

XV Krajowa Konferencja Automatyki

Tom II



**Redaktorzy:
Zdzisław Bubnicki
Roman Kulikowski
Janusz Kacprzyk**

XV Krajowa Konferencja Automatyki Tom II



Redaktorzy:
Zdzisław BUBNICKI
Roman KULIKOWSKI
Janusz KACPRZYK

ORGANIZATOR

Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓŁORGANIZATORZY

Politechnika Warszawska

Przemysłowy Instytut Automatyki i Pomiarów

Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

ORGANIZATOR

Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓLORGANIZATORZY

Politechnika Warszawska
Przemysłowy Instytut Automatyki i Pomiarów
Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

KOMITET PROGRAMOWY

Przewodniczący	Zdzisław BUBNICKI
Zastępca Przewodniczącego	Roman KULIKOWSKI

CZŁONKOWIE

Stanisław BAŃKA	Michał BIAŁKO
Mikołaj BUSŁOWICZ	Władysław FINDEISEN
Ryszard GESSING	Henryk GÓRECKI
Jakub GUTENBAUM	Jerzy JÓZEFczyk
Stanisław KACZANOWSKI	Tadeusz KACZOREK
Janusz KACPRZYK	Jerzy KLAMKA
Józef KORBICZ	Zbigniew KOWALSKI
Krzysztof KOZŁOWSKI	Juliusz L. KULIKOWSKI
Krzysztof KUŹMIŃSKI	Kazimierz MALANOWSKI
Krzysztof MALINOWSKI	Wojciech MITKOWSKI
Antoni NIEDERLIŃSKI	Władysław PEŁCZEWSKI
Tadeusz PUCHAŁKA	Leszek RUTKOWSKI
Stanisław SKOCZOWSKI	Roman SŁOWIŃSKI
Jerzy ŚWIĄTEK	Andrzej ŚWIERNIAK
Ryszard TADEUSIEWICZ	Piotr TATJEWSKI
Krzysztof TCHOŃ	Leszek TRYBUS
Jan WĘGLARZ	Andrzej P. WIERZBICKI

KOMITET ORGANIZACYJNY

Przewodniczący	Roman KULIKOWSKI
Zastępcy Przewodniczącego	Janusz KACPRZYK
	Stanisław KACZANOWSKI
	Tadeusz KACZOREK
	Krzysztof MALINOWSKI
Członkowie	Roman OSTROWSKI
	Tadeusz PUCHAŁKA
	Dariusz WAGNER
Sekretarze naukowci	Jan STUDZIŃSKI
	Jan W. OWSIŃSKI

ISBN 83-89475-01-4

Copyright © Instytut Badań Systemowych Polskiej Akademii Nauk
All rights reserved

Druk: ARGRAF, Warszawa

ROBOTY

REGULATORY ROZMYTE W STEROWANIU ROBOTA PRZEMYSŁOWEGO IRP-6[†]

Mieczysław ZACZYK

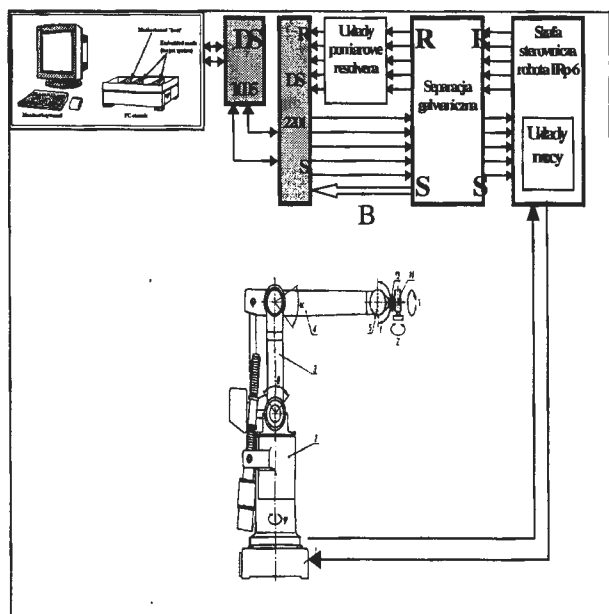
Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Al. Mickiewicza 30, 30-059 Kraków, e-mail: mza@ia.agh.edu.pl

Streszczenie: W pracy przedstawiono implementację regulatorów rozmytych w sterowniku robota oraz wyniki weryfikacji doświadczalnej zastosowania takich regulatorów w procesie sterowania robota przemysłowego IRP-6. Sterownik robota zrealizowany został z wykorzystaniem technologii szybkiego prototypowania oferowanej przez system dSPACE bazujący na karcie procesorowej DS1005.

Słowa kluczowe: Roboty przemysłowe, regulator rozmyty, szybkie prototypowanie.

