

**AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA  
W KRAKOWIE**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,  
INFORMATYKI I ELEKTRONIKI**

**KATEDRA AUTOMATYKI**

**PRACA DYPLOMOWA**

**Tomasz Tęcza**

**KIERUNEK: AUTOMATYKA I ROBOTYKA**

**TEMAT: PROJEKT, IMPLEMENTACJA I TESTOWANIE  
WYBRANEGO REGULATORA ROZMYTEGO**

Ocena:.....

pracą pod kierunkiem  
dr hab. inż. Antoniego Ligęzy

KRAKÓW 1998

## **Spis treści.**

## Spisrzsunków

RYS. 2.1- RÓŻNE FORMY FUNKCJI PRZYNALEŻNOŚCI: A) TRÓJKĄTNA, B) TRAPEZOIDALNA, C) W KSZTAŁCIE DZWONU.....	12
RYS. 3.1.WEWNĘTRZNA STRUKTURA FLC W UKŁADZIE STEROWANIA.....	20
RYS. 3.2.FUNKCJE PRZYNALEŻNOŚCI ZBIORÓW ROZMYTYCH ODNIESIENIA REPREZENTUJĄCE ZMIENNE LINGWISTYCZNE: DODATNI, W PRZYBLIŻENIU ZERO, UJEMNY. TE TRZY ZMIENNE LINGWISTYCZNE TWORZĄ ZBIÓR TERMINU UCHYB. SYMBOL $\mu$ UCHYBU OZNACZA FUNKCJĘ PRZYNALEŻNOŚCI ZBIORU ROZMYTEGO ODNIESIENIA ZWIĄZANĄ Z DOWOLNYMI ZMIENNYMI LINGWISTYCZNYMI, KTÓRE NALEŻĄ DO ZBIORU TERMINU UCHYB. WSZYSTKIE TRZY ZBIORY ROZMYTE SĄ OKREŚLONE NA TEJ SAMEJ PRZESTRZENI [-1, 1].....	20
RYS. 3.3.FUNKCJE PRZYNALEŻNOŚCI ZBIORÓW ROZMYTYCH ODNIESIENIA ZWIĄZANE ZE ZBIORAMI TERMINU UCHYBU E, ZMIANY UCHYBU $\Delta E$ I ZMIANY STEROWANIA $\Delta U$ STOSOWANE W BAZIE REGUŁ W TABELI 3.2.KAŻDY ZBIÓR TERMINU ZAWIERA SIEDEM ETYKIET LINGWISTYCZNYCH DUŻY DODATNI (DD), ŚREDNI DODATNI (SD), MAŁY DODATNI (MD), ZERO (Z), MAŁY UJEMNY (MU), ŚREDNI UJEMNY (SU) I DUŻY UJEMNY (DU). WSZYSTKIE ZBIORY ROZMYTE ODNIESIENIA SĄ OKREŚLONE NA PRZESTRZENI [-1, 1].....	23
RYS. 3.4.OBLICZANIE POZIOMU ZAPŁONU I TEJ REGUŁY: A) ZMIENNE WEJŚCIOWE FLC PRZYJMUJĄ WARTOŚCI NIEROZMYTE $U1=X1*U2=X2*$ ; STOSUJE SIĘ WYRAŻENIE 3.5; B) ZMIENNE WEJŚCIOWE FLC PRZYJMUJĄ WARTOŚCI ROZMYTE $U1=A1$ I $U2=A2$ ; STOSUJE SIĘ WYRAŻENIE 3.6.....	26
RYS. 3.5.REPREZENTACJA GRAFICZNA ALGORYTMU WNIOSKOWANIA: WEJŚCIE NIEROZMYTE $X1*=E(K)$ I $X2*=E(K)$ ; METODA WYOSTRZANIA MOM; WYJŚCIE NIEROZMYTE JEST WYPROWADZONE PRZEZ BAZĘ REGUŁ FLC I RÓWNA SIĘ $Y*=U(K)$ .....	30
RYS. 5.1-ZALEŻNOŚĆ POMIĘDZY RUCHEM OBROTOWYM A LINIOWYM W SILNIKU.	36
RYS. 5.2- SCHEMAT TWORNIKA SILNIKA.....	37
RYS. 5.3- SCHEMAT OBWODU SILNIKA.....	38
RYS. 6.1- SCHEMAT SYSTEMU STEROWANIA I SYMULACJI DLA REGULATORS FLC TYPU PI.....	43
RYS. 6.2- MATLABOWSKI SCHEMAT MODELU SILNIKA.....	43
RYS. 6.3- SCHEMAT SYSTEMU STEROWANIA I SYMULACJI DLA REGULATORS FLC TYPU PD.....	46
RYS. 7.1- OGÓLNY WYGLĄD EDYTORA FIS DLA ZMIENNEJ DCMAM2.....	48
RYS. 7.2- EDYTOR FUNKCJI PRZYNALEŻNOŚCI - MEMBERSHIP FUNCTION EDITOR.	50
RYS. 7.3- ZBIORY FUNKCJI PRZYNALEŻNOŚCI DLA FLC TYPU PI: A) BŁĄD, B) POCHODNĄ BŁĘDU, C) ZMIANA STEROWANIA.....	52
RYS. 7.4- EDYTOR ZBIORU REGUŁ - RULE EDITOR.....	53
RYS. 7.5- WPŁYW ILOŚCI REGUŁ NA OBROTY SILNIKA PRZY ROZRUCHU DLA FLC TYPU PI.....	56

