

**Piotr Tatjewski**

**Sterowanie zaawansowane  
obiektów przemysłowych  
Struktury i algorytmy**

**Wydanie drugie zmienione**

Akademicka Oficyna Wydawnicza **EXIT**  
Warszawa 2016

© Copyright by Piotr Tatjewski  
Warszawa 2016

© Copyright by Akademicka Oficyna Wydawnicza EXIT  
Warszawa 2016

**Opiniodawcy pierwszego wydania:**

prof. dr inż. Władysław Findeisen  
prof. dr hab. inż. Józef Korbicz

Wszystkie nazwy produktów są zastrzeżonymi nazwami handlowymi lub znakami towarowymi odpowiednich firm.

Powyższej monografii w całości lub części nie wolno powielać ani przekazywać w żaden sposób, nawet za pomocą nośników mechanicznych i elektronicznych (np. zapis magnetyczny), w tym też umieszczać ani rozpowszechniać w postaci cyfrowej zarówno w Internecie, jak i w sieciach lokalnych, bez uzyskania pisemnej zgody Akademickiej Oficyny Wydawniczej EXIT.

**Skład komputerowy**

Piotr Tatjewski

**Projekt okładki**

Jarosław Skowroński

**Wydawca**

Akademicka Oficyna Wydawnicza EXIT  
www.exit.pl, biuro@exit.pl

**ISBN 978-83-7837-570-8**

*Pamięci Mamy*



## Przedmowa do wydania drugiego

Pierwsze wydanie niniejszej monografii ukazało się w 2002 roku, jako tom 5 w serii „Monografie Komitetu Automatyki i Robotyki Polskiej Akademii Nauk”. Od tamtej chwili minęło już 14 lat, stąd prezentowane obecnie czytelnikowi wydanie drugie jest istotnie zmienione, uaktualnione i w wielu fragmentach znacznie rozszerzone. Bardziej odpowiada aktualnemu stanowi naszej wiedzy w omawianej dziedzinie, w tym przedstawia również nowe rezultaty uzyskane przez autora.

Rozdział pierwszy, stanowiący wprowadzenie w problematykę sterowania warstwowego procesami przemysłowymi, pozostał w istocie niezmienniony. W rozdziale drugim poświęconym regulacji rozmytej zmian jest niewiele, dodano jedynie nowe, nieco silniejsze rezultaty dotyczące warunków stabilności układów sterowania z regulatorami rozmytymi. Istotne zmiany i wiele rozszerzeń wprowadzono w najobszerniejszym rozdziale trzecim pracy, poświęconym regulacji predykcyjnej. Poprawiono i nieco rozszerzono część dotyczącą regulatora GPC. Gruntownie przeformułowano i znacząco rozszerzono, a w istocie napisano od nowa, część dotyczącą regulacji predykcyjnej z modelem procesu w postaci liniowych równań stanu. Z najważniejszych zmian w tej części warto wymienić uwzględnienie wprowadzonego przez autora w pracy [144] nowatorskiego deterministycznego modelu zakłóceń niemierzalnych i dodanie nowych rezultatów (w tym autora) w napisanej od nowa części dotyczącej estymacji stanu na potrzeby regulatora predykcyjnego. Znacząco rozszerzono część dotyczącą regulacji predykcyjnej z nieliniowymi modelami procesu, w szczególności dotyczącą regulacji z optymalizacją nieliniową i regulatorów z nieliniowymi modelami w postaci rozmytych równań stanu. Istotnie zmieniono rozdział czwarty omawiający optymalizację punktu pracy procesu sterowanego, w pierwszej części wprowadzając nowe zagadnienia dotyczące bieżącego dostrajania punktu pracy we współdziałaniu z regulatorem predykcyjnym (tzw. *steady state target optimization*). Natomiast w dalszej części zmieniono i ograniczono przedstawienie zagadnień dotyczących bieżącej optymalizacji stanu ustalonego procesu.

Dalsza część niniejszej Przedmowy zawiera tekst Przedmowy do wydania pierwszego, z uaktualnieniami wynikającymi z wymienionych powyżej zmian i rozszerzeń.

Przedmiotem rozważań przedstawionych w książce jest *sterowanie zaawansowane* obiektów przemysłowych, rozumiane jako wzajemnie uzależnione, w odpowiednich strukturach, algorytmy regulacji zaawansowanej i optymalizacji bieżącej. Punktem wyjścia dla tak określonej problematyki jest *struktura warstwowa* sterowania procesami przemysłowymi. Od wielu lat obiekty techniczne, szczególnie większej czy dużej skali, sterowane są w takiej strukturze. Umożliwia ona bowiem racjonalne, bezpieczne sterowanie i zarządzanie, poprzez podział zadań i odpowiedzialności na dobrze postawione zadania cząstkowe o bardziej jednorodnym charakterze i ograniczonej skali, wzajemnie między sobą powiązane. Podstawowe warstwy takiej struktury to warstwa regulacji i zabezpieczeń oraz bezpośrednio nadrzędna wobec niej warstwa optymalizacji [43, 90, 42, 54, 87, 15, 144, 145].

Zadaniem warstwy regulacji i zabezpieczeń jest utrzymywanie obiektu w określonych stanach wymuszanych przede wszystkim przez wartości zadane dla regulatorów, ponadto reagowanie na stany niebezpieczne czy awaryjne poprzez odpowiednio zaprojektowany system blokad i zabezpieczeń. Rozwój elektroniki i techniki komputerowej doprowadził w ostatnich dekadach do powstania potężnych systemów zbierania, wizualizacji i przetwarzania danych (systemy SCADA – *Supervisory Control and Data Acquisition*), które w połączeniu z potencjałem obliczeniowym współczesnych sterowników komputerowych umożliwiają bardzo zaawansowaną realizację podstawowych zadań warstwy regulacji i zabezpieczeń, jak i rozszerzenie ich zakresu. W szczególności, możliwa jest realizacja w czasie rzeczywistym tzw. zaawansowanych technik regulacji (*advanced control techniques*), za które uważa się algorytmy bardziej złożone niż klasyczne algorytmy typu PID [87, 9, 10]. Techniki te adresowane są do obiektów opisanych modelami liniowymi, ale trudnych dla klasycznej regulacji np. ze względu na duże opóźnienia, ograniczenia sygnałów czy wielowymiarowość, jak też do regulacji obiektów nieliniowych i opisanych modelami nieliniowymi.

Zadaniem warstwy optymalizacji jest ustalanie najlepszych z ekonomicznego punktu widzenia wartości zadanych dla regulatorów warstwy regulacji, czyli optymalnych punktów pracy obiektu sterowanego. W obliczu rosnących mocy obliczeniowych realna stała się możliwość optymalizacji powtarzanej na bieżąco (*on-line*), czyli automatycznego czy półautomatycznego dopasowywania optymalnego punktu pracy do zmieniających się warunków zewnętrznych. Co więcej, firmy, które pierwsze wprowadziły systemy optymalizacji bieżącej, wysunęły się na czoło walki konkurencyjnej,

wyznaczając nowe standardy. Wyliczane na bieżąco w warstwie optymalizacji punkty pracy wymagają od regulatorów poprawnego działania nie tylko w jednym takim punkcie, ale w szerszym zakresie zmienności wielkości wejściowych i wyjściowych obiektu. Realne obiekty są z reguły nieliniowe, stąd potrzeba nieliniowych lub adaptacyjnych układów regulacji. Co więcej, optymalizacja przynosi tym większe zyski, im dokładniejszym modelem obiektu dysponuje optymalizator i im precyzyjniej ustawiane i utrzymywane są optymalne wartości wielkości regulowanych. Dlatego optymalizacja postawiła nowe wymagania przed układami regulacji, stwarzając potrzebę rozwoju algorytmów lepszych niż powszechnie stosowana technika PID, potrzebę rozwoju zaawansowanych technik regulacji (w sensie zdefiniowanym powyżej).

Tematyka książki mieści się w głównym nurcie przedstawionych wyzwań. W rozdziale pierwszym przedstawiono ogólne zagadnienia związane ze sterowaniem w strukturze warstwowej. Wychodząc od podstawowego celu sterowania omówiono zagadnienie dekompozycji prowadzące do takiej struktury, przedstawiono podstawowe zadania i cechy realizacyjne warstw regulacji bezpośredniej i nadrzędnej oraz optymalizacji, wspomniano o warstwach wyższych.

Rozdział drugi poświęcono algorytmom regulacji nieliniowej wykorzystującej struktury rozmyte (*fuzzy*) typu Takagi-Sugeno (TS). Po krótkim, ogólnym wprowadzeniu do logiki rozmytej i modelowania nieliniowego rozmytego typu TS omówiono zagadnienia projektowania i analizy stabilności nieliniowych algorytmów regulacji rozmytych typu TS. Zdaniem nie tylko autora jest to technika skuteczna, stosunkowo łatwa w projektowaniu i przekonywająca, gdyż może być traktowana jako naturalne nieliniowe uogólnienie znanych liniowych algorytmów regulacji. W szczególności, jest to uogólnienie ugruntowanej techniki PID w postaci nieliniowych rozmytych regulatorów PID, kiedy jednym z zasadniczych etapów projektowania jest dobór nastaw liniowych regulatorów PI czy PID w odpowiednio wybranym zestawie punktów pracy obiektu. Stanowiąc pełną alternatywę projektową, nieliniowe rozmyte regulatory TS są jednocześnie dobrą propozycją tam, gdzie potrzeba przejść od algorytmów liniowych do nielinowych, nie tracąc przy tym zgromadzonych doświadczeń.

Rozdział trzeci poświęcono predykcyjnym algorytmom regulacji, określanym zwięźle powszechnie używanym akronimem MPC (*Model-based Predictive Control*). Regulacja MPC jest jedyną z zaawansowanych technik regulacji, która odniosła olbrzymi, niekwestionowany sukces w aplikacjach praktycznych wywierając w ostatnich latach dominujący wpływ zarówno na kierunek rozwoju przemysłowych układów regulacji i sterowania, jak

i prac badawczych w tym obszarze. Z kilku powodów. Algorytmy regulacji predykcyjnej MPC są w istocie pierwszą techniką w sposób bezpośredni uwzględniającą ograniczenia zarówno sygnałów sterujących, jak i wielkości wyjściowych. Przy tym generują sterowania uwzględniając również, dzięki bezpośredniemu wykorzystaniu modelu, interakcje wewnętrzne w obiekcie. Stąd są skuteczne w zastosowaniu do obiektów wielowymiarowych, również przy nierównej liczbie wejść sterujących i wielkości regulowanych. Ponadto są to algorytmy, których zasada działania i dobór nastaw są zrozumiałe, stosunkowo łatwe do wyjaśnienia personelowi inżynierskiemu, jak i operatorskiemu – aspekt bardzo istotny przy wprowadzaniu nowych technik do praktyki przemysłowej. Ze względu na większe wymagania obliczeniowe, w początkowym okresie algorytmy MPC znalazły zastosowanie przede wszystkim w regulacji nadrzędnej, gdzie typowe są dłuższe okresy próbkowania, a jednocześnie kluczowe są kwestie ograniczeń i wielowymiarowości. Wraz ze zwiększaniem mocy obliczeniowych i niezawodności oraz spadkiem cen procesorów możliwe stało się również wprowadzanie algorytmów predykcyjnych do regulacji bezpośredniej, do podstawowych pętli regulacyjnych.

W rozdziale trzecim przedstawiono najpierw liniowe algorytmy predykcyjne, rozpoczynając od najważniejszego dla praktyki przemysłowej sformułowania DMC (*Dynamic Matrix Control*), następnie przechodząc do algorytmu GPC (*Generalized Predictive Control*). Wyczerpująco przedstawiono algorytmy predykcyjne z modelem w postaci liniowych równań stanu (algorytmy MPC), ta część została w istocie napisana od nowa i jest znacznie obszerniejsza niż w pierwszym wydaniu książki. Przedstawiając algorytmy z modelami liniowymi, omówiono zarówno wersje analityczne (prawa regulacji), jak i numeryczne, kiedy w każdej chwili próbkowania rozwiązywane jest numerycznie zadanie programowania kwadratowego. Wychodząc od sformułowań liniowych, przedstawiono następnie podstawowe struktury nieliniowych algorytmów MPC, zwracając szczególną uwagę na kluczowe dla efektywnych zastosowań wersje z linearyzacjami. Pokazano, że wersje te szczególnie łatwo implementować dla modeli obiektów rozmytych typu Takagi-Sugeno. Omówiono zagadnienia stabilności algorytmów MPC oraz kwestie interpretacji i doboru nastaw. Rozdział trzeci stanowi w niniejszym wydaniu drugim najobszerniejszą część monografii, ponad jej połowę.

Rozdział czwarty i ostatni poświęcony jest algorytmom optymalizacji punktu pracy procesu sterowanego. W porównaniu do analogicznego rozdziału w pierwszym wydaniu, zmiany są tu bardzo duże. Pierwsza część rozdziału, omawiająca zagadnienia dotyczące szybkiego bieżącego dostrajania punktu pracy we współdziałaniu z regulatorem predykcyjnym (tzw. *steady state target optimization*), jest całkowicie nowa. Natomiast pozostała część

tego rozdziału przedstawiająca zagadnienia dotyczące algorytmów bieżącej optymalizacji stanu ustalonego obiektu regulowanego, w warunkach istotnej niepewności powodowanej niedokładnym modelem i błędami estymacji zakłóceń, została nieco zmieniona i skrócona.

W książce przedstawiono, przede wszystkim w rozdziałach drugim i centralnym rozdziale trzecim, ważne aktualnie i dobrze ugruntowane zaawansowane algorytmy regulacji, uzupełnione oryginalnymi wynikami i spojrzeniem autora. Dlatego też książka adresowana jest zarówno do czytelników zainteresowanych poznaniem mechanizmów prezentowanych technik zaawansowanego sterowania, w tym inżynierów praktyków, jak i do studentów wyższych lat studiów magisterskich, doktorantów i pracowników naukowych. Od czytelnika wymaga się znajomości podstaw automatyki, przede wszystkim w zakresie ciągłych i dyskretnych algorytmów regulacji, oraz wiedzy z zakresu podstaw optymalizacji. Potrzebne jest też odpowiadające tym wymaganiom wykształcenie matematyczne.

Dla wygody czytelnika oraz dla jednoznaczności odniesień do literatury anglojęzycznej, przy wprowadzaniu terminów polskich podaje się z reguły ich odpowiedniki w języku angielskim.

Na końcu pracy zamieszczono skorowidz przyjmując zasadę, że podane odniesienia dotyczą przede wszystkim numerów tych stron w tekście książki, gdzie dany termin jest definiowany lub omawiany.

Na końcu Przedmowy do pierwszego wydania tej monografii autor zamieścił szereg podziękowań osobom i instytucjom, które przyczyniły się do powstania pierwszego wydania. Podziękowania nie są tu powtórzone, mimo iż nie straciły na znaczeniu. Autor chciałby jedynie przekazać wyrazy wdzięczności Panu Profesorowi Władysławowi Findeisenowi, swemu wieloletniemu nauczycielowi i mistrzowi oraz Panu Profesorowi Krzysztofowi Malinowskiemu, od wielu lat koledze i przyjacielowi w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. Ponadto, autor pragnie podziękować za wieloletnią współpracę swym uczniom i najbliższym współpracownikom w kierowanym przez siebie Zakładzie Automatyki i Inżynierii Oprogramowania.

Na koniec autor chciałby przeprosić tych czytelników, którzy nadal preferują tradycyjną drukowaną postać książek, za dostępność niniejszego drugiego, zmienionego wydania tej monografii jedynie w postaci elektronicznej.



# Spis treści

<b>Przedmowa do wydania drugiego</b> . . . . .	V
<b>Rozdział 1. Warstwowa struktura sterowania</b> . . . . .	1
1.1. System sterowania . . . . .	1
1.2. Cele sterowania . . . . .	2
1.3. Warstwy sterowania . . . . .	5
1.4. Modelowanie obiektu w strukturze warstwowej . . . . .	10
Przykład 1.1 . . . . .	16
1.5. Warstwa optymalizacji . . . . .	26
Przykład 1.2 . . . . .	29
1.6. Nadzór, diagnostyka, adaptacja . . . . .	30
1.7. Wnioski . . . . .	33
<b>Rozdział 2. Nieliniowa regulacja rozmyta</b> . . . . .	35
2.1. Systemy rozmyte typu TS . . . . .	38
2.1.1. Zbiory rozmyte i zmienne lingwistyczne . . . . .	38
2.1.2. Wnioskowanie rozmyte . . . . .	42
2.1.3. Projektowanie modeli rozmytych TS . . . . .	50
2.1.4. System TS jako rozmyta sieć neuronowa . . . . .	51
2.2. Dyskretne rozmyte algorytmy regulacji typu TS . . . . .	58
2.2.1. Regulator rozmyty TS ze sprzężeniem od stanu . . . . .	62
2.2.2. Regulatory rozmyte TS ze sprzężeniem od wyjścia . . . . .	76
2.3. Ciągłe rozmyte algorytmy regulacji typu TS . . . . .	90
2.4. Kompensacja zakłóceń, strojenie automatyczne . . . . .	110
<b>Rozdział 3. Regulacja predykcyjna</b> . . . . .	115
3.1. Zasada regulacji predykcyjnej . . . . .	117
3.2. Algorytm DMC (Dynamic Matrix Control) . . . . .	130
3.2.1. Predykcja wyjść modelem odpowiedzi skokowych . . . . .	130
3.2.2. Algorytm DMC w wersji analitycznej . . . . .	136
3.2.3. Uwzględnianie ograniczeń, algorytm w wersji numerycznej . . . . .	146
3.2.4. Niepewność modelu, zakłócenia . . . . .	155
3.3. Algorytm GPC (Generalized Predictive Control) . . . . .	162
3.3.1. Algorytm GPC dla obiektu jednowymiarowego . . . . .	165
3.3.2. Przypadek zakłócenia typu DMC . . . . .	182
3.3.3. Algorytm GPC dla obiektu wielowymiarowego . . . . .	185
3.3.4. Wersja numeryczna algorytmu GPC . . . . .	187

3.4.	Algorytmy MPCs – z modelem w postaci liniowych równań stanu .	195
3.4.1.	Algorytm MPCs z pomiarem wektora stanu . . . . .	196
3.4.2.	Algorytmy MPCs z obserwatorami stanu . . . . .	207
3.5.	Algorytmy nieliniowe regulacji predykcyjnej . . . . .	225
3.5.1.	Struktury nieliniowych algorytmów MPC . . . . .	225
3.5.2.	Algorytmy MPC-NPL (z nieliniową predykcją i linearyzacją)	237
3.5.3.	Algorytmy FMPC (Fuzzy MPC) w wersji numerycznej . . .	254
3.5.4.	Algorytmy FMPC (Fuzzy MPC) w wersji analitycznej . . .	270
3.6.	Stabilność, ograniczenia, dobór parametrów . . . . .	278
3.6.1.	Stabilność algorytmów MPC . . . . .	278
3.6.2.	Dopuszczalność ograniczeń, dobór parametrów . . . . .	293
<b>Rozdział 4. Optymalizacja bieżąca punktu pracy . . . . .</b>		<b>305</b>
4.1.	Miejsce optymalizacji punktu pracy w strukturze warstwowej . . . .	305
4.2.	Dostrajanie punktu pracy dla regulatorów MPC (zadanie SSTO) .	308
4.2.1.	Zadania SSTO dla regulacji MPC z modelami liniowymi . .	309
4.2.2.	Zadania SSTO dla regulacji MPC z modelami nieliniowymi	312
4.3.	Iteracyjna optymalizacja punktu pracy w warunkach niepewności .	313
4.3.1.	Modelowanie obiektu regulowanego i zadanie optymalizacji w stanach ustalonych . . . . .	313
4.3.2.	Algorytm powtarzanej optymalizacji z korektą estymat zakłóceń . . . . .	318
4.3.3.	Metoda powtarzanej modyfikowanej optymalizacji (PMO) .	320
4.3.4.	Algorytm dualny PMO (algorytm DPMO) . . . . .	329
4.3.5.	Przykłady symulacji sterowania . . . . .	337
4.4.	Iteracyjna optymalizacja punktu pracy w strukturze z wyjściami ograniczonymi regulacyjnie . . . . .	354
4.4.1.	Algorytm powtarzanej optymalizacji z korektą estymat zakłóceń . . . . .	357
4.4.2.	Metoda powtarzanej modyfikowanej optymalizacji (PMO) .	359
4.5.	Procedury optymalizacji modeli nieliniowych . . . . .	360
<b>Podsumowanie . . . . .</b>		<b>363</b>
<b>Bibliografia . . . . .</b>		<b>369</b>
<b>Indeks . . . . .</b>		<b>379</b>
<b>Abstract . . . . .</b>		<b>387</b>

## Rozdział 1

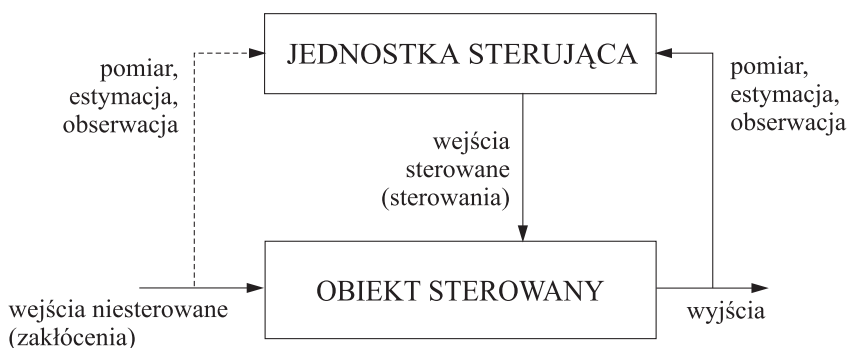
# Warstwowa struktura sterowania

### 1.1. System sterowania

*Sterowanie* obiektem można ogólnie zdefiniować jako wpływanie na obiekt w taki sposób, aby powodować jego działanie (zachowanie się) zgodnie z założonymi wymaganiami, zob. np. [42]. Definicja ta dotyczy wszelkich sytuacji i obiektów podlegających tak określone sterowaniu. Mówimy więc o sterowaniu obiektów technicznych, takich jak samoloty czy generatory prądu, procesów technologicznych zachodzących w reaktorach chemicznych, kolumnach destylacyjnych czy oczyszczalniach ścieków, obiektów ekonomicznych, takich jak firma, koncern czy wręcz cała gałąź gospodarki (tu zamiast słowa sterowanie używane jest na ogół określenie zarządzanie), obiektów przekazu i wymiany informacji (informacyjnych), takich jak sieć telekomunikacyjna czy sieć komputerowa itd.

Obiekt podlegający sterowaniu (obiekt sterowany) jest zawsze pewną wyodrębnioną częścią środowiska, w którym występuje, podlegając kontrolowanym bądź niekontrolowanym przez jednostkę sterującą (komputer sterujący, człowieka itp.) wpływom otoczenia. Dla przykładu, na zachowanie samolotu oddziałuje zarówno pilot, w celu wymuszenia kierunku, wysokości i innych parametrów lotu, jak i kierunek i siła wiatru czy prądy powietrzne będące oddziaływaniami nie podlegającymi kontroli i zaburzającymi wymuszane przez pilota parametry lotu. Podobnie komputer sterujący czy też operator stara się wymuszać pożądane parametry procesu technologicznego w reaktorze chemicznym czy kolumnie destylacyjnej poprzez odpowiednie ustawianie wartości tych zmiennych, na które ma wpływ (poziomów, przepływów, temperatur, ciśnień itp.) – w obecności oddziaływań otoczenia obiektu zakłócających pożądany bieg procesu, takich jak niesterowane wahania parametrów strumieni surowców czy mediów energetycznych (składu surowca, ciśnienia pary grzewczej, temperatury wody chłodzącej itp.). Przykłady można by tu dalej przytaczać, również w zakresie obiektów natury ekonomicznej czy informacyjnej. Istotna cecha wspólna, na którą zwracamy w tych przykładach uwagę, to fakt występowania sterowanych i niesterowanych oddziaływań zewnętrznych na wyodrębniony z otoczenia obiekt ste-

rowania – mówimy o *sterowanych i niesterowanych wielkościach wejściowych* (krótko: *wejściach*) *obiektu*. Wejścia niesterowane nazywane są często *wejściami swobodnymi* lub *zakłócającymi* (krótko: *zakłóceniami*), gdyż zakłócają one pożądane zachowanie obiektu. Z kolei ocena stanu obiektu sterowanego, tego czy spełnia on założone wymagania, czy właściwe jest oddziaływanie wejść sterujących, dokonywana jest na podstawie pomiarów, a ogólniej obserwacji wartości i cech odpowiednich wielkości charakteryzujących obiekt sterowany. Nazywamy je *wielkościami wyjściowymi* (krótko: *wyjściami*) *obiektu*. W przypadku sterowania samolotem wyjściami będą parametry pozycji i prędkości. W przypadku sterowania reaktorem chemicznym czy kolumną destylacyjną przykładami wielkości wyjściowych są parametry mieszaniny reagującej czy destylowanej, takie jak temperatura czy skład, jak również parametry charakteryzujące stan aparatów technologicznych (zapełnienia, poziomy itp.). Znając cele sterowania, na podstawie analizy wartości wyjść obiektu oraz tych wejść niesterowanych, które są znane (mierzone, estymowane), jednostka sterująca podejmuje decyzje dotyczące utrzymania bądź odpowiedniej zmiany wartości wejść sterowanych (sterowań). Omówiona ogólna struktura systemu sterowania przedstawiona jest na rysunku 1.1.



Rysunek 1.1. Ogólna struktura systemu sterowania

## 1.2. Cele sterowania

Cele sterowania mogą być różnej natury. Dla przykładu, celem sterowania samolotem pasażerskim jest przelot do określonego lotniska zgodnie z założoną trajektorią lotu. Pierwotne, podstawowe cele sterowania procesami technologicznymi czy gospodarczymi w gospodarce rynkowej są natury

ekonomicznej – osiągnięcie zysku z prowadzonej działalności produkcyjnej, handlowej czy produkcyjno-handlowej. Podobnie, pierwotny cel sterowania siecią telekomunikacyjną stacjonarną czy bezprzewodową jest natury ekonomicznej – długookresowy zysk operatora. Jednakże, aby efektywnie realizować podstawowy cel ekonomiczny, trzeba z reguły zapewniać realizację szeregu celów częściowych, warunkujących możliwość przede wszystkim odpowiednio bezpiecznej realizacji celu podstawowego, a ponadto gwarantujących wymagane parametry jakościowe oferowanych produktów czy usług – i to przy braku czy niepełnej informacji o zachowaniu otoczenia obiektu sterowanego. Ponadto, wiele obiektów sterowania jest złożonej natury, z wieloma wejściami sterowanymi i niesterowanymi oraz wieloma wielkościami wyjściowymi, o skomplikowanej naturze wzajemnych oddziaływań między wejściami i wyjściami. Skomplikowanym obiektem może już być nawet pojedynczy reaktor czy kolumna destylacyjna, bardzo złożonymi obiektami sterowania są linie technologiczne składające się z kilku obiektów technologicznych wzajemnie na siebie oddziałujących. Sterowanie centralne automatyczne takimi złożonymi obiektami, aczkolwiek współcześnie nierzadko teoretycznie możliwe, jest bardzo trudne i charakteryzuje się szeregiem wad praktycznie takie rozwiązania eliminujących, zob. np. [90, 42]. Najważniejsze z nich to trudność zapewnienia odpowiedniego bezpieczeństwa przebiegu procesu sterowanego, trudność niezbędnego włączenia ludzi w proces nadzoru i reakcji na zjawiska niekontrolowane i nieprzewidziane, związana z koniecznością jednoczesnego i szybkiego przetwarzania dużych ilości informacji. Stąd w sterowaniu (jak i zarządzaniu) złożonymi obiektami od lat ukształtowała się praktyka podejścia *hierarchicznego*, a w szczególności warstwowego, praktyka potwierdzona przez teorię, zob. np. [80, 43, 42]. Istotą podejścia hierarchicznego jest *dekompozycja* pierwotnego, podstawowego zadania (celu) sterowania na szereg zadań cząstkowych, mniej złożonych i wzajemnie ze sobą powiązanych, z których każde związane jest z przetwarzaniem mniejszej ilości informacji i realizacją na ogół jedynie celu cząstkowego.

Spotykamy dwa podstawowe sposoby dekompozycji zadania sterowania [42]:

- dekompozycję funkcjonalną,
- dekompozycję przestrzenną.

Pierwsza z nich dotyczy obiektu traktowanego jako całość i polega na wydzieleniu szeregu funkcjonalnie różnych cząstkowych celów sterowania wzajemnie ze sobą powiązanych – w strukturze pionowej, hierarchicznej współzależności zwanej *strukturą warstwową*. Jednostka decyzyjna związana z każdą z warstw podejmuje decyzje odnoszące się do tego samego

obiektu, ale każda z nich podejmuje decyzje różnego rodzaju. Dekompozycja przestrzenna związana jest z kolei z przestrzenną strukturą złożonego obiektu sterowanego i polega na podziale zadania sterowania (czy zadania funkcjonalnie cząstkowego – np. w ramach jednej warstwy wspomnianej struktury warstwowej) na lokalne podzadania funkcjonalnie tego samego rodzaju – jedynie o mniejszej wymiarowości, mniejszej ilości przetwarzanej informacji, powiązane z poszczególnymi przestrzennie wydzielonymi częściami całego złożonego obiektu sterowania. Postępowanie to prowadzi do tzw. *struktur wielopoziomowych* [40, 44, 43, 139, 42]. Przedmiotem zainteresowania niniejszej pracy są *warstwowe struktury sterowania obiektami przemysłowymi*.

Struktura warstwowa sterowania jest efektem dekompozycji funkcjonalnej podstawowego, całościowego zadania (celu) sterowania. Realizacja podstawowego, ekonomicznego celu sterowania obiektem przemysłowym (technologicznym) wiąże się nierozzerwalnie z realizacją pewnych celów szczegółowych, z których najważniejsze to:

1. Zapewnienie bezpiecznego przebiegu procesów w obiekcie sterowanym, tzn. ograniczenie do akceptowalnego poziomu możliwości wystąpienia awarii i zjawisk niekontrolowanych,
2. Zapewnienie odpowiednich cech wyjść obiektu (np. jakości produktów), tj. utrzymywanie zmiennych wyjściowych w obszarze dopuszczalnych wartości,
3. Optymalizacja bieżącej efektywności działania obiektu, tj. z reguły maksymalizacja zysku przy ograniczeniach surowcowych czy energetycznych lub minimalizacja kosztów (surowcowych, energetycznych) przy założonym poziomie produkcji.

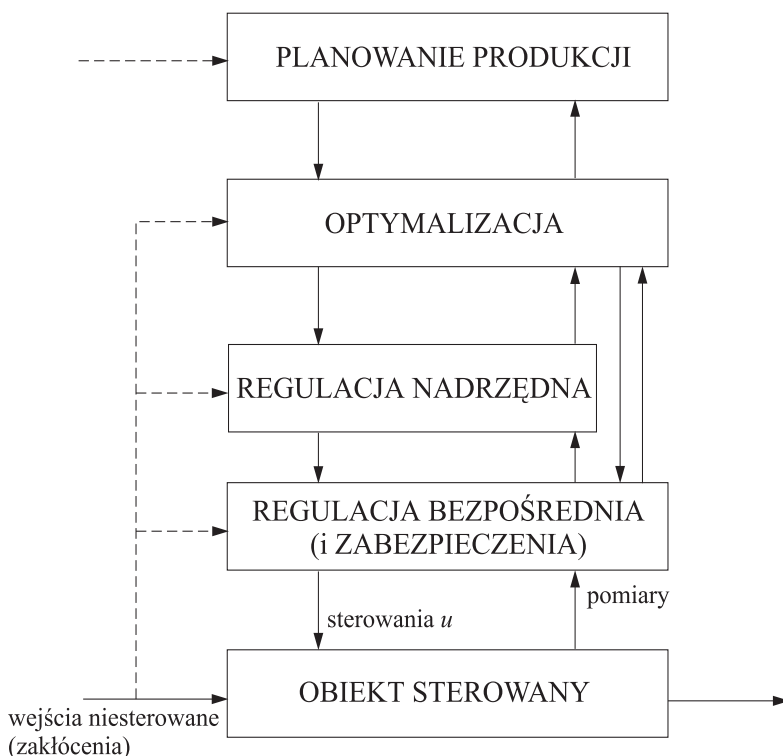
Nietrudno zauważyć, że pierwsze dwa z wymienionych celów też mają podłoże natury ekonomicznej, związane są z podstawowym efektywnościowym celem sterowania obiektem. Wystąpienie awarii prowadzi bowiem z reguły do bardzo poważnych strat związanych z jej usuwaniem i przerwą w produkcji, strat bezpośrednich i pośrednich (niedotrzymanie terminów, utrata odbiorców). Są to straty z reguły większe niż straty wynikające z nieoptymalnego, ale bezpiecznego przebiegu produkcji. Z kolei niedotrzymanie parametrów jakościowych prowadzi w najlepszym przypadku do utraty części zysku z powodu konieczności obniżenia ceny, może też prowadzić do utraty produktu (konieczności wyrzucenia czy powtórnego przerobienia), jeśli produkt nie spełniający wymaganych norm nie znajduje nabywcy lub jest nieprzydatny do dalszego przerobu. I tutaj straty są na ogół większe niż w przypadku nieoptymalnego, ale gwarantującego spełnianie wymagań

jakościowych prowadzenia procesu. Dodajmy, że z reguły im wyższa jakość produktu (np. czystość w procesie destylacji), tym wyższe koszty. Wówczas opłaca się prowadzić proces możliwie blisko granicy wymagań jakościowych – ale wymaga to precyzyjniejszych układów sterowania i może być bardziej ryzykowne wobec zawsze występującej niepewności związanej z wpływem zakłóceń.

### 1.3. Warstwy sterowania

Kolejność wymienienia przedstawionych w rozdziale poprzednim trzech najważniejszych szczegółowych, cząstkowych celów sterowania obiektem przemysłowym nie jest przypadkowa. Najważniejsze jest bezpieczeństwo sterowania obiektem, w następnej kolejności trzeba dbać o jakość produktów, a dopiero w zakresie swobody pozostawionej po zapewnieniu realizacji tych dwóch celów jest miejsce na bieżącą ekonomiczną optymalizację wartości zmiennych określających przebieg procesów w obiekcie. W takiej też kolejności umieszczone są ponad usytuowanym na samym dole *obiektem sterowania*, realizujące kolejno wymienione cele warstwy *podstawowej warstwowej struktury sterowania* przedstawionej na rysunku 1.2.

Warstwa *regulacji bezpośredniej* (i zabezpieczeń) odpowiada za bezpieczeństwo przebiegu procesów dynamicznych w obiekcie. Tylko ta warstwa ma bezpośredni dostęp do obiektu sterowanego, ma możliwość bezpośredniego wpływania na wielkości wejściowe sterujące (zwane też wielkościami manipulowanymi) oznaczone na rysunku przez  $u$ . Realizacja techniczna zadań warstwy regulacji i zabezpieczeń zapewniana jest wspólnie, dla większych obiektów, przez tzw. *rozproszone systemy sterowania*, popularne w przemyśle systemy typu DCS (*Distributed Control System*). Są to kompleksowe komputerowe systemy pomiarów, regulacji i zabezpieczeń, wyposażone z reguły we własne, zintegrowane pakiety oprogramowania typu SCADA (*Supervisory Control and Data Acquisition*) do wizualizacji, operatorskiego i inżynierskiego nadzoru oraz archiwizacji danych. Jednym z pierwszych był opracowany w latach siedemdziesiątych ubiegłego stulecia system TDC 2000 firmy Honeywell, wszystkie liczące się firmy oferują obecnie systemy tej klasy. Zadania regulacji i zabezpieczeń dla mniejszych obiektów, czy fragmentów większych obiektów, realizować można z wykorzystaniem sterowników programowanych (PLC – *Programmable Logic Controllers*) czy sterowników automatyki (PAC – *Programmable Automation Controllers*), czy też tzw. regulatorów wielofunkcyjnych. Granica między tymi klasami sprzętu jest nieostra – bardziej rozbudowane wersje sterowników programowanych umożliwiają realizację nawet wielu pętli regulacyj-



Rysunek 1.2. Warstwowa struktura sterowania obiektem przemysłowym

nych, z kolei współczesne regulatory wielofunkcyjne zapewniają możliwość realizacji wielu funkcji logicznych, zob. np. [156]. Nowszym, wyraźnie zaznaczającym się trendem jest zastosowanie do zadań regulacji komputerów klasy PC, często w wykonaniu przemysłowym, z reguły ze specjalizowanymi kartami i oprogramowaniem umożliwiającym realizację sterowań w czasie rzeczywistym.

Algorytmy regulacji bezpośredniej powinny być niezawodne i stosunkowo proste, stąd nadal dominują tu klasyczne algorytmy PID. Jednakże możliwości systemów DCS, współczesnych sterowników PLC czy PAC, czy komputerów klasy PC, są dużo większe. Stąd, tam gdzie klasyczna regulacja PID nie spełnia wymagań, coraz częściej stosowane są bardziej zaawansowane algorytmy regulacji, przede wszystkim odpowiednie modyfikacje algorytmu PID oraz, szczególnie w ostatnim okresie, proste realizacje regulatorów predykcyjnych. Wymienić tu można struktury regulacji PID z bezpośrednią korekcją od wielkości zakłócającej czy wartości zadanej, ze zmiennym wzmocnieniem uzależnionym od wybranej wielkości procesowej–

tw. *gain scheduling*, regulatory PI o strukturze Smitha dla pętli z dużymi opóźnieniami, nieliniowe regulatory PI czy PID typu rozmytego (*fuzzy logic controllers*), proste algorytmy predykcyjne, algorytmy adaptacyjne.

W opracowaniach, materiałach informacyjnych firm z branży automatyki spotyka się dość powszechnie określenie *regulacja podstawowa* (emphasis regulatory control, direct control), w odróżnieniu od określenia *regulacja zaawansowana* (*advanced control*). Należy tu jednak zdecydowanie podkreślić, że podstawową cechą wyróżniającą wszelkie algorytmy regulacji bezpośredniej jest *bezpośredni dostęp do obiektu sterowanego i wysoka częstotliwość interwencji* (mały okres próbkowania), a nie rodzaj zastosowanego algorytmu.

Zmiennymi wyjściowymi regulatorów wchodzących w skład *warstwy regulacji nadrzędnej* nie są bezpośrednio wielkości wejściowe sterujące (manipulowane) obiektu, ale wartości zadane dla regulatorów warstwy regulacji bezpośredniej. Ponadto, w pętlach regulacji nadrzędnej *częstotliwość interwencji jest mniejsza*, tzn. większe są okresy próbkowania – rzędu np. minuty, podczas gdy w warstwie regulacji bezpośredniej są to ułamki sekundy, co najwyżej pojedyncze sekundy.

Celem regulacji nadrzędnej jest kontrolowanie wolniej zmiennych wielkości procesowych *decydujących o jakości, o parametrach produkcji*, jak np. stężenia w reaktorach czy kolumnach destylacyjnych. Dla przykładu, dobra stabilizacja (charakteryzująca się małą wariancją uchybu regulacji) stężenia kluczowego zanieczyszczenia w strumieniu produktu destylacji pozwala prowadzić proces w punkcie pracy leżącym bliżej maksymalnej dopuszczalnej wartości tego zanieczyszczenia – wytwarzamy produkt nieco bardziej zanieczyszczony, a przez to taniej. Dlatego też od algorytmów regulacji nadrzędnej wymaga się z reguły wysokiej jakości działania, najczęściej w sytuacji procesu wielowymiarowego, nieliniowego. Najbardziej typowe, nowoczesne rozwiązania to przede wszystkim algorytmy opartej na modelach procesu regulacji predykcyjnej z przesuwającym horyzontem (algorytmy MPC – *Model Predictive Control*). Jednym z najbardziej znanych przykładów są tu aplikacje oparte na wykorzystaniu algorytmu DMC (*Dynamic Matrix Control*), powstałego w przemyśle petrochemicznym w latach siedemdziesiątych ubiegłego stulecia.

Historia oraz znaczenie warstwy regulacji nadrzędnej bezpośrednio związane są z rozwojem algorytmów zaawansowanej regulacji, w tym szczególnie z aplikacjami algorytmów predykcyjnych. Upřednio nie rozróżniano bowiem w literaturze przedmiotu warstw regulacji bezpośredniej i nadrzędnej, zob. np. [80, 43]. Dopiero rozwój techniki komputerowej, umożliwiając realizację w czasie rzeczywistym bardziej wymagających obliczeniowo, za-

awansowanych algorytmów opartych na modelach procesu, takich jak algorytm DMC i inne algorytmy predykcyjne, spowodował wyraźne wyodrębnienie warstwy regulacji nadrzędnej. Od tego czasu wyróżnienie tej warstwy jest powszechnie spotykane w materiałach, opracowaniach wielu czołowych firm produkujących sprzęt i oprogramowanie do celów sterowania, a także w podstawowych podręcznikach, szczególnie poświęconych sterowaniu procesami technologicznymi (*process control*), zob. np. [90, 54, 87, 144, 145]. Trzeba jednakże zaznaczyć, że warstwa regulacji nadrzędnej nie zawsze występuje. Nie trzeba jej wyodrębniać tam, gdzie nie ma potrzeby regulacji nadrzędnej w wyżej opisanym sensie. Ponadto, nie musi ona rozdzielać w pełni warstwy regulacji bezpośredniej od warstwy optymalizacji – część wartości zadanych dla regulatorów bezpośrednich może być wyznaczana i bezpośrednio przekazywana z warstwy optymalizacji, co zostało zaznaczone na rysunku 1.2. Nie zalicza się też do wyróżnionej warstwy regulacji nadrzędnej regulatorów głównych standardowych kaskadowych układów regulacji.

Warstwa *optymalizacji* jest kolejną, usytuowaną bezpośrednio nad warstwami regulacji, zob. rys. 1.2. Celem jej działania jest wyznaczanie *optymalnych wartości zadanych* dla bezpośrednio jej podporządkowanych regulatorów warstw regulacji nadrzędnej i bezpośredniej. Wartości te wynikają zwykle z optymalizacji określonej funkcji celu natury ekonomicznej, która określa zysk lub bieżące koszty działania obiektu. Rozwiązywane zadanie optymalizacji to najczęściej zadanie optymalizacji statycznej, wyznaczany jest wówczas *optymalny punkt pracy*, tzn. optymalne dla aktualnego stanu otoczenia obiektu wartości zadane dla regulatorów.

Częstotliwość rozwiązywania zadania optymalizacji, tj. częstotliwość interwencji warstwy optymalizacji jest z reguły znacznie mniejsza niż warstw niższych. Warstwa optymalizacji może przy tym działać w trybie synchronicznym lub asynchronicznym. W tym ostatnim przypadku zadanie optymalizacji punktu pracy obiektu aktywowane jest obserwowanymi czy estymowanymi na bieżąco zmianami niesterowanych wejść obiektu lub zmianami parametrów produkcji przekazywanymi z wyższej warstwy planowania produkcji. Zagadnieniom algorytmów warstwy optymalizacji poświęcony jest rozdział 4. W szczególności, rozważane jest tam zagadnienie wyznaczania optymalnego punktu pracy obiektu w sytuacji istotnej niepewności, przejawiającej się dysponowaniem przybliżonym jedynie modelem obiektu oraz niedostępną czy też niepełną informacją o aktualnych wartościach wejść niesterowanych. W wymienionej sytuacji proponuje się iteracyjne algorytmy wyznaczania optymalnej wartości punktu pracy. Polegają one na kilkukrotnym, iteracyjnym wyznaczaniu coraz to lepszych wartości, po-

prawianych na podstawie obserwacji efektów ich zastosowania w obiekcie. Nie dokonujemy jedynie jednokrotnej optymalizacji, której wynik byłby nieoptymalny ze względu na niepełną informację o obiekcie i wpływie otoczenia, ale algorytm optymalizacji pracuje w pętli sprzężenia zwrotnego redukując w ten sposób wpływ niepewności.

Interesującym zjawiskiem obserwowanym w ostatnich latach, związanym z rozwojem możliwości sprzętu i oprogramowania dla kompleksowego sterowania obiektów przemysłowych, jest integracja oprogramowania do predykcyjnej regulacji nadrzędnej i optymalizacji bieżącej punktów pracy. Algorytmy nadrzędne regulacji predykcyjnej złożonych obiektów działają z reguły w oparciu o rozwiązywane w każdej chwili próbkowania numeryczne zadanie optymalizacji sterowań. Algorytmy tego typu wymagają zarówno dość znacznej mocy obliczeniowej, jak i dość dobrego modelu obiektu (algorytmom predykcyjnym poświęcony jest rozdział 3 książki). Stąd komercyjne pakiety oprogramowania oferujące algorytmy predykcyjne regulacji wielowymiarowej z ograniczeniami są z reguły kompleksowe i drogie, zawierają również procedury do modelowania i identyfikacji obiektu sterowanego, jak też procedury do optymalizacji bieżącej punktów pracy – bezpośrednio w pakiecie lub w module z nim ściśle związanym. Zbierane na bieżąco z systemu DCS pomiary zmiennych wyjściowych i wejściowych obiektu wykorzystywane mogą być w algorytmach zarówno identyfikacji (dostrajania, adaptacji) modelu, jak i regulacji oraz optymalizacji. Procedura optymalizacji dostarcza algorytmowi regulacji nadrzędnej odpowiednich wartości zadanych dla wyjść regulowanych, uruchamiana jest w odpowiednim rytmie dopasowanym do działania algorytmu nadrzędnej regulacji predykcyjnej. Interesującym z punktu widzenia integracji regulacji i optymalizacji jest przypadek, kiedy liczba sterowań wyznaczanych przez nadrzędny regulator predykcyjny jest większa niż liczba wyjść regulowanych. W tej sytuacji możliwe jest uwzględnienie w funkcji celu algorytmu predykcyjnego również składników o charakterze ekonomicznym.

Najwyższą z warstw przedstawionych na rysunku 1.2 jest warstwa *planowania produkcji*. Jej zadaniem jest ustalanie warunków pracy warstwy optymalizacji, tzn. celów, parametrów produkcji i jej ograniczeń. Warstwa ta działa na styku ekonomicznego otoczenia obiektu, reagując odpowiednio na zamówienia dotyczące asortymentu i wielkości produkcji, ceny, zbyt itp. wielkości przychodzące bezpośrednio z otoczenia rynkowego czy środowiska dużego zakładu, którego obiekt sterowany jest elementem. Częstotliwość interwencji tej warstwy odpowiada okresowi czasu zmiany produkcyjnej, czy nawet kilku dni. Algorytmy jej działania i stosowane przez nią modele procesu nie są przedmiotem zainteresowania niniejszej książki.

Podstawowymi cechami wyróżniającymi poszczególne warstwy są odrębne, *wydzielone cele sterowania i różne częstotliwości interwencji*. W tabeli 1.1 zestawiono podstawowe zadania poszczególnych warstw struktury sterowania obiektem przemysłowym i odpowiadające im typowe okresy interwencji, por. np. [54].

Warstwa	Podstawowe zadanie	Typowy okres interwencji
regulacja bezpośrednia (i zabezpieczenia)	stabilizacja, nadzorowanie procesu	ułamek sekundy
regulacja nadrzędna	sterowanie jakością – zaawansowana regulacja kluczowych zmiennych (często blisko ograniczeń)	minuta
optymalizacja	maksymalizacja bieżących efektów ekonomicznych	godzina
zarządzanie produkcją	maksymalizacja efektów ekonomicznych w dłuższych okresach	dość, kilka dni

Tablica 1.1. Podstawowe zadania i okresy interwencji warstw sterowania

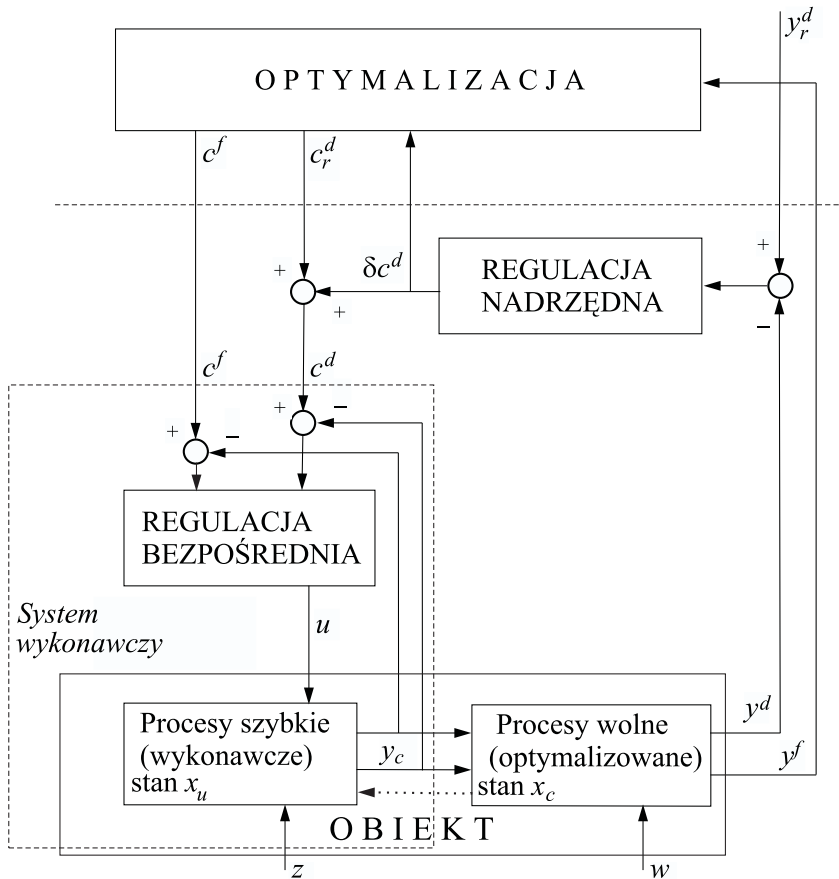
*Podstawowa przyczyna znaczenia, powszechnego stosowania warstwowych struktur sterowania jest następująca :*

*Podział (dekompozycja) wyjściowego, całościowego problemu sterowania na prostsze, funkcjonalnie wzajemnie powiązane między sobą podzadania upraszcza projektowanie, sterowania i nadzorowanie procesu – projektujemy i nadzorujemy układy sterowania dla poszczególnych warstw, realizujące wydzielone cele cząstkowe, a nie jeden układ centralny dla całego procesu.*

Dlatego też dla wielu procesów złożonych podejście warstwowe jest jedynym możliwym.

#### 1.4. Modelowanie obiektu w strukturze warstwowej

Rozważymy teraz nieco szczegółowiej strukturę warstwową sterowania składającą się z warstw regulacji i optymalizacji przedstawioną na rysunku 1.3. Charakterystyczną cechą tej struktury jest przedstawienie zachodzących w obiekcie procesów dynamicznych w postaci zdekomponowanej na procesy *szybkie* poddawany wpływom zmiennych sterujących  $u$  i zakłóceń



Rysunek 1.3. Struktura warstwowa regulacji i optymalizacji z dekompozycją obiektu sterowanego

szybkozmiennych  $z$  oraz na procesy *wolne*. Wyjściami części „szybkiej” obiektu są zmienne oznaczone przez  $y_c$ , będące z kolei wejściami do części obiektu charakteryzującej się procesami wolnymi, która poddana jest bezpośrednio wpływem wolniej zmiennych zakłóceń  $w$ . Wyjścia  $y = (y^f, y^d)$  to pewne zmienne istotne dla zadania regulacji nadrzędnej i zadania optymalizacji.

Przedstawionym na rysunku 1.3 celem działania regulacji nadrzędnej jest wymuszanie osiągnięcia przez pewne składowe wektora  $y$  wyjść obiektu, oznaczone jako podwektor  $y^d$ , wartości zadanych  $y_r^d$ , tzn. wymuszanie ograniczenia równościowego

$$y^d(t) = y_r^d.$$

Odpowiada to spotykanej w zastosowaniach praktycznych sytuacji regulacyjnej kontroli ważnych, w normalnych warunkach pracy zawsze aktywnych ograniczeń określonych zmiennych wyjściowych, np. utrzymywania stężenia zanieczyszczenia w strumieniu produktu na określonej wartości odpowiednio (bezpiecznie) bliskiej jego maksymalnej wartości dopuszczalnej. Bliżej tej wartości nie można ze względu na nieuchronną wariancję uchybu regulacji i stąd groźbę przekroczenia wartości dopuszczalnej, zaś produkt mniej zanieczyszczony to z reguły wyższe koszty produkcji. Pozostałe, swobodne (nieregulowane) składowe wektora wyjść  $y$  oznaczono jako podwektor  $y^f$ .

Kluczową rolę w przedstawionej dekompozycji obiektu odgrywa odpowiedni wybór zmiennych regulowanych  $y_c$ , wartości zadane dla zmiennych regulowanych oznaczone przez  $c = (c^f, c^d)$  są zmiennymi decyzyjnymi w zadaniu optymalizacji. Wybór zmiennych regulowanych wynika na ogół z doświadczenia projektantów i operatorów procesów sterowanych oraz postawionych zadań optymalizacji, w celu zapoznania się z aspektami formalnymi tego wyboru odsyłamy czytelnika do pracy [42].

Wybór zmiennych regulowanych  $y_c$  powinien być taki, aby:

- stabilizacja wartości  $y_c$  na sensownie dobranych wielkościach zadanych  $c$  zapewniała bezpieczne panowanie nad obiektem, tzn. określała jednoznacznie istotne współrzędne jego stanu,
- wartości zadane  $c$  jako zmienne decyzyjne zadania optymalizacji pozwalały realizować to zadanie, tzn. zapewniały pełne wykorzystanie możliwości wpływania na zmienne wyjściowe obiektu  $y$  istotne dla wyznaczania wartości optymalizowanego kryterium działania obiektu (funkcji celu) i wartości ograniczeń.

Podkreślimy, że pierwszy z wymienionych warunków zawsze musi być spełniony, jest to podstawowe wymaganie prawidłowego, bezpiecznego działania warstwy regulacji bezpośredniej. Jeśli wybór zmiennych regulowanych spełniających ponadto drugie z wymagań nie umożliwia pełnej realizacji celów postawionych przed warstwą optymalizacji, to dysponujemy zbyt małym zestawem zmiennych decyzyjnych. Należy wtedy albo nieco mniej ambitnie postawić cele optymalizacji, albo wzbogacić obiekt o dodatkowe zmienne manipulowane i w rezultacie dodatkowe zmienne regulowane umożliwiając w ten sposób pełną realizację postawionych celów (zmiennych regulowanych nie może być w normalnych warunkach pracy więcej niż manipulowanych, zob. [42]).

Rozważymy bliżej strukturę warstwową przedstawioną na rysunku 1.3. Jeśli układy regulacji bezpośredniej działają dobrze, to pomijając okresy następujące bezpośrednio po szybkich (np. skokowych) zmianach wartości

zadanych  $c$  czy zakłóceń, możemy przyjąć, że zachodzi

$$y_c(t) = c(t). \quad (1.1)$$

Można wówczas założyć, że z punktu widzenia regulatorów nadrzędnych i algorytmów optymalizacji widoczne są jedynie procesy dynamiczne wolne. Relację wejście-wyjście definiowaną przez te procesy opiszemy operatorem  $F$ ,

$$y(t) = F(c(t), w(t)). \quad (1.2)$$

Stąd, dla regulatorów nadrzędnych i algorytmów optymalizacji procesy szybkie obiektu wraz z warstwą regulacji bezpośredniej mogą być traktowane jako swego rodzaju *system wykonawczy* – realizujący zadane wartości  $c(t)$  zmiennych regulowanych  $y_c(t)$ , tzn. wymuszający  $y_c(t) \cong c(t)$ . Określenie „system wykonawczy” (*actuating system* [37]) wprowadzone zostało przez analogię do „elementu wykonawczego”, jakim jest np. zawór z pozycjonerem. Stąd też procesy szybkie nazwano na rysunku 1.3 procesami *wykonawczymi*, zaś z kolei procesy wolne procesami *optymalizowanymi*, jako że zachowanie obiektu charakteryzowane tymi właśnie procesami jest widziane przez warstwy nadrzędne, w szczególności przez warstwę optymalizacji.

Analityczna postać operatora modelu obiektu (1.2) rzadko jest dostępna. Wynika ona z rozwikłania następującej, typowo przyjmowanej dla układów ciągłych o parametrach skupionych postaci modelu

$$\begin{aligned} \frac{dx_c(t)}{dt} &= f_c(x_c(t), c(t), w(t)), \\ y(t) &= g_c(x_c(t), c(t)), \end{aligned} \quad (1.3)$$

gdzie  $x_c$  jest wektorem stanu procesów wolnych obiektu, zob. rys. 1.3, zaś konsekwentnie założono  $y_c(t) = c(t)$  eliminując w ten sposób z opisu zmienne odpowiadające wyjściom regulowanym  $y_c(t)$ , tzn. zakładając idealne działanie systemu wykonawczego.

Opis *całego* obiektu dynamicznego można natomiast przyjąć, analogicznie, w następującej ogólnej postaci

$$\begin{aligned} \frac{dx_u(t)}{dt} &= f_1(x_u(t), x_c(t), u(t), z(t), w(t)), \\ \frac{dx_c(t)}{dt} &= f_2(x_u(t), x_c(t), u(t), w(t)), \\ y(t) &= g(x_u(t), x_c(t), u(t)), \end{aligned} \quad (1.4)$$

gdzie wektor stanu  $x(t)$  zapisano w postaci podzielonej odpowiadającej stanom szybkim i wolnym,  $x(t) = (x_u(t), x_c(t))$  oraz, konsekwentnie, założono

brak *bezpośredniego* wpływu zakłóceń szybkozmiennych  $z(t)$  na podwektor stanów wolnych  $x_c(t)$ . Dekompozycja, czyli podział całościowego wektora stanu  $x$  na podwektory stanów „szybkich” i „wolnych”,  $x_u$  i  $x_c$ , jest w każdym przypadku kwestią indywidualną, wynikającą z cech obiektu i określonych wyżej wymagań dotyczących zmiennych regulowanych. Modele (1.4) i (1.3) należy oczywiście uzupełnić zestawem odpowiednich warunków początkowych, niezbędnych przy formalnych rozważaniach analitycznych czy jakichkolwiek obliczeniach numerycznych.

Przyjmując równanie zmiennych regulowanych w postaci (zob. [42])

$$y_c(t) = h(x(t), u(t)),$$

można rozważyć związki między opisami (1.4) i (1.3). Z przyjęcia równości (1.1),  $c(t) = y_c(t)$ , wynika wówczas bezpośrednio

$$\begin{aligned} \frac{dx_c(t)}{dt} &= f_c(x_c(t), h(x_u(t), x_c(t), u(t)), w(t)), \\ &= f_2(x_u(t), x_c(t), u(t), w(t)), \end{aligned}$$

oraz podobnie

$$\begin{aligned} y(t) &= g_c(x_c(t), h(x_u(t), x_c(t), u(t))), \\ &= g(x_u(t), x_c(t), u(t)). \end{aligned}$$

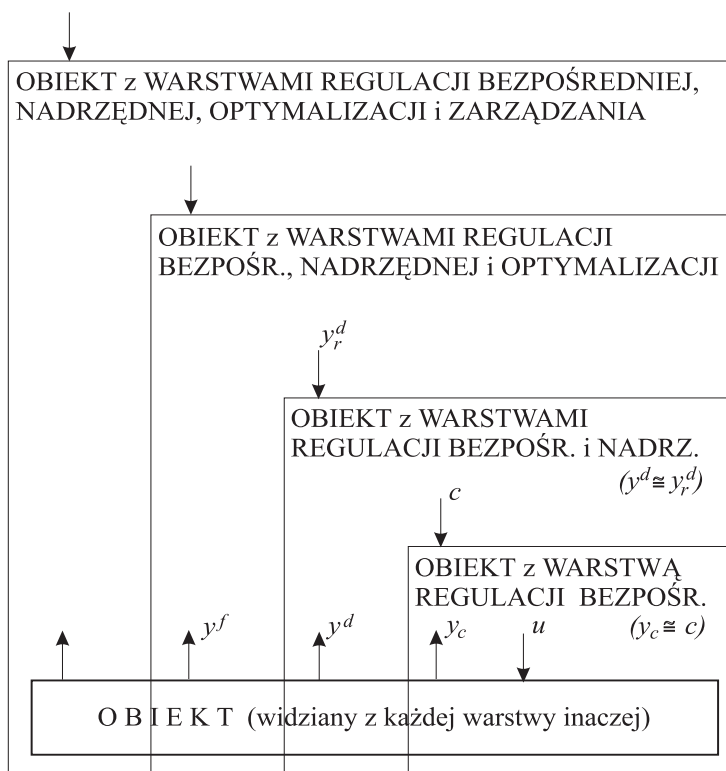
W strukturze warstwowej każda z warstw steruje w istocie tym samym obiektem, ale każda inaczej. Warstwa regulacji bezpośredniej jako jedyna ma bezpośredni dostęp do sterowanych zmiennych wejściowych obiektu (zmiennych manipulowanych) i otrzymuje pomiary wszystkich dostępnych wielkości wyjściowych istotnych dla jego stabilizacji i bezpiecznego prowadzenia, w tym przede wszystkim pomiary zmiennych wyjściowych szybciej zmiennych decydujących o możliwości odpowiednio szybkiej reakcji układów regulacji. Warstwa regulacji nadrzędnej, interweniująca i otrzymująca pomiary z reguły rzadziej, postrzega już obiekt nieco inaczej. Widzi go łącznie z układami regulatorów bezpośredniego działania, dla których wyznacza decyzje w postaci wartości zadanych. Jest to już nieco inny obiekt, o dynamice determinowanej dynamiką procesów przebiegających wolniej.

Ponadto, przy dobrym działaniu regulatorów bezpośredniego działania, przebiegi szybkozmiennie wywoływane wpływem szybkozmiennych zakłóceń są dla warstwy nadrzędnej mało istotne, często pomijalne. Podobnie, warstwa optymalizacji postrzega obiekt łącznie ze wszystkimi układami regulacji. Stąd modelowanie obiektu dla potrzeb sterowania, a więc ujmujące jedynie podstawowe zależności istotne dla projektowania sterowania, jest w każdej warstwie inne, prowadzi w efekcie do innych modeli. W tabeli 1.2 zestawiono typowe, podstawowe cechy modeli w poszczególnych

warstwach struktury sterowania. Zwróćmy uwagę, że im wyższa warstwa, tym model mniej szczegółowy, bardziej zagregowany, z wolniejszą dyna-

Warstwa sterowania	Model
Regulacja bezpośrednia (i zabezpieczenia)	szybki dynamiczny (liniowy, ew. nieliniowy)
Regulacja nadrzędna	wolny dynamiczny (liniowy, nieliniowy)
Optymalizacja	nieliniowy statyczny (ew. dynamiczny)
Zarządzanie produkcją	liniowy bilansowy (stacyjny)

Tablica 1.2. Cechy modeli procesu sterowanego w poszczególnych warstwach sterowania

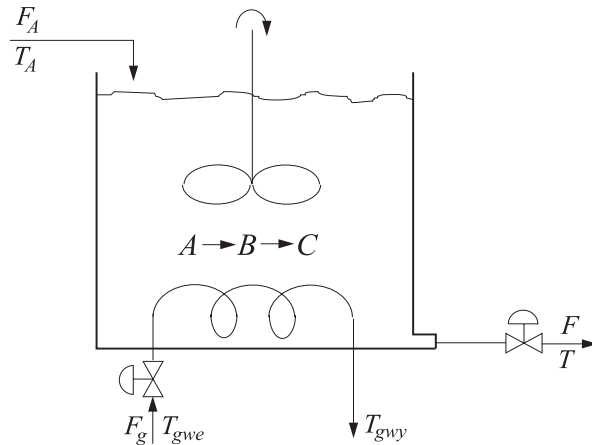


Rysunek 1.4. Modelowanie obiektu sterowanego w strukturze warstwowej (oznaczenia zmiennych jak na rysunku 1.3)

miką lub statyczny. Obiekt wraz z wszystkimi niższymi warstwami jest dla rozważanej warstwy sterowania pewnym "systemem wykonawczym", który powinien realizować decyzje tej warstwy. Dla przykładu, zaznaczony na rysunku 1.3 „system wykonawczy” złożony z obiektu łącznie z układami regulacji bezpośredniej realizuje (wymusza) zadane przez  $c = (c^d, c^f)$  wartości zmiennych regulowanych  $y_c$ . Na rysunku 1.4 przedstawiono strukturę modelowania obiektu w kolejnych warstwach sterowania, wyraźnie pokazującą charakterystyczną cechę *zagnieżdżenia*. W danej warstwie modeluje się obiekt wraz z układami sterowania warstw niższych – oczywiście łącznie ze skutkami ich działania umożliwiającymi odpowiednie uproszczenia i agregację, adekwatnie do zadania i skali czasu warstwy. Jest to oczywiście podejście uproszczone, ale umożliwiające skuteczne i sprawdzone w praktyce projektowanie oraz bieżącą realizację sterowania i optymalizacji procesów.

### Przykład 1.1

Zagadnienie dekompozycji modelu obiektu i regulacji nadrzędnej zilustrowane zostanie na prostym przykładzie reaktora o pracy ciągłej, przedstawionego na rysunku 1.5.



Rysunek 1.5. Schemat reaktora przepływowego z idealnym wymieszaniem (do przykładu 1.1)

Reaktor zasilany jest substancją  $A$  dopływającą strumieniem o natężeniu  $F_A$  [ $kg/min$ ] i temperaturze  $T_A$  [ $K$ ]. W reaktorze zachodzą reakcje  $A \rightarrow B \rightarrow C$  z prędkościami odpowiednio  $r_B[\frac{1}{min}]$  i  $r_C[\frac{1}{min}]$ , przy czym

pierwsza reakcja jest endotermiczna, a druga egzotermiczna. Mieszanina w zbiorniku ogrzewana jest substancją grzewczą przepływającą przez rurowy wymiennik ciepła umieszczony w reaktorze.

Natężenia dopływu medium grzewczego  $F_g$  i wypływu produktu z reaktora  $F$  mogą być regulowane przez odpowiednie zawory.

Przyjmujemy następujące *założenia upraszczające*:

1. W reaktorze następuje idealne wymieszanie, stąd skład (opisany stężeniami  $C_A$  substancji A i  $C_B$  substancji B) oraz temperatura  $T$  mieszaniny są jednolite w całej objętości.
2. Prędkości reakcji opisują modele:

$$r_B = k_1(T) C_A = k_{10} \exp\left(-\frac{E_1}{RT}\right) C_A, \quad (1.5)$$

$$r_C = k_2(T) C_B = k_{20} \exp\left(-\frac{E_2}{RT}\right) C_B, \quad (1.6)$$

gdzie

$$k_{10} = 9000 \ll k_{20} = 35000 \left[\frac{1}{\text{min}}\right],$$

$$\frac{E_1}{R} = 4000 < \frac{E_2}{R} = 5200 [K],$$

zaś ciepła reakcji są znane i wynoszą odpowiednio  $h_1 \left[\frac{J}{kg}\right]$  i  $h_2 \left[\frac{J}{kg}\right]$ .

3. Pomijamy wpływ energii mechanicznej ruchu mieszadła i wymianę ciepła z otoczeniem, przyjmujemy jako jednakowe gęstości  $\rho \left[\frac{kg}{m^3}\right]$  i ciepła właściwe  $c_w \left[\frac{J}{K kg}\right]$  wszystkich substancji i mieszaniny.
4. Za siłę napędową wymiany ciepła przyjmujemy średnią z temperatur medium grzewczego na wejściu i na wyjściu rury wymiennika ciepła, wymianę ciepła przez powierzchnię wymiennika opisujemy wzorem empirycznym

$$H = a(F_g)^b (T_{g\acute{s}r} - T), \quad (1.7)$$

gdzie  $a$  i  $b$  to współczynniki zależne od parametrów konstrukcyjnych wymiennika ciepła [90].

5. Zakres regulacji natężenia przepływu  $F_g$  umożliwia uzyskanie i stabilizację temperatury reaktora w zakresie 300 – 360 K.

Przy powyższych założeniach reaktor można potraktować jako obiekt charakteryzowany czterema zmiennymi stanu opisującymi bilanse masy mieszaniny i poszczególnych substancji oraz bilans ciepły. Dla celów modelowania przyjmiemy następujące *zmiennne stanu* obiektu:

- $W$  – ilość mieszaniny w zbiorniku,  
 $C_A$  – stężenie substancji  $A$ ,  
 $C_B$  – stężenie substancji  $B$ ,  
 $T$  – temperatura mieszaniny w zbiorniku.

Układamy *równania stanu*:

— dla zapełnienia  $W$  (bilans masy mieszaniny):

$$\frac{dW}{dt} = F_A - F,$$

— dla stężenia  $C_A$  (bilans masy substancji  $A$ ):

$$\begin{aligned} \frac{d(WC_A)}{dt} &= W \frac{dC_A}{dt} + C_A \frac{dW}{dt} \\ &= F_A - FC_A - Wk_1(T)C_A, \\ W \frac{dC_A}{dt} &= -C_A(F_A - F) + F_A - FC_A - Wk_1(T)C_A \\ &= -C_A F_A + F_A - Wk_1(T)C_A, \end{aligned}$$

czyli

$$\frac{dC_A}{dt} = \frac{1 - C_A}{W} F_A - k_1(T)C_A;$$

— dla stężenia  $C_B$  (bilans masy substancji  $B$ ):

$$\begin{aligned} \frac{d(WC_B)}{dt} &= W \frac{dC_B}{dt} + C_B \frac{dW}{dt} \\ &= -FC_B + Wk_1(T)C_A - Wk_2(T)C_B, \\ W \frac{dC_B}{dt} &= -C_B(F_A - F) - FC_B + Wk_1(T)C_A - Wk_2(T)C_B \\ &= -C_B F_A + Wk_1(T)C_A - Wk_2(T)C_B, \end{aligned}$$

czyli

$$\frac{dC_B}{dt} = -\frac{C_B}{W} F_A + k_1(T)C_A - k_2(T)C_B;$$

— dla temperatury  $T$  (bilans cieplny):

$$\begin{aligned} \frac{d(c_w WT)}{dt} &= c_w W \frac{dT}{dt} + c_w T \frac{dW}{dt} \\ &= c_w F_A T_A - c_w FT + F_g c_g (T_{gwe} - T_{gwy}) - h_1 W k_1(T) C_A + h_2 W k_2(T) C_B, \end{aligned}$$

$$c_w W \frac{dT}{dt} = -c_w T F_A + c_w F_A T_A + F_g c_g (T_{gwe} - T_{gwy}) - h_1 W k_1(T) C_A + \\ + h_2 W k_2(T) C_B.$$

Spośród zmiennych występujących po prawej stronie powyższego równania temperatura wyjściowa medium grzewczego  $T_{gwy}$  jest wielkością zależną od zmiennych wejściowych. Uwzględniając przyjęte w punkcie 4 założenia dotyczące wymiany ciepła, w tym wzór (1.7), można wyeliminować temperaturę  $T_{gwy}$  z równania bilansu cieplnego. Mamy bowiem równocześnie dwie zależności

$$H = a(F_g)^b \left( \frac{T_{gwe} + T_{gwy}}{2} - T \right),$$

$$H = F_g c_g (T_{gwe} - T_{gwy}).$$

Wyznaczając  $T_{gwy}$  z drugiego równania,

$$T_{gwy} = -\frac{H}{F_g c_g} + T_{gwe}$$

i wstawiając do pierwszego dostajemy

$$H = a(F_g)^b \left( \frac{1}{2} (T_{gwe} - \frac{H}{F_g c_g} + T_{gwe}) - T \right) \\ = a(F_g)^b (T_{gwe} - T) - \frac{a}{2} (F_g)^b \frac{H}{F_g c_g}.$$

Stąd wyznaczamy wzór na ilość wymienianego ciepła  $H$

$$H = \frac{a(F_g)^{b+1}}{F_g + \frac{a(F_g)^b}{2c_g}} (T_{gwe} - T). \quad (1.8)$$

Równanie bilansu cieplnego przyjmuje więc postać

$$c_w W \frac{dT}{dt} = c_w F_A (T_A - T) + H(T_{gwe} - T, F_g) - h_1 W k_1(T) C_A + h_2 W k_2(T) C_B,$$

czyli

$$\frac{dT}{dt} = \frac{F_A}{W} (T_A - T) + \frac{1}{c_w W} H(T_{gwe} - T, F_g) - \frac{h_1}{c_w} k_1(T) C_A + \frac{h_2}{c_w} k_2(T) C_B,$$

gdzie  $H(T_{gwe} - T, F_g)$  dane jest wzorem (1.8).

Reasumując, układ równań stanu rozpatrywanego reaktora ma postać

$$\frac{dW}{dt} = F_A - F, \quad (1.9a)$$

$$\begin{aligned} \frac{dT}{dt} = \frac{F_A}{W}(T_A - T) + \frac{1}{c_w W} H(T_{gwe} - T, F_g) - \frac{h_1}{c_w} k_1(T) C_A + \\ + \frac{h_2}{c_w} k_2(T) C_B, \end{aligned} \quad (1.9b)$$

$$\frac{dC_A}{dt} = \frac{1 - C_A}{W} F_A - k_1(T) C_A, \quad (1.9c)$$

$$\frac{dC_B}{dt} = -\frac{C_B}{W} F_A + k_1(T) C_A - k_2(T) C_B, \quad (1.9d)$$

gdzie prędkości reakcji  $k_1(T)$  i  $k_2(T)$  dane są wzorami (1.5) i (1.6), zaś ilość doprowadzonego ciepła  $H(T_{gwe} - T, F_g)$  zależnością (1.8).

Kolejność, w jakiej zostały ustawione równania stanu (1.9a)–(1.9d) nie jest przypadkowa. Z natury zjawisk fizykochemicznych zachodzących w reaktorze wynika, że zmienne stanu  $W$  i  $T$  są szybciej zmienne niż stężenia  $C_A$  i  $C_B$ , pod wpływem zmian wielkości wejściowych sterowanych ( $F$ ,  $F_g$ ) czy niesterowanych ( $F_A$ ,  $T_A$ ,  $T_{gwe}$ ). Co więcej, są to wielkości, których stabilizacja zapewnia bezpieczną pracę reaktora, tzn. bez przepelnienia czy nadmiernego opróżnienia zbiornika, bez przekraczania dozwolonych temperatur. Stąd dekompozycja wektora stanu, wg opisu (1.4):

$$\begin{aligned} x_u &= [W \ T]^T, \\ x_c &= [C_A \ C_B]^T. \end{aligned}$$

Wektorem zmiennych sterujących  $u$  jest tu

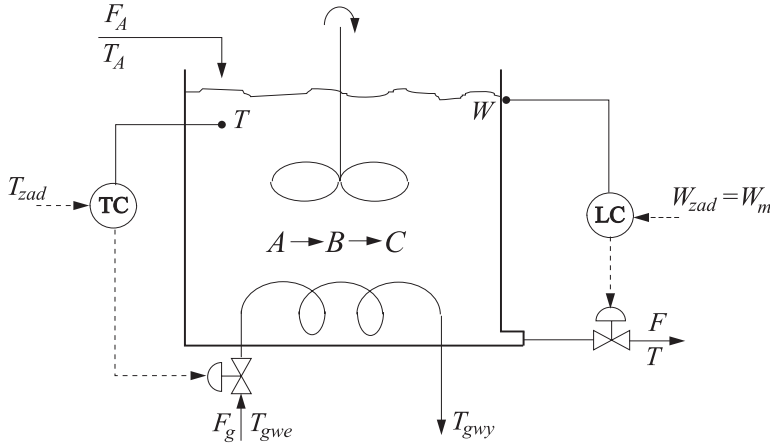
$$u = [F \ F_g]^T,$$

zaś za wyjścia regulowane przez regulatory warstwy sterowania bezpośredniego przyjmiemy bezpośrednio szybciej zmienne współrzędne stanu, tzn. temperaturę  $T$  i zapelnienie  $W$ ,

$$y_c = [T \ W]^T.$$

Jest to oczywiście wybór najlepszy z punktu widzenia panowania nad obiektem. Rysunek 1.6 przedstawia reaktor wraz z układami regulacji bezpośredniej poziomu i temperatury. Wartościami zadanymi dla regulatorów warstwy regulacji bezpośredniej, w terminologii rysunku 1.3, są składowe  $T_{zad}$ ,  $W_m$  wektora  $c$ ,

$$c = [T_{zad} \ W_m]^T.$$



Rysunek 1.6. Reaktor wraz z układami regulacji bezpośredniej (LC – regulator poziomu, TC – regulator temperatury)

Dla projektowania warstwy nadrzędnej struktury sterowania zakłada się dobre działanie regulatorów bezpośrednich, tzn.

$$\begin{aligned} W(t) &= W_m, \\ T(t) &= T_{zad}(t), \end{aligned}$$

przy czym szczególnie pierwsza z tych równości może być stosunkowo dokładnie spełniana (szybka stabilizacja zapęfnienia poprzez sterowanie natężeniem wypływu). Przy poczynionych założeniach dynamika procesów wolnych, zob. (1.3), opisana jest równaniami

$$\frac{dC_A}{dt} = \frac{1 - C_A}{W_m} F_A - k_1(T) C_A, \quad (1.10)$$

$$\frac{dC_B}{dt} = -\frac{C_B}{W_m} F_A + k_1(T) C_A - k_2(T) C_B, \quad (1.11)$$

gdzie  $T = T_{zad}$ .

Rozważmy następujące sformułowanie (a) *celu sterowania* (dalej rozważymy inne sformułowanie oznaczone przez (b)):

- (a) *Stabilizacja stężenia  $C_B$  substancji B w strumieniu produktu na wartości  $C_{Bzad} = 0.25$ , przy stałym maksymalnym użytecznym zapęfnieniu zbiornika  $W(t) = W_m = const$  i wolnozmiennych wahaniami dopływu  $W/F_A = 25 \pm 5$  [min], przy dostępnym bieżącym pomiarze stężenia  $C_B$*

analizatorem o czasie pomiaru znacznie większym od okresu próbkowania układów regulacji poziomu i temperatury.

Realizacja powyższego celu zakłada utrzymywanie obiektu regulowanego w stanie ustalonym – mamy *stabilizować* stężenie. Obiekt (1.10)-(1.11) znajduje się w stanie ustalonym, kiedy

$$\frac{dC_A(t)}{dt} = 0, \quad \frac{dC_B(t)}{dt} = 0,$$

pomimo wpływu wolnozmienniej wartości zakłócenia  $F_A(t)$ .

Równania modelu obiektu wolniejszego dla *stanów ustalonych*:

$$0 = \frac{1 - C_A}{W_m} F_A - k_{10} \exp\left(-\frac{E_1}{RT}\right) C_A,$$

$$0 = -\frac{C_B}{W_m} F_A + k_{10} \exp\left(-\frac{E_1}{RT}\right) C_A - k_{20} \exp\left(-\frac{E_2}{RT}\right) C_B.$$

Wyznaczając  $C_A$  z pierwszego równania,

$$C_A = \frac{\frac{F_A}{W_m}}{\frac{F_A}{W_m} + k_{10} \exp\left(-\frac{E_1}{RT}\right)},$$

i wstawiając do drugiego dostajemy

$$C_B \left( \frac{F_A}{W_m} + k_{20} \exp\left(-\frac{E_2}{RT}\right) \right) = k_{10} \exp\left(-\frac{E_1}{RT}\right) \frac{\frac{F_A}{W_m}}{\frac{F_A}{W_m} + k_{10} \exp\left(-\frac{E_1}{RT}\right)},$$

skąd mamy

$$C_B = \frac{\frac{F_A}{W_m} k_{10} \exp\left(-\frac{E_1}{RT}\right)}{\left( \frac{F_A}{W_m} + k_{10} \exp\left(-\frac{E_1}{RT}\right) \right) \left( \frac{F_A}{W_m} + k_{20} \exp\left(-\frac{E_2}{RT}\right) \right)}.$$

Wzór powyższy jest wygodniejszy w postaci:

$$C_B = \frac{\frac{W_m}{F_A} k_{10} \exp\left(-\frac{E_1}{RT}\right)}{\left( 1 + \frac{W_m}{F_A} k_{10} \exp\left(-\frac{E_1}{RT}\right) \right) \left( 1 + \frac{W_m}{F_A} k_{20} \exp\left(-\frac{E_2}{RT}\right) \right)}. \quad (1.12)$$

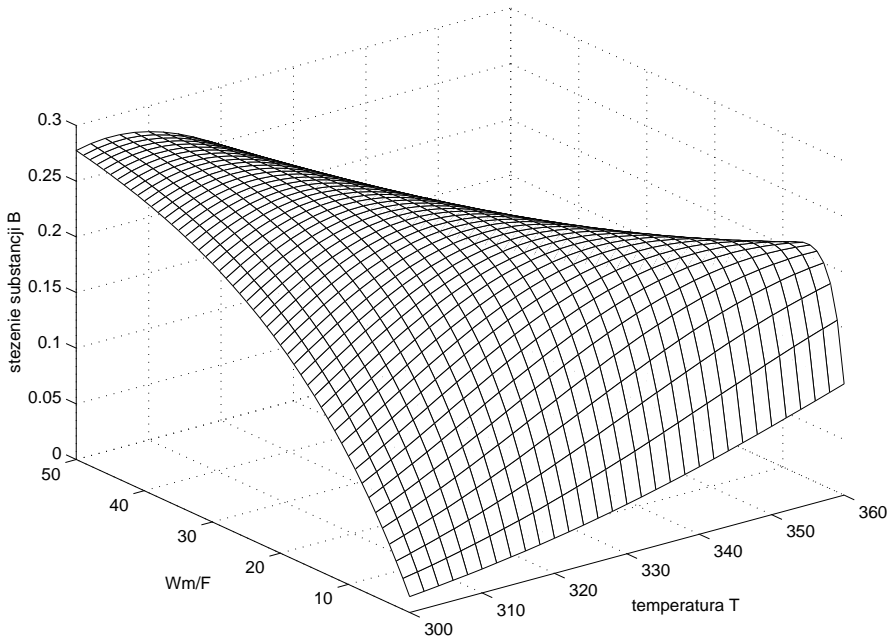
Wzór (1.12) opisuje model stanów ustalonych, czyli charakterystykę statyczną stężenia  $C_B$ ,

$$C_B = C_B(T, F_A), \quad (1.13)$$

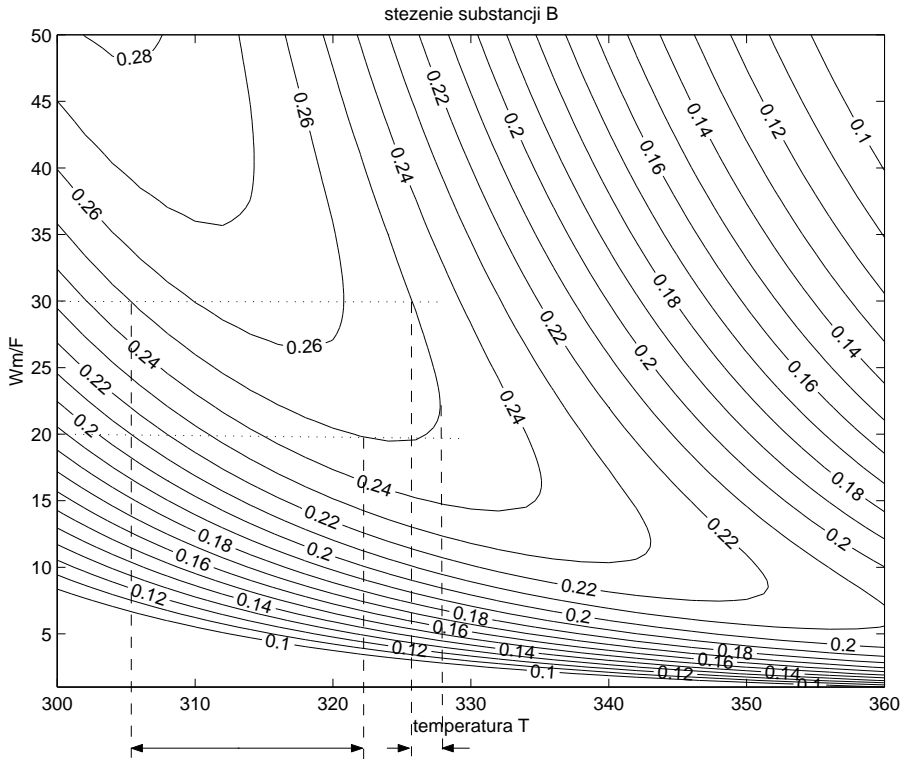
obiektu opisanego przez (1.10)-(1.11). W ogólnej terminologii opisu warstwowej struktury sterowania, wzór (1.13) stanowi konkretną postać funkcji  $y = F(c, w)$ , por. (1.2).

Charakterystykę (1.13) wygodniej jest przedstawić w nieco innym układzie współrzędnych, w funkcji  $\frac{W_m}{F_A}$  zamiast  $F_A$ , gdyż wartość ilorazu  $\frac{W_m}{F_A}$  – tzw. *czas przebywania* substancji w zbiorniku (czas napełniania zbiornika do zawartości  $W_m$  dopływem  $F_A$ ), jest dobrze znaną i interpretowalną wielkością. Kształt powierzchni  $C_B(T, \frac{W_m}{F_A})$  dla wartości liczbowych podanych w przykładzie przedstawiono na rysunku 1.7, zaś zbiory poziomicowe tej powierzchni na rysunku 1.8. Na rysunkach przyjęto  $F = F_A$ , równość oczywistą ze względu na zakładaną stałość zapełnienia  $W(t) = W_m$ .

Sformułowany cel sterowania polegający na stabilizacji stężenia  $C_B$  można zrealizować przez zastosowanie regulatora nadrzędnego, jak przedstawiono na rysunku 1.9. Zwróćmy uwagę, że zastosowanie regulacji nadrzędnej wynika nie tylko z wolniejszej dynamiki zmian stężenia, ale jest wręcz wymuszone czasem pomiaru jego wartości znacznie dłuższym od okresu próbkowania regulatorów poziomu i temperatury. Czas ten ogranicza od dołu wielkość okresu próbkowania regulatora nadrzędnego, a więc i momentów zmian jego wartości wyjściowej  $T_{zad}$ . Jeśli dynamika układów regulacji bezpośredniej poziomu i temperatury jest szybsza od dynamiki modelu (1.10), (1.11), to przy projektowaniu regulatora stężenia można



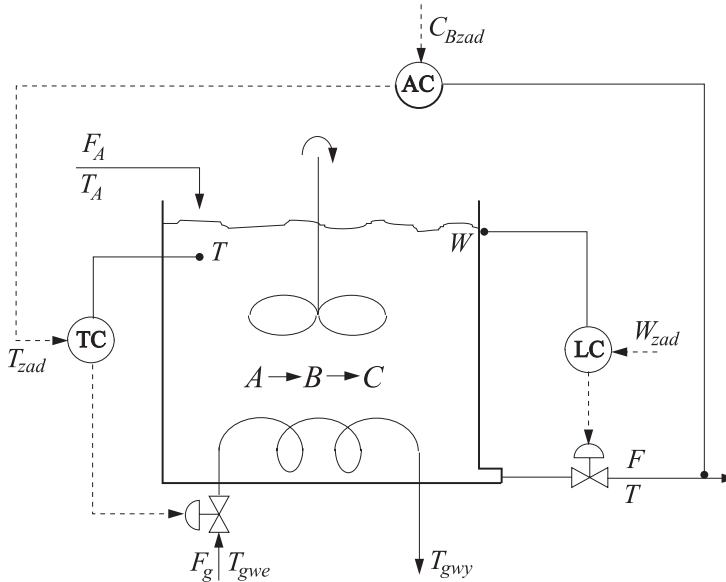
Rysunek 1.7. Powierzchnia  $C_B(T, \frac{W_m}{F})$



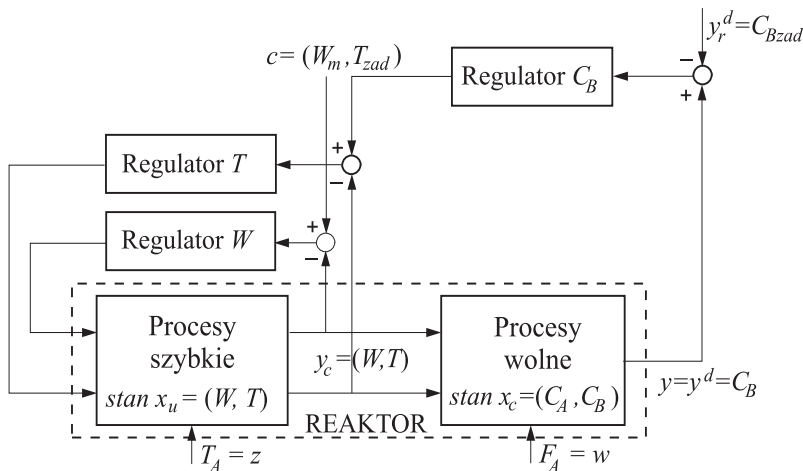
Rysunek 1.8. Zbiory poziomicowe powierzchni  $C_B(T, \frac{W_m}{F})$ , z zaznaczonymi możliwymi obszarami wartości wyjścia regulatora stabilizacji stężenia

posłużyć się jedynie tym modelem, *znacznie prostszym* niż model dynamiki całego układu. Należy jednakże potem zweryfikować zachowanie pełnego układu regulacji, np. symulacyjnie z uwzględnieniem dynamiki obu warstw czy eksperymentalnie na obiekcie, korygując w razie potrzeby nastawy regulatora nadrzędnego. Zauważmy, że układ regulacji stężenia  $C_B$  jest układem regulacji kaskadowej, ale o różnych okresach próbkowania regulatorów w pętli głównej i pętli pomocniczej (tzw. *multirate control system*). Schemat struktury sterowania z rysunku 1.9 został przedstawiony na rysunku 1.10 w języku schematu blokowego struktury warstwowej z rysunku 1.3, ze zdekomponowanym opisem obiektu.

Z kształtu zbiorów poziomicowych powierzchni  $C_B(T, \frac{W_m}{F})$  wynika, że obiekt jest nieliniowy i w szerokim zakresie zmienności  $\frac{W_m}{F}$  te same wartości stężenia można realizować różnymi wartościami temperatury. Na rysunku 1.8 przedstawiono dwa zakresy zmienności temperatury  $T$  stabilizującej stężenie na wartości 0.25. Czytelnikowi pozostawiamy analizę, który z



Rysunek 1.9. Reaktor z układami regulacji bezpośredniej (LC i TC) oraz nadrzędnej (AC) realizującej stabilizację stężenia  $C_B$



Rysunek 1.10. Zdekomponowany opis blokowy struktury sterowania reaktora realizującej stabilizację stężenia  $C_B$

tych zakresów jest korzystniejszy dla pracy regulatora stężenia i jak w tej sytuacji regulator projektować. Nadmienimy tylko, że projektując np. regulator liniowy typu PID standardowo wyznaczamy liniowy model obiektu w

określonym punkcie pracy i na podstawie tego modelu dobieramy nastawy regulatora. Każdemu z dwóch obszarów zaznaczonych na rysunku 1.8 odpowiadać oczywiście będzie inny punkt pracy spełniający  $\frac{W_m}{F_A} = 0.25$ .

## 1.5. Warstwa optymalizacji

Zadaniem *warstwy optymalizacji* jest dobór najlepszych z ekonomicznego punktu widzenia wartości zadanych zmiennych regulowanych  $c$ . W ogólności, stan optymalny procesu może być dynamiczny i proces może być prowadzony w sposób dynamiczny. W takiej sytuacji w warstwie optymalizacji wyznaczone są dynamiczne przebiegi optymalne zmiennych  $c(t)$  w wyniku rozwiązania zadania optymalizacji dynamicznej z ograniczeniami. *Funkcja celu* (*wskaźnik jakości*, *funkcja kryterialna*) takiego zadania może być sformułowana np. w postaci

$$J(c,y) = \int_{t_0}^{t_k} Q(c(t),y(t))dt, \quad (1.14)$$

gdzie  $t_k - t_0$  jest horyzontem optymalizacji. Funkcja celu optymalizowana jest przy uwzględnieniu *ograniczeń*:

- równań dynamiki procesu (1.3),
- ograniczeń nierównościowych natury fizycznej i technologicznej nałożonych na wartości zmiennych decyzyjnych  $c$  i wyjściowych  $y$ . Ograniczenia zapisać można w postaci ogólnej, zdefiniowanej przez pewne funkcje  $g$  i  $h$ ,

$$c \in C = \{c : g(c) \leq 0\}, \quad (1.15)$$

$$y \in Y = \{y : h(y) \leq 0\}. \quad (1.16)$$

W praktyce są to przede wszystkim proste ograniczenia chwilowych wartości maksymalnych czy minimalnych poszczególnych zmiennych,

$$c(t) \in C = \{c(t) : c(t) \geq c_{\min}, c(t) \leq c_{\max}\}, \quad (1.17)$$

$$y(t) \in Y = \{y(t) : y(t) \geq y_{\min}, y(t) \leq y_{\max}\}. \quad (1.18)$$

Nie uwzględniliśmy tu jawnie ograniczeń nałożonych na zmienne stanu  $x_c$  – zawsze tak można uczynić zakładając, że ograniczane składowe wektora  $x_c$  wchodzi w skład wektora wyjść. Założenie jest rozsądne, gdyż dla kontroli ograniczeń zmienne te muszą być bezpośrednio czy pośrednio mierzone, a więc należy je traktować jako wyjścia.

Używając wprowadzonych zależności, zadanie *optymalizacji dynamicznej* (sterowania optymalnego) jest zadaniem optymalizacji funkcji celu (1.14) przy ograniczeniach równościowych (1.3) i nierównościowych (1.15) i (1.16), tzn. w przypadku minimalizacji zadanie

$$\begin{aligned} \min \{ J(c,y) = \int_{t_0}^{t_k} Q(c(t),y(t))dt \} \\ \text{z ogr. : } dx_c(t)/dt = f_c(x_c(t), c(t),w(t)), \\ y(t) = g_c(x_c(t), c(t)), \\ c \in C, y \in Y. \end{aligned} \quad (1.19)$$

W ogólności jest to albo zadanie z rozwiązaniem okresowym – wtedy okres determinuje horyzont optymalizacji, albo zadanie z horyzontem wynikającym z dynamiki wejść niesterowanych  $w(t)$  i dynamiki samego procesu. W tym ostatnim przypadku sformułowane zadanie optymalizacji jest z reguły elementem wyznaczania sterowania w strukturze repetycyjnej z przesuwany horyzontem, zob. [42]. Jednakże, przy sterowaniu ciągłymi procesami przemysłowymi zakłócenia są z reguły natury nie pozwalającej na prognozy długoterminowe, stąd rozwiązania optymalne są stałe lub okresowe.

W praktyce dominuje *sterowanie stanem ustalonym*, tzn. sterowanie polegające na wyznaczeniu ustalonych optymalnych wartości zadanych dla układów regulacji, wartości optymalnych dla sytuacji obiektu charakteryzowanej aktualnymi wartościami wejść niesterowanych  $w$ , co można zapisać formalnie jako wyznaczanie wartości optymalnej  $\hat{c}(w)$ . Rozwiązujemy wówczas *zadanie optymalizacji statycznej*

$$\begin{aligned} \min Q(c,y) \\ \text{z ogr. : } 0 = f_c(x_c, c,w), \\ y = g_c(x_c, c), \\ c \in C, y \in Y. \end{aligned} \quad (1.20)$$

W przypadku dysponowania rozwikłanym statycznym modelem procesu (1.2) zadanie powyższe przyjmuje formalnie prostszą postać

$$\begin{aligned} \min Q(c,y) \\ \text{z ogr. : } y = F(c,w), \\ c \in C, y \in Y. \end{aligned} \quad (1.21)$$

Nie zawsze ograniczenia wartości wyjść obiektu, będące ograniczeniami *technologicznymi*, są realizowane jedynie przez ich uwzględnianie w sformu-

łowaniu zadania optymalizacji. W przypadku ważnych ograniczeń determinujących parametry jakości czy efektywności produkcji, aktywnych w nominalnych warunkach pracy, typowe jest *wymuszanie spełnienia ograniczeń przez regulatory nadrzędne*. Ten sposób postępowania, choć kosztowniejszy w realizacji, ma dwie podstawowe zalety:

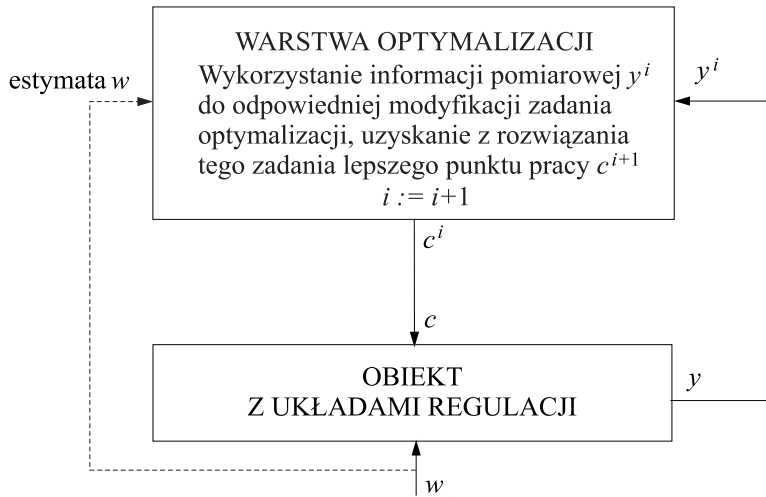
- ograniczenie „pilnowane” jest nie tylko w stanach ustalonych, ale i w okresach dynamicznych zmian w obiekcie, np. po zmianach punktu pracy czy wejść niesterowanych,
- zastosowanie pętli sprzężenia zwrotnego eliminuje (z dokładnością do uchybu regulacji) nieuchronny wpływ niedokładności modelu i estymat zakłóceń wykorzystywanych w zadaniu optymalizacji, prowadzący do możliwości przekraczania w rzeczywistym obiekcie ograniczeń spełnionych na modelu.

Na rysunku 1.3 w rozdziale poprzednim przedstawiono strukturę regulacji tego właśnie rodzaju, a w szczególności mający duże znaczenie praktyczne przypadek realizowanego przez regulatory nadrzędne ograniczenia równościowego składowych  $y^d$  wektora  $y$  wyjść obiektu,

$$y^d(t) = y_r^d.$$

Odpowiada to spotykanej w wielu zastosowaniach sytuacji regulacyjnej kontroli ważnych, w normalnych warunkach pracy zawsze aktywnych ograniczeń określonych wyjść procesu, np. utrzymywania stężenia kluczowego zanieczyszczenia w strumieniu produktu na określonej wartości odpowiednio bliskiej (ale nie równej) nierównościowemu ograniczeniu jego maksymalnej wartości dopuszczalnej.

W zadaniu optymalizacji statycznej (1.20) czy (1.21) występuje wektor  $w$  reprezentujący wolnozmiennne wielkości wejściowe niesterowane obiektu (zakłócenia). Z punktu widzenia optymalizacji są to parametry. Problem jest jedynie w tym, że model obiektu  $F$  powinien być modelem dobrze opisującym rzeczywistość, ponadto wartości zakłóceń powinny być znane, czyli zmierzone lub dobrze zidentyfikowane. Jedynie wtedy rozwiązanie zadania optymalizacji statycznej z modelem procesu daje w efekcie punkt pracy optymalny dla obiektu rzeczywistego, a dokładniej dostatecznie bliski optymalnemu – ze względu na zawsze nieuchronną pewną niepewność, chociażby powodowaną błędami pomiarów. Jeśli tak nie jest, to punkt optymalny dla (niedokładnego) modelu może być daleki od optymalnego dla obiektu rzeczywistego. W takiej sytuacji, podejściem umożliwiającym poprawę punktu pracy jest wykorzystanie dodatkowych informacji pomiarowych z obiektu, przez zastosowanie tzw. *iteracyjnych algorytmów sterowania optymalizującego punkty pracy*. Struktura tego rodzaju sterowania przedstawiona jest



Rysunek 1.11. Struktura iteracyjnego sterowania optymalizującego stan ustalony

na rysunku 1.11. Najprostszym sposobem zaznaczonej tam modyfikacji zadania optymalizacji jest poprawa (adaptacja) modelu procesu poprzez wykorzystanie nowego punktu pomiarowego. Takie postępowanie może być jednak niewystarczające, algorytm wymaga wówczas istotniejszych modyfikacji. Algorytmy iteracyjnej poprawy punktu pracy obiektów regulowanych rozważone będą w rozdziale 4.

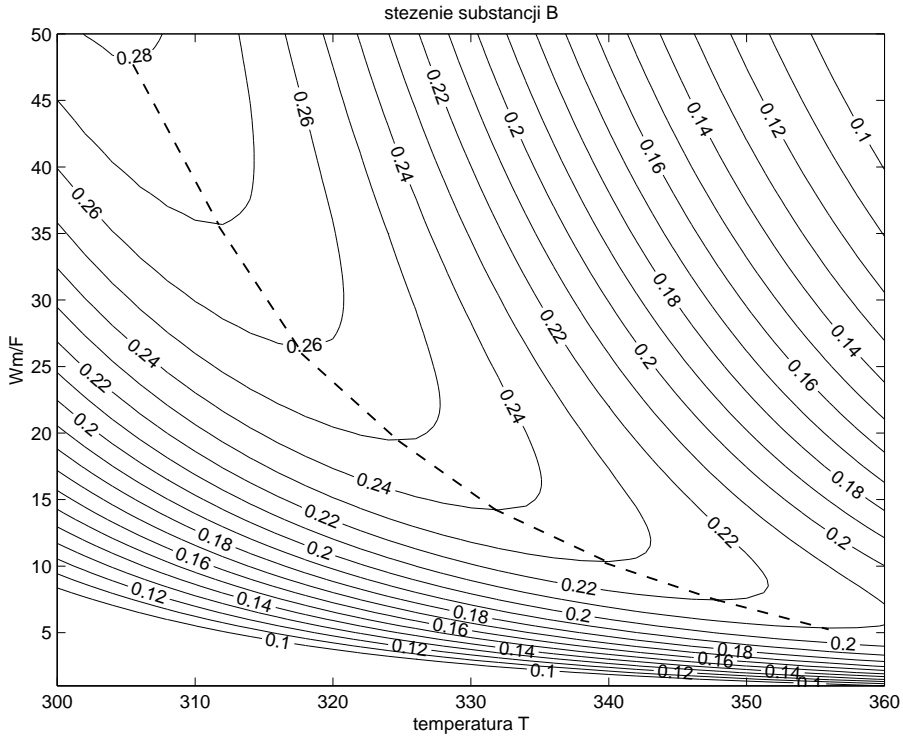
### Przykład 1.2

Wróćmy do rozważonego uprzednio w przykładzie 1.1 reaktora przepływowego z idealnym wymieszaniem. Tym razem postawimy jednak inny, optymalizacyjny *cel sterowania* w następującej postaci (b):

- (b) *Optymalizować na bieżąco sterowania tak, aby maksymalizować stężenie  $C_B(t)$  w strumieniu produktu w sytuacji wolnozmiennego, niesterowanego wejścia (zakłócenia)  $F_A(t)$ .*

Założenie zmian natężenia dopływu  $F_A$  wolnych w stosunku do dynamiki układu regulowanego pozwala na zastosowanie *sterowania stanem ustalonym*. Dla każdej wartości zakłócenia  $F_A$  można wyznaczyć z równania (1.12) wartość temperatury maksymalizującą wartość stężenia  $C_B$ . Na rysunku 1.12 zaznaczono linią kreskowaną krzywą optymalnej temperatury, tzn. następującą krzywą punktów maksymalizujących funkcję (1.12):

$$\hat{T}\left(\frac{W_m}{F_A}\right) = \arg \max_T C_B\left(T, \frac{W_m}{F_A}\right).$$

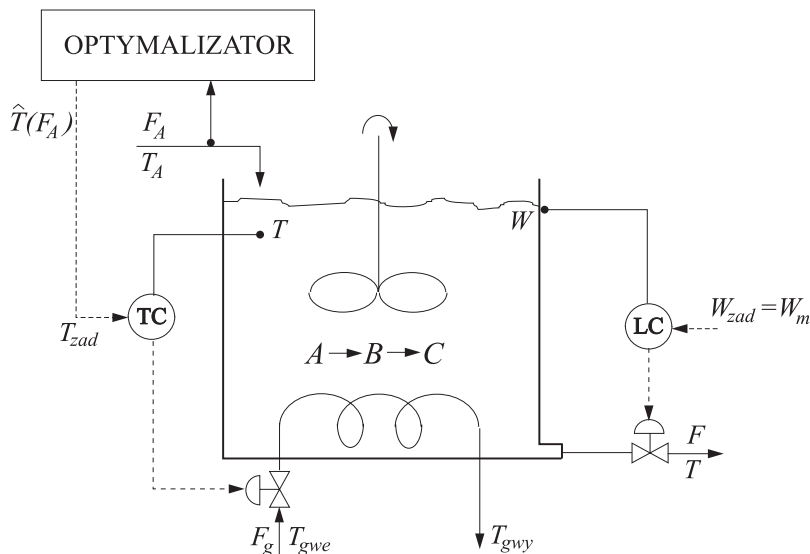


Rysunek 1.12. Zbiory poziomicowe powierzchni  $C_B(T, \frac{W_m}{F})$ , z zaznaczoną linią przerywaną zbiorem punktów maksymalizujących stężenie (optymalnych temperatur)

Zadaniem warstwy optymalizacji jest bieżące wyznaczanie punktu tej krzywej odpowiadającego aktualnej wartości zmiennej wejściowej niesterowanej  $F_A$ . Optymalizator może zadanie to realizować drogą bieżącej, wykonywanej za każdym razem optymalizacji. Ze względu na prostotę zadania można jednakże postąpić inaczej: wyznaczyć najpierw całą krzywą  $\hat{T}(\frac{W_m}{F_A})$ , przeprowadzając optymalizacje dla odpowiednio gęsto wybranych wartości z zakresu zmienności  $F_A$  i zapamiętać optymalną krzywą w pamięci optymalizatora. Struktura sterowania reaktorem realizująca cel sterowania (b) przedstawiona została na rysunku 1.13.

## 1.6. Nadzór, diagnostyka, adaptacja

W klasycznych już obecnie pracach poświęconych sterowaniu warstwowemu obiektami przemysłowymi i złożonymi systemami wyróżniano, przy dekompozycji funkcjonalnej zadania sterowania, trzy następujące podsta-

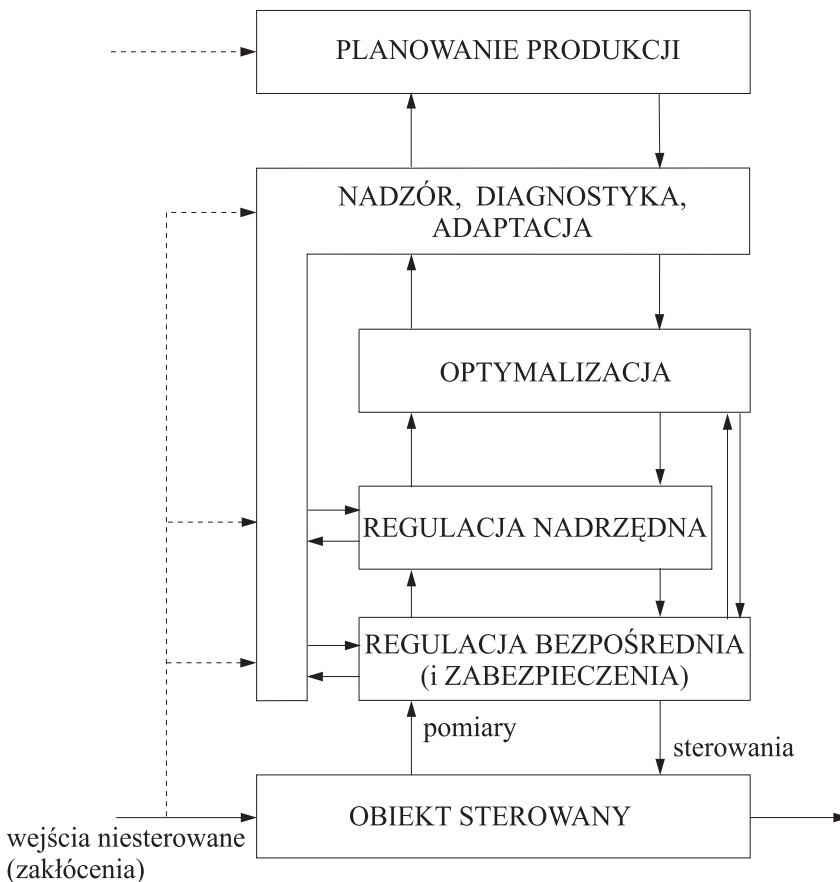


Rysunek 1.13. Warstwowa struktura sterowania reaktorem realizująca bieżącą optymalizację punktu pracy

wowe warstwy: regulacji (stabilizacji), optymalizacji i adaptacji, zob. np. [40]. Prace te stanowiły podsumowanie doświadczeń lat sześćdziesiątych i siedemdziesiątych, kiedy w układach regulacji dominowały jednopętlowe regulatory mechaniczne (pneumatyczne) i elektroniczne regulatory analogowe. Natomiast drogie i dość jeszcze zawodne w owych czasach komputery wprowadzano do sterowania jako elementy realizacji zadań warstw wyższych, głównie do zbierania, archiwizacji i obróbki danych dla celów wspomaganie decyzji operatorów i kadry inżynierskiej, decyzji o różnym stopniu zautomatyzowania. Warstwa adaptacji odpowiadała wydzieleniu podstawowych funkcji nadzoru i adaptacji w centrum sterowania i wspomaganie decyzji, realizującym z reguły również zadania związane z optymalizacją. W centrum tym zapadały, na podstawie analizy aktualnej sytuacji procesowej i wymagań zewnętrznych, decyzje o zmianach punktów pracy procesu, sposobach realizacji wymagań dotyczących jakości, reakcji na zjawiska nieprzewidziane typu awarii itp. – przekazywane następnie do niższych warstw regulacji i optymalizacji, w warstwie regulacji realizowane nierzadko ręcznie przez operatorów.

Rozwój mikroelektroniki i techniki komputerowej doprowadził do radykalnych zmian w sprzęcie automatyki. W związku z lawinowo postępującą miniaturyzacją, z pojawieniem się układów scalonych dużej i bardzo dużej skali integracji nie tylko kolosalnie wzrosły możliwości obliczeniowe, ale

i spadły wymiary oraz ceny sprzętu elektronicznego, mikroprocesorowego. Najważniejsza z punktu widzenia zastosowań do sterowania była jednakże równie radykalna poprawa niezawodności sprzętu. Dlatego też wraz z pojawieniem się regulatorów mikroprocesorowych i następnie przemysłowych rozproszonych systemów sterowania typu DCS nastąpiła nowa epoka w zakresie techniki regulacji, techniki sterowania obiektami przemysłowymi. W każdej z funkcjonalnych i sprzętowych warstw układu sterowania, a nawet w pojedynczej pętli regulacji bezpośredniej istnieje obecnie możliwość instalowania bardziej złożonych algorytmów, w tym algorytmów nieliniowych i algorytmów adaptacyjnych z automatyczną adaptacją do założonych wymagań. Obok algorytmów regulacji można w tych samych lokalnych ste-



Rysunek 1.14. Warstwowa struktura sterowania uwzględniająca zadania nadzoru, diagnostyki i adaptacji

rownikach realizować zadania diagnostyki zarówno sygnałów pomiarowych jak i poprawności działania samych regulatorów, z automatycznym przełączaniem na regulatory redundantne. Stąd zadania nadzoru i diagnostyki oraz adaptacji ulegają w znacznym stopniu rozproszeniu. Nie można już mówić o jednej, nadrzędnej wobec warstwy optymalizacji warstwie nadzoru, diagnostyki i adaptacji. Obok nadal istniejących centralnych zadań nadzoru i diagnostyki oraz zadania adaptacji struktury czy parametrów warstwy optymalizacji mamy zadania nadzoru i adaptacji układów regulacji bezpośredniej czy nadrzędnej, które w znacznej części realizowane są lokalnie, jako algorytmy lokalne mniej lub bardziej zintegrowane z algorytmami regulacji i zabezpieczeń. Taka struktura sterowania zapewnia bowiem większą szybkość działania i niezawodność, zgodnie z ogólną zasadą skracania pętli informacyjno-decyzyjnych. Warstwowa struktura sterowania uwzględniająca współczesne realizacje zadań nadzoru, diagnostyki i adaptacji przedstawiona jest na rysunku 1.14 (por. rys. 1.2).

Zagadnienia diagnostyki nie są omawiane w niniejszej pracy, zainteresowanemu czytelnikowi polecamy wyczerpujące, napisane na wysokim poziomie książki [65, 68, 66, 67].

## 1.7. Wnioski

Rosnąca konkurencja, globalizacja rynku produktów przemysłowych wymusza zwiększanie efektywności produkcji. Oznacza to coraz powszechniejsze stosowanie bieżącej optymalizacji sterowania w strukturze warstwowej. W następnych rozdziałach książki omawiane są algorytmy dla takich właśnie zastosowań. W rozdziale 4 przedstawione są algorytmy bieżącej optymalizacji punktu pracy obiektu. Natomiast w rozdziałach 2 i 3 omawia się algorytmy regulacji nadające się dobrze, zdaniem autora, do stosowania w warstwach regulacji bezpośredniej i nadrzędnej pracujących w warunkach optymalizacji bieżącej.

Podczas takiej optymalizacji wartości zadane dla regulatorów mogą być zmieniane w dość szerokim zakresie, zależnym od zmienności wejść niesterowanych i wymaganych asortymentów, cech produktów. Ponieważ obiekty sterowane są z reguły nieliniowe, to układy regulacji powinny być również nieliniowe, zapewniając dobrą jakość regulacji w różnych punktach pracy i w czasie procesów przejściowych przy ich zmianach. Dlatego też w rozdziale 2 książki omawiane są nieliniowe układy regulacji, poświęcona jest im też znacząca część rozważań rozdziału 3.

Przy większych zmianach punktów pracy (czy też silniejszych zakłóceniach) dobra jakość regulacji wymaga na ogół stosowania maksy-

malnych możliwych amplitud czy szybkości zmian sygnałów sterujących, a więc pracy na ograniczeniach. Ponadto, optymalny punkt pracy położony jest często na ograniczeniach określonych wielkości wyjściowych obiektu. Wynika stąd potrzeba stosowania algorytmów regulacji pozwalających uwzględniać ograniczenia. Jedyne algorytmy predykcyjne spełniają powyższe wymagania, nadają się przy tym świetnie do regulacji wielowymiarowej, w tym przy różnej ilości wejść sterujących i wyjść regulowanych. Stąd też burzliwy ich rozwój w ostatnich latach oraz olbrzymi sukces aplikacyjny, komercyjny. Omówieniu algorytmów predykcyjnych z liniowymi i nieliniowymi modelami obiektów poświęcony jest obszerny rozdział 3 książki.

## Rozdział 2

# Nieliniowa regulacja rozmyta

Zadaniem układów regulacji jest utrzymywanie zmiennych regulowanych jak najbliższe wartości dla nich zadanych, określających punkt pracy obiektu sterowanego. Klasycznym podejściem przy doborze algorytmu regulacji jest wyznaczenie liniowego modelu obiektu w danym punkcie pracy, a następnie, dla tego modelu, zaprojektowanie regulatora liniowego czy też dobór nastaw regulatora liniowego standardowego, z reguły typu PID, zob. np. [47, 90, 3, 54]. Ponieważ liniowy model obiektu jest tylko jego przybliżonym opisem oraz wobec możliwości zmian cech samego obiektu w trakcie sterowania powodowanych wpływem niesterowanych wielkości wejściowych (zakłóceń) czy zmianami jego cech wewnętrznych, projektuje się regulatory odpowiednio niezawodne, odporne. Takie regulatory powinny zapewniać stabilność układu regulacji i zachowanie podstawowych cech poprawnej regulacji przy określonym, dość znacznym zakresie różnic między obiektem a jego modelem użytym do projektowania. Przykładowo, przy klasycznej syntezy w dziedzinie częstotliwości zakłada się zachowanie określonych wartości zapasów modułu i fazy wyznaczanych na podstawie charakterystyk częstotliwościowych modelu obiektu wraz z regulatorem, wartości wynikających przede wszystkim z wieloletnich doświadczeń praktycznych i przyjmowanego kompromisu między odpornością na zmiany cech obiektu a jakością regulacji. Podobna filozofia stosowana jest przy projektowaniu regulatorów innymi metodami, takimi jak metoda linii pierwiastkowych (*root locus*), metoda lokowania biegunów układu zamkniętego (*pole placement*), eksperyment Zieglera-Nicholsa, czy popularne w dziedzinie sterowania procesami przemysłowymi metody doboru nastaw regulatorów PID z wykorzystaniem prostego modelowania obiektu na podstawie odpowiedzi skokowej i następnie tabel doboru nastaw, zob. np. [41, 47, 90]. Projektowanie takie zapewnia z reguły poprawną pracę układu regulacji w określonym otoczeniu punktu pracy, dla którego regulator był wyznaczany. Im mniejsze nieliniowości obiektu, tym otoczenie to jest w ogólności większe.

Występujące w praktyce obiekty sterowania są jednakże z reguły nieliniowe, nierzadko silnie nieliniowe. Jeśli pracują w niewielkim otoczeniu

punktu pracy i wymagania co do jakości regulacji nie są wygórowane, to przedstawiona wyżej filozofia projektowania wystarcza. Tak było i nierzadko po prostu musiało być w czasach konstrukcji regulatorów w technice mechanicznej (np. pneumatycznych), jak i nawet elektronicznej analogowej. Wprowadzenie do praktyki sterowania komputerów sterujących i dostatecznie niezawodnych regulatorów mikroprocesorowych radykalnie zmieniło sytuację. Nadrzędne komputery sterujące umożliwiły efektywną realizację bieżącej (*on-line*) optymalizacji biegu procesu sterowanego, tzn. odpowiedniego dopasowywania punktów pracy obiektu do aktualnej sytuacji technologicznej określanej wartościami niesterowanych wejść obiektu i wymagań warstwy zarządzania. Z kolei regulatory mikroprocesorowe pozwalają na realizację nawet bardzo zaawansowanych obliczeniowo algorytmów regulacji, w tym regulacji wielowymiarowej i regulacji nieliniowej. Trzeba podkreślić, że oba wymienione zjawiska silnie warunkują się wzajemnie – bieżąca optymalizacja biegu procesu zakłada nierzadko częste, a zawsze szybkie i niezawodne (a więc automatyczne) zmiany punktów pracy obiektu, realizowane poprzez zmiany wartości zadanych dla regulatorów. Nie jest to możliwe bez zastosowania regulatorów zdolnych dobrze i niezawodnie regulować nie tylko w otoczeniu jednego, ustalonego punktu pracy, ale w szerokim zakresie zmienności wielkości wejściowych i wyjściowych obiektu, a więc zdolnych regulować obiekt nieliniowy.

Najogólniej rzecz ujmując, rozwój algorytmów regulacji poszedł w takiej sytuacji w dwóch kierunkach: regulacji adaptacyjnej i regulacji nieliniowej. Istotą *regulacji adaptacyjnej* jest dopasowywanie na bieżąco parametrów regulatora, z reguły standardowego regulatora liniowego (np. PID), do zmieniających się cech obiektu. Adaptację prowadzi się w sposób bezpośredni czy też pośredni, poprzez bieżącą identyfikację liniowego modelu obiektu odpowiadającego aktualnemu obszarowi pracy i dobór parametrów regulatora, z reguły metodą klasyczną jak wspomniane wyżej, zob. np. [2, 109]. Podejście takie odpowiednie jest przede wszystkim w sytuacjach, kiedy nie jesteśmy w stanie a priori zaprojektować regulatora nieliniowego, nie jesteśmy w stanie uniknąć konieczności przeprowadzania bieżącej identyfikacji w trakcie pracy układu regulacji. Bieżąca identyfikacja niesie bowiem pewne ryzyko zawodności działania, szczególnie w okresach małej zmienności wielkości mierzonych. Dlatego też w zakresie regulacji przemysłowej, obiektów typu chemicznego, petrochemicznego, spożywczego, itp. regulacja adaptacyjna w wyżej omówionym sensie znalazła jak dotąd dość ograniczone zastosowanie.

Istotą *regulacji nieliniowej* (nieadaptacyjnej) jest natomiast projektowanie regulatora od razu dla obiektu modelowanego w szerokim zakresie

zmian wielkości wejściowych i wyjściowych, modelowanego jako obiekt nieliniowy – a więc projektowanie regulatora nieliniowego. Teoria regulacji nieliniowej jest bardzo bogata, zob. np. [57, 64] i nie jest celem tej pracy jej omawianie. W rozdziale niniejszym będziemy się zajmować projektowaniem i analizą regulatora nieliniowego w oparciu o teorię zbiorów rozmytych i logikę rozmytą, w szczególności modele rozmyte o tzw. strukturze Takagi-Sugeno, zaproponowane w roku 1985 [136].

Pomysł, definicja zbiorów rozmytych (*fuzzy sets*) i logiki rozmytej (*fuzzy logic*) zaproponowane zostały przez L. A. Zadeha już w roku 1964 [163], spotykając się początkowo z dużym krytycyzmem. Mimo iż pierwsze udane zastosowanie przemysłowe do celów sterowania powstało w roku 1976 (w duńskiej cementowni), to prawdziwy boom w pracach nad teorią i zastosowaniami logiki rozmytej do sterowania nastąpił w końcu lat osiemdziesiątych i trwa w zasadzie do dzisiaj. Okazało się, że regulatory rozmyte mogą być silnym, skutecznym narzędziem przede wszystkim tam, gdzie trudno o pełny analityczny opis obiektu sterowanego. Dysponujemy natomiast wiedzą empiryczną, doświadczeniem operatorów sterujących mniej lub bardziej skutecznie takimi obiektami, czy też wzorcami pożądanego i realizowalnego zachowania się obiektów sterowanych. Z kolei modelowanie rozmyte obiektów nieliniowych okazało się bardzo skutecznym narzędziem pozwalającym efektywnie wykorzystywać w podejściu analitycznym elementy wiedzy jakościowej o obiekcie. Modelowanie rozmyte obiektów nieliniowych do celów sterowania okazało się, obok modelowania sieciami neuronowymi, podejściem najintensywniej rozwijanym i praktycznie stosowanym poczynając od dekady lat dziewięćdziesiątych ub. stulecia.

Nie zamierzamy w niniejszej pracy omawiać podstaw teorii zbiorów rozmytych i logiki rozmytej oraz możliwych zastosowań do celów sterowania, istnieje tu bardzo wiele często doskonałych pozycji książkowych, jak np. [162, 116], gdzie znaleźć też można bardzo obszerną literaturę przedmiotu. Podstawy teorii zbiorów rozmytych i logiki rozmytej przytoczymy w punkcie następnym, jedynie w zakresie potrzebnym dla wprowadzenia modelowania rozmytego o strukturze Takagi-Sugeno. Przedstawimy następnie projektowanie i analizę stabilności *regulatorów nieliniowych rozmytych o strukturze Takagi-Sugeno*, nazywanych również regulatorami rozmytymi *wieloobszarowymi* [32, 33], czy też *wielomodelowymi* [21]. Zdaniem nie tylko autora niniejszej książki jest to jedna z najbardziej udanych konstrukcji regulatorów nieliniowych, szczególnie z praktycznego, aplikacyjnego punktu widzenia.

## 2.1. Systemy rozmyte typu TS

### 2.1.1. Zbiory rozmyte i zmienne lingwistyczne

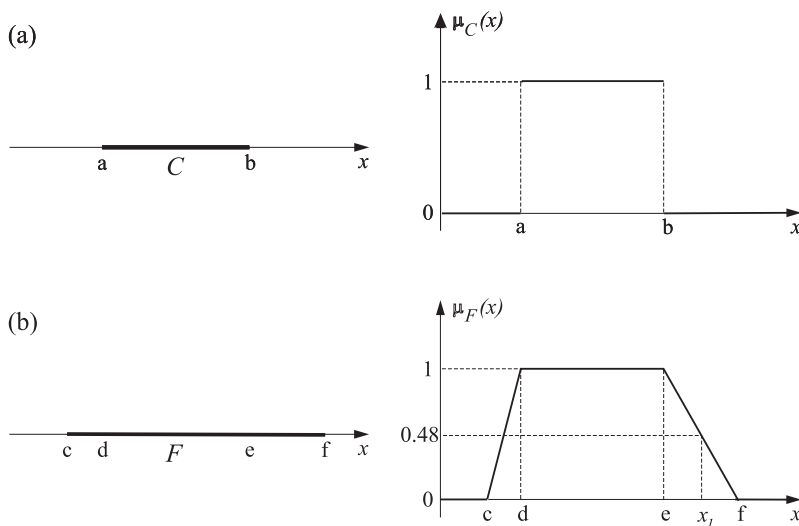
Na rysunku 2.1 b) przedstawiono sposób określania przynależności elementu  $x$  osi liczbowej  $\mathbb{R}^1$  do przykładowego zbioru rozmytego  $F$ . Natomiast na rysunku 2.1 a) pokazano, dla porównania, przynależność elementu  $x$  osi liczbowej  $\mathbb{R}^1$  do zbioru  $C$  definiowanego w sposób klasyczny, dla odróżnienia w teorii zbiorów rozmytych nazywanego zbiorem ostrym (*crisp set*). Każdy z elementów osi liczbowej należy do zbioru  $C$  lub nie, funkcja przynależności  $\mu_C(x)$  zbioru  $C$  może przyjmować jedynie wartości 0 lub 1,

$$\mu_C(x) = \begin{cases} 1, & \text{jeśli } x \in C, \\ 0, & \text{jeśli } x \notin C. \end{cases}$$

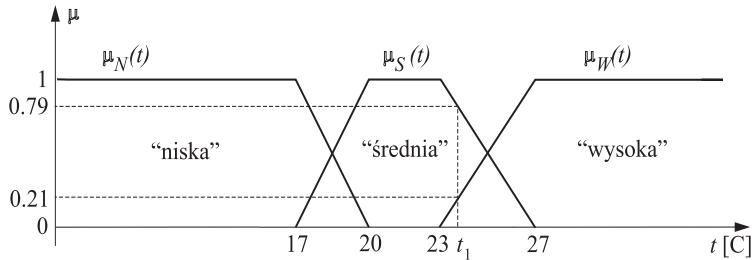
Natomiast funkcja przynależności  $\mu_F(x)$  zbioru rozmytego  $F$  może przyjmować również każdą z wartości pomiędzy 0 a 1,

$$\mu_F(x) = \begin{cases} \in (0, 1], & \text{jeśli } x \in F, \\ 0, & \text{jeśli } x \notin F. \end{cases}$$

Na rysunku 2.1 pokazano przykładową funkcję przynależności  $\mu_F(x)$  zbioru rozmytego  $F$  o kształcie trapezowym. Każdy z punktów odcinka  $[d, e]$  należy w pełni do zbioru  $F$ , tzn. z wartością funkcji przynależności równą 1,



Rysunek 2.1. (a) Zbiór ostry  $C$  na osi liczbowej, w porównaniu z (b) przykładowym zbiorem rozmytym  $F$ ;  $\mu_C(x)$ ,  $\mu_F(x)$  – funkcje przynależności zbiorów  $C$  i  $F$



Rysunek 2.2. Ilustracja zmiennej lingwistycznej „temperatura” przyjmującej trzy wartości: „niska”, „średnia” i „wysoka”

podobnie jak każdy z punktów odcinka  $[a, b]$  należy do ostrego zbioru  $C$ . Natomiast punkty odcinków  $(c, d)$  i  $(e, f)$  „nie w pełni” należą do zbioru  $F$ , albowiem odpowiadające im wartości funkcji przynależności są zawarte w przedziale  $(0, 1)$ . W tym sensie granice zbioru  $F$  są *rozmyte*, stąd nazwa *zbiór rozmyty*. Dla przykładu, pokazany na rysunku 2.1 b) punkt  $x_1$  należy do zbioru  $F$  z wartością funkcji przynależności 0.48, mówimy też krócej: ze stopniem przynależności (*degree of membership*) 0.48.

Zdefiniowanie zbioru rozmytego pozwala wprowadzić kolejne pojęcie kluczowe dla logiki rozmytej, pojęcie *zmiennej lingwistycznej* (*linguistic variable*). Zmienna lingwistyczna, zwana też rozmytą, jest pośrednią między zmienną numeryczną a zmienną symboliczną, której wartości są symbolami – np. zmienna symboliczna „kształt” zdefiniowana jako przyjmująca 3 wartości: „koło”, „kwadrat”, „trójkąt”. Pojęcie zmiennej lingwistycznej wyjaśnia przykład pokazanej na rysunku 2.2 zmiennej „temperatura”, przyjmującej wartości: „niska”, „średnia” i „wysoka”, zdefiniowane zbiorami rozmytymi określonymi odpowiednio przez funkcje przynależności  $\mu_N(t)$ ,  $\mu_S(t)$ , i  $\mu_W(t)$ . Dla przykładu, pokazana na rysunku temperatura  $t_1$  jest średnia ze stopniem przynależności 0.79 (z wartością funkcji przynależności 0.79) i jednocześnie wysoka ze stopniem przynależności 0.21.

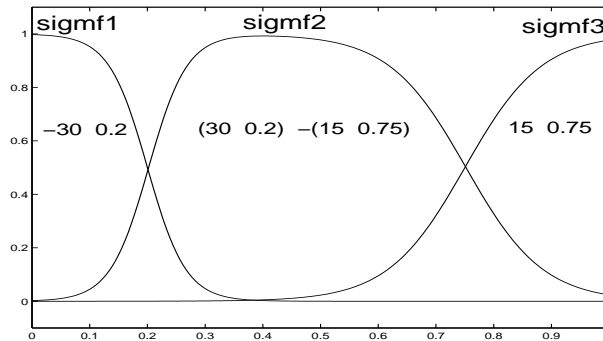
Na rysunkach 2.1 i 2.2 zastosowano zbiory rozmyte z *trapezowymi funkcjami przynależności*. Łącznie z *funkcjami trójkątnymi* (będącymi szczególnym przypadkiem funkcji trapezowych, jak na rysunku 2.1 przy  $d = e$ ) są one najpopularniejszymi funkcjami przynależności w tych zastosowaniach, gdzie nie jest potrzebna gładkość funkcji. Jeśli różniczkowalność funkcji przynależności jest potrzebna, jak np. przy projektowaniu systemów rozmytych jako sieci neuronowych (*neuro-fuzzy systems*, zob. rozdz. 2.1.4), to najpopularniejsze w zastosowaniach są funkcje sigmoidalne, również dzwonowe (dzwonopodobne) lub gaussowskie, zob. np. [112, 59, 162, 116].

Funkcja przynależności (do zbioru  $X_i$ ) *sigmoidalna jednostronna* jest definiowana wzorem

$$\mu_{X_i}(x) = \frac{1}{1 + \exp[-\alpha_i(x - c_i)]}, \quad (2.1)$$

gdzie parametry  $\alpha_i$  i  $c_i$  określają kształt zbioru rozmytego  $X_i$ . Dla  $\alpha_i < 0$  funkcja jest lewostronnie otwarta, a dla  $\alpha_i > 0$  prawostronnie otwarta. Wartość  $c_i$  określa położenie funkcji ( $\mu_{X_i}(c_i) = 0.5$ ), zaś  $\alpha_i$  stromość jej zbocza. Funkcje sigmoidalne jednostronne mogą definiować jedynie krańcowe zbiory rozmyte. Natomiast z dwóch funkcji przynależności sigmoidalnych jednostronnych skonstruować można funkcję dwustronną (zamkniętą), mogącą reprezentować wartości pośrednie, o wartości dążącej do zera przy  $|x| \rightarrow \infty$ . Można to uczynić na dwa proste sposoby: przez odjęcie od siebie dwóch funkcji prawostronnie (lub lewostronnie) otwartych odpowiednio położonych względem siebie lub przez przemnożenie funkcji prawostronnie otwartej przez położoną na prawo od niej funkcję lewostronnie otwartą (zob. np. *Fuzzy Logic Toolbox* pakietu MATLAB). Na rysunku 2.3 przedstawiono trzy sigmoidalne funkcje przynależności:

- sigmf1: lewostronnie otwartą z parametrami  $\alpha_i = -30$ ,  $c_i = 0.2$ ;
- sigmf2: dwustronną powstałą z odjęcia funkcji sigmf3 od funkcji prawostronnie otwartej z parametrami  $\alpha_i = 30$ ,  $c_i = 0.2$ ;
- sigmf3: prawostronnie otwartą z parametrami  $\alpha_i = 15$ ,  $c_i = 0.75$ .



Rysunek 2.3. Przykładowe sigmoidalne funkcje przynależności

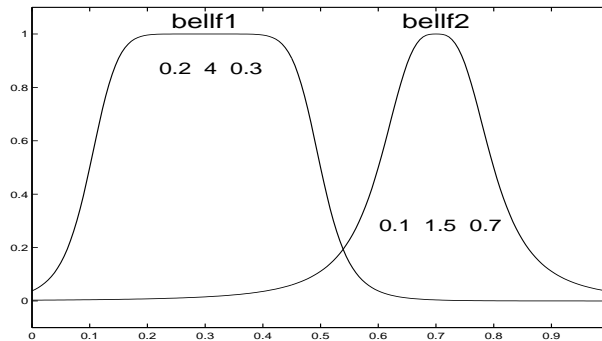
Popularność funkcji sigmoidalnych wynika nie tylko z faktu, że łatwo konstruować funkcje zarówno jednostronne, jak i dwustronne, ale również z łatwości uzyskania cech asymetrii funkcji dwustronnych – parametry lewego i prawego zbocza mogą być zupełnie różne, jako parametry składowych funkcji jednostronnych, a jednocześnie wielkość obszaru środkowego (o wartości funkcji zbliżonej do jedności) może być swobodnie kształtowana.

Funkcja przynależności dzwonowa uogólniona (*generalized bell function*, zob. np. [112, 59, 162]) definiowana jest w postaci

$$\mu_{X_i}(x) = \frac{1}{1 + \left[\left(\frac{x-c_i}{\alpha_i}\right)^2\right]^{\beta_i}}, \quad (2.2)$$

gdzie  $\alpha_i, \beta_i, c_i$  są parametrami określającymi kształt zbioru rozmytego  $X_i$ . W szczególności,  $c_i$  definiuje środek symetrii,  $\beta_i$  wpływa głównie na nachylenie zboczy, a  $\alpha_i$  przede wszystkim na szerokość „dzwonu”. Na rysunku 2.4 pokazano dwie przykładowe funkcje dzwonowe:

- bellf1: funkcja o parametrach  $\alpha_i = 0.2$ ,  $\beta_i = 4$ ,  $c_i = 0.3$ ;
- bellf2: funkcja o parametrach  $\alpha_i = 0.1$ ,  $\beta_i = 1.5$ ,  $c_i = 0.7$ .



Rysunek 2.4. Przykładowe funkcje przynależności uogólnione dzwonowe

Zwróćmy uwagę, że funkcja dzwonowa jest symetryczna względem osi pionowej  $x = c_i$ . Chcąc uzyskać funkcję dwustronną niesymetryczną należy ją konstruować z połówek funkcji dzwonowych o różnych parametrach. W literaturze oraz pakietach oprogramowania występują również z reguły funkcje gaussowskie, zob. np. [112, 59, 162, 116].

Zmienne lingwistyczne (rozmyte) pozwalają ujmować zjawiska nieprecyzyjnie zdefiniowane lepiej niż zmienne numeryczne (zwane ostrymi – *crisp*). Dla przykładu, przedstawione na rysunku 2.2 określenie zmiennej lingwistycznej „temperatura” może wynikać z sondażu przeprowadzonego wśród reprezentacyjnej próbki klientów lub użytkowników budynku, np. w celu określenia warunków włączania urządzeń grzewczych (gdy temperatura jest niska) bądź klimatyzacyjnych (gdy temperatura jest wysoka).

Dla konkretnej numerycznej wartości zmiennej proces określania jej wartości lingwistycznej łącznie z odpowiednią wartością funkcji przynależności nazywany jest procesem *rozmywania* (fuzyfikacji, *fuzzyfication*). W przykładzie na rysunku 2.2 mamy trzy wartości lingwistyczne dla zmiennej

temperatura: „niska”, „średnia” i „wysoka”, zaś wartości numerycznej  $t_1$  zmiennej temperatura zostają przyporządkowane, w procesie rozmywania, wartości lingwistyczne „średnia” i „wysoka”, z wartościami funkcji przynależności odpowiednio 0.79 i 0.21. Podkreślmy, że wartość lingwistyczna definiowana jest przez funkcję przynależności do tej wartości, tzn. tożsama jest z definiowanym przez funkcję przynależności zbiorem rozmytym.

### 2.1.2. Wnioskowanie rozmyte

Podstawowym elementem systemów rozmytych (*fuzzy systems*), wykorzystywanych do modelowania, sterowania itd., jest *zbiór reguł* zwany też *bazą wiedzy* (jak w systemach ekspertowych). Składa się on z reguł wnioskowania operujących na zmiennych rozmytych (lingwistycznych), stąd też nazwa *reguły rozmyte* (*fuzzy rules*).

Każda reguła wnioskowania składa się z dwóch elementów: części „JEŚLI” („IF-part”) zwanej poprzednikiem reguły (*antecedent*) i części „TO” („THEN-part”) zwanej następnikiem reguły (*consequent*). Struktura pojedynczej reguły może być więc przedstawiona w postaci:

JEŚLI <poprzednik> TO <następnik>.

Poprzednik określa warunek, a następnik konkluzję, jaka będzie realizowana, jeśli warunek jest prawdziwy.

Poprzednik rozmytej reguły wnioskowania zawiera w najprostszym przypadku pojedynczy warunek, wówczas reguła jest postaci

JEŚLI  $x$  jest  $A$  TO <następnik>,

gdzie  $x$  jest zmienną lingwistyczną, zaś  $A$  zbiorem rozmytym określonym funkcją przynależności  $\mu_A(x)$ . W przypadku ogólnym poprzednik reguły może zawierać wiele prostych warunków powiązanych wzajemnie operatorami koniunkcji (i), dysjunkcji (lub) oraz negacji (nie), np.

JEŚLI  $x_1$  jest  $A_1$  i  $x_2$  jest (nie $A_2$ )  $\cdots$  lub  $x_k$  jest  $A_k \cdots$   
TO <następnik>.

Następnik rozmytej reguły wnioskowania może natomiast, w ogólności, przyjmować jedną z następujących postaci:

1. Następnik *ostry* (*crisp consequent*):

JEŚLI <poprzednik > TO  $y = y_a$ ,

gdzie  $y_a$  jest wartością numeryczną (liczbą) lub symboliczną,

2. Następnik *rozmyty* (*fuzzy consequent*):

JEŚLI <poprzednik > TO  $y$  jest  $Y_k$ ,

gdzie  $y$  jest zmienną rozmytą (lingwistyczną), a  $Y_k$  zbiorem rozmytym,

3. Następnik *funkcyjny* (*functional consequent*):

JEŚLI  $x_1$  jest  $A_1$  i  $x_2$  jest  $A_2$   $\cdots$  i  $x_n$  jest  $A_n$

TO  $y = f(x_1, x_2, \dots, x_n)$ ,

gdzie  $f$  jest pewną funkcją zmiennych  $x_1, x_2, \dots, x_n$ .

W książce niniejszej interesować nas będą jedynie następniki funkcyjne, zaproponowane w roku 1985 przez Takagi i Sugeno [136]. Okazało się, że wykorzystanie reguł wnioskowania z następnikami funkcyjnymi umożliwia efektywne modelowanie zależności nieliniowych z użyciem niewielkiej liczby reguł. Systemy rozmyte wykorzystujące reguły z następnikami funkcyjnymi nazywane są, od nazwisk autorów, *systemami rozmytymi Takagi-Sugeno* (*Takagi-Sugeno fuzzy systems*, lub krócej *TS fuzzy systems*). Spotyka się też, choć rzadziej, określenie systemy Takagi-Sugeno-Kanga (TSK), gdyż Kang był kolejnym z młodych współpracowników prof. Sugeno zajmującym się systemami rozmytymi o takiej strukturze. Spotkać też można, chociaż rzadko, określenie „systemy wielomodelowe” (*multiple model systems*, zob. np. [21]). Zwróćmy uwagę, że następnik funkcyjny redukuje się do ostrego (numerycznego), gdy funkcja  $f$  przyjmuje krańcową postać stałej,  $f(x_1, x_2, \dots, x_n) = a_0$ . Najczęściej stosuje się następniki funkcyjne w postaci wielomianów pierwszego rzędu

$$f(x_1, x_2, \dots, x_n) = a_0 + \sum_{i=1}^n a_i x_i,$$

gdzie  $a_0, a_1, \dots, a_n$  są parametrami. Powody są dwa: po pierwsze i najważniejsze, użycie jedynie liniowych funkcji następników wystarcza do zadowalająco dokładnego modelowania nawet silnie nieliniowych zależności, przy odpowiednio dobranym położeniu i z reguły niezbyt dużej ilości zbiorów rozmytych poprzedników reguł. Po drugie, istnieje szereg prostych, dobrze opanowanych metod konstrukcji i identyfikacji modeli liniowych czy projektowania liniowych regulatorów, które można bezpośrednio zastosować w poszczególnych podobszarach wydzielonych poprzednikami reguł systemu TS. Dlatego też w pracy niniejszej będziemy wykorzystywać jedynie liniowe funkcje następników reguł, chociaż prezentowane struktury modelowania i regulacji można również zastosować w przypadku bardziej ogólnym.

Przypominamy, że zbiór wszystkich reguł wnioskowania systemu rozmytego, np. rozmytego modelu obiektu nieliniowego, nazywany jest *bazą wiedzy* tego systemu. Będziemy przyjmować, że baza wiedzy systemu rozmytego TS składa się z  $r$  reguł następującej postaci

$$\begin{aligned} R^i: \text{ JEŚLI } x_1 \text{ jest } A_1^i \text{ i } x_2 \text{ jest } A_2^i \cdots \text{ i } x_n \text{ jest } A_n^i \\ \text{ TO } y^i = a_0^i + \sum_{j=1}^n a_j^i x_j, \end{aligned} \quad (2.3)$$

$i = 1, \dots, r$ . Każdy ze zbiorów rozmytych  $A_j^i$  w poprzedniku reguły  $R^i$  jest jednym z elementów zbioru wartości lingwistycznych (zbiorów rozmytych) zmiennej  $x_j$

$$A_j^i \in \mathbb{X}_j = \{X_{j1}, X_{j2}, \dots, X_{jr_j}\}, \quad (2.4)$$

gdzie  $r_j$  jest liczbą elementów zbioru  $\mathbb{X}_j$ ,  $j = 1, \dots, n$ . Przedstawioną strukturę bazy wiedzy systemu rozmytego TS wyjaśnimy za pomocą przykładu.

### Przykład 2.1.

Rozważmy modelowanie systemem rozmytym typu TS dwuwymiarowej zależności nieliniowej  $y = f(x_1, x_2)$ , przy dwuelementowej reprezentacji każdej ze zmiennych lingwistycznych  $x_1$  i  $x_2$ ,

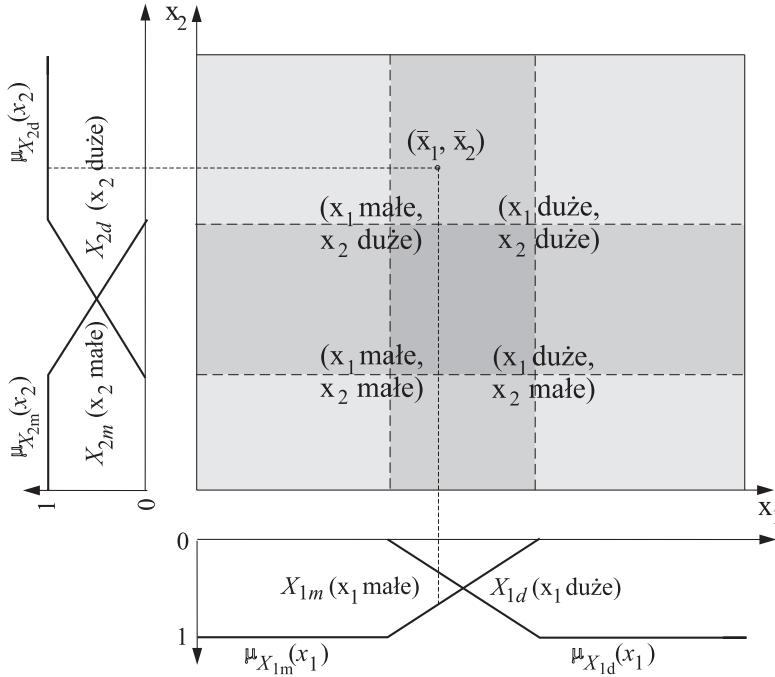
$$\begin{aligned} x_1 \in \{\text{„małe”}, \text{„duże”}\} &= \{X_{1m}, X_{1d}\}, \\ x_2 \in \{\text{„małe”}, \text{„duże”}\} &= \{X_{2m}, X_{2d}\}. \end{aligned}$$

Baza wiedzy będzie zbiorem czterech reguł  $R^i$ ,  $i = 1, 2, 3, 4$  :

$$\begin{aligned} R^1: \text{ JEŚLI } x_1 \text{ jest } X_{1m} \text{ i } x_2 \text{ jest } X_{2m} \quad \text{ TO } y = y^1 &= a_0^1 + a_1^1 x_1 + a_2^1 x_2, \\ R^2: \text{ JEŚLI } x_1 \text{ jest } X_{1m} \text{ i } x_2 \text{ jest } X_{2d} \quad \text{ TO } y = y^2 &= a_0^2 + a_1^2 x_1 + a_2^2 x_2, \\ R^3: \text{ JEŚLI } x_1 \text{ jest } X_{1d} \text{ i } x_2 \text{ jest } X_{2m} \quad \text{ TO } y = y^3 &= a_0^3 + a_1^3 x_1 + a_2^3 x_2, \\ R^4: \text{ JEŚLI } x_1 \text{ jest } X_{1d} \text{ i } x_2 \text{ jest } X_{2d} \quad \text{ TO } y = y^4 &= a_0^4 + a_1^4 x_1 + a_2^4 x_2, \end{aligned}$$

gdzie  $X_{1m}$ ,  $X_{2m}$ , i  $X_{1d}$ ,  $X_{2d}$  są zbiorami rozmytymi określającymi odpowiednio wartości „małe” i „duże” zmiennych lingwistycznych  $x_1$  i  $x_2$ , zaś następniki funkcyjne są liniowymi przybliżeniami modelowanej zależności w odpowiednich dwuwymiarowych obszarach (zbiorach) rozmytych. Przedstawiając sformułowane powyżej cztery reguły w ogólnej postaci (2.3) mamy

$$\begin{aligned} A_1^1 &= X_{1m}, & A_2^1 &= X_{2m}, \\ A_1^2 &= X_{1m}, & A_2^2 &= X_{2d}, \\ A_1^3 &= X_{1d}, & A_2^3 &= X_{2m}, \\ A_1^4 &= X_{1d}, & A_2^4 &= X_{2d}, \end{aligned}$$



Rysunek 2.5. Przykładowy podział dziediny dwuwymiarowej w przypadku dwu-elementowych zmiennych lingwistycznych  $x$  i  $y$

przy czym zbiory  $A_1^i$ ,  $i = 1, \dots, 4$ , są elementami zbioru (2.4) postaci

$$\mathbb{X}_1 = \{X_{11}, X_{12}\} = \{X_{1m}, X_{1d}\},$$

gdzie  $r_1 = 2$ , zaś zbiory  $A_2^i$ ,  $i = 1, \dots, 4$ , są elementami zbioru (2.4) postaci

$$\mathbb{X}_2 = \{X_{21}, X_{22}\} = \{X_{2m}, X_{2d}\},$$

gdzie  $r_2 = 2$ .

Przykładowa ilustracja omówionej dziediny modelowanej zależności przedstawiona jest na rysunku 2.5. Zaznaczony na rysunku punkt  $(\bar{x}_1, \bar{x}_2)$  należy, z niezerowymi wartościami funkcji przynależności, do zbiorów  $X_{1m} \times X_{2d}$  ( $x_1$  małe,  $x_2$  duże) i  $X_{1d} \times X_{2d}$  ( $x_1$  duże,  $x_2$  duże). W szczególności, współrzędna  $\bar{x}_1$  należy do zbioru  $X_{1m}$  z wartością funkcji przynależności  $\mu_{X_{1m}}(\bar{x}_1) = 0.62$  i do zbioru  $X_{1d}$  z wartością funkcji przynależności  $\mu_{X_{1d}}(\bar{x}_1) = 0.38$ . Natomiast współrzędna  $\bar{x}_2$  należy jedynie do zbioru  $X_{2d}$ , z wartością funkcji przynależności  $\mu_{X_{2d}}(\bar{x}_2) = 1.0$ .

Przedstawiony przykład ilustruje nie tylko zbiór reguł, pokazuje również, jak w naturalny sposób tworzone są iloczyny kartezjańskie zbiorów rozmytych.  $\square$

Zbiór reguł służy do wnioskowania rozmytego. W ogólności, *wnioskowanie rozmyte* składa się z trzech podstawowych etapów oraz, dla pewnych zastosowań, dodatkowo z etapu czwartego (zob. np. [162]):

1. Obliczenie *poziomów aktywacji* (*levels, degrees of activation*) wszystkich reguł, określanych również jako *sily odpalenia* reguł (*firing strengths*, zob. np. [115, 58]), odpowiadających aktualnym wartościom numerycznym zmiennych wejściowych.
2. Wyznaczenie *konkluzji* poszczególnych reguł.
3. *Połączenie konkluzji* wszystkich reguł w jedną konkluzję finalną – wniosek końcowy.