1. WPROWADZENIE

Prototyp układu sterowania zrealizowany został na bazie stanowiska badawczego, w którym robot przemysłowy IRP-6 połączony jest z komputerem poprzez specjalistyczny system do szybkiego prototypowania (dSPACE), obejmujący kartę procesorową DS1005 oraz wielofunkcyjną kartę wejść/wyjść analogowych i cyfrowych DS2201. Schemat blokowy stanowiska pokazano na rys.1.



Rys. 1. Schemat blokowy stanowiska badawczego.

2. STEROWNIK ROBOTA

Algorytm realizowany przez sterownik robota wykonany został w środowisku Simulink jako schemat blokowy (rys.2). Przy wykorzystaniu narzędzi pakietu Matlab/Simulink (Real-Time Workshop) oraz pakietu dSPACE (Real-Time Interface) z utworzonego schematu blokowego generowany jest automatycznie program wykonywalny czasu rzeczywistego. Wykonywanie tej aplikacji przejmują karty procesorowe DS1005, która poprzez układy pomiarowe karty DS2201 komunikuje się z podłączonym robotem (pomiar położenia ramion z resolverów, pomiar sygnałów z czujników synchronizacyjnych oraz wysyłanie sygnałów sterujących do układów mocy robota). Komputer, na którym przygotowano schemat sterownika przejmując (poprzez pakiet ControlDesk) rolę urządzenia umożliwiającego komunikację w czasie rzeczywistym z działającą aplikacją (możliwość zmiany parametrów, odczytu pomiarów, rejestracji przebiegów itp.).

Widoczne na schemacie (rys.2) bloki DS2201ADC oraz DS2201DAC zapewniają połączenie z robotem tzn. realizują pomiar położenia kąтового ramion oraz wysyłanie wypracowanego sygnału sterującego do układów mocy robota, poprzez interfejsy pomiarowe karty DS2201.

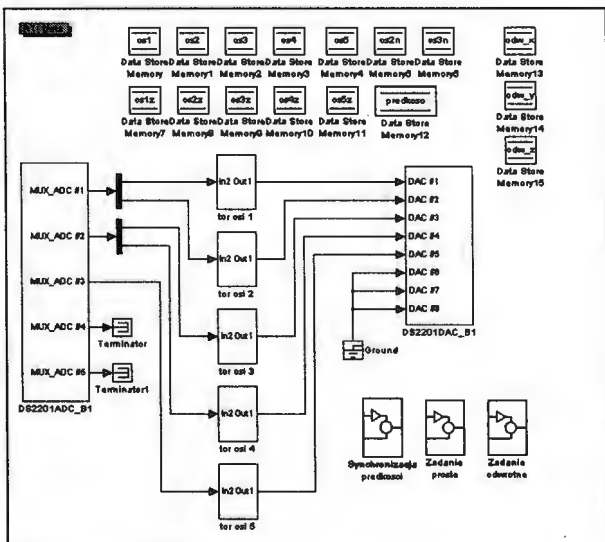
Pozostałe podsystemy realizują wszystkie typowe dla sterownika robota operacje:

- bazowanie robota (doprowadzanie ramion do pozycji bazowej)
- synchronizację prędkości ruchu ramion (jednakowy czas trwania ruchu dla ramion) oraz zadawanie prędkości roboczej
- pozycjonowanie w układzie współrzędnych konfiguracyjnych (kątowych)
- pozycjonowanie w układzie współrzędnych kartezjańskich (realizacja prostego i odwrotnego zadania kinematyki) z uwzględnieniem ograniczeń wynikających z zakresu przestrzeni roboczej.

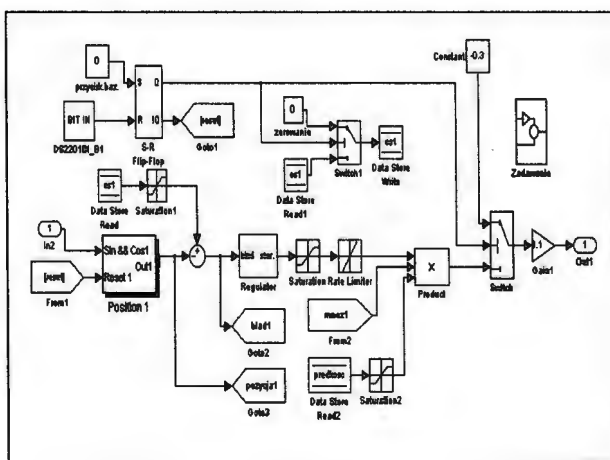
Dodatkowo w algorytmie sterownika umieszczono funkcję umożliwiającą wybór typu regulatora w torze regulacji położenia dla poszczególnych osi robota.