<b>RYS. 7.6- WPŁYW METODY WYOSTRZANIA NA STABILIZACJĘ OBROTÓW DLA FLC TYPU PI.....</b>	<b>57</b>
<b>RYS. 7.7- RULE VIEWER.....</b>	<b>58</b>
<b>RYS. 7.8- POWIERZCHNIA ZALEŻNOŚCI ZMIENNEJ ZMIANA STEROWANIA OD ZMIENNYCH WEJŚCIOWYCH DLA FLC TYPU PI.....</b>	<b>58</b>
<b>RYS. 7.9- OGÓLNY WYGLĄD EDYTORA FIS DLA ZMIENNEJ DCMAMDAN.....</b>	<b>59</b>
<b>RYS. 7.10- ZBIORY FUNKCJI PRZYNALEŻNOŚCI DLA FLC TYPU PD: A) BŁĄD, B) ZMIANA BŁĘDU, C) STEROWANIE.....</b>	<b>62</b>
<b>RYS. 7.11- WPŁYW METODY WYOSTRZANIA NA ZACHOWANIE SIĘ OBIEKTU DLA FLC TYPU PD.....</b>	<b>64</b>
<b>RYS. 7.12- ZALEŻNOŚĆ ZMIENNEJ WYJŚCIOWEJ OD ZMIENNYCH WEJŚCIOWYCH DLA FLC TYPU PD.....</b>	<b>64</b>
<b>RYS. 8.1- PRZEBIEGI DLA PRACY SILNIKA BEZ OBCIĄŻENIA PRZY STEROWANIU FLC TYPU PI: A) PRĘDKOŚĆ, B) PRĄD, C) BŁĄD, D) POCHODNA BŁĘDU, E) NAPIĘCIE, F) ZMIANA STEROWANIA.....</b>	<b>65</b>
<b>RYS. 8.2- PRZEBIEG ROZRUCHU SILNIKA DLA RÓŻNYCH WARTOŚCI ZADANYCH OBROTÓW PRZY STEROWANIU FLC TYPU PI.....</b>	<b>66</b>
<b>RYS. 8.3- PRZEBIEGI PRACY SILNIKA DLA OBCIĄŻENIA SKOKIEM JEDNOSTKOWYM W CHWILI <math>T=3S</math> PRZY STEROWANIU FLC TYPU PI: A) PRĘDKOŚĆ, B) PRĄD, C) BŁĄD, D) POCHODNA BŁĘDU, E) NAPIĘCIE.....</b>	<b>67</b>
<b>RYS. 8.4- PRZEBIEGI PRACY SILNIKA DLA PRZYŁOŻONEGO OBCIĄŻENIA SKOKIEM JEDNOSTKOWYM I FALĄ PROSTOKĄTNĄ PRZY STEROWANIU FLC TYPU PI: A) PRĘDKOŚĆ, B) PRĄD, C) BŁĄD, D) POCHODNA BŁĘDU, E) NAPIĘCIE, F) ZMIANA STEROWANIA.....</b>	<b>69</b>
<b>RYS. 8.5- PRZEBIEGI PRACY SILNIKA DLA OBCIĄŻENIA FALĄ PROSTOKĄTNĄ PRZY STEROWANIU FLC TYPU PI: A) PRĘDKOŚĆ, B) PRĄD, C) BŁĄD, D) POCHODNA BŁĘDU, E) NAPIĘCIE, F) ZMIANA STEROWANIA.....</b>	<b>70</b>
<b>RYS. 8.6- PRZEBIEGI DLA ROZRUCHU SILNIKA BEZ OBCIĄŻENIA PRZY STEROWANIU REGULATOREM FLC TYPU PD: A) OBROTY, B) PRĄD, C) BŁĄD, D) ZMIANA BŁĘDU, E) NAPIĘCIE.....</b>	<b>72</b>
<b>RYS. 8.7- PRZEBIEG ROZRUCHU SILNIKA DLA RÓŻNYCH WARTOŚCI ZADANYCH OBROTÓW PRZY STEROWANIU FLC TYPU PD.....</b>	<b>73</b>
<b>RYS. 8.8- PRZEBIEGI DLA SILNIKA PRZY OBCIĄŻENIU PRZYŁOŻONYM W TRAKCIE ROZRUCHU DLA STEROWANIA FLC TYPU PD: A) OBROTY, B) PRĄD, C) BŁĄD, D) ZMIANA BŁĘDU, E) NAPIĘCIE.....</b>	<b>74</b>
<b>RYS. 8.9- PRZEBIEGI DLA SILNIKA PRZY OBCIĄŻENIU FALĄ PROSTOKĄTNĄ I SKOKIEM JEDNOSTKOWYM DLA STEROWANIA FLC TYPU PD: A) OBROTY, B) PRĄD, C) BŁĄD, D) ZMIANA BŁĘDU, E) NAPIĘCIE.....</b>	<b>75</b>
<b>RYS. 8.10- PRZEBIEGI PRACY SILNIKA DLA OBCIĄŻENIA FALĄ PROSTOKĄTNĄ PRZY STEROWANIU FLC TYPU PD: A) PRĘDKOŚĆ, B) PRĄD, C) BŁĄD, D) ZMIANA BŁĘDU, E) NAPIĘCIE.....</b>	<b>76</b>
<b>RYS. 8.11- PORÓWNANIE DZIAŁANIA REGULATORA ROZMYTEGO Z REGULATOREM KONWENCJONALNYM PRZY STAŁYM OBCIĄŻENIU.....</b>	<b>78</b>

**RYS. 8.12 - PORÓWNANIE DZIAŁANIA REGULATORA ROZMYTEGO Z REGULATOREM KONWENCJONALNYM PRZY OBCIĄŻENIU IMPULSOWYM.....78**

**RYS. 8.13- - PORÓWNANIE DZIAŁANIA REGULATORA ROZMYTEGO Z REGULATOREM KONWENCJONALNYM PRZY OBCIĄŻENIU W FORMIE FALI PROSTOKĄTNEJ... 79**

## **Spis tabel**

**TABELA 3.1- PRZYKŁAD BAZY REGUŁ PROSTEGO FLC..... 19**

**TABELA 3.2.POPRAWIONA BAZA REGUŁ FLC DO STEROWANIA OBIEKTÓW APROKSIMOWANYCH PRZEZ UKŁAD PIERWSZEGO RZĘDU Z CZASEM RZECZYWISTYM... 22**

**TABELA 7.1- ZBIÓR REGUŁ DLA REGULATORA FLC TYPU PI..... 54**

**TABELA 7.2- ZBIÓR REGUŁ DLA REGULATORA FLC TYPU PD..... 62**

**TABELA 8.1- WPŁYW CZYNNIKÓW SKŁADOWYCH REGULATORA NA ZACHOWANIE SIĘ OBIEKTU I EFEKTYWNOŚĆ STEROWANIA..... 81**

## **Wstęp**

### ***Cel pracy uzasadnieniu wyboru tematu***

Zagadnienia logiki rozmytej sterowania rozmytego, w przeciwieństwie do logiki klasycznej i konwencjonalnej teorii sterowania, są stosunkowo nowe i ulegają ciągłej ewolucji. Literatura tej dziedziny stale się powiększa, lecz nie ma w niej zbyt wielu publikacji, które traktowałyby sterowanie rozmyte od strony prostego, a jednocześnie praktycznego zastosowania. Sterowanie rozmyte, które ma już zastosowanie w przemyśle, jest godne uwagi ze względu na odmienny, nowatorski charakter, jak i na fakt wykorzystania wiedzy zapisanej w sposób lingwistyczny bezpośrednio do procesu regulacji.

Celem pracy jest zaprojektowanie, zaimplementowanie i przetestowanie wybranego regulatora rozmytego. Podstawowym założeniem jest sprawdzenie, czy zaprojektowany regulator jest przynajmniej równie skuteczny jak regulator konwencjonalny. Ponadto przeprowadzone zostaną:

- praktyczna analiza procesu syntezy regulatora,
- analiza elementów strojenia,
- ocena skuteczności zaprojektowanego regulatora i porównanie z regulatorem konwencjonalnym,
- zebranie uzyskanych w trakcie pracy doświadczeń.

W związku z tym, że praca ma charakter testowy i eksperymentalny, zostaną pokazane zarówno plusy jak i minusy sterowania rozmytego. W pracy zostanie położony nacisk na przebadanie pracy samego regulatora, a nie na najefektowniejsze sterowanie konkretnym obiektem. Praca ma również pokazać, że logika rozmyta jest łatwa do zrozumienia a sterowanie rozmyte bardzo elastyczne pod względem strojenia, toleruje jednocześnie nieprecyzyjność informacji. Pokaże w jaki sposób można wykorzystać wiedzę wyrażoną w języku naturalnym bezpośrednio do syntezy regulatora.