W ogólnym przypadku zmienna następników reguł jest zmienną rozmytą, stąd również konkluzja finalna ma postać rozmytą, tzn.  $y \in Y$ , gdzie  $y$  jest zmienną wyjściową, a  $Y$  zbiorem rozmytym powstałym w wyniku etapów 1, 2 i 3 wnioskowania rozmytego. Dla szeregu zastosowań, np. do sterowania, przekształcić jeszcze trzeba uzyskaną wartość rozmytą (tzn. zbiór rozmyty) zmiennej wyjściowej do postaci ostrej, numerycznej.

Wówczas wykonuje się:

4. *Wyostrowanie* (*defuzyfikację, defuzzification*) – przekształcenie rozmytej wartości zmiennej wyjściowej do wartości numerycznej.

Wnioskowanie rozmyte bardzo się upraszcza, jeśli zmienne wyjściowe następników wszystkich reguł nie są zmiennymi rozmytymi, są ostre lub funkcyjne. Ta właśnie sytuacja ma miejsce w przypadku systemów rozmytych typu TS, które są przedmiotem zainteresowania w niniejszej pracy. Dlatego ograniczymy się dalej jedynie do tego przypadku – odsyłając czytelnika zainteresowanego ogólniejszymi aspektami wnioskowania rozmytego do bogatej literatury przedmiotu, np. [115, 162, 116].

### Wnioskowanie rozmyte w strukturach TS

W przypadku struktur rozmytych typu TS *schemat wnioskowania rozmytego* można sformułować w następującej postaci:

1. Wyznaczenie *poziomów aktywacji* poszczególnych reguł, odpowiadających aktualnym wartościom numerycznym zmiennych wejściowych.
2. Obliczenie wartości następników funkcyjnych poszczególnych reguł.

3. Wyznaczenie *konkluzji finalnej* – ważone i normowane zsumowanie wartości zmiennej wyjściowej uzyskanych w następnikach reguł, z uwzględnieniem poziomów aktywacji poszczególnych reguł.

W rozważanym przypadku systemu TS każda z reguł  $R^i$  ma postać (2.3),  $i = 1, \dots, r$ , gdzie  $r$  jest liczbą reguł. *Obliczenie poziomu aktywacji  $w^i$*  reguły, nawet w rozważanym jednorodnym przypadku poprzednika zawierającego jedynie proste warunki, wszystkie połączone operatorem koniunkcji, nie jest operacją jednoznaczną. Powszechnie stosowane są tu operator mnożenia lub operator minimum. Przy zastosowaniu *operatora mnożenia*, dla wartości zmiennych wejściowych  $x = [x_1 \ x_2 \ \dots \ x_n]^T$  poziom aktywacji reguły  $i$ -tej wyznaczany jest *iloczynem algebraicznym*

$$w^i(x) = \mu_{A_1^i}(x) \cdot \mu_{A_2^i}(x) \cdot \dots \cdot \mu_{A_n^i}(x), \quad i = 1, \dots, r.$$

Natomiast przy zastosowaniu *operatora minimum* poziom aktywacji wyznaczany jest *iloczynem logicznym*

$$w^i(x) = \min\{\mu_{A_1^i}(x), \mu_{A_2^i}(x), \dots, \mu_{A_n^i}(x)\}, \quad i = 1, \dots, r.$$

W dwuwymiarowym przykładzie 2.1 przedstawionym w punkcie poprzednim, dla wartości  $x = \bar{x} = [\bar{x}_1 \ \bar{x}_2]^T$  wektora zmiennych wejściowych (zob. rys. 2.5) dostajemy następujące wartości poziomów aktywacji poszczególnych reguł, przy zastosowaniu operacji minimum:

$$\begin{aligned} w^1(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1m}}(\bar{x}_1) = 0.62, \mu_{X_{2m}}(\bar{x}_2) = 0\} = 0, \\ w^2(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1m}}(\bar{x}_1) = 0.62, \mu_{X_{2d}}(\bar{x}_2) = 1\} = 0.62, \\ w^3(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1d}}(\bar{x}_1) = 0.38, \mu_{X_{2m}}(\bar{x}_2) = 0\} = 0, \\ w^4(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1d}}(\bar{x}_1) = 0.38, \mu_{X_{2d}}(\bar{x}_2) = 1\} = 0.38. \end{aligned}$$

Wartościami następników poszczególnych reguł będą w tym przykładzie następujące wartości zmiennej wyjściowej:

$$\begin{aligned} R^1 : \quad \bar{y}^1 &= a_0^1 + a_1^1 \bar{x}_1 + a_2^1 \bar{x}_2, \\ R^2 : \quad \bar{y}^2 &= a_0^2 + a_1^2 \bar{x}_1 + a_2^2 \bar{x}_2, \\ R^3 : \quad \bar{y}^3 &= a_0^3 + a_1^3 \bar{x}_1 + a_2^3 \bar{x}_2, \\ R^4 : \quad \bar{y}^4 &= a_0^4 + a_1^4 \bar{x}_1 + a_2^4 \bar{x}_2. \end{aligned}$$

Zwróćmy uwagę, że w przykładzie powyższym zastosowanie operatora mnożenia prowadzi do takiego samego wyniku, co jest raczej przypadkiem szczególnym. Niezerowe poziomy aktywacji liczone operatorami mnożenia i minimum przyjmują inne wartości, jeśli co najmniej dwa stopnie przynależności występujące w danej regule są mniejsze od 1. Natomiast bezpośrednio

z definicji obu operatorów wynika, że zerowe poziomy aktywacji występują zawsze dla tych samych reguł.

Ostatnim etapem wnioskowania rozmytego jest ważone, normowane zsumowanie wartości następników poszczególnych reguł. Dla  $r$  reguł opisanych wzorami (2.3) sumowanie to wykonuje się zgodnie z wzorem

$$y = \frac{\sum_{i=1}^r w^i(x) \cdot y^i}{\sum_{l=1}^r w^l(x)} = \frac{\sum_{i=1}^r w^i(x)[a_0^i + \sum_{j=1}^n a_j^i x_j]}{\sum_{l=1}^r w^l(x)}, \quad (2.5)$$

gdzie  $y^i$  są wartościami następników poszczególnych reguł w punkcie  $x$ , a  $y$  jest konkluzją finalną – wartością zmiennej wyjściowej systemu rozmytego typu TS w punkcie  $x$ . Dla wartości  $\bar{x}$  wektora zmiennych wejściowych w rozważanym powyżej przykładzie 2.1 mamy

$$\begin{aligned} \bar{y} &= \frac{\sum_{i=1}^4 w^i(\bar{x}) \cdot \bar{y}^i}{\sum_{l=1}^4 w^l(\bar{x})} = \\ &= 0.62 \cdot (a_0^2 + a_1^2 \bar{x}_1 + a_2^2 \bar{x}_2) + 0.38 \cdot (a_0^4 + a_1^4 \bar{x}_1 + a_2^4 \bar{x}_2). \end{aligned}$$

Zwróćmy uwagę, że w przykładzie powyższym  $\sum_{l=1}^r w^l(\bar{x}) = 1$  dla każdej wartości  $x$ , co wynika z konstrukcji funkcji przynależności. W ogólności warunek ten nie musi być spełniony, stąd we wzorze (2.5) w mianowniku ułamka występuje normująca suma wartości poziomów aktywacji wszystkich reguł. Wartości

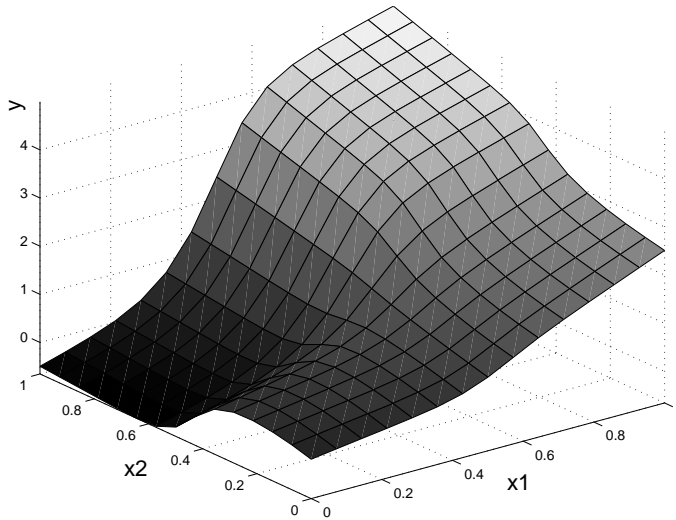
$$\tilde{w}^i(x) = \frac{w^i(x)}{\sum_{l=1}^r w^l(x)}, \quad i = 1, \dots, r, \quad (2.6)$$

nazywamy *unormowanymi poziomami aktywacji* reguł modelu rozmytego. Spełniają one zawsze warunek  $\sum_{i=1}^r \tilde{w}^i(x) = 1$ .

Z punktu widzenia relacji między zmiennymi wejściowymi  $x_1, x_2, \dots, x_n$  oraz zmienną wyjściową  $y$  system rozmyty TS można traktować jako pewne *odwzorowanie funkcyjne* o cechach zależnych od cech funkcji przynależności i funkcji następników reguł. Jeśli funkcje przynależności są funkcjami różniczkowalnymi, jak np. funkcje sigmoidalne, oraz funkcje następników reguł są też różniczkowalne (z reguły są liniowe), zaś poziomy aktywacji reguł wyznaczamy operatorem mnożenia, to odwzorowanie nieliniowe generowane modelem rozmytym TS też będzie odwzorowaniem ciągłym i różniczkowalnym. W przypadku dwuwymiarowym odwzorowanie systemu TS wygodnie jest przedstawić w formie graficznej, jako powierzchnię  $y = \varphi(x_1, x_2)$ .

### Przykład 2.2.

Rysunek 2.6 przedstawia powierzchnię przykładowego systemu rozmytego



Rysunek 2.6. Powierzchnia systemu rozmytego TS w przykładzie 2.2

TS o dwóch wejściach  $x_1$  i  $x_2$ , ze zbiorami rozmytymi opisanymi sigmoidalnymi funkcjami przynależności przedstawionymi na rysunkach 2.7 i 2.8 oraz następującą bazą reguł, o ogólnej strukturze (2.3):

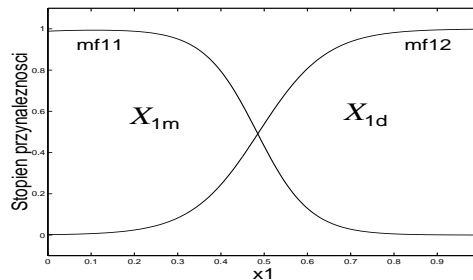
$$R^1: \text{ JEŚLI } x_1 \text{ jest } X_{1m} \text{ i } x_2 \text{ jest } X_{2m} \text{ TO } y = 0.2 + x_1 + 2x_2,$$

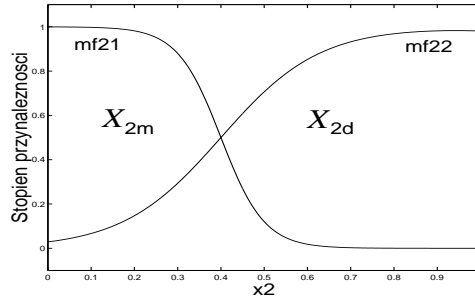
$$R^2: \text{ JEŚLI } x_1 \text{ jest } X_{1m} \text{ i } x_2 \text{ jest } X_{2d} \text{ TO } y = -1 + 2x_1 + 0.5x_2,$$

$$R^3: \text{ JEŚLI } x_1 \text{ jest } X_{1d} \text{ i } x_2 \text{ jest } X_{2m} \text{ TO } y = 1 + 2x_1 + 2x_2,$$

$$R^4: \text{ JEŚLI } x_1 \text{ jest } X_{1d} \text{ i } x_2 \text{ jest } X_{2d} \text{ TO } y = -0.5 + 3x_1 + 1x_2.$$

Zauważmy, że mimo prostych liniowych następników otrzymane odwzorowanie jest silnie nieliniowe.

Rysunek 2.7. Funkcje przynależności zbiorów rozmytych zmiennej  $x_1$



Rysunek 2.8. Funkcje przynależności zbiorów rozmytych zmiennej  $x_2$

### 2.1.3. Projektowanie modeli rozmytych TS

Projektowanie systemu rozmytego typu TS, w szczególności modelu rozmytego znanej lub nieznannej zależności między zmiennymi wejściowymi a zmienną wyjściową, polega na:

1. Podziale przedziału zmienności każdej zmiennej wejściowej na częściowo na siebie nachodzące zbiory rozmyte (*fuzzy partitioning*), tzn. określeniu liczby zbiorów rozmytych oraz wyznaczeniu kształtu i wartości parametrów funkcji przynależności każdego zbioru.
2. Określeniu postaci i parametrów następników funkcyjnych poszczególnych reguł. Reguł powinno być tyle, ile utworzonych obszarów (wielowymiarowych zbiorów rozmytych) rozważanego zbioru zmiennych wejściowych  $X \in \mathbb{R}^n$ , gdzie  $n$  jest liczbą zmiennych – po jednej regule typu TS dla każdego obszaru.
3. Wybraniu sposobu wyznaczania poziomów aktywacji poszczególnych reguł: operatorem iloczynu lub minimum.

Najważniejsze są dwa pierwsze punkty, punkt trzeci to prosta czynność techniczna. Kluczowym dla sukcesu modelowania typu TS jest *właściwe dobranie zbiorów rozmytych* dla zmiennych wejściowych. Zbyt mała ich liczba, niewłaściwe ułożenie względem siebie prowadzą do złego zaprojektowania modelu rozmytego, który nie spełnia wymagań, jest zbyt mało dokładny. Z kolei przyjęcie zbyt dużej liczby zbiorów rozmytych prowadzi do nadmierne rozbudowanego modelu, ze zbyt dużą liczbą parametrów, trudniejszego do projektowania i wolniejszego w działaniu.

Istnieje wiele propozycji automatyzacji procesu podziału na zbiory rozmyte, np. na podstawie dostatecznie reprezentatywnego zbioru par wartości zmiennych wejściowych i wyjściowych modelowanej zależności. Jednakże, nadal na ogół najskuteczniejsze w praktyce okazuje się podjęcie przez człowieka zasadniczej decyzji o liczbie zbiorów i ich orientacyjnym położeniu,

w razie potrzeby w trybie interakcyjnym. To właśnie tu jest miejsce na wykorzystanie trudno formalizowalnej wiedzy empirycznej o problemie, np. o charakterze nieliniowości. Udział człowieka nie stanowi bynajmniej wady podejścia, ale wręcz przeciwnie – jego siłę, gdyż rola człowieka-eksperta jest dobrze określona. Wykorzystanie wiedzy eksperta potrzebne jest do zdefiniowania bardzo ważnej konkretnej rzeczy – właściwej ogólnej struktury systemu rozmytego. Nie chodzi tu bowiem o precyzyjne określenie funkcji przynależności, trzeba jedynie określić ich liczbę i orientacyjne położenie. Precyzyjniejsze dostrojenie wartości parametrów funkcji możliwe jest na ogół już w sposób automatyczny, poprzez zastosowanie odpowiedniego algorytmu optymalizacji na dalszym etapie projektowania.

Sposób określenia *postaci i parametrów następników funkcyjnych* zależy od rodzaju zastosowania. Z reguły przyjmuje się funkcje wielomianowe pierwszego rzędu. Po pierwsze, na ogół to wystarcza; po drugie, prowadzi do możliwości prostej i korzystnej interpretacji systemu rozmytego jako sieci neuropodobnej – jak to przedstawimy dalej. W przypadku modelowania rozmytego funkcje następników będą przybliżeniami liniowymi modelowanej zależności w poszczególnych obszarach, np. linearyzacjami w odpowiednio dobranych punktach charakterystycznych obszarów. W przypadku projektowania systemu rozmytego jako regulatora, funkcje następników będą funkcjami regulatorów lokalnych, dobranych do lokalnych modeli obiektu regulowanego wyznaczonych w poszczególnych obszarach, czy bezpośrednio do zestawów danych charakteryzujących obiekt.

Wybór *rodzaju operatora* (minimum czy iloczyn) do wyznaczania poziomów aktywacji reguł nie ma większego znaczenia dla istoty działania systemu TS, jest arbitralny lub podyktowany potrzebami sposobu projektowania. Operator iloczynu jest różniczkowalny (nie ma tej cechy operator minimum), stąd należy go stosować, jeśli chcemy, aby reprezentowane przez system rozmyty TS nieliniowe odwzorowanie było różniczkowalne. W szczególności, sytuacja taka ma miejsce przy strojeniu parametrów funkcji przynależności w trybie uczenia rozmytej sieci neuronowej.

#### 2.1.4. System TS jako rozmyta sieć neuronowa

System rozmyty TS może być przedstawiony w postaci struktury sieci neuronowej (neuropodobnej), zwanej *rozmytą siecią neuronową* (*fuzzy neural network* – FNN), także systemem ANFIS (*Adaptive Neuro-Fuzzy Inference System*), zob. np. [58, 59, 31, 113, 162, 116]. Sieć taką można interpretować jako perceptron wielowarstwowy, w którym nieliniowe węzły neuronów odpowiadają nieliniowym funkcjom przynależności. Przykładową strukturę rozmytej sieci neuronowej przedstawimy dla konkretnego prostego

systemu rozmytego TS, w którym dziedziną każdej z dwóch zmiennych wejściowych,  $x_1$  i  $x_2$ , podzielona jest na trzy zbiory rozmyte (odpowiednio  $X_{11}$ ,  $X_{12}$  i  $X_{13}$  oraz  $X_{21}$ ,  $X_{22}$  i  $X_{23}$ ) opisane sigmoidalnymi funkcjami przynależności, kolejno lewostronną (jednostronną otwartą lewostronnie), dwustronną i prawostronną (jednostronną otwartą prawostronnie),

$$\begin{aligned}\mu_{X_{j1}}(x_j) &= \frac{1}{1 + \exp[-\alpha_{j1}(x_j - c_{j1})]}, \quad \alpha_{j1} < 0, \\ \mu_{X_{j2}}(x_j) &= \frac{1}{1 + \exp[-\alpha_{j2+}(x_j - c_{j2+})]} - \frac{1}{1 + \exp[-\alpha_{j2-}(x_j - c_{j2-})]}, \\ \mu_{X_{j3}}(x_j) &= \frac{1}{1 + \exp[-\alpha_{j3}(x_j - c_{j3})]}, \quad \alpha_{j3} > 0,\end{aligned}$$

gdzie  $\alpha_{j2+} > 0$ ,  $\alpha_{j2-} > 0$ ,  $c_{j1} < c_{j2+} < c_{j2-} < c_{j3}$ ,  $j = 1, 2$ . Rozważymy bazę reguł składającą się z trzech następujących elementów, w których każdy z poprzedników zawiera jedynie dwa proste warunki,

$R^1$  : JEŚLI  $x_1$  jest  $X_{11}$  i  $x_2$  jest  $X_{21}$  TO  $y^1 = f_1(x) = a_0^1 + a_1^1 x_1 + a_2^1 x_2$ ,

$R^2$  : JEŚLI  $x_1$  jest  $X_{12}$  i  $x_2$  jest  $X_{23}$  TO  $y^2 = f_2(x) = a_0^2 + a_1^2 x_1 + a_2^2 x_2$ ,

$R^3$  : JEŚLI  $x_1$  jest  $X_{13}$  i  $x_2$  jest  $X_{22}$  TO  $y^3 = f_3(x) = a_0^3 + a_1^3 x_1 + a_2^3 x_2$ ,

gdzie  $x = [x_1 \ x_2]^T$ . Rozważamy jedynie trzy reguły dla prostoty prezentacji, pełna baza zawierałaby 9 reguł. Dla wyznaczenia poziomów aktywacji użyty jest operator mnożenia

$$w^1(x) = \mu_{X_{11}}(x) \cdot \mu_{X_{21}}(x),$$

$$w^2(x) = \mu_{X_{12}}(x) \cdot \mu_{X_{23}}(x),$$

$$w^3(x) = \mu_{X_{13}}(x) \cdot \mu_{X_{22}}(x),$$

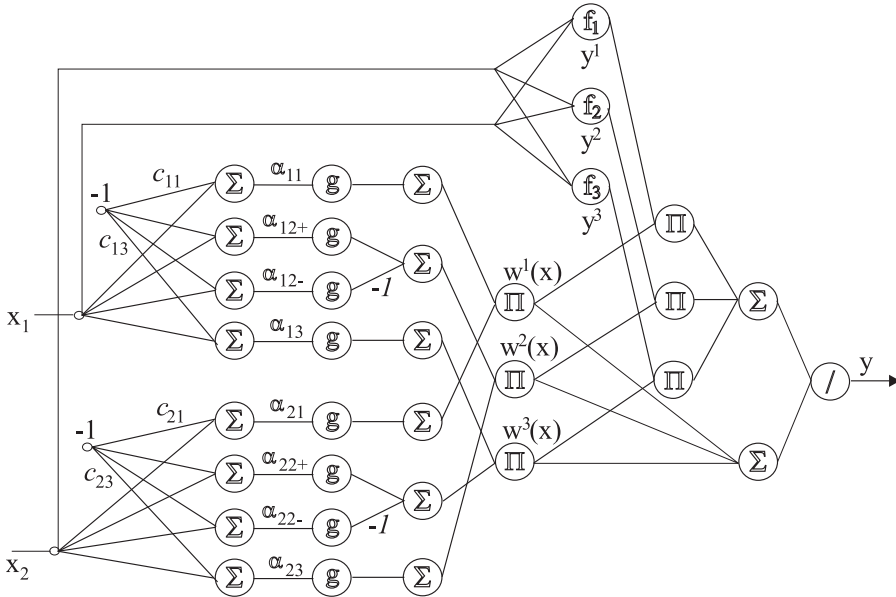
zaś wyjście systemu dane jest według standardowej formuły

$$y = \frac{\sum_{i=1}^3 w^i(x) \cdot y^i}{\sum_{l=1}^3 w^l(x)} = \sum_{i=1}^3 \tilde{w}^i(x) \cdot y^i, \quad (2.7)$$

gdzie przez  $\tilde{w}^i(x)$  oznaczono unormowane poziomy aktywacji reguł.

Struktura rozmytej sieci neuronowej (FNN), dla przypadku ustalonych parametrów funkcji następników reguł, przedstawiona jest na rysunku 2.9. Sieć składa się z ośmiu warstw (nie wliczając w to trywialnej warstwy węzłów wejściowych):

— Cztery pierwsze warstwy związane są z poprzednikami reguł i realizują wyznaczenie poziomów aktywacji  $w^i(x)$  poszczególnych reguł. Trzy



Rysunek 2.9. Reprezentacja przykładowego systemu TS z dwiema zmiennymi wejściowymi i trzema regułami w postaci rozmytej sieci neuronowej, dla przypadku ustalonych parametrów funkcji  $f_i$  następników reguł

pierwsze warstwy obliczają wartości funkcji przynależności, przy czym węzły warstwy drugiej realizują funkcje nieliniowe postaci

$$g(z) = \frac{1}{1 + \exp[-z]},$$

zaś węzły warstwy czwartej, oznaczone przez „ $\Pi$ ”, to węzły mnożące.

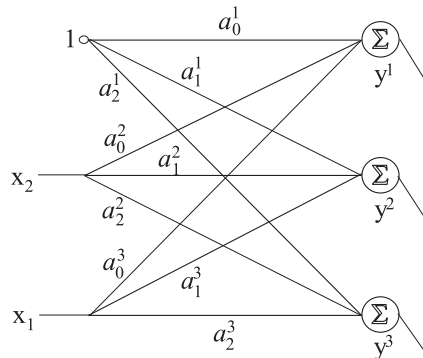
- Każdy z węzłów warstwy piątej oblicza wartość funkcji  $f_i(x)$  następnika reguły – w założonym przypadku ustalonych parametrów funkcji  $f_i$  (nie są dostrajalnymi parametrami sieci) obliczenia te można zrealizować w jednym węźle dla każdej reguły.
- Warstwy szósta, siódma i ósma realizują konkluzję finalną, zgodnie z wzorem (2.7).

Na rysunku 2.9 przyjęto następującą *konwencję*: parametry  $c_{ji}$  i  $\alpha_{ji}$  umieszczone przy łukach, w miarę możliwości na środku, oznaczają dostrajalne parametry sieci, przez które mnożone są sygnały wychodzące z węzłów (w przypadku parametrów  $c_{ji}$ , z braku miejsca nie zaznaczono  $c_{j2+}$  i  $c_{j2-}$ , a  $c_{j3}$  umieszczono bliżej początku łuków,  $j = 1, 2$ ); natomiast przy węzłach przetwarzających sygnały zaznaczono wartości sygnałów z nich wychodzą-

cych. Łuki sieci, przy których nie ma żadnych oznaczeń, należy traktować jako łuki o wartości „1”.

Rozmyta sieć neuronowa może być stosowana do dopasowania (optymalizacji) parametrów  $\alpha_{ji}$  i  $c_{ji}$  postaci funkcji przynależności, poprzez zastosowanie jednego ze standardowych algorytmów uczenia sieci, zob. np. [31, 59, 162, 113]. Oczywiście, możliwe to jest w sytuacji, kiedy dysponujemy zbiorem danych uczących, danych typu wejście-wyjście (w przykładzie:  $x \rightarrow y$ ). Dokonując modelowania rozmytego typu TS znanej zależności nieliniowej sami możemy taki zbiór wygenerować. Typowa jest natomiast sytuacja, kiedy konstruujemy model rozmyty TS nieznannej zależności dysponując jedynie zbiorem danych typu wejście-wyjście. Po wybraniu przez użytkownika liczby zbiorów rozmytych i wstępnego kształtu funkcji przynależności oraz struktury następników reguł, dostrajać można metodami uczenia sieci zarówno parametry funkcji przynależności, jak i parametry funkcji następników. Do wyboru są tu dwa sposoby:

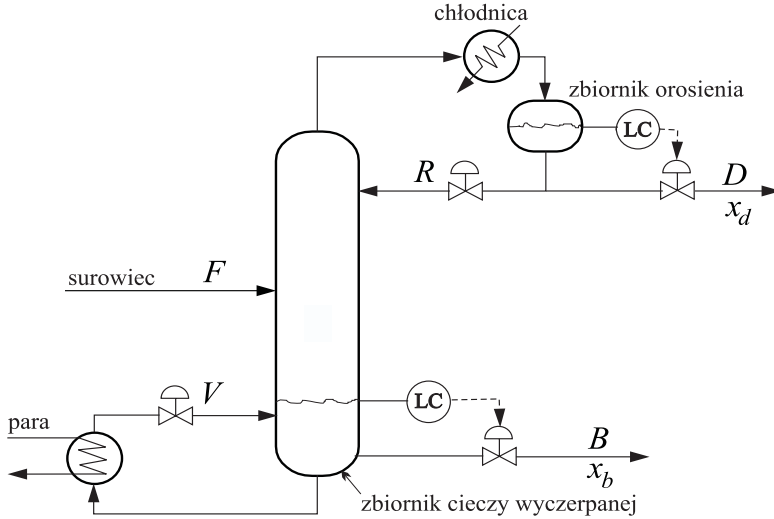
- Rozbudowujemy sieć zastępując węzły „ $f_i$ ” dodatkowymi fragmentami przedstawionymi na rysunku 2.10 – i dostrajamy łącznie parametry funkcji przynależności i następników reguł wybraną metodą uczenia sieci, np. optymalizacją gradientową z wyliczaniem gradientu metodą propagacji wstecznej.
- Stosujemy algorytm *hybrydowy* (zob. [59, 113]): parametry poprzedników reguł dostrajamy metodą uczenia sieci (optymalizacja gradientowa z wykorzystaniem algorytmu propagacji wstecznej dla wyznaczania gradientu), na zmianę z dostrajaniem parametrów następników metodą najmniejszych kwadratów. Autorzy pracy [59] przeprowadzili badania porównawcze wskazujące na lepsze własności algorytmu hybrydowego.



Rysunek 2.10. Fragment struktury sieci realizujący następniki reguł

**Przykład 2.3.**

Przedstawimy wyniki modelowania rozmytego charakterystyk statycznych kolumny destylacyjnej służącej do rozdzielania dwuskładnikowej mieszanki metanolu i wody, przedstawionej schematycznie na rysunku 2.11.

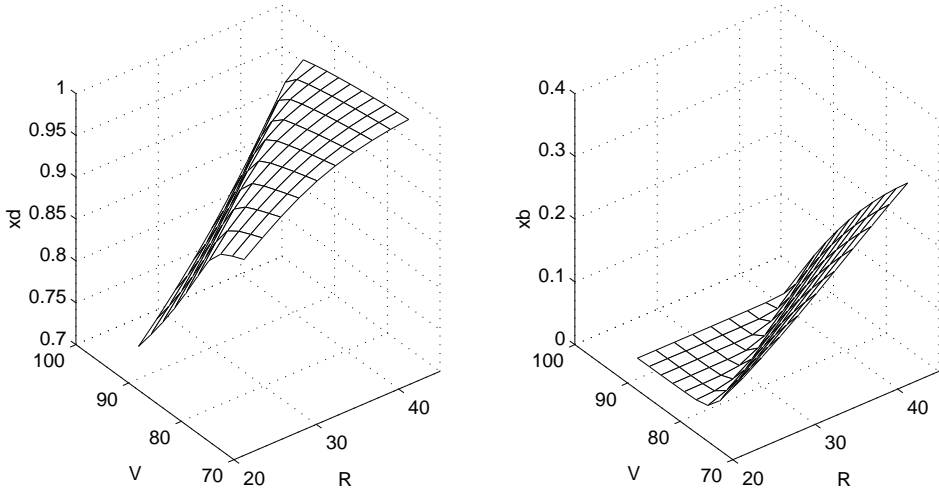


Rysunek 2.11. Schemat kolumny destylacyjnej do przykładu 2.3

Zgodnie z zasadami opisu zjawisk zachodzących w takiej kolumnie [86, 117, 49], zbudowano model w postaci zestawu równań różniczkowych i algebraicznych opisujących zjawiska zachodzące na poszczególnych półkach kolumny, w zbiorniku orosienia i zbiorniku cieczy wyczerpanej [70]. Model uzupełniono regulatorami stabilizującymi poziomy cieczy w zbiorniku orosienia i zbiorniku cieczy wyczerpanej (dół kolumny), poprzez sterowanie strumieniami destylatu  $D$  i cieczy wyczerpanej  $B$ .

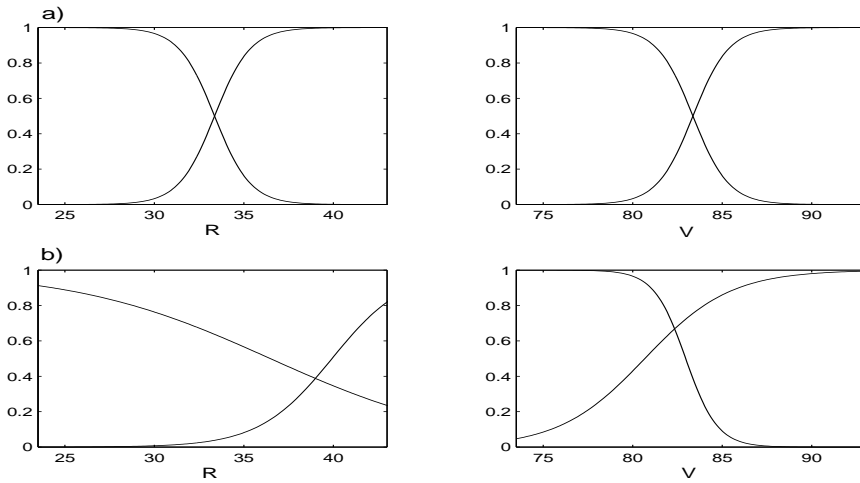
Przyrównując wszystkie pochodne do zera uzyskano następnie układ nieliniowych równań algebraicznych opisujących charakterystyki statyczne kolumny. Wielkościami wejściowymi (sterowaniami) tych charakterystyk są: natężenie strumienia orosienia (strumienia recyklu ze zbiornika orosienia do kolumny)  $R$  i natężenia strumienia pary  $V$  ogrzewającej walek (w  $kmol/godz$ ). Wyjściami są natomiast stężenia produktu (metanolu)  $x_d$  i  $x_b$ , odpowiednio w destylacie  $D$  i w cieczy wyczerpanej  $B$ . Na rysunku 2.12 przedstawiono oryginalne, nieliniowe powierzchnie charakterystyk statycznych kolumny.

Przystępując do budowy modelu rozmytego TS przedstawionych charakterystyk statycznych przyjęto podział każdej z dziedzin zmiennych wejściowych  $R$  i  $V$  na dwie równe partycje (dwa zbiory rozmyte), przyjmując



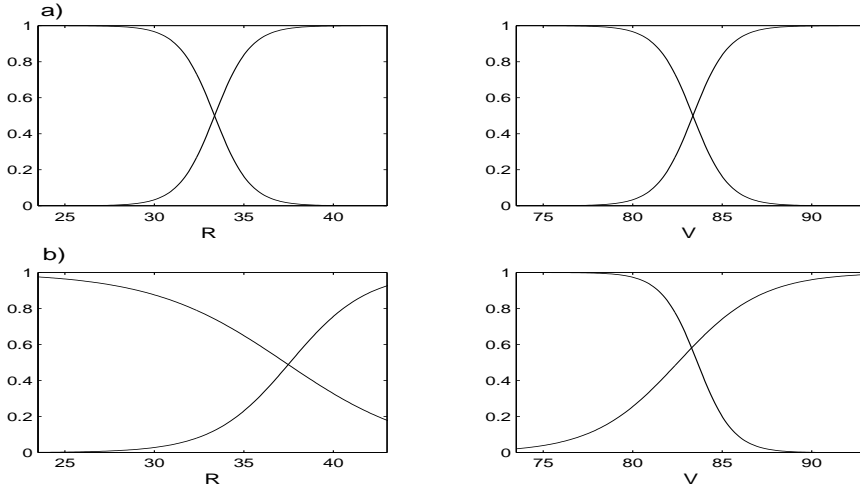
Rysunek 2.12. Powierzchnie charakterystyk statycznych obiektu, przykład 2.3

arbitralnie pokazane na rysunkach 2.13 a) i 2.14 a) kształty sigmoidalnych funkcji przynależności. W dziedzinie każdej z charakterystyk utworzono



Rysunek 2.13. Funkcje przynależności zbiorów rozmytych modelu TS charakterystyki  $x_d$ : a) początkowe, b) po procesie optymalizacji parametrycznej

w ten sposób cztery rozmyte podobszary (dwuwymiarowe zbiory rozmyte), porównaj z rysunkiem 2.5 i przykładem 2.1. W każdym z nich utworzono



Rysunek 2.14. Funkcje przynależności zbiorów rozmytych modelu TS charakterystyki  $x_b$ : a) początkowe, b) po procesie optymalizacji parametrycznej

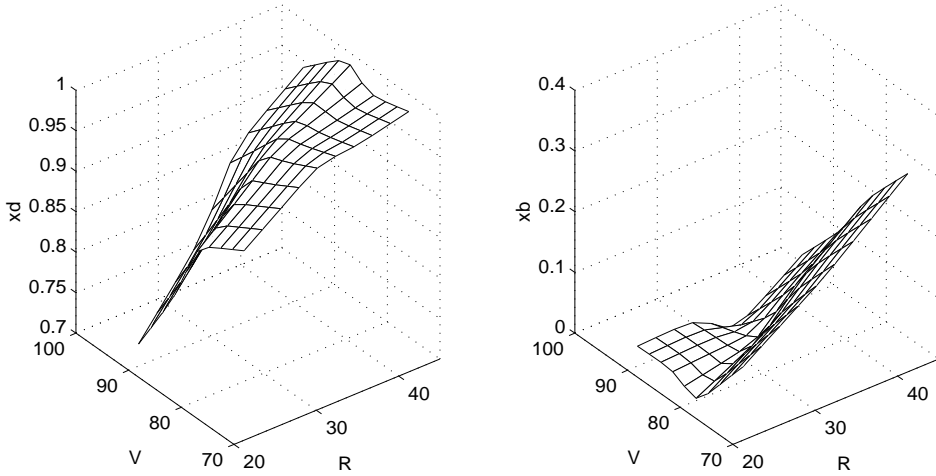
liniowy model dla stężenia  $x_d$  i liniowy model dla stężenia  $x_b$

$$x_d = a_{d0}^i + a_{d1}^i R + a_{d2}^i V, \quad i = 1, \dots, 4,$$

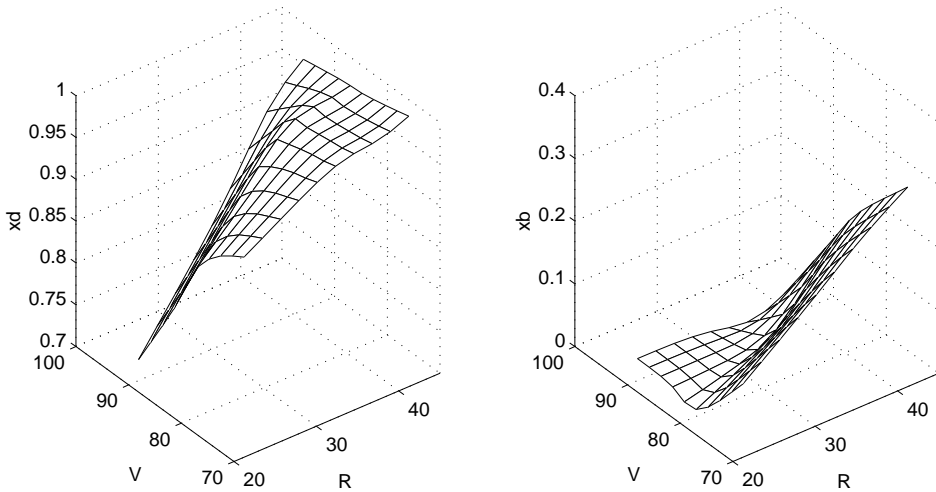
$$x_b = a_{b0}^i + a_{b1}^i R + a_{b2}^i V, \quad i = 1, \dots, 4.$$

Dla celów identyfikacji parametrów  $a_{dj}^i$  i  $a_{bj}^i$  modeli lokalnych utworzono zestaw danych do modelowania: w 441 ( $21 \times 21$ ) równomiernie rozłożonych punktach pokrywających cały zakres zmienności  $R$  i  $V$  obliczono wartości charakterystyk statycznych z rysunku 2.12. Następnie dostosowano do tych danych parametry modeli lokalnych metodą minimalizacji odchylenia średniokwadratowego. Powierzchnie uzyskanych w ten sposób modeli rozmytych TS pokazano na rysunku 2.15.

Zoptymalizowano następnie parametry funkcji przynależności metodą hybrydową: wykonując, przemiennie, kroki modyfikujące wartości parametrów funkcji przynależności w kierunku ujemnego gradientu (liczonego metodą propagacji wstecznej) funkcji średniokwadratowego błędu modelowania i kroki dostrajające parametry modeli liniowych, metodą najmniejszych kwadratów. Wykonano 500 iteracji, po których wartość błędu modelowania wyniosła  $8.45 \times 10^{-3}$  dla modelu  $x_d$  i  $7.49 \times 10^{-3}$  dla modelu  $x_b$ . Kształty uzyskanych w ten sposób funkcji przynależności pokazano na rysunkach 2.13 b) i 2.14 b), zaś powierzchnie modeli rozmytych na rysunku 2.16. Zwraca uwagę dobra dokładność modelowania silnie nieliniowych zależności, przy bardzo niewielkiej liczbie zbiorów rozmytych.



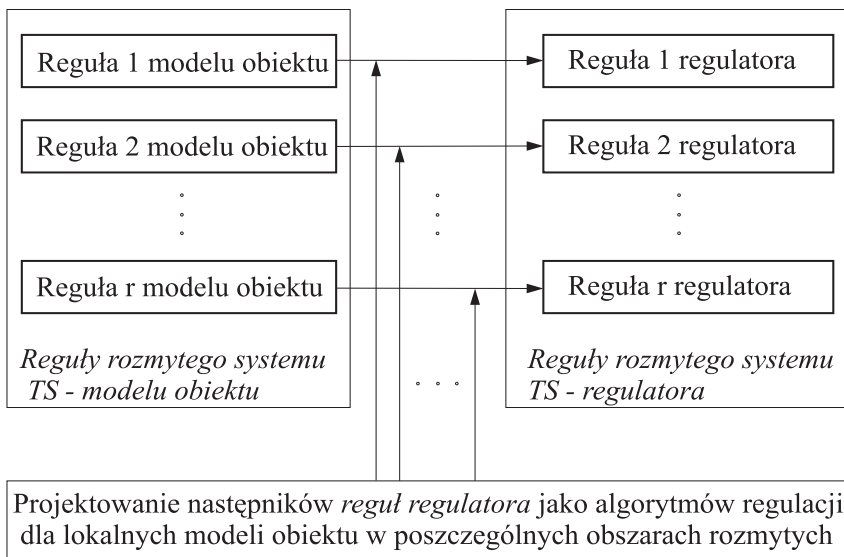
Rysunek 2.15. Powierzchnie modeli rozmytych TS charakterystyk statycznych stężeń  $x_d$  i  $x_b$  dla początkowych wartości parametrów funkcji przynależności



Rysunek 2.16. Powierzchnie modeli rozmytych TS charakterystyk statycznych stężeń  $x_d$  i  $x_b$  dla optymalnych wartości parametrów funkcji przynależności

## 2.2. Dyskretne rozmyte algorytmy regulacji typu TS

Przez *algorytm regulacji rozmyty TS* określamy system rozmyty typu TS, w którym następnik każdej reguły jest funkcją definiującą *algorytm regulacji*, zaprojektowany dla regulacji obiektu lokalnie w obsza-



Rysunek 2.17. Projektowanie regulatora rozmytego TS dla obiektu danego modelem rozmytym

rze (zbiornym rozmytym) określonym poprzednikami reguły. Przedstawione podejście nazywane bywa również w literaturze *kompensacją równoległą rozproszoną* (PDC – *Parallel Distributed Compensation*), zob. np. [160], algorytmem rozmytym *wieloobszarowym* [32, 33] czy *wielomodelowym* [21]. Algorytmy regulacji następników reguł projektowane są na ogół lokalnie, dla lokalnych modeli opisujących obiekt w poszczególnych obszarach. W ogólności, lokalne modele obiektu mogą być liniowe lub nieliniowe, podobnie algorytmy lokalnych regulatorów. W praktyce stosujemy lokalne modele i algorytmy regulacji liniowe, gdyż wystarcza to w pełni do opisu nieliniowego, jeśli tylko obszary są odpowiednio dobrane do nieliniowości obiektu. Natomiast przy lokalnym podejściu liniowym projektowanie jest bardzo efektywne, można wykorzystać znane i sprawdzone w praktyce proste algorytmy regulacji. Stąd ograniczamy się w niniejszej pracy do *liniowych* lokalnych modeli i algorytmów regulacji. Struktura projektowania reguł algorytmu regulacji rozmytego TS, dla przypadku obiektu opisanego modelem rozmytym TS, przedstawiona jest na rysunku 2.17.

Dla zaprojektowania regulatora rozmytego TS należy:

1. a) Zdefiniować zmienne występujące w poprzednikach reguł. Wartości tych zmiennych określać będą przynależność aktualnego stanu (punktu pracy) obiektu do wydzielonych *lokalnych obszarów*, w któ-

rych obiekt można do celów regulacji przybliżać modelami liniowymi.

- b) Dokonać podziału zakresu zmienności każdej ze zmiennych poprzedników reguł na wzajemnie się zazębiające *zbiory rozmyte*, określając ich liczbę oraz wyznaczając kształt i wartości parametrów funkcji przynależności każdego zbioru, w relacji do nieliniowości obiektu.
- c) Sformułować modele liniowe obiektu w każdym z powstałych w ten sposób rozmytych lokalnych obszarów (każdy z obszarów definiowany jest przez jedną regułę, por. przykład 2.2).

Postępowanie powyższe to skonstruowanie modelu rozmytego TS rozważanego nieliniowego obiektu regulacji, z liniowymi następnikami funkcyjnymi będącymi dynamicznymi liniowymi modelami obiektu w poszczególnych obszarach. Zakres zmienności sygnałów wejściowych i wyjściowych powinien pokrywać cały interesujący obszar pracy projektowanego regulatora.

2. Dla każdego z liniowych modeli (podmodeli) obiektu *zaprojektować regulator liniowy* pracujący prawidłowo w danym obszarze, wybraną techniką odpowiednią do zastosowanych liniowych modeli obiektu, np. regulator liniowy ze sprzężeniem od stanu dla modeli w postaci liniowych równań stanu, regulator PID dla modeli typu ARX, itp. Utworzyć regulator rozmyty TS – system rozmyty typu TS z poprzednikami reguł utworzonymi w punkcie poprzednim (podział na obszary jak w modelu rozmytym TS obiektu) i następnikami funkcyjnymi będącymi wyznaczonymi algorytmami regulacji.
3. Przeprowadzić *symulacyjną i teoretyczną analizę* uzyskanego rozmytego algorytmu TS: sprawdzić, na ile to możliwe, spełnienie warunków stabilności metodami analitycznymi oraz przesymulować działanie układu regulacji w przewidywanych warunkach. Jeśli wyniki są niezadowalające, powrócić do punktu poprzedniego i skorygować parametry lokalnych regulatorów liniowych. Powtórzyć analizę symulacyjną i teoretyczną uzyskanego regulatora. Jeśli występują trudności z uzyskaniem wymaganej jakości regulacji poprzez jedynie korekty regulatorów lokalnych, to powrócić do punktu 1 i zmodyfikować zbiory rozmyte poprzedników reguł, tzn. parametry funkcji przynależności, lub nawet liczbę partycji (liczbę zbiorów rozmytych), na jakie dzielimy zakresy poszczególnych zmiennych występujących w poprzednikach reguł.

Należy zaznaczyć, że punkt ostatni jest istotny. Cechy regulatora rozmytego TS nie zawsze są bowiem prostym złożeniem cech regulatorów lokalnych. Znane są przykłady, zob. np. [138], kiedy mimo stabilności wszystkich regulatorów lokalnych regulator nieliniowy rozmyty TS jest niestabilny.

Zwiększenie niezawodności i mocy systemów komputerowych oraz silna konkurencja na rynku produktów przemysłowych prowadzą nieuchronnie do stosowania nowoczesnego sterowania warstwowego z bieżącą optymalizacją punktów pracy. Stąd w wielu pętlach regulacji bezpośredniej trzeba myśleć o zastosowaniu regulatorów nieliniowych, gdyż w sytuacji bieżących zmian punktów pracy i istotnych nieliniowości powszechnie występujące regulatory liniowe typu PID nie zawsze zapewniają odpowiednią jakość regulacji. Muszą one wówczas być strojone na najgorszy przypadek, tzn. dla punktu pracy krytycznego dla stabilności i odporności, a stąd w punktach mniej krytycznych są zbyt wolne. Rozważane w niniejszym rozdziale regulatory rozmyte typu TS nadają się nie tylko do *regulacji nadrzędnej* (gdzie czasy próbkowania są z reguły większe), ale pasują również znakomicie do zastosowania w *regulacji bezpośredniej*. Jeśli np. w następnikach reguł regulatora rozmytego TS występują funkcje regulatorów liniowych PID, to regulator taki jest naturalnym nieliniowym uogólnieniem regulatora liniowego PID, uogólnieniem które powinno być stosunkowo łatwo zrozumiane i akceptowane przez personel obsługujący system sterowania obiektem.

*Dziedzina czasu* jest dziedziną naturalną dla systemów rozmytych o strukturze TS stosowanych do modelowania obiektów czy konstruowania układów dynamicznych, takich jak regulatory, gdyż dla zmiennych w dziedzinie czasu formułowane są warunki występujące w poprzednikach reguł modelu obiektu czy regulatora. Stąd też w następnikach funkcyjnych reguł występują dynamiczne modele liniowe określające wartości zmiennych wyjściowych modelu obiektu czy regulatora w określonej chwili. Natomiast standardowym, praktycznym podejściem przy doborze nastaw regulatora przemysłowego typu PID jest wykorzystanie znacznie uproszczonego modelu obiektu, przede wszystkim w postaci pewnej zastępczej transmitancji, np. jednonercyjnej z opóźnieniem, uzyskanej na podstawie otrzymanej eksperymentalnie odpowiedzi na skok jednostkowy czy impuls prostokątny – i wyznaczenie nastaw z wykorzystaniem odpowiednich tabel, zob. np. [41, 47]. Nic nie stoi na przeszkodzie zastosowania tego sposobu do projektowania regulatorów rozmytych typu PID, jedynie lokalne modele obiektów odpowiadające poszczególnym obszarom rozmytym należy przedstawić w postaci transmitancji wykorzystywanych do doboru nastaw.

Strojenie parametrów regulatorów lokalnych, lub nawet łącznie parametrów regulatorów i funkcji przynależności, może być też dokonywane w sposób globalny, automatyczny i zalgorytmizowany – jeśli dysponujemy wiarygodnym symulacyjnym modelem obiektu regulacji. Możemy wówczas zastosować procedurę optymalizacji funkcji sformułowanej jako odpowiedni wskaźnik jakości regulacji, na ogół również z ograniczeniami, jak np. na

przeregulowanie. Jeśli natomiast dysponujemy zestawem danych wzorcowych, to przedstawiając projektowany regulator rozmyty TS w postaci rozmytej sieci neuronowej można zastosować algorytm optymalizacji (uczenia) takiej sieci (zob. rozdz. 2.1.4).

Przedstawiona powyżej dyskusja dotyczy ogólnych zagadnień projektowania nieliniowych regulatorów rozmytych typu TS. W aktualnym stanie techniki regulatory te implementowane są w technice mikroprocesorowej, a więc zawsze są to regulatory dyskretne (cyfrowe). Jednakże, jeśli okres próbkowania w układzie regulacji jest wystarczająco mały w stosunku do dominujących stałych czasowych obiektu, to w pełni uprawnionym sposobem projektowania jest synteza ciągłego algorytmu regulacji – i jedynie cyfrowa jego realizacja, zob. np. [47, 54]. Dlatego też, po przedstawieniu w kolejnych rozdziałach zagadnień projektowania dyskretnych rozmytych regulatorów TS, poruszymy następnie analogiczne zagadnienia dla przypadku ciągłego.

### 2.2.1. Regulator rozmyty TS ze sprzężeniem od stanu

Pierwszym rozważanym w literaturze regulatorem rozmytym TS był regulator w postaci liniowych sprzężeń od zmiennych stanu obiektu [138, 160], bazujący na modelu rozmytym TS obiektu z następnikami w postaci liniowych równań stanu. Dopiero później zaproponowano regulatory oparte na modelach typu wejście-wyjście, ze sprzężeniem od wyjścia.

Rozważmy model dyskretny rozmyty TS obiektu dynamicznego o regulacjach w następującej postaci:

$$\begin{aligned} R_{ob}^i : \quad & \text{JEŚLI } x_1(k) \text{ jest } A_1^i \text{ i } x_2(k) \text{ jest } A_2^i \text{ i } \dots \text{ i } x_{n_x}(k) \text{ jest } A_{n_x}^i \\ & \text{TO } x^i(k+1) = \mathbf{A}_i x(k) + \mathbf{B}_i u(k), \quad (2.8) \end{aligned}$$

gdzie

$$\begin{aligned} x(k) &= [x_1(k) \ x_2(k) \ \dots \ x_{n_x}(k)]^T, \\ u(k) &= [u_1(k) \ u_2(k) \ \dots \ u_{n_u}(k)]^T \end{aligned}$$

są odpowiednio wektorami stanu i sterowań obiektu dynamicznego w chwili  $k$ -tej,  $A_j^i \in \mathbb{X}_j = \{X_{j1}, \dots, X_{jr_j}\}$  są zbiorami rozmytymi współrzędnej  $x_j$  wektora stanu  $x$  (zob. (2.4)),  $j = 1, \dots, n_x$ , zaś  $\mathbf{A}_i$  i  $\mathbf{B}_i$  są macierzami stanu i sterowań lokalnych modeli liniowych utworzonych dla poszczególnych obszarów,  $i = 1, \dots, r$ .

Przy siatkowym podziale obszaru zmiennych wejściowych (*grid partition* [58]), którego przykładem jest podział przedstawiony na rys. 2.5, dostajemy system rozmyty o ilości obszarów (i reguł) równej

$$r = r_1 \cdot r_2 \cdot \dots \cdot r_{n_x} = \prod_{j=1}^{n_x} r_j.$$

Wynika stąd, że liczba obszarów rośnie szybko wraz ze wzrostem wymiarowości  $n_x$  wektora stanu  $x$  i liczby  $r_j$  zbiorów rozmytych odpowiadających każdej ze współrzędnych  $x_j$  wektora stanu. Istotne jest więc, aby zakresy zmienności poszczególnych zmiennych stanu były podzielone na jak najmniejszą liczbę zbiorów rozmytych, tzn. aby wartości  $r_j$  były małe. Jest to możliwe, przy jednoczesnym zachowaniu dokładności modelowania, właśnie przy stosowaniu struktur typu TS – w odróżnieniu od klasycznego modelowania rozmytego, gdzie następnikami reguł są zbiory rozmyte.

Dla danych wartości wektora stanu  $x(k)$  i sterowania  $u(k)$  wyjście modelu rozmytego, tj. stan w chwili następnej, wyliczane jest zgodnie z ogólnym wzorem postaci (2.5), tj.

$$x(k+1) = \frac{\sum_{i=1}^r w^i(k) [\mathbf{A}_i x(k) + \mathbf{B}_i u(k)]}{\sum_{l=1}^r w^l(k)}, \quad (2.9)$$

gdzie  $w^i(k)$  są poziomami aktywacji poszczególnych reguł (2.8) w chwili  $k$ -tej,

$$w^i(k) = \prod_{j=1}^{n_x} \mu_{A_j^i}(x_j(k)).$$

Przyjęliśmy powyżej naturalne założenie, że zawsze  $\sum_{l=1}^r w^l(k) > 0$ , tj. dla każdej wartości zmiennych stanu z rozważanego zakresu ich zmienności model jest określony – co najmniej jedna reguła jest aktywowana. Wygodnie jest operować unormowanymi poziomami aktywacji reguł, zob. (2.6), które z definicji spełniają warunek  $\sum_{i=1}^r \tilde{w}^i(k) = 1$ . Wówczas model obiektu niesterowanego (przy  $u(k) \equiv 0$ ) przyjmuje postać

$$x(k+1) = \sum_{i=1}^r \tilde{w}^i(k) \mathbf{A}_i x(k). \quad (2.10)$$

Dla takiego modelu dynamicznego można stosunkowo łatwo, korzystając bezpośrednio z twierdzenia Lapunowa (zob. np. [61, 159]), wyprowadzić warunek dostateczny stabilności asymptotycznej, zob. [138].

**Twierdzenie 2.1.** *Punkt równowagi systemu dynamicznego (2.10) opisanego regułami (2.8) z  $u(k) = 0$  jest globalnie asymptotycznie stabilny, jeśli*

istnieje macierz symetryczna dodatnio określona  $\mathbf{P}$  taka, że dla macierzy stanu  $\mathbf{A}_i$  każdego z modeli lokalnych spełniona jest nierówność

$$\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < \mathbf{0}, \quad i = 1, 2, \dots, r. \quad (2.11)$$

**Dowód.** Utworzymy funkcję skalarną postaci

$$V(x(k)) = x^T(k) \mathbf{P} x(k), \quad (2.12)$$

gdzie  $\mathbf{P}$  jest macierzą symetryczną dodatnio określoną. Funkcja (2.12) spełnia następujące warunki:

$$\begin{aligned} V(0) &= 0, \\ V(x(k)) &> 0 \quad \text{dla } x \neq 0, \\ V(x(k)) &\rightarrow \infty \quad \text{dla } \|x(k)\| \rightarrow \infty. \end{aligned}$$

Ponadto jej przyrost wzdłuż trajektorii układu dynamicznego (2.10) określony jest wzorem

$$\begin{aligned} \Delta V(x(k)) &= V(x(k+1)) - V(x(k)) \\ &= x^T(k+1) \mathbf{P} x(k+1) - x^T(k) \mathbf{P} x(k) \\ &= x^T(k) \left[ \sum_{i=1}^r \tilde{w}^i(k) \mathbf{A}_i^T \mathbf{P} \sum_{j=1}^r \tilde{w}_j(k) \mathbf{A}_j - \mathbf{P} \right] x(k) \\ &= \sum_{i=1}^r \sum_{j=1}^r \tilde{w}^i(k) \tilde{w}^j(k) x^T(k) [\mathbf{A}_i^T \mathbf{P} \mathbf{A}_j - \mathbf{P}] x(k) \\ &= \sum_{i=1}^r (\tilde{w}^i(k))^2 x^T(k) [\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P}] x(k) + \\ &+ \sum_{i=1}^r \sum_{j < i}^r \tilde{w}^i(k) \tilde{w}^j(k) x^T(k) [\mathbf{A}_i^T \mathbf{P} \mathbf{A}_j + \mathbf{A}_j^T \mathbf{P} \mathbf{A}_i - 2\mathbf{P}] x(k). \end{aligned}$$

Mamy

$$\begin{aligned} \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j + \mathbf{A}_j^T \mathbf{P} \mathbf{A}_i - 2\mathbf{P} &= -(\mathbf{A}_i - \mathbf{A}_j)^T \mathbf{P} (\mathbf{A}_i - \mathbf{A}_j) \\ &+ [\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P}] + [\mathbf{A}_j^T \mathbf{P} \mathbf{A}_j - \mathbf{P}], \end{aligned}$$

zaś z założenia dodatniej określoności  $\mathbf{P}$  wynika

$$-(\mathbf{A}_i - \mathbf{A}_j)^T \mathbf{P} (\mathbf{A}_i - \mathbf{A}_j) \leq \mathbf{0}.$$

Ponadto

$$\tilde{w}^i(k) \geq 0 \quad \text{dla każdego } i = 1, 2, \dots, r,$$

stąd  $\Delta V(x(k)) < 0$  dla  $x(k) \neq 0$  jeśli tylko  $\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < \mathbf{0}$  dla każdego  $i = 1, 2, \dots, r$ . Wówczas funkcja  $V(x(k))$  jest funkcją Lapunowa dla nieliniowego układu dynamicznego (2.10), co kończy dowód.  $\square$

Podkreślmy, że w twierdzeniu powyższym występuje tylko jedna macierz  $\mathbf{P}$ , wspólna dla wszystkich lokalnych liniowych modeli dynamicznych postaci  $x^i(k+1) = \mathbf{A}_i x(k)$ . Rozwiązanie (lub stwierdzenie jego braku) układu liniowych nierówności macierzowych

$$\begin{aligned} \mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < \mathbf{0}, \quad i = 1, 2, \dots, r, \\ \mathbf{P} > \mathbf{0}, \quad \mathbf{P} \text{ symetryczna,} \end{aligned} \quad (2.13)$$

nie nastęrcza aktualnie większych trudności, możemy tu zastosować procedury z *LMI Toolbox* pakietu MATLAB (LMI – *Linear Matrix Inequalities*).

Warto w tym miejscu zaznaczyć, że dla nieosobliwych macierzy  $\mathbf{A}_i$  warunkiem koniecznym istnienia macierzy  $\mathbf{P}$  spełniającej twierdzenie 2.1 jest, aby każda z macierzy  $\mathbf{A}_i \mathbf{A}_j$ ,  $i, j = 1, 2, \dots, r$  była macierzą układu asymptotycznie stabilnego, tj. posiadała wartości własne o modułach mniejszych od jedności [138]. Z warunku (2.11) wynika bowiem

$$\mathbf{P} - (\mathbf{A}_j^{-1})^T \mathbf{P} (\mathbf{A}_j)^{-1} < \mathbf{0},$$

co zsumowane z (2.11) daje

$$\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - (\mathbf{A}_j^{-1})^T \mathbf{P} (\mathbf{A}_j)^{-1} < \mathbf{0},$$

czyli

$$\mathbf{A}_j^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_i \mathbf{A}_j - \mathbf{P} < \mathbf{0}, \quad i = 1, 2, \dots, r.$$

Z ostatniej nierówności wynika, że macierz  $\mathbf{A}_i \mathbf{A}_j$  musi być asymptotycznie stabilna. Tak więc wykrycie, że którakolwiek z macierzy  $\mathbf{A}_i \mathbf{A}_j$  nie jest asymptotycznie stabilna, przesądza o nieistnieniu macierzy  $\mathbf{P}$  spełniającej warunki twierdzenia 2.1.

W pracy [161] przedstawiono wersję twierdzenia 2.1 z osłabionymi założeniami, dla obiektów dynamicznych z ograniczonymi przejściami między różnymi podzbiórami obszaru działania. Aby przedstawić ten rezultat, cały zbiór stanów obiektu trzeba podzielić dodatkowo na podzbiory ciągłej aktywacji reguł rozmytych (*constant activation of fuzzy rules*) [60, 161], które dalej można nazwać komórkami ciągłej aktywacji lub, jeszcze krócej, komórkami aktywacji (*constant activation cells, activation cells*). Komórki aktywacji oznaczymy przez  $S_l$ ,  $l = 1, \dots, L$ , gdzie  $L$  jest liczbą wszystkich komórek. Wewnątrz każdej komórki aktywacji poziom aktywacji każdej reguły rozmytej jest albo zero (reguła nieaktywna) albo dodatni (reguła aktywna – jej poziom aktywacji może być równy zero jedynie na granicy komórki).

Komórki aktywacji można podzielić na takie, które są w reżimie pracy (*operating regime*) i takie, które są w reżimie interpolacyjnym (*interpolating regime*). Komórka jest w reżimie pracy, jeśli  $\tilde{w}^l(x(k)) = 1$  dla pewnej wartości  $l \in L$  i wszystkie pozostałe znormalizowane poziomy aktywacji reguł są równe zero – czyli dynamika obiektu jest w pełni określona przez lokalny model liniowy będący następnikiem reguły rozmytej definiującej rozmyty podzbiór zawierający rozważaną komórkę. Z drugiej strony, komórki z reżimem interpolacyjnym definiuje się jako takie, dla których  $0 < \tilde{w}^l(x(k)) < 1$  for  $x(k) \in \text{int}S_l$  (wnętrze zbioru  $S_l$ ) dla przynajmniej dwóch wartości  $l \in L$  (przynajmniej dwóch, ponieważ  $\sum_{l \in L} \tilde{w}^l(x(k)) = 1$  dla każdej chwili próbkowania  $k$ ). Zauważmy, że wszystkie komórki aktywacji są zbiorami ostrymi, nie zachodzącymi na siebie. Dzielą one cały zbiór stanów obiektu w sposób ściśle powiązany z podziałem tego zbioru na podzbiory rozmyte, definiowane dotychczas w niniejszym rozdziale (zob. początek rozdziału 2.2), jednakże nieco inaczej – jest ich więcej i nie zachodzą na siebie.

Przykład podziału zbioru stanów obiektu dwuwymiarowego na podzbiory rozmyte i na komórki aktywacji reguł rozmytych pokazuje rysunek 2.18, dla modelu TS zawierającego następujące trzy reguły:

$$\begin{aligned} R_p^1: & \text{ JEŚLI } x_1(k) \text{ jest } X_1 \text{ TO } x(k+1) = A_1x(k) + B_1u(k), \\ R_p^2: & \text{ JEŚLI } x_1(k) \text{ jest } X_2 \text{ TO } x(k+1) = A_2x(k) + B_2u(k), \\ R_p^3: & \text{ JEŚLI } x_1(k) \text{ jest } X_3 \text{ TO } x(k+1) = A_3x(k) + B_3u(k), \end{aligned}$$

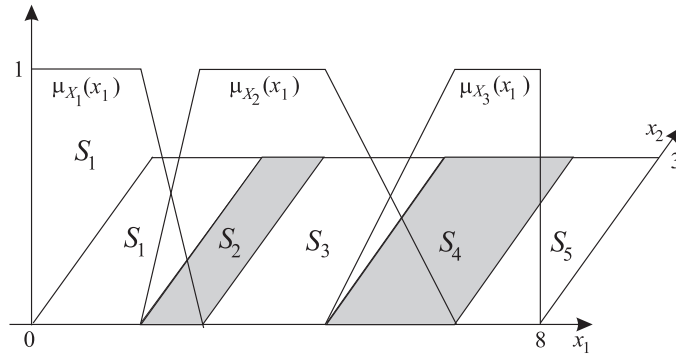
gdzie  $x = (x_1, x_2)$ ,  $0 \leq x_2 \leq 3$ ,  $X_i$  są zbiorami rozmytymi definiowanymi przez funkcje przynależności  $\mu_{X_i}(x_1)$ ,  $i = 1, 2, 3$ , natomiast trzy dwuwymiarowe rozmyte podzbiory są zdefiniowane przez iloczyny kartezjańskie  $X_i \times [0, 3]$ ,  $i = 1, 2, 3$ .

Zauważmy, że podział zbioru stanów obiektu na komórki aktywacji jest formalnie możliwy jedynie dla trapezoidalnych (czy podobnych) funkcji przynależności, ponieważ funkcje sigmoidalne, gaussowskie czy dzwonowe posiadają, ze ściśle matematycznego punktu widzenia, dodatnie poziomy aktywacji w całym obszarze zmienności argumentu.

Oznaczając, dla każdej komórki aktywności  $S_l$ , przez  $K(l)$  zbiór indeksów wszystkich reguł rozmytych z niezerowymi poziomami aktywacji wewnątrz  $S_l$  (dla komórki  $S_l$  z reżimem operacyjnym odpowiadający jej zbiór  $K(l)$  jest jednoelementowy), można opisać dynamikę obiektu autonomicznego (bez sterowania) w następujący sposób:

$$x(k+1) = \sum_{i \in K(l)} \tilde{w}^i(k) [\mathbf{A}_i x(k)], \quad x(k) \in S_l, \quad l \in L, \quad (2.14)$$

gdzie  $0 < \tilde{w}^i(k) = \tilde{w}^i(x(k)) \leq 1$  dla każdego  $i \in K(l)$  i  $\sum_{i \in K(l)} \tilde{w}^i(k) = 1$ .



Rysunek 2.18. Funkcje przynależności  $\mu_{X_i}(x_1)$  definiujące 3 zbiory rozmyte  $X_i$ , 3 obszary rozmyte  $X_i \times [0, 3]$  i odpowiadające im 5 komórek aktywacji: 3 z reżimem operacyjnym ( $S_1, S_3, S_5$  – kolor biały) i 2 z reżimem interpolacyjnym ( $S_2, S_4$  – kolor szary)

Zdefiniujmy ponadto, dla dalszej analizy, zbiór  $\Omega$  opisujący *wszystkie możliwe jednokrokowe (między kolejnymi chwilami impulsowania) przejścia stanu obiektu* między różnymi komórkami aktywacji  $S_l$ ,

$$\Omega = \{(l, m) : x(k) \in S_l, x(k+1) \in S_m, \text{ for every } l, m \in L\}, \quad (2.15)$$

gdzie  $m \neq l$  kiedy obiekt przechodzi (w jednym kroku) z komórki  $S_l$  do komórki  $S_m$ , podczas gdy  $m = l$  kiedy system pozostaje w tej samej komórce  $S_l$ . Można teraz sformułować wersję twierdzenia 2.1 z osłabionymi warunkami stabilności, możliwymi do uzyskania dzięki wykorzystaniu odcinkowo-kwadratowych funkcji Lapunowa, zamiast jednej globalnej kwadratowej.

**Twierdzenie 2.2.** [161] *Punkt równowagi obiektu dynamicznego (2.14) opisanego regułami (2.8) z  $u(k) \equiv 0$  jest globalnie asymptotycznie stabilny, jeśli istnieje  $L$  symetrycznych i dodatnio określonych macierzy  $\mathbf{P}_l$ ,  $l \in L$  takich, że spełniony jest układ nierówności*

$$\mathbf{A}_i^T \mathbf{P}_m \mathbf{A}_i - \mathbf{P}_l < 0, \quad \text{dla każdego } (l, m) \in \Omega, \quad i \in K(l). \quad (2.16)$$

*Dowód.* Rozumowanie jest podobne do zastosowanego w dowodzie twierdzenia 2.1, jedynie należy zastosować opis dynamiki obiektu przez (2.14) i odcinkowo-kwadratowe funkcje Lapunowa

$$V(k) = x(k)^T \mathbf{P}_l x(k), \quad x(k) \in S_l, \quad l \in L, \quad (2.17)$$

zamiast funkcji globalnej (2.12). □

Twierdzenia 2.1 i 2.2 formułują jedynie warunki dostateczne stabilności – niespełnienie takiego warunku nie rozstrzyga kwestii stabilności. Nie udało się jak dotąd sformułować warunków koniecznych i jednocześnie dostatecznych. Próbowano też uzyskać sformułowanie warunków dostatecznych stabilności w oparciu o rozważenie cech lokalnych macierzy stanu  $\mathbf{A}_i$ , bez konieczności rozwiązywania równań czy nierówności globalnych, takich jak (2.11) czy (2.16). W pracy [19] wypisano warunki tego typu dokonując bezpośredniego szacowania normy stanu układu opisanego równaniem (2.10),

$$\begin{aligned}
 \|x(k+1)\| &= \left\| \sum_{i=1}^r \tilde{w}^i(k) \cdot \mathbf{A}_i x(k) \right\| \\
 &\leq \sum_{i=1}^r \tilde{w}^i(k) \|\mathbf{A}_i\| \|x(k)\| \\
 &\leq \max_{1 \leq i \leq r} \|\mathbf{A}_i\| \sum_{i=1}^r \tilde{w}^i(k) \|x(k)\| \\
 &= \max_{1 \leq i \leq r} \|\mathbf{A}_i\| \cdot \|x(k)\|, \tag{2.18}
 \end{aligned}$$

gdzie  $\|\mathbf{A}\|$  oznacza normę macierzy  $\mathbf{A}$  zgodną z normą wektorową. Stąd spełnienie warunku

$$\sum_{i=1}^r \tilde{w}^i(k) \|\mathbf{A}_i\| < 1, \tag{2.19}$$

lub nieco ostrzejszego warunku

$$\|\mathbf{A}_i\| < 1, \quad i = 1, 2, \dots, r, \tag{2.20}$$

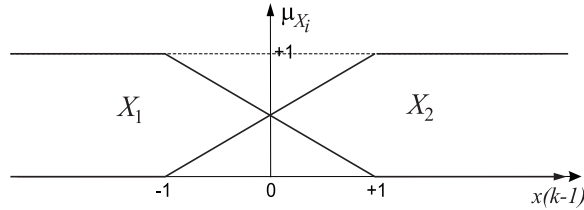
zapewnia asymptotyczną stabilność układu dynamicznego (2.10). Warunki powyższe okazują się jednak w praktyce na ogół bardzo konserwatywne, a przez to mało użyteczne.

Podany niżej, za [138], przykład pokazuje, że stabilność poszczególnych liniowych podsystemów  $x(k+1) = \mathbf{A}_i x(k)$  w ogólności nie wystarcza do stabilności nieliniowego systemu rozmytego. Konieczne są więc dla zapewnienia tej stabilności dodatkowe warunki, takie jak w twierdzeniu 2.1.

#### Przykład 2.4.

Rozważmy zbiory rozmyte  $X_1$  i  $X_2$  przedstawione na rysunku 2.19 oraz model rozmyty TS składający się z dwóch następujących reguł

$$\begin{aligned}
 R_{ob}^1 : \text{JEŚLI } x(k-1) \text{ jest } X_1 \quad \text{TO } x^1(k+1) &= x(k) - 0.5x(k-1), \\
 R_{ob}^2 : \text{JEŚLI } x(k-1) \text{ jest } X_2 \quad \text{TO } x^2(k+1) &= -x(k) - 0.5x(k-1),
 \end{aligned}$$



Rysunek 2.19. Zbiory rozmyte modelu, przykład 2.4

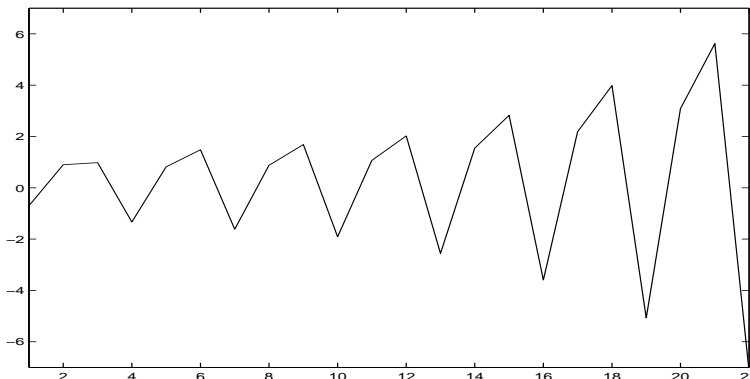
które są równoważne regułom (2.8) z macierzami stanu

$$\mathbf{A}_1 = \begin{bmatrix} 1 & -0.5 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} -1 & -0.5 \\ 1 & 0 \end{bmatrix}.$$

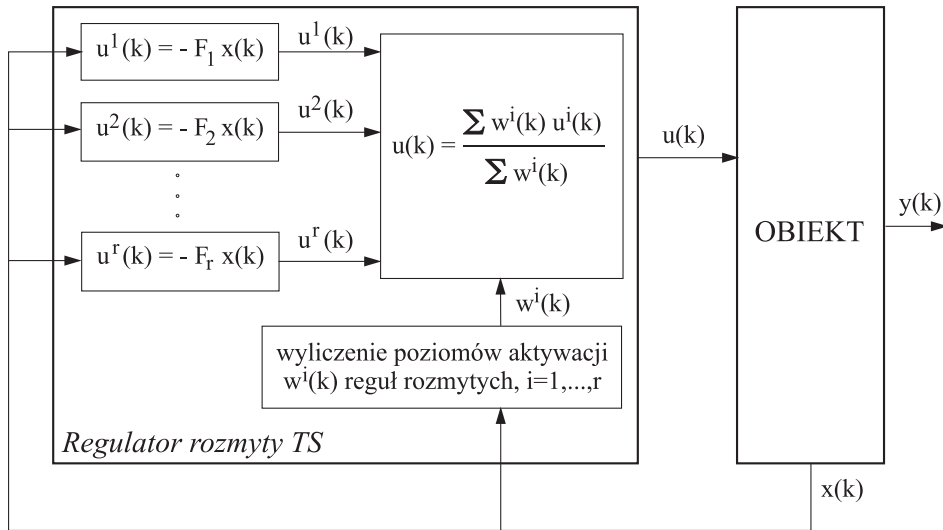
Wartości własne macierzy  $\mathbf{A}_1$  i  $\mathbf{A}_2$  wynoszą odpowiednio  $0.5 \pm 0.5i$  oraz  $-0.5 \pm 0.5i$ , są to więc macierze układów asymptotycznie stabilnych. Natomiast uruchamiając system rozmyty z punktu początkowego  $x_1 = 0.9$ ,  $x_2 = -0.7$  uzyskujemy przebieg pokazany na rysunku 2.20, jednoznacznie pokazujący niestabilność rozmytego modelu nieliniowego. Założenia twierdzenia 2.1 nie mogą zatem być w tym przypadku spełnione. I rzeczywiście, jak łatwo sprawdzić macierz  $\mathbf{A}_1\mathbf{A}_2$ ,

$$\mathbf{A}_1\mathbf{A}_2 = \begin{bmatrix} -1.5 & -0.5 \\ -1 & -0.5 \end{bmatrix},$$

ma wartości własne równe  $\lambda_1 = 0.134$ ,  $\lambda_2 = 1.866$ , a więc nie jest macierzą stabilnego dyskretnego układu dynamicznego – stąd nie istnieje macierz  $\mathbf{P}$  spełniająca założenia twierdzenia 2.1.  $\square$



Rysunek 2.20. Niestabilna trajektoria stanu w przykładzie 2.4



Rysunek 2.21. Struktura układu regulacji z regulatorem rozmytym TS ze sprzężeniem od stanu

Dla każdego z lokalnych liniowych modeli dynamicznych występujących w następniku reguł (2.8), tj. dla każdego z obszarów modelu rozmytego, można zaprojektować metodą standardową liniowy regulator ze sprzężeniem od zmiennych stanu. Uzyskamy w ten sposób reguły regulatora rozmytego TS w postaci

$$R_{reg}^j : \quad \text{JEŚLI } x_1(k) \text{ jest } A_1^j \text{ i } x_2(k) \text{ jest } A_2^j \text{ i } \dots \text{ i } x_{n_x}(k) \text{ jest } A_{n_x}^j \\ \text{TO } u^j(k) = -\mathbf{F}_j x(k), \quad (2.21)$$

gdzie  $\mathbf{F}_j$  są macierzami współczynników sprzężeń od stanu,  $j = 1, 2, \dots, r$ . Kompletny nieliniowy regulator rozmyty TS będzie opisany zależnością

$$u(k) = - \sum_{j=1}^r \tilde{w}^j(k) \mathbf{F}_j x(k). \quad (2.22)$$

Struktura układu regulacji z takim regulatorem przedstawiona jest na rysunku 2.21.

Modelując układ regulacji możemy, w ogólności, użyć do opisu obiektu modelu rozmytego TS innego niż wykorzystany do projektowania regulatora rozmytego. Np. modelu o innej liczbie partycji dla pewnych zmiennych stanu  $x_j$  (innej liczbie i kształcie odpowiadających tym zmiennym zbiorów rozmytych), a stąd innej liczbie i poprzednikach reguł. Jeśli liczbę tych reguł

oznaczymy przez  $ro$ , a poziomy ich aktywacji przez  $w_o^i(k)$ ,  $i = 1, \dots, ro$ , to wyjście modelu dane będzie następującym wzorem, analogicznym do (2.9),

$$x(k+1) = \frac{\sum_{i=1}^{ro} w_o^i(k) [\mathbf{A}_i x(k) + \mathbf{B}_i u(k)]}{\sum_{l=1}^{ro} w_o^l(k)} = \sum_{i=1}^{ro} \tilde{w}_o^i(k) [\mathbf{A}_i x(k) + \mathbf{B}_i u(k)], \quad (2.23)$$

gdzie  $\tilde{w}_o^i(k)$  to unormowane poziomy aktywacji reguł. Oczywiście, jeśli model użyty do opisu obiektu jest inny niż model (2.8) wykorzystany do projektowania regulatora, to i jego macierze  $\mathbf{A}_i$  i  $\mathbf{B}_i$  są w ogólności inne (choć nie wprowadzamy innych oznaczeń, bo nie prowadzi to do nieporozumień: przy analizie układu regulacji występują jedynie macierze modelu obiektu (2.23) i macierze  $\mathbf{F}_j$  regulatora). Wstawiając do (2.23) równania regulatora (2.22) dostajemy opis zamkniętego układu regulacji w postaci

$$x(k+1) = \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(k) \tilde{w}^j(k) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) x(k). \quad (2.24)$$

Do badania stabilności układu dynamicznego (2.24) można bezpośrednio zastosować twierdzenie 2.1, przyjmując w jego sformułowaniu zamiast  $\mathbf{A}_i$  macierze  $\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j$ ,  $i = 1, 2, \dots, ro$ ,  $j = 1, 2, \dots, r$ .

Jeśli natomiast modelując przedstawiony na rys. 2.21 zamknięty układ regulacji zastosujemy do opisu obiektu ten sam model, który wykorzystano do projektowania regulatora rozmytego (*z tą samą liczbą i poprzednikami reguł*), to formuła opisująca dynamikę układu regulacji będzie postaci

$$x(k+1) = \sum_{i=1}^r \sum_{j=1}^r \tilde{w}^i(k) \tilde{w}^j(k) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) x(k). \quad (2.25)$$

Uzyskamy wówczas nieco wygodniejsze, prostsze sformułowanie warunków stabilności, wykorzystując równość

$$\tilde{w}^i(k) \tilde{w}^j(k) = \tilde{w}^j(k) \tilde{w}^i(k), \quad i, j = 1, 2, \dots, r,$$

wynikającą z faktu stosowania takich samych poprzedników w regułach zarówno modelu obiektu, jak i regulatora. Wzór (2.25) zapisać bowiem wówczas możemy w postaci

$$x(k+1) = \sum_{i=1}^r \tilde{w}^i(k) \tilde{w}^i(k) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i) x(k) + 2 \sum_{i < j}^r \tilde{w}^i(k) \tilde{w}^j(k) \mathbf{D}_{ij} x(k), \quad (2.26)$$

gdzie

$$\mathbf{D}_{ij} = \frac{1}{2} [(\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) + (\mathbf{A}_j - \mathbf{B}_j \mathbf{F}_i)], \quad i < j, \quad i, j = 1, 2, \dots, r, \quad (2.27)$$

zaś symbol  $\sum_{i<j}^r$  oznacza sumowanie po wszystkich parach indeksów  $i, j$  takich, że  $i < j$ ,  $i, j = 1, 2, \dots, r$ , np.

$$\sum_{i<j}^3 \mathbf{D}_{ij} = \mathbf{D}_{12} + \mathbf{D}_{13} + \mathbf{D}_{23}.$$

Stąd, bezpośrednio z twierdzenia 2.1, uzyskujemy

**Wniosek 2.3.** *Punkt równowagi układu regulacji (2.25) opisanego regularami modelu obiektu (2.8) i regulatora (2.21) jest globalnie asymptotycznie stabilny, jeśli istnieje macierz dodatnio określona  $\mathbf{P}$  spełniająca warunki*

$$(\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i)^T \mathbf{P} (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i) - \mathbf{P} < 0, \quad i = 1, \dots, r, \quad (2.28)$$

$$\mathbf{D}_{ij}^T \mathbf{P} \mathbf{D}_{ij} - \mathbf{P} < 0, \quad i < j, \quad i, j = 1, \dots, r, \quad (2.29)$$

dla wszystkich par  $i, j$  z wyjątkiem takich, dla których zawsze  $w^i(k)w^j(k) = 0$  (tzn. reguły o indeksach  $i$  oraz  $j$  wykluczają się).  $\square$

W pracy [137] uzyskano pewne osłabienie powyższego rezultatu, dowód przebiega podobnie jak w twierdzeniu 2.1 z wykorzystaniem tej samej postaci funkcji Lapunowa.

**Twierdzenie 2.4.** *Niech maksymalna liczba reguł aktywowanych jednocześnie w układzie regulacji (2.25) opisanym regularami modelu obiektu (2.8) i regulatora (2.21) jest nie większa od  $s$ ,  $1 < s \leq r$ . Punkt równowagi tego układu jest globalnie asymptotycznie stabilny, jeśli istnieją macierze dodatnio określona  $\mathbf{P}$  i dodatnio półokreślona  $\mathbf{Q}$  spełniające warunki*

$$(\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i)^T \mathbf{P} (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i) - \mathbf{P} + (s - 1)\mathbf{Q} < 0, \quad i = 1, \dots, r, \quad (2.30)$$

$$\mathbf{D}_{ij}^T \mathbf{P} \mathbf{D}_{ij} - \mathbf{P} - \mathbf{Q} \leq 0 \quad i < j, \quad i, j = 1, \dots, r, \quad (2.31)$$

dla wszystkich par  $i, j$  z wyjątkiem takich, dla których  $w^i(k)w^j(k) = 0$  dla każdej chwili  $k$ .  $\square$

Zauważmy, że warunki (2.30), (2.31) redukują się do (2.28), (2.29), jeśli przyjmiemy  $\mathbf{Q} = \mathbf{0}$ .

Sformułowania warunków stabilności przedstawione powyżej opierają się na globalnej kwadratowej funkcji Lapunowa. W wielu przypadkach można uzyskać słabsze warunki stosując odcinkowo-kwadratowe funkcje lapunowa zamiast jednej globalnej, jak to pokazano w twierdzeniu 2.2. Aby zastosować ten rezultat do przypadku ze sprzężeniem od stanu, obszar stanów procesu trzeba podzielić na komórki aktywności  $S_l$ ,  $l = 1, \dots, L$ ,

jak w przedstawianej poprzednio sytuacji obiektu niesterowanego. Przypomnijmy, że  $K(l)$  oznacza zbiór zawierający indeksy wszystkich reguł rozmytych z niezerowymi poziomami aktywności wewnątrz  $S_l$ . Regulator ze sprzężeniem od stanu przyjmuje teraz postać (por. z (2.22))

$$u(k) = - \sum_{j \in K(l)} \tilde{w}^j(k) \mathbf{F}_j x(k), \quad x(k) \in S_l, \quad l \in L. \quad (2.32)$$

Dynamikę systemu ze sprzężeniem zwrotnym można teraz opisać następująco:

$$x(k+1) = \sum_{i \in K(l)} \tilde{w}^i(k) [\mathbf{A}_i - \mathbf{B}_i \sum_{j \in K(l)} \tilde{w}^j(k) \mathbf{F}_j] x(k), \quad x(k) \in S_l, \quad l \in L, \quad (2.33)$$

gdzie  $\sum_{j \in K(l)} \tilde{w}^j(k) = 1$ . Korzystając z ostatniej równości, zależność (2.33) może być przedstawiona w postaci

$$x(k+1) = \sum_{i \in K(l)} \sum_{j \in K(l)} \tilde{w}^i(k) \tilde{w}^j(k) [\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j] x(k), \quad x(k) \in S_l, \quad l \in L, \quad (2.34)$$

która jest rozproszoną (podzieloną) postacią opisu ogólnej dynamiki (2.25), odpowiadającą podziałowi całego obszaru działania obiektu na  $L$  komórek stałej aktywności  $S_l$ . Twierdzenie 2.2 można teraz bezpośrednio zastosować do rozważanego przypadku regulacji ze sprzężeniem zwrotnym od stanu, co prowadzi do następującego rezultatu:

**Wniosek 2.5.** *Punkt równowagi układu regulacji (2.34) opisanego regułami rozmytymi modelu obiektu (2.8) i regulatora (2.21) jest globalnie asymptotycznie stabilny, jeśli istnieje  $L$  symetrycznych i dodatnio określonych macierzy  $\mathbf{P}_l$ ,  $l \in L$  takich, że spełniony jest następujący układ nierówności*

$$[\mathbf{A}_i^T - \mathbf{B}_i \mathbf{F}_j] \mathbf{P}_m [\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j] - \mathbf{P}_l < 0, \quad \text{dla każdych } (l, m) \in \Omega, \quad i, j \in K(l), \quad (2.35)$$

gdzie  $\Omega$  oznacza zbiór par indeksów opisujących wszystkie możliwe jednokrokowe przejścia stanu obiektu między komórkami  $S_l$  w układzie regulacji, zob. (2.15).  $\square$

Słabym punktem powyższego rezultatu jest konieczność zdefiniowania, a priori, wszystkich możliwych jednokrokowych przejść stanu między komórkami aktywacji, tzn. zdefiniowania zbioru  $\Omega$ . Trzeba podkreślić, że możliwe jednokrokowe przejścia stanu w układzie regulacji mogą różnić się od takich przejść dla obiektu niesterowanego (autonomicznego), a te pierwsze trzeba uwzględnić w definicji zbioru  $\Omega$  w powyższym wniosku. Oczywiście,

niedoszacowanie zbioru  $\Omega$  może być niebezpieczne, ponieważ akurat opuszczone przejścia mogą być tymi prowadzącymi do niestabilności. Dlatego też przeszacowanie jest bardziej prawdopodobne w praktycznym zastosowaniu, co z kolei może prowadzić do zwiększonej liczby nierówności (2.35) – a im więcej tych nierówności, tym bardziej ograniczające warunki stabilności.

Przedstawione dotychczas projektowanie i analiza dotyczyły rozmytych układów Takagi-Sugeno z lokalnie zaprojektowanymi macierzami sprzężeń od stanu, jak w pierwszej części rozdziału 2.2. Podejście to wydaje się być przekonującym i naturalnym uogólnieniem tradycyjnego podejścia do projektowania sterowania ze sprzężeniem od stanu dla liniowych modeli obiektu, w wybranym punkcie pracy. Możliwe jest natomiast również podejście globalne polegające na projektowaniu macierzy współczynników sprzężeń zwrotnych  $\mathbf{F}_i$ ,  $i = 1, 2, \dots, r$ , zapewniających stabilność układu regulacji w oparciu o rozwiązanie układu nierówności podanych w twierdzeniu 2.4, względem  $\mathbf{P}$ ,  $\mathbf{Q}$  i  $\mathbf{F}_1, \dots, \mathbf{F}_r$ . Układ nierówności (2.30)-(2.31) trzeba wówczas przeformułować do postaci układu liniowych nierówności macierzowych i zastosować np. *LMI Toolbox* do pakietu MATLAB, zob. [137]. Nie będziemy tego podejścia tutaj przedstawiać, gdyż zdaniem autora bardziej naturalny, prawidłowy jest sposób projektowania regulatora rozmytego TS przedstawiony we wstępnej części rozdziału 2.2, polegający na zdecentralizowanym, lokalnym projektowaniu poszczególnych lokalnych regulatorów (w rozważanym przypadku macierzy sprzężeń od stanu). Tym bardziej, że w razie niezadowalającego wyniku trzeba korygować nie tylko wartości współczynników sprzężeń, ale sprawdzić prawidłowość zaprojektowania liczby, położenia i kształtu funkcji przynależności.

W zastosowaniach praktycznych często nie jest dostępny pomiarowo cały wektor zmiennych stanu. W takiej sytuacji klasyczna teoria regulacji liniowo-kwadratowej sugeruje zastosowanie obserwatora stanu Luenbergera lub filtru Kalmana. Podobnie można postąpić przy projektowaniu układu regulacji z regulatorem TS ze sprzężeniem od stanu, konstruuując obserwator stanu jako system rozmyty TS o strukturze reguł identycznej jak reguły modelu i regulatora oraz następnikach funkcyjnych będących klasycznymi formułami liniowych obserwatorów stanu. Przedstawienie zagadnienia projektowania rozmytego obserwatora stanu oraz analizy stabilności układu regulacji z takim obserwatorem pominiemy, odsyłając zainteresowanego czytelnika do literatury [137]. Alternatywą jest zastosowanie rozszerzonego obserwatora Luenbergera lub rozszerzonego filtru Kalmana, które przedstawimy w rozdziale 3.

### Przykład 2.5.

Rozważmy ponownie model rozmyty TS obiektu przedstawiony w przy-

kładzie 2.4, uzupełniony o zmienną sterującą  $u$ . Zbiory rozmyte  $X_1$  i  $X_2$  zdefiniowane trapezoidalnymi funkcjami przynależności przedstawiono na rys. 2.19, natomiast reguły modelu można przedstawić w postaci

$$\begin{aligned} R_{ob}^1 : \text{JEŚLI } x_2(k) \text{ jest } X_1 \text{ TO } x^1(k+1) &= \mathbf{A}_1 x(k) + \mathbf{B}u(k), \\ R_{ob}^2 : \text{JEŚLI } x_2(k) \text{ jest } X_2 \text{ TO } x^2(k+1) &= \mathbf{A}_2 x(k) + \mathbf{B}u(k), \end{aligned}$$

gdzie  $x(k) = [x_1(k) \ x_2(k)]^T \in \mathbb{R}^2$ ,  $u(k) \in \mathbb{R}$ ,

$$\mathbf{A}_1 = \begin{bmatrix} 1 & -0.5 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} -1 & -0.5 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

W przykładzie 2.4 pokazano, że układ autonomiczny (bez oddziaływania sterowania) generuje, dla pewnych warunków początkowych, niestabilne przebiegi stanu.

Zaprojektujemy regulator rozmyty TS ze sprzężeniem od stanu zapewniający stabilną pracę układu regulacji. Reguły regulatora będą postaci

$$\begin{aligned} R_{reg}^1 : \text{JEŚLI } x_2(k) \text{ jest } X_1 \text{ TO } u(k) &= -\mathbf{F}_1 x(k), \\ R_{reg}^2 : \text{JEŚLI } x_2(k) \text{ jest } X_2 \text{ TO } u(k) &= -\mathbf{F}_2 x(k). \end{aligned}$$

Najprościej będzie założyć identyczne położenie wartości własnych macierzy stanu  $\mathbf{A}_1 - \mathbf{B}\mathbf{F}_1$  i  $\mathbf{A}_2 - \mathbf{B}\mathbf{F}_2$  poszczególnych lokalnych układów ze sprzężeniem zwrotnym. Przyjmując wartości własne równe 0.5 i 0.4 wyliczono elementy wektorów  $\mathbf{F}_1$  i  $\mathbf{F}_2$  (wykorzystując funkcję *place* z *Control System Toolbox* pakietu MATLAB)

$$\mathbf{F}_1 = \begin{bmatrix} 0.1 & -0.3 \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} -1.9 & -0.3 \end{bmatrix}.$$

Rozważamy sytuację identycznych reguł regulatora i modelu obiektu użytego do symulacji układu regulacji. Stąd, dla badania stabilności wystarczy wyliczyć 3 macierze, które dzięki przyjętym założeniom mają identyczne wartości

$$\begin{aligned} \mathbf{A}_1 - \mathbf{B}\mathbf{F}_1 &= \mathbf{A}_2 - \mathbf{B}\mathbf{F}_2 = \\ &= \mathbf{D}_{12} = \frac{1}{2}[(\mathbf{A}_1 - \mathbf{B}\mathbf{F}_2) + (\mathbf{A}_2 - \mathbf{B}\mathbf{F}_1)] = \begin{bmatrix} 0.9 & -0.2 \\ 1 & 0 \end{bmatrix}. \end{aligned}$$

Zamknięty układ regulacji rozmyty TS będzie stąd opisany jednym równaniem

$$x(k+1) = \mathbf{D}_{12}x(k),$$

i jest oczywiście stabilny, o założonych wartościach własnych macierzy stanu równych 0.5 i 0.4.

Dla pełniejszej ilustracji badania stabilności układu zamkniętego powtórzmy projektowanie regulatora przyjmując nieco inny wektor sprzężeń zwrotnych w drugim podobieckie, a mianowicie  $\mathbf{F}_2 = [-2.1 \ -0.2]$ , co odpowiada wartościom własnym macierzy  $\mathbf{A}_2 - \mathbf{BF}_2$  równym 0.5 i 0.6. Mamy teraz

$$\begin{aligned}\mathbf{A}_1 - \mathbf{BF}_1 &= \begin{bmatrix} 0.9 & -0.2 \\ 1 & 0 \end{bmatrix}, \\ \mathbf{A}_2 - \mathbf{BF}_2 &= \begin{bmatrix} 1.1 & -0.3 \\ 1 & 0 \end{bmatrix}, \\ \mathbf{D}_{12} &= \frac{1}{2}[(\mathbf{A}_1 - \mathbf{BF}_2) + (\mathbf{A}_2 - \mathbf{BF}_1)] = \begin{bmatrix} 1 & -0.25 \\ 1 & 0 \end{bmatrix}.\end{aligned}$$

Warunkiem dostatecznym stabilności układu zamkniętego jest, na mocy wniosku 2.3, istnienie symetrycznej dodatnio określonej macierzy  $\mathbf{P}$  spełniającej układ nierówności

$$\begin{aligned}(\mathbf{A}_1 - \mathbf{BF}_1)^T \mathbf{P} (\mathbf{A}_1 - \mathbf{BF}_1) - \mathbf{P} &< \mathbf{0}, \\ (\mathbf{A}_2 - \mathbf{BF}_2)^T \mathbf{P} (\mathbf{A}_2 - \mathbf{BF}_2) - \mathbf{P} &< \mathbf{0}, \\ (\mathbf{D}_{12})^T \mathbf{P} \mathbf{D}_{12} - \mathbf{P} &< \mathbf{0}.\end{aligned}$$

Stosując *LMI Toolbox* pakietu MATLAB sprawdzono, że taka macierz istnieje, uzyskując

$$\mathbf{P} = \begin{bmatrix} 1.9755 & -0.7968 \\ -0.7968 & 0.9459 \end{bmatrix}.$$

□

### 2.2.2. Regulatory rozmyte TS ze sprzężeniem od wyjścia

W układach regulacji procesów technologicznych stosuje się tradycyjnie regulatory liniowe ze sprzężeniem od wyjścia, przede wszystkim dobrze znane regulatory typu PID. Wraz z coraz szerszym stosowaniem do bieżącego sterowania struktur warstwowych z bieżącą optymalizacją punktów pracy rośnie potrzeba stosowania regulatorów nieliniowych, zdolnych pracować w szerokim zakresie zmian punktów pracy. Stąd w zastosowaniach przemysłowych istotne jest dysponowanie stosunkowo prostym regulatorem nieliniowym, najlepiej będącym nieliniowym uogólnieniem regulatora PID. Konstrukcję dyskretnych rozmytych regulatorów typu TS ze sprzężeniem

od wyjścia, spełniających powyższe postulaty, zaproponowano w pracy [31], zob. też [32, 33], gdzie nazywano je regulatorami *wieloobszarowymi*.

Rozważmy model obiektu rozmyty TS, który wykorzystamy do projektowania regulatora rozmytego. Reguły tego modelu przyjmujemy w postaci

$$\begin{aligned} R_{ob}^i : \text{JEŚLI } y(k) \text{ jest } A_0^i \text{ i } y(k-1) \text{ jest } A_1^i \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^i \\ \text{i } u(k-1) \text{ jest } B_1^i \text{ i } \dots \text{ i } u(k-m_R) \text{ jest } B_{m_R}^i \\ \text{TO } y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \dots + a_{n_A}^i y(k-n_A+1) + \\ + b_0^i u(k) + b_1^i u(k-1) + \dots + b_{m_B}^i u(k-m_B), \quad (2.36) \end{aligned}$$

gdzie  $i = 1, \dots, r$  indeksuje reguły i jednocześnie zbiory (obszary) rozmyte w dziedzinie modelu TS (w ogólności wielowymiarowe),  $y(k)$  jest wartością wyjścia obiektu w chwili  $k$ -tej,  $u(k)$  jest wartością sterowania (wejścia) w chwili  $k$ -tej,  $A_j^i \in \mathbb{Y}_j$ ,  $B_j^i \in \mathbb{U}_j$ , zaś  $a_j^i$  i  $b_j^i$  są współczynnikami funkcji następników reguł,  $i = 1, \dots, r$ . Elementami każdego ze zbiorów  $\mathbb{Y}_j = \{Y_{j1}, \dots, Y_{jr_{y_j}}\}$  są zbiory rozmyte pokrywające obszar zmiennej  $y(k-j)$ ,  $j = 0, \dots, n_R$ , analogicznie  $\mathbb{U}_j = \{U_{j1}, \dots, U_{jr_{u_j}}\}$  dla  $u(k-j)$ ,  $j = 1, \dots, m_R$ .

Przedstawiona sytuacja jest najbardziej ogólna, najczęściej mamy do czynienia z przypadkiem znacznie prostszym, kiedy

$$\mathbb{Y}_0 = \dots = \mathbb{Y}_{n_R} = \mathbb{Y}, \quad \mathbb{Y} = \{Y_1, \dots, Y_{r_y}\},$$

czyli takie same są partycje zakresu zmiennej wyjściowej dla chwili aktualnej i chwil wstecz, tzn. zmiennych  $y(k)$ ,  $y(k-1)$ ,  $\dots$ ,  $y(k-n_R)$ ; analogicznie takie same są też partycje sterowania,  $\mathbb{U}_1 = \dots = \mathbb{U}_{m_R} = \mathbb{U}$ ,  $\mathbb{U} = \{U_1, \dots, U_{r_u}\}$ .

Dynamiczne modele liniowe występujące w następnikach reguł (2.36) to często stosowane modele postaci ARX. Modele te obejmują również przypadek układu z opóźnieniem równym kilku, powiedzmy  $\tau$ , okresom próbkowania. Wówczas jedynie w następnikach reguł wystąpi  $b_0^i = b_1^i = \dots = b_\tau^i = 0$ .

Wyjście modelu rozmytego wyliczane jest zgodnie z ogólnym wzorem (2.5), tj.

$$y(k+1) = \frac{\sum_{i=1}^r w^i(k) y^i(k+1)}{\sum_{l=1}^r w^l(k)}, \quad (2.37)$$

gdzie  $w^i(k)$  są poziomami aktywacji poszczególnych reguł (2.36) w chwili  $k$ -tej,

$$w^i(k) = \prod_{j=0}^{n_R} \mu_{A_j^i}(y(k-j)) \prod_{p=1}^{m_R} \mu_{B_p^i}(u(k-p)). \quad (2.38)$$

Dla modelu obiektu opisanego regułami (2.36) sformułujemy następujące reguły regulatora rozmytego TS:

$$\begin{aligned} R_{reg}^j : \text{JEŚLI } & y(k) \text{ jest } A_0^j \text{ i } y(k-1) \text{ jest } A_1^j \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^j \\ & \text{i } u(k-1) \text{ jest } B_1^j \text{ i } \dots \text{ i } u(k-m_R) \text{ jest } B_{m_R}^j \\ \text{TO } & u^j(k) = c_1^j e(k) + c_2^j e(k-1) + \dots + c_{n_C}^j e(k-n_C+1) + \\ & + d_1^j u(k-1) + d_2^j u(k-2) + \dots + d_{m_D}^j u(k-m_D), \end{aligned} \quad (2.39)$$

gdzie  $j = 1, \dots, r$  indeksuje reguły, których poprzedniki są identyczne jak w modelu (2.36),  $e(k) = y^{zad}(k) - y(k)$  jest uchybem regulacji w chwili  $k$ , zaś  $c_j^j$  i  $d_j^j$  są współczynnikami funkcji następników reguł. Występująca w następnikach reguł (2.39) postać dyskretnego algorytmu regulacji jest dość ogólna, jej przypadkiem szczególnym są algorytmy *dyskretnego regulatora PID*, czy też *analitycznych wersji regulatorów predykcyjnych DMC i GPC* (omawianych w rozdziale 3).

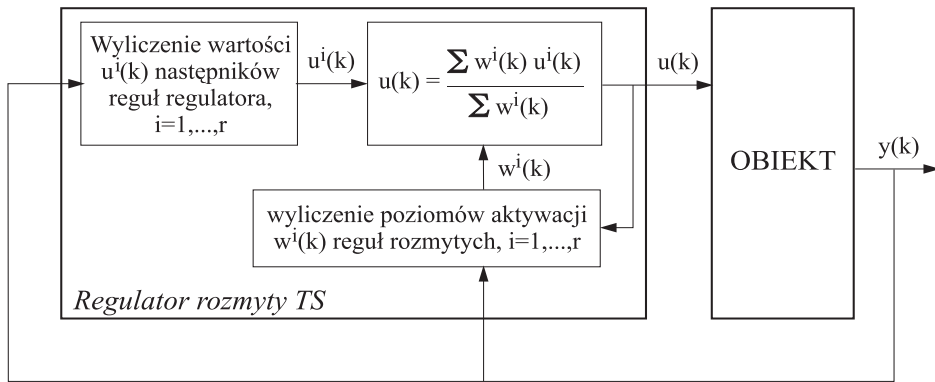
Sygnal wyjściowy regulatora przyjmuje standardową dla struktur TS postać

$$u(k) = \frac{\sum_{j=1}^r w^j(k) u^j(k)}{\sum_{l=1}^r w^l(k)} = \sum_{j=1}^r \tilde{w}^j(k) u^j(k), \quad (2.40)$$

gdzie  $w^j(k)$  są poziomami aktywacji poszczególnych reguł (2.39) w chwili  $k$ -tej, określonymi wzorami (2.38) identycznymi jak dla reguł (2.36), ze względu na takie same poprzedniki reguł. Struktura układu regulacji z przedstawionym powyżej regulatorem rozmytym TS ze sprzężeniem od wyjścia obiektu jest przedstawiona na rysunku 2.22.

Zwróćmy uwagę, że w poprzednikach reguł regulatora (2.39) nie występuje warunek „ $u(k)$  jest  $B_0^j$ ”, prowadziłyby to bowiem do *niespójności logicznej*. Albowiem, w chwili  $k$ -tej w następnikach funkcyjnych reguł regulatora dopiero wyznaczana ma być wartość sterowania  $u(k)$ , tak więc  $u(k)$  nie może występować w tej chwili w poprzednikach. Stąd najprościej jest przyjąć poprzedniki w regułach regulatora bez warunku „ $u(k)$  jest  $B_0^j$ ”, jak to uczyniliśmy w (2.39). Jest to naturalne szczególnie przy sterowaniu dyskretnym schodkowym, z ekstrapolatorem zerowego rzędu. Wówczas bowiem tuż przed wyznaczeniem sterowania w chwili  $k$ -tej, tj. na okres próbkowania zaczynający się w tej chwili, mierzymy wartość wyjścia obiektu  $y(k)$ , ale obiekt jest jeszcze pod działaniem sterowania  $u(k-1)$ .

Do badań symulacyjnych czy teoretycznych zamkniętego układu regulacji z regulatorem rozmytym określonym regułami (2.39), przedstawionego na rysunku 2.22, można użyć modelu obiektu innego niż wykorzystany do



Rysunek 2.22. Struktura układu regulacji z regulatorem rozmytym TS ze sprzężeniem od wyjścia

konstrukcji regulatora model dany regułami (2.36). W szczególności, może to być nieco inny model rozmyty TS, z liczbą reguł  $ro$  (w ogólności  $ro \neq r$ ) i postaci

$R_{ob}^i$  : JEŚLI  $y(k)$  jest  $A_0^i$  i  $y(k-1)$  jest  $A_1^i$  i  $\dots$  i  $y(k-n_O)$  jest  $A_{n_O}^i$   
i  $u(k)$  jest  $B_0^i$  i  $u(k-1)$  jest  $B_1^i$  i  $\dots$  i  $u(k-m_O)$  jest  $B_{m_O}^i$

TO  $y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \dots + a_{n_A}^i y(k-n_A+1) +$   
 $+ b_0^i u(k) + b_1^i u(k-1) + \dots + b_{m_B}^i u(k-m_B)$ , (2.41)

gdzie  $i = 1, \dots, ro$ , zaś parametry  $n_A$  i  $m_B$  reguł (2.36) i (2.41) mogą przyjmować różne wartości (nie wprowadziliśmy różnych oznaczeń, by nie mnożyć ich liczby; nie prowadzi to do nieporozumień, gdyż w analizie układu regulacji wystąpi jeden model obiektu (2.41)). Poziomy aktywacji reguł (2.41) oznaczymy przez  $w_o^i(k)$ , a ich unormowane wartości przez  $\tilde{w}_o^i(k)$ ,

$$y(k+1) = \frac{\sum_{i=1}^{ro} w_o^i(k) y^i(k+1)}{\sum_{l=1}^{ro} w_o^l(k)} = \sum_{i=1}^{ro} \tilde{w}_o^i(k) y^i(k+1). \quad (2.42)$$

Zwróćmy uwagę, że w poprzednikach reguł modelu obiektu (2.41) występuje, w ogólności, również warunek „ $u(k)$  jest  $B_0^i$ ”, gdyż następniki tych reguł wyznaczają wartość wyjścia na chwilę następną,  $(k+1)$ -szą, i nie grozi tu wystąpienie niespójności logicznej.

Jeśli jednak w określonej sytuacji wskazane jest, aby w poprzednikach reguł modelu wykorzystanego do projektowania regulatora, a więc i w poprzednikach reguł regulatora, występował warunek „ $u(k)$  jest  $B_0^j$ ”, to jest

to możliwe. Jednakże, dla wyeliminowania niezgodności logicznej trzeba wówczas zastosować w następnikach reguł algorytm regulacji wyznaczający sterowanie z wyprzedzeniem o jeden okres próbkowania (zob. [165, 33]), tzn. zastosować reguły regulatora postaci

$$\begin{aligned}
 R_{reg}^j : \text{ JEŚLI } & y(k) \text{ jest } A_0^j \text{ i } y(k-1) \text{ jest } A_1^j \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^j \\
 & \text{ i } u(k) \text{ jest } B_0^j \text{ i } u(k-1) \text{ jest } B_1^j \text{ i } \dots \text{ i } u(k-m_R) \text{ jest } B_{m_R}^j \\
 \text{ TO } & u^j(k+1) = c_1^j e(k) + c_2^j e(k-1) + \dots + c_{n_C}^j e(k-n_C+1) + \\
 & + d_1^j u(k) + d_2^j u(k-1) + d_3^j u(k-2) + \dots + d_{m_D}^j u(k-m_D+1).
 \end{aligned} \tag{2.43}$$

Warto dodać, że warunki na sterowania często w ogóle nie występują w poprzednikach reguł modelu obiektu, stąd m.in. rozważanie problemu występowania bądź nie warunku „ $u(k)$  jest  $B_0^i$ ” dla bardzo wielu zastosowań nie ma znaczenia. Kluczowymi zmiennymi poprzedników reguł modelu rozmytego TS są bowiem na ogół zmienne wyjściowe obiektu  $y(k)$ ,  $y(k-1)$ , .... Jest tak dlatego, że określają one, w pełni lub częściowo, stan obiektu dynamicznego. Dlatego dla istotnej klasy problemów wystarcza zastosowanie modelu obiektu (a stąd i regulatora) z regułami, w których warunki poprzedników określone są jedynie dla zmiennych wyjściowych, tj. reguł

$$\begin{aligned}
 R_{ob}^i : \text{ JEŚLI } & y(k) \text{ jest } A_0^i \text{ i } y(k-1) \text{ jest } A_1^i \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^i \\
 \text{ TO } & y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \dots + a_{n_A}^i y(k-n_A+1) + \\
 & + b_0^i u(k) + b_1^i u(k-1) + \dots + b_{m_B}^i u(k-m_B).
 \end{aligned} \tag{2.44}$$

Rozważymy teraz zagadnienie *stabilności zamkniętego układu regulacji* z regulatorem rozmytym TS i modelem obiektu też w postaci systemu rozmytego TS. Rozpoczniemy od ogólnego przypadku, kiedy model użyty do opisu obiektu, choć też rozmyty TS, jest jednakże inny niż model użyty do projektowania regulatora. Przyjmiemy model obiektu z regułami postaci (2.41), zaś regulator z regułami (2.39). Dla skrócenia zapisu będziemy w przedstawianych dalej wzorach oznaczać poziomy aktywacji reguł obiektu i regulatora przez  $w_o^i$ ,  $w^j$  zamiast w  $w_o^i(k)$ ,  $w^j(k)$ , a unormowane poziomy aktywacji, zob. (2.6), przez  $\tilde{w}_o^i$ ,  $\tilde{w}^j$ . Dla przejrzystości zapisu przekształceń algebraicznych przyjmiemy też, w odpowiednim momencie, założenie

$$\begin{aligned}
 n_A = n_C = n, \\
 m_B = m_D = m,
 \end{aligned} \tag{2.45}$$

które należy rozumieć następująco:

- $n = \max\{n_A, n_C\}$  i brakujące współczynniki  $a_p^i$  (jeśli  $n_A < n_C$ ) lub  $c_p^i$  (jeśli  $n_A > n_C$ ) są zerami; podobnie
- $m = \max\{m_B, m_D\}$  i brakujące współczynniki  $b_p^i$  (jeśli  $m_B < m_D$ ) lub  $d_p^i$  (jeśli  $m_B > m_D$ ) są zerami.

Wstawiając zależności opisujące lokalne modele obiektu do (2.42) uzyskujemy

$$y(k+1) = \sum_{i=1}^{r_o} \tilde{w}_o^i \left\{ \sum_{p=1}^{n_A} a_p^i y(k-p+1) + \sum_{q=0}^{m_B} b_q^i u(k-q) \right\}. \quad (2.46)$$

Analogicznie, wstawiając do (2.40) równania lokalnych regulatorów (2.39) dostajemy

$$\begin{aligned} u(k) &= \sum_{j=1}^r \tilde{w}^j \left\{ \sum_{t=1}^{n_C} c_t^j e(k-t+1) + \sum_{s=1}^{m_D} d_s^j u(k-s) \right\} \\ &= \sum_{j=1}^r \tilde{w}^j \left\{ \sum_{t=1}^{n_C} c_t^j y^{zad}(k-t+1) - \sum_{t=1}^{n_C} c_t^j y(k-t+1) + \sum_{s=1}^{m_D} d_s^j u(k-s) \right\}. \end{aligned} \quad (2.47)$$

Wstawiając teraz tę zależność do (2.46) i uwzględniając, że  $\sum_{j=1}^r \tilde{w}^j = 1$ , uzyskujemy równanie opisujące zamknięty układ regulacji

$$\begin{aligned} y(k+1) &= \sum_{i=1}^{r_o} \tilde{w}_o^i \sum_{j=1}^r \tilde{w}^j \left\{ \sum_{p=1}^{n_A} a_p^i y(k-p+1) + \sum_{q=1}^{m_B} b_q^i u(k-q) \right\} + \\ &+ \sum_{i=1}^{r_o} \tilde{w}_o^i b_0^i \sum_{j=1}^r \tilde{w}^j \left\{ \sum_{t=1}^{n_C} c_t^j y^{zad}(k-t+1) - \sum_{t=1}^{n_C} c_t^j y(k-t+1) + \right. \\ &\quad \left. + \sum_{s=1}^{m_D} d_s^j u(k-s) \right\}. \end{aligned}$$

Wprowadzając teraz umożliwiające zwięzły i przejrzysty zapis założenie (2.45) możemy, po odpowiednim pogrupowaniu składników, uzyskane równanie przedstawić w postaci

$$\begin{aligned} y(k+1) &= \sum_{i=1}^r \tilde{w}_o^i \sum_{j=1}^r \tilde{w}^j \left\{ \sum_{p=1}^n (a_p^i - b_0^i c_p^j) y(k-p+1) + \right. \\ &\quad \left. + \sum_{q=1}^m (b_q^i + b_0^i d_q^j) u(k-q) + \sum_{t=1}^n b_0^i c_t^j y^{zad}(k-t+1) \right\}. \end{aligned} \quad (2.48)$$

Zdefiniujemy teraz wektor stanu  $x(k) \in \mathbb{R}^{n+m-1}$  opisujący dynamikę daną równaniami (2.48) i (2.47), następująco

$$x(k) = [y(k), y(k-1), \dots, y(k-n+1), u(k-1), u(k-2), \dots, u(k-m)]^T. \quad (2.49)$$

Wówczas, definiując dla zwięzłości zapisu parametry

$$\begin{aligned} ac_p^{i,j} &= a_p^i - b_0^i c_p^j, & p &= 1, \dots, n, \\ bd_q^{i,j} &= b_q^i + b_0^i d_q^j, & q &= 1, \dots, m, \end{aligned}$$

i uwzględniając, że (2.47) jest równoważne równaniu

$$u(k) = \sum_{i=1}^{ro} \tilde{w}_o^i \sum_{j=1}^r \tilde{w}^j \left\{ \sum_{t=1}^n c_t^j [y^{zad}(k-t+1) - y(k-t+1)] + \sum_{s=1}^m d_s^j u(k-s) \right\},$$

przedstawimy opis zamkniętego układu regulacji w postaci

$$x(k+1) = \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i \tilde{w}^j \mathbf{A}_{ij} x(k) + \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i \tilde{w}^j \sum_{t=1}^n \mathbf{E}_{ij} \tilde{y}_{zad}(k), \quad (2.50)$$

gdzie

$$\mathbf{A}_{ij} = \begin{bmatrix} ac_1^{i,j} & ac_2^{i,j} & \cdots & ac_{n-1}^{i,j} & ac_n^{i,j} & bd_1^{i,j} & bd_2^{i,j} & \cdots & bd_{m-1}^{i,j} & bd_m^{i,j} \\ 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -c_1^j & -c_2^j & \cdots & -c_{n-1}^j & -c_n^j & d_1^j & d_2^j & \cdots & d_{m-1}^j & d_m^j \\ 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad (2.51)$$

$$\mathbf{E}_{ij} = \begin{bmatrix} b_0^i c_1^j & b_0^i c_2^j & \cdots & b_0^i c_{n-1}^j & b_0^i c_n^j \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ c_1^j & c_2^j & \cdots & c_{n-1}^j & c_n^j \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad (2.52)$$

zaś wektor  $\tilde{y}^{zad}(k)$  zdefiniowany jest jako

$$\tilde{y}^{zad}(k) = [y^{zad}(k) \ y^{zad}(k-1) \ \cdots \ y^{zad}(k-n+1)]^T. \quad (2.53)$$

Dynamika układu autonomicznego opisana jest zależnością

$$x(k+1) = \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(k) \tilde{w}^j(k) \mathbf{A}_{ij} x(k), \quad (2.54)$$

dla której, stosując twierdzenie Lapunowa, możemy uzyskać rezultat analogiczny do twierdzenia 2.1 (jest to bezpośrednie zastosowanie twierdzenia 2.1, z dokładnością do wyłączenia macierzy  $\mathbf{A}_{ij}$  o indeksach, dla których zawsze  $w_o^i(k)w^j(k) = 0$ ):

**Wniosek 2.6.** *Punkt równowagi układu regulacji rozmytego TS opisanego zależnością (2.54) jest globalnie asymptotycznie stabilny, jeśli istnieje macierz symetryczna dodatnio określona  $\mathbf{P}$  taka, że dla każdej macierzy  $\mathbf{A}_{ij}$  spełnione jest równanie*

$$\mathbf{A}_{ij}^T \mathbf{P} \mathbf{A}_{ij} - \mathbf{P} < \mathbf{0}, \quad (2.55)$$

dla wszystkich par  $(i, j)$ ,  $i = 1, 2, \dots, ro$ ,  $j = 1, 2, \dots, r$  z wyjątkiem takich, dla których zawsze  $w_o^i(k)w^j(k) = 0$ .  $\square$

Jeśli założymy, że poprzedniki reguł modelu zastosowanego do opisu obiektu w układzie regulacji i poprzedniki reguł regulatora są identyczne – co ma miejsce, gdy do opisu obiektu w układzie regulacji wykorzystujemy model użyty do projektowania regulatora – to wówczas zachodzą równości  $ro = r$ ,  $\tilde{w}_o^i(k) = \tilde{w}^i(k)$ ,  $i = 1, \dots, r$  oraz  $\tilde{w}^i(k)\tilde{w}^j(k) = \tilde{w}^j(k)\tilde{w}^i(k)$ , dla każdych wartości  $i, j = 1, 2, \dots, r$ . Wykorzystując tę ostatnią zależność można zmniejszyć liczbę macierzy  $\mathbf{A}_{ij}$  występujących w (2.55), postępując

analogicznie jak w rozdziale poprzednim. Mianowicie, równanie dynamiki (2.54) możemy wówczas przedstawić w postaci

$$x(k+1) = \left[ \sum_{i=1}^r \tilde{w}^i(k) \tilde{w}^i(k) \mathbf{A}_{ii} + 2 \sum_{i<j}^r \tilde{w}^i(k) \tilde{w}^j(k) \overline{\mathbf{A}}_{ij} \right] x(k), \quad (2.56)$$

gdzie

$$\overline{\mathbf{A}}_{ij} = \frac{\mathbf{A}_{ij} + \mathbf{A}_{ji}}{2}, \quad i < j, \quad i, j = 1, 2, \dots, r,$$

zaś symbol  $\sum_{i<j}^r$  oznacza sumowanie po wszystkich parach indeksów  $i, j$  takich, że  $i < j$ . W warunkach stabilności (2.55) wystarczy teraz uwzględnić, zamiast macierzy  $\mathbf{A}_{ij}$ ,  $i = 1, 2, \dots, r$ ,  $j = 1, 2, \dots, r$ , jedynie macierze  $\mathbf{A}_{ii}$ ,  $i = 1, \dots, r$  oraz  $\overline{\mathbf{A}}_{ij}$ ,  $i < j$ ,  $i, j = 1, 2, \dots, r$ . Warunki (2.55) można w tym przypadku jeszcze nieco osłabić, stosując do układu dynamicznego (2.56) twierdzenie 2.4.

**Wniosek 2.7.** *Niech maksymalna liczba reguł aktywowanych jednocześnie jest nie większa od  $s$ ,  $1 < s \leq r$ . Wówczas punkt równowagi układu dynamicznego opisanego zależnością (2.56) jest globalnie asymptotycznie stabilny, jeśli istnieją macierze dodatnio określona  $\mathbf{P}$  i dodatnio półokreślona  $\mathbf{Q}$  spełniające warunki*

$$\begin{aligned} \mathbf{A}_{ii}^T \mathbf{P} \mathbf{A}_{ii} - \mathbf{P} + (s-1)\mathbf{Q} &< 0, \quad i = 1, 2, \dots, r, \\ \overline{\mathbf{A}}_{ij}^T \mathbf{P} \overline{\mathbf{A}}_{ij} - \mathbf{P} - \mathbf{Q} &\leq 0, \quad i < j, \quad i, j = 1, 2, \dots, r, \end{aligned}$$

dla wszystkich par  $(i, j)$  z wyjątkiem takich, dla których  $w^i(k)w^j(k) = 0$  dla każdej wartości  $k$ .  $\square$

Przeprowadzona analiza stabilności nie obejmuje bezpośrednio przypadku układu regulacji z regulatorem zdefiniowanym regułami (2.43), których poprzedniki zawierają dodatkowo warunek "  $u(k)$  jest  $B_0^j$ ", a stąd następni zdefiniowane są nieco innymi wzorami, por. (2.39) i (2.43). Analizę stabilności takiego układu regulacji przeprowadzić można w identyczny sposób jak to uczyniliśmy powyżej, uzyskując analogiczne wyniki – jedynie macierze  $\mathbf{A}_{ij}$  i  $\mathbf{E}_{ij}$  będą zdefiniowane nieco innymi wzorami. Można uzyskać te wzory bezpośrednio z już uzyskanych, przeddefiniowując jedynie współczynniki następników regulatora, od postaci

$$\begin{aligned} w^j(k+1) &= c_1^j e(k) + c_2^j e(k-1) + \dots + c_{n_C}^j e(k-n_C+1) + \\ &+ d_1^j u(k) + d_2^j u(k-1) + \dots + d_{m_D}^j u(k-m_D+1), \end{aligned}$$

do postaci

$$u^j(k) = \bar{c}_1^j e(k) + \bar{c}_2^j e(k-1) + \dots + \bar{c}_{\bar{n}_C}^j e(k - \bar{n}_C + 1) + \\ + d_1^j u(k-1) + d_2^j u(k-3) + \dots + d_{m_D}^j u(k - m_D),$$

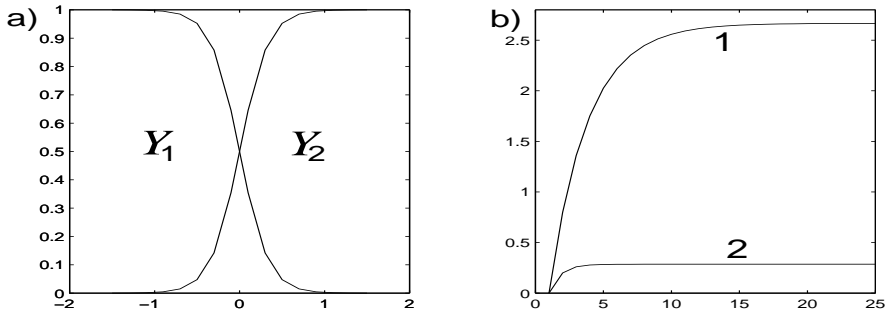
gdzie  $\bar{n}_C = n_C + 1$ ,  $\bar{c}_1^j = 0$ ,  $\bar{c}_p^j = c_{p-1}^j$ ,  $p = 2, \dots, n_C + 1$ . Ponadto, konsekwentnie, należy wówczas wstawić do wszystkich wzorów  $w^j(k-1)$  zamiast  $w^j(k)$ ,  $j = 1, \dots, r$ .

### Przykład 2.6.

Rozważmy model obiektu regulacji opisany następującymi regułami rozmytymi postaci (2.44)

$$R_{obR}^1 : \text{JEŚLI } y(k) \text{ jest } Y_1 \text{ TO } y^1(k+1) = 0.7y(k) + 0.8u(k), \quad (2.57)$$

$$R_{obR}^2 : \text{JEŚLI } y(k) \text{ jest } Y_2 \text{ TO } y^2(k+1) = 0.3y(k) + 0.2u(k). \quad (2.58)$$

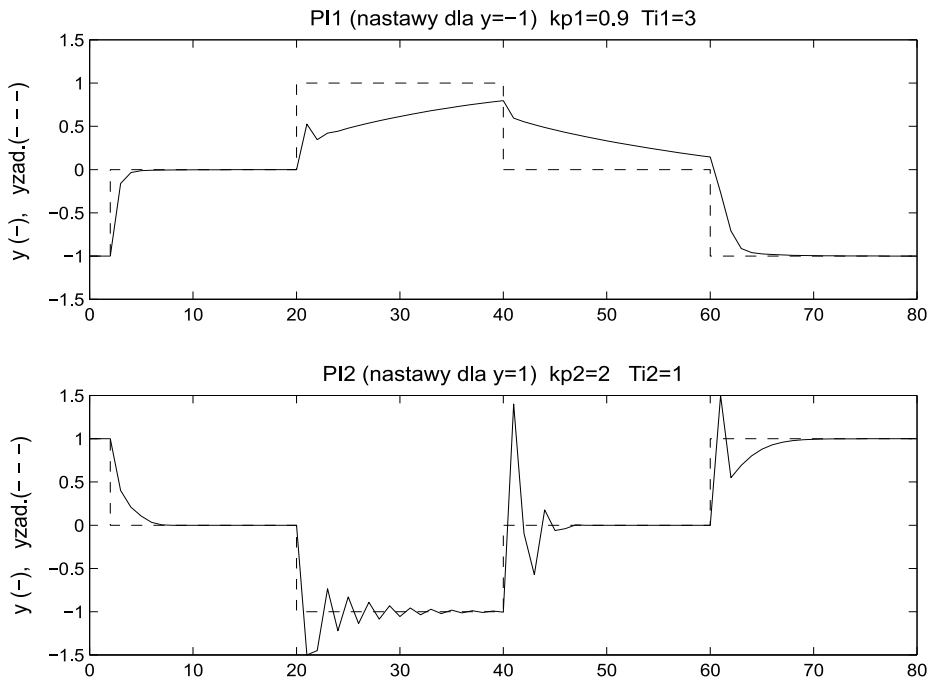


Rysunek 2.23. a) sigmoidalne funkcje przynależności, b) odpowiedzi skokowe obiektów lokalnych, w przykładzie 2.6

Na rysunku 2.23 a) przedstawiono zbiory rozmyte  $Y_1$  i  $Y_2$ , zaś na tym samym rysunku b) odpowiedzi skokowe obiektów lokalnych występujących w następnikach reguł (2.57) i (2.58). Porównanie tych odpowiedzi pokazuje, że rozważany obiekt mimo prostej struktury jest silnie nieliniowy, przy czym nieliniowa jest zarówno statyka (wzmocnienie), jak i dynamika (stała czasowa). Obiekt z drugiej reguły jest trudniejszy do sterowania, ze względu na małą w stosunku do okresu próbkowania stałą czasową.

Dla modeli obiektu z następników reguł (2.57) i (2.58) zaprojektowano regulatory dyskretne przyrostowe PI w wersji z całkowaniem metodą trapezów, tzn. postaci

$$u(k) = u(k-1) + k_p \left(1 + \frac{T_p}{2T_i}\right) e(k) + k_p \left(-1 + \frac{T_p}{2T_i}\right) e(k-1),$$

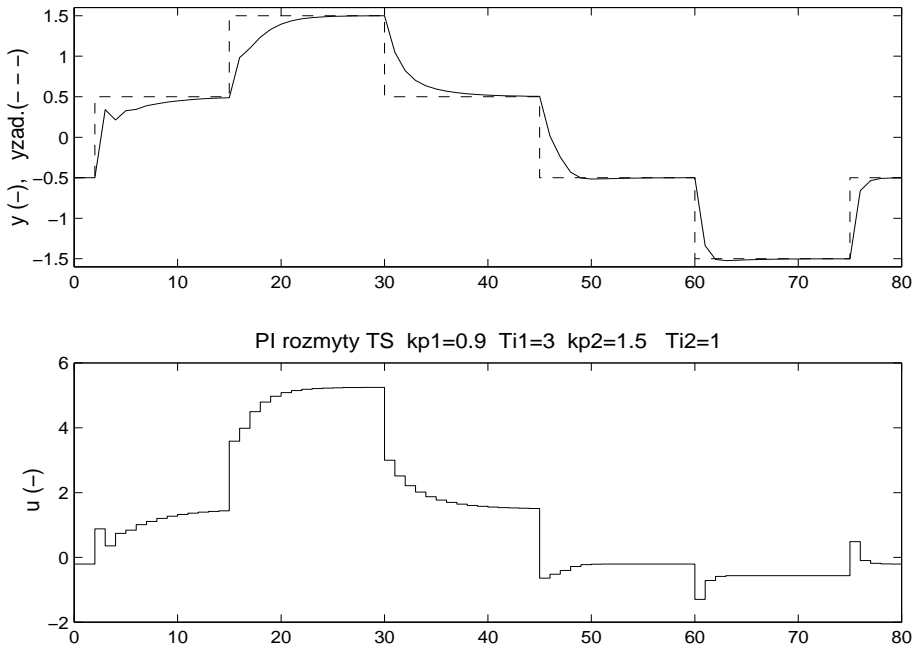


Rysunek 2.24. Przebiegi wielkości regulowanej w układach regulacji z nieliniowym obiektem rozmytym i regulatorem PI zaprojektowanym dla punktu  $y = -1$  (rysunek górny) i dla punktu  $y = +1$  (rysunek dolny)

gdzie  $k_p$  jest wzmocnieniem, a  $T_i$  stałą czasową całkowania. Wymienione nastawy dobrano dla poszczególnych obiektów lokalnych metodą korygowanego eksperymentu Zieglera-Nicholsa (zob. np. [3]), uzyskując

$$\begin{aligned} k_p^1 &= 0.9, & T_i^1 &= 3, \\ k_p^2 &= 2.0, & T_i^2 &= 1. \end{aligned}$$

Poszczególne regulatory PI działają bardzo dobrze w otoczeniach swych punktów pracy, ale fatalnie w całym obszarze. Na rysunku 2.24 pokazano przebiegi w układzie z obiektem nieliniowym oraz, kolejno, z pierwszym regulatorem PI zaprojektowanym dla punktu pracy  $y = -1$  i z drugim regulatorem PI zaprojektowanym dla punktu pracy  $y = +1$ . Pierwszy regulator jest zdecydowanie za wolny w zakresie dodatnich wartości  $y$ . Drugi jest natomiast zbyt agresywny, powoduje przebiegi zbyt oscylacyjne nie tylko dla ujemnych wartości wielkości regulowanej  $y$ , ale już w otoczeniu zera. Co więcej, rysunek pokazuje przebieg zbyt agresywny nawet w otoczeniu własnego punktu pracy  $y = +1$ , jeśli następuje on po skoku wartości za-



Rysunek 2.25. Przebiegi wielkości regulowanej i sterowania w układzie z regulatorem PI rozmytym TS

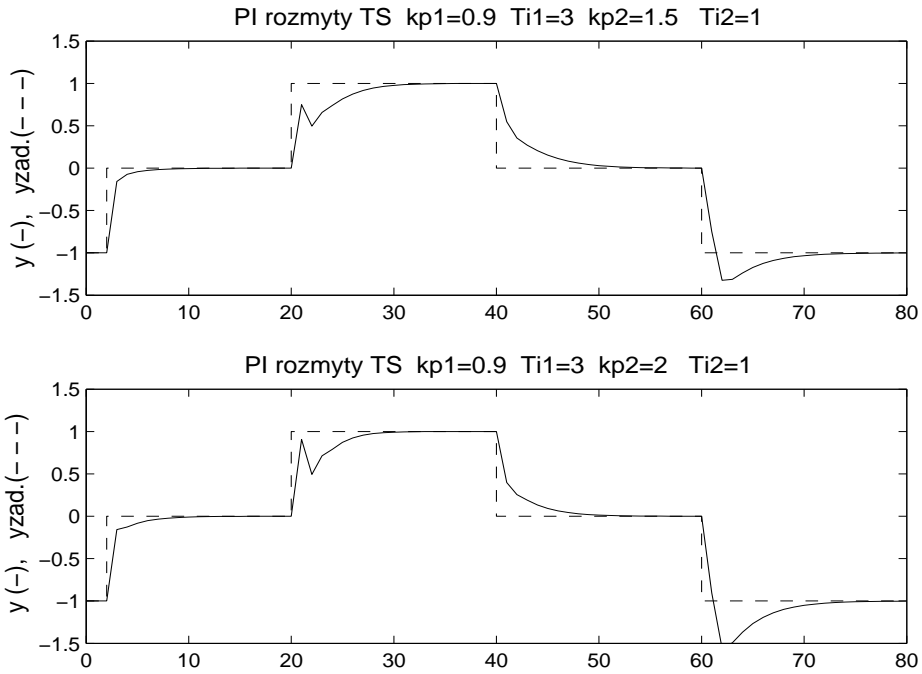
danej od zera. Dlatego też przed przystąpieniem do konstrukcji regulatora rozmytego TS zmniejszono wzmocnienie regulatora drugiego do wartości  $k_p^2 = 1.5$ .

Przyjmując poprzedniki takie jak w modelu obiektu oraz następniki w postaci omówionych regulatorów lokalnych PI dostajemy stąd następujące reguły regulatora rozmytego TS

$$R_{reg}^1: \text{JEŚLI } y(k) \text{ jest } Y_1 \text{ TO } u^1(k) = u(k-1) + 1.05e(k) - 0.75e(k-1), \quad (2.59)$$

$$R_{reg}^2: \text{JEŚLI } y(k) \text{ jest } Y_2 \text{ TO } u^2(k) = u(k-1) + 2.25e(k) - 0.75e(k-1). \quad (2.60)$$

Na rysunku 2.25 przedstawiono przebiegi zmiennej regulowanej i sterowania w układzie regulacji z zaprojektowanym regulatorem rozmytym TS, w dość szerokim zakresie zmian wartości zadanej, zaś na rysunku 2.26 przebiegi zmiennej regulowanej w nieco węższym, krytycznym zakresie zmian między  $-1$  a  $1$ , dla przypadku  $k_p^2 = 1.5$  (rysunek górny) i dla  $k_p^2 = 2$  (rysunek dolny), czyli przebieg z regulatorem rozmytym bez dokonanego uprzed-



Rysunek 2.26. Przebiegi wielkości regulowanej w układzie z regulatorem PI rozmytym: z  $k_p^2 = 1.5$  (rysunek górny) i z  $k_p^2 = 2$  (rysunek dolny)

nio zmniejszenia wzmocnienia  $k_p^2$ . Jak można było oczekiwać, zwiększenie współczynnika wzmocnienia  $k_p^2$  przyspiesza (niewiele) działanie układu regulacji w zakresie dodatnich wartości wyjścia. Niestety, jednocześnie różnie przeregulowanie przy zmianach wartości zadanej w kierunku wartości ujemnych, dobrze widoczne dla skoku od zera do  $-1$ . Stąd też zasadność dokonanej uprzednio korekty wzmocnienia  $k_p^2$ . W rozdziale 3 pokażemy, że występujący konflikt można zredukować poprzez zastosowanie nieliniowego regulatora predykcyjnego, uzyskując przyspieszenie działania dla dodatnich wartości wyjścia przy jednoczesnej redukcji przeregulowania.

Dla analizy stabilności uzyskanego nieliniowego rozmytego układu regulacji (z  $k_p^2 = 1.5$ ) wykorzystamy wynikający z twierdzenia 2.1 wniosek 2.7. Trzeba wyznaczyć w tym celu macierze  $\mathbf{A}_{ij}$  (2.51). Rozszerzony wektor stanu w rozważanym przykładzie to  $x(k) = [y(k) \ y(k-1) \ u(k-1)]^T$ , z postaci następników reguł obiektu i regulatora wynikają wartości  $n = 2$ ,

$m = 1$ . Stąd macierze  $A_{ij}$  są o wymiarze 3, postaci ogólnej

$$\mathbf{A}_{ij} = \begin{bmatrix} a_1^i - b_0^i c_1^j & a_2^i - b_0^i c_2^j & b_1^i + b_0^i d_1^j \\ 1 & 0 & 0 \\ -c_1^j & -c_2^j & d_1^j \end{bmatrix}, \quad i, j = 1, 2.$$

Przyjmując, dla teoretycznego sprawdzenia warunków stabilności, model obiektu (2.57), (2.58), czyli takie same poprzedniki reguł w obiekcie i regulatorze, wystarczy wziąć pod uwagę macierze  $\mathbf{A}_{11}$ ,  $\mathbf{A}_{22}$  i  $\overline{\mathbf{A}}_{12}$ . Są one następującej postaci

$$\mathbf{A}_{11} = \begin{bmatrix} -0.14 & 0.6 & 0.8 \\ 1 & 0 & 0 \\ -1.05 & 0.75 & 1 \end{bmatrix},$$

$$\mathbf{A}_{22} = \begin{bmatrix} -0.15 & 0.15 & 0.2 \\ 1 & 0 & 0 \\ -2.25 & 0.75 & 1 \end{bmatrix},$$

$$\overline{\mathbf{A}}_{12} = \begin{bmatrix} -0.5050 & 0.3750 & 0.5 \\ 1 & 0 & 0 \\ -1.65 & 0.75 & 1 \end{bmatrix}.$$

Korzystając z *LMI Toolbox* do pakietu MATLAB sprawdzono, że układ nierówności macierzowych

$$\begin{aligned} (\mathbf{A}_{11})^T \mathbf{P} \mathbf{A}_{11} - \mathbf{P} &< 0, \\ (\mathbf{A}_{22})^T \mathbf{P} \mathbf{A}_{22} - \mathbf{P} &< 0, \\ (\overline{\mathbf{A}}_{12})^T \mathbf{P} \overline{\mathbf{A}}_{12} - \mathbf{P} &< 0, \\ -\mathbf{P} &< 0, \end{aligned}$$

posiada rozwiązanie postaci

$$\mathbf{P} = 10^3 \begin{bmatrix} 5.4140 & -2.8931 & -3.0451 \\ -2.8931 & 2.2731 & 2.1858 \\ -3.0451 & 2.1858 & 2.5867 \end{bmatrix}.$$

Stąd na mocy wniosku 2.7 (w wersji z  $\mathbf{Q} = \mathbf{0}$ ) spełnione są warunki dostateczne stabilności rozważanego nieliniowego układu regulacji. Przeprowadzona analiza dotyczy stabilności układu nominalnego, pozostaje jeszcze zbadanie odporności zaprojektowanego układu regulacji. Sprawdzono, że układ pracuje prawidłowo przy dość znacznych odchyleniach parametrów.  $\square$

Jak już wcześniej wspomniano, rozważona w tym rozdziale postać dyskretnego rozmytego regulatora TS obejmuje również przypadki regulatorów predykcyjnych DMC i GPC w wersji analitycznej. W rozdziale 3 pokażemy m.in. projektowanie nieliniowych regulatorów predykcyjnych dla obiektu z przedstawionego wyżej przykładu, w tym regulatora GPC.

### 2.3. Ciągłe rozmyte algorytmy regulacji typu TS

W dotychczasowych rozważaniach przyjmowaliśmy dyskretny opis obiektu i stąd, konsekwentnie, regulatora. W praktyce, dyskretny opis obiektu dostosowany jest przede wszystkim do sytuacji, kiedy okres próbkowania w układzie regulacji nie jest wyraźnie mniejszy od dominujących stałych czasowych obiektu. Dla bardzo małych okresów próbkowania dostajemy np. dyskretne odpowiedzi skokowe o bardzo dużej liczbie istotnych elementów czy modele ARX o wysokim rzędzie, co niepotrzebnie utrudnia projektowanie układów regulacji. Z drugiej strony, w rezultacie rozwoju techniki mikrokomputerowej coraz częściej dostępne są sterowniki posiadające duże moce obliczeniowe, umożliwiające stosowanie małych okresów próbkowania nawet przy bardziej złożonych algorytmach regulacji. Stąd w układach regulacji bezpośredniej coraz powszechniej stosuje się małe okresy próbkowania, tym bardziej, że podejście takie umożliwia bezpośrednie przechodzenie z techniki analogowej na cyfrową – nie trzeba zmieniać nierzadko żmudnie w przeszłości dostrajanych nastaw regulatorów analogowych.

W sytuacji, kiedy okres próbkowania w pętli regulacji jest dostatecznie mały w stosunku do dynamiki obiektu, podejściem w pełni uprawnionym jest projektowanie ciągłego układu regulacji, a następnie cyfrowa realizacja regulatora ciągłego jedną z metod emulacji, zob. np. [47, 54]. Dlatego też w podrozdziale niniejszym zajmiemy się zagadnieniami projektowania i analizy ciągłych regulatorów rozmytych TS. Po pierwsze, lokalne algorytmy regulacji takiego regulatora będą definiowane jako ciągłe. Następnie, konstrukcję całego regulatora nieliniowego i warunki stabilności układu regulacji rozważymy też w dziedzinie czasu ciągłego. Nadmienmy ponadto, że symulacyjną weryfikację układu regulacji też można przeprowadzić formułując jego opis w dziedzinie ciągłej (np. korzystając z programu *Simulink*), lub lepiej w środowisku hybrydowym z ciągłą realizacją obiektu modelowanego układem równań różniczkowych całkowanych z odpowiednio małym krokiem i dyskretną realizacją regulatora.

Struktury ciągłych regulatorów rozmytych typu TS są *analogiczne* do prezentowanych w rozdziale poprzednim struktur dyskretnych. Stąd, przed-

stawiając konstrukcję takich regulatorów oraz warunki stabilności układów regulacji uczynimy to w sposób skrótowy, koncentrując się przede wszystkim na sformułowaniach i cechach różniących oba podejścia. W szczególności, część wstępna rozdziału 2.2 została napisana w sposób ogólny, odnosi się do modelowania obiektu i projektowania algorytmów regulacji rozmytych TS zarówno w wersji dyskretnej, jak i ciągłej.

### Ciągły regulator rozmyty TS ze sprzężeniem od stanu

Rozważmy obiekt dynamiczny o wektorach stanu i sterowań

$$x(t) = [x_1(t) \ x_2(t) \ \cdots \ x_{n_x}(t)]^T,$$

$$u(t) = [u_1(t) \ u_2(t) \ \cdots \ u_{n_u}(t)]^T,$$

opisany modelem rozmytym TS o regułach postaci

$$\begin{aligned} R_{ob}^i : \quad & \text{JEŚLI } x_1(t) \text{ jest } A_1^i \text{ i } x_2(t) \text{ jest } A_2^i \text{ i } \cdots \text{ i } x_{n_x}(t) \text{ jest } A_{n_x}^i \\ & \text{TO } \dot{x}^i(t) = \mathbf{A}_i x(t) + \mathbf{B}_i u(t), \end{aligned} \quad (2.61)$$

gdzie  $A_j^i \in \mathbb{X}_j = \{X_{j1}, \dots, X_{jr_j}\}$  są zbiorami rozmytymi współrzędnych  $x_j$  wektora stanu  $x$  (por. (2.4)),  $j = 1, \dots, n_x$ , zaś  $\mathbf{A}_i$  i  $\mathbf{B}_i$  macierzami stanu i sterowań lokalnych modeli liniowych utworzonych dla poszczególnych obszarów rozmytych dziedziny modelu,  $i = 1, \dots, r$ . Przy siatkowym podziale obszaru zmiennych wejściowych dostajemy system rozmyty o liczbie obszarów (i reguł) równej  $r = r_1 r_2 \cdots r_{n_x} = \prod_{j=1}^{n_x} r_j$ .

Dla danych wartości  $x(t)$  i  $u(t)$  wyjście modelu rozmytego, tj. wartość prawej strony równania stanu całościowego modelu obiektu, wyliczane jest zgodnie z ogólnym wzorem (2.5), tj.

$$\dot{x}(t) = \frac{\sum_{i=1}^r w^i(t) [\mathbf{A}_i x(t) + \mathbf{B}_i u(t)]}{\sum_{l=1}^r w^l(t)}, \quad (2.62)$$

gdzie  $w^i(t)$  są poziomami aktywacji poszczególnych reguł (2.61),

$$w^i(t) = \prod_{j=1}^{n_x} \mu_{A_j^i}(x_j(t)).$$

Przyjęliśmy powyżej naturalne założenie, że zawsze  $\sum_{i=1}^r w^i(k) > 0$ , tj. dla każdej wartości zmiennych stanu z rozważanego zakresu ich zmienności model jest określony – co najmniej jedna reguła jest aktywowana.

Model obiektu niesterowanego (przy  $u(t) \equiv 0$ ) można zapisać w postaci

$$\dot{x}(t) = \sum_{i=1}^r \tilde{w}^i(t) \mathbf{A}_i x(t), \quad (2.63)$$

gdzie przez  $\tilde{w}^i(t)$  oznaczono unormowane poziomy aktywacji reguł. Dla takiego modelu dynamicznego, bezpośrednio z zastosowania twierdzenia Lapunowa wynika następujący warunek dostateczny stabilności asymptotycznej:

**Twierdzenie 2.8.** *Punkt równowagi systemu dynamicznego (2.63) jest globalnie asymptotycznie stabilny, jeśli istnieje macierz symetryczna dodatnio określona  $\mathbf{P}$  taka, że dla macierzy stanu  $\mathbf{A}_i$  każdego z modeli lokalnych spełnione jest równanie*

$$\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i < \mathbf{0}, \quad i = 1, 2, \dots, r. \quad (2.64)$$

**Dowód.** Przebiega podobnie jak w twierdzeniu 2.1. Tworzymy funkcję skalarną postaci

$$V(x(t)) = x^T(t) \mathbf{P} x(t), \quad (2.65)$$

gdzie  $\mathbf{P}$  jest macierzą symetryczną dodatnio określoną. Funkcja ta wprost z definicji spełnia wszystkie warunki funkcji Lapunowa z wyjątkiem ujemności pochodnej wzdłuż trajektorii obiektu (2.63), co pokażemy poniżej. Mamy bowiem

$$\begin{aligned} \dot{V}(x(t)) &= \dot{x}^T(t) \mathbf{P} x(t) + x^T(t) \mathbf{P} \dot{x}(t) \\ &= x^T(t) \left[ \sum_{i=1}^r \tilde{w}^i(t) \mathbf{A}_i^T \mathbf{P} + \sum_{i=1}^r \tilde{w}^i(t) \mathbf{P} \mathbf{A}_i \right] x(t) \\ &= \sum_{i=1}^r \tilde{w}^i(t) x^T(t) \left[ \mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i \right] x(t). \end{aligned}$$

Ponieważ

$$\begin{aligned} \tilde{w}^i(t) &\geq 0 \quad \text{dla każdego } i = 1, 2, \dots, r, \\ \sum_{i=1}^r \tilde{w}^i(t) &> 0, \end{aligned}$$

to  $\dot{V}(x(t)) < 0$  dla  $x(t) \neq 0$ , jeśli tylko spełniony jest warunek (2.64). Wówczas funkcja  $V(x(t))$  jest funkcją Lapunowa dla nieliniowego układu dynamicznego (2.63).  $\square$

W pracy [155] pokazano, że warunkiem koniecznym istnienia macierzy  $\mathbf{P}$  spełniającej (2.64) jest, aby każda z sum  $\mathbf{A}_i + \mathbf{A}_j$ ,  $i, j = 1, 2, \dots, r$  (a w istocie każda suma macierzy spośród  $\mathbf{A}_i$ ,  $i = 1, 2, \dots, r$ ) była macierzą układu asymptotycznie stabilnego, tzn. miała ujemne części rzeczywiste wartości własnych. Fakt ten wynika bezpośrednio z dodania do siebie nierówności (2.64) dla  $i = i$  oraz  $i = j$ . Jeśli bowiem  $\mathbf{P}$  jest dodatnio określona

i zachodzi  $(\mathbf{A}_i + \mathbf{A}_j)^T \mathbf{P} + \mathbf{P}(\mathbf{A}_i + \mathbf{A}_j) < \mathbf{0}$ , to  $\mathbf{A}_i + \mathbf{A}_j$  jest macierzą asymptotycznie stabilną. Tak więc wykrycie, że którakolwiek z sum  $\mathbf{A}_i + \mathbf{A}_j$  taką macierzą nie jest, przesądza o nieistnieniu macierzy  $\mathbf{P}$  spełniającej warunki twierdzenia 2.8.

