

Mgr inż. Maciej Ławryńczuk  
Mgr inż. Piotr Marusak  
Prof. nzw. dr hab. Piotr Tatjewski  
Instytut Automatyki i Informatyki Stosowanej  
Politechniki Warszawskiej

## **REGZA – PAKIET PROGRAMÓW DO PROJEKTOWANIA ZAAWANSOWANYCH UKŁADÓW REGULACJI<sup>1</sup>**

*Opisano pakiet oprogramowania ułatwiający wykonanie poszczególnych etapów procesu projektowania układów regulacji automatycznej, ze szczególnym uwzględnieniem syntezy algorytmów regulacji predykcyjnej. Programy wchodzące w skład pakietu umożliwiają analizę sygnałów pozyskanych z procesu, identyfikację modeli i dobór, na podstawie uzyskanego modelu, algorytmów regulacji, których działanie można następnie przetestować. Pakiet został zaprojektowany jako zbiór efektywnych narzędzi pozwalających na wykorzystanie złożonych algorytmów identyfikacji i regulacji, z których część jest oryginalnymi rozwiązaniami powstałymi w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej.*

### **REGZA – SOFTWARE FOR COMPUTER AIDED DESIGN OF ADVANCED CONTROL SYSTEMS**

*A software system for computer aided control systems design is described in the paper. The system enables process data analysis, model identification and control algorithm synthesis and testing. The software offers many advanced identification and control algorithms easily accessed using intuitive program interface. Many of the implemented algorithms have been developed at the Institute of Control and Computation Engineering, Warsaw University of Technology.*

#### **1. WPROWADZENIE**

Celem niniejszego referatu jest przedstawienie pakietu REGZA (REGulacja ZAawan-sowana), opracowanego w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej [8], rozszerzone w stosunku do [5] o opis nowych elementów, dodanych ostatnio do programów wchodzących w skład pakietu.

Programy wchodzące w skład pakietu umożliwiają wykonanie poszczególnych etapów związanych z opracowaniem układu regulacji, tzn. analizę sygnałów pozyskanych z obiektu, identyfikację modelu, a na końcu procesu projektowania – synteze i testowanie algorytmu regulacji. Oprogramowanie umożliwia zaprojektowanie kilku wersji algorytmów regulacji predykcyjnej, zarówno dla procesów liniowych jak i nieliniowych.

---

<sup>1</sup> Praca finansowana w ramach projektu CATID i działalności statutowej WEiTI PW

Algorytmy tego typu są przedmiotem szczególnego zainteresowania projektantów systemu, ze względu na ich liczne zalety umożliwiające efektywną regulację procesów wielowymiarowych, w obecności ograniczeń [1, 11, 16].

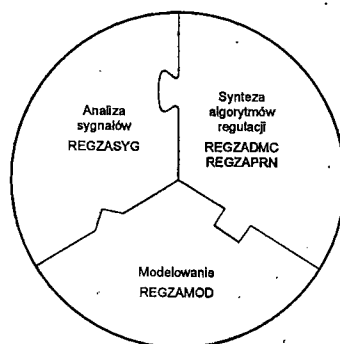
Zamiarem autorów pakietu było opracowanie wygodnych narzędzi pozwalających na wykorzystanie złożonych algorytmów obliczeniowych nie tylko przez środowiska akademickie, lecz również przez praktyków. Dlatego też starano się, aby obsługa opisywanych programów była intuicyjna i nie wymagała szczegółowej wiedzy z zakresu teorii identyfikacji i sterowania a także, aby było możliwe wykonanie wielu złożonych eksperymentów w stosunkowo krótkim czasie. Warto jednak nadmienić, że opracowany pakiet, umożliwiając testowanie nowych wersji algorytmów oraz ich badania porównawcze, jest szczególnie przydatny do celów badawczych.

Warto zaznaczyć, że niektóre algorytmy identyfikacji oraz regulacji zaimplementowane w pakiecie REGZA są oryginalnymi rozwiązaniami powstałymi w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej [6, 9, 10, 12, 13, 14, 17].

## 2. STRUKTURA PAKIETU

Pakiet REGZA składa się z czterech programów (rys. 1), z których każdy może pracować niezależnie. Jest jednak możliwa wymiana danych pomiędzy tymi programami, za pomocą plików o określonym formacie [8]. Zadania poszczególnych elementów pakietu REGZA są następujące:

- a) analiza i obróbka danych – program REGZASYG,
- b) budowa i identyfikacja modeli – program REGZAMOD,
- c) synteza algorytmów regulacji – programy REGZADMC (regulacja DMC) oraz REGZAPRN (predykcyjne regulatory neuronowe).