[†]Praca zrealizowana w ramach Badań Statutowych
Nr 11.11.120.231

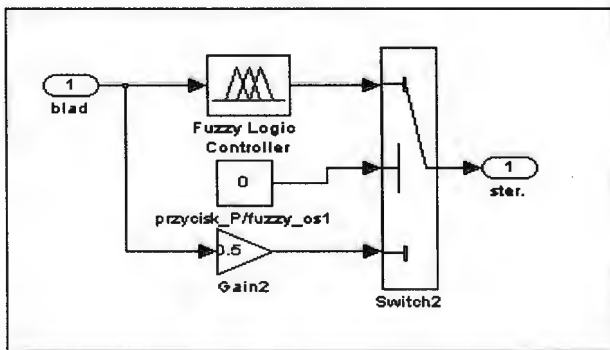
Podsystem realizujący regulację położenia pierwszej osi robota pokazano na rys.3, natomiast wybór typu regulatora na rys.4. Istnieje możliwość wybrania regulatora proporcjonalnego P lub regulatora rozmytego. Współczynnik wzmocnienia regulatora proporcjonalnego został dobrany tak, by zapewnić w układzie przebiegi bez przeregulowania. Podsystemy realizujące regulację położenia dla pozostałych osi są zrealizowane bardzo podobnie.



Rys. 2. Schemat blokowy sterownika.

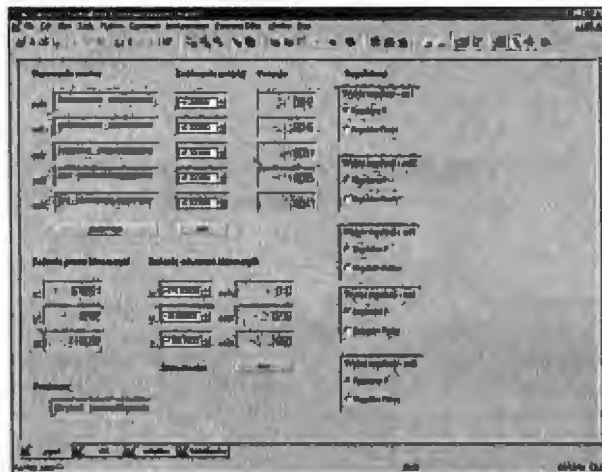


Rys. 3. Podsystem *Tor osi 1* – tor regulacji położenia wybranej osi.



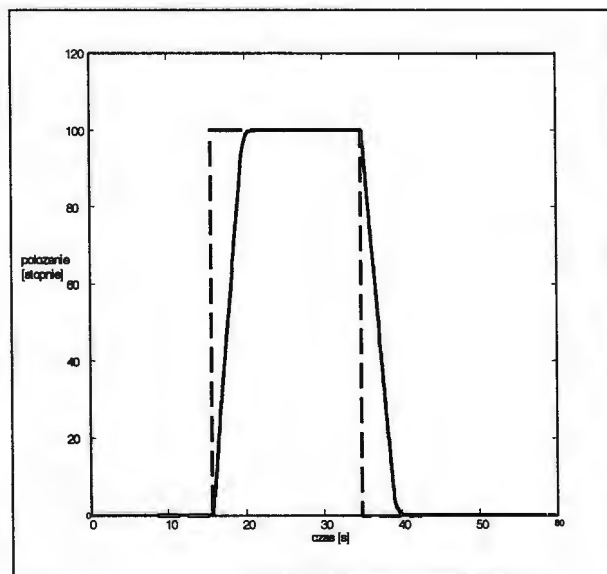
Rys. 4. Podsystem *Regulator* – wybór typu regulatora.

Komunikacja z procesem realizowana jest poprzez wykonany przy pomocy pakietu ControlDesk pulpit sterujący (rys.5). Pozwala on na wymuszanie procesu bazowania, zadawanie położenia kątowych lub kartezjańskich, zadawanie prędkości roboczej, wybór typu regulatora a także obserwację przebiegów czasowych wybranych zmiennych procesu.



Rys. 5. Panel sterowania.

Przebieg procesu pozycjonowania wybranej osi robota, z regulatorem proporcjonalnym w torze regulacji położenia, przedstawiony został na rys. 6.

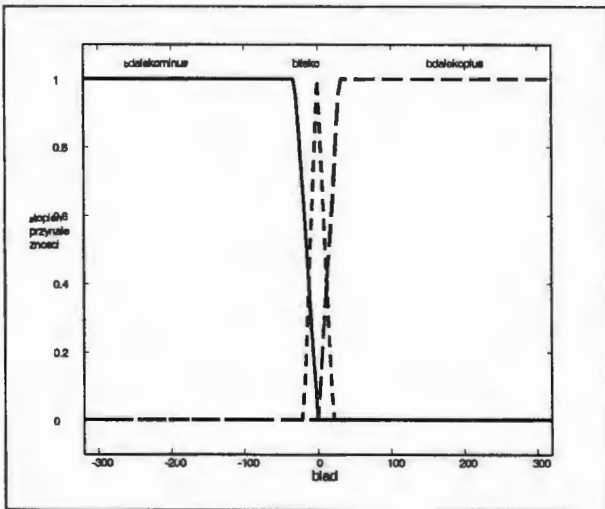


Rys. 6. Pozycjonowanie osi robota – regulator P.