Istnienie jednej wspólnej dodatnio określonej macierzy  $\mathbf{P}$  spełniającej nierówności (2.64) jest warunkiem dostatecznym stabilności, warunki konieczne są jak dotąd nieznanne. Przy tym, warunki dostateczne sformułowane w twierdzeniu 2.8 nie biorą pod uwagę wartości poziomów aktywacji poszczególnych reguł (czyli kształtów funkcji przynależności). Dlatego też próbowano warunki te osłabić, zob. [22, 20, 8], stosując do badania stabilności układu dynamicznego (2.63) nieco inną funkcję Lapunowa, a mianowicie postaci

$$V(x(t)) = x^T(t) \mathbf{P}(x(t)) x(t) = x^T(t) \sum_{i=1}^r \tilde{w}^i(t) \mathbf{P}_i x(t), \quad (2.66)$$

gdzie  $\mathbf{P}_i$  są macierzami symetrycznymi dodatnio określonymi. Licząc pochodną tej funkcji wzdłuż trajektorii układu (2.63) dostajemy

$$\begin{aligned} \dot{V}(x(t)) &= \dot{x}^T(t) \mathbf{P}(x(t)) x(t) + x^T(t) \mathbf{P}(x(t)) \dot{x}(t) + x^T(t) \dot{\mathbf{P}}(x(t)) x(t) \\ &= x^T(t) \left[ \sum_{i=1}^r \tilde{w}^i(t) \mathbf{A}_i^T \sum_{j=1}^r \tilde{w}^j(t) \mathbf{P}_j + \sum_{i=1}^r \tilde{w}^i(t) \sum_{j=1}^r \tilde{w}^j(t) \mathbf{P}_j \mathbf{A}_i \right] x(t) + \\ &\quad + x^T(t) \sum_{i=1}^r \left( \frac{d}{dt} \tilde{w}^i(x(t)) \mathbf{P}_i \right) x(t) \\ &= x^T(t) \left[ \sum_{i=1}^r \sum_{j=1}^r \tilde{w}^j(t) \tilde{w}^i(t) \mathbf{A}_i^T \mathbf{P}_j + \sum_{i=1}^r \sum_{j=1}^r \tilde{w}^i(t) \tilde{w}^j(t) \mathbf{P}_j \mathbf{A}_i \right] x(t) + \\ &\quad + x^T(t) \sum_{i=1}^r \left( \left[ \frac{\partial \tilde{w}^i(x(t))}{\partial x_j(t)} \right] \dot{x}(t) \mathbf{P}_i \right) x(t). \end{aligned}$$

Pierwsze dwa składniki stosunkowo łatwo oszacować, będą one mianowicie ujemne dla  $x(t) \neq 0$ , jeśli spełnione są nierówności

$$\begin{aligned} \mathbf{A}_i^T \mathbf{P}_i + \mathbf{P}_i \mathbf{A}_i &< \mathbf{0}, \quad i = 1, 2, \dots, r, \\ \frac{\mathbf{A}_i^T \mathbf{P}_j + \mathbf{P}_j \mathbf{A}_i + \mathbf{A}_j^T \mathbf{P}_i + \mathbf{P}_i \mathbf{A}_j}{2} &< \mathbf{0}, \quad i \neq j, \quad i, j = 1, 2, \dots, r, \end{aligned}$$

dla wszystkich par indeksów  $i, j$  z wyjątkiem takich, dla których zawsze zachodzi  $\tilde{w}^i(t) \tilde{w}^j(t) = 0$ . Niestety, trudno jest uzyskać konstruktywne warunki ujemności ostatniego składnika, co prowadziłoby do spełnienia przez

funkcję (2.66) warunków funkcji Lapunowa, a więc dowodziłoby stabilności. Przyczyną jest trudność w oszacowaniu pochodnej wektora stanu  $\dot{x}(t)$ . Jak dotąd uzyskane warunki są albo mało konstruktywne (założenie a priori odpowiedniego ograniczenia normy gradientu pochodnej wektora stanu [20]), albo sformułowane w złożonej postaci, o niejasnej na obecnym etapie wiedzy przydatności [8]. Stąd nie będziemy ich tu przytaczać odsyłając zainteresowanego czytelnika do cytowanych prac.

Natomiast w [21] podano zupełnie inne sformułowanie warunków dostatecznych stabilności układu dynamicznego (2.63).

**Twierdzenie 2.9.** *Punkt równowagi dynamicznego systemu rozmytego (2.63) jest globalnie asymptotycznie stabilny, jeśli*

$$\lambda_{\max}(\mathbf{A}_i + \mathbf{A}_i^T) < 0, \quad i = 1, 2, \dots, r, \quad (2.67)$$

gdzie  $\lambda_{\max}(\mathbf{A}_i + \mathbf{A}_i^T)$  oznacza maksymalną wartość własną macierzy symetrycznej  $\mathbf{A}_i + \mathbf{A}_i^T$ .  $\square$

Twierdzenie powyższe przytaczamy bez dowodu (zob. [21]), gdyż podany w nim warunek stabilności (2.67), chociaż innej natury niż warunek z twierdzenia 2.8 i sformułowany w terminach macierzy lokalnych, jest trudny do interpretacji i okazuje się na ogół bardzo konserwatywny, a stąd mało użyteczny.

Dla każdego z lokalnych liniowych modeli dynamicznych występujących w następnikach reguł (2.61), tj. dla każdego z obszarów modelu rozmytego TS, projektujemy liniowy regulator ze sprzężeniem od zmiennych stanu, metodą klasyczną. Uzyskujemy w ten sposób reguły regulatora rozmytego TS w postaci

$$\begin{aligned} R_{reg}^j : \quad & \text{JEŚLI } x_1(t) \text{ jest } A_1^j \text{ i } x_2(t) \text{ jest } A_2^j \text{ i } \dots \text{ i } x_{n_x}(t) \text{ jest } A_{n_x}^j \\ & \text{TO } u^j(t) = -\mathbf{F}_j x(t), \end{aligned} \quad (2.68)$$

gdzie  $\mathbf{F}_i$  są macierzami współczynników sprzężeń od stanu,  $j = 1, 2, \dots, r$ . Nieliniowy regulator rozmyty TS jest stąd opisany zależnością

$$u(t) = -\sum_{j=1}^r \tilde{w}^j(t) \mathbf{F}_j x(t), \quad (2.69)$$

gdzie unormowane poziomy aktywacji reguł są, oczywiście, takie same jak w modelu obiektu (2.61) użyty do projektowania regulatora.

Analogicznie jak zakładano w rozdziale 2.2.1, do opisu obiektu w zamkniętym układzie regulacji można w ogólności użyć modelu rozmytego TS innego, niż wykorzystany do projektowania regulatora rozmytego, tzn.

o innej liczbie czy postaci reguł. Liczbę reguł tego modelu oznaczamy przez  $ro$ , a poziomy ich aktywacji przez  $w_o^i(k)$ ,  $i = 1, \dots, ro$ . Wstawiając teraz opis regulatora (2.69) do rozmytego modelu obiektu postaci (2.62) – ale z  $ro$  regułami o poziomach aktywacji  $w_o^i(k)$ , uzyskujemy formułę opisującą zamknięty układ regulacji

$$\begin{aligned} \dot{x}(t) &= \frac{\sum_{i=1}^{ro} \sum_{j=1}^r w_o^i(t) w^j(t) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) x(t)}{\sum_{l=1}^{ro} \sum_{p=1}^r w_o^l(t) w^p(t)} \\ &= \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(t) \tilde{w}^j(t) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) x(t). \end{aligned} \quad (2.70)$$

Oczywiście, jeśli model użyty do opisu obiektu jest inny niż model (2.61), (2.62) wykorzystany do projektowania regulatora, to jego macierze  $\mathbf{A}_i$  i  $\mathbf{B}_i$  są też w ogólności inne (choć nie wprowadzamy innych oznaczeń, bo nie prowadzi to do nieporozumień – w równaniu układu regulacji (2.70) występują tylko te macierze). Do badania stabilności układu dynamicznego opisanego wzorem (2.70) można bezpośrednio wykorzystać twierdzenie 2.8 lub twierdzenie 2.9, jedynie zamiast macierzy  $\mathbf{A}_i$ ,  $i = 1, \dots, r$  trzeba brać pod uwagę macierze  $\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j$ ,  $i = 1, \dots, ro$ ,  $j = 1, \dots, r$ . Jeśli natomiast model wykorzystany do opisu obiektu w układzie regulacji jest równocześnie modelem wykorzystanym do projektowania regulatora, tzn. identyczne są poprzedniki tych modeli (stąd  $ro = r$  i  $w_o^i(k) = w^i(k)$ ), to wykorzystując ten fakt zależność (2.70) można uprościć, postępując analogicznie, jak to uczyniliśmy w rozdziale 2.2.1 ze wzorem (2.25) dla układów z czasem dyskretnym. Wykorzystując bowiem równość  $w^i(k) w^j(k) = w^j(k) w^i(k)$ ,  $i, j = 1, 2, \dots, r$  zależność (2.70) można zapisać w postaci

$$\dot{x}(t) = \sum_{i=1}^r \tilde{w}^i(t) \tilde{w}^i(t) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i) x(t) + 2 \sum_{i < j}^r \tilde{w}^i(t) \tilde{w}^j(t) \mathbf{D}_{ij} x(t), \quad (2.71)$$

gdzie

$$\mathbf{D}_{ij} = \frac{1}{2} [(\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) + (\mathbf{A}_j - \mathbf{B}_j \mathbf{F}_i)], \quad i < j, \quad i, j = 1, 2, \dots, r. \quad (2.72)$$

Warunki dostateczne stabilności można teraz sformułować w następującej postaci, zob. [137].

**Twierdzenie 2.10.** *Niech maksymalna liczba reguł aktywowanych jednocześnie w układzie regulacji (2.71) opisanym regułami modelu obiektu (2.61) i regulatora (2.68) jest nie większa od  $s$ ,  $1 < s \leq r$ . Punkt równowagi tego*

układu jest globalnie asymptotycznie stabilny, jeśli istnieją macierze dodatnio określona  $\mathbf{P}$  i dodatnio półokreślona  $\mathbf{Q}$  spełniające warunki

$$\begin{aligned} (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i)^T \mathbf{P} + \mathbf{P}(\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i) + (s-1)\mathbf{Q} &< 0, \quad i = 1, 2, \dots, r, \\ \mathbf{D}_{ij}^T \mathbf{P} + \mathbf{P} \mathbf{D}_{ij} - \mathbf{Q} &< 0, \quad i < j, \quad i, j = 1, 2, \dots, r, \end{aligned}$$

z wyjątkiem par  $(i, j)$  takich, dla których zawsze  $w^i(t)w^j(t) = 0$ .  $\square$

W szczególności, dla  $\mathbf{Q} = \mathbf{0}$  uzyskujemy nieco ostrzejsze sformułowanie warunków stabilności, które może być zastosowane do wstępnego testowania. W pracy [154] uzyskano dalsze (wydaje się, że bardzo niewielkie) osłabienie założeń twierdzenia 2.10, posługując się tą samą funkcją Lapunowa  $V(x(t)) = x^T(t)\mathbf{P}x(t)$ . Uzyskane warunki typu LMI (liniowych nierówności macierzowych) są jednakże znacznie bardziej złożone.

Można również próbować stosować do przypadku zamkniętego układu regulacji twierdzenie 2.9, uzyskujemy wówczas

**Wniosek 2.11.** *Punkt równowagi układu dynamicznego (2.71) jest globalnie asymptotycznie stabilny, jeśli*

$$\begin{aligned} \lambda_{\max}([\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i] + [\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i]^T) &< 0, \quad i = 1, 2, \dots, r, \\ \lambda_{\max}(\mathbf{D}_{ij} + \mathbf{D}_{ij}^T) &< 0, \quad i < j, \quad i, j = 1, 2, \dots, r, \end{aligned}$$

z wyjątkiem par  $(i, j)$  takich, dla których zawsze  $w^i(t)w^j(t) = 0$ .  $\square$

W przypadku konieczności rekonstrukcji wektora stanu można zastosować obserwator stanu jako system rozmyty TS o strukturze reguł identycznej jak reguły modelu i regulatora oraz następnikach funkcyjnych będących klasycznymi formułami liniowych obserwatorów stanu, zob. [137].

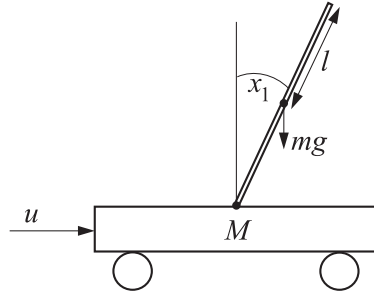
### Przykład 2.7.

Rozważmy zadanie stabilizacji odwróconego wahadła zamocowanego na ruchomym wózku, przedstawionego na rysunku 2.27. Równania opisujące dynamikę obiektu są następujące (zob. np. [111, 155])

$$\dot{x}_1 = x_2, \quad (2.73a)$$

$$\dot{x}_2 = \frac{g \sin x_1 - aml(x_2)^2 \sin(2x_1)/2 - a \cos(x_1)u}{4l/3 - aml \cos^2 x_1}, \quad (2.73b)$$

gdzie przez  $x_1$  oznaczono kąt między położeniem aktualnym a pionowym wahadła (pion  $= 0$  [rad]), stąd  $\dot{x}_1 = x_2$  oznacza prędkość kątową.  $M$  i  $m$  oznaczają masy wózka i wahadła,  $2l$  jest jego długością, zaś  $a = 1/(M+m)$ . Sterowaniem  $u$  jest siła przyłożona do wózka, przez  $g = 9.81m/sec^2$  oznaczono przyspieszenie ziemskie. Przyjmijmy następujące wartości liczbowe:



Rysunek 2.27. Schemat obiektu złożonego z odwróconego wahadła zamocowanego na ruchomym wózku ( $x_1$  – kąt między wahadłem a pionem)

$m = 2 [kg]$ ,  $M = 8 [kg]$ ,  $2l = 1 [m]$  ( jak w [160, 155], gdzie również rozważano sterowanie przedstawionym obiektem).

Wiadomo, że stabilizacja pionowego położenia wahadła regulatorem liniowym działa prawidłowo w ograniczonym zakresie odchyłeń kątowych. Zaprojektujemy prosty regulator rozmyty TS składający się jedynie z dwóch reguł, działający skuteczniej i w większym zakresie odchyłeń. Pierwszym obszarem pracy będzie obszar  $X_1$  wokół położenia zerowego zdefiniowany dwustronną sigmoidalną funkcją przynależności postaci

$$\text{sigmf1}(x) = \frac{1}{1 + \exp[-15(x_1 - \pi/8)]} - \frac{1}{1 + \exp[-15(x_1 + \pi/8)]}, \quad (2.74)$$

zaś drugim obszar  $X_2$  wokół położeń  $x_1 = \pm\pi/4$  określony funkcją przynależności

$$\text{sigmf2}(x) = 1 - \text{sigmf1}(x). \quad (2.75)$$

Dla zaprojektowania regulatora dokonamy aproksymacji nieliniowego opisu (2.73a), (2.73b) modelem rozmytym TS z regułami

$$R_{ob}^1: \text{ JEŚLI } x_1(t) \text{ jest } X_1 \text{ TO } \dot{x}^1(t) = \mathbf{A}_1 x(t) + \mathbf{B}_1 u(t), \quad (2.76a)$$

$$R_{ob}^2: \text{ JEŚLI } x_1(t) \text{ jest } X_2 \text{ TO } \dot{x}^2(t) = \mathbf{A}_2 x(t) + \mathbf{B}_2 u(t), \quad (2.76b)$$

gdzie macierze  $\mathbf{A}_i$  i  $\mathbf{B}_i$  wyznaczmy poniżej.

Pierwszy z modeli liniowych wyznaczamy w punkcie równowagi  $[0 \ 0]$ . Dokonując standardowej linearyzacji dostajemy

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ \frac{g}{4l/3 - aml} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 17.3118 & 0 \end{bmatrix},$$

$$\mathbf{B}_1 = \begin{bmatrix} 0 \\ \frac{-a}{4l/3 - aml} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.1765 \end{bmatrix}.$$

Natomiast przeprowadzając standardową, opartą na rozwinięciu w szereg Taylora, linearyzację oryginalnego nieliniowego modelu (2.73a), (2.73b) w *niezerowym* punkcie  $[\pi/4 \ 0]$  uzyskalibyśmy model *afiniczny*, a nie liniowy, którego wymagamy dla skonstruowania macierzy lokalnych sprzężeń od stanu. Dlatego też wykorzystamy nieco inny sposób linearyzacji, analogicznie jak w [155]. Wykorzystamy przy tym ogólną strukturę naszego modelu nieliniowego, a mianowicie

$$\dot{x}(t) = \mathbf{f}(x(t)) + \mathbf{G}(x(t))u(t).$$

Wymagamy, aby model liniowy  $\dot{x}(t) = \mathbf{A}_2x(t) + \mathbf{B}_2u(t)$  aproksymował w otoczeniu punktu  $x_0 = [\pi/4 \ 0]$  opis nieliniowy w taki sposób, że

$$\mathbf{f}(x) + \mathbf{G}(x)u \approx \mathbf{A}_2x + \mathbf{B}_2u,$$

oraz

$$\mathbf{f}(x_0) + \mathbf{G}(x_0)u = \mathbf{A}_2x_0 + \mathbf{B}_2u, \quad u \in \mathbb{R}.$$

Z warunków tych wynika bezpośrednio

$$\mathbf{B}_2 = \mathbf{G}(x_0).$$

Pozostaje więc do spełnienia

$$\mathbf{f}(x) \approx \mathbf{A}_2x \quad \text{i} \quad \mathbf{f}(x_0) = \mathbf{A}_2x_0.$$

Oznaczmy przez  $\mathbf{a}_{2j}$  wiersze macierzy  $\mathbf{A}_2$ ,  $j = 1, 2$ . Jak pokazano w [155], dla spełnienia wyżej postawionych warunków wiersze  $\mathbf{a}_{2j}$  należy wyznaczać z rozwiązania zadania optymalizacji

$$\min_{\mathbf{a}_{2j}} \{0.5 \|\nabla f_j(x_0) - \mathbf{a}_{2j}\|_2^2\}$$

$$\text{przy ogr.: } \mathbf{a}_{2j}^T x_0 = f_j(x_0),$$

gdzie  $\mathbf{f} = [f_1 \ f_2]^T$ . Rozwiązanie analityczne sformułowanego zadania optymalizacji kwadratowej prowadzi do wzoru

$$\mathbf{a}_{2j} = \nabla f_j(x_0) + \frac{f_j(x_0) - x_0^T \nabla f_j(x_0)}{\|x_0\|^2} x_0, \quad x_0 \neq 0.$$

Wykorzystując powyższe rozumowanie dostajemy następujące postacie macierzy  $\mathbf{A}_2$  i  $\mathbf{B}_2$ :

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ \frac{4g \sin(\pi/4)}{\pi[4l/3 - aml \cos^2(\pi/4)]} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 14.3223 & 0 \end{bmatrix},$$

$$\mathbf{B}_2 = \begin{bmatrix} 0 \\ \frac{-a \cos(\pi/4)}{4l/3 - aml \cos^2(\pi/4)} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.1147 \end{bmatrix}.$$

Regulator rozmyty TS będzie określony regułami

$$R_{reg}^1 : \text{JEŚLI } x_1(t) \text{ jest } X_1 \quad \text{TO } u^1(t) = -\mathbf{F}_1 x(t),$$

$$R_{reg}^2 : \text{JEŚLI } x_1(t) \text{ jest } X_2 \quad \text{TO } u^2(t) = -\mathbf{F}_2 x(t),$$

i wnioskowaniem finalnym

$$u(t) = w^1(t)u^1(t) + w^2(t)u^2(t),$$

gdzie  $w^1(t)$  i  $w^2(t)$  to poziomy aktywacji reguł (zauważmy, że z konstrukcji funkcji przynależności (2.74) i (2.75) wynika  $w^1(t) + w^2(t) = 1$ ). Macierze  $\mathbf{F}_1$  i  $\mathbf{F}_2$  wyznaczmy w standardowy sposób zakładając położenie wartości własnych macierzy  $\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i$ ,  $i = 1, 2$ . Wymagając, aby dla obu modeli liniowych wartości te wynosiły  $-2$  i  $-2.5$  dostajemy (stosując komendę *place* z *Control Toolbox* pakietu MATLAB)

$$\mathbf{F}_1 = [-126.4125 \quad -25.4958],$$

$$\mathbf{F}_2 = [-168.4595 \quad -39.2328].$$

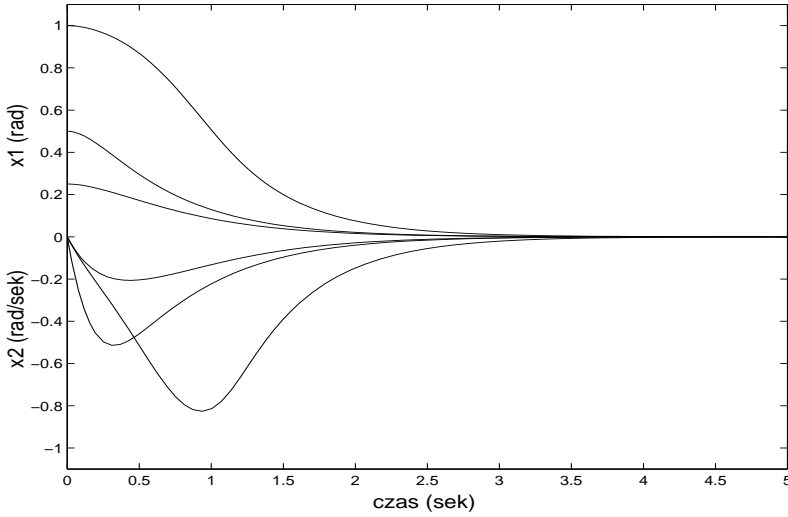
Do symulacji zamkniętego układu regulacji wykorzystano oryginalny nieliniowy model obiektu (2.73a), (2.73b) i zaprojektowany rozmyty regulator TS generujący sterowanie

$$u(t) = -w^1(t)\mathbf{F}_1 x(t) - w^2(t)\mathbf{F}_2 x(t).$$

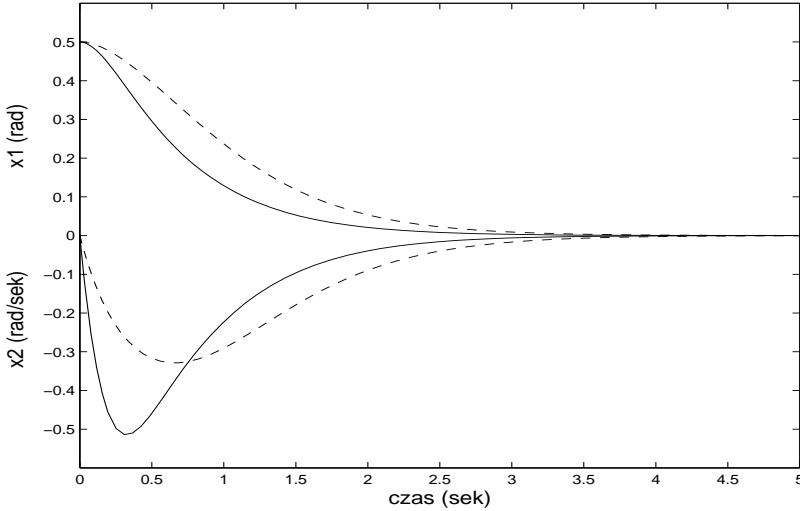
Przykładowe przebiegi zmiennych stanu  $x(t)$  dla stanów początkowych  $x_1(0)$  równych 0.25, 0.5 i 1.0 [rad] oraz  $x_2(0) = 0$  [rad/sek] przedstawiono na rysunku 2.28.

Dla porównania, na rysunku 2.29 pokazano przebiegi z regulatorem rozmytym i regulatorem liniowym zaprojektowanym dla pierwszego obszaru ( $u(t) = -\mathbf{F}_1 x(t)$ ), dla stanu początkowego  $x_1(0) = 0.5$ ,  $x_2(0) = 0$ . Generalnie, im dalej od położenia pionowego, tym nieliniowy regulator rozmyty działa szybciej. Ponadto, doprowadza on wahadło do pozycji pionowej z szerszego zakresu stanów początkowych. Pracuje jeszcze od  $x_1(0) = 1.0$ , podczas gdy regulator liniowy jest już niestabilny dla  $x_1(0) = 0.85$ .

Dla sprawdzenia jakości modelowania rozmytego dość silnie przeciw nieliniowego obiektu, dokonano również symulacji układu regulacji z obiektem w postaci modelu rozmytego TS danego regułami (2.76a) i (2.76b). Na rysunku 2.30 pokazano linią przerywana przebiegi w takim właśnie układzie regulacji, jednocześnie z przebiegami w układzie regulacji z oryginalnym obiektem nieliniowym (linia ciągła). W zakresie mniejszych odchyień zgodność obu rodzajów przebiegów jest bardzo dobra (dla  $x_1(0) = 0.25$  mamy przebiegi prawie nierozróżnialne).

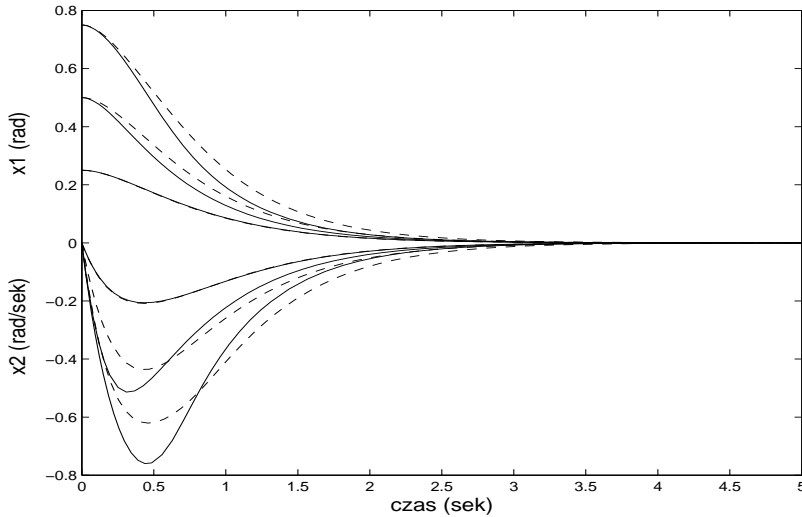


Rysunek 2.28. Przebiegi w układzie z regulatorem rozmytym TS dla różnych warunków początkowych



Rysunek 2.29. Przebiegi w układzie z regulatorem rozmytym TS (linia ciągła) i regulatorem liniowym (linia przerywana)

Analiza powyższa jest o tyle interesująca, że dla układu regulacji z modelami rozmytymi zarówno obiektu, jak i regulatora możemy zastosować kryteria badania stabilności. Zastosowano twierdzenie 2.10 z  $\mathbf{Q} = \mathbf{0}$ , stwierdzając spełnienie warunku dostatecznego stabilności, z macierzą



Rysunek 2.30. Przebiegi w układach regulacji z regulatorem rozmytym TS i obiektem w postaci oryginalnego nieliniowego modelu (linie ciągłe) oraz modelu rozmytego TS użytego do konstrukcji regulatora (linie przerywane)

$$\mathbf{P} = \begin{bmatrix} 1.2529 & 0.1025 \\ 0.1025 & 0.1461 \end{bmatrix}.$$

Stwierdzono również, że układ regulacji jest też stabilny z innymi macierzami sprzężeń zwrotnych  $\mathbf{F}_i$ , odpowiadających wartościom własnym macierzy  $\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i$  z zakresu  $-1$  do  $-4$ .

Gdybyśmy dążyli do podwyższenia wiarygodności tego rodzaju analizy, to należałoby zbudować model rozmyty TS dokładniejszy, o większej liczbie reguł, pokrywający szerzej interesujący obszar pracy.

Zastosowanie wniosku 2.11 do badania stabilności rozważanego układu regulacji nie prowadzi niestety do konstruktywnych rezultatów, wszystkie trzy macierze badane w tezie wniosku okazują się mieć dodatnie wartości własne. Potwierdza to pogląd autora o konserwatywnym charakterze twierdzenia 2.9 i o niejasnym, jak dotąd, zakresie jego użyteczności.  $\square$

### Ciągły regulator rozmyty TS ze sprzężeniem od wyjścia

W praktyce regulacji procesów przemysłowych naważniejszą rolę odgrywają regulatory ze sprzężeniem od wyjścia obiektu, ponieważ wektor stanu nie jest na ogół pomiarowo dostępny, rzadko dysponujemy precyzyjnym modelem dynamiki. Szczególnie w pętlach regulacji bezpośredniej nadal dominują regulatory typu PID. Jak zaznaczyliśmy we wstępie do niniejszego

rozdziału, w sytuacji bieżących zmian punktów pracy i istotnych nieliniowości regulatory liniowe typu PID nie zawsze zapewniają odpowiednią jakość regulacji. Jednym z najskuteczniejszych rozwiązań jest wówczas zastosowanie regulacji rozmytej typu TS, ze sprzężeniem od dostępnych pomiarowo wyjść obiektu. W rozdziale 2.2.2 przedstawiono zagadnienia projektowania i analizy dyskretnych regulatorów rozmytych typu TS ze sprzężeniem od wyjścia. W niniejszym rozdziale zajmiemy się ciągłymi regulatorami rozmytymi TS ze sprzężeniem od wyjścia, projektowanymi dla ciągłego opisu obiektu. Zaprojektowany regulator ciągły jest następnie przekształcany w dyskretny metodą emulacji i implementowany cyfrowo w sterowniku komputerowym, z okresem próbkowania odpowiednio małym w stosunku do dynamiki obiektu regulacji.

Ograniczymy się do najistotniejszego w praktyce przypadku regulatora PID, przede wszystkim dla uproszczenia zapisu. Metodyka postępowania ma jednak charakter ogólny, słuszny dla dowolnego liniowego dynamicznego algorytmu regulacji.

Podstawowym zagadnieniem przy projektowaniu ciągłego regulatora rozmytego TS, analogicznie jak w przypadku regulatora dyskretnego, jest wybór zmiennych występujących w poprzednikach reguł regulatora i określenie zbiorów rozmytych dla tych zmiennych. Jest to więc podział dziedziny zmiennych określających punkty pracy obiektu na wzajemnie na siebie zachodzące obszary rozmyte tak, aby w każdym z nich obiekt mógł być regulowany odpowiednio dobranym regulatorem liniowym. Punkty pracy obiektu związane są z określonymi stanami obiektu, stąd i zmienne występujące w poprzednikach reguł modelu rozmytego TS będą przede wszystkim zmiennymi stanu określającymi nieliniowe zachowanie obiektu. Jest to głównie sama zmienna regulowana i jej pochodne. Zmienne poprzedników reguł oznaczymy przez  $x_1(t), \dots, x_n(t)$ , a zbiory rozmyte określające podział dziedziny każdej z nich przez  $A_j^i, j = 1, \dots, n_x$ .

Po wyznaczeniu obszarów rozmytych pracy obiektu, w każdym z nich dobieramy nastawy lokalnego ciągłego regulatora PID, np. jedną ze standardowych metod dla lokalnego liniowego modelu obiektu w punkcie centralnym każdego z obszarów. Uzyskujemy w ten sposób *regulator rozmyty ciągły TS-PID* zdefiniowany regułami

$$\begin{aligned} R_{reg}^j : \text{JEŚLI } x_1(t) \text{ jest } A_1^j \text{ i } x_2(t) \text{ jest } A_2^j \text{ i } \dots \text{ i } x_{n_x}(t) \text{ jest } A_{n_x}^j \\ \text{TO } u^j(t) = k_P^j \left[ e(t) + \frac{1}{T_I^j} \int_0^t e(\tau) d\tau + T_D^j \frac{de(t)}{dt} \right], \quad (2.77) \end{aligned}$$

gdzie  $j = 1, \dots, r$  indeksuje reguły (i jednocześnie lokalne obszary pracy),  $e(t) = y^{zad}(t) - y(t)$  jest wartością uchybu regulacji w chwili  $t$ , zaś nastawy

regulatorów lokalnych: wzmocnienie  $k_P^j$ , stała czasowa całkowania  $T_I^j$  i stała czasowa różniczkowania  $T_D^j$ , są współczynnikami funkcji następników reguł.

Sygnał wyjściowy regulatora przyjmuje standardową dla struktur TS postać

$$u(t) = \frac{\sum_{j=1}^r w^j(t) u^j(t)}{\sum_{l=1}^r w^l(t)} = \sum_{j=1}^r \tilde{w}^j(t) u^j(t), \quad (2.78)$$

gdzie  $w^j(t)$  są poziomami aktywacji poszczególnych reguł (2.77) w chwili  $t$ ,

$$w^j(t) = \prod_{p=1}^{n_x} \mu_{A_p^j}(x_p(t)), \quad (2.79)$$

a  $\tilde{w}^j(t)$  ich wartościami unormowanymi.

Implementując zależności (2.77) i (2.78) w środowisku symulacyjnym, takim jak np. Simulink, dostajemy realizację ciągłego regulatora rozmytego TS-PID. W celu symulacji całego układu regulacji należy jeszcze dysponować nieliniowym modelem obiektu regulowanego, sformułowanym w dowolnej postaci akceptowanej przez wykorzystywane środowisko symulacyjne. Jeśli chcemy wesprzeć badania symulacyjne analizą teoretyczną stabilności układu regulacji, to dysponujemy aktualnie metodami takiej analizy wykorzystującymi modele obiektu i regulatora sformułowane w postaci równań stanu. Równania stanu opisują bowiem dynamikę bezpośrednio w dziedzinie czasu, dziedzinie naturalnej dla formułowania nieliniowych modeli rozmytych i analizy stabilności ich punktów równowagi. Podstawowym zaś narzędziem tej analizy jest metoda Lapunowa, zob. np. [61, 159, 137, 155, 20, 154], przy czym praktycznie wszystkie publikacje dotyczą układów z regulatorami z liniowym sprzężeniem od stanu (wyjątkiem jest praca [22] dotycząca układu z regulatorami ze sprzężeniem od wyjścia). Poniżej przeprowadzimy analizę stabilności układu regulacji ciągłej z regulatorem rozmytym TS-PID, ograniczając wyprowadzenie odpowiednich zależności do przypadku regulatora typu PI.

Dla opisu dynamiki układu regulacji trzeba dysponować *modelem rozmytym TS obiektu*, najwygodniej z poprzednikami reguł takimi samymi, jak w regułach regulatora (2.77). W ogólności może to być jednakże model rozmyty z inną liczbą i poprzednikami reguł. Podobnie jak przy opisach układów regulacji w podrozdziałach poprzednich, oznaczymy liczbę jego reguł przez  $ro \neq r$ . Następnik funkcyjny każdej reguły jest lokalnym liniowym modelem obiektu danym w postaci równań stanu

$$\begin{aligned} \dot{x}^i(t) &= \mathbf{A}_i x(t) + \mathbf{B}_i u(t), \\ y^i(t) &= \mathbf{C} x^i(t), \end{aligned}$$

$i = 1, 2, \dots, ro$ . Oznaczając unormowane poziomy aktywacji reguł modelu obiektu przez  $\tilde{w}_o^i(k)$  dostajemy równania tego nieliniowego, kompletnego modelu w postaci

$$\dot{x}(t) = \sum_{i=1}^{ro} \tilde{w}_o^i(t) [\mathbf{A}_i x(t) + \mathbf{B}_i u(t)], \quad (2.80)$$

$$y(t) = \mathbf{C}x(t). \quad (2.81)$$

Równania te rozszerzymy o dynamikę integratora uchybu regulacji

$$\dot{q}(t) = y^{zad} - \mathbf{C}x(t).$$

Reguły regulatora PI mają postać

$$\begin{aligned} R_{reg}^j : \text{ JEŚLI } x_1(t) \text{ jest } A_1^j \text{ i } x_2(t) \text{ jest } A_2^j \text{ i } \dots \text{ i } x_{n_x}(t) \text{ jest } A_{n_x}^j \\ \text{ TO } u^j(t) = k_P^j e(t) + \frac{k_P^j}{T_I^j} q(t), \end{aligned} \quad (2.82)$$

$j = 1, 2, \dots, r$ , zaś wyjście regulatora

$$\begin{aligned} u(t) &= \sum_{j=1}^r \tilde{w}^j(t) [k_P^j e(t) + \frac{k_P^j}{T_I^j} q(t)] \\ &= - \sum_{j=1}^r \tilde{w}^j(t) k_P^j \mathbf{C}x(t) + \sum_{j=1}^r \tilde{w}^j(t) \frac{k_P^j}{T_I^j} q(t) + \sum_{j=1}^r \tilde{w}^j(t) k_P^j y^{zad}. \end{aligned} \quad (2.83)$$

Wstawiając to równanie do (2.80) uzyskujemy

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{ro} \tilde{w}_o^i(t) \left\{ [\mathbf{A}_i - \mathbf{B}_i \sum_{j=1}^r \tilde{w}^j(t) k_P^j \mathbf{C}] x(t) + \right. \\ &\quad \left. + \mathbf{B}_i \sum_{j=1}^r \tilde{w}^j(t) \frac{k_P^j}{T_I^j} q(t) + \mathbf{B}_i \sum_{j=1}^r \tilde{w}^j(t) k_P^j y^{zad} \right\} \\ &= \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(t) \tilde{w}^j(t) \left\{ [\mathbf{A}_i - \mathbf{B}_i k_P^j \mathbf{C}] x(t) + \mathbf{B}_i \frac{k_P^j}{T_I^j} q(t) + \mathbf{B}_i k_P^j y^{zad} \right\}. \end{aligned}$$

Wprowadzając rozszerzony wektor stanu

$$v(t) = [x(t)^T \ q(t)]^T, \quad (2.84)$$

oraz wykorzystując równości  $\sum_{i=1}^{ro} \tilde{w}_o^i(t) = \sum_{j=1}^r \tilde{w}^j(t) = 1$ , można równania dynamiki zamkniętego układu regulacji zapisać w postaci

$$\begin{aligned} \dot{v}(t) &= \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(t) \tilde{w}^j(t) \left\{ \begin{bmatrix} \mathbf{A}_i - \mathbf{B}_i k_P^j \mathbf{C} & \mathbf{B}_i \frac{k_P^j}{T_I^j} \\ -\mathbf{C} & 0 \end{bmatrix} v(t) + \begin{bmatrix} \mathbf{B}_i k_P^j \\ 1 \end{bmatrix} y^{zad} \right\} \\ &= \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(t) \tilde{w}^j(t) \left\{ [\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}] v(t) \right\} + \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(t) \tilde{w}^j(t) \tilde{\mathbf{B}}_{ij} y^{zad}, \end{aligned}$$

gdzie

$$\tilde{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{0} \\ -\mathbf{C} & 0 \end{bmatrix}, \quad \mathbf{G}_{ij} = \begin{bmatrix} \mathbf{B}_i k_P^j \mathbf{C} & -\mathbf{B}_i \frac{k_P^j}{T_I^j} \\ \mathbf{0} & 0 \end{bmatrix}, \quad \tilde{\mathbf{B}}_{ij} = \begin{bmatrix} \mathbf{B}_i k_P^j \\ 1 \end{bmatrix}.$$

Nie zmniejszając ogólności rozważań można przyjąć  $y^{zad} = 0$ . Wówczas równania dynamiki naszego układu przyjmują postać

$$\dot{v}(t) = \sum_{i=1}^{ro} \sum_{j=1}^r \tilde{w}_o^i(t) \tilde{w}^j(t) [\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}] v(t), \quad (2.85)$$

identyczną jak opisana zależnością (2.70) dynamika układu regulacji ze sprzężeniem od stanu rozważana w poprzednim rozdziale, przy czym macierzom  $\mathbf{A}_i$  i  $\mathbf{B}_i \mathbf{F}_j$  w (2.70) odpowiadają macierze  $\tilde{\mathbf{A}}_i$  i  $\mathbf{G}_{ij}$  w (2.85). Do badania stabilności układu dynamicznego opisanego wzorem (2.85) można bezpośrednio wykorzystać twierdzenie 2.8 lub twierdzenie 2.9, jedynie zamiast macierzy  $\mathbf{A}_i$ ,  $i = 1, \dots, r$  trzeba brać pod uwagę macierze  $\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}$ ,  $i = 1, \dots, ro$ ,  $j = 1, \dots, r$ .

Jeśli ponadto założymy, że liczba i poprzedniki reguł modelu obiektu i regulatora są takie same, to definiując macierz

$$\tilde{\mathbf{D}}_{ij} = \frac{1}{2} [(\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}) + (\tilde{\mathbf{A}}_j - \mathbf{G}_{ji})], \quad i < j, \quad i, j = 1, 2, \dots, r, \quad (2.86)$$

analogiczną do macierzy  $\mathbf{D}_{ij}$  (2.72), uzyskujemy zależność (2.85) w postaci przekształconej

$$\dot{v}(t) = \sum_{i=1}^r w^i(t) w^i(t) (\tilde{\mathbf{A}}_i - \mathbf{G}_{ii}) v(t) + 2 \sum_{i < j}^r w^i(t) w^j(t) \tilde{\mathbf{D}}_{ij} v(t), \quad (2.87)$$

odpowiadającej równaniu (2.71). Do badania stabilności układu dynamicznego (2.87) można więc zastosować bezpośrednio twierdzenie 2.10 i wniosek

2.11, zastępując jedynie występujące tam macierze  $\mathbf{A}_i$ ,  $\mathbf{B}_i\mathbf{F}_i$  i  $\mathbf{D}_{ij}$  macierzami  $\tilde{\mathbf{A}}_i$ ,  $\mathbf{G}_{ii}$  i  $\tilde{\mathbf{D}}_{ij}$ .

Zaprezentowany powyżej sposób podejścia do analizy ciągłego układu regulacji ze sprzężeniem od wyjścia obiektu przedstawiony został na przykładzie regulatora PI. W podobny sposób można potraktować ogólny przypadek z dowolnym dynamicznym regulatorem liniowym ze sprzężeniem od wyjścia, w szczególności z regulatorem PID.

Zauważmy, że ze względu na liniowość następników reguł (2.77) zależność (2.78) możemy zapisać w postaci

$$\begin{aligned} u(t) &= \sum_{j=1}^r \tilde{w}^j(t) \cdot k_P^j \left[ e(t) + \frac{1}{T_I^j} \int_0^t e(\tau) d\tau + T_D^j \frac{de(t)}{dt} \right] \\ &= \sum_{j=1}^r \tilde{w}^j(t) k_P^j \cdot e(t) + \sum_{j=1}^r \tilde{w}^j(t) \frac{k_P^j}{T_I^j} \cdot \int_0^t e(\tau) d\tau + \sum_{j=1}^r \tilde{w}^j(t) k_P^j T_D^j \cdot \frac{de(t)}{dt}. \end{aligned} \quad (2.88)$$

Rozważany regulator rozmyty można więc traktować jako nieliniowy regulator PID, o nieliniowo zmiennych nastawach płynnie dopasowujących się do zmian punktu pracy obiektu, gdyż  $\tilde{w}^j(t) = \tilde{w}^j(x(t))$ , zob. (2.79). Ta cecha, w połączeniu ze względną łatwością i zrozumiałością sposobu konstruowania, zadecydowała o dużym zainteresowaniu, powodzeniu regulatorów rozmytych TS, w szczególności regulatorów rozmytych TS-PID. Interpretacja powyższa wskazuje, że w przypadku identycznej struktury następników poszczególnych reguł liniowego regulatora rozmytego TS, tak jak w przypadku regulatora TS-PID, zmiana jego nastaw może być traktowana jako zmiana typu *gain scheduling*, zob. np. [2]. Jednakże sposób prowadzenia tej zmiany jest zupełnie inny: nie przez nieliniową funkcję stanowiącą odwrotność nieliniowości obiektu, ale wykorzystując ogólniejszy i znacznie łatwiejszy w realizacji mechanizm wnioskowania rozmytego.

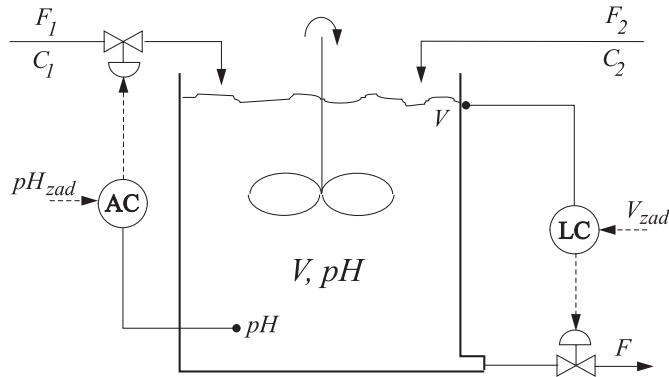
### Przykład 2.8.

Rozważona będzie nieliniowa regulacja wartości pH mieszaniny w reaktorze przepływowym o idealnym wymieszaniu przedstawionym na rysunku 2.31. Dynamika reaktora opisana jest następującymi równaniami:

$$\frac{dV\xi}{dt} = F_1 C_1 - (F_1 + F_2)\xi, \quad (2.89)$$

$$\frac{dV\psi}{dt} = F_2 C_2 - (F_1 + F_2)\psi, \quad (2.90)$$

$$\frac{dV}{dt} = F_1 + F_2 - F, \quad (2.91)$$



Rysunek 2.31. Struktura układu regulacji poziomu i pH w reaktorze przepływowym, przykład 2.8

$$[H^+]^3 + (K_a + \psi) [H^+]^2 + (K_a(\psi - \xi) - K_w) [H^+] - K_a K_w = 0, \quad (2.92)$$

gdzie

$$\xi \sim [HAC] + [AC^-],$$

$$\psi \sim [Na^+],$$

$$pH = -\log_{10} [H^+],$$

$$C_1 = 0.32 \left[ \frac{mol}{l} \right] \text{ jest stężeniem kwasu w dopływie } F_1,$$

$$C_2 = 0.05005 \left[ \frac{mol}{l} \right] \text{ jest stężeniem kwasu w dopływie } F_2,$$

$$V = 1000 [l] \text{ jest objętością mieszaniny},$$

$K_a$  i  $K_w$  są stałymi równowagowymi kwasu i wody,

$$K_a = 1.8 \times 10^{-5}, \quad K_w = 1.0 \times 10^{-14},$$

oraz przyjęto następujące wartości nominalne dopływów:

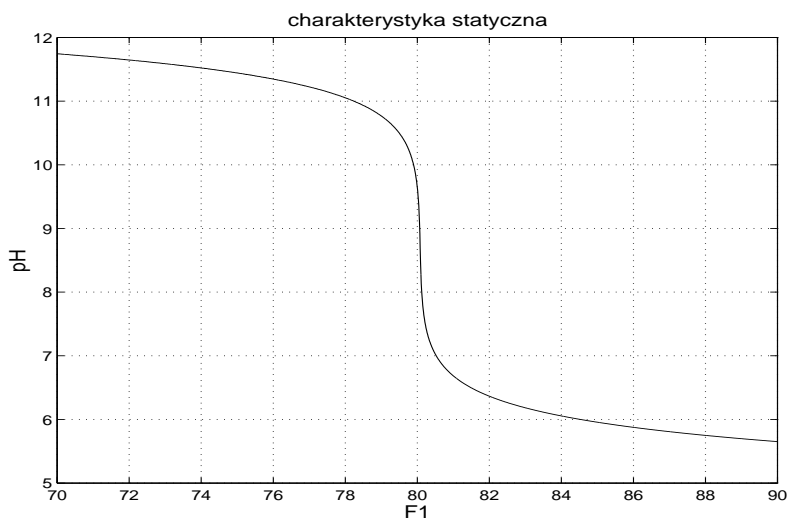
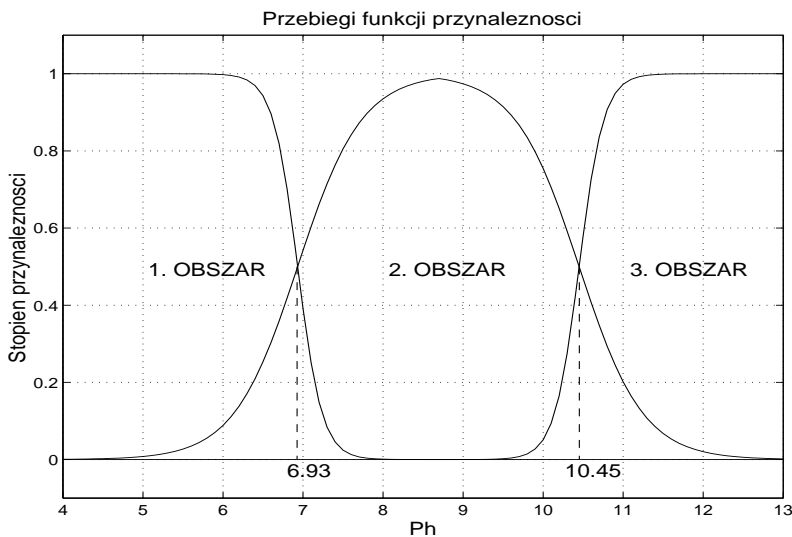
$$F_1(0) = 81 \left[ \frac{l}{min} \right], \quad F_2(0) = 512 \left[ \frac{l}{min} \right].$$

W układzie mamy dwie pętle regulacji, zob. rys. 2.31:

- regulację objętości  $V$  (regulacja poziomu), gdzie zmienną sterującą jest natężenie strumienia wypływu  $F$  z reaktora,
- stężenie  $pH$ , gdzie zmienną sterującą jest natężenie strumienia dopływu  $F_1$ ,

natomiast zakłóceniem są zmiany natężenia dopływu  $F_2$ .

Pierwsza pętla regulacji zapelnienia  $V$  jest standardowa i typowy regulator liniowy łatwo zapewnia odpowiednią jakość regulacji. Natomiast

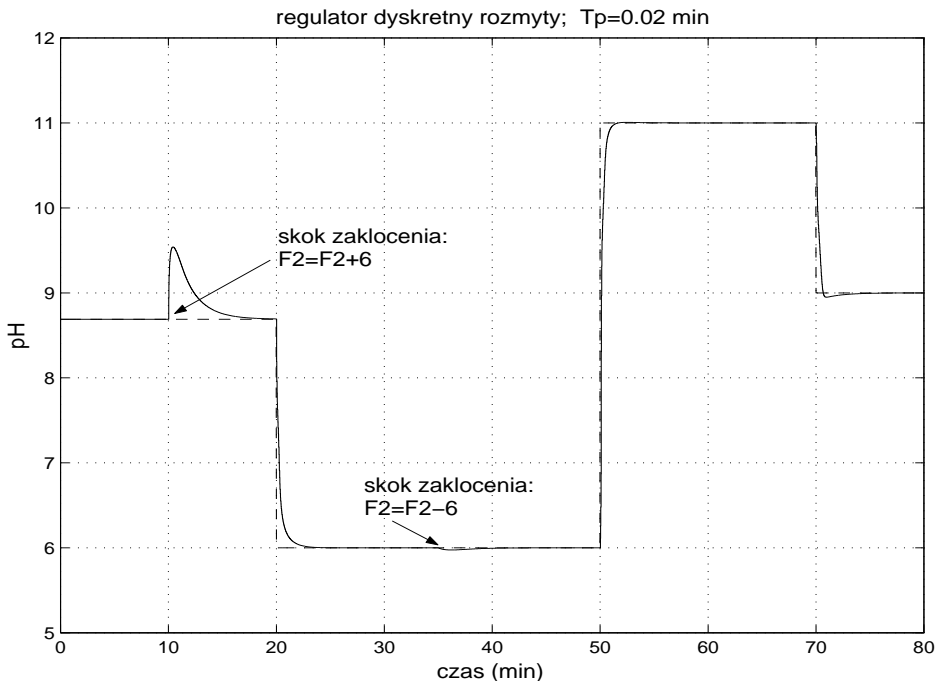
Rysunek 2.32. Charakterystyka statyczna zależności stężenia  $pH$  od dopływu  $F_1$ Rysunek 2.33. Funkcje przynależności reguł regulatora rozmytego  $pH$ 

regulacja wartości  $pH$  okazuje się bardzo trudna, co wynika z silnie nieliniowej charakterystyki statycznej stężenia  $pH$  w zależności od  $F_1$ , jak to pokazano na rysunku 2.32.

Tym bardziej, że interesujący punkt pracy ( $pH = 8.7$ ) znajduje się

w obszarze silnej nieliniowości, która powoduje, że nie można sensownie dobrać jednego regulatora PI dla całego obszaru pracy. Regulator pracujący właściwie w obszarze dużego wzmocnienia ( $pH = 7.5 \div 10$ ) reguluje nieakceptowalnie wolno w obszarach mniejszego wzmocnienia, tj. małych i dużych wartości pH. Z kolei regulator pracujący szybciej w tych obszarach wzbudza się (staje się niestabilny) w obszarze dużego wzmocnienia. Dlatego też zaprojektowano regulator nieliniowy, rozmyty TS [82]. Zmienną występującą w poprzednikach reguł regulatora jest wartość  $pH$ . Przyjęto podział obszaru zmienności  $pH$  na trzy zbiory rozmyte, z sigmoidalnymi funkcjami przynależności pokazanymi na rysunku 2.33. W poszczególnych obszarach zastosowano regulatory PI o nastawach podanych w poniższej tabelce:

$j$	$K_p^j$	$T_I^j$
$j = 1$	-20	1.57
$j = 2$	-1	1.7
$j = 3$	-20.2	1.55



Rysunek 2.34. Przebiegi w układzie regulacji pH z regulatorem rozmytym TS

Algorytm regulatora zastosowano w postaci (2.88), tzn.

$$u(t) = \sum_{j=1}^3 \tilde{w}^j(t) k_P^j \cdot e(t) + \sum_{j=1}^3 \tilde{w}^j(t) \frac{k_P^j}{T_I^j} \cdot \int_0^t e(\tau) d\tau + \sum_{j=1}^3 \tilde{w}^j(t) k_P^j T_D^j \cdot \frac{de(t)}{dt},$$

gdzie  $e(t) = pH^{zad}(t) - pH(t)$ , zaś  $\tilde{w}^j(t)$  są unormowanymi poziomami aktywacji reguł z poprzednikami postaci: „ $pH(t)$  jest OBSZAR  $j$ ”,  $j = 1, 2, 3$ .

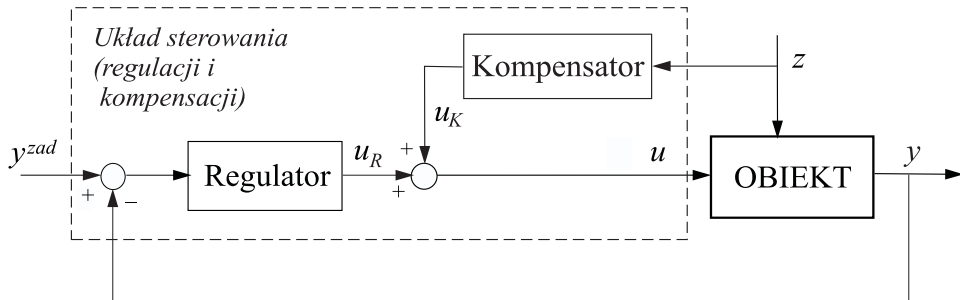
Przykładowy przebieg regulacji, przy skokowych zmianach wartości zadanej i zakłócenia (dopływ  $F_2$ ), pokazany jest na rysunku 2.34. Jest to wynik symulacji z obiektem ciągłym i regulatorem zdyskretyzowanym, z okresem próbkowania  $T_p = 0.02$  min.  $\square$

## 2.4. Kompensacja zakłóceń, strojenie automatyczne

### Kompensacja zakłóceń mierzonych

Przy projektowaniu układów sterowania obowiązuje zasada, że *wpływ zakłóceń mierzonych*, tzn. wejść niesterowanych, ale możliwych do mierzenia, *powinien być kompensowany w otwartej strukturze sterowania* – jeśli tylko wpływ tych zakłóceń na wielkości regulowane jest istotny, dynamika torów zakłócenia i sterowania umożliwia skuteczną realizację kompensacji oraz dysponujemy odpowiednio dokładnymi modelami. Sterowanie w układzie otwartym jest bowiem wówczas szybsze i efektywniejsze niż w strukturze zamkniętej. Dla sterowania w pętli sprzężenia zwrotnego (regulacji) pozostaje wówczas redukcja wpływu pozostałych zakłóceń czy niedokładności modelu oraz jedynie nieidealności kompensacji w torze otwartym. Struktura regulacji z kompensacją zakłócenia mierzonego pokazana jest na rysunku 2.35.

Tor kompensacji zakłócenia działa niezależnie od pętli regulacji nie wpływając na warunki stabilności tej pętli, sterowanie  $u$  oddziałujące na obiekt składa się z sygnału regulatora  $u_R$  oraz sygnału kompensatora  $u_K$ . Algorytm kompensatora projektujemy niezależnie od projektowania algorytmu regulatora. Rozważone w niniejszym rozdziale nieliniowe, rozmyte regulatory TS, zarówno w wersji dyskretnej, jak i ciągłej, generują sygnał regulacji  $u_R$ . Struktury sterowania z tymi regulatorami można łatwo uzupełnić o elementy realizujące kompensację zakłóceń mierzonych w torze otwartym, zgodnie z rysunkiem 2.35. Zasada działania algorytmu kompensatora polega na wytwarzaniu sygnału  $u_K$  znoszącego oddziaływanie



Rysunek 2.35. Struktura regulacji z kompensacją zakłóceń mierzonych

zakłócenia  $z$  na wyjście  $y$  obiektu. Jeśli przez  $G(s)$  i  $G_z(s)$  oznaczymy transmitancje toru sterowania i zakłócenia obiektu, tzn.

$$Y(s) = G(s)U(s) + G_z(s)Z(s),$$

to sygnał wyjściowy idealnego członu kompensującego powinien być równy

$$U_K(s) = -G^{-1}(s)G_z(s)Z(s). \quad (2.93)$$

Oczywiście, trudność projektowania kompensatora polega na tym, że nie zawsze odwrotność transmitancji  $G(s)$  jest stabilna i realizowalna. Przede wszystkim, warunkiem pełnej (modelowej) kompensacji jest, aby opóźnienie w torze sterowania było nie większe niż w torze zakłócenia oraz aby wyliczony zgodnie z wzorem (2.93) liniowy człon kompensujący był układem realizowalnym i stabilnym, zob. np. [48, 3, 54]. Ponadto, dla jakości sterowania w torze otwartym istotna jest zawsze dokładność modelu – ale dla struktury z rysunku 2.35 ten aspekt nie jest krytyczny, gdyż niedokładności kompensacji są zredukowane w pętli sprzężenia zwrotnego.

Do zaprojektowania kompensatora potrzebny jest model obiektu. W rozdziale niniejszym rozważane są obiekty nieliniowe opisywane modelami rozmytymi TS. W obecności istotnych dla zachowania obiektu mierzonych wejść niesterowanych (mierzonych zakłóceń) modele te będą również zależeć od wartości tych zmiennych. Zakłócenia będą więc wpływać, w ogólności, zarówno na postać następników, jak i poprzedników reguł. Rozważmy, dla ustalenia uwagi, przypadek modelu dyskretnego typu ARX. Oznaczając wartość zmierzonego w chwili  $k$ -tej zakłócenia przez  $z(k)$ , poprzedniki i następniki reguł *modelu obiektu rozmytego TS, wykorzystywanego do projektowania regulatora rozmytego z kompensacją zakłóceń* można wówczas zapisać w postaci (por. z (2.36))

$$\begin{aligned}
R_{ob}^i : \text{ JEŚLI } & y(k) \text{ jest } A_0^i \text{ i } y(k-1) \text{ jest } A_1^i \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^i \\
& \text{ i } z(k) \text{ jest } Z_0^i \text{ i } z(k-1) \text{ jest } Z_1^i \text{ i } \dots \text{ i } z(k-p_R) \text{ jest } Z_{p_R}^i \\
& \text{ i } u(k-1) \text{ jest } B_1^i \text{ i } \dots \text{ i } u(k-m_R) \text{ jest } B_{m_R}^i \\
\text{TO } & y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \dots + a_{n_A}^i y(k-n_A+1) + \\
& + b_0^i u(k) + b_1^i u(k-1) + \dots + b_{m_B}^i u(k-m_B) + \\
& + f_0^i z(k) + f_1^i z(k-1) + \dots + f_{p_F}^i z(k-p_F), \quad (2.94)
\end{aligned}$$

gdzie  $i = 1, \dots, r$  indeksuje reguły (i jednocześnie zbiory rozmyte, w ogólności wielowymiarowe) w dziedzinie modelu TS,  $y(k)$ ,  $z(k)$  i  $u(k)$  są wartościami wyjścia, zakłócenia i sterowania w chwili  $k$ -tej,  $A_j^i \in \mathbb{Y}_j$ ,  $B_j^i \in \mathbb{U}_j$ ,  $Z_j^i \in \mathbb{Z}_j$ , zaś  $a_j^i$ ,  $b_j^i$  i  $f_j^i$  są współczynnikami funkcji następników reguł. Elementami każdego ze zbiorów  $\mathbb{Y}_j = \{Y_{j1}, \dots, Y_{j r_{y_j}}\}$  są zbiory rozmyte pokrywające obszar zmiennej  $y(k-j)$ ,  $j = 0, \dots, n_R$ , analogicznie  $\mathbb{U}_j = \{U_{j1}, \dots, U_{j r_{u_j}}\}$  dla  $u(k-j)$ ,  $j = 1, \dots, m_R$  i  $\mathbb{Z}_j = \{Z_{j1}, \dots, Z_{j r_{z_j}}\}$  dla  $z(k-j)$ ,  $j = 0, \dots, p_R$  (zob. komentarz po wzorze (2.36)). Postać poprzednika reguły (2.94) wygląda nieco skomplikowanie, ze względu na ogólność zapisu. W konkretnych aplikacjach występuje z reguły niewiele warunków, w szczególności często nie ma zmiennych opóźnionych, jeśli występują wyjścia  $y$ , to nie występują sterowania  $u$  itp.

Wyjście modelu rozmytego wyliczane jest zgodnie z ogólnym wzorem

$$y(k+1) = \frac{\sum_{i=1}^r [w^i(k) y^i(k+1)]}{\sum_{l=1}^r w^l(k)}, \quad (2.95)$$

gdzie  $w^i(k)$  są poziomami aktywacji poszczególnych reguł (2.94) w chwili  $k$ -tej,

$$w^i(k) = \prod_{j=0}^{n_R} \mu_{A_j^i}(y(k-j)) \prod_{j=0}^{p_R} \mu_{Z_j^i}(z(k-j)) \prod_{j=1}^{m_R} \mu_{B_j^i}(u(k-j)). \quad (2.96)$$

Dla każdego z modeli liniowych występujących w następnikach reguł (2.94) modelu obiektu projektuje się zarówno algorytm regulatora, jak i algorytm kompensatora. Projektowanie regulatora rozważyliśmy w rozdziale 2.2.2 – jedyną różnicą wprowadzoną przez wykorzystanie pomiaru zakłóceń mierzonych jest wystąpienie większej, w ogólności, liczby reguł regulatora, ze względu na możliwość uwzględnienia nieliniowości zależnej od zmiennej zakłócającej. Natomiast algorytm kompensatora projektuje

się zgodnie z ogólną zasadą liniowej kompensacji w torze otwartym przypominaną krótko powyżej (zal. (2.93)), dla rozważanego sposobu modelowania w postaci dyskretnego równania różnicowego. Otrzymujemy w ten sposób następującą ogólną postać *reguł dyskretnego, nieliniowego kompensatora rozmytego TS*

$$\begin{aligned}
 R_{reg}^i : \text{JEŚLI } & y(k) \text{ jest } A_0^i \text{ i } y(k-1) \text{ jest } A_1^i \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^i \\
 & \text{i } z(k) \text{ jest } Z_0^i \text{ i } z(k-1) \text{ jest } Z_1^i \text{ i } \dots \text{ i } z(k-p_R) \text{ jest } Z_{p_R}^i \\
 & \text{i } u(k-1) \text{ jest } B_1^i \text{ i } \dots \text{ i } u(k-m_R) \text{ jest } B_{m_R}^i \\
 \text{TO } & u_K^i(k) = g_1^i u_K(k-1) + g_2^i u_K(k-2) + \dots + g_{m_G}^i u_K(k-m_G) + \\
 & + h_0^i z(k) + h_1^i z(k-1) + \dots + h_{p_H}^i z(k-p_H),
 \end{aligned} \tag{2.97}$$

gdzie  $i = 1, \dots, r$  indeksuje reguły, zaś  $g_1^i$  i  $h_j^i$  są współczynnikami funkcji następników reguł. Sygnał wyjściowy kompensatora przyjmuje standardową dla struktur TS postać

$$u_K(k) = \sum_{i=1}^r \tilde{w}^i(k) u_K^i(k), \tag{2.98}$$

gdzie  $\tilde{w}^i(k)$  są unormowanymi poziomami aktywacji poszczególnych reguł (2.97) w chwili  $k$ -tej. Kompensacja wpływu zakłóceń jest liniowa w każdym z  $r$  lokalnych kompensatorów, ale w efekcie wnioskowania rozmytego cały kompensator rozmyty TS zdefiniowany wzorami (2.97) i (2.98) jest oczywiście *nieliniowy*.

### Strojenie automatyczne regulatora rozmytego TS-PID

W nowoczesnych mikroprocesorowych przemysłowych regulatorach PID jest już standardem oprogramowanie do automatycznego doboru nastaw (tzw. *self-tuning*) na podstawie pewnego, z góry zaprogramowanego eksperymentu czynnego, z reguły typu sterowania przekąźnikowego. Uruchamiając taką opcję użytkownik automatycznie uzyskuje nastawy obliczone przez sam regulator w aktualnym punkcie pracy obiektu, nie musi znać algorytmu czy modelu obiektu wykorzystanego do ich wyznaczenia.

Jeśli wykorzystamy wymieniony powyżej sposób automatycznego doboru nastaw lokalnych regulatorów liniowych, to w pierwszym etapie projektowania regulatora rozmytego TS nie będzie potrzebne wyznaczanie lokalnych modeli obiektu niezbędnych do projektowania następników reguł regulatora rozmytego. Potrzebne jest natomiast zawsze podzielenie zakresów zmiennych określających nieliniowe cechy obiektu na odpowiednie ob-

szary rozmyte (*fuzzy partitioning*) i wyznaczenie położenia punktów centralnych tych obszarów jako punktów pracy, w których dobierane będą automatycznie nastawy regulatorów PID.

W przypadku dużego stopnia zaufania do nastaw regulatorów PID wyznaczanych w punktach pracy (do odporności lokalnych układów regulacji) i przy wystarczająco gęstym, w stosunku do nieliniowości obiektu, pokryciu obszaru pracy oraz zaufaniu do prawidłowości oprogramowania regulatora rozmytego TS-PID można zaryzykować testowanie regulatora od razu na rzeczywistym obiekcie. Użytkownik tego regulatora określa liczbę i położenie punktów pracy, w każdym z nich regulator przeprowadza automatyczny dobór nastaw (*self-tuning*) i na bazie tak uzyskanych regulatorów lokalnych tworzony jest automatycznie regulator rozmyty TS-PID. Zwróćmy uwagę, że regulator jest tu już w finalnej, dyskretnej postaci i nie jest dla użytkownika istotne ani widoczne, jaką metodą regulator dobrał nastawy. Oczywiście, ważnym parametrem projektowym jest okres próbkowania regulatora.

## Rozdział 3

# Regulacja predykcyjna

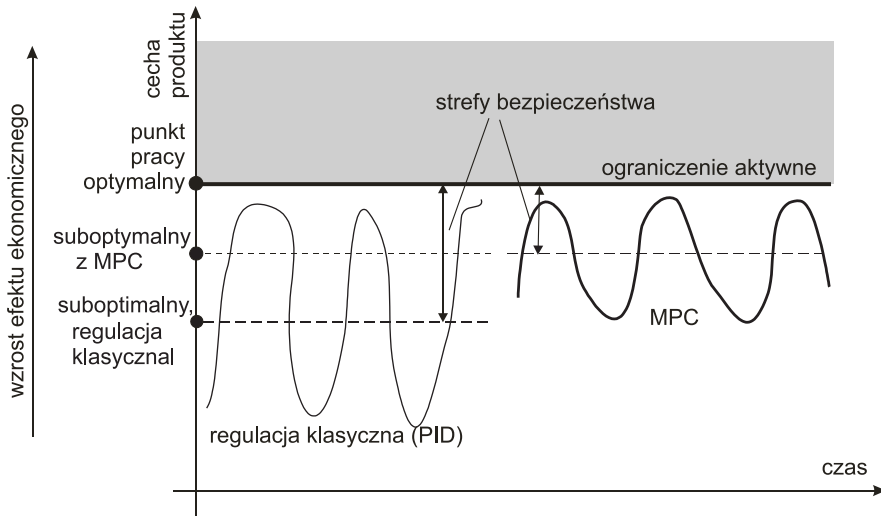
Klasyczne metody projektowania układów regulacji, polegające na doborze regulatora zapewniającego odpowiedni przebieg charakterystyk częstotliwościowych czy położenie biegunów układu zamkniętego, zakładają opis obiektu pomijający ograniczenia sygnałów sterowania. Podobnie, bardziej współczesne metody projektowania oparte na modelu obiektu w postaci układu liniowych równań stanu i prowadzące do regulatora w postaci liniowego sprzężenia od zmiennych stanu, również nie uwzględniają istnienia ograniczeń, zob. np. [47, 54]. Teoretycznie, metodami tymi projektuje się układy regulacji przy założeniu ich działania w niewielkim otoczeniu punktu równowagi obiektu, położonego we wnętrzu zbioru sterowań dopuszczalnych. Zakłada się więc niewielkie zmiany wartości wejść niesterowanych (zakłóceń) i wartości wielkości regulowanych, warunkujące możliwość przyjęcia liniowego opisu dynamiki obiektu i nie uwzględniania ograniczeń sygnałów sterujących. Przyczyną takiego postawienia zadania był oczywiście fakt braku efektywnych metod projektowania algorytmów regulacji uwzględniających nieliniowości i występowanie ograniczeń sygnałów.

Jak pokazuje praktyka przemysłowa, ograniczenia wielkości sterujących (jak i regulowanych) są często bardzo istotne, gdyż sygnały wyprowadzające obiekt ze stanu równowagi mogą przyjmować wartości tak duże, że wygenerowane przez regulatory liniowe sterowania są nierealizowalne fizycznie – o zbyt dużych amplitudach czy też szybkościach narastania. Ignorowanie w pracy regulatora faktu aktywności ograniczenia sygnału sterującego może wówczas prowadzić do dramatycznego pogorszenia jakości regulacji. Próbowano temu zaradzić, stąd w powszechnie wykorzystywanych w praktyce regulatorach przemysłowych PID standardem jest stosowanie struktury z ograniczeniem całkowania (*anti-windup*), w której stan regulatora (integratora) jest dodatkowo korygowany sygnałem z wyjścia elementu wykonawczego (czy modelu nieliniowości elementu wykonawczego), lub różnicą sygnałów przed i po elemencie wykonawczym, zob. np. [54]. Postępowanie takie okazuje się zadowalające w wielu przypadkach, szczególnie w jednowymiarowym układzie regulacji (SISO). Nie rozwiązuje natomiast podstawowo-

wego dla praktyki zadania konstrukcji regulatora w obecności ograniczeń. Zadanie to zostało po raz pierwszy skutecznie rozwiązane przez zaproponowanie *algorytmów regulacji predykcyjnej*, określanych aktualnie powszechnie używanym angielskim skrótem MPC – *Model* (czy też *Model-based*) *Predictive Control*. Algorytmy predykcyjne wyznaczają w każdej chwili próbkowania sterowanie przez optymalizację pewnej funkcji kryterialnej (funkcji celu) zdefiniowanej na horyzoncie o skończonej długości, na którym w oparciu o model predykowane (przewidywane) jest dynamiczne zachowanie obiektu. Przy tym cechą typową jest horyzont o stałej długości, z kroku na krok *przesuwany* (*receding horizon*).

Technika regulacji MPC jest jedyną z tzw. *zaawansowanych technik regulacji* (*advanced control techniques* – bardziej zaawansowanych niż standardowa regulacja PID), która odniosła olbrzymi sukces w aplikacjach praktycznych wywierając dominujący wpływ zarówno na kierunek rozwoju przemysłowych układów regulacji i sterowania, jak i prac badawczych w tym obszarze, zob. np. [102, 1, 38, 98, 87, 9, 144, 10]. Jest kilka przyczyn tego sukcesu. Po pierwsze, algorytmy regulacji predykcyjnej MPC w sposób naturalny potrafią uwzględniać ograniczenia zarówno sygnałów sterujących, jak i ograniczenia wielkości wyjściowych decydujące często o jakości, efektywności i bezpieczeństwie produkcji. Przy tym generują aktualne sterowania uwzględniając zarówno wspomniane ograniczenia, jak i, dzięki bezpośredniemu wykorzystaniu modelu, interakcje wewnętrzne w obiekcie. Stąd w sposób naturalny można je stosować do obiektów wielowymiarowych, również przy nierównej liczbie wejść sterujących i wielkości regulowanych. Po trzecie, są to algorytmy, których zasada działania jest zrozumiała, stosunkowo łatwa do wyjaśnienia personelowi inżynierskiemu, jak i operatorskiemu – aspekt bardzo istotny przy wprowadzaniu nowych technik do praktyki przemysłowej.

Algorytmy predykcyjne były początkowo zastosowane do regulacji nadrzędnej zmiennych procesowych decydujących o jakości, efektywności produkcji (np. DMC do regulacji kolumn destylacyjnych w firmie Shell), gdzie potrafiły wykazać swą skuteczność, wyższość prowadząc do istotnego zmniejszenia wariacji uchybów regulacji, szczególnie w regulacji wielowymiarowej przy występowaniu silnych interakcji. Pozwoliło to na skuteczniejszą optymalizację ekonomiczną prowadzenia procesów, przede wszystkim poprzez przesunięcie punktów pracy obiektów bliżej istotnych ekonomicznie ograniczeń technologicznych, a przez to zwiększenie efektów ekonomicznych. Sytuacja taka pokazana jest poglądowo na rysunku 3.1.



Rysunek 3.1. Typowa sytuacja prowadząca do wzrostu efektu ekonomicznego produkcji po zastosowaniu regulacji MPC

### 3.1. Zasada regulacji predykcyjnej

Ogólna zasada regulacji predykcyjnej (z przesuwającym horyzontem) jest następująca:

w każdej iteracji algorytmu, czyli w każdej kolejnej chwili  $k$  (dokładniej,  $kT_p$ , gdzie  $T_p$  oznacza okres próbkowania, czyli czas powtarzania interwencji regulatora,  $k = 0, 1, \dots$ ), dysponując:

- dynamicznym modelem obiektu, zakładającym określony model zakłóceń,
- pomiarami zmiennych wyjściowych obiektu w chwilach bieżącej i poprzednich oraz poprzednimi wartościami sterowania,
- znaną bądź założoną trajektorią wartości zadanych wyjść regulowanych obiektu w chwili bieżącej  $k$  i chwilach przyszłych,

wyznaczamy wartości sterowań  $u(k) = u(k|k), u(k+1|k), \dots, u(k+N_u-1|k)$ , przyjmując dalej  $u(k+p|k) = u(k+N_u-1|k)$  dla  $p \geq N_u$ , gdzie  $N_u$  to tzw. *horyzont sterowania*. Zastosowana notacja „ $k+p|k$ ” oznacza wyznaczenie w chwili  $k$  wartości przewidywanej na chwilę  $k+p$ . Sterowania wyznaczane są tak, aby zminimalizować różnice między wartościami regulowanych wyjść obiektu  $y(k+p|k)$  przewidywanymi (predykowanymi) w chwili  $k$  a wartościami zadanymi dla tych wyjść  $y^{zad}(k+p|k)$ , na *horyzontie predykcji*  $N$  ( $p = 1, 2, \dots, N$ ). Minimalizacja różnic rozumiana jest

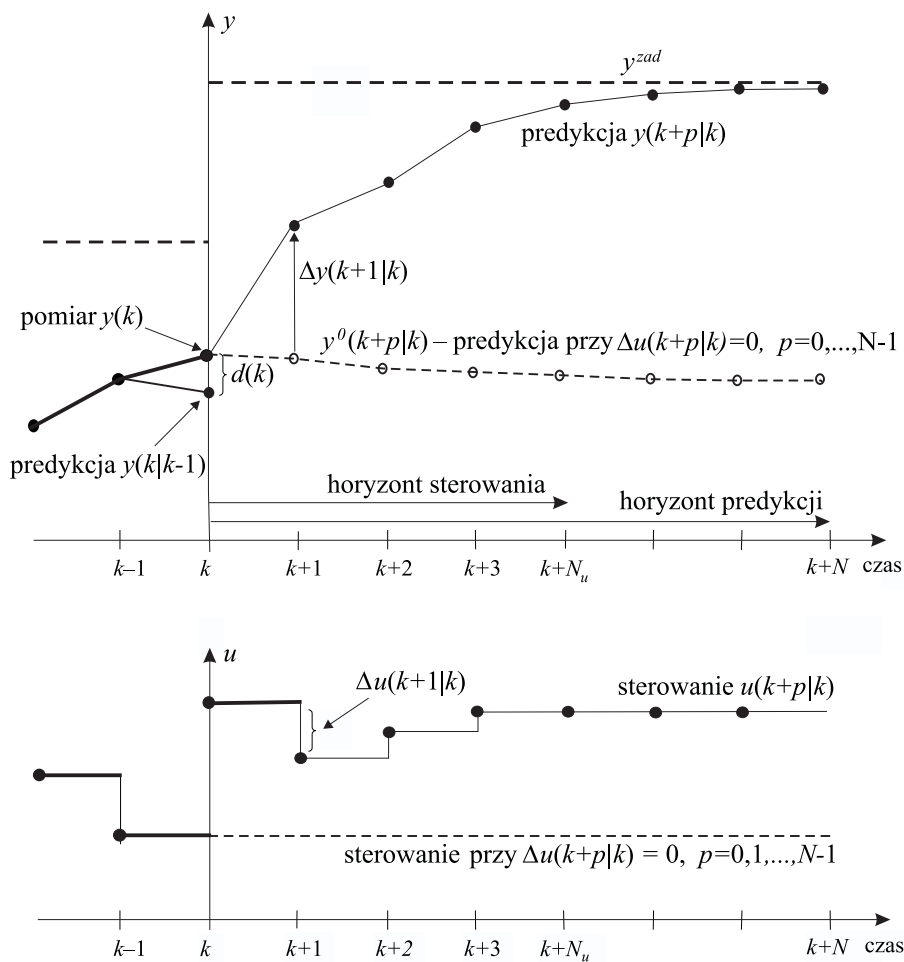
w sensie minimalizacji określonego kryterium jakości regulacji. Do sterowania obiektu wykorzystywany jest jedynie pierwszy element wyznaczonego optymalnego ciągu wartości sterowań, tzn. sterowanie  $u(k) = u(k|k)$ . W kolejnej chwili,  $(k+1)$ -szej, następuje nowy pomiar wyjść obiektu i cała procedura jest powtarzana, z horyzontem predykcji o nie zmienionej długości  $N$ . Stosuje się więc zasadę przesuwanego horyzontu, zwaną też zasadą sterowania repetycyjnego [40]. Cała procedura wyznaczania bieżącej wartości sterowania powtarzana jest w każdej z kolejnych iteracji algorytmu (w każdej z kolejnych chwil  $kT_p$ ). Przedstawiona zasada działania regulacji predykcyjnej zilustrowana jest, dla przypadku obiektu o jednym wejściu i jednym wyjściu (SISO), na rysunku 3.2.

Na rysunku zilustrowano wielkości wykorzystywane do wyznaczania sterowania w bieżącej dyskretniej chwili  $k$ , przedstawiając odpowiednie trajektorie wielkości wyjściowej (zmiennej regulowanej) i sterowania. Na osi poziomej zaznaczono dyskretnie chwile czasu, przy czym  $k$  jest chwilą bieżącą, w której podejmowana jest decyzja o sterowaniu  $u(k) = u(k|k)$ , o wartości ustalonej na czas  $t \in [kT_p, (k+1)T_p)$ . Na rysunku przedstawiono po dwie trajektorie wyjścia i sterowania oraz trajektorię wartości zadanej, w szczególności przedstawiona jest:

- Przewidywana trajektoria wyjścia  $y^0(k+p|k)$ ,  $p = 1, 2, \dots, N$ , odpowiadająca sytuacji utrzymywania przez cały horyzont predykcji sterowania o wartości  $u(k-1)$  wyznaczonej i zastosowanej w chwili poprzedniej, tj.  $u(k+p-1) = u(k-1)$ , dla każdej wartości  $p = 1, 2, \dots, N$ . Obie trajektorie (wyjścia i sterowania) oznaczono dla tego przypadku liniami przerywanymi. Zdefiniowana w ten sposób trajektoria  $y^0(k+p|k)$  przedstawia przebieg wyjścia zależny tylko od *poprzednich* wartości sterowań i w chwili  $k$  nie mamy już na nią wpływu, stąd nazywana jest *składową swobodną* prognozowanej trajektorii wyjść (krócej: *trajektorią swobodną* wyjść prognozowanych).
- Przewidywana trajektoria wyjścia  $y(k+p|k)$  zależna od wartości sterowań zarówno *poprzednich*, jak i *przyszłych* wyznaczanych w chwili  $k$ , tzn. sterowań poprzednich do chwili  $k-1$  i przyszłych  $u(k+p-1|k)$ ,  $p = 1, 2, \dots, N_u$ . Zakłada się przy tym, że horyzont sterowania (dokładniej: horyzont zmienności sterowania)  $N_u$  może być krótszy od horyzontu predykcji  $N$ ,  $N_u \leq N$ . Wymienione trajektorie przedstawione są na rysunku liniami ciągłymi. Przy tym cienkie fragmenty tych linii oznaczają części przewidywane trajektorii wyjścia i sterowania, a pogrubione części trajektorii wyjścia już zrealizowane (odpowiadające pomiarom) i trajektorii sterowania już zrealizowane i realizowane w obiekcie w chwili  $k$ .

- Znana bądź przewidywana trajektoria wartości zadanych dla wyjścia  $y^{zad} = y^{zad}(k+p|k)$ ,  $p = 1, 2, \dots, N$ , przedstawiona pogrubioną linią przerywaną. Wartość zadana wyjścia obiektu przedstawiona na rysunku uległa skokowej zmianie w chwili  $k$ , a następnie pozostaje ustalona. W ogólności może być oczywiście zmienna na horyzoncie predykcji.

Model obiektu używany do wyznaczania przyszłych sterowań zwykle odbiega od rzeczywistości, ponadto występują niedokładnie modelowane czy też niemierzalne zakłócenia – stąd predykcje z reguły różnią się od mierzonych potem rzeczywistych wartości wyjść obiektu. Fakt ten zaznaczono na



Rysunek 3.2. Zasada regulacji predykcyjnej

rysunku jako występujące na wyjściu obiektu w chwili  $k$  zakłócenie oznaczone przez  $d(k)$ ,  $d(k) = y(k) - y(k|k-1)$ , gdzie  $y(k|k-1)$  to wartość wyjścia przewidywana na chwilę  $k$  w chwili poprzedniej,  $(k-1)$ -szej. Uzależnienie w chwili  $k$  trajektorii przewidywanych wyjść obiektu od aktualnie zmierzonej wartości  $y(k)$ , a nie od wartości  $y(k|k-1)$ , oznacza zastosowanie w układzie sterowania dyskretnego *sprzężenia zwrotnego*.

Wyznaczenie wartości sterowań w chwili bieżącej i następnych realizowane jest w algorytmach predykcyjnych w oparciu o model, przez minimalizację określonej funkcji kryterialnej określającej jakość regulacji na horyzoncie predykcji, której zasadniczym składnikiem jest *koszt odchyłeń prognozowanej trajektorii wyjść od trajektorii zadanej*, tzn. koszt prognozowanej trajektorii uchybu regulacji. Ponadto, typowym jest też uwzględnianie w funkcji kryterialnej kar za zmienność wyznaczanych wartości sterowań. Biorąc pod uwagę wymienione dwa składniki formułuje się następującą, najczęściej spotykaną kwadratową postać *funkcji kryterialnej (funkcji celu) regulacji predykcyjnej* wykorzystywaną do wyznaczania sterowania w chwili bieżącej  $k$ :

$$\begin{aligned}
 J(k) &= \sum_{p=N_1}^N (y^{zad}(k+p|k) - y(k+p|k))^T \Psi(p) (y^{zad}(k+p|k) - y(k+p|k)) \\
 &\quad + \sum_{p=0}^{N_u-1} \Delta u(k+p|k)^T \Lambda(p) \Delta u(k+p|k) \\
 &= \sum_{p=N_1}^N \left\| y^{zad}(k+p|k) - y(k+p|k) \right\|_{\Psi(p)}^2 + \sum_{p=0}^{N_u-1} \left\| \Delta u(k+p|k) \right\|_{\Lambda(p)}^2,
 \end{aligned} \tag{3.1}$$

gdzie wektory  $y^{zad}(k+p|k)$  i  $y(k+p|k)$  są o wymiarowości  $n_y = \dim y$  równej liczbie wyjść regulowanych, zaś wektor przyrostów sterowań  $\Delta u(k+p|k)$  jest o wymiarowości  $n_u = \dim u$  równej liczbie sterowań.

W sformułowaniu (3.1) uwzględniono różnice  $y^{zad}(k+p|k) - y(k+p|k)$  między wartościami zadanymi a przewidywanymi wyjść od chwili  $k+N_1$  do chwili końca horyzontu predykcji  $N$ , gdzie  $1 \leq N_1 \leq N$ . Przyjęcie wartości  $N_1 > 1$  jest sensowne wówczas, gdy w obiekcie występuje opóźnienie powodujące w chwilach  $k+1, \dots, k+N_1-1$  brak reakcji wyjść na zmianę sterowania w chwili  $k$ . Oczywiście, przyjęcie  $N_1 = 1$  nie jest błędem również w sytuacji występowania opóźnienia – jedynie wówczas pierwszych  $N_1-1$  składników pierwszej z sum funkcji (3.1) będzie w procesie optymalizacji niepotrzebnie obliczanych, jako niezależnych od wyliczanych wartości sterowań. Czasem jednak przyjmuje się w literaturze przedmiotu  $N_1 = 1$ ,

dla jednolitości zapisu, co jak wspomniano powyżej nie prowadzi do utraty poprawności sformułowań. Długość horyzontu sterowania  $N_u$  spełnia ograniczenia  $0 < N_u \leq N$ . Z reguły przyjmuje się  $N_u < N$ , a powodem takiego postępowania jest dążenie do zmniejszenia wymiarowości zadania optymalizacji, a stąd nakładu obliczeń.

Macierz  $\Psi(p) \geq \mathbf{0}$  jest macierzą wag umożliwiającą różnicowanie wpływu poszczególnych składowych wektora wyjść względem siebie w chwili  $k + p$ , zwykle jest to macierz diagonalna. Jeśli macierze  $\Psi(p)$  są różne dla różnych wartości  $p$ , to następuje też różnicowanie wpływu prognozowanych w poszczególnych chwilach wartości uchybów regulacji  $y^{zad}(k + p|k) - y(k + p|k)$  na wartość funkcji kryterialnej. Jeśli różnicowanie zarówno składowych wektora uchybów względem siebie w poszczególnych chwilach, jak i w różnych chwilach w zakresie horyzontu predykcyjnego nie jest potrzebne, to uzyskujemy najprostszy przypadek  $\Psi(p) = \mathbf{I}$ , gdzie  $\mathbf{I}$  oznacza macierz jednostkową o wymiarze  $n_y \times n_y$ . Rolą macierzy  $\Lambda(p) \geq 0$  jest, z kolei, nie tylko różnicowanie wzajemnego wpływu poszczególnych składowych wektora przyrostów sterowania na rezultaty optymalizacji, ale przede wszystkim określanie wagi wpływu składników drugiej sumy w funkcji (3.1) w stosunku do sumy pierwszej, tzn. wagi składników związanych z tłumieniem zmienności sterowań wobec składników odpowiadających prognozowanym uchybom regulacji. W najprostszym przypadku braku różnicowania wpływu wektorów przyrostów sterowań w zależności od chwili horyzontu predykcyjnego, jak i poszczególnych składowych wektora przyrostów sterowań względem siebie w tych samych chwilach, dostajemy  $\Lambda(p) = \lambda \mathbf{I}$ , gdzie  $\mathbf{I}$  jest macierzą jednostkową o wymiarze  $n_u \times n_u$ . W przypadku  $\Psi(p) = \mathbf{I}$  i  $\Lambda(p) = \lambda \mathbf{I}$  funkcja kryterialna (3.1) przyjmuje prostszą i często spotykaną postać

$$J(k) = \sum_{p=N_1}^N \left\| y^{zad}(k + p|k) - y(k + p|k) \right\|^2 + \lambda \sum_{p=0}^{N_u-1} \left\| \Delta u(k + p|k) \right\|^2, \quad (3.2)$$

gdzie skalar  $\lambda \geq 0$  określa wagę tłumienia zmienności sterowań w stosunku do redukcji uchybów regulacji. Podkreślimy, że przyjęcie  $\lambda = 0$  jest możliwe, ale przy braku ograniczeń na wartości i szybkość zmiany sterowań prowadzi to najczęściej do praktycznie nieakceptowalnych właściwości regulatora, w tym w szczególności do małej odporności na błędy modelowania. Nieco innym, dodatkowym sposobem wpływania na spokojniejszy przebieg sterowania i pożądaną przebieg wyjść, jest przyjmowanie w funkcji kryterialnej (3.1) zamiast trajektorii wartości zadanych wyjść  $y^{zad}(k + p|k)$  tzw. *trajektorii referencyjnej*  $y^{ref}(k + p|k)$ . Stanowi ona pewną trajektorię pośrednią definiującą ciągłe, o zadanym charakterze, przejście między wartością ak-

tualnie zmierzoną wyjść regulowanych  $y(k)$  a trajektorią wartości dla nich zadanych. Zagadnienie to omówimy bliżej w rozdziale 3.6.2.

Wśród wielkości występujących w funkcji kryterialnej (3.1), czy też jej uproszczonej i często wystarczającej postaci (3.2), występują (zob. rys. 3.2):

- znane lub bezpośrednio prognozowane wektory wartości zadanych zmiennych regulowanych  $y^{zad}(k + p|k)$ ,
- przyrosty sterowań  $\Delta u(k + p|k)$  stanowiące zmienne decyzyjne zadania optymalizacji,
- przewidywane wartości zmiennych regulowanych  $y(k + p|k)$  zależne od dotychczasowych (przeszłych) wyjść i sterowań oraz wyznaczanych przyrostów sterowań  $\Delta u(k + p|k)$ .

Dla wyznaczania wartości  $y(k + p|k)$  na horyzoncie predykcji, dla  $p = N_1, \dots, N$ , niezbędne jest dysponowanie odpowiednim *modelem* obiektu regulacji. W ogólności może to być model nieliniowy. Największe znaczenie praktyczne mają jednakże, jak dotąd, *algorytmy MPC z liniowymi modelami obiektów*. Po pierwsze, dość szeroki jest zakres ich bezpośredniego zastosowania. Po drugie, stanowią one podstawę konstrukcji stosunkowo prostych i bardzo często skutecznych algorytmów nieliniowych z linearyzacjami modelu (przedstawimy je w dalszych częściach niniejszego rozdziału).

W przypadku liniowym można, korzystając z zasady superpozycji, przedstawić trajektorię wyjść predykowanych  $y(k + p|k)$  w postaci sumy *trajektorii swobodnej*  $y^0(k + p|k)$  zależnej tylko od już zrealizowanych (przeszłych) sterowań i trajektorii  $\Delta y(k + p|k)$  zależnej tylko od zmiennych decyzyjnych (przyszłych przyrostów sterowań)  $\Delta u(k + p|k)$ . Stąd trajektoria  $\Delta y(k + p|k)$ ,  $p = N_1, \dots, N$  nazywana jest *trajektorią wymuszaną* (precyzyjniej, jest to składowa wymuszana prognozowanej trajektorii wyjść). Mamy więc

$$y(k + p|k) = y^0(k + p|k) + \Delta y(k + p|k), \quad p = N_1, \dots, N.$$

Rozbicie powyższe jest wygodne, choć nie konieczne dla realizacji algorytmu regulacji predykcyjnej, gdyż wartości  $y^0(k + p|k)$ , jako zależne tylko od przeszłości obiektu, oblicza się w danym kroku czasowym  $k$  algorytmu regulacji *tylko jeden raz* i są one dalej dla procesu optymalizacji sterowania jedynie *ustalonymi parametrami*. Natomiast przyrosty  $\Delta y(k + p|k)$ , jako zależne od wyznaczanych zmian sterowania  $\Delta u(k + p|k)$ , wyliczane są w procesie optymalizacji numerycznej *wielokrotnie*, w procesie tym stosowany jest w istocie model zależności tych właśnie przyrostów wyjść od przyrostów sterowań, a więc postać funkcyjna zależności  $\Delta y(k + p|k)$  od  $\Delta u(k + j|k)$ ,  $p = N_1, \dots, N$ ,  $j = 0, 1, \dots, p - 1$ . Uwzględniając dekompozycję

prognozowanej trajektorii wyjść funkcję kryterialną (3.1) można przepisać w postaci

$$J(k) = \sum_{p=N_1}^N \left\| [y^{zad}(k+p|k) - y^0(k+p|k)] - \Delta y(k+p|k) \right\|_{\Psi(p)}^2 + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\Lambda(p)}^2, \quad (3.3)$$

zaś funkcję (3.2) jako

$$J(k) = \sum_{p=N_1}^N \left\| [y^{zad}(k+p|k) - y^0(k+p|k)] - \Delta y(k+p|k) \right\|^2 + \lambda \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|^2. \quad (3.4)$$

Oznaczmy modele liniowe wykorzystywane do predykcji elementów trajektorii wymuszanej  $\Delta y(k+p|k)$  przez macierze  $\mathbf{M}_p$ ,

$$\Delta y(k+p|k) = \mathbf{M}_p [\Delta u(k|k)^T \Delta u(k+1|k)^T \cdots \Delta u(k+p-1|k)^T]^T \quad (3.5)$$

$p = N_1, \dots, N$ , gdyż zależność liniowa w rozważanych przestrzeniach skończenie wymiarowych może być zawsze przedstawiona jako macierz. Dzięki liniowości modeli (3.5) zadanie minimalizacji funkcji (3.3), czy jej nieco uproszczonej postaci (3.4), jest zadaniem minimalizacji *funkcji kwadratowej wypukłej*, ściśle wypukłej jeśli  $\Lambda(p) > 0$  i  $\Psi(p) \geq 0$  ( $\lambda > 0$  przy  $\Lambda(p) = \lambda \mathbf{I}$  i  $\Psi(p) = \mathbf{I}$ ). Jest to więc wówczas zadanie posiadające jednoznaczne minimum będące minimum globalnym. Ponadto jest to zadanie łatwe do rozwiązania analitycznego, jeśli nie ma dodatkowych ograniczeń nierównościowych – lub do szybkiego i niezawodnego rozwiązania numerycznego, gdy takie ograniczenia występują.

Jeśli natomiast do predykcji wartości wyjść regulowanych  $y(k+p|k)$ ,  $p = N_1, \dots, N$ , stosujemy *nieliniowy model obiektu*, to sytuacja jest dużo trudniejsza. Trajektorię swobodną  $y^0(k+p|k)$  możemy również stosunkowo łatwo wyznaczyć, przyjmując w modelu nieliniowym zerowe przyrosty przyszłych sterowań. Nie jest jednakże, w ogólności, możliwe rozdzielenie modelu nieliniowego na niezależne składniki generujące trajektorię swobodną i trajektorię wymuszaną. Przy tym największym problemem jest nieliniowa zależność trajektorii predykowanej  $y(k+p|k)$  od zmiennych decyzyjnych, tzn. wyznaczanych przyrostów sterowania  $\Delta u(k+p|k)$ ,  $p = 0, 1, \dots, N_u - 1$ . Nieliniowość tej zależności powoduje, że zadanie minimalizacji funkcji kryterialnej (3.1) staje się zadaniem *optymalizacji nieliniowej, niewypukłej*.

Dla takich zadań na ogół nie potrafimy wyznaczyć rozwiązania analitycznego nawet w przypadku bez dodatkowych ograniczeń nierównościowych. Ponadto, proces optymalizacji numerycznej jest trudniejszy i znacznie mniej efektywny – można napotkać minima lokalne, czas optymalizacji jest trudny do oszacowania, a rezultat w postaci znalezienia minimum nie jest gwarantowany. Zagadnieniom algorytmów regulacji predykcyjnej z modelami nieliniowymi poświęcimy więcej miejsca w rozdziale 3.5.

W pierwszych latach rozwoju algorytmów MPC wykorzystywano wyłącznie modele liniowe obiektu, później modele nieliniowe – ale przede wszystkim jedynie dla predykcji trajektorii swobodnej wyjść  $y^0(k+p|k)$ . Programy komercyjnych regulatorów MPC do modelowania zależności  $\Delta y(k+p|k)$  w funkcji zmiennych  $\Delta u(k+p|k)$  wykorzystywały modele liniowe i nadal wykorzystują przede wszystkim takie modele.

Optymalny ciąg przyrostów sterowań  $\Delta u(k|k)$ ,  $\Delta u(k+1|k)$ ,  $\Delta u(k+2|k)$ , ...,  $\Delta u(k+N_u-1|k)$  uzyskiwany jest w regulatorach predykcyjnych poprzez *minimalizację funkcji kryterialnej* (3.1). Na początkowym etapie rozwoju algorytmów MPC była to z reguły optymalizacja bez dodatkowych ograniczeń, stosowano też dodatkowy mechanizm wpływu na kształt trajektorii rozwiązania poprzez zastosowanie trajektorii referencyjnej (zob. rozdz. 3.6.2). Dla liniowego modelu obiektu (3.5) optymalizacja bez ograniczeń prowadzi bowiem do rozwiązań analitycznych, określonych formuł matematycznych możliwych do implementacji i obliczania w czasie rzeczywistym (*on-line*) nawet przy niewielkich zasobach i mocach obliczeniowych sprzętu sterującego. Rozwój technologii mikroprocesorowej i wynikający stąd wzrost możliwości obliczeniowych umożliwił rozwiązywanie w czasie rzeczywistym zadań minimalizacji funkcji kryterialnej przy uwzględnianiu dodatkowych ograniczeń nierównościowych, co stało się z kolei jednym z podstawowych powodów rozwoju i znaczącego wzrostu liczby aplikacji przemysłowych algorytmów predykcyjnych.

*Ograniczeniami* istotnymi w zastosowaniach i możliwymi do bezpośredniego uwzględniania w algorytmach MPC są:

— ograniczenia wartości sygnałów sterujących

$$u_{\min} \leq u(k+p|k) \leq u_{\max}, \quad p = 0, 1, \dots, N_u - 1, \quad (3.6)$$

— ograniczenia przyrostów wartości sygnałów sterujących

$$-\Delta u_{\max} \leq \Delta u(k+p|k) \leq \Delta u_{\max}, \quad p = 0, 1, \dots, N_u - 1, \quad (3.7)$$

— ograniczenia wartości sygnałów wyjściowych regulowanych, które w związanej postaci można zapisać jako

$$y_{\min} \leq y(k+p|k) \leq y_{\max}, \quad p = N_1, N_1 + 1, \dots, N. \quad (3.8)$$

Ograniczenia wyjść regulowanych bywają też nakładane nie w każdej z chwil od  $N_1$  do  $N$ , ale jedynie w chwilach *okna ograniczeń* (*constraint window*)  $[N_{cw1}, N_{cw}]$ , gdzie  $N_{cw1} \geq N_1$ ,  $N_{cw} \leq N$ . Jest to jeden ze sposobów przeciwdziałania pustości zbioru dopuszczalnego powodowanej niemożliwością spełnienia ograniczeń wyjść na pierwszym odcinku horyzontu predykcyjnego, jeśli aktualnie zmierzona ich wartość istotnie wykracza poza ograniczenia, z powodu np. wystąpienia silnego zakłócenia. Do kwestii tej powrócimy w rozdziale 3.6.2 omawiając zastosowanie koncepcji okna ograniczeń.

Postać (3.8) przedstawia bezpośrednio najogólniejszy przypadek *ograniczeń dwustronnych*, tzw. ograniczeń pasmowych (*band constraints*). Oczywiście, przypadkiem szczególnym są *ograniczenia jednostronne*, jak np.

$$y(k+p|k) \leq y_{\max}, \quad p = N_1, N_1 + 1, \dots, N.$$

Wartości ograniczające mogą być również zależne od czasu, sytuacja spotykana przede wszystkim właśnie w przypadku ograniczeń wyjść. Przy tym zależność od czasu może być zarówno w sensie położenia w ramach horyzontu predykcyjnego definiowanego indeksem  $p$ , np.  $y_{\max} = y_{\max}(p)$ ,  $p = N_1, N_1 + 1, \dots, N$ , jak i od bieżącej chwili zdefiniowanej indeksem  $k$  (ten ostatni przypadek prowadzi do *niestacjonarności* algorytmu regulacji predykcyjnej).

W zastosowaniach występują też ograniczenia wartości *sygnałów wyjściowych nieregulowanych*  $y^n(k)$ , sformułować je można w analogicznej do (3.8) ogólnej postaci

$$y_{\min}^n \leq y^n(k+p|k) \leq y_{\max}^n, \quad p = N_1, N_1 + 1, \dots, N. \quad (3.9)$$

Zmienne wyjściowe nieregulowane  $y^n(k)$  o wartościach ograniczanych są na ogół bezpośrednio mierzone, zaś wartości prognozowane  $y^n(k+p|k)$  wyznaczamy z wykorzystaniem odpowiedniego modelu, analogicznie jak wartości  $y(k+p|k)$ . W rozważaniach niniejszego rozdziału nie będziemy rozróżniać zmiennych wyjściowych regulowanych i nieregulowanych, aby nie wydłużać zapisu.

Zdefiniowaliśmy powyżej wszystkie elementy *zadania optymalizacji* rozwiązywanego w pojedynczym kroku (dyskretnej chwili  $k$ ) *algorytmu predykcyjnego MPC*, dla wyznaczenia przebiegów sterowań na horyzoncie sterowania  $N_u$ . Zadanie to, dla liniowego modelu obiektu, można sformułować

następująco:

$$\min_{\Delta u(k|k), \dots, \Delta u(k+N_u-1|k)} \left\{ \sum_{p=N_1}^N \|[y^{zad}(k+p|k) - y^0(k+p|k)] + \right. \\ \left. -\Delta y(k+p|k)\|_{\Psi(p)}^2 + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\mathbf{A}(p)}^2 \right\}$$

z ogr. : (3.6), (3.7), (3.8), (3.10)

$$\Delta y(k+p|k) = \mathbf{M}_p[\Delta u(k|k)^T \cdots \Delta u(k+p-1|k)^T]^T, \\ p = N_1, \dots, N.$$

Istotnym problemem przy rozwiązywaniu zadania optymalizacji (3.167), zadania z ograniczeniami nierównościami na wartości sterowań i wyjść regulowanych, jest możliwość wystąpienia *пустego zbioru rozwiązań dopuszczalnych*. Oznacza to brak możliwości jednoczesnego spełnienia ograniczeń (3.6), (3.7), (3.8) i równań wyjść (3.5) dla pewnych stanów zależnych od dynamiki obiektu i realizacji zakłóceń, czyli dla wartości aktualnych i poprzednich wyjść regulowanych i nieregulowanych, sterowań i zakłóceń. Zwróćmy uwagę, że sytuacja taka może zaistnieć jedynie wówczas, gdy uwzględnia się ograniczenia na wartości wyjść. Same ograniczenia wartości i prędkości zmiany sterowań nie powodują pustości zbioru dopuszczalnego – oczywiście, o ile od samego początku nie są sprzecznie sformułowane, co jest wymagającym korekty błędem określenia ograniczeń. Sytuacja braku rozwiązań dopuszczalnych nie powinna wystąpić w praktycznych implementacjach algorytmu, poza co najwyżej specjalnie traktowanymi odosobnionymi przypadkami. Do kwestii zapewniania niepustości zbioru dopuszczalnego powrócimy w rozdziale 3.6.2.

Historycznie rzecz ujmując, pierwsze praktyczne realizacje predykcyjnych algorytmów regulacji z przesuwającym horyzontem pojawiły się w latach siedemdziesiątych ubiegłego stulecia, opracowane niezależnie w różnych ośrodkach – chociaż sama idea została sformułowana wcześniej [78], zob. [98]. Daty pierwszych publikacji nie są tu szczególnie miarodajne, gdyż algorytmy predykcyjne były proponowane i implementowane przede wszystkim w kręgach związanych z przemysłem, gdzie udane aplikacje nowych pomysłów nierzadko znacznie wyprzedzają ich opublikowanie. Algorytmy te, oparte na ogólnej zasadzie regulacji predykcyjnej, różniły się przede wszystkim sposobem modelowania obiektu regulacji, poza tym szczegółami formułowania funkcji kryterialnej i zadania optymalizacji, sposobem jego rozwiązania, podejściem do traktowania ograniczeń.

Pierwszymi publikacjami dotyczącymi przemysłowo zastosowanego algorytmu typu MPC były prace Richaleta i in. [123, 124], dotyczące algorytmu nazwanego *Model Predictive Heuristic Control* (MPHC), zaimplementowanego w komercyjnym oprogramowaniu pod nazwą IDCOM (*IDentification and COMmand*), znanego bardziej pod nazwą algorytm MAC (*Model Algorithmic Control*). Do predykcji wyjść regulowanych zastosowano tu model liniowy obiektu w postaci skończonej odpowiedzi impulsowej, funkcję kryterialną sformułowano jako sumę kwadratowych składników kary za odchylenia trajektorii prognozowanej od trajektorii referencyjnej. Trajektorja ta jest „pośrednią” między trajektoriami odpowiedzi swobodnej modelu i wartości zadanej, zaczyna się w aktualnie zmierzonych wartościach wyjść i zdąża na horyzoncie predykcji do trajektorii wartości zadanych, z reguły opisana jest dynamiką pierwszego rzędu, zob. rozdz. 3.6.2. W algorytmie MAC uwzględniano ograniczenia, a sterowania wyliczano heurystycznym algorytmem iteracyjnym. Duży nacisk położono na łatwość strojenia algorytmu.

Jednym z pierwszych był również algorytm *Dynamic Matrix Control* (DMC), opracowany w pierwszej połowie lat siedemdziesiątych ub. stulecia przez pracowników firmy Shell Oil i nieco później opublikowany, wraz z informacjami o udanych aplikacjach [29, 119]. Algorytm ten stał się bardzo popularny w komercyjnych aplikacjach algorytmów regulacji predykcyjnej. Algorytm DMC wykorzystuje liniowy model obiektu w postaci odpowiedzi skokowych oraz kwadratową funkcję kryterialną ze składnikami kary za zmiany sterowań (jak w (3.4)).

Uważa się, że IDCOM i DMC reprezentują *pierwszą generację* algorytmów MPC, zob. np. [1, 54, 87]. Algorytmy te wywarły olbrzymi wpływ na rozwój praktyki i teorii regulacji przemysłowej. W algorytmach tych sterowanie nie było jeszcze wyznaczane na drodze numerycznego rozwiązywania zadania optymalizacji z ograniczeniami w każdej iteracji.

Główną wadą algorytmów MPC pierwszej generacji było traktowanie ograniczeń w sposób przybliżony, heurystyczny. W algorytmie DMC wadę tę usunięto proponując jego wersję pod nazwą Quadratic Dynamic Matrix Control (QDMC) [51]. W każdej chwili próbkowania sterowania są wyznaczane z rozwiązania zadania minimalizacji kwadratowej funkcji kryterialnej (takiej jak w DMC) przy liniowych ograniczeniach na sterowania i wyjścia, tzn. zadania programowania kwadratowego. Algorytm DMC w wersji QDMC uważany jest za reprezentanta *drugiej generacji* algorytmów MPC, której cechą charakterystyczną jest rozwiązywanie zadania programowania kwadratowego zapewniającego bezpośrednio uwzględnianie ograniczeń nierównościowych.

Bardzo popularnym okazał się zaproponowany znacznie później przez Clarka i in. algorytm *Generalized Predictive Control* (GPC) [26, 25], wykorzystujący model procesu w postaci transmitancji dyskretnej (równań różnicowych). Ten sposób modelowania pozwolił na uwzględnianie w algorytmie MPC szerszej klasy modeli zakłóceń, w porównaniu do stosowanych w algorytmach DMC i MPHC. Interesującym rozwinięciem algorytmu GPC jest zaproponowany przez Clarka i Scattoliniego algorytm CRHPC (Constrained Receding Horizon Predictive Control), jedno z pierwszych sformułowań algorytmów predykcyjnych ze skończonym, przesuwającym horyzontem i z teoretycznie wykazaną stabilnością pętli regulacyjnej [27].

W algorytmie SMOC [91] zastosowano po raz pierwszy model obiektu w postaci równań stanu – podejście, które dominuje aktualnie w pracach badawczych dotyczących algorytmów MPC. Zastosowano również obserwator stanu dla estymacji pomiarowo niedostępnych współrzędnych wektora stanu. Wprowadzono ponadto rozróżnienie między regulowanymi i mierzonymi zmiennymi wyjściowymi obiektu.

SMOC jest już przykładem algorytmu MPC nowoczesnego, *trzeciej generacji*, podobnie jak algorytmy IDCOM-M firmy Setpoint, RMPCT firmy Honeywell, 3dMPC firmy ABB i inne.

Aktualnie w ofercie wszystkich znaczących producentów sprzętu i oprogramowania dla sterowania i regulacji obiektów przemysłowych znajdują się pakiety oprogramowania z algorytmami MPC, istnieją też firmy specjalizujące się w aplikacjach tego typu. Bardziej znane produkty dostępne na rynku wymieniono poniżej (przegląd autora wykonany w 2012 r.):

- **DMCplus** firmy AspenTech, sukcesor klasycznego algorytmu DMC (Cutler i Ramaker, 1979),
- **IDCOM-HEICON** firmy Sherpa Engineering, sukcesor algorytmu **IDCOM** (Richalet i in, 1978), do 2006 r. rozwijanego i oferowanego przez firmę Adersa,
- **SMOC (Supervisory Multivariable Optimizing Controller)** oferowany przez Shell Global Solutions,
- **3dMPC** oferowany przez firmę ABB,
- **Profit Controller** i **Profit NLC Controller** (Nonlinear Controller) firmy Honeywell (elementy szerszego środowiska Profit Suite),
- **Pavilion8** firmy Pavilion Technologies (grupa Rockwell Automation),
- **Connoisseur** firmy Invensys,
- **INCA MPC** firmy IPCOS,

- **DeltaV PredictPro** firmy Emerson (może być też realizowany w sterownikach systemu DCS DeltaV),
- **MVC** firmy GE Energy (dawniej produkt firmy Continental Controls).

W pracach prowadzonych w ostatnich latach i obecnie, w zakresie badań i zastosowań algorytmów MPC, dominują problemy uwzględniania nieliniowych modeli obiektów, stabilności, zapewniania odporności na niepewność modelowania i estymacji zakłóceń. Można uznać, że problematyka ta wyróżnia algorytmy MPC *czwartej generacji*.

W kolejnych częściach niniejszego rozdziału przedstawimy wybrane algorytmy predykcyjne, uwzględniając przy tym wyniki prac badawczych autora. Na początku przedstawimy szczegółowo konstrukcję i podstawowe cechy algorytmu predykcyjnego na przykładzie algorytmu DMC z modelem liniowym obiektu. Wybór ten podyktowany jest przede wszystkim faktem, że na przykładzie wykorzystywanego w algorytmie DMC modelu obiektu w postaci odpowiedzi skokowej najłatwiej, a przy tym w najbardziej naturalny sposób można wyprowadzić i zinterpretować zależności opisujące predykcyjny algorytm regulacji. Jest to bowiem model sformułowany w dziedzinie czasu, w której to dziedzinie formułowane są algorytmy predykcyjne. Algorytm DMC powstał w przemyśle petrochemicznym, był jednym z pierwszych z dużym powodzeniem zastosowanych w praktyce i stąd popularnych algorytmów predykcyjnych. Co więcej, oprogramowanie wykorzystujące zasadę DMC jest nadal jednym z najbardziej rozpowszechnionych w zastosowaniach przemysłowych, gdyż identyfikacja modelu obiektu w postaci jego odpowiedzi skokowej jest jednym z najprostszych, a zarazem najsukuczniejszych sposobów. Następnie przedstawimy algorytm GPC, wykorzystujący model w postaci liniowego dyskretnego równania różnicowego (model transmitancyjny), będący w historii regulacji predykcyjnej drugim istotnym przedstawicielem rodziny algorytmów MPC. W ostatniej kolejności, ale za to bardzo obszernie, omówimy algorytm MPC z modelem w postaci liniowych równań stanu (określany w książce akronimem MPCs). Przedstawimy najpierw algorytm dla sytuacji ze stanem mierzonym, następnie ze stanem estymowanym, omawiając zagadnienia estymacji stanu dla wykorzystania w tym algorytmie. Przedstawiając poszczególne algorytmy podkreślać będziemy ich cechy wspólne i różnice między nimi.

Zaprezentowane algorytmy z modelem liniowym będą punktem wyjścia do przedstawienia koncepcji algorytmów MPC wykorzystujących modele nieliniowe obiektu regulacji. W obszarze tym skoncentrujemy się przede wszystkim na algorytmach wykorzystujących bieżące, powtarzane linearyzacje modelu nieliniowego – jako na algorytmach pewnych i szybkich w implementacji, a jednocześnie skutecznych dla szerokiej klasy obiek-

tów nieliniowych. Algorytmy te omówimy ponadto nieco szczegółowiej dla przypadku obiektów opisanych nieliniowymi, rozmytymi modelami Takagi-Sugeno. W tym bowiem przypadku szczególnie łatwa jest realizacja algorytmów nieliniowych z linearyzacjami.

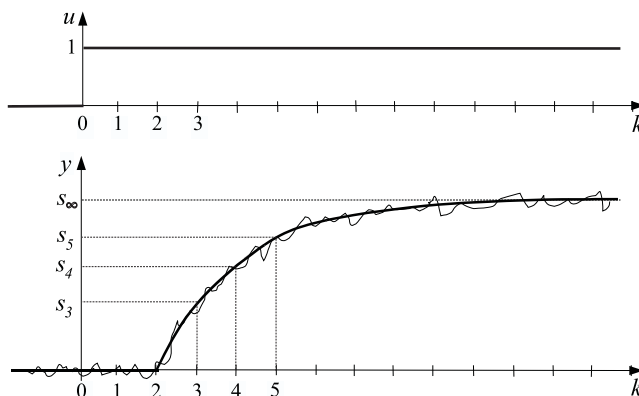
W końcowej części rozdziału przedstawimy krótko zagadnienia stabilności algorytmów predykcyjnych i modyfikacji gwarantujących stabilność. Poruszymy ponadto kwestie doboru parametrów, jak również specyficzne i istotne dla algorytmów predykcyjnych zagadnienie projektowania zapewniającego niepustość zbioru dopuszczalnego w zadaniu optymalizacji bieżącej.

## 3.2. Algorytm DMC (Dynamic Matrix Control)

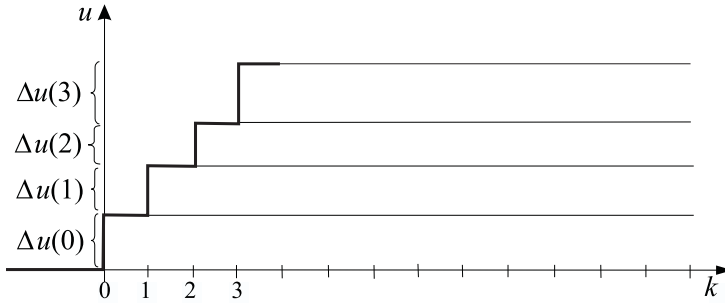
### 3.2.1. Predykcja wyjść modelem odpowiedzi skokowych

#### Obiekt jednowymiarowy (SISO)

W algorytmie DMC dynamika obiektu regulacji modelowana jest dyskretnymi odpowiedziami skokowymi opisującymi reakcję wyjść na skoki jednostkowe na wejściach sterujących. Odpowiedź wyjścia obiektu na skok sterowania jest w wielu zastosowaniach wygodnym sposobem modelowania dynamiki obiektu. Na rysunku 3.3 przedstawiono przykładową odpowiedź skokową obiektu jednoinercyjnego z opóźnieniem, gdzie opóźnienie  $\tau = 2T_p$  ( $T_p$  – okres próbkowania), przy czym zaznaczono zarówno odpowiedź rzeczywistą, jak i odpowiedź z idealnie odfiltrowanymi zakłóceniami szybkozmiennymi (krzawa pogrubiona).



Rysunek 3.3. Przykład odpowiedzi wyjścia obiektu  $y$  na skok sterowania  $u$



Rysunek 3.4. Reprezentacja sygnału schodkowego jako sumy sygnałów skokowych o różnych amplitudach  $\Delta u(k)$  i momentach skoku

Znając dyskretną odpowiedź skokową obiektu  $\{s_1, s_2, s_3, \dots\}$  można modelować dyskretną odpowiedź wyjścia na sterowanie w postaci dowolnego sygnału dyskretnego (schodkowego). Można bowiem traktować sygnał schodkowy jako sumę skoków rozpoczynających się w kolejnych chwilach, o różnych amplitudach – jak to przedstawiono na rysunku 3.4. Korzystając z zasady superpozycji możemy wtedy napisać:

$$y(1) = y(0) + s_1 \Delta u(0),$$

$$y(2) = y(0) + s_2 \Delta u(0) + s_1 \Delta u(1),$$

$$y(3) = y(0) + s_3 \Delta u(0) + s_2 \Delta u(1) + s_1 \Delta u(2),$$

$$y(4) = y(0) + s_4 \Delta u(0) + s_3 \Delta u(1) + s_2 \Delta u(2) + s_1 \Delta u(3),$$

i tak dalej dla kolejnych chwil czasu. Stąd dla dowolnego  $k = 1, 2, 3, \dots$  dostajemy zależność

$$y(k) = y(0) + \sum_{j=1}^k s_j \Delta u(k-j). \quad (3.11)$$

Wartość wyliczana na podstawie (3.11) na chwilę  $k+p$  jest równa

$$y(k+p) = y(0) + \sum_{j=1}^{k+p} s_j \Delta u(k+p-j). \quad (3.12)$$

Wzór (3.12) zastosujemy do wyznaczenia *równań predykcji* w algorytmie DMC. Przypomnijmy, że wartość wyjścia przewidywaną w chwili (bieżącej)  $k$  na chwilę  $k+p$  horyzontu predykcji oznaczamy przez  $y(k+p|k)$ , zaś wartości przyrostów sterowań wyznaczane w chwili  $k$  na chwilę przyszłe przez  $\Delta u(k+p|k)$ . Wartość  $y(k+p|k)$  równa jest sumie wartości wyliczanej

z modelu (3.12) oraz przewidywanej na chwilę  $k + p$  wartości  $d(k + p|k)$  zakłócenia spowodowanego do wyjścia

$$y(k + p|k) = y(0) + \sum_{j=1}^p s_j \Delta u(k + p - j|k) + \sum_{j=p+1}^{k+p} s_j \Delta u(k + p - j) + d(k + p|k).$$

Zakłócenie w chwili  $k$  przyjmujemy jako równe różnicy wartości zmierzonej wyjścia  $y(k)$  i wartości wyliczanej z modelu,

$$d(k) = y(k) - \left[ y(0) + \sum_{j=1}^k s_j \Delta u(k - j) \right]. \quad (3.13)$$

W algorytmie DMC zakłada się brak wiedzy o zmianach wartości zakłócenia na horyzoncie predykcji, stąd przyjmuje się następujący jego model (zwany też *modelem zakłócenia typu DMC*, zob. np. [87])

$$d(k + 1|k) = d(k + 2|k) = \dots = d(k + N|k) = d(k),$$

tzn. na horyzoncie predykcji wartość zakłócenia przyjmujemy jako stałą, równą wyznaczonej w chwili  $k$  wartości (3.13).

Łącząc uzyskane powyżej zależności dostajemy następujący wzór na predykcje wyjścia w chwili  $k$

$$\begin{aligned} y(k + p|k) &= y(k) + \sum_{j=1}^p s_j \Delta u(k + p - j|k) + \\ &+ \sum_{j=p+1}^{k+p} s_j \Delta u(k + p - j) - \sum_{j=1}^k s_j \Delta u(k - j), \quad p = 1, \dots, N, \end{aligned}$$

gdzie  $y(k)$  jest *wartością wyjścia zmierzoną w chwili  $k$* , a  $\Delta u(k - 1)$ ,  $\Delta u(k - 2)$ , ... zrealizowanymi już przyrostami sterowań. Przekształcając dalej mamy

$$\begin{aligned} y(k + p|k) &= \sum_{j=1}^p s_j \Delta u(k + p - j|k) + \\ &+ y(k) + \sum_{j=1}^k s_{j+p} \Delta u(k - j) - \sum_{j=1}^k s_j \Delta u(k - j) \\ &= \sum_{j=1}^p s_j \Delta u(k + p - j|k) + y(k) + \sum_{j=1}^k (s_{j+p} - s_j) \Delta u(k - j). \end{aligned}$$

Pierwsza z sum po prawej stronie uzyskanej zależności zależy od przyrostów sterowania w chwilach bieżącej i przyszłych,  $\Delta u(k|k)$ ,  $\Delta u(k+1|k)$ , ..., które są *zmiennymi decyzyjnymi* zadania optymalizacji sterowań w chwili  $k$ . Stąd nazywamy ją *składową wymuszaną* trajektorii wyjść prognozowanych:

$$\Delta y(k+p|k) = \sum_{j=1}^p s_j \Delta u(k+p-j|k), \quad p = 1, 2, \dots, N. \quad (3.14)$$

Natomiast pozostała część trajektorii prognozowanej jest *składową swobodną* zależną jedynie od poprzednich przyrostów sterowania i dana jest wzorem

$$y^0(k+p|k) = y(k) + \sum_{j=1}^k (s_{j+p} - s_j) \Delta u(k-j), \quad p = 1, 2, \dots, N. \quad (3.15)$$

Dla rzeczywistych obiektów stabilnych (bez całkowania) stan wyjścia po wymuszeniu skokowym ustala się,  $\lim_{k \rightarrow \infty} s_k = s_\infty$ , zob. rys. 3.3. Stąd, wystarczy znać  $D$  współczynników odpowiedzi skokowej, tzn. liczbę kroków dyskretyzacji, po której wartość odpowiedzi skokowej można uznać za ustaloną, równą wzmocnieniu statycznemu obiektu  $k_m = s_\infty$ .  $D$  nazwiemy *horyzontem dynamiki* obiektu, na ogół słuszne jest następujące oszacowanie

$$D \cong (\tau + (3 \div 4)T)/T_p,$$

gdzie  $T$  to tzw. zastępcza stała czasowa obiektu. Jeśli  $s_j = k_m$  dla  $j \geq D$ , to zależność (3.15) przyjmuje postać

$$y^0(k+p|k) = y(k) + \sum_{j=1}^{D-1} (s_{j+p} - s_j) \Delta u(k-j), \quad p = 1, 2, \dots, N, \quad (3.16)$$

gdyż  $s_{j+p} - s_j = 0$  dla  $j \geq D$ .

Przedstawiony sposób modelowania w postaci skończonych odpowiedzi skokowych jest stosowalny do obiektów asymptotycznie stabilnych. W praktyce przemysłowej istotną rolę odgrywają natomiast jeszcze *obiekty stabilne z całkowaniem*. Dla tej klasy obiektów łatwo jest uogólnić rozważane modelowanie odpowiedzią na pobudzenie skokowe – stosując zamiast rozważanej powyżej zwykłej odpowiedzi skokowej tzw. *przyrostową odpowiedź skokową*. Obliczamy mianowicie, jako wyjścia modelu, przyrosty wyjść obiektu  $\Delta y(k) = y(k) - y(k-1)$ ,  $k = 1, 2, 3, \dots$ . Zakładając dla  $k < 0$  stan ustalony oraz skok jednostkowy sterowania w chwili  $k = 0$ ,  $\Delta u(0) = 1$ , mamy wówczas

$$\begin{aligned}\Delta y(1) &= y(1) - y(0) = \tilde{s}_1 = s_1, \\ \Delta y(2) &= y(2) - y(1) = \tilde{s}_2 = (s_2 - s_1), \\ \Delta y(3) &= y(3) - y(2) = \tilde{s}_3 = (s_3 - s_2), \\ &\vdots\end{aligned}$$

gdzie  $\tilde{s}_k$  oznaczają elementy przyrostowej odpowiedzi skokowej, a  $s_k$  elementy uprzednio wprowadzonej zwykłej odpowiedzi skokowej. Dla obiektu stabilnego z pojedynczym całkowaniem różnicy między kolejnymi wartościami wyjść w zwykłej odpowiedzi skokowej ustalają się po zaniknięciu procesu przejściowego odpowiadającego horyzontowi dynamiki  $D$  – ustala się wartość końcowa przyrostowej odpowiedzi skokowej, tzw. wzmocnienie prędkościowe  $\tilde{k}_m$ . Dla dowolnego wymuszenia traktowanego jako suma skoków dostajemy stąd odpowiednik podstawowej zależności (3.11) w postaci

$$\Delta y(k) = \sum_{j=1}^k \tilde{s}_j \Delta u(k-j). \quad (3.17)$$

Wszystkie rozważania zamieszczone dalej, dotyczące projektowania algorytmu regulacji predykcyjnej dla obiektu stabilnego modelowanego odpowiedzią skokową, można łatwo rozszerzyć na obiekt z całkowaniem, modelowany przyrostową odpowiedzią skokową.

### Obiekt wielowymiarowy (MIMO)

Rozważmy teraz obiekt wielowymiarowy (MIMO – *Multi-Input Multi-Output*), o  $n_y$  wyjściach regulowanych i  $n_u$  sterowaniach. Rozpatrzmy model obiektu w postaci zestawu  $n_y \cdot n_u$  skończonych odpowiedzi skokowych  $\{s_l^{ij}, l = 1, 2, \dots, D\}$ , gdzie indeks  $i$  określa numer zmiennej wyjściowej,  $i = 1, 2, \dots, n_y$ , a indeks  $j$  numer sterowania,  $j = 1, 2, \dots, n_u$ . Tak więc elementy wektora  $s^{ij} = [s_1^{ij} \ s_2^{ij} \ \dots \ s_D^{ij}]$  są elementami odpowiedzi  $i$ -tego wyjścia na skok jednostkowy na  $j$ -tym wejściu, przy pozostałych wejściach ustalonych.  $D$  jest horyzontem dynamiki wspólnym dla wszystkich odpowiedzi, tzn. można przyjąć  $s_l^{ij} = \text{const.}$  dla  $l \geq D$ .

Przedstawiony opis obiektu wielowymiarowego w postaci  $n_y \cdot n_u$  wektorów skończonych odpowiedzi skokowych  $[s_1^{ij} \ s_2^{ij} \ \dots \ s_D^{ij}]$ ,  $i = 1, 2, \dots, n_y$ ,  $j = 1, 2, \dots, n_u$  jest naturalny. Naturalnym bowiem i podstawowym w praktyce sposobem identyfikacji odpowiedzi skokowych jest dokonywanie skokowej zmiany jednego sterowania (przy pozostałych ustalonych) i rejestrowanie odpowiedzi wyjść, kolejno dla wszystkich sterowań,  $j = 1, 2, \dots, n_u$ . Zaprezentowany opis w postaci zbioru wektorów odpowiedzi skokowych korzystnie jest jednakże przekształcić do nieco innej, równoważnej postaci, co

umożliwi przejrzysty zapis modelu obiektu wielowymiarowego dla potrzeb algorytmu DMC. Zdefiniujemy w tym celu następujące macierze

$$\mathbf{S}_l = \begin{bmatrix} s_l^{11} & s_l^{12} & s_l^{13} & \cdots & s_l^{1n_u} \\ s_l^{21} & s_l^{22} & s_l^{23} & \cdots & s_l^{2n_u} \\ s_l^{31} & s_l^{32} & s_l^{33} & \cdots & s_l^{3n_u} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_l^{n_y 1} & s_l^{n_y 2} & s_l^{n_y 3} & \cdots & s_l^{n_y n_u} \end{bmatrix}, \quad l = 1, 2, \dots, D. \quad (3.18)$$

Każda z macierzy  $\mathbf{S}_l$  składa się z odpowiadających chwili  $l$  współczynników  $s_l^{ij}$  wszystkich odpowiedzi skokowych,  $i = 1, 2, \dots, n_y$ ,  $j = 1, 2, \dots, n_u$ . Tak więc obiekt może być reprezentowany przez  $D$  macierzy  $\mathbf{S}_l$  o wymiarze  $n_y \times n_u$ , zamiast przez  $n_y \cdot n_u$  wektorów o wymiarze  $D$ .

Zestaw macierzy  $\mathbf{S}_l$  można traktować jako *wielowymiarową (macierzową) odpowiedź skokową*  $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_D\}$ . Wówczas wszystkie uzyskane dotychczas dla obiektu SISO wzory *bezpośrednio przenoszą się na przypadek obiektu MIMO*, jedynie zamiast każdego ze skalarnych współczynników  $s_l$  pojedynczej odpowiedzi skokowej należy podstawić macierz  $\mathbf{S}_l$ ,  $l = 1, 2, \dots, D$ . Podstawowy dla modelowania odpowiedzią skokową wzór (3.11) przyjmuje w ten sposób postać

$$y(k) = y(0) + \sum_{j=1}^k \mathbf{S}_j \Delta u(k-j), \quad (3.19)$$

gdzie zmienne wyjściowe i sterowania obiektu są teraz wektorami,  $y(k) \in \mathbb{R}^{n_y}$ ,  $\Delta u(k-j) \in \mathbb{R}^{n_u}$ . Analogicznie, zależność opisująca wyjścia przewidywane jest następująca

$$y(k+p|k) = \sum_{j=1}^p \mathbf{S}_j \Delta u(k+p-j|k) + y(k) + \sum_{j=1}^{D-1} (\mathbf{S}_{j+p} - \mathbf{S}_j) \Delta u(k-j), \quad (3.20)$$

stąd wzory na elementy trajektorii swobodnej i wymuszonej wyjść przewidywanych są postaci

$$\Delta y(k+p|k) = \sum_{j=1}^p \mathbf{S}_j \Delta u(k+p-j|k), \quad p = 1, 2, \dots, N, \quad (3.21)$$

$$y^0(k+p|k) = y(k) + \sum_{j=1}^{D-1} (\mathbf{S}_{j+p} - \mathbf{S}_j) \Delta u(k-j), \quad p = 1, 2, \dots, N. \quad (3.22)$$

### 3.2.2. Algorytm DMC w wersji analitycznej

Zdefiniujmy wektory

$$\mathcal{Y}^{zad}(k) = \begin{bmatrix} y^{zad}(k + N_1|k) \\ \vdots \\ y^{zad}(k + N|k) \end{bmatrix}, \quad \mathcal{Y}^0(k) = \begin{bmatrix} y^0(k + N_1|k) \\ \vdots \\ y^0(k + N|k) \end{bmatrix},$$

$$\Delta\mathcal{Y}(k) = \begin{bmatrix} \Delta y(k + N_1|k) \\ \vdots \\ \Delta y(k + N|k) \end{bmatrix}, \quad \Delta\mathcal{U}(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k + N_u - 1|k) \end{bmatrix},$$

$$\mathcal{Y}^{pred}(k) = \mathcal{Y}^0(k) + \Delta\mathcal{Y}(k) = [y(k + N_1|k)^T \cdots y(k + N|k)^T]^T,$$

oraz macierze

$$\underline{\Psi} = \begin{bmatrix} \Psi(1) & 0 & \cdots & 0 \\ 0 & \Psi(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Psi(N) \end{bmatrix}, \quad (3.23)$$

$$\underline{\Lambda} = \begin{bmatrix} \Lambda(0) & 0 & \cdots & 0 \\ 0 & \Lambda(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Lambda(N_u - 1) \end{bmatrix}. \quad (3.24)$$

Wówczas funkcję kryterialną (3.3) można zapisać w postaci

$$J(k) = \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] - \Delta\mathcal{Y}(k) \right\|_{\underline{\Psi}}^2 + \|\Delta\mathcal{U}(k)\|_{\underline{\Lambda}}^2. \quad (3.25)$$

Jeśli natomiast dodatkowo  $\underline{\Psi} = \mathbf{I}$  oraz  $\underline{\Lambda} = \lambda\mathbf{I}$ , tak jak we wzorze (3.4), to funkcja kryterialna przyjmuje postać

$$J(k) = \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] - \Delta\mathcal{Y}(k) \right\|^2 + \lambda \|\Delta\mathcal{U}(k)\|^2 \quad (3.26)$$

Zdefiniujmy ponadto wektory

$$\mathcal{Y}(k) = \begin{bmatrix} y(k) \\ \vdots \\ y(k) \end{bmatrix}, \quad \Delta\mathcal{U}^P(k) = \begin{bmatrix} \Delta u(k-1) \\ \vdots \\ \Delta u(k-(D-1)) \end{bmatrix},$$

gdzie  $\dim \mathcal{Y}(k) = n_{\mathcal{Y}} = n_y \cdot (N - N_1 + 1)$ ,  $\dim \Delta\mathcal{U}^P(k) = n_u \cdot (D - 1)$ .

Załóżmy na wstępie, dla większej przejrzystości wzorów, że *obiekt regulacji jest jednowymiarowy (SISO)*,  $n_y = n_u = 1$ . Wektory  $\mathcal{Y}^0(k)$ ,  $\mathcal{Y}(k)$  i  $\Delta\mathcal{Y}(k)$  są wówczas o wymiarze  $n_y = N - (N_1 - 1)$ . Wektory te w algorytmie DMC wyznaczone są w oparciu o model procesu w postaci skończonej odpowiedzi skokowej  $\{s_l, l = 1, 2, \dots, D, s_l = s_D \text{ dla } l \geq D\}$ . Oznacza to, że rozważane są obiekty asymptotycznie stabilne, bez całkowania. Wykorzystując wzór (3.16) możemy zależność opisującą *składową swobodną* trajektorii prognozowanej zapisać w postaci

$$\mathcal{Y}^0(k) = \mathcal{Y}(k) + \mathbf{M}^P \Delta\mathcal{U}^P(k), \quad (3.27)$$

gdzie macierz  $\mathbf{M}^P$  jest wymiaru  $n_y \times (D - 1)$  i dla  $N_1 = 1$  ma charakterystyczną postać

$$\mathbf{M}^P = \begin{bmatrix} s_2 - s_1 & s_3 - s_2 & s_4 - s_3 & \cdots & s_D - s_{D-1} \\ s_3 - s_1 & s_4 - s_2 & s_5 - s_3 & \cdots & s_{D+1} - s_{D-1} \\ s_4 - s_1 & s_5 - s_2 & s_6 - s_3 & \cdots & s_{D+2} - s_{D-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{N+1} - s_1 & s_{N+2} - s_2 & s_{N+3} - s_3 & \cdots & s_{N+D-1} - s_{D-1} \end{bmatrix}.$$

Postać ogólna dla  $N_1 > 1$  powstaje poprzez pominięcie w pokazanej powyżej macierzy pierwszych  $N_1 - 1$  wierszy, tzn.

$$\mathbf{M}^P = \begin{bmatrix} s_{1+N_1} - s_1 & s_{2+N_1} - s_2 & s_{3+N_1} - s_3 & \cdots & s_{D-1+N_1} - s_{D-1} \\ s_{2+N_1} - s_1 & s_{3+N_1} - s_2 & s_{4+N_1} - s_3 & \cdots & s_{D+N_1} - s_{D-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{N+1} - s_1 & s_{N+2} - s_2 & s_{N+3} - s_3 & \cdots & s_{N+D-1} - s_{D-1} \end{bmatrix}. \quad (3.28)$$

Indeks górny „ $P$ ” macierzy wprowadzono dla zaznaczenia, że macierz ta wyznacza predykcje wyjść w zależności jedynie od przeszłych ( $P$  – „*Past*”) przyrostów sterowań. Oczywiście,  $s_l = s_\infty = k_m$  dla każdego elementu  $s_l$  o indeksie  $l \geq D$ . Na przykład, dla  $N = D = 6$  i  $N_1 = 2$  (co jest standardową wartością dla opóźnienia  $\tau = 1$ , tj. o czas  $T_p$ ) dostajemy

$$\mathbf{M}^P = \begin{bmatrix} s_3 - s_1 & s_4 - s_2 & s_5 - s_3 & k_m - s_4 & k_m - s_5 \\ s_4 - s_1 & s_5 - s_2 & k_m - s_3 & k_m - s_4 & k_m - s_5 \\ s_5 - s_1 & k_m - s_2 & k_m - s_3 & k_m - s_4 & k_m - s_5 \\ k_m - s_1 & k_m - s_2 & k_m - s_3 & k_m - s_4 & k_m - s_5 \\ k_m - s_1 & k_m - s_2 & k_m - s_3 & k_m - s_4 & k_m - s_5 \end{bmatrix}.$$

Wykorzystując z kolei wzór (3.14) możemy zapisać *składową wymuszaną* trajektorii prognozowanej w postaci

$$\Delta\mathcal{Y}(k) = \mathbf{M} \Delta\mathcal{U}(k), \quad (3.29)$$

gdzie macierz  $\mathbf{M}$  jest tzw. *macierzą dynamiczną* (*dynamic matrix*) o wymiarze  $n_y \times n_{\Delta U}$  ( $n_{\Delta U} = N_u$  dla obiektu jednowymiarowego), postaci

$$\mathbf{M} = \begin{bmatrix} s_{N_1} & s_{N_1-1} & \cdots & s_1 & 0 & \cdots & 0 \\ s_{N_1+1} & s_{N_1} & \cdots & s_2 & s_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{N_u} & s_{N_u-1} & \cdots & s_{N_u-N_1+1} & s_{N_u-N_1} & \cdots & s_1 \\ s_{N_u+1} & s_{N_u} & \cdots & s_{N_u-N_1+2} & s_{N_u-N_1+1} & \cdots & s_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_N & s_{N-1} & \cdots & s_{N-N_1+1} & s_{N-N_1} & \cdots & s_{N-N_u+1} \end{bmatrix}. \quad (3.30)$$

Dla standardowej wartości  $N_1 = \tau + 1$ , gdy pierwsze  $\tau = N_1 - 1$  współczynników odpowiedzi skokowej jest zerami, macierz  $\mathbf{M}$  przyjmuje charakterystyczną postać

$$\mathbf{M} = \begin{bmatrix} s_{N_1} & 0 & 0 & 0 & \cdots & 0 \\ s_{N_1+1} & s_{N_1} & 0 & 0 & \cdots & 0 \\ s_{N_1+2} & s_{N_1+1} & s_{N_1} & 0 & \cdots & 0 \\ s_{N_1+3} & s_{N_1+2} & s_{N_1+1} & s_{N_1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{N_u+N_1-1} & s_{N_u+N_1-2} & s_{N_u+N_1-3} & s_{N_u+N_1-4} & \cdots & s_{N_1} \\ s_{N_u+N_1} & s_{N_u+N_1-1} & s_{N_u+N_1-2} & s_{N_u+N_1-3} & \cdots & s_{N_1+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_N & s_{N-1} & s_{N-2} & s_{N-3} & \cdots & s_{N-N_u+1} \end{bmatrix}. \quad (3.31)$$

Na przykład, dla  $N = 6$ ,  $N_u = 3$  i  $\tau = 1$ , a stąd  $N_1 = 2$  mamy

$$\mathbf{M} = \begin{bmatrix} s_2 & 0 & 0 \\ s_3 & s_2 & 0 \\ s_4 & s_3 & s_2 \\ s_5 & s_4 & s_3 \\ s_6 & s_5 & s_4 \end{bmatrix}.$$

Przejdźmy teraz do ogólnego przypadku *obektu regulacji wielowymiarowego* (MIMO). Wykorzystując model w postaci wielowymiarowej, macierzowej odpowiedzi skokowej  $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_D\}$  uzyskujemy odpowiedniki wzorów (3.27) i (3.29) w takiej samej postaci,

$$\mathcal{Y}^0(k) = \mathcal{Y}(k) + \mathbf{M}^P \Delta \mathcal{U}^P(k), \quad (3.32)$$

$$\Delta \mathcal{Y}(k) = \mathbf{M} \Delta \mathcal{U}(k), \quad (3.33)$$

jedynie teraz w definicjach macierzy  $\mathbf{M}^P$  i  $\mathbf{M}$  wystąpią macierze  $\mathbf{S}_j$  w miejscach skalarów  $s_j$ . Tak więc mamy

$$\mathbf{M}^P = \begin{bmatrix} \mathbf{S}_{1+N_1} - \mathbf{S}_1 & \mathbf{S}_{2+N_1} - \mathbf{S}_2 & \cdots & \mathbf{S}_{D-1+N_1} - \mathbf{S}_{D-1} \\ \mathbf{S}_{2+N_1} - \mathbf{S}_1 & \mathbf{S}_{3+N_1} - \mathbf{S}_2 & \cdots & \mathbf{S}_{D+N_1} - \mathbf{S}_{D-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{N+1} - \mathbf{S}_1 & \mathbf{S}_{N+2} - \mathbf{S}_2 & \cdots & \mathbf{S}_{N+D-1} - \mathbf{S}_{D-1} \end{bmatrix}, \quad (3.34)$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}_{N_1} & \mathbf{S}_{N_1-1} & \cdots & \mathbf{S}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{S}_{N_1+1} & \mathbf{S}_{N_1} & \cdots & \mathbf{S}_2 & \mathbf{S}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{N_u} & \mathbf{S}_{N_u-1} & \cdots & \mathbf{S}_{N_u-N_1+1} & \mathbf{S}_{N_u-N_1} & \cdots & \mathbf{S}_1 \\ \mathbf{S}_{N_u+1} & \mathbf{S}_{N_u} & \cdots & \mathbf{S}_{N_u-N_1+2} & \mathbf{S}_{N_u-N_1+1} & \cdots & \mathbf{S}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_N & \mathbf{S}_{N-1} & \cdots & \mathbf{S}_{N-N_1+1} & \mathbf{S}_{N-N_1} & \cdots & \mathbf{S}_{N-N_u+1} \end{bmatrix}. \quad (3.35)$$

Zgodnie z definicjami wektorów  $\mathcal{Y}^0(k)$ ,  $\Delta\mathcal{Y}(k)$ ,  $\Delta\mathcal{U}^P(k)$  i  $\Delta\mathcal{U}(k)$  macierz  $\mathbf{M}^P$  ma wymiar  $n_y \times n_{\Delta\mathcal{U}^P} = n_y(N - N_1 + 1) \times n_u(D - 1)$ , zaś macierz dynamiczna  $\mathbf{M}$  wymiar  $n_y \times n_{\Delta\mathcal{U}} = n_y(N - N_1 + 1) \times n_u N_u$ .

Wykorzystując macierz  $\mathbf{M}$ , funkcję kryterialną (3.25) możemy zapisać w postaci

$$J(k) = \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] - \mathbf{M}\Delta\mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \|\Delta\mathcal{U}(k)\|_{\underline{\Lambda}}^2. \quad (3.36)$$

Przy poczynionych założeniach dotyczących macierzy wagowych jest to funkcja ściśle wypukła. Warunkiem minimum bez ograniczeń jest zerowanie się jej gradientu, tj.

$$\begin{aligned} -\mathbf{M}^T \underline{\Psi} [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k) - \mathbf{M}\Delta\mathcal{U}(k)] + \underline{\Lambda}\Delta\mathcal{U}(k) &= \mathbf{0}, \\ [\mathbf{M}^T \underline{\Psi} \mathbf{M} + \underline{\Lambda}] \Delta\mathcal{U}(k) - \mathbf{M}^T \underline{\Psi} [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] &= \mathbf{0}. \end{aligned}$$

Dostajemy stąd *wektor optymalnych przyrostów sterowań*  $\Delta\hat{\mathcal{U}}(k)$

$$\begin{aligned} \Delta\hat{\mathcal{U}}(k) &= [\mathbf{M}^T \underline{\Psi} \mathbf{M} + \underline{\Lambda}]^{-1} \mathbf{M}^T \underline{\Psi} [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] \\ &= \mathbf{K} [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)], \end{aligned} \quad (3.37)$$

gdzie

$$\mathbf{K} = [\mathbf{M}^T \underline{\Psi} \mathbf{M} + \underline{\Lambda}]^{-1} \mathbf{M}^T \underline{\Psi} \quad (3.38)$$

jest macierzą wymiaru  $n_{\Delta\mathcal{U}} \times n_y$ . Dla przypadku  $\underline{\Psi} = \mathbf{I}$  i  $\underline{\Lambda} = \lambda \mathbf{I}$  ma ona postać

$$\mathbf{K} = [\mathbf{M}^T \mathbf{M} + \lambda \mathbf{I}]^{-1} \mathbf{M}^T. \quad (3.39)$$

Oznaczmy

$$\mathbf{K} = \begin{bmatrix} \overline{\mathbf{K}}_1 \\ \overline{\mathbf{K}}_2 \\ \vdots \\ \overline{\mathbf{K}}_{N_u} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} & \cdots & \mathbf{K}_{1,N-N_1+1} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} & \cdots & \mathbf{K}_{2,N-N_1+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{N_u,1} & \mathbf{K}_{N_u,2} & \cdots & \mathbf{K}_{N_u,N-N_1+1} \end{bmatrix}, \quad (3.40)$$

gdzie każda z podmacierzy  $\overline{\mathbf{K}}_i$  jest wymiaru  $n_u \times n_y = n_u \times (N - N_1 + 1)n_y$ , a każda z podmacierzy  $\mathbf{K}_{ij}$  jest wymiaru  $n_u \times n_y$ .

W algorytmie regulacji predykcyjnej wykorzystuje się do sterowania obiektem jedynie przyrosty sterowań dla bieżącej chwili  $k$ , tzn. wyznaczoną optymalną wartość  $\Delta \hat{u}(k|k)$  złożoną z pierwszych  $n_u$  elementów wektora optymalnych przyrostów sterowań (3.37). Oznaczmy tę wartość przez  $\Delta \hat{u}(k)$ ,

$$\Delta \hat{u}(k) = \Delta \hat{u}(k|k) = \overline{\mathbf{K}}_1 [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)]. \quad (3.41)$$

Tak więc uzyskane *prawo regulacji jest liniowym sprzężeniem zwrotnym od trajektorii uchybu regulacji, definiowanego jako różnica trajektorii zadanej i składowej swobodnej trajektorii prognozowanej* (tzn. bez sterowania).

Przedstawimy prawo regulacji DMC w postaci zależnej bezpośrednio od znanych wyjść i sterowań obiektu. Wykorzystując zależność (3.32) mamy

$$\Delta \hat{u}(k) = \overline{\mathbf{K}}_1 [\mathcal{Y}^{zad}(k) - \mathcal{Y}(k) - \mathbf{M}^P \Delta \mathcal{U}^P(k)].$$

Przedstawiając macierz  $\mathbf{M}^P$  jako

$$\mathbf{M}^P = \begin{bmatrix} \mathbf{M}_1^P & \mathbf{M}_2^P & \cdots & \mathbf{M}_{D-1}^P \end{bmatrix},$$

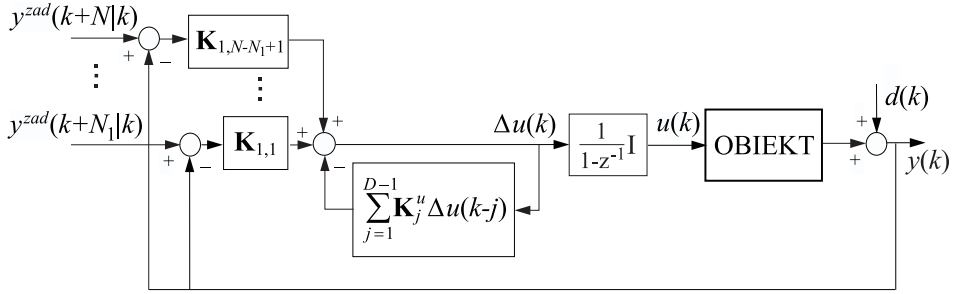
gdzie każda z podmacierzy  $\mathbf{M}_j^P$  jest wymiaru  $n_y \times n_u = n_y(N - N_1 + 1) \times n_u$ , oraz wykorzystując strukturę (3.40) macierzy  $\mathbf{K}$  możemy napisać

$$\begin{aligned} \Delta \hat{u}(k) &= \overline{\mathbf{K}}_1 [\mathcal{Y}^{zad}(k) - \mathcal{Y}(k)] - \sum_{j=1}^{D-1} (\overline{\mathbf{K}}_1 \mathbf{M}_j^P) \Delta u(k-j) \\ &= \sum_{p=N_1}^N \mathbf{K}_{1,p-N_1+1} [y^{zad}(k+p|k) - y(k)] + \\ &\quad - \sum_{j=1}^{D-1} \mathbf{K}_j^u \Delta u(k-j), \end{aligned} \quad (3.42)$$

gdzie

$$\mathbf{K}_j^u = \overline{\mathbf{K}}_1 \mathbf{M}_j^P, \quad j = 1, 2, \dots, D-1.$$

Wzór (3.42) przedstawia strukturę regulatora predykcyjnego DMC wyznaczonego *analytycznie* w przypadku braku ograniczeń nierównościowych, czy



Rysunek 3.5. Struktura regulatora DMC w wersji analitycznej (bez ograniczeń)

też świadomego ich nie uwzględniania w trakcie wyznaczania prawa regulacji. Stąd będziemy tu mówić o *regulatorze w wersji analitycznej*. Struktura takiego regulatora zilustrowana jest schematycznie na rysunku 3.5, gdzie blok oznaczony diagonalną macierzą transmitancji dyskretnych  $\frac{1}{1-z^{-1}}\mathbf{I}$  realizuje dyskretne całkowanie – sumowanie kolejnych przyrostów wektora sterowań w celu przejścia z sygnałów przyrostów na sygnały wartości sterowań. Natomiast blok wewnętrznej pętli sprzężenia zwrotnego zasilany kolejnymi wartościami przyrostów sterowania  $\Delta u(k)$  należy traktować jako realizujący wszystko to, co jest potrzebne do wykonywania wpisanego tam wzoru, tzn. również pamiętający  $D - 1$  ostatnio zrealizowanych przyrostów sterowań.

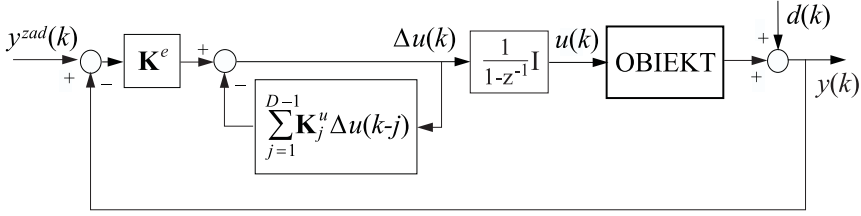
Zwróćmy uwagę, że regulator DMC analityczny jest *regulatorem liniowym*, zaś macierze jego współczynników  $\mathbf{K}_{1,p}$ ,  $p = 1, \dots, N - N_1 + 1$  i  $\mathbf{K}_j^u$ ,  $j = 1, \dots, D - 1$  wyznaczamy tylko raz, w fazie obliczania regulatora (*off-line*).