## **Zakres pracy**

W zakres pracy wchodzi dokładne przedstawienie zagadnień logiki rozmytej jak i jej praktyczne zastosowanie do sterowania konkretnym obiektem. W rozdziale drugim przedstawiona jest historia rozwoju logiki rozmytej jak i sterowania rozmytego, począwszy od początków lat sześćdziesiątych skończywszy na chwili obecnej. Rozdział ten przedstawia również podstawy teorii logiki rozmytej, różnice pomiędzy klasyczną teorią zbiorów a teorią zbiorów rozmytych. Rozdział trzeci wchodzi już w zagadnienia sterowania rozmytego. Po krótkim porównaniu ze sterowaniem konwencjonalnym, uwaga skupia się już na samym regulatorze rozmytym. Wyjaśniona jest jego struktura i zasada działania, opisany sposób projektowania oraz bardzo ważny w tej dziedzinie proces wnioskowania. Rozdział kończy się bardzo przydatnym opisem podstawowego algorytmu wnioskowania. Rozdział czwarty to w pewnym sensie rozwinięcie rozdziału trzeciego. Opisuje bowiem już całe modele rozmyte, ich rodzaje, postępy w rozwoju oraz bezpośrednio podejście do konstruowania modeli lingwistycznych. Rozdział piąty to krótki opis wybranego modelu sterowanego obiektu - silnika obcowzbudnego prądu stałego - zawierający opis matematyczny oraz parametry wewnętrzne silnika. Rozdział szósty wyjaśnia sposób w jaki został zamodelowany system do sterowania i symulacji obiektu silnika. Natomiast rozdział siódmy to szczegółowy opis całego procesu strojenia regulatora rozmytego, doboru parametrów i nastaw. W ósmym rozdziale zawarte są wyniki przeprowadzonych testów sterowania oraz komentarz dotyczący efektywności tego sterowania. Zamieszczony jest również porównanie z regulatorem konwencjonalnym. Rozdział dziewiąty zawiera podsumowanie pracy i wnioski dotyczące przydatności sterowania rozmytego.

## Historia rozwoju i teoria logiki rozmytej

### *Historia rozwoju stan aktualny*

Minęło ponad trzydzieści lat od sformułowania podstaw teorii zbiorów rozmytych przez Lotfi A. Zadeha z Uniwersytetu Kalifornijskiego [1]. W ciągu ostatniego ćwierćwiecza teoria ta znakomicie się rozwinęła, wkraczając zarówno do wielu dziedzin matematyki (topologia, teoria miary, lingwistyka i in.), techniki (automatyka, robotyka, informatyka, sztuczna inteligencja), jak również do ekonomii i wielu innych dziedzin. Jednym z najbardziej widocznych dowodów przydatności teorii zbiorów rozmytych jest jej zastosowanie w technice sterowania i systemach ekspertowych o różnym przeznaczeniu, a ostatnio w technologii maszyn. Zadeh, w swoich kolejnych publikacjach z lat 1968 i 1973, przedstawił istotne z punktu widzenia zastosowań praktycznych pojęcia, takie jak: zmienna lingwistyczna (*linguistic variable*), funkcja przynależności (*membership function*), rozmyte zdania warunkowe (*fuzzy conditional statement*), złożeniowa reguła wnioskowania (*compositional rule of inference*) czy algorytm rozmyty (*fuzzy algorithm*) [2, 3].

Prace Zadeha, uważanego za ojca logiki rozmytej, były silnym impulsem do zastosowań teorii zbiorów rozmytych w najróżnorodniejszych dziedzinach, od biologii i medycyny, przez nauki społeczne i ekonomię, do nauk przyrodniczych i technicznych. Już na początku lat siedemdziesiątych podjęto próbę zastosowania nowej teorii do sterowania procesami technologicznymi. Początkowym centrum badawczym w tej dziedzinie był Queen Mary College (Londyn).

Począwszy od pierwszego zastosowania regulatorów rozmytych do sterowania ciśnieniem pary w kotle i szybkością silnika parowego przez Mamdaniego w 1974 r., w rozwoju dziedziny logiki rozmytej można wyróżnić dwa główne okresy [4]. Pierwszy okres obejmował eksperymenty na skalę laboratoryjną i prototypowe stanowiska, a drugi instalacje przemysłowe.

Jak już wspomniano Mamdani jako pierwszy zaproponował zastosowanie aparatu zmiennych lingwistycznych do syntezy regulatorów rozmytych. Drugi eksperyment to sterowanie procesem grzewczym wody, przeprowadzony przez Kickert i Van Nauta Lemke w 1976 r. [5]. Następnie Pappis i Mamdani (1977 r.) [6] zastosowali regulator rozmyty do sterowania ruchem ulicznym na prostym skrzyżowaniu dwupasmowych jezdni. Regulator rozmyty uzyskał lepsze wskaźniki niż regulator konwencjonalny.



W tym samym roku dwie niezależne grupy badaczy: pierwsza z nich to Tong, Beck, Latten [7], a druga to Yagishita, Itoh, Sugeno [8], opisały zastosowanie sterowania rozmytego do oczyszczania wody. Również w 1977 r. Van Lauta Lemke, van Amerongen i van der Veen zaprojektowali autopilota statków ze wspomaganiami systemu rozmytego wnioskowania. W tym okresie powstały też pierwsze prace teoretyczne polskich naukowców, Kacprzyk i Pedrycz, dotyczącej logiki rozmytej [9].

Wpływ na rozwój teorii zbiorów rozmytych w latach osiemdziesiątych miały dwa zdarzenia. Pierwsze z nich to zastosowanie regulatora rozmytego dla sterowania istniejącą instalacją przemysłową pieców obrotowych w cementowni (Dania, 1979 r.), opisane przez Larsena w 1980 r. [10]. W tym eksperymencie regulator rozmyty wykorzystywał wiedzę operatora, zapisaną w tzw. protokole sterowania. Okazało się, że zastosowanie regulatora rozmytego pozwoliło na poprawę sterowania, jak również na pewne oszczędności paliwa w wyżej wymienionych piecach. Odpowiednie urządzenie i oprogramowanie zostało wykonane przez duńską firmę Smith i spółka. Drugie zdarzenie to pierwsza w Japonii (Sendai, 1985 r.), na tak dużą skalę, zakończona sukcesem adaptacja regulatora rozmytego do sterowania kolejką podziemną. Te bardzo ważne eksperymenty na rzeczywistych instalacjach przemysłowych pokazały, że analogiczne zastosowania będą możliwe w bliskiej przyszłości.