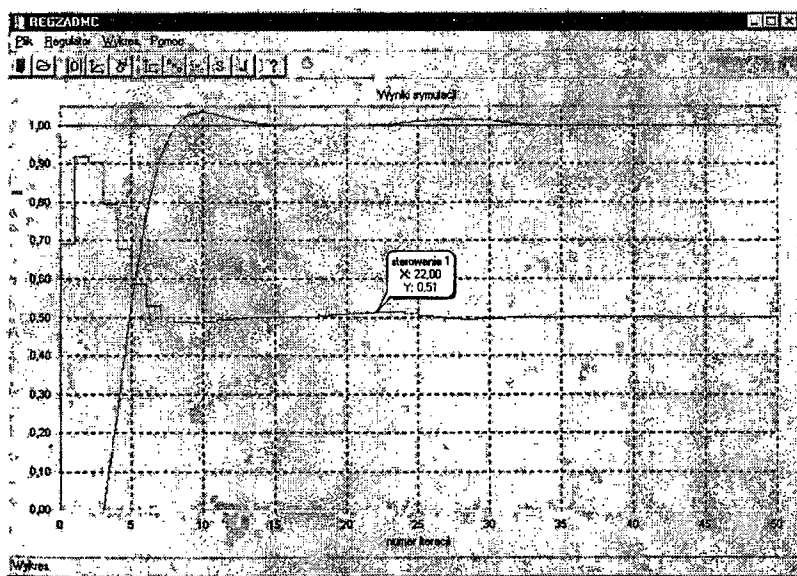


Rys. 1. Elementy i struktura pakietu REGZA

Całość procesu projektowania regulatora można wykonać używając poszczególnych składników pakietu, a tok postępowania zmierzającego do tego celu, może być następujący (zob. przykład z [5]): po wstępnym przygotowaniu danych (program REGZASYG) są one wykorzystywane do identyfikacji modeli (program REGZAMOD). Otrzymane pliki, zawierające strukturę i parametry modeli służą następnie do syntezy algorytmów regulacji (programy REGZADMC i REGZAPRN).

W celu zapewnienia możliwie dużej przenośności, wszystkie procedury obliczeniowe zostały napisane w języku C. System komunikacji z użytkownikiem został natomiast opracowany przy wykorzystaniu środowiska programistycznego C++ Builder firmy Borland, przy czym wykorzystano elementy interfejsu graficznego standardowo oferowane przez system operacyjny Windows.

Główne okno każdego z programów składa się z kilku części (przykład – rys. 2). Połączając od góry okna, pod paskiem tytułowym znajduje się menu rozwijalne, z którego wybiera się żądane funkcje programu. Poniżej, częściej używane opcje programu są wyprowadzone w postaci przycisków na pasku narzędzi. Centralną część okna zajmuje wykres z przebiegami wybranych zmiennych. Na samym dole okna znajduje się pasek stanu, na którym są wyświetlane podpowiedzi dla użytkownika. Taka postać interfejsu, standardowo stosowana w większości programów przeznaczonych dla systemu Windows, umożliwia intuicyjną obsługę oprogramowania oraz pozwala na szybkie wywoływanie poszczególnych procedur obliczeniowych, a także łatwe wprowadzanie i modyfikację parametrów.



Rys. 2. Główne okno programu REGZADMC

### 3. ANALIZA SYGNAŁÓW

Program służący do analizy i obróbki sygnałów (REGZASYG) umożliwia przeprowadzenie szeregu operacji na zgromadzonych danych oraz wizualizację tych danych. W programie wykorzystano bibliotekę funkcji służących do obróbki sygnałów opracowaną przez Pawła Wnuka na potrzeby pakietu SYGNAŁ [4], odpowiednio ją adaptując. Ponadto do zestawu podstawowych operacji możliwych do przeprowadzenia na sygnałach poddawanych obróbce zostały dodane operacje zaawansowane.