W regulacji obiektów przemysłowych zwykle występuje sytuacja, gdy nie znamy, stąd też nie zakładamy, zmian wartości zadanych na horyzoncie predykcji w momencie wyznaczania sterowania, tzn. przyjmujemy w każdej z kolejnych chwil próbkowania

$$y^{zad}(k + N_1|k) = y^{zad}(k + N_1 + 1|k) = \dots = y^{zad}(k + N|k) = y^{zad}(k).$$

Wówczas prawo regulacji (3.42) upraszcza się do postaci

$$\begin{aligned} \Delta \hat{u}(k) &= \sum_{p=N_1}^N \mathbf{K}_{1,p-N_1+1} [y^{zad}(k) - y(k)] - \sum_{j=1}^{D-1} \mathbf{K}_j^u \Delta u(k-j) \\ &= \mathbf{K}^e [y^{zad}(k) - y(k)] - \sum_{j=1}^{D-1} \mathbf{K}_j^u \Delta u(k-j), \end{aligned} \quad (3.43)$$



Rysunek 3.6. Struktura analitycznej wersji regulatora DMC (bez ograniczeń) z predykcją wartości zadanych równą wartościom aktualnym  $y^{zad}(k)$

gdzie

$$\mathbf{K}^e = \sum_{p=N_1}^N \mathbf{K}_{1,p-N_1+1}. \quad (3.44)$$

Struktura prawa regulacji (3.43) zilustrowana jest na rysunku 3.6.

Przepiszmy prawo regulacji (3.43) w formie

$$\Delta u(k) + \sum_{j=1}^{D-1} \mathbf{K}_j^u z^{-j} \Delta u(k) = \mathbf{K}^e e(k),$$

gdzie  $e(k) = y^{zad}(k) - y(k)$ , zaś  $z^{-1}$  oznacza operator jednostkowego przesunięcia w czasie (opóźnienia). Wzór powyższy zapiszmy dalej w postaci

$$\Delta u(k) = [\mathbf{I} + \sum_{j=1}^{D-1} \mathbf{K}_j^u z^{-j}]^{-1} \mathbf{K}^e e(k).$$

W przypadku układu SISO macierze  $\mathbf{K}^e$  i  $\mathbf{K}_j^u$  są skalarami  $k^e$  i  $k_j^u$ , a używany wzór przyjmuje postać opisującą dyskretny, jednowymiarowy, szeregowy regulator przyrostowy

$$\frac{\Delta u(k)}{e(k)} = \frac{k^e}{1 + k_1^u z^{-1} + k_2^u z^{-2} + \dots + k_{D-1}^u z^{-(D-1)}}. \quad (3.45)$$

Dla konkretnego zastosowania, dysponując dyskretnym liniowym modelem obiektu i zaprojektowanym dla niego regulatorem DMC w postaci (3.45), można badać własności zamkniętego układu regulacji, takie jak położenie biegunów, zapasy stabilności itp. – w zależności od wartości parametrów dostrojonych regulatora. Daje to istotny wgląd we własności regulatora DMC, przede wszystkim w rozważanej sytuacji bez ograniczeń. Ma też znaczenie dla ważniejszych praktycznie przypadków z ograniczeniami, choć wówczas układ regulacji staje się nieliniowym i jego bezpośrednia analiza wskazanym sposobem nie jest możliwa.

**Przykład 3.1.**

Rozważymy prosty obiekt jednowymiarowy (SISO) opisany modelem w postaci odpowiedzi skokowej

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$
0	0	0.2	0.5	0.6	0.62

(3.46)

gdzie  $D = 6$ . Przyjmijmy  $N = 6$ ,  $N_u = 3$ , oraz  $N_1 = 3$  ze względu na opóźnienie o dwa okresy próbkowania widoczne w odpowiedzi skokowej.

Przybliżenie wyjść zależna od przeszłych sterowań,  $\mathcal{Y}^0(k) = \mathcal{Y}(k) + \mathbf{M}^P \Delta \mathcal{U}^P(k)$ , jest postaci

$$\begin{bmatrix} y^0(k+3|k) \\ y^0(k+4|k) \\ y^0(k+5|k) \\ y^0(k+6|k) \end{bmatrix} = \begin{bmatrix} y(k) \\ y(k) \\ y(k) \\ y(k) \end{bmatrix} + \mathbf{M}^P \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \Delta u(k-3) \\ \Delta u(k-4) \\ \Delta u(k-5) \end{bmatrix},$$

gdzie

$$\begin{aligned} \mathbf{M}^P &= \begin{bmatrix} 0.5 & 0.6 & 0.42 & 0.12 & 0.02 \\ 0.6 & 0.62 & 0.42 & 0.12 & 0.02 \\ 0.62 & 0.62 & 0.42 & 0.12 & 0.02 \\ 0.62 & 0.62 & 0.42 & 0.12 & 0.02 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{M}_1^P & \mathbf{M}_2^P & \mathbf{M}_3^P & \mathbf{M}_4^P & \mathbf{M}_5^P \end{bmatrix}. \end{aligned}$$

Macierz dynamiczna ma natomiast postać

$$\mathbf{M} = \begin{bmatrix} 0.2 & 0 & 0 \\ 0.5 & 0.2 & 0 \\ 0.6 & 0.5 & 0.2 \\ 0.62 & 0.6 & 0.5 \end{bmatrix}. \quad (3.47)$$

Przyjmijmy  $\underline{\Psi} = \mathbf{I}$  oraz  $\underline{\Lambda} = \lambda \mathbf{I}$ , wówczas

$$\mathbf{M}^T \mathbf{M} + \lambda \mathbf{I} = \begin{bmatrix} 1.0344 + \lambda & 0.772 & 0.43 \\ 0.772 & 0.65 + \lambda & 0.4 \\ 0.43 & 0.4 & 0.29 + \lambda \end{bmatrix}.$$

Stąd, dla  $\lambda = 0.01$  dostajemy

$$\begin{aligned} \mathbf{K} &= (\mathbf{M}^T \mathbf{M} + \lambda \mathbf{I})^{-1} \mathbf{M}^T \\ &= \begin{bmatrix} 1.7171 & 1.5996 & -0.4519 & 0.0678 \\ -2.6932 & -0.9300 & 2.5508 & -0.6326 \\ 1.1297 & -1.0529 & -2.0868 & 2.4130 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{K}}_1 \\ \bar{\mathbf{K}}_2 \\ \bar{\mathbf{K}}_3 \end{bmatrix}. \end{aligned}$$

Przyjmując  $y^{zad}(k+3|k) = \dots = y^{zad}(k+6|k) = y^{zad}(k)$  otrzymujemy prawo regulacji w postaci (3.43)

$$\begin{aligned}\Delta \hat{u}(k) &= \left( \sum_{p=3}^6 k_{1,p-2} \right) (y^{zad}(k) - y(k)) - \sum_{j=1}^5 (\bar{\mathbf{K}}_1 \mathbf{M}_j^P) \Delta u(k-j) \\ &= k^e (y^{zad}(k) - y(k)) - \sum_{j=1}^5 k_j^u \Delta u(k-j),\end{aligned}$$

gdzie

$$k^e = 2.9327, \quad \mathbf{k}^u = \begin{bmatrix} 1.5802 & 1.7839 & 1.2317 & 0.3519 & 0.0587 \end{bmatrix}.$$

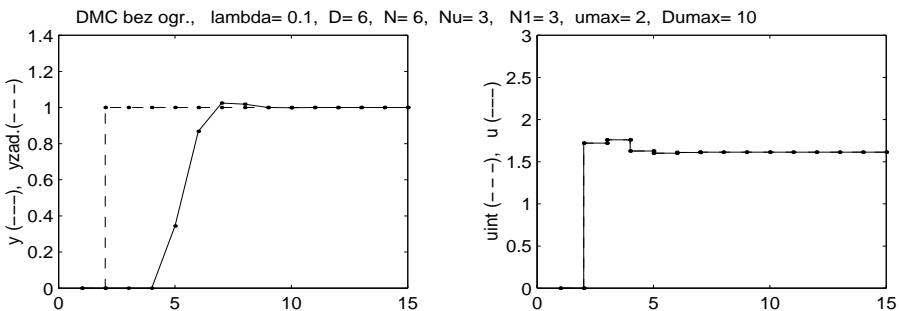
Dla  $\lambda = 0.05$  wartości parametrów regulatora wynoszą

$$k^e = 2.0787, \quad \mathbf{k}^u = \begin{bmatrix} 1.1622 & 1.2715 & 0.8730 & 0.2494 & 0.0416 \end{bmatrix},$$

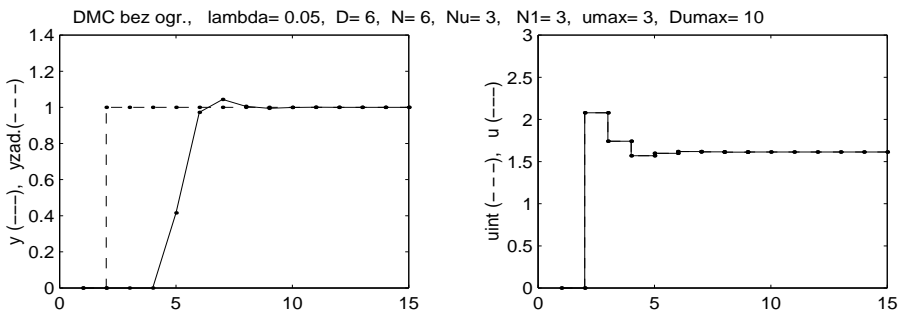
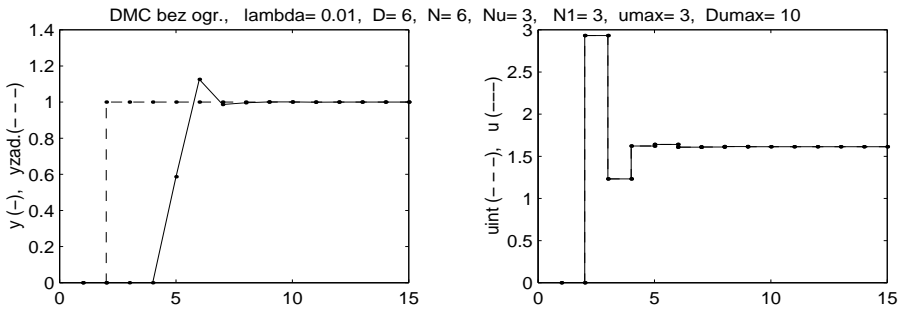
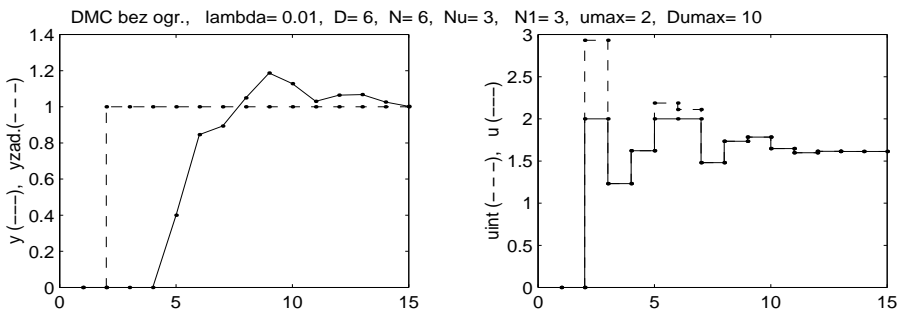
zaś dla  $\lambda = 0.1$

$$k^e = 1.7201, \quad \mathbf{k}^u = \begin{bmatrix} 0.9774 & 1.0546 & 0.7225 & 0.2064 & 0.0344 \end{bmatrix}.$$

Na rysunkach 3.7, 3.8, 3.9 i 3.10 przedstawiono przykładowe przebiegi zmiennych wyjściowej i sterującej w układzie regulacji z wyliczonym powyżej regulatorem DMC. Rysunki 3.7, 3.8 i 3.9 pokazują, jak zmniejszanie wartości  $\lambda$  wpływa na przyspieszanie przebiegu wyjścia, poprzez stosowanie coraz większych amplitud sygnału sterowania – i odwrotnie, zwiększanie wartości  $\lambda$  uspokaja przebieg sygnału sterowania spowalniając sygnał wyjścia. Ma to znaczenie, jeśli zbyt duże wartości sterowania są potem ograniczane przez nasycenie elementu wykonawczego, co może prowadzić do istotnego pogorszenia przebiegów – jeśli nie jest brane pod uwagę przy projektowaniu regulatora. Przypadek taki przedstawiono na rysunku 3.10,



Rysunek 3.7. Regulator analityczny, przebiegi bez ograniczeń sterowania,  $\lambda = 0.10$

Rysunek 3.8. Regulator analityczny, przebiegi bez ograniczeń sterowania,  $\lambda = 0.05$ Rysunek 3.9. Regulator analityczny, przebiegi bez ograniczeń sterowania,  $\lambda = 0.01$ Rysunek 3.10. Regulator analityczny, przebiegi z ograniczeniem amplitudy sterowania nie uwzględnionym w strukturze regulatora,  $\lambda = 0.01$

gdzie linią przerywaną oznaczono sterowanie wyznaczone przez regulator (zsumowane przyrosty sygnału sterowania), a linią ciągłą sterowanie realizowane w obiekcie po przejściu przez element ograniczający amplitudę do wartości  $u_{\max} = 2$  (porównaj z rys. 3.9). Poprzez ograniczenie zmian sterowania, regulatory z większymi wartościami  $\lambda$  są z reguły bardziej odporne na niedokładność modelowania, czy zmiany parametrów obiektu w trakcie pracy układu regulacji.  $\square$

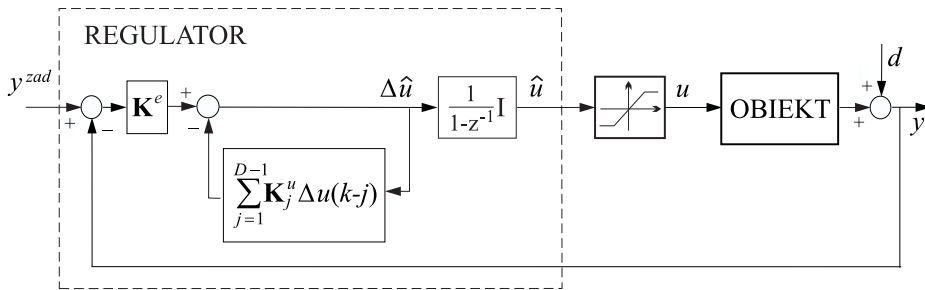
### 3.2.3. Uwzględnianie ograniczeń, algorytm w wersji numerycznej

#### Ograniczenie sygnału sterującego przez rzutowanie (przycięcie)

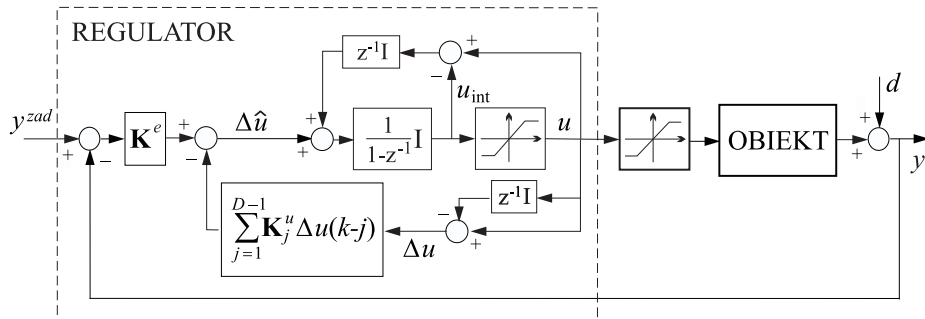
W realnych układach regulacji zawsze istnieją ograniczenia sygnałów sterujących wynikające z możliwości fizycznych elementów wykonawczych. Ograniczeń tych przekroczyć nie można – zwane są dlatego *twardymi* (*hard constraints*). Ponadto, występować mogą ograniczenia sygnałów wyjściowych obiektu – które są na ogół natury technologicznej i fizycznie mogą być naruszane, choć grozi to określonymi konsekwencjami, jak np. uszkodzeniami aparatury czy nieodpowiednimi parametrami produktów. Dlatego też ograniczenia te traktowane mogą być na ogół jako *miękkie* (*soft constraints*), czyli takie, których chwilowe naruszenie jest możliwe. O ograniczeniach zarówno sterowań, jak i wyjść regulowanych wspomniano w rozdziale 3.1 omawiając zasady regulacji predykcyjnej, zob. (3.6), (3.7), (3.8), kwestia poruszona jest też w rozdziale 3.6.2.

Klasyczne regulatory, przede wszystkim najbardziej rozpowszechnione w zastosowaniach przemysłowych regulatory typu PID, projektuje się (stroi się) nie uwzględniając ograniczeń. Ścisłe rzecz biorąc, regulator projektuje się jedynie do pracy przy małych wartościach sygnałów, w dostatecznie małym otoczeniu punktu pracy. Pomimo tego, w wielu sytuacjach praktycznych regulatory PID pracują zupełnie zadowolająco również w sytuacji przekraczania przez wyznaczone przez nie sygnały sterujące ograniczeń amplitudy czy prędkości zmiany, jeśli tylko struktura prawa regulacji PID uzupełniona jest układem ograniczenia nasycenia integratora (tzw. struktura *anti-windup*) wykorzystującym model nieliniowości elementu wykonawczego lub bezpośrednio sygnał z jego wyjścia, zob. np. [54].

Wyprowadzając w rozdziale poprzednim analityczne prawo sterowania regulatora predykcyjnego DMC, postaci (3.42) czy (3.43), pominęliśmy ograniczenia – tylko w takiej sytuacji prawo takie można w ogólności analitycznie wyprowadzić. Powstaje pytanie, jak pracowałby układ regulacji z regulatorem DMC analitycznym w sytuacji generacji sygnału sterującego



Rysunek 3.11. Struktura układu z regulatorem analitycznym DMC nie wykorzystująca informacji o ograniczeniach sygnałów sterujących – *struktura niepoprawna*



Rysunek 3.12. Struktura układu z regulatorem analitycznym DMC wykorzystująca informacje o ograniczeniach sygnałów do korekty przyrostów sterowania, z dodatkową pętlą *anti-windup* (górna pętla sprzężenia zwrotnego)

$\Delta \hat{u}(k)$  o większych amplitudach czy prędkościach zmiany niż możliwe do przeniesienia przez element wykonawczy, tzn. sygnału potencjalnie naruszającego fizyczne ograniczenia i stąd nierealizowalnego. Okazuje się, że jeśli strukturę regulatora analitycznego DMC odpowiednio uzupełnić wykorzystując model ograniczeń sterowania, to regulator taki w wielu sytuacjach będzie zadowalająco, czy wręcz dobrze pracował, szczególnie dla obiektów jednowymiarowych [93].

Na rysunku 3.11 pokazano niewłaściwe zastosowanie regulatora analitycznego DMC w sytuacji dysponowania informacją o ograniczeniach amplitudy sygnału sterującego. Błędem, i to o istotnych konsekwencjach dla jakości regulacji, jest zasilanie bloku wewnętrznego sprzężenia zwrotnego algorytmu regulatora poprzednimi przyrostami sterowania wyznaczanymi przez prawo regulacji, tzn. wartościami  $\Delta \hat{u}(k-j)$  wynikającymi ze wzoru (3.42), a nie przyrostami  $\Delta u(k-j)$  odpowiadającymi sterowaniu  $u(k-j)$ ,

które po przejściu przez charakterystykę nasycenia elementu wykonawczego realnie oddziaływało na obiekt. Efekty zastosowania takiej błędnej struktury pokazano na rysunku 3.10, przy prezentacji przykładu 3.1. Poprawne wykorzystanie poprzednich przyrostów sterowań realizuje struktura pokazana na rysunku 3.12, gdzie  $z^{-1}\mathbf{I}$  oznacza jednostkowe (o okres próbkowania) opóźnienie sygnałów wektora sterowań, zaś blok nieliniowy też należy rozumieć wektorowo – jako wektor  $n_u$  ograniczeń amplitudy poszczególnych składowych wektora sterowań. W strukturze tej zastosowano ponadto dodatkową pętlę sprzężenia zwrotnego ograniczającą całkowanie (korygującą stan integratora), w sposób podobny do stosowanego w strukturze ograniczenia całkowania (*anti-windup*) regulatora PID, zob. np. [3, 54]. Zastosowanie dodatkowej pętli *anti-windup* wprowadza *całkowitz korekcję stanu integratora*, z opóźnieniem jednego kroku – w chwili  $k$ -tej sygnał wyjściowy regulatora  $\Delta\hat{u}(k)$  dodawany jest do sygnału  $u(k-1)$  sterującego obiektem w chwili poprzedniej, tzn. sygnału  $u_{\text{int}}(k-1)$  po przejściu przez blok ograniczenia amplitudy. Algorytmiczna realizacja przedstawionej struktury, dla przypadku sterowania skalarnego, podana jest poniżej:

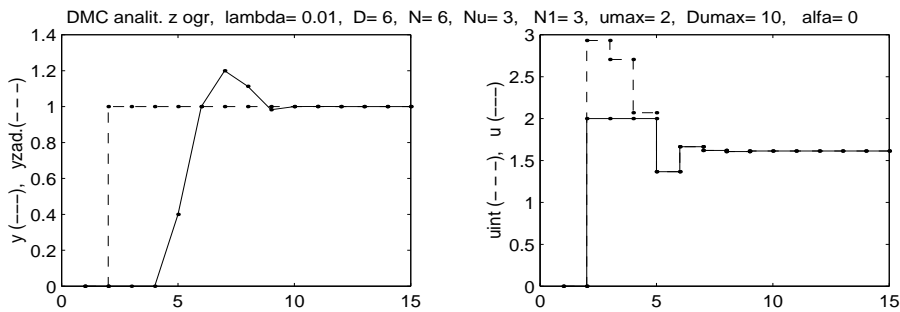
$$\begin{aligned}
 \Delta\hat{u}(k) &= k^e(y^{\text{zad}}(k) - y(k)) - \sum_{j=1}^{D-1} k_j^u \Delta u(k-j) \\
 u_{\text{int}}(k) &= u(k-1) + \Delta\hat{u}(k) \\
 \text{if } u_{\text{int}}(k) &\geq u_{\text{max}} \\
 \quad u(k) &= u_{\text{max}} \\
 \text{else } u(k) &= \max\{u_{\text{int}}(k), u_{\text{min}}\} \\
 \text{end} \\
 \Delta u(k) &= u(k) - u(k-1)
 \end{aligned} \tag{3.48}$$

Funkcjonalnie, algorytm powyższy realizuje *rzutowanie* sterowania na zbiór dopuszczalny

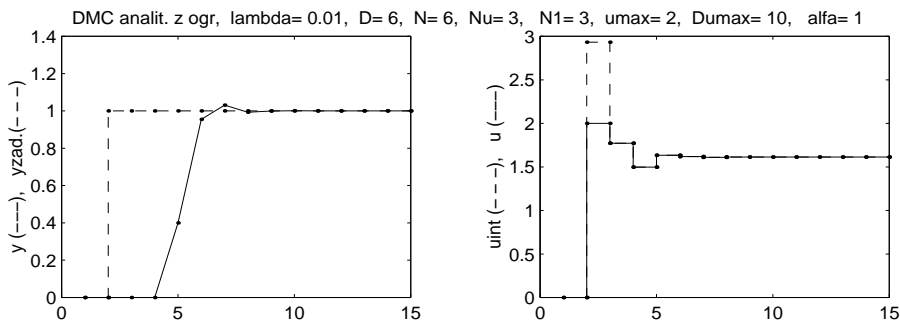
$$u_{\text{min}} \leq u(k) \leq u_{\text{max}}. \tag{3.49}$$

Omówiony sposób uwzględniania ograniczenia amplitudy określany też bywa terminem *przycinanie* (*clipping*) sygnału sterowania.

W strukturze na rysunku 3.12 i odpowiadającym jej algorytmie (3.48) wartości poprzednich przyrostów sterowania  $\Delta u(k-j)$  wykorzystywane w algorytmie DMC są tymi, które zostały zmodyfikowane w sposób odpowiadający przejściu oryginalnie wyznaczonych sygnałów sterowania przez ograniczenia elementu wykonawczego. Stąd, zachowano tu ogólną zasadę konstrukcji struktur regulacji uwzględniających ograniczenia sterowań: *sygnały sterowania oddziaływujące zwrotnie na stan regulatora powinny być sygnałami realnie wykonanych, ograniczonych sterowań*, zob. [54]. Na rysunku 3.13 pokazano przebiegi dla regulatora analitycznego z



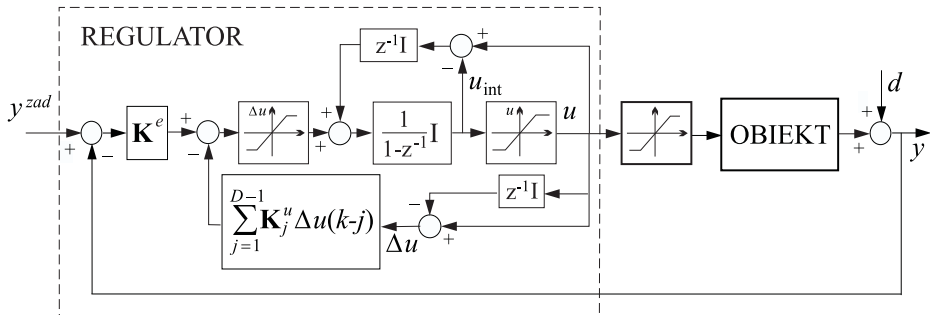
Rysunek 3.13. Przebiegi z regulatorem analitycznym z korektami przyrostów sterowania uwzględniającymi ograniczenia amplitudy, ale bez pętli *anti-windup*



Rysunek 3.14. Przebiegi z regulatorem analitycznym z korektami przyrostów sterowania uwzględniającymi ograniczenia amplitudy i z korektą *anti-windup*

przykładu 3.1 implementowanego w strukturze z rysunku 3.12, ale bez pętli sprzężenia zwrotnego *anti-windup* ograniczającego stan integratora. Natomiast na rysunku 3.14 pokazano analogiczne przebiegi dla pełnej struktury z rysunku 3.12, tj. również przy włączonym sprzężeniu korekcyjnym *anti-windup*. Porównanie z przebiegami z rysunku 3.10 pokazuje istotne polepszenie nawet bez pętli *anti-windup*, włączenie tej pętli dodatkowo znacznie poprawia działanie regulatora.

Oprócz ograniczeń wartości (3.49) występują również ograniczenia szybkości zmiany sygnałów sterujących. Ponieważ algorytm DMC operuje właśnie na przyrostach sterowań, to ograniczenia te jest łatwo uwzględnić. Na rysunku 3.15 pokazano strukturę z uwzględnieniem zarówno nasycień sygnałów sterujących, jak i szybkości ich zmian – te ostatnie są realizowane przez blok nasycenia umieszczony przed integratorem, przy czym wartości nasycień tego bloku są wartościami granicznymi nierówności ograniczają-

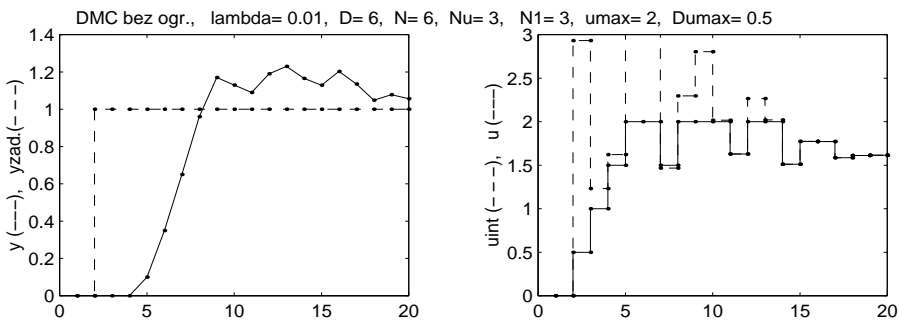


Rysunek 3.15. Regulator analityczny DMC z wykorzystaniem informacji o nasyceniach i ograniczeniach szybkości zmian sygnałów sterujących

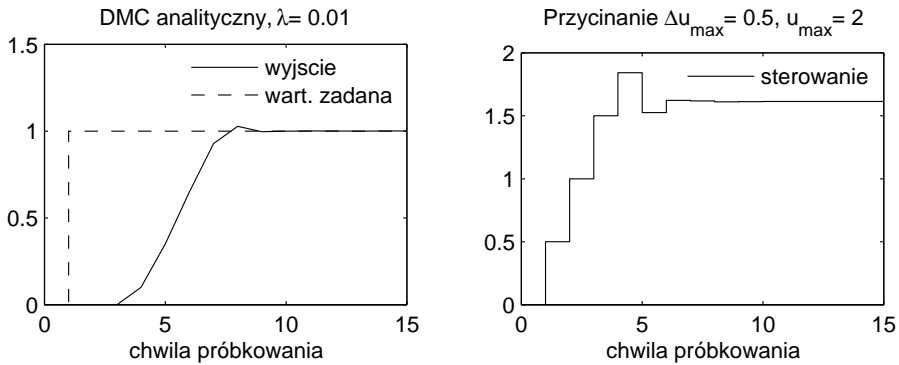
cych

$$-\Delta u_{\max} \leq \Delta u(k) \leq \Delta u_{\max}.$$

Na rysunku 3.16 przedstawiono przebiegi wyjścia i sterowania uzyskane dla obiektu z przykładu 3.1, w układzie regulacji przedstawionym na rysunku 3.11, tzn. z regulatorem analitycznym DMC nie wykorzystującym informacji o ograniczeniach, w sytuacji elementu wykonawczego wprowadzającego aktywne ograniczenia amplitudy (nasycenie) i przyrostu sterowania. Natomiast na rysunku 3.17 przedstawiono przebiegi uzyskane w identycznych warunkach, ale z uwzględniającym ograniczenia regulatorem o strukturze z rysunku 3.15. Warto porównać te przebiegi z przedstawionymi na rysunkach 3.13 i 3.14, w stosunku do których dodano jedynie ograniczenie przyrostu sterowania  $\Delta u_{\max} = 0.5$ .



Rysunek 3.16. Przebiegi w układzie z regulatorem analitycznym, z ograniczeniami amplitudy ( $\leq 2$ ) i przyrostu ( $\leq 0.5$ ) sterowania nie uwzględnionymi w strukturze regulatora



Rysunek 3.17. Przebiegi w układzie z regulatorem analitycznym, z ograniczeniami amplitudy i przyrostu sterowania uwzględnionymi w strukturze regulatora, z dodatkową korektą *anti-windup*

### Algorytm DMC w wersji numerycznej

Uwzględnianie ograniczeń sterowania przedstawione w poprzednim punkcie jest heurystyczne, w ogólności suboptymalne, chociaż często bliskie optymalnemu, szczególnie w przypadku jednowymiarowym. Ponadto, trudniejsze jest uwzględnianie w ten sposób ograniczeń nałożonych na wyjścia (regulowane i niergulowane) – należy wówczas, wykorzystując model obiektu, dokonać przekształcenia wartości ograniczeń wyjść na równoważne wartości ograniczeń sterowań, zob. [95].

W rozdziale 3.1 przedstawiliśmy ogólną zasadę regulacji predykcyjnej. Centralnym jej elementem jest wykorzystywanie w każdym kroku sterowania uzyskiwanego z rozwiązania zadania optymalizacji z ograniczeniami (3.167). Przy liniowym modelu i braku ograniczeń zadanie to ma rozwiązanie analityczne, co bardzo upraszcza realizację i implementację algorytmu, tak jak to pokazaliśmy dla algorytmu DMC. Jednakże, zaletą algorytmu realizowanego z wykorzystaniem w każdym kroku numerycznego rozwiązania zadania optymalizacji z ograniczeniami jest możliwość dokładnego, optymalnego uwzględniania ograniczeń sterowań i wyjść. Jeśli model procesu jest liniowy i ograniczenia też są liniowe, to zadanie optymalizacji jest zadaniem *programowania kwadratowego*. Dla takich zadań istnieją efektywne i niezawodne procedury optymalizacji, umożliwiające efektywną realizację algorytmu predykcyjnego.

W algorytmie DMC model obiektu dany jest w postaci odpowiedzi skokowych, funkcję kryterialną dla takiego modelu sformułowaliśmy już w postaci (3.36). Należy jeszcze odpowiednio przeformułować ograniczenia, jako funkcje wektora zmiennych decyzyjnych  $\Delta U(k)$ . Najłatwiejsze jest to dla

ograniczeń szybkości narastania sterowań

$$-\Delta u_{\max} \leq \Delta u(k+p|k) \leq \Delta u_{\max}, \quad p = 0, 1, \dots, N_u - 1.$$

Definiując wektor  $\Delta \mathcal{U}_{\max}$  o wymiarowości  $n_u \cdot N_u$ ,

$$\Delta \mathcal{U}_{\max} = \begin{bmatrix} \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix},$$

możemy ograniczenia szybkości zmiany zapisać w postaci

$$-\Delta \mathcal{U}_{\max} \leq \Delta \mathcal{U}(k) \leq \Delta \mathcal{U}_{\max}.$$

Sytuacja jest nieco trudniejsza dla ograniczeń wartości sterowań

$$u_{\min} \leq u(k+p|k) \leq u_{\max}, \quad p = 0, 1, \dots, N_u - 1.$$

Po zdefiniowaniu następujących wektorów granic, o wymiarze  $n_u \cdot N_u$ ,

$$\mathcal{U}_{\min} = \begin{bmatrix} u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix}, \quad \mathcal{U}_{\max} = \begin{bmatrix} u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix},$$

należy jeszcze wyrazić wartości sterowań poprzez ich przyrosty. Wykorzystując zależność

$$u(k+p|k) = u(k-1) + \sum_{j=0}^p \Delta u(k+j|k), \quad p = 0, 1, \dots, N_u - 1,$$

możemy ograniczenia wartości sterowań zapisać w postaci

$$\mathcal{U}_{\min} \leq \mathcal{U}(k-1) + \mathbf{J} \Delta \mathcal{U}(k) \leq \mathcal{U}_{\max},$$

gdzie

$$\mathbf{J} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix},$$

a każda z macierzy jednostkowych  $\mathbf{I}$  jest o wymiarze  $n_u$ . Na koniec należy jeszcze odpowiednio zapisać ograniczenia wyjść (3.8)

$$y_{\min} \leq y(k+p|k) \leq y_{\max}, \quad p = N_1, N_1 + 1, \dots, N. \quad (3.50)$$

Definiując, analogicznie do wektorów  $\mathcal{U}_{\min}$  i  $\mathcal{U}_{\max}$ , wektory  $\mathcal{Y}_{\min}$  i  $\mathcal{Y}_{\max}$  o wymiarze  $n_y \cdot (N - N_1 + 1)$  każdy i wykorzystując zależność (3.33), ograniczenia wyjść można zapisać w postaci

$$\mathcal{Y}_{\min} \leq \mathcal{Y}^0(k) + \mathbf{M}\Delta\mathcal{U}(k) \leq \mathcal{Y}_{\max},$$

gdzie  $\mathcal{Y}^0(k)$  dane jest wzorem (3.32).

Reasumując, zadanie optymalizacji kwadratowej, rozwiązywane w każdym kroku algorytmu DMC w wersji numerycznej, można zapisać w postaci:

$$\begin{aligned} \min_{\Delta\mathcal{U}(k)} & \left\{ \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}(k) - \mathbf{M}^P \Delta\mathcal{U}^P(k)] - \mathbf{M}\Delta\mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \|\Delta\mathcal{U}(k)\|_{\underline{\Lambda}}^2 \right\} \\ \text{z ogr. : } & -\Delta\mathcal{U}_{\max} \leq \Delta\mathcal{U}(k) \leq \Delta\mathcal{U}_{\max}, \\ & \mathcal{U}_{\min} \leq \mathcal{U}(k-1) + \mathbf{J}\Delta\mathcal{U}(k) \leq \mathcal{U}_{\max}, \\ & \mathcal{Y}_{\min} \leq \mathcal{Y}^0(k) + \mathbf{M}\Delta\mathcal{U}(k) \leq \mathcal{Y}_{\max}. \end{aligned} \quad (3.51)$$

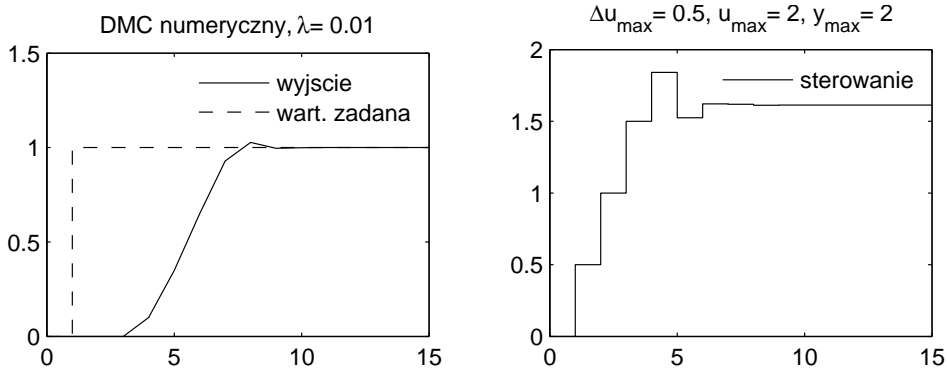
Zadanie (3.51) można łatwo zapisać w równoważnej postaci standardowej dla zadania programowania kwadratowego, np. w wymaganej przez procedurę QUADPROG (*Quadratic Programming*) pakietu MATLAB formie

$$\begin{aligned} \min & \left\{ J(x) = \frac{1}{2}x^T \mathbf{H}x + f^T x \right\} \\ \text{z ogr. : } & x_{\min} \leq x \leq x_{\max}, \\ & \mathbf{A}x \leq b. \end{aligned} \quad (3.52)$$

Łatwo bowiem sprawdzić, że zadania (3.51) i (3.52) są równoważne, jeśli

$$\begin{aligned} x &= \Delta\mathcal{U}(k), \quad x_{\min} = -\Delta\mathcal{U}_{\max}, \quad x_{\max} = \Delta\mathcal{U}_{\max}, \\ \mathbf{H} &= 2(\mathbf{M}^T \underline{\Psi} \mathbf{M} + \underline{\Lambda}), \\ f &= -2\mathbf{M}^T \underline{\Psi} (\mathcal{Y}^{zad}(k) - \mathcal{Y}(k) - \mathbf{M}^P \Delta\mathcal{U}^P(k)), \\ \mathbf{A} &= \begin{bmatrix} -\mathbf{J} \\ \mathbf{J} \\ -\mathbf{M} \\ \mathbf{M} \end{bmatrix}, \quad b = \begin{bmatrix} -\mathcal{U}_{\min} + \mathcal{U}(k-1) \\ \mathcal{U}_{\max} - \mathcal{U}(k-1) \\ -\mathcal{Y}_{\min} + \mathcal{Y}^0(k) \\ \mathcal{Y}_{\max} - \mathcal{Y}^0(k) \end{bmatrix}. \end{aligned}$$

Aktualnie, algorytmy numeryczne z grupy metod zbiorów aktywnych (*active set methods*) oraz z grupy metod punktu wewnętrznego (*interior*



Rysunek 3.18. Przebiegi w układzie z regulatorem numerycznym, z ograniczeniami amplitudy i przyrostu sterowania

*point methods*) uważane są za najefektywniejsze w rozwiązywaniu zadania programowania kwadratowego, zob. np. [17, 1].

Regulator predykcyjny DMC numeryczny, wyliczający w każdym kroku przyrost sterowania na drodze rozwiązania przedstawionego zadania programowania kwadratowego, zastosowano również do regulacji obiektu przedstawionego w przykładzie 3.1, dla warunków pracy takich samych jak przy stosowaniu regulatora analitycznego w strukturze z rysunku 3.15. Na rysunku 3.18 przedstawiono przykładowe przebiegi, uzyskane przy  $\lambda = 0.01$  i przy ograniczeniach amplitudy i przyrostów sterowania, są one niemal identyczne z otrzymanymi dla regulatora analitycznego, przedstawionymi na rysunku 3.17. Podobnie było w innych przypadkach, stąd nie będziemy ich tu osobno przytaczać. Fakty te potwierdzają tezę, że przynajmniej w przypadku jednowymiarowym regulator analityczny w strukturze z rysunku 3.15 (z pętlą anti-windup) *działa dobrze*, najczęściej praktycznie optymalnie. Autorowi pracy nie jest znane porównanie, czy teoretyczna analiza, kiedy regulator analityczny z uwzględnieniem ograniczeń przez rzutowanie (przycinanie) pracuje jak optymalny regulator numeryczny.

Jak wspomnieliśmy, zadanie programowania kwadratowego (3.51) można dokładnie i efektywnie rozwiązywać – pod jednym wszakże warunkiem: że posiada ono rozwiązanie, czyli że zbiór dopuszczalny definiowany przez ograniczenia nierównościowe jest niepusty (ograniczenia są niesprzeczne). Problem ten poruszyliśmy już w rozdziale 3.1 omawiając sformułowane tam ogólne zadanie optymalizacji sterowań algorytmu regulacji predykcyjnej ze skończonym, przesuwającym horyzontem. Zadanie (3.51) jest szczególnym przypadkiem tego zadania i wszystkie uwagi poczynione w rozdziale 3.1 mają tu zastosowanie. Przypomnijmy, że zjawisko pustego

zbioru dopuszczalnego grozi jedynie wówczas, gdy występują ograniczenia wartości wyjść (3.50) oraz że ograniczenia te mają z reguły charakter ograniczeń „miękkich”. Zagadnieniu zapewniania niepustości zbioru rozwiązań poświęcony jest pierwszy punkt rozdziału 3.6.2.

### 3.2.4. Niepewność modelu, zakłócenia

Model liniowy użyty do prognozowania wartości zmiennych wyjściowych w algorytmie DMC jest pewnym przybliżeniem zachowania się wyjść realnego obiektu, z reguły ze względu zarówno na niedokładność *strukturalną* (założenie liniowości), jak i *parametryczną* (niedokładne wartości współczynników). Ponadto, na obiekt na ogół oddziałują *wielkości wejściowe niesterowane*, czyli *zakłócenia*. Można wśród nich wyróżnić zakłócenia *mierzone* i *niemierzone* (*niemierzalne*).

#### Niepewność, zakłócenia niemierzalne

Typowym sposobem postępowania jest traktowanie błędów prognozowania wynikających z niedokładności modelowania i wpływu zakłóceń niemierzalnych łącznie, jako efektu oddziaływania pewnych zakłóceń niemierzalnych. W algorytmie DMC zakłócenia te traktuje się jako addytywne, sprowadzone do wyjścia obiektu. W bieżącej chwili  $k$  efekt ich oddziaływania  $d(k)$  oblicza się jako różnicę między wartościami wyjść aktualnie zmierzonymi i prognozowanymi modelem na chwilę bieżącą w chwili poprzedniej,  $(k - 1)$ -szej, tak jak to omówiliśmy w rozdziale 3.2.1 definiując model zakłóceń typu DMC. Model ten polega na przyjęciu prognozy zakłóceń niemierzalnych w postaci *wartości stałej na całym horyzoncie predykcji*, równej wartości  $d(k)$  wyliczonej w chwili bieżącej.

Model zakłóceń typu DMC sprawdza się dobrze zarówno dla zakłóceń skokowo, ale rzadko zmiennych, jak i wolnozmiennych, a więc w sytuacjach typowych w regulacji przemysłowych procesów technologicznych. Co więcej, zapewnia on *zerowe uchyby ustalone* (jeśli tylko odpowiednie sterowania są dopuszczalne) – tak jak klasyczne regulatory z *całkowaniem*, PI czy PID. Fakt ten łatwo wyjaśnić przyjmując, że układ regulacji z regulatorem DMC jest asymptotycznie stabilny i istnieje sterowanie dopuszczalne  $u_{zad}$  takie, że

$$y^{zad} - \mathbf{K}_{ob}u_{zad} = 0,$$

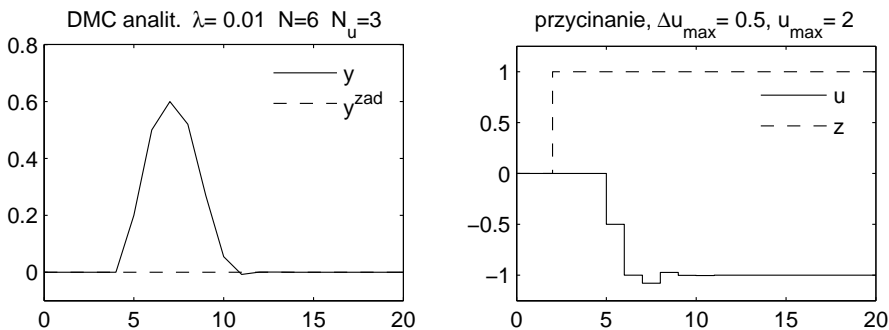
gdzie  $y^{zad}$  oznacza ustalone wartości zadane wyjść regulowanych, a  $\mathbf{K}_{ob}$  macierz wzmocnień statycznych obiektu. Przy ustalonych wartościach zadanych i zakłóceniach asymptotycznie stabilny układ regulacji osiąga stan ustalony. Oznaczmy sterowania i wyjścia obiektu w tym stanie przez  $u_{\infty}$

i  $y_\infty$ ,  $y_\infty = \mathbf{K}_{ob}u_\infty$ . W stanie ustalonym wszystkie, tzn. przeszłe (na horyzoncie dynamiki obiektu) i przyszłe optymalne (na horyzoncie sterowania) przyrosty sterowań są zerowe, tak więc po osiągnięciu stanu ustalonego w każdej chwili  $k$  wartość optymalna funkcji kryterialnej w zadaniu (3.51) wynosi

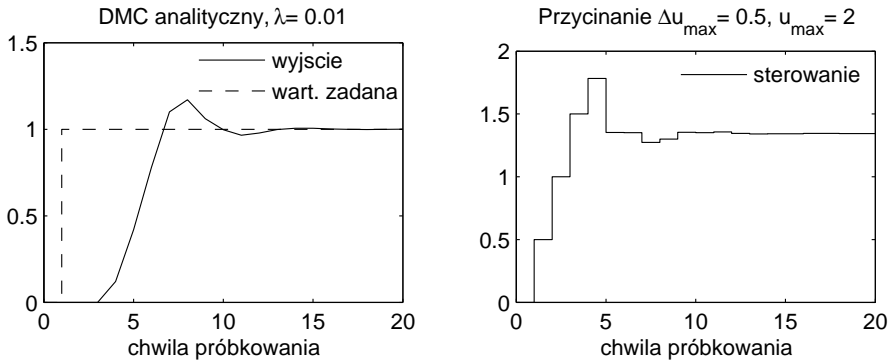
$$J(k)_{\text{opt}} = \left\| \mathcal{Y}^{zad}(k) - \mathcal{Y}(k) \right\|_{\underline{\Psi}}^2 = \sum_{p=N_1}^N \left\| y^{zad} - y_\infty \right\|_{\Psi(p)}^2.$$

Stąd w stanie ustalonym musi być  $y_\infty = y^{zad}$ , gdyż w przeciwnym razie w każdej chwili  $k$  istniałyby niezerowe przyrosty zmieniające wartości sterowań  $u_\infty$  w kierunku  $u_{zad}$  i zmniejszające wartość funkcji kryterialnej (albowiem z założenia  $y^{zad} = \mathbf{K}_{ob}u_{zad}$ ), co przeczyłoby oczywiście optymalności zerowych przyrostów sterowań w stanie ustalonym.

Na dotychczasowych rysunkach przedstawiających przebiegi w układzie regulacji DMC dla obiektu z przykładu 3.1 widać, że uchyb regulacji po skokowej zmianie wartości zadanej od zera do jedności dąży do zera. Na rysunku 3.19 przedstawiono przebiegi dla tego samego obiektu pokazujące tłumienie zakłócenia, przy skokowej zmianie zakłócenia niemierzalnego na wejściu obiektu w chwili symulacji  $k = 2$ , od wartości zero do wartości 1 utrzymującej się do końca horyzontu symulacji (skok jednostkowy  $z(k)$  w chwili  $k = 2$ , wartość zadana cały czas zerowa). Na rysunku 3.20 przedstawiono natomiast sytuację jedynie ze skokiem jednostkowym wartości zadanej, ale dodatkowo zwiększono wzmocnienie w torze sterowania obiektu o 20%, pozostawiając model nie zmieniony. Tak więc regulator musiał poradzić sobie z błędami modelowania. Przebiegi są, jak się można było spo-



Rysunek 3.19. Odpowiedź na skok jednostkowy zakłócenia niemierzalnego  $z(k)$  na wejściu obiektu w chwili  $k=2$ , w układzie z regulatorem analitycznym uwzględniającym ograniczenia



Rysunek 3.20. Wyniki symulacji w z regulatorem analitycznym uwzględniającym ograniczenia, przy wzmacnieniu obiektu zwiększonym o 20%

dziewać, pogorszone, z większym przeregulowaniem (spowodowanym przez wzrost wzmacnienia w pętli sprzężenia zwrotnego) – ale niewiele, regulator nadal skutecznie wyzerowuje uchyb regulacji.

### Kompensacja zakłóceń mierzonych

Jak pamiętamy, przy projektowaniu układów sterowania obowiązuje zasada, że *wpływ zakłóceń mierzalnych*, tzn. wejść niesterowanych, ale możliwych do mierzenia, *powinien być kompensowany w otwartej strukturze sterowania* – jeśli tylko wpływ tych zakłóceń na wielkości regulowane jest istotny, dynamika torów zakłócenia i sterowania umożliwia skuteczną realizację kompensacji oraz dysponujemy odpowiednio dokładnymi modelami. Sterowanie w układzie otwartym jest bowiem wówczas szybsze i efektywniejsze niż w strukturze zamkniętej. Dla sterowania w pętli sprzężenia zwrotnego (regulacji) pozostaje wówczas redukcja wpływu pozostałych zakłóceń oraz niedokładności kompensacji w torze otwartym.

Zasada powyższa powinna być oczywiście przestrzegana również w regulacji predykcyjnej. Dla kompensacji mierzonych zmiennych zakłócających w układzie otwartym potrzebny jest model wpływu tych zmiennych na wyjścia regulowane. W algorytmie DMC model ten tworzymy w postaci takiej samej jak podstawowy model obiektu „sterowania–wyjścia”, tzn. w postaci odpowiedzi skokowych. Oznaczmy przez  $z$  wektor mierzonych zmiennych zakłócających, o wymiarowości  $n_z = \dim z$ , a przez  $\{\mathbf{S}_l^z, l = 1, 2, 3, \dots\}$  ciąg macierzowych odpowiedzi skokowych wyjść obiektu na skoki wartości zmiennych  $z$ , analogicznie jak przez  $\{\mathbf{S}_l, l = 1, 2, 3, \dots\}$  oznaczyliśmy ciąg macierzowych odpowiedzi skokowych wyjść obiektu na skoki sterowań  $u$  w rozdziale 3.2.1. Zgodnie z zasadą superpozycji, poszukiwany model

obiekту możemy wówczas zapisać jako rozszerzenie zależności (3.19)

$$y(k) = y(0) + \sum_{j=1}^k \mathbf{S}_j \Delta u(k-j) + \sum_{j=1}^k \mathbf{S}_j^z \Delta z(k-j), \quad (3.53)$$

gdzie  $\Delta z(k-j) = z(k-j) - z(k-j-1)$ ,  $j = 1, \dots, k$ .

Wynikające ze wzoru (3.53) równania prognozy w zależności jedynie od wejść zakłócających (przy zerowych przyrostach sterowań) wyprowadza się identycznie jak dla wejść sterujących, zob. rozdz. 3.2.1. Dostajemy w ten sposób zależność

$$\begin{aligned} y(k+p|k) &= y(k) + \sum_{j=1}^{p-1} \mathbf{S}_j^z \Delta z(k+p-j|k) + \mathbf{S}_p^z \Delta z(k) + \\ &\quad + \sum_{j=p+1}^{D_z-1+p} \mathbf{S}_j^z \Delta z(k+p-j) - \sum_{j=1}^{D_z-1} \mathbf{S}_j^z \Delta z(k-j) \\ &= \sum_{j=1}^{p-1} \mathbf{S}_j^z \Delta z(k+p-j|k) + \\ &\quad + y(k) + \mathbf{S}_p^z \Delta z(k) + \sum_{j=1}^{D_z-1} (\mathbf{S}_{j+p}^z - \mathbf{S}_j^z) \Delta z(k-j), \end{aligned}$$

gdzie  $D_z$  oznacza horyzont dynamiki zakłóceń, tzn. dla  $j \geq D_z$  możemy przyjąć  $\mathbf{S}_j^z = \mathbf{S}_{D_z}^z$ . W uzyskanym wzorze pierwsza suma reprezentuje wpływ prognozowanych wartości zakłóceń mierzonych, natomiast pozostałe składniki wpływ wartości już zrealizowanych. Łącząc te składniki z zależnością (3.22) uzyskujemy wzór opisujący *składową swobodną* trajektorii przewidywanej wyjść regulowanych, zależną od *przeszłych* wartości sterowań oraz *przeszłych i aktualnej* wartości zakłóceń mierzonych,

$$\mathcal{Y}^0(k) = \mathcal{Y}(k) + \mathbf{M}^P \Delta \mathcal{U}^P(k) + \mathbf{M}^{zP} \Delta \mathcal{Z}^P(k), \quad (3.54)$$

gdzie  $\Delta \mathcal{Z}^P(k)$  jest wektorem zrealizowanych przyrostów zakłóceń,

$$\Delta \mathcal{Z}^P(k) = \begin{bmatrix} \Delta z(k) \\ \Delta z(k-1) \\ \vdots \\ \Delta z(k - (D_z - 1)) \end{bmatrix},$$

zaś macierz  $\mathbf{M}^{zP}$  jest postaci

$$\mathbf{M}^{zP} = \begin{bmatrix} \mathbf{S}_{N_1}^z & \mathbf{S}_{1+N_1}^z - \mathbf{S}_1^z & \mathbf{S}_{2+N_1}^z - \mathbf{S}_2^z & \cdots & \mathbf{S}_{D_z-1+N_1}^z - \mathbf{S}_{D_z-1}^z \\ \mathbf{S}_{N_1+1}^z & \mathbf{S}_{2+N_1}^z - \mathbf{S}_1^z & \mathbf{S}_{3+N_1}^z - \mathbf{S}_2^z & \cdots & \mathbf{S}_{D_z+N_1}^z - \mathbf{S}_{D_z-1}^z \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_N^z & \mathbf{S}_{N+1}^z - \mathbf{S}_1^z & \mathbf{S}_{N+2}^z - \mathbf{S}_2^z & \cdots & \mathbf{S}_{N+D_z-1}^z - \mathbf{S}_{D_z-1}^z \end{bmatrix}. \quad (3.55)$$

Zauważmy, że  $\mathbf{M}^{zP}$  jest macierzą o strukturze analogicznej jak macierz  $\mathbf{M}^P$ , por. (3.34), ale rozszerzonej o pierwszą kolumnę (macierzową) odpowiadającą pierwszej kolumnie macierzy  $\mathbf{M}$ , por. (3.35). Wynika to z faktu, że w chwili  $k$  wektor  $\Delta z(k)$  przedstawia już zrealizowane (i zmierzone) wartości zakłóceń, podczas gdy  $\Delta u(k|k)$  należy do zmiennych decyzyjnych.

Na wartości wyjść w chwilach przyszłych mają wpływ, oprócz już zrealizowanych, *wyznaczane w chwili  $k$*  aktualna i przyszłe wartości sterowań  $u(k+p|k)$ ,  $p = 0, 1, \dots, N_u - 1$  oraz prognozowane wartości zakłóceń mierzonych  $z(k+p|k)$ ,  $p = 1, 2, \dots, N$ . Pierwsze z tych zmiennych to *zmiennne decyzyjne*, natomiast drugie należy odpowiednio przyjąć – dokonać ich *prognozy*. Najprostszy sposób to przyjęcie na horyzoncie predykcji wartości prognozowanych równych wartości bieżącej, czyli  $z(k+p|k) = z(k)$ ,  $p = 1, \dots, N-1$ , stąd  $\Delta z(k+p|k) = z(k+p|k) - z(k+p-1|k) = 0$ ,  $p = 1, \dots, N-1$ . Prognoza powyższa odpowiada modelowi zakłóceń zakładającemu brak informacji o ich zmienności w przyszłych chwilach.

Dysponując mechanizmem regulacji predykcyjnej i pomiarami zakłóceń można pokusić się o więcej, jeśli przebieg wartości zakłóceń mierzonych na przedziałach czasu odpowiadających horyzontowi predykcji (a szczególnie horyzontowi sterowania) jest istotnie zmienny, a przy tym o charakterze ciągłym, gładkim. Wówczas, wykorzystując zgromadzone dane, można przy projektowaniu regulatora predykcyjnego skonstruować *model prognostyczny zakłóceń*, np. prosty model o strukturze AR (autoregresyjnej) i parametrach wyznaczonych jedną ze standardowych metod identyfikacji. Zwróćmy uwagę, że najistotniejsza dla właściwej pracy regulatora predykcyjnego jest dokładność prognozy w pierwszych chwilach horyzontu predykcji, gdyż i tak w każdej z następujących chwil cały proces prognozowania i wyznaczania sterowania zostanie powtórzony. Stąd model AR może być stosunkowo niskiego rzędu. Co więcej, przy dłuższych horyzontach predykcji  $N$  sensowne może być wprowadzenie *horyzontu prognozy zakłóceń*  $N_z < N$ , tzn. przyjęcie  $z(k+p|k) = z(k+N_z|k)$ ,  $p = N_z + 1, \dots, N-1$ .

Oznaczając przez  $z(k+p|k)$ ,  $p = 1, \dots, N-1$ , wartości przyszłych zakłóceń prognozowane w chwili  $k$  na podstawie już zrealizowanych ich wartości  $z(k-j)$ ,  $j = 0, 1, \dots$  i z wykorzystaniem liniowego modelu ich dynamiki,

zdefiniujmy wektor

$$\Delta \mathcal{Z}(k) = \begin{bmatrix} \Delta z(k+1|k) \\ \vdots \\ \Delta z(k+N_z|k) \end{bmatrix}.$$

Można teraz zamiast (3.54) zastosować wzór określający przyszłe wartości wyjść regulowanych w zależności od zrealizowanych wartości sterowań i zakłóceń oraz *prognozowanych wartości zakłóceń*,

$$\mathcal{Y}^0(k) = \mathcal{Y}(k) + \mathbf{M}^P \Delta \mathcal{U}^P(k) + \mathbf{M}^{zP} \Delta \mathcal{Z}^P(k) + \mathbf{M}^z \Delta \mathcal{Z}(k), \quad (3.56)$$

gdzie macierz  $\mathbf{M}^z$ , o wymiarze  $n_y \times n_{\Delta \mathcal{Z}} = n_y(N - N_1 + 1) \times n_z N_z$ , jest postaci

$$\mathbf{M}^z = \begin{bmatrix} \mathbf{S}_{N_1-1}^z & \mathbf{S}_{N_1-2}^z & \cdots & \mathbf{S}_1^z & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{S}_{N_1}^z & \mathbf{S}_{N_1-1}^z & \cdots & \mathbf{S}_2^z & \mathbf{S}_1^z & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{N_z}^z & \mathbf{S}_{N_z-1}^z & \cdots & \mathbf{S}_{N_z-N_1+2}^z & \mathbf{S}_{N_z-N_1+1}^z & \cdots & \mathbf{S}_1^z \\ \mathbf{S}_{N_z+1}^z & \mathbf{S}_{N_z}^z & \cdots & \mathbf{S}_{N_z-N_1+3}^z & \mathbf{S}_{N_z-N_1+2}^z & \cdots & \mathbf{S}_2^z \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{N-1}^z & \mathbf{S}_{N-2}^z & \cdots & \mathbf{S}_{N-N_1+1}^z & \mathbf{S}_{N-N_1}^z & \cdots & \mathbf{S}_{N-N_z}^z \end{bmatrix}. \quad (3.57)$$

Macierz ta ma strukturę podobną do struktury macierzy dynamicznej  $\mathbf{M}$ , por. (3.35). Różnice wynikają stąd, że wektory  $\Delta \mathcal{Z}(k)$  i  $\Delta \mathcal{U}(k)$  są przesunięte względem siebie w czasie o takt: wektor  $\Delta \mathcal{Z}(k)$  składa się z  $N_z$  podwektorów  $\Delta z(k+p|k)$  indeksowanych przez  $p = 1, 2, \dots, N_z$ , podczas gdy wektor  $\Delta \mathcal{U}(k)$  składa się z  $N_u$  podwektorów  $\Delta u(k+p|k)$  indeksowanych przez  $p = 0, 1, \dots, N_u - 1$ . Stąd, w porównaniu do macierzy  $\mathbf{M}$ , macierz  $\mathbf{M}^z$  jest „przesunięta o kolumnę w prawo” – jej pierwsza kolumna odpowiada drugiej kolumnie macierzy  $\mathbf{M}$ , itd.

Oznaczmy przez  $\mathbf{Z}$  macierz reprezentującą liniowy *model prognozy przyszłych zakłóceń* w zależności od ich poprzednich i aktualnej wartości, tj.

$$\Delta \mathcal{Z}(k) = \mathbf{Z} \Delta \mathcal{Z}^P(k).$$

Możemy teraz przekształcić wzór (3.56) do postaci

$$\begin{aligned} \mathcal{Y}^0(k) &= \mathcal{Y}(k) + \mathbf{M}^P \Delta \mathcal{U}^P(k) + \mathbf{M}^{zP} \Delta \mathcal{Z}^P(k) + \mathbf{M}^z \mathbf{Z} \Delta \mathcal{Z}^P(k) \\ &= \mathcal{Y}(k) + \mathbf{M}^P \Delta \mathcal{U}^P(k) + [\mathbf{M}^{zP} + \mathbf{M}^z \mathbf{Z}] \Delta \mathcal{Z}^P(k) \\ &= \mathcal{Y}(k) + \mathbf{M}^P \Delta \mathcal{U}^P(k) + \mathbf{Z}^P \Delta \mathcal{Z}^P(k), \end{aligned} \quad (3.58)$$

gdzie

$$\mathbf{Z}^P = \mathbf{M}^z \mathbf{Z}^P + \mathbf{M}^z \mathbf{Z}. \quad (3.59)$$

Zależność (3.58) definiuje *składową swobodną prognozy wyjść regulowanych*, będącą rozszerzeniem zależności (3.32) na przypadek z zakłóceniami mierzonymi. Postępowanie prowadzące do wyprowadzenia analitycznego prawa regulacji DMC przedstawione w rozdziale 3.2.2 można teraz dokładnie powtórzyć, stosując jedynie zależność (3.58) zamiast wzoru (3.32). Dlatego też podajemy od razu wynik końcowy, odpowiadający wzorowi (3.41):

$$\Delta \hat{u}(k) = \bar{\mathbf{K}}_1 [\mathcal{Y}^{zad}(k) - \mathcal{Y}(k)] - \bar{\mathbf{K}}_1 \mathbf{M}^P \Delta \mathcal{U}^P(k) - \bar{\mathbf{K}}_1 \mathbf{Z}^P \Delta \mathcal{Z}^P(k). \quad (3.60)$$

Rozpisując macierz  $\mathbf{Z}^P$ , podobnie jak poprzednio macierz  $\mathbf{M}^P$ , jako

$$\mathbf{Z}^P = [\mathbf{Z}_0^P \ \mathbf{Z}_1^P \ \mathbf{Z}_2^P \ \dots \ \mathbf{Z}_{D_z-1}^P],$$

i wykorzystując przedstawioną w (3.40) strukturę macierzy  $\mathbf{K}$  dostajemy

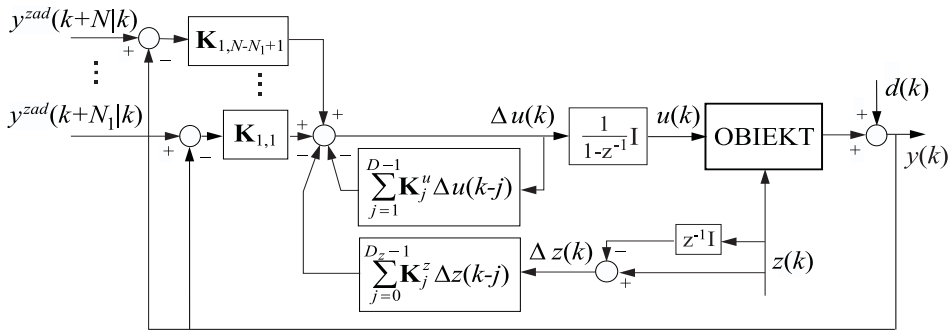
$$\begin{aligned} \Delta \hat{u}(k) &= \sum_{p=N_1}^N \mathbf{K}_{1,p-N_1+1} [y^{zad}(k+p|k) - y(k)] - \sum_{j=1}^{D-1} (\bar{\mathbf{K}}_1 \mathbf{M}_j^P) \Delta u(k-j) + \\ &\quad - \sum_{j=0}^{D_z-1} (\bar{\mathbf{K}}_1 \mathbf{Z}_j^P) \Delta z(k-j) \\ &= \sum_{p=N_1}^N \mathbf{K}_{1,p-N_1+1} [y^{zad}(k+p|k) - y(k)] - \sum_{j=1}^{D-1} \mathbf{K}_j^u \Delta u(k-j) + \\ &\quad - \sum_{j=0}^{D_z-1} \mathbf{K}_j^z \Delta z(k-j), \end{aligned} \quad (3.61)$$

gdzie

$$\mathbf{K}_j^z = \bar{\mathbf{K}}_1 \mathbf{Z}_j^P, \quad j = 0, 1, \dots, D_z - 1.$$

Wzór (3.61) przedstawia strukturę analitycznego regulatora predykcyjnego DMC, będącą rozszerzeniem struktury (3.42) na przypadek z kompensacją zakłóceń mierzonych. Struktura takiego regulatora zilustrowana jest schematycznie na rysunku 3.21 (porównaj z rysunkiem 3.5 i komentarzem przy nim zamieszczonym).

Stosując algorytm analityczny z kompensacją zakłóceń przy możliwości występowania ograniczeń sygnału sterującego, strukturę tę należy odpowiednio uzupełnić, analogicznie jak to przedstawiono na rysunkach 3.12 i 3.15 dla struktury bez kompensacji zakłóceń.



Rysunek 3.21. Struktura prawa regulacji DMC w wersji analitycznej, z kompensacją zakłóceń mierzonych (bez ograniczeń)

Przy zastosowaniu kompensacji zakłóceń mierzonych struktura zadania optymalizacji kwadratowej *numerycznego algorytmu DMC* nie ulega zmianie, jedynie należy rozszerzyć funkcję kryterialną do postaci

$$J(k) = \left\| [y^{zad}(k) - y(k) - \mathbf{M}^P \Delta \mathcal{U}^P(k) - \mathbf{Z}^P \Delta \mathcal{Z}^P(k)] - \mathbf{M} \Delta \mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \left\| \Delta \mathcal{U}(k) \right\|_{\underline{\Lambda}}^2,$$

gdzie macierz  $\mathbf{Z}^P$  dana jest zależnością (3.59). W przypadku braku modelu zmienności zakłóceń w granicach horyzontu predykcji macierz  $\mathbf{Z}^P$  redukuje się do macierzy  $\mathbf{M}^{z^P}$  (3.55).

### 3.3. Algorytm GPC (Generalized Predictive Control)

Pierwsze, komercyjne algorytmy MPC wykorzystywały model obiektu w postaci dyskretnej odpowiedzi impulsowej – algorytm *Model Algorithmic Control* (MAC) [123] i dyskretnej odpowiedzi skokowej – algorytm DMC. Ten ostatni przedstawiliśmy szczegółowo w poprzednim rozdziale. Algorytm predykcyjny wykorzystujący model w postaci odpowiedzi impulsowej jest bardzo podobny, odpowiedzi impulsowa i skokowa obiektu dynamicznego są przecież ściśle ze sobą powiązane. Zdecydowaliśmy się przedstawić algorytm DMC, gdyż odpowiedź skokowa jest opisem obiektu bardziej naturalnym, łatwym do bezpośredniego uzyskania – stąd popularność produktów komercyjnych opartych na tym właśnie algorytmie. Podejścia wykorzystującego model w postaci odpowiedzi impulsowej nie będziemy osobno prezentować. Jego znaczenie jest znacznie mniejsze, a wszystkie wzory oparte na odpo-

wiedzi skokowej można przekształcić we wzory wykorzystujące odpowiedź impulsową, korzystając ze znanej zależności

$$s_k = \sum_{j=0}^k h_j,$$

gdzie  $\{s_j\}$  i  $\{h_j\}$  są ciągami współczynników odpowiedzi skokowej i impulsowej.

Jednym z najpopularniejszych algorytmów predykcyjnych okazał się algorytm GPC (*Generalized Predictive Control*), zaproponowany nieco później niż algorytmy oparte na nieparametrycznych modelach odpowiedzi skokowej czy impulsowej. Algorytm GPC wykorzystuje model obiektu w postaci dyskretnego równania różnicowego opisującego relację wejście–wyjście (transmitancji dyskretniej) [26], zob. też [16, 17]. W rozdziale niniejszym przedstawimy algorytm predykcyjny GPC, odwołując się do ogólnych cech algorytmów MPC przedstawionych wcześniej przy prezentacji algorytmu DMC w rozdziale poprzednim.

W funkcji kryterialnej algorytmu GPC rozważa się często zastosowanie, w miejsce trajektorii wartości zadanych  $y^{zad}(k+p|k)$ ,  $p = N_1, \dots, N$ , trajektorii referencyjnej (*reference trajectory*)  $y^{ref}(k+p|k)$ ,  $p = N_1, \dots, N$ , zdefiniowanej zależnością

$$y^{ref}(k+p|k) = \gamma y^{ref}(k+p-1|k) + (1-\gamma)y^{zad}(k+p|k), \quad p = 1, \dots, N \quad (3.62)$$

przy czym  $y^{ref}(k|k) = y(k)$ , zob. [16, 17]. W zależności powyższej  $\gamma$ ,  $0 \leq \gamma < 1$ , jest parametrem określającym szybkość przybliżania się trajektorii referencyjnej, startującej z aktualnie zmierzonej wartości wyjścia obiektu  $y(k)$ , do trajektorii  $y^{zad}(k+p|k)$ ,  $p = 1, \dots, N$ . Dla  $\gamma = 0$  trajektoria referencyjna staje się trajektorią wartości zadanych, wraz ze wzrostem  $\gamma$  zbliżanie się trajektorii referencyjnej do wartości zadanych staje się powolniejsze, łagodniejsze – stawiając przez to mniejsze wymagania dla sygnału sterującego. Rolę i znaczenie trajektorii referencyjnej omówimy bliżej w rozdziale 3.6.2 w punkcie poświęconym doborowi wartości parametrów regulatorów predykcyjnych. Trajektorja referencyjna może być zastosowana w każdym algorytmie MPC jako pewien filtr różnicy między  $y(k)$  a  $y^{zad}(k+p|k)$ , występując w funkcji kryterialnej i wynikających z niej dalej wzorach dokładnie w miejscu wartości  $y^{zad}(k+p|k)$ . Mając powyższe stwierdzenie na uwadze, dalej we wzorach stosować będziemy, konsekwentnie, trajektorię wartości zadanych.

W algorytmie GPC wykorzystujemy model obiektu w postaci dyskretnego równania różnicowego. W ogólnym przypadku obiektu wielowymiarowego o  $n_u$  wejściach i  $n_y$  wyjściach jest to model postaci

$$\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})u(k-1) + \mathbf{C}(z^{-1})\frac{\varepsilon(k)}{\Delta}, \quad (3.63)$$

gdzie  $\mathbf{A}$ ,  $\mathbf{B}$  i  $\mathbf{C}$  są macierzami wielomianowymi

$$\begin{aligned} \mathbf{A}(z^{-1}) &= \mathbf{1} + \mathbf{A}_1 z^{-1} + \mathbf{A}_2 z^{-2} + \dots + \mathbf{A}_{n_A} z^{-n_A}, \\ \mathbf{B}(z^{-1}) &= \mathbf{B}_0 + \mathbf{B}_1 z^{-1} + \mathbf{B}_2 z^{-2} + \dots + \mathbf{B}_{n_B} z^{-n_B}, \\ \mathbf{C}(z^{-1}) &= \mathbf{1} + \mathbf{C}_1 z^{-1} + \mathbf{C}_2 z^{-2} + \dots + \mathbf{C}_{n_C} z^{-n_C}, \end{aligned}$$

$z^{-1}$  oznacza operator jednostkowego opóźnienia,  $\varepsilon(k)$  jest wektorem białych szumów o zerowej wartości średniej, zaś  $\Delta = 1 - z^{-1}$  oznacza operator różnicy wstecznej (czyli  $1/\Delta$  oznacza operację całkowania). Jeśli traktować  $z^{-1}$  jako zmienną zespoloną przekształcenia  $\mathcal{Z}$ , to można rozważać wykorzystywany model jako transmitancje dyskretne między wejściami a wyjściami obiektu.

Model (3.63) okreśłany jest jako model ARIMAX (*Auto-Regressive Integrated Moving Average with exogenous Input*), zob. np. [109], zwany jest on także modelem CARIMA (*Controlled Auto-Regressive Integrated Moving Average*), np. [17]. W modelu tym szum biały  $\varepsilon(k)$  podlega całkowaniu, stąd w efekcie na układ działają zakłócenia niestacjonarne. Jeśli  $\mathbf{C}(z^{-1}) = 1$ , to zakłócenia działające na obiekt są scałkowanymi szumami białymi, jeśli  $\mathbf{C}(z^{-1}) \neq 1$ , to są scałkowanymi szumami kolorowymi.

Dla  $\mathbf{C}(z^{-1}) = 1$  opis obiektu (3.63) przyjmuje następującą postać

$$\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})u(k-1) + \frac{\varepsilon(k)}{\Delta}. \quad (3.64)$$

Przypadek ten jest bardzo ważny z praktycznego punktu widzenia. Przede wszystkim dlatego, że identyfikacja wielomianu  $\mathbf{C}$  nastęrcza na ogół trudności: oceny są niezbędne lub zbiegają wolno [109]. Stąd też bywa on traktowany inaczej: nie jako część modelu, a jako filtr o parametrach podlegających strojeniu, dla polepszenia określonych cech regulatora [17, 87]. Ponadto, w przypadku  $\mathbf{C}(z^{-1}) = 1$  łatwiej wyprowadza się zależności opisujące predykcje wyjść i analityczne prawo regulacji GPC w sytuacji bez ograniczeń. Dlatego wielu autorów rozważa regulator GPC tylko dla tego przypadku. Podobnie postąpimy i w tej pracy, odsyłając czytelnika zainteresowanego modelem z szumem kolorowym przede wszystkim do książki [17], poświęconej głównie algorytmowi regulacji GPC, w wielu jego wersjach.

### 3.3.1. Algorytm GPC dla obiektu jednowymiarowego

Rozważmy na początku układ o jednym wejściu i jednym wyjściu (SISO). Ze względu na większą złożoność dokonywanych przekształceń niż w przypadku regulatora DMC, pozwoli to na łatwiejsze zrozumienie ich zasady. Uzyskane rezultaty rozszerzymy potem na przypadek wielowymiarowy.

W przypadku obiektu SISO model (3.64) redukuje się do równania

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + \frac{\varepsilon(k)}{\Delta}, \quad (3.65)$$

gdzie  $A(z^{-1})$ ,  $B(z^{-1})$  są wielomianami zmiennej  $z^{-1}$ , odpowiednio

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_A} z^{-n_A}, \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_B} z^{-n_B}. \end{aligned}$$

W modelu powyższym nie wydzielamy odrębnie opóźnienia  $\tau$ , poza opóźnieniem jednostkowym związanym z dyskretyzacją modelu obiektu jako układu przyczynowego. Jeśli w obiekcie (ciągłym) występuje opóźnienie, to odpowiadających mu pierwszych  $\tau$  współczynników  $b_j$  wielomianu  $B$  będzie równych zero.

Przy takim samym sformułowaniu postaci funkcji kryterialnej, różne liniowe algorytmy predykcyjne różnią się sposobami wyznaczania przewidywanych na horyzoncie predykcji wartości trajektorii wyjść  $y(k+p|k)$ ,  $p = 1, \dots, N$ , zob. rys. 3.2. W algorytmie GPC wartości te wyznacza się opierając się na modelu obiektu (3.65), zob. np. [16, 17]. Rozważmy w tym celu tożsamość Bézout (równanie diofantyczne z  $C(z^{-1}) = 1$ )

$$E_p(z^{-1})\bar{A}(z^{-1}) + F_p(z^{-1})z^{-p} = 1, \quad (3.66)$$

gdzie

$$\bar{A}(z^{-1}) = \Delta A(z^{-1}),$$

zaś wielomiany  $E_p$  i  $F_p$  są stopnia, odpowiednio,  $p-1$  i  $n_A$ . Można je uzyskać dzieląc 1 przez  $\bar{A}(z^{-1})$  aż do momentu, kiedy reszta ma postać  $F_p(z^{-1})z^{-p}$ , zob. np. [3]. Wynikają stąd bezpośrednio następujące wzory na współczynniki wielomianów  $E_p$  i  $F_p$ :

$$\text{wartości początkowe: } e_{1,0} = 1, \quad f_{1,i} = -\bar{a}_{i+1}, \quad i = 0, \dots, n_A,$$

oraz zależności rekurencyjne dla  $p = 2, \dots, N$ ,

$$e_{p,i} = \begin{cases} e_{p-1,i} & \text{gd } i = 0, \dots, p-2, \\ f_{p-1,0} & \text{gd } i = p-1, \end{cases}$$

$$f_{p,i} = \begin{cases} -f_{p-1,0}\bar{a}_{i+1} + f_{p-1,i+1} & \text{gd}y \ i = 0, \dots, n_A - 1, \\ -f_{p-1,0}\bar{a}_{i+1} & \text{gd}y \ i = n_A. \end{cases}$$

Mnożąc równanie modelu obiektu (3.65) stronami przez  $E_p(z^{-1})z^p$  dostajemy

$$E_p(z^{-1})\bar{A}(z^{-1})y(k+p) = E_p(z^{-1})B(z^{-1})\Delta u(k+p-1) + E_p(z^{-1})\varepsilon(k+p).$$

Zapisując tożsamość Bezout (3.66) w postaci

$$E_p(z^{-1})\bar{A}(z^{-1}) = 1 - F_p(z^{-1})z^{-p},$$

i wstawiając do poprzedniej zależności uzyskujemy

$$(1 - F_p(z^{-1})z^{-p})y(k+p) = E_p(z^{-1})B(z^{-1})\Delta u(k+p-1) + E_p(z^{-1})\varepsilon(k+p),$$

czyli

$$y(k+p) = F_p(z^{-1})y(k) + E_p(z^{-1})B(z^{-1})\Delta u(k+p-1) + E_p(z^{-1})\varepsilon(k+p).$$

Ponieważ wielomian  $E_p$  jest stopnia co najwyżej  $p-1$ , to składnik  $E_p(z^{-1})\varepsilon(k+p)$  reprezentuje jedynie *przyszłe wartości* szumu zakłócającego. W regulatorze GPC stosuje się predykcję minimalnowariancyjną, która osiągnana jest przy

$$y(k+p|k) = E\{y(k+p)|k\},$$

gdzie przez  $E\{\cdot|k\}$  oznaczono operator warunkowej wartości oczekiwanej, w warunkach informacji dostępnej w chwili  $k$ , zob. np. [26, 25, 17, 3]. Ponieważ składnik  $F_p(z^{-1})y(k)$  zawiera znane (aktualną i przeszłe) wartości wyjścia, określone przez  $E_p(z^{-1})B(z^{-1})\Delta u(k+p-1|k)$  składniki zawierają jedynie deterministyczne przyrosty sterowań:

$$\Delta u(k+p-1|k) = \begin{cases} \Delta u(k+p-1|k), & p-1 \geq 0 \text{ - zmienne decyzyjne} \\ \Delta u(k+p-1), & p-1 < 0 \text{ - wartości przeszłe} \end{cases} \quad (3.67)$$

oraz wartość średnia sumy równa się sumie wartości średnich, to

$$y(k+p|k) = F_p(z^{-1})y(k) + E_p(z^{-1})B(z^{-1})\Delta u(k+p-1|k) + E\{E_p(z^{-1})\varepsilon(k+p)\}.$$

Ponieważ składnik  $E_p(z^{-1})\varepsilon(k+p)$  reprezentuje jedynie przyszłe wartości szumu białego o zerowej wartości średniej, to  $E\{E_p(z^{-1})\varepsilon(k+p)\} = 0$ . Stąd otrzymujemy predyktor minimalnowariancyjny w postaci

$$\begin{aligned} y(k+p|k) &= F_p(z^{-1})y(k) + E_p(z^{-1})B(z^{-1})\Delta u(k+p-1|k) \\ &= F_p(z^{-1})y(k) + G_p(z^{-1})\Delta u(k+p-1|k), \end{aligned} \quad (3.68)$$

gdzie  $G_p(z^{-1}) = E_p(z^{-1})B(z^{-1})$ . Ponieważ stopień wielomianu  $G_p$  jest  $n_B + (p-1)$ , to w zależności (3.68) występują przyszłe i przeszłe wartości przyrostów sterowania, wstecz aż do chwili  $k+p-1-n_B-(p-1) = k-n_B$ , oraz  $n_A + 1$  znanych wartości wyjścia, w chwilach od  $k$ -tej do  $k-n_A$ .

Każdą z sum  $G_p(z^{-1})\Delta u(k+p-1|k)$ ,  $p = 1, \dots, N$  można rozłożyć na część  $G_p^{FG}(z)$  zależną od aktualnego i przyszłych oraz część  $G_p^{PG}(z^{-1})$  zależną od przeszłych przyrostów sterowania

$$G_p(z^{-1})\Delta u(k+p-1|k) = G_p^{FG}(z)\Delta u(k|k) + G_p^{PG}(z^{-1})\Delta u(k-1), \quad (3.69)$$

gdzie rząd każdego z wielomianów  $G_p^{PG}(z^{-1})$  jest nie większy od  $n_B$ . Ponieważ

$$G_p(z^{-1})\Delta u(k+p-1|k) = G_p(z^{-1})z^{p-1}u(k|k),$$

to możemy napisać

$$G_{N_1}^{PG}(z^{-1}) = G_{N_1}(z^{-1})z^{N_1-1} - s_{N_1}, \quad (3.70a)$$

$$G_{N_1+1}^{PG}(z^{-1}) = G_{N_1+1}(z^{-1})z^{N_1} - s_{N_1}z - s_{N_1+1}, \quad (3.70b)$$

$$\vdots$$

$$G_{N_1+j}^{PG}(z^{-1}) = G_{N_1+j}(z^{-1})z^{N_1+j-1} - s_{N_1}z^j - s_{N_1+1}z^{j-1} - \dots - s_{N_1+j}, \quad (3.70c)$$

$$\vdots$$

$$G_N^{PG}(z^{-1}) = G_N(z^{-1})z^{N-1} - s_{N_1}z^{N-N_1} - s_{N_1+1}z^{N-N_1-1} - \dots - s_N, \quad (3.70d)$$

gdzie, przypomnijmy,  $s_{N_1} = s_{\tau+1}$  jest pierwszym niezerowym współczynnikiem odpowiedzi skokowej obiektu, a  $s_{N_1+1}, s_{N_1+2}, \dots$  są kolejnymi współczynnikami. Interpretacja powyższych zależności jest dość oczywista: od wielomianów  $G_p(z^{-1})$  definiujących całkowitą predykcję każdego z wyjść w zależności od przyszłych i przeszłych sterowań odejmujemy części odpowiadające zależności od wartości aktualnej i przyszłych sterowań (tzn. od zmiennych decyzyjnych).

Przypomnijmy definicje wektorów  $\mathcal{Y}^{zad}(k)$ ,  $\Delta\mathcal{Y}(k)$ ,  $\mathcal{Y}^0(k)$  i  $\Delta\mathcal{U}(k)$  oraz zdefiniujmy wektory  $\Delta\mathcal{U}^{PG}(k)$  i  $\mathcal{Y}^{PG}(k)$

$$\mathcal{Y}^{zad}(k) = \begin{bmatrix} y^{zad}(k+N_1|k) \\ \vdots \\ y^{zad}(k+N|k) \end{bmatrix}, \quad \Delta\mathcal{Y}(k) = \begin{bmatrix} \Delta y(k+N_1|k) \\ \vdots \\ \Delta y(k+N|k) \end{bmatrix},$$

$$\mathcal{Y}^0(k) = \begin{bmatrix} y^0(k + N_1|k) \\ \vdots \\ y^0(k + N|k) \end{bmatrix}, \quad \Delta\mathcal{U}(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k + N_u - 1|k) \end{bmatrix},$$

$$\Delta\mathcal{U}^{PG}(k) = \begin{bmatrix} \Delta u(k-1) \\ \vdots \\ \Delta u(k - n_B) \end{bmatrix}, \quad \mathcal{Y}^{PG}(k) = \begin{bmatrix} y(k) \\ \vdots \\ y(k - n_A) \end{bmatrix}.$$

Możemy teraz (3.69) zapisać w postaci

$$G_p(z^{-1})\Delta u(k + p - 1|k) = \mathbf{g}_p^{FG}\Delta\mathcal{U}(k) + \mathbf{g}_p^{PG}\Delta\mathcal{U}^{PG}(k), \quad p = N_1, \dots, N, \quad (3.71)$$

gdzie  $\mathbf{g}_p^{FG}$  i  $\mathbf{g}_p^{PG}$  są wektorami (wierszowymi) złożonymi ze współczynników wielomianów  $G_p^{FG}$  i  $G_p^{PG}$  (w razie potrzeby uzupełnionymi zerami). W przypadku  $N_u < N$  może się zdarzyć, że rzędy wielomianów definiowanych przez  $\mathbf{g}_j^{FG}$  są dla wyższych wartości  $j$  większe od  $N_u - 1$ , oznaczymy przez  $\bar{\mathbf{g}}_j^{FG}$  ich obcięcie do rzędu  $N_u - 1$ . Oznaczając

$$\mathbf{G}^{FG} = \begin{bmatrix} \bar{\mathbf{g}}_{N_1}^{FG} \\ \vdots \\ \bar{\mathbf{g}}_N^{FG} \end{bmatrix}, \quad \mathbf{G}^{PG} = \begin{bmatrix} \mathbf{g}_{N_1}^{PG} \\ \vdots \\ \mathbf{g}_N^{PG} \end{bmatrix},$$

możemy składowe wymuszaną i swobodną predykcji wyjść zapisać w postaci

$$\Delta\mathcal{Y}(k) = \mathbf{G}^{FG}\Delta\mathcal{U}(k), \quad (3.72)$$

$$\mathcal{Y}^0(k) = \mathbf{F}\mathcal{Y}^{PG}(k) + \mathbf{G}^{PG}\Delta\mathcal{U}^{PG}(k). \quad (3.73)$$

Z postaci równania składowej wymuszonej (3.72) wynika, że musi zachodzić równość  $\mathbf{G}^{FG} = \mathbf{M}$ , gdzie  $\mathbf{M}$  to wprowadzona w rozdziale (3.2.2) macierz dynamiczna (3.30), dla  $N_1 = \tau + 1$  o charakterystycznej postaci (3.31).

Dla wyznaczenia odpowiedzi swobodnej trzeba jeszcze wyznaczyć macierz  $\mathbf{F}$ , której wiersze składają się ze współczynników wielomianów  $F_p$ ,

$$F_p = f_{p,0} + f_{p,1}z^{-1} + \dots + f_{p,n_A}z^{-n_A}, \quad p = N_1, \dots, N,$$

$$\mathbf{F} = \begin{bmatrix} f_{N_1,0} & f_{N_1,1} & \cdots & f_{N_1,n_A} \\ f_{N_1+1,0} & f_{N_1+1,1} & \cdots & f_{N_1+1,n_A} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N,0} & f_{N,1} & \cdots & f_{N,n_A} \end{bmatrix}.$$

Korzystając z wyprowadzonych zależności funkcję kryterialną można zapisać w postaci (3.36)

$$J(k) = \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] - \mathbf{M}\Delta\mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \|\Delta\mathcal{U}(k)\|_{\underline{\Lambda}}^2, \quad (3.74)$$

identycznej jak dla algorytmu DMC rozważanego w rozdziale 3.2.2 – jedynie składowa swobodna  $\mathcal{Y}^0(k)$  będzie teraz w postaci (3.73), a nie (3.27). Rozumowanie prowadzące do wyliczenia analitycznej postaci regulatora predykcyjnego (bez uwzględniania ograniczeń nierównościowych) będzie więc takie jak w rozdziale 3.2.2, otrzymujemy wektor optymalnych przyrostów sterowań

$$\Delta\hat{\mathcal{U}}(k) = \mathbf{K}[\mathcal{Y}^{zad}(k) - (\mathbf{F}\mathcal{Y}^{PG}(k) + \mathbf{G}^{PG}\Delta\mathcal{U}^{PG}(k))], \quad (3.75)$$

oraz *prawo regulacji GPC*

$$\Delta\hat{u}(k) = \Delta\hat{u}(k|k) = \bar{\mathbf{K}}_1[\mathcal{Y}^{zad}(k) - (\mathbf{F}\mathcal{Y}^{PG}(k) + \mathbf{G}^{PG}\Delta\mathcal{U}^{PG}(k))], \quad (3.76)$$

w typowej postaci liniowego sprzężenia zwrotnego od przyszłych uchybów regulacji określonych przez różnicę między trajektorią zadaną a składową swobodną trajektorii prognozowanych wyjść regulowanych, gdzie  $\bar{\mathbf{K}}_1$  oznacza pierwszy wiersz macierzy  $\mathbf{K} = [\mathbf{M}^T \underline{\Psi} \mathbf{M} + \underline{\Lambda}]^{-1} \mathbf{M}^T \underline{\Psi}$ , zob. (3.38), (3.40).

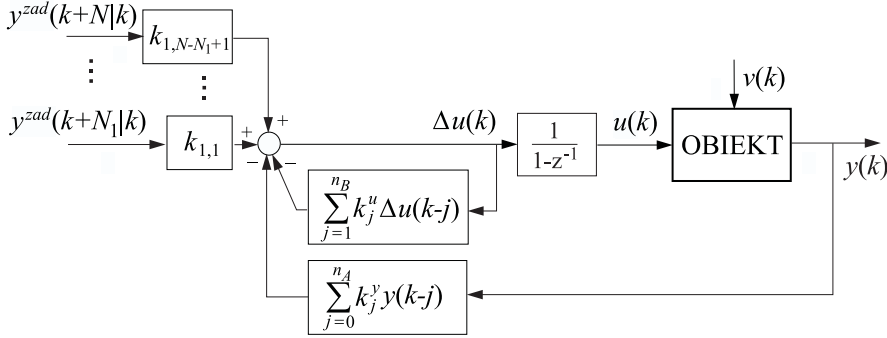
Przedstawimy prawo regulacji GPC w postaci zależnej bezpośrednio od wyjść i sterowań obiektu. Oznaczając

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_0 & \mathbf{F}_1 & \cdots & \mathbf{F}_{n_A} \end{bmatrix},$$

$$\mathbf{G}^{PG} = \begin{bmatrix} \mathbf{G}_1^{PG} & \mathbf{G}_2^{PG} & \cdots & \mathbf{G}_{n_B}^{PG} \end{bmatrix},$$

gdzie  $\mathbf{F}_j$  i  $\mathbf{G}_j^{PG}$  są kolumnami macierzy  $\mathbf{F}$  i  $\mathbf{G}^{PG}$ , oraz wykorzystując strukturę (3.40) macierzy  $\mathbf{K}$  możemy napisać

$$\begin{aligned} \Delta\hat{u}(k) &= \sum_{p=N_1}^N k_{1,p-N_1+1} y^{zad}(k+p|k) + \\ &\quad - \sum_{j=0}^{n_A} (\bar{\mathbf{K}}_1 \mathbf{F}_j) y(k-j) - \sum_{j=1}^{n_B} (\bar{\mathbf{K}}_1 \mathbf{G}_j^{PG}) \Delta u(k-j), \\ &= \sum_{p=N_1}^N k_{1,p-N_1+1} y^{zad}(k+p|k) + \\ &\quad - \sum_{j=0}^{n_A} k_j^y y(k-j) - \sum_{j=1}^{n_B} k_j^u \Delta u(k-j), \end{aligned} \quad (3.77)$$



Rysunek 3.22. Struktura analitycznej wersji regulatora GPC

gdzie, ze względu na rozpatrywanie przypadku układu jednowymiarowego, współczynniki odpowiednich wzmocnień  $k_{1p}$ ,  $k_j^y$  i  $k_j^u$  są skalarami. Uzyskana *struktura wersji analitycznej (prawa regulacji) regulatora GPC* przedstawiona jest na rysunku 3.22.

W ważnym praktycznie przypadku stałej na horyzoncie predykcji wartości zadanej, tzn.  $y^{zad}(k+1|k) = \dots = y^{zad}(k+N|k) = y^{zad}(k)$ , dostajemy prawo regulacji GPC w postaci

$$\Delta \hat{u}(k) = k^e y^{zad}(k) - \sum_{j=0}^{n_A} k_j^y y(k-j) - \sum_{j=1}^{n_B} k_j^u \Delta u(k-j), \quad (3.78)$$

gdzie  $k^e = \sum_{p=N_1}^N k_{1,p-N_1+1}$ .

Struktura prawa regulacji GPC (3.77), czy (3.78), odpowiada spotykanym w literaturze, zob. np. [17], ale jest niezbyt dogodna do pewnych porównań i analiz. Dlatego też przekształcimy to prawo do postaci nieco innej, korzystając z udowodnionego niżej lematu pokazującego dokładniej strukturę macierzy  $\mathbf{F}$ . Przypomnijmy, że wiersze macierzy  $\mathbf{F}$  składają się ze współczynników wielomianów  $F_p(z^{-1})$ ,  $p = N_1, \dots, N$ , wyznaczonych z rozwiązań tożsamości Bézout (3.66).

**Lemat 3.1.** *Każdy z wielomianów*

$$F_p(z^{-1}) - 1 = f_{p,0} - 1 + f_{p,1}z^{-1} + \dots + f_{p,n_A}z^{-n_A}, \quad p = 1, 2, \dots, N,$$

*jest podzielny przez  $\Delta = 1 - z^{-1}$ .*

**Dowód.** Tezę wykażemy korzystając z zasady indukcji matematycznej. Dla  $p = 1$  tożsamość Bézout (3.66) ma postać

$$E_1(z^{-1})\bar{A}(z^{-1}) + F_1(z^{-1})z^{-1} = 1.$$

Odejmując od obu stron  $z^{-1}$  uzyskujemy

$$E_1(z^{-1})\Delta A(z^{-1}) + (F_1(z^{-1}) - 1)z^{-1} = \Delta,$$

gdź z definicji  $\bar{A}(z^{-1}) = \Delta A(z^{-1})$ . Stąd, jeśli wielomian  $F_1(z^{-1})$  jest rozwiązaniem tożsamości Bezout, to wielomian  $F_1(z^{-1}) - 1$  musi być podzielny przez  $\Delta$ .

Założmy teraz, że dla pewnego  $p \geq 1$  wielomian  $F_p(z^{-1}) - 1$  jest podzielny przez  $\Delta$ . Z zasady uzyskiwania wielomianów  $E_p$  i  $F_p$  poprzez dzielenie 1 przez wielomian  $\bar{A}(z^{-1})$  wynika bezpośrednio zależność rekurencyjna

$$F_{p+1}(z^{-1})z^{-1} = F_p(z^{-1}) - f_{p,0}\bar{A}(z^{-1}).$$

Odejmując od lewej strony tej równości  $z^{-1}$ , a od prawej  $z^{-1} = z^{-1} - 1 + 1$  dostajemy

$$(F_{p+1}(z^{-1}) - 1)z^{-1} = (F_p(z^{-1}) - 1) + (1 - z^{-1}) - f_{p,0}\bar{A}(z^{-1}).$$

Ponieważ  $\bar{A}(z^{-1}) = \Delta A(z^{-1})$ , to z założenia podzielności  $F_p(z^{-1}) - 1$  przez  $\Delta$  wynika podzielność  $F_{p+1}(z^{-1}) - 1$  przez  $\Delta$ , co kończy dowód.  $\square$

Na mocy lematu, każdy z wielomianów  $F_p(z^{-1})$  można przedstawić w postaci

$$\begin{aligned} F_p(z^{-1}) &= 1 + \bar{F}_p(z^{-1})\Delta \\ &= 1 + \bar{f}_{p,1}\Delta + \bar{f}_{p,2}z^{-1}\Delta + \cdots + \bar{f}_{p,n_A}z^{-n_A+1}\Delta, \end{aligned} \quad (3.79)$$

gdzie

$$\begin{aligned} \bar{f}_{p,1} &= f_{p,0} - 1, \\ \bar{f}_{p,j+1} &= f_{p,j} + \bar{f}_{p,j}, \quad j = 2, \dots, n_A - 1. \end{aligned}$$

Wynika stąd alternatywne przedstawienie iloczynu  $\mathbf{FY}^{PG}(k)$

$$\begin{aligned} \mathbf{FY}^{PG}(k) &= \mathbf{I}y(k) + \begin{bmatrix} \bar{f}_{N_1,1} & \bar{f}_{N_1,2} & \cdots & \bar{f}_{N_1,n_A} \\ \bar{f}_{N_1+1,1} & \bar{f}_{N_1+1,2} & \cdots & \bar{f}_{N_1+1,n_A} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{f}_{N,1} & \bar{f}_{N,2} & \cdots & \bar{f}_{N,n_A} \end{bmatrix} \begin{bmatrix} \Delta y(k) \\ \Delta y(k-1) \\ \vdots \\ \Delta y(k-n_A+1) \end{bmatrix}, \\ &= \mathbf{I}y(k) + \bar{\mathbf{F}}\Delta\mathcal{Y}^{PG}(k) \end{aligned} \quad (3.80)$$

gdzie  $\Delta \mathcal{Y}^{PG}(k) = [\Delta y(k) \ \Delta y(k-1) \ \cdots \ \Delta y(k-n_A+1)]^T$ . Oznaczając kolumny macierzy  $\bar{\mathbf{F}}$  przez  $\bar{\mathbf{F}}_j$ ,  $\bar{\mathbf{F}} = [\bar{\mathbf{F}}_1 \ \cdots \ \bar{\mathbf{F}}_{n_A}]$ , prawo regulacji GPC (3.77) można teraz przekształcić do postaci

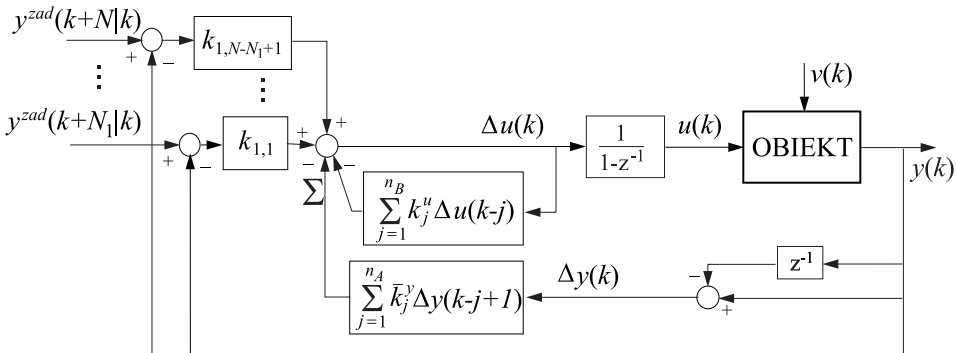
$$\begin{aligned} \Delta \hat{u}(k) &= \sum_{p=N_1}^N k_{1,p-N_1+1} (y^{zad}(k+p|k) - y(k)) + \\ &\quad - \sum_{j=1}^{n_A} (\bar{\mathbf{K}}_1 \bar{\mathbf{F}}_j) \Delta y(k-j+1) - \sum_{j=1}^{n_B} k_j^u \Delta u(k-j), \\ &= \sum_{p=N_1}^N k_{1,p-N_1+1} (y^{zad}(k+p|k) - y(k)) + \\ &\quad - \sum_{j=1}^{n_A} \bar{k}_j^y \Delta y(k-j+1) - \sum_{j=1}^{n_B} k_j^u \Delta u(k-j), \end{aligned} \quad (3.81)$$

redukującej się przy stałej na horyzoncie predykcji wartości zadanej do

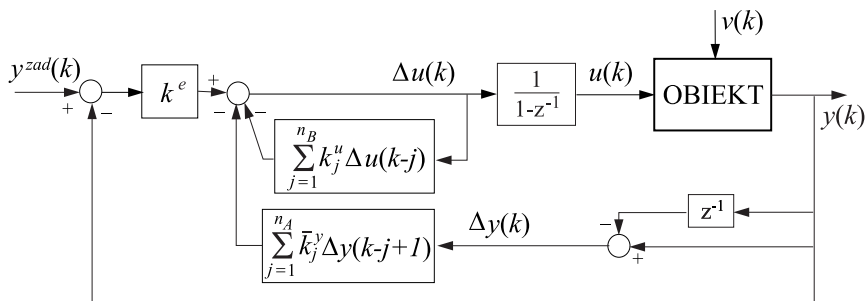
$$\Delta \hat{u}(k) = k^e (y_k^{zad} - y(k)) - \sum_{j=1}^{n_A} \bar{k}_j^y \Delta y(k-j+1) - \sum_{j=1}^{n_B} k_j^u \Delta u(k-j). \quad (3.82)$$

Rysunki 3.23 i 3.24 ilustrują uzyskaną alternatywną strukturę prawa regulacji GPC.

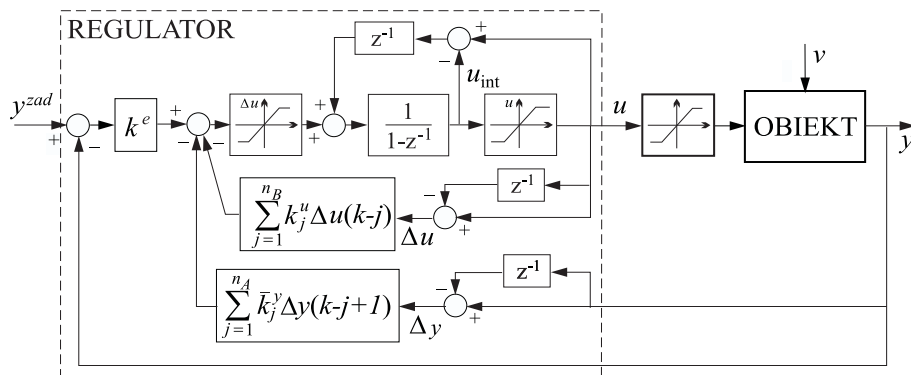
W przypadku występowania ograniczeń sygnału sterującego należy strukturę regulatora analitycznego GPC odpowiednio uzupełnić, analogicznie jak w przypadku regulatora DMC – uwzględniając ograniczone przyrosty i amplitudy sterowania oraz wprowadzając pętlę typu *anti-wind-up*. Przykład takiej struktury, odpowiadający zilustrowanemu rysunkiem 3.24 prawu regulacji (3.82), pokazany jest na rysunku 3.25 (por. rys. 3.15).



Rysunek 3.23. Alternatywna struktura analitycznej wersji regulatora GPC



Rysunek 3.24. Alternatywna struktura analitycznej wersji regulatora GPC w przypadku stałej trajektorii wartości zadanej na horyzoncie predykcji



Rysunek 3.25. Struktura regulatora GPC z rysunku poprzedniego uzupełniona o elementy uwzględnienia ograniczeń przyrostu i amplitudy sterowania

Prawo regulacji GPC przedstawione w postaci (3.81), czy też (3.82) dla stałej prognozy wartości zadanych, łatwo jest porównać z prawem regulacji DMC danym wzorem (3.42) czy, odpowiednio, (3.43). Analogicznie można porównać ilustrujące strukturę regulatora analitycznego GPC rysunki 3.23 i 3.24 z rysunkami 3.5 i 3.6 przedstawiającymi strukturę regulatora analitycznego DMC. W regulatorze DMC mamy jedno sprzężenie zwrotne od wyjścia, od jego aktualnej wartości  $y(k)$  oraz sprzężenia od  $D - 1$  wartości poprzednich przyrostów sterowania, gdzie  $D$  jest horyzontem dynamiki obiektu. Natomiast w regulatorze GPC występują sprzężenia zwrotne od aktualnej wartości wyjścia i  $n_A$  ostatnich jego przyrostów oraz od  $n_B$  poprzednich wartości przyrostów sterowania, gdzie  $n_A$  i  $n_B$  są stopniami wielomianów modelu obiektu (3.65).

Jeśli okres próbkowania jest mały w stosunku do horyzontu dynamiki

obiektu, to odpowiedź skokowa zawiera dużą liczbę elementów, jest „długa”. Jednocześnie może być ona nierzadko opisana transmitancją dyskretną z wielomianami niskiego rzędu. Wówczas  $D \gg n_A$  i  $D \gg n_B$ , model dla algorytmu GPC ma oszczędniejszą reprezentację, w strukturze prawa regulacji występuje mniej sprzężeń zwrotnych. Pamiętajmy jednak, że model w postaci odpowiedzi skokowej jest modelem nieparametrycznym, uzyskiwanym najczęściej na podstawie bezpośredniego, prostego i zrozumiałego eksperymentu na obiekcie. Dla uzyskania modelu wykorzystywanego w algorytmie GPC należy natomiast przeprowadzić odpowiednią identyfikację jego struktury i parametrów.

### Przykład 3.2.

Dla zilustrowania sposobu wyznaczania i pokazania postaci prawa regulacji GPC obliczymy to prawo dla obiektu z przykładu 3.1, opisanego dyskretną odpowiedzią skokową (3.46). Podobnie jak poprzednio przyjmijmy  $N_1 = 3$ ,  $N_u = 3$ ,  $N = 6$  oraz  $\underline{\Psi} = \mathbf{I}$ ,  $\underline{\Lambda} = \lambda \mathbf{I}$ . Dla potrzeb algorytmu regulacji GPC obiekt opiszemy modelem w postaci dyskretnego równania różnicowego

$$y(k) + ay(k-1) = b_2u(k-3) + b_3u(k-4) + \varepsilon(k)/\Delta,$$

odpowiadającego inercji z opóźnieniem  $\tau$  równym dwu okresom próbkowania – wielkość opóźnienia wynika bezpośrednio z przebiegu odpowiedzi skokowej obiektu. Dopasowując, przy  $\varepsilon(k) = 0$ , parametry modelu do punktów odpowiedzi skokowej (metodą najmniejszych kwadratów) uzyskujemy

$$(1 - 0.2676z^{-1})y(k) = (0.1989z^{-2} + 0.2552z^{-3})u(k-1), \quad (3.83)$$

tnz. model z macierzami wielomianowymi

$$\begin{aligned} A(z^{-1}) &= 1 - 0.2676z^{-1}, \\ B(z^{-1}) &= 0.1989z^{-2} + 0.2552z^{-3}. \end{aligned}$$

Ponieważ

$$\bar{A}(z^{-1}) = A(z^{-1})(1 - z^{-1}) = 1 - 1.2676z^{-1} + 0.2676z^{-2},$$

to tożsamość Bézout ma w rozważanym przypadku postać

$$E_p(z^{-1})(1 - 1.2676z^{-1} + 0.2676z^{-2}) + F_p(z^{-1})z^{-p} = 1.$$

Rozwiązania tożsamości dla kolejnych wartości  $p = 1, 2, \dots, 6$ , uzyskane drogą dzielenia 1 przez  $\bar{A}(z^{-1})$ , są przedstawione w tablicy 3.1, gdzie w ostatniej kolumnie podano również wielomiany  $\bar{F}_p(z^{-1})$  występujące w przekształconej postaci (3.79) wielomianów  $F_p(z^{-1})$ .

p	$E_p(z^{-1})$	$F_p(z^{-1})$	$\bar{F}_p(z^{-1})$
1	1	$1.2676 - 0.2676z^{-1}$	0.2676
2	$1 + 1.2676z^{-1}$	$1.3392 - 0.3392z^{-1}$	0.3392
3	$1 + 1.2676z^{-1} + 1.3392z^{-2}$	$1.3584 - 0.3584z^{-1}$	0.3584
4	$1 + 1.2676z^{-1} + 1.3392z^{-2} + 1.3584z^{-3}$	$1.3635 - 0.3635z^{-1}$	0.3635
5	$1 + 1.2676z^{-1} + 1.3392z^{-2} + 1.3584z^{-3} + 1.3635z^{-4}$	$1.3649 - 0.3649z^{-1}$	0.3649
6	$1 + 1.2676z^{-1} + 1.3392z^{-2} + 1.3584z^{-3} + 1.3635z^{-4} + 1.3649z^{-5}$	$1.3652 - 0.3652z^{-1}$	0.3652

Tablica 3.1. Wielomiany rozwiązań tożsamości Bézout w przykładzie

Wyznaczamy następnie wielomiany  $G_p(z^{-1}) = E_p(z^{-1})B(z^{-1})$ , ze względu na opóźnienie  $\tau = 2$  poczynając od trzeciego:

$$\begin{aligned} G_3(z^{-1}) &= (1 + 1.2676z^{-1} + 1.3392z^{-2})(0.1989z^{-2} + 0.2552z^{-3}) \\ &= 0.1989z^{-2} + 0.5073z^{-3} + 0.5899z^{-4} + 0.3418z^{-5}, \end{aligned}$$

$$\begin{aligned} G_4(z^{-1}) &= (1 + 1.2676z^{-1} + 1.3392z^{-2} + 1.3584z^{-3})(0.1989z^{-2} + 0.2552z^{-3}) \\ &= 0.1989z^{-2} + 0.5073z^{-3} + 0.5899z^{-4} + 0.6119z^{-5} + 0.3467z^{-6}, \end{aligned}$$

$$\begin{aligned} G_5(z^{-1}) &= (1 + 1.2676z^{-1} + 1.3392z^{-2} + 1.3584z^{-3} + 1.3635z^{-4}) \cdot \\ &\quad \cdot (0.1989z^{-2} + 0.2552z^{-3}) \\ &= 0.1989z^{-2} + 0.5073z^{-3} + 0.5899z^{-4} + 0.6119z^{-5} + 0.6179z^{-6} + \\ &\quad + 0.3480z^{-7}, \end{aligned}$$

$$\begin{aligned} G_6(z^{-1}) &= (1 + 1.2676z^{-1} + 1.3392z^{-2} + 1.3584z^{-3} + 1.3635z^{-4} + \\ &\quad + 1.3649z^{-5})(0.1989z^{-2} + 0.2552z^{-3}) \\ &= 0.1989z^{-2} + 0.5073z^{-3} + 0.5899z^{-4} + 0.6119z^{-5} + 0.6179z^{-6} + \\ &\quad + 0.6194z^{-7} + 0.3483z^{-8}. \end{aligned}$$

Jeśli teraz zapiszemy te wielomiany w postaci  $G_p(z^{-1})z^{p-1}$ , por. zależności (3.70a)-(3.70d), to dobrze widoczny będzie podział na współczynniki określające odpowiedź wymuszaną (elementy odpowiedzi skokowej) i współczynniki określające odpowiedź swobodną, tzn. elementy wektorów  $\mathbf{g}_p^{PG}$

i  $\mathbf{g}_p^{FG}$ , odpowiednio, zob. (3.71). Otrzymujemy bowiem wówczas:

$$G_3(z^{-1})z^2 = 0.1989 + 0.5073z^{-1} + 0.5899z^{-2} + 0.3418z^{-3},$$

$$G_4(z^{-1})z^3 = 0.1989z^1 + 0.5073 + 0.5899z^{-1} + 0.6119z^{-2} + 0.3467z^{-3},$$

$$G_5(z^{-1})z^4 = 0.1989z^2 + 0.5073z^1 + 0.5899 + 0.6119z^{-1} + 0.6179z^{-2} + 0.3480z^{-3},$$

$$G_6(z^{-1})z^5 = 0.1989z^3 + 0.5073z^2 + 0.5899z^1 + 0.6119 + 0.6179z^{-1} + 0.6194z^{-2} + 0.3483z^{-3}.$$

Możemy teraz zapisać odpowiedź wymuszaną  $\Delta\mathcal{Y}(k)$  w postaci wektorowej (3.72),  $\Delta\mathcal{Y}(k) = \mathbf{G}^{FG} \Delta\mathcal{U}(k)$ , tzn.  $\Delta\mathcal{Y}(k) = \mathbf{M} \Delta\mathcal{U}(k)$ ,

$$\begin{bmatrix} \Delta y(k+3|k) \\ \Delta y(k+4|k) \\ \Delta y(k+5|k) \\ \Delta y(k+6|k) \end{bmatrix} = \begin{bmatrix} 0.1989 & 0 & 0 \\ 0.5073 & 0.1989 & 0 \\ 0.5899 & 0.5073 & 0.1989 \\ 0.6119 & 0.5899 & 0.5073 \end{bmatrix} \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \Delta u(k+2|k) \end{bmatrix},$$

zaś macierz  $\mathbf{G}^{PG}$  jest postaci

$$\mathbf{G}^{PG} = \begin{bmatrix} 0.5073 & 0.5899 & 0.3418 \\ 0.5899 & 0.6119 & 0.3467 \\ 0.6119 & 0.6179 & 0.3480 \\ 0.6179 & 0.6194 & 0.3483 \end{bmatrix},$$

gdzie, wykorzystując ostatni wielomian  $G_6(z^{-1})z^5$  do określenia odpowiedzi wymuszonej dla  $k = N = 6$  pominęliśmy pierwszy jego składnik, ze względu na długość horyzontu sterowania  $N_u$  równą 3.

Dla określenia pełnej odpowiedzi swobodnej trzeba jeszcze wykorzystać macierz  $\mathbf{F}$ , której współczynniki podane są bezpośrednio w trzeciej kolumnie macierzy 3.1. Pełna, wektorowa postać odpowiedzi swobodnej (3.73) jest stąd następująca

$$\begin{bmatrix} y^0(k+3|k) \\ y^0(k+4|k) \\ y^0(k+5|k) \\ y^0(k+6|k) \end{bmatrix} = \begin{bmatrix} 0.5073 & 0.5899 & 0.3418 \\ 0.5899 & 0.6119 & 0.3467 \\ 0.6119 & 0.6179 & 0.3480 \\ 0.6179 & 0.6194 & 0.3483 \end{bmatrix} \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \Delta u(k-3) \end{bmatrix} + \begin{bmatrix} 1.3584 & -0.3584 \\ 1.3635 & -0.3635 \\ 1.3649 & -0.3649 \\ 1.3652 & -0.3652 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix},$$

$$\text{tzn. } \mathcal{Y}^0(k) = \mathbf{G}^{PG} \Delta \mathcal{U}^{PG}(k) + \mathbf{F} \mathcal{Y}^{PG}(k).$$

Zwróćmy uwagę na różnice w wartościach elementów macierzy dynamicznych  $\mathbf{M}$  uzyskanych powyżej i w przykładzie 3.1, porównaj z (3.47). Różnice spowodowane są tym, że macierz uzyskana powyżej składa się z elementów odpowiedzi skokowej układu dynamicznego opisanego równaniem (3.83), a nie oryginalnej odpowiedzi skokowej (3.46).

Rozwiązanie zadania optymalizacji dane jest wzorem (3.75), gdzie macierze  $\mathbf{K}$  są, dla tych samych wartości  $\lambda$ , zbliżone do uzyskanych w przykładzie 3.1, z odstępstwami wynikającymi z różnic w wartościach macierzy  $\mathbf{M}$ . Zakładając  $y^{zad}(k+3|k) = \dots = y^{zad}(k+6|k) = y^{zad}(k)$  dostajemy prawo regulacji (3.77) w postaci

$$\begin{aligned} \Delta \hat{u}(k) &= \sum_{p=1}^4 k_{1p} y^{zad}(k) - \sum_{j=0}^1 (\bar{\mathbf{K}}_1 \mathbf{F}_{j+1}) y(k-j) - \sum_{j=1}^3 (\bar{\mathbf{K}}_1 \mathbf{G}_j^{PG}) \Delta u(k-j) \\ &= k^e y^{zad}(k) - \sum_{j=0}^1 k_j^y y(k-j) - \sum_{j=1}^3 k_j^u \Delta u(k-j), \end{aligned}$$

gdzie  $\bar{\mathbf{K}}_1$  jest pierwszym wierszem macierzy  $\mathbf{K}$ , a  $\mathbf{F}_j$  i  $\mathbf{G}_j^{PG}$  kolumnami macierzy  $\mathbf{F}$  i  $\mathbf{G}^{PG}$ .

Dla  $\lambda = 0.01$  uzyskujemy parametry prawa regulacji GPC o wartościach:

$$\begin{aligned} k^e &= 2.8618, \quad [k_0^y \ k_1^y] = [3.8933 \ -1.0314], \\ [k_1^u \ k_2^u \ k_3^u] &= [1.5454 \ 1.7131 \ 0.9835], \end{aligned}$$

dla  $\lambda = 0.05$  dostajemy wartości:

$$\begin{aligned} k^e &= 2.0653, \quad [k_0^y \ k_1^y] = [2.8177 \ -0.7464], \\ [k_1^u \ k_2^u \ k_3^u] &= [1.1484 \ 1.2453 \ 0.7120], \end{aligned}$$

a dla  $\lambda = 0.1$  wartości:

$$\begin{aligned} k^e &= 1.7153, \quad [k_0^y \ k_1^y] = [2.3362 \ -0.6208], \\ [k_1^u \ k_2^u \ k_3^u] &= [0.9678 \ 1.0381 \ 0.5922]. \end{aligned}$$

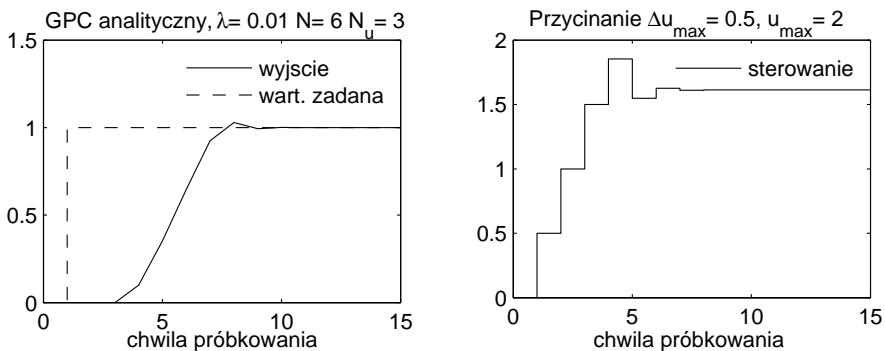
Zauważmy, że wartości modułów współczynników sprzężeń zwrotnych  $k^e$ ,  $k_j^u$  i  $k_j^y$  maleją wraz ze wzrostem wartości współczynnika wagowego (kary)  $\lambda$ , podobnie jak to było w przypadku algorytmu DMC.

Działanie układu regulacji z wyznaczonym algorytmem GPC zasymulowano najpierw bez ograniczeń. Uzyskanych przebiegów nie zamieszczamy, ponieważ są one praktycznie identyczne z zamieszczonymi w przykładzie 3.1, otrzymanymi algorytmem DMC, zob. rys. 3.7, 3.8 i 3.9. Podobne

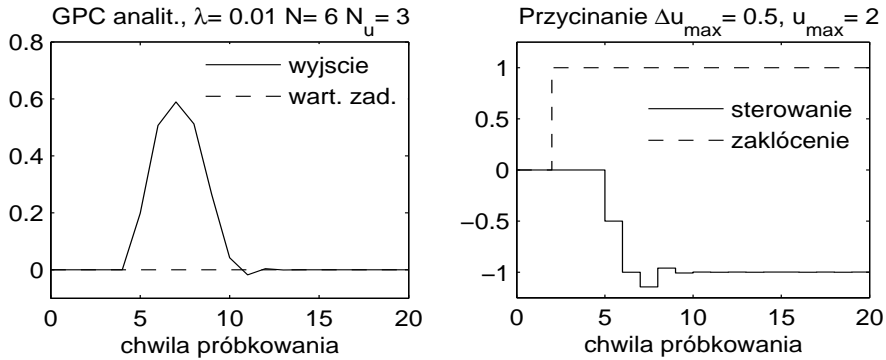
do uzyskanych algorytmem DMC przebiegi uzyskano również w przypadku ograniczeń amplitudy i szybkości zmiany sygnału sterującego, przy stosowaniu struktury z rysunku 3.25, wynik taki zamieszczamy na rysunku 3.26, por. rys. 3.17. Na rysunku 3.27 zamieszczono wyniki symulacji analogiczne do przedstawionych na rysunku 3.19 dla regulatora DMC, ze skokiem zakłócenia niemierzonego na wejściu obiektu, zaś na rysunku 3.28 przebiegi przy wzmacnieniu obiektu zwiększonym o 20%, por. rys. 3.20. Na rysunku 3.29 pokazano natomiast przypadek z dodatkowym skokiem niemierzalnego zakłócenia na wyjściu obiektu, od zera do 0.2 w chwili  $k = 12$ . Przebiegi przedstawione na rysunkach 3.30 i 3.31 uzyskano stosując w prawie regulacji GPC trajektorię referencyjną (3.62), zamiast trajektorii wartości zadanej. Ilustrują one wpływ zastosowania trajektorii referencyjnej, z wartościami, odpowiednio,  $\gamma = 0.3$  i  $\gamma = 0.6$ , przy  $\lambda = 0.1$  (porównaj też z rysunkiem 3.7 odpowiadającym wartości  $\gamma = 0$ ).

Zastosowanie trajektorii referencyjnej powoduje łagodniejszy przebieg zmiennej regulowanej, wydłużając czas dojścia do wartości zadanych. Skutek ten może być oczywiście osiągnąć jedynie poprzez odpowiednie kształtowanie trajektorii sterowania, ograniczanie jego zmienności – ale wpływ na sterowanie jest słabszy niż poprzez współczynnik wagowy  $\lambda$ .  $\square$

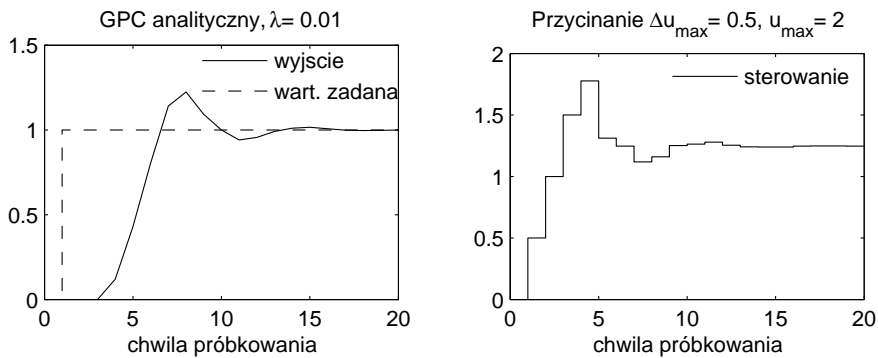
Porównując przykłady 3.2 i 3.1, dotyczące tego samego obiektu i regulatorów GPC oraz DMC należy podkreślić, że rozpatrywana odpowiedź skokowa obiektu (3.46) jest bardzo krótka, zawiera jedynie 6 elementów ( $D = 6$ ). Stąd liczby współczynników odpowiedzi skokowej i parametrów modelu dyskretnego transmitancyjnego są podobne. Wynika to oczywiście z dużego, w stosunku do dynamiki obiektu, okresu próbkowania. Komen-



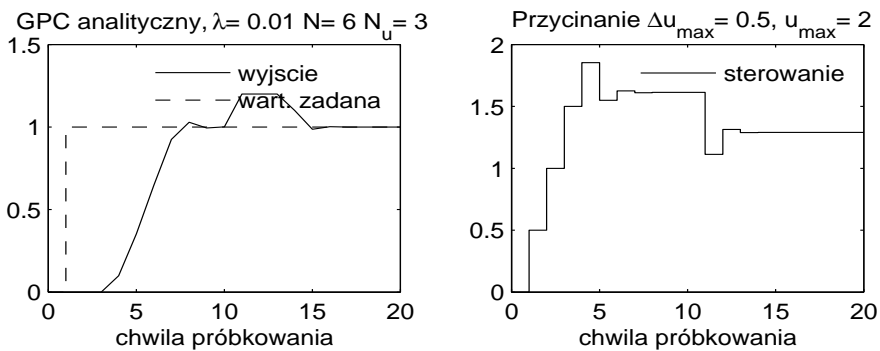
Rysunek 3.26. Wyniki symulacji z regulatorem GPC analitycznym z ograniczeniami amplitudy i przyrostu sterowania



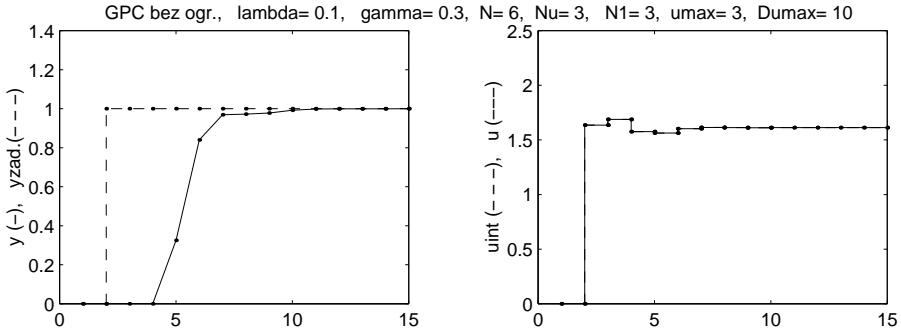
Rysunek 3.27. Wyniki symulacji z regulatorem GPC analitycznym, ze skokiem jednostkowym zakłócenia niemierzalnego  $z(k)$  na wejściu obiektu w chwili  $k = 2$



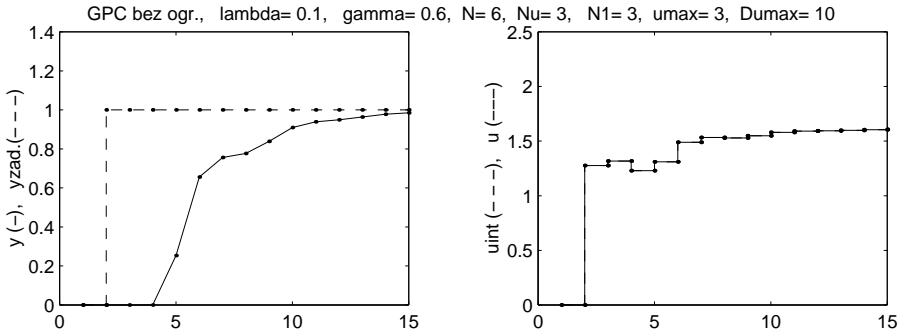
Rysunek 3.28. Wyniki symulacji z regulatorem GPC analitycznym z ograniczeniami, wzmocnienie obiektu zwiększone o 20%



Rysunek 3.29. Wyniki symulacji z regulatorem GPC analitycznym, dodatkowo skok zakłócenia niemierzalnego na wyjściu obiektu o 0.2 w chwili  $k=12$



Rysunek 3.30. Przebiegi w układzie regulacji GPC bez ograniczeń z trajektorią referencyjną dla  $\gamma = 0.3$



Rysunek 3.31. Przebiegi w układzie regulacji GPC bez ograniczeń z trajektorią referencyjną dla  $\gamma = 0.6$

tarza wymaga natomiast fakt uzyskiwania praktycznie identycznych przebiegów w symulacjach układów regulacji.

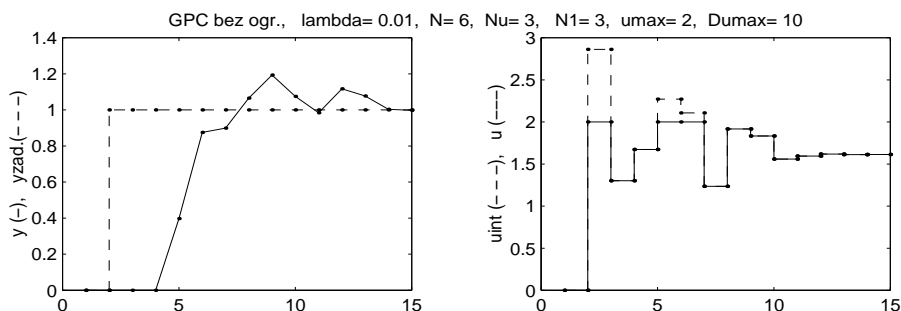
Predyktor wyjść w regulacji GPC wyprowadza się jako minimalnowariancyjny, dla modelu (3.65) z zakłóceniem w postaci całkowanego białego szumu  $\varepsilon(k)$ . Uzyskane wzory są identyczne jak dla modelu

$$\begin{aligned} A(z^{-1})y(k) &= B(z^{-1})u(k-1) + \frac{\varepsilon(k)}{\Delta} \\ &= B(z^{-1})u(k-1) + d(k) \end{aligned}$$

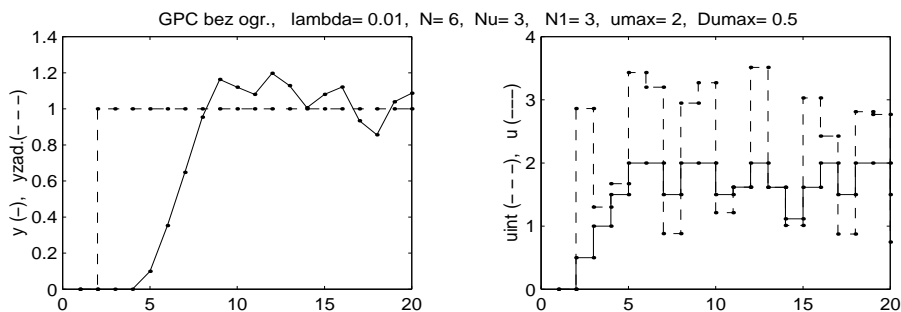
z deterministyczną prognozą zakłócenia  $d(k)$ , gdzie  $\varepsilon(k)$  jest przebiegiem będącym impulsem idealnym o wartości równej  $d(k)$  w chwili  $k$ -tej i zerem na horyzoncie predykcji od  $k+1$  do  $k+N$ ; zatem  $d(k)$  jest zakłóceniem skokowo zmiennym w chwili  $k$ , z prognozą typu DMC nie przewidującą zmiany wartości na horyzoncie predykcji. Tak więc *algorytm GPC stosować*

można również przy modelu zakłóceń odpowiadającym algorytmowi DMC, inne wzory na predykcję w obu algorytmach traktując jako inne modelowanie tego samego liniowego obiektu. Stąd też, przy tej samej postaci funkcji kryterialnej, identyczne wyniki symulacji w przykładach 3.2 i 3.1.

Natomiast w przypadku występowania różnicy między obiektem a jego modelem użytym do projektowania, efekty zastosowania regulatorów DMC i GPC mogą się różnić. Modele liniowe są z reguły przybliżonymi opisami obiektu nieliniowego w punkcie pracy. Stąd, jeśli zakłócenia czy zmiany wartości zadanych wyprowadzają obiekt poza zakres dobrego przybliżenia liniowego, to efekty niedokładnego modelowania stają się widoczne, na ogół w nieco różny sposób, zależnie od rodzaju zastosowanego regulatora liniowego. Widać to już porównując rysunek 3.28 z 3.20. Dodatkowo, dla porównania, zasymulowano układ regulacji z analitycznym algorytmem GPC z przykładu 3.2 w strukturze nie uwzględniającej przy wyliczaniu sterowa-



Rysunek 3.32. Przebiegi z regulatorem analitycznym GPC, z ograniczeniem amplitudy sterowania nie uwzględnionym w strukturze regulatora



Rysunek 3.33. Przebiegi z regulatorem analitycznym GPC, z ograniczeniami amplitudy ( $\leq 2$ ) i przyrostu ( $\leq 0.5$ ) sterowania nie uwzględnionymi w strukturze regulatora

nia informacji o ograniczeniach, w sytuacjach odpowiadających sterowaniu analitycznym algorytmem DMC przedstawionych na rysunkach 3.10 i 3.16. W pierwszym przypadku aktywne jest ograniczenie amplitudy, przebiegi dla algorytmu GPC przedstawiono na rysunku 3.32. Przebieg wyjścia jest nieco zbliżony do przedstawionego na rysunku 3.10, ale przebieg sterowania jest od pewnego momentu wyraźnie różny. Natomiast na rysunku 3.33 przedstawiono przebiegi przy dodaniu jeszcze aktywnego ograniczenia szybkości zmiany sterowania. Porównując z odpowiadającymi tej sytuacji przebiegami z rysunku 3.16 widzimy duże różnice.

Algorytm GPC zapewnia, tak jak DMC, *zerowe uchyby w stanie ustalonym*. Fakt ten można prosto uzasadnić rozumując tak, jak w przypadku algorytmu DMC w pierwszej części rozdziału 3.2.4. Należy jedynie wykozystać równania predykcji w alternatywnej postaci

$$y(k+p|k) = y(k) + \sum_{j=1}^{n_A} \bar{f}_{p,j} \Delta y(k-j+1) + \sum_{j=1}^{n_B} g_{p,j}^{PG} \Delta u(k-j) + \\ + \sum_{j=1}^t g_{p,j}^{FG} \Delta u(k+t-j|k), \quad t = \min\{p, N_u\}, \quad p = N_1, \dots, N,$$

która w zapisie wektorowym przyjmuje formę

$$\mathcal{Y}^{pred}(k) = \mathcal{Y}(k) + \bar{\mathbf{F}} \Delta \mathcal{Y}^{PG}(k) + \mathbf{G}^{PG} \Delta \mathcal{U}^{PG}(k) + \mathbf{M} \Delta \mathcal{U}(k),$$

oraz zauważyć, że w stanie ustalonym wszystkie przyrosty, zarówno sterowań jak i wyjść, są zerowe.

Istotniejszą różnicą między algorytmami GPC i DMC jest natomiast fakt, że modelem odpowiedzi skokowej możemy opisywać jedynie *obiekty stabilne, co najwyżej z całkowaniem* (wówczas stosujemy przyrostową odpowiedź skokową, zob. rozdz. 3.2.1). Natomiast model transmitancyjny może opisywać również *obiekty niestabilne*, z biegunami transmitancji dyskretniej o module większym od jedności. Dla zapewnienia poprawnej pracy regulatora ważny jest wówczas właściwy dobór horyzontów sterowania i predykcji. Muszą one być odpowiednio długie tak, aby wytłumiać wpływ niestabilnych modów obiektu w czasie nie przekraczającym długości horyzontu predykcji, zob. [128, 87].

### 3.3.2. Przypadek zakłócenia typu DMC

Przedstawiony sposób formułowania algorytmu predykcyjnego GPC, oparty na zastosowaniu równania diofantycznego (tożsamości Bézout) i z modelem zakłócenia w postaci scałkowanego białego szumu, jest ogólny

i pozwala na pełne sformułowanie i analizę prawa regulacji – ale jest dość skomplikowany. Definiujące algorytm GPC formuły *można wyprowadzić prościej*, zakładając model zakłócenia taki jak w algorytmie DMC, co pokazujemy poniżej.

Zakładając skok jednostkowy sterowania, z modelu (typu ARX)

$$y(k) = -a_1 y(k-1) - \dots - a_{n_A} y(k-n_A) + b_0 u(k-1) + \dots + b_{n_B} u(k-n_B-1) \quad (3.84)$$

wyznaczyć można bezpośrednio ciąg elementów  $\{s_1, s_2, \dots\}$  odpowiedzi skokowej, co opisane jest zależnością

$$s_k = - \sum_{i=1}^{\min\{k-1, n_A\}} a_i s_{k-i} + \sum_{i=0}^{\min\{k-1, n_B\}} b_i. \quad (3.85)$$

Elementy odpowiedzi skokowej określają macierz dynamiczną  $\mathbf{M}$  pozwalającą wyliczać składową wymuszaną trajektorii prognozowanej wyjść (3.72).

Dla wyznaczenia elementów  $y^0(k+p|k)$  trajektorii swobodnej przyjmujemy na horyzoncie predykcji sterowanie stałe, równe wartości wyznaczonej w chwili poprzedniej

$$u(k+p|k) = u(k-1), \quad p = 0, 1, \dots, N-1. \quad (3.86)$$

Przy takich sterowaniach wyznaczamy  $y^0(k+p|k)$  kolejno, dla  $p = 1, \dots, N$ , z wzoru

$$y^0(k+p|k) = y(k+p) + d(k), \quad (3.87)$$

gdzie wartości  $y(k+p)$  liczymy z modelu (3.84) stosowanego dla  $k = k+p$ , przyjmując dla chwil  $k+p-j > k$  w miejsce jeszcze niezrealizowanych wartości  $y(k+p-j)$  uprzednio wyliczone wartości  $y^0(k+p-j|k)$ . Prognozowane na podstawie modelu (3.84) wyjścia swobodne są uzupełnione estymatą zakłócenia  $d(k)$  w chwili  $k$ , stałą na horyzoncie predykcji i równą wartości wyliczonej w chwili bieżącej  $k$  (zgodnie z przyjętym modelem zakłócenia typu DMC)

$$\begin{aligned} d(k) &= y(k) - y(k|k-1) \\ &= y(k) - \left[ - \sum_{i=1}^{n_A} a_i y(k-i) + \sum_{i=0}^{n_B} b_i u(k-1-i) \right]. \end{aligned} \quad (3.88)$$

Przedstawiony sposób postępowania prowadzi do następującego wzoru na elementy trajektorii swobodnej wyjść przewidywanych na horyzoncie predykcji

$$y^0(k+p|k) = - \sum_{i=1}^{\min\{n_A, p-1\}} a_i y^0(k+p-i|k) - \sum_{i=\min\{n_A, p-1\}+1}^{n_A} a_i y(k+p-i) \\ + \sum_{i=0}^{\min\{n_B, p\}} b_i u(k-1) + \sum_{i=\min\{n_B, p\}+1}^{n_B} b_i u(k-1+p-i) + d(k), \quad (3.89)$$

$p = 1, 2, \dots, N$ , gdzie wartość zakłócenia  $d(k)$  dana jest zależnością (3.88).

Zauważmy, że wzór (3.89) jest rekurencyjny. Nie podaje on ogólnej, jawnej postaci zależności trajektorii swobodnej od przeszłych wyjść i sterowań, czyli nie może być bezpośrednio wykorzystany do formułowania prawa regulacji. Może być natomiast użyty przy implementacji numerycznej wersji algorytmu GPC, gdzie w każdej chwili próbkowania wyliczana jest odpowiedź swobodna i rozwiązywane zadanie optymalizacji kwadratowej. Możliwe jest również oczywiście wyznaczenie jawnej zależności trajektorii swobodnej od przeszłych wyjść i sterowań poprzez kolejne wyeliminowanie rekurencji z wzoru (3.89). Pokazujemy to w przykładach poniżej.

### Przykład 3.3.

Dla obiektu z przykładu 3.2 pokażemy wyznaczanie wzorów na odpowiedź swobodną wykorzystując zależności (3.89).

Model dany jest zależnością (3.83), tzn.

$$a_1 = -0.2676, \quad n_A = 1, \\ b_0 = b_1 = 0, \quad b_2 = 0.1989, \quad b_3 = 0.2552, \quad n_B = 3.$$

Z (3.88) mamy

$$d(k) = y(k) + a_1 y(k-1) - b_2 u(k-3) - b_3 u(k-4).$$

Z wzoru (3.89) dostajemy natomiast, dla  $p = 1$

$$y^0(k+1|k) = -a_1 y(k) + b_2 u(k-2) + b_3 u(k-3) + \\ + y(k) + a_1 y(k-1) - b_2 u(k-3) - b_3 u(k-4) \\ = 1.2676y(k) - 0.2676y(k-1) + \\ + 0.1989u(k-2) + 0.0563u(k-3) - 0.2552u(k-4).$$

Dla  $p = 2$  z (3.89) mamy

$$y^0(k+2|k) = -a_1 y^0(k+1|k) + b_2 u(k-1) + b_3 u(k-2) + \\ + y(k) + a_1 y(k-1) - b_2 u(k-3) - b_3 u(k-4).$$

Wstawiając do tej zależności uzyskany uprzednio wzór na  $y^0(k+1|k)$  dostajemy

$$y^0(k+2|k) = 1.3392y(k) - 0.3392y(k-1) + \\ + 0.1989u(k-1) + 0.3084u(k-2) - 0.1838u(k-3) - 0.3235u(k-4).$$

Analogicznie, dla  $p=3$  mamy

$$y^0(k+3|k) = -a_1y^0(k+2|k) + b_2u(k-1) + b_3u(k-1) + \\ + y(k) + a_1y(k-1) - b_2u(k-3) - b_3u(k-4),$$

i wstawiając do tej zależności uzyskany uprzednio wzór na  $y^0(k+2|k)$  dostajemy

$$y^0(k+3|k) = 1.3584y(k) - 0.3584y(k-1) + \\ + 0.5073u(k-1) + 0.0825u(k-2) - 0.2481u(k-3) + 0.3418u(k-4).$$

Jak łatwo porównać, uzyskane wyrażenie jest równoważne wzorowi obliczonemu w przykładzie 3.2, danemu pierwszym wierszem uzyskanej tam zależności macierzowej określającej  $\mathcal{Y}^0(k)$ . Wyznaczenie pozostałych wzorów, dla  $y^0(k+4|k)$ ,  $y^0(k+5|k)$  i  $y^0(k+6|k)$ , wykonuje się w analogiczny sposób, obliczenia pozostawiamy czytelnikowi.  $\square$

### 3.3.3. Algorytm GPC dla obiektu wielowymiarowego

W przypadku obiektu wielowymiarowego, o  $n_u$  wejściach sterowanych i  $n_y$  wyjściach regulowanych, model obiektu dany jest wzorem (3.64)

$$\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})u(k-1) + \frac{\varepsilon(k)}{\Delta}. \quad (3.90)$$

Oparta na wykorzystaniu tożsamości Bézout analiza prowadząca do wyznaczenia wzorów na predykcje i prawa regulacji regulatora GPC, przeprowadzona dla przypadku jednowymiarowego, przenosi się bezpośrednio na obiekt wielowymiarowy. Tożsamość Bézout ma teraz postać macierzową (porównaj z (3.66))

$$\mathbf{E}_p(z^{-1})\overline{\mathbf{A}}(z^{-1}) + \mathbf{F}_p(z^{-1})z^{-p} = \mathbf{1},$$

gdzie  $\overline{\mathbf{A}}(z^{-1}) = \Delta\mathbf{A}(z^{-1})$ . Rozumując analogicznie jak w przypadku jednowymiarowym dostajemy wzory na predykcje wyjść

$$y(k+p|k) = \mathbf{F}_p(z^{-1})y(k) + \mathbf{E}_p(z^{-1})\mathbf{B}(z^{-1})\Delta u(k+p-1|k) \\ = \mathbf{F}_p(z^{-1})y(k) + \mathbf{G}_p(z^{-1})\Delta u(k+p-1|k),$$

przy czym  $\mathbf{G}_p(z^{-1})\Delta u(k+p-1|k)$  można rozbić na składniki zależne od aktualnego i przyszłych przyrostów sterowań (decyzje) i składniki zależne od poprzednich przyrostów sterowań, otrzymując wzory na predykcję wyjść w postaci identycznej z (3.72) i (3.73),

$$\Delta \mathcal{Y}(k) = \mathbf{M} \Delta \mathcal{U}(k), \quad (3.91)$$

$$\mathcal{Y}^0(k) = \mathbf{F} \mathcal{Y}^{PG}(k) + \mathbf{G}^{PG} \Delta \mathcal{U}^{PG}(k). \quad (3.92)$$

Teraz  $\Delta \mathcal{Y}(k)$ ,  $\mathcal{Y}^0(k)$  itd. składają się z odpowiadających poszczególnym chwilom wektorów, a nie skalarów. Struktura macierzy  $\mathbf{F}$  też jest analogiczna, opisana tezą lematu 3.1, skąd

$$\mathbf{F} \mathcal{Y}^{PG}(k) = \mathbf{I} y(k) + \bar{\mathbf{F}} \Delta \mathcal{Y}^{PG}(k).$$

Struktura analitycznych praw regulacji też będzie taka sama. Rysunki ilustrujące tę strukturę będą stąd analogiczne do rysunków (3.22) i (3.23) – tylko we wzorach skalarne współczynniki wzmocnień regulatora trzeba zastąpić macierzowymi, podobnie jak i bloczki dyskretnego integratora i opóźnienia (porównaj z odpowiednimi rysunkami dla regulatora DMC, sporządzonymi od razu dla sytuacji wielowymiarowej).

Szczegółowe przedstawienie i dyskusja regulacji GPC dla obiektu wielowymiarowego w ujęciu macierzowym wykracza poza ramy niniejszej pracy, tym bardziej, że mamy tu wspomnianą wyżej analogię z przypadkiem jednowymiarowym. Zainteresowanego czytelnika odsyłamy przede wszystkim do książki [17], poświęconej głównie algorytmowi GPC. Nie znajdziemy tam jednakże prawa regulacji GPC danego wzorami (3.81) czy (3.82), jak i zależności na bezpośrednie, rekurencyjne wyliczanie elementów składowych wielowymiarowej odpowiedzi swobodnej, które formułujemy poniżej.

Zakładając model zakłóceń typu DMC dla każdego z wyjść obiektu, można prosto uogólnić na przypadek wielowymiarowy wyznaczone bezpośrednio z równań modelu zależności (3.85) i (3.89), określając w ten sposób elementy odpowiedzi skokowych i (rekurencyjnie) składowych swobodnych trajektorii wyjść. Rozważać będziemy macierz  $\mathbf{A}(z^{-1})$  diagonalną, oznaczając przez  $A^m(z^{-1})$  i  $B^{m,j}(z^{-1})$  wielomiany składowe macierzy wielomianowych  $\mathbf{A}(z^{-1})$  i  $\mathbf{B}(z^{-1})$  definiujące model odpowiedzi  $m$ -tego wyjścia na  $n_u$  sterowań

$$y_m(k) = - \sum_{i=1}^{n_A} a_i^m y_m(k-i) + \sum_{j=1}^{n_u} \sum_{i=0}^{n_B} b_i^{m,j} u_j(k-1-i), \quad (3.93)$$

$m = 1, \dots, n_y$  (założyliśmy tu, że każdy z wielomianów jest tego samego stopnia,  $n_A^m = n_A$ ,  $n_B^{m,j} = n_B$ ). Z zależności powyższej dostajemy natychmiast wzór na elementy odpowiedzi skokowej, dla każdej pary wejście ( $j$ -te) – wyjście ( $m$ -te) ,

$$s_k^{m,j} = - \sum_{i=1}^{\min\{k-1, n_A\}} a_i^m s_{k-i}^{m,j} + \sum_{i=0}^{\min\{k-1, n_B\}} b_i^{m,j}, \quad (3.94)$$

$m = 1, \dots, n_y$ ,  $j = 1, \dots, n_u$  (porównaj z (3.85)). Znając elementy wielowymiarowej odpowiedzi skokowej możemy od razu sformułować postać macierzy dynamicznej określającej trajektorie wymuszane wyjść na horyzoncie predykcji, zob. rozdz. 3.2.2.

Rozumując analogicznie jak w przypadku jednowymiarowym wyznaczamy też, będące odpowiednikiem wzoru (3.89), zależności rekurencyjne na elementy składowych swobodnych trajektorii wyjść prognozowanych

$$\begin{aligned} y_m^0(k+p|k) = & - \sum_{i=1}^{\min\{n_A, p-1\}} a_i^m y_m^0(k+p-i|k) + \\ & - \sum_{i=\min\{n_A, p-1\}+1}^{n_A} a_i^m y_m(k+p-i) + \sum_{j=1}^{n_u} \left[ \sum_{i=0}^{\min\{n_B, p\}} b_i^{m,j} u_j(k-1) + \right. \\ & \left. + \sum_{i=\min\{n_B, p\}+1}^{n_B} b_i^{m,j} u_j(k-1+p-i) \right] + d_m(k), \quad (3.95) \end{aligned}$$

$p = 1, \dots, N$ , gdzie

$$d_m(k) = y_m(k) - \left[ - \sum_{i=1}^{n_A} a_i^m y_m(k-i) + \sum_{j=1}^{n_u} \sum_{i=0}^{n_B} b_i^{m,j} u_j(k-1-i) \right]$$

są addytywnymi zakłóceniami typu DMC wyliczonymi w chwili  $k$ ,  $m = 1, \dots, n_y$ .

Zdefiniowany powyżej sposób postępowania zilustrujemy w przykładzie 3.4 zamieszczonym w rozdziale 3.3.4, pokazującym jednocześnie działanie algorytmu predykcyjnego GPC w wersji zarówno analitycznej, jak i numerycznej, dla obiektu dwuwymiarowego.

### 3.3.4. Wersja numeryczna algorytmu GPC

Sformułowania zadania optymalizacji kwadratowej dla różnych algorytmów predykcyjnych z modelem liniowym obiektu różnią się przede wszystkim sposobami wyliczania wartości wyjść regulowanych przewidywanych na horyzont predykcji, w tym głównie składowej swobodnej  $\mathcal{Y}^0(k)$  trajektorii wyjść. Tym też różni się zadanie optymalizacji kwadratowej algorytmu GPC, w stosunku do analogicznego zadania (3.51) dla algorytmu DMC.

Zadanie optymalizacji kwadratowej rozwiązywane w każdym kroku algorytmu GPC w wersji numerycznej ma postać

$$\begin{aligned} \min_{\Delta \mathcal{U}(k)} & \left\{ \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] - \mathbf{M} \Delta \mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \|\Delta \mathcal{U}(k)\|_{\underline{\Lambda}}^2 \right\} \\ \text{z ogr. : } & -\Delta \mathcal{U}_{\max} \leq \Delta \mathcal{U}(k) \leq \Delta \mathcal{U}_{\max}, \\ & \mathcal{U}_{\min} \leq \mathcal{U}(k-1) + \mathbf{J} \Delta \mathcal{U}(k) \leq \mathcal{U}_{\max}, \\ & \mathcal{Y}_{\min} \leq \mathcal{Y}^0(k) + \mathbf{M} \Delta \mathcal{U}(k) \leq \mathcal{Y}_{\max}, \end{aligned} \quad (3.96)$$

gdzie odpowiedź swobodna  $\mathcal{Y}^0(k)$  wyliczna jest z zależności (3.92)

$$\mathcal{Y}^0(k) = \mathbf{F} \mathcal{Y}^{PG}(k) + \mathbf{G}^{PG} \Delta \mathcal{U}^{PG}(k).$$

Wszystkie wielkości występujące w powyższym sformułowaniu są wprowadzonymi uprzednio w rozdziale 3.2.3 przy omawianiu algorytmu numerycznego DMC. Również podana tam dyskusja dotycząca formułowania zadania optymalizacji kwadratowej w postaci standardowej, algorytmów jego rozwiązywania, dopuszczalności rozwiązań itp. pozostaje oczywiście w mocy.