3. REGULATORY ROZMYTE

Regulatory rozmyte zostały stworzone za pomocą narzędzia „fuzzy” znajdującego się w pakiecie Fuzzy Logic Toolbox, będącym dodatkiem do oprogramowania Matlab/Simulink. Umożliwia ono ręczny dobór parametrów regulatora, czyli funkcji przynależności dla wejść i dla wyjść, reguł opisujących sposób działania regulatora czy sposobu defuzyfikacji. W ten sposób

utworzono regulatory typu Mamdani – z trzema i z pięcioma funkcjami przynależności.
 Na rys.7 przedstawiono przykładowe dobrane funkcje przynależności dla wejścia, a na rys.8 dla wyjścia, dla trzech funkcji.



Rys. 7. Funkcje przynależności dla wejścia.

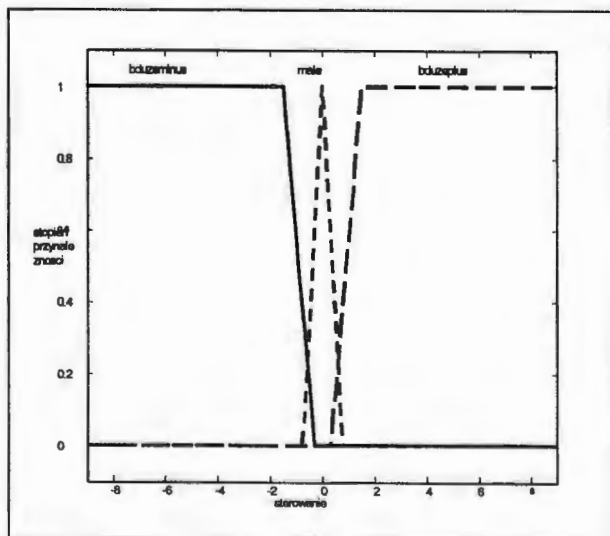
Dobrane zostały trzy funkcje przynależności przypisujące uchyb wejściowy do zbiorów: *bdalekominus* (uchyby w zakresie $[-320^\circ, 0^\circ]$), *blisko* ($[-20^\circ, 20^\circ]$) i *bdalekoplus* ($[0^\circ, 320^\circ]$).

Zaproponowane sterowania to: *bduzeminus* ($[-5, -0.3]$), *male* ($[-0.75, 0.75]$) oraz *bduzeplus* ($[0.3, 5]$).

Dla opisu działania regulatora ustalone zostały trzy proste reguły:

1. If (input1 is *bdalekominus*) then (output1 is *bduzeminus*) (1)
2. If (input1 is *blisko*) then (output1 is *male*) (1)
3. If (input1 is *bdalekoplus*) then (output1 is *bduzeplus*) (1).

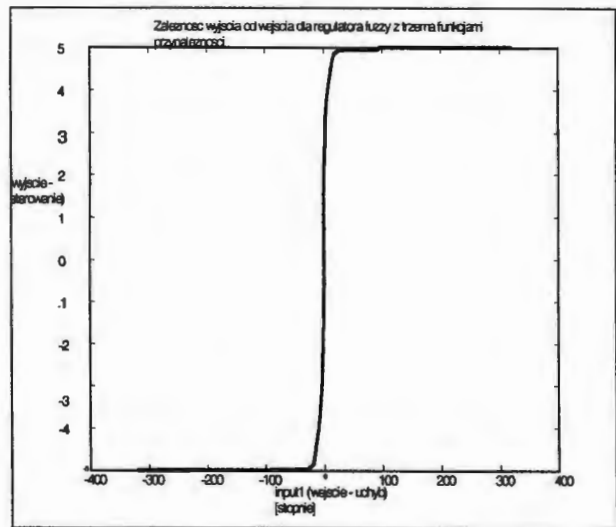
Dla realizacji procesu defuzyfikacji wybrana została metoda środka ciężkości.