Dane są przechowywane w plikach, wraz z informacjami takimi, jak: krok dyskretyzacji, liczba próbek, nazwa zmiennej (ze względu na kompatybilność przyjęto format

zgodny z pakietem SYGNAŁ). Program umożliwia, w zależności od potrzeb, dzielenie zbiorów z danymi na części, jak również łączenie zbiorów tej samej zmiennej. Jest także możliwy zapis wybranych danych do osobnego pliku, w formacie akceptowanym przez program do identyfikacji modeli procesów.

Operacje podstawowe oferowane przez program do analizy danych (zaadaptowane z pakietu SYGNAŁ) są następujące: usuwanie wartości spoza ograniczeń maksymalnego i minimalnego, obliczanie wartości średniej, maksymalnej oraz minimalnej w wybranym przedziale, obliczanie wariancji w przedziale, filtracja pierwszego rzędu, normalizacja sygnału, wyznaczanie trendu wartości średniej, trendu liniowego oraz możliwość ich usunięcia, agregacja równomierna i możliwość edycji poszczególnych próbek.

Operacje zaawansowane oferowane przez program REGZASYG są następujące:

- wykrywanie okresów stanów ustalonych,
- wykrywanie nagłych zmian,
- wykrywanie okresów dużej zmienności (poprzez analizę zmian wartości średniej w oknie oraz analizę wariancji),
- wykrywanie skoków wartości średniej (zmodyfikowany algorytm Page'a-Hinkley'a).

#### 4. MODELOWANIE

Program REGZAMOD służy do szeroko pojętej identyfikacji dyskretnych modeli procesów dynamicznych oraz do ich testowania. Procesy wielowymiarowe mogą być modelowane przy wykorzystaniu:

- modeli liniowych z opóźnieniem (ARX),
- modeli złożonych z rzędnych odpowiedzi skokowych (procesy liniowe),
- modeli neuronowych (dwuwarstwowa sieć perceptronowa),
- modeli liniowo-neuronowych (połączenie modeli ARX i sieci neuronowych),
- modeli obszarowych (rozmytych) typu Takagi-Sugeno,
- obszarowych modeli neuronowych (połączenie sieci perceptronowych z koncepcją modeli obszarowych) [9].

Ocena jakości predykcji możliwa jest w sposób ilościowy (wskaźniki jakości), a także w sposób jakościowy – dzięki rozbudowanym opcjom wizualizacji. Z uwagi na przeznaczenie modeli można wyznaczyć predykcję jednokrokową lub też sprawdzić działanie modelu rekurencyjnego.

Z uwagi na prowadzone prace badawcze szczególne znaczenie mają modele neuronowe. Oprócz typowych modeli bazujących na sieci dwuwarstwowej typu perceptronowego dostępne są modele liniowo-neuronowe oraz tzw. obszarowe modele neuronowe [9]. Zaimplementowano kilka efektywnych algorytmów optymalizacji parametrów modeli neuronowych (algorytmy zmiennej metryki oraz gradientów sprzężonych). Z uwagi na różne przeznaczenie modeli (np. dla potrzeb algorytmów predykcyjnych) istnieje możliwość identyfikacji predyktora jednokrokowego (NARX) lub wielokrokowego (NOE). Rekurencyjny tryb uczenia modeli neuronowych jest szczególnie polecany wówczas, gdy modele te mają być później wykorzystane w algorytmie regulacji predykcyjnej. Program pozwala wyznaczyć odpowiedź skokową modelu neuronowego dla wybranego

punktu pracy oraz jego charakterystykę statyczną. Aby umożliwić osiągnięcie dobrych właściwości predykcji istnieje możliwość douczania modeli neuronowych.

## 5. SYNTEZA ALGORYTMÓW REGULACJI

Programy REGZADMC oraz REGZAPRN umożliwiają projektowanie i symulację liniowych i nieliniowych cyfrowych algorytmów regulacji wielowymiarowych obiektów dynamicznych. Ocena jakości algorytmu regulacji oraz możliwość stosunkowo łatwej poprawy jego właściwości wspomagają oferowane przez programy rozbudowane opcje wizualizacji, wyznaczania wskaźników jakości regulacji oraz korekcji parametrów. Istnieje także możliwość sprawdzenia wpływu zakłóceń na działanie projektowanych układów regulacji.