#### Przykład 3.4.

Dla zilustrowania wielowymiarowego algorytmu GPC, zarówno w wersji analitycznej, jak i numerycznej, przytoczmy wyniki symulacji dla dwuwymiarowego modelu liniowego danego macierzą transmitancji

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+0.7s} & \frac{5}{1+0.3s} \\ \frac{1}{1+0.5s} & \frac{2}{1+0.4s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix},$$

reprezentującego małosygnalowy model reaktora przepływowego w punkcie pracy ([17], str. 143).

Wykorzystując funkcję „c2d” w Matlabie, przyjmując okres próbkowania  $T_p = 0.03$ , uzyskujemy dyskretną reprezentację modelu

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \frac{0.041951z^{-1}}{1-0.958048z^{-1}} & \frac{0.475812z^{-1}}{1-0.904837z^{-1}} \\ \frac{0.058235z^{-1}}{1-0.941764z^{-1}} & \frac{0.144513z^{-1}}{1-0.927743z^{-1}} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}. \quad (3.97)$$

Powyższy model transmitancyjny MIMO ( $2 \times 2$ ) zapiszemy w postaci dwóch modeli MISO:

$$y_1(k) = \frac{0.041951z^{-1}}{1 - 0.958048z^{-1}}u_1(k) + \frac{0.475812z^{-1}}{1 - 0.904837z^{-1}}u_2(k), \quad (3.98)$$

$$y_2(k) = \frac{0.058235z^{-1}}{1 - 0.941764z^{-1}}u_1(k) + \frac{0.144513z^{-1}}{1 - 0.927743z^{-1}}u_2(k). \quad (3.99)$$

Sprowadzając prawą stronę każdego z powyższych równań do wspólnego mianownika dostajemy:

$$y_1(k) = \frac{0.041951z^{-1}(1 - 0.904837z^{-1})u_1(k)}{(1 - 0.958048z^{-1})(1 - 0.904837z^{-1})} + \frac{0.475812z^{-1}(1 - 0.958048z^{-1})u_2(k)}{(1 - 0.958048z^{-1})(1 - 0.904837z^{-1})},$$

$$y_2(k) = \frac{0.058235z^{-1}(1 - 0.927743z^{-1})u_1(k)}{(1 - 0.941764z^{-1})(1 - 0.927743z^{-1})} + \frac{0.144513z^{-1}(1 - 0.941764z^{-1})u_2(k)}{(1 - 0.941764z^{-1})(1 - 0.927743z^{-1})},$$

co można równoważnie zapisać w postaci dwóch równań opisujących dyskretne modele MISO:

$$(1 - 1.862885z^{-1} + 0.866877z^{-2})y_1(k) = (0.041951 + 0.037959z^{-1})u_1(k-1) + (0.475812 - 0.455851z^{-1})u_2(k-1), \quad (3.100)$$

$$(1 - 1.869508z^{-1} + 0.873715z^{-2})y_2(k) = (0.058235 + 0.054027z^{-1})u_1(k-1) + (0.144513 - 0.136097z^{-1})u_2(k-1). \quad (3.101)$$

Modele te można przedstawić w jednym modelu postaci MIMO  $\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})z^{-1}u(k)$ , gdzie:

$$\mathbf{A}(z^{-1}) = \begin{bmatrix} 1 + a_1^1 z^{-1} + a_2^1 z^{-2} & 0 \\ 0 & 1 + a_1^2 z^{-1} + a_2^2 z^{-2} \end{bmatrix}$$

$$= \begin{bmatrix} 1 - 1.862885z^{-1} + 0.866877z^{-2} & 0 \\ 0 & 1 - 1.869508z^{-1} + 0.873715z^{-2} \end{bmatrix},$$

$$\begin{aligned}
\mathbf{B}(z^{-1}) &= \begin{bmatrix} b_0^{1,1} + b_1^{1,1}z^{-1} & b_0^{1,2} + b_1^{1,2}z^{-1} \\ b_0^{2,1} + b_1^{2,1}z^{-1} & b_0^{2,2} + b_1^{2,2}z^{-1} \end{bmatrix} \\
&= \begin{bmatrix} 0.041951 - 0.037959z^{-1} & 0.475812 - 0.455851z^{-1} \\ 0.058235 - 0.054027z^{-1} & 0.144513 - 0.136097z^{-1} \end{bmatrix}.
\end{aligned} \tag{3.102}$$

Przyjęto wartości horyzontów: predykcji  $N = 3$  i sterowania  $N_u = 2$ , a także  $N_1 = 1$  (opóźnienia nie występują). Korzystając z wzoru (3.94) wyznaczamy elementy odpowiedzi skokowych, a stąd macierz dynamiczną

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{S}_2 & \mathbf{S}_1 \\ \mathbf{S}_3 & \mathbf{S}_2 \end{bmatrix} = \begin{bmatrix} 0.04195 & 0.47581 & 0 & 0 \\ 0.05824 & 0.14451 & 0 & 0 \\ 0.08214 & 0.90635 & 0.04195 & 0.47581 \\ 0.11308 & 0.27858 & 0.05824 & 0.14451 \\ 0.12065 & 1.29591 & 0.08214 & 0.90635 \\ 0.16473 & 0.40297 & 0.11308 & 0.27858 \end{bmatrix}.$$

Wzór na predykcję,  $\mathcal{Y}(k) = \mathcal{Y}^0(k) + \mathbf{M}\Delta\mathcal{U}(k)$ , przyjmuje w rozważanym przypadku postać

$$\begin{bmatrix} y_1(k+1|k) \\ y_2(k+1|k) \\ y_1(k+2|k) \\ y_2(k+2|k) \\ y_1(k+3|k) \\ y_2(k+3|k) \end{bmatrix} = \begin{bmatrix} y_1^0(k+1|k) \\ y_2^0(k+1|k) \\ y_1^0(k+2|k) \\ y_2^0(k+2|k) \\ y_1^0(k+3|k) \\ y_2^0(k+3|k) \end{bmatrix} + \mathbf{M} \begin{bmatrix} \Delta u_1(k|k) \\ \Delta u_2(k|k) \\ \Delta u_1(k+1|k) \\ \Delta u_2(k+1|k) \end{bmatrix}.$$

Składową swobodną  $\mathcal{Y}^0(k)$  będziemy liczyć w sposób rekurencyjny wykorzystując wzory (3.95):

$$\begin{aligned}
y_1^0(k+1|k) &= -a_1^1 y_1(k) - a_2^1 y_1(k-1) + b_0^{1,1} u_1(k-1) + \\
&\quad + b_1^{1,1} u_1(k-1) + b_0^{1,2} u_2(k-1) + b_1^{1,2} u_2(k-1) + d_1(k),
\end{aligned}$$

$$\begin{aligned}
y_2^0(k+1|k) &= -a_1^2 y_2(k) - a_2^2 y_2(k-1) + b_0^{2,1} u_1(k-1) + \\
&\quad + b_1^{2,1} u_1(k-1) + b_0^{2,2} u_2(k-1) + b_1^{2,2} u_2(k-1) + d_2(k),
\end{aligned}$$

$$y_1^0(k+2|k) = -a_1^1 y_1^0(k+1|k) - a_2^1 y_1(k) + b_0^{1,1} u_1(k-1) + \\ + b_1^{1,1} u_1(k-1) + b_0^{1,2} u_2(k-1) + b_1^{1,2} u_2(k-1) + d_1(k),$$

itd. aż do:

$$y_2^0(k+3|k) = -a_1^2 y_2^0(k+2|k) - a_2^2 y_2^0(k+1|k) + b_0^{2,1} u_1(k-1) + \\ + b_1^{2,1} u_1(k-1) + b_0^{2,2} u_2(k-1) + b_1^{2,2} u_2(k-1) + d_2(k),$$

gdzie stałe na horyzoncie predykcji zakłócenia są różnicą między wyjściami (zmiierzonymi) obiektu w chwili  $k$  a wyjściami modelu na chwilę  $k$  obliczonymi w chwili  $k-1$ :

$$d_1(k) = y_1(k) - [-a_1^1 y_1(k-1) - a_2^1 y_1(k-2) + \\ + b_0^{1,1} u_1(k-1) + b_1^{1,1} u_1(k-2) + b_0^{1,2} u_2(k-1) + b_1^{1,2} u_2(k-2)],$$

$$d_2(k) = y_2(k) - [-a_1^2 y_2(k-1) - a_2^2 y_2(k-2) + \\ + b_0^{2,1} u_1(k-1) + b_1^{2,1} u_1(k-2) + b_0^{2,2} u_2(k-1) + b_1^{2,2} u_2(k-2)].$$

Przyjęto  $\Psi = \mathbf{I}$  i  $\Lambda = \lambda \mathbf{I}$ , przy  $\lambda = 0.075$ . Badanie symulacyjne algorytmu przeprowadzono dla dwóch sytuacji:

1. Algorytm analityczny GPC, bez uwzględniania ograniczeń amplitud i szybkości zmian sterowań (zarówno w algorytmie, jak i w obiekcie). W tym przypadku wektor rozwiązania zadania optymalizacji wyznaczony jest analitycznie ze wzoru

$$\Delta \hat{\mathbf{U}}(k) = \mathbf{K}[\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)],$$

gdzie

$$\mathbf{K} = (\mathbf{M}^T \mathbf{M} + \lambda \mathbf{I})^{-1} \mathbf{M}^T \\ = \begin{bmatrix} -0.04364 & 0.45453 & -0.20244 & 0.76346 & -0.32797 & 1.05657 \\ 0.90230 & 0.22863 & 0.44040 & 0.04689 & 0.02006 & -0.12258 \\ -0.12893 & -0.11711 & -0.15232 & 0.35308 & -0.15013 & 0.79786 \\ -1.27727 & -0.39466 & -0.19277 & -0.11970 & 0.78629 & 0.13457 \end{bmatrix},$$

zaś do sterowania w chwili  $k$  wykorzystywane są, oczywiście, dwa pierwsze elementy tego wektora,  $\Delta \hat{u}_1(k) = \Delta \hat{u}_1(k|k)$  i  $\Delta \hat{u}_2(k) = \Delta \hat{u}_2(k|k)$ .

2. Algorytm numeryczny GPC, przy ograniczeniach amplitud i szybkości zmian sygnałów sterowań

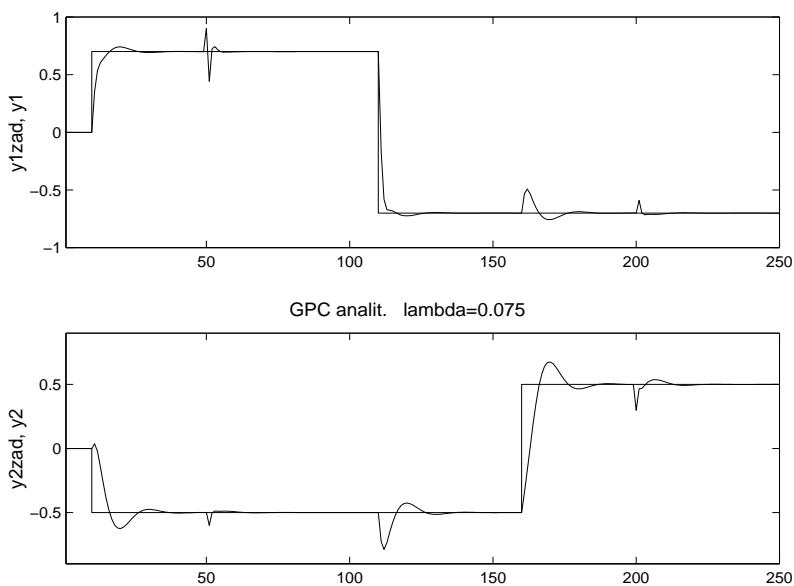
$$\begin{aligned} u_{1 \min} &= -2.5, \\ u_{1 \max} &= 2.5, \\ \Delta u_{1 \max} &= 0.5, \\ u_{2 \min} &= -0.6, \\ u_{2 \max} &= 0.6, \\ \Delta u_{2 \max} &= 0.3, \end{aligned}$$

gdzie w każdym kroku rozwiązywane jest zadanie optymalizacji kwadratowej.

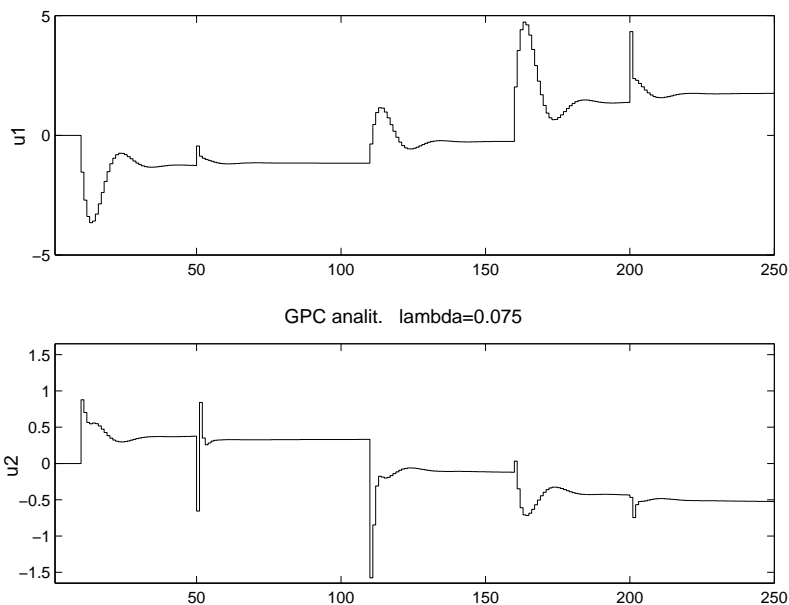
Wyniki symulacji z regulatorem analitycznym przedstawiono na rysunkach 3.34 (trajektorie wielkości regulowanych) i 3.35 (trajektorie sygnałów sterujących), natomiast wyniki symulacji z regulatorem numerycznym, odpowiednio, na rysunkach 3.36 i 3.37. Przebiegi pokazane są na odcinku czasu odpowiadającym 250 okresom próbkowania, przy czym w chwilach 10, 110 i 160 następują skoki wartości zadanych, zaś w chwilach 50 i 200 skoki zakłócenia niemierzalnego o amplitudzie 0.2, odpowiednio na wyjściu 1 i wyjściu 2.

W obu sytuacjach widać dobre własności odsprzęgające regulatora predykcyjnego. W przypadku regulatora analitycznego szybszemu nadążaniu za zmianami wartości zadanych towarzyszą większe (bo nie ograniczane) amplitudy i zmiany sygnałów sterujących. W sytuacji z ograniczeniami przebiegi sterowania są znacznie spokojniejsze. Oczywiście, mniejszym (i optymalnym) amplitudom i szybkościom zmian sygnałów sterujących odpowiada wolniejsze nadążanie za zmianami wartości zadanych i wolniejsze tłumienie wpływu zakłóceń.

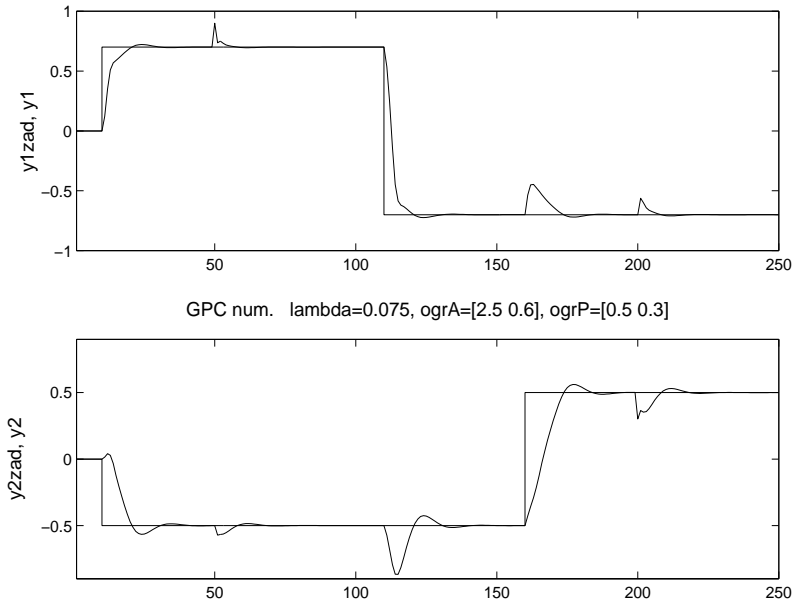
Zauważmy na koniec, że przedstawiona w niniejszym rozdziale procedura uzyskiwania modelu MIMO postaci  $\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})z^{-1}u(k)$  z macierzą  $\mathbf{A}(z^{-1})$  diagonalną, tzn. składającego się z  $n_y$  modeli MISO, prowadzi w przypadku wielu wejść sterujących obiektu do wysokich rzędów takich modeli. Każdy z wielomianów  $A_i(z^{-1})$  znajdujących się na diagonalu macierzy  $\mathbf{A}(z^{-1})$ ,  $i = 1, \dots, n_y$ , może być w ogólności rzędu będącego sumą rzędów wielomianów znajdujących się w mianownikach transmitancji tworzących  $i$ -ty wiersz oryginalnej dyskretnej macierzy transmitancji obiektu (jak w przykładzie powyżej). Może to prowadzić do poważnych kłopotów przy realizacji równań predykcyjnych. Stąd w przypadkach takich zaleca się, na potrzeby projektowania regulatora GPC, wstępne uproszczenie poszczególnych elementów oryginalnej macierzy transmitancji do postaci zastępczych jak najniższego rzędu.  $\square$



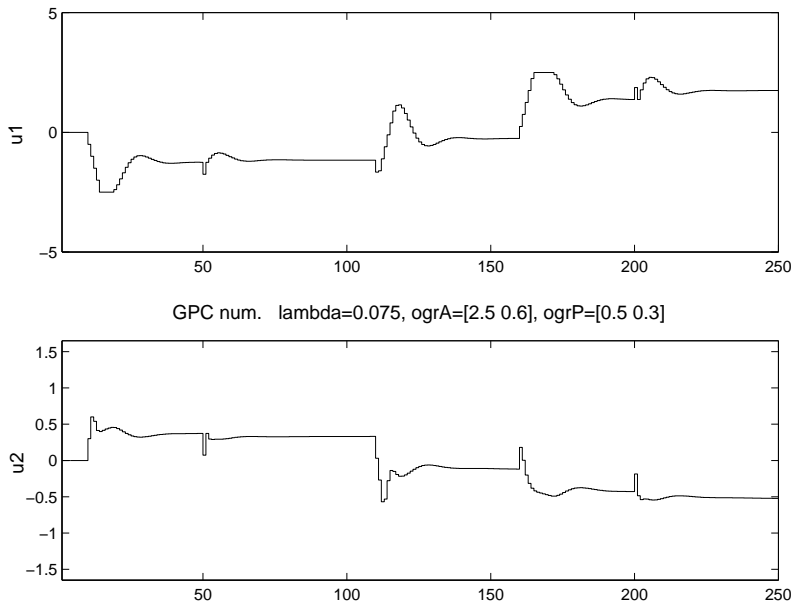
Rysunek 3.34. Trajektorie wielkości regulowanych w układzie z regulatorem analitycznym (sytuacja bez ograniczeń)



Rysunek 3.35. Trajektorie sterowań w układzie z regulatorem analitycznym



Rysunek 3.36. Trajektorie wielkości regulowanych w układzie z regulatorem numerycznym (sytuacja z ograniczeniami)



Rysunek 3.37. Trajektorie sterowań w układzie z regulatorem numerycznym

### 3.4. Algorytmy MPC – z modelem w postaci liniowych równań stanu

Pierwsze generacje algorytmów regulacji predykcyjnej operowały modelami obiektów w postaci odpowiedzi impulsowych i skokowych czy, nieco później, modelem w postaci dyskretnych równań różnicowych (algorytm GPC) – a więc modelami typu wejście-wyjście (zob. rozdz. 3.1). Modele takie dominują w zastosowaniach typu regulacji przemysłowej (*process control*), a więc w obszarze, gdzie narodziły się i zostały z dużym powodzeniem zastosowane algorytmy MPC. Algorytmy te, szczególnie w wersji numerycznej zakładającej rozwiązywanie w każdym kroku (okresie próbkowania) zadania programowania kwadratowego numeryczną procedurą optymalizacyjną, wymagają stosunkowo dużego jak na regulację nakładu obliczeń. Stąd naturalne, najłatwiejsze było zastosowanie algorytmów MPC do regulacji nadrzędnej procesów przemysłowych, charakteryzującej się z reguły długimi okresami próbkowania (zob. rozdz. 1). Poza tym, pojawienie się i rozpowszechnienie w instalacjach przemysłowych elektronicznych, rozproszonych systemów sterowania (DCS) stworzyło naturalne środowisko dla realizacji złożonych obliczeniowo algorytmów nadrzędnego sterowania, takich jak nadrzędna regulacja predykcyjna i optymalizacja punktów pracy.

Zaproponowanie algorytmów regulacji predykcyjnej z modelem obiektu w postaci układu liniowych równań stanu nastąpiło stosunkowo późno, na przełomie lat osiemdziesiątych i dziewięćdziesiątych – kiedy znacznie wzrosły moce obliczeniowe i niezawodność komputerowego sprzętu sterującego, a jednocześnie drastycznie spadły jego ceny i wymiary. Możliwe stało się zastosowanie algorytmów MPC do regulacji bezpośredniej obiektów mniejszych, wyodrębnionych, często dobrze modelowanych właśnie równaniami stanu, jak np. obiekty elektromechaniczne. Ponadto opis obiektu w postaci równań stanu okazał się wygodniejszy dla szeregu analiz teoretycznych. Lata dziewięćdziesiąte ubiegłego stulecia to rozwój i wzrost popularności algorytmów predykcyjnych opartych na równaniach stanu. Np. wypuszczony na rynek po raz pierwszy w roku 1995 przybornik *Model Predictive Control Toolbox* do pakietu MATLAB zawierał procedury do tworzenia i badania algorytmów MPC oparte na dwóch postaciach modelu obiektu: odpowiedzi skokowej lub modelu danego układem liniowych równań stanu [104].

Zastosowanie modelu obiektu w postaci równań stanu jest podejściem bardziej ogólnym, wygodnym do analizy teoretycznej algorytmów MPC [87], w sposób naturalny umożliwiającym przejście do modeli nieliniowych. Natomiast w wielu zastosowaniach przemysłowych, gdzie dominuje modelowanie typu wejście-wyjście, formułowanie modelu w przestrzeni stanów może być nienaturalne, szczególnie w typowej sytuacji niedostępności po-

miarowej pełnego wektora stanu. Ponadto, algorytm predykcji oparty na opisie obiektu układem liniowych równań stanu ma niezbyt dobre cechy numeryczne w przypadkach długich horyzontów predykcji i dużej wymiarowości wektora stanu w stosunku do wektora wyjść regulowanych. Dla zwiezłości zapisu algorytm ten będzie dalej określany jako *algorytm MPSC* (*Model Predictive Control with State-space model*).

### 3.4.1. Algorytm MPSC z pomiarem wektora stanu

Rozważmy następującą postać równań stanu modelu obiektu

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k) + v(k), \quad (3.103)$$

$$y(k) = \mathbf{C}x(k), \quad (3.104)$$

gdzie  $x$  jest *wektorem stanu* o wymiarze  $n_x$ ,  $y$  jest *wektorem wyjść regulowanych* o wymiarze  $n_y$ , zaś  $v(k)$  jest *wektorem estymaty zakłóceń stanu* o wymiarze  $n_v = n_x$ . Zakładamy, że jest to estymata zakłóceń niemierzalnych, reprezentujących również błędy modelowania. Do obliczania tej estymaty zastosujemy zależność

$$v(k) = x(k) - [\mathbf{A}x(k-1) + \mathbf{B}u(k-1)], \quad (3.105)$$

$$v(k) = v(k+1|k) = \dots = v(k+N-1|k), \quad (3.106)$$

tzn. zakładamy brak informacji o zmianach zakłóceń niemierzalnych stanu na horyzoncie predykcji, a stąd stałość estymaty tych zakłóceń na horyzoncie predykcji. Prezentowany sposób modelowania zakłóceń niemierzalnych w algorytmach MPC z modelem w postaci równań stanu został wprowadzony przez autora w monografii [144], jest on prosty w realizacji i okazał się skuteczny w praktyce.

Wyprowadzimy teraz wzory określające przewidywane wartości zmiennych stanu zakładając (w tym rozdziale) dostępność pomiaru pełnego wektora stanu. Oznaczając przez  $x(k+p|k)$  stan przewidywany w chwili  $k$  na chwilę  $k+p$  horyzontu predykcji i postępując rekurencyjnie, dostajemy

$$\begin{aligned} x(k+1|k) &= \mathbf{A}x(k) + \mathbf{B}u(k|k) + v(k) \\ &= \mathbf{A}x(k) + \mathbf{B}(\Delta u(k|k) + u(k-1)) + v(k), \end{aligned}$$

$$\begin{aligned} x(k+2|k) &= \mathbf{A}x(k+1|k) + \mathbf{B}(\Delta u(k+1|k) + \Delta u(k|k) + u(k-1)) + v(k) \\ &= \mathbf{A}^2x(k) + \mathbf{A}\mathbf{B}(\Delta u(k|k) + u(k-1)) + \mathbf{A}v(k) \\ &\quad + \mathbf{B}(\Delta u(k+1|k) + \Delta u(k|k) + u(k-1)) + v(k) \end{aligned}$$

$$= \mathbf{A}^2 x(k) + (\mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + \mathbf{B}\Delta u(k+1|k) + \\ + (\mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) + (\mathbf{A} + \mathbf{I})v(k),$$

$$x(k+3|k) = \mathbf{A}x(k+2|k) + \mathbf{B}(\Delta u(k+2|k) + \Delta u(k+1|k) + \\ + \Delta u(k|k) + u(k-1)) + v(k) \\ = \mathbf{A}^3 x(k) + (\mathbf{A}^2 + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + (\mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+1|k) + \\ + \mathbf{B}\Delta u(k+2|k) + (\mathbf{A}^2 + \mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) + (\mathbf{A}^2 + \mathbf{A} + \mathbf{I})v(k), \\ \vdots$$

$$x(k+N_u|k) = \mathbf{A}x(k+N_u-1|k) + \mathbf{B}(\Delta u(k+N_u-1|k) + \dots + \\ + \Delta u(k|k) + u(k-1)) + v(k) \\ = \mathbf{A}^{N_u} x(k) + (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + \\ + (\mathbf{A}^{N_u-2} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+1|k) + \\ + \dots + \mathbf{B}\Delta u(k+N_u-1|k) + \\ + (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) + (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})v(k),$$

$$x(k+N_u+1|k) = \mathbf{A}x(k+N_u|k) + \mathbf{B}(\Delta u(k+N_u-1|k) + \dots + \\ + \Delta u(k|k) + u(k-1)) \\ = \mathbf{A}^{N_u+1} x(k) + (\mathbf{A}^{N_u} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + \\ + (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+1|k) + \\ + \dots + (\mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+N_u-1|k) + \\ + (\mathbf{A}^{N_u} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) + (\mathbf{A}^{N_u} + \dots + \mathbf{A} + \mathbf{I})v(k), \\ \vdots$$

$$x(k+N|k) = \mathbf{A}^N x(k) + (\mathbf{A}^{N-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + \\ + (\mathbf{A}^{N-2} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+1|k) + \dots + \\ + (\mathbf{A}^{N-N_u} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+N_u-1|k) + \\ + (\mathbf{A}^{N-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) + (\mathbf{A}^{N-1} + \dots + \mathbf{A} + \mathbf{I})v(k).$$

Definiując wektor  $\mathcal{X}(k)$  i przypominając definicję wektora  $\Delta\mathcal{U}(k)$

$$\mathcal{X}(k) = \begin{bmatrix} x(k+1|k) \\ \vdots \\ x(k+N|k) \end{bmatrix}, \quad \Delta\mathcal{U}(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k+N_u-1|k) \end{bmatrix},$$

uzyskane zależności można przedstawić w postaci macierzowej:

$$\begin{aligned}
 \mathcal{X}(k) = & \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^{N_u} \\ \mathbf{A}^{N_u+1} \\ \vdots \\ \mathbf{A}^N \end{bmatrix} x(k) + \\
 & + \begin{bmatrix} \mathbf{B} & \cdots & \mathbf{0} \\ (\mathbf{A} + \mathbf{I})\mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ (\mathbf{A}^{N_u-1} + \cdots + \mathbf{A} + \mathbf{I})\mathbf{B} & \cdots & \mathbf{B} \\ (\mathbf{A}^{N_u} + \cdots + \mathbf{A} + \mathbf{I})\mathbf{B} & \cdots & (\mathbf{A} + \mathbf{I})\mathbf{B} \\ \vdots & \ddots & \vdots \\ (\mathbf{A}^{N-1} + \cdots + \mathbf{A} + \mathbf{I})\mathbf{B} & \cdots & (\mathbf{A}^{N-N_u} + \cdots + \mathbf{A} + \mathbf{I})\mathbf{B} \end{bmatrix} \Delta \mathcal{U}(k) + \\
 & + \begin{bmatrix} \mathbf{I} \\ \mathbf{A} + \mathbf{I} \\ \vdots \\ \mathbf{A}^{N_u-1} + \cdots + \mathbf{A} + \mathbf{I} \\ \mathbf{A}^{N_u} + \cdots + \mathbf{A} + \mathbf{I} \\ \vdots \\ \mathbf{A}^{N-1} + \cdots + \mathbf{A} + \mathbf{I} \end{bmatrix} [\mathbf{B}u(k-1) + v(k)]. \quad (3.107)
 \end{aligned}$$

Zauważmy, że zarówno definiując wektor  $\mathcal{X}(k)$  jak i w następującym wzorze (3.107) założyliśmy  $N_1 = 1$ . Powodem jest możliwość złożonej struktury opóźnień w przypadku modelu równań stanu, modelowanej na ogół poprzez odpowiednio większy wymiar wektora stanu w modelu bez jawnych opóźnień. Jedynie w przypadku prostego, jednolitego opóźnienia  $\tau$  w sterowaniach można go łatwo jawnie uwzględnić, prowadzi to do pominięcia  $\tau$  pierwszych wierszy ( $N_1 = \tau + 1$ ) w formule (3.107).

Oznaczając teraz macierze występujące we wzorze (3.107) przez  $\tilde{\mathbf{A}}$ ,  $\mathbf{M}_x$  and  $\mathbf{V}$ , odpowiednio, możemy ten wzór zapisać w postaci

$$\mathcal{X}(k) = \tilde{\mathbf{A}}x(k) + \mathbf{M}_x \Delta \mathcal{U}(k) + \mathbf{V}[\mathbf{B}u(k-1) + v(k)]. \quad (3.108)$$

Definiując macierz  $\tilde{\mathbf{C}}$  o wymiarze  $N \cdot n_y \times N \cdot n_x$ ,

$$\tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C} \end{bmatrix},$$

uzyskujemy zależność opisującą trajektorię przewidywaną wyjść  $\mathcal{Y}^{pred}(k)$

$$\begin{aligned} \mathcal{Y}^{pred}(k) &= \tilde{\mathbf{C}}\mathcal{X}(k) \\ &= \left[ \tilde{\mathbf{C}}\tilde{\mathbf{A}}x(k) + \tilde{\mathbf{C}}\mathbf{V}[\mathbf{B}u(k-1) + v(k)] \right] + \tilde{\mathbf{C}}\mathbf{M}_x\Delta\mathcal{U}(k) \\ &= \mathcal{Y}^0(k) + \Delta\mathcal{Y}(k), \end{aligned} \quad (3.109)$$

lub równoważnie, w rozbiciu na poszczególne chwile predykcji,

$$\begin{aligned} y(k+p|k) &= \mathbf{C}\mathbf{A}^p x(k) + \mathbf{C} \left[ \sum_{j=0}^{p-1} \mathbf{A}^j \right] [\mathbf{B}u(k-1) + v(k)] + \\ &\quad + \mathbf{C} \sum_{i=0}^{p-1} \left[ \sum_{j=0}^{p-1-i} \mathbf{A}^j \right] \mathbf{B}\Delta u(k+i|k), \quad \text{for } p \leq N_u, \end{aligned}$$

$$\begin{aligned} y(k+p|k) &= \mathbf{C}\mathbf{A}^p x(k) + \mathbf{C} \left[ \sum_{j=0}^{p-1} \mathbf{A}^j \right] [\mathbf{B}u(k-1) + v(k)] + \\ &\quad + \mathbf{C} \sum_{i=0}^{N_u-1} \left[ \sum_{j=0}^{p-1-i} \mathbf{A}^j \right] \mathbf{B}\Delta u(k+i|k), \quad \text{for } p > N_u, \end{aligned}$$

gdzie  $\Delta\mathcal{Y}(k) = \tilde{\mathbf{C}}\mathbf{M}_x\Delta\mathcal{U}(k)$  jest składową wymuszaną trajektorii wyjść prognozowanych. Oczywiście, musi zachodzić równość  $\tilde{\mathbf{C}}\mathbf{M}_x = \mathbf{M}$ , gdzie  $\mathbf{M}$  jest macierzą dynamiczną (3.35). Zwróćmy uwagę, że macierz  $\tilde{\mathbf{C}}\mathbf{M}_x$  zdefiniowana jest dość skomplikowanym wzorem, a przecież jej elementy są po prostu współczynnikami odpowiedzi skokowych. Wzór (3.109) jest tym bardziej niekorzystny obliczeniowo (numerycznie), im bardziej wymiar wektora stanu  $n_x$  jest większy od wymiaru wektora wyjść regulowanych  $n_y$  oraz im dłuższy jest horyzont predykcji.

Zakładając teraz postać funkcji celu (3.36) i brak ograniczeń, dostajemy rozwiązanie zadania optymalizacji regulatora MPCs w postaci analitycznej (3.37)

$$\begin{aligned}\Delta\hat{u}(k) &= \mathbf{K}[\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] \\ &= \mathbf{K} \left[ \mathcal{Y}^{zad}(k) - \tilde{\mathbf{C}}\tilde{\mathbf{A}}x(k) - \tilde{\mathbf{C}}\mathbf{V}[\mathbf{B}u(k-1) + v(k)] \right],\end{aligned}\quad (3.110)$$

co prowadzi dalej do *analitycznego prawa regulacji MPC*

$$\Delta\hat{u}(k) = \bar{\mathbf{K}}_1 \left[ \mathcal{Y}^{zad}(k) - \tilde{\mathbf{C}}\tilde{\mathbf{A}}x(k) - \tilde{\mathbf{C}}\mathbf{V}[\mathbf{B}u(k-1) + v(k)] \right], \quad (3.111)$$

gdzie, przypomnijmy,  $\bar{\mathbf{K}}_1$  jest pierwszym (macierzowym) wierszem macierzy  $\mathbf{K}$ , zob. (3.40). Zakładając stałość wartości zadanych na horyzoncie predykcji i wykorzystując struktury macierzy  $\tilde{\mathbf{A}}$ ,  $\mathbf{V}$  i  $\tilde{\mathbf{C}}$ , prawo sterowania (3.111) może być zapisane w postaci

$$\Delta u(\hat{k}) = \sum_{p=1}^N \mathbf{K}_{1,p} \left[ y^{zad}(k) - \mathbf{C}\mathbf{A}^p x(k) - \mathbf{C}\mathbf{V}_p [\mathbf{B}u(k-1) + v(k)] \right], \quad (3.112)$$

gdzie  $\mathbf{K}_{1,p}$  są podmacierzami (w przypadku obiektów SISO skalarami) macierzy  $\bar{\mathbf{K}}_1$  (zob.(3.40)),

$$\bar{\mathbf{K}}_1 = [\mathbf{K}_{1,1} \quad \mathbf{K}_{1,2} \quad \cdots \quad \mathbf{K}_{1,N}],$$

a  $\mathbf{V}_p$  są podmacierzami macierzy  $\mathbf{V}$ ,

$$\mathbf{V}_p = \sum_{j=0}^{p-1} \mathbf{A}^j, \quad p = 1, \dots, N. \quad (3.113)$$

Wykorzystując teraz definicję  $\mathbf{K}^e$  (3.44), prawo regulacji (3.112) przekształcamy do postaci

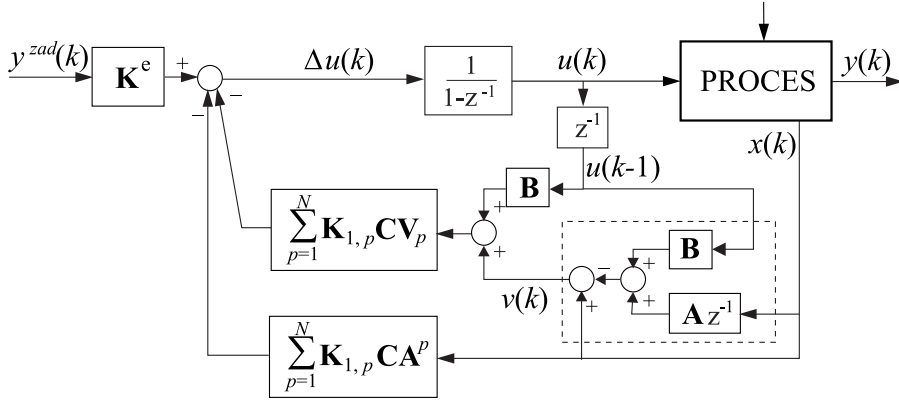
$$\Delta\hat{u}(k) = \mathbf{K}^e y^{zad}(k) - \sum_{p=1}^N \mathbf{K}_{1,p} [\mathbf{C}\mathbf{A}^p x(k) + \mathbf{C}\mathbf{V}_p [\mathbf{B}u(k-1) + v(k)]], \quad (3.114)$$

gdzie estymata zakłóceń stanu  $v(k)$  dana jest zależnością (3.105).

Struktura uzyskanego prawa sterowania MPC przedstawiona jest na rys. 3.38, gdzie część dotyczącą estymaty zakłócenia otoczono linią przerywaną.

Model procesu jest często formułowany w postaci typu ARX (równania różnicowe, transmitancje dyskretne), tak jak w algorytmie GPC. Załóżmy postać modelu tej postaci,

$$\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})u(k-1), \quad (3.115)$$



Rysunek 3.38. Struktura analitycznego regulatora MPCs z pomiarem stanu (bez elementów rzutowania sterowania na ograniczenia i anti-windup).

gdzie, przypomnijmy,  $\mathbf{A}(z^{-1})$  i  $\mathbf{B}(z^{-1})$  to macierze wielomianowe ( $\mathbf{A}(z^{-1})$  to zupełnie inna macierz niż macierz  $\mathbf{A}$  w równaniach stanu (3.103) !),

$$\mathbf{A}(z^{-1}) = \mathbf{1} + \mathbf{A}_1 z^{-1} + \mathbf{A}_2 z^{-2} + \dots + \mathbf{A}_{n_A} z^{-n_A},$$

$$\mathbf{B}(z^{-1}) = \mathbf{B}_0 + \mathbf{B}_1 z^{-1} + \mathbf{B}_2 z^{-2} + \dots + \mathbf{B}_{n_B} z^{-n_B},$$

zob. (3.63). Reprezentacja tego modelu w postaci równań stanu, ze stanem całkowicie mierzalnym, może być łatwo uzyskana, jeśli zdefiniujemy wektor stanu w postaci

$$x(k) = [y(k)^T \ y(k-1)^T \ \dots \ y(k-n_A+1)^T \ u(k-1)^T \ \dots \ u(k-n_B)^T]^T. \quad (3.116)$$

Wówczas macierze  $A, B, C$  modelu w postaci równań stanu (3.103)-(3.104) odpowiadającego definicji wektora stanu (3.116) są następujące

$$A = \begin{bmatrix} -\mathbf{A}_1 & -\mathbf{A}_2 & \dots & -\mathbf{A}_{n_A-1} & -\mathbf{A}_{n_A} & \mathbf{B}_1 & \dots & \mathbf{B}_{n_B-1} & \mathbf{B}_{n_B} \\ \mathbf{I}_y & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_y & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_y & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{I}_u & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_u & \mathbf{0} \end{bmatrix},$$

$$B = [\mathbf{B}_0^T \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0} \quad \mathbf{I}_u \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]^T,$$

$$C = [\mathbf{I}_y \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}],$$

gdzie  $\dim \mathbf{I}_y = n_y \times n_y$ ,  $\dim \mathbf{I}_u = n_u \times n_u$ . Przedstawione podejście do modelowania procesu w postaci równań stanu prowadzi zwykle do nieminimalnej reprezentacji stanu, ale unikamy konieczności konstruowania obserwatora stanu.

Zmienne wyjściowe to z reguły elementy wektora stanu, bądź kombinacje liniowe takich elementów. Równania wyjścia są w tym przypadku równaniami dokładnymi, bez błędów modelowania, a zmienne wyjściowe można traktować jako mierzone, poprzez pomiar zmiennych stanu. Jednakże występują również przypadki, kiedy równania wyjścia są nieliniowe, jak to jest np. w przypadku reaktora polimeryzacji rozważanego w [34], gdzie wyjście jest nieliniową funkcją zmiennych stanu. Wykorzystywanie wówczas modelu liniowego, tzn. zlinearyzowanego w pewnym punkcie pracy procesu, prowadzi do stosowania również zlinearyzowanych równań wyjścia, stąd wprowadza błędy modelowania tych równań. Błędów takich nie są w stanie skompensować zakłócenia addytywne stanu  $v(k)$ , gdyż nie są one objęte pętlą sprzężenia zwrotnego od stanu. Stąd powstaje błąd stanu ustalonego wyjścia, tzw. *offset*. Dla jego wyeliminowania mamy dwa sposoby:

1. Dodane sygnału kompensującego  $d$ , wyliczanego tak, aby w stanach ustalonych (w punktach równowagi) wartości oryginalnych, nieliniowych równań wyjścia  $y = g(x)$  i równań zlinearyzowanych,  $y = \mathbf{C}x + d$ , były równe, tzn.  $g(x^{ss}) = \mathbf{C}x^{ss} + d(y^{zad})$ , gdzie  $x^{ss}$  jest stanem równowagi odpowiadającym wartości zadanej  $y^{zad}$ . Wartość przesunięcia kompensującego zależy od tej wartości, dlatego powinna być dostrajana on-line do zmian  $y^{zad}$ . Sposobem realizacji jest dodawanie potrzebnej wartości różnicy  $d(k)$  do wartości zadanej,

$$d(k) = -\mathbf{C}x(k) + g(x(k)). \quad (3.117)$$

2. Potraktowanie różnicy między oryginalnymi a modelowymi (z modelu liniowego) równaniami wyjścia, powodowanej błędami modelowania, jako niemierzalnego zakłócenia dodawanego do wyjścia i potraktowanie tego zakłócenia tak, jak w algorytmach DMC i GPC prezentowanych w rozdziałach poprzednich. Model zakłóceń jest wówczas w postaci

$$d(k) = y(k) - y(k|k-1) = y(k) - \mathbf{C}x(k|k-1)$$

$$= y(k) - \mathbf{C}[\mathbf{A}x(k-1) + \mathbf{B}u(k-1) + v(k-1)], \quad (3.118)$$

gdzie  $y(k) = g(x(k))$ , i  $d(k)$  powinno być dodawane do równan predykcji (3.109). Wówczas zależność opisująca część swobodną trajektorii wyjścia będzie postaci

$$\mathcal{Y}^0(k) = \tilde{\mathbf{C}}\tilde{\mathbf{A}}x(k) + \tilde{\mathbf{C}}\mathbf{V}[\mathbf{B}u(k-1) + v(k)] + \tilde{\mathbf{I}}_y d(k), \quad (3.119)$$

gdzie  $\tilde{\mathbf{I}}_y = [\mathbf{I}_y \ \mathbf{I}_y \ \dots \ \mathbf{I}_y]^T$ ,  $\dim \tilde{\mathbf{I}}_y = N \cdot n_y \times n_y$ . W przypadku tym analityczne prawo regulacji MPCs (3.114) jest rozszerzone do postaci

$$\Delta \hat{u}(k) = \mathbf{K}^e [y^{sp}(k) - d(k)] - \sum_{p=1}^N \mathbf{K}_{1,p} [\mathbf{C}\mathbf{A}^p x(k) + \mathbf{C}\mathbf{V}_p [\mathbf{B}u(k-1) + v(k)]] \quad (3.120)$$

### Przykład 3.5.

Zastosujemy algorytm analityczny MPCs do obiektu z przykładów 3.1 i 3.2, opisanego równaniem

$$(1 - 0.2676z^{-1})y(k) = (0.1989z^{-2} + 0.2552z^{-3})u(k-1). \quad (3.121)$$

Tak jak poprzednio przyjmiemy  $N_u = 3$ ,  $N = 6$  oraz  $\underline{\Psi} = \mathbf{I}$ ,  $\underline{\Lambda} = \lambda \mathbf{I}$ . Dla potrzeb algorytmu MPCs obiekt opiszemy modelem w postaci równań stanu, ze wektorem stanu zdefiniowanym wg. (3.116), w postaci

$$x(k) = [y(k) \ u(k-1) \ u(k-2) \ u(k-3)].$$

Wykorzystując tę definicję, dostajemy następujące równania stanu i wyjścia odpowiadające transmitancji (3.162)

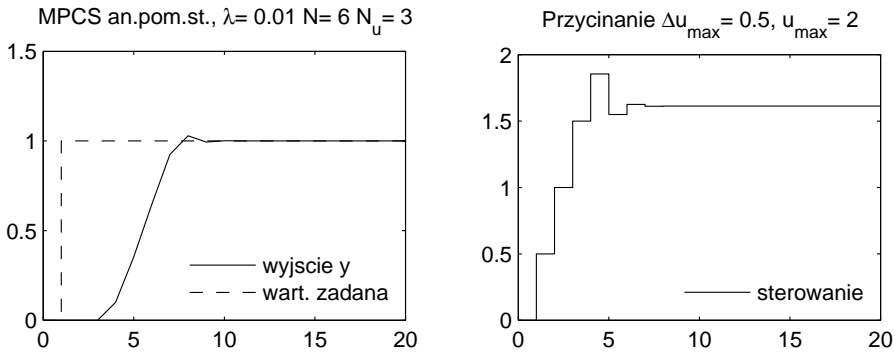
$$x(k+1) = \begin{bmatrix} 0.2676 & 0 & 0.1989 & 0.2552 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u(k) + v(k),$$

$$y(k) = [1 \ 0 \ 0 \ 0] x(k),$$

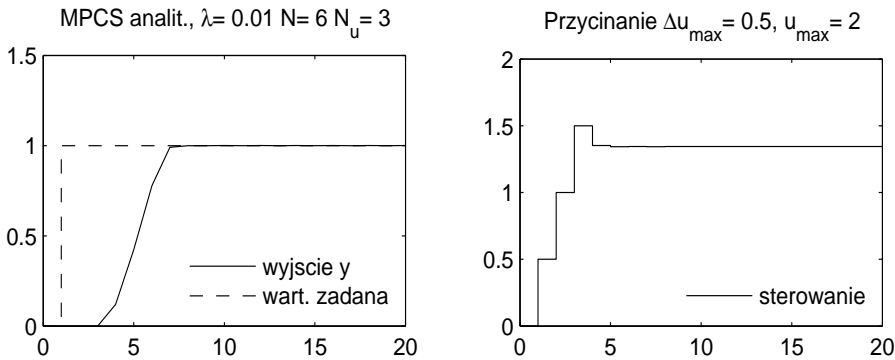
z dodanym wektorem estymaty niemierzalnych zakłóceń stanu  $v(k)$ . Można sprawdzić, że akurat w tym przypadku uzyskany model jest o minimalnym wymiarze stanu.

Wynik uzyskane w przypadku nominalnym (obiekt w pętli regulacji i jego model użyty do projektowania regulatora identyczne) dla skoku wartości zadanej pokazany jest na rysunku 3.39, jest on oczywiście praktycznie identyczny z uzyskanym dla regulatorów DMC i GPC. Natomiast na

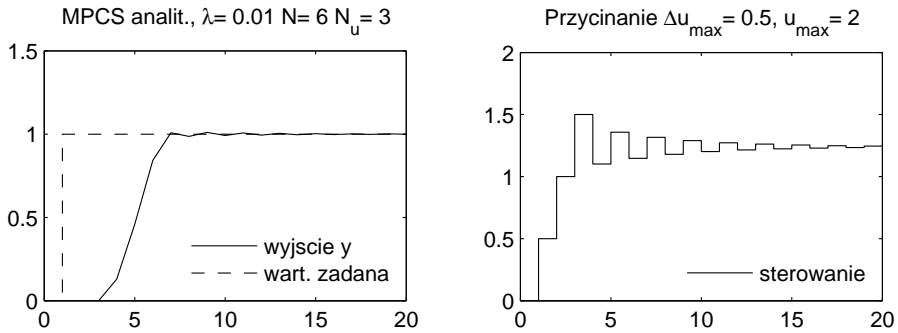
rysunku 3.40 podajemy wyniki symulacji z obiektem o wzmacnieniu zwiększonym o 20%, a na rysunku 3.41 przy wzmacnieniu obiektu zwiększonym o 30% – w tym ostatnim przypadku uwagę zwraca znacznie bardziej oscylacyjny charakter sterowania. Jednakże zwiększenie wartości współczynnika wagowego  $\lambda$  z 0.01 do 0.05 stabilizuje układ regulacji, co pokazano na rysunku 3.42. Z kolei na rysunku 3.43 pokazano działanie układu regulacji przy skoku jednostkowym zakłócenia na wejściu obiektu w chwili  $k = 2$ . Z porównania tego ostatniego rysunku z analogicznymi rysunkami dla regulatorów DMC i GPC, por. rys. 3.19 i 3.27, widać, że w badanym przykładzie regulator MPCs z pomiarem stanu lepiej poradził sobie z tłumieniem skoku zakłócenia na wejściu obiektu. Wynika to z cech zastosowanego modelu zakłóceń  $v(k)$ , opartego na bezpośrednim pomiarze stanu obiektu.  $\square$



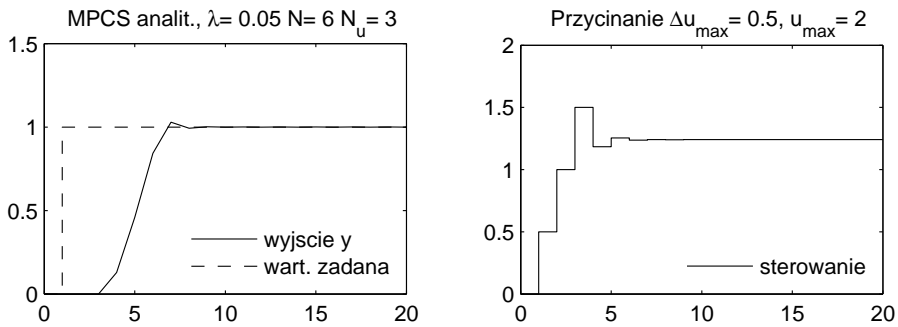
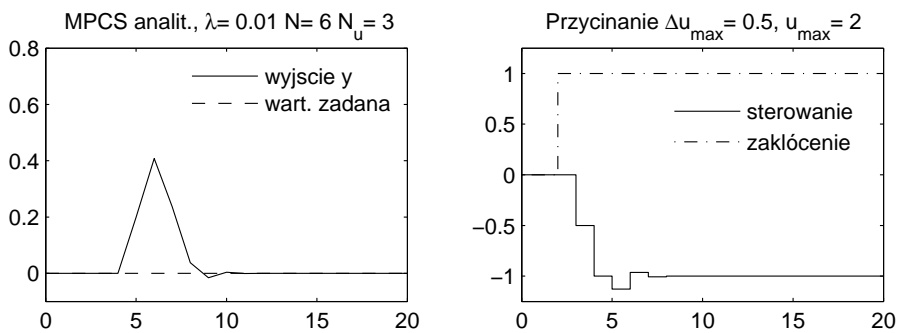
Rysunek 3.39. Wyniki symulacji z regulatorem MPCs analitycznym z pomiarem stanu



Rysunek 3.40. Wyniki symulacji z regulatorem MPCs analitycznym z pomiarem stanu, wzmacnienie obiektu zwiększone o 20%



Rysunek 3.41. Wyniki symulacji z regulatorem MPCs analitycznym z pomiarem stanu, wzmacnienie obiektu zwiększone o 30%

Rysunek 3.42. Wyniki symulacji z regulatorem MPCs analitycznym z pomiarem stanu, wzmacnienie obiektu zwiększone o 30%,  $\lambda = 0.05$ Rysunek 3.43. Wyniki symulacji z regulatorem MPCs analitycznym z pomiarem stanu, przy skoku jednostkowym zakłócenia na wejściu w chwili  $k = 2$

**Przykład 3.6.**

Wyznaczymy równania stanu dla dwuwymiarowego obiektu zaprezentowanego w przykładzie 3.4, danego równaniami różnicowymi (3.100)-(3.101):

$$\begin{aligned}(1 - 1.862885z^{-1} + 0.866877z^{-2})y_1(k) &= (0.041951 + \\ &\quad - 0.037959z^{-1})u_1(k-1) + (0.475812 - 0.455851z^{-1})u_2(k-1), \\ (1 - 1.869508z^{-1} + 0.873715z^{-2})y_2(k) &= (0.058235 - \\ &\quad - 0.054027z^{-1})u_1(k-1) + (0.144513 - 0.136097z^{-1})u_2(k-1).\end{aligned}$$

Rozważmy dwie reprezentacje równań stanu: minimalnowymiarową i wynikającą z definicji stanu zgodnie z (3.116).

1. Reprezentacja stanu jako mierzalnego, wg definicji (3.116).

Definiując wektor stanu wg (3.116) dostajemy wektor 6-wymiarowy:

$$x(k) = [y_1(k) \ y_2(k) \ y_1(k-1) \ y_2(k-1) \ u_1(k-1) \ u_2(k-1)]^T, \quad (3.122)$$

oraz równania stanu  $x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k)$ ,  $y(k) = \mathbf{C}x(k)$  w postaci

$$x(k+1) = \begin{bmatrix} -\mathbf{A}_1 & -\mathbf{A}_2 & \mathbf{B}_1 \\ \mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix} x(k) + \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{0}_2 \\ \mathbf{I}_2 \end{bmatrix} u(k), \quad (3.123a)$$

$$y(k) = [\mathbf{I}_2 \ \mathbf{0}_2 \ \mathbf{0}_2] x(k), \quad (3.123b)$$

gdzie poszczególne macierzowe elementy równań stanu i wyjścia wynikające bezpośrednio z przytoczonych równań różnicowych modelu (3.100)-(3.101), czy z przekształconych do postaci (3.102), są postaci:

$$\mathbf{A}_1 = \begin{bmatrix} -1.862885 & 0 \\ 0 & -1.869508 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0.866877 & 0 \\ 0 & 0.873715 \end{bmatrix},$$

$$\mathbf{B}_0 = \begin{bmatrix} 0.041951 & 0.475812 \\ 0.058235 & 0.144513 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} -0.037959 & -0.455851 \\ -0.054027 & -0.136097 \end{bmatrix}.$$

2. Reprezentacja stanu minimalnowymiarowa.

Zastosowanie funkcji „ss” pakietu Matlab prowadzi do przetransfor-

mowania macierzy transmitancji dyskretnych (3.97) do następującego 4-wymiarowego modelu równań stanu:

$$x(k+1) = \begin{bmatrix} 0.958 & 0 & 0 & 0 \\ 0 & 0.9418 & 0 & 0 \\ 0 & 0 & 0.9048 & 0 \\ 0 & 0 & 0 & 0.9277 \end{bmatrix} + \begin{bmatrix} 0.25 & 0 \\ 0.25 & 0 \\ 0 & 0.5 \\ 0 & 0.5 \end{bmatrix} u(k),$$

$$y(k) = \begin{bmatrix} 0.1678 & 0 & 0.9516 & 0 \\ 0 & 0.2329 & 0 & 0.289 \end{bmatrix} x(k). \quad (3.124)$$

Model uzyskany w drugim przypadku ma mniejszy wymiar, ale konieczne jest zastosowanie obserwatora stanu.  $\square$

### 3.4.2. Algorytmy MPCs z obserwatorami stanu

Jeśli nie stosujemy podejścia z wektorem stanu złożonym ze zmiennych wejściowych i wyjściowych (3.116), to cały wektor stanu jest najczęściej pomiarowo bezpośrednio niedostępny, konieczne jest zastosowanie *obserwatora stanu*. Dla obiektu opisanego równaniami (3.103), (3.104) obserwator stanu (deterministyczny, typu Luenbergera) można przedstawić w postaci [3]

$$\hat{x}(k+1) = \mathbf{A}\hat{x}(k) + \mathbf{B}u(k) + \mathbf{K}_{obs}[y(k) - \mathbf{C}\hat{x}(k)], \quad (3.125)$$

gdzie  $\hat{x}(k)$  oznacza estymatę wektora stanu  $x(k)$ , zaś  $\mathbf{K}_{obs}$  jest macierzą wzmocnień określającą dynamikę obserwatora. Z teorii sterowania wiadomo, że obserwowalność liniowego układu dynamicznego opisanego macierzami  $\mathbf{A}$  i  $\mathbf{C}$  jest warunkiem umożliwiającym dowolne ulokowanie biegunów macierzy stanu  $\mathbf{A} - \mathbf{K}_{obs}\mathbf{C}$  obserwatora (3.125), będącej jednocześnie macierzą dynamiki błędu estymacji  $e_{obs}(k) = x(k) - \hat{x}(k)$ , poprzez odpowiedni dobór macierzy  $\mathbf{K}_{obs}$ . Wstawiając bowiem równanie wyjścia  $y(k) = \mathbf{C}x(k)$  do (3.125) i odejmując to równanie od równania stanu obiektu  $x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k)$ , dostajemy

$$e_{obs}(k+1) = \mathbf{A}e_{obs}(k) - \mathbf{K}_{obs}\mathbf{C}e_{obs}(k) = (\mathbf{A} - \mathbf{K}_{obs}\mathbf{C})e_{obs}(k). \quad (3.126)$$

Stąd, błąd estymacji zanika do zera z dynamiką układu liniowego o macierzy stanu  $\mathbf{A} - \mathbf{K}_{obs}\mathbf{C}$ ,

$$e_{obs}(k) = (\mathbf{A} - \mathbf{K}_{obs}\mathbf{C})^k e_{obs}(0), \quad (3.127)$$

gdzie  $e_{obs}(0)$  to różnica stanów obiektu i obserwatora w chwili początkowej.

Dysponując dobrą estymatą stanu  $\hat{x}(k)$  i wstawiając ją zamiast wektora stanu do zależności (3.105)

$$v(k) = \hat{x}(k) - [\mathbf{A}\hat{x}(k-1) + \mathbf{B}u(k-1)], \quad (3.128)$$

oraz w miejsce wektora stanu  $x(k)$  we wzorze (3.109) uzyskujemy wzór na składową swobodną prognozowanej trajektorii wyjść przy estymacji stanu:

$$\mathcal{Y}^0(k) = \tilde{\mathbf{C}}\tilde{\mathbf{A}}\hat{x}(k) + \tilde{\mathbf{C}}\mathbf{V}\mathbf{B}u(k-1) + \tilde{\mathbf{C}}\mathbf{V}v(k). \quad (3.129)$$

Przedstawiony obserwator estymuje w układzie regulacji stan obiektu poprawnie jedynie wówczas, w szczególności przy występowaniu zakłóceń deterministycznych, gdy zakłócenia te są o zerowej wartości średniej czy ustalonej (np. impuls na wejściu szybko zanikający). Przy skokowym zakłóceniu np. na wejściu obiektu stan estymowany nie będzie równy stanowi obiektu, a będzie równy pewnemu zastępczemu stanowi uwzględniającemu wpływ wartości zakłócenia. Wynika to bezpośrednio z równań obiektu zakłócanego i obserwatora. Mamy wówczas bowiem równanie dynamiki obiektu w postaci

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}(u(k) + z(k)), \quad (3.130)$$

gdzie  $z(k)$  jest zakłóceniem deterministycznym. W połączeniu z równaniem obserwatora (3.125) i równaniem wyjścia  $y(k) = \mathbf{C}x(k)$  dostajemy stąd

$$e_{obs}(k+1) = \mathbf{A}e_{obs}(k) + \mathbf{B}z(k) - \mathbf{K}_{obs}\mathbf{C}e_{obs}(k) = (\mathbf{A} - \mathbf{K}_{obs}\mathbf{C})e_{obs}(k) + \mathbf{B}z(k), \quad (3.131)$$

auchyb obserwacji w stanie ustalonym będzie niezerowy dla niezerowej wartości ustalonej zakłócenia  $z_{ust}$ , równy

$$e_{obs,ust} = (\mathbf{I} - \mathbf{A} + \mathbf{K}_{obs}\mathbf{C})^{-1}\mathbf{B}z_{ust}. \quad (3.132)$$

W takiej sytuacji:

- obserwator stanu estymuje stan zakłócony obiektu,  $\hat{x}(k) \neq x(k)$  i regulator otrzymuje do predykcji zakłóconą wartość wyjść regulowanych  $\hat{y}(k) = \mathbf{C}\hat{x}(k) \neq y(k)$ ,
- algorytm regulacji MPCs wyzerowuje uchyb regulacji liczony w stosunku do predykcji wyjść wyliczonej przy zakłóconym „pomiarze”  $\hat{y}(k) = \mathbf{C}\hat{x}(k)$ , tzn. z uchybem  $\mathbf{C}\hat{x}(k) - y(k)$ , stąd w stanie ustalonym dostajemy uchyb statyczny estymacji wyjścia, równy  $\mathbf{C}\hat{x}_{ust} - y_{ust}$ .

Niemniej, regulator MPCs z takim obserwatorem stanu obiektu będzie działał poprawnie, tzn. zerował uchyb regulacji, *jeśli rozszerzymy formułę predykcji trajektorii swobodnej (3.129), bezpośrednio korygując błąd estymacji wyjścia*, tzn. do postaci [147]:

$$y^0(k+p|k) = \mathbf{C}\mathbf{A}^p x(k) + \mathbf{C}\mathbf{V}_p[\mathbf{B}u(k-1) + v(k)] - [\mathbf{C}\hat{x}(k) - y(k)], \quad p = 1, \dots, N, \quad (3.133)$$

gdzie  $\mathbf{V}_p = [\mathbf{A}^{p-1} + \mathbf{A}^{p-2} + \dots + \mathbf{I}]$ , lub w zapisie wektorowym składowej swobodnej prognozowanej trajektorii wyjść (por. z (3.129)):

$$\mathcal{Y}^0(k) = \tilde{\mathbf{C}}\tilde{\mathbf{A}}\hat{x}(k) + \tilde{\mathbf{C}}\mathbf{V}[\mathbf{B}u(k-1) + v(k)] + \mathbf{I}_{yN}[y(k) - \mathbf{C}\hat{x}(k)],$$

gdzie

$$\mathbf{I}_{yN} = \left[ \begin{array}{c} \mathbf{I}_y \\ \vdots \\ \mathbf{I}_y \end{array} \right] \left. \vphantom{\begin{array}{c} \mathbf{I}_y \\ \vdots \\ \mathbf{I}_y \end{array}} \right\} N \text{ razy } \mathbf{I}_y, \quad \dim \mathbf{I}_y = n_y \times n_y.$$

Analityczne prawo regulacji MPCs jest teraz postaci (por. z (3.111))

$$\begin{aligned} \Delta\hat{u}(k) = \bar{\mathbf{K}}_1 \{ \mathcal{Y}^{zad}(k) - \tilde{\mathbf{C}}\tilde{\mathbf{A}}\hat{x}(k) - \tilde{\mathbf{C}}\mathbf{V}[\mathbf{B}u(k-1) + v(k)] + \\ + \mathbf{I}_{yN}[y(k) - \mathbf{C}\hat{x}(k)] \}, \end{aligned} \quad (3.134)$$

gdzie, przypomnijmy,  $\bar{\mathbf{K}}_1$  jest pierwszym (macierzowym) wierszem macierzy  $\mathbf{K}$ , zob. (3.40). Zakładając stałość wartości zadanych na horyzoncie predykcji i wykorzystując struktury macierzy  $\tilde{\mathbf{A}}$ ,  $\mathbf{V}$  i  $\tilde{\mathbf{C}}$ , prawo regulacji można przekształcić do postaci (por. z (3.112))

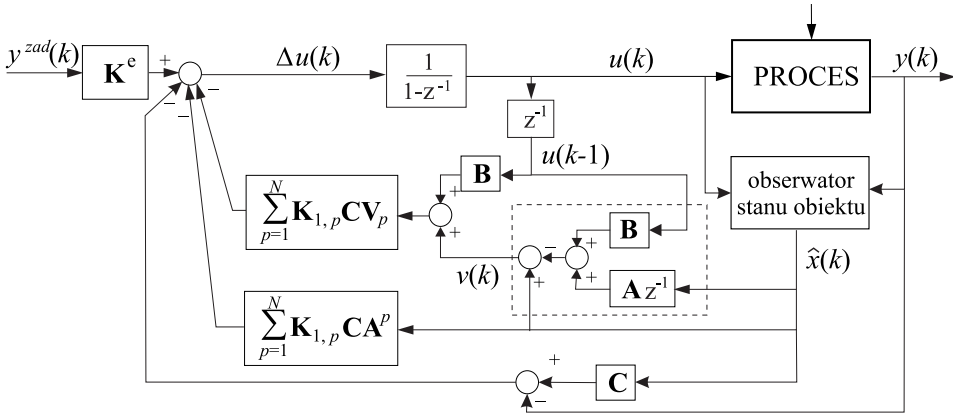
$$\begin{aligned} \Delta\hat{u}(k) = \sum_{p=1}^N \mathbf{K}_{1,p} \{ y^{zad}(k) - \mathbf{C}\mathbf{A}^p\hat{x}(k) - \mathbf{C}\mathbf{V}_p[\mathbf{B}u(k-1) + v(k)] + \\ + [y(k) - \mathbf{C}\hat{x}(k)] \}, \end{aligned} \quad (3.135)$$

gdzie  $\mathbf{K}_{1,p}$  są podmacierzami (w przypadku obiektów SISO skalarami) macierzy  $\bar{\mathbf{K}}_1 = [\mathbf{K}_{1,1} \ \mathbf{K}_{1,2} \ \dots \ \mathbf{K}_{1,N}]$  (zob. (3.40)), a  $\mathbf{V}_p$  są podmacierzami macierzy  $\mathbf{V}$ , zob. (3.113). Wykorzystując teraz definicję  $\mathbf{K}^e$  (3.44), prawo regulacji (3.135) przyjmuje postać

$$\begin{aligned} \Delta\hat{u}(k) = \mathbf{K}^e y^{zad}(k) - \sum_{p=1}^N \mathbf{K}_{1,p} \{ \mathbf{C}\mathbf{A}^p\hat{x}(k) + \mathbf{C}\mathbf{V}_p[\mathbf{B}u(k-1) + v(k)] + \\ + [y(k) - \mathbf{C}\hat{x}(k)] \}, \end{aligned} \quad (3.136)$$

gdzie estymata zakłóceń stanu  $v(k)$  dana jest zależnością (3.128). Struktura regulatora przedstawiona jest na rysunku 3.44 (por. rys. 3.38). Jest to struktura bez elementów uwzględniania ograniczeń i anti-windup, które należy dodać w analogiczny sposób, jak to przedstawiono w przypadku regulacji DMC.

Należy podkreślić, że prawidłowe działanie rozważanego regulatora MPCs ze standardowym (nie rozszerzonym) obserwatorem (3.125) zachodzi dzięki korekcie wprowadzonej w równaniach na predykcję trajektorii swobodnej wyjść oraz stosowaniu zakłóceń stanu  $v(k)$  i ich modelu (3.128), uwzględnianemu w tych równaniach. Regulator będzie też prawidłowo działał przy zakłóceniach deterministycznych wpływających na wyjścia obiektu



Rysunek 3.44. Struktura analitycznego regulatora MPCs z obserwatorem stanu obiektu (bez elementów rzutowania sterowania na ograniczenia i anti-windup)

(zakłócających równania wyjścia) – co więcej, nie jest w tym przypadku konieczna odrębna estymacja wartości tych zakłóceń i ich dodatkowa kompensacja, jak to ma miejsce dla deterministycznych zakłóceń równań wyjścia dla regulatora MPCs z pełnym pomiarem stanu rozważanego w rozdziale poprzednim, zob. (3.117), (3.118) i (3.119). Przyczyną jest fakt, że teraz sprzężenie zwrotne jest od wyjścia (poprzez obserwator), a nie bezpośrednio od stanu, stąd obejmuje również zakłócenie oddziałujące bezpośrednio na relację stan-wyjście.

Alternatywnym podejściem do estymacji stanu w regulatorze predykcyjnym z modelem w postaci równań stanu, historycznie wcześniejszym od zaprezentowanego dotychczas w niniejszym rozdziale, jest zastosowanie estymacji *rozszerzonego wektora stanu*. Model dynamiki obiektu rozszerzamy o założoną dynamikę zakłóceń deterministycznych i projektujemy obserwator estymujący taki rozszerzony wektor stanu, zob. np. [105, 87, 144, 88, 147]. Estymata stanu samego obiektu ( $\hat{x}(k)$ ) może być w tym przypadku dokładniejsza, choć nie zawsze nieobciążona uchybem w stanach ustalonych – zależy to od relacji między strukturą oryginalnych zakłóceń w samym obiekcie i strukturą modelu zakłóceń przyjętego w wykorzystywanym modelu obiektu.

Przykładowo, dla modelu obiektu zakładającego oddziaływanie zakłóceń deterministycznych  $z(k)$  typu skokowego na wejścia sterujące, rozszerzony, łączny model dynamiki obiektu i zakłóceń będzie postaci

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} u(k), \\ y(k) &= \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix}, \end{aligned} \quad (3.137)$$

a wynikające stąd równania obserwatora rozszerzonego stanu postaci

$$\begin{aligned} \begin{bmatrix} \hat{x}(k+1) \\ \hat{z}(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{z}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} u(k) + \\ &+ \mathbf{K}_{obsr} \left( y(k) - \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{z}(k) \end{bmatrix} \right). \end{aligned} \quad (3.138)$$

Aby powyższy obserwator stanu działał poprawnie, model rozszerzony musi być obserwowalny (przynajmniej wykrywalny – *detectable*).

Z kolei, dla przypadku modelu obiektu zakładającego oddziaływanie zakłóceń deterministycznych  $d(k)$  typu skokowego na wyjścia, rozszerzony, łączny model dynamiki obiektu i zakłóceń będzie postaci

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ d(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} u(k), \\ y(k) &= \begin{bmatrix} \mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix}, \end{aligned} \quad (3.139)$$

a wynikające stąd równania obserwatora rozszerzonego stanu postaci

$$\begin{aligned} \begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} u(k) + \\ &+ \mathbf{K}_{obsr} \left( y(k) - \begin{bmatrix} \mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \end{bmatrix} \right). \end{aligned} \quad (3.140)$$

Analogicznie jak poprzednio, aby powyższy obserwator stanu działał poprawnie, model rozszerzony musi być obserwowalny (przynajmniej wykrywalny).

W pracy [105] rozważono ogólny model rozszerzony o deterministyczne zakłócenia stanu i wyjścia:

$$\begin{bmatrix} x(k+1) \\ z(k+1) \\ d(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{G}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} u(k),$$

$$y(k) = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{G}_d \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \\ d(k) \end{bmatrix}. \quad (3.141)$$

Zauważmy, że dla  $\mathbf{G}_z = \mathbf{B}$  zakłócenia  $z(k)$  są zakłóceniami wejścia sterowanego, jak we wzorze (3.137), a dla  $\mathbf{G}_d = \mathbf{I}$  zakłócenia  $d(k)$  są zakłóceniami wyjścia, jak we wzorze (3.139). W [105] sformułowano też warunki wykrywalności tego modelu: pokazano, że jest on wykrywalny wtedy i tylko wtedy, gdy system  $(\mathbf{C}, \mathbf{A})$  jest wykrywalny oraz

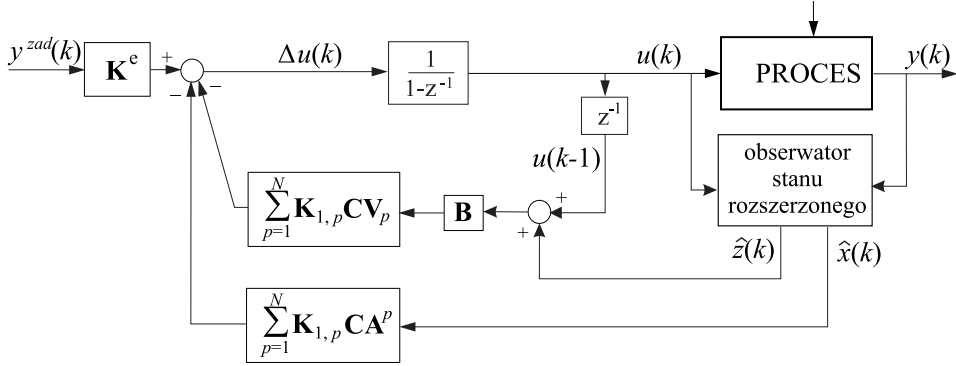
$$\text{rank} \begin{bmatrix} (\mathbf{I} - \mathbf{A}) & -\mathbf{G}_z & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \mathbf{G}_d \end{bmatrix} = \dim x + \dim z + \dim d. \quad (3.142)$$

Z powyższego wynika, że model rozszerzony nie jest wykrywalny, jeśli liczba rozszerzających stanów zakłóceń  $z$  i  $d$  jest większa od liczby wyjść  $y$  (zakładając pełne kolumnowe rzędy macierzy  $\mathbf{G}_z$  i  $\mathbf{G}_d$ ). W szczególności, nie możemy zastosować modelu z pewnymi zakłóceniami  $z(k)$  wejść sterujących i jednocześnie zakłóceniami  $d(k)$  wszystkich wyjść – łączna liczba zakłóceń nie może przekraczać liczby wyjść mierzonych, tj. wymiaru wektora  $y(k)$ .

Estymata stanu realizowana obserwatorem rozszerzonym zakłada inny model zakłócenia niż prezentowany poprzednio w niniejszym rozdziale – zamiast zakłócenia stanu  $v(k)$  modelowanego zależnością (3.128) mamy wybrane spośród zakłóceń  $z(k)$  i  $d(k)$  występujących w zależności (3.141) – przypomnijmy, mamy ograniczenie  $\dim z + \dim d \leq \dim y$  wynikające z warunku obserwowalności. I to te wybrane zakłócenia należy uwzględnić w równaniach predykcji. Przykładowo, jeśli modelujemy zakłócenia niemierzalne jedynie jako  $z(k)$ , tzn. oddziałujące na wejście obiektu, to mamy z reguły spełniony wymieniony warunek obserwowalności, gdyż mamy z reguły więcej pomiarów wyjść niż wejść sterujących  $\dim u \leq \dim y$  (wyjść regulowanych bywa często mniej niż mierzonych). Przy obliczaniu składowej swobodnej prognozowanej trajektorii wyjść należy w tym przypadku uwzględnić zamiast równań (3.103)-(3.104) równania:

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}(u(k) + z(k)), \quad (3.143)$$

$$y(k) = \mathbf{C}x(k). \quad (3.144)$$



Rysunek 3.45. Struktura analitycznego regulatora MPCs z obserwatorem stanu obiektu  $x(k)$  rozszerzonego o zakłócenia wejścia  $z(k)$  (bez elementów rzutowania sterowania na ograniczenia i anti-windup)

Prowadzi to do zależności na składową swobodną trajektorii wyjść

$$\mathcal{Y}^0(k) = \tilde{\mathbf{C}}\tilde{\mathbf{A}}\hat{x}(k) + \tilde{\mathbf{C}}\mathbf{V}\mathbf{B}[u(k-1) + \hat{z}(k)], \quad (3.145)$$

lub w zapisie rozbitym na poszczególne elementy na horyzoncie predykcji:

$$y^0(k+p|k) = \mathbf{C}\mathbf{A}^p\hat{x}(k) + \mathbf{C}\mathbf{V}_p\mathbf{B}[u(k-1) + \hat{z}(k)], \quad p = 1, \dots, N, \quad (3.146)$$

gdzie  $\hat{x}(k)$  i  $\hat{z}(k)$  to estymaty stanu i zakłócenia uzyskiwane przez obserwator rozszerzonego stanu (3.138). Prowadzi to dalej do *analitycznego prawa regulacji MPCs*

$$\Delta\hat{u}(k) = \bar{\mathbf{K}}_1\{\mathcal{Y}^{zad}(k) - \tilde{\mathbf{C}}\tilde{\mathbf{A}}\hat{x}(k) - \tilde{\mathbf{C}}\mathbf{V}\mathbf{B}[u(k-1) + \hat{z}(k)]\}, \quad (3.147)$$

por. (3.134). Zakładając ponadto stałość wartości zadanych na horyzoncie predykcji prawo to można przekształcić do postaci (por. (3.136))

$$\Delta\hat{u}(k) = \mathbf{K}^e y^{zad}(k) - \sum_{p=1}^N \mathbf{K}_{1,p} \{\mathbf{C}\mathbf{A}^p\hat{x}(k) + \mathbf{C}\mathbf{V}_p\mathbf{B}[u(k-1) + \hat{z}(k)]\}. \quad (3.148)$$

Struktura tego prawa regulacji pokazana jest na rysunku 3.45, por. rys. 3.44.

Podobnie, przy modelowaniu zakłóceń niemierzalnych jedynie jako zakłóceń wyjścia  $d(k)$  przy obliczaniu składowej swobodnej prognozowanej trajektorii wyjść należy uwzględnić zamiast (3.103)-(3.104) równania:

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k), \quad (3.149)$$

$$y(k) = \mathbf{C}x(k) + d(k). \quad (3.150)$$

Prowadzi to do zależności na składową swobodną trajektorii wyjść

$$\mathcal{Y}^0(k) = \tilde{\mathbf{C}}\tilde{\mathbf{A}}\hat{x}(k) + \tilde{\mathbf{C}}\mathbf{V}\mathbf{B}u(k-1) + \mathbf{I}_{yN}\hat{d}(k), \quad (3.151)$$

lub w zapisie rozbitym na poszczególne elementy na horyzoncie predykcji:

$$y^0(k+p|k) = \mathbf{C}\mathbf{A}^p\hat{x}(k) + \mathbf{C}\mathbf{V}_p\mathbf{B}u(k-1) + \hat{d}(k), \quad p = 1, \dots, N, \quad (3.152)$$

gdzie  $\hat{x}(k)$  i  $\hat{d}(k)$  to estymaty stanu i zakłócenia uzyskiwane przez obserwator rozszerzonego stanu (3.140).

Obserwator stanu Luenbergera, dany równaniem (3.125)

$$\hat{x}(k+1) = \mathbf{A}\hat{x}(k) + \mathbf{B}u(k) + \mathbf{K}_{obs}[y(k) - \mathbf{C}\hat{x}(k)], \quad (3.153)$$

nazywany jest "predykcyjnym", zob. np. [3, 144]. Powodem jest fakt, że aktualna informacja w chwili  $k$ , tzn. pomiar wyjść  $y(k)$ , wykorzystywany jest do obliczenia przyszłej wartości estymaty wartości stanu  $\hat{x}(k+1)$ , a nie aktualnej  $\hat{x}(k)$  – czyli dokonujemy predykcji stanu. Aktualna informacja jest wykorzystywana z opóźnieniem o takt (okres impulsowania). Stąd często stosowana jest nieco inna, „poprawiona” wersja tego obserwatora, zwany jest on wówczas „bieżącym” (*current observer*), postaci:

$$\hat{x}(k|k-1) = \mathbf{A}\hat{x}(k-1) + \mathbf{B}u(k-1), \quad (3.154a)$$

$$\hat{x}(k) = \hat{x}(k|k-1) + \mathbf{K}_{obsb}[y(k) - \mathbf{C}\hat{x}(k|k-1)]. \quad (3.154b)$$

Równania (3.154a) i (3.154b) można sprowadzić do postaci

$$\hat{x}(k) = \mathbf{A}\hat{x}(k-1) + \mathbf{B}u(k-1) + \mathbf{K}_{obsb}[y(k) - \mathbf{C}(\mathbf{A}\hat{x}(k-1) + \mathbf{B}u(k-1))] \quad (3.155)$$

Dla projektowania macierzy wzmocnień  $\mathbf{K}_{obsb}$  obserwatora bieżącego sprowadza się jednakże równania (3.154a) i (3.154b) do nieco innego pojedynczego równania dynamiki: wstawiając (3.154b) do (3.154a) przesuniętego o takt do przodu dostajemy

$$\hat{x}(k+1|k) = \mathbf{A}\hat{x}(k|k-1) + \mathbf{B}u(k) + \mathbf{A}\mathbf{K}_{obsb}[y(k) - \mathbf{C}\hat{x}(k|k-1)]. \quad (3.156)$$

Jest to równanie dynamiczne identyczne do równania obserwatora predykcyjnego (3.153), jeśli przyjąć  $\mathbf{K}_{obs} = \mathbf{A}\mathbf{K}_{obsb}$ . Stąd macierz  $\mathbf{K}_{obsb}$  można projektować wyznaczając najpierw elementy macierzy wzmocnień  $\mathbf{K}_{obs}$  obserwatora predykcyjnego odpowiednio do założonych wartości własnych jego dynamiki, a następnie wyliczając elementy macierzy  $\mathbf{K}_{obsb}$  [48, 3].

Przy stochastycznych zaburzeniach zmiennych stanu i wyjścia do estymacji stanu (ew. rozszerzonego o stan zakłóceń typu deterministycznego,

np. postaci skoku na wyjściu czy wejściu modelu obiektu) zaleca się stosowanie filtru Kalmana, zob. np. [62, 106, 48, 3, 55].

Dla sformułowania równań filtru Kalmana rozważa się następującą postać równań stanu obiektu

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k) + \varepsilon_x(k) \quad (3.157a)$$

$$y(k) = \mathbf{C}x(k) + \varepsilon_y(k) \quad (3.157b)$$

gdzie  $x$  jest wektorem stanu o wymiarze  $n_x$ ,  $y$  jest wektorem wyjść mierzonych regulowanych o wymiarze  $n_y$ ,  $\varepsilon_x$  jest wektorem szumów stanu, zaś  $\varepsilon_y$  jest wektorem szumów pomiarowych. Zakłada się, że szумы są białe, są zmiennymi losowymi wzajemnie nieskorelowanymi, o rozkładach normalnych z zerowymi wartościami średnimi, o macierzach kowariancji  $\mathbf{Q}$  i  $\mathbf{R}$ , odpowiednio.

Oznaczmy estymatę stanu przez  $\hat{x}(k)$ , wówczas błąd estymacji to  $e(k) = x(k) - \hat{x}(k)$ . Wartościami początkowymi dla filtru Kalmana są początkowa estymata stanu  $\hat{x}(k-1)$  i początkowa wartość macierzy kowariancji błędu estymacji  $\mathbf{P}_{k-1}$ .

Równania filtru można zapisać następująco:

$$\hat{x}(k|k-1) = \mathbf{A}\hat{x}(k-1) + \mathbf{B}u(k-1), \quad (3.158a)$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}, \quad (3.158b)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R})^{-1}, \quad (3.158c)$$

$$\hat{x}(k) = \hat{x}(k|k-1) + \mathbf{K}_k[y(k) - \mathbf{C}\hat{x}(k|k-1)], \quad (3.158d)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_{k|k-1}. \quad (3.158e)$$

Zauważmy, że na równania powyższe można spojrzeć w następujący sposób: równania (3.158a) i (3.158d) można zapisać łącznie w postaci

$$\hat{x}(k) = \mathbf{A}\hat{x}(k-1) + \mathbf{B}u(k-1) + \mathbf{K}_k[y(k) - \mathbf{C}(\mathbf{A}\hat{x}(k-1) + \mathbf{B}u(k-1))], \quad (3.159)$$

które to równanie można traktować jako właściwe równanie filtru, natomiast pozostałe trzy równania macierzowe służą do adaptacji macierzy wzmocnień  $\mathbf{K}_k$ . Należy tu zaznaczyć, że możliwe jest ogólniejsze sformułowanie równań filtru Kalmana, ze zmiennymi z kroku na krok macierzami kowariancji szumów  $\mathbf{Q}$  i  $\mathbf{R}$ .

Jeśli równania procesu i szумы są stacjonarne, to macierz kowariancji błędu  $\mathbf{P}_k$  zbiega do wartości ustalonej  $\mathbf{P}$  (w sensie wartości średniej), będącej rozwiązaniem równania Riccatiego:

$$\mathbf{P} = \mathbf{A}[\mathbf{P} - \mathbf{P}\mathbf{C}^T(\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R})^{-1}\mathbf{C}\mathbf{P}]\mathbf{A}^T + \mathbf{Q}. \quad (3.160)$$

Dostajemy stąd stacjonarny filtr Kalmana z macierzą wzmocnień

$$\mathbf{K} = \mathbf{P}\mathbf{C}^T(\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R})^{-1}. \quad (3.161)$$

Zauważmy, że równania obserwatora Luenbergera (3.154a)-(3.154b) (bieżącego) są identyczne z równaniami (3.158a) i (3.158d), jedynie macierze wzmocnień są różne, odpowiednio  $\mathbf{K}_{obsb}$  i  $\mathbf{K}_k$  (lub  $\mathbf{K}$  dla statycznego filtru Kalmana). Stąd różnica między obserwatorem stanu a filtrem Kalmana jest jedynie w sposobie wyznaczania macierzy wzmocnień, ponadto w filtrze Kalmana macierz ta może być zmienna z kroku na krok (dla bieżącej minimalizacji błędów estymacji stanu). Ważna jest oczywiście istota sposobu wyznaczania obu macierzy: w obserwatorze Luenbergera wyznaczamy macierz dla osiągnięcia odpowiedniej dynamiki malenia błędu estymacji, określanej przez położenie wartości własnych macierzy dynamiki układu zamkniętego obserwatora, a macierz filtru Kalmana projektowana jest dla minimalizacji wariancji błędu estymacji w warunkach zakłóceń stochastycznych.

### Przykład 3.7.

Dla zilustrowania projektowania regulatora MPCs z obserwatorem stanu wyznaczmy taki regulator dla prostego obiektu z przykładu 3.6 (a poprzednio z przykładów 3.1 i 3.2), opisanego dyskretnym równaniem różnicowym postaci (3.83)

$$(1 - 0.2676z^{-1})y(k) = (0.1989z^{-2} + 0.2552z^{-3})u(k - 1). \quad (3.162)$$

Podobnie jak poprzednio przyjmijmy  $N_1 = 3$ ,  $N_u = 3$ ,  $N = 6$  oraz  $\Psi(p) = \mathbf{I}$ ,  $\Lambda(p) = \lambda\mathbf{I}$ .

Korzystając z *Control System Toolbox* pakietu MATLAB wyznaczono układ równań stanu odpowiadający powyższej transmitancji dyskretniej, w postaci

$$x(k+1) = \begin{bmatrix} 0.2676 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(k), \quad (3.163a)$$

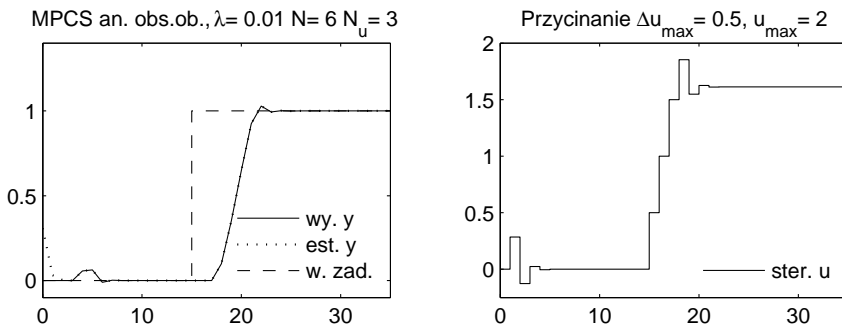
$$y(k) = \begin{bmatrix} 0.61685 & 0.1989 & 0 & 0 \end{bmatrix} x(k). \quad (3.163b)$$

Dla przyjętego modelu badano początkowo układ regulacji MPCs z obserwatorem stanu samego obiektu (stanu nie rozszerzonego o stany zakłóceń). Wyznaczono macierze prawa regulacji (3.136) oraz obserwator bieżący

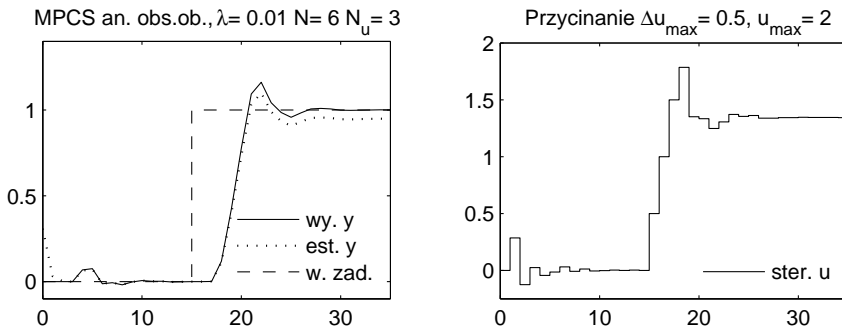
stanu (3.155). Ponieważ trzy ostatnie współrzędne wektora stanu odpowiadają opóźnieniu sterowania, to obserwator bieżący stanu przyjęto z wektorem wzmocnień w postaci  $\mathbf{K}_{obsb} = [k_1 \ 0 \ 0 \ 0]^T$ . Dla  $k_1 = 1$  daje to wartości własne macierzy dynamiki stanu obserwatora  $\mathbf{A} - \mathbf{A}\mathbf{K}_{obsb}\mathbf{C}$  (por. zależność (3.156)) równe 0.1025, 0, 0, 0.

Układ regulacji badano początkowo dla zerowego stanu początkowego obiektu i różniącego się stanu początkowego obserwatora, równego  $[0.5 \ 0 \ 0 \ 0]$  – aby mieć możliwość zaobserwowania dynamiki samego obserwatora. Ponadto w chwili  $k = 15$  zastosowano skok wartości zadanej od zera od wartości równej jeden (skok jednostkowy). Dla takich warunków przeprowadzono kilka symulacji, ze zmienianym wzmocnieniem obiektu i niezmiennym regulatorem (wyznaczonym dla wartości nominalnych modelu). Rysunki 3.46, 3.47 i 3.48 przedstawiają uzyskane trajektorie wyjścia obiektu, wyjścia estymowanego (liczonego na podstawie estymaty stanu) i sterowania dla regulatora MPCs zaprojektowanego dla modelu równego obiektowi nominalnemu i dla obiektu ze wzmocnieniem zwiększanym, kolejno, o 20% i 30%. Zaobserwować można szybkie działanie obserwatora. Ponadto im większy błąd modelowania, tym większa różnica w stanie ustalonym między wyjściem mierzonym i liczonem z estymaty stanu. Wystąpienie tej różnicy jest oczywiście zgodne z teorią regulatora z estymacją samego stanu obiektu (nie rozszerzonego) przy występowaniu zakłóceń deterministycznych. Rysunek 3.49 odpowiada natomiast sytuacji z zerowymi stanami początkowymi obiektu i obserwatora oraz jednostkowym skokiem zakłócenia na wejściu obiektu.

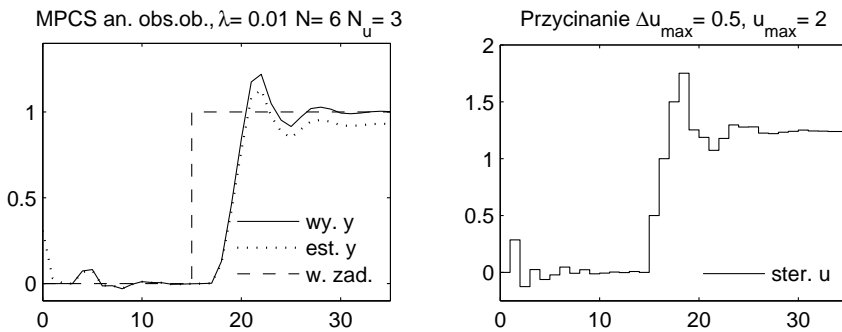
Następnie zbadano, dla porównania, układ regulacji MPCs z obserwatorem stanu obiektu rozszerzonym o stan zakłócenia. Macierz wzmocnień (w tym przypadku wektor – jedno wyjście obiektu) obserwatora bieżącego rozszerzonego stanu była w postaci  $\mathbf{K}_{obsb} = [1.237 \ 1.812 \ 1.962 \ 1.962 \ 1.962]^T$ . Wynika to z wyznaczenia początkowo wektora wzmocnień obserwatora predykcyjnego, a następnie odpowiedniego przeliczenia na wektor wzmocnień obserwatora bieżącego z zależności  $\mathbf{K}_{obs} = \mathbf{A}_r \mathbf{K}_{obsb}$  (przeliczenie jest niejednoznaczne, gdyż macierz stanu obiektu rozszerzonego jest osobliwa). Wartości własne macierzy  $\mathbf{A}_r - \mathbf{A}_r \mathbf{K}_{obsb} \mathbf{C}_r$  dynamiki obserwatora stanu rozszerzonego (3.138) wynoszą  $0.057 \pm 0.044i$ , 0, 0, 0. Na rysunku 3.50 pokazano trajektorie odpowiedzi na skok jednostkowy zakłócenia na wejściu obiektu, przy zerowych stanach początkowych obiektu i obserwatora. Zauważmy, że jest to sytuacja idealnego dopasowania przyjętego w modelu rozszerzonym rodzaju zakłócenia (zakłócenia wejścia  $z(k)$ ) do rzeczywistego miejsca oddziaływania zakłócenia w obiekcie. Stąd oczywiście estymata stanu, a stąd i wyjścia jest w tym przypadku dokładna. Zauważmy



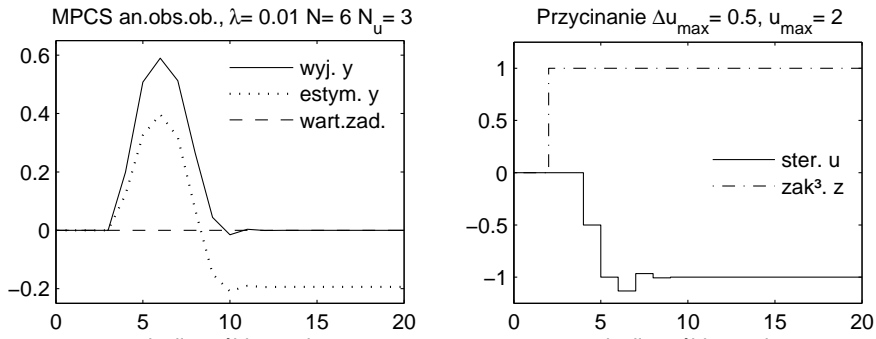
Rysunek 3.46. Odpowiedź na skok jednostkowy wartości zadanej w układzie regulacji MPC z obserwatorem stanu obiektu



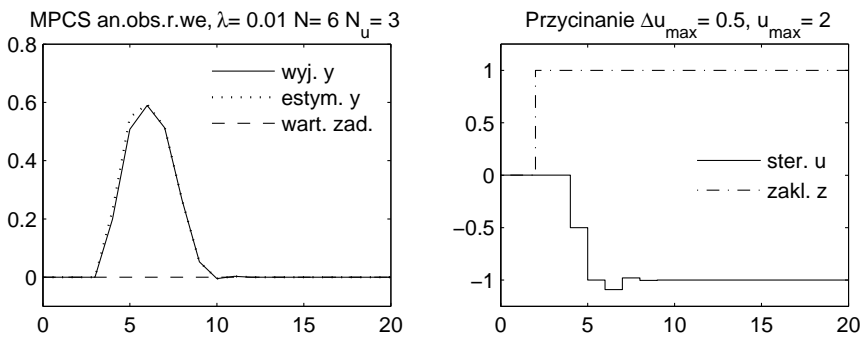
Rysunek 3.47. Odpowiedź na skok wartości zadanej w układzie regulacji MPC z obserwatorem stanu obiektu, ze wzmacnieniem obiektu zwiększonym o 20%



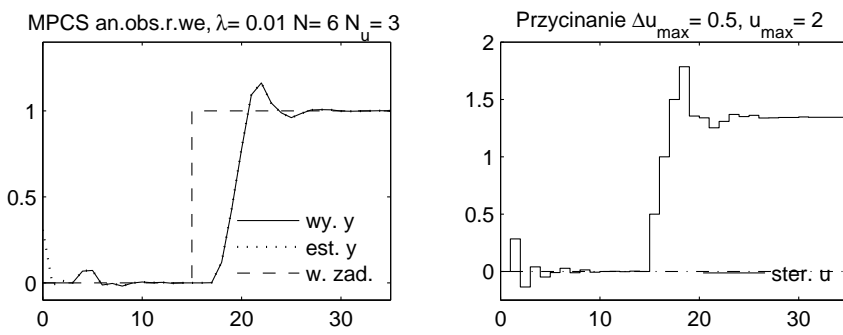
Rysunek 3.48. Odpowiedź na skok wartości zadanej w układzie regulacji MPC z obserwatorem stanu obiektu, ze wzmacnieniem obiektu zwiększonym o 30%



Rysunek 3.49. Odpowiedź na skok jednostkowy zakłócenia na wejściu obiektu w układzie regulacji MPCs z obserwatorem stanu obiektu



Rysunek 3.50. Odpowiedź na skok jednostkowy zakłócenia na wejściu obiektu w układzie regulacji MPCs z obserwatorem stanu obiektu rozszerzonego o stan zakłócenia na wejściu



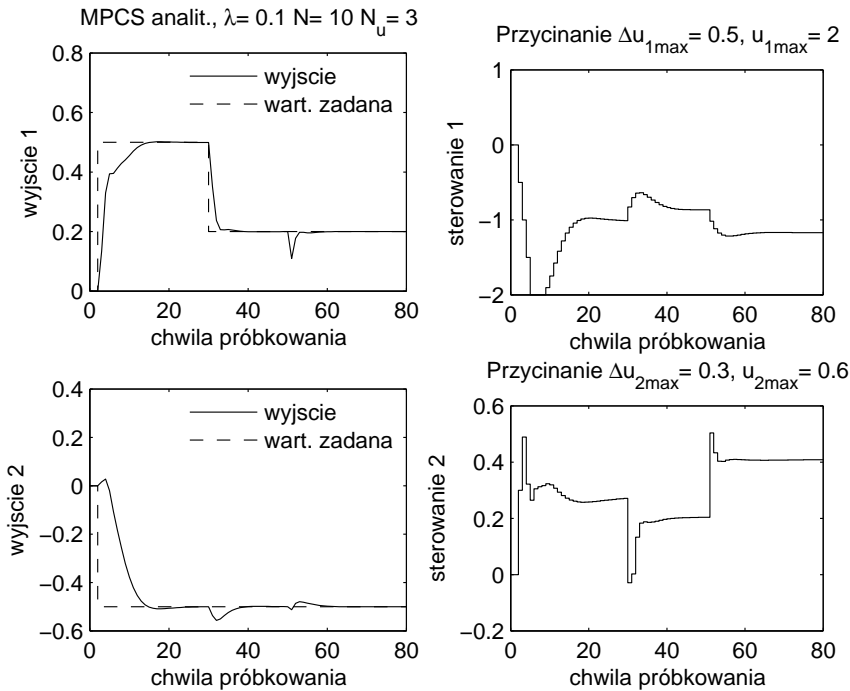
Rysunek 3.51. Odpowiedź na skok jednostkowy wartości zadanej w układzie regulacji MPCs z obserwatorem stanu obiektu rozszerzonego o stan zakłócenia na wejściu, ze wzmocnieniem obiektu zwiększonym o 20%

natomiast, że trajektorie wyjścia i sterowania są praktycznie identyczne z tymi na rysunku 3.49. Z kolei rysunek 3.51 przedstawia wyniki symulacji w warunkach takich samych jak uzyskane dla sytuacji z obserwatorem stanu samego obiektu przedstawionych na rysunku 3.47. Widać drobne różnice w początkowym okresie czasu odpowiadającym dostrojeniu stanu obserwatorów, natomiast trajektorie wyjścia i sterowania w odpowiedzi na skok jednostkowy wartości zadanej są praktycznie identyczne.  $\square$

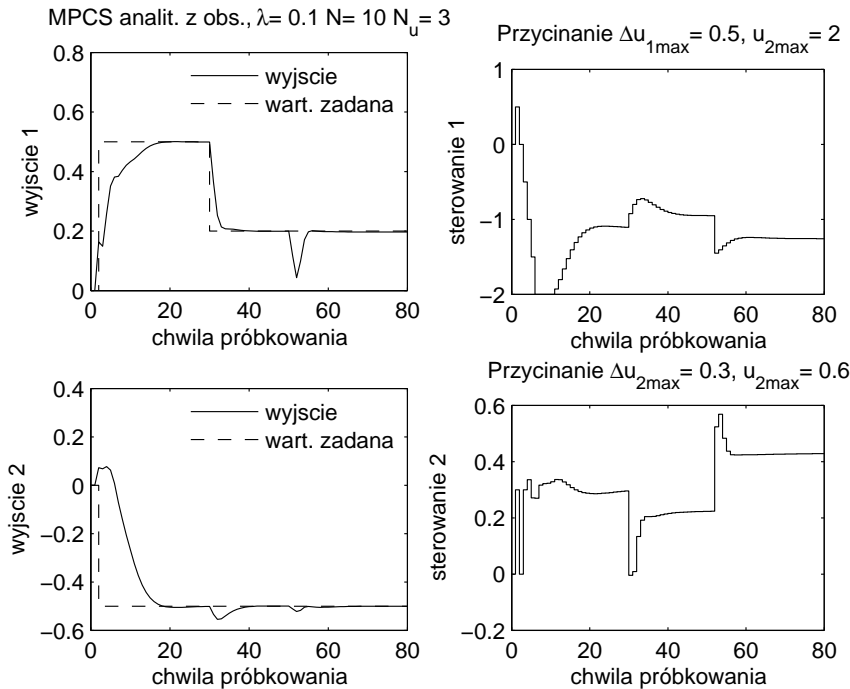
### Przykład 3.8.

Rozważmy model obiektu dwuwymiarowego przedstawiony w Przykładzie 3.6, z minimalnowymiarową reprezentacją stanu (Przypadek 2). Rysunek 3.52 przedstawia wyniki symulacji w przypadku stanu mierzonego (bez obserwatora), gdzie oprócz widocznych skoków wartości zadanych, w chwili  $k = 50$  nastąpił skok zakłóceń na wejściach obiektu: na wejściu pierwszym od zera do 0.3, a na wejściu drugim od zera do -0.2.

Następnie, korzystając z funkcji "place" pakietu Matlab, wyznaczono



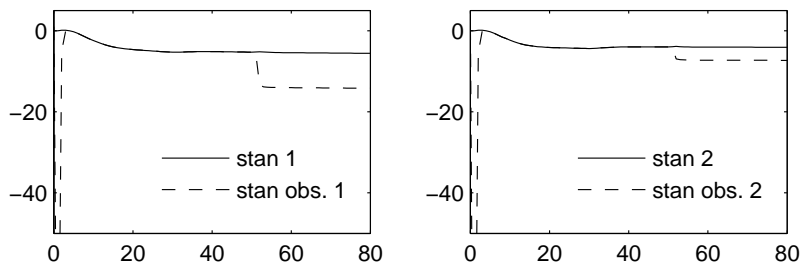
Rysunek 3.52. Wyniki symulacji z regulatorem MPCŚS analitycznym z pomiarem stanu, z ograniczeniami sterowania uwzględnianymi przez przycinanie, obiekt 2x2 z modelem (3.124)



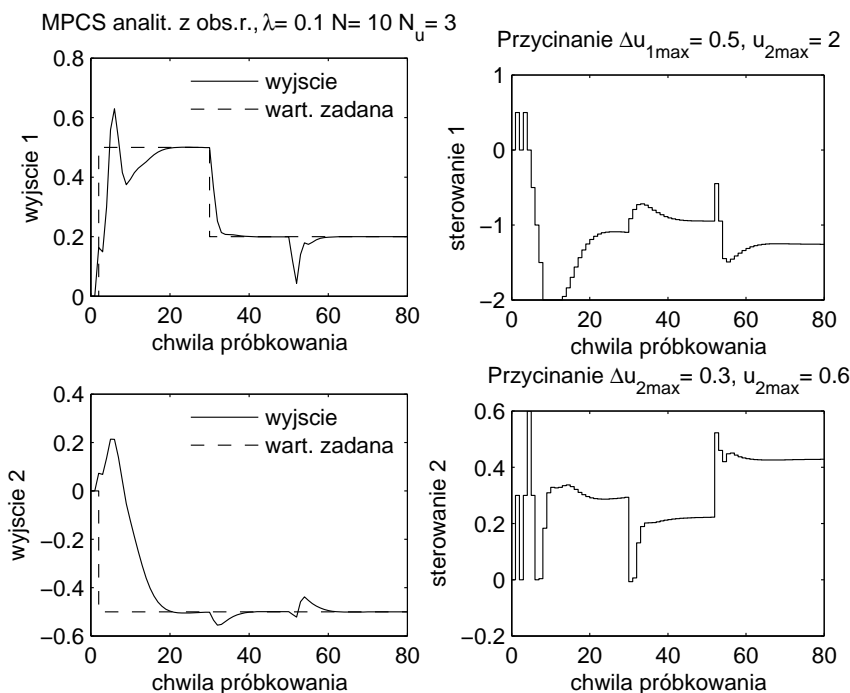
Rysunek 3.53. Wyniki symulacji z regulatorem MPCs analitycznym z obserwatorem stanu (3.125), z ograniczeniami sterowania uwzględnianymi przez przycinanie, obiekt 2x2 z modelem (3.124)

macierz wzmocnień  $\mathbf{K}_{obs}$  obserwatora stanu (3.125), zakładając bieguny dynamiki obserwatora o wartościach 0.02, 0.021, 0.023 i 0.024 ( $\mathbf{K}_{obs} = \text{place}(\mathbf{A}', \mathbf{C}', [0.02 \ 0.021 \ 0.023 \ 0.024])'$ ). Na rysunku 3.53 przedstawiono wyniki symulacji w identycznych warunkach jak poprzednio, ale z wyznaczonym obserwatorem stanu, startującym z wartości początkowej  $[1 \ 1 \ 1 \ 1]$ , przy stanie początkowym obiektu  $[0 \ 0 \ 0 \ 0]$ . Natomiast rysunek 3.54 pokazuje przebiegi dwóch pierwszych składowych stanu obiektu i wyjścia obserwatora (amplituda estymat stanu w pierwszych chwilach symulacji wykracza znacznie poza skalę na rysunkach przyjętą tak, aby dobrze pokazać różnice na ostatnim odcinku symulacji) – widać, że po pojawieniu się skoku zakłóceń na wejściu obiektu, obserwator wyznacza stan przesunięty, zgodnie z zal. (3.132). Regulator zaś działa poprawnie, realizując wartości zadane, tłumiąc wpływ zakłócenia.

Zrealizowano też układ regulacji z obserwatorem stanu rozszerzonego (3.138), wyniki przedstawia rysunek 3.55, dla identycznego jak poprzednio scenariusza zmian wartości zadanych i zakłóceń skokowych na wejściu



Rysunek 3.54. Przebiegi dwóch pierwszych składowych stanu i wyjścia obserwatora, symulacja z rys. 3.53



Rysunek 3.55. Wyniki symulacji z regulatorem MPC analytycznym z rozszerzonym obserwatorem stanu, z ograniczeniami sterowania uwzględnianymi przez przycinanie, obiekt 2x2 z modelem (3.124)

obiektu. Bieguny obserwatora rozszerzonego stanu założono na podobnym poziomie wartości, co poprzednio, konkretnie założono bieguny o wartościach 0.02, 0.021, 0.023, 0.024, 0.025 i 0.028. Obserwator startował z wartości początkowych  $[1 \ 1 \ 1 \ 1 \ 0 \ 0]$ , przy stanie początkowym obiektu  $[0 \ 0 \ 0 \ 0]$ . Tym razem obserwacja stanu jest dokładna, prawidłowo są też estymowane

zakłócenia wejścia. Okazało się jednakże, że wartości elementów macierzy wzmocnień obserwatora rozszerzonego  $K_{obsr}$  są o rząd wielkości większe od elementów macierzy obserwatora (nierozszerzonego)  $K_{obs}$ . Skutkuje to większą wrażliwością na błąd stanu początkowego, stąd większymi błędami estymacji stanu i uchybami regulacji w początkowym okresie symulacji (okresie redukcji błędu estymacji stanu). Ponadto, z testów porównawczych wynika, że układ regulacji z rozszerzonym obserwatorem stanu jest bardziej wrażliwy na błędy modelowania obiektu.

Podobnie jak w poprzednim przykładzie 3.7, wyniki uzyskane w niniejszym przykładzie pokazują, że jakość regulacji w układzie z bezpośrednim pomiarem stanu jest lepsza niż w układach regulacji z obserwatorami stanu.  $\square$

W rozdziale niniejszym, poświęconym algorytmom MPCs, zajmowaliśmy się dotąd przede wszystkim algorytmami wyznaczania składowej swobodnej  $\mathcal{Y}^0(k)$  prognozowanej trajektorii wyjść na horyzoncie predykcji, dla kilku wariantów: pomiaru stanu, estymacji stanu samego obiektu i estymacji stanu rozszerzonego o zmienne modelujące zakłócenia deterministyczne. Niezależnie bowiem jakiej sytuacji odpowiada wyznaczona składowa swobodna  $\mathcal{Y}^0(k)$ , dalej postępowanie jest takie samo: wyznaczamy przyrost sterowania stosując albo prawo regulacji z podmacierzą  $\bar{\mathbf{K}}_1$  macierzy  $\mathbf{K}$  identyczną dla wszystkich wariantów (macierz  $\mathbf{K}$  zależy tylko od macierzy dynamicznej  $\mathbf{M}$  i współczynników wagowych w funkcji celu), albo rozwiązujemy zadanie numerycznego wyznaczania przyrostu sterowania z rozwiązaniem zadania optymalizacji, w którym  $\mathcal{Y}^0(k)$  jest jedynie wektorem parametrów. Postacie praw sterowania regulatora MPCs dla wszystkich wymienionych wyżej wariantów podawaliśmy wcześniej w niniejszym rozdziale, po wyprowadzeniu odpowiedniego wzoru na  $\mathcal{Y}^0(k)$ . Natomiast zadanie optymalizacji regulatora numerycznego MPCs ma standardową postać, jak dla wszystkich algorytmów MPC rozważanych dotąd w niniejszym rozdziale, a mianowicie:

$$\begin{aligned} \min_{\Delta \mathcal{U}(k)} \left\{ \left\| [\mathcal{Y}^{sp}(k) - \mathcal{Y}^0(k)] - \tilde{\mathbf{C}}\mathbf{M}_x \Delta \mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \left\| \Delta \mathcal{U}(k) \right\|_{\underline{\Lambda}}^2 \right\} \\ \text{z ogr.: } -\Delta \mathcal{U}_{\max} \leq \Delta \mathcal{U}(k) \leq \Delta \mathcal{U}_{\max}, \\ \mathcal{U}_{\min} \leq \mathcal{U}(k-1) + \mathbf{J}\Delta \mathcal{U}(k) \leq \mathcal{U}_{\max}, \\ \mathcal{Y}_{\min} \leq \mathcal{Y}^0(k) + \tilde{\mathbf{C}}\mathbf{M}_x \Delta \mathcal{U}(k) \leq \mathcal{Y}_{\max}, \end{aligned} \quad (3.164)$$

gdzie  $\tilde{\mathbf{C}}\mathbf{M}_x = \mathbf{M}$  jest macierzą dynamiczną (3.35). Powyższe zadanie optymalizacji regulatora MPCs można łatwo przeformułować do standardo-

wej postaci zadania programowania kwadratowego, tak jak to pokazaliśmy w rozdziale 3.2.3.

Porównując wyniki uzyskane dla układów regulacji MPCs z pomiarem stanu i z obserwatorami stanu (stanu obiektu i stanu rozszerzonego), zaprezentowane w przykładach 3.6 i 3.7 wyraźnie widać, że jakość regulacji w układzie regulacji z pomiarem stanu jest wyraźnie lepsza zarówno w nadążaniu za trajektorią (po skoku wartości zadanej) jak i przy tłumieniu zakłócenia oddziałującego na wejście obiektu. Przyczyną tej sytuacji jest fakt, że zastosowany model zakłóceń stanu  $v(k)$  i sposób jego wyliczania przy bezpośrednim pomiarze stanu jest w stanie szybciej zareagować na wpływ zakłóceń objawiający się najpierw zmianami wartości zmiennych stanu, a dopiero potem zmianami wyjścia obiektu – w porównaniu do wartości  $v(k)$  czy estymaty zakłócenia  $\hat{z}(k)$  obliczanych na podstawie estymaty stanu samego obiektu czy stanu rozszerzonego uzyskiwanych z obserwatora. W układach regulacji z obserwatorami regulator dostaje bowiem informacje o pojawieniu się zakłócenia dopiero w momencie jego oddziaływania na wyjście obiektu – są to układy regulacji ze sprzężeniem od wyjścia obiektu, nie od stanu. Opisane zjawisko dotyczy zarówno zakłóceń wynikających z błędów modelowania, jak i zakłócenia zewnętrznego działającego na wejście obiektu. Przedstawiony fakt manifestuje się jeszcze wyraźniej, gdy porównamy rysunki 3.47, 3.51 i 3.49, 3.50 z odpowiadającymi im rysunkami dla układów regulacji DMC i GPC, odpowiednio z rysunkami 3.19, 3.28 i 3.20, 3.27. Widać prawie identyczne przebiegi – układy regulacji DMC i GPC są też układami ze sprzężeniem od wyjścia, z tym samym obiektem opisywanym równoważnymi modelami liniowymi.

Porównując natomiast algorytmy MPCs z estymacją stanu, stanu samego obiektu i stanu rozszerzonego, trzeba zaznaczyć, że projektowanie układu regulacji z estymacją stanu samego obiektu jest prostsze – nie trzeba zastanawiać się nad przyjęciem miejsc oddziaływania zakłóceń na obiekt przy ograniczeniu ich maksymalnej liczby (ograniczający warunek obserwowalności obiektu rozszerzonego). Model zakłóceń stanu  $v(k)$  jest bowiem jednoznacznie i prosto zdefiniowany dla każdego równania stanu. Ponadto, obserwator stanu samego obiektu ma mniejszy wymiar, jest prostszy do projektowania. Natomiast układ regulacji z estymacją stanu rozszerzonego jest bardziej ogólny, umożliwia również estymację zakłóceń nie tylko skokowo (i rzadko) zmiennych, ale i zmiennych w sposób ciągły. W pracy [87] poświęconej głównie algorytmom regulacji predykcyjnej z modelami w przestrzeni stanów znaleźć można przykład projektowania algorytmu MPCs dla zakłóceń okresowych.

Porównując cechy algorytmu MPCs z algorytmami DMC czy GPC,

zwraca uwagę inne podejście do modelowania obiektu. Modele wykorzystywane w algorytmach DMC i GPC to modele typu wejście-wyjście, w algorytmie DMC jest to model nieparametryczny dany odpowiedziami skokowymi, zaś w algorytmie GPC model transmitancji dyskretnych (równań różnicowych). Modelowanie równaniami stanu jest, jak wiadomo, podejściem ogólniejszym, ale przy braku możliwości pomiaru całego wektora stanu wymaga zastosowania obserwatora stanu (filtru Kalmana). Obliczenia prowadzące do określenia wartości przewidywanych wyjść na horyzoncie predykcyjnym wykonywane są, konsekwentnie, na bazie równań stanu. Prowadzi to do dość złożonych zależności o niezbyt korzystnych własnościach obliczeniowych, numerycznych, szczególnie przy wymiarze wektora stanu istotnie większym od wymiaru wektora wyjść regulowanych i przy długim horyzoncie predykcyjnym. W przypadku zastosowania obserwatora stanu rozszerzona wymieniona cecha ulega jeszcze pogłębieniu. Sytuacja jest szczególnie niekorzystna przy modelach obiektów ze słabo uwarunkowanymi macierzami stanu  $\mathbf{A}$  (wartości własne bliskie jedności), czy wręcz niestabilnych (wartości własne większe od jedności). W przypadkach tych zaleca się wstępne wystabilizowanie obiektu np. poprzez liniowe sprzężenie od stanu, a następnie zastosowanie regulatora MPC do obiektu wstępnie wystabilizowanego, zob. np. [127].

Modelowanie dynamiki układem równań stanu jest w przypadku np. wielu modeli obiektów elektromechanicznych naturalne. Natomiast dla procesów przemysłowych, gdzie modele dla celów regulacji są konstruowane przede wszystkim za pomocą struktur typu wejście-wyjście, nie jest rozwiązaniem najbardziej typowym. Biorąc jednakże pod uwagę możliwość uzyskania lepszej jakości regulacji przy zastosowaniu regulatora MPC z pomiarem stanu, wartym rozważania jest wykorzystanie modelu równań stanu ze stanem złożonym z wartości aktualnych i opóźnionych wyjść i opóźnionych wejść procesu, zob. definicja (3.116). Stan taki jest z definicji w pełni mierzalny, choć na ogół jego wymiar nie jest minimalny.

### 3.5. Algorytmy nieliniowe regulacji predykcyjnej

#### 3.5.1. Struktury nieliniowych algorytmów MPC

Zastosowanie predykcyjnych algorytmów regulacji z liniowymi modelami obiektów było niewątpliwie kamieniem milowym w rozwoju techniki regulacji automatycznej. Rzeczywiste obiekty regulacji są jednak z reguły nieliniowe i nie zawsze wystarcza regulacja liniowa w otoczeniu założonego punktu pracy. Szczególnie w sytuacji optymalnego sterowania w struktu-

rze warstwowej, kiedy optymalizator na bieżąco zmienia punkty pracy dostosowując je do wartości wejść niesterowanych, potrzebna jest regulacja nieliniowa, zob. rozdz. 1 i 4. Trajektorie z założenia zmienne w czasie występują przy sterowaniu procesów wsadowych (szarżowych). Stąd już w latach osiemdziesiątych ub. stulecia pojawiły się próby sformułowań algorytmów MPC dla obiektów nieliniowych, szczególnie w obszarze badań związanych z aplikacjami algorytmu DMC w przemyśle chemicznym. Dominowało podejście oparte na sukcesywnej linearyzacji zapoczątkowane przez Garcję [50], zob. [6, 56]. Dekada lat dziewięćdziesiątych to okres rozwoju badań nad nieliniowymi algorytmami regulacji, zob. artykuły przeglądowe [56, 1, 98].

Zastosowania algorytmów MPC z pełnym modelem nieliniowym wykorzystywanym w zadaniu optymalizacji rozwiązywanym w każdej chwili próbkowania wybraną procedurą programowania nieliniowego były i nadal są przedmiotem wielu prac badawczych. Algorytmy takie mają zastosowanie przede wszystkim do obiektów silnie nieliniowych i wówczas, gdy należy się spodziewać sterowania agresywnego, o większych amplitudach. Podstawowe problemy związane z projektowaniem i implementacją algorytmów z nieliniową, niekwadratową optymalizacją w każdej chwili próbkowania omówimy krótko w początkowej części rozdziału.

Tendencją wyraźnie widoczną w pracach o charakterze bardziej aplikacyjnym jest dążenie do stosowania, w przypadku potrzeby modelowania i regulacji nieliniowej, *algorytmów MPC z linearyzacją*. M. Morari i J.H. Lee w artykule przeglądowym [102] stwierdzają, że „linearyzacja jest jedyną metodą, która znalazła szersze zastosowanie w przemyśle, pomijając projekty demonstracyjne. Dla przemysłu musi być jasno uzasadniona potrzeba rozwiązywania *on-line* nieliniowych zadań dynamicznych, a jak dotąd nie ma przykładów wykazujących taką potrzebę w sposób przekonujący” (tłumaczenie autora). Stosując linearyzację dążymy do takiego sformułowania algorytmów predykcyjnych, aby wykorzystując model nieliniowy rozwiązywać jednakże w każdym kroku odpowiednio skonstruowane *zadanie wypukłego programowania kwadratowego*, zadanie z reguły z kroku na krok modyfikowane, dopasowywane do nieliniowości obiektu, czy nawet kilkakrotnie rozwiązywane w pojedynczym kroku algorytmu. Przytoczyć tu należy rozwiązania dążące do wykorzystania ogólnych modeli nieliniowych w strukturze algorytmu predykcyjnego poprzez odpowiednie linearyzacje w każdej chwili próbkowania, zob. [83, 53, 79, 107, 99, 152, 72, 73], w pracy przeglądowej [56] znaleźć można ponadto obszerniejszą bibliografię.

Należy zaznaczyć, że o ile w ostatniej dekadzie pojawiło się bardzo wiele prac dotyczących analizy stabilności rozmaicie sformułowanych algo-

rytmów wykorzystujących ogólny model nieliniowy, to praktycznie nie ma takich prac w obszarze algorytmów z linearyzacją. Pogląd ten potwierdzają M. Morari i J.H. Lee we wspomnianym powyżej artykule przeglądowym, pisząc że "theoretical purists tend to stay away from linearization approaches" (teoretycy stronią od badania podejść linearyzacyjnych). Niewątpliwie, jedną z podstawowych przyczyn tego stanu rzeczy jest fakt, że algorytmy z linearyzacjami są *suboptimalne* – a tego typu algorytmy z reguły znacznie trudniej poddają się analizie formalnej. Jednakże, już od początku powstania i rozwoju algorytmów MPC brak ścisłych analiz teoretycznych nie jest czynnikiem hamującym ich rozwój i zastosowania.

Struktury nieliniowych, suboptimalnych algorytmów MPC wykorzystujących odpowiednie linearyzacje modelu nieliniowego będą głównym przedmiotem rozważań niniejszego rozdziału.

### Algorytmy MPC-NO (z nieliniową optymalizacją)

Na pierwszy rzut oka, *idea przejścia od liniowej do nieliniowej regulacji predykcyjnej* wydaje się prosta: pozostawiając strukturę, zasadę wyznaczania sterowania niezmienną należy *jedynie* do przewidywania wyjść na horyzoncie predykcji używać, zamiast liniowego, *nieliniowego* modelu obiektu. Prowadzi to do sformułowania zdania optymalizacji w postaci (porównaj z (3.167))

$$\min_{\Delta u(k|k), \dots, \Delta u(k+N_u-1|k)} \left\{ J(k) = \sum_{p=N_1}^N \|y^{zad}(k+p|k) - y(k+p|k)\|_{\Psi(p)}^2 + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\Lambda(p)}^2 \right\}$$

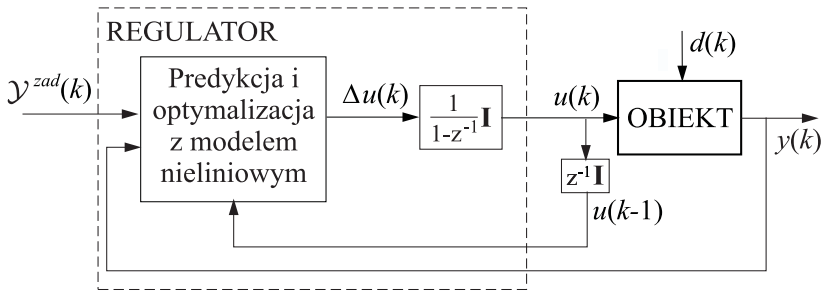
z ogr. : (3.6), (3.7) i (3.8),

gdzie wartości  $y(k+p|k)$  wyliczane są *modelem nieliniowym*.

(3.165)

Strukturę pełnej nieliniowej regulacji MPC, z predykcją całej trajektorii wyjść opartą na modelu nieliniowym określać będziemy jako *MPC-NO* (*MPC with Nonlinear Optimization*). Jest ona zilustrowana na rysunku 3.56, gdzie  $\mathcal{Y}^{zad}(k) = [y^{zad}(k+N_1|k)^T \dots y^{zad}(k+N|k)^T]^T$  reprezentuje trajektorię wartości zadanych na horyzoncie predykcji.

Z pozoru prosta zmiana polegająca na użyciu, zamiast liniowego, nieliniowego modelu obiektu jest jednak fundamentalna. Nieliniowa zależność wartości wyjść przewidywanych  $y(k+p|k)$  od zmiennych decyzyjnych decyduje o tym, że zadanie optymalizacji (3.165) jest *niekwadratowe i w ogólności niewypukłe*. Do rozwiązywania takich zadań nie ma uniwersalnych



Rysunek 3.56. Struktura pełnej nieliniowej regulacji MPC (z predykcją i optymalizacją opartą na nieliniowym modelu – struktura MPC-NO)

procedur minimalizacji zapewniających znalezienie rozwiązania niezawodnie i szybko, a co najważniejsze, w przewidywalnym czasie i z gwarantowaną dokładnością. Ponadto, bardziej efektywne procedury gradientowe znajdują z reguły jedynie minima lokalne. Wymienione fakty powodują, że algorytmy MPC z modelem nieliniowym w zadaniu optymalizacji rozwiązywanym w każdym kroku wybraną procedurą programowania nieliniowego znajdują zastosowanie przede wszystkim do obiektów silnie nieliniowych i wówczas, gdy należy się spodziewać sterowania agresywnego, o większych amplitudach. Henson w artykule przeglądowym [56] podał jedne z pierwszych list przykładów symulacyjnego i, znacznie już krótszą, eksperymentalnego badania nieliniowych algorytmów predykcyjnych.

Nieliniowy modelu procesu może być dany np. w postaci nieliniowego równania różnicowego:

$$y(k+1) = g(y(k), \dots, y(k-n_A), u(k), \dots, u(k-n_B-1)),$$

lub nieliniowych równań stanu:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)), \\ y(k) &= g(x(k)). \end{aligned}$$

Nieliniowe zadanie optymalizacji dynamicznej dla regulatora MPC-NO jest często wygodniej formułować przyjmując wartości (nie przyrosty) sterowań jako zmienne decyzyjne, tzn. wektor zmiennych decyzyjnych postaci:

$$\mathcal{U}(k) = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N_u-1|k) \end{bmatrix}. \quad (3.166)$$

**Zadanie optymalizacji dynamicznej regulatora MPC** przyjmuje wówczas postać:

$$\min_{u(k|k), \dots, u(k+N_u-1|k)} \left\{ J(k) = \sum_{p=1}^N \| [y^{zad}(k+p|k) - y(k+p|k)] \|^2 + \lambda \sum_{p=0}^{N_u-1} \| u(k+p|k) - u(k+p-1|k) \|^2 \right\}$$

z ogr. :  $u_{min} \leq u(k+p|k) \leq u_{max}, \quad p = 0, \dots, N_u - 1,$

$-\Delta u_{max} \leq u(k+p|k) - u(k+p-1|k) \leq \Delta u_{max}, \quad p = 0, \dots, N_u - 1,$

$y_{min} \leq y(k+p|k) \leq y_{max}, \quad p = 1, \dots, N,$  (3.167)

gdzie predykcje wyjść  $y(k+p|k)$ ,  $p = 1, \dots, N$ , liczone są z użyciem *nieliniowego modelu procesu*, zaś  $u(k-1|k) = u(k-1)$ .

Obliczanie predykowanej trajektorii wyjść wykonujemy **rekurencyjnie**, dla każdej aktualnej trajektorii sterowań  $u(k+p|k)$ ,  $p = 0, \dots, N-1$ , na horyzoncie predykcji – przy tym przypomnijmy, że jeśli  $N_c < N$ , to dla  $p = N_u, \dots, N-1$ :  $u(k+p|k) = u(k+N_u-1|k)$ .

Dla **nieliniowego równania różnicowego**

$$y(k+1) = g(y(k), \dots, y(k-n_A), u(k), \dots, u(k-n_B-1)), \quad (3.168)$$

obliczenia te wyglądają następująco:

$$y(k+1|k) = g(y(k), \dots, y(k-n_A), u(k|k), \dots, u(k-n_B-1)) + d(k),$$

$$y(k+2|k) = g(y(k+1|k), y(k), \dots, y(k-n_A+1), u(k+1|k), \dots, u(k-n_B)) + d(k),$$

$$y(k+3|k) = g(y(k+2|k), y(k+1|k), y(k), \dots, y(k-n_A+2), u(k+2|k), u(k+1|k), \dots, u(k-n_B+1)) + d(k),$$

⋮

$$y(k+N|k) = g(y(k+N-1|k), y(k+N-2|k), \dots, y(k+N-1-n_A|k), u(k+N-1|k), \dots, u(k+N-2-n_B|k)) + d(k), \quad (3.169)$$

gdzie  $d(k)$  jest estymatą zakłóceń stanu (niemierzalnych):

$$\begin{aligned} d(k) &= y(k) - y(k|k-1) \\ &= y(k) - g(y(k-1), \dots, y(k-n_A-1), u(k-1), \dots, u(k-n_B-2)). \end{aligned}$$

Dla nieliniowego równania stanu:

$$\begin{aligned}x(k+1) &= f(x(k), u(k)), \\y(k) &= g(x(k)),\end{aligned}\tag{3.170}$$

predykcja wartości zmiennych stanu i zmiennych wyjściowych wygląda następująco (dla każdej aktualnej trajektorii sterowań  $u(k+p|k)$ ,  $p = 0, \dots, N-1$ ):

$$\begin{aligned}x(k+1|k) &= f(x(k), u(k|k)) + v(k), & y(k+1|k) &= g(x(k+1|k)), \\x(k+2|k) &= f(x(k+1|k), u(k+1|k)) + v(k), & y(k+2|k) &= g(x(k+2|k)), \\&\vdots & & \\x(k+N|k) &= f(x(k+N-1|k), u(k+N-1|k)) + v(k), & y(k+N|k) &= g(x(k+N|k)),\end{aligned}\tag{3.171}$$

gdzie  $v(k)$  jest estymatą zakłóceń stanu (niemierzalnych) :

$$\begin{aligned}v(k) &= x(k) - x(k|k-1) \\&= x(k) - f(x(k-1), u(k-1)).\end{aligned}$$

Przy *estymacji stanu*, do równań powyższych należy zamiast  $x(k)$  wstawić estymatę  $\hat{x}(k)$ .

Jeśli estymujemy stan samego procesu  $x(k)$  i występują niemierzalne zakłócenia deterministyczne, to dla zagwarantowania zerowych błędów w stanach ustalonych (*offset-free control*) predykcję wyjść należy korygować przez dodanie odpowiedniej poprawki,

$$y(k+p|k) = g(x(k+p|k)) + [y(k) - g(\hat{x}(k))], \quad p = 1, \dots, N, \tag{3.172}$$

gdzie poprawka występuje w nawiasie kwadratowym. Przyczyna potrzeby korekcji i sposób postępowania są analogiczne, jak przedstawione uprzednio dla regulatora MPC z liniowymi równaniami stanu.

Przy numerycznej optymalizacji nieliniowego zadania MPC *ważny jest punkt startowy dla procedury optymalizacji*. Regulator MPC rozwiązuje w kolejnych krokach (okresach impulsowania) zadania nieliniowe, różniące się jedynie wartościami pewnych parametrów – ostatnich pomiarów wyjść (stanu) i zakłóceń mierzalnych oraz zastosowanych wartości sterowań. W większości sytuacji trajektorie optymalne z sąsiednich kroków nie powinny się wiele różnić – stąd zaleca się stosowanie *zasady "gorącego startu"*:

wartości optymalne sterowań z poprzedniego kroku stają się podstawą punktu startowego w kroku następnym. Dokładniej, zasada ta w zastosowaniu do zadania optymalizacji MPC polega na tym, że przyjmując *trajektorię początkową sterowań*  $u^0(k+p|k)$ ,  $p = 0, \dots, N-1$ , dla kroku  $k$  wykorzystujemy *trajektorię optymalną sterowań wyliczoną w poprzednim kroku*,  $(k-1)$ -szym – przy czym skracamy ją o pierwsze sterowanie (już zastosowane) i wydłużamy o jedno sterowanie, przez powielenie:

$$\mathcal{U}^0(k) = \begin{bmatrix} u^0(k|k) \\ u^0(k+1|k) \\ \vdots \\ u^0(k+N-3|k) \\ u^0(k+N-2|k) \\ u^0(k+N-1|k) \end{bmatrix} = \begin{bmatrix} \hat{u}(k|k-1) \\ \hat{u}(k+1|k-1) \\ \vdots \\ \hat{u}(k+N-3|k-1) \\ \hat{u}(k+N-2|k-1) \\ \hat{u}(k+N-1|k-1) \end{bmatrix}. \quad (3.173)$$

Podkreślmy, że w algorytmie MPC-NO w każdym kroku (okresie impulsowania) *predykcja trajektorii wyjść obliczanych modelem nieliniowym wykonywana jest wielokrotnie*, dla każdego kolejnego wektora zmiennych decyzyjnych wyznaczanych przez algorytmu optymalizacji, poczynając od trajektorii początkowej  $\mathcal{U}^0(k)$  i kończąc na trajektorii optymalnej  $\hat{\mathcal{U}}(k)$ .

Przy projektowaniu estymatora stanu procesu nieliniowego sytuacja jest następująca: mamy nieznaną dokładnie *nieliniowy proces*, opisany przez (nieznane) nieliniowe równania stanu:

$$\begin{aligned} x(k+1) &= f_p(x(k), u(k)), \\ y(k) &= g_p(x(k)), \end{aligned} \quad (3.174)$$

oraz *model procesu* (znany):

$$\begin{aligned} x(k+1) &= f(x(k), u(k)), \\ y(k) &= g(x(k)), \end{aligned} \quad (3.175)$$

z ew. uzupełnieniami o zakłócenia stochastyczne (szумы) stanu i wyjścia. Estymujemy stan na podstawie modelu, podstawowe metody to:

- rozszerzony obserwator Luenbergera (ELO – Extended Luenberger Observer), skuteczny gdy szумы są pomijalne,
- rozszerzony filtr Kalmana (EKF – Extended Kalman Filter).

*Rozszerzony obserwator Luenbergera* jest postaci:

$$\hat{x}(k+1) = f(\hat{x}(k), u(k)) + \mathbf{K}_{ELO}[y(k) - g(\hat{x}(k))],$$

gdzie macierz wzmocnień  $\mathbf{K}_{ELO}$  wyznaczamy dla linearyzacji modelu nieliniowego w punkcie pracy (jak dla klasycznego obserwatora Luenbergera dla modelu liniowego). Zastosować można ELO rozmyte o strukturze Takagi-Sugeno: obszar zmienności stanu dzielimy wówczas na podobszary i określamy funkcje przynależności stanu do podobszarów, w każdym podobszarze wyznaczamy linearyzację modelu i macierz wzmocnień obserwatora – macierz wzmocnień rozmytego ELO jest rozmytą kombinacją wzmocnień z podobszarów aktywnych w aktualnym stanie.

*Rozszerzony filtr Kalmana (EKF)* projektujemy dla nieliniowych równań stanu postaci:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) + \varepsilon_x(k), \\ y(k) &= g(x(k)) + \varepsilon_y(k), \end{aligned} \quad (3.176)$$

gdzie szумы  $\varepsilon_x$  i  $\varepsilon_y$  są białe gaussowskie, o macierzach kowariancji  $\mathbf{Q}$  i  $\mathbf{R}$  (w ogólności mogą być niestacjonarne,  $\mathbf{Q}_k$  i  $\mathbf{R}_k$ ).

Wartości początkowe dla iteracji  $k$ -tej rozszerzonego filtru Kalmana to wartość estymaty stanu  $\hat{x}(k-1)$  i wartość estymaty macierzy kowariancji błędu estymacji  $\mathbf{P}_{k-1}$ . Równania rozszerzonego filtru Kalmana wyglądają następująco:

$$\text{predykcja : } \hat{x}(k|k-1) = f(\hat{x}(k-1), u(k-1)),$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q},$$

$$\text{macierz wzmocnień : } \mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_{k|k-1}^T(\mathbf{C}_{k|k-1}\mathbf{P}_{k|k-1}\mathbf{C}_{k|k-1}^T + \mathbf{R})^{-1},$$

$$\text{korekcja : } \hat{x}(k) = \hat{x}(k|k-1) + \mathbf{K}_k[y(k) - g(\hat{x}(k|k-1))],$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_{k|k-1})\mathbf{P}_{k|k-1},$$

gdzie macierze  $\mathbf{A}_{k-1}$  i  $\mathbf{C}_{k|k-1}$  wynikają z linearyzacji funkcji nieliniowych wokół aktualnej estymaty stanu i sterowania:

$$\mathbf{A}_{k-1} = \left[ \frac{\partial f_i}{\partial x_j}(\hat{x}(k-1), u(k-1)) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\hat{x}(k-1), u(k-1)) & \cdots & \frac{\partial f_1}{\partial x_{n_x}}(\hat{x}(k-1), u(k-1)) \\ \frac{\partial f_2}{\partial x_1}(\hat{x}(k-1), u(k-1)) & \cdots & \frac{\partial f_2}{\partial x_{n_x}}(\hat{x}(k-1), u(k-1)) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial x_1}(\hat{x}(k-1), u(k-1)) & \cdots & \frac{\partial f_{n_x}}{\partial x_{n_x}}(\hat{x}(k-1), u(k-1)) \end{bmatrix},$$

$$\mathbf{C}_{k|k-1} = \left[ \frac{\partial g_i}{\partial x_j}(\hat{x}(k|k-1)) \right] = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(\hat{x}(k|k-1)) & \cdots & \frac{\partial g_1}{\partial x_{n_x}}(\hat{x}(k|k-1)) \\ \frac{\partial g_2}{\partial x_1}(\hat{x}(k|k-1)) & \cdots & \frac{\partial g_2}{\partial x_{n_x}}(\hat{x}(k|k-1)) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_{n_x}}{\partial x_1}(\hat{x}(k|k-1)) & \cdots & \frac{\partial g_{n_x}}{\partial x_{n_x}}(\hat{x}(k|k-1)) \end{bmatrix}.$$

Rozszerzony filtr Kalmana *nie jest już estymatorem optymalnym* – staje się takim, gdy równania stanu i wyjścia są liniowe. Może być rozbieżny, np. przy źle dobranym punkcie początkowym, choć zdarza się to rzadko. Dlatego też jest najczęściej stosowany.

### Przykład 3.9.

Nieliniowy obiekt opisany jest następującym modelem równań stanu [103]:

$$x_{p1}(k+1) = \mathbf{0.95}x_{p1}(k) - \mathbf{0.25}x_{p1}(k)x_{p2}(k) + x_{p2}(k) + \mathbf{d}_1(\mathbf{k}), \quad (3.177a)$$

$$x_{p2}(k+1) = \mathbf{0.7}x_{p2}(k) + \mathbf{0.1}x_{p2}(\mathbf{k})\mathbf{d}_2(\mathbf{k}) + u(k), \quad (3.177b)$$

$$y_p(k) = x_{p1}(k) + \mathbf{d}_3(\mathbf{k}), \quad (3.177c)$$

gdzie  $x_p(k) = [x_{p1}(k) \ x_{p2}(k)]^T$  jest wektorem stanu, a  $d_1(k)$ ,  $d_2(k)$  i  $d_y(k)$  są nieznanymi (niemierzalnymi) zakłóceniami.

Do projektowania regulatora MPC-NO wykorzystano model obiektu:

$$x_1(k+1) = 0.9x_1(k) - 0.3x_1(k)x_2(k) + x_2(k), \quad (3.178a)$$

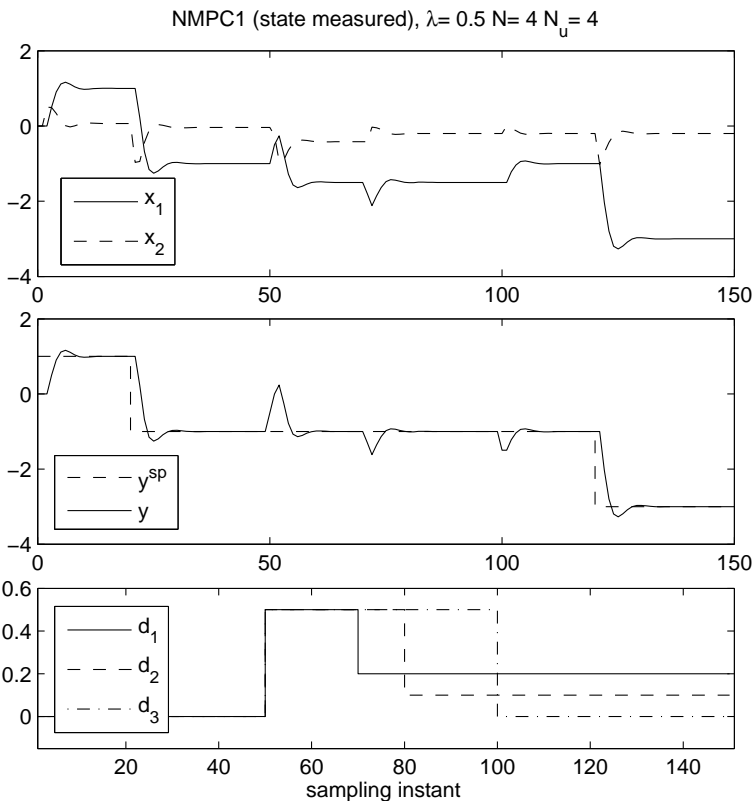
$$x_2(k+1) = 0.8x_2(k) + u(k), \quad (3.178b)$$

$$y(k) = x_1(k), \quad (3.178c)$$

gdzie  $x(k) = [x_1(k) \ x_2(k)]^T$  i  $y(k)$  są stanem i wyjściem, odpowiednio. W równaniach opisujących obiekt wyłuszczone różnice w stosunku do modelu: niedokładnie zidentyfikowane parametry, pominięty człon nieliniowy i zakłócenia addytywne.

Regulator MPC-NO zaprojektowano dla horyzontów predykcji i sterowania takich samych jak przyjęte w pracy [103], do ew. porównania rezultatów, tzn.  $N = 4$  i  $N_u = 4$  (choć przy  $N = 6$  i  $N_u = 3$  można uzyskać nieco lepsze rezultaty). Przyjęto jeden skalarny współczynnik wagowy w funkcji celu  $\lambda = 0.5$ , tzn.  $\Psi = \mathbf{I}$ ,  $\Lambda = \lambda\mathbf{I}$ .

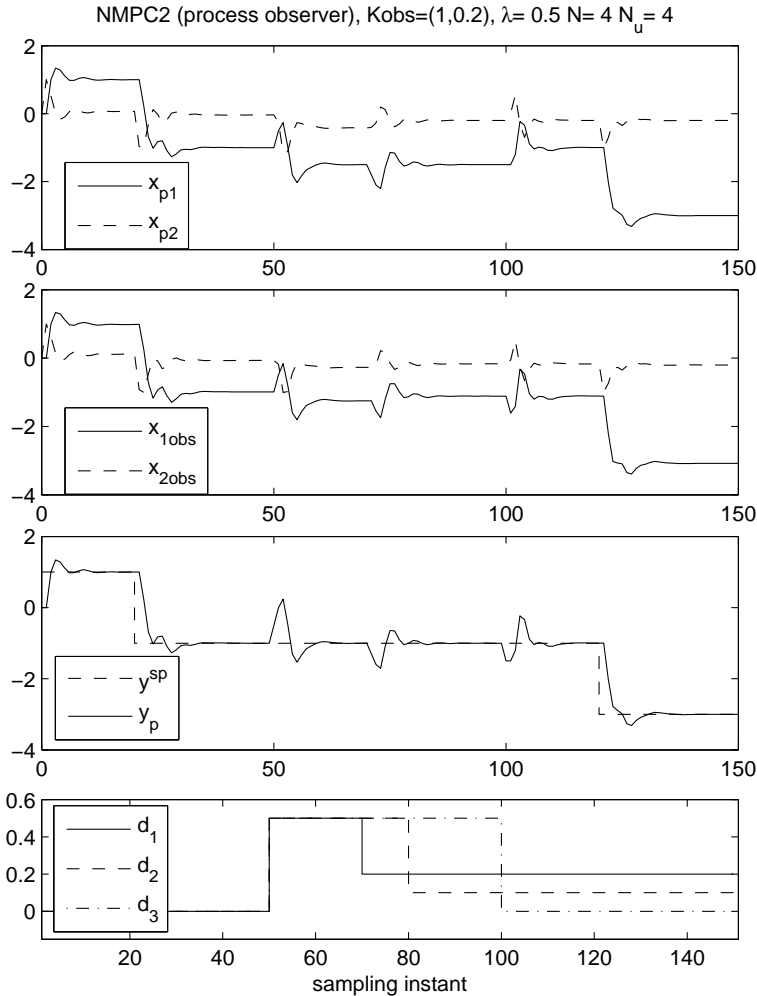
Dla przypadku z estymowanym stanem, do estymacji stanu procesu okazało się wystarczające zastosowanie rozszerzonego obserwatora Luenbergera (ELO) ze stałą macierzą wzmocnień, wyznaczoną dla zlinearyzowanego modelu obiektu w przyjętym punkcie pracy (standardowo, jak dla obserwatora Luenbergera dla modelu liniowego).



Rysunek 3.57. Algorytm z pomiarem stanu: trajektorie stanu, wartości zadanej i wyjścia, niemierzalnych zakłóceń (odpowiednio na kolejnych rysunkach).

Na rysunkach 3.57 i 3.58 przedstawiono wyniki symulacji pętli regulacji z obiektem (3.177a)-(3.177c) z nieliniowymi regulatorami MPCs wykorzystującymi uproszczony model (3.178a)-(3.178c). Rysunek 3.57 przedstawia wyniki z regulatorem wykorzystującym pomiar wektora stanu, zaś rysunek 3.58 wyniki z regulatorem wykorzystującym estymację wektora stanu obiektu (nie rozszerzonego o model zakłóceń), przy identycznym scenariuszu zmian wartości zadanej i zakłóceń. Porównując te wyniki łatwo zauważamy lepszą jakość regulacji w przypadku z pomiarem wektora stanu.  $\square$

Warto zaznaczyć, że jednym z obiecujących kierunków badań w obszarze nieliniowych algorytmów predykcyjnych jest zastosowanie modeli neuronowych o określonych strukturach, zob. np. [114, 75, 76, 70, 151, 71, 77, 72, 73]. Przy tym w pierwszej z tych prac nie sam obiekt, ale całe działanie algorytmu predykcyjnego jest modelowane odpowiednią, podlegającą uczeniu siecią neuronową. Interesującym podejściem jest również próba wy-



Rysunek 3.58. Algorytm z estymacją stanu: trajektorie stanu, estymat stanu, wartości zadanej i wyjścia, niemierzalnych zakłóceń (od góry, odpowiednio).

korzystania do nieliniowej, globalnej optymalizacji w każdym z kroków algorytmu MPC-NO metody programowania nieliniowego typu podziału i ograniczeń (*branch and bound method*), dokonując odpowiedniego przeformułowania zadania optymalizacji [131]. Dla zastosowania metody podziału i ograniczeń trzeba dokonać dyskretyzacji każdego ze sterowań  $\Delta u(k+p|k)$ ,  $p = 0, 1, \dots, N_u - 1$ , od poziomu tej dyskretyzacji zależy poziom suboptymalności rezultatów i nakład obliczeń. Aby ograniczyć ten nakład, zaproponowano w [158] dyskretyzacje jedynie na pewnych podzbiorach wokół wartości poprzednich, podzbiorach adaptacyjnie skalowanych z wykorzy-

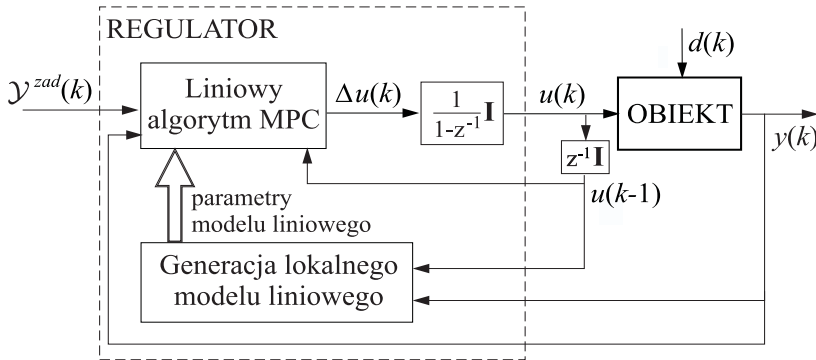
staniem logiki rozmytej. Podejście, aczkolwiek skuteczne, okazało się nadal wymagającym dużego nakładu obliczeń.

Zaznaczmy na koniec, że algorytmy klasy MPC-NO cieszą się największym zainteresowaniem teoretyków. Zaproponowano szereg struktur takich algorytmów i nadal pojawiają się koncepcje tego typu, często wraz z analizą stabilności. Do tematu tego powrócimy w rozdziale 3.6.1, dotyczącym mechanizmów zapewniania stabilności algorytmów predykcyjnych.