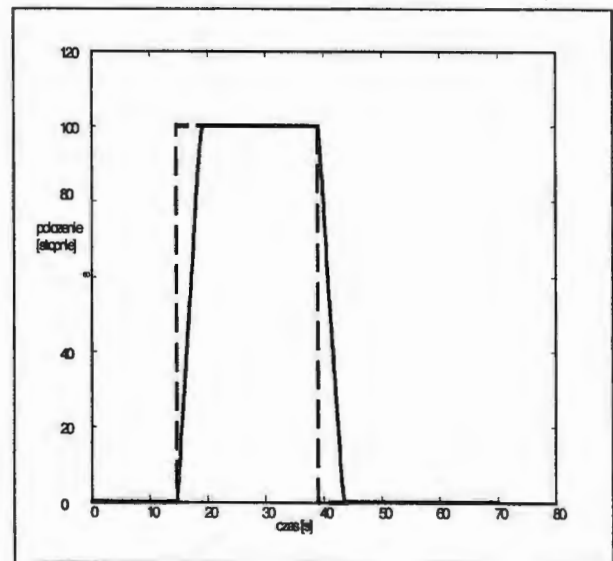
Rys. 8. Funkcje przynależności dla wyjścia.

Dla takich funkcji przynależności oraz przyjętych reguł uzyskana została zależność sterowania od uchybu przedstawiona na rys. 9.

Przebieg pozycjonowania osi robota dla skokowej zmiany wartości zadanej pokazano na rys.10.



Rys. 9. Zależność sterowania od uchybu.



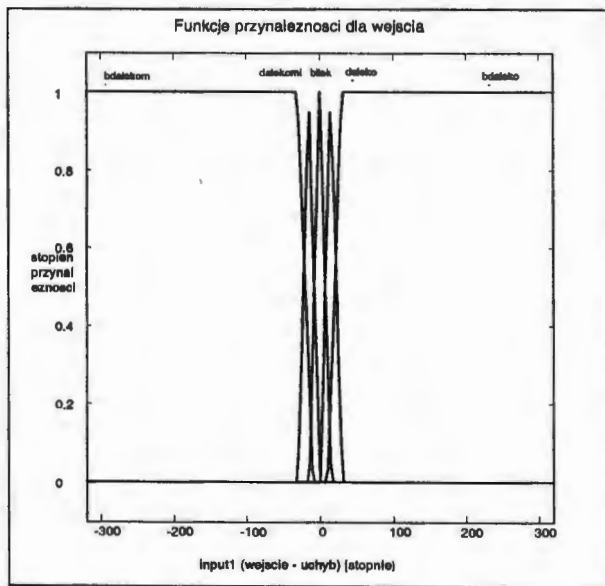
Rys. 10. Pozycjonowanie wybranej osi robota.

W dalszej części eksperymentu został stworzony nieco bardziej złożony regulator, ale dający większe możliwości ustalania sposobu dochodzenia ramienia do pozycji zadanej.

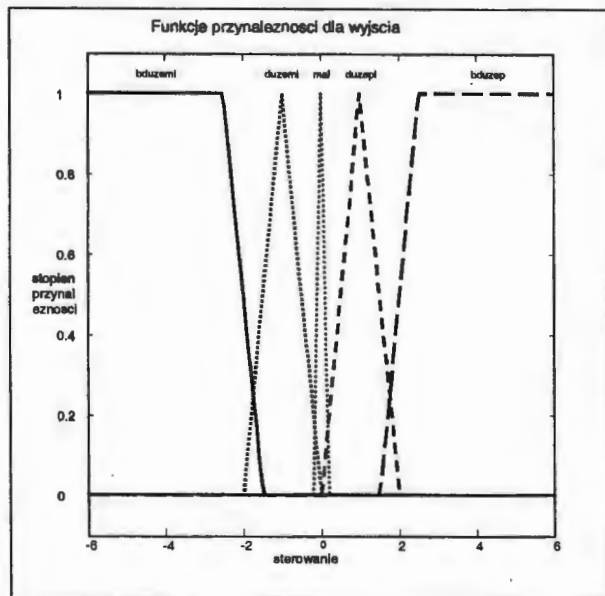
Stworzono pięć funkcji przynależności, przyporządkowujących wejściowy uchyb do zbiorów: *bdalekominus* (uchyby w zakresie $[-320^\circ, -10^\circ]$), *dalekominus* ($[-30^\circ, -15^\circ]$), *blisko* ($[-15^\circ, 15^\circ]$), *dalekoplus* ($[15^\circ, 30^\circ]$), *bdalekoplus* ($[10^\circ, 320^\circ]$).

Utworzenie dwóch dodatkowych funkcji przynależności pozwoliło zróżnicować wartości sterowania w zależności od wartości uchybu: *bduzeminus* ($[-5, -1.5]$), *duzeminus* ($[-2, 0]$), *male* ($[-0.2, 0.2]$), *duzeplus* ($[0, 2]$), *bduzeplus* ($[1.5, 5]$).

Na rys. 11 przedstawione zostały przebiegi funkcji przynależności dla wejścia, a na rys.12 funkcje przynależności dla wyjścia.



Rys. 11. Funkcje przynależności dla wejścia.