Drugi okres rozpoczął się w połowie lat osiemdziesiątych i objął szeroką gamę przemysłowych zastosowań, które dokonały również przełomu w sposobie konstruowania złożonych układów elektronicznych. Jako pierwsze należy wymienić zastosowania systemów opartych na zbiorach rozmytych opracowane przez firmę Matsushita Electric Ind. do konstrukcji pralek automatycznych. Regulatory rozmyte decydują czy ładunek prania jest mały czy też duży. Decydują też o długości prania na podstawie stopnia zabrudzenia mierzonego przez specjalne sensory. One to decydują również jakich detergentów i ile wody należy użyć, w celu optymalnego wykonania takiego prania.

Ta sama firma wprowadziła na rynek interesujący odkurzacz, którego ssanie jest regulowane za pomocą regulatora rozmytego w zależności od stopnia zabrudzenia odkurzonej powierzchni. Firma Mitsubishi Heavy Industry opracowała system klimatyzacji pomieszczeń, bazujący na technikach zbiorów rozmytych. Uzyskano rewelacyjne obniżenie zużycia energii (ok. 24%) w porównaniu z konwencjonalnymi systemami. Firma Sanyo Fisher's wprowadziła na rynek kamerę video 8 mm stosując zbiory rozmyte do automatycznego ogniskowania i określania warunków świetlnych. Innym osiągnięciem w tej dziedzinie jest elektroniczny stabilizator obrazu opracowany we wspomnianej już firmie Matsushita. W połowie lat osiemdziesiątych pojawiły się również

pierwszego osiągnięcia opracowaniu struktury układu scalonego opartej na logice rozmytej - *Fuzzy Chip*. W 1989r. powstał pierwszy taki procesor konstruowany przez firmę Togai Infralogic (Irvine, USA).

Jednym z najbardziej spektakularnych osiągnięć ostatnich lat w dziedzinie sterowania rozmytego jest zaprojektowanie i wykonanie przez Sugeno modelu helikoptera bezzałogowego reagującego na rozmyte rozkazy przekazywane mu drogą radiową.

Ponadto do dnia dzisiejszego zastosowanie też sterowania rozmytego w wielu złożonych procesach przemysłu metalowego. Szczególnie interesujące są zastosowania metod zbiorów rozmytych w przemyśle samochodowym. Dotyczy to na przykład automatycznych przekładni biegów, układów zapobiegających blokowaniu hamulców, sterowania silnika i innych układów samochodu. Do przemysłowych zastosowań wyżej wymienionych metod należy zaliczyć sterowanie rozmyte oczyszczalnią wody i ścieków, sterowanie rozmyte różnego rodzaju procesami grzewczymi i wiele innych. Inne zastosowania systemów bazujących na teorii zbiorów rozmytych to: systemy wspomagania decyzji, rozmyte systemy ekspertowe, zastosowanie tych systemów w rozpoznawaniu mowy, obrazów, wspomaganiu diagnozy medycznej, ekonomii, lingwistyce i innych dziedzinach. Obecnie dziedzina logiki rozmytej jest jedną z najszybciej rozwijających się znajdujących rozwiązania coraz szerszej gamie oferowanych produktów.

### **Podstawowe pojęcia teorii zbiorów rozmytych**

Przed zaprezentowaniem podstawowych pojęć teorii zbiorów rozmytych, zostanie przedstawione czym zajmuje się ta teoria i dlaczego jej znaczenie jest takie ważne a następnie omówiona będzie jej geneza.

W życiu codziennym człowiek ma do czynienia z niedoskonałością informacji. W tym przypadku zostanie ona ograniczona do jednej formy jaką jest niepewność rozmyta. Informację w warunkach niepewności rozmytej można spotkać często w języku naturalnym, którą można krótko nazwać informacją lingwistyczną. Jako przykład może posłużyć różnica pomiędzy precyzyjną informacją numeryczną a nieprecyzyjną informacją lingwistyczną. Informacja numeryczna to na przykład stwierdzenie: wzrost chłopca wynosi 160 cm. Przykład informacji lingwistycznej to określenia wzrostu jak: „wysoki”, „bardzo niski” itp., które są zrozumiałe dla człowieka, ale trudno do wyrażenia w postaci liczbowej. Każde z wyżej wymienionych określeń lingwistycznych stanowi pojęcie nieostre, niejednoznaczne, rozmyte, co odpowiada angielskiemu terminowi *fuzzy*. Stąd pochodzi nazwa *fuzzy set*, w języku polskim zbiór rozmyty.

Należy tutaj zauważyć, że do czasu sformułowania teorii zbiorów rozmytych, komputery mogły prawie w całości wykorzystywać przetwarzanie informacji numerycznej, natomiast informację lingwistyczną w bardzo niewielkim stopniu. Stąd też m.in. wynikała potrzeba stworzenia aparatu matematycznego, który umożliwiłby przetwarzanie informacji lingwistycznej dla celów wspomagania decyzji. Takim narzędziem służącym do formalizowania przybliżonego wnioskowania w otoczeniu nieostrych terminów jest teoria zbiorów rozmytych. Stworzyła ona możliwość konstrukcji automatycznych systemów przybliżonego wnioskowania. Oprócz szerokiego zastosowania praktycznego w wielu dziedzinach życia codziennego i techniki, teoria zbiorów rozmytych wywiera głęboki wpływ na nauki podstawowe jak matematyka, fizyka i chemia, co świadczy o jej dużym stopniu uniwersalności.

Obserwując cały rozwój logiki, od czasów starożytnych do pierwszych dziesięcioleci obecnego wieku, można podstawową gałąź matematyki jaką jest logika, utożsamić z logiką dwuwartościową, mimo tego, że zagadnienia logiki wielowartościowej były już wówczas znane [11]. Związane to było z tendencją do ścisłego i precyzyjnie sformalizowanego opisu wszelkich obiektów jak i pewnej niechęci dla niejednoznaczności. Należy tu zaznaczyć, że logiki wielowartościowe zajmują naczelne miejsce wśród aparatów formalnych, które doprowadziły do sformułowania teorii zbiorów rozmytych. Sytuacja zmieniła się po opublikowaniu przez profesora Zadeha artykułu pt. „Fuzzy sets” w czasopiśmie „Information and Control” w roku 1965. W tym artykule zostały przedstawione fundamenty nowej logiki nieskończenie wartościowej, w której przyjmuje się wartości prawdy z przedziału  $[0,1]$ . Należy podkreślić, że sformułowanie podstaw logiki rozmytej wzbudziło wzrost zainteresowania logikami wielowartościowymi, które mogą posłużyć do sformalizowania przybliżonego rozumowania w terminach nieostrych i niejednoznacznych.

Podstawowym pojęciem klasycznej teorii zbiorów w sensie Cantora jest relacja przynależności elementu do zbioru. Ostra relacja przynależności oznacza, że element może do zbioru należeć lub nie należeć. Nie może być sytuacji pośredniej.