W zależności od klasy nieliniowości procesu oraz możliwości implementacyjnych można zastosować następujące dyskretne struktury sterowania:

- algorytm PID oraz nieliniowy rozmyty algorytm PID typu Takagi–Sugeno,
- algorytm regulacji predykcyjnej DMC (Dynamic Matrix Control) dla obiektów liniowych [1, 3, 11],
- rozmyty algorytm regulacji predykcyjnej FDMC (Fuzzy DMC) z sukcesywną linearyzacją, wykorzystujący nieliniowe modele rozmyte typu Takagi–Sugeno [12, 14, 16],
- nieliniowy algorytm regulacji predykcyjnej NDMC (Nonlinear DMC) z predykcją odpowiedzi swobodnej dokonywaną przy użyciu nieliniowego modelu typu Takagi–Sugeno i sukcesywną linearyzacją [12, 14, 16],
- nieliniowy algorytm regulacji predykcyjnej NDMC z predykcją odpowiedzi obiektu na sterowania wyznaczone w poprzednim kroku przy użyciu modelu nieliniowego typu Takagi–Sugeno i sukcesywną linearyzacją [12, 16],
- nieliniowy algorytm regulacji predykcyjnej NDMC z predykcją odpowiedzi obiektu na sterowania wyznaczone w poprzednim kroku przy użyciu modelu nieliniowego typu Takagi–Sugeno oraz z wielokrotną linearyzacją i optymalizacją w każdym kroku pracy regulatora w celu poprawienia prognozy i wyznaczanych sterowań [12, 16],
- algorytm regulacji predykcyjnej GPC (Generalized Predictive Control) dla obiektów liniowych [2, 11],
- nieliniowy algorytm regulacji predykcyjnej z sukcesywną linearyzacją w punkcie pracy wykorzystujący modele neuronowe [16],
- nieliniowy algorytm regulacji predykcyjnej z nieliniową predykcją i linearyzacją w punkcie pracy wykorzystujący modele neuronowe [6],
- nieliniowy algorytm regulacji predykcyjnej z iteracyjnie modyfikowaną nieliniową predykcją i linearyzacją w punkcie pracy wykorzystujący modele neuronowe [16],
- nieliniowy algorytm regulacji predykcyjnej z nieliniową optymalizacją wykorzystujący modele neuronowe [10].

Oprócz algorytmów predykcyjnych, z uwagi na liczne zastosowania praktyczne, oprogramowanie umożliwia projektowanie jednowymiarowych algorytmów PID oraz roz-

mytych (typu Takagi–Sugeno) PID. Nastawy dobierane są przez użytkownika lub automatycznie, przy wykorzystaniu optymalizacji wybranego kryterium.

Algorytmy predykcyjne zostały zaimplementowane w wersji ogólnej umożliwiającej ich zastosowanie do regulacji procesów wielowymiarowych. Użytkownik ma możliwość zadawania standardowych parametrów algorytmów, jakimi są: horyzont predykcji, horyzont sterowania oraz wartość współczynnika kary za przyrosty sterowań.

Algorytmy predykcyjne DMC, GPC oraz algorytm FDMC są dostępne w dwóch wersjach: analitycznej (w postaci analitycznego prawa regulacji) oraz numerycznej polegającej na powtarzonym rozwiązywaniu problemu optymalizacji kwadratowej z ograniczeniami. W przypadku algorytmów w wersji analitycznej istnieje możliwość uwzględnienia ograniczeń nałożonych na sterowania, dzięki zastosowaniu mechanizmu rzutowania sterowań na zbiór ograniczeń. Pozostałe algorytmy nieliniowe dostępne są w wersji numerycznej, co umożliwia uwzględnienie ograniczeń zarówno wartości i przyrostów sygnałów sterujących, jak i wartości sygnałów wyjściowych.

Sterowania generowane w algorytmie DMC są wyznaczone na podstawie modelu obiektu w postaci odpowiedzi skokowych. Dla modeli innych niż odpowiedź skokowa, najpierw jest ona wyznaczana w bieżącym punkcie pracy. Algorytm GPC wykorzystuje natomiast modele wielomianowe.

W przypadku regulatorów FDMC zakłada się, że obszary regulatora FDMC są takie same, jak w modelu obiektu i dla każdego z obszarów modelu obiektu wyznacza się odpowiedź skokową. W przypadku regulatora w wersji analitycznej, dla każdej z użytych odpowiedzi skokowych oblicza się współczynniki lokalnego regulatora liniowego. Regulator FDMC w wersji analitycznej jest złożony z uprzednio wyznaczonych regulatorów lokalnych. Współczynniki równań tego regulatora są więc nieliniowo zmienne i ustalane w każdej iteracji, w zależności od aktualnego punktu pracy, przy użyciu logiki rozmytej. W algorytmie numerycznym FDMC, w każdym kroku są w sposób rozmyty ustalane współczynniki odpowiedzi skokowej, wykorzystywane następnie podczas formułowania zadania optymalizacji.