### Algorytmy MPC-NSL (nieliniowe z sukcesywną linearyzacją)

Już przy liniowych algorytmach MPC zaznaczaliśmy, że różnice między nimi wynikają z użycia różnych modeli obiektu do prognozowania wartości wyjść na horyzoncie predykcyjnym. Prowadziło to do różnych struktur i postaci formuł predykcyjnych składowej swobodnej trajektorii wyjść regulowanych oraz jedynie różnej postaci wzorów opisujących elementy tej samej macierzy dynamicznej wyznaczającej składową wymuszaną. W przypadku *modelu nieliniowego* obiektu, nieliniowe zależności określają trajektorie przewidywaną wyjść regulowanych  $y(k+p|k)$ ,  $p = 1, 2, \dots, N$ . Brak liniowości, a co za tym idzie nieobowiązywanie zasady superpozycji, uniemożliwia zachowującą optymalność dekompozycję tej trajektorii na nieliniowe składowe odpowiadające części swobodnej (zależnej od już zrealizowanych sterowań i zmierzonych wyjść) i części wymuszonej (zależnej od zmiennych decyzyjnych – wyznaczanych przyrostów sterowań). Wstawiając jednakże zerowe przyrosty sterowań na horyzoncie predykcyjnym, nieliniowe zależności opisujące trajektorie  $y(k+p|k)$  przekształcamy w *nieliniowy model odpowiedzi swobodnej* wyjść przewidywanych  $y^0(k+p|k)$ ,  $p = 1, 2, \dots, N$ . Postępowanie takie nie będzie jednakże konstruktywne, jeśli nie będziemy w stanie wyprowadzić niezależnych formuł dla składowej wymuszonej trajektorii wyjść predykcyjnych. Minimalizując bowiem funkcję kryterialną w (3.165) będziemy wówczas i tak musieli bezpośrednio wykorzystywać pełne równania nieliniowe do wyliczania wartości  $y(k+p|k)$ ,  $p = 1, 2, \dots, N$ , w zależności od wartości zmiennych decyzyjnych  $\Delta u(k+p|k)$ ,  $p = 0, \dots, N_u - 1$ .

Rozwiązaniem najprostszym, umożliwiającym naturalne przeniesienie korzystnych cech algorytmów predykcyjnych liniowych (tj. z liniowym modelem obiektu) do regulacji predykcyjnej nieliniowej są algorytmy wykonujące w każdej chwili próbkowania *linearyzację modelu nieliniowego* w aktualnym stanie obiektu, a następnie wyliczenie sterowania z zastosowaniem liniowego algorytmu MPC, zob. np. [56, 99, 4, 94, 152, 144, 73]. Jest to podejście *suboptymalne*, ale umożliwiające zachowanie fundamentalnej dla praktycznych zastosowań cechy *niezawodności i gwarantowanej szybkości* rozwiązywania zadania optymalizacji w każdej z kolejnych chwil próbkowania, tzn. niezawodności wyznaczania sterowania. Omówioną strukturę



Rysunek 3.59. Struktura regulacji MPC nieliniowej z sukcesywnymi linearyzacjami modelu (algoritm MPC-NSL)

suboptymalną nieliniowej regulacji MPC z linearyzacjami modelu określamy dalej skrótowo jako *strukturę MPC-NSL* (*MPC Nonlinear with Successive Linearization*), przedstawiono ją na rysunku 3.59.

Dla modeli słabo nieliniowych, czy też pozostających przez dłuższe okresy w pobliżu punktów równowagi, linearyzacja może nie być potrzebna w każdym kroku algorytmu, wystarczy wykonywanie jej rzadziej, np. co określoną liczbę chwil interwencji, przykład taki podano w [87]. Z drugiej strony, przedstawione podejście może się okazać niewystarczające dla obiektów silnie nieliniowych, czy też trajektorii odpowiadających szybkiemu przechodzeniu do odległych punktów równowagi, np. po wystąpieniu silnych zakłóceń czy też przy uruchamianiu lub wyłączaniu procesów pracujących w reżimie ciągłym. W wymienionych przypadkach może być celowe zastosowanie algorytmów MPC w większym stopniu wykorzystujących nieliniowość modelu. Są to przede wszystkim algorytmy z nieliniową predykcją i linearyzacją, które przedstawimy w następnym rozdziale.

### 3.5.2. Algorytmy MPC-NPL (z nieliniową predykcją i linearyzacją)

Rozwiązaniem strukturalnie dokładniejszym niż omówiony powyżej algorytm regulacji MPC-NSL, a przy tym nadal prostym w realizacji i zachowującym szybkość i niezawodność rozwiązywania zadania optymalizacji bieżącej, jest użycie modeli linearyzowanych jedynie dla zadania optymalizacji, natomiast wyznaczanie trajektorii początkowej wyjść predykowanych na modelu nieliniowym, w każdej chwili próbkowania, zob.

[50, 53, 56, 99, 158, 94, 152, 76]. Strukturę tę określać będziemy dalej jako *MPC-NPL* (*MPC with Nonlinear Prediction and Linearization*).

Nie ma w istocie powodu, aby rezygnować z nieliniowej predykcji trajektorii początkowej („składowej swobodnej”) dysponując modelem nieliniowym obiektu. Jest to bowiem zadanie stosunkowo proste, wykonywane tylko raz w każdym kroku algorytmu i nie wpływające na decydujący o efektywności nakład obliczeń związany z rozwiązaniem zadania optymalizacji kwadratowej. Stąd, należy w ogólności *preferować algorytmy MPC-NPL* w stosunku do algorytmów MPC-NSL. Oczywiście, o ile nasza wiedza o problemie czy wyniki badań symulacyjnych nie pokazują, że w zupełności wystarczy zastosowanie algorytmu typu MPC-NSL. W szczególności w sytuacji, kiedy wystarcza dokonywanie linearyzacji jedynie co kilka kroków algorytmu.

Do wyznaczenia trajektorii początkowej wyjść na horyzoncie predykcji  $\mathcal{Y}^0(k)$  potrzebne są nie tylko wartości zrealizowanych już (przeszłych) wyjść i sterowań, ale również przyjęcie określonych (początkowych) wartości *przyszłych sterowań* na horyzoncie predykcji, tzn. *trajektorii początkowej sterowań*  $\mathcal{U}^0(k)$ . Im trajektoria ta będzie bliższa wyznaczonej w danym kroku trajektorii optymalnej, tym mniejsze błędy będą popełnione przez zastosowanie w algorytmie optymalizacji modelu zlinearyzowanego. Stąd zalecane jest zastosowanie trajektorii początkowej sterowań w postaci (3.173).

W algorytmie MPC-NPL predykcja elementów trajektorii zmiennych wyjściowych regulowanych w chwili  $k$  są wyznaczana jest z zależności

$$\begin{aligned} y(k+p|k) &= y^0(k+p|k) + \Delta y(k+p|k) \\ &= y^0(k+p|k) + \bar{\mathbf{M}}_p(k)\Delta\mathcal{U}(k), \quad p = N_1, \dots, N, \end{aligned}$$

gdzie  $y^0(k+p|k)$  są elementami trajektorii początkowej  $\mathcal{Y}^0(k)$ , zaś trajektoria poprawek („trajektoria wymuszana”)  $\Delta\mathcal{Y}(k)$ ,

$$\Delta\mathcal{Y}(k) = \begin{bmatrix} \Delta y(k+N_1|k) \\ \Delta y(k+N_1+1|k) \\ \vdots \\ \Delta y(k+N|k) \end{bmatrix}, \quad (3.179)$$

liczona jest z zależności

$$\Delta\mathcal{Y}(k) = \mathbf{M}(k)\Delta\mathcal{U}(k) = \begin{bmatrix} \bar{\mathbf{M}}_{N_1}(k) \\ \bar{\mathbf{M}}_{N_1+1}(k) \\ \vdots \\ \bar{\mathbf{M}}_N(k) \end{bmatrix} \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_u-1|k) \end{bmatrix}. \quad (3.180)$$

Stąd zadanie optymalizacji algorytmu MPC-NPL to zadanie programowania kwadratowego, postaci

$$\min_{\Delta \mathcal{U}(k)} \left\{ \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}^0(k)] - \mathbf{M}(k) \Delta \mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \left\| \Delta \mathcal{U}(k) \right\|_{\underline{\Lambda}}^2 \right\}$$

$$\text{z ogr. : } -\Delta \mathcal{U}_{\max} \leq \Delta \mathcal{U}(k) \leq \Delta \mathcal{U}_{\max}, \quad (3.181)$$

$$\mathcal{U}_{\min} \leq \mathcal{U}^0(k) + \mathbf{J} \Delta \mathcal{U}(k) \leq \mathcal{U}_{\max},$$

$$\mathcal{Y}_{\min} \leq \mathcal{Y}^0(k) + \mathbf{M}(k) \Delta \mathcal{U}(k) \leq \mathcal{Y}_{\max},$$

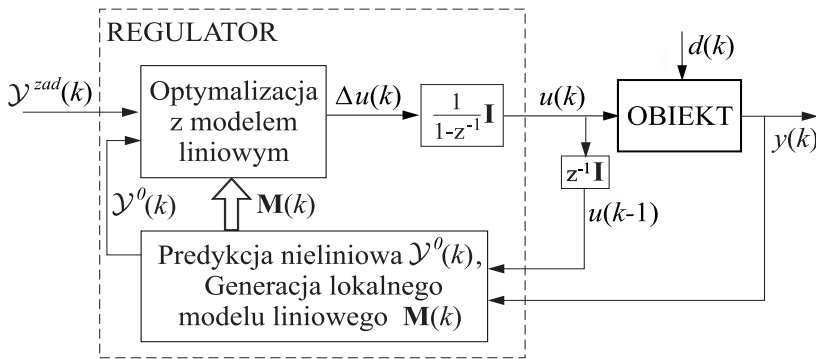
porównaj z (3.51) i (3.165).

W przypadku opisu modelu równaniem różnicowym, trajektorię początkową wyjść  $\mathcal{Y}^0(k)$  wyznaczamy identycznie jak trajektorię wyjść prognozowanych w algorytmie MPC-NO, tzn. wg formuł (3.169), jedynie przyjmując trajektorię początkową sterowań,  $u(k+p|k) = u^0(k+p|k)$ ,  $p = 0, \dots, N-1$ . W przypadku opisu modelu równaniami stanu, trajektorię początkową wyjść wyznaczamy podobnie, jedynie zamiast formuł (3.169) wykorzystując formuły (3.171).

Podkreślmy, że w zadaniu optymalizacji (3.181) *przyrosty sterowania*  $\Delta \mathcal{U}(k)$  są wyznaczane w stosunku do trajektorii początkowej  $\mathcal{Y}^0(k)$  wyznaczonej modelem nieliniowym, zaś  $\mathbf{M}(k)$  oznacza macierz odwzorowania liniowego wyznaczonego w chwili  $k$  jako aktualne, lokalne liniowe przybliżenie modelu nieliniowego (linearyzacja modelu nieliniowego). Oczywiście, elementy tej macierzy to współczynniki odpowiedzi skokowych odpowiadającego jej modelu liniowego, jest to macierz dynamiczna tego odwzorowania. Struktura *algorytmu regulacji MPC-NPL* została przedstawiona na rysunku 3.60.

Zaznaczyliśmy już, że algorytm MPC-NPL powinien działać tym lepiej, im mniejsze są przyrosty sterowania wyznaczone w zadaniu optymalizacji kwadratowej. A więc nawet dla obiektów silnie nieliniowych i przy przejściach między odległymi punktami pracy, byle trajektorie były łagodne, realizowane przy niezbyt dużych z kroku na krok zmianach sterowania. Jeśli jednakże wymagane są szybkie i duże zmiany punktów pracy, czy też szybkie reagowanie na zakłócenia skokowe o dużych amplitudach, to zastosowanie przyrostów sterowania wyznaczonych algorytmem MPC-NPL w jego najprostszej, przedstawionej powyżej wersji może nie być zadowalające. Są dwa podstawowe sposoby poprawy, wzmocnienia nieliniowego działania tego algorytmu, o rosnącym stopniu złożoności:

1. Pierwszy, nieco prostszy sposób to poprawa jakości linearyzacji – zamiast linearyzować model nieliniowy w aktualnym punkcie, można linearyzować go wokół trajektorii przewidywanej na horyzoncie predykcji.



Rysunek 3.60. Struktura regulacji MPC z nieliniową predykcją trajektorii swobodnej i linearyzacją modelu dla optymalizacji (algorytm MPC-NPL(T))

Algorytmy tej klasy oznaczać będziemy przez MPC-NPLT (Nieliniowa Predykcja z Linearyzacją wokół Trajektorii – *Nonlinear Prediction with Linearisation along Trajectory*).

2. Kolejny sposób to zastosowanie kilku powtórzeń (iteracyjne powtarzanie) cyklu „wyznaczanie trajektorii początkowej – linearyzacja – optymalizacja przyrostów sterowań na modelu liniowym”. Algorytmy tej klasy oznaczymy przez MPC-NPL+ lub MPC-NPLT+, odpowiednio.

W literaturze znaleźć można propozycje zarówno poprawy jakości linearyzacji, jak i wprowadzenia postępowania iteracyjnego do algorytmów klasy MPC-NPL, zob. np. [83, 79, 107, 56, 30, 143, 72, 73].

Kwestia poprawy jakości linearyzacji to kwestia poprawy wyznaczania liniowego przybliżenia modelu, tzn. wyznaczania macierzy współczynników  $\mathbf{M}(k)$  określających liniową zależność między przyrostami wyjść a przyrostami sterowań w zadaniu optymalizacji kwadratowej. W podanym wyżej opisie algorytmu MPC-NPL w jego najprostszej wersji linearyzację zaproponowano w punkcie, w którym obiekt znajduje się w chwili  $k$ , a więc w chwili początkowej horyzontu predykcji. Natomiast trajektoria początkowa wyjść regulowanych na horyzoncie predykcji, którą będziemy dalej umownie nazywać również „trajektoria swobodną”, wyliczana jest w sposób nieliniowy. W konsekwencji, podczas jej obliczania przewidywany stan obiektu zmienia się z chwili na chwilę predykcji w ogólności w sposób nieliniowy, tzn. przybliżenie liniowe obiektu w każdej z kolejnych chwil na horyzoncie predykcji może być nieco inne – i można je korygować powtarzając linearyzację w punktach odpowiadających każdej z kolejnych chwil na horyzoncie predykcji, tzn. dokonując *linearyzacji wokół trajektorii*. W kon-

sekwencji, macierz  $\mathbf{M}(k)$ , zob. (3.180), będzie się składać z podmacierzy (w przypadku skalarnej wyjścia  $y$  – wierszy)  $\overline{\mathbf{M}}_p(k)$  wyliczanych kolejno dla nieco innych modeli liniowych odpowiadających poszczególnym chwilom  $p = 0, 1, \dots, N - 1$  horyzontu predykcyjnego.

Powtórzenie w danym kroku algorytmu MPC-NPL, jednokrotne czy nawet kilkakrotne, cyklu składającego się z wyznaczenia trajektorii początkowej („swobodnej”) na modelu nieliniowym i następnie obliczenia optymalnych przyrostów sterowań poprzez rozwiązanie zadania programowania kwadratowego jest koncepcyjnie i realizacyjnie klarownym, dalszym sposobem poprawy nieliniowych cech algorytmu MPC-NPL w momentach wymagających silniejszego uwzględnienia nieliniowości, przy występowaniu większych zmian sygnału sterowania. Procedura ta może być realizowalna przy obecnym poziomie rozwoju mocy obliczeniowej procesorów nawet dla niezbyt wolnych obiektów, kilkakrotne powtarzanie rozwiązania zadania programowania kwadratowego w jednym kroku algorytmu regulacji jest tylko problemem wyboru mocy procesora zastosowanego do sterowania. Nawet w czasach znacznie wolniejszych i droższych procesorów próbowano stosować ten sposób postępowania, zmniejszając nakład obliczeń przez radykalne skrócenie horyzontu sterowania, nawet do  $N_u = 1$  [50]. Może to być w praktyce skuteczne, gdyż najważniejsza jest poprawa cech nieliniowych algorytmu MPC-NPL w pierwszej chwili horyzontu sterowania, przy wyznaczeniu pierwszego przyrostu sterowania  $\Delta \hat{u}(k|k)$ . Powody są dwa: po pierwsze, po silnej (np. skokowej) zmianie wartości zadanej czy zakłócenia algorytm reaguje najsilniej, tj. zmianą sterowania o największej amplitudzie, z reguły właśnie w chwili pierwszej. Po drugie, tylko pierwsze ze sterowań wyznaczonych przez algorytm predykcyjny na horyzoncie sterowania jest przekazywane do realizacji, następne będą od nowa wyznaczone zgodnie z ogólną zasadą sterowania predykcyjnego z przesuwającym horyzontem. Opis takiego algorytmu (o strukturze MPC-NPL+) znaleźć można np. w [143, 144]. Wobec wspomnianego gwałtownego rozwoju techniki mikroprocesorowej ograniczanie długości horyzontu sterowania do  $N_u = 1$  nie wydaje się aktualnie zasadne, stąd poniżej przedstawimy opis przykładowego algorytmu MPC-NPLT+.

Organizacja algorytmu MPC-NPLT+ jest w każdym kroku następująca:

1. Korzystając z modelu nieliniowego, na podstawie poprzednich wartości wyjść i sterowań obiektu oraz początkowej trajektorii sterowań  $\mathcal{U}^0(k)$  (3.173) wyznacz trajektorie początkową wyjść przewidywanych  $\mathcal{Y}^0(k)$ . Wyznacz macierz dynamiczną modelu zlinearyzowanego  $\mathbf{M}^0(k)$  poprzez linearyzację modelu nieliniowego wokół trajektorii początkowej sterowań. Ustaw indeks iteracji wewnętrznych  $j = 0$ .

2. Rozwiąż zadanie optymalizacji kwadratowej

$$\min_{\Delta \mathcal{U}(k)} \left\{ \left\| [\mathcal{Y}^{zad}(k) - \mathcal{Y}^j(k)] - \mathbf{M}^j(k) \Delta \mathcal{U}(k) \right\|_{\underline{\Psi}}^2 + \left\| \Delta \mathcal{U}(k) \right\|_{\underline{\Lambda}}^2 \right\}$$

$$\text{z ogr. : } -\Delta \mathcal{U}_{\max} \leq \Delta \mathcal{U}(k) \leq \Delta \mathcal{U}_{\max}, \quad (3.182)$$

$$\mathcal{U}_{\min} \leq \mathcal{U}^j(k) + \mathbf{J} \Delta \mathcal{U}(k) \leq \mathcal{U}_{\max},$$

$$\mathcal{Y}_{\min} \leq \mathcal{Y}^j(k) + \mathbf{M}^j(k) \Delta \mathcal{U}(k) \leq \mathcal{Y}_{\max},$$

uzyskując wektor optymalnych przyrostów sterowań  $\Delta \hat{\mathcal{U}}^j(k)$ , tzn. optymalne przyrosty  $\Delta \hat{u}^j(k+p|k)$ ,  $p = 0, \dots, N_u - 1$ .

3. Jeśli  $j > 0$  oraz

$$\|\hat{u}^j(k|k) - \hat{u}^{j-1}(k|k)\| < \varepsilon \|\hat{u}^j(k|k)\|, \quad (3.183)$$

to *zakończ* przekazując przyrost sterowania

$$\Delta \hat{u}(k) = \sum_{i=0}^j \Delta \hat{u}^i(k|k)$$

do realizacji w układzie regulacji.

W przeciwnym razie przejdź do punktu 4.

4. Dokonaj korekty trajektorii sterowań na horyzoncie predykcji:

$$\mathcal{U}^{j+1}(k) = \begin{bmatrix} u^j(k|k) + \Delta \hat{u}^j(k|k) \\ u^j(k+1|k) + \sum_{i=0}^1 \Delta \hat{u}^j(k+i|k) \\ \vdots \\ u^j(k+N_u-1|k) + \sum_{i=0}^{N_u-1} \Delta \hat{u}^j(k+i|k) \\ u^j(k+N_u|k) \\ \vdots \\ u^j(k+N-1|k) \end{bmatrix}.$$

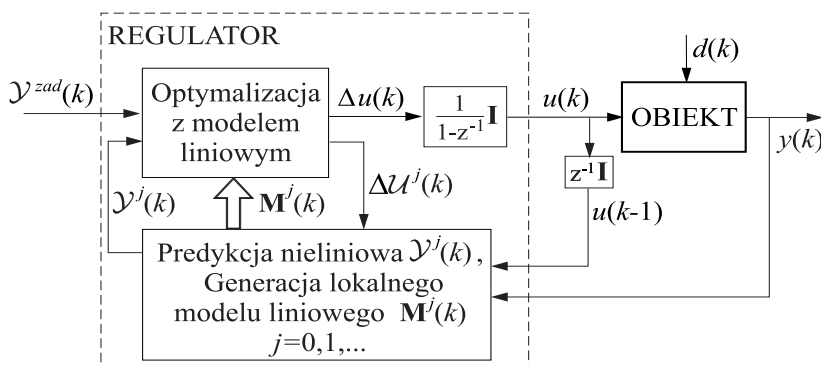
5. Dokonaj *korekty trajektorii początkowej*: na nieliniowym modelu wyznacz trajektorię wyjść przewidywanych  $\mathcal{Y}^{j+1}(k)$  składającą się z elementów  $y^{j+1}(k+p|k)$ ,  $p = 1, \dots, N$ , liczonych dla zrealizowanych wyjść i sterowań z chwil poprzednich oraz sterowań przewidywanych na horyzont optymalizacji danych przez  $\mathcal{U}^{j+1}(k)$  (stosując formuły (3.169) lub (3.171), dla  $u(k+p|k) = u^{j+1}(k+p|k)$ ,  $p = 0, \dots, N-1$ ). Wyznacz linearyzację modelu nieliniowego wokół wyznaczonej trajektorii

sterowań  $\mathcal{U}^{j+1}(k)$ , oblicz macierz dynamiczną odpowiadającą modelowi zlinearyzowanemu  $\mathbf{M}^{j+1}(k)$ .

Ustaw  $j = j + 1$  i wróć do punktu 2.

W algorytmie powyższym przyjęto, w punkcie 3, prosty test stopu oparty o obserwację zmian wartości pierwszego sterowania  $\Delta \hat{u}^j(k|k)$  w kolejnych iteracjach. Test ten można rozszerzyć, kontrolując zmiany całej trajektorii sterowania, co może być rozwiązaniem lepszym w sytuacji aktywności ograniczeń sterowania na początku horyzontu predykcyjnego. Wartość współczynnika  $\varepsilon$  występującego w kryterium zakończenia iteracji wewnętrznych w punkcie 3 algorytmu nie powinna być ustawiona na zbyt małej wartości, jedynie przy istotniejszej różnicy powinno dochodzić do powtórzeń predykcyjnej i optymalizacyjnej. Teoretycznie, iterowanie może wymagać wielu kroków, ale ze względu na konieczność wyliczenia przez algorytm przyrostu sterowania w określonym czasie maksymalna liczba iteracji jest z reguły dodatkowo arbitralnie ograniczana do niewielu powtórzeń. Tym bardziej, że zbieżność tego typu iteracji (choć na ogół szybko zbieżnych) nie jest w ogólności gwarantowana, stąd dopuszczając wielokrotne iterowanie trzeba pamiętać o odpowiednich zabezpieczeniach, w razie potrzeby przerywających iteracje.

Strukturę algorytmu MPC-NPLT+ przedstawia rys. 3.61. Zauważmy, że jego pętlę iteracyjną można też traktować jako *rozszerzenie wbudowane w algorytm MPC-NPLT*, włączane dodatkowo jedynie w momentach silnie nieliniowych kroków spowodowanych większymi wymuszeniami. Struktura MPC-NPLT+ jest najbardziej rozbudowaną spośród przedstawionych dotąd wykorzystujących model liniowy w zadaniu optymalizacji, jest adresowana do problemów silnie nieliniowych i sytuacji bardziej wymagających,



Rysunek 3.61. Wzbogacenie struktury MPC-NPLT przez dodanie możliwości iteracyjnej poprawy predykcyjnej i optymalizacji (algorytm MPC-NPLT+)

kiedy potrzebne są odpowiednie reakcje sterowań na większe i szybsze zmiany wymuszeń czy zakłóceń.

Przedstawione struktury algorytmów regulacji nieliniowej MPC: optymalna (MPC-NO) i suboptymalne (MPC-NSL, MPC-NPL) są ogólne, nie ograniczają możliwych postaci modeli nieliniowych. Odpowiadają zadaniu regulacji w obecności ograniczeń, w każdym kroku algorytmu MPC rozwiązywane jest zadanie optymalizacji nieliniowej z ograniczeniami (MPC-NO), bądź wypukłe kwadratowe z ograniczeniami (MPC-NSL, MPC-NPL, MPC-NPLT).

### Przykład 3.10.

Rozważmy przypadek modelu dynamiki obiektu opisanego nieliniową funkcją  $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ ,

$$y(k+1) = g(y(k), u(k-1), u(k-2)) \quad (3.184)$$

Zilustrujemy na tym przykładzie, modelu opisanego równaniem różniczkowym, konstruowanie trajektorii początkowej wyjść na horyzoncie predykcji oraz linearyzację modelu i konstrukcję macierzy  $\mathbf{M}(k)$  definiującą odpowiedź wymuszaną, w algorytmach MPC-NPL i MPC-NPLT. Założymy długości horyzontu sterowania  $N_u = 4$  i predykcji  $N = 6$ , oraz  $N_1 = 2$  (w obiekcie występuje opóźnienie  $\tau = 1$ ).

Obliczenie rekurencyjne nieliniowej (na podstawie modelu nieliniowego) *trajektorii początkowej wyjść* (tzn. składowej swobodnej) w chwili  $k$ , dla danej początkowej trajektorii sterowań  $\mathcal{U}^0(k)$  (tzn.  $u^0(k+p|k)$ ,  $p = 0, \dots, N-1$ ), wygląda następująco:

$$\begin{aligned} y^0(k+1|k) &= g(y(k), u(k-1), u(k-2)) + d(k), \\ y^0(k+2|k) &= g(y^0(k+1|k), u^0(k|k), u(k-1)) + d(k), \\ y^0(k+3|k) &= g(y^0(k+2|k), u^0(k+1|k), u^0(k|k)) + d(k), \\ y^0(k+4|k) &= g(y^0(k+3|k), u^0(k+2|k), u^0(k+1|k)) + d(k), \\ y^0(k+5|k) &= g(y^0(k+4|k), u^0(k+3|k), u^0(k+2|k)) + d(k), \\ y^0(k+6|k) &= g(y^0(k+5|k), u^0(k+4|k), u^0(k+3|k)) + d(k), \end{aligned}$$

gdzie  $d(k)$  jest estymatą zakłócenia,

$$d(k) = y(k) - y(k|k-1) = y(k) - g(y(k-1), u(k-2), u(k-3)).$$

Zdefiniujmy wektor aktualnego punktu (stanu) obiektu, będącego punktem linearyzacji  $x_L(k) \in \mathbb{R}^3$ :

$$x_L(k) = \begin{bmatrix} x_{L1}(k) \\ x_{L2}(k) \\ x_{L3}(k) \end{bmatrix} = \begin{bmatrix} y(k) \\ u(k-1) \\ u(k-2) \end{bmatrix}.$$

Oznaczając:

$$a_1(k) = a_1(x_L(k)) = -\frac{\partial g}{\partial y(k)}(y(k), u(k-1), u(k-2)),$$

$$b_1(k) = b_1(x_L(k)) = \frac{\partial g}{\partial u(k-1)}(y(k), u(k-1), u(k-2)),$$

$$b_2(k) = b_2(x_L(k)) = \frac{\partial g}{\partial u(k-2)}(y(k), u(k-1), u(k-2)),$$

dostajemy równanie modelu zlinearyzowanego (dla zmiennych przyrostowych !) w postaci:

$$\Delta y(k+1) = -a_1(k)\Delta y(k) + b_1(k)\Delta u(k-1) + b_2(k)\Delta u(k-2).$$

Równanie to jest potrzebne jedynie dla wyznaczenia macierzy dynamicznej  $\mathbf{M}(k)$ , znajomość  $\mathbf{M}(k)$  wystarcza do liczenia składowej wymuszanej prognozowanej trajektorii wyjść.

Dla wyliczenia macierzy dynamicznej, liczymy współczynniki odpowiedzi skokowej modelu zlinearyzowanego tak samo jak dla algorytmu GPC, z wzoru (3.85):

$$s_j(k) = - \sum_{i=1}^{\min\{j-1, n_A\}} a_i(k) s_{j-i}(k) + \sum_{i=0}^{\min\{j-1, n_B\}} b_i(k).$$

W przypadku rozważanego modelu dostajemy stąd:

$$\begin{aligned} s_1(k) &= 0, \\ s_2(k) &= b_1(k), \\ s_3(k) &= -a_1(k)s_2 + b_1(k) + b_2(k), \\ s_4(k) &= -a_1(k)s_3 + b_1(k) + b_2(k), \\ s_5(k) &= -a_1(k)s_4 + b_1(k) + b_2(k), \\ s_6(k) &= -a_1(k)s_5 + b_1(k) + b_2(k). \end{aligned}$$

Macierz dynamiczną  $\mathbf{M}(k)$  (wymiaru  $5 \times 4$ ) wyznaczamy ze wzoru (3.31),

$$\mathbf{M}(k) = \begin{bmatrix} s_2(k) & 0 & 0 & 0 \\ s_3(k) & s_2(k) & 0 & 0 \\ s_4(k) & s_3(k) & s_2(k) & 0 \\ s_5(k) & s_4(k) & s_3(k) & s_2(k) \\ s_6(k) & s_5(k) & s_4(k) & s_3(k) \end{bmatrix}.$$

Składowa wymuszana  $\Delta\mathcal{Y}(k)$  trajektorii przewidywanej na horyzoncie predykcji dana będzie oczywiście wzorem  $\Delta\mathcal{Y}(k) = \mathbf{M}(k)\Delta\mathcal{U}(k)$ , gdzie

$$\Delta\mathcal{U}(k) = [\Delta u(k|k) \ \Delta u(k+1|k) \ \Delta u(k+2|k) \ \Delta u(k+3|k)]^T.$$

Przedstawiona powyżej linearyzacja modelu nieliniowego przeprowadzona była w punkcie  $x_L(k)$  odpowiadającym wyjściu procesu  $y(k)$  w aktualnej chwili  $k$ . Odpowiada to potrzebom algorytmu MPC-NPL.

Algorytm MPC-NPLT wymaga linearyzacji wokół trajektorii początkowej na horyzoncie predykcji, w naszym przypadku długości horyzontu predykcji  $N = 6$ , w chwilach  $k+p|k$ ,  $p = 0, 1, \dots, 5$ , wokół trajektorii sterowań  $\{u^0(k|k), u^0(k+1|k), \dots, u^0(k+5|k)\}$  i wyjść prognozowanych  $\{y(k), y^0(k+1|k), \dots, y^0(k+5|k)\}$ . Linearyzację w chwili początkowej horyzontu, tzn. w punkcie  $x_L(k|k) = x_L(k)$  już przeprowadziliśmy. W analogiczny sposób linearyzujemy w kolejnych chwilach, w punktach

$$\begin{aligned} x_L(k+1|k) &= \begin{bmatrix} y^0(k+1|k) \\ u^0(k|k) \\ u(k-1) \end{bmatrix}, \\ x_L(k+2|k) &= \begin{bmatrix} y^0(k+2|k) \\ u^0(k+1|k) \\ u^0(k|k) \end{bmatrix}, \\ &\vdots \\ x_L(k+5|k) &= \begin{bmatrix} y^0(k+5|k) \\ u^0(k+4|k) \\ u^0(k+3|k) \end{bmatrix}. \end{aligned}$$

W każdym z tych punktów uzyskujemy nieco inny model zlinearyzowany (przyrostowy), tzn. nieco inne wartości współczynników  $a_1, b_1$  i  $b_2$ , a stąd i współczynników odpowiedzi skokowej. Oznaczmy te współczynniki przez  $a_1(k+p|k), b_1(k+p|k), b_2(k+p|k)$  oraz  $s_j(k+p|k)$ ,  $p = 0, \dots, 5$ ,  $j = 1, 2, \dots, 6$ . Macierz dynamiczna ma w tym przypadku postać

$$\mathbf{M}(k) = \begin{bmatrix} s_2(k+1|k) & 0 & 0 & 0 \\ s_3(k+2|k) & s_2(k+2|k) & 0 & 0 \\ s_4(k+3|k) & s_3(k+3|k) & s_2(k+3|k) & 0 \\ s_5(k+4|k) & s_4(k+4|k) & s_3(k+4|k) & s_2(k+4|k) \\ s_6(k+5|k) & s_5(k+5|k) & s_4(k+5|k) & s_3(k+5|k) \end{bmatrix}.$$

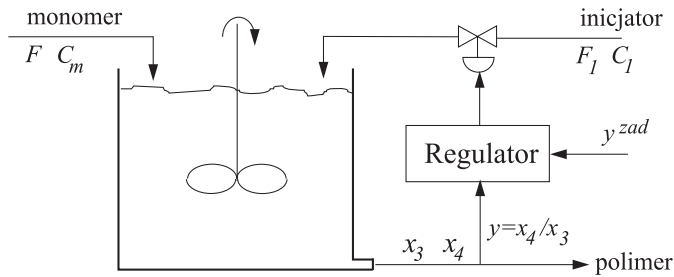
□

**Przykład 3.11.**

Rozważmy regulację predykcyjną ciągłego izotermicznego reaktora opisanego w pracy [34], w którym zachodzi proces polimeryzacji. Przyjmując przyjęte tam założenia (idealne wymieszanie, stałe zapełnienie, stała temperatura równa  $335K$ , stała pojemność cieplna itp.) oraz wartości parametrów fizykochemicznych uzyskuje się następujące równania reaktora:

$$\begin{aligned}\dot{x}_1 &= 10(6 - x_1) - 2.4568x_1\sqrt{x_2}, \\ \dot{x}_2 &= 80u - 10.1022x_2, \\ \dot{x}_3 &= 0.0024121x_1\sqrt{x_2} + 0.112191x_2 - 10x_3, \\ \dot{x}_4 &= 245.978x_1\sqrt{x_2} - 10x_4, \\ y &= x_4/x_3,\end{aligned}$$

gdzie  $x_1 = C_m$  jest stężeniem monomeru,  $x_2 = C_1$  jest stężeniem substancji inicjującej (inicjatora), zaś wielkością wyjściową regulowaną  $y$  jest tzw. ciężar molekularny (NAMV – *number-average molecular weight*,  $[kg/kmol]$ ), będący ilorazem zmiennych stanu czwartej i trzeciej,  $y = x_4/x_3$ . Wielkością sterującą  $u$  jest natężenie dopływu inicjatora  $F_1$ . Ogólny schemat układu regulacji reaktora pokazano na rysunku 3.62.



Rysunek 3.62. Schemat reaktora polimeryzacji, do przykładu 3.11

Przedstawiony obiekt jest silnie nieliniowy, o nieliniowej zarówno statyce, jak i dynamice. Przyjęto punkt pracy tak jak w [34]:  $x_{10} = 5.50677$ ,  $x_{20} = 0.132906$ ,  $x_{30} = 0.0019752$ ,  $x_{40} = 49.3818$ ,  $u_0 = 0.016783$ , stąd  $y_0 = 25000$ . Należy też uwzględnić ograniczenia amplitudy sterowania:  $u_{\min} = 0.001$ ,  $u_{\max} = 0.5$ .

Eksperymentami symulacyjnymi objęto układy regulacji z trzema regulatorami predykcyjnymi:

- liniowym regulatorem GPC (w wersji numerycznej, z bieżącą optymalizacją) zaprojektowanym dla podanego punktu pracy,

- regulatorem MPC-NPL, z nieliniową predykcją wyliczaną z wykorzystaniem zdyskretyzowanej wersji modelu nieliniowego i optymalizacją kwadratową z modelem linearyzowanym w każdym kroku algorytmu,
- pełnym nieliniowym regulatorem MPC-NO, z optymalizacją na zdyskretyzowanym modelu nieliniowym w każdym kroku.

Dla wymienionych regulatorów przyjęto parametry:

$N = 10$ ,  $N_u = 2$ ,  $\Psi = 1$ ,  $\Lambda = \lambda = 700$ , okres próbkowania  $T_p = 1.8$  min.

Wybrane rezultaty eksperymentów symulacyjnych [70], wykonane z wykorzystaniem programów pakietu REGZA [74], zamieszczone są na rysunkach od 3.63 do 3.66.

Rysunki 3.63 i 3.64 pokazują przebiegi po skoku wartości zadanej do wartości 28000. Widoczne są już różnice w działaniu regulatorów, większe szczególnie w przebiegach sygnału sterującego (rysunków dla mniejszych skoków wartości zadanej nie zamieszczamy, praktycznie nie różnią się od siebie). Zmiana punktu pracy jest w kierunku większych wzmocnień obiektu, stąd regulator GPC zaprojektowany dla punktu pracy o mniejszym wzmocnieniu wykazuje najszybsze działanie, ale działa jeszcze stabilnie.

Rysunki 3.65 i 3.66 pokazują przebiegi po większym skoku wartości zadanej, do wartości 32000. Różnice w działaniu regulatorów są teraz bardzo wyraźne, zarówno w przebiegach wielkości regulowanej, jak i sygnału sterującego. Układ z regulatorem GPC uległ destabilizacji, natomiast układ z najprostszą wersją regulatora MPC-NPL działa gorzej (wolniej) od regulatora MPC-NO, ale skutecznie.

### Przykład 3.12.

Przedstawimy wybrane wyniki badań symulacyjnych dwuwymiarowej regulacji predykcyjnej stężeń  $x_d$  i  $x_b$  produktu, odpowiednio w destylacie  $D$  i cieczy wyczerpanej  $B$  w kolumnie destylacyjnej przedstawionej w przykładzie 2.3 w rozdziale 2.1. Przyjmując okres próbkowania  $T_p = 10$  min zaprojektowano dwa regulatory predykcyjne [70]:

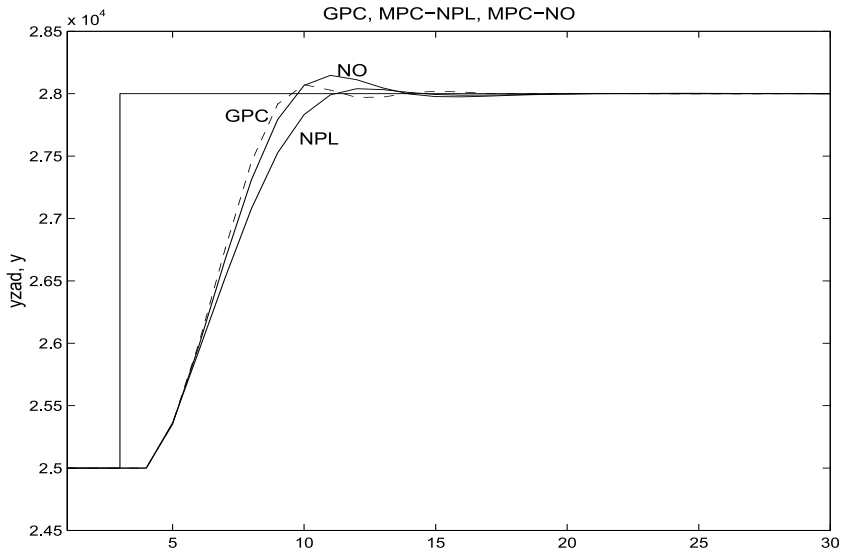
- Regulator liniowy GPC o strukturze modeli (3.93)

$$\begin{aligned} y_1(k) &= -a_1^1 y_1(k-1) + b_0^{1,1} u_1(k-1) + b_0^{1,2} u_2(k-1), \\ y_2(k) &= -a_1^2 y_2(k-1) + b_0^{2,1} u_1(k-1) + b_0^{2,2} u_2(k-1), \end{aligned}$$

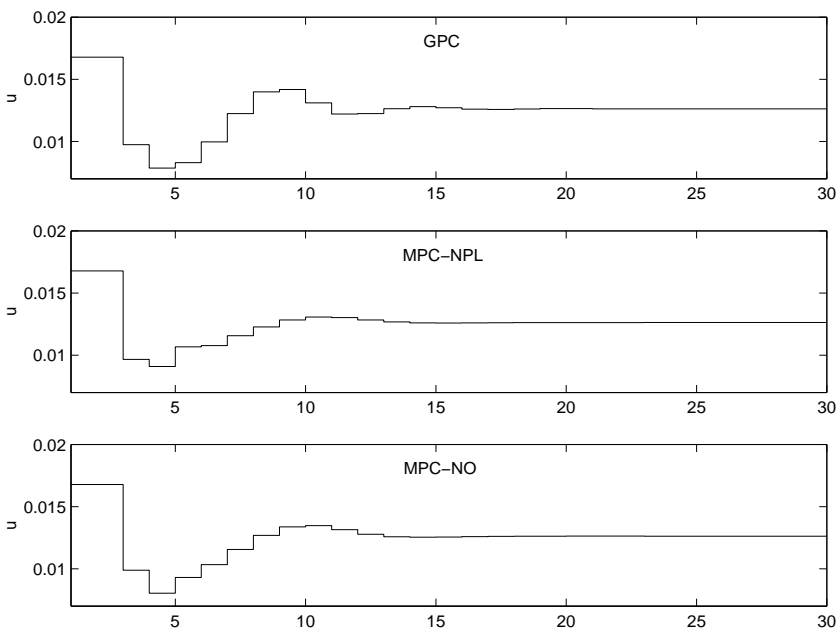
wyznaczony dla punktu pracy określonego sterowaniami  $R_0 = 33.34$ ,  $V_0 = 83, 34$  [kmol/h], gdzie  $y = [x_d \ x_b]^T$ ,  $u = [R \ V]^T$ .

- Regulator MPC-NO, z nieliniową optymalizacją modelu kolumny.

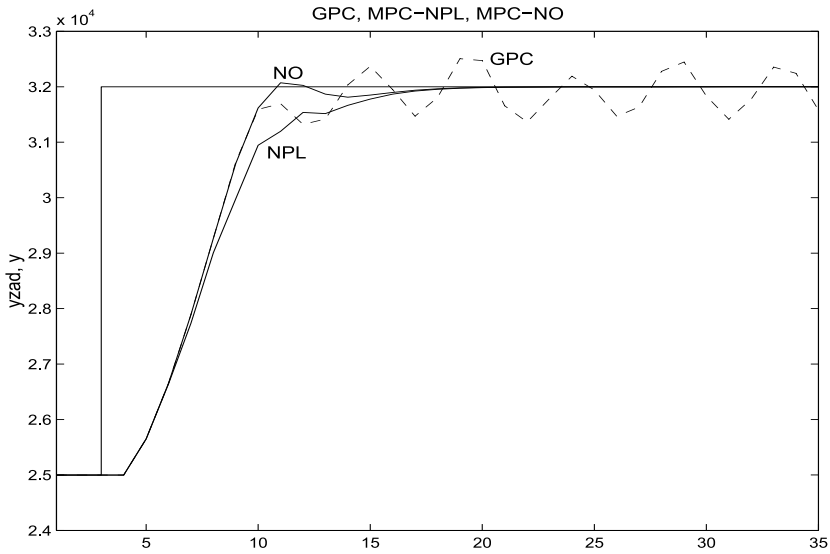
Przyjęto  $\Psi(p) = \mathbf{I}$ ,  $\Lambda(p) = \lambda \mathbf{I}$ . Badano regulatory dla różnych horyzontów predykcji i różnych wartości  $\lambda$ . Na rysunkach od 3.67 do 3.70 przedsta-



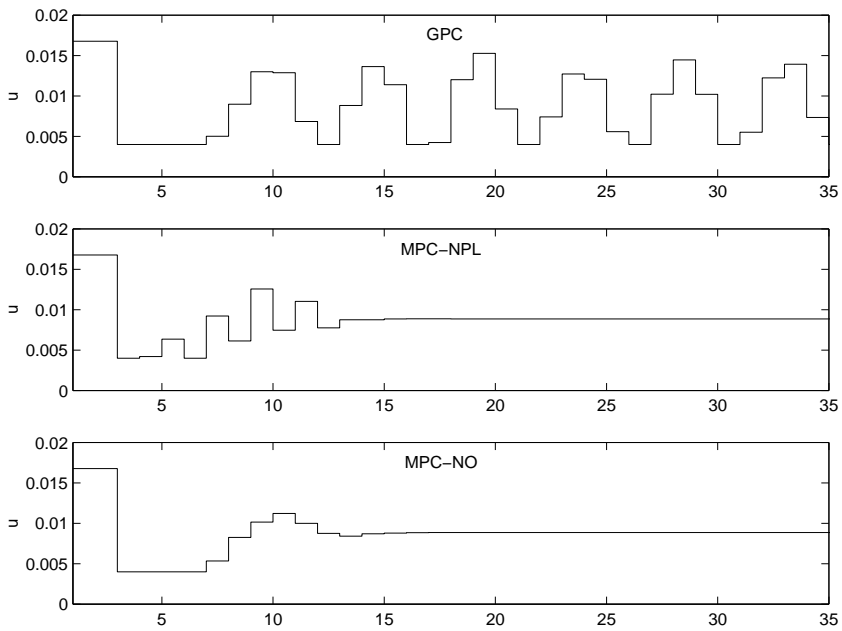
Rysunek 3.63. Przebiegi trajektorii wyjścia w eksperymentach z reaktorem polimeryzacji, skok wartości zadanej do 28000



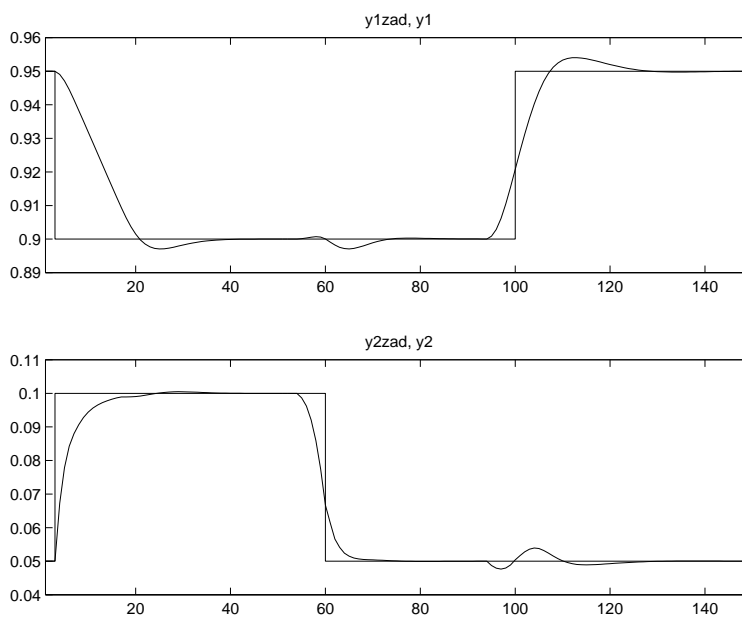
Rysunek 3.64. Przebiegi trajektorii sterowania w eksperymentach z reaktorem polimeryzacji, skok wartości zadanej do 28000



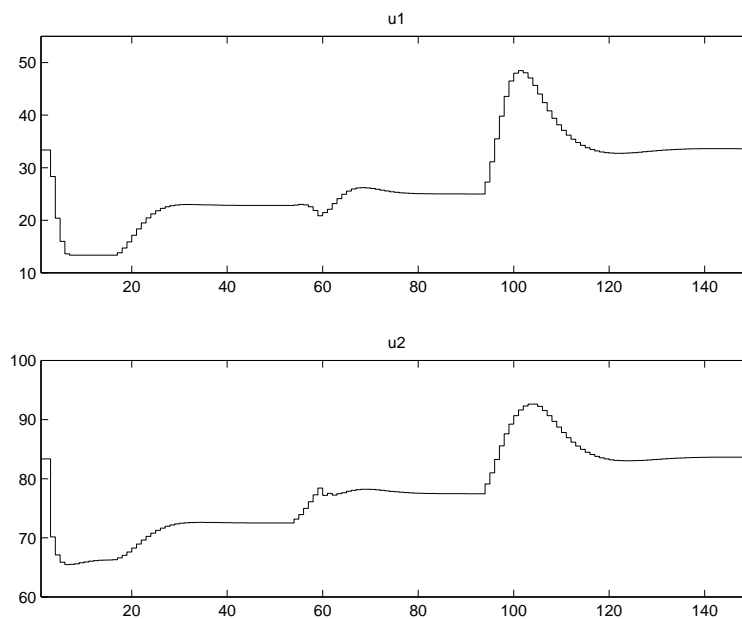
Rysunek 3.65. Przebiegi trajektorii wyjścia w eksperymentach z reaktorem polimeryzacji, skok wartości zadanej do 32000



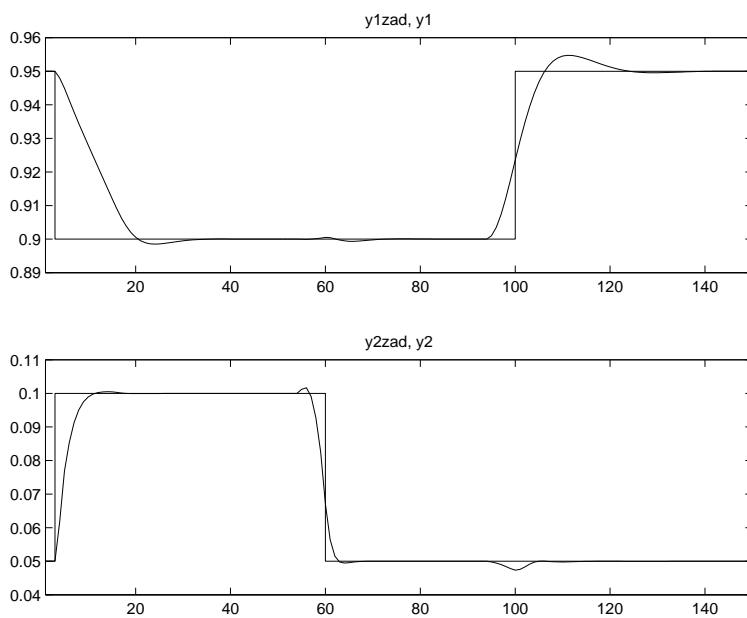
Rysunek 3.66. Przebiegi trajektorii sterowania w eksperymentach z reaktorem polimeryzacji, skok wartości zadanej do 32000



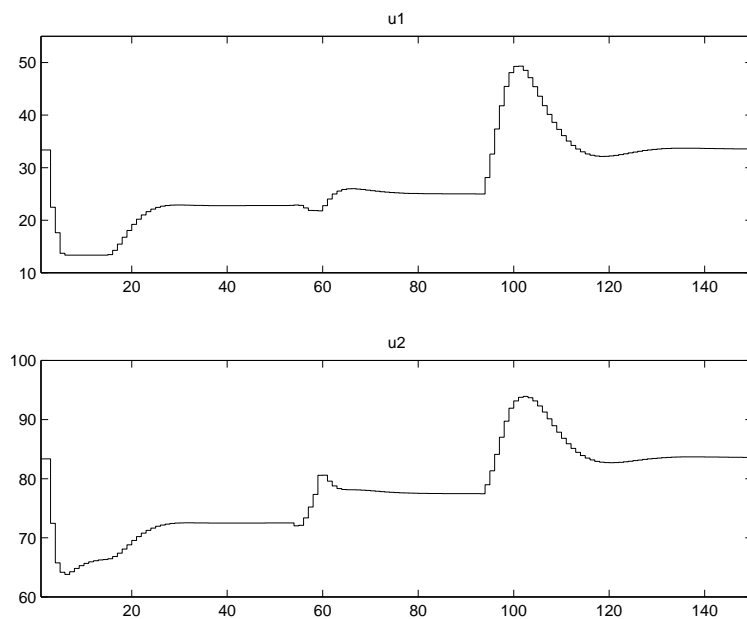
Rysunek 3.67. Przebiegi wyjść w układzie regulacji GPC kolumny



Rysunek 3.68. Przebiegi sterowań w układzie regulacji GPC kolumny



Rysunek 3.69. Przebiegi wyjść w układzie regulacji MPC-NO kolumny



Rysunek 3.70. Przebiegi sterowań w układzie regulacji MPC-NO kolumny

wiono dość reprezentatywne przebiegi uzyskane dla obu algorytmów przy  $N = 6$ ,  $N_u = 2$  i  $\lambda = 0.05$ .

Z porównania wynika, że mimo silnej nieliniowości kolumny (o czym świadczą charakterystyki statyczne przedstawione w przykładzie 2.3) regulator GPC zupełnie nieźle funkcjonuje. Regulator MPC-NO (z pełną nieliniową optymalizacją) jest nieco szybszy oraz lepiej tłumi interakcje między wyjściami. W prezentowanych symulacjach założono, odmiennie niż w innych przykładach, że regulator predykcyjny dysponuje z wyprzedzeniem wiedzą o mających nastąpić skokach wartości zadanych. Widać wyraźnie antycypacyjny charakter sterowania, reagującego z wyprzedzeniem na mające nastąpić zmiany.  $\square$

Ważnymi z punktu widzenia zastosowań w algorytmach regulacji predykcyjnej, w tym szczególnie w algorytmach z bieżącymi linearyzacjami modelu nieliniowego, są dwa przypadki modelowania obiektu nieliniowego:

- modelowanie siecią neuronową,
- modelowanie systemem rozmytym typu TS (Takagi-Sugeno) z liniowymi następnikami reguł, zob. rozdz. 2.1.

Modelowania siecią neuronową nie będziemy tu omawiać, gdyż istnieją bardzo aktualne i wyczerpujące monografie poświęcone temu tematowi [72, 73]. Będziemy natomiast omawiać wykorzystanie modeli rozmytych.

Dla nieliniowego modelu w postaci systemu rozmytego typu TS w naturalny sposób generować można lokalne modele liniowe potrzebne w poszczególnych krokach suboptymalnych algorytmów nieliniowych MPC. Ponadto, w przypadku projektowania regulatora bez uwzględniania ograniczeń można zastosować ogólną strukturę regulacji nieliniowej rozmytej typu TS, ze sprzężeniem od wyjścia lub stanu, zob. rozdz. 2.2. Definiujemy wówczas następniki reguł nieliniowego regulatora rozmytego jako liniowe, analityczne prawa regulacji MPC. Warto przypomnieć w tym miejscu, że analityczne wersje regulatorów MPC (prawa regulacji MPC) odpowiednio implementowane w strukturach typu *anti-wind-up* mają istotne znaczenie praktyczne, szczególnie w regulacji jednowymiarowej. Ponadto, przy projektowaniu regulatorów MPC bywa zalecane najpierw zaprojektowanie regulatora bez uwzględniania ograniczeń, zob. np. [52, 104]. Łatwiej wówczas dobrać parametry wpływające na dynamikę i odporność układu regulacji.

Dla zwięzłości zapisu określać dalej będziemy algorytmy MPC z modelem obiektu nieliniowym rozmytym typu TS jako *algorytmy FMPC (Fuzzy Model Predictive Control)*. Realizacje przedstawionych dotychczas algorytmów nieliniowych dla modeli rozmytych typu TS omówimy nieco szczegółowiej w kolejnych rozdziałach.

### 3.5.3. Algorytmy FMPC (Fuzzy MPC) w wersji numerycznej

Model nieliniowy rozmyty typu TS jest przykładem ogólnego modelu nieliniowego. Jeśli wykorzystujemy ten model w strukturze *algorytmu FMPC-NO* (Fuzzy MPC-NO), to w każdym jego kroku wykonywana jest optymalizacja nieliniowej, niekwadratowej funkcji celu z ograniczeniami liniowymi sterowań i nieliniowymi ograniczeniami wyjść (jeśli występują), czyli rozwiązywane jest zadanie (3.165) (czy (3.167)) z trajektorią przewidywaną wyjść wyznaczaną z wykorzystaniem nieliniowego modelu rozmytego TS. Kluczowym problemem jest tu odpowiedni dobór numerycznej procedury optymalizacji, tak aby zapewnić jej niezawodne i odpowiednio szybkie działanie. Sformułowanie zaleceń projektowych nie jest łatwe, mogą to być jedynie ogólne wskazówki dotyczące doboru procedur minimalizacji do cech problemów nieliniowych znane z teorii i praktyki optymalizacji, zob. np. [46, 7, 132]. Procedury numeryczne typu SQP (*Sequential Quadratic Programming*) są najczęściej rekomendowane. W prostszych przypadkach regulacji jednowymiarowej problem wyboru procedury optymalizacji staje się mniej krytyczny [45]. Dyskusję dotyczącą zastosowania metod optymalizacji w predykcyjnych algorytmach regulacji nieliniowej znajdzie czytelnik w pracach [1, 87]. Natomiast zagadnienie stabilności omówimy ogólnie dalej, w rozdziale 3.6.1.

Sytuacja obliczeniowo, implementacyjnie znacznie korzystniejsza występuje w przypadku nieliniowych algorytmów FMPC *suboptymalnych*, w każdej chwili  $k$  wykorzystujących dla potrzeb optymalizacji bieżącej zlinearyzowaną postać modelu rozmytego – a więc algorytmów z modelem sukcesywnie linearyzowanym, czy z nieliniową predykcją i optymalizacją na modelu linearyzowanym (FMPC-NSL, FMPC-NPL, FMPC-NPLT). Stąd też duże znaczenie praktyczne tych rozwiązań. Tym bardziej, że różnica w działaniu układu regulacji wynikająca z suboptymalności rozwiązania nie ma często istotniejszego znaczenia, szczególnie w przypadku stosowania nieliniowej predykcji trajektorii swobodnej (jak w strukturach NPL, NPLT).

Jak pokażemy poniżej, przy stosowaniu algorytmów FMPC suboptymalnych wykorzystujących modele liniowe (zlinearyzowane) ważne jest, aby w modelu rozmytym TS obiektu *następniki wszystkich reguł miały identyczną strukturę*. Oznaczając wektor zmiennych ogólnego modelu obiektu przez  $x$ ,  $x = [x_1, x_2, \dots, x_n]^T$ , gdzie elementy wektora  $x$  mogą składać się ze współrzędnych stanu, sterowań czy wyjść obiektu (aktualnych i opóźnionych), reguły  $R^i$  modelu TS można zapisać w postaci

$$R^i : \text{ JEŚLI } x_1(k) \text{ jest } A_1^i \text{ i } \dots \text{ i } x_n(k) \text{ jest } A_n^i \\ \text{ TO } y^i(k+1) = a_0^i + a_1^i x_1(k) + \dots + a_n^i x_n(k).$$

Wyjście modelu rozmytego dane jest formułą

$$y(k+1) = \frac{\sum_{i=1}^r w^i(k)y^i(k+1)}{\sum_{l=1}^r w^l(k)} = \sum_{i=1}^r \tilde{w}^i(k)[a_0^i + \sum_{j=1}^n a_j^i x_j(k)], \quad (3.185)$$

gdzie  $r$  oznacza liczbę reguł,  $w^i(k)$  poziomy aktywacji reguł modelu rozmytego w chwili  $k$  (w punkcie  $x(k)$ ), zaś  $\tilde{w}^i(k)$  unormowane poziomy aktywacji, zob. rozdz. 2.1.2. Korzystając z liniowości i identycznej struktury następników reguł możemy model (3.185) przedstawić w postaci

$$y(k+1) = a_0(k) + \sum_{j=1}^n a_j(k)x_j(k), \quad (3.186)$$

gdzie

$$a_j(k) = \sum_{i=1}^r \tilde{w}^i(k)a_j^i, \quad j = 0, \dots, n. \quad (3.187)$$

Uzyskujemy w ten sposób model rozmyty w postaci *modelu liniowego o zmiennych (rozmytych) współczynnikach*, zależnych od aktualnego stanu obiektu. Jeśli teraz w bieżącej chwili  $k$  stan obiektu wynosi  $x(k)$ , to najprostszym, naturalnym sposobem generacji lokalnego modelu liniowego dla wykorzystania w zadaniu optymalizacji algorytmu predykcyjnego jest przyjęcie modelu postaci (3.186). Jest to model ze współczynnikami o wartościach odpowiadających aktualnemu stanowi obiektu w chwili  $k$ , a dokładniej, odpowiadających poziomom aktywacji reguł wyliczonych dla aktualnego stanu obiektu w chwili  $k$ .

### Algorytmy FMPC-NSL (Fuzzy MPC-NSL)

Realizacja *algorytmów typu FMPC-NSL* jest dla modelu rozmytego z następnikami o identycznej strukturze naturalna i stosunkowo prosta. W każdej iteracji algorytmu (chwili próbkowania) wyznaczamy model liniowy postaci (3.186), w zależności od przyjętego sposobu modelowania będzie to np. model zdefiniowany współczynnikami odpowiedzi skokowych, model w postaci dyskretnego równania różnicowego, czy liniowych równań stanu. W rezultacie, w każdym kroku nieliniowego algorytmu predykcyjnego typu MPC-NSL sterowanie generowane będzie jak w algorytmie liniowym, odpowiednio DMC, GPC czy MPCs – prowadząc do algorytmów nieliniowych, które nazwać możemy, odpowiednio, FDMC-NSL, FGPC-NSL czy FMPCS-NSL. Wzory definiujące elementy trajektorii swobodnej i wymuszanej będą w tych algorytmach standardowymi wzorami algorytmów DMC, GPC czy MPCs, ale ze zmieniającymi się z kroku na krok współczynnikami, odpowiadającymi zmieniającym się współczynnikom stosowanych modeli liniowych.

Rozważmy *algorytm FDMC-NSL*. Zgodnie z przyjętą strukturą modelowania rozmytego dziedzina modelu podzielona jest na  $r$  zbiorów rozmytych i każdemu z tych zbiorów odpowiada jeden lokalny model liniowy obiektu dany współczynnikami odpowiedzi skokowej  $s_j^i$ ,  $j = 1, \dots, D$ ,  $i = 1, \dots, r$ , gdzie  $D$  jest wspólną dla wszystkich modeli liniowych liczbą elementów odpowiedzi skokowych (zob. rozdz. 3.2.1). Jedynymi związanymi z modelem elementami algorytmu DMC są definiujące odpowiedzi swobodną i wymuszaną macierze  $\mathbf{M}^P$  i  $\mathbf{M}$ , zob. (3.34) i (3.35) w rozdz. 3.2.2. W przypadku rozważanego algorytmu rozmytego, będą one zbudowane z elementów odpowiedzi skokowych o wartościach zależnych od aktualnej chwili  $k$ ,  $s_j = s_j(k) = \sum_{i=1}^r \tilde{w}^i(k) s_j^i$ , gdzie współczynniki  $\tilde{w}^i(k)$  oznaczają unormowane poziomy aktywacji reguł modelu rozmytego w chwili  $k$ . W każdym z kolejnych momentów próbkowania przyrost sterowania generowany będzie na podstawie algorytmu DMC z macierzami  $\mathbf{M}^P = \mathbf{M}^P(k)$  i  $\mathbf{M} = \mathbf{M}(k)$ , zbudowanymi z elementów  $s_j(k)$  (w przypadku wielowymiarowym: elementów macierzy  $\mathbf{S}_j(k)$ ).

W *algorytmie FGPC-NSL* od poziomów aktywacji reguł modelu rozmytego w chwili  $k$  zależą wartości współczynników  $a_j = a_j(k)$  i  $b_j = b_j(k)$  równań różnicowych modelujących obiekt, na podstawie których konstruowane będą macierze  $\mathbf{F} = \mathbf{F}(k)$  i  $\mathbf{G}^{PG} = \mathbf{G}^{PG}(k)$  służące do wyznaczenia trajektorii swobodnej algorytmem GPC oraz macierz dynamiczna  $\mathbf{M} = \mathbf{M}(k)$  ( $\mathbf{G}^{FG}(k)$ ) potrzebna do wyznaczania trajektorii wymuszonej, zob. rozdz. 3.3. Elementy trajektorii swobodnej mogą być też wyznaczone rekurencyjnie z wykorzystaniem wzorów (3.89).

Podobnie, w *algorytmie FMPCS-NSL* od poziomów aktywacji reguł modelu rozmytego w każdej chwili zależą współczynniki równań stanu, określających macierze  $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}(k)$ ,  $\mathbf{M}_x = \mathbf{M}_x(k)$  i  $\mathbf{V} = \mathbf{V}(k)$ , definiujące w danym kroku  $k$  predykcje algorytmem MPC, zob. rozdz. 3.4. W sytuacjach, gdy pełny wektor stanu jest niedostępny, predykcje te zależą od estymowanych wartości stanu i zakłóceń. W przypadku nieliniowym powinien być zastosowany wówczas nieliniowy obserwator stanu. Projektowanie takich obserwatorów nie jest sprawą tak prostą, jak w przypadku liniowym. Sposobem postępowania jest tu zastosowanie rozszerzonego obserwatora Luenbergera lub rozszerzonego filtru Kalmana.[79] Ponieważ stosujemy nieliniowy rozmyty model TS obiektu oparty na lokalnych modelach w postaci liniowych równań stanu, to polecić tu można również zastosowanie nieliniowego rozmytego obserwatora stanu konstruowanego w analogiczny sposób, jako system rozmyty TS, w oparciu o lokalne liniowe obserwatory, zob. [137].

### Algorytmy FMPC-NPL (Fuzzy MPC-NPL)

Przejdziemy teraz do nieco trudniejszych w realizacji *algorytmów typu NPL*, tzn. FMPC-NPL i FMPC-NPLT, z nieliniową predykcją i linearyzacją dla potrzeb optymalizacji. Rozpocznijemy od przypadku z modelami liniowymi o postaci wykorzystywanej w algorytmie GPC.

#### Algorytmy FGPC-NPL(T)

Podobnie jak w rozdziale 3.3.1 ograniczymy się do przypadku skalarowego. Przyjmijmy, że model rozmyty obiektu składa się z  $r$  reguł, przy czym następnikiem każdej z nich jest model liniowy w postaci dyskretnego równania różnicowego odpowiadającego strukturze (3.84). *Model rozmyty do celów nieliniowej predykcji* zapiszemy stąd w postaci zbioru reguł

$$\begin{aligned} R^i : \text{ JEŚLI } & y(k) \text{ jest } A_0^i \text{ i } y(k-1) \text{ jest } A_1^i \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^i \\ & \text{ i } u(k) \text{ jest } B_0^i \text{ i } u(k-1) \text{ jest } B_1^i \text{ i } \dots \text{ i } u(k-m_R) \text{ jest } B_{m_R}^i \\ \text{ TO } & y^i(k+1) = -a_1^i y(k) - \dots - a_{n_A}^i y(k-n_A+1) + \\ & + b_0^i u(k) + b_1^i u(k-1) + \dots + b_{n_B}^i u(k-n_B), \quad (3.188) \end{aligned}$$

gdzie  $i$  indeksuje reguły i jednocześnie zbiory rozmyte w dziedzinie modelu TS (w ogólności wielowymiarowe),  $A_j^i \in \mathbb{Y}_j$ ,  $B_j^i \in \mathbb{U}_j$ , zaś  $a_j^i$  i  $b_j^i$  są współczynnikami funkcji następników reguł,  $i = 1, \dots, r$ . Elementami każdego ze zbiorów  $\mathbb{Y}_j = \{Y_{j1}, \dots, Y_{jr_{y_j}}\}$  są zbiory rozmyte pokrywające obszar zmiennej  $y(k-j)$ ,  $j = 0, \dots, n_R$ , analogicznie  $\mathbb{U}_j = \{U_{j1}, \dots, U_{jr_{u_j}}\}$  dla  $u(k-j)$ ,  $j = 0, \dots, m_R$ . Przedstawiona sytuacja jest najbardziej ogólna, często mamy  $\mathbb{Y}_0 = \dots = \mathbb{Y}_{n_R} = \mathbb{Y}$ ,  $\mathbb{Y} = \{Y_1, \dots, Y_{r_y}\}$ , tzn. partycje zakresu zmienności  $y(k)$  są takie same niezależnie od chwili  $k$ , i analogicznie  $\mathbb{U}_1 = \dots = \mathbb{U}_{m_R} = \mathbb{U}$ ,  $\mathbb{U} = \{U_1, \dots, U_{r_u}\}$  (zob. rozdz. 2.1.2 i 2.2.2).

Zbiór reguł dopełnia standardowy wzór na konkluzję finalną, czyli wyjście modelu

$$y(k+1) = \sum_{i=1}^r \tilde{w}^i(k) y^i(k+1), \quad (3.189)$$

gdzie  $\tilde{w}^i(k)$  to unormowane poziomy aktywacji reguł (3.188). Zwróćmy uwagę, że poziomy aktywacji reguł w chwili  $k$  zależą od wartości  $n_R + 1$  wyjść  $y(k)$ ,  $y(k-1)$ , ...,  $y(k-n_R)$  oraz  $m_R$  wejść  $u(k-1)$ ,  $u(k-2)$ , ...,  $u(k-m_R)$  – a dokładniej, zależą od wartości funkcji przynależności tych wyjść i wejść do zbiorów rozmytych  $A_j^i$  i  $B_j^i$ , gdzie, w ogólności,  $n_R \neq n_A$  i  $m_R \neq n_B$ .

Związłą postacią zapisu modelu (3.188), (3.189), zgodną z ogólnym wzorem (3.186), jest

$$y(k+1) = -a_1(k)y(k) - \dots - a_{n_A}(k)y(k-n_A+1) + \\ + b_0(k)u(k) + b_1(k)u(k-1) + \dots + b_{n_B}(k)u(k-n_B), \quad (3.190)$$

gdzie

$$a_j(k) = \sum_{i=1}^r \tilde{w}^i(k)a_j^i, \quad b_j(k) = \sum_{i=1}^r \tilde{w}^i(k)b_j^i. \quad (3.191)$$

Przyjmując model zakłócenia  $d(k)$  dodawanego do wyjścia, z prognozą typu DMC,

$$d(k) = y(k) - \left[ -\sum_{j=1}^{n_A} a_j(k-1)y(k-j) + \sum_{j=0}^{n_B} b_j(k-1)u(k-1-j) \right],$$

możemy przystąpić do sformułowania równań *nieliniowej predykcji*.

Wyjście  $y^0(k+1|k)$  przewidywane w chwili  $k$  na chwilę  $k+1$  można wyznaczyć bezpośrednio z zależności (3.188) i (3.189), czy równoważnie ze wzoru (3.190). Dostajemy stąd

$$y^0(k+1|k) = -a_1(k)y(k) - \dots - a_{n_A}(k)y(k-n_A+1) + \\ + b_0(k)u^0(k|k) + b_1(k)u(k-1) + \dots + b_{n_B}(k)u(k-n_B) + d(k), \quad (3.192)$$

gdzie  $u^0(k|k)$  jest pierwszym elementem trajektorii początkowej przewidywanych sterowań  $\mathcal{U}^0(k)$ , zob. (3.173).

Wartości wyjścia predykowane na chwile następne wyliczać będziemy stosując, rekurencyjnie, predykcję jednokrokową z wykorzystaniem zależności (3.188) i (3.189). Stąd wyjście przewidywane na chwilę  $(k+2|k)$  wyznaczamy z wzorów

$$\text{JEŚLI } y^0(k+1|k) \text{ jest } A_0^i \text{ i } y(k) \text{ jest } A_1^i \text{ i } \dots \text{ i } y(k-n_R+1) \text{ jest } A_{n_R}^i \\ \text{i } u^0(k+1|k) \text{ jest } B_0^i \text{ i } u^0(k|k) \text{ jest } B_1^i \text{ i } \dots \text{ i } u(k-m_R+1) \text{ jest } B_{m_R}^i$$

$$\text{TO } y^{0i}(k+2|k) = -a_1^i y^0(k+1|k) - a_2^i y(k) - \dots - a_{n_A}^i y(k-n_A+2) + \\ + b_0^i u^0(k+1|k) + b_1^i u^0(k|k) + \dots + b_{n_B}^i u(k-n_B+1) + d(k), \quad (3.193)$$

$$y^0(k+2|k) = \sum_{i=1}^r \tilde{w}^i(k+1|k)y^{0i}(k+2|k), \quad (3.194)$$

gdzie  $u^0(k+1|k)$  jest drugim elementem trajektorii początkowej sterowań, zaś  $\tilde{w}^i(k+1|k)$  to unormowane poziomy aktywacji reguł (3.193). Wzory powyższe uzupełnione o konkluzję finalną zapisać można w zwężłej postaci

$$y^0(k+2|k) = -a_1(k+1|k)y^0(k+1|k) - \dots - a_{n_A}(k+1|k)y(k-n_A+2) \\ + b_0(k+1|k)u^0(k+1|k) + \dots + b_{n_B}(k+1|k)u(k-n_B+1) + d(k), \quad (3.195)$$

gdzie

$$a_j(k+1|k) = \sum_{i=1}^r \tilde{w}^i(k+1|k)a_j^i, \quad b_j(k+1|k) = \sum_{i=1}^r \tilde{w}^i(k+1|k)b_j^i. \quad (3.196)$$

Predykcję wartości wyjść na kolejne chwile  $(k+j|k)$  kontynuujemy, analogicznie, aż do chwili  $(k+N|k)$ . Ogólna postać wzoru predykcyjnego, którego szczególnymi przypadkami dla  $j=1$  i  $j=2$  są wzory (3.192) i (3.195), jest następująca

$$y^0(k+p|k) = - \sum_{j=1}^{\min\{n_A, p-1\}} a_j(k+p-1|k)y^0(k+p-j|k) + \\ - \sum_{j=\min\{n_A, p-1\}+1}^{n_A} a_j(k+p-1|k)y(k+p-j) + \\ + \sum_{j=0}^{\min\{n_B+1, p\}-1} b_j(k+p-1|k)u^0(k+p-1-j|k) + \\ + \sum_{j=\min\{n_B+1, p\}}^{n_B} b_j(k+p-1|k)u(k+p-1-j) + d(k), \quad (3.197)$$

gdzie

$$a_j(k+p-1|k) = \sum_{i=1}^r \tilde{w}^i(k+p-1|k)a_j^i, \quad (3.198)$$

$$b_j(k+p-1|k) = \sum_{i=1}^r \tilde{w}^i(k+p-1|k)b_j^i, \quad (3.199)$$

przy czym  $a_j(k|k) = a_j(k)$ ,  $b_j(k|k) = b_j(k)$ , zob. (3.191),  $\tilde{w}^i(k+p-1|k)$  to unormowane poziomy aktywacji nie widocznych bezpośrednio w powyższym wzorze reguł modelu rozmytego, wyznaczone na chwilę  $k+p-1$  w chwili  $k$ , zaś  $u^0(k+p|k)$ ,  $p=0, \dots, N-1$  to elementy trajektorii początkowej sterowań  $\mathcal{U}^0(k)$ .

Dla obliczania składowej wymuszanej trajektorii wyjść na horyzoncie predykcji trzeba wyznaczyć macierz dynamiczną. Znając współczynniki (3.191) modelu liniowego (zlinearyzowanego) w chwili  $k$ , możemy wyznaczyć elementy odpowiedzi skokowej tego modelu  $s_j(k)$ ,  $j = 1, 2, \dots$ , bezpośrednio ze wzoru (3.85)

$$s_j(k) = - \sum_{i=1}^{\min\{j-1, n_A\}} a_i(k) s_{j-i}(k) + \sum_{i=0}^{\min\{j-1, n_B\}} b_i(k). \quad (3.200)$$

Macierz dynamiczną  $\mathbf{M}(k)$  odpowiadającą tym współczynnikom obliczamy następnie zgodnie z zależnością (3.30), wstawiając jedynie uzyskane powyżej wartości  $s_j(k)$  w miejsce  $s_j$ ,  $j = N_1, \dots, N$ . Wyznaczona w ten sposób macierz dynamiczna odpowiada modelowi rozmytemu dla aktualnej chwili  $k$ , tzn. chwili początkowej horyzontu predykcji – a więc strukturze NPL.

Można rozważyć nieco inny, prostszy w realizacji obliczeniowej on-line sposób wyznaczania macierzy  $\mathbf{M}(k)$  w algorytmie FGPC-NPL: obliczenie, na etapie projektowania regulatora (off-line), korzystając z wzoru (3.85), elementów odpowiedzi skokowych wszystkich modeli liniowych występujących w następnikach reguł (3.188). Następnie, dla każdej z odpowiedzi skokowych, wyznaczenie macierzy dynamicznych  $\mathbf{M}^i$ ,  $i = 1, \dots, r$ . Natomiast w trakcie działania algorytmu predykcyjnego wyznaczanie w każdej chwili  $k$  macierzy  $\mathbf{M}(k)$  ze wzoru

$$\mathbf{M}(k) = \sum_{i=1}^r \tilde{w}^i(k) \mathbf{M}^i.$$

Do omówienia pozostaje jeszcze nieliniowy sposób wyznaczania macierzy  $\mathbf{M}(k)$  odpowiadający strukturze NPLT (NPL z linearyzacją wokół trajektorii). Dla modelu rozmytego typu GPC, linearyzacja wokół trajektorii na horyzoncie predykcji jest naturalna i prosta, odpowiednie współczynniki kolejnych modeli liniowych są bowiem generowane już przy obliczaniu predykowanej trajektorii wyjść. W przypadku struktury NPLT, kolejne wiersze macierzy dynamicznej  $\mathbf{M}(k)$  (w przypadku wyjścia  $y$  wektorowego, bloki poziome stanowiące podmacierze  $\mathbf{M}(k)$  o liczbie wierszy równej wymiarowi wektora wyjść obiektu, tzn. poziome podmacierze o wymiarowości  $n_y \times (n_u \cdot N_u)$  każda) składają się ze współczynników odpowiedzi skokowych odpowiadających sukcesywnie wyliczanym współczynnikiem  $a_j(k+p|k)$  i  $b_j(k+p|k)$  modeli liniowych generowanych przy wyznaczaniu elementów trajektorii początkowej wyjść przewidywanych  $y^0(k+p+1|k)$ ,  $p = 0, 1, \dots, N-1$ . Stąd, dla  $n_y = 1$ , macierz ta będzie postaci, por. (3.30):

$$\mathbf{M}(k) = \begin{bmatrix} s_{N_1}^{(N_1-1|k)} & \cdots & s_1^{(N_1-1|k)} & 0 & \cdots & 0 \\ s_{N_1+1}^{(N_1|k)} & \cdots & s_2^{(N_1|k)} & s_1^{(N_1|k)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{N_u}^{(N_u-1|k)} & \cdots & s_{N_u-N_1+1}^{(N_u-1|k)} & s_{N_u-N_1}^{(N_u-1|k)} & \cdots & s_1^{(N_u-1|k)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_N^{(N-1|k)} & \cdots & s_{N-N_1+1}^{(N-1|k)} & s_{N-N_1}^{(N-1|k)} & \cdots & s_{N-N_u+1}^{(N-1|k)} \end{bmatrix}, \quad (3.201)$$

gdzie  $s_j^{(p|k)} = s_j(k+p|k)$  to elementy odpowiedzi skokowej modelu liniowego ze współczynnikami  $a_j(k+p|k)$  i  $b_j(k+p|k)$ , zob. (3.198) i (3.199).

Dysponując trajektorią trajektorią początkową  $\mathcal{Y}^0(k) = [y^0(k+N_1|k) \cdots \cdots y^0(k+N|k)]^T$  oraz macierzą dynamiczną  $\mathbf{M}(k)$ , potrzebną do wyznaczenia trajektorii wymuszanej  $\Delta\mathcal{Y}(k) = \mathbf{M}(k)\Delta\mathcal{U}(k)$ , można przeprowadzić optymalizację sterowań rozwiązując zadanie programowania kwadratowego, co kończy działanie algorytmu FGPC-NPLT w chwili próbkowania  $k$ .

W przypadku algorytmu FGPC-NPLT+, kolejne iteracje odpowiedzi swobodnej w chwili  $k$  wyliczamy identycznie, jedynie należy przyjmować trajektorię sterowań początkowych  $\mathcal{U}^j(k)$  w miejsce trajektorii  $\mathcal{U}^0(k)$ . Natomiast przyrosty sterowań wyliczamy z zadania optymalizacji (3.182).

### Algorytmy FDMC-NPL(T)

Dla algorytmów FDMC-NPL(T) postępowanie jest analogiczne jak w omówionym wyżej przypadku algorytmów FGPC-NPL(T). Jedynie w następnikach reguł modelu rozmytego obiektu wystąpią inne formuły, odpowiadające modelowi opartemu na współczynnikach odpowiedzi skokowej, zob. rozdz. 3.2.1.

Model rozmyty do celów nieliniowej predykcji trajektorii swobodnej zapiszemy w postaci zbioru  $r$  reguł z poprzednikami w ogólnej postaci takiej, jak wyżej w algorytmie FGPC, oraz z następnikami w postaci

$$y^{0i}(k+p|k) = y(0) + \sum_{j=p+1}^{k+p} s_j^i \Delta u(k+p-j) + d(k),$$

por. (3.12), gdzie  $s_j^i$  to  $j$ -ty element odpowiedzi skokowej modelu liniowego występującego w  $i$ -tej regule,  $i = 1, \dots, r$ . Zakłócenie  $d(k)$  przyjmujemy jako stałe na horyzoncie predykcji i równe

$$d(k) = y(k) - \left[ y(0) + \sum_{j=1}^k s_j(k-1) \Delta u(k-j) \right],$$

gdzie  $s_j(k-1)$  to współczynniki odpowiedzi skokowej modelu liniowego wykorzystanego w chwili  $k-1$ . Łącząc dwa powyższe równania dostajemy

$$y^{0i}(k+p|k) = y(k) + \sum_{j=1}^{D-1} [s_{j+p}^i - s_j(k-1)] \Delta u(k-j),$$

gdzie  $D$  oznacza horyzont dynamiki odpowiedzi skokowych, zob. rozdz. 3.2.1. Stąd, wykonywana w chwili  $k$  *nieliniowa predykcja trajektorii początkowej (swobodnej)* na horyzoncie  $N$  realizowana jest przez *kolejne (rekurencyjne)* zastosowanie reguł modelu

$$\begin{aligned} R_{pr}^i : \text{JEŚLI } & y^0(k+p-1|k) \text{ jest } A_0^i \text{ i } \dots \text{ i } y^0(k+1|k) \text{ jest } A_{p-2}^i \\ & \text{i } y(k) \text{ jest } A_{p-1}^i \text{ i } \dots \text{ i } y(k+p-1-n_R) \text{ jest } A_{n_R}^i \\ & \text{i } u^0(k+p-1|k) \text{ jest } B_0^i \text{ i } \dots \text{ i } u^0(k|k) \text{ jest } B_{p-1}^i \\ & \text{i } u(k-1) \text{ jest } B_p^i \text{ i } \dots \text{ i } u(k+p-1-m_R) \text{ jest } B_{m_R}^i \end{aligned}$$

$$\text{TO } y^{0i}(k+p|k) = y(k) + \sum_{j=1}^{D-1} [s_{j+p}^i - s_j(k-1)] \Delta u(k-j),$$

i konkluzji wnioskowania rozmytego

$$y^0(k+p|k) = \sum_{i=1}^r \tilde{w}^i(k+p-1|k) y^{0i}(k+p|k), \quad p = 1, 2, \dots, N,$$

gdzie  $\tilde{w}^i(k+p-1|k)$  to unormowane poziomy aktywacji reguł w chwili  $(k+p-1|k)$ , zaś  $u^0(k|k), \dots, u^0(k+N-1|k)$  to składowe trajektorii początkowej sterowań  $\mathcal{U}^0(k)$  (3.173). Wzory powyższe można interpretować jako użycie do obliczenia wartości początkowej  $y^0(k+p|k)$  w chwili  $p$  horyzontu predykcji modelu liniowego o współczynnikach  $s_j(k+p-1|k)$ , gdzie

$$s_j(k+p-1|k) = \sum_{i=1}^r \tilde{w}^i(k+p-1|k) s_j^i, \quad j = 1, 2, \dots, N. \quad (3.202)$$

Stąd wzór na predykcję można zapisać w postaci

$$y^0(k+p|k) = y(k) + \sum_{j=1}^{D-1} [s_{j+p}(k+p-1|k) - s_j(k-1)] \Delta u(k-j),$$

$$p = 1, \dots, N \quad (3.203)$$

Podkreślmy, że wyznaczanie składowych nieliniowej odpowiedzi swobodnej trzeba prowadzić kolejno, rekurencyjnie, gdyż dla wyznaczenia wartości  $y^0(k+p|k)$  potrzebne jest wyznaczenie poziomów aktywacji reguł

$\tilde{w}^i(k+p-1|k)$  zależnych, w ogólności, od poprzednich elementów trajektorii swobodnej, poprzez ich występowanie w poprzednikach reguł modelu rozmytego.

Ponieważ znamy wartości współczynników  $s_j(k)$  odpowiedzi skokowej modelu liniowego (zlinearyzowanego) wykorzystywanego w chwili  $k$ , to mamy stąd bezpośrednio macierz dynamiczną  $\mathbf{M}(k)$  potrzebną do wyznaczania odpowiedzi wymuszanej na horyzoncie predykcji w algorytmie FDMC-NPL. Natomiast równie łatwo policzyć też macierz  $\mathbf{M}(k)$  dla algorytmu FDMC-NPLT, gdyż mamy już policzone współczynniki  $s_j(k+p|k)$  modeli liniowych odpowiadających linearyzacji w każdej z chwil  $p$  horyzontu predykcji,  $p = 0, \dots, N-1$ . W rozważanym przypadku są to współczynniki odpowiednich odpowiedzi skokowych, można je bezpośrednio wykorzystać do konstrukcji macierzy dynamicznej postaci (3.201).

### Algorytmy FMPCS-NPL(T)

Dla algorytmów FMPCS-NPL(T) postępowanie jest analogiczne jak w omówionych wyżej przypadkach algorytmów FGPC-NPL(T) czy FDMC-NPL(T). Jedynie w następnikach reguł modelu rozmytego obiektu wystąpią inne formuły, odpowiadające modelowi opartemu na równaniach stanu.

Model rozmyty do celów nieliniowej predykcji trajektorii początkowej (swobodnej) zapisujemy w postaci zbioru  $r$  reguł z poprzednikami i następnikami w ogólnej postaci (zob. rozdz. 2.2.1)

$$\begin{aligned} R_{ob}^i : \quad & \text{JEŚLI } x_1(k) \text{ jest } A_1^i \text{ i } x_2(k) \text{ jest } A_2^i \text{ i } \dots \text{ i } x_{n_x}(k) \text{ jest } A_{n_x}^i \\ & \text{TO } x^i(k+1) = \mathbf{A}_i x(k) + \mathbf{B}_i u(k), \quad y^i(k) = \mathbf{C}_i x(k), \end{aligned} \quad (3.204)$$

$i = 1, \dots, r$ .

Dla przypadku stanu mierzonego  $x(k)$ , model zakłóceń niemierzalnych stanu  $v(k)$  przyjmuje postać

$$\begin{aligned} v(k) &= x(k) - x(k|k-1) \\ &= x(k) - \sum_{i=1}^r \tilde{w}^i(k-1) [\mathbf{A}_i x(k-1) + \mathbf{B}_i u(k-1)], \end{aligned} \quad (3.205)$$

gdzie  $\tilde{w}^i(k-1)$  to unormowane poziomy aktywacji reguł w chwili  $k-1$ . Predykcję stanu i wyjść wykonujemy, stosując rekurencyjnie reguły modelu

$$\begin{aligned} R_{pr}^i : \quad & \text{JEŚLI } x_1^0(k+p|k) \text{ jest } A_1^i \text{ i } \dots \text{ i } x_{n_x}^0(k+p|k) \text{ jest } A_{n_x}^i \\ & \text{TO } x^{0i}(k+p+1|k) = \mathbf{A}_i x^0(k+p|k) + \mathbf{B}_i u^0(k+p|k) + v(k), \end{aligned} \quad (3.206)$$

a następnie konkluzję wnioskowania rozmytego

$$x^0(k+p+1|k) = \sum_{i=1}^r \tilde{w}^i(k+p|k)x^{0i}(k+p+1|k), \quad p = 0, 1, \dots, N-1,$$

gdzie  $\tilde{w}^i(k+p|k)$  to unormowane poziomy aktywacji reguł w chwili predykcji  $(k+p|k)$ ,  $x^0(k|k) = x(k)$ , zaś  $u^0(k|k), \dots, u^0(k+N-1|k)$  to składowe trajektorii początkowej sterowań  $\mathcal{U}^0(k)$  (3.173). Wzory powyższe można interpretować jako użycie do obliczenia wartości trajektorii początkowej stanu  $x^0(k+p+1|k)$  modelu liniowego o macierzach

$$\mathbf{A}(k+p|k) = \sum_{i=1}^r \tilde{w}^i(k+p|k)\mathbf{A}_i, \quad (3.207)$$

$$\mathbf{B}(k+p|k) = \sum_{i=1}^r \tilde{w}^i(k+p|k)\mathbf{B}_i, \quad p = 0, \dots, N-1. \quad (3.208)$$

Trajektoria początkowa wyjścia dana jest oczywiście wzorem

$$y^0(k+p|k) = \mathbf{C}x^0(k+p|k), \quad p = 1, \dots, N. \quad (3.209)$$

Podkreślmy, że wyznaczanie składowych nieliniowej trajektorii początkowej trzeba stanu prowadzić kolejno, rekurencyjnie, gdyż dla wyznaczenia wartości  $x^0(k+p+1|k)$  potrzebne jest wyznaczenie poziomów aktywacji  $\tilde{w}^i(k+p|k)$  zależnych, w ogólności, od elementów trajektorii początkowej stanu  $x^0(k+p|k)$  wyznaczonych na chwilę poprzednią horyzontu predykcji, poprzez ich występowanie w poprzednikach reguł modelu rozmytego.

Dla przypadku stanu estymowanego obiektu  $\hat{x}(k)$ , i zastosowania modelu zakłóceń niemierzalnych stanu  $v(k)$ , przyjmuje on postać

$$v(k) = \hat{x}(k) - \sum_{i=1}^r \tilde{w}^i(k-1)[\mathbf{A}_i\hat{x}(k-1) + \mathbf{B}_iu(k-1)], \quad (3.210)$$

zaś predykcję stanu i wyjść wykonujemy rekurencyjnie, stosując te same reguły (3.206), jedynie teraz zapoczątkowane (dla  $p=0$ ) nie pomiarem, a estymatorem stanu,  $x^0(k|k) = \hat{x}(k)$ . Stosując estymację stanu jedynie samego obiektu, formułę na trajektorię początkową wyjścia (3.209) należy uzupełnić członem korekcyjnym, zob. rozdz. 3.4.2,

$$y^0(k+p|k) = \mathbf{C}x^0(k+p|k) + [y(k) - \mathbf{C}\hat{x}(k)] \quad p = 1, \dots, N. \quad (3.211)$$

Macierz dynamiczną  $\mathbf{M}(k)$  dla algorytmu NPL wyznaczamy standardowo jak w algorytmie MPSC z modelem liniowym, jedynie zamiast macierzy  $\mathbf{A}$  i  $\mathbf{B}$  uwzględniamy macierze  $\mathbf{A}(k) = \mathbf{A}(k|k)$  i  $\mathbf{B}(k) = \mathbf{B}(k|k)$ , dane

wzorami (3.207) i (3.208) dla  $p = 0$ . Przy zastosowaniu linearyzacji wokół trajektorii (algorytm NPLT), kolejne bloki wierszowe macierzy dynamicznej (o liczbie wierszy  $n_y$  każdy) wyznaczamy też z wzorów (3.207) i (3.208), ale tym razem kolejno dla  $p = 0, 1, \dots, N - 1$ , por. (3.180) w rozdz. 3.5.2.

Zauważmy, że w przypadku rozmytego modelu obiektu z następnikami reguł w postaci lokalnych równań stanu, szczególnie łatwo jest realizować rozszerzony filtr Kalmana. Model nieliniowy rozmyty można bowiem przedstawić jako model liniowy ze zmiennymi elementami macierzy modelu, zgodnie z zależnościami (3.207)-(3.208) dla  $p = 0$ . Nie trzeba w każdym kroku dokonywać klasycznej linearyzacji modelu nieliniowego.

### Przykład 3.13.

Jest to przykład dla prostego zilustrowania projektowania i analizy regulatorów nieliniowych MPC wykorzystujących rozmyty model obiektu regulacji rozpatrywanego już w przykładzie 2.6 w rozdziale 2. Model ten opisany jest następującymi regułami rozmytymi

$$R^1 : \text{JEŚLI } y(k) \text{ jest } Y_1 \text{ TO } y^1(k+1) = 0.7y(k) + 0.8u(k), \quad (3.212)$$

$$R^2 : \text{JEŚLI } y(k) \text{ jest } Y_2 \text{ TO } y^2(k+1) = 0.3y(k) + 0.2u(k). \quad (3.213)$$

Zbiory rozmyte  $Y_1$  i  $Y_2$  zdefiniowane są sigmoidalnymi funkcjami przynależności

$$\mu_{Y_1}(y(k)) = \frac{1}{1 + \exp(6y(k))},$$

$$\mu_{Y_2}(y(k)) = 1 - \mu_{Y_1}(y(k)),$$

przedstawionymi na rys. 2.23 a). Liniowe modele występujące w następnikach reguł (3.212) i (3.213) mają istotnie różne wzmocnienia, jak i dynamikę, co pokazano zamieszczając na rysunku 2.23 b) ich odpowiedzi skokowe.

Postać (3.186) modelu rozmytego jest w przykładzie następująca

$$y(k+1) = \mu_{Y_1}(y(k))[0.7y(k) + 0.8u(k)] + \mu_{Y_2}(y(k))[0.3y(k) + 0.2u(k)]$$

$$= -a_1(k)y(k) + b_0(k)u(k), \quad (3.214)$$

gdzie

$$-a_1(k) = 0.7\mu_{Y_1}(y(k)) + 0.3\mu_{Y_2}(y(k)),$$

$$b_0(k) = 0.8\mu_{Y_1}(y(k)) + 0.2\mu_{Y_2}(y(k)),$$

przy czym poziomy aktywacji reguł są dane wzorami

$$w^1(k) = \mu_{Y_1}(y(k)),$$

$$w^2(k) = \mu_{Y_2}(y(k)) = 1 - \mu_{Y_1}(y(k)).$$

Przedstawiony model jest istotnie nieliniowy i nie jest możliwe zaprojektowanie regulatora liniowego dobrze pracującego w całym obszarze zmienności. Fakt ten sprawdzono, projektując liniowe regulatory GPC dla punktów pracy odpowiadających wartościom  $y = -1$ ,  $y = +1$  i  $y = 0$ . Dwa pierwsze regulatory poprawnie pracowały jedynie w małych otoczeniach swoich punktów pracy, będąc całkowicie nie do przyjęcia w całym zakresie (niestabilność lub bardzo wolne działanie w sąsiednim obszarze). Trzeci, zaprojektowany dla punktu środkowego (najkorzystniejszego), okazał się lepszy, ale też nie spełniający oczekiwań – pokazują to przebiegi wielkości regulowanej i sterowania przedstawione na rysunku 3.50.

Dla przedstawionego obiektu zaprojektowano i porównano regulatory nieliniowe typu NSL, NPL i NO (z nieliniową optymalizacją), przyjmując  $N = 6$ ,  $N_u = 3$ ,  $N_1 = 1$ ,  $\Psi(p) = 1$ ,  $\Lambda(p) = \lambda$ .

*Algorytm NSL* wyznaczamy w każdym kroku  $k$  jako algorytm liniowy (GPC) dla modelu liniowego

$$y(k) = -a_1(k)y(k-1) + b_0(k)u(k-1).$$

Wykorzystujemy w tym celu jeden ze sposobów projektowania podanych w rozdz. 3.3.1.

W *algorytmie NPL* wyznaczamy nieliniową predykcję trajektorii początkowej wyjść zakładając brak przyrostów sterowania na horyzoncie predykcji – tzn. trajektoria początkowa sterowań składa się z wartości  $u(k-1)$ , co prowadzi do wzorów na predykcje wyjść

$$y^0(k+1|k) = -a_1(k)y(k) + b_0(k)u(k-1) + d(k),$$

$$y^0(k+p|k) = -a_1(k+p-1)y^0(k+p-1|k) + b_0(k+p-1|k)u(k-1) + d(k), \quad p = 2, \dots, N,$$

gdzie

$$-a_1(k+p-1|k) = 0.7\mu_{Y_1}(y^0(k+p-1|k)) + 0.3\mu_{Y_2}(y^0(k+p-1|k)),$$

$$b_0(k+p-1|k) = 0.8\mu_{Y_1}(y^0(k+p-1|k)) + 0.2\mu_{Y_2}(y^0(k+p-1|k)), \quad p = 2, \dots, N,$$

$$d(k) = y(k) - [-a_1(k-1)y(k-1) + b_0(k-1)u(k-1)]. \quad (3.215)$$

Potrzebną do wyznaczania trajektorii wymuszanej macierz dynamiczną  $\mathbf{M}(k)$  najprościej liczyć wyznaczając współczynniki odpowiedzi skokowej modelu zlinearyzowanego ze wzoru (3.85).

W algorytmie typu NO wykonujemy w każdym kroku nieliniową optymalizację, wyliczając elementy trajektorii wyjść przewidywanych na horyzoncie predykcji nieliniowym modelem (3.214) uzupełnionym o estymatę zakłócenia  $d(k)$  daną wzorem (3.215). Postępowanie jest rekurencyjne, kolejne wzory wyglądają następująco:

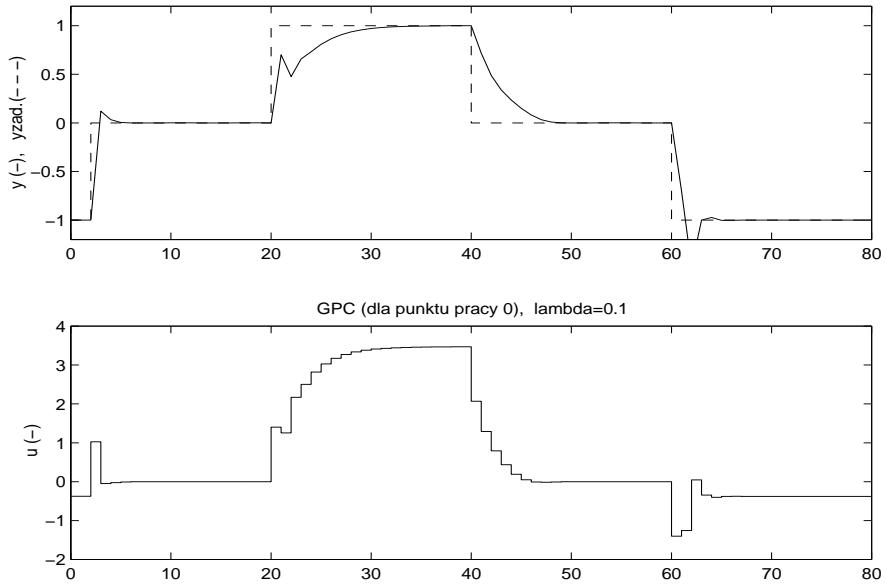
$$\begin{aligned}
 y(k+1|k) &= -a_1(k)y(k) + b_0(k)[u(k-1) + \Delta u(k|k)] + d(k), \\
 y(k+2|k) &= -a_1(k+1|k)y(k+1|k) + b_0(k+1|k)[u(k-1) + \Delta u(k|k) + \\
 &\quad + \Delta u(k+1|k)] + d(k), \\
 y(k+3|k) &= -a_1(k+3|k)y(k+3|k) + b_0(k+3|k)[u(k-1) + \Delta u(k|k) + \\
 &\quad + \Delta u(k+1|k) + \Delta u(k+2|k)] + d(k), \\
 &\quad \vdots \\
 y(k+6|k) &= -a_1(k+5|k)y(k+5|k) + b_0(k+5|k)[u(k-1) + \Delta u(k|k) + \\
 &\quad + \Delta u(k+1|k) + \Delta u(k+2|k)] + d(k).
 \end{aligned}$$

Tym razem współczynniki modelu liczone są na pełnej nieliniowej trajektorii przewidywanej wyjść, zależnej od aktualnej wartości wektora zmiennych decyzyjnych zadania nieliniowej optymalizacji  $\Delta \mathcal{U}(k) = [\Delta u(k|k) \Delta u(k+1|k) \Delta u(k+2|k)]^T$ , tzn.

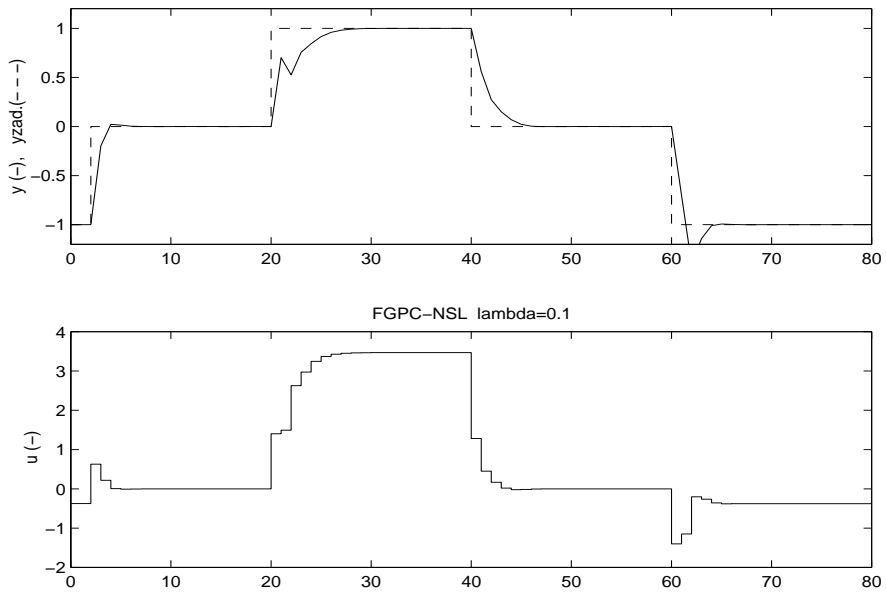
$$\begin{aligned}
 -a_1(k+p-1|k) &= 0.7\mu_{Y_1}(y(k+p-1|k)) + 0.3\mu_{Y_2}(y(k+p-1|k)), \\
 b_0(k+p-1|k) &= 0.8\mu_{Y_1}(y(k+p-1|k)) + 0.2\mu_{Y_2}(y(k+p-1|k)), \\
 &\quad p = 2, \dots, 6.
 \end{aligned}$$

Obliczenia symulacyjne przeprowadzono dla przedstawionych algorytmów i różnych wartości współczynnika  $\lambda$ . Na rysunkach od 3.71 do 3.74 pokazano przebiegi trajektorii wielkości regulowanej i sterowania uzyskane dla  $\lambda = 0.1$  algorytmami, kolejno, liniowym GPC oraz nieliniowymi typu NSL, NPL i NO.

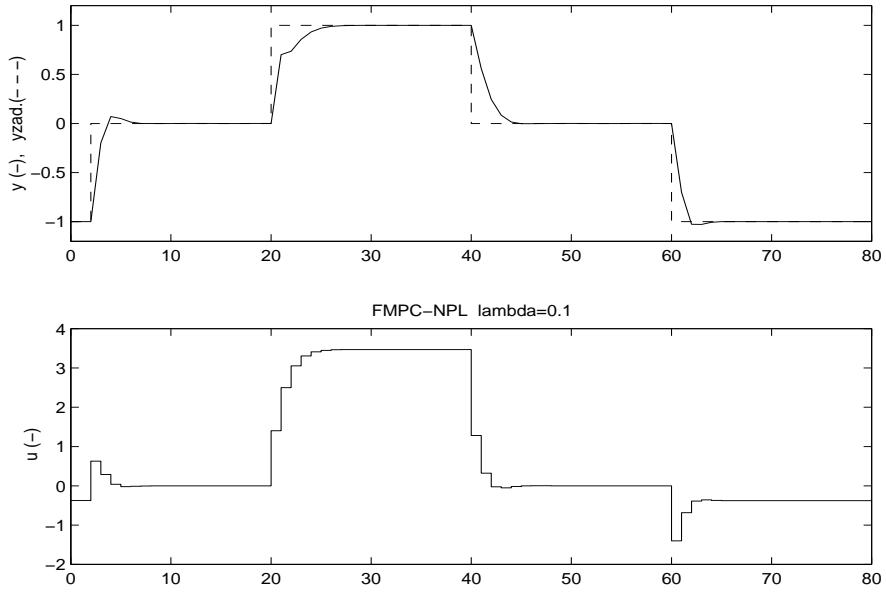
Algorytm FGPC-NSL okazał się nieco lepszy od najlepszego w rozważanej sytuacji liniowego, tzn. algorytmu GPC zaprojektowanego dla punktu pośredniego nieliniowości ( $y = 0$ ), przede wszystkim poprzez zmniejszenie czasu regulacji dla zakresu dodatnich wartości wyjścia. Wprowadzenie algorytmu FGPC-NPL (o porównywalnej złożoności obliczeniowej!) najwyraźniej poprawiło własności układu, dalej redukując czasy regulacji w obszarze  $y \geq 0$  oraz eliminując praktycznie dość silne, niekorzystne przeregulowanie w zakresie ujemnych wartości  $y$  (obserwowane przy skoku wartości zada-



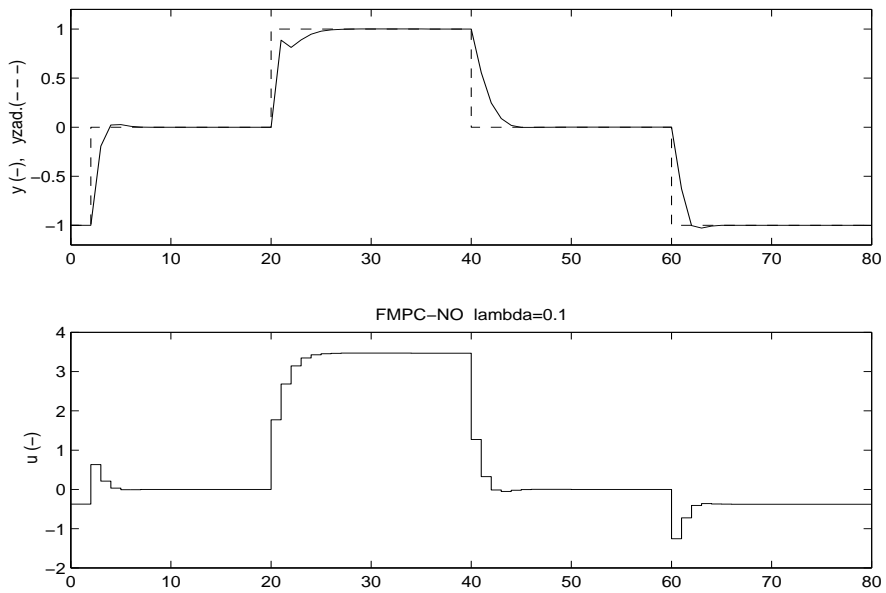
Rysunek 3.71. Trajektorie wielkości regulowanej i sterowania w układzie z regulatorem GPC zaprojektowanym dla punktu  $y = 0$



Rysunek 3.72. Trajektorie wielkości regulowanej i sterowania w układzie z regulatorem FGPC-NSL



Rysunek 3.73. Trajektorie wielkości regulowanej i sterowania w układzie z regulatorem FGPC-NPL



Rysunek 3.74. Trajektorie wielkości regulowanej i sterowania w układzie z regulatorem MPC-NO

nej do  $-1$ ). Zastosowanie algorytmu NO spowodowało już tylko niewielką poprawę.

Warto zaznaczyć, że przebiegi uzyskane przy zastosowaniu algorytmu FGPC-NPL niewiele odbiegają od uzyskanych ze znacznie bardziej wymagającym obliczeniowo algorytmem FMPC-NO, z nieliniową optymalizacją wykonywaną w każdym kroku z wykorzystaniem procedury z *Optimization Toolbox* pakietu MATLAB. Można oczekiwać, że zastosowanie algorytmu FGPC-NPLT czy FGPC-NPLT+ różnicę tę dalej by zmniejszyło.

Interesujące jest też porównanie uzyskanych rezultatów z wynikami w układzie z regulatorem PI rozmytym TS, zamieszczonymi w przykładzie 2.6 w rozdziale 2.  $\square$

#### 3.5.4. Algorytmy FMPC (Fuzzy MPC) w wersji analitycznej

Jeśli model obiektu jest liniowy zarówno dla celów predykcji składowej swobodnej trajektorii wyjść, jak i wyliczania składowej wymuszanej oraz przy rozwiązywaniu zadania optymalizacji algorytmu predykcyjnego nie są uwzględniane dodatkowe ograniczenia wartości sterowań czy wyjść, to zadanie takie ma rozwiązanie analityczne będące liniowym prawem regulacji. Prawa takie wyznaczyliśmy dla algorytmów DMC, GPC i MPCs w rozdziałach, odpowiednio, 3.2, 3.3 i 3.4. W wymienionej sytuacji można projektować algorytmy FMPC inaczej niż to czyniliśmy w rozdziale poprzednim, poświęconym wersjom numerycznym, w ogólności z ograniczeniami uwzględnianymi w zadaniu optymalizacji. Można bowiem najpierw zaprojektować w wydzielonych, lokalnych obszarach pracy obiektu predykcyjne regulatory liniowe. Wykorzystując je następnie łącznie z wnioskowaniem rozmytym uzyskujemy rozmyte prawa regulacji predykcyjnej w wersji analitycznej. Istotą podejścia jest więc zastosowanie ogólnej struktury projektowania regulatora nieliniowego rozmytego TS przedstawionej w rozdziale 2.2. Ponieważ przy projektowaniu regulatorów lokalnych stosowano modele liniowe i prawa regulacji są modyfikowane w każdym kroku algorytmu nieliniowego, to algorytmy regulacji są zbliżone do typu FMPC-NSL. Nie są jednakże identyczne, gdyż wnioskowanie rozmyte dotyczące lokalnych praw regulacji nie jest w ogólności równoważne wnioskowaniu rozmytemu dotyczącemu modeli lokalnych i wyznaczaniu (w każdym kroku) prawa regulacji dla modelu tak linearyzowanego.

Pierwszym etapem jest, jak zawsze przy systemach rozmytych, *zaprojektowanie poprzedników i następników reguł modelu rozmytego TS* obiektu. Etap ten polega na zdefiniowaniu wielkości określających punkty pracy obiektu i podział zakresów ich zmienności na zbiory rozmyte w taki sposób, aby w każdym z powstałych wydzielonych podobszarów (odpowiadających

jednej regule modelu rozmytego) obiekt mógł być przybliżony lokalnym modelem liniowym. Następnie, dla każdego z podobszarów projektujemy *lokalny liniowy regulator predykcyjny w wersji analitycznej*, w zależności od przyjętego sposobu modelowania i projektowania będzie to prawo regulacji DMC, GPC czy MPC – jeśli rozważać algorytmy liniowe przedstawione wcześniej w niniejszej pracy. Finalny nieliniowy rozmyty regulator predykcyjny TS uzyskujemy jako kombinację regulatorów lokalnych, drogą wnioskowania rozmytego.

Rozważmy projektowanie regulatora analitycznego FMPC na przykładzie jednowymiarowego *analitycznego regulatora FDMC* – z lokalnymi liniowymi regulatorami predykcyjnymi DMC [94]. Załóżmy, że wykonano etap pierwszy projektowania uzyskując  $r$  obszarów rozmytych zdefiniowanych poprzednikami reguł modelu rozmytego TS obiektu oraz w każdym z tych obszarów wyznaczono liniowy model obiektu w postaci odpowiedzi skokowej skończonej długości. Przyjmujemy, że każda z tych odpowiedzi skokowych jest o tej samej długości (nie zmniejsza to ogólności, wszystkie krótsze odpowiedzi można odpowiednio przedłużyć). Załóżmy następnie, że dla każdego z lokalnych modeli zaprojektowano prawo regulacji DMC. Przyjmując, jedynie dla zwięzłości zapisu, że wartość zadana zmiennych regulowanych nie zmienia się na horyzoncie predykcji, każde z lokalnych praw regulacji dane będzie wzorem (3.43). W efekcie uzyskujemy regulator analityczny FDMC opisany następującym zestawem reguł

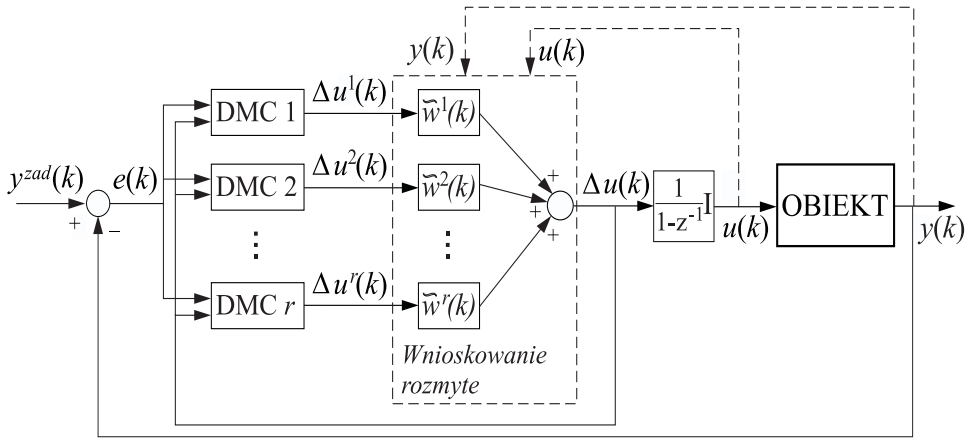
$$\begin{aligned} R_{reg}^i : \text{JEŚLI } & y(k) \text{ jest } A_0^i \text{ i } y(k-1) \text{ jest } A_1^i \text{ i } \dots \text{ i } y(k-n_R) \text{ jest } A_{n_R}^i \\ & \text{i } u(k-1) \text{ jest } B_1^i \text{ i } \dots \text{ i } u(k-m_R) \text{ jest } B_{m_R}^i \\ \text{TO } & \Delta u^i(k) = (k^e)^i e(k) - \sum_{q=1}^{D-1} (k_q^u)^i \Delta u(k-q), \end{aligned} \quad (3.216)$$

gdzie  $i = 1, \dots, r$ , indeksuje reguły, a stąd i lokalne prawa regulacji DMC, zaś  $e(k) = y^{zad}(k) - y(k)$  jest uchybem regulacji. Sygnał wyjściowy regulatora przyjmuje standardową dla struktur TS postać

$$\Delta u(k) = \frac{\sum_{i=1}^r w^i(k) \Delta u^i(k)}{\sum_{l=1}^r w^l(k)} = \sum_{i=1}^r \tilde{w}^i(k) \Delta u^i(k), \quad (3.217)$$

gdzie  $\tilde{w}^i(k)$  są unormowanymi poziomami aktywacji poszczególnych reguł (3.216) w chwili  $k$ -tej. Strukturę regulatora analitycznego FDMC przedstawiono na rysunku 3.75, gdzie liniami przerywanymi zaznaczono blok wnioskowania rozmytego i sygnały do niego prowadzące.

Oczywiście, dla aplikacji w których możliwe jest powodowane przez element wykonawczy wchodzenie na ograniczenia amplitudy czy szybkości



Rysunek 3.75. Struktura regulatora FDMC w wersji analitycznej (wartość zadana stała na horyzoncie predykcyj)

narastania sygnału sterowania, strukturę z rysunku należy zmodyfikować uzupełniając o odpowiednie elementy dokładnie tak, jak to przedstawiono dla regulatora DMC w rozdz. 3.2.2, zob. rys. 3.15.

W sformułowaniu reguł (3.216) przyjęliśmy postać poprzedników ogólną, taką samą jak stosowana w rozdziale 2.2.2 poświęconym dyskretnym regulatorom rozmytym TS ze sprzężeniem od wyjścia. Zaprezentowany regulator analityczny FDMC jest bowiem przypadkiem szczególnym rozważanego tam regulatora z regułami (2.39), dla  $n_C = 1$ ,  $m_D = D$  oraz

$$\begin{aligned}
 c_1^j &= (k^e)^j, \\
 d_1^j &= 1 - (k_1^u)^j, \\
 d_q^j &= (k_{q-1}^u)^j - (k_q^u)^j, \quad q = 2, \dots, D - 1, \\
 d_D^j &= (k_{D-1}^u)^j,
 \end{aligned} \tag{3.218}$$

gdzie  $D$  jest długością odpowiedzi skokowych przyjętych do projektowania regulatorów lokalnych DMC. Dlatego też można wykorzystać sformułowany w rozdziale 2.2.2 wniosek 2.6 do *badania stabilności* układu regulacji z regulatorem analitycznym FDMC [96, 97]. Analiza dotyczy układu regulacji z obiektem modelowanym również systemem rozmytym TS, dlatego też przyjmujemy model obiektu regulowanego w ogólnej postaci takiej, jak w rozdziale 2.2.2. Mianowicie, w postaci zestawu następujących *ro* reguł

(w ogólności  $ro \neq r$  i poprzedniki też mogą być różne niż w regułach regulatora)

$R_{ob}^i$  : JEŚLI  $y(k)$  jest  $A_0^i$  i  $y(k-1)$  jest  $A_1^i$  i  $\dots$  i  $y(k-n_O)$  jest  $A_{n_O}^i$   
i  $u(k)$  jest  $B_0^i$  i  $u(k-1)$  jest  $B_1^i$  i  $\dots$  i  $u(k-m_O)$  jest  $B_{m_O}^i$

$$\text{TO } y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \dots + a_{n_A}^i y(k-n_A+1) \\ + b_0^i u(k) + b_1^i u(k-1) + \dots + b_{m_B}^i u(k-m_B), \quad (3.219)$$

gdzie  $i = 1, \dots, ro$ . Poziomy aktywacji reguł (3.219) oznaczamy przez  $w_o^i(k)$ , a ich unormowane wartości przez  $\tilde{w}_o^i(k)$ , stąd wyjście modelu obiektu dane jest zależnością

$$y(k+1) = \frac{\sum_{i=1}^{ro} w_o^i(k) y^i(k+1)}{\sum_{l=1}^{ro} w_o^l(k)} = \sum_{i=1}^{ro} \tilde{w}_o^i(k) y^i(k+1). \quad (3.220)$$

Zwróćmy uwagę na drobną różnicę w stosunku do ogólnej postaci modelu typu ARX stosowanego w niniejszym rozdziale (por. (3.84)), jaką jest zastosowanie w następnikach reguł (3.219) oznaczenia  $m_B$  w miejsce  $n_B$  – dla zgodności z notacją stosowaną w rozdziale 2. Na wyniki przedstawione w tym rozdziale będziemy się bowiem powoływać.

Warunkiem dostatecznym stabilności układu regulacji z regulatorem FDMC i modelem (3.219), (3.220) jest (zob. wniosek 2.6) istnienie takiej macierzy symetrycznej dodatnio określonej  $\mathbf{P}$ , że dla każdej macierzy  $\mathbf{A}_{ij}$  spełnione jest równanie  $\mathbf{A}_{ij}^T \mathbf{P} \mathbf{A}_{ij} - \mathbf{P} < \mathbf{0}$ , dla wszystkich par  $(i, j)$ ,  $i = 1, \dots, ro$ ,  $j = 1, \dots, r$  z wyjątkiem takich, dla których zawsze  $w_o^i(k)w^j(k) = 0$ . Macierze  $\mathbf{A}_{ij}$  dane są zależnościami (2.51), przy czym w rozważanym przypadku regulatora FDMC elementy  $(n+1)$ -szych wierszy tych macierzy wynikają bezpośrednio ze wzorów (3.218), zaś elementy pierwszych wierszy wyrażają się zależnościami

$$ac_1^{i,j} = a_1^i - b_0^i (k^e)^j, \\ ac_p^{i,j} = a_p^i, \quad p = 2, \dots, n_A, \\ bd_1^{i,j} = b_1^i + b_0^i [1 - (k_1^u)^j], \\ bd_q^{i,j} = b_q^i + b_0^i [(k_{q-1}^u)^j - (k_q^u)^j], \quad q = 2, \dots, D-1, \\ bd_D^{i,j} = b_D^i + b_0^i (k_{D-1}^u)^j, \quad (3.221)$$

gdzie  $n = n_A$  oraz przyjęto  $m = D$ , tzn.  $D \geq m_B$ , gdyż liczba elementów odpowiedzi skokowej jest nie mniejsza niż liczba wyrazów zależnych od sterowania w modelu ARX obiektu – przypadkiem granicznym jest właśnie

model oparty bezpośrednio na odpowiedzi skokowej. Jeśli  $m_B < D$ , to w powyższych wzorach należy ponadto przyjąć  $b_q^i = 0$  dla  $m_B < q \leq D$ .

W przypadku identycznych poprzedników reguł w modelach obiektu przyjętego do projektowania regulatora i do opisu obiektu w układzie regulacji mamy  $r = r_0$ ,  $w_o^i(k) = w^i(k)$ ,  $i = 1, \dots, r$ . Wówczas można zmniejszyć liczbę macierzy  $\mathbf{A}_{ij}$  oraz do analizy stabilności wykorzystać sformułowanie podane w postaci wniosku 2.7, zob. rozdz. 2.2.2.

W analogiczny sposób projektuje się *rozmyty analityczny regulator FGPC* – z lokalnymi liniowymi regulatorami GPC. Zamiast reguł (3.216) mamy jedynie reguły z następnikami odpowiadającymi zależności (3.78)

$R_{reg}^i$  : JEŚLI  $y(k)$  jest  $A_0^i$  i  $y(k-1)$  jest  $A_1^i$  i  $\dots$  i  $y(k-n_R)$  jest  $A_{n_R}^i$   
i  $u(k-1)$  jest  $B_1^i$  i  $\dots$  i  $u(k-m_R)$  jest  $B_{m_R}^i$

$$\text{TO } \Delta u^i(k) = (k^e)^i y^{zad}(k) - \sum_{q=0}^{n_{AR}} (k_q^y)^i y(k-q) - \sum_{q=1}^{m_{BR}} (k_q^u)^i \Delta u(k-q), \quad (3.222)$$

$i = 1, \dots, r$ , gdzie  $n_{AR}$  i  $m_{BR}$  odpowiadają parametrom  $n_A$  i  $n_B$  wielomianów  $A(z^{-1})$  i  $B(z^{-1})$  modelu obiektu (3.65) przyjętego do projektowania regulatora GPC w rozdz. 3.3.1. Wprowadziliśmy tu oznaczenia  $n_{AR}$  i  $m_{BR}$ , gdyż wartości  $n_{AR}$  i  $m_{BR}$  występujące w następnikach reguł (3.222) (tzn. w lokalnych modelach ARX obiektu użytych do projektowania lokalnych praw regulacji GPC) oraz wartości  $n_A$  i  $m_B$  w następnikach reguł (3.219) modelu reprezentującego obiekt w układzie regulacji mogą być w ogólności różne, w przypadku innych modeli obiektu użytych do projektowania regulatora i analizy układu regulacji.

Do analizy stabilności układu regulacji z regulatorem FGPC i modelem obiektu (3.219) również możemy wykorzystać twierdzenie 2.1, a dokładniej wniosek 2.6 przedstawiony w rozdz. 2.2.2. Zauważmy jednakże, że formuły lokalnych regulatorów GPC występujące w następnikach reguł (3.222) są nieco innej postaci niż formuły następników reguł (2.39) w rozdziale 2.2.2 – wyjścia regulatorów lokalnych zależą tam od uchybów regulacji, a nie bezpośrednio od wyjść obiektu jak w (3.222). Nie stanowi to jednak przeszkody w analizie stabilności, gdyż teza zarówno twierdzenia 2.1, jak i wniosku 2.6 nie zależy od wartości sygnału  $y^{zad}$ , stąd bez utraty ogólności możemy przyjąć  $y^{zad}(k) = y^{zad}(k-1) = \dots = 0$ . Wówczas prawa regulacji GPC są szczególnymi przypadkami praw regulacji z następników reguł (2.39), dla  $n_C = n_{AR} + 1$ ,  $m_D = m_{BR} + 1$  oraz

$$c_p^j = (k_{p-1}^y)^j, \quad p = 1, \dots, n_{AR} + 1, \quad (3.223)$$

$$d_1^j = 1 - (k_1^u)^j,$$

$$d_q^j = (k_{q-1}^u)^j - (k_q^u)^j, \quad q = 2, \dots, m_{BR},$$

$$d_{m_{BR}+1}^j = (k_{m_{BR}}^u)^j. \quad (3.224)$$

Przyjmując  $n = \max\{n_{AR} + 1, n_A\}$ ,  $m = \max\{m_{BR} + 1, m_B\}$ , uzyskujemy następujące wzory na elementy pierwszych wierszy macierzy  $\mathbf{A}_{ij}$  (2.51) występujących we wniosku 2.6:

$$\begin{aligned} ac_p^{i,j} &= a_p^i - b_0^i (k_{p-1}^y)^j, \quad p = 1, \dots, n, \\ bd_1^{i,j} &= b_1^i + b_0^i [1 - (k_1^u)^j], \\ bd_q^{i,j} &= b_q^i + b_0^i [(k_{q-1}^u)^j - (k_q^u)^j], \quad q = 2, \dots, m. \end{aligned} \quad (3.225)$$

Przy tym we wzorach powyższych:

- jeśli  $n_{AR} + 1 > n_A$ , to należy przyjąć  $a_p^i = 0$  dla  $p > n_A$ , zaś jeśli  $n_{AR} + 1 < n_A$  to należy przyjąć  $(k_{p-1}^y)^j = 0$  dla  $p > n_{AR} + 1$ , oraz analogicznie,
- jeśli  $m_{BR} + 1 > m_B$  to przyjmujemy  $b_q^i = 0$  dla  $q > m_B$ , zaś zawsze przyjmujemy  $(k_q^u)^j = 0$  dla  $q > m_{BR}$ .

Z wzorów (3.224) wynikają natomiast bezpośrednio zależności na elementy  $(n + 1)$ -szych wierszy macierzy  $\mathbf{A}_{ij}$ .

Projektowanie *rozmytego analitycznego regulatora FMPCS* (Fuzzy MPC opartego na modelu w postaci równań stanu) polega, w pierwszym etapie, na uzyskaniu  $r$  obszarów rozmytych zdefiniowanych poprzednikami reguł modelu rozmytego TS obiektu oraz w każdym z tych obszarów wyznaczeniu liniowego modelu obiektu w postaci równań stanu. Następnie, dla każdego z tych liniowych modeli projektujemy analityczne prawo regulacji MPC, w sposób omówiony w rozdz. 3.4. Oznaczając przez  $x_1(k), \dots, x_n(k)$  wartości zmiennych stanu w chwili  $k$  uzyskujemy regulator rozmyty FMPCS, który w przypadku *pomiaru stanu* można zapisać w postaci zbioru reguł z następnikami realizującymi prawo regulacji (3.114)

$R_{reg}^i$  : JEŚLI  $x_1(k)$  jest  $A_1^i$  i  $\dots$  i  $x_n(k)$  jest  $A_n^i$

$$\begin{aligned} \text{TO } \Delta u^i(k) &= (k^e)^i y^{zad}(k) + \\ &- \sum_{p=1}^N k_{1,p}^i \{ \mathbf{C}(\mathbf{A}_i)^p x(k) + \mathbf{C}\mathbf{V}_{pi} [\mathbf{B}_i u(k-1) + v(k)] \}, \end{aligned} \quad (3.226)$$

gdzie macierz  $\mathbf{V}_{pi}$  zdefiniowana jest przez (3.113) z  $\mathbf{A}_i$  zamiast  $\mathbf{A}$ . Reguły uzupełnia konkluzja finalna w standardowej postaci (3.217).

W następnikach reguł (3.226) występują mierzone wartości stanu nieliniowego obiektu rozmytego. Jeśli stosowany jest *estymator stanu obiektu* (nie rozszerzonego o stany zakłóceń deterministycznych), to w regułach lokalnych należy zastosować prawo regulacji (3.136), tzn. w następnikach reguł trzeba też uwzględnić odpowiednią poprawkę:

$$\begin{aligned} R_{reg}^i : \text{ JEŚLI } x_1(k) \text{ jest } A_1^i \text{ i } \dots \text{ i } x_n(k) \text{ jest } A_n^i \\ \text{ TO } \Delta u^i(k) = (k^e)^i y^{zad}(k) - \sum_{p=1}^N k_{1,p}^i \{ \mathbf{C}(\mathbf{A}_i)^p \hat{x}(k) + \\ + \mathbf{C}\mathbf{V}_{pi} [\mathbf{B}_i u(k-1) + v(k)] + [y(k) - \mathbf{C}\hat{x}(k)] \}. \end{aligned} \quad (3.227)$$

Przypomnijmy, że w przypadku rozmytego modelu obiektu z następnikami reguł w postaci lokalnych równań stanu, szczególnie łatwo jest realizować rozszerzony filtr Kalmana. Model nieliniowy można bowiem w tym przypadku przedstawić jako model liniowy ze zmiennymi elementami macierzy modelu, zgodnie z zależnościami (3.207)-(3.208) dla  $p = 0$ .

#### Przykład 3.14.

Rozważymy układ regulacji z regulatorem analitycznym FGPC i obiektem z przykładu 3.13. Regulator analityczny FGPC uzyskuje się stosując lokalne prawa regulacji liniowych regulatorów GPC łączone w każdej chwili wnioskowaniem rozmytym.

Przyjmując parametry ustalone w przykładzie 3.13, tzn.  $N = 6$ ,  $N_u = 3$ ,  $N = 1$  i  $\lambda = 0.1$  uzyskuje się dla obiektów lokalnych występujących w następnikach reguł (3.212) i (3.213) modelu obiektu prawa regulacji GPC, będące następnikami reguł regulatora FGPC

$$\begin{aligned} R_{reg}^1 : \text{ JEŚLI } y(k) \text{ jest } Y_1 \\ \text{ TO } u^1(k) = 1.0012y^{zad}(k) - 1.7904y(k) + 0.7892y(k-1), \end{aligned} \quad (3.228)$$

$$\begin{aligned} R_{reg}^2 : \text{ JEŚLI } y(k) \text{ jest } Y_2 \\ \text{ TO } u^2(k) = 2.1948y^{zad}(k) - 2.9854y(k) + 0.7906y(k-1). \end{aligned} \quad (3.229)$$

Przyjmując funkcje przynależności takie jak w przykładzie 3.13 i oznaczając przez  $w^1(k)$  i  $w^2(k)$  poziomy ich aktywacji (nie trzeba normować tych

wartości, gdyż z konstrukcji  $w^1(k) + w^2(k) = 1$ ), prawo regulacji FGPC można zapisać w postaci

$$\begin{aligned} u(k) &= w^1(k)u^1(k) + w^2(k)u^2(k) \\ &= [1.0012w^1(k) + 2.1948w^2(k)]y^{zad}(k) - [1.7904w^1(k) + \\ &+ 2.9854w^2(k)]y(k) + [0.7892w^1(k) + 0.7906w^2(k)]y(k-1). \end{aligned}$$

Przebiegi uzyskane w układzie regulacji z takim regulatorem są bardzo zbliżone do pokazanych na rysunku 3.72 w przykładzie 3.13 (regulator typu FGPC-NSL, sytuacja bez ingerencji ograniczeń), stąd ich nie przytaczamy.

Zastosujemy wniosek 2.7, z  $\mathbf{Q} = \mathbf{0}$ , do analizy stabilności układu regulacji z modelem obiektu danym regułami (3.212) i (3.213) oraz zaprojektowanym regulatorem FGPC. W tym celu należy wyznaczyć macierze  $\mathbf{A}_{ij}$  (2.51). W rozważanym przykładzie mamy  $n = 2$  i  $m = 1$ , stąd macierze  $A_{ij}$  są o wymiarze 3, postaci ogólnej

$$\mathbf{A}_{ij} = \begin{bmatrix} ac_1^{ij} & ac_2^{ij} & bd_1^{ij} \\ 1 & 0 & 0 \\ -c_1^j & -c_2^j & d_1^j \end{bmatrix}, \quad i, j = 1, 2,$$

o elementach danych wzorami (3.224) i (3.225), gdzie wartości (niezerowe) parametrów lokalnych modeli obiektów i regulatorów wynoszą

$$\begin{aligned} a_1^1 &= 0.7, \quad b_0^1 = 0.8, \quad (k_1^y)^1 = 1.7904, \quad (k_2^y)^1 = -0.7892, \\ a_1^2 &= 0.3, \quad b_0^2 = 0.2, \quad (k_1^y)^2 = 2.9854, \quad (k_2^y)^2 = -0.7906. \end{aligned}$$

Zgodnie z tezą wniosku 2.7 wystarczy wziąć pod uwagę macierze  $\mathbf{A}_{11}$ ,  $\mathbf{A}_{22}$  i  $\overline{\mathbf{A}}_{12} = (\mathbf{A}_{12} + \mathbf{A}_{21})/2$ , które w analizowanym przypadku przyjmują wartości

$$\begin{aligned} \mathbf{A}_{11} &= \begin{bmatrix} -0.7496 & 0.6361 & 0.8000 \\ 1 & 0 & 0 \\ -1.8120 & 0.7951 & 1 \end{bmatrix}, \\ \mathbf{A}_{22} &= \begin{bmatrix} -0.3169 & 0.1626 & 0.2000 \\ 1 & 0 & 0 \\ -3.0846 & 0.8131 & 1 \end{bmatrix}, \\ \overline{\mathbf{A}}_{12} &= \begin{bmatrix} -0.9150 & 0.4048 & 0.5000 \\ 1 & 0 & 0 \\ -2.4483 & 0.8041 & 1 \end{bmatrix}. \end{aligned}$$

Korzystając z *LMI Toolbox* do pakietu MATLAB sprawdzono, że układ nierówności macierzowych

$$\begin{aligned}(\mathbf{A}_{11})^T \mathbf{P} \mathbf{A}_{11} - \mathbf{P} &< 0, \\ (\mathbf{A}_{22})^T \mathbf{P} \mathbf{A}_{22} - \mathbf{P} &< 0, \\ (\overline{\mathbf{A}}_{12})^T \mathbf{P} \overline{\mathbf{A}}_{12} - \mathbf{P} &< 0, \\ -\mathbf{P} &< 0,\end{aligned}$$

posiada rozwiązanie w postaci następującej symetrycznej macierzy  $\mathbf{P}$

$$\mathbf{P} = \begin{bmatrix} 12.1556 & -5.8906 & -5.6196 \\ -5.8906 & 4.2805 & 3.4626 \\ -5.6196 & 3.4626 & 3.4601 \end{bmatrix}.$$

Na mocy wniosku 2.7 spełnione są warunki dostateczne stabilności rozważanego nieliniowego układu regulacji.  $\square$

W pracach [96, 97, 92] znaleźć można przykłady projektowania i analizy stabilności regulatorów analitycznych FDMC, w szczególności również dla obiektu rozważonego w ostatnim przykładzie, dla układu regulacji stężenia zanieczyszczenia w kolumnie etylenowej i wielowymiarowej regulacji stacji wyparki.

### 3.6. Stabilność, ograniczenia, dobór parametrów

#### 3.6.1. Stabilność algorytmów MPC

Jest zjawiskiem frapującym, że algorytmy regulacji predykcyjnej ze skończonym, przesuwającym horyzontem i ograniczeniami były przez szereg lat z wielkim powodzeniem stosowane w praktyce przemysłowej, zanim pojawiły się prace teoretyczne dotyczące analizy stabilności. Co więcej, fakt ten nie był przeszkodą w rozwoju, w powstaniu produktów komercyjnych o dużych walorach użytkowych i popularności praktycznej. Podejściem wystarczającym okazała się wszechstronna, komputerowa analiza symulacyjna i doświadczenia gromadzone w kolejnych zastosowaniach. Dzisiaj wiemy już nieporównanie więcej o warunkach stabilności algorytmów MPC, a przede wszystkim o gwarantujących stabilność sposobach ich formułowania czy modyfikowania. Wiedza ta jednakże wykorzystywana jest nie tylko do projektowania algorytmów o gwarantowanej nominalnej stabilności, ale przede wszystkim pozwala lepiej rozumieć mechanizm działania regulacji

predykcyjnej, dostarcza wskazówek dla lepiej uzasadnionego doboru elementów struktury i wartości parametrów. Wobec złożoności algorytmów MPC, szczególnie w sytuacji wielu ograniczeń na sterowania i wyjścia, problem stabilności jest ważnym, ale nie jedynym. Nie mniej istotne jest zapewnienie niepustości zbioru dopuszczalnych rozwiązań w każdej chwili próbkowania. Podejścia heurystyczne i komputerowa analiza symulacyjna nadal odgrywają tu podstawową rolę.

W niniejszej części rozdziału poświęconego regulacji predykcyjnej przedstawimy wybrane, podstawowe rezultaty dotyczące stabilności algorytmów MPC, przede wszystkim algorytmów z modelami liniowymi. Prezentacja nie pretenduje do wyczerpującej ani rygorystycznej matematycznie – wykraczałoby to znacznie poza założenia i ramy niniejszej pracy. Czytelnikowi zainteresowanemu obszerniejszymi studiami polecamy znakomity artykuł przeglądowy Mayne'a i in. [98] oraz prace wymienione w zawartej tam wyczerpującej bibliografii tematu, zob. też [102, 1, 87]. Celem rozważań niniejszego rozdziału jest przede wszystkim pokazanie podstawowych mechanizmów warunkujących stabilność i wynikających stąd wskazówek wyjaśniających zarówno kierunki modyfikacji struktur, jak i pokazujących wytyczne doboru wartości parametrów regulatorów predykcyjnych.

W rozdziale niniejszym poruszano już zagadnienie stabilności – w przypadkach analitycznych sformułowań algorytmów w postaci określonych praw regulacji, możliwych do uzyskania, kiedy nie uwzględnia się ograniczeń sterowań i wyjść. Dla regulatorów z liniowymi modelami obiektu, a więc algorytmów DMC, GPC czy MPC, prawo regulacji jest liniowym sprzężeniem od wyjść i wejść czy stanu, zob. rozdz. 3.2, 3.3 i 3.4. Stabilność układu regulacji z takim regulatorem może być analizowana znanymi w liniowej teorii regulacji metodami. Wykorzystując specyficzną strukturę nieliniowych modeli rozmytych typu TS (Takagi-Sugeno) udało się również wyprowadzić analityczny sposób formułowania warunków dostatecznych stabilności nieliniowych regulatorów predykcyjnych z takimi modelami, oczywiście też w przypadku projektowania bez ograniczeń, zob. rozdz. 3.5.4. Podkreślimy, że rozważenie sytuacji bez ograniczeń stanowi zalecany etap projektowania regulatora predykcyjnego również przy jego przewidywanej pracy w obecności ograniczeń. Chociażby dlatego, że punkty równowagi mogą leżeć we wnętrzu zbioru dopuszczalnego, że znacznie lepiej rozumiemy problematykę stabilności pętli regulacji w sytuacji bez ograniczeń. Natomiast w przypadku ograniczeń dotyczących jedynie sterowań regulator analityczny zaimplementowany w odpowiedniej strukturze typu *anti-windup* może dobrze funkcjonować, szczególnie w sytuacji regulacji jednowymiarowej, zob. np. rozdz. 3.2. Nie zmienia to oczywiście faktu, że

głównym problemem i wyzwaniem jest zagadnienie stabilności algorytmów MPC w wersji numerycznej, z rozwiązywanym w każdej chwili próbkowania zadaniem optymalizacji z ograniczeniami.

### Podstawowe mechanizmy zapewniania stabilności

Dwa elementy w sformułowaniu algorytmu MPC mają kluczowe znaczenie dla stabilności układu regulacji:

- *ograniczenie stanu końcowego*, tzn. ograniczenie stanu w chwili końcowej  $k + N$  horyzontu predykcji,
- *długość horyzontu predykcji*  $N$ .

Wymuszenie określonej wartości stanu w chwili końcowej horyzontu predykcji wystarcza do wykazania stabilności nawet w ogólnym przypadku nieliniowym – ale przy założeniu realizowalności, czyli niepustości zbioru dopuszczalnych sterowań w każdej chwili działania algorytmu predykcyjnego. Uzasadnienie jest stosunkowo proste, jeśli zastosować rozumowanie oparte na wykorzystaniu funkcji Lapunowa, jak to pokazano w oryginalnych pracach Chena i Shaw [23] dla układu regulacji z czasem ciągłym, zob. też [110], czy przez Keerthi i Gilbera [63] dla układu z czasem dyskretnym. Rozważmy bowiem obiekt opisany nieliniowym równaniem różnicowym

$$x(k+1) = f(x(k), u(k)), \quad (3.230)$$

zakładając, że stan  $(x, u) = (0, 0)$  jest jego punktem równowagi. Przyjmując dla uproszczenia równość horyzontów sterowania i predykcji,  $N_u = N$ , oznaczmy przez  $x(k+p|k)$  wyznaczony z wykorzystaniem powyższego modelu stan przewidywany w chwili  $k$  na chwilę  $k+p$  przy wartościach sterowania  $u(k+p-1|k)$ ,  $p = 1, \dots, N$ . Następnie, rozważmy funkcję kryterialną postaci

$$J(k) = \sum_{p=1}^N h(x(k+p|k), u(k+p-1|k)), \quad (3.231)$$

gdzie  $h(x, u) \geq 0$  oraz  $h(x, u) = 0$  wtedy i tylko wtedy, gdy  $x = 0$  i  $u = 0$ . Ponadto uwzględnimy ograniczenia

$$u(k+p|k) \in U, \quad p = 0, 1, \dots, N-1, \quad (3.232a)$$

$$x(k+p|k) \in X, \quad p = 1, 2, \dots, N-1, \quad (3.232b)$$

$$x(k+N|k) = 0. \quad (3.232c)$$

Ostatnie ze sformułowanych ograniczeń to wspomniane *ograniczenie stanu końcowego*. Do wyznaczania kolejnych sterowań zastosujemy ogólną za-

sadę regulacji predykcyjnej, tzn. w każdej chwili próbkowania optymalizowana będzie funkcja kryterialna przy wszystkich wymienionych ograniczeniach, zaś do sterowania obiektem brany będzie jedynie pierwszy element  $\hat{u}(k) = \hat{u}(k|k)$  każdorazowo wyznaczanego wektora optymalnych sterowań  $[\hat{u}(k|k) \cdots \hat{u}(k+N-1|k)]^T$ . Oznaczmy przez  $\hat{J}(k)$  optymalną wartość funkcji kryterialnej w chwili  $k$  przy sformułowanych powyżej ograniczeniach, podobnie  $\hat{J}(k+1)$  dla chwili  $k+1$ , zaś przez  $\hat{x}(k+p|k)$ ,  $p = 1, 2, \dots, N$  oznaczmy odpowiadające  $\hat{J}(k)$  optymalne wartości stanu wyznaczone w chwili  $k$ . Możemy wówczas napisać

$$\begin{aligned} \hat{J}(k+1) &= \min_{u(k+1), \dots, u(k+N)} \sum_{p=1}^N h(x(k+1+p|k+1), u(k+p)) \\ &= \min_{u(k+1), \dots, u(k+N)} \left\{ \sum_{p=1}^N h(x(k+p|k+1), u(k+p-1)) + \right. \\ &\quad \left. - h(x(k+1|k+1), u(k)) + h(x(k+1+N|k+1), u(k+N)) \right\} \\ &\leq \hat{J}(k) - h(\hat{x}(k+1|k), \hat{u}(k|k)) + \\ &\quad + \min_{u(k+N)} h(x(k+1+N|k+1), u(k+N)), \quad (3.233) \end{aligned}$$

ponieważ wektor

$$[\hat{u}(k+1|k) \cdots \hat{u}(k+N-1|k) \ u^*(k+N|k+1)]^T,$$

gdzie  $N-1$  pierwszych elementów to składowe wektora optymalnego z poprzedniej chwili  $k$ , zaś  $u^*(k+N|k+1)$  to wartość minimalizująca ostatni składnik nierówności (3.233), nie może być lepszy w chwili  $k+1$  od wektora optymalnego dla tej chwili. Ponieważ po optymalizacji w chwili  $k$  spełnione jest ograniczenie  $x(k+N|k) = 0$  i punkt  $(0, 0)$  jest z założenia punktem równowagi, to  $u^*(k+N|k+1) = 0$ , gdyż wówczas  $x(k+1+N|k+1) = 0$  i ostatni składnik po prawej stronie nierówności (3.233) przyjmuje najmniejszą możliwą wartość równą zero. Stąd dostajemy

$$\hat{J}(k+1) - \hat{J}(k) \leq -h(\hat{x}(k+1|k), \hat{u}(k|k)) \leq 0.$$

Jeśli teraz przyjąć funkcję  $\hat{J}(k)$  jako funkcję Lapunowa, to z twierdzenia Lapunowa wynika stabilność układu regulacji z rozważonym regulatorem predykcyjnym.

Możliwość uzyskania przytoczonego, bardzo silnego rezultatu wynika przede wszystkim z założenia niepustości zbioru dopuszczalnego określonego zestawem ograniczeń (3.232), niepustości w każdej chwili próbkowania. Ponadto, zakłada się milcząco osiągnięcie w każdej chwili globalnego minimum

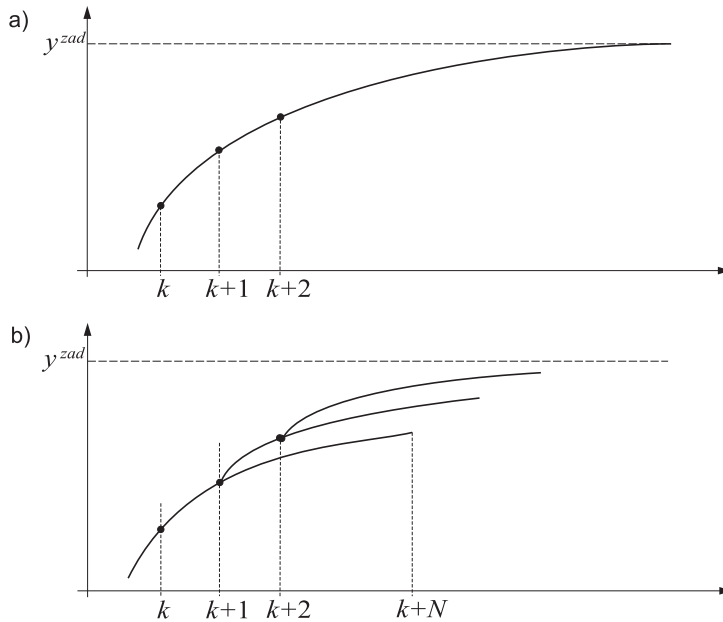
funkcji  $J(k)$  – co jest dla modeli liniowych i typowych kwadratowych funkcji kryterialnych założeniem naturalnym i zawsze spełnionym, natomiast dla modeli nieliniowych trudnym w ogólności do zagwarantowania. Niewątpliwie kluczowe jest jednak założenie o niepustości zbioru dopuszczalnego, w ogólności silnie ograniczonego dodatkowym wymaganiem  $x(k+N|k) = 0$ .

Wykorzystując dla uzyskania stabilności pomysł narzucania stanu końcowego Clarke i Scattolini zaproponowali wersję analitycznego algorytmu GPC z dodanym tego typu ograniczeniem, nazwanego algorytmem CRHPC (*Constrained Receding-Horizon Predictive Control*) [27]. W pracy [129] sformułowanie rozszerzono na przypadek algorytmu numerycznego, z ograniczeniami nierównościowymi. Sformułowanie zadania optymalizacji algorytmu CRHPC jest rozszerzeniem zadania dla GPC (zob. rozdz. 3.3) poprzez narzucenie dodatkowych ograniczeń

$$y(k+N+j|k) = y^{zad}(k+N), \quad j = 1, \dots, m. \quad (3.234)$$

W [129] wykazano stabilność algorytmu CRHPC dla  $N_u = N - \tau + 1$  i  $N_u \geq m$ , przy  $m = \max\{n_A, n_B - \tau + 1\} + 1$ , gdzie  $n_A$  i  $n_B$  to stopnie wielomianów  $A(z^{-1})$  i  $B(z^{-1})$  w modelu obiektu wykorzystywanym w algorytmie GPC, zob. rozdz. 3.3, zaś  $\tau$  to opóźnienie w torze sterowania obiektu. Oczywiście, trzeba również przyjmować jako założenie niepustość zbioru dopuszczalnego – tzn. realizowalność ograniczeń nierównościowych łącznie z ograniczeniami równościowymi (3.234). Zwróćmy uwagę, że predykcję wartości wyjść wykonuje się w istocie na horyzoncie o długości  $N + m$ . Ograniczenia trzeba bowiem uwzględniać do chwili  $N + m$ , mimo że w funkcji kryterialnej jest tylko sumowanie do chwili  $N$ . Dodatkowe ograniczenie równościowe narzucane jest nie w jednej, ale w  $m$  kolejnych chwilach po to, aby zapewnić wystabilizowanie *całego stanu* obiektu. I wreszcie, stosuje się długi, równy  $N + 1$ , horyzont sterowania, co minimalizuje niebezpieczeństwo nierealizowalności ograniczeń (3.234) przy danej wartości  $N$ . Autorzy zalecają algorytm CRHPC dla obiektów trudnych, gdy zastosowanie GPC sprawia kłopoty ze stabilizacją obiektu.

Rozważmy teraz standardowy algorytm GPC, ale przy horyzoncie sterowania  $N_u = N + 1$ , takim jak w CRHPC. Zauważmy, że jeśli oddziaływanie współczynników funkcji kryterialnej wyrażone macierzami  $\Psi(p)$  i  $\Lambda(p)$  nie maleje ze wzrostem  $p$  (np.  $\Psi(p) = \psi \mathbf{I}$  i  $\Lambda(p) = \lambda \mathbf{I}$ , nie zależą od  $p$ ) oraz długość horyzontu predykcji  $N$  jest dostatecznie duża, to na końcowym odcinku tego horyzontu wyjścia powinny stabilizować się na wartościach zadanych, tzn. z dostateczną dokładnością powinny zachodzić  $\hat{y}(k+N-j|k) = y^{zad}$ ,  $\Delta \hat{u}(k+N-j) = 0$ ,  $j = 0, \dots, m-1$  (oczywiście przy założeniu osiągalności, dopuszczalności wartości  $y^{zad}$ ). Osiągamy więc dokładnie sytuację wymuszaną w algorytmie CRHPC dodatkowym



Rysunek 3.76. Przykład trajektorii obiektu realizowanych przy regulacji predykcyjnej (a) z nieskończonym horyzontem predykcji i sterowania i (b) ze skończonym horyzontem równym  $N$  (przy założeniu dokładnego modelu)

ograniczeniem (3.234), bez narzucenia tego ograniczenia, a jedynie wydłużając horyzont. Można stąd wyciągnąć wniosek, że *wydłużanie horyzontu ma własności stabilizujące*.