Aby scharakteryzować zbiór rozmyty można stosować różne ujęcia. Przyjmuje się, że w zbiorze rozmytym nie ma ostrej granicy między elementami, które do danego zbioru należą, a tymi, które należą do jego dopełnienia. W takim przypadku określa się stopnie przynależności poszczególnych elementów do zbioru. Innymi słowy, każdemu elementowi można przyporządkować inny stopień przynależności do określonego zbioru. Pozwala to scharakteryzować zbiór rozmyty poprzez funkcję przynależności, przyjmującą wartości z przedziału  $[0,1]$ . Pojęcie zbioru rozmytego wymaga też określe

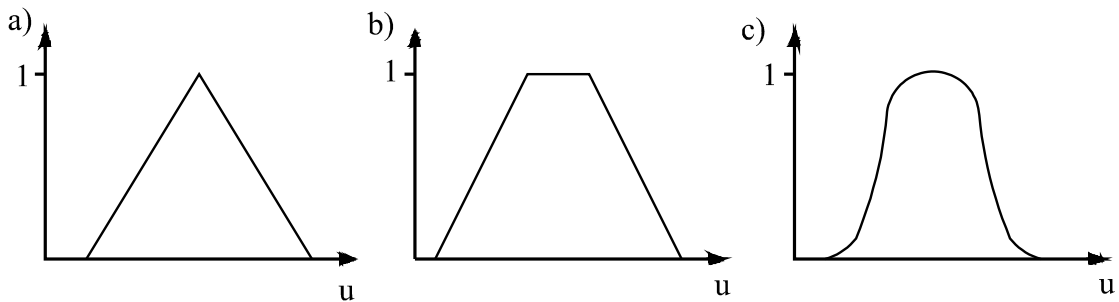
nia tzw. przestrzeni rozważań  $U$ , będącej zbiorem zwykłym, którą nazywamy też krótko uniwersum. A zatem, zbiorem rozmytym  $F$  można nazwać zbiór par uporządkowanych, w których w odpowiedniej kolejności występują: element i stopień przynależności elementu do zbioru. Można to zapisać:

$$F = \{(u, \mu_F(u)) \mid u \in U\} \quad (2.0)$$

gdzie  $\mu_F$  funkcja przynależności:

$$\mu_F : U \rightarrow [0,1] \quad (2.0)$$

Rysunek 2.1 przedstawia typowe przypadki funkcji przynależności.



Rys. 2.1 - Różne formy funkcji przynależności: a) trójkątna, b) trapezoidalna, c) w kształcie dzwonu

Zbiór rozmyty nie jest rozłączny ze swoim dopełnieniem (przykładowo człowiek może odczuwać zimno i ciepło w różnym stopniu).

Poniżej zostaną scharakteryzowane najważniejsze operacje na zbiorach rozmytych. Wśród ich bardzo dużej liczby można wyróżnić operacje mnogościowe i algebraiczne.

Z operacji mnogościowych należy wymienić następujące operacje:

- jednoargumentowa operacja dopełnienia zbioru rozmytego:



- dwuargumentowa operacja sumy mnogościowej dwóch zbiorów:

$$A, B, C \in X$$

$$C = A \cup B$$

$$\forall x \in X:$$

$$\mu_C(x) = \max[\mu_A(x), \mu_B(x)]$$

- przecięcie:

$$A, B, D \in X$$

$$D = A \cap B$$

$$\forall x \in X:$$

$$\mu_D(x) = \min[\mu_A(x), \mu_B(x)]$$

Powyższe definicje nadają zbiorowi podzbiorów rozmytych zbioru  $X$  strukturę zbliżoną do algebry Boole'a, jednak odmienną, ponieważ przestają obowiązywać fundamentalne zasady teorii klasycznych:

- zasada niesprzeczności, bo:  $A \cap \bar{A} \neq \emptyset$
- zasada wyłączonego środka, ponieważ:  $A \cup \bar{A} \neq \bar{\emptyset}$

Oprócz operacji mnogościowych należy wymienić jeszcze całą szereg operacji algebraicznych na zbiorach rozmytych. Do nich należą operacje mnożenia i potęgowania funkcji przynależności rozważanych zbiorów rozmytych. Do operacji potęgowania należą: operacja koncentracji zaostrzającą zbiór rozmyty, operacja rozcieńczania spłaszczająca zbiór rozmyty, a następnie intensyfikacja (zwiększenie) kontrastu zbioru rozmytego, zwiększającą stopnie przynależności mniejsze od 1/2 i operacja jej przeciwna.

W dalszej kolejności należy wymienić inne operacje algebraiczne na zbiorach rozmytych. Do nich należą: suma algebraiczna, suma ograniczona, różnica symetryczna, różnica ograniczona i inne. Wszystkie te operacje definiujemy za pomocą odpowiednich operacji algebraicznych na funkcjach przynależności.

Istnieje też wiele innych operacji. Interesujące są niektóre własności ważniejszych operacji na zbiorach rozmytych:

- przemienność:

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

- idempotentność:

$$A \cup A = A$$

$$A \cap A = A$$

- łączność:

$$A \cup (B \cap C) = (A \cup B) \cap C = A \cup B \cap C$$

$$A \cap (B \cup C) = (A \cap B) \cup C = A \cap B \cup C$$

- rozdzielność:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

- prawa de Morgana:

$$\overline{(A \cup B)} = \bar{A} \cap \bar{B}$$

$$\overline{(A \cap B)} = \bar{A} \cup \bar{B}$$

Należy też podkreślić problem adekwatności definicji operacji na zbiorach rozmytych. Suma i przekrój zbiorów rozmytych wydają się być intuicyjnie dość dobrze uzasadnione, a pozatym posiadają ważną zaletę – są proste analitycznie i obliczeniowo.

Poniżej zostaną przedstawione kolejne ważne pojęcia teorii zbiorów rozmytych decydujące o ich praktycznym znaczeniu, czyli o zastosowaniu do wnioskowania w warunkach nieprecyzyjnej informacji.

Pierwszym jest pojęcie zmiennej lingwistycznej. Jej definicja jest dość złożona i wymaga pewnych formalizmów matematycznych. Można ją jednak przedstawić w sposób opisowy. Zmienna lingwistyczna to wielkość, która może przyjmować lingwistyczne wartości (które wartości to słowa lub zdania w języku naturalnym lub sztucznym). Przykładem takiej zmiennej może być wielkość o nazwie *szybkość*. Zmienna ta może przyjmować różne wartości lingwistyczne (zwaneteż *termami*) takie jak: *bardzo mała, mała, średnią, dużą, bardzo dużą* itp. Termy mogą być już opisane numerycznie za pomocą odpowiednich funkcji przynależności.

Przetwarzanie danych przez człowieka oparte jest na zmiennych lingwistycznych. Umysł ludzki ma nie tylko zadziwiającą zdolność przetwarzania takich zmiennych, ale również rozumowania na ich podstawie. Aby to wykorzystać należy jeszcze określić zależności między zmiennymi lingwistycznymi. Mogą do tego służyć rozmyte zdania warunkowe, które opisują zależności przyczynowo skutkowe. Zdania te nazywane są też rozmytymi regułami wnioskowania. Najczęściej są one typu: *jeśli A to B*. A i B to twierdzenia rozmyte (np. *e jest duża*). Można je tworzyć na podstawie doświadczenia eksperta, bądź też na podstawie wyników numerycznych eksperymentów. Jeżeli dana sytuacja opisana jest zbiorem reguł, to stanowią one wtedy bazę reguł. Baza taka stanowi punkt wyjściowy do problemu wnioskowania w warunkach nieprecyzyjnej informacji lub wnioskowania przybliżonego. Wnioskowanie takie jest realizowane w sposób automatyczny w systemach wnioskowania rozmytego.