W najprostszej wersji algorytmu NDMC (Nonlinear DMC – nieliniowy, z rozmytym modelem obiektu) predykcja odpowiedzi swobodnej jest wykonywana z użyciem modelu nieliniowego i wyznaczana jest odpowiedź skokowa przy aktualnym stanie zmiennych. Następnie w każdej iteracji rozwiązywane jest zadanie optymalizacji kwadratowej z ograniczeniami. Algorytm ten jest także dostępny w bardziej zaawansowanych wersjach, w których na podstawie modelu nieliniowego wyznacza się predykcję odpowiedzi obiektu na ciąg sterowań wygenerowanych przez regulator w poprzednim kroku. Model liniowy jest natomiast wykorzystywany jedynie podczas obliczania poprawek względem przyjętego ciągu przyszłych sterowań. W najbardziej skomplikowanym algorytmie NDMC, jest możliwe iteracyjne poprawianie prognozy, a tym samym, także jakości sterowania. Każda z wersji algorytmów regulacji DMC, FDMC oraz NDMC umożliwia uwzględnienie prognozy zakłóceń mierzalnych.

Oprogramowanie umożliwia syntezę czterech wersji nieliniowych algorytmów predykcyjnych z modelami neuronowymi. Istnieje możliwość uwzględnienia ograniczeń wartości oraz szybkości narastania sygnałów sterujących oraz ograniczeń wartości sygnałów wyjściowych. Dostępne są efektywne obliczeniowo trzy algorytmy suboptymalne z

linearyzacją oraz algorytm z nieliniową optymalizacją. W przypadku najprostszego algorytmu z sukcesywną linearyzacją, w każdej iteracji, na podstawie modelu nieliniowego, wyznaczany jest lokalny model liniowy, do obliczenia sygnałów sterujących wykorzystywany jest następnie klasyczny algorytm liniowy. Jakość regulacji można zwiększyć wykorzystując model neuronowy do wyznaczenia nieliniowej trajektorii swobodnej (algorytm z nieliniową predykcją i linearyzacją). Dalsza poprawa jakości regulacji może być osiągnięta dzięki modyfikacji tej trajektorii w iteracyjnej wersji algorytmu. Zaletą wszystkich algorytmów z linearyzacją jest niewielka złożoność obliczeniowa: problem optymalizacji, analogicznie jak w przypadku liniowym jest wypukły, sekwencja sygnałów sterujących wyznaczana jest w wyniku rozwiązania zadania liniowo-kwadratowego, nie istnieje więc problem fałszywych minimów lokalnych.

W przypadku najbardziej złożonego obliczeniowo algorytmu predykcyjnego nieliniowy model procesu jest wykorzystywany zarówno do celów predykcyjnych jak i optymalizacji. W każdej iteracji algorytmu wartości sygnałów sterujących wyznaczane są w wyniku rozwiązania nieliniowego problemu optymalizacji z ograniczeniami, w ogólności niewypukłego. W rezultacie algorytm ten umożliwia regulację wielu złożonych procesów nieliniowych (nie są konieczne żadne założenia upraszczające). Oprócz standardowego wskaźnika jakości (suma kwadratów błędów predykcyjnych) można zastosować normę wartości bezwzględnej lub też normę maksimum [7].

Uwzględnienie zbyt restrykcyjnych ograniczeń sygnałów wyjściowych w algorytmie sterowania może spowodować, że zbiór rozwiązań będzie pusty. Dlatego też, w oferowanych przez programy algorytmach regulacji, zaimplementowano mechanizm umożliwiający relaksację ograniczeń, poprzez traktowanie ich jako „ograniczeń miękkich” i zastosowanie odpowiednich funkcji kary [11, 16].