Dla nieskończonego horyzontu predykcji i sterowania ( $N = N_u = \infty$ ) stabilność układu regulacji wynika bezpośrednio z zasady działania regulatora predykcyjnego. Dla dowodu wystarczy wziąć pod uwagę zasadę optymalności Bellmana (w największym skrócie, dla addytywnych funkcji kryterialnych: każdy odcinek końcowy trajektorii optymalnej jest też, počynając od swojego punktu początkowego, trajektorią optymalną). Z zasady tej wynika, że jeśli tylko nie zmieniają się warunki zewnętrzne (trajektoria wartości zadanej), to układ z regulatorem predykcyjnym będzie się poruszał dokładnie po trajektorii optymalnej  $[\hat{u}(0|0) \hat{u}(1|0) \hat{u}(2|0) \dots]^T$  wyznaczonej w zerowej chwili początkowej, tzn.  $\hat{u}(k|k) = \hat{u}(k|0)$ ,  $k = 0, 1, 2, \dots$ , a więc będzie stabilny. Sytuację tę zilustrowano na rysunku 3.76 a). Dla porównania, na rysunku 3.76 b) przedstawiono przykładowe trajektorie występujące przy skończonym horyzontie predykcji.

W praktycznych aplikacjach algorytmów predykcyjnych długość horyzontów predykcji i sterowania, a szczególnie długość horyzontu sterowania, silnie wpływają na nakład i czas obliczeń potrzebny do wyliczenia sterowa-

nia w każdej z chwil próbkowania. Im dłuższy horyzont, tym dłuższy czas obliczeń. Czas wyliczenia sterowania może być krytyczny w zastosowaniach praktycznych, stąd uzyskiwanie gwarantowanej stabilności przez znaczne, z odpowiednim zapasem bezpieczeństwa, wydłużanie horyzontów jest nie do zaakceptowania jako ogólna metoda postępowania. Z drugiej strony, właśnie dla krótszych horyzontów zastosowanie alternatywnego podejścia przez narzucanie stanu końcowego zbyt łatwo grozi wystąpieniem zjawiska nieosiągalności założonych stanów końcowych, czyli pustością zbioru dopuszczalnego – i też jest nie do zaakceptowania jako zbyt restrykcyjna ogólna metoda postępowania. Na szczęście łatwo wykazać, że kluczowy dla stabilności jest nieskończony horyzont *predykcyj*, nie sterowania.

Rozważamy zagadnienie regulacji stabilnego obiektu nieliniowego (3.230) do zerowego punktu równowagi, przyjmując równanie wyjść  $y(k) = g(x(k))$ . Funkcję kryterialną definiujemy na nieskończonym horyzoncie, przy założeniu niezależności macierzy współczynników wagowych od czasu,  $\Psi(p) = \Psi$  i  $\Lambda(p) = \Lambda$  oraz uwzględniając oprócz przyrostów sterowań również ich wartości [106]

$$J(k) = \sum_{p=1}^{\infty} \left( \|y(k+p|k)\|_{\Psi}^2 + \|\Delta u(k+p-1|k)\|_{\Lambda}^2 + \|u(k+p-1|k)\|_{\mathbf{R}}^2 \right), \quad (3.235)$$

przy czym macierze  $\Psi$  i  $\Lambda$  są dodatnio półokreślone, macierz  $\mathbf{R}$  dodatnio określona, zaś przez  $y(k+p|k)$ ,  $u(k+p|k)$  oznaczono predykcje wyjść obiektu i sterowania wyznaczane na chwilę  $k+p$  w chwili  $k$ . Załóżmy ponadto *skończony horyzont sterowania* o długości  $N_u \geq 1$ ,

$$\Delta u(k+p|k) = 0 \quad \text{dla } p \geq N_u, \quad k = 0, 1, \dots \quad (3.236)$$

Oznaczmy, podobnie jak poprzednio, przez  $\hat{J}(k)$  optymalną wartość funkcji kryterialnej uzyskiwaną przez jej optymalizację względem wektora  $[\Delta u(k|k) \Delta u(k+1|k) \dots \Delta u(k+N_u-1|k)]^T$  przy dodatkowych ograniczeniach stanu czy sterowania, zakładając niepustość zbioru dopuszczalnego. Mamy wówczas

$$\hat{J}(k) = \sum_{p=1}^{\infty} \|\hat{y}(k+p|k)\|_{\Psi}^2 + \sum_{p=0}^{N_u-1} \left( \|\Delta \hat{u}(k+p|k)\|_{\Lambda}^2 + \|\hat{u}(k+p|k)\|_{\mathbf{R}}^2 \right), \quad (3.237)$$

gdyż musi zachodzić  $\hat{u}(k+N_u-1|k) = 0$  — w przeciwnym razie wartość  $J(k)$  byłaby nieskończona. We wzorze powyższym oznaczono przez  $\hat{u}(k+p-1|k)$ ,  $p = 1, \dots, N_u$  optymalne wartości sterowań wyznaczonych w chwili  $k$ , a przez  $\hat{y}(k+p|k)$ ,  $p = 1, 2, \dots$  odpowiadające im wartości wyjść. Traktując,

podobnie jak poprzednio w przypadku dodatkowego ograniczenia na stan końcowy,  $\hat{J}(k)$  jako kandydatkę na funkcję Lapunowa, możemy napisać

$$\begin{aligned} \hat{J}(k+1) &= \sum_{p=1}^{\infty} \|\hat{y}(k+1+p|k+1)\|_{\Psi}^2 + \\ &\quad + \sum_{p=0}^{N_u-1} \left( \|\Delta\hat{u}(k+1+p|k+1)\|_{\Lambda}^2 + \|\hat{u}(k+1+p|k+1)\|_{\mathbf{R}}^2 \right) \\ &\leq \hat{J}(k) - \|\hat{y}(k+1|k)\|_{\Psi}^2 - \|\Delta\hat{u}(k|k)\|_{\Lambda}^2 - \|\hat{u}(k|k)\|_{\mathbf{R}}^2, \end{aligned}$$

ponieważ sterowania optymalne wyznaczone w chwili próbkowania  $k$  na przyszłe chwile nie mogą w następnej chwili próbkowania,  $(k+1)$ -szej, dać rezultatu lepszego niż sterowania optymalne dla tej chwili. Dostajemy stąd  $\hat{J}(k+1) - \hat{J}(k) \leq 0$ , a ponieważ ciąg wartości  $\hat{J}(k)$  jest ograniczony od dołu, to jest zbieżny. Bezpośrednio stąd wynika

$$\|\hat{y}(k+1|k)\|_{\Psi}^2 + \|\Delta\hat{u}(k|k)\|_{\Lambda}^2 + \|\hat{u}(k|k)\|_{\mathbf{R}}^2 \rightarrow 0, \quad (3.238)$$

a dodatnia określoność macierzy  $\mathbf{R}$  powoduje, że  $\hat{u}(k|k) = u(k) \rightarrow 0$ . Założenie stabilności obiektu implikuje wówczas  $x(k) \rightarrow 0$ , czyli *stabilność regulatora predykcyjnego*. Przypomnijmy jeszcze, że cały czas zakładamy niepustość zbioru dopuszczalnego i znajdowanie minimum globalnego zadania optymalizacji w każdej chwili próbkowania.

Udowodniony rezultat sugeruje, że wydłużanie horyzontu predykcji przy dowolnym, skończonym horyzoncie sterowania stabilizuje działanie układu regulacji predykcyjnej. Chcąc stabilność zagwarantować, trzeba rozwiązać problem praktyczny efektywnego liczenia wartości funkcji kryterialnej na nieskończonym horyzoncie. Skuteczne rozwiązanie tego problemu w klasie obiektów liniowych zaproponowano w pracach [122, 106]. Przytoczymy poniżej ten rezultat.

### Regulacja z modelem liniowym i nieskończonym horyzontem predykcji

Punktem wyjścia jest opis obiektu układem liniowych, stabilnych równań stanu

$$\begin{aligned} x(k+1) &= \mathbf{A}x(k) + \mathbf{B}u(k), \\ y(k) &= \mathbf{C}x(k), \end{aligned}$$

oraz sformułowanie problemu jako zadania regulacji do zerowego punktu równowagi.

Zanim przejdziemy do problemu liczenia funkcji kryterialnej na nieskończonym horyzoncie zauważmy, że w typowych sformułowaniach regulatorów predykcyjnych z liniowymi modelami, jak w przypadku DMC czy GPC, nie występują w funkcji kryterialnej składniki związane z kwadratami wartości sterowań, tzn. mamy  $\mathbf{R} = \mathbf{0}$ . Przedstawione w punkcie poprzednim rozumowanie jest jednakże nadal słuszne również przy  $\mathbf{R} = \mathbf{0}$ , jeśli tylko  $\Psi > \mathbf{0}$  oraz  $y(k) \rightarrow 0$  implikuje  $x(k) \rightarrow 0$ , czyli nie występują mody obiektu nieobserwowalne poprzez wyjścia regulowane (badamy stabilność nominalną, czyli z dokładnym modelem i bez zakłóceń, wówczas przy  $\Psi > \mathbf{0}$  z (3.238) wynika  $\hat{y}(k+1|k) = y(k+1) \rightarrow 0$ ). Dodajmy, że w algorytmach wykorzystujących jedynie opis obiektu typu wejście-wyjście, jak DMC i GPC, rozważany model obiektu reprezentuje jedynie jego część sterowalną i obserwowalną i stabilność jest rozważana jedynie w sensie stabilnych zachowań wyjść obiektu, tzn. modów sterowalnych i obserwowalnych.

Dla podania efektywnego sposobu obliczania pierwszej (nieskończonej) sumy w wyrażeniu (3.235) rozbijmy tę sumę na dwa składniki

$$\sum_{p=1}^{\infty} \|y(k+p|k)\|_{\Psi}^2 = \sum_{p=1}^{N_u-1} \|y(k+p|k)\|_{\Psi}^2 + \sum_{p=N_u}^{\infty} \|y(k+p|k)\|_{\Psi}^2.$$

Wówczas drugą z sum można przedstawić w postaci

$$\begin{aligned} \sum_{p=N_u}^{\infty} \|y(k+p|k)\|_{\Psi}^2 &= \sum_{p=N_u}^{\infty} x(k+p|k)^T \mathbf{C}^T \Psi \mathbf{C} x(k+p|k) \\ &= x(k+N_u|k)^T \left[ \sum_{i=0}^{\infty} (\mathbf{A}^T)^i \mathbf{C}^T \Psi \mathbf{C} \mathbf{A}^i \right] x(k+N_u|k), \end{aligned}$$

gdyż, jak to już uzasadniono wyżej, musi zachodzić  $\hat{u}(k+p-1|k) = 0$  dla  $p \geq N_u$ , skąd

$$\begin{aligned} y(k+N_u|k) &= \mathbf{C}x(k+N_u|k), \\ y(k+N_u+1|k) &= \mathbf{C}\mathbf{A}x(k+N_u|k), \\ y(k+N_u+2|k) &= \mathbf{C}\mathbf{A}^2x(k+N_u|k), \quad \text{itd.} \end{aligned} \quad (3.239)$$

Macierz  $\Psi^{\infty}$ ,

$$\Psi^{\infty} = \sum_{i=0}^{\infty} (\mathbf{A}^T)^i \mathbf{C}^T \Psi \mathbf{C} \mathbf{A}^i,$$

istnieje dla macierzy  $\mathbf{A}$  obiektu stabilnego i może być wyznaczona z rozwiązania dyskretnego równania Lapunowa

$$\mathbf{A}^T \Psi^{\infty} \mathbf{A} + \mathbf{C}^T \Psi \mathbf{C} = \Psi^{\infty},$$

gdź

$$\Psi^\infty = \mathbf{C}^T \Psi \mathbf{C} + \sum_{i=1}^{\infty} (\mathbf{A}^T)^i \mathbf{C}^T \Psi \mathbf{C} \mathbf{A}^i = \mathbf{C}^T \Psi \mathbf{C} + \mathbf{A}^T \Psi^\infty \mathbf{A}.$$

Reasumując, zadanie optymalizacji regulatora predykcyjnego z nieskończonym horyzontem predykcji można przedstawić w następującej postaci zadania ze skończonym horyzontem i definiowanym za pomocą macierzy  $\Psi^\infty$  składnikiem kary za stan w chwili  $k + N_u|k$

$$\begin{aligned} J(k) = & x(k + N_u|k)^T \Psi^\infty x(k + N_u|k) + \sum_{p=1}^{N_u-1} \|y(k + p|k)\|_{\Psi}^2 + \\ & + \sum_{p=0}^{N_u-1} \left( \|\Delta u(k + p|k)\|_{\Lambda}^2 + \|u(k + p|k)\|_{\mathbf{R}}^2 \right). \end{aligned}$$

Jeśli nie ma ograniczeń stanu (implikowanych np. ograniczeniami wyjść), które mają być spełnione na nieskończonym horyzoncie, to oczywiście należy uznać  $N_u$  za horyzont predykcji  $N$ , nie ma potrzeby wyznaczać wartości stanu  $x(k + p|k)$  dla  $p > N_u$ . Jeśli natomiast takie ograniczenia istnieją, to w sformułowaniu zadania optymalizacji należy je uwzględnić aż do chwili  $N \geq N_u$ , gdzie  $N$  jest najmniejszą liczbą naturalną taką, że spełnienie ograniczeń na horyzoncie predykcji do chwili  $k + N$  zapewnia ich spełnienie dla  $k + p > k + N$  – w pracy [122] pokazano, że taka skończona liczba istnieje. W przedstawionej sytuacji łatwo może się zdarzyć przypadek  $N > N_u$ , szczególnie przy mniejszych wartościach  $N_u$ . Oczywiście, efektywnym horyzontem predykcji wymagającym wyliczenia wyjść przewidywanych jest wówczas  $N$ .

W praktyce wartości zadane wyjść są z reguły niezerowe, odpowiadając niezerowym punktom równowagi. Trzeba wówczas stosować funkcję kryterialną w postaci

$$\begin{aligned} J(k) = & [x(k + N_u|k) - x_s]^T \Psi^\infty [x(k + N_u|k) - x_s] + \\ & + \sum_{p=1}^{N_u-1} \left\| y(k + p|k) - y^{zad} \right\|_{\Psi}^2 + \\ & + \sum_{p=0}^{N_u-1} \left( \|\Delta u(k + p|k)\|_{\Lambda}^2 + \|u(k + p|k) - u_s\|_{\mathbf{R}}^2 \right), \quad (3.240) \end{aligned}$$

gdzie  $u_s$  jest sterowaniem, a  $x_s$  stanem odpowiadającym w stanie ustalonym wartości zadanej  $y^{zad}$  [106]. Wartości te należy wyliczyć z równań statyki obiektu,  $x_s = \mathbf{A}x_s + \mathbf{B}u_s$ ,  $y^{zad} = \mathbf{C}x_s$ .

Do tej pory rozważane były jedynie obiekty stabilne. Dla *obektu niestabilnego* opisanego równaniami stanu należy dokonać dekompozycji Jordana macierzy  $\mathbf{A}$  modelu obiektu na części stabilną i niestabilną, dodać do sformułowania zadania optymalizacji regulatora predykcyjnego ograniczenie równościowe wymuszające wyzerowanie niestabilnych modów obiektu w chwili  $(k + N|k)$  oraz liczyć sumę nieskończoną tylko dla modów stabilnych, analogicznie jak dla obiektu stabilnego (mody niestabilne pozostają zerowe dla  $k + p > k + N$ ). Zainteresowanym szczegółami czytelnika odsyłamy do prac [122, 106, 87].

Sformułowanie (3.240) jest ogólnym zapewniającym stabilność nominalną przez wprowadzenie składnika kary odpowiadającego pominiętej części nieskończonego horyzontu predykcji, dla  $\mathbf{R} = \mathbf{0}$  uzyskujemy przypadek odpowiadający najczęściej stosowanej, standardowej dla regulatorów MPC postaci funkcji kryterialnej. Wspomniany składnik kary za stan końcowy zdefiniowany jest poprzez zmienne stanu obiektu, wyliczony został z wykorzystaniem modelu obiektu w postaci równań stanu, zob. (3.239). Dlatego też chcąc bezpośrednio zastosować przedstawione podejście w algorytmach MPC operujących na modelach typu wejście-wyjście (DMC, GPC), trzeba dla potrzeb wyliczania składnika kary przejść na opis zmiennymi stanu lub opis równoważny.

Rozważając algorytm DMC zauważmy, że przy skończonej długości  $D$  odpowiedzi skokowych predykowane wartości wyjść stabilizują się po  $N_u + D - 1$  chwilach, z (3.22), (3.21) wynika bowiem

$$y^0(k + p|k) = y(k) + \sum_{j=1}^{D-1} (\mathbf{S}_D - \mathbf{S}_j) \Delta u(k - j) = y^{0\infty}(k), \quad p \geq D - 1,$$

$$\Delta y(k + p|k) = \sum_{j=1}^{N_u} \mathbf{S}_D \Delta u(k + N_u - j|k) = \Delta y^\infty(k), \quad p \geq N_u + D - 1.$$

Stąd, oznaczając  $y^\infty(k) = y^{0\infty}(k) + \Delta y^\infty(k)$ , mamy

$$J(\infty) = \sum_{p=N_1}^{N_u+D-2} \left\| [y^{zad}(k + p|k) - y^0(k + p|k)] - \Delta y(k + p|k) \right\|_{\Psi}^2 +$$

$$+ \lim_{p \rightarrow \infty} [p - (N_u + D - 2)] \left\| [y^{zad\infty}(k) - y^\infty(k)] \right\|_{\Psi}^2 +$$

$$+ \sum_{p=0}^{N_u-1} \left\| \Delta u(k + p|k) \right\|_{\Lambda}^2,$$

gdzie przyjęliśmy

$$y^{zad}(k + p|k) = const. = y^{zad\infty}(k), \quad p \geq N_u + D - 1.$$

Z uzyskanej zależności wynika, że sterowanie minimalizujące funkcję kryterialną  $J(\infty)$  musi zapewniać równość  $y^\infty(k) = y^{zad\infty}(k)$ , o ile tylko takie dopuszczalne sterowanie istnieje i  $\Psi > 0$ . Stąd *równoważnym* sformulowaniem algorytmu DMC z nieskończonym horyzontem predykcji jest przyjęcie w zadaniu optymalizacji funkcji kryterialnej ze skończonym horyzontem predykcji równym  $N = N_u + D - 2$  oraz dodanie do zestawu ograniczeń ograniczenia równościowego  $y^\infty(k) = y^{zad\infty}(k)$ , tzn. ograniczenia

$$y^0(k + N_u + D - 1|k) + \Delta y(k + N_u + D - 1|k) = y^{zad\infty}(k).$$

W sytuacjach krótkich horyzontów dynamiki  $D$  zaproponowane sformułowanie może okazać się obliczeniowo niezbyt praktyczne. Przy występowaniu ograniczeń przyrostów czy wartości sterowań może bowiem wówczas zdarzyć się przypadek pustości zbioru dopuszczalnego, który należy odpowiednio traktować (zob. następny rozdział). Jednym ze sposobów takiego traktowania jest wprowadzenie krytycznego ograniczenia do funkcji kryterialnej w postaci składnika kary, stosując minimalizację następującej jej postaci

$$J^p(\infty) = \sum_{p=N_1}^{N_u+D-2} \left\| [y^{zad}(k+p|k) - y^0(k+p|k)] - \Delta y(k+p|k) \right\|_{\Psi}^2 + \\ + \rho \left\| [y^{zad\infty}(k) - y^\infty(k)] \right\|_{\Psi}^2 + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\Lambda}^2,$$

gdzie wartość współczynnika kary  $\rho$  powinna być tak dobrana, aby zapewnić spełnienie równości  $y^\infty(k) = y^{zad\infty}(k)$  z pożądaną dokładnością. Przyjmując  $\Psi(p) = \Psi$  dla  $p = 1, \dots, N_u + D - 2$ ,  $\Psi(N_u + D - 1) = \rho\Psi$  sformułowaną powyżej funkcję kryterialną można przedstawić w równoważnej postaci standardowej dla algorytmu DMC (zob. rozdz. 3.2)

$$J^p(\infty) = \sum_{p=N_1}^{N_u+D-1} \left\| [y^{zad}(k+p|k) - y^0(k+p|k)] - \Delta y(k+p|k) \right\|_{\Psi(p)}^2 + \\ + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\Lambda}^2. \quad (3.241)$$

Zauważmy, że zaproponowane sposoby obliczania wartości funkcji kryterialnej z nieskończonym horyzontem w algorytmie DMC prowadzą do dużych wymiarowości macierzy  $\mathbf{M}^P$  i macierzy dynamicznej  $\mathbf{M}$  w przypadku dużych wartości horyzontu dynamiki obiektu  $D$ , zob. (3.34) i (3.35). Macierze

te liczymy bowiem dla skończonego horyzontu predykcyjnego  $N = N_u + D - 1$ . Dlatego też, ze względów obliczeniowych, istotny jest dobór odpowiednio dużego okresu próbkowania. Skutkuje to wówczas mniejszą wartością  $D$ .

Algorytm GPC z nieskończonym horyzontem predykcyjnym, nazywany algorytmem  $\text{GPC}^\infty$ , rozważał Scokaert w [128]. Zdefiniował on chwilę końcową jako  $N = \max\{N_u + n_B, n_A, n_C\}$ , gdzie  $n_B$ ,  $n_A$  i  $n_C$  to parametry modelu wykorzystywanego w algorytmie GPC, zob. (3.63). Natomiast za uchyb stanu (odpowiednik  $x(k+p|k) - x_s$ ) przyjął wektor uchybów regulacji  $y^{zad} - y(k+p|k)$  w  $\bar{n}_A$  kolejnych chwilach od  $(k+N|k)$ -tej poczynając, gdzie  $\bar{n}_A$  to stopień stabilnej części wielomianu  $A(z^{-1})$  występującego w modelu obiektu. Przyjęty sposób zdefiniowania stanu doprowadził Scokaerta do dość złożonego sformułowania algorytmu na skończonym horyzoncie, z dodatkowym ograniczeniem równościowym. Czytelnika zainteresowanego szczegółami realizacji algorytmu  $\text{GPC}^\infty$  odsyłamy do pracy [128].

Należy podkreślić, że opracowanie sformułowań algorytmów regulacji predykcyjnej gwarantujących nominalną stabilność pozwoliło przede wszystkim lepiej uzasadnić, zrozumieć *mechanizmy wpływające na stabilizowanie pętli regulacji* z algorytmami MPC. Są to, zauważone już przecież wcześniej empirycznie, stabilizujący wpływ wydłużania horyzontu predykcyjnego oraz wpływ ograniczeń stanu końcowego. Wyniki dotyczące stabilności wpłynęły też zapewne na formułowanie kolejnych wersji produktów komercyjnych, stosowanych już przecież od wielu lat, z dużym powodzeniem.

### **Stabilne teoretycznie algorytmy z modelem nieliniowym i ograniczeniami**

Jak zaznaczono we wstępie, przedstawienie zagadnień stabilności regulacji predykcyjnej z zamierzenia ograniczone jest do pokazania podstawowych mechanizmów zapewniania stabilności, przede wszystkim dla potrzeb prezentowanych algorytmów. Dlatego też nie jest to wyczerpujący przegląd problematyki, bardzo obszernej szczególnie w zakresie algorytmów nieliniowych.

Przedstawiliśmy już i uzasadniliśmy podstawowe mechanizmy zapewniające nominalną stabilność: narzucenie ograniczenia równościowego na stan obiektu na końcu horyzontu predykcyjnego i wydłużenie horyzontu predykcyjnego do nieskończoności. Uzasadnienia były ogólne, dotyczące obiektów zarówno liniowych, jak i nieliniowych. Zwróciliśmy uwagę, że narzucanie ograniczenia na stan końcowy jest, szczególnie dla krótszych horyzontów, zbyt restrykcyjne. Z drugiej strony, przy wydłużeniu horyzontu predykcyjnego do nieskończoności niezbędne jest dysponowanie efektywnym sposobem liczenia wartości funkcji kryterialnej na takim horyzoncie. Sposób obliczania, polegający na zastąpieniu kosztu odpowiadającego końcowemu nieskończo-

nemu odcinkowi horyzontu odpowiednim kosztem stanu końcowego, podano dla liniowych modeli obiektów w postaci równań stanu. Niestety, dla obiektów nieliniowych problem jest znacznie trudniejszy. Stąd szereg propozycji algorytmów o różnych stopniach złożoności, dla uzyskania stabilności i realizowalności numerycznej wykorzystujących, w różnych konfiguracjach, następujące podstawowe mechanizmy:

- zbiór ograniczający stany końcowe – zrelaksowany warunek narzucenia określonej wartości stanu na końcu horyzontu predykcji,
- koszt stanu końcowego – jako odpowiednik wartości odrzuconej nieskończonej sumy składników funkcji kryterialnej od końca horyzontu predykcji do nieskończoności,
- stabilizujące liniowe sprzężenie od stanu wyznaczone w rozważanym punkcie równowagi obiektu nieliniowego, najczęściej dla liniowego przybliżenia obiektu w tym punkcie.

Przedstawienie stabilnych algorytmów predykcyjnych dla nieliniowych obiektów z ograniczeniami sterowań i stanów (wyjść) wykracza znacznie poza ramy niniejszej książki. Zainteresowanego czytelnika odsyłamy tu przede wszystkim do znakomitego, obszernego artykułu przeglądowego Mayne'a i in. [98] zawierającego wyczerpującą bibliografię tematu, zob. też [102, 1, 87]. Poniżej skomentujemy jedynie krótko kilka wybranych prac, zdaniem autora ważnych, reprezentatywnych dla rozwoju problematyki, formułowania nowych koncepcji.

Niewątpliwie prekursorskie są prace [69, 23, 63], pokazujące zapewniający stabilność wpływ narzucenia wartości stanu końcowego, przy czym tylko ostatnia z nich dotyczy regulacji dyskretnej z dodatkowymi ograniczeniami sterowania i stanu. Interesującą analizę stabilności i cech regulatora predykcyjnego z czasem ciągłym zawiera praca [110].

W pracy [101] Michalska i Mayne zaproponowali algorytm, w którym zamiast narzucania stanu końcowego wprowadzono zbiór dopuszczalnych stanów końcowych  $W_\alpha$ , będący pewnym otoczeniem punktu równowagi. Koncepcja algorytmu polega na sprowadzaniu stanu obiektu algorytmem predykcyjnym ze skończonym horyzontem, w skończonym czasie do zbioru  $W_\alpha$ , gdzie sterowanie jest przełączane na liniowe sprzężenie zwrotne od stanu doprowadzające obiekt do punktu równowagi – stąd zaproponowana przez autorów nazwa „algorytm *dual-mode*”. W omawianym algorytmie długość skończonego horyzontu predykcji jest zmienna, na bieżąco dopasowywana w trakcie jego działania.

Chen i Allgöwer [24] zaproponowali (dla czasu ciągłego, podobnie jak w [101]) algorytm z *quasi-nieskończonym horyzontem*, w którym wykorzystano wszystkie trzy wymienione wyżej mechanizmy zapewniania stabilno-

ści. Sterowanie jest wyznaczane na bieżąco z rozwiązania zadania optymalizacji na skończonym horyzoncie, ze zbiorem ograniczającym stany końcowe i z kwadratowym składnikiem kary za stan końcowy w funkcji kryterialnej. Składnik ten jest wyznaczany jako ograniczenie górne wartości funkcji kryterialnej na odcinku od stanu końcowego do nieskończoności, ze sterowaniem generowanym liniowym sprzężeniem od stanu (macierz składnika kary jest wyznaczana off-line). Interesującą cechą algorytmu jest zapewnienie stabilności, nawet jeśli w zadaniu optymalizacji nie jest znajdowane minimum globalne, konieczne jest natomiast znajdowanie w każdej chwili próbkowania rozwiązania dopuszczalnego (cecha zwana *feasibility implies stability*). Właściwość tę wykazano także w pracy [130] dla suboptymalnych wersji dyskretnych klasycznego algorytmu z narzuconym stanem końcowym i algorytmu *dual-mode* [101], przy założeniach dopuszczalności rozwiązań w każdym kroku i wymaganiu jedynie pewnej poprawy wartości funkcji kryterialnej.

Algorytm dyskretny o strukturze prawie analogicznej do rozważanej w [24] analizowany był w serii prac De Nicolo, Magniego i Scattoliniego, zob. np. [108]. Wartość składnika kary za stan końcowy jest tu jednakże wyznaczana przez bezpośrednie policzenie wartości funkcji kryterialnej na przedziale od chwili końcowej horyzontu optymalizacji  $N$  do pewnego arbitralnego (dostatecznie odległego) momentu  $M$ , ze sterowaniem wyznaczanym na tym odcinku przez liniowe sprzężenie od stanu. Stąd wadą podejścia był większy nakład obliczeniowy i arbitralność wyboru wartości  $M$ . W pracy [89] cechy algorytmu poprawiono podając zależność na wyznaczenie  $M$  oraz obniżając nakład obliczeń poprzez wprowadzenie horyzontu sterowania  $N_u \leq N$ , analogicznie jak to jest w typowych algorytmach MPC z modelami liniowymi. Na odcinku czasu predykowanego od chwili bieżącej  $k$  do chwili  $k + N_u - 1$  sterowanie jest wyznaczane z rozwiązania zadania optymalizacji, a na odcinku od  $k + N_u$  do  $k + N - 1$  sterowanie wynika z wyznaczonego na etapie projektowania liniowego sprzężenia od stanu.

Interesującą koncepcję znacznie efektywniejszego, w sensie obliczeń koniecznych do wykonywania *on-line*, algorytmu przedstawiono w pracy [18], dla zadania z ograniczeniami narzuconymi jedynie na wartości sterowania. Pomysł polega na zastosowaniu optymalnego liniowego sprzężenia od stanu na całym horyzoncie predykcyjnym i optymalizowaniu jedynie prostej poprawki sterowania dla zapewnienia jego dopuszczalności.

Przedstawiony powyżej bardzo zwięzły przegląd pokazuje, że algorytmy regulacji predykcyjnej z optymalizacją na nieliniowych modelach obiektów i z ograniczeniami to obszar intensywnej działalności badawczej. Pomimo faktu, że zidentyfikowano już podstawowe mechanizmy zapewniania nomi-

nalnej stabilności, algorytmy nie wyszły jeszcze w istocie poza stadium prac teoretycznych ilustrowanych przykładami symulacji komputerowych, nie osiągnęły jeszcze etapu pozwalającego skutecznie przekonywać potencjalnych użytkowników poważnych zastosowań. W tej sytuacji najsensowniej jest wprowadzać do praktyki algorytmy stanowiące bezpośrednio rozszerzenia sprawdzonych algorytmów z modelami liniowymi, takie jak będące przedmiotem rozważań rozdziału 3.5 algorytmy z linearyzacjami modelu nieliniowego.

Teoretycy zajmujący się badaniem stabilności układów regulacji stronią, jak dotąd, od badania tych algorytmów – najprawdopodobniej ich analiza nie jest w obecności ograniczeń łatwa. Jednakże, jak pokazuje wieloletnie już doświadczenie, brak rezultatów teoretycznych nie przeszkadzał dotychczas w zastosowaniach intuicyjnie przekonywających i zweryfikowanych symulacyjnie czy eksperymentalnie algorytmów regulacji predykcyjnej.

### 3.6.2. Dopuszczalność ograniczeń, dobór parametrów

#### Zapewnianie niepustości zbioru dopuszczalnego

W rozdziale 3.1 przedstawiliśmy klasy ograniczeń występujących w algorytmach regulacji predykcyjnej: ograniczenia wartości i przyrostów sygnałów sterujących (3.6) i (3.7), ograniczenia wartości wyjść regulowanych (3.8) i wyjść nieregulowanych (3.9). W przypadku występowania ograniczeń wyjść, w pewnych chwilach pracy algorytmu predykcyjnego może wystąpić pusty zbiór rozwiązań zadania optymalizacji. Na przykład, po wystąpieniu silnego zakłócenia może nie być możliwe, przy ograniczeniach na sterowania, sprowadzenie wyjść do wartości spełniających ograniczenia szybciej niż w kilku krokach algorytmu – wówczas niemożliwe jest spełnienie ograniczeń wyjść (3.8) czy (3.9) w chwili  $k + N_1$  i ewentualnie jeszcze w kilku kolejnych chwilach. Jednym ze sposobów postępowania jest wówczas zastosowanie koncepcji tzw. *okna ograniczeń* (*constraint window*).

Stosowanie okna ograniczeń polega na uwzględnianiu, przy wyliczaniu sterowania w chwili  $k$ , ograniczeń wyjść jedynie w prognozach na chwile  $k+p$  dla  $N_{cw1} \leq p \leq N_{cw}$ , gdzie wartości  $N_{cw1}$  i  $N_{cw}$ ,  $N_1 \leq N_{cw1} < N_{cw} \leq N$ , definiują szerokość okna ograniczeń. Odpowiednio duża wartość  $N_{cw1}$  powinna zapewnić spełnienie ograniczeń w każdej chwili okna ograniczeń, zaś wartość  $N_{cw}$  powinna być taka, aby wszystkie ograniczenia były spełnione również za oknem, do końca horyzontu predykcyjnego, co oczywiście jest zapewnione, jeśli przyjąć  $N_{cw} = N$ , por. (3.8), (3.9). Dla stabilnego obiektu liniowego Rawlings i Muske wykazali w [122] istnienie takich wartości  $N_{cw1}$  i  $N_{cw}$ , że przy występowaniu ograniczeń sterowań  $Gu(k+p|k) \leq g$ ,  $p = 0, 1, \dots$ ,

ograniczenia wyjść  $Hx(k+p|k) \leq h$  (zapisane poprzez zmienne stanu) są spełnione dla  $p = N_{cw1}, \dots, N_{cw}$  i dla  $p > N_{cw}$  ( $G$  i  $H$  to macierze określające postacie ograniczeń). Jest faktem, że strategia typu okna ograniczeń była stosowana w komercyjnym pakiecie DMC. Oczywiście, strategia ta nie usuwa przekraczania ograniczeń w pierwszych chwilach horyzontu predykcji i może być stosowana, jeśli takie przekraczanie w krótkich odcinkach czasu jest w obiekcie możliwe do zaakceptowania.

Często stosowanym rozwiązaniem praktycznym problemu zapewnienia niepustości zbioru dopuszczalnego jest potraktowanie wszystkich lub tylko niektórych ograniczeń wartości wyjść jako tzw. ograniczeń *miękkich*. Takie ograniczenia mogą być naruszane, ale naruszenie to jest odpowiednio silnie karane składnikami kary dodanymi do funkcji kryterialnej tak, że występuje w istocie jedynie w sytuacji braku rozwiązań dopuszczalnych.

Przypomnijmy zadanie optymalizacji regulatora predykcyjnego, sformułowane w standardowej dla metod numerycznych postaci (3.52), ale z odpowiadającą ograniczeniom nierównościowym macierzą  $\mathbf{A}$  podzieloną na części odpowiadające, odpowiednio, ograniczeniom sterowań i wyjść:

$$\begin{aligned} \min \left\{ J(x) = \frac{1}{2} x^T \mathbf{H} x + f^T x \right\} \\ \text{z ogr.: } x_{\min} \leq x \leq x_{\max}, \\ \mathbf{A}_u x \leq b_u, \\ \mathbf{A}_y x \leq b_y, \end{aligned} \quad (3.242)$$

gdzie  $\mathbf{A}^T = [\mathbf{A}_u^T \ \mathbf{A}_y^T]^T = [-\mathbf{J}^T \ \mathbf{J}^T \ -\mathbf{M}^T \ \mathbf{M}^T]^T$ ,  $b^T = [b_u^T \ b_y^T]^T$ , por. (3.52).

Zrelaksowanie ograniczeń  $\mathbf{A}_y x \leq b_y$  można uzyskać formułując zadanie (3.242) w postaci

$$\begin{aligned} \min \left\{ J(x) = \frac{1}{2} x^T \mathbf{H} x + f^T x + \rho_i \sum_{i=1}^{n_v} v_i^2 \right\} \\ \text{z ogr.: } x_{\min} \leq x \leq x_{\max}, \\ \mathbf{A}_u x \leq b_u, \\ \mathbf{A}_y x \leq b_y + v, \\ v \geq 0, \end{aligned} \quad (3.243)$$

gdzie  $v = [v_1 \ \dots \ v_{n_v}]^T$  jest wektorem dodatkowych zmiennych o wymiarowości  $n_v$  równej liczbie relaksowanych ograniczeń,  $n_v = \dim b_y$ , zaś  $\rho_i > 0$  to współczynniki kary, w ogólności różne dla różnych ograniczeń. Składniki kary nie muszą być kwadratowe, zamiast  $\sum_{i=1}^{n_v} v_i^2$  można zastosować

$\sum_{i=1}^{n_v} v_i$  – co jest sformulowaniem tzw. dokładnej funkcji kary (*exact penalty function*). Funkcja ta staje się „dokładna”, tzn. ograniczenia nie są naruszone, dla skończonych, odpowiednio dużych wartości współczynników  $\rho_i$  – oczywiście w przypadku niepustego zbioru dopuszczalnego.

Ponieważ w praktyce waga spełnienia poszczególnych ograniczeń wyjść bywa różna, to można stosować odpowiednie szeregowanie ograniczeń w kolejności ważności ich przestrzegania, czyli odpowiednią *priorytetyzację* ograniczeń. Przykładem takiej prostej priorytetyzacji jest wprowadzenie różnych współczynników kary dla różnych relaksowanych ograniczeń w sformułowaniu zadania optymalizacji (3.243). W pewnych problemach może być sensowne traktowanie jako ograniczeń miękkich jedynie tych o niższych priorytetach, czy też szeregowanie naruszania ograniczeń w ramach ograniczeń miękkich odpowiednio sklasyfikowanych. Można to zorganizować przez kolejne sprawdzanie niepustości zbioru dopuszczalnego poczynając od „twardego” traktowania wszystkich ograniczeń, poprzez relaksowanie ograniczeń kolejno od tych o najniższych priorytetach, aż do uzyskania dopuszczalności rozwiązań [157].

### Dobór parametrów algorytmów predykcyjnych

W punkcie niniejszym omówimy zagadnienia znaczenia i doboru podstawowych parametrów algorytmów regulacji predykcyjnej. Będzie to wystarczające dla algorytmów z modelami liniowymi, tzn. omawianych we wcześniejszych częściach rozdziału algorytmów DMC, GPC i MPC (dla MPC parametrami dostrajalnymi są ponadto współczynniki macierzy obserwatora stanu, zob. rozdz. 3.4). Natomiast w przypadku algorytmów z modelami nieliniowymi występują z reguły dodatkowe parametry związane z rodzajem algorytmu, jak np. liczba i parametry zbiorów rozmytych występujących w poprzednikach reguł regulatorów predykcyjnych rozmytych o strukturze Takagi-Sugeno, omawianych w rozdziale 3.5.

Podstawowymi *parametrami dostrajalnymi* algorytmu MPC generującego sterowanie w wyniku optymalizacji funkcji kryterialnej (3.1) są:

$N$  – horyzont predykcji,

$N_u$  – horyzont sterowania,  $N_u \leq N$ ,

$\Psi(p)$ ,  $\Lambda(p)$  – macierze współczynników wagowych w funkcji kryterialnej,  $N_{cw1}$ ,  $N_{cw}$  – granice okna ograniczeń, lub inne parametry określające strategię zapewniania niepustości zbioru dopuszczalnego (np. wartości współczynników kary  $\rho_i$  w (3.243)),

$T_p$  – okres próbkowania (czas powtarzania interwencji regulatora),

$\gamma$  – współczynnik filtra liniowego określającego przebieg trajektorii referencyjnej.

Za dodatkowy parametr algorytmu predykcyjnego można też uważać wartość  $N_1$  – chwilę początku uwzględniania uchybu regulacji w funkcji kryterialnej. Z reguły przyjmuje się  $N_1 = \tau + 1$ , gdzie  $\tau$  oznacza liczbę chwil opóźnienia w odpowiedzi skokowej obiektu. Wybór  $N_1 < \tau$  nie prowadzi do błędów — ale jest mało sensowny, ponieważ predykcja wyjść obiektu dla chwil  $k + p \leq k + \tau$  nie zależy od zmiennych decyzyjnych w chwili  $k$  (sterowań  $u(k + p|k)$ ).

Wartość  $N$  określa długość horyzontu predykcji. Nieskończony horyzont zapewnia nominalną stabilność przy każdych wartościach pozostałych parametrów, pod warunkiem niepustego zbioru rozwiązań, jak to przedstawiliśmy w rozdz. 3.6.1. Jeśli horyzont predykcji  $N$  jest dostatecznie długi w stosunku do cech dynamiki obiektu i układ regulacji pracuje prawidłowo, to dalsze zwiększanie  $N$  z reguły niewiele zmienia, własności układu regulacji stają się niewrażliwe na wartość  $N$ .

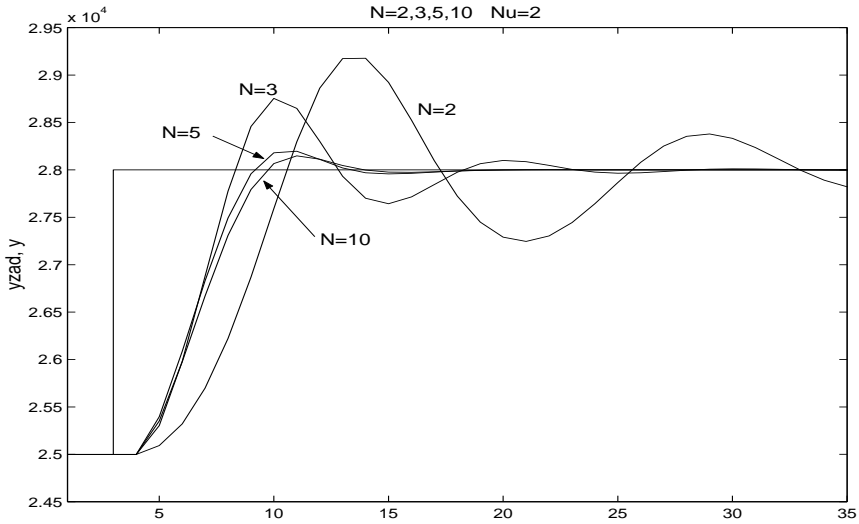
Stosunkowo mała jest wrażliwość układu regulacji na długość horyzontu sterowania  $N_u$ , byle tylko horyzont ten nie był za krótki [102]. Ponieważ wartość  $N_u$  decyduje o liczbie zmiennych decyzyjnych zadania optymalizacji (równiej  $N_u n_u$ ), to przyjmuje się na ogół wartości niewielkie. Ma to szczególnie duże znaczenie w przypadku algorytmów typu NO, z nieliniową optymalizacją wykonywaną w każdym kroku. Możliwe są wartości skrajnie małe, aż do  $N_u = 1$ , jak to pokazujemy w przykładzie poniżej (interesującą dyskusję takich przypadków znaleźć można w [87]).

### Przykład 3.15.

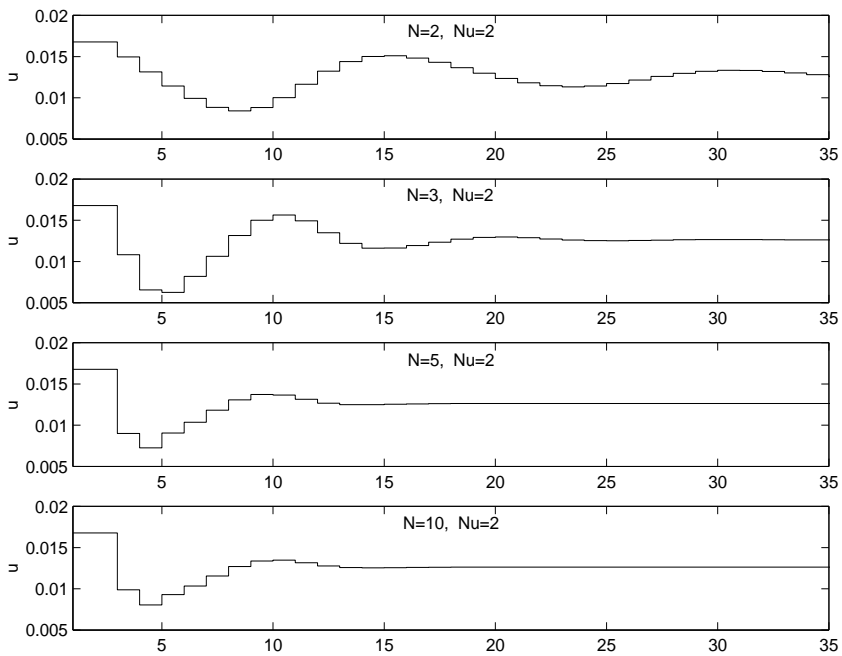
Wróćmy do przykładu silnie nieliniowego przepływowego reaktora z procesem polimeryzacji przedstawionego w przykładzie 3.11. Pokażemy teraz, jak zmienia się charakter pracy układu regulacji z algorytmem MPC-NO (z nieliniową optymalizacją), przy zmianach długości horyzontu predykcji  $N$  i długości horyzontu sterowania  $N_u$ .

Na rysunkach 3.77 i 3.78 przedstawiono przebiegi trajektorii wyjścia i sterowania dla horyzontu sterowania  $N_u = 2$  i horyzontów predykcji, kolejno,  $N = 2, 3, 5$  i  $10$ . Zwraca uwagę niewielka różnica między przebiegami dla  $N = 5$  i  $N = 10$ . Wynik dla  $N = 10$  podano jako przykładowy dla zakresu wartości horyzontów predykcji na tyle dużych, że już nie wpływających na przebiegi w układzie regulacji. Zbadano bowiem również zachowanie się układu regulacji przy zastosowaniu większych wartości  $N$ , uzyskując przebiegi praktycznie pokrywające się z zaprezentowanymi dla  $N = 10$ .

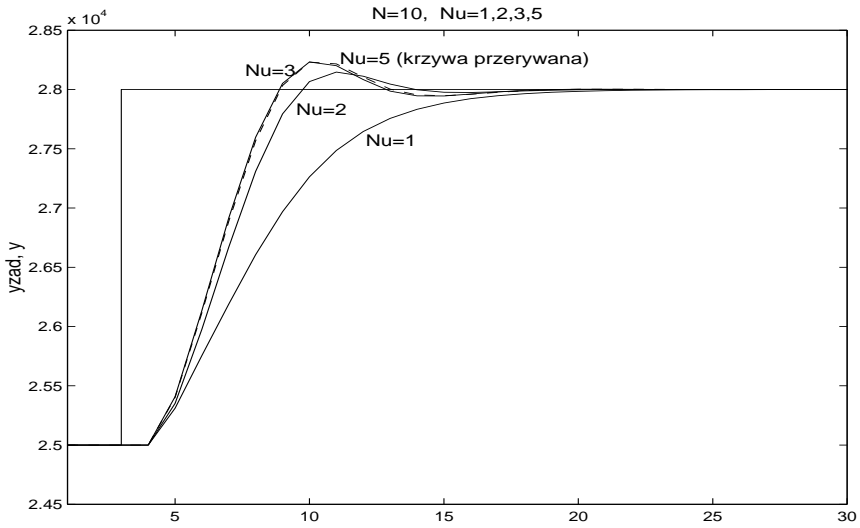
Przedstawione wyniki eksperymentów symulacyjnych wyraźnie pokazują stabilizujący wpływ wydłużania horyzontu predykcji. Przy okazji wy-



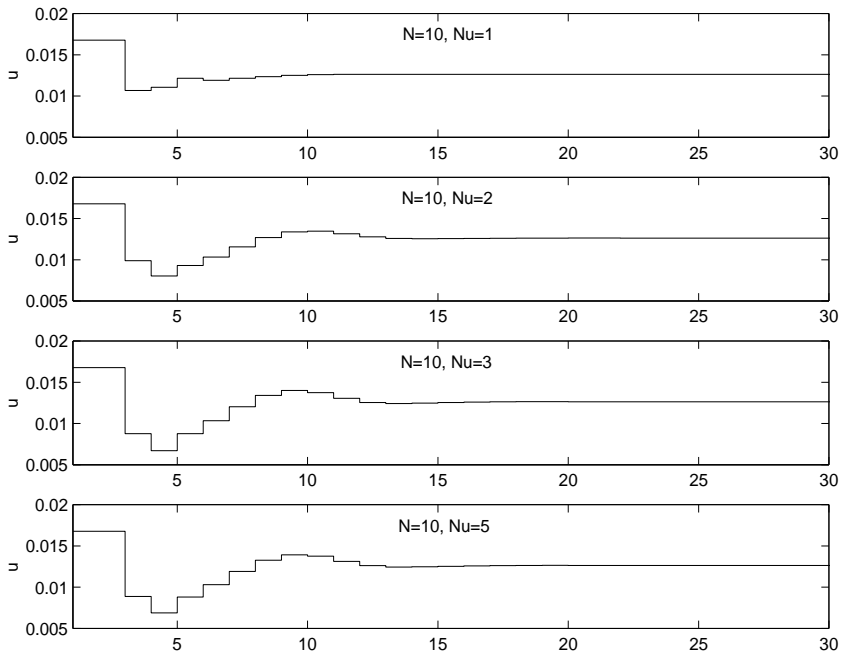
Rysunek 3.77. Trajektorie zmiennej regulowanej dla różnych wartości horyzontu predykcji



Rysunek 3.78. Trajektorie sterowania dla różnych wartości horyzontu predykcji



Rysunek 3.79. Trajektorie zmiennej regulowanej dla różnych wartości horyzontu sterowania



Rysunek 3.80. Rysunek 3.60: Trajektorie sterowania dla różnych wartości horyzontu sterowania

jaśniają, dlatego w przykładzie 3.11 przyjęto horyzont predykcji  $N = 10$ , przy znacznie krótszym horyzoncie sterowania  $N_u = 2$ .

Na rysunkach 3.79 i 3.80 przedstawiono przebiegi trajektorii wyjścia i sterowania dla horyzontu predykcji  $N = 10$  i horyzontów sterowania, kolejno,  $N_u = 1, 2, 3$  i  $5$ . Różnica jakościowa występuje między przebiegami uzyskanymi przy  $N_u = 1$  oraz dla wartości większych. Między trajektoriami dla  $N_u = 2$  i  $N_u = 3$  mamy już tylko różnice ilościowe, i to niezbyt duże. Natomiast praktycznie nie ma różnic w przebiegach dla wartości  $N_u = 3$  i większych – pokazane na rysunkach trajektorie dla  $N_u = 3$  oraz, przykładowo, dla  $N_u = 5$  różnią się minimalnie. Symulowano regulację dla kilku większych wartości  $N_u$  uzyskując wyniki pokrywające się na wykresach z uzyskanymi dla  $N_u = 5$ .

Przedstawione rezultaty są typowe: pokazują, że na ogół wystarcza przyjęcie niewielkich wartości  $N_u$ , przebiegi w układzie regulacji szybko stają się niewrażliwe na wzrost długości horyzontu sterowania.  $\square$

Podstawowymi parametrami dostrajalnymi wpływającymi na dynamikę układu regulacji predykcyjnej, w tym na stabilność przy skończonej długości horyzontu predykcji, są wartości elementów macierzy  $\Psi(p)$  i  $\Lambda(p)$ . Macierze te przyjmowane są w postaci diagonalnej i najczęściej nie zależą od  $p$ ,  $\Psi(p) = \Psi$  i  $\Lambda(p) = \Lambda$ . Stosunki elementów  $\psi_{ii}$  macierzy  $\Psi$  określają wzajemne relacje wagi pomiędzy uchybami regulacji odpowiadającymi poszczególnym wielkościom regulowanym. Natomiast stosunki elementów  $\lambda_{ii}$  macierzy  $\Lambda$  określają wzajemne relacje wagi pomiędzy poszczególnymi zmiennymi sterującymi. Najistotniejsze są jednakże relacje między elementami macierzy  $\Psi$  i macierzy  $\Lambda$ , określają one bowiem koszt zmiany sterowań w stosunku do kosztu uchybów regulacji. Zwiększanie wartości  $\lambda_{ii}$  w stosunku do wartości  $\psi_{ii}$  powoduje ograniczanie zmienności sterowań, wolniejszą ale bardziej płynną pracę układu regulacji, zmniejsza ryzyko ewentualnej aktywności ograniczeń sterowań, zwiększa ponadto z reguły marginesy stabilności, odporność. W najprostszym przypadku, gdy  $\Psi(p) = \mathbf{I}$  oraz  $\Lambda(p) = \lambda \mathbf{I}$ , odpowiadającym funkcji kryterialnej (3.4), pozostaje  $\lambda$  jako skalarny parametr dostrajalny. Jest on dla działania algorytmów liniowych (DMC, GPC czy MPCs) *podstawowym parametrem dostrajalnym*, zob. np. [17]. Wpływ wartości  $\lambda$  na odpowiedzi skokowe w układzie regulacji liniowej (DMC) analizowany był w przykładzie 3.1 w rozdziale 3.2.2.

W przypadku występowania ograniczeń wyjść istotne są zarówno parametry, jak i struktury algorytmu predykcyjnego zapewniające w każdej chwili próbkowania niepustość zbioru dopuszczalnego. Zagadnienie to, bardzo ważne, zostało omówione w poprzednim punkcie niniejszego rozdziału.

Dobór odpowiedniego okresu próbkowania  $T_p$  jest ważny i powiązany

z długościami horyzontów. Szczególnie dobrze jest to widoczne na przykładzie algorytmu DMC. Przyjęcie małej wartości  $T_p$  skutkuje dużą liczbą elementów odpowiedzi skokowej, tzn. długim horyzontem dynamiki obiektu  $D$ , a to z kolei prowadzi do większych wymiarów macierzy  $\mathbf{M}^P$  i  $\mathbf{M}$ , zob. rozdz. 3.2. Stąd większa wymiarowość i czas obliczeń zadania optymalizacji, jeśli chcemy utrzymać podobną proporcję horyzontów predykcji i sterowania do horyzontu dynamiki obiektu. Natomiast przyjęcie zbyt dużej wartości  $T_p$  powoduje, iż przebieg schodkowy sterowania nie jest dopasowany do dynamiki obiektu, pogarsza z reguły dynamikę układu regulacji przez wprowadzenie większego opóźnienia.

W każdym z algorytmów regulacji predykcyjnej można zastosować w funkcji kryterialnej w miejsce trajektorii wartości zadanych  $y^{zad}(k+p|k)$ ,  $p = N_1, \dots, N$ , tzw. trajektorię referencyjną (*reference trajectory*)  $y^{ref}(k+p|k)$ ,  $p = N_1, \dots, N$ , wykorzystywaną już w algorytmie *Model Predictive Heuristic Control* (MPHC) [123, 124], znanym później bardziej pod nazwą MAC (*Model Algorithmic Control*). Trajektorja referencyjna definiowana jest zależnością

$$y^{ref}(k+p|k) = \gamma y^{ref}(k+p-1|k) + (1-\gamma)y^{zad}(k+p|k), \quad p = 1, \dots, N,$$

przy czym  $y^{ref}(k|k) = y(k)$ , zob. np. [16, 17].

Dla najistotniejszego przypadku przyjmowania stałej na horyzoncie predykcji wartości zadanej,  $y^{zad}(k+p|k) = y^{zad}(k)$ ,  $p = 1, \dots, N$ , cała trajektorja referencyjna jest w chwili  $k$  jednoznacznie zdefiniowana wartościami  $y(k)$  i  $y^{zad}(k)$ , stąd można ją przedstawić w nieco dogodniejszej postaci

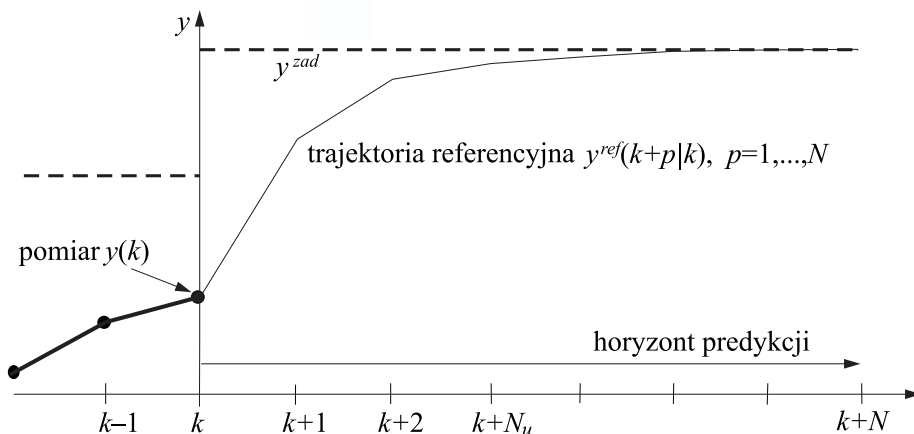
$$\begin{aligned} y^{ref}(k+p|k) &= (1-\gamma^p)y^{zad}(k) + \gamma^p y(k) \\ &= y^{zad}(k) - \gamma^p [y^{zad}(k) - y(k)], \quad p = 1, \dots, N, \end{aligned} \quad (3.244)$$

gdzie  $y^{zad}(k) - y(k)$  jest uchybem regulacji w chwili  $k$ . Z postaci (3.244) widać, że wymaga się redukcji uchybu regulacji do zera wykładniczo w czasie, z szybkością określoną wartością  $\gamma$ . Przykład przebiegu trajektorii referencyjnej pokazano na rysunku 3.81.

Przy zastosowaniu trajektorii referencyjnej funkcja celu algorytmu MPC przyjmuje postać

$$J(k) = \sum_{p=N_1}^N \left\| y^{ref}(k+p|k) - y(k+p|k) \right\|_{\Psi(p)}^2 + \sum_{p=0}^{N_u-1} \left\| \Delta u(k+p|k) \right\|_{\Lambda(p)}^2, \quad (3.245)$$

gdzie, standardowo, wartości  $y(k+p|k)$  reprezentują trajektorię wyjść predykowanych, a  $\Delta u(k+p|k)$  zmienne decyzyjne, porównaj z (3.1). Za-



Rysunek 3.81. Ilustracja trajektorii referencyjnej (wartość zadana stała na horyzoncie predykcji)

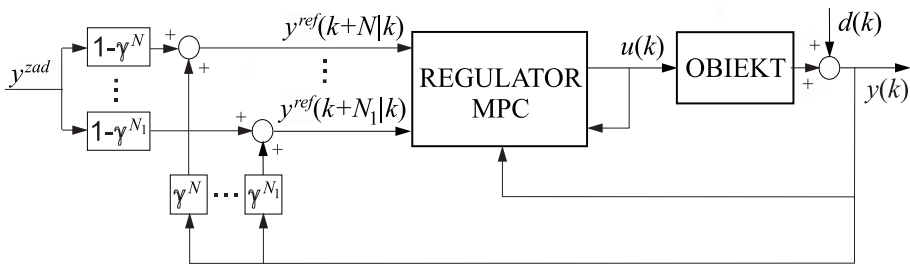
stosowanie trajektorii referencyjnej ma na celu wstępne przetworzenie przebiegu trajektorii wyjść zadanych do postaci o łagodniejszym, ciągłym przebiegu zmiennym nie szybciej niż wzorec wykładniczy (inercyjny) definiowany przez  $\gamma$ . Wartość  $\gamma$ ,  $0 \leq \gamma < 1$ , jest parametrem dostrajalnym algorytmu, dla  $\gamma = 0$  trajektoria referencyjna staje się trajektorią wartości zadanych  $y^{zad}(k+p|k)$ ,  $p = 1, \dots, N$ . Wraz ze wzrostem  $\gamma$  zbliżanie się trajektorii referencyjnej do trajektorii wartości zadanych staje się powolniejsze, łagodniejsze – stawiając przez to mniejsze wymagania dla sygnału sterującego. Podobny efekt można uzyskać i bez trajektorii referencyjnej, jedynie odpowiednio dobierając wartości elementów macierzy wagowych  $\Psi(p)$  i  $\Lambda(p)$ , a w szczególności (gdyż chodzi o relacje pomiędzy chwilami czasu) przyjmując  $\Psi(p) = \psi(p)\mathbf{I}$ ,  $\Lambda(p) = \lambda(p)\mathbf{I}$  i odpowiednio dobierając wartości elementów  $\psi(p)$  i  $\lambda(p)$ . Zastosowanie trajektorii referencyjnej jest nieco prostsze, do dostrojenia mamy tylko jeden parametr  $\gamma$ .

W przypadku skoku wartości zadanej działanie trajektorii referencyjnej ma w pierwszej chwili podobny efekt jak zastosowanie filtra inercyjnego wartości zadanej w klasycznym układzie regulacji o dwóch stopniach swobody, pokazanym na rysunku 3.82.

Jednakże *nie są to struktury równoważne*: filtr wartości zadanej w układzie z rys. 3.82 działa w układzie otwartym, natomiast zastosowanie trajektorii referencyjnej w układzie regulacji predykcyjnej jest wprowadzeniem dodatkowego sprzężenia zwrotnego. Pokazuje to rysunek 3.83, słuszny dla dowolnego regulatora MPC (np. DMC, GPC, MPCs, regulatory nieliniowe), w którym w funkcji kryterialnej zastosowano trajektorię referen-



Rysunek 3.82. Klasyczny układ regulacji o dwóch stopniach swobody



Rysunek 3.83. Struktura układu regulacji MPC z zastosowaniem trajektorii referencyjnej (wartość zadana stała na horyzoncie predykcji)

cyjną  $y^{ref}(k + p|k)$  zamiast trajektorii wartości zadanych  $y^{zad}(k + p|k)$ ,  $p = N_1, \dots, N$ . Różnica między zastosowaniem filtra wartości zadanej a trajektorii referencyjnej jest szczególnie dobrze widoczna w przypadku zmiany wartości zakłócenia spowodowanego do wyjścia. Filtr nie ma wówczas żadnego wpływu na przebiegi w układzie z rysunku 3.82, natomiast uchyb regulacji spowodowany zakłóceniem jest traktowany podobnie jak spowodowany zmianą wartości zadanej – jedynie przyczyną jego powstania jest inna, skok wartości wyjścia  $y(k)$ , a nie wartości zadanej.

Trajektoria referencyjna, traktowana jako jeden z praktycznych sposobów wpływania na zachowanie algorytmu predykcyjnego, jest obecnie w literaturze przedmiotu rzadziej spotykana. Ma ona mniejsze znaczenie dla algorytmów z ograniczeniami sterowań, realizowanych w wersji numerycznej (z bieżącą optymalizacją). Na ogół nie ma bowiem racjonalnych powodów, aby spowalniać redukcję uchybów regulacji w sytuacji pełnej kontroli nad ograniczeniami zarówno amplitud, jak i prędkości zmian sterowań. Natomiast na zachowanie samego układu regulacji, jego cechy stabilnościowe czy, przede wszystkim, odpornościowe, większy wpływ ma odpowiedni dobór współczynników wagowych, przede wszystkim elementów macierzy  $\mathbf{\Lambda}$  (zob. przykład 3.2).

Wart uwagi jest pogląd wyrażony przez Maciejowskiego [87]: wydaje

się sensownym stosowanie filtra wartości zadanej dla unikania długotrwałych nasyczeń elementu wykonawczego wywoływanych skokami wartości zadanej, natomiast dla szybkiej redukcji uchybów regulacji spowodowanych zakłóceniami wystarczy polegać na mechanizmie regulatora predykcyjnego z ograniczeniami (bez trajektorii referencyjnej).



## Rozdział 4

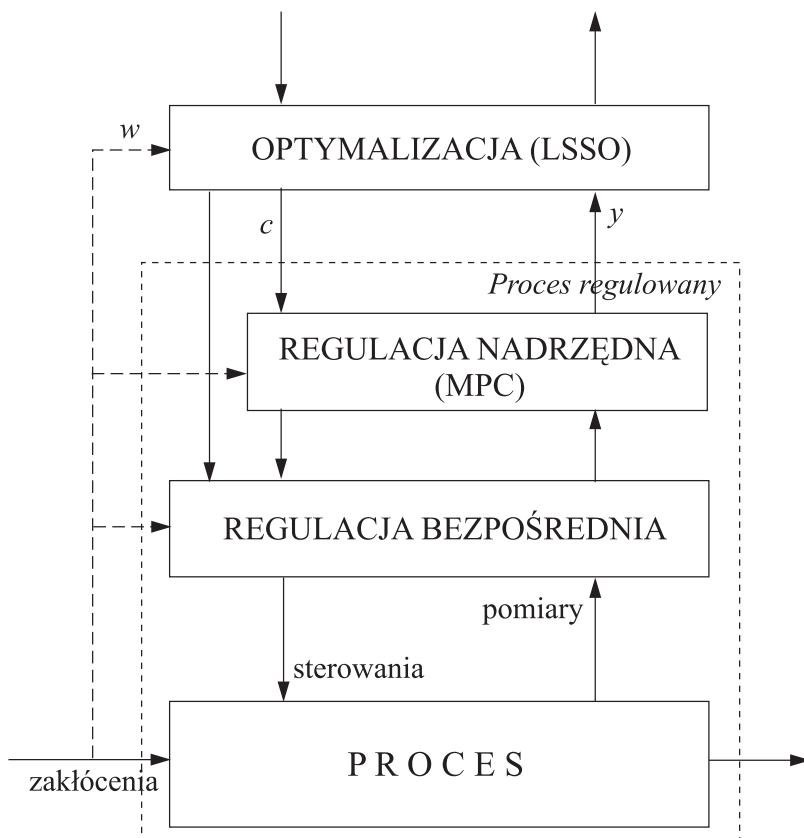
# Optymalizacja bieżąca punktu pracy

### 4.1. Miejsce optymalizacji punktu pracy w strukturze warstwowej

Rozważając w rozdziale 1 warstwową strukturę sterowania obiektami przemysłowymi wyróżniliśmy warstwę optymalizacji, zob. rys. 1.2. Zadaniem warstwy optymalizacji jest wyznaczanie *optymalnych wartości zadanych* dla regulatorów bezpośrednio niżej usytuowanych układów regulacji. Na rysunku 1.3 przedstawiono strukturę warstwową w rozbudowanej postaci, z dekompozycją obiektu sterowanego, wydzieleniem regulacji nadrzędnej i bezpośredniej oraz zdefiniowaniem systemu wykonawczego. W rozdziale niniejszym obiektem zainteresowania będzie optymalizacja punktu pracy, czyli wartości zadanych dla regulatorów warstwy regulacji. Na rysunku 4.1 przytoczono strukturę warstwową z pominięciem warstwy najwyższej, tzn. tylko z warstwami regulacji i lokalnej optymalizacji. Jest to optymalizacja lokalna, gdyż dotyczy wydzielonego procesu z własnymi autonomicznymi układami regulacji, w odróżnieniu od nadrzędnej do niej optymalizacji globalnej dotyczącej całości obiektu – np. optymalizacja punktu pracy kolumny destylacyjnej będącej fragmentem większego ciągu technologicznego. Jest to ponadto z reguły optymalizacja statyczna, tj. optymalizacji stanu ustalonego – stąd w literaturze używane jest często określenie „*local steady-state optimization*” (LSSO).

Na rysunku 4.1 zaznaczono, że w warstwie lokalnej optymalizacji stanu ustalonego (LSSO) w ogólności wyznaczone są punkty pracy (wartości zadane dla regulatorów) nie tylko dla bezpośrednio poniżej położonej warstwy optymalizacji nadrzędnej, ale też bezpośrednio dla pewnych układów regulacji bezpośredniej – w skrajnym przypadku regulacja nadrzędna może w ogóle nie występować (zob. przykład w rozdziale pierwszym).

Warstwa LSSO ma dobierać punkty pracy w szerokim zakresie operacyjnym, stąd operuje z zasady na modelu odpowiednio dokładnym, a więc złożonym, nieliniowym. Jej okresy interwencji („okresy próbkowania”) są znacznie dłuższe niż okresy próbkowania regulatorów – punkty pracy zmieniają się przez LSSO w odstępach rzędu co najmniej godziny, podczas gdy



Rysunek 4.1. Struktura sterowania obiektem przemysłowym z warstwami regulacji i optymalizacji

okresy próbkowania regulatorów nadrzędnych MPC to co najwyżej pojedyncze minuty, a regulatorów bezpośrednich co najwyżej pojedyncze sekundy.

Zadanie optymalizacji LSSO można sformułować w postaci

$$\begin{aligned} & \min Q(c, y) \\ \text{z ogr. : } & y = F(c, w), \\ & c \in C, \\ & y \in Y, \end{aligned} \quad (4.1)$$

gdzie  $y = F(c, w)$  jest statycznym modelem procesu, opisującym relacje między wartościami zmiennych regulowanych  $c$  (których wartości optymalne będą wartościami zadanymi dla regulatorów) i wyjściami procesu

$y$  w stanach ustalonych, zaś  $C$  i  $Y$  reprezentują odpowiednie zbiory ograniczeń. Funkcja  $Q(c, y)$  definiowana jest zwykle jako określająca bieżący zysk uzyskiwany w procesie sterowanym, typową jest stąd następująca jej postać

$$Q(c, y) = \sum_{j=1}^{n^J} p_j c_j - \sum_{j=1}^{m^J} q_j y_j, \quad (4.2)$$

gdzie  $p_j$  oznaczają ceny strumieni wejściowych (surowce, energia), zaś  $q_j$  ceny produktów wyjściowych.

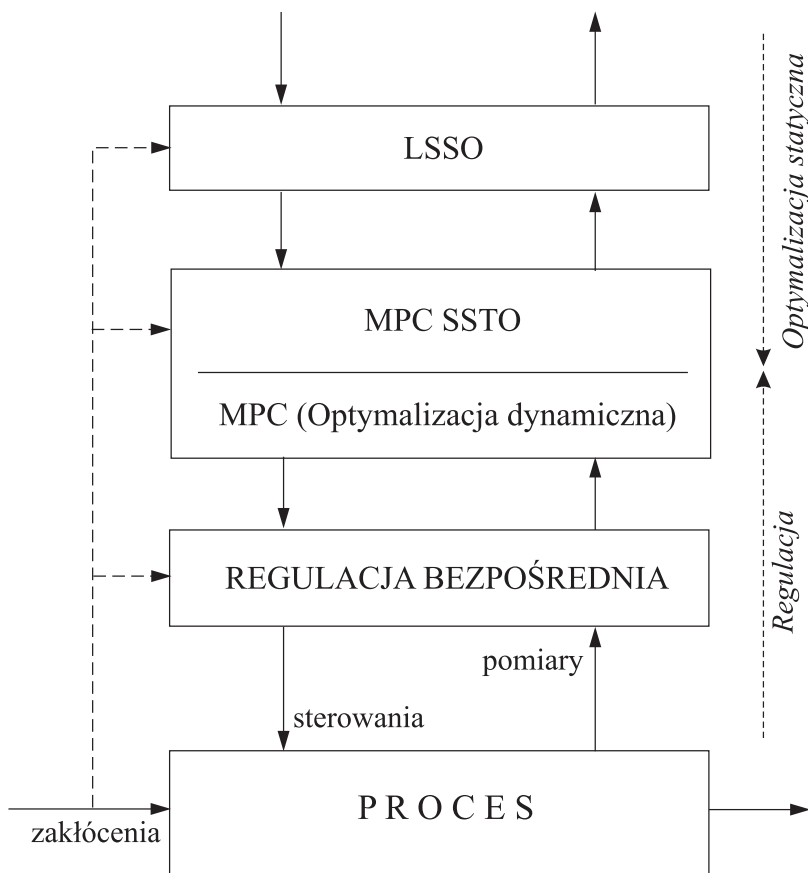
Zmienne  $w$  w modelu procesu reprezentują zakłócenia, mierzone lub estymowane. Powinny to być jedynie zakłócenia wolno lub rzadko zmienne, ze względu na częstotliwość interwencji warstwy optymalizacji. Natomiast jeśli w procesie występują zakłócenia istotnie zmienne pomiędzy chwilami interwencji warstwy optymalizacji, istotnie przesuujące optymalny punkt pracy procesu między kolejnymi interwencjami tej warstwy – to tradycyjnym podejściem było przyjmowanie do optymalizacji punktów pracy wartości średnich takich zakłóceń, na odpowiednim oknie uśredniania (estymowania wartości średniej). Jest to podejście praktyczne, ale suboptymalne i często nie akceptowalne w sytuacji, kiedy regulatorem bezpośrednio podległym warstwie optymalizacji jest regulator predykcyjny MPC i istotne jest przestrzeganie ograniczeń. W takiej sytuacji uzupełnia się strukturę sterowania o dodatkowe, optymalizacyjne dostrajanie wartości zadanych dla regulatora MPC – wprowadza się tzw. „*steady-state target optimization*” (SSTO) [9, 120, 144]. Uzupełnienie tego rodzaju występuje praktycznie we wszystkich komercyjnych pakietach oprogramowania dedykowanych nadrzędnej regulacji predykcyjnej. Zadanie SSTO omówimy w pierwszej części tego rozdziału.

Z punktu widzenia matematycznego zadania optymalizacji, zakłócenia  $w$  są parametrami w modelu  $F$  stanu ustalonego procesu. Stąd wartości tych parametrów (zakłóceń) powinny być znane – mierzone lub estymowane. Tylko wówczas zadanie optymalizacji stanu ustalonego jest w pełni matematycznie sformułowane, umożliwiając wyznaczenie optymalnego punktu pracy procesu. Dokładniej, punktu położonego w pobliżu dokładnego optimum, ze względu na zawsze występujące niedokładności modelowania równań procesu, tzn. określania struktury modelu i wyznaczania wartości jego parametrów. W sytuacjach istotnych niedokładności modelowania, ale kiedy zakłócenia (w tym błędy modelowania) mogą być traktowane jako praktycznie stałe w dłuższych okresach czasu, tzn. wielokrotnie dłuższych niż czas ustalania przebiegów w układzie regulacji procesu, możliwe jest zastosowanie iteracyjnych algorytmów poprawy punktu pracy wykorzystu-

jących bezpośrednio pomiary stanów ustalonych procesu, zob. np. [15]. Tego typu algorytmy przedstawione będą w drugiej części tego rozdziału.

#### 4.2. Dostrajanie punktu pracy dla regulatorów MPC (zadanie SSTO)

Struktura sterowania z dostrajaniem punktu pracy (SSTO) dla potrzeb optymalizacji dynamicznej regulatora MPC przedstawiona jest na rysunku 4.2. Zwróćmy uwagę, że funkcjonalnie zadanie SSTO jest zadaniem optymalizacji punktu pracy (optymalizacji statycznej), a więc takiego rodzaju jak zadanie LSSO. Jednakże jest ono uruchamiane w każdym kroku



Rysunek 4.2. Warstwowa struktura sterowania obiektem przemysłowym z warstwą regulacji MPC uzupełnionej o dostrajanie punktu pracy (SSTO)

(okresie próbkowania) regulatora MPC i stąd zostało przyporządkowane do tej samej warstwy co zadanie regulacji MPC – gdyż częstotliwość interwencji jest jedną z podstawowych cech wyróżniających poszczególne warstwy struktury warstwowej sterowania.

Zadanie SSTO jest zadaniem optymalizacji uruchamianym często, w każdym kroku (okresie interwencji) regulatora MPC, stąd musi ono opierać się na uproszczonym modelu statyki procesu, istotnie prostszym od modelu nieliniowego używanego w LSSO. Taki uproszczony, liniowy model statyki procesu to macierz wzmocnień  $\mathbf{G}$ . Macierz wzmocnień najłatwiej wyznaczyć jako zależność statyczną modelu dynamicznego używanego w regulatorze MPC.

#### 4.2.1. Zadania SSTO dla regulacji MPC z modelami liniowymi

Rozważmy na początku *regulatory MPC z liniowymi modelami obiektów*, tzn. regulatory DMC, GPC lub MPCs. W regulatorze DMC modelem liniowym jest model odpowiedzi skokowych, w tym przypadku macierz wzmocnień jest postaci

$$\mathbf{G} = \begin{bmatrix} s_D^{11} & s_D^{12} & s_D^{13} & \cdots & s_D^{1n_u} \\ s_D^{21} & s_D^{22} & s_D^{23} & \cdots & s_D^{2n_u} \\ s_D^{31} & s_D^{32} & s_D^{33} & \cdots & s_D^{3n_u} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_D^{n_y 1} & s_D^{n_y 2} & s_D^{n_y 3} & \cdots & s_D^{n_y n_u} \end{bmatrix}, \quad (4.3)$$

gdzie  $D$  jest horyzontem dynamiki obiektu, a  $n_u$  i  $n_y$  to liczby sterowań i wyjść regulowanych procesu.

Dla modelu (3.64) wykorzystywanego w algorytmie GPC mamy

$$\mathbf{G} = \mathbf{A}(1)^{-1}\mathbf{B}(1). \quad (4.4)$$

Natomiast dla modelu w postaci liniowych równania stanu (3.103)-(3.104), wykorzystywanego w algorytmie MPCs, macierz wzmocnień będzie miała postać

$$\mathbf{G} = \mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}. \quad (4.5)$$

Celem zadania SSTO jest dostrojenie wartości zadanych dla regulatora MPC do aktualnych wartości zakłóceń. Aby to uczynić, musimy mieć model procesu z estymatami tych zakłóceń. Naturalnym podejściem jest tu wykorzystanie, w danej chwili czasu, wartości wyjść predykowanych wyliczanych w algorytmie MPC, gdyż *predykcje te zawierają pomiar wyjść i oparte na nim wartości estymat zakłóceń niemierzonych*. Interesuje nas stan ustalony,

stąd poszukujemy wartości wyjść odpowiadających, w stanie ustalonym, aktualnej (ostatniej) wartości sterowań  $u(k-1)$ . Wartości takie to  $y^0(k+N)$ , tzn. wartości trajektorii swobodnej wyjść na końcu horyzontu predykcji, wyznaczanej przy stałej na całym horyzoncie predykcji wartości sterowań równej  $u(k-1)$  – pod warunkiem odpowiednio długiego horyzontu, tak aby zanikły procesy przejściowe, a wyjścia osiągnęły na końcu horyzontu predykcji stan ustalony.

Zadanie SSTO może być formułowane w dwojaki sposób, dla jednego z dwóch różnych celów, w zależności od potrzeb:

1. Wyznaczenie dopuszczalnych celów (targets) położonych jak najbliżej aktualnych wymaganych stanów ustalonych optymalnych, pochodzących z warstwy LSSO lub wyższej [121, 120]. Zadanie to występuje np. w sytuacji, kiedy wymagane stany to pożądane stężenia szkodliwych zanieczyszczeń w gazach wylotowych elektrowni, a celem regulacji predykcyjnej jest optymalna minimalizacja odchyłek od tych stanów, mimo ograniczeń. Zadanie SSTO można sformułować w tym przypadku jako odpowiednie zadanie programowania kwadratowego (QP) z ograniczeniami i liniowym modelem statyki procesu (macierzą wzmocnień  $\mathbf{G}$ ). Oznaczając stan wymagany przez  $y^{opt}$ ,  $u^{opt}$ , zadanie SSTO można w tym przypadku sformułować w postaci

$$\begin{aligned} \min_{\Delta u^{ss}, \Delta y^{ss}} \{ & \|y^{opt} - y^{ss}\|^2 + \|u^{opt} - u^{ss}\|^2 \} \\ \text{z ogr. : } & \Delta y^{ss} = \mathbf{G} \Delta u^{ss}, \\ & y^{ss} = y^0(k+N|k) + \Delta y^{ss}, \\ & u^{ss} = u(k-1) + \Delta u^{ss}, \\ & u_{min} \leq u^{ss} \leq u_{max}, \\ & y_{min} \leq y^{ss} \leq y_{max}, \end{aligned} \quad (4.6)$$

lub równoważnie, eliminując zmienne  $\Delta y^{ss}$ ,

$$\begin{aligned} \min_{\Delta u^{ss}} \{ & \|y^{opt} - y^0(k+N|k) - \mathbf{G} \Delta u^{ss}\|^2 + \|u^{opt} - u^{ss}\|^2 \} \\ \text{z ogr. : } & u^{ss} = u(k-1) + \Delta u^{ss}, \\ & u_{min} \leq u^{ss} \leq u_{max}, \\ & y_{min} \leq y^0(k+N|k) + \mathbf{G} \Delta u^{ss} \leq y_{max}. \end{aligned} \quad (4.7)$$

Po rozwiązaniu tego zadania, z wynikiem  $\hat{u}^{ss}$ , wyznaczane są odpowiednie wartości zadane wyjść  $\hat{y}^{ss} = y^0(k+N|k) + \mathbf{G} \hat{u}^{ss}$  i przekazywane do zadania optymalizacji regulatora MPC.

2. Rozwiązanie zadania programowania liniowego (LP) z oryginalną ekonomiczną funkcją celu (4.2), oryginalnymi ograniczeniami i liniowym modelem statyki procesu w postaci macierzy wzmocnień  $\mathbf{G}$ , zob. np.[9]. Zadanie SSTO może być wówczas sformułowane w następującej postaci

$$\begin{aligned} \min_{\Delta u^{ss}, \Delta y^{ss}} & \left\{ -q^T \Delta y^{ss} + p^T \Delta u^{ss} \right\} \\ \text{z ogr. : } & \Delta y^{ss} = \mathbf{G} \Delta u^{ss}, \\ & y^{ss} = y^0(k + N|k) + \Delta y^{ss}, \\ & u^{ss} = u(k - 1) + \Delta u^{ss}, \\ & u_{min} \leq u^{ss} \leq u_{max}, \\ & y_{min} \leq y^{ss} \leq y_{max}, \end{aligned} \quad (4.8)$$

lub równoważnie, eliminując zmienne  $\Delta y^{ss}$ ,

$$\begin{aligned} \min_{\Delta u^{ss}} & \left\{ -q^T \mathbf{G} \Delta u^{ss} + p^T \Delta u^{ss} \right\} \\ \text{z ogr. : } & u^{ss} = u(k - 1) + \Delta u^{ss}, \\ & u_{min} \leq u^{ss} \leq u_{max}, \\ & y_{min} \leq y^0(k + N|k) + \mathbf{G} \Delta u^{ss} \leq y_{max}. \end{aligned} \quad (4.9)$$

Po rozwiązaniu zadania SSTO wartość  $\hat{y}^{ss} = y^0(k + N|k) + \mathbf{G} \hat{u}^{ss}$  przekazywana jest do regulatora MPC jako wartość zadana  $y^{zad}(k)$  w bieżącym okresie próbkowania, w razie potrzeby (gdy liczba sterowań przekracza liczbę wyjść regulowanych) przekazywana jest też wartość  $\hat{u}^{ss}$  jako zadana wartość wejść procesu  $u^{zad}(k)$ .

Jeśli występują ograniczenia wartości wyjść i możliwa jest sytuacja pustego (sprzecznego) zbioru ograniczeń w zadaniu SSTO, to trzeba zrelaksować niektóre lub wszystkie ograniczenia wyjść, traktując je jako ograniczenia miękkie (chwilowo przekraczalne). Wówczas zadanie SSTO może być sformułowane w postaci, np. dla przypadku odpowiadającego (4.8),

$$\begin{aligned} \min_{\Delta u^{ss}, \Delta y^{ss}, \epsilon_m, \epsilon_M} & \left\{ -q^T \Delta y^{ss} + p^T \Delta u^{ss} + \rho_1^T \epsilon_m + \rho_2^T \epsilon_M \right\} \\ \text{z ogr. : } & \Delta y^{ss} = \mathbf{G} \Delta u^{ss}, \\ & y^{ss} = y^0(k + N|k) + \Delta y^{ss}, \\ & u^{ss} = u(k - 1) + \Delta u^{ss}, \\ & u_{min} \leq u^{ss} \leq u_{max}, \\ & -\epsilon_m + y_{min} \leq y^{ss} \leq y_{max} + \epsilon_M, \\ & \epsilon_m \geq 0, \quad \epsilon_M \geq 0. \end{aligned} \quad (4.10)$$

#### 4.2.2. Zadania SSTO dla regulacji MPC z modelami nieliniowymi

W przypadku *regulacji MPC wykorzystującej model nieliniowy*, jedynie przygotowanie danych dla zadania SSTO jest nieco bardziej rozbudowane. Problemem jest tu bowiem wyznaczenie stanu ustalonego wyjścia odpowiadającego ostatnio wyznaczonej (w poprzednim kroku), aktualnej wartości sterowania  $u(k-1)$ . Powodem jest fakt, że jako reguła wykorzystujemy tu jako trajektorię początkową sterowań  $\mathcal{Y}^0(k)$  trajektorię (3.173), a więc trajektorię zmienną na horyzoncie predykcji. Stąd wartość  $y^0(k+N|k)$ , nawet jeśli ustalona – co wystąpi jeśli odcinek horyzontu predykcji od chwili  $(k+N_u|k)$  do chwili  $(k+N|k)$  jest odpowiednio długi – nie będzie odpowiadała wartości sterowania  $u(k-1)$ . Rozwiązaniem jest tu wyliczenie, na początku działania algorytmu MPC w  $k$ -tej chwili czasu, trajektorii początkowej wyjścia, oznaczymy ją przez  $\mathcal{Y}^{00}(k)$ , dla trajektorii początkowej sterowań takiej, jak stosowana w algorytmach MPC z modelem liniowym, tzn. stałej trajektorii początkowej, którą oznaczymy przez  $\mathcal{U}^{00}(k)$ ,

$$\mathcal{U}^{00}(k) = \begin{bmatrix} u^0(k|k) \\ u^0(k+1|k) \\ \vdots \\ u^0(k+N-1|k) \end{bmatrix} = \begin{bmatrix} u(k-1) \\ u(k-1) \\ \vdots \\ u(k-1) \end{bmatrix}. \quad (4.11)$$

Przy odpowiednio długim horyzoncie predykcji, uzyskana na jego końcu wartość  $y^0(k+N|k)$  wyliczana dla trajektorii sterowań (4.11) będzie odpowiadała stanowi ustalonemu ze sterowaniem  $u(k-1)$ .

Drugim elementem odróżniającym formułowanie zadania SSTO w przypadku algorytmów MPC z modelem nieliniowym jest zależność wzmocnienia od punktu pracy. Dlatego też macierz wzmocnień będzie teraz zależała od punktu pracy (z wartością wejścia  $u(k-1)$ ),  $\mathbf{G} = \mathbf{G}(k)$ .

Reasumując, na początku działania algorytmu MPC-NO, czy algorytmów z linearyzacjami (MPC-NSL, MPC-NPL(T)), należy wyznaczyć z wykorzystaniem modelu nieliniowego trajektorię początkową wyjść  $\mathcal{Y}^{00}(k)$  odpowiadającą trajektorii początkowej sterowań (4.11) oraz macierz wzmocnień  $\mathbf{G}(k)$  odpowiadającą aktualnej chwili czasu  $k$  (sterowaniu  $u(k-1)$ ). Następnie zadania SSTO formułujemy identycznie jak poprzednio dla algorytmów z modelami liniowymi, jedynie wykorzystujemy jako  $y^0(k+N|k)$  wartość ostatnią trajektorii  $\mathcal{Y}^{00}(k)$ , a zamiast macierzy stałej  $\mathbf{G}$  macierz  $\mathbf{G}(k)$ , w ogólności zmienną z kroku na krok. Podkreślimy, że w ogólności  $y^0(k+N|k) \neq \mathbf{G}(k)u(k-1)$ , gdyż *wartość  $y^0(k+N|k)$  wyznaczana jest na modelu nieliniowym i z uwzględnieniem aktualnej estymaty zakłóceń niemierzonych*. Po rozwiązaniu zadania SSTO, przesyłamy do zadania opty-

malizacji MPC wyznaczone wartości zadane i przystępujemy do realizacji właściwego algorytmu MPC z modelem nieliniowym. Jeśli w kolejnych krokach wartość sterowania  $u(k-1)$  niewiele się zmienia, to można adaptację macierzy wzmocnień  $\mathbf{G}(k)$  przeprowadzać rzadziej, co kilka kroków.

### 4.3. Iteracyjna optymalizacja punktu pracy w warunkach niepewności

#### 4.3.1. Modelowanie obiektu regulowanego i zadanie optymalizacji w stanach ustalonych

Wychodząc od opisu obiektu regulowanego dla potrzeb warstwy optymalizacji (1.3) i dodając jawnie zależność modelu od pewnych parametrów  $\alpha$ , w stanie ustalonym mamy

$$0 = f_c(x_c, c, w, \alpha), \quad (4.12)$$

$$y = g_c(x_c, c). \quad (4.13)$$

Układ równań (4.12) jest w istocie uwikłanym zapisem pewnego odwzorowania  $\zeta$  określającego wartości stanu procesu  $x_c$  w zależności od ustalonych wartości wejść sterowanych  $c$ , niesterowanych  $w$  i parametrów modelu  $\alpha$ ,

$$x_c = \zeta(c, w, \alpha). \quad (4.14)$$

Zakładając jednoznaczność odwzorowania (4.14) – typową dla dobrze wybranych zmiennych wejściowych i wyjściowych oraz dla dobrze określonego modelu – można formalnie wyeliminować zmienne stanu z układu równań (4.12), (4.13), uzyskując w ten sposób odwzorowanie typu wejścia – wyjścia obiektu regulowanego w stanach ustalonych, czyli jego *model statyczny*  $F$ ,

$$y = F(c, w, \alpha), \quad (4.15)$$

gdzie

$$F(c, w, \alpha) = g_c(\zeta(c, w, \alpha), c). \quad (4.16)$$

Trzeba wyraźnie podkreślić, że jawna postać odwzorowania (4.14) nie zawsze jest możliwa do uzyskania w praktycznych zastosowaniach, podobnie jak jawna postać modelu (4.15) powstająca, z matematycznego punktu widzenia, drogą rozwikłania układu nieliniowych równań algebraicznych (4.12) i (4.13). Wówczas jednak, wartości zmiennych wyjściowych  $y$  mogą być wyliczone, dla każdego punktu  $(c, w, \alpha)$ , poprzez rozwikłanie (obliczenie) numeryczne układu równań (4.12) i wstawienie uzyskanej wartości  $x_c = \zeta(c, w, \alpha)$  do równań wyjść (4.13).

W dotychczasowych rozważaniach wyprowadziliśmy postać opisu modelu statycznego obiektu regulowanego wychodząc od równań jego dynamiki i przyrównując pochodne zmiennych stanu do zera. Jest to elegancki, poglądowy sposób pokazujący relacje między dynamiką a statyką. Jednak w poważnych zastosowaniach przemysłowych, np. w profesjonalnych pakietach programów do modelowania statyki, szczegółowe modele statyczne są budowane na podstawie praw fizykochemicznych wyprowadzanych dla stanów ustalonych, opisywanych złożonymi równaniami nieliniowymi czy nieliniowymi schematami iteracyjnymi zbiegającymi do poszukiwanych relacji równowagowych. Równania te, bądź schematy iteracyjne, prowadzą do wysokich dokładności modelowania, na ogół trudnych czy wręcz niemożliwych do osiągnięcia przy konstruowaniu użytecznych modeli dynamicznych obiektów. Przy tym niektóre, czy też prawie wszystkie zmienne wyjściowe występują często w sposób uwikłany, nie dają się jawnie wyrazić w postaci odpowiadającej równaniu (4.13). Wówczas, zamiast (4.12) i (4.13), mamy nieco ogólniejszą postać układu równań

$$0 = f_c(x_c, c, w, \alpha), \quad (4.17)$$

$$0 = g_{cy}(y, x_c, c). \quad (4.18)$$

Układ równań nieliniowych (4.17) i (4.18), z reguły o wysokiej wymiarowości (nierzadko setki czy więcej równań) musi być rozwiązywany odpowiednimi procedurami numerycznymi, zdolnymi radzić sobie, i to nie zawsze niezawodnie, z iteracyjnymi pętlami algebraicznymi. Takie bardzo złożone modele statyczne były dotąd zwykle opracowywane przede wszystkim dla celów symulacyjnych, z reguły z pomocą specjalizowanych i drogich pakietów oprogramowania, takich jak ASPEN+, ChemCAD, itp. Z powodu swej złożoności i nie zawsze pełnej gwarancji poprawnego wyniku (dla całego rozważanego obszaru zmienności wejść i parametrów procesu) modele takie w wielu przypadkach nie nadają się do wyznaczania optymalnych punktów pracy na bieżąco, w czasie rzeczywistym. Stanowią one natomiast dobrą podstawę do opracowania prostszych modeli o strukturze (4.15), modeli opracowywanych specjalnie dla zastosowań optymalizacji bieżącej (on-line). Takie uproszczone modele mogą być konstruowane jako pewne funkcyjne aproksymacje dużych zbiorów danych wejściowo-wyjściowych generowanych przez pełne modele symulacyjne, np. z użyciem aproksymacji metodą najmniejszych kwadratów dla estymacji parametrów. Dla bieżącego dostrajania modelu uproszczonego, z wykorzystaniem aktualnych danych rzeczywistych, pozostawia się jedynie pewną, możliwie małą liczbę kluczowych parametrów.

Trzeba podkreślić, że ciągły rozwój techniki komputerowej i algorytmów numerycznych umożliwia w coraz większej liczbie przypadków stosowanie

bardziej złożonych fizykochemicznych modeli statycznych do zastosowań optymalizacyjnych on-line. Tendencja ta z pewnością będzie kontynuowana.

Sterowanie optymalizujące stan ustalony punktu pracy nie oznacza, że wartości zadane dla regulatorów  $c$  są raz optymalnie ustalone i następnie przez bardzo długi czas utrzymywane na wyliczonych wartościach, niezależnie od zmian istotnych niesterowanych wielkości wejściowych czy cech samego obiektu. Taka sytuacja oznaczałaby stabilizację na pewnych wartościach średnich, być może w sensie średniej wartości kryterium ekonomicznego optymalnie wyznaczonych, i ma sens w sytuacji szybkozmiennych zakłóceń o stałych wartościach średnich. Sterowanie optymalizujące punkt prac polega natomiast na sukcesywnym dostrajaniu wartości zadanych dla regulatorów do wartości optymalnych zmieniających się w wyniku zmian wejść niesterowanych i cech samego obiektu, zmian wolnych w stosunku do dynamiki obiektu regulowanego. Praktycznie istotna jest również sytuacja, kiedy wszystkie czy niektóre wejścia niesterowane są stosunkowo szybko zmienne, ale o wolno zmiennych wartościach średnich. Wówczas punkt pracy obiektu należy optymalnie dostrajać do tych wartości średnich.

Założymy, że podlegające optymalizacji statycznej kryterium ekonomiczne działania obiektu (funkcja celu) ma postać

$$J = Q(c, y), \quad (4.19)$$

gdzie  $c \in \mathbb{R}^{n_c}$  i  $y \in \mathbb{R}^{n_y}$ . Funkcja (4.19) wyraża cel ekonomiczny produkcji, zwykle zysk. Zakładając liniową zależność kosztów odpowiednich substancji od ich cen, funkcja celu przyjmuje prostą postać

$$Q(c, y) = - \sum_{j=1}^{n_J} p_j^c c_j + \sum_{j=1}^{m_J} p_j^y y_j, \quad (4.20)$$

gdzie  $p_j^c$  oznaczają ceny strumieni surowców i mediów energetycznych, zaś  $p_j^y$  ceny produktów wyjściowych ( $n_J$  oznacza liczbę strumieni surowców i mediów energetycznych,  $m_J$  liczbę produktów). Zmienne wyjściowe  $y$  mogą w ogólności reprezentować intensywność produkcji zarówno produktów finalnych, jak półproduktów czy nawet substancji odpadowych. W ostatnim przypadku ceny będą miały ujemne znaki. Mogą również występować, i często występują, składowe wektora  $y$  nie wchodzące do funkcji celu, ale istotne z punktu widzenia parametrów jakościowych produktów i stąd wchodzące w sformułowania ograniczeń, jak np. stężenie zanieczyszczeń czy czystego składnika w strumieniu produktu. Stąd sytuacja  $m_J < n_y$  jest możliwa i często spotykana.

Wartości zadane regulatorów  $c$  (punkty pracy obiektu) są zmiennymi decyzyjnymi warstwy optymalizacji i są w praktyce zawsze ograniczone, zbiór tych ograniczeń oznaczamy będziemy przez  $C$ ,

$$c \in C = \{c \in \mathbb{R}^{n_c} : g(c) \leq 0\}, \quad (4.21)$$

gdzie  $g : \mathbb{R}^{n_c} \mapsto \mathbb{R}^{r_c}$  oznacza wektor funkcji ograniczeń. Składowe wektora  $g$  odpowiadają najczęściej prostym ograniczeniom zakresów,  $c_j \geq c_{jmin}$  i  $c_j \leq c_{jmax}$ , mogą jednakże również reprezentować ograniczenia bardziej złożone definiowane przez kilka składowych wektora  $c$ , np. ograniczenie natężenia przepływu medium grzejnego zasilającego kilka niezależnie sterowanych układów grzewczych.

Zadanie optymalizacji bieżącej punktów pracy wykonywane jest z reguły w obecności zakłóceń modelowanych jedynie w sposób przybliżony oraz w sytuacji niedokładnego modelu. Dlatego też dla sformułowania tego zadania rozróżniamy między dostępnym modelem obiektu regulowanego w stanach ustalonych (4.15) a z założenia *nieznanym dokładnie odwzorowaniem rzeczywistym*  $F_*$

$$y = F_*(c, w). \quad (4.22)$$

Zadanie bieżącej optymalizacji wartości ustalonych punktów pracy w warunkach niepewności nazwiemy *zadaniem sterowania optymalizującego* punkty pracy (ZSO) i sformułujemy w następującej postaci:

*Dla aktualnych wartości wejść niesterowanych w wyznaczyć optymalne wartości zadane  $c$  dla zmiennych regulowanych minimalizujące funkcję celu  $Q(c, y)$ , przy spełnieniu ograniczeń nierównościowych  $g(c) \leq 0$ , gdzie  $y$  jest pomiarem wyjść w stanie ustalonym odpowiadającym zastosowanym wartościom  $c$ , tzn.  $y = F_*(c, w)$ .*

Sformułowanie matematyczne przedstawionego zadania ZSO jest stąd następującej postaci:

$$\begin{aligned} & \min Q(c, y) \\ & \text{z ogr. : } y = F_*(c, w), \\ & \quad g(c) \leq 0. \end{aligned} \quad (4.23)$$

Zadanie ZSO jest oczywiście w praktyce nierealizowalne ze względu na nieznajomość odwzorowania rzeczywistego  $y = F_*(c, w)$ , jest ono jedynie sformulowaniem pewnego celu, do którego powinniśmy dążyć.

W sytuacjach, kiedy model obiektu regulowanego  $F(c, w, \alpha)$  jest dostatecznie dokładny, zadanie ZSO można zastąpić odpowiadającym mu zadaniem optymalizacji wykorzystującym zamiast odwzorowania rzeczywistego

$y = F_*(c, w)$  odwzorowanie modelowe  $y = F(c, w, \alpha)$ . Jednakże, w odwzorowaniu tym trzeba w odpowiedni sposób określać wartości wejść niesterowanych  $w$ . Są to wielkości wolnozmienne, które w wielu przypadkach możemy mierzyć lub z określoną dokładnością estymować. Stąd najczęściej stosowanym sposobem reprezentacji wolnozmiennych wejść niesterowanych w aktualnej chwili (w której dokonujemy optymalizacji) są deterministyczne wartości  $w_d$  odpowiadające aktualnym pomiarom bezpośrednim czy pośrednim, wartościom średnim, itp. Taki sposób traktowania wejść niesterowanych prowadzi do odpowiadającego zadaniu (4.23), realizowalnego deterministycznego zadania optymalizacji modelowej (ZOM), postaci

$$\begin{aligned} \min Q(c, y) \\ \text{z ogr. : } y = F(c, w_d, \alpha), \\ g(c) \leq 0. \end{aligned} \quad (4.24)$$

Jest to typowe zadanie optymalizacji statycznej (programowania matematycznego), nieliniowe przy nieliniowym modelu obiektu regulowanego, ograniczeń czy funkcji celu.

W przedstawionych powyżej rozważaniach sformułowaliśmy zadanie sterowania optymalizującego punkt pracy obiektu, zadanie ZSO (4.23). Następnie rozważyliśmy sytuację, w której dysponujemy na tyle wiarygodnym modelem statyki, że zadanie sterowania optymalizującego można realizować przez rozwiązanie zadania optymalizacji opartego na modelu, pod warunkiem odpowiednio wiarygodnej struktury, parametrów modelu oraz wartości wejść niesterowanych. W dalszej części niniejszego rozdziału zajmujemy się przede wszystkim sytuacją, kiedy model procesu i/lub reprezentacja wejść niesterowanych są na tyle mało dokładne, że rozwiązanie odpowiedniego statycznego zadania optymalizacji modelowej prowadzi do uzyskania punktów pracy istotnie różnych od naprawdę optymalnych, odpowiadających rozwiązaniu zadania ZSO. Nie jesteśmy tu jednakże bez szans na poprawę, ale pod warunkiem, że zmienność procesu niedokładnie modelowanego i zmienność niedokładnie znanych wejść niesterowanych są dostatecznie wolnozmienne. Dokładniej, są na tyle wolne, że mogą być uznane za niezmienne w dłuższych okresach czasu, wielokrotnie dłuższych od pojedynczego czasu ustalania się przebiegów przejściowych w procesie regulowanym (po skokowej zmianie wejść, tzw. *settling time*). Wówczas można zastosować algorytmy oparte na iteracyjnej poprawie stanu ustalonego, bazujące na pomiarach kolejnych stanów ustalonych obiektu przy założeniu, że w czasie trwania takiego procesu iteracyjnego (kilku kolejnych iteracji) niedokładnie znany obiekt nie podlega zmianom [15].

Jedynym mechanizmem, jakim dysponuje warstwa optymalizacji dla uzyskania punktów pracy lepszych niż wynikające z jednokrotnego rozwiązania zadania ZOM (4.24), jest wykorzystanie bieżących pomiarów wyjść  $y$  obiektu, tj. pomiarowego sprzężenia zwrotnego od wyjść i postępowanie kilkietapowe, iteracyjne. Prowadzi to do *iteracyjnych algorytmów bieżącej optymalizacji punktów pracy*, dla których będziemy też używać nieco krótszego określenia: *iteracyjne algorytmy sterowania optymalizującego* – gdyż algorytmy te powinny dążyć do uzyskania punktów pracy będących rozwiązaniami zadania sterowania optymalizującego ZSO. Poświęcimy główną uwagę najistotniejszej klasie takich algorytmów, opartej na *metodzie powtarzanej modyfikowanej optymalizacji* (PMO), w literaturze anglojęzycznej znanej jako metoda ISOPE (*Integrated System Optimization and Parameter Estimation* [15, 144]. Zanim to tego przejdziemy, przedstawimy stosunkowo prosty algorytm stosowalny dla sytuacji z dobrze określonym modelem i z niepewnością głównie w wartościach estymat istotnych zakłóceń.

#### 4.3.2. Algorytm powtarzanej optymalizacji z korektą estymat zakłóceń

Rozważymy teraz szczególną sytuację, a mianowicie założymy, że:

- Model obiektu regulowanego (jego struktura, parametry) jest wystarczająco precyzyjny dla potrzeb optymalizacji,
- Niedokładność modelowania zawarta jest przede wszystkim w estymatach  $w_d$  wartości wejść niesterowanych (zakłóceń)  $w$ , których liczba  $n_w = \dim w$  jest równa lub większa od liczby wyjść  $n_y = \dim y$ .

Jeśli  $n_w > n_y$ , to wybieramy spośród  $n_w$  wejść niesterowanych  $n_y$  najbardziej istotnych, tj. najsilniej wpływających na wartości wyjść  $y$ , a pozostałe pomijamy w aktualnych rozważaniach – traktując je jako stałe i mniej interesujące parametry  $\alpha$  modelu. Stąd mamy  $n_w = n_y$ .

Przyjmijmy teraz, że dla aktualnej wartości estymaty istotnych zakłóceń  $w_d = w_d^{(0)}$  rozwiązano zadanie optymalizacji modelowej ZOM, oznaczając rozwiązanie przez  $\hat{c}_m^{(0)}$ . Punkt pracy  $\hat{c}_m^{(0)}$  zastosowano następnie do obiektu regulowanego i zmierzono odpowiadające mu wartości wyjść

$$y^{(0)} = F_*(\hat{c}_m^{(0)}, w).$$

Punktowi pracy  $\hat{c}_m^{(0)}$  odpowiada modelowo wyliczona wartość wyjść

$$y_m^{(0)} = F(\hat{c}_m^{(0)}, w_d^{(0)}, \alpha).$$

Jeśli modelowanie obiektu byłoby idealne (w tym estymaty zakłóceń też idealne), to mielibyśmy równość

$$y_m^{(0)} = y^{(0)}.$$

Niedokładność estymat  $w_d^{(0)}$  powoduje jednak, że  $y_m^{(0)} \neq y^{(0)}$ .

Możliwość korekty estymaty zakłóceń wynika z możliwości wykorzystania informacji zawartej w różnicy wartości wyjść rzeczywistych (pomiar) i wyznaczonych z modelu. Mianowicie, można wyliczyć skorygowaną wartość  $w_d^{(1)}$  estymaty zakłóceń z równania

$$F(\hat{c}_m^{(0)}, w_d^{(1)}, \alpha) = y^{(0)}. \quad (4.25)$$

W tym celu należy rozwiązać, w ogólności nieliniowy, układ  $n_y$  równań (4.25) względem  $w_d^{(1)}$ . W praktyce może wystarczyć policzenie zadania prostszego, pewnego i szybkiego w rozwiązaniu, a mianowicie rozwiązanie jedynie liniowego przybliżenia układu równań (4.25)

$$F'_w(\hat{c}_m^{(0)}, w_d^{(0)}, \alpha)[w_d^{(1)} - w_d^{(0)}] = y^{(0)} - y_m^{(0)}. \quad (4.26)$$

Zadanie optymalizacji modelowej ZOM może być teraz ponownie rozwiązane, z estymatą istotnych zakłóceń  $w_d = w_d^{(1)}$ , itd.

Reasumując, algorytm *powtarzanej optymalizacji z korekcją estymat zakłóceń* składa się z dwóch następujących, iteracyjnie powtarzanych etapów:

- Etapu rozwiązania modelowego zadania optymalizacji ZOM z aktualną wartością estymaty istotnych zakłóceń, zastosowania uzyskanego punktu pracy do obiektu regulowanego i pomiaru wartości wyjść  $y$ , w stanie ustalonym po zaniknięciu przebiegów przejściowych.
- Etapu korekcji estymaty istotnych zakłóceń  $w_d$  w oparciu o pomiar wyjść, poprzez rozwiązanie układu równań liniowych (4.26), lub oryginalnego układu równań nieliniowych (4.25).

Przedstawiony sposób postępowania ma strukturę powtarzanych adaptacji parametrów i optymalizacji – etap adaptacji parametrów modelu stanu ustalonego realizowany jest tutaj jako etap stosunkowo prostej *deterministycznej korekcji estymat istotnych zakłóceń*  $w_d$ . Jest to więc algorytm *suboptymalny*, o stopniu nieoptymalności zależnym od jakości stosowanego modelowania obiektu regulowanego. Przy czym najistotniejsza jest tu dokładność strukturalna postaci odwzorowania  $F(c, w, \alpha)$ , a nie precyzja wyznaczenia początkowej wartości estymaty istotnych zakłóceń, która będzie potem modyfikowana. Zauważmy ponadto, że przedstawiony sposób

korekty może prowadzić do wartości estymat  $w_d^{(i)}$  korygujących nie tylko niedokładności estymacji odpowiadających im zakłóceń  $w$ , ale i innych, parametrycznych i strukturalnych niedokładności modelu będących powodem różnic w wartościach wyjść obiektu regulowanego mierzonych i wartościach wyliczanych za pomocą modelu.

### 4.3.3. Metoda powtarzanej modyfikowanej optymalizacji (PMO)

Założyliśmy powyżej, że pojedynczy proces iteracyjnego wyznaczania (korekty) punktu pracy  $c$  trwa na tyle krótko, iż można założyć stałość wartości wejść niesterowanych  $w$  w trakcie trwania tych iteracji, jak również przez pewien czas po ich zakończeniu. Stąd w rozważaniach niniejszego rozdziału zakładać będziemy stałość wartości reprezentacji deterministycznej  $w_d$  zakłóceń w czasie pojedynczego procesu iteracyjnego, całą ewentualną zmienność modelu umieszczając w wektorze parametrów  $\alpha$ . Nie zmniejsza to ogólności rozważań, a prowadzi do uproszczenia zapisu poprzez możliwość pominięcia we wzorach zmiennych  $w$  i  $w_d$ , będących w tej sytuacji stałymi parametrami. Dlatego też pisać dalej będziemy  $F_*(c)$  zamiast  $F_*(c, w)$  oraz  $F(c, \alpha)$  zamiast  $F(c, w_d, \alpha)$ , zaś dla oznaczenia rozwiązań zadań ZSO i ZOM odpowiednio  $\hat{c}_*$  i  $\hat{c}_m(\alpha)$  (z wyjątkiem sytuacji, gdy wypisanie jawne zależności od  $w$  czy  $w_d$  może być potrzebne).

W roku 1979 Roberts [125] zaproponował podejście iteracyjne prowadzące do punktu ściśle optymalnego, tj. punktu będącego rozwiązaniem zadania ZSO – mimo dysponowania jedynie przybliżonym modelem matematycznym  $F$  odwzorowania rzeczywistego  $F_*$ . Metoda ta znana jest przede wszystkim pod nazwą ISOPE (*Integrated System Optimization and Parameter Estimation*) [12, 153, 15, 144]. W rozdziale niniejszym przedstawimy na początku iteracyjną metodę sterowania optymalizującego będącą metodą ISOPE z pewną modyfikacją, polegającą na jawnym wprowadzeniu parametru addytywnego modelu (parametru przesunięcia modelu). Następnie opiszemy wariant dualny metody, wprowadzający skuteczne rozwiązanie kluczowego problemu estymacji pochodnych. Rozważaną metodę nazywać będziemy *metodą powtarzanej modyfikowanej optymalizacji* (PMO).

Można stosunkowo łatwo uzasadnić [143], że warunek

$$F'_c(\hat{c}_m(\alpha), \alpha) = (F'_*)'(\hat{c}_*) \quad (4.27)$$

jest warunkiem koniecznym uzyskania rzeczywistego punktu optymalnego  $\hat{c}_*$  z rozwiązania zadania optymalizacji opartego jedynie na modelu. Warunek ten to wymaganie równości, w punkcie optymalnym, pochodnych

rzeczywistego odwzorowania opisującego statykę obiektu  $F_*$  i jego modelu matematycznego  $F$ .

Załóżmy, że w aktualnym punkcie pracy  $c^i$  wartość pochodnej  $(F_*)'(c^i)$  odwzorowania  $F_*$  może zostać obliczona (estymowana), w praktyce będzie to oczywiście możliwe jedynie poprzez wykorzystanie bieżących pomiarów wartości wyjść. Okazuje się, że taka jedynie punktowa informacja o wartości pochodnej odwzorowania obiektowego może być wykorzystana do konstrukcji iteracyjnych algorytmów sterowania optymalizującego zbiegających do rzeczywistego punktu optymalnego. Dla przedstawienia idei tych algorytmów rozważmy ponownie zadanie optymalizacji modelowej ZOM (4.24), upraszczając formalnie zapis tego zadania poprzez eliminację zmiennych wyjściowych  $y$ , co prowadzi do równoważnej postaci

$$\begin{aligned} & \min Q(c, F(c, \alpha)), \\ & \text{z ogr. : } g(c) \leq 0. \end{aligned} \quad (4.28)$$

Oznaczmy optymalne rozwiązanie tego zadania przez  $\hat{c}_m(\alpha)$ . Odpowiadająca zadaniu (4.28) uproszczona postać zadania sterowania optymalizującego (4.23) jest następująca

$$\begin{aligned} & \min Q(c, F_*(c)) \\ & \text{z ogr. : } g(c) \leq 0, \end{aligned} \quad (4.29)$$

z optymalnym rozwiązaniem  $\hat{c}_*$ .

Dysponując w aktualnym punkcie pracy  $c^i$  wartością pochodnej  $(F_*)'(c^i)$ , rozważmy następujące zmodyfikowane zadanie optymalizacji modelowej (ZZOM)

$$\begin{aligned} & \min_c \{Q(c, F(c, \alpha) + a^i) - \lambda(c^i, a^i)^T c\} \\ & \text{z ogr. : } g(c) \leq 0, \end{aligned} \quad (4.30)$$

gdzie

$$\lambda(c^i, a^i)^T = Q'_y(c^i, F(c^i, \alpha) + a^i) \cdot [F'_c(c^i, \alpha) - (F_*)'(c^i)], \quad (4.31)$$

$$a^i = F_*(c^i) - F(c^i, \alpha). \quad (4.32)$$

Oznaczmy rozwiązanie ZZOM przez  $\hat{c}_{zm} = \hat{c}_{zm}(c^i, a^i)$ , zaś przez  $q_{zm}$  funkcję celu zadania (4.30),

$$q_{zm}(c, a, \lambda, \alpha) = Q(c, F(c, \alpha) + a) - \lambda(c, a)^T c. \quad (4.33)$$

Pochodna względem  $c$  funkcji celu zadania ZZOM jest równa

$$\begin{aligned} (q_{zm})'_c(c, a^i, \lambda^i, \alpha) &= Q'_c(c, F(c, \alpha) + a^i) + Q'_y(c, F(c, \alpha) + a^i) \cdot F'_c(c, \alpha) + \\ & - Q'_y(c^i, F(c^i, \alpha) + a^i) \cdot [F'_c(c^i, \alpha) - (F_*)'(c^i)]. \end{aligned} \quad (4.34)$$

Uwzględniając (4.32), w punkcie  $c = c^i$  pochodna ta wynosi

$$(q_{zm})'_c(c^i, a^i, \lambda^i, \alpha) = Q'_c(c^i, F_*(c^i)) + Q'_y(c^i, F_*(c^i)) \cdot (F_*)'(c^i), \quad (4.35)$$

czyli uzyskujemy równość

$$(q_{zm})'_c(c^i, a^i, \lambda^i, \alpha) = (q_*)'(c^i), \quad (4.36)$$

gdzie  $q_*$  to funkcji celu oryginalnego zadania ZSO,

$$q_*(c) = Q(c, F_*(c)). \quad (4.37)$$

Tak więc *pochodne funkcji celu zadań ZZOM i ZSO są w punkcie  $c=c^i$  równe* – wynik, który w istocie leży u podstaw konstrukcji postaci zmodyfikowanego zadania optymalizacji modelowej (4.30).

Wyobraźmy sobie teraz, że iteracje punktów pracy są prowadzone w taki sposób, że każde kolejne rozwiązanie  $\hat{c}_{zm}^i$  zadania ZZOM zostaje użyte jako następny punkt pracy  $c^{i+1}$  procesu,  $c^{i+1} = \hat{c}_{zm}^i$ , itd. Wówczas, jeśli ciąg  $\{c^i\}$  jest zbieżny do pewnego punktu  $\tilde{c}$ , to punkt ten spełnia  $\tilde{c} = \hat{c}_{zm}^i$ . Punkt  $\tilde{c}$  jest stąd jednocześnie punktem startowym  $c^i$  (w którym spełniony jest warunek (4.36)) i punktem optymalnym  $\hat{c}_{zm}^i$  zadania ZZOM. Jest więc punktem, w którym spełnione są warunki konieczne optymalności tego zadania. W przypadku nieaktywności ograniczeń (tj.  $g(\tilde{c}) < 0$ ) warunki te przyjmują znaną postać zerowania pochodnej

$$(q_{zm})'_c(\tilde{c}, \tilde{a}, \tilde{\lambda}, \alpha) = (q_*)'(\tilde{c}) = 0,$$

gdzie  $\tilde{a} = F_*(\tilde{c}) - F(\tilde{c}, \alpha)$ ,  $\tilde{\lambda} = \lambda(\tilde{c}, \tilde{a})$ . W przypadku aktywnego (i wypukłego) zbioru ograniczeń  $C$ , warunki konieczne optymalności zadania ZZOM w punkcie  $\tilde{c}$  można zapisać w postaci

$$(q_{zm})'_c(\tilde{c}, \tilde{a}, \tilde{\lambda}, \alpha) \cdot [c - \tilde{c}] \geq 0 \quad \text{dla } c \in C, \quad (4.38)$$

co wobec równości (4.36) daje od razu warunki konieczne optymalności zadania ZSO

$$(q_*)'(\tilde{c}) \cdot [c - \tilde{c}] \geq 0 \quad \text{dla } c \in C.$$

Reasumując, *pokazaliśmy, że jeśli w strukturze powtarzanej optymalizacji zastosujemy zadanie optymalizacji zmodyfikowane ZZOM zamiast zadania ZOM, to w przypadku zbieżności generowanego ciągu punktów pracy  $\{c^i\}$  granicą będzie punkt spełniający warunki konieczne optymalności oryginalnego zadania sterowania optymalizującego (ZSO).*

Przedstawiony pomysł iteracyjnego algorytmu sterowania optymalizującego punkt pracy to koncepcja metody *powtarzanej modyfikowanej optymalizacji* (PMO) [143], w stosunku do oryginalnej metody ISOPE nieco

zmodyfikowanej przez wprowadzenie parametru  $a^i$  (4.32). Modyfikacja ta powoduje zrelaksowanie warunku

$$F(c^i, \alpha^i) = F_*(c^i), \quad (4.39)$$

występującego w oryginalnej metodzie ISOPE. Co więcej, dzięki adaptacji parametru przesunięcia modelu  $a^i$  nie jest konieczna w każdej iteracji metody PMO adaptacja parametrów modelu  $\alpha^i$ . Adaptacja parametrów  $\alpha^i$  w każdej iteracji nie jest potrzebna w oryginalnej metodzie ISOPE jedynie wtedy, gdy funkcja  $Q$  jest addytywna i liniowa względem  $y$ , tzn. gdy

$$Q(c, y) = \tilde{Q}(c) + b^T y, \quad (4.40)$$

gdzie  $b \in \mathbb{R}^{n_y}$  jest wektorem parametrów. W przypadku tym nie ma też potrzeby wprowadzania wektora przesunięć modelu  $a^i$  (4.32) dla uzyskania fundamentalnej równości (4.36) w PMO. Sformułowania metod ISOPE i PMO stają się wówczas równoważne [143]. Struktura metody PMO przedstawiona jest na rysunku 4.3, gdzie blok estymacji parametrów  $\alpha$  oznaczono linią przerywaną, jako nie konieczny w każdej iteracji.

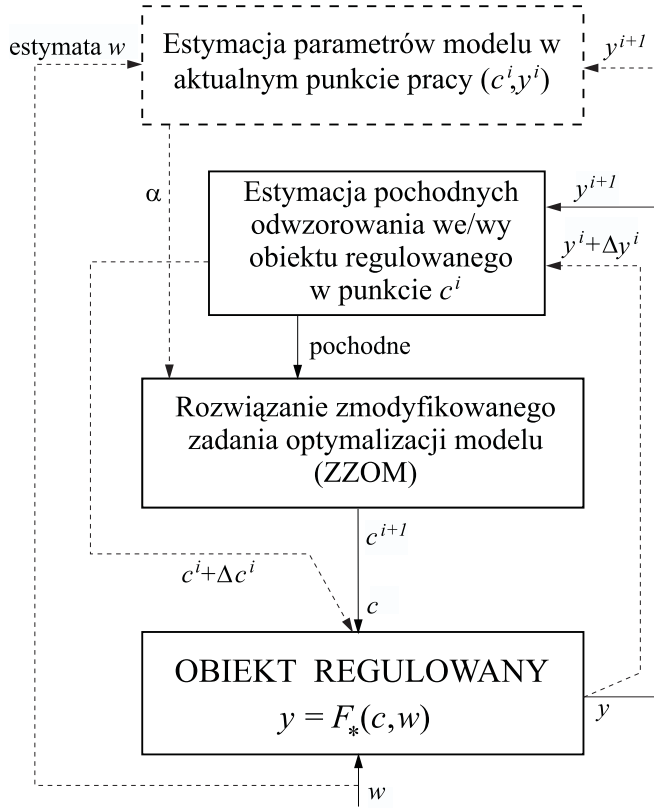
W dotychczasowych rozważaniach wprowadziliśmy zadanie ZZOM i uzasadniliśmy jego sens, nie podając jednakże formalnego wyprowadzenia zastosowanej modyfikacji funkcji celu. Wyprowadzenie to można uzyskać odpowiednio przekształcając warunki konieczne optymalności pierwotnego zadania sterowania optymalizującego (ZSO). Zadanie to, zob. (4.29), można zapisać formalnie w następującej równoważnej postaci

$$\begin{aligned} & \min_{c, \alpha} Q(c, F(c, \alpha) + a) \\ & \text{z ogr. : } F(c, \alpha) + a = F_*(c), \\ & \quad g(c) \leq 0, \end{aligned} \quad (4.41)$$

a następnie przekształcić do kolejnej równoważnej postaci

$$\begin{aligned} & \min_{c, c^i, a} \{Q(c, F(c, \alpha) + a) + \rho \|c^i - c\|^2\} \\ & \text{z ogr. : } F(c^i, \alpha) + a = F_*(c^i), \\ & \quad g(c) \leq 0, \\ & \quad c^i = c. \end{aligned} \quad (4.42)$$

W zadaniu (4.42) rozróżniono aktualny punkt pracy  $c^i$ , w którym wyliczana jest wartość  $a$  i od którego startuje procedura optymalizacji oraz zmienne decyzyjne tej optymalizacji  $c$ . Ponadto, dodano do funkcji celu człon regularyzujący (uwypuklający) ze współczynnikiem kary  $\rho > 0$ , który



Rysunek 4.3. Struktura metody powtarzanej modyfikowanej optymalizacji (PMO)

nie zmieniając istoty zadania odgrywa ważną rolę zarówno z teoretycznego (osłabienie warunków stosowalności), jak i z praktycznego punktu widzenia. Dla zachowania równoważności zadań (4.41) i (4.42) dodano w tym ostatnim ograniczenie równościowe  $c^i = c$ , którego spełnienie będzie warunkiem stopu algorytmu iteracyjnego PMO.

Zapisując teraz warunki konieczne optymalności dla zadania (4.42) i eliminując z nich mnożniki Lagrange'a odpowiadające ograniczeniu  $F(c^i, \alpha) + a = F_*(c^i)$  możemy te warunki przekształcić do następującej postaci

$$(q_a)'_c(c, a, \alpha)^T - \lambda(c^i, a) - 2\rho(c^i - c) + g'(c)^T \mu = 0, \quad (4.43a)$$

$$g(c) \leq 0, \quad \mu \geq 0, \quad \mu^T g(c) = 0, \quad (4.43b)$$

$$F(c^i, \alpha) + a - F_*(c^i) = 0, \quad (4.43c)$$

$$\text{gdzie} \quad c^i - c = 0, \quad (4.43d)$$

$$q_a(c, a, \alpha) = Q(c, F(c, \alpha) + a), \quad (4.44)$$

$$\lambda(c^i, a) = [F'_c(c^i, \alpha) - F'_*(c^i)]^T Q'_y(c^i, F(c^i, \alpha) + a)^T, \quad (4.45)$$

czyli dostaliśmy wyprowadzenie podanego wcześniej wzoru (4.31). Algorytmy PMO mogą być traktowane jako algorytmy iteracyjnego znajdowania punktu spełniającego powyższe warunki konieczne optymalności zadania ZSO.

### Algorytm podstawowy PMO

Podstawowymi elementami każdego z algorytmów PMO są: zmodyfikowane zadanie optymalizacji modelowej ZZOM oraz sposób wyznaczania aproksymacji pochodnych  $(F_*)'(c^i)$  potrzebnych do wyliczenia mnożników (4.45). Poniżej przytoczymy algorytm podstawowy PMO, odpowiadający algorytmowi ISOPE zaproponowanemu i wyczerpująco zbadanemu w latach osiemdziesiątych [125, 13, 12]).

Zmodyfikowane zadanie optymalizacji modelowej formułowane jest w następującej, rozszerzonej postaci:

$$\begin{aligned} \min_c \{ & Q(c, F(c, \alpha) + a^i) - \lambda(c^i, a^i)^T c + \rho \|c^i - c\|^2 \} \\ \text{z ogr. : } & g(c) \leq 0, \end{aligned} \quad (4.46)$$

$$\text{gdzie} \quad a^i = F_*(c^i) - F(c^i, \alpha) \quad (4.47)$$

Zadanie (4.46) różni się od zadania (4.30) jedynie dodatkowym członem regularyzującym  $\rho \|c - c^i\|^2$  dodanym do zmodyfikowanej funkcji celu. Zauważmy, że zależności (4.43a) i (4.43b) to dokładnie warunki konieczne optymalności zadania (4.46), a definicja (4.47) odpowiada warunkowi (4.43c).

Możemy teraz sformułować *algorytm podstawowy PMO*:

**Start.** Dane startowe: punkt początkowy  $c^0$ , współczynnik relaksacji  $k_c$  ( $0 < k_c \leq 1$ ), parametr testu stopu  $\epsilon > 0$ , ustaw  $i := 0$ . Zmierz wartości ustalone wyjść  $y^0 = F_*(c^0)$ .

**Krok 1.** Wyznacz wartość pochodnej  $(F_*)'(c^i)$  odwzorowania obiektu regulowanego w punkcie  $c^i$  – metody wyznaczania omówione zostaną dalej w kolejnej części rozdziału. Dokonaj adaptacji parametrów modelu  $\alpha$  (opcjonalnie).

**Krok 2.** Dla  $a^i = F_*(c^i) - F(c^i, \alpha)$  i  $\lambda(c^i, a^i)$  rozwiąż rozszerzone zmodyfikowane zadanie optymalizacji modelowej postaci (4.46), oznacz uzyskany punkt optymalny przez  $\hat{c}_{zm}^i = \hat{c}_{zm}(c^i, a^i)$ .

**Krok 3.** Wyznacz 
$$c^{i+1} := c^i + k_c(\hat{c}_{zm}^i - c^i). \quad (4.48)$$

**Krok 4.** Jeśli 
$$\|c^{i+1} - c^i\| \leq \epsilon, \quad (4.49)$$

to zakończ – optymalny punkt pracy jest znaleziony (**Stop**).

**Krok 5.** Zastosuj wartości zadane  $c^{i+1}$  do obiektu regulowanego, po ustaniu procesów przejściowych zmierz wartości wyjść  $y^{i+1} = F_*(c^{i+1})$ . Dokonaj adaptacji parametrów modelu  $\alpha$  (opcjonalnie).

Ustaw  $i := i + 1$  i idź do **Kroku 1**.

Jak łatwo zauważyć, przedstawiony algorytm jest skonstruowany jako "solver" warunków koniecznych optymalności (4.43a)-(4.43d) zadania sterowania optymalizującego w postaci (4.42). W każdej iteracji, po rozwiązaniu zadania (4.46) w kroku 2 spełnione są odpowiednio warunki (4.43a), (4.43b) i (4.43c), zaś algorytm kończy działanie po spełnieniu warunku (4.43d), co jest implikowane przez spełnienie kryterium stopu (4.49).

Cały algorytm PMO może być również traktowany jako algorytm znajdowania punktu stałego, ponieważ kolejne wartości  $c^i$  sterowań optymalizujących są iterowane w taki sposób, aby spełnić równanie (4.43d), tj. dążąc do spełnienia równości

$$\hat{c}_{zm}(c, a(c)) = c, \quad (4.50)$$

gdzie  $a(c) = F_*(c) - F(c, \alpha)$ . Wzór (4.48) jest prostą regułą iteracyjną wyznaczania punktu stałego odwzorowania  $\hat{c}_{zm}(c, a(c))$ , typu relaksacyjnego ze współczynnikiem relaksacji  $k_c$ . Warto zauważyć, że dla  $k_c = 1$  wzór ten staje się bezpośrednim podstawieniem  $c^{i+1} := \hat{c}_{zm}^i$ , a więc algorytmem tzw. iteracji prostej. Oprócz prostoty, wzór iteracyjny (4.48) charakteryzuje się ponadto bardzo ważną z punktu widzenia zastosowań cechą zachowania dopuszczalności kolejnego punktu pracy, w dominującym praktycznie przypadku wypukłego zbioru  $C$ :

*Jeśli  $C$  jest zbiorem wypukłym i punkt początkowy  $c^0 \in C$ , to  $c^1 \in C$  i każdy kolejny punkt  $c^i$  generowanej iteracyjnie wzorem (4.48) sekwencji jest też dopuszczalny, dla  $0 < k_c \leq 1$ .*

W celu zmniejszenia liczby iteracji opracowano dla metody ISOPE również bardziej złożone niż (4.48) formuły iteracyjne, typu quasi-newtonowskiego [153] – można je łatwo adaptować dla PMO. Algorytmy z tymi formułami cechują się jednakże nieco silniejszymi warunkami stosowalności i wymagają specjalnych zabiegów dla zachowania dopuszczalności kolejnych punktów pracy. W momencie pojawienia się algorytmu dualnego ISOPE znaczenie praktyczne tych formuł zmalało, dlatego też nie będziemy ich tu przytaczać.

Dowód zbieżności podstawowego algorytmu PMO ze strategią koordynacji (4.48), przy założeniu dokładnego obliczania wartości pochodnych  $(F_*)'(c^i)$  w każdej iteracji, można przeprowadzić analogicznie jak to zostało pokazane dla podstawowego algorytmu ISOPE w pracach [13, 12]. Przy rozsądnych założeniach dotyczących cech odwzorowań występujących w sformułowaniach zadań ZSO i ZOM wykazuje się [12], że dla każdej wartości współczynnika kary  $\rho$  większej od pewnej wartości progowej  $\rho_1 \geq 0$  (zależnej od cech konkretnego problemu) algorytm ma następujące cechy:

- Każdy z ciągu punktów  $c^i$  spełnia ograniczenia oraz, jeśli nie spełnia kryterium optymalności, to

$$q_*(c^{i+1}) < q_*(c^i), \quad i = 0, 1, 2, \dots \quad (4.51)$$

dla każdego  $k_c$ ,  $0 < k_{c \min} \leq k_c \leq k_{c \max} \leq 1$ , gdzie wartości  $k_{c \min}$  i  $k_{c \max}$  zależą od cech konkretnego problemu, zaś  $q_*(c^i)$  oznacza wartość rzeczywistej funkcji celu w punkcie  $c^i$ , zob. (4.37). Jest to rezultat ważny dla zastosowań praktycznych: *w każdej iteracji algorytmu następuje poprawa rzeczywistego kryterium oceny efektywności procesu.*

- Istnieje co najmniej jeden punkt skupienia ciągu  $\{c^i\}$  oraz każdy z takich punktów spełnia zależność (4.50), czyli spełnia warunki konieczne optymalności zadania sterowania optymalizującego punkt pracy.

### Estymacja pochodnych odwzorowania wyjść obiektu regulowanego

Kluczowym elementem metody PMO, jak i metody ISOPE, jest konieczność uzyskania w każdej iteracji algorytmu wartości pochodnej  $(F_*)'(c^i)$  rzeczywistego odwzorowania wejścia-wyjścia obiektu regulowanego, w aktualnym punkcie pracy  $c^i$ . Pochodna ta jest potrzebna dla wyznaczenia określonego wzorem (4.31) mnożnika  $\lambda(c^i, a^i)$ , niezbędnego dla sformułowania zadania ZZOM. Sposób wyznaczania wartości  $(F_*)'(c^i)$  jest kluczowy dla efektywności algorytmów PMO (ISOPE), definiuje klasy tych algorytmów. Przyczyną jest fakt, że czas i koszt poświęcony obliczeniu wartości przybliżenia pochodnej w trakcie każdej iteracji może być dominujący. Tak było

właśnie w koncepcyjnie najprostszym i historycznie pierwszym sposobie zaproponowanym przez Robertsa i stosowanym dalej w wielu wczesnych pracach [125, 126, 13, 12, 153]. Sposób ten polega na wyznaczaniu numerycznych przybliżeń kolejnych pochodnych cząstkowych techniką różnic skończonych, w oparciu o  $n_c$  ( $n_c = \dim c$ ) kolejno generowanych odchyłeń wartości zadanych regulatorów od aktualnego punktu  $c^i$  (w kierunkach poszczególnych osi układu współrzędnych) i odpowiadających im pomiarów wartości wyjść. Stosowanie takiego algorytmu wyznaczania pochodnych oznacza  $n_c$  dodatkowych, dynamicznych procesów przejściowych w obiekcie regulowanym w każdej iteracji algorytmu, dodatkowo poza pojedynczym procesem przejściowym związanym ze zmianą punktu pracy z  $c^{i-1}$  do  $c^i$ . Jest to więc procedura kosztowna. Dlatego też podejmowano badania w celu opracowania sposobów bardziej efektywnych, czy też innej organizacji algorytmu ISOPE umożliwiającej efektywniejsze szacowanie pochodnych. Badania poszły w dwóch kierunkach, które można scharakteryzować następująco:

- Dążenie do znalezienia sposobu aproksymacji pochodnych jedynie z pomiarów wyjść w stanach ustalonych, lecz bardziej efektywnie niż w omówionym wyżej algorytmie stosującym dodatkowe  $n_c$  odchyłeń od aktualnego punktu pracy.
- Obliczanie pochodnych na podstawie liniowych modeli dynamicznych procesu regulowanego, identyfikowanych w otoczeniu każdego z kolejnych punktów pracy  $c^i$ .

Pierwszy z wymienionych kierunków poszukiwań doprowadził do sformułowania tzw. dualnego algorytmu ISOPE [14]. Będący jego odpowiednikiem dualny algorytm PMO omówimy dalej, w punkcie następnym.

Drugie podejście polega na wykorzystaniu idei ekstrakowania punktowej informacji o pochodnej charakterystyki statycznej z identyfikowanego on-line modelu dynamicznego obiektu regulowanego, idei wywodzącej się od oryginalnego pomysłu Bambergera i Isermanna [5]. Zhang i Roberts [164] zaproponowali zastosowanie tego podejścia w algorytmie ISOPE do estymacji pochodnej  $(F_*)'(c^i)$ . Postępowanie jest następujące: w każdej iteracji obiekt regulowany jest początkowo pobudzany przez pewien odcinek czasu dodatkowym sygnałem dynamicznym (typu PRBS) dodawanym do stałej wartości zadanej  $c^i$ . Na podstawie pomiarów wyjścia identyfikowany jest następnie liniowy model dynamiczny obiektu regulowanego wybraną procedurą identyfikacji, odpowiadając aktualnemu stanowi obiektu. Następnie z modelu dynamicznego wyliczany jest model statyczny, a stąd poszukiwana informacja o jego pochodnej. Przedstawiony sposób wydobywania informacji statycznej z modelu dynamicznego wydaje się atrakcyjny, ale

z procesem identyfikacji związane są trudności, w tym przede wszystkim z dokładnością uzyskiwanych pochodnych odwzorowania statycznego. Po pierwsze, aby uzyskać wystarczająco dokładne wyniki identyfikacji proces identyfikowany musi być odpowiednio pobudzany, gdyż trudno założyć, że bierny eksperyment identyfikacyjny oparty na danych zgromadzonych w trakcie dynamicznego procesu przejściowego przy przechodzeniu między ostatnimi punktami pracy,  $c^{i-1}$  i  $c^i$ , będzie wystarczający. Po drugie, procedury identyfikacji bazujące na zbiorach danych dynamicznych są z reguły konstruowane w celu uzyskiwania liniowych modeli dynamicznych jak najlepiej odzwierciedlających lokalną *dynamikę* procesu, a nie jego własności *statyczne*. Ta właśnie trudność została zauważona już w oryginalnych badaniach Bambergera i Isermanna [5], którzy uznali ją jako szczególnie istotną w sytuacji pomiarów zaszumionych i dla rozważanego przez siebie algorytmu optymalizacji opracowali specjalną procedurę aktywnej identyfikacji bazującą na technice korelacyjnej.

Reasumując, estymacja pochodnej  $(F_*)'(c^i)$  modelu statycznego obiektu regulowanego na podstawie liniowego modelu dynamicznego identyfikowanego on-line w każdej iteracji algorytmu PMO (czy ISOPE) jest wartą uwagi alternatywą, jednakże nie jest to sposób łatwy w realizacji. Szczególnie w sytuacji zaszumionych pomiarów należy dążyć do wykorzystywania jedynie pomiarów stanów ustalonych, przy których szумы mogą być dobrze odfiltrowane. Efektywny sposób estymacji pochodnej spełniający powyższy postulat został zaproponowany w przedstawionym dalej algorytmie dualnym.

#### 4.3.4. Algorytm dualny PMO (algorytm DPMO)

Własności użytkowe algorytmów PMO (ISOPE) zależą w dominującym stopniu od efektywności estymacji pochodnej  $(F_*)'(c^i)$  odwzorowania obiektu regulowanego. Zostałyby one radykalnie poprawione, gdyby wyeliminować konieczność generacji dodatkowych punktów pracy potrzebnych jedynie dla estymacji pochodnych, czy też przynajmniej istotnie zmniejszyć ich liczbę. Myśl o wykorzystaniu w celu estymacji pochodnych punktów pracy wyznaczanych w poprzednich iteracjach algorytmu jest frapująca. Jednakże, z teoretycznego punktu widzenia, punkty  $c^i$ ,  $i = 0, 1, \dots$  generowane w kolejnych iteracjach algorytmu podstawowego mogą nie zawierać i często nie zawierają informacji wystarczającej do estymacji pochodnych, tzn. nie są rozmieszczone w przestrzeni  $\mathbb{R}^{n_c}$  w sposób umożliwiający taką estymację. Stąd pierwsze próby czynione w tym kierunku nie były udane, zob. np. [85]. Dopiero pomysł zaprezentowany w pracy [14] okazał się skuteczny.

Zdając sobie sprawę, że tylko *pasywne* wykorzystywanie informacji zgromadzonej w trakcie iteracji algorytmu podstawowego nie zapewnia wystarczającej informacji do aproksymacji wartości pochodnych  $(F_*)'(c^i)$ , zaproponowano w [14] sposób *aktywnego* gromadzenia potrzebnych informacji. Sformułowano tam algorytm ISOPE w taki sposób, że w każdej iteracji punkt następny  $c^{i+1}$  wyznaczany jest w sposób łączący w sobie jednocześnie realizację zadania zmodyfikowanej modelowej optymalizacji i zadania właściwego rozmieszczenia w przestrzeni  $\mathbb{R}^{n_c}$ , w stosunku do punktów poprzednich, tak aby umożliwić poprawną estymację pochodnej w następnej,  $(i + 1)$ -szej iteracji. Jak łatwo zauważyć, jest to zgodne z wprowadzonym przez Feldbauma [39] określeniem *sterowania dualnego*. Stąd też, uzyskany algorytm sterowania optymalizującego nazwano *algorytmem dualnym ISOPE (algorytmem ISOPED)* [14]. Przedstawimy poniżej zaprezentowaną tam ideę estymacji pochodnych odwzorowania wejście-wyście obiektu regulowanego, a następnie sformułujemy wykorzystujący tę ideę *algorytm dualny PMO* (algorytm DPMO).

Algorytm ten, jak pokażemy poniżej, jest w istocie algorytmem *zintegrowanej modyfikowanej optymalizacji i estymacji gradientu*.

Założmy, że dysponujemy  $n_c + 1$  ( $n_c = \dim c$ ) punktami  $c^i, c^{i-1}, \dots, c^{i-n_c}$  takimi, że wszystkie wektory

$$s^{ik} = c^{i-k} - c^i, \quad k = 1, \dots, n_c, \quad (4.52)$$

są wzajemnie liniowo niezależne, tzn. macierz

$$\mathbf{S}^i = [s^{i1} \ s^{i2} \ \dots \ s^{in_c}]^T \quad (4.53)$$

jest nieosobliwa. Pochodną kierunkową  $DF_{*j}(c^i; s^{ik})$   $j$ -tego wyjścia obiektu regulowanego w punkcie  $c^i$  ( $y_j^i = F_{*j}(c^i)$ ) w kierunku  $s^{ik}$  definiuje się następująco (zob. np. [81]):

$$DF_{*j}(c^i; s^{ik}) = \lim_{\beta \rightarrow 0} \frac{F_{*j}(c^i + \beta \frac{s^{ik}}{\|s^{ik}\|}) - F_{*j}(c^i)}{\beta}. \quad (4.54)$$

Jeśli funkcja  $F_{*j}$  ma w otoczeniu punktu  $c^i$  ciągle pochodne cząstkowe, to pochodna kierunkowa (4.54) związana jest z gradientem funkcji w tym punkcie zależnością (zob. np. [81])

$$DF_{*j}(c^i; s^{ik}) = \frac{1}{\|s^{ik}\|} (s^{ik})^T \nabla F_{*j}(c^i), \quad (4.55)$$

którą możemy zapisać w postaci

$$\|s^{ik}\| DF_{*j}(c^i; s^{ik}) = (s^{ik})^T \nabla F_{*j}(c^i).$$

Zapisując  $n_c$  powyższych równań, tj. dla wszystkich  $n_c$  kierunków  $s^{ik}$ ,  $k = 1, \dots, n_c$ , w postaci macierzowej dostajemy

$$\mathbf{S}^i \nabla F_{*j}(c^i) = \begin{bmatrix} \|s^{i1}\| DF_{*j}(c^i; s^{i1}) \\ \vdots \\ \|s^{in_c}\| DF_{*j}(c^i; s^{in_c}) \end{bmatrix}, \quad j = 1, \dots, n_y. \quad (4.56)$$

Oznaczmy  $\beta = \gamma \|s^{ik}\|$ . Wówczas definicja (4.54) przyjmuje postać równoważną

$$DF_{*j}(c^i; s^{ik}) = \lim_{\gamma \|s^{ik}\| \rightarrow 0} \frac{F_{*j}(c^i + \gamma s^{ik}) - F_{*j}(c^i)}{\gamma \|s^{ik}\|}.$$

Jeśli punkty  $c^{i-k}$  są położone dostatecznie blisko  $c^i$ , tzn. skalary  $\|s^{ik}\|$  są dostatecznie małe, to (przyjmując  $\gamma = 1$ ) uzyskujemy następujące przybliżenie

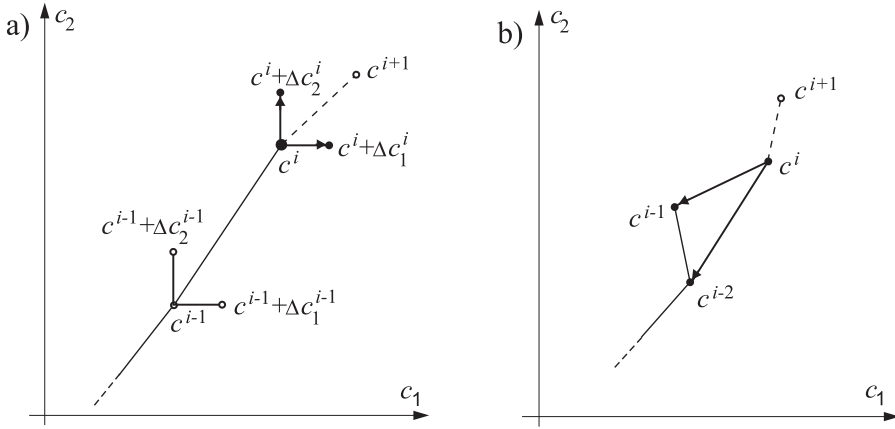
$$\|s^{ik}\| DF_{*j}(c^i; s^{ik}) \cong F_{*j}(c^i + s^{ik}) - F_{*j}(c^i).$$

Zależność (4.56) może być wówczas przedstawiona w przybliżonej postaci

$$\mathbf{S}^i \nabla F_{*j}(c^i) \cong \begin{bmatrix} F_{*j}(c^{i-1}) - F_{*j}(c^i) \\ \vdots \\ F_{*j}(c^{i-n_c}) - F_{*j}(c^i) \end{bmatrix}, \quad j = 1, \dots, n_y. \quad (4.57)$$

W algorytmie DPMO kolejne punkty  $c^i$  generowane są w taki sposób, aby układ równań liniowych (4.57) mógł być wykorzystany w każdej iteracji do wyliczenia aproksymacji pochodnych  $(F_{*j})'(c^i) = \nabla F_{*j}(c^i)^T$ ,  $j = 1, \dots, n_y$ . Na rys. 4.4 b) zilustrowano, dla przypadku dwuwymiarowego ( $n_c = 2$ ), wektory  $s^{i1} = c^{i-1} - c^i$  i  $s^{i2} = c^{i-2} - c^i$  tworzące macierz  $\mathbf{S}^i = [s^{i1} \ s^{i2}]^T$  wykorzystywaną we wzorze (4.57). Dla porównania, na rys. 4.4 a) zilustrowano wektory dodatkowych odchyżeń od każdego kolejnego punktu  $c^i$  służące do bezpośredniego wyliczenia pochodnych cząstkowych w tradycyjnych algorytmach ISOPE [125, 126, 13, 12, 153], tj. wektory różnic generowane są tu niezależnie od algorytmu podstawowego na podstawie pomiarów w dodatkowych punktach tworzących z punktem  $c^i$  liniowo niezależne kierunki osi układu współrzędnych.

Aby wyliczenie pochodnych na podstawie układu równań liniowych (4.57) było praktycznie użyteczne, macierz  $\mathbf{S}^i$  musi być nie tylko nieosobliwa, ale i *dostatecznie dobrze uwarunkowana*, gdyż prawa strona tego układu jest wektorem obciążonym zarówno błędem metody (przybliżenie prawej strony układu równań (4.56)), jak i błędami pomiaru wartości wyjść  $y_j^i = F_{*j}(c^i)$ . Dobre uwarunkowanie można uzyskać jedynie w przypadku odpowiednio dobrego rozlokowania kolejnych punktów  $c^i$  w przestrzeni  $\mathbb{R}^{n_c}$ .



Rysunek 4.4. Wektory różnicowe służące do estymacji pochodnych odwzorowania procesowego: a)  $n_c$  dodatkowych odchyżeń w każdym punkcie  $c^i$  (algorytm podstawowy), b) na podstawie punktów samego algorytmu (algorytm dualny)

Dla uzyskania tego celu zmodyfikowane zadanie optymalizacji modelowej ZZOM zostało przeformułowane przez wprowadzenie dodatkowego ograniczenia, zwanego dalej *ograniczeniem uwarunkującym*, postaci

$$d(c^{i+1}(c), c^i, \dots, c^{i-n_c+1}) = \frac{\sigma_{\min}(\mathbf{S}^{i+1}(c))}{\sigma_{\max}(\mathbf{S}^{i+1}(c))} \geq \delta, \quad (4.58)$$

gdzie

$$c^{i+1}(c) = c^i + k_c(c - c^i),$$

$$\mathbf{S}^{i+1}(c) = [c^i - c^{i+1}(c) \quad c^{i-1} - c^{i+1}(c) \quad \dots \quad c^{i-n_c+1} - c^{i+1}(c)]^T, \quad (4.59)$$

$\sigma_{\min}(\mathbf{S}^{i+1}(c))$  i  $\sigma_{\max}(\mathbf{S}^{i+1}(c))$  oznaczają odpowiednio minimalną i maksymalną wartość szczególną (singularną)<sup>1</sup> macierzy  $\mathbf{S}^{i+1}(c)$ , zaś wartość  $\delta$ ,  $0 < \delta < 1$ , określa wymagane uwarunkowanie macierzy  $\mathbf{S}^{i+1}(c)$ . Należy pamiętać, że  $\frac{\sigma_{\min}(\mathbf{S}^{i+1})}{\sigma_{\max}(\mathbf{S}^{i+1})}$  jest odwrotnością standardowo używanego wskaźnika uwarunkowania macierzy  $\mathbf{S}^{i+1}$ , liczonego w normie spektralnej (normie drugiej).

*Algorytm DPMO (Dualny PMO)* sformułujemy w następującej postaci:

**Start.** Dane startowe: punkt początkowy  $c^0$ , współczynnik relaksacji  $k_c$  ( $0 < k_c \leq 1$ ), współczynnik kary  $\rho > 0$ , parametr ograniczenia uwarunkującego  $\delta > 0$ , parametr testu stopu  $\epsilon > 0$ , ustaw  $i := 0$ .

<sup>1</sup> Wartości szczególne macierzy  $\mathbf{S}$  to pierwiastki wartości własnych macierzy  $\mathbf{S}^T \mathbf{S}$ , zob. np. [133, 118, 146, 147].

**Krok 0 (faza wstępna).** Zmierz wartości ustalone wyjść  $y^0 = F_*(c^0)$  obiektu. Wybierz  $n_c$  punktów  $c^{-n_c}, c^{-n_c+1}, \dots, c^{-1}$  takich, że macierz  $S^0$  jest dobrze uwarunkowana (np. dokonując odchyień od  $c^0$  w kierunkach osi przestrzeni  $\mathbb{R}^{n_c}$ ), zastosuj te punkty kolejno jako wartości zadane regulatorów obiektu sterowanego mierząc, po ustaniu procesów przejściowych, wartości wyjść  $y^k = F_*(c^k)$ ,  $k = -n_c, -n_c + 1, \dots, -1$ .

**Krok 1.** Wyznacz estymatę pochodnej  $(F_*)'(c^i)$  odwzorowania wyjść obiektu regulowanego względem  $c^i$  rozwiązując  $n_y$ ,  $n_y = \dim y$ , układów równań liniowych (4.57).

**Krok 2.** Dla  $a^i = F_*(c^i) - F(c^i, \alpha)$  i  $\lambda(c^i, a^i)$  rozwiąż *Zmodyfikowane Zadanie Optymalizacji Modelowej Uwarunkowane ZZOMU* (tzn. z ograniczeniem uwarunkującym (4.58)):

$$\begin{aligned} & \min_c \left\{ Q(c, F(c, \alpha) + a^i) - \lambda(c^i, a^i)^T c + \rho \|c^i - c\|^2 \right\} \\ & \text{z ogr. : } g(c) \leq 0, \\ & d(c^i + k_c(c - c^i), c^i, \dots, c^{i-n_c+1}) \geq \delta, \end{aligned} \quad (4.60)$$

oznaczając rozwiązanie przez  $\hat{c}_{zm}^i = \hat{c}_{zm}(c^i, a^i)$ .