### **Wnioskowanie przybliżone**

Wnioskowanie to stanowi podstawę do budowy tzw. rozmytych systemów wnioskowania, które mogą być już zaprogramowane np. w komputerze. Wnioskowanie rozmyte można wstępnie scharakteryzować jako procedurę, która pozwala otrzymywać

wnioski z rozmytych zdań warunkowych oraz znanych rozmytych lub nierozmytych faktów.

Kluczową rolę w algorytmach wnioskowania pełni złożeniowa reguła wnioskowania. Została ona zaproponowana przez Zadeh'a w 1968r. [12] i jak dotąd stanowi formalną podstawę wnioskowania rozmytego. Jest daleko idącym uogólnieniem prostej koncepcji funkcji; stosuje relację rozmytą do wyrażenia związku między dwoma twierdzeniami rozmytymi.

Dana jest funkcja jednej zmiennej  $y = f(x)$ . Wykres tej funkcji obrazuje relację między zmienną  $x$  a zmienną  $y$ . Jeżeli dany jest fakt  $x = a$ , to fakt ten w połączeniu z zadaną funkcją pozwalającą się wyrazić regułą:

*Jeśli zmienna  $x$  przyjmuje wartość  $a$ , to zmienna  $y$  przyjmuje wartość  $b = f(a)$*   
pozwolana otrzymanie wniosku  $y = b = f(a)$ .

Podstawowa reguła wnioskowania niezawodnego to reguła *modus ponens*. Zgodnie z tą regułą można wnioskować o prawdziwości konkluzji  $B$  na podstawie prawdziwości przesłanki  $A$  i implikacji z  $A$  wynika  $B$ . Na przykład jeżeli  $A$  to „pomidor jest czerwony” a  $B$  to „pomidor jest dojrzały” to wtedy z prawdziwości przesłanki  $A$  wynika także prawdziwość wniosku  $B$ . Jednakże w większości procesów wnioskowania podana powyżej reguła *modus ponens* jest stosowana w sposób przybliżony.

Podstawowa struktura systemów wnioskowania rozmytego składa się z trzech koncepcyjnie różnych komponentów:

- baza reguł składająca się ze zbiorów reguł rozmytych,
- baza danych - opisy funkcji przynależności zbiorów rozmytych w bazie reguł,
- mechanizm wnioskowania - wykonuje procedury wnioskowania i bierze pod uwagę reguły i znane fakty.

Baza reguł i baza danych składają się na bazę wiedzy regulatora rozmytego omówionego w dalszej części. W zastosowaniach technicznych wielkościami wejściowymi i wyjściowymi są zwykle wartości mierzalne. Aby operować takimi wielkościami należy wbudować na wejściu i wyjściu systemu wnioskowania rozmytego dwa dodatkowe bloki zwane odpowiednio blokiem *rozmywania* i blokiem *wyostrzania*. Układ bloku rozmywania przekształca bieżącą wartość ostrą zmiennej stanu procesu (pojedynczą wartość sygnału wejściowego) w zbiór rozmyty, aby uczynić ją kompatybilną z odwzorowaniem zbioru rozmytego zmiennej stanu procesu w poprzedniku reguły. Zadaniem bloku wyostrzania natomiast, jest operacja odwrotna, czyli odwrócenie zmodyfikowanej wartości zmiennej sterującej w pojedynczą wartość punktową.

W szerokiej grupie systemów wnioskowania rozmytego rozróżniamy cztery podstawowe grupy. Różnice między nimi leżą w interpretacji reguł rozmytych jak również w doborze zbioru reprezentujących konkluzje. Dwie pierwsze grupy to tzw. systemy konstruktywne oraz systemy destruktywne przy czym podziału dokonuje się zgodnie z dwiema możliwymi interpretacjami reguł rozmytych. Pierwszą grupą są systemy, w których reguły rozmyte są interpretowane za pomocą spójnika „i”. Są to systemy typu Mamdaniego zwane też konstruktywnymi ze względu na sposób tworzenia wniosku końcowego w wnioskowaniu przybliżonym (*implikacja Mamdaniego*:  $p \rightarrow q \equiv p \wedge q$ ). Drugą grupą są systemy, w których reguły są interpretowane za pomocą implikacji rozmytych. Pozostałe dwie grupy to systemy typu Takagi-Sugeno-Kanga oraz rzadziej używane systemy wnioskowania typu Tsukamoto. Jeżeli chodzi o systemy typu Takagi-Sugeno-Kanga to są one najbardziej popularne w modelowaniu układów na podstawie eksperymentalnych danych dyskretnych, z uwagi na małą czasochłonność obliczeń i możliwość uniknięcia trudnej operacji wyostrozania. Informacje na ten temat można znaleźć w publikacji [13].



## Zagadnienie sterowania rozmytego

### ***Sterowanie rozmyte a sterowanie konwencjonalne***

Ogólne podstawy teorii sterowania są związane z matematyczną teorią sterowania, rozwijaną intensywnie po II wojnie światowej. Podstawowe zasady sterowania ze sprzężeniem zwrotnym, w postaci doświadczeń, intuicji i umiejętności praktycznych, były znane i stosowane od wieków. Czasy, po rewolucyjnym odkryciu przez Watt sterowania ze sprzężeniem zwrotnym, były zdominowane przez techniki matematyczne. Pomijano fakt, że ludzie potrafią zadowalająco sterować różnymi procesami technologicznymi.

Podstawowa idea sterowania automatycznego polega na tym, że element sterujący (regulator) powinien w każdej chwili  $t$  wytwarzać na podstawie sygnałów pomiarowych sygnał sterujący  $u(t)$  - odpowiedni dla celów sterowania. Celem tym może być stabilizacja, śledzenie zadanej trajektorii itd. Jednym z istotnych wymagań regulacji jest zapewnienie dużej odporności algorytmu regulacji na zmiany parametrów procesu (*robust control*).