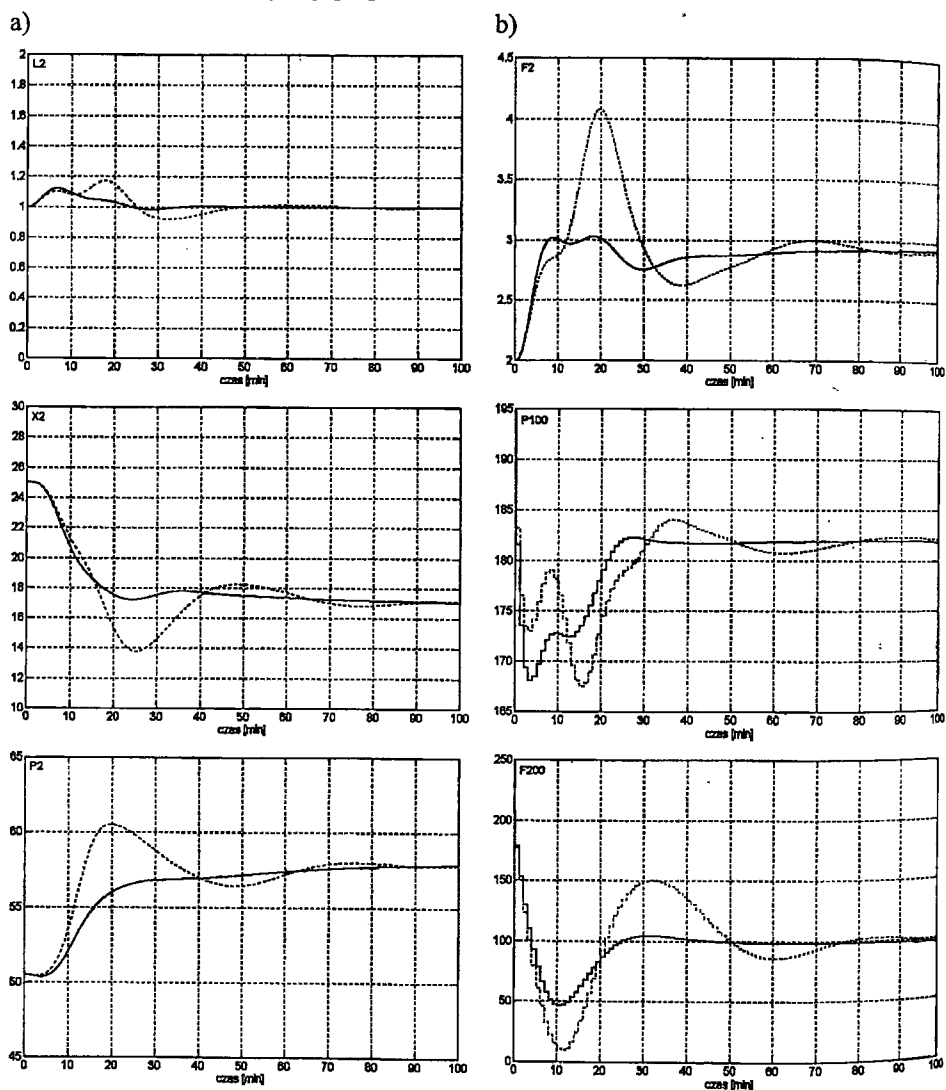
## 6. PRZYKŁAD WYKORZYSTANIA PAKIETU

W celu zademonstrowania wybranych możliwości pakietu przedstawiony zostanie przykład układu regulacji obiektu wielowymiarowego – wyparki opisanej w pracy [15]. Przyjęto przy tym, że jedna z wielkości regulowanych jest stabilizowana regulatorem PI tak, jak zaproponowano w [15]. Ponieważ rozważany obiekt regulacji był przedmiotem badań prowadzonych przez wielu autorów, zob. np. [11], a dokładny opis modelu matematycznego podano w pracy [15], w kolejnym akapicie wyszczególniono jedynie wielkości wejściowe i wyjściowe obiektu.

Wielkości regulowane obiektu to:  $L2$  – poziom cieczy w separatorze (przyjęto, że jest on stabilizowany wokół wartości  $L2_{zad} = 1,0$  m za pomocą regulatora PI o nastawach  $K_p = 5,6$  oraz  $T_i = 8,84$  min [15]),  $X2$  – skład produktu,  $P2$  – ciśnienie w wyparce. Wielkości sterujące są następujące:  $F2$  – przepływ produktu (jest to wielkość używana do stabilizacji poziomu  $L2$ ),  $P100$  – ciśnienie pary,  $F200$  – przepływ wody chłodzącej.