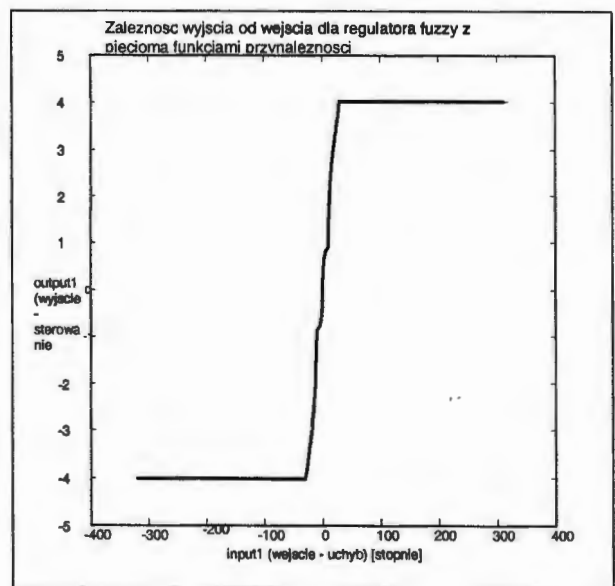


Rys. 12. Funkcje przynależności dla wyjścia.

Poniżej przedstawiono zbiór reguł określających działanie regulatora:

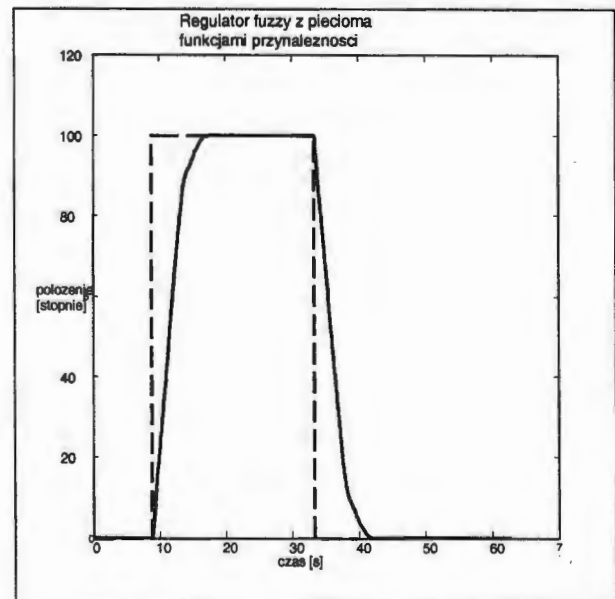
1. If (input1 is bdalekominus) then (output1 is bduzeminus) (1)
2. If (input1 is dalekominus) then (output1 is duzeminus) (1)
3. If (input1 is blisko) then (output1 is male) (1)
4. If (input1 is dalekoplus) then (output1 is duzepus) (1)
5. If (input1 is bdalekoplus) then (output1 is bduzepus) (1)

Dla tak dobranych funkcji przynależności otrzymano zależność sterowania od uchybu przedstawioną na Rys.13.



Rys. 13. Zależność sterowania od uchybu.

Przebieg procesu pozycjonowania wybranej osi robota, dla skokowej zmiany wartości zadanej, pokazano na rys.14.



Rys. 14. Pozycjonowanie wybranej osi robota.

Jak widać na wykresie, zgodnie z zamierzeniem uzyskano efekt wcześniejszego hamowania ramienia. Moment rozpoczęcia hamowania można zmieniać, modyfikując rozmieszczenie zbiorów rozmywających wejście, wielkość sterowania można natomiast dostosować zmieniając przynależność wartości sterowań do jego zbiorów rozmywających.

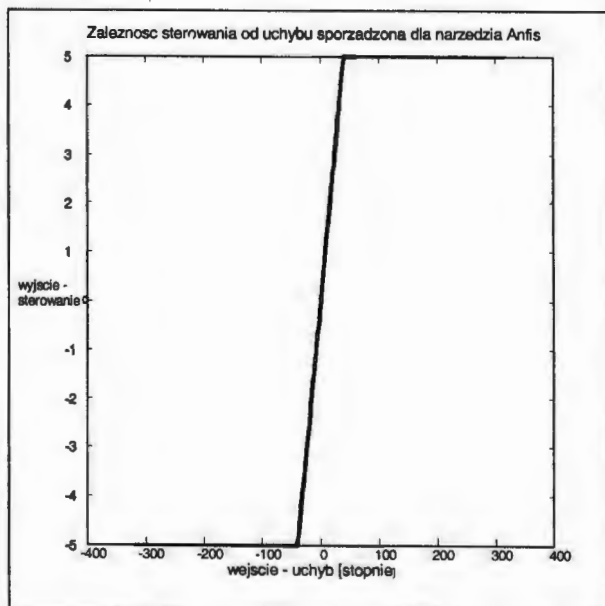
Przeprowadzony eksperyment pokazuje, że regulator fuzzy wprowadza większą możliwość wpływania na przebieg dochodzenia ramienia do jego pozycji zadanej.