Cechą wspólną sterowania konwencjonalnego jest fakt, że algorytm sterowania jest wyrażony analitycznie przez równania algebraiczne, różnicowe, różniczkowe itp. Mówiąc ogólnie, synteza takiego algorytmu sterowania wymaga sformalizowanego opisu analitycznego układu sterowania w postaci modelu matematycznego. Praktyka pokazuje, że większość regulatorów przemysłowych działa zgodnie z algorytmem PID (proporcjonalno-całkująco-różniczkującym), często wystarczającym pomimo swojej prostoty. Znane są jednak przykłady, w których nie wolno stosować klasycznego liniowego sprzężenia zwrotnego, a stosowanie bardziej zaawansowanych metod syntezy algorytmów nieliniowych na ogół wykracza poza zakres umiejętności inżyniera. Z drugiej strony istnieje wiele przykładów potwierdzających wydajność algorytmów sterowania wykorzystujących reakcje ludzkie. Teoria zbiorów rozmytych proponuje prawidłowe narzędzia operowania na algorytmach opisanych heurystycznie lub lingwistycznie. Rozwijający teorię Zadeha Mamdani i Assilian [14] zademonstrowali, że reguły logiczne z nieokreślonymi predykatami mogą być wykorzystane do wprowadzenia wniosków z niejasno sformułowanych danych i doszli do konkluzji, że algorytmy sterowania lingwistycznego mogą być wykorzystane do sterowania złożonymi systemami, ludzkimi i technicznymi. Pomysł algorytmów sterowania lingwistycznego był błyskotliwym uogólnieniem ludzkich doświadczeń w stosowaniu reguł lingwistycznych z nieokreślono-

nymi predykatami w celu sformułowania sterowania działaniem. Formalizacja tego pojęcia przez teorię zbiorów rozmytych wprowadziła je do dziedziny sterowania przy użyciu aparatu wnioskowania przybliżonego.

Główną zasadą budowy regulatora rozmytego mówi, że algorytm sterowania jest algorytmem na bazie wiedzy, opisanym przez metody logiki rozmytej. Układy sterowania w logice rozmytej są rodzajem systemu ekspertowego na bazie wiedzy, który zawiera algorytm sterowania w postaci prostej bazy reguł. Wiedza zakodowana w bazie reguł jest wyprowadzana na podstawie ludzkiego doświadczenia i intuicji oraz na podstawie teoretycznego i praktycznego zrozumienia dynamiki obiektu sterowanego. Tym, co różni sterowanie rozmyte pod względem pojęciowym od sterowania konwencjonalnego, jest brak opisu analitycznego. Mechanizm wnioskowania przybliżonego przekształca wiedzę wpisaną w bazę reguł, w nierozmyty algorytm sterowania.

### **Regulator rozmyty FLC - pojęcia podstawowe**

Pod nazwą regulator rozmyty (*Fuzzy Logic Controller* - FLC) rozumie się prawo sterowania, które jest opisane przez system bazy wiedzy zawierający reguły JEŻELI - TO przy nieokreślonych predykatami mechanizmu sterowania o logice rozmytej.

Baza reguł jest główną częścią FLC w postaci podobnej do konwencjonalnego prawa sterowania:

$$u(k) = F(e(k), e(k-1), \dots, e(k-v), u(k-1), u(k-2), \dots, u(k-v)) \quad (3.0)$$

przy czym funkcja  $F$ , prawo sterowania, jest opisane przez bazę reguł a  $v$  określa rząd regulatora. Jednakże nie znaczy to, że FLC jest pewnym rodzajem funkcji przejścia czy równania różniczkowego. Naturalną podstawą bazy reguł regulatora FLC dyktuje ograniczone użycie znanych z przeszłości wartości uchybu (błędu)  $e$  i sterowania  $u$  ponieważ jest nierozsądne oczekiwać, że  $e(k-3)$ ,  $e(k-4)$ ,  $u(k-3)$ ,  $u(k-4)$ , itd. będą dobrze określone w sposób lingwistyczny.

Typowy FLC opisuje zależność między zmianą sterowania  $\Delta u(k) = u(k) - u(k-1)$  z jednej strony i uchybem  $e(k) = w(k) - y(k)$  oraz jego zmianą  $\Delta e(k) = e(k) - e(k-1)$  z drugiej strony. Takie prawo sterowania można sformalizować jako:

$$\Delta u(k) = F(e(k), \Delta e(k)) \quad (3.0)$$

Rzeczywiście wyjście regulatora  $u(k)$  otrzymujemy z przeszłej wartości sterowania  $u(k-1)$  i jej aktualizacji przez  $\Delta u(k)$ :

$$u(k) = u(k-1) + \Delta u(k) \quad (3.0)$$

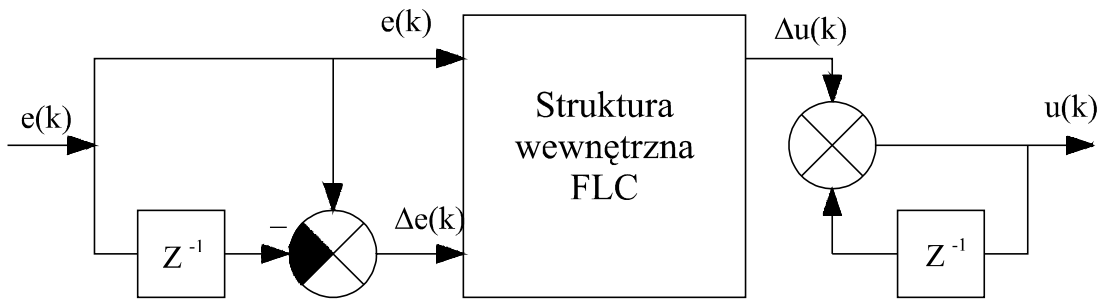
FLC tego typu zaproponowali po raz pierwszy Mamdani i Assilian w 1975 roku [14] i nazywany jest FLC typu Mamdaniego.

Prototypowa baza prostego FLC realizującego prawo (3.2) jest wymieniona w tabeli 3.1.

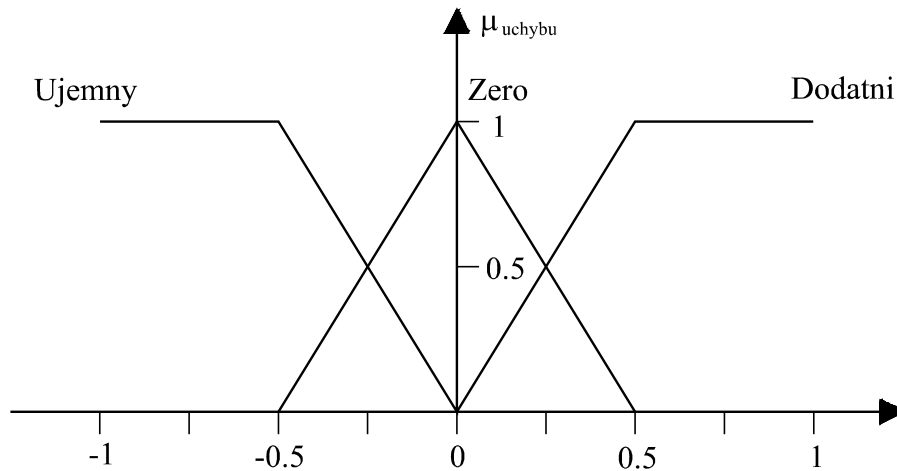
Tabela 3.1 - Przykład bazy reguł prostego FLC