Do rozważanego obiektu dobrano dwa regulatory używając do tego celu programu REGZADMC z wprowadzonymi odpowiednimi zmianami, potrzebnymi do wykonania zaplanowanych eksperymentów. W pierwszej kolejności opracowano konwencjonalny regulator DMC, wykorzystując do tego celu odpowiedzi z okolicy punktu równowagi  $X2 = 25\%$ ,  $P2 = 50,5$  kPa. Następnie w celu poprawienia działania układu regulacji, opra-

cowano rozmyty regulator FDMC w wersji numerycznej, szczegóły konstrukcji którego zostały zamieszczone w pracy [12].



Rys. 3. Odpowiedzi układów regulacji z regulatorami DMC (linia przerywana) i FDMC (linia ciągła) na skoki wartości zadanych do  $X2_{zad} = 17\%$  i  $P2_{zad} = 58$  kPa; a) wyjścia od góry:  $L2$ ,  $X2$  i  $P2$ , b) sterowania od góry:  $F2$ ,  $P100$  i  $F200$

Regulator ten opracowano na podstawie odpowiedzi otrzymanych z okolic czterech punktów pracy:  $X2 = 15\%$ ,  $P2 = 40$  kPa;  $X2 = 15\%$ ,  $P2 = 60$  kPa;  $X2 = 35\%$ ,  $P2 = 40$  kPa;  $X2 = 35\%$ ,  $P2 = 60$  kPa. W przypadku obydwu regulatorów przyjęto taką samą wartość współczynnika kary za zmienność sygnałów sterujących  $\lambda = 0,1$ .



W przypadku większości wymuszeń (skoków zakłóceń dodawanych do wyjść obiektu lub skoków wartości zadanych), odpowiedzi otrzymane w obydwu układach regulacji (zarówno z regulatorem konwencjonalnym, jak i z regulatorem rozmytym) były podobne, zwłaszcza w okolicy punktu pracy  $X_2 = 25\%$ ,  $P_2 = 50,5$  kPa, jedynie między przebiegami sterowań były dostrzegalne różnice. Nieco większe różnice w odpowiedziach otrzymanych po zastosowaniu obydwu regulatorów pojawiły się w przypadku odchylenia od punktu, dla którego został zaprojektowany regulator konwencjonalny.

Duże różnice w działaniu układów regulacji pojawiły się w przypadku pracy w okolicy punktu  $X_2 = 15\%$ ,  $P_2 = 60$  kPa. Wówczas wyraźnie ujawniły się zalety nieliniowego, rozmytego algorytmu regulacji predykcyjnej. Skok wartości zadanej ciśnienia  $P_2$  w górę, przy jednoczesnym skokowym zmniejszeniu wartości zadanej składu  $X_2$  okazał się być najbardziej wymagający dla regulatora konwencjonalnego. Prawdopodobnie właśnie dlatego ten przypadek posłużył jako przykład w [11]. Przebiegi otrzymane w takich warunkach pokazano na rys. 3, przy czym wartości zadane dobrano tak, aby ograniczenia nie były aktywne. Linia przerywaną oznaczono przebiegi uzyskane w układzie regulacji z regulatorem konwencjonalnym, zaś linią ciągłą – z regulatorem rozmytym. W przypadku regulatora konwencjonalnego przeregulowanie było wyraźnie większe. Większe były również zmiany sterowań.

W przykładzie zastosowano jedną z prostszych struktur regulatora rozmytego i najprostszymi sposobami jego projektowania. Sposób ten polega na pozyskaniu odpowiedzi skokowych z okolic kilku punktów pracy, a następnie zastosowaniu regulatora rozmytego FDMC. Warto zauważyć, że w rozważanym przypadku kosztem stosunkowo niewielkich nakładów (dzięki prostocie zastosowanej metody projektowania regulatora), otrzymano poprawę jakości regulacji w stosunku do regulatora konwencjonalnego.

## 7. PODSUMOWANIE

Pakiet REGZA umożliwia stosunkowo szybkie i wygodne opracowanie algorytmów sterowania wielowymiarowych obiektów regulacji zarówno w wersji liniowej jak i nieliniowej, począwszy od obróbki danych pozyskanych z procesu, które służą następnie do identyfikacji modelu obiektu, a skończywszy na doborze struktury i parametrów regulatora oraz przetestowaniu otrzymanego układu regulacji.