4. REGULATOR FUZZY WYGENEROWANY ZA POMOCĄ NARZĘDZIA ANFIS

Środowisko Matlab wraz z Fuzzy Logic Toolbox dostarcza jeszcze innego sposobu tworzenia regulatorów rozmytych. Jest nim narzędzie Anfis wraz z edytorem posiadającym interfejs graficzny – Anfis Editor. Program umożliwia utworzenie regulatora rozmytego na podstawie dostarczonych mu danych w postaci charakterystyki przedstawiającej zależność wyjścia regulatora od jego wejścia. Możliwe jest tworzenie tylko regulatorów typu Sugeno.

Program wymaga także podania m.in. ilości i rodzaju funkcji rozmywających wejście oraz ilości punktów, w których będzie dokonywana optymalizacja. Zwiększanie tych parametrów (w pewnych granicach) wpływa korzystnie na przebieg optymalizacji, rośnie jednak nakład obliczeniowy a co za tym idzie czas optymalizacji. Przeprowadzono szereg eksperymentów dla podanej wyżej charakterystyki, zadowalający efekt został uzyskany dla dziesięciu funkcji przynależności typu „trim” oraz 640 punktów.

Na Rys.15 przedstawiono przebieg charakterystyki uczącej dla narzędzia Anfis, a na Rys.16 porównanie przebiegu danych uczących z charakterystyką otrzymaną w wyniku działania programu.

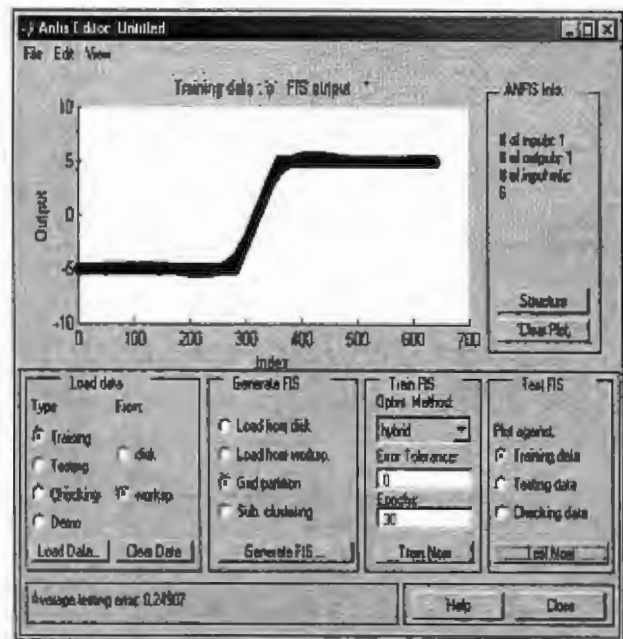


Rys. 15. Zależność sterowania od uchybu – charakterystyka dla programu Anfis

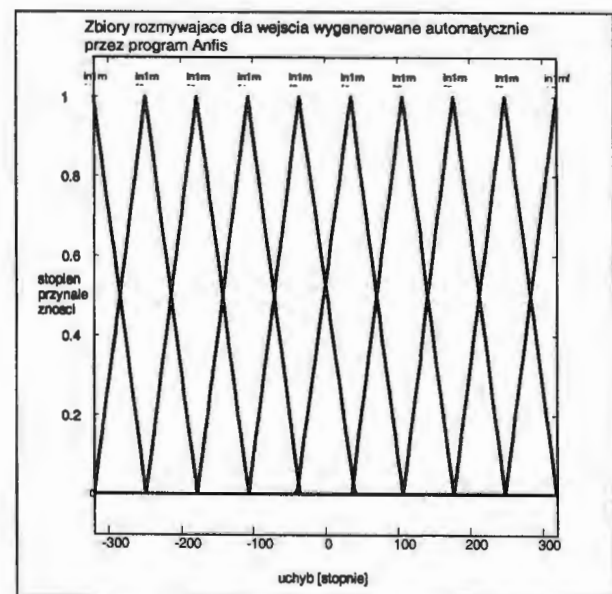
Reguły wygenerowane automatycznie przez program:

1. If (input1 is in1mf1) then (output is out1mf1) (1)
2. If (input1 is in1mf2) then (output is out1mf2) (1)
3. If (input1 is in1mf3) then (output is out1mf3) (1)
4. If (input1 is in1mf4) then (output is out1mf4) (1)
5. If (input1 is in1mf5) then (output is out1mf5) (1)
6. If (input1 is in1mf6) then (output is out1mf6) (1)
7. If (input1 is in1mf7) then (output is out1mf7) (1)
8. If (input1 is in1mf8) then (output is out1mf8) (1)
9. If (input1 is in1mf9) then (output is out1mf9) (1)
10. If (input1 is in1mf10) then (output is out1mf10) (1)

W wyniku działania tego narzędzia otrzymano zbiór funkcji przynależności przedstawiony na Rys.17.