- 
- |    |   |
|----|---|
| 1. | <b>JEŻELI</b> uchyb $e(k)$ jest dodatni <b>I</b> zmiana uchybu $\Delta e(k)$ jest w przybliżeniu zero<br><b>TO</b> zmiana sterowania $\Delta u(k)$ jest dodatnia<br><b>TAKŻE</b>                        |
| 2. | <b>JEŻELI</b> uchyb $e(k)$ jest ujemny <b>I</b> zmiana uchybu $\Delta e(k)$ jest w przybliżeniu zero<br><b>TO</b> zmiana sterowania $\Delta u(k)$ jest ujemna<br><b>TAKŻE</b>                           |
| 3. | <b>JEŻELI</b> uchyb $e(k)$ jest w przybliżeniu zero <b>I</b> zmiana uchybu $\Delta e(k)$ jest w przybliżeniu zero<br><b>TO</b> zmiana sterowania $\Delta u(k)$ jest w przybliżeniu zero<br><b>TAKŻE</b> |
| 4. | <b>JEŻELI</b> uchyb $e(k)$ jest w przybliżeniu zero <b>I</b> zmiana uchybu $\Delta e(k)$ jest dodatnia<br><b>TO</b> zmiana sterowania $\Delta u(k)$ jest dodatnia<br><b>TAKŻE</b>                       |
| 5. | <b>JEŻELI</b> uchyb $e(k)$ jest w przybliżeniu zero <b>I</b> zmiana uchybu $\Delta e(k)$ jest ujemna<br><b>TO</b> zmiana sterowania $\Delta u(k)$ jest ujemna   |
- 

Każda reguła FLC składa się z części JEŻELI, nazwanej poprzednikiem oraz części TO, zwanej następnikiem. Poprzednik reguły zawiera zbiór warunków; następnik zawiera wniosek. Każda reguła działa w następujący sposób: jeżeli są spełnione warunki poprzednika, to wykonuje się wniosek następnika. Można spojrzeć na FLC jak na system, który ma na wejściach zmienne zawarte w poprzednikach reguł, a na wyjściu zmienną zawartą w następniku. Można przyjąć uchyb  $e(k)$  oraz jego zmianę  $\Delta e(k)$  jako wejścia, a zmianę sterowania  $\Delta u(k)$  jako wyjście FLC opisanego równaniem (3.2). Można zauważyć, że wejścia i wyjścia FLC są w istocie poprzednikami następnikiem zmiennych tych reguł, które są zawarte w jego bazie reguł. Są one związane ze strukturą wewnętrzną FLC i w zasadzie są takie same jak wejście  $e$  i wyjście  $u$  regulatora (rysunek 3.1).



Rys. 3.1 Wewnętrzna struktura FLC w układzie sterowania



Rys. 3.2. Funkcje przynależności zbiorów rozmytych odniesienia reprezentujące zmiennę lingwistyczną: dodatni, w przybliżeniu zero, ujemny. Te trzy zmiennę lingwistyczną tworzą zbiór terminu uchyb. Symbol  $\mu_{uchybu}$  oznacza funkcję przynależności zbioru rozmytego odniesienia związaną z dowolnymi zmiennymi lingwistycznymi, które należą do zbioru terminu uchyb. Wszystkie trzy zbiory rozmyte są określone na tej samej przestrzeni  $[-1, 1]$

Rodzina reguł w tabeli 3.1 działa na etykietach lingwistycznych wejść i wyjść FLC. Zmiennę lingwistyczną reprezentowaną przez zbiory rozmyte, zwane *zbiórami rozmytymi odniesienia*, zbiór etykiet lingwistycznych jednej zmiennej tworzy zbiór terminu. Zbiory rozmyte odniesienia określone są na przestrzeniach, które są wyznaczone przez zakres działania odpowiedniej zmiennej. Na rysunku 3.2 pokazano funkcje przynależności trzech zbiorów rozmytych odniesienia reprezentujących etykiety lingwistyczne uchyb: dodatni, w przybliżeniu zero, ujemny. Te etykiety lingwistyczne tworzą zbiór terminu *uchyb e*, zmiennej *e*. Zakres działania uchybu *e* jest z założenia w przedziale  $[-1, 1]$ .

Analogicznie definiuje się zbiory terminu innych zmiennych FLC jako zmianę uchybu  $\Delta e$  i zmianę sterowania  $\Delta u$ .

Poniżej skomentowane jest znaczenie poszczególnych reguł zawartych w tabeli 3.1. Dwie pierwsze reguły zapewniają szybkie dojście przez wyjście układu  $y$  do wartości zadanej. Dodanie wartości uchybów w regule 1 odnosi się do przypadków, dla których  $e(k) > Q$  to jest  $w(k) > y(k)$ . Zgodnie z przyjętą prostą dynamiką układu (3.3), zwiększenie zmiennej sterowania  $u$  daje większą wartość wyjścia  $y$  układu. Rozsądne działanie sterujące polega zatem na zwiększeniu sterowania  $u$ , co z kolei daje dodatnią zmianę  $\Delta u$ . Reguła 2 działa podobnie dla ujemnych wahań względem wartości zadanej. Te dwie reguły wyznaczają zdolność układu sterowania do reakcji. Reguła 4 opisuje strategię sterowania, gdy wyjście układu  $y$  jest bliskie wartości zadanej  $w$ , to jest uchyb jest w przybliżeniu zero. Dodatnie wartości zmian uchybu  $\Delta e$  wskazują, że gradient uchybu względem czasu jest dodatni, tzn.  $e(k) > e(k-1)$  i odpowiednio

$$w(k) - y(k) > w(k-1) - y(k-1)$$

oraz małych zmian wartości zadanej pomiędzy chwilami próbkowania ( $w(k) = w(k-1)$ ) mamy  $y(k) < y(k-1)$ . Dlatego reguła ta odnosi się do przypadku, gdy bieżące wyjście  $y$  jest bliskie wartości zadanej, ale maleje w funkcji czasu. Inaczej mówiąc, gradient wyjścia  $y$  względem czasu jest ujemny. Rozsądne działanie sterujące polega na zwiększeniu  $u$  (i zwiększeniu  $y$ ), z czego wynika dodatnia zmiana sterowania  $\Delta u$ , określona przez tę regułę. W przypadku przeciwnym, gdy zmiana uchybu  $\Delta e$  jest ujemna, reguła 5 ustala ujemną zmianę sterowania  $\Delta u$ . Oczywiście reguły 4 i 5 zapewniają zredukowanie przekroczenia. Wreszcie reguła 3 opisuje sterowanie w stanie ustalonym. Mówi ona, że jeżeli wyjście jest bliskie wartości zadanej i różnica uchybu jest bliska zera, to sterowanie nie powinno się zmieniać.

To proste sformułowanie algorytmu sterowania umożliwia zrealizowanie strategii heurystycznych, określonych przez zdania opisane lingwistycznie. Algorytm sterowania rozmytego odzwierciedla mechanizm sterowania realizowany przez ludzi, bez użycia żadnej sformalizowanej wiedzy o sterowanym obiekcie w postaci modeli matematycznych i bez opisu analitycznego algorytmu sterowania. Nie można jednak oczekiwać, że powyższy algorytm może rozwiązać bardziej złożone zadania sterowania. Oczywiście do skomplikowanych zagadnień sterowania potrzebne jest więcej wiedzy, a więc bardziej szczegółowa baza reguł, która podaje bardziej kompletny opis strategii sterowania.