Dużą zaletą pakietu, oprócz stosunkowo szerokiej oferty dostępnych funkcji i algorytmów, jest intuicyjna obsługa, dzięki której opisywane programy mogą być przydatne szczególnie wówczas, gdy synteza i badania algorytmów regulacji powinny być wykonane stosunkowo szybko.

## LITERATURA

- [1] E. F. Camacho, C. Bordons: Model predictive control. Springer, 1999.
- [2] D. W. Clarke, C. Mohtadi, P. S. Tuffs: Generalized predictive control – I. The basic algorithm. *Automatica*, 1987, tom 23, nr 2, str. 137–148, II. Extensions and interpretations, str. 149–160.
- [3] C. R. Cutler, B. L. Ramaker: Dynamic matrix control – a computer control algorithm. AICHE National Meeting, Houston, Texas, 1979; *Proceedings of Joint. Aut. Control Conf.*, San Francisco, 1980.

- [4] K. Janiszowski, P. Wnuk: Pakiet programowy do celów identyfikacji modeli dynamicznych SIGNAL IDENTIFICATION. *Konferencja naukowo techniczna Automation 2000*, Warszawa, 2000, str. 57–62.
- [5] M. Ławryńczuk, P. Marusak, P. Tatjewski: Oprogramowanie do projektowania zaawansowanych układów regulacji. *Krajowa Konferencja Automatyki*, Zielona Góra, 2002, str. 483–488.
- [6] M. Ławryńczuk, P. Tatjewski: A computationally efficient nonlinear predictive control algorithms based on neural models. *Proceedings of the 8<sup>th</sup> International Conference on Methods and Models in Automation and Robotics*, Międzyzdroje, 2002, str. 781–786.
- [7] M. Ławryńczuk: Alternatywne wskaźniki jakości w nieliniowym algorytmie regulacji predykcyjnej. *Krajowa Konferencja Automatyki*, Zielona Góra, 2002, str. 159–164.
- [8] M. Ławryńczuk, P. Marusak: REGZA – pakiet oprogramowania do projektowania układów regulacji zaawansowanej – instrukcja użytkownika (wersja 2.0), Raport IAiIS PW 00–14, Warszawa, czerwiec 2001.
- [9] M. Ławryńczuk, P. Tatjewski: Obszarowe modele neuronowe procesów dynamicznych – struktura oraz algorytmy identyfikacji. *V Krajowa Konferencja Naukowo-Techniczna Diagnostyka Procesów Przemysłowych*, Łągów Lubuski, 2001, str. 145–148.
- [10] M. Ławryńczuk, P. Tatjewski: A nonlinear predictive control algorithm for processes modelled by means of neural networks. *Proceedings of the 7<sup>th</sup> International Conference on Methods and Models in Automation and Robotics*, Międzyzdroje, 2001, str. 489–494.
- [11] J. M. Maciejowski: *Predictive control with constraints*. Prentice Hall, 2002.
- [12] P. Marusak: Regulacja predykcyjna obiektów nieliniowych z zastosowaniem techniki DMC i modelowania rozmytego (rozprawa doktorska). Wydział Elektroniki i Technik Informacyjnych Politechniki Warszawskiej, Warszawa, 2002.
- [13] P. Marusak, P. Tatjewski: Output constraints in fuzzy DMC algorithms with parametric uncertainty in process models. *Proceedings of the 7<sup>th</sup> International Conference on Methods and Models in Automation and Robotics*, Międzyzdroje, 2001, str. 517–522.
- [14] P. Marusak, P. Tatjewski: Fuzzy Dynamic Matrix Control algorithms for nonlinear plants. *Proceedings of the 6<sup>th</sup> International Conference on Methods and Models in Automation and Robotics*, Międzyzdroje, 2000, str. 749–754.
- [15] R. B. Newell, P. L. Lee: *Applied process control – a case study*. Prentice Hall, 1989.
- [16] P. Tatjewski: Sterowanie zaawansowane obiektów przemysłowych; Struktury i algorytmy. Akademska Oficyna Wydawnicza EXIT, Warszawa, 2002.
- [17] P. Tatjewski, M. Ławryńczuk, P. Marusak: Wybrane algorytmy regulacji predykcyjnej procesów nieliniowych. *Krajowa Konferencja Automatyki*, Zielona Góra, 2002, str. 145–152.