**Krok 3.** Wyznacz

$$c^{i+1} := c^i + k_c(\hat{c}_{zm}^i - c^i).$$

**Krok 4.** Jeśli

$$\|c^{i+1} - c^i\| \leq \epsilon,$$

to zakończ, punkt optymalny został znaleziony (**Stop**).

**Krok 5.** Zastosuj wartości zadane  $c^{i+1}$  do obiektu regulowanego, po ustaniu procesów przejściowych zmierz wartości wyjść  $y^{i+1} = F_*(c^{i+1})$ . Dokonaj adaptacji parametrów modelu  $\alpha$  (opcjonalnie).

Ustaw  $i := i + 1$  i idź do **Kroku 1**.

Dla uzyskania dobrej *efektywności obliczeniowej* zaleca się następującą organizację kroku 2 algorytmu: najpierw rozwiązywane jest *Zmodyfikowane Zadanie Optymalizacji Modelowej ZZOM* (4.46) oraz sprawdza się, czy uzyskany punkt  $\hat{c}_{zm}^i = \hat{c}_{zm}(c^i, a^i)$  spełnia ograniczenie uwarunkujące, tzn. czy

$$d(c^i + k_c(\hat{c}_{zm}^i - c^i), c^i, \dots, c^{i-n_c+1}) \geq \delta.$$

Jeśli ograniczenie jest spełnione, to przechodzimy do następnego kroku algorytmu. Natomiast jeśli nie jest spełnione, to dopiero wtedy przystępujemy do minimalizacji zadania ZZOMU, najlepiej startując procedurę optymalizacji z punktu uzyskanego jako rozwiązanie zadania ZZOM. Postępowanie

powyższe uzasadnione jest tym, że ograniczenie uwarunkowujące jest silnie nieliniowe. Ponadto, obliczanie wartości lewej jego strony jest numerycznie złożone (liczenie wartości szczególnych macierzy).

Jeśli w danej iteracji ograniczenie uwarunkowujące nie jest aktywne w kroku 2 algorytmu, to uzyskany punkt minimalizuje zadanie ZZOM, takie jak w algorytmie podstawowym PMO – stąd bez jakiegokolwiek utraty optymalności. Jeśli natomiast ograniczenie uwarunkowujące jest aktywne, to następuje w danym kroku pewne odstępstwo od ścisłej optymalności – dla uzyskania odpowiednich warunków estymacji w kolejnej iteracji algorytmu. Jest to więc mechanizm zintegrowanej, wzajemnie się uwarunkowującej optymalizacji i estymacji, mechanizm *sterowania dualnego*.

Zwróćmy uwagę, że człon regularyzujący  $\rho \|c^i - c\|^2$  pełni w algorytmie dualnym w istocie rolę podwójną: *regularyzacji* (uwypuklenia i ujednoznacznienia) zadania optymalizacji oraz pośrednio *kontroli długości kroku*. Ze względu na wykorzystanie punktów generowanych w kolejnych iteracjach algorytmu również do aproksymacji wartości pochodnych, odległości między nimi nie powinny być zbyt duże. Pośrednio kontrolę długości kroków można uzyskać właśnie przez dobór odpowiedniej wartości współczynnika kary  $\rho$ .

Z praktycznego punktu widzenia istotne są cechy występującego w zadaniu ZZOMU (4.60) dodatkowego ograniczenia określonego przez nierówność (4.58) (dodatkowego w stosunku do zadania ZZOM), definiującego zbiór

$$D^i = \{c \in \mathbb{R}^{n_c} : d(c^i + k_c(c - c^i), c^i, \dots, c^{i-n_c+1}) \geq \delta\}. \quad (4.61)$$

Analiza teoretyczna własności tego zbioru w przypadku ogólnym jest trudna. Dla  $n_c = 2$  można wykazać, że składa się on z dwóch podzbiorów spójnych będących kołami o promieniu i położeniu zależnym od usytuowania punktów  $c^i, c^{i-1}$  oraz wartości  $\delta$  i  $k_c$ . Ponadto można wykazać, że jeśli zbiór dopuszczalny  $C \cap D^i$  zadania ZZOMU jest niepusty w pierwszej iteracji ( $i = 0$ ), to jest on również niepusty w drugiej i wszystkich następnych iteracjach [134, 135]. W pracach tych zamieszczono też rezultaty teoretyczne dotyczące analizy zbieżności nieco uproszczonej wersji algorytmu, a mianowicie z sukcesywnymi linearyzacjami funkcji celu na początku każdej iteracji (tzw. algorytm DPMOL). Przy rozsądnych założeniach, w przypadku  $C = \mathbb{R}^{n_c}$  pokazano, że gradient rzeczywistej funkcji celu  $Q(c^i, F_*(c^i))$  zbiega do zera przy  $i \rightarrow \infty$ , jeśli tylko  $\delta \lesssim 0.284$ . Jest to rezultat ważny z praktycznego punktu widzenia, gdyż rekomendowane na podstawie badań symulacyjnych wartości  $\delta$  są rzędu 0.05 do 0.2 (w przypadku pomiarów silnie zaszumionych), a więc leżą w obszarze teoretycznie zagwarantowanej zbieżności. Pokazano ponadto na przykładach symulacyjnych, że algorytmy

DPMO i DPMOL praktycznie identycznie reagują na wartości  $\delta$ , stąd hipoteza o takiej samej czy zbliżonej granicznej wartości  $\delta$  również dla algorytmu DPMO powinna być prawdziwa. Niestety, algorytm DPMO (jak i ISOPED) ma dzięki wprowadzeniu ograniczenia uwarunkowującego skomplikowaną, silnie nieliniową strukturę powodującą w ogólności niemonotoniczne jego zachowanie, stąd analiza teoretyczna okazała się nawet w przypadku dwuwymiarowym bardzo trudna [134].

### Algorytm DPMO z optymalizacją fazy wstępnej

Zasadnicze cele fazy wstępnej (**Krok 0**) algorytmu DPMO to:

- Zgromadzenie punktów  $c^{-1}, c^{-2}, \dots, c^{-n_c}$  takich, że wraz z punktem  $c^0$  rozpinają przestrzeń  $\mathbb{R}^{n_c}$ , stanowiąc podstawę dla utworzenia macierzy  $\mathbf{S}^0$  w pierwszej regularnej iteracji algorytmu.
- Zgromadzenie informacji umożliwiającej wyliczenie pochodnej  $(F_*)'(c^0)$  w punkcie startowym  $c^0$ , tzn. sukcesywne zastosowanie punktów  $c^{-1}, c^{-2}, \dots, c^{-n_c}$  do obiektu regulowanego i zgromadzenie pomiarów wyjść w tych punktach.

Oczywiście, jedynym sposobem realizacji wymienionych celów jest przeprowadzenie  $n_c$  zmian punktu pracy, najłatwiej w kierunkach osi układu współrzędnych. Technika taka nie bierze pod uwagę, w ogólności, aspektu ekonomicznego zmian punktu pracy. W [140] zaproponowano efektywniejsze sformułowanie fazy wstępnej algorytmu dualnego ISOPE. W tym celu rozszerza się definicję macierzy  $\mathbf{S}^{i+1}(c)$  na ujemne wartości  $i$ , następująco:

$$\mathbf{S}^{i+1}(c) = [c^i - c \quad c^{i-1} - c \quad \dots \quad c^{-n_c} - c]^T, \quad -n_c < i < 0, \quad (4.62)$$

tzn.

$$\begin{aligned} \mathbf{S}^{-n_c+1}(c) &= [c^{-n_c} - c]^T, \\ \mathbf{S}^{-n_c+2}(c) &= [c^{-n_c+1} - c \quad c^{-n_c} - c]^T, \\ &\vdots \\ \mathbf{S}^0(c) &= [c^{-1} - c \quad c^{-2} - c \quad \dots \quad c^{-n_c} - c]^T. \end{aligned}$$

Porównując tę definicję ze sformułowaniem (4.59) widać, że w fazie wstępnej przyjmujemy teraz  $c^{i+1}(c) = c$ , tzn.  $k_c = 1$ . Zauważmy, że dla  $i < -1$  macierz  $\mathbf{S}^{i+1}$  jest niekwadratowa, z wymiarem rosnącym wraz ze wzrostem  $i$ , dla  $i = -1$  staje się dokładnie macierzą kwadratową  $\mathbf{S}^0 = \mathbf{S}^0(c^0)$  potrzebną do realizacji pierwszej regularnej iteracji algorytmu DPMO.

Konsekwentnie, rozszerzamy definicję ograniczenia uwarunkowującego:

$$d(c, c^i, c^{i-1}, \dots, c^{-n_c}) = \frac{\sigma_{\min}(\mathbf{S}^{i+1}(c))}{\sigma_{\max}(\mathbf{S}^{i+1}(c))} \geq \delta, \quad (4.63)$$

gdzie  $\sigma_{\min}(\mathbf{S}^{i+1}(c))$  i  $\sigma_{\max}(\mathbf{S}^{i+1}(c))$  oznaczają minimalną i maksymalną wartość szczególną macierzy  $\mathbf{S}^{i+1}(c)$ ,

Zoptymalizowaną fazę wstępną algorytmu DPMO sformułujemy w następującej postaci:

### Krok 0 (Faza wstępna zoptymalizowana):

**Start 0.** Ustal dodatnie parametry  $\rho_1, \gamma$ . Ustaw  $i := -n_c$ , oznacz punkt początkowy  $c^0$  przez  $c^{-n_c}$ .

**Krok 0.1.** Zastosuj wartości zadane regulatorów  $c^i$  do obiektu regulowanego, po ustaniu procesów przejściowych zmierz wartości wyjść  $y^i = F_*(c^i)$ . Dodaj pomiary do zestawu danych i dokonaj adaptacji parametrów  $\alpha$  modelu statycznego (opcjonalnie).

**Krok 0.2.** Rozwiąż następujące rozszerzone i uwarunkowane zadanie optymalizacji modelowej

$$\begin{aligned} \min_c \{ & Q(c, F(c, \alpha) + a^i) + \rho_1 \|c^i - c\|^2 \} \\ \text{z ogr. : } & g(c) \leq 0, \\ & \|c^i - c\| \geq \gamma, \\ & d(c, c^i, c^{i-1}, \dots, c^{-n_c}) \geq \delta, \end{aligned} \quad (4.64)$$

oznaczając rozwiązanie przez  $c^{i+1}$ .

**Krok 0.3.** Ustaw  $i := i + 1$ . Jeśli  $i < 0$  to wróć do **Kroku 0.1**, w przeciwnym przypadku przejdź do **Kroku 1** algorytmu DPMO.

Z przyczyn identycznych jak podane w przypadku kroku 2 regularnej iteracji algorytmu DPMO, dla uzyskania dobrej *efektywności obliczeniowej* zaleca się następującą organizację kroku 0.2: najpierw rozwiązywane jest zadanie optymalizacji (4.64) bez ograniczenia uwarunkowującego  $d(c, c^i, c^{i-1}, \dots, c^{-n_c}) \geq \delta$ . Następnie sprawdza się, czy uzyskany punkt  $c = c^{i+1}$  spełnia to ograniczenie. Tylko jeśli nie jest ono spełnione, to minimalizujemy zadanie optymalizacji w pełnej wersji (4.64).

Zauważmy, że tworząc zadanie optymalizacji (4.64) dodano do standardowego sformułowania zadania optymalizacji modelowej ZOM (tj. minimalizuj  $Q(c, F(c, \alpha))$  przy ograniczeniach  $g(c) \leq 0$ ) trzy elementy:

- Po pierwsze, funkcja celu została rozszerzona o wektor przesunięć wyjść  $a^i$  i składnik regularyzujący (kary), podobnie jak w fazie zasadniczej algorytmów PMO czy DPMO. Przyczyny są podobne: zbliżenie wartości funkcji celu do rzeczywistej (przesunięcie) oraz regularyzacja (uwypuklenie) i przeciwdziałanie zbyt dużym odchyleniom rozwiązania od

punktu bieżącego. Stąd wartość współczynnika  $\rho_1$  może być potrzebna w tej fazie nawet większa niż potem w regularnych iteracjach ( $\rho_1 \geq \rho$ ).

- Po drugie, dodano ograniczenie  $\|c^i - c\| \geq \gamma$ , wymuszające nie za małe zmiany punktu pracy, odpowiednie dla późniejszej estymacji pochodnej metodą różnic skończonych z wykorzystaniem macierzy  $\mathbf{S}^0$ . W szczególności, ograniczenie to jest niezbędne, jeśli algorytm startuje z punktu optymalnego dla modelu (czy bliskiego takiemu) i przesunięcia modelu wyjść  $a^i$  nie odgrywają roli (np. funkcja celu liniowa względem  $y$ ).
- Po trzeciej, dodano ograniczenie uwarunkowujące (4.63).

Trzeba zaznaczyć, że poprawa wartości funkcji celu w trakcie zoptymalizowanej fazy wstępnej zależy przede wszystkim od dokładności modelu, którym dysponujemy – im dokładniejszy model, tym większej poprawy wartości funkcji celu można oczekiwać. Oczywiście, jeśli nie startujemy z wstępnie wyznaczonego i zastosowanego punktu optymalnego dla modelu. Jeśli dysponujemy modelem mało dokładnym lub startujemy z punktu modelowo optymalnego i przesunięcia wyjść  $a^i$  nie mają wpływu na rezultaty optymalizacji, to zoptymalizowana faza wstępna może przynieść niewielką lub żadną poprawę rzeczywistego wskaźnika jakości. W takich sytuacjach zastosowanie podstawowej wersji fazy wstępnej jest nadal sensowną alternatywą, szczególnie jeśli przy doborze odchyleń punktu pracy uwzględnia się kierunki największej zmienności funkcji celu. Algorytm DPMO ze zoptymalizowaną fazą wstępną nazywać będziemy *algorytmem DPMO+*.

#### 4.3.5. Przykłady symulacji sterowania

##### Przykład 4.1a.

Rozważymy na wstępie bardzo proste zadanie sterowania optymalizującego punkt pracy przedstawione w [14], dla przejrzystej ilustracji sformułowań i działania algorytmów.

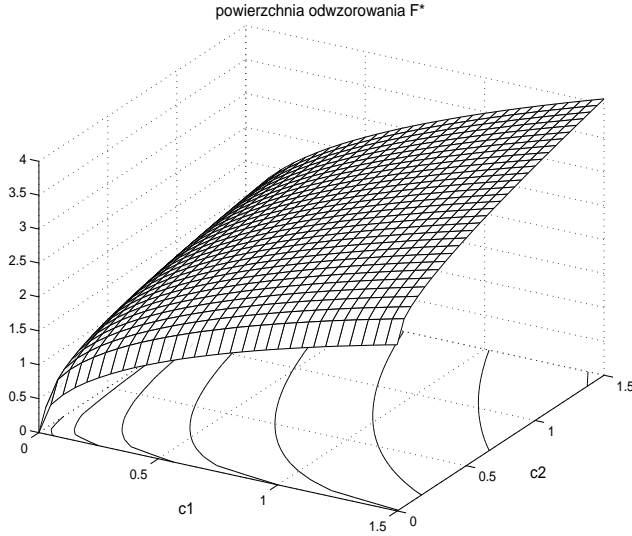
Obiekt regulowany opisany jest w stanach ustalonych następującym odwzorowaniem, przedstawionym na rysunku 4.5,

$$y = F_*(c_1, c_2) = 2c_1^{0.5} + c_2^{0.4} + 0.2c_1c_2. \quad (4.65)$$

Minimalizowana funkcja celu i zbiór ograniczeń są postaci

$$Q(c, y) = -y + (c_1 - 0.5)^2 + (c_2 - 0.5)^2, \quad (4.66)$$

$$C = \{c \in \mathbb{R}^2 : 0 \leq c_1 \leq 1.5, \quad 0 \leq c_2 \leq 1.5\}.$$



Rysunek 4.5. Powierzchnia odwzorowania obiektu regulowanego, przykład 4.1

Punkt optymalny sformułowanego zadania to  $(\hat{c}_1, \hat{c}_2) = (1.067, 0.830)$ , przy czym optymalna wartość rzeczywistej funkcji celu to  $Q(\hat{c}, F_*(\hat{c})) = -2.7408$ . Punkt optymalny leży we wnętrzu zbioru dopuszczalnego  $C$ .

Zakładamy, że dysponujemy modelem  $F$  odwzorowania (4.65) postaci:

$$y = F(c_1, c_2, \alpha) = 0.6c_1 + 0.4c_2 + \alpha. \quad (4.67)$$

Zwróćmy uwagę, że funkcja celu (4.66) jest liniowa względem zmiennej wyjściowej  $y$ , stąd nie ma potrzeby wyliczania przesunięcia modelu  $a^i$  w PMO czy DPMO, zob. rozdz. 4.3.3.

Zmodyfikowane zadanie optymalizacji modelowej (ZZOM – (4.46)) algorytmu PMO ma dla naszego przykładu postać

$$\begin{aligned} \min_c \left\{ Q(c, F(c, \alpha)) - \lambda(c^i)^T c + \rho \|c^i - c\|^2 \right\} \\ \text{z ogr. : } 0 \leq c_1 \leq 1.5, \\ 0 \leq c_2 \leq 1.5, \end{aligned}$$

gdzie  $c = [c_1 \ c_2]^T$ ,

$$Q(c, F(c, \alpha)) = -0.6c_1 - 0.4c_2 - \alpha + (c_1 - 0.5)^2 + (c_2 - 0.5)^2,$$

$$\begin{aligned} \lambda(c^i)^T &= - \left( [0.6 \ 0.4] - \left[ \frac{\partial F_*}{\partial c_1}(c^i) \quad \frac{\partial F_*}{\partial c_2}(c^i) \right] \right) \\ &= \left[ -0.6 + \frac{\partial F_*}{\partial c_1}(c^i) \quad -0.4 + \frac{\partial F_*}{\partial c_2}(c^i) \right], \end{aligned}$$

przy czym wektor pochodnej  $(F_*)'(c^i) = [\frac{\partial F_*}{\partial c_1}(c^i) \quad \frac{\partial F_*}{\partial c_2}(c^i)]$  będzie w trakcie działania algorytmu estymowany metodą różnic skończonych, na podstawie wartości wyjść  $F_*(c)$ .

W algorytmie DPMO zamiast ZZOM rozwiązywane jest zadanie ZZOMU (4.60), powstające przez dodanie do ZZOM ograniczenia uwarunkowującego (4.58)

$$d(c^{i+1}(c), c^i, c^{i-1}) = \frac{\sigma_{\min}(\mathbf{S}^{i+1}(c))}{\sigma_{\max}(\mathbf{S}^{i+1}(c))} \geq \delta,$$

gdzie  $c^{i+1}(c) = c^i + k_c(c - c^i)$ ,

$$\mathbf{S}^{i+1}(c) = \begin{bmatrix} c_1^i - c_1^{i+1}(c) & c_1^{i-1} - c_1^{i+1}(c) \\ c_2^i - c_2^{i+1}(c) & c_2^{i-1} - c_2^{i+1}(c) \end{bmatrix}^T,$$

zaś  $\sigma_{\min}(\mathbf{S}^{i+1}(c))$  i  $\sigma_{\max}(\mathbf{S}^{i+1}(c))$  to mniejsza i większa wartości szczególne macierzy  $\mathbf{S}^{i+1}(c)$ . Wektor pochodnej  $(F_*)'(c^i) = [\frac{\partial F_*}{\partial c_1}(c^i) \quad \frac{\partial F_*}{\partial c_2}(c^i)]$  będzie w tym przypadku estymowany z rozwiązania układu dwóch równań liniowych

$$\mathbf{S}^i \begin{bmatrix} \frac{\partial F_*}{\partial c_1}(c^i) \\ \frac{\partial F_*}{\partial c_2}(c^i) \end{bmatrix} = \begin{bmatrix} F_*(c^{i-1}) - F_*(c^i) \\ F_*(c^{i-2}) - F_*(c^i) \end{bmatrix},$$

gdzie

$$\mathbf{S}^i = \begin{bmatrix} c^{i-1} - c^i & c^{i-2} - c^i \end{bmatrix}^T, \quad i = 1, 2.$$

Dla przedstawionego obiektu regulowanego symulowano sterowanie optymalizujące punkt pracy trzema algorytmami:

**PMO** – algorytm podstawowy, z dodatkowymi  $n_c$  odchyleniami punktów pracy w każdej iteracji dla estymacji pochodnych odwzorowania obiektu regulowanego metodą różnic skończonych, zob. rozdz. 4.3.3,

**DPMO** – algorytm dualny z generacją pierwszej wartości pochodnej w fazie wstępnej metodą odchylen punktów pracy jak w PMO, zob. rozdz. 4.3.4,

**DPMO+** – algorytm dualny DPMO ze zoptymalizowaną fazą wstępną, zob. rozdz. 4.3.4.

Do symulacji przyjęto punkt startowy  $c^0 = (0.5 \ 0.5)$  oraz następujące nominalne wartości parametrów rozważanych algorytmów:

- współczynnik relaksacji  $k_c = 0.5$  (zob. (4.48)),
- współczynnik kary (regularyzacji)  $\rho = 0.1$ ,

- parametr stopu (najmniejsza akceptowalna zmiana punktu)  $\varepsilon = 0.04$ ,
- wielkość odchyłek dla estymacji pochodnych w PMO i fazie wstępnej DPMO  $\Delta c_j = \varepsilon$ ,
- wartość prawej strony ograniczenia uwarunkowującego  $\delta = 0.1$ ,
- wartość prawej strony ograniczenia dodatkowego w zadaniu (4.64) w fazie wstępnej DPMO+  $\gamma = \varepsilon$ ,
- współczynnik kary w zadaniu (4.64)  $\rho_1 = 4 * \rho$ .

Wartości wybranych parametrów były następnie celowo zmieniane, aby zbadać ich wpływ na przebieg symulacji. Zwróćmy uwagę na małą, z matematycznego punktu widzenia, dokładność punktu końcowego wynikającą z dość dużej wartości parametru stopu  $\varepsilon$ , jak na kryteria stopu algorytmów typu optymalizacyjnego. Nie badamy bowiem tutaj matematycznego algorytmu optymalizacji, ale *symulujemy działanie algorytmu optymalizującego sterowanie obiektem technicznym*. O zakończeniu działania algorytmu, tj. przerwaniu zmian wartości zadanych regulatorów decyduje najmniejsza sensowna zmiana tych punktów, uwarunkowana dokładnością ustawiania elementów wykonawczych i pomiarów wielkości wyjściowych obiektu regulowanego. Stąd sensowne odchyłki punktów pracy dla estymacji pochodnych  $\Delta c_j$  też nie mogą być mniejsze od omówionej wartości, podobnie jak i najmniejsza akceptowalna zmiana wynikająca z rozwiązania zadania optymalizacji w fazie wstępnej algorytmu DPMO+ (przyjęto  $\Delta c_j = \varepsilon$ ,  $\gamma = \varepsilon$ ).

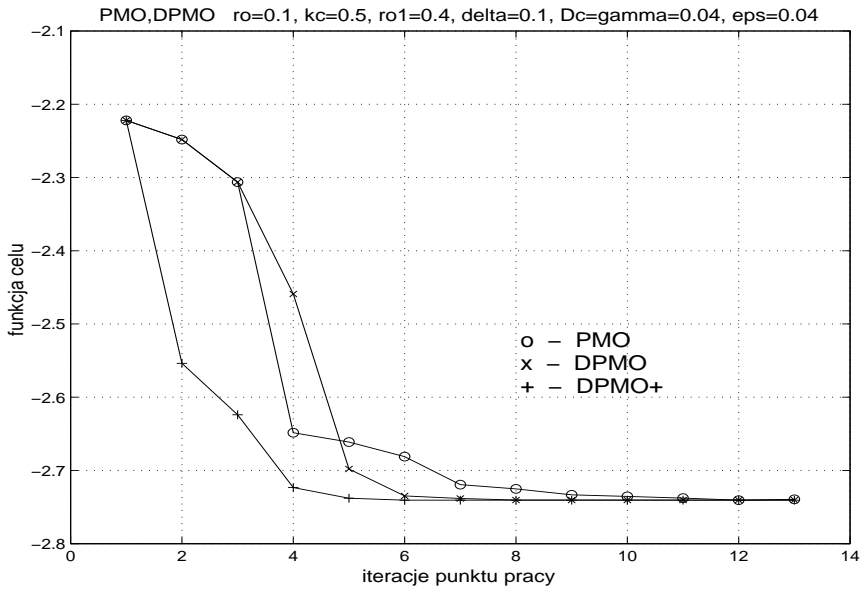
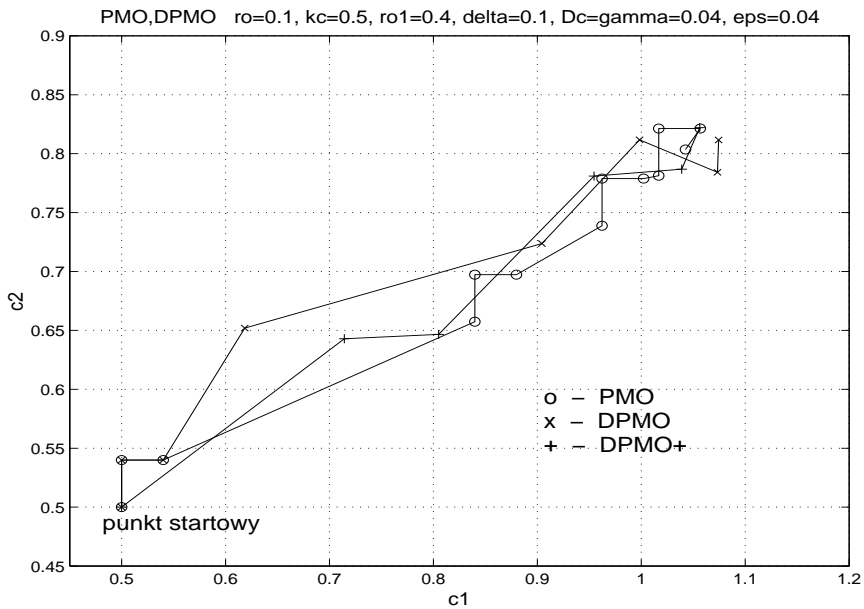
Przebiegi prezentowanych przykładowych przebiegów symulacji przedstawione są w postaci dwóch rysunków:

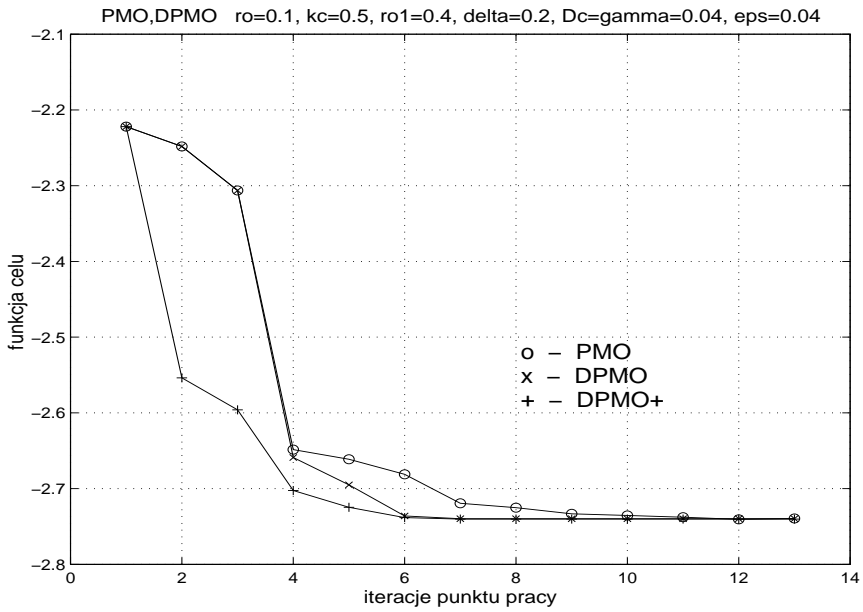
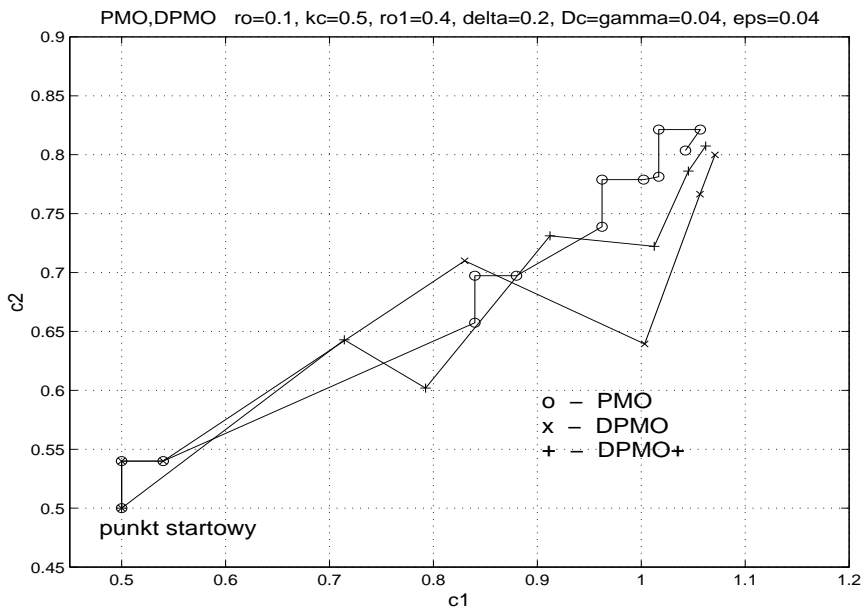
- trajektorii uzyskiwanych wartości rzeczywistych funkcji celu  $Q(c^i, F_*(c^i))$  w funkcji wszystkich kolejnych punktów pracy, tj. uwzględniając również ew. dodatkowe odchyłki punktów pracy dla uzyskania pochodnych odwzorowania obiektu regulowanego,
- trajektorii punktów pracy na płaszczyźnie zmiennych  $(c_1, c_2)$ .

Na rysunkach 4.6 i 4.7 przedstawiono trajektorie uzyskane dla wszystkich trzech algorytmów, dla nominalnych wartości parametrów podanych wyżej.

Na rysunku 4.7 widać charakterystyczne dwa ruchy testowe (odchyłki) poprzedzające każdy krok główny (iterację) algorytmu PMO oraz charakterystyczną "zygzakowatość" trajektorii algorytmów DPMO wynikającą z interwencji ograniczenia uwarunkowującego. Ze względu na optymalizację modelową wykonywaną również w fazie wstępnej, zdecydowanie najefektywniejszy jest algorytm DPMO+ (punkt pracy startowy nie został wstępnie ustawiony na wartości optymalnej dla modelu).

Na rysunkach 4.8 i 4.9 przedstawiono trajektorie uzyskane po zwięk-

Rysunek 4.6. Trajektorie funkcji celu,  $\delta = 0.1$ , przykład 4.1aRysunek 4.7. Trajektorie punktów pracy,  $\delta = 0.1$ , przykład 4.1a

Rysunek 4.8. Trajektorie funkcji celu,  $\delta = 0.2$ , przykład 4.1aRysunek 4.9. Trajektorie punktów pracy,  $\delta = 0.2$ , przykład 4.1a

szczeniu wartości prawej strony ograniczenia uwarunkowującego z 0.1 na 0.2. Na płaszczyźnie punktów pracy widać wyraźnie zwiększenie „zygzakowatości” trajektorii dla algorytmów dualnych, zaś ich efektywność nie uległa istotnym zmianom.  $\square$

#### Przykład 4.1b.

Jest to przykład poprzedni, ale ze zmienionym zbiorem ograniczeń:

$$C = \{c \in \mathbb{R}^2 : 0 \leq c_1 \leq 0.8, 0 \leq c_2 \leq 1.5\}, \quad (4.68)$$

tak więc jedno z ograniczeń ( $c_1 \leq 0.8$ ) jest aktywne w punkcie optymalnym. Przykład ten pokazuje działanie algorytmów dla sytuacji aktywnego ograniczenia.

Na rysunkach 4.10 i 4.11 pokazano trajektorie wszystkich trzech algorytmów: PMO, DPMO i DPMO+, dla nominalnych wartości parametrów takich jak wymienione poprzednio w przykładzie 1. Zwróćmy uwagę na bardzo wysoką, akurat w tym przypadku, efektywność algorytmu DPMO+, oraz na konieczność zmieniania znaku odchyłki testowej dla zmiennej  $c_2$  w pobliżu ograniczenia w algorytmie PMO.  $\square$

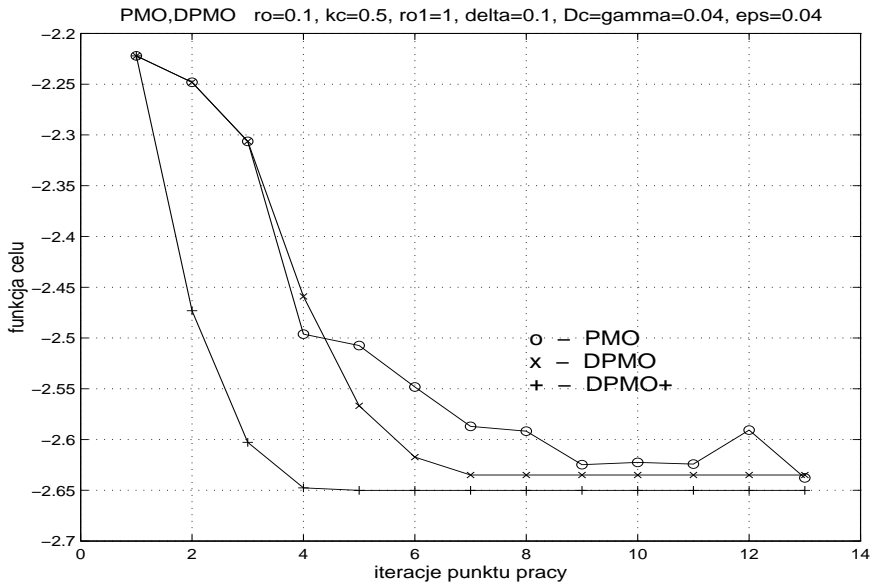
#### Przykład 4.1c.

Jest to przykład 4.1a ze zmodyfikowanym równaniem wyjścia obiektu regulowanego, a mianowicie zamiast (4.65) mamy

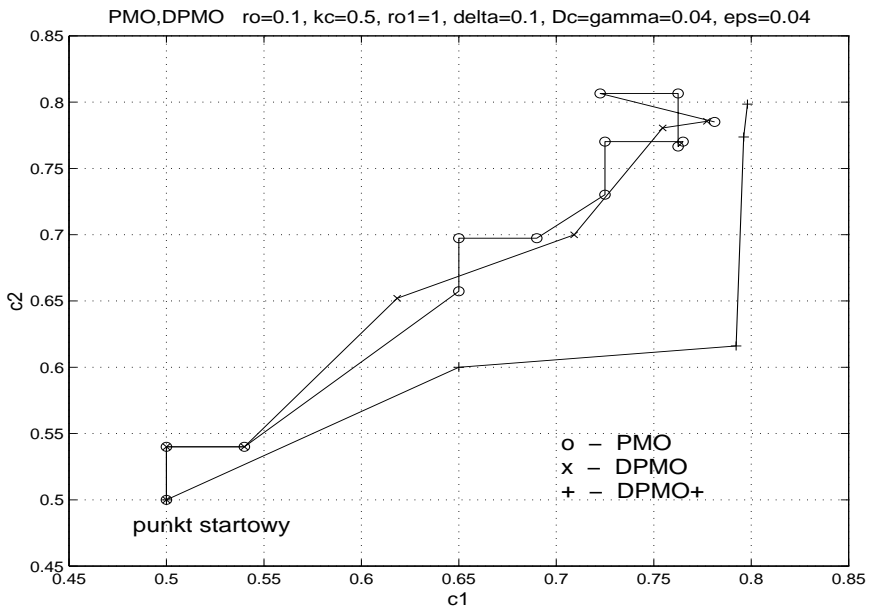
$$y = F_*(c_1, c_2) = 2c_1^{0.5} + c_2^{0.4} + 0.2c_1c_2 + \nu, \quad (4.69)$$

gdzie  $\nu$  jest sygnałem pseudolosowym reprezentującym błąd pomiaru, o rozkładzie jednostajnym w przedziale  $[-0.03, +0.03]$ . Odpowiada to błędowi pomiaru z zakresu ok. 1.5%, gdyż w trakcie biegu symulacji wyjścia przyjmują wartości  $y \cong 2 \div 3$ . Zauważmy, że możliwy błąd różnicy wartości wyjść może być znacznie większy, w zależności od krzywizny charakterystyki statycznej. Podkreślmy, że mówimy o błędach pomiaru *stanu ustalonego*, wyliczanego w praktyce jako średnia z pewnej liczby pomiarów chwilowych, stąd można traktować szum pomiarowy jako odfiltrowany.

Zwróćmy uwagę na pierwszorzędne znaczenie odpowiedniego doboru minimalnej dopuszczalnej wartości zmiany punktu pracy  $c$ , tj parametrów  $\varepsilon$ ,  $\Delta c_j$  i  $\gamma$  dla konkretnego rozważanego procesu, w realnej przecież sytuacji pomiaru wyjść z zakłóceniami. We wszystkich algorytmach estymacja pochodnych odbywa się bowiem na podstawie różnic wartości mierzonych wyjść obiektu regulowanego – różnic powodowanych specjalnie odchylanymi punktami pracy (PMO, faza wstępna DPMO) lub różnic między wyjściami



Rysunek 4.10. Trajektorie funkcji celu, przykład 4.1b



Rysunek 4.11. Trajektorie punktów pracy, przykład 4.1b

w kolejnych iteracjach algorytmu (iteracje fazy głównej w DPMO, iteracje obu faz w DPMO+). Stąd różnice między kolejnymi punktami pracy muszą być takie, aby różnice odpowiadających im wyjść (niezakłóconych) były co najmniej kilkakrotnie większe od błędu pomiarowego. W sytuacji rozważanego obiektu mamy następującą przybliżoną zależność, w zakresie występujących w symulacjach wartości wejść  $c$  i wyjść  $y$ :

$$\Delta y \cong (0.6 \div 1.2)\Delta c.$$

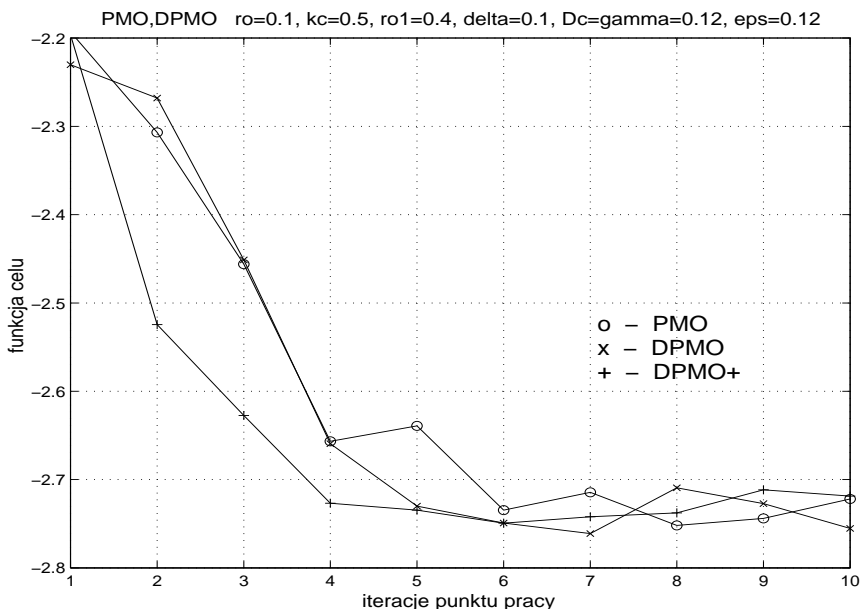
Na rysunkach 4.12 i 4.13 przedstawiono wyniki typowej symulacji wszystkich trzech algorytmów: PMO, DPMO i DPMO+, przy przyjęciu  $\Delta c_j = 0.12$ .

Mamy wówczas

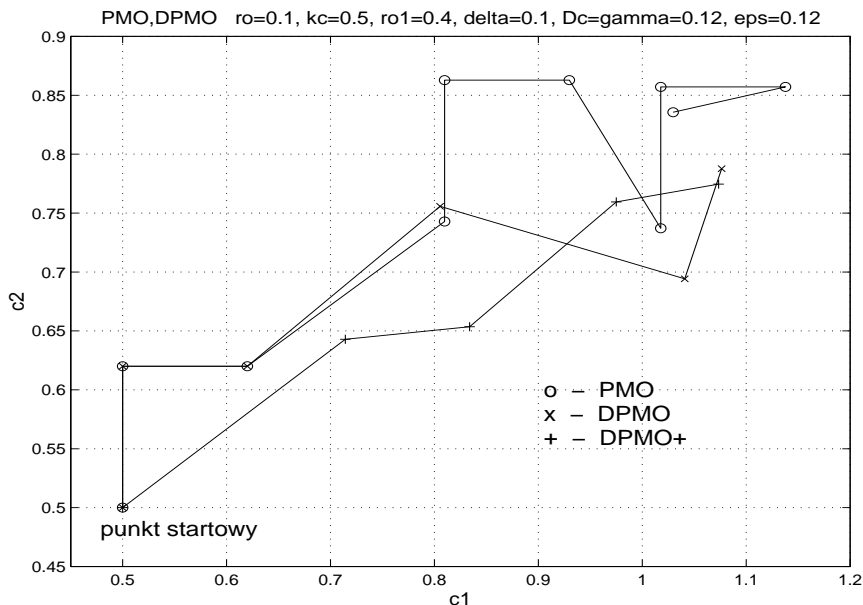
$$\Delta y \cong 0.072 \div 0.144,$$

czyli z numerycznego punktu widzenia dokładność estymacji pochodnych bardzo małą. Trajektorie wyznaczono dla nominalnych wartości parametrów takich jak w poprzednich symulacjach, z wyjątkiem związanych z wartością  $\Delta c_j$ , tj. przyjęto  $\varepsilon = \gamma = \Delta c_j = 0.12$ .

Badania przeprowadzono również dla wartości minimalnych zmian punktu pracy zmniejszonej dwukrotnie, do  $\varepsilon = \Delta c_j = \gamma = 0.06$ . Mimo



Rysunek 4.12. Trajektorie funkcji celu, przykład 4.1c



Rysunek 4.13. Trajektorie punktów pracy, przykład 4.1c

mniej dokładności estymacji wszystkie algorytmy zachowywały się w początkowych iteracjach jeszcze poprawnie, szybko dochodząc w pobliżu punktu optymalnego nawet przy małych, w stosunku do błędów, maksymalnych zmianach punktów pracy. Przebiegów nie zamieszczamy, gdyż charakter trajektorii był podobny do przedstawionych na rysunkach 4.12 i 4.13, jedynie większe były rozrzuty trajektorii i mniejsza dokładność końcowa. □

### Przykład 4.2.

Rozważany obiekt to przedstawiona na rysunku 4.14 kolumna destylacyjna, w której następuje destylacja etylenu z mieszaniny zawierającej głównie etan i etylen.

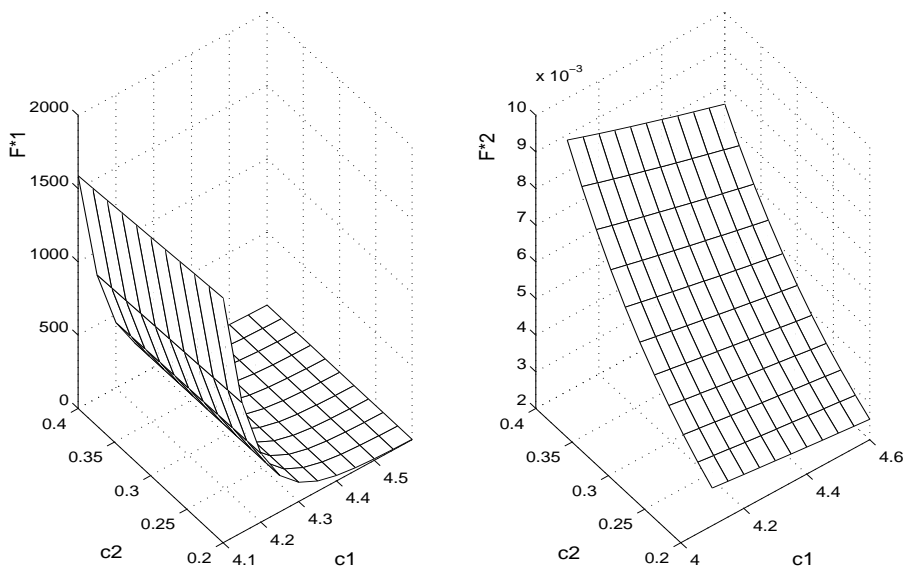
Wektor wielkości wyjściowych  $y = [y_1 \ y_2]^T$  to wartości (w stanie ustalonym) następujących zmiennych:

$y_1$  – stężenie etanu w strumieniu produktu  $P$ , podane w *ppm* (*parts per million*),

$y_2$  – stężenie etylenu w strumieniu dolnym (cieczy wyczerpanej)  $W$ , podane jako ułamek molowy.

Natomiast wektor zmiennych decyzyjnych to wartości zadane  $c = [c_1 \ c_2]^T$  regulatorów wymuszających:



Rysunek 4.15. Powierzchnie odwzorowań obiektu  $F_{*1}(c_1, c_2)$  i  $F_{*2}(c_1, c_2)$ 

gdzie

$$\begin{aligned} m_1 &= 999.9923, & \bar{c}_{11m} &= 4.6261, & \bar{c}_{12m} &= 2.6481, \\ m_{21} &= -0.0016, & m_{22} &= 0.0254, & m_{23} &= 0.0043. \end{aligned}$$

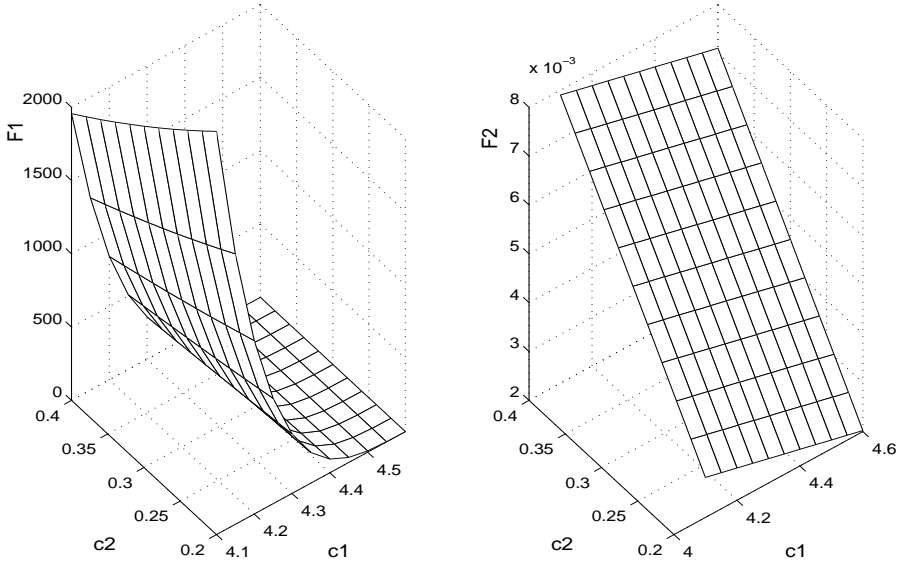
Kształt powierzchni odwzorowań modelu przedstawiony jest na rysunku 4.16. Celem optymalizacji punktów pracy jest osiągnięcie przez wyjścia pożądaných wartości  $y_{r1}$  i  $y_{r2}$ , w stanach ustalonych. Stąd funkcja celu ma postać

$$Q(c, y) = \mu_1(y_1 - y_{r1})^2 + \mu_2(y_2 - y_{r2})^2, \quad (4.72)$$

gdzie  $\mu_1 = 2 \cdot 10^{-5}$  i  $\mu_2 = 10^6$  są współczynnikami skalującymi niezbędnymi ze względu na diametralnie różne zakresy zmienności  $y_1$  (setki) i  $y_2$  ( tysięczne części). Typowe wartości pożądanę wyjść to  $y_{r1} = 500$ ,  $y_{r2} = 0.005$  (przyjęte w prezentowanych dalej symulacjach). Wartości zmiennych decyzyjnych warstwy optymalizacji  $c$  są ograniczone do zbioru

$$C = \{ c \in \mathbb{R}^2 : 4.1 \leq c_1 \leq 4.6, \quad 0.2 \leq c_2 \leq 0.4 \}.$$

Sformułowana funkcja celu ma pośrednio sens ekonomiczny. Im bardziej bowiem zanieczyszczony destylat, tym mniejsze zużycie mediów energetycznych decydujących o kosztach produkcji. Stąd optymalnie jest prowadzić destylację na wartości stężenia  $y_{r1}$  etanu w produkcji jak najbliższym poziomowi granicznemu, ale w odległości strefy bezpieczeństwa o szerokości

Rysunek 4.16. Powierzchnie odwzorowań modelu  $F_1(c_1, c_2)$  i  $F_2(c_1, c_2)$ 

wynikającej ze stopnia niepewności, jakości stosowanego systemu sterowania. Z kolei, nie powinno być za dużo strat etylenu, stąd ograniczenie na stężenie etylenu w cieczy wyczerpanej i pożądana jego wartość  $y_{r2}$ .

Dokonano optymalizacji funkcji celu (4.72) przy ograniczeniach w postaci równań modelu (4.71a), (4.71b) uzyskując wartości zmiennych decyzyjnych optymalnych dla modelu  $(\hat{c}_{m1}, \hat{c}_{m2}) = (4.2685, 0.2964)$ . Odpowiadają im wartości wyjść odwzorowań obiektu (4.70a), (4.70b)  $F_*(\hat{c}_{m1}, \hat{c}_{m2}) = [191.6 \ 0.0048]^T$  – są więc dalekie od wartości założonych  $y_{r1} = 500$ ,  $y_{r2} = 0.005$ . Przeprowadzono następnie symulacje sterowania optymalizującego punkt pracy algorytmami DPMO i DPMO+, w warunkach niedokładnego modelu.

Zwróćmy uwagę, że funkcja celu (4.72) nie jest liniowa względem zmiennych wyjściowych  $y$ , stąd należy wprowadzić przesunięcia modelu  $a^i = [a_1^i \ a_2^i]^T$ ,  $a^i = F_*(c^i) - F(c^i)$ , zob. (4.32) w rozdz. 4.3.3. Zmodyfikowane zadanie optymalizacji modelowej uwarunkowane (ZZOMU – (4.60)) w algorytmie DPMO (i DPMO+) ma dla naszego przykładu postać

$$\begin{aligned} & \min_c \left\{ Q(c, F(c) + a^i) - \lambda(c^i, a^i)^T c + \rho \|c^i - c\|^2 \right\} \\ & \text{z ogr. : } 4.1 \leq c_1 \leq 4.6, \\ & \quad 0.2 \leq c_2 \leq 0.4, \\ & \quad d(c^{i+1}(c), c^i, c^{i-1}) = \frac{\sigma_{\min}(\mathbf{S}^{i+1}(c))}{\sigma_{\max}(\mathbf{S}^{i+1}(c))} \geq \delta, \end{aligned}$$

gdzie  $c = [c_1 \ c_2]^T$ ,  $c^{i+1}(c) = c^i + k_c(c - c^i)$ ,  $\sigma_{\min}(\mathbf{S}^{i+1}(c))$  i  $\sigma_{\max}(\mathbf{S}^{i+1}(c))$  to mniejsza i większa wartości szczególne macierzy  $\mathbf{S}^{i+1}(c)$ ,

$$\mathbf{S}^{i+1}(c) = \begin{bmatrix} c_1^i - c_1^{i+1}(c) & c_1^{i-1} - c_1^{i+1}(c) \\ c_2^i - c_2^{i+1}(c) & c_2^{i-1} - c_2^{i+1}(c) \end{bmatrix}^T.$$

Ponadto w sformułowaniu zadania ZZOMU mamy

$$\begin{aligned} Q(c, F(c) + a^i) &= \mu_1(F_1(c) + a_1^i - y_{r1})^2 + \mu_2(F_2(c) + a_2^i - y_{r2})^2, \\ \lambda(c^i, a^i)^T &= [2\mu_1(F_1(c^i) + a_1^i - y_{r1}) \quad 2\mu_2(F_2(c^i) + a_2^i - y_{r2})] \times \\ &\quad \times \left( \begin{bmatrix} \frac{\partial F_1}{\partial c_1}(c^i) & \frac{\partial F_1}{\partial c_2}(c^i) \\ \frac{\partial F_2}{\partial c_1}(c^i) & \frac{\partial F_2}{\partial c_2}(c^i) \end{bmatrix} - \begin{bmatrix} \frac{\partial F_{*1}}{\partial c_1}(c^i) & \frac{\partial F_{*1}}{\partial c_2}(c^i) \\ \frac{\partial F_{*2}}{\partial c_1}(c^i) & \frac{\partial F_{*2}}{\partial c_2}(c^i) \end{bmatrix} \right), \end{aligned}$$

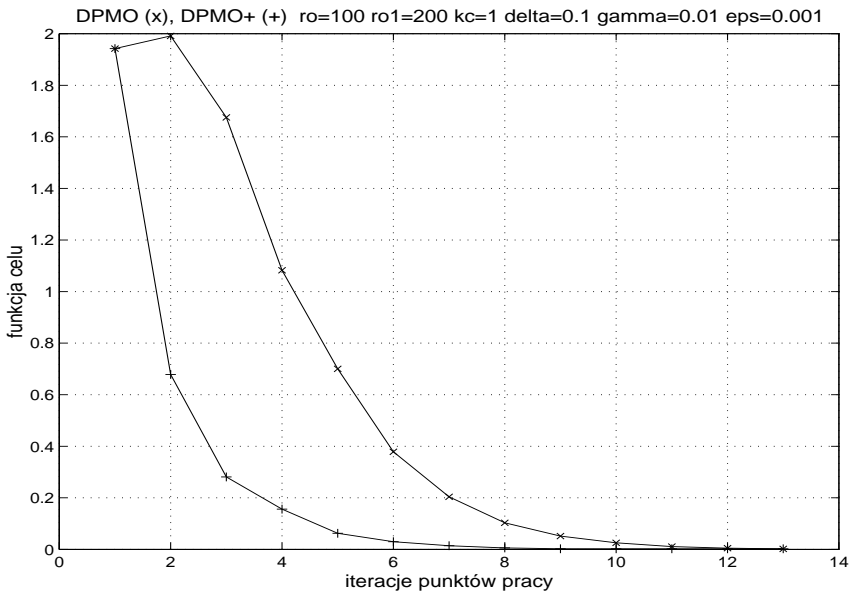
przy czym wektory pochodnych  $(F_j)'(c^i) = [\frac{\partial F_j}{\partial c_1}(c^i) \quad \frac{\partial F_j}{\partial c_2}(c^i)]$ ,  $j = 1, 2$ , liczymy bezpośrednio ze wzorów (4.71a) i (4.71b), zaś wektory pochodnych odwzorowania obiektu  $(F_{*j})'(c^i) = [\frac{\partial F_{*j}}{\partial c_1}(c^i) \quad \frac{\partial F_{*j}}{\partial c_2}(c^i)]$ ,  $j = 1, 2$  będą w trakcie działania algorytmów DPMO estymowane z rozwiązania układów równań liniowych

$$\mathbf{S}^i \begin{bmatrix} \frac{\partial F_{*j}}{\partial c_1}(c^i) \\ \frac{\partial F_{*j}}{\partial c_2}(c^i) \end{bmatrix} = \begin{bmatrix} F_{*j}(c^{i-1}) - F_{*j}(c^i) \\ F_{*j}(c^{i-2}) - F_{*j}(c^i) \end{bmatrix}, \quad j = 1, 2,$$

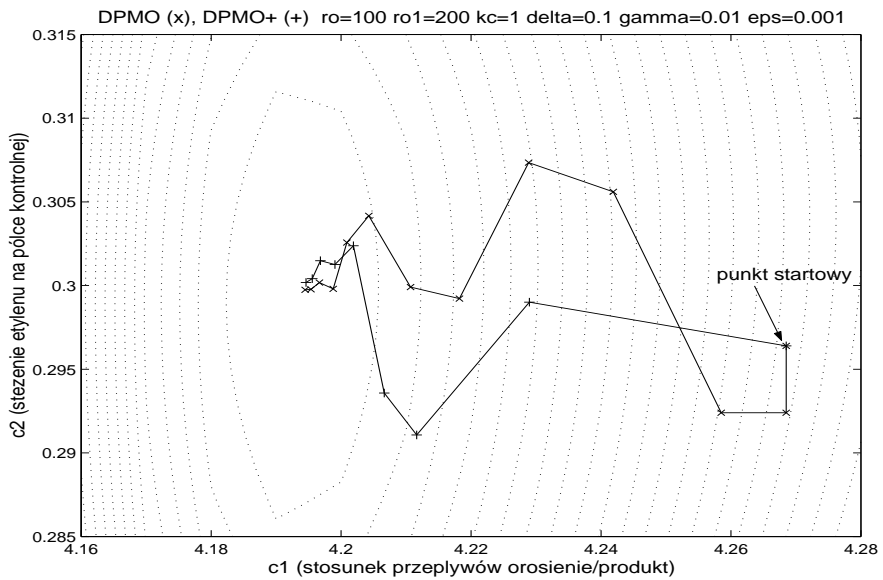
gdzie  $\mathbf{S}^i = \begin{bmatrix} c^{i-1} - c^i & c^{i-2} - c^i \end{bmatrix}^T$ .

Dla opisanego obiektu i jego modelu przeprowadzono symulację sterowania optymalizującego punkt pracy algorytmami DPMO i DPMO+. Na rysunkach 4.17, 4.18 i 4.19 przedstawiono przebieg sterowania od punktu startowego równego punktowi optymalnemu dla modelu,  $(\hat{c}_{m1}, \hat{c}_{m2}) = (4.2685, 0.2964)$ , tzn. wyznaczonemu z rozwiązania zadania optymalizacji modelowej ZOM.

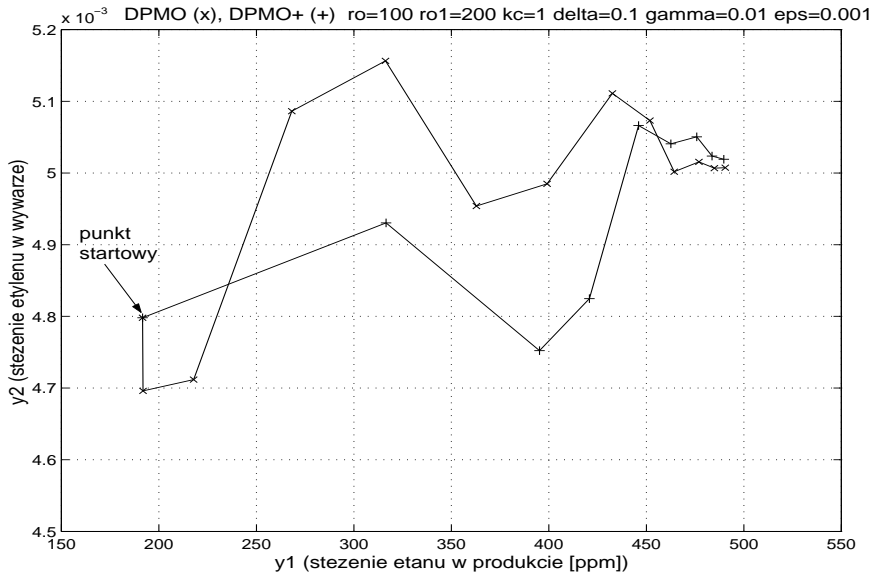
Przyjęto parametry algorytmu:  $\rho = 100$ ,  $k_c = 1$ ,  $\epsilon = 0.001$  oraz dla fazy wstępnej algorytmu DPMO+  $\rho_1 = 200$  i  $\gamma = 0.01$ . Celowo przyjęto bardzo małą wartość  $\epsilon$ , aby pokazać zbieżność nawet w bliskim otoczeniu optimum (w aplikacjach praktycznych sterowania wartość  $\epsilon$  należy przyjmować większą, odpowiednio do błędów pomiarów wyjść). Na rysunku 4.17 przedstawiono trajektorie funkcji celu, na rysunku 4.18 trajektorie zmiennych decyzyjnych na tle poziomic rzeczywistej funkcji celu  $Q(c, F_*(c))$ , a na rysunku 4.19 trajektorie wyjść obiektu. Zwraca uwagę istotne poprawienie efektywności w pierwszych iteracjach przy zastosowaniu zoptymalizowanej



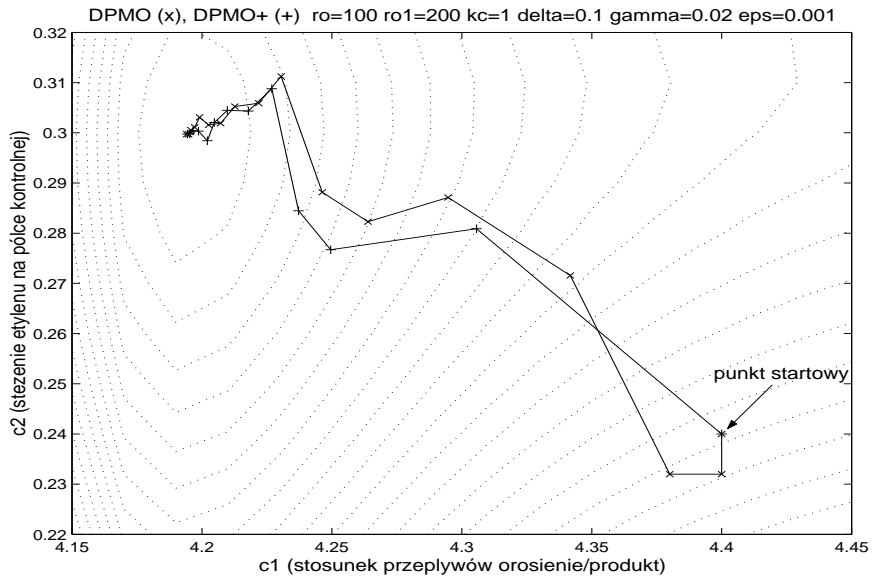
Rysunek 4.17. Trajektorie funkcji celu dla algorytmów DPMO i DPMO+, punkt startowy optymalny zadania optymalizacji modelowej



Rysunek 4.18. Trajektorie punktów pracy dla algorytmów DPMO i DPMO+, punkt startowy optymalny zadania optymalizacji modelowej



Rysunek 4.19. Trajektorie wyjść obiektu dla algorytmów DPMO i DPMO+, punkt startowy optymalny zadania optymalizacji modelowej



Rysunek 4.20. Trajektorie punktów pracy dla algorytmów DPMO i DPMO+, punkt startowy bardziej odległy

fazy wstępnej (algorytm DPMO+), mimo iż startujemy z punktu modelowo optymalnego. Powodem jest zastosowanie wektora przesunięć  $a^i$  i stosunkowo dobre oddanie trendów zmienności wyjść przez odwzorowania modelowe, mimo ich strukturalnej niedokładności. Symulowano również sterowanie startując z innych, w tym bardziej odległych punktów początkowych. Przykładowe trajektorie zmiennych decyzyjnych pokazane są na rysunku 4.20. Nie przytaczamy trajektorii funkcji celu i wyjść, gdyż wyglądają one podobnie jak poprzednio, są tylko nieco dłuższe – analogicznie jak trajektorie na rysunku 4.20 w stosunku do trajektorii na rysunku 4.18.

### Ograniczenia wyjść

Rozważaliśmy dotychczas sytuację bez ograniczeń na zmienne wyjściowe  $y$  obiektu regulowanego. Ograniczenia na te zmienne mają charakter ograniczeń technologicznych reprezentując maksymalne czy minimalne wartości, których przekroczenie może powodować bądź zagrożenie bezpieczeństwa prowadzenia procesu, bądź przekroczenie wymagań na jakość produktu (np. przekroczenie maksymalnego dopuszczalnego poziomu zanieczyszczeń). Przy występowaniu ograniczenia technologicznego na wyjściu wyróżnić można dwie sytuacje:

1. Ograniczenie powinno być aktywne zawsze w nominalnych warunkach pracy, tj. przy zmieniających się wartościach punktu pracy i zakłóceń,
2. Ograniczenie na wyjścia jest aktywne lub nie, w zależności od aktualnej sytuacji technologicznej, od aktualnych wartości punktu pracy i zakłóceń.

W pierwszym przypadku, szczególnie jeśli ograniczenia są ważne i istotne jest ich spełnienie również w stanach przejściowych, dynamicznych, celowe jest zastosowanie dla ich kontroli regulacji nadrzędnej, tj. sterowania w pełnej strukturze warstwowej przedstawionej na rysunku 1.3. Zagadnieniu temu poświęcony jest rozdział 4.4. Natomiast w przypadku drugim sytuacja jest trudniejsza. Trzeba tu stosować albo układy regulacji nadrzędnej o zmiennej strukturze (z przełączaniem regulatorów), albo rezygnując z regulatora nadrzędnego o zmiennej strukturze, zdać się na jedynie modelowe spełnianie ograniczeń w stanach ustalonych, wynikające z wartości punktów pracy  $c$  wyliczanych w zadaniu optymalizacji uwzględniającym ograniczenia. Przy czym można je uwzględniać w funkcji celu (jak np. w przykładzie 4.2), lub bezpośrednio jako ograniczenia. Oczywiście, wartości ograniczeń trzeba wówczas ustawiać z odpowiednimi strefami bezpieczeństwa – z reguły tym większymi, im większa wariancja sygnałów wyjściowych, powodowana wpływem zakłóceń (błędów pomiarowych) oraz

im większa niepewność modelu, co prowadzi do określonej utraty optymalności.

Przedstawione w niniejszym rozdziale algorytmy iteracyjnego sterowania optymalizującego punkty pracy można stosunkowo łatwo rozszerzyć na przypadek bezpośredniego uwzględniania modelowych ograniczeń wyjść, nawet dość ogólnej postaci

$$h(c, y) \leq 0. \quad (4.73)$$

Odpowiednie algorytmy zaproponowano już w pracach [11, 84], dla podstawowej wersji metody ISOPE. Uwzględnienie ograniczeń (4.73) prowadzi do wzoru na mnożnik  $\lambda$  zależny również od mnożników Kuhna-Tuckera  $\eta$  odpowiadających tym ograniczeniom. Zmodyfikowane zadanie optymalizacji modelowej rozwiązywane jest przy uwzględnieniu ograniczeń (4.73), oprócz punktu minimalizującego wyznaczamy też wartość mnożników  $\eta^i$  odpowiadających tym ograniczeniom. Następnie iterujemy łącznie punkty pracy  $c^i$  i mnożniki  $\eta^i$ . Zainteresowanego czytelnika odsyłamy do oryginalnych opracowań [11, 84], koncentrując się dalej na rozważeniu regulacyjnej kontroli ograniczeń wyjść.

#### 4.4. Iteracyjna optymalizacja punktu pracy w strukturze z wyjściami ograniczonymi regulacyjnie

Rozdział niniejszy poświęcony jest iteracyjnym algorytmom sterowania optymalizującego punkty pracy (wartości zadane regulatorów) w pełnej strukturze warstwowej przedstawionej na rysunku 1.3, czyli z zastosowaniem regulacji nadrzędnej do kontroli ograniczeń na pewne składowe wektora wyjść obiektu regulowanego, ograniczeń zawsze aktywnych w nominalnych warunkach pracy.

Rozważać będziemy przypadek ograniczeń postaci

$$y \in Y = \{y \in \mathbb{R}^{n_y} : y_j \leq y_{rj}^d, j = 1, \dots, n_{yd}\},$$

gdzie  $n_{yd} \leq n_y$ . Dokładnie takiej sytuacji odpowiada przedstawiona na rys. 1.3 struktura sterowania warstwowego z regulacyjną nadrzędną kontrolą ograniczeń części wyjść procesowych. W sytuacji tej rozróżniamy wyjścia swobodne  $y^f \in \mathbb{R}^{n_{yf}}$  oraz wyjścia regulowane  $y^d \in \mathbb{R}^{n_{yd}}$ ,  $n_{yf} + n_{yd} = n_y$ ,  $y = (y^f, y^d) \in \mathbb{R}^{n_y}$ , tak jak to przedstawiono na rys. 1.3. Regulator nadrzędny oddziałuje bezpośrednio na część  $c^d \in \mathbb{R}^{n_{cd}}$  ("d" – od ang. *dependent*) wektora wartości zadanych dla warstwy regulacji bezpośredniej  $c$ , aby wymusić równość  $y^d = y_{rj}^d$ . Pozostała, podlegająca swobodnej optymalizacji część wektora  $c \in \mathbb{R}^{n_c}$  oznaczona jest przez  $c^f$  ("f" – od ang. *free*),

$c^f \in \mathbb{R}^{n_{cf}}$ ,  $n_{cf} + n_{cd} = n_c$ . Odwzorowanie obiektu regulowanego (4.22) przyjmuje teraz postać zdekomponowaną na dwie części

$$y^f = F_*^f(c^f, c^d, w), \quad (4.74)$$

$$y^d = F_*^d(c^f, c^d, w). \quad (4.75)$$

Stąd zadanie bieżącej optymalizacji punktów pracy w warunkach niepewności, czyli *zadanie sterowania optymalizującego* punktu pracy, w rozważanym przypadku z wyjściami swobodnymi i ograniczonymi można zapisać w postaci

$$\begin{aligned} & \min Q(c^f, c^d, y^f, y^d) \\ \text{z ogr.: } & y^f = F_*^f(c^f, c^d, w), \\ & y^d = F_*^d(c^f, c^d, w), \quad y^d = y_r^d, \\ & g(c^f) \leq 0. \end{aligned} \quad (4.76)$$

Zadanie (4.76) może być bezpośrednio, zadowalająco zrealizowane poprzez obliczenie odpowiadającego mu zadania optymalizacji z modelami odwzorowań  $F_*^f$  i  $F_*^d$  jedynie w sytuacji dostatecznie dokładnych modeli. Jednakże w praktyce przemysłowej jest to sytuacja raczej nietypowa. Stąd istotą sterowania optymalizującego jest generowanie takich wartości punktów pracy  $c$ , aby uzyskać wartości funkcji celu jak najbardziej zbliżone do optymalnych rozwiązań zadania (4.76) – mimo niedokładności modelowania odwzorowań obiektu i estymat zakłóceń. Środkiem do tego celu, analogicznie jak to przedstawiliśmy uprzednio dla sytuacji bez ograniczeń na wyjścia, są algorytmy iteracyjne wykorzystujące sprzężenie zwrotne [149, 141, 36, 15].

Zwróćmy uwagę, że w zadaniu (4.76) ograniczenia nierównościowe przyjęto jedynie na swobodne sterowania optymalizujące  $c^f$ ,  $g(c^f) \leq 0$ . Powodem takiego postępowania jest fakt, że jako zmienne  $c^d$  powinny być wybrane takie spośród składowych wektora  $c$ , które nie dochodzą do wartości ograniczających podczas pracy procesu – w przeciwnym przypadku regulator nadrzędny nie będzie dysponował swobodą konieczną do realizacji swego celu, tj. utrzymywania ograniczenia  $y^d = y_r^d$ . Zauważmy bowiem (zob. rys. 1.3), że ograniczenia na  $c^d$  to ograniczenia na  $c_r^d + \delta c^d$ ,  $c^d = c_r^d + \delta c^d$ , gdzie  $c_r^d$  jest wartością ustaloną.

Sformułowanie (4.76) to *pełne* zadanie sterowania optymalizującego, występują tu zarówno wyjścia swobodne  $y^f$ , jak i wyjścia ograniczone  $y^d$  dynamicznie stabilizowane przez regulator nadrzędny na wartościach  $y_r^d$ . Algorytmy dla takiego zadania są w istocie złożeniem prezentowanych uprzednio algorytmów dla przypadku z wszystkimi wyjściami swobodnymi oraz algorytmów, w których wszystkie składowe wektora wyjść są ograniczone

(algorytmów dla wyjść ograniczonych). Dlatego też w rozdziale niniejszym, dla uproszczenia wyводу, pominiemy wyjścia swobodne i rozważymy algorytm dla następującego zadania sterowania optymalizującego z wyjściami ograniczonymi, które oznaczymy przez ZSOY:

$$\begin{aligned} & \min Q(c^f, c^d, y^d) \\ \text{z ogr.: } & y^d = F_*^d(c^f, c^d, w), \quad y^d = y_r^d, \\ & g(c^f) \leq 0. \end{aligned} \quad (4.77)$$

Czytelnika zainteresowanego sformułowaniem i analizą przypadku pełnego (jednocześnie z wyjściami swobodnymi i ograniczonymi) odsyłamy do pracy [150].

O obiekcie regulowanym, wejściach niesterowanych (zakłóceniach) i o algorytmach sterowania optymalizującego będziemy czynić założenia analogiczne jak sformułowane uprzednio dla sytuacji bez ograniczeń wyjść. W szczególności, zakładamy odpowiednią wolnozmiennność zakłóceń  $w$  (reprezentowanych modelem w postaci estymaty  $w_d$ ), przyjmowanych jako stałe w trakcie pojedynczego przebiegu algorytmu sterowania optymalizującego.

Model  $F^d$  rzeczywistego odwzorowania procesowego  $F_*^d$  zapiszemy w postaci

$$y^d = F^d(c^f, c^d, w_d). \quad (4.78)$$

Zastępując w sformułowaniu zadania ZSOY (4.77) odwzorowanie  $F_*^d$  modelem  $F^d$  otrzymujemy następujące zadanie optymalizacji modelowej, które oznaczymy przez ZOMY:

$$\begin{aligned} & \min Q(c^f, c^d, y^d) \\ \text{z ogr.: } & y^d = F^d(c^f, c^d, w_d), \quad y^d = y_r^d, \\ & g(c^f) \leq 0. \end{aligned} \quad (4.79)$$

Podobnie jak w przypadku zadania sterowania optymalizującego bez ograniczeń na wyjścia, mamy dwie klasy algorytmów dla rozważanej tu sytuacji z ograniczeniami:

1. Algorytm suboptymalny metody powtarzanej optymalizacji z korektą estymat zakłóceń, odpowiednio sformułowany.
2. Algorytmy optymalne powtarzanej modyfikowanej optymalizacji (PMO), odpowiednio dostosowane.

#### 4.4.1. Algorytm powtarzanej optymalizacji z korektą estymat zakłóceń

W rozważanym w rozdziale 4.3.2 algorytmie powtarzanej optymalizacji z korektą estymat zakłóceń informacją zwrotną była różnica wartości wyjść (swobodnych) mierzonych w procesie i wyliczanych w modelu, dla tych samych wartości zadanych dla regulatorów (tych samych punktów pracy)  $c^i$ . W strukturze z wyjściami stabilizowanymi przez regulator nadrzędny na zadanych, ustalonych wartościach różnica ta jest zerowa. Dysponujemy natomiast innego rodzaju informacją, dostarczaną w stanach ustalonych w pętli sprzężenia zwrotnego. Mianowicie, regulator nadrzędny stabilizując  $y^d$  na wartościach  $y_r^d$  używa w tym celu niezerowych sterowań (zob. rys. 1.3), jeśli tylko model lub estymata wejść niesterowanych obciążone są błędami. Sterowania  $\delta c^d$  ("poprawki") są znane i mogą być wykorzystane w algorytmie sterowania optymalizującego do korekty estymaty zakłóceń, jak to przedstawimy poniżej.

Założymy, że liczba  $\dim w$  wejść niesterowanych (i ich estymat) jest równa lub większa od liczby  $n_{yd}$  wyjść regulowanych  $y^d$ . Wybieramy  $n_{yd}$  wejść niesterowanych najbardziej istotnych, tj. najsilniej wpływających na wartości wyjść  $y^d$ , a pozostałe pomijamy w aktualnych rozważaniach – traktując je jako stałe i mniej interesujące (nie wyróżniane) parametry modelu wyjść. Stąd będziemy mieli  $\dim w_d = n_{yd}$ . Założymy ponadto, że poziomy stabilizacji  $y_r^d$  są stałe, niezależne od zakłóceń i pominiemy błędy wynikające z niedokładności (nieidealności) działania regulacji nadrzędnej.

Przyjmijmy teraz, że dla aktualnej estymaty istotnych zakłóceń  $w_d = w_d^{(0)}$  rozwiązano zadanie optymalizacji modelowej ZOMY (4.79), a rozwiązanie oznaczono przez  $\hat{c}_m^{(0)} = (\hat{c}_m^{f(0)}, \hat{c}_r^{d(0)})$ . Wartości zadane  $\hat{c}_m^{(0)}$  zastosowano następnie w układach regulacji. Jednakże jedynie punkt  $\hat{c}_m^{f(0)}$  jest bezpośrednio stosowany jako wartości zadane  $c^f$  regulatorów bezpośredniego działania – wartość  $\hat{c}_r^{d(0)}$  jest w istocie jedynie *modelowo rekomendowaną* dla  $c^d$ , do korekty przez regulator nadrzędny (zob. rys. 1.3). Jeśli model procesu byłby idealny i estymaty zakłóceń też idealne, to oczywiście korekta nie byłaby potrzebna ( $\delta c^d = 0$ ). Jednakże, w interesujących nas sytuacjach praktycznych tak nie jest i regulator nadrzędny koryguje wartość zadaną  $\hat{c}_r^{d(0)}$  do wartości  $c^{d(0)} \neq \hat{c}_r^{d(0)}$ .

Bez działania regulatora nadrzędnego wartości wyjść  $y^d$  ustaliłyby się na wartościach  $y^{d(0)} \neq y_r^d$ ,

$$y^{d(0)} = F_*^d(\hat{c}_m^{f(0)}, \hat{c}_r^{d(0)}, w). \quad (4.80)$$

Regulator nadrzędny nie dopuszcza do tego korygując  $\hat{c}_r^{d(0)}$  do takiej wartości  $c^{d(0)} = \hat{c}_r^{d(0)} + \delta c^d$ , że

$$F_*^d(\hat{c}_m^{f(0)}, c^{d(0)}, w) = y_r^d.$$

Wykorzystamy informację zawartą w wartości wyjścia regulatora nadrzędnego  $\delta c^d$  do wyliczenia skorygowanej wartości estymaty istotnych zakłóceń: takiej wartości  $w_d^{(1)}$ , że spełnione jest modelowe równanie stabilizacji wyjść

$$F^d(\hat{c}_m^{f(0)}, c^{d(0)}, w_d^{(1)}) = y_r^d. \quad (4.81)$$

W tym celu należy rozwiązać, w ogólności nieliniowy, układ  $n_{yd}$  równań (4.81) względem  $w_d^{(1)}$ . Ponieważ operujemy na modelu niedokładnym, to w praktyce powinno wystarczyć policzenie zadania prostszego i pewnego w rozwiązaniu, a mianowicie rozwiązanie liniowego przybliżenia układu równań (4.81), tj. układu równań postaci

$$(F^d)'_{c^d}(\hat{c}_m^{f(0)}, \hat{c}_r^{d(0)}, w_d^{(0)})[c^{d(0)} - \hat{c}_r^{d(0)}] + (F^d)'_w(\hat{c}_m^{f(0)}, \hat{c}_r^{d(0)}, w_d^{(0)})[w_d^{(1)} - w_d^{(0)}] = 0. \quad (4.82)$$

Zadanie optymalizacji modelowej ZOMY może być teraz ponownie rozwiązane, z estymatą istotnych zakłóceń  $w_d = w_d^{(1)}$  itd. dopóki wyznaczane korekcyjne estymaty wejść niesterowanych  $w_d^{(i)}$  są na tyle istotne, że prowadzą do wartych do zastosowania zmian punktów pracy  $\hat{c}_m^{(i)} = (\hat{c}_m^{f(i)}, \hat{c}_r^{d(i)})$ .

Reasumując, *metoda powtarzanej optymalizacji z korekcją estymat zakłóceń* składa się z dwóch następujących, iteracyjnie powtarzanych etapów:

- Etapu rozwiązania modelowego zadania optymalizacji ZOMY z aktualną wartością estymat istotnych wejść niesterowanych (zakłóceń), zastosowania uzyskanego punktu pracy do obiektu regulowanego i odczytu wartości wyjść regulatora nadrzędnego  $\delta c^d$ , w stanie ustalonym po zaniknięciu przebiegów przejściowych.
- Etapu korekcji estymat istotnych zakłóceń  $w_d$  w oparciu o wartość wyjść  $\delta c^d$  regulatora nadrzędnego, poprzez rozwiązanie układu równań liniowych (4.82) lub oryginalnego układu równań (4.81).

Przedstawiony powyżej algorytm został zaproponowany w pracy [35] (zob. też [36]), gdzie znaleźć też można wyniki symulacji układu sterowania dla przykładu pojedynczej kolumny destylacyjnej. Podstawowa struktura algorytmu jest analogiczna do struktury algorytmu przedstawionego w rozdziale 4.3.2, stąd i jego cechy są analogiczne do tam omówionych.

Jeśli obok wyjść stabilizowanych  $y^d$  występują także wyjścia swobodne  $y^f$ , to należy zastosować bezpośrednie połączenie omówionego wyżej algorytmu i algorytmu powtarzanej optymalizacji z korektą zakłóceń przedstawionego w rozdziale 4.3.2. Wówczas w etapie pierwszym, po optymalizacji i zastosowaniu uzyskanego punktu pracy do obiektu regulowanego,

mierzymy zarówno wartości wyjść swobodnych, jak i wyjść regulatora nadrzędnego, a w etapie drugim (korekty) dokonujemy modyfikacji istotnych zakłóceń występujących zarówno w modelach wyjść swobodnych, jak i stabilizowanych procesu.

#### 4.4.2. Metoda powtarzanej modyfikowanej optymalizacji (PMO)

Rozważamy nadal *zadanie sterowania optymalizującego z wyjściami ograniczonymi* ZSOY (4.77). Dla zwięzłości zapisu pominiemy natomiast dalej symbole wejść niesterowanych  $w$  w odwzorowaniu  $F_*^d$  i ich estymat  $w_d$  w odwzorowaniu  $F$  (analogicznie jak to czyniliśmy w rozdziale 4.3.3) – są to bowiem w trakcie iteracji algorytmu wartości stałe. Stąd mamy zadanie ZSOY w postaci

$$\begin{aligned} \min Q(c^f, c^d, y^d) \\ \text{z ogr.: } y^d = F_*^d(c^f, c^d), \quad y^d = y_r^d, \\ g(c^f) \leq 0. \end{aligned} \quad (4.83)$$

Zauważmy, że w wyniku działania regulatora nadrzędnego zmienne  $c^d$  są *zależne* od zmiennych swobodnych  $c^f$  poprzez odwzorowanie obiektu regulowanego – dla każdej zadanej wartości  $c^f$  regulator nadrzędny ustala  $c^d$  na wartościach (zob. rys. 1.3)

$$c^d = c_r^d + \delta c^d.$$

Istnieje więc zależność funkcyjna  $c^d$  od  $c^f$ , którą oznaczymy przez  $P_*$ ,

$$c^d = P_*(c^f, y_r^d). \quad (4.84)$$

Matematycznie, odwzorowanie  $P_*$  wynika z (4.75) przy warunku  $y^d = y_r^d$ , tj. z rozwikłania względem  $c^d$  układu równań:

$$F_*^d(c^f, c^d) = y_r^d.$$

Założymy, że  $P_*$  jest dobrze określonym jednoznacznym odwzorowaniem i że dysponujemy modelem  $P$  tego odwzorowania,

$$c^d = P(c^f, y_r^d, \beta), \quad (4.85)$$

gdzie przez  $\beta$  oznaczono parametry modelu.

Oznaczmy

$$a^{(i)} = P_*(c^{f(i)}, y_r^d) - P(c^{f(i)}, y_r^d, \beta), \quad (4.86)$$

oraz przeformułujmy zadanie ZSOY (4.83) do następującej równoważnej postaci (podobnie jak to czyniliśmy z zadaniem ZSO w rozdziale 4.3.3, por. z (4.42)):

$$\begin{aligned} \min_{c^f, c^{f(i)}, a} \{ & Q(c^f, P(c^f, y_r^d, \beta) + a, y_r^d) + \rho \|c^{f(i)} - c^f\|^2 \} \\ \text{z ogr. : } & P(c^{f(i)}, y_r^d, \beta) + a = P_*(c^{f(i)}, y_r^d), \\ & g(c^f) \leq 0, \\ & c^{f(i)} = c^f. \end{aligned} \quad (4.87)$$

gdzie  $\rho$  jest współczynnikiem kary dodatkowego członu uwypuklającego (regularyzującego).

Zauważmy, że zadanie sterowania optymalizującego (4.87) ma analogiczną postać jak sformułowane w rozdziale 4.3.3 zadanie (4.42) - jedynie w miejsce zmiennych  $c$  mamy zmienne  $c^f$ , w miejsce odwzorowań  $F$  i  $F_*$  odwzorowania  $P$  i  $P_*$ , parametry modelu oznaczono przez  $\beta$ , a nie przez  $\alpha$  i występuje dodatkowy parametr  $y_r^d$  - stały, a więc nieistotny dla działania algorytmu. Stąd algorytmy powtarzanej modyfikowanej optymalizacji (PMO) wyprowadza się postępując analogicznie jak w rozdziale 4.3.3 - z oczywistymi modyfikacjami wynikającymi z nieco innej struktury sterowania, struktury z regulatorem nadrzędnym. Dlatego też nie będziemy ich tu przedstawiać. Tym bardziej, że po pierwszym wydaniu niniejszej monografii w roku 2002, w roku 2005 ukazała się monografia [15] poświęcona algorytmom iteracyjnego sterowania optymalizującego, gdzie w wyczerpujący sposób przedstawiono omawiane zagadnienie.

## 4.5. Procedury optymalizacji modeli nieliniowych

W rozdziale 4.3.1 sformułowano zadanie sterowania optymalizującego punkt pracy obiektu (zadanie ZSO). Rozważano też sytuację dysponowania na tyle wiarygodnym modelem stanów ustalonych (statyki) obiektu regulowanego, że zadanie sterowania optymalizującego można realizować przez rozwiązywanie zadania optymalizacji opartego na modelu.

Rozważmy zadanie optymalizacji modelowej z deterministyczną reprezentacją wejść niesterowanych  $w_d$  (których jako stałych parametrów nie wymieniamy w formułach poniżej, analogicznie jak w rozdziałach poprzednich)

$$\begin{aligned} \min Q(c, y) \\ \text{z ogr. : } y = F(c, \alpha), \\ g(c) \leq 0, \\ h(y) \leq 0. \end{aligned} \quad (4.88)$$

Jest to na ogół zadanie nieliniowe, przy czym głównym źródłem nieliniowości jest model odwzorowania wyjść obiektu  $y=F(c,\alpha)$ , często dany w postaci niejawnej względem wyjść, jak to przedyskutowano w rozdz. 4.3.1. Dla znajdowania punktów optymalnych zadań optymalizacji modelowej stosować trzeba numeryczne procedury optymalizacji nieliniowej (programowania matematycznego), odpowiednio dobierane do cech rozwiązywanego zadania. Nie są one przedmiotem rozważań niniejszej książki, istnieje bowiem bogata literatura przedmiotu, zob. np. [46, 7, 132] oraz szereg komercyjnych pakietów czy procedur, np. z przybornika *Optimization Toolbox* pakietu MATLAB.

Przedstawione w niniejszej pracy algorytmy powtarzanej modyfikowanej optymalizacji (PMO czy DPMO) też mogą być stosowane do rozwiązywania zadań o strukturze (4.88), stanowiąc pewną alternatywę dla procedur standardowych. Przyczyną jest struktura zadania (4.88), gdzie głównym źródłem nieliniowości i trudności obliczeniowych jest nieliniowe ograniczenie  $y=F(c,\alpha)$ . Jeśli ograniczenie to dane jest w postaci niejawnej,  $\varphi(y, c, \alpha) = 0$ , to wyliczenie każdej wartości wyjść  $y$  wiąże się na ogół z trudnym, iteracyjnym rozwikłaniem nieliniowych równań. W takiej sytuacji procedury optymalizacji zadania (4.88) powinny być dobierane pod kątem jak najmniejszej liczby iteracji, jak najmniejszej liczby zmian sterowań  $c$  i obliczeń wartości wyjść  $y$ . Zauważmy w tym miejscu, że algorytmy sterowania optymalizującego, PMO czy DPMO, skonstruowano właśnie pod tym kątem, tak aby uzyskać jak najmniejszą liczbę zmian punktów pracy.

Przyjmując trudny nieliniowy model obiektu  $F(c,\alpha)$  jako reprezentację odwzorowania obiektu regulowanego  $F_*(c)$ , oraz konstruując jego *znacznie prostszą obliczeniowo aproksymację*  $F_a(c,\alpha_a)$  (np. model liniowy) traktowaną jako „model do sterowania” możemy bezpośrednio zastosować algorytm PMO czy DPMO do optymalizacji zadania (4.88). W trakcie działania tych algorytmów rozwiązywane zadania optymalizacji (zmodyfikowane zadania optymalizacji modelowej ZZOM czy ZZOMU, zob. rozdz. 4.3.3, 4.3.4) będą ze znacznie prostszym odwzorowaniem wartości wyjść  $F_a(c,\alpha_a)$  zamiast  $F(c,\alpha)$ . Oczywiście, odpadają też wówczas kłopoty związane z niedokładnością pomiarów wyjść występujące w przypadku sterowania rzeczywistym obiektem, wartości wyjść „obektu”  $F(c,\alpha)$  wyliczane będą z dokładnością stosowanych procedur numerycznych.

Podobnie można postąpić rozwiązując zadanie optymalizacji modelowej z wyjściami ograniczonymi regulacyjnie (4.79)

$$\begin{aligned} & \min Q(c^f, c^d, y^d) \\ \text{z ogr.: } & y^d = F^d(c^f, c^d, \alpha), \quad y^d = y_r^d, \\ & g(c^f) \leq 0, \end{aligned}$$

które przeformułowuje się do postaci (zob. rozdz. 4.4)

$$\begin{aligned} & \min Q(c^f, c^d, y_r^d) \\ \text{z ogr.: } & c^d = P(c^f, y_r^d, \beta), \\ & g(c^f) \leq 0, \end{aligned} \tag{4.89}$$

gdzie odwzorowanie  $P(c^f, y_r^d, \beta)$  wynika z operacji rozwikłania układu równań  $y_r^d = F^d(c^f, c^d, \alpha)$ , czy z bezpośredniej identyfikacji modelu statyki obiektu regulowanego, z regulatorami bezpośrednimi i z regulatorem nadrzędnym realizującym równanie  $y^d = y_r^d$ . Przyjmując teraz  $P(c^f, y_r^d, \beta)$  jako odwzorowanie „rzeczywistego obiektu regulowanego  $P_*(c^f, y_r^d)$ ” oraz konstruując znacznie prostszy obliczeniowo jego „model”  $c^d = P_a(c^f, y_r^d, \beta_a)$  możemy bezpośrednio zastosować algorytmy PMO czy DPMO.

Zauważmy na koniec, że ogólna struktura algorytmów PMO, polegająca na rozwiązywaniu ciągu prostszych, z iteracji na iterację modyfikowanych zadań optymalizacji jest podobna do struktury algorytmów SQP (*Sequential Quadratic Programming*), uważanych aktualnie za jedno z najefektywniejszych dla nieliniowych zadań optymalizacji z ograniczeniami, zob. np. [46, 7], jak również do struktury algorytmów optymalizacji opartych na tzw. zasadzie modelu pomocniczego (*auxiliary problem principle*) [28].

## Podsumowanie

W książce przedstawiono algorytmy sterowania zaawansowanego stanowiące odpowiedź na współczesne wyzwania, wynikające z możliwości stosowania w czasie rzeczywistym bieżącej optymalizacji punktów pracy sterowanych obiektów, w powiązaniu ze złożonymi algorytmami regulacji.

Dominującym nurtem omawianych zagadnień są algorytmy regulacji zaawansowanej, którym poświęcono drugi i trzeci rozdział książki. Autor postawił przed sobą zadanie przedstawienia technik nowoczesnych, a jednocześnie dobrze już ugruntowanych w praktyce przemysłowej, czy też w jego przekonaniu mających wszelkie szanse na taką pozytywną weryfikację praktyczną. Stąd w rozdziale drugim skoncentrowano się na nieliniowych algorytmach rozmytych typu Takagi-Sugeno (TS), zaś w rozdziale trzecim na algorytmach predykcyjnych z przesuwającym horyzontem.

Zdaniem nie tylko autora, algorytmy rozmyte TS oferują nieliniową technikę regulacji skuteczną, stosunkowo łatwą w projektowaniu i przekonującą, która może być traktowana jako naturalne nieliniowe uogólnienie znanych liniowych algorytmów regulacji. W szczególności, mamy tu uogólnienie z dawna ugruntowanej techniki PID w postaci nieliniowych rozmytych regulatorów PID, kiedy jednym z zasadniczych etapów projektowania jest dobór nastaw liniowych regulatorów PI czy PID w odpowiednio wybranym zestawie punktów pracy obiektu. Stąd, stanowiąc pełną alternatywę projektową, nieliniowe rozmyte regulatory TS są jednocześnie szczególnie warte stosowania tam, gdzie trzeba przejść od algorytmów liniowych do nieliniowych, nie tracąc przy tym zgromadzonych doświadczeń. W odróżnieniu od standardowej techniki rozmytej, gdzie następnikami reguł wnioskowania są zmienne rozmyte, w algorytmach typu TS następniki reguł są funkcyjne, co umożliwia projektowanie złożonych algorytmów nieliniowych przy stosunkowo niewielkiej liczbie reguł. W procesie nieliniowego modelowania czy projektowania reguł regulatora można przy tym wykorzystać bogatą wiedzę dotyczącą algorytmów liniowych. Przy obecnym poziomie techniki elektronicznej algorytmy rozmyte TS mogą być z powodzeniem stosowane

w podstawowych pętlach warstwy regulacji bezpośredniej. Mogą być również wykorzystywane, w razie potrzeby, w pętlach regulacji nadrzędnej.

Omawiana w rozdziale trzecim regulacja predykcyjna jest zaawansowaną techniką regulacji, która odniosła olbrzymi, niekwestionowany sukces w zastosowaniach praktycznych wywierając w ostatnich latach dominujący wpływ zarówno na kierunek rozwoju przemysłowych układów regulacji i sterowania, jak i prac badawczych w tym obszarze. Powody tego faktu wymienialiśmy już uprzednio zarówno w przedmowie, jak i szerzej w rozdziale trzecim. Przypomnijmy, że algorytmy regulacji predykcyjnej są w istocie pierwszą techniką w sposób bezpośredni uwzględniającą ograniczenia zarówno sygnałów sterujących, jak i wielkości wyjściowych, generując przy tym sterowania biorące pod uwagę również interakcje wewnętrzne w obiekcie, dzięki bezpośredniemu wykorzystaniu modelu. Stąd są aktualnie bezkonkurencyjne w zastosowaniu do obiektów wielowymiarowych z silnymi interakcjami, nawet przy braku ograniczeń. Ponadto są to algorytmy, których zasada działania i dobór nastaw są łatwe do wyjaśnienia personelowi technicznemu. W związku ze zwiększeniem mocy obliczeniowych i niezawodności oraz spadkiem cen procesorów możliwe jest aktualnie wprowadzanie algorytmów predykcyjnych nie tylko do regulacji nadrzędnej (gdzie są techniką dominującą), ale i do regulacji bezpośredniej, do podstawowych pętli regulacyjnych.

W pierwszej części rozdziału trzeciego przedstawiono najważniejsze, ugruntowane już w praktyce przemysłowej algorytmy liniowe oparte na modelach typu wejście-wyjście (algorytmy DMC i GPC), jak i wykorzystujący model w postaci liniowych równań stanu algorytm MPC. Samo przedstawienie jest w szeregu punktach niestandardowe, nowatorskie: można tu wymienić implementacje praw regulacji predykcyjnej w strukturach uwzględniających działanie typu *anti-windup*, tzw. alternatywną postać prawa regulacji GPC, proste sformułowanie algorytmu GPC bezpośrednio dla modelu zakłóceń typu DMC, sformułowania algorytmu MPC (MPC z modelem równań stanu) z modelem zakłóceń deterministycznych w postaci zakłóceń stanu, dla przypadków zarówno z pomiarem stanu jak i z estymacją stanu. Druga część rozdziału poświęcona jest nieliniowym algorytmom regulacji predykcyjnej. Wychodząc od ogólnego nieliniowego sformułowania zadania regulacji predykcyjnej, wskazano na jego podstawową dla aplikacji praktycznych wadę, jaką jest brak gwarancji znajdowania minimum globalnego nieliniowego zadania optymalizacji, jak i zakończenia działania procedury optymalizacji nieliniowej w określonym czasie. Stąd dużo uwagi poświęcono algorytmom nieliniowym suboptymalnym z linearyzacjami, rozwiązującym w każdym kroku (okresie próbkowania) jedynie zadanie optymalizacji

zacji kwadratowej (wypukłego programowania kwadratowego). Przedstawiono algorytmy z sukcesywną linearyzacją (typu NSL), z nieliniową predykcją i linearyzacją w punkcie (typu NPL) lub wokół trajektorii (typu NPLT), zaproponowano algorytm z iteracyjnym dostrajaniem nieliniowej predykcji i zadania optymalizacji kwadratowej w każdym kroku (NPLT+). Zaprezentowane techniki nieliniowej regulacji predykcyjnej omówiono bliżej dla przypadku modelu obiektu nieliniowego rozmytego typu TS, kiedy to szczególnie łatwo przeprowadza się bieżącą linearyzację modelu. Przedstawiono mechanizmy zapewniania nominalnej stabilności zarówno liniowych, jak i nieliniowych algorytmów regulacji predykcyjnej. Końcowe fragmenty rozdziału poświęcono zagadnieniom zapewniania realizowalności (niepustości zbioru dopuszczalnego) oraz doboru wartości parametrów dostrajalnych algorytmów.

Rozdział czwarty poświęcony jest algorytmom optymalizacji bieżącej stanu ustalonego (optymalizacji bieżącej wartości zadanych regulatorów). Po omówieniu miejsca optymalizacji w strukturze warstwowej i zadania optymalizacji stanu ustalonego, w pierwszej części rozdziału skoncentrowano się na algorytmach bieżącego dostrajania punktu pracy regulatorów MPC, tzw. algorytmach SSTO (*Steady State target Optimization*), bezpośrednio współpracujących z tymi regulatorami w każdym okresie próbkowania. Po omówieniu algorytmów SSTO dla regulacji MPC z liniowymi modelami obiektów, zaproponowano sformułowanie algorytmu SSTO dla algorytmów z nieliniową predykcją i linearyzacją, typu NPL i NPLT. Druga część rozdziału koncentruje się na algorytmach iteracyjnej optymalizacji punktów pracy w warunkach istotnej niepewności powodowanej niedokładnym modelem obiektu regulowanego i błędami estymacji zakłóceń. Podano sformułowania algorytmów i wyniki badań zarówno dla podstawowej warstwowej struktury sterowania, jak i dla struktury z nadrzędną regulacyjną kontrolą ważnych ograniczeń technologicznych narzuconych na wyjścia obiektu.

Autor postawił sobie zadanie takiego napisania niniejszej książki, aby była ona przydatna nie tylko dla stosunkowo wąskiego kręgu pracowników naukowych i doktorantów zajmujących się problematyką regulacji zaawansowanej i optymalizacji punktów pracy, ale aby służyła również (a może przede wszystkim) inżynierom projektującym czy eksploatującym zaawansowane algorytmy sterowania. Chodziło też o to, by była ona przydatna studentom wyższych lat studiów. Stąd też dużą uwagę poświęcono pełnemu przedstawieniu samych technik i algorytmów, ich ilustracji licznymi przykładami. Nie rezygnując z analizy teoretycznej ograniczono ją jednak do niezbędnego minimum, podając przy tym odnośniki do literatury, w tym do prac własnych autora i współpracowników. Odnośniki te są adresowane

przede wszystkim do czytelnika zainteresowanego aspektami teoretycznymi, pragnącego zdobyć głębszą czy szerszą wiedzę. Konsekwencją postawionego celu była również eliminacja pewnych zagadnień mniej, zdaniem autora, przydatnych z praktycznego punktu widzenia – ale też z odesłaniem zainteresowanego czytelnika do odpowiednich pozycji literatury.

Zrezygnowano z omawiania zagadnień ważnych, ale bardziej zaawansowanych teoretycznie i znajdujących się ciągle na etapie bardziej badań i weryfikacji, niż sprawdzonych technik. Chodzi tu przede wszystkim o zagadnienie projektowania odpornych algorytmów regulacji, tzn. pracujących zadowalająco przy określonym zakresie zmian zakłóceń czy niepewności modelu. Jest to zagadnienie trudne i wciąż na etapie badań nawet dla algorytmów z modelami liniowymi (z ograniczeniami), nie mówiąc już o algorytmach nieliniowych [98]. Podstawową techniką projektową jest tu *analiza symulacyjna*, wsparta rozumieniem wpływu zmian poszczególnych parametrów dostrajalnych. Stąd duża waga przykładana w pracy do tego zagadnienia, spora liczba odpowiednio dobranych przykładów symulacji układów sterowania, zarówno prostych, jak i trudniejszych, powiązanych z modelami praktycznymi. W najszerszej potraktowanym obszarze algorytmów predykcyjnych wskazano, że kluczowe znaczenie dla odporności ma odpowiedni dobór współczynników wagowych funkcji kryterialnej związanych z przyrostami sygnałów sterujących, starano się pokazać charakter wpływu wartości horyzontów predykcji i sterowania na cechy algorytmu oraz sposoby zapewniania niepustości zbioru dopuszczalnego.

Pominięto w książce kwestię uwzględniania wpływu skończonego czasu potrzebnego na wyliczenie wartości sterowania w sterowniku komputerowym realizującym dyskretne algorytmy regulacji, poczynając od momentu pomiaru wartości wyjść w każdej z kolejnych chwil próbkowania. Mimo nieprawdopodobnego rozwoju mocy obliczeniowych procesorów zagadnienie to jest nadal ważne. Przede wszystkim ma to miejsce w przypadku bardziej obliczeniowo wymagających algorytmów, jakimi są uwzględniające ograniczenia algorytmy predykcyjne w realizacjach numerycznych, w każdym kroku rozwiązujące zadanie optymalizacji. W największym skrócie, czas obliczeń powoduje wprowadzenie do pętli regulacyjnej dodatkowego opóźnienia. Projektant układu regulacji może uwzględnić bezpośrednio opóźnienie szacowane na podstawie czasu obliczeń, ale czasem łatwiejsze i sensowne jest też wprowadzenie opóźnienia równego okresowi impulsowania. Czytelnika odsyłamy tu do literatury poświęconej regulacji dyskretnej, np. [3]. W odniesieniu do modelu obiektu w postaci liniowych równań stanu, zagadnienie poruszone jest w [87].

Problematyka omawiana w niniejszej książce to nadal obszar zarówno

intensywnie rozwijanego i stosowanego w praktyce komercyjnego oprogramowania, jak i wielu prac badawczych. Co więcej, ze względu na stopień trudności badań analitycznych, praktyka wyprzedzała i nadal często wyprzedza teorię. Dobrze to świadczy o przydatności rozważanych podejść i metod. Na podstawowe problemy, kierunki rozwoju zarówno aplikacji, jak i badań wielokrotnie zwracano już uwagę. Zasadnicze z nich w obszarze regulacji zaawansowanej to algorytmy dla procesów nieliniowych, algorytmy odpowiednio niezawodne numerycznie i odporne, w tym potrafiące zapewnić spełnienie ograniczeń technologicznych w warunkach niepewności. Jednym z trendów, widocznym w oprogramowaniu komercyjnym, a raczej słabo w obszarze publikowanych prac badawczych, są próby integracji algorytmów regulacji predykcyjnej i optymalizacji bieżącej punktów pracy. Ma to głęboki sens w sytuacjach dynamiki swobodnych (niesterowanych) wielkości wejściowych porównywalnej z dynamiką obiektu sterowanego, czy też wolniejszej ale na tyle szybkiej, że utrzymywanie przez dłuższe okresy czasu ustalonych wartości zadanych zmiennych regulowanych prowadzi do obniżenia efektywności ekonomicznej.

W głębokim przekonaniu autora algorytmy zaawansowanej regulacji, w tym przede wszystkim regulacji predykcyjnej, oraz bieżącej optymalizacji będą nadal rozwijane, będą stosowane coraz szerzej i efektywniej, nie tylko w instalacjach typowo przemysłowych.



## Bibliografia

- [1] F. Allgöwer, T. Badgwell, J. Qin, J. Rawlings, S. Wright. Nonlinear predictive control and moving horizon estimation – an introductory overview. P. Frank, redaktor, *Advances in Control – Highlights of ECC'99*, rozdział 12. Springer, London, 1999.
- [2] K. Åström, B. Wittenmark. *Adaptive Control*. Addison-Wesley, Reading, Massachusetts, 1995.
- [3] K. Åström, B. Wittenmark. *Computer Controlled Systems*. Prentice Hall, Upper Saddle River, 1997.
- [4] R. Babuska, J. Sousa, H. Verbruggen. Predictive control of nonlinear systems based on fuzzy and neural models. *5th European Control Conference ECC'99*, CD-ROM F1032-5, Karlsruhe, 1999.
- [5] W. Bamberger, R. Isermann. Adaptive on-line steady-state optimization of slow dynamic processes. *Automatica*, 14:223–230, 1978.
- [6] B. Bequette. Nonlinear control of chemical processes: a review. *Industrial & Engineering Chemistry Research*, 30:1391–1413, 1991.
- [7] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, 1995.
- [8] Y. Blanco, W. Perruquetti, P. Borne. Non quadratic stability of nonlinear systems in the Takagi-Sugeno form. *Proceedings 6th European Control Conference ECC'01*, strony 3917–3922, Porto, 2001.
- [9] T. Blevins, G. McMillan, W. Wojsznis, M. Brown. *Advanced Control Unleashed*. The ISA Society, Research Triangle Park, NC, 2003.
- [10] T. Blevins, W. Wojsznis, M. Nixon. *Advanced Control Foundation*. The ISA Society, Research Triangle Park, NC, 2013.
- [11] M. Brdyś, S. Chen, P. D. Roberts. An extension to the modified two-step algorithm for steady-state system optimization and parameter estimation. *International Journal of Systems Science*, 17:1229–1243, 1986.
- [12] M. Brdyś, J. Ellis, P. Roberts. Augmented integrated system optimization and parameter estimation technique: derivation, optimality and convergence. *IEE Proceedings Part D*, 34(3):201–209, 1987.
- [13] M. Brdyś, P. Roberts. Convergence and optimality of modified two-step algorithm for integrated system optimization and parameter estimation. *International Journal of Systems Science*, 18(7):1305–1322, 1987.
- [14] M. Brdyś, P. Tatjewski. An algorithm for steady-state optimising dual control of uncertain plants. *Proceedings 1st IFAC Workshop New Trends in Design of Control Systems*, strony 249–254, Smolenice, 1994.

- [15] M. Brdys, P. Tatjewski. *Iterative Algorithms for Multilayer Optimizing Control*. Imperial College Press/World Scientific, London/Singapore, 2005.
- [16] E. Camacho, C. Bordons. *Model Predictive Control in the Process Industry*. Springer, Berlin, 1995.
- [17] E. Camacho, C. Bordons. *Model Predictive Control*. Springer, London, 1999.
- [18] M. Cannon, B. Kouvaritakis, Y. Lee, A. Brooms. Efficient nonlinear model-based predictive control. *International Journal of Control*, 74(4):361–372, 2001.
- [19] S. Cao, N. Rees, G. Feng. Stability analysis of fuzzy control systems. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, 26(1):201–204, 1996.
- [20] M. Chadli, D. Maquin, J. Ragot. Relaxed stability conditions for Takagi-Sugeno fuzzy systems. *Proceedings IEEE Conference Systems, Man and Cybernetics*, strony 3514–3519, 2000.
- [21] M. Chadli, D. Maquin, J. Ragot. On the stability analysis of multiple model systems. *Proceedings 6th European Control Conference ECC'01*, strony 1894–1899, Porto, 2001.
- [22] J.-S. Chai, S. Tan, Q. Chen, C.-C. Hang. A general fuzzy control scheme for nonlinear processes with stability analysis. *Fuzzy Sets and Systems*, 100:179–195, 1998.
- [23] C. Chen, L. Shaw. On receding horizon feedback control. *Automatica*, 18(3):349–352, 1982.
- [24] H. Chen, F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.
- [25] D. Clarke, C. Mohtadi. Properties of generalised predictive control. *Automatica*, 25:859–875, 1989.
- [26] D. Clarke, C. Mohtadi, P. Tuffs. Generalised predictive control – Parts I and II. *Automatica*, 23:137–160, 1987.
- [27] D. Clarke, R. Scattolini. Constrained receding-horizon predictive control. *IEE Proceedings Part D*, 138(4):347–354, 1991.
- [28] G. Cohen. Optimization by decomposition and coordination: a unified approach. *IEEE Transactions on Automatic Control*, 23:222–232, 1978.
- [29] C. Cutler, B. Ramaker. Dynamic Matrix Control – a computer control algorithm. *Proceedings Joint Automatic Control Conference*, San Francisco, 1980.
- [30] F. Declercq, R. de Keyser. Suboptimal nonlinear predictive controllers. *International Journal of Applied Mathematics and Computer Science*, 9(1):129–148, 1999.
- [31] P. Domański. *Zastosowanie metod jakościowych do modelowania i projektowania układów regulacji*. Praca doktorska. Wydział Elektroniki i Techniki Informatycznych Politechniki Warszawskiej, Warszawa, 1996.
- [32] P. Domański, M. Brdyś, P. Tatjewski. Fuzzy logic multi-regional controllers - design and stability analysis. *Proceedings 4th European Control Conference ECC'97*, CD-ROM TU-E G5, Brussels, 1997.
- [33] P. Domański, M. A. Brdyś, P. Tatjewski. Design and stability of fuzzy-logic

- multi-regional output controllers. *International Journal of Applied Mathematics and Computer Science*, 9(4):883–897, 1999.
- [34] F. Doyle III, B. Ogunnaike, R. Pearson. Nonlinear model-based control using second-order Volterra models. *Automatica*, 31(5):697–714, 1995.
- [35] J. Duda. *Wybrane zagadnienia syntezy algorytmów sterowania nadrzędnego wolnozmiennymi, ciągłymi procesami przemysłowymi*. Zeszyty Naukowe AGH, nr 56 (ser. Automatyka), Kraków, 1991.
- [36] J. Duda. *Modele matematyczne, struktury i algorytmy nadrzędnego sterowania komputerowego*. Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Kraków, 2003.
- [37] J. Duda, M. Brdyś, P. Tatjewski, W. Byrski. Multilayer decomposition for optimizing control of technological processes. *Proceedings IFAC/IFORS/IMACS Symposium on Large Scale Systems Theory and Applications LSS'95*, strony 111–116, London, 1995.
- [38] H. Eder. MBPC benefits and key success factors. *Proceedings 5th European Control Conference ECC'99*, CD-ROM F1033-6, Karlsruhe, 1999.
- [39] A. Feldbaum. *Optimal Control Systems*. Academic Press, New York, 1965.
- [40] W. Findeisen. *Wielopoziomowe układy sterowania*. PWN, Warszawa, 1974.
- [41] W. Findeisen. *Technika regulacji automatycznej*. PWN, Warszawa, 1978.
- [42] W. Findeisen. *Struktury sterowania dla złożonych procesów*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1997.
- [43] W. Findeisen, F. Bailey, M. Brdyś, K. Malinowski, P. Tatjewski, A. Woźniak. *Control and Coordination in Hierarchical Systems*. J. Wiley & Sons, Chichester - New York - Brisbane - Toronto, 1980.
- [44] W. Findeisen, M. Brdyś, K. Malinowski, P. Tatjewski, A. Woźniak. On-line hierarchical control for steady-state systems. *IEEE Transactions on Automatic Control*, 23(2):189–208, 1978.
- [45] M. Fischer, O. Nelles, R. Isermann. Adaptive predictive control of a heat exchanger based on a fuzzy model. *Control Engineering Practice*, 6:259–269, 1998.
- [46] R. Fletcher. *Practical Methods of Optimization*. J. Wiley & Sons, Chichester - New York - Brisbane - Toronto, 1987.
- [47] G. Franklin, D. Powell, A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, Reading, Massachusetts, 1994.
- [48] G. Franklin, J. Powell, M. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, Reading, Massachusetts, 1994.
- [49] R. Franks. *Mathematical Modelling in Chemical Engineering*. J. Wiley, 1972.
- [50] C. Garcia. Quadratic/dynamic matrix control of nonlinear processes: an application to a batch reaction process. *Proceedings AIChE Annual Meeting*, San Francisco, 1984.
- [51] C. Garcia, A. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46:73–87, 1986.
- [52] C. Garcia, D. Prett, M. Morari. Model predictive control: theory and practice – a survey. *Automatica*, 25:335–348, 1989.
- [53] G. Gattu, E. Zafiriou. Nonlinear quadratic dynamic matrix control

- with state estimation. *Industrial & Engineering Chemistry Research*, 31:1096–1104, 1992.
- [54] G. Goodwin, S. Graebe, M. Salgado. *Control System Design*. Prentice Hall, Upper Saddle River, 2001.
- [55] M. Grewal, A. Andrews. *Kalman Filtering - Theory and Practice Using MATLAB*. J. Wiley, 2001.
- [56] M. Henson. Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering*, 23:187–202, 1998.
- [57] A. Isidori. *Nonlinear Control Systems*. Springer, Berlin, 1995.
- [58] J.-S. Jang, C.-T. Sun. Neuro-fuzzy modeling and control. *Proceedings of the IEEE*, 83(3):378–406, 1995.
- [59] J.-S. R. Jang, C.-T. Sun, E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, Upper Saddle River, 1997.
- [60] M. Johansson, A. Rantzer, K.-E. Arzen. Piecewise quadratic stability of fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 7(6):713–722, 1999.
- [61] R. Kalman, J. Bertram. Control system analysis and design via the second method of Lyapunov. I continuous-time systems. II discrete-time systems. *Journal of Basic Engineering Transactions ASME*, 82(2):371–400, 1960.
- [62] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [63] S. Keerthi, E. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizons approximations. *Journal of Optimization Theory and Applications*, 57:265–293, 1988.
- [64] H. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, 1996.
- [65] J. Kościelny. *Diagnostyka zautomatyzowanych procesów przemysłowych*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2001.
- [66] J. Korbicz, J. Kościelny, redaktorzy. *Modelowanie, diagnostyka i sterowanie nadrzędne procesami. Implementacja w systemie DiaSter*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2009.
- [67] J. Korbicz, J. Kościelny, redaktorzy. *Modeling, Diagnostics and Process Control. Implementation in the DiaSter System*. Springer-Verlag, Berlin Heidelberg, 2011.
- [68] J. Korbicz, J. Kościelny, Z. Kowalczyk, W. Cholewa, redaktorzy. *Diagnostyka procesów*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2002.
- [69] W. Kwon, A. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Transactions on Automatic Control*, 22(5):838–842, 1977.
- [70] M. Ławryńczuk. *Nieliniowe algorytmy regulacji predykcyjnej z neuronowymi modelami obiektów*. Rozprawa doktorska. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Warszawa, 2003.
- [71] M. Ławryńczuk. Efficient nonlinear predictive control based on structured neural models. *International Journal of Applied Mathematics and Computer Science*, 19(2):233–246, 2009.
- [72] M. Ławryńczuk. *Sieci neuronowe w efektywnych obliczeniowo algorytmach regulacji predykcyjnej, Prace Naukowe Politechniki Warszawskiej, Elektro-*

- nika, nr 180. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2012.
- [73] M. Ławryńczuk. *Computationally efficient model predictive control algorithms: a neural network approach*. Springer, Heidelberg, 2014.
- [74] M. Ławryńczuk, P. Marusak, P. Tatjewski. Oprogramowanie do projektowania zaawansowanych układów regulacji. *Materiały XIV Krajowej Konferencji Automatyki*, wolumen 1, strony 483–488, Zielona Góra, 2002.
- [75] M. Ławryńczuk, P. Tatjewski. A multivariable neural predictive control algorithm. *Proceedings IFAC Workshop Advanced Fuzzy-Neural Control AFNC'01*, strony 191–196, Valencia, 2001.
- [76] M. Ławryńczuk, P. Tatjewski. A computationally efficient nonlinear predictive control algorithm based on neural models. *Proceedings 8th IEEE Conference Methods and Models in Automation and Robotics MMAR'02*, strony 781–786, Szczecin, 2002.
- [77] M. Ławryńczuk, P. Tatjewski. Nonlinear predictive control based on neural multi-models. *International Journal of Applied Mathematics and Computer Science*, 20(1):7–21, 2010.
- [78] E. Lee, L. Markus. *Foundations of Optimal Control Theory*. J. Wiley, New York, 1967.
- [79] J. Lee, N. Ricker. Extended Kalman filter based nonlinear model predictive control. *Industrial Engineering & Chemistry Research*, 33:1530–1541, 1994.
- [80] I. Lefkowitz. Multilevel approach applied to control system design. *Journal of Basic Engineering Transactions ASME, ser. B*, 2:392–398, 1966.
- [81] F. Leja. *Rachunek różniczkowy i całkowity*. PWN, Warszawa, 1969.
- [82] R. Lewandowski. *Regulacja nieliniowa i wizualizacja w środowisku oprogramowania SCADA i Soft Control*. Praca dyplomowa inżynierska. Politechnika Warszawska, Instytut Automatyki i Informatyki Stosowanej, Warszawa, 2001.
- [83] W. Li, T. Biegler. Multistep, Newton-type control strategies for constrained, nonlinear processes. *Chemical Engineering Research and Design*, 67:562–577, 1989.
- [84] J. Lin, S. Chen, P. Roberts. Modified algorithm for steady-state integrated system optimization and parameter estimation. *IEE Proceedings Part D*, 35:119–126, 1988.
- [85] Z. Liu, P. Roberts. Non-derivative algorithm for optimization and control of steady-state systems. *International Journal of Systems Science*, 20(8):1483–1479, 1989.
- [86] W. Luyben. *Modelowanie, symulacja i sterowanie procesów przemysłu chemicznego*. WNT, Warszawa, 1976.
- [87] J. Maciejowski. *Predictive Control*. Prentice Hall, Harlow, England, 2002.
- [88] U. Maeder, F. Borelli, M. Morari. Linear offset-free model predictive control. *Automatica*, 45:2214 – 2222, 2009.
- [89] L. Magni, G. D. Nicolao, L. Magnani, R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37:1351–1362, 2001.
- [90] T. E. Marlin. *Process Control*. McGraw-Hill, New York, 1995.

- [91] P. Marquis, J. Broustail. SMOC, a bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit. T. McAvoy, Y. Arkun, E. Zafriou, redaktorzy, *Proceedings 1988 IFAC Workshop Model Based Process Control*, strony 37–43. Pergamon Press, Oxford, 1988.
- [92] P. Marusak. *Regulacja predykcyjna obiektów nieliniowych z wykorzystaniem techniki macierzy dynamicznej*. Rozprawa doktorska. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Warszawa, 2002.
- [93] P. Marusak, J. Pułaczewski, P. Tatjewski. Algorytmy DMC z uwzględnieniem ograniczeń sterowania. *Prace XIII Krajowej Konferencji Automatyki*, wolumen 1, strony 113–118, Opole, 1999.
- [94] P. Marusak, P. Tatjewski. Fuzzy Dynamic Matrix Control algorithms for nonlinear plants. *Proceedings 6th International Conference Methods and Models in Automation and Robotics MMAR'00*, wolumen 2, strony 749–754, Międzyzdroje, 2000.
- [95] P. Marusak, P. Tatjewski. Output constraints in fuzzy DMC algorithms with parametric uncertainty in process models. *Proceedings 7th IFAC Conference Methods and Models in Automation and Robotics MMAR'01*, wolumen 1, strony 517–522, Międzyzdroje, 2001.
- [96] P. Marusak, P. Tatjewski. Stability analysis of nonlinear control systems with fuzzy DMC controllers. *Proceedings IFAC Workshop Advanced Fuzzy and Neural Control AFNC'01*, strony 21–26, Valencia, 2001.
- [97] P. Marusak, P. Tatjewski. Stability analysis of nonlinear control systems with unconstrained fuzzy predictive controllers. *Archives of Control Sciences*, 12:267–288, 2002.
- [98] D. Mayne, J. Rawlings, C. Rao, P. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36:789–814, 2000.
- [99] D. Megias, J. Serrano, M. E. Ghomari. Extended linearized predictive control: practical control algorithms for non-linear systems. *Proceedings 5th European Control Conference ECC'99*, CD-ROM F883, Karlsruhe, 1999.
- [100] A. Meszaros, M. Brdyś, P. Tatjewski, P. Lednicky. Multilayer adaptive control of continuous bioprocesses using optimising control technique. *Bio-process Engineering*, 12:1–9, 1995.
- [101] H. Michalska, D. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38:1623–1633, 1993.
- [102] M. Morari, J. Lee. Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23:667–682, 1999.
- [103] M. Morari, U. Maeder. Nonlinear offset-free model predictive control. *Automatica*, 48:2059–2067, 2012.
- [104] M. Morari, N. Ricker. *Model Predictive Control Toolbox, User's Guide ver. 1*. The MathWorks Inc., Natick, MA, 1998.
- [105] K. Muske, T. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12:617–632, 2002.
- [106] K. Muske, J. Rawlings. Model predictive control with linear models. *AICHE Journal*, 39(2):262–287, 1993.

- [107] R. Mutha, W. Cluett, A. Penlidis. Nonlinear model-based predictive control of control nonaffine systems. *Automatica*, 33(5):907–913, 1997.
- [108] G. D. Nicolao, L. Magni, R. Scattolini. Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Transactions on Automatic Control*, 43(7):1030–1036, 1998.
- [109] A. Niederliński, J. Mościński, Z. Ogonowski. *Regulacja adaptacyjna*. PWN, Warszawa, 1995.
- [110] K. Nowosad. *Strukturalne cechy regulacji predykcyjnej*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1995.
- [111] K. Ogata. *Modern Control Engineering*. Prentice Hall, Englewood Cliffs, NJ, 1997.
- [112] S. Osowski. *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa, 1996.
- [113] S. Osowski. *Sieci neuronowe do przetwarzania informacji*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2000.
- [114] T. Parisini, M. Sanguinetti, R. Zoppoli. Nonlinear stabilization by receding-horizon neural regulators. *International Journal of Control*, 70(3):341–362, 1998.
- [115] W. Pedrycz. *Fuzzy Control and Fuzzy Systems*. Research Studies Press, Taunton, England, 1993.
- [116] A. Piegat. *Modelowanie i sterowanie rozmyte*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 1999.
- [117] A. Płanowski, W. Ramm, S. Kagan. *Procesy i aparaty w technologii chemicznej*. PWN, Warszawa, 1974.
- [118] W. Press, S. Teukolsky, W.T.Vetterling, B.P.Flannery. *Numerical Recipes. Third Edition*. Cambridge University Press, Cambridge, USA, 2007.
- [119] D. Pretz, R. Gillette. Optimization and constrained multivariable control of catalytic cracking unit. *Proceedings Joint Automatic Control Conference*, San Francisco, 1980.
- [120] S. Qin, T. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.
- [121] V. Rao, J. B. Rawlings. Steady states and constraints in model predictive control. *AIChE Journal*, 45(6):1266–1278, 1999.
- [122] J. Rawlings, K. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38:1512–1516, 1993.
- [123] J. Richalet, A. Rault, J. Testud, J. Papon. Algorithmic control of industrial processes. *Proceedings 4th IFAC Symposium Identification and System Parameter Estimation*, strony 1119–1167, 1976.
- [124] J. Richalet, A. Rault, J. Testud, J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.
- [125] P. Roberts. An algorithm for steady-state optimization and parameter estimation. *International Journal of Systems Science*, 10:719–734, 1979.
- [126] P. Roberts, T. Williams. On an algorithm for combined system optimization and parameter estimation. *Automatica*, 17(4):455–472, 1981.
- [127] J. Rossiter. *Model-Based Predictive Control*. CRC Press, Boca Raton - London - New York - Washington, D.C., 2003.

- [128] P. Scokaert. Infinite horizon generalized predictive control. *International Journal of Control*, 66(5):161–175, 1997.
- [129] P. Scokaert, D. Clarke. Stabilising properties of constrained predictive control. *IEE Proceedings – Control Theory Applications*, 141(5):295–304, 1994.
- [130] P. Scokaert, D. Mayne, J. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 1999.
- [131] M. Setnes, J. Sousa. Fuzzy rule-based optimization in nonlinear predictive control. *Proceedings 5th European Control Conference ECC'99*, paper F907(CD-ROM), Karlsruhe, 1999.
- [132] A. Stachurski, A. Wierzbicki. *Podstawy optymalizacji*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1999.
- [133] J. Stoer, R. Bulirsch. *Wstęp do analizy numerycznej*. Państwowe Wydawnictwo Naukowe, Warszawa, 1987.
- [134] W. Tadej. *Analiza własności dualnego algorytmu optymalizacji punktów pracy procesów nieliniowych*. Rozprawa doktorska. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Warszawa, 2001.
- [135] W. Tadej, P. Tatjewski. Analysis of an ISOPE-type dual algorithm for optimizing control and nonlinear optimization. *International Journal of Applied Mathematics and Computer Science*, 11(2):429–457, 2001.
- [136] T. Takagi, M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *Transactions on Systems, Man and Cybernetics*, 15(1), 1985.
- [137] K. Tanaka, T. Ikeda, H. Wang. Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs. *IEEE Transactions on Fuzzy Systems*, 6(2):250–265, 1998.
- [138] K. Tanaka, M. Sugeno. Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems*, 45:135–156, 1992.
- [139] P. Tatjewski. *Hierarchiczne metody rozwiązywania niwypukłych zadań optymalizacji i sterowania stanem ustalonym złożonych procesów*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1988.
- [140] P. Tatjewski. Two-phase dual-type optimising control algorithm for uncertain plants. *Proceedings 5th International Symposium Methods and Models in Automation and Robotics MMAR'98*, wolumen 1, strony 171–176, Międzydroje, 1998.
- [141] P. Tatjewski. Two-phase dual-type optimising control algorithm for uncertain plants with active output constraints. *Proceedings 5th European Control Conference ECC'99*, D-ROM FO347, Karlsruhe, 1999.
- [142] P. Tatjewski. Optimizing control of uncertain plants under varying external inputs. *Proceedings 6th European Control Conference ECC'01*, strony 2975–2980, Porto, 2001.
- [143] P. Tatjewski. *Sterowanie zaawansowane obiektów przemysłowych, Struktury i algorytmy*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2002.
- [144] P. Tatjewski. *Advanced Control of Industrial Processes*. Springer Verlag, London, 2007.

- [145] P. Tatjewski. Advanced control and on-line process optimization in multi-layer structures. *Annual Reviews in Control*, 32(1):71–85, 2008.
- [146] P. Tatjewski. *Metody numeryczne*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2013.
- [147] P. Tatjewski. Disturbance modeling and state estimation for offset-free predictive control with state-spaced process models. *International Journal of Applied Mathematics and Computer Science*, 24(2):313–323, 2014.
- [148] P. Tatjewski, N. Abdullah, P. Roberts. Comparative study and development of integrated optimization and parameter estimation algorithms for hierarchical steady-state control. *International Journal of Control*, 51(2):421–443, 1990.
- [149] P. Tatjewski, M. Brdyś, J. Duda. Optimising control of uncertain plants with feedback controlled output constraints. *Proceedings IFAC/IFORS/IMACS Symposium on Large Scale Systems Theory and Applications LSS'95*, strony 123–128, London, 1995.
- [150] P. Tatjewski, M. Brdyś, J. Duda. Optimising control of uncertain plants with constrained, feedback controlled outputs. *International Journal of Control*, 74(15):1510–1526, 2001.
- [151] P. Tatjewski, M. Ławryńczuk. Soft computing in model-based predictive control. *International Journal of Applied Mathematics and Computer Science*, 16(1):7–26, 2006.
- [152] P. Tatjewski, M. Ławryńczuk, P. Marusak. Wybrane algorytmy regulacji predykcyjnej procesów nieliniowych. *Materiały XIV Krajowej Konferencji Automatyki*, wolumen 1, strony 145–152, Zielona Góra, 2002.
- [153] P. Tatjewski, P. Roberts. Newton-like algorithm for integrated system optimization and parameter estimation technique. *International Journal of Control*, 46(4):1155–1170, 1987.
- [154] M. Teixeira, E. Assunção, H. Pietrobom. On relaxed LMI-based designs for fuzzy regulators and fuzzy observers. *Proc. 6th European Control Conference ECC'01*, strony 120–125, Porto, 2001.
- [155] M. Teixeira, S. Żak. Stabilizing controller design for uncertain nonlinear systems using fuzzy models. *IEEE Transactions on Fuzzy Systems*, 7(2):133–142, 1999.
- [156] L. Trybus. *Regulatory wielofunkcyjne*. WNT, Warszawa, 1992.
- [157] J. Vada, O. Slupphaug, T. Johansen, B. Foss. Linear MPC with optimal prioritized infeasibility handling: application, computational issues and stability. *Automatica*, 37:1835–1843, 2001.
- [158] P. van der Veen, R. Babuska, H. Verbruggen. Comparison of nonlinear predictive control methods for a waste-water treatment benchmark. *Proc. 5th European Control Conference ECC'99*, CD-ROM F1060-1, Karlsruhe, 1999.
- [159] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, Englewood Cliffs, N.J., 1993.
- [160] H. Wang, K. Tanaka, M. Griffin. An approach to fuzzy control of nonlinear systems: stability and design issues. *IEEE Transactions on Fuzzy Systems*, 4(1):14–23, 1996.

- [161] L. Wang, G. Feng. Piecewise  $H_\infty$  controller design of discrete time fuzzy systems. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 34(1):682–686, 2004.
- [162] J. Yen, R. Langari. *Fuzzy Logic - Intelligence, Control and Information*. Prentice Hall, Upper Saddle River, New Jersey, 1999.
- [163] L. Zadeh. Fuzzy sets. *Information and Control*, 8, 1965.
- [164] H. Zhang, P. Roberts. On-line steady-state optimization of nonlinear constrained processes with slow dynamics. *Transactions of the Institute of Measurement and Control*, 12(5):251–261, 1990.
- [165] J. Zhao, V. Wertz, R. Gorez. Dynamic fuzzy state-feedback controller and its limitations. *Proceedings 13th IFAC Triennial World Congress*, strony 121–126, San Francisco, 1996.

# Indeks

- adaptacja, 31
- algorytm
  - powtarzanej optymalizacji z korektą estymat zakłóceń, 319, 357
- algorytm regulacji
  - bezpośredniej, 6
  - CRHPC, 128, 282
  - DMC, 127
    - analityczny, 136
    - analityczny z kompensacją zakłóceń, 161
    - model zakłóceń, 132
    - nieskończony horyzont predykcji, 288
    - numeryczny, 151
    - prawo regulacji, 140, 141
    - równania predykcji, 131
    - trajektoria swobodna, 137
    - trajektoria wymuszana, 137
    - zadanie optymalizacji kwadratowej, 153
- FDMC
  - analityczny, 271
  - NPL, 261
  - NSL, 256
- FGPC
  - analityczny, 274
  - NPL, 257
  - NPLT, 257
  - NSL, 256
- FMPC, 253
  - analityczny, 271
  - NO, 254
  - NPL, 257
  - NSL, 255
- FMPCS
  - analityczny, 275
  - NPL, 263
  - NSL, 256
- GPC, 128, 163
  - analityczny, 169
  - model obiektu, 164
  - model obiektu, obiekt SISO, 165
  - nieskończony horyzont predykcji, 290
  - numeryczny, 187
  - prawo regulacji, 169, 170
  - prawo regulacji, struktura alternatywna, 172
  - trajektoria swobodna, 168, 184, 187
  - trajektoria wymuszana, 168
  - z modelem zakłóceń typu DMC, 183
  - zadanie optymalizacji kwadratowej, 188
- MAC, 127, 162
- MPC-NO, 227
  - trajektoria początkowa sterowań, 231
  - zadanie optymalizacji, 227, 229
- MPC-NPL, 238
  - trajektoria początkowa wyjść, 238
  - trajektoria przewidywana wyjść, 238
  - trajektoria wymuszana, 238
  - zadanie optymalizacji kwadratowej, 239
- MPC-NPL+, 240
- MPC-NPLT, 240
- MPC-NPLT+, 240, 241

- trajektoria początkowa wyjść, 242
  - zadanie optymalizacji kwadratowej, 242
- MPC-NSL, 237
- MPCS, 196
  - analityczny, 200
  - model obiektu, 196
  - numeryczny, 223
  - prawo regulacji, 200, 209, 213
  - trajektoria przewidywana wyjścia, 199
  - trajektoria swobodna, 200, 203, 208, 213, 214
  - trajektoria wymuszana, 199
  - z obserwatorem stanu, 207
  - z obserwatorem stanu rozszerzonego, 210
  - z pomiarem stanu, 196
  - zadanie optymalizacji kwadratowej, 223
- MPHC, 127
- predykcijnej, 116
  - dobór parametrów, 295
  - dual-mode, 291
  - nieliniowy, 226
  - nieliniowy z linearyzacją, 226
  - niestacjonarny, 125
  - z modelem neuronowym, 234
  - z quasi-nieskończonym horyzontem, 291
- QDMC, 127
- rozmyty TS (Takagi-Sugeno) *patrz też*: regulator, 58
- SMOC, 128
- algorytm sterowania
  - optymalizującego, 28, 318
- ANFIS, 51
- anti-windup, 115, 148
- baza wiedzy, 42, 44
- cel sterowania, 2, 10
  - dekompozycja, 4
- częstotliwość interwencji, 7, 10
- czas przebywania, 23
- DCS, 5
- dekompozycja, 3
  - funkcjonalna, 3
  - przestrzenna, 3
- diagnostyka, 33
- DPMO (metoda dualna powtarzanej modyfikowanej optymalizacji), 330
- dual-mode, 291
- filtr Kalmana, 215
  - rozszerzony (EKF), 232
  - stacjonarny, 216
- funkcja
  - kary, 295
  - kwadratowa, 123
  - Lapunowa, 64, 281
- funkcja celu (kryterialna)
  - ekonomiczna, 315
  - regulacji predykcijnej, 120, 121, 123
    - z nieskończonym horyzontem, 284
    - z trajektorią referencyjną, 300
  - zadania optymalizacji dynamicznej, 26
- funkcja przynależności, 38
  - gaussowska, 41
  - sigmoidalna, 40
  - trójkątna, 39
  - trapezowa, 38
  - uogólniona dzwonowa, 41
- horyzont
  - dynamiki obiektu, 133
    - oszacowanie, 133
  - predykcji, 117
    - dobór wartości, 296
    - nieskończony, 283
    - quasi-nieskończony, 291
  - prognozy zakłóceń mierzonych, 159
  - przesuwany, 116, 118
  - sterowania, 117
    - dobór wartości, 296
- IDCOM, 127

- ISOPE, 320
- kolumna destylacyjna, 55
  - ciecz wyczerpana, 55
  - orosienie, 55
- kompensacja równoległa rozproszona, 59
- linearyzacja, 236
  - wokół trajektorii, 240
- LMI, 65
- logika rozmyta, 37
- LSSO (local steady state optimization), 306
- macierz
  - dynamiczna
    - dla obiektu MIMO, 139
    - dla obiektu SISO, 138
    - nieliniowa, 241, 260
  - wag przyrostów sterowania, 121
    - dobór wartości, 299
  - wag uchybów regulacji, 121
    - dobór wartości, 299
  - wskaźnik uwarunkowania , 332
- metoda
  - dualna powtarzanej modyfikowanej optymalizacji (DPMO), 330
    - zoptymalizowana faza wstępna, 336
  - algorytm, 332
  - ograniczenie uwarunkujące, 332
  - powtarzanej modyfikowanej optymalizacji (PMO), 320
    - algorytm dualny, 329
    - algorytm podstawowy, 325
    - estymacja pochodnych, 328
    - struktura, 323
      - z wyjściami ograniczonymi regulacyjnie, 359
  - powtarzanej optymalizacji z korektą estymat zakłóceń
    - z wyjściami ograniczonymi regulacyjnie, 358
- MIMO obiekt
  - odpowiedzi skokowe, 134
- model obiektu
  - ARX, 183
    - wielowymiarowy, 186
  - dynamiczny stanowy, 13
  - liniowy o zmiennych współczynnikach, 255
  - nieliniowy, 236
  - regulowanego
    - dynamiczny, 13
    - statyczny, 313
  - rozmyty TS ciągły
    - z równaniami stanu, 91
  - rozmyty TS dyskretny, 254
    - dla nieliniowej predykcji, 257
    - z następnikami ARX, 77
    - z równaniami stanu, 62
- modelowanie
  - obektu, 14
    - warstwowe, 16
- MPC (model predictive control), 116
- nadzór, 33
- obiekt sterowania, 5
  - niestabilny, 182
- obserwator Luenbergera
  - bieżący, 214
    - predykcyjny, 207, 214
    - rozszerzony (ELO), 231
- obserwator stanu, 207
  - rozmyty TS, 256
- odpowiedź impulsowa, 162
- odpowiedź skokowa, 130
  - modelu ARX, 183
    - wielowymiarowa, 186
  - przyrostowa, 133
    - wielowymiarowa (macierzowa), 135
- odwzorowanie obiektu statyczne, 316
- odwzorowanie statyki obiektu
  - zdekomponowane, 355
- ograniczenia, 26, 124
  - stanu końcowego, 280
- sterowania
  - przyrostu sygnału, 124

- twarde, 146
  - wartości sygnału, 124
- wartości zadanych regulatorów, 26, 316
- wyjść
  - miękkie, 146, 294
  - nieregulowanych, 125
  - priorytetyzacja, 295
  - regulacyjnie kontrolowane, 28
  - regulowanych dwustronne, 124
  - regulowanych jednostronne, 125
  - relaksacja, 295
  - w stanach ustalonych, 26
- ograniczenie uwarunkowujące, 332
- okno ograniczeń, 125, 293
- okres próbkowania, 117
  - dobór wartości, 299
- optymalizacja
  - dynamiczna, 27
  - zadanie, 27
  - nieliniowa
    - metoda podziału i ograniczeń, 235
    - metoda SQP, 254
  - statyczna
    - zadanie, 27
- optymalności zasada, 283
- parametry algorytmu MPC
  - dobór (strojenie), 295
- perceptron wielowarstwowy, 51
- PMO (metoda powtarzanej modyfikowanej optymalizacji), 320, 323
- pochodna kierunkowa, 330
- prawo regulacji
  - DMC, 140
  - GPC, 169
  - MPCS, 200
- priorytetyzacja ograniczeń, 295
- procesy dynamiczne
  - szybkie, 10
    - (wykonawcze), 13
  - wolne, 10
    - (optymalizowane), 13
- punkt pracy
  - optymalny, 8
- reaktor polimeryzacji, 247
- reguła wnioskowania rozmyta, 42
  - konkluzja, 46
    - operatorem minimum, 47
    - operatorem mnożenia, 47
- następnik, 42
  - funkcyjny, 43
  - funkcyjny liniowy, 43
  - ostry, 42
  - rozmyty, 43
- poprzednik, 42
- poziom aktywacji, 46, 47
  - unormowany, 48
- reguły wnioskowania
  - kompensatora TS dyskretnego, 113
  - modelu obiektu TS ciągłego
    - z równaniami stanu, 91
  - modelu obiektu TS dyskretnego
    - z następnikami ARX, 77, 79, 80
    - z równaniami stanu, 62
  - regulatora TS ciągłego
    - PID, 102
    - ze sprzężeniem od stanu, 94
  - regulatora TS dyskretnego
    - niespójność logiczna, 78
    - ze sprzężeniem od stanu, 70
    - ze sprzężeniem od wyjścia, 78, 80
- regulacja
  - adaptacyjna, 36
  - kaskadowa, 24
  - nadrzędna, 11
  - nieliniowa, 36
  - podstawowa, 7
  - predykcyjna, 115
    - zasada, 117
  - zaawansowana, 7, 116
- regulator
  - rozmyty TS ciągły
    - PID, 103, 106
    - ze sprzężeniem od stanu, 94
  - rozmyty TS dyskretny

- FDMC, 271
  - FGPC, 274
  - FMPCS, 275
  - ze sprzężeniem od stanu, 70
  - ze sprzężeniem od wyjścia, 77
- wielofunkcyjny, 6
- rozmywanie, 41
- rzutowanie sterowania, 148
  
- SCADA, 5
- sieć neuronowa rozmyta, 51
  - algorytm optymalizacji, 54
  - hybrydowy, 54
- SSTO (steady-state target optimization), 307
- stability conditions
  - of discrete TS fuzzy
  - state-feedback control, 73
- stabilność
  - algorytmu DMC z nieskończonym horyzontem predykcji, 288
  - algorytmu MPC, 279
    - liniowego z nieskończonym horyzontem, 285
    - podstawowe mechanizmy zapewniania, 280, 291
  - regulatora ciągłego rozmytego TS PID, 105
- warunki
  - modelu rozmytego TS ciągłego autonomicznego, 92, 94
  - modelu rozmytego TS dyskretnego autonomicznego, 63
  - regulacji ciągłej TS ze sprzężeniem od stanu, 95, 96
  - regulacji dyskretnej TS ze sprzężeniem od stanu, 72
  - regulacji dyskretnej TS ze sprzężeniem od wyjścia, 83, 84
  - regulacji FDMC, 273
  - regulacji FGPC, 274
- stanu równania, 196
- steady-state target optimization (SSTO), 307
  - sformułowanie celu, 310
  - zadanie, 310, 311
    - z relaksacją ograniczeń, 311
- sterowanie, 1
  - dualne, 334
  - optymalne, 27
  - repetycyjne
    - zasada, 118
    - stanem ustalonym, 27
- stopień przynależności, 39
- struktura
  - metody PMO, 323
- struktura regulacji
  - DMC w wersji analitycznej, 141, 142
    - z kompensacją zakłóceń, 161
    - z ograniczeniami, 148, 149
  - FDMC w wersji analitycznej, 271
  - GPC w wersji analitycznej, 170
    - alternatywna, 172
  - MPC z trajektorią referencyjną, 302
  - MPC-NO, 227
  - MPC-NPL, 239
  - MPC-NPLT+, 243
  - MPC-NSL, 237
  - MPCS w wersji analitycznej
    - z obserwatorem stanu, 209
    - z obserwatorem stanu rozszerzonego, 213
    - z pomiarem stanu, 200
    - o dwóch stopniach swobody, 301
- struktura sterowania
  - warstwowa, 3, 5, 305
    - przyczyna stosowania, 10
    - z MPC z SSTO, 308
  - wielopoziomowa, 4
    - z kompensacją zakłóceń, 111, 157
- superpozycji zasada, 131
- system rozmyty, 42
  - Takagi-Sugeno (TS), 43
  - projektowanie, 50

- system sterowania, 2
  - rozproszony, 5
  - wykonawczy, 13
- tożsamość Bezout, 165
- trajektoria początkowa sterowań, 231
- trajektoria przewidywana
  - wartości zadanych, 119
  - wyjścia, 118
    - dekompozycja, 122
    - nieliniowa, 227
    - składowa swobodna (trajektoria swobodna), 118, 122, 133, 137, 158
    - składowa wymuszana (trajektoria wymuszana), 122, 133, 137, 199
- trajektoria przewidywana wyjść
  - algorytmu MPC-NO, 238
- trajektoria referencyjna, 121, 163, 300
- twierdzenie Lapunowa, 63, 281
- uchyb ustalony
  - zerowy algorytmu DMC, 155
  - zerowy algorytmu GPC, 182
- warstwa
  - optymalizacji, 8, 26, 305
  - planowania produkcji, 9
  - regulacji
    - bezpośredniej, 5
    - nadrzędnej, 7
  - sterowania, 5
- wartość szczególna macierzy, 332
- wartości zadane
  - zmiennych regulowanych, 122
  - optymalne, 8, 305
- wyjścia
  - niesterowane (swobodne), 2
  - sterowane, 2
- wnioskowanie rozmyte, 46
- wskaźnik uwarunkowania macierzy, 332
- wyjścia, 2
  - regulowane, 11, 354
  - swobodne (nieregulowane), 12, 354
- wyostrzanie, 46
- zadanie optymalizacji
  - algorytmu predykcyjnego liniowego, 126
  - MPC-NO (nieliniowego), 227, 229
  - kwadratowej, 153, 294
    - algorytmu DMC, 153
    - algorytmu MPC-NPL, 239
    - algorytmu MPC-NPLT+, 242
    - z relaksacją ograniczeń, 294
  - LSSO, 306
    - funkcja celu, 307
- zadanie optymalizacji modelowej (ZOM), 317, 321
- zadanie optymalizacji modelowej z wyjściami (ZOMY), 356
- zadanie sterowania optymalizującego (ZSO), 316, 321
- zadanie sterowania optymalizującego z wyjściami (ZSOY), 356, 359
- zakłócenia, 2, 120
  - mierzone
    - kompensacja, 110, 157
    - model prognostyczny, 159
    - model typu DMC, 132
  - niemierzalne, 155
- zbiór
  - dopuszczalny algorytmu MPC
    - pusty, 126
    - zapewnianie niepustości, 293
  - ostry, 38
  - reguł, 42
  - rozmyty, 37, 38
    - zasady doboru, 50
- zmienna lingwistyczna, 39
- zmiennie regulowane, 12
  - warunki wyboru, 12
- zmodyfikowanie zadanie optymalizacji modelowej (ZZOM), 321

- zmodyfikowane zadanie optymalizacji modelowej uwarunkowane (ZZOMU), 333
- ZOM (zadanie optymalizacji modelowej), 317, 321
- ZOMY (zadanie optymalizacji modelowej z wyjściami), 356
- ZSO (zadanie sterowania optymalizującego), 316
- ZSOY (zadanie sterowania optymalizującego z wyjściami), 356
- ZZOM (zmodyfikowane zadanie optymalizacji modelowej), 321
- ZZOMU (zmodyfikowane zadanie optymalizacji modelowej uwarunkowane), 333



# Abstract

## **Advanced Control of Industrial Processes Structures and Algorithms, Second Edition**

The book is devoted to advanced algorithms of industrial process control, in the framework of a multilayer control structure.

First chapter introduces the multilayer control structure, starting from basic definitions and a functional decomposition of the overall control task. Basic layers are described: the direct (local loop) control layer, the supervisory feedback control layer, the set-point optimization layer and the management layer (the latter being beyond the scope of the book). Process modelling within the multilayer structure is discussed, stressing the fact that different models and time scales are applied at different layers. A detailed example of a CSTR reactor with two nonlinear reactions illustrates theoretical considerations. Finally, aspects of adaptation and emergency handling in the multilayer hierarchy are commented. It is pointed out, that developments in hardware and software technology in the last decades led to wide applications of advanced regulatory control, often combined with on-line set-point optimization.

Second chapter is devoted to nonlinear fuzzy control based on Takagi-Sugeno (TS) models and controller structures. After a short introduction to fuzzy logic and fuzzy modelling, design procedures and stability analysis tools of nonlinear fuzzy TS controllers are presented. State-space feedback and dynamic output feedback controllers are considered, for both cases of discrete as well as continuous time design. In the author's opinion, the described techniques are effective, relatively easy and convincing, because they can be treated as a natural nonlinear extension of well known linear control structures, like e.g. widely accepted linear PID controllers.

Model Predictive Control (MPC) is the subject of the third chapter. Special attention is paid to this case, because MPC is the advanced control technique that has made most significant and widespread impact on industrial process control in the last decades, handling efficiently multivariable and constrained cases, finding numerous successful applications. In

this chapter linear predictive control techniques are first presented, first the popular DMC (*Dynamic Matrix Control*) algorithm, then the GPC (*Generalized Predictive Control*) algorithm, and finally the approach based on state-space modelling of the control plant. Both analytical formulation (obtained from unconstrained design) leading to feedback control laws, as well as numerical formulation when a numerical constrained quadratic programming procedure is applied at each sample to find the control vector, are considered. Presentation of these, rather standard techniques contains certain original results like anti-windup structures for predictive control law implementations, derivation of an alternative view on the structure of the GPC control law, direct and simple derivation of the GPC algorithm for the case of the DMC type disturbance representation and novel approach to predictive control with state-space process models based on state disturbance model, for both state measurement and state estimation cases. Basing on linear designs, several structures of nonlinear predictive controllers, differing in increasing computational complexity, are then introduced. Special attention is focused on linearization approaches, in the author's opinion most important for practical implementations. It is shown that these approaches are especially easy applicable when the nonlinear plant model is of the fuzzy TS type. The subject of stability is then tackled, indicating key sources of nominal stability of both linear and nonlinear MPC controllers and their impact on controller structure and design parameters.

Chapter four is devoted to algorithms for the on-line set-point optimization. First a discussion of the role and place of the optimization in the multilayer control structure is presented. Then the problems of steady-state target optimization, closely combined with the model predictive control, are discussed. In the last part of the chapter, the problem of steady-state set-point optimizing control is formulated and on-line iterative set-point optimization algorithms under uncertainty in process models and disturbance estimates are presented. Formulations of the algorithms and their analyses are given both for the basic multilayer control structure and for the structure with feedback controlled output constraints.

The book is addressed both to a wide group of readers interested in learning basics and more involved details of the presented advanced control algorithms, as well as to the students of more advanced control courses, PhD. students and researchers. Therefore, main attention of the author is focused on presentation of structures and properties of the algorithms, illustrating the design on many simple and more practical simulation examples. The amount of theoretical analysis is therefore limited, but numerous references are given for the reader interested in deeper or wider knowledge.