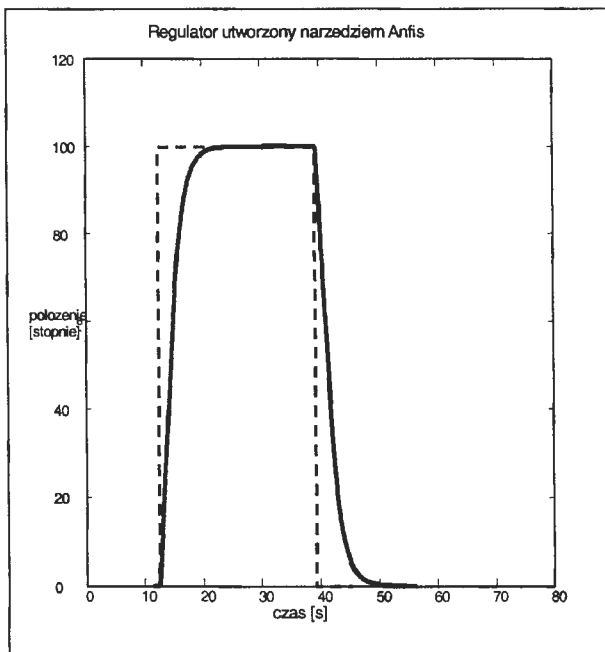
Rys. 16. Porównanie danych uczących z charakterystyką otrzymaną



Rys. 17. Funkcje przynależności dla wejścia wygenerowane przez program Anfis

Przebieg procesu pozycjonowania wybranej osi (dla skokowej zmiany wartości zadanej) dla tak wygenerowanego regulatora rozmytego pokazano na Rys.18.

Uzyskany przebieg dochodzenia osi do wartości zadanej jest zbliżony do tego, jaki uzyskany został dla regulatora proporcjonalnego. Wynika to z zadanej charakterystyki uczącej, której przebieg odzwierciedla zmiany sygnału sterującego dla regulatora proporcjonalnego.



Rys. 18. Pozycjonowanie wybranej osi robota.

5. PODSUMOWANIE

Przeprowadzone badania wykazały poprawne działanie zaimplementowanych regulatorów rozmytych. Pierwszy ze stworzonych regulatorów to układ zapewniający sterowanie osią robota przy użyciu algorytmu opisanego zaledwie przez trzy funkcje przynależności klasyfikujące wejście i wyjście. Takie rozwiązanie jest wystarczające do skutecznego sterowania ramieniem i jego efekt jest zbliżony do działania regulatora proporcjonalnego. Wprowadzenie dodatkowych funkcji przynależności i reguł pozwala w łatwy sposób kształtować charakterystykę ruchu ramienia. W przedstawionym przykładzie (algorytm opisany pięcioma funkcjami przynależności) uzyskano efekt wcześniejszego hamowania osi robota.

Pewną automatyzację przebiegu projektowania regulatorów wprowadza narzędzie „Anfis” wykorzystane do stworzenia przykładowego regulatora.

FUZZY CONTROLLERS IN THE PROCESS OF CONTROL OF INDUSTRIAL ROBOT IRP-6

Abstract: In the paper the implementation of fuzzy controllers in the process of control of industrial robot IRP-6 are presented. The controller for the robot has been realized on the base of the dSPACE board DS1005 with the use of rapid control prototyping technology. The results of experimental verification are also presented.

Literatura

- [1] Craig J. (1995) *Wprowadzenie do robotyki: mechanika i sterowanie*. WNT, Warszawa.
- [2] Piegat A. (1999) *Modelowanie i sterowanie rozmyte*. Wydawnictwo EXIT, Warszawa.
- [3] Spong, M.W., Vidyasagar M. (1997) *Dynamika i sterowanie robotów*. WNT, Warszawa
- [4] Zaczyk M. (2003) Wykorzystanie obrazu z kamery w procesie śledzenia trajektorii przez robota IRp. *Materiały IV Krajowej Konferencji MSK'03*, Kraków, 443-448
- [5] *Real-Time Interface - Implementation Guide*. dSPACE GmbH, Germany 1999.
- [6] *ControlDesk-Experiment Guide*. dSPACE GmbH, Germany 1999.
- [7] *Fuzzy Logic Toolbox-User's Guide*. MathWorks, Inc. 2001.



Instytut Badań Systemowych
Polskiej Akademii Nauk

ISBN 83-89475-01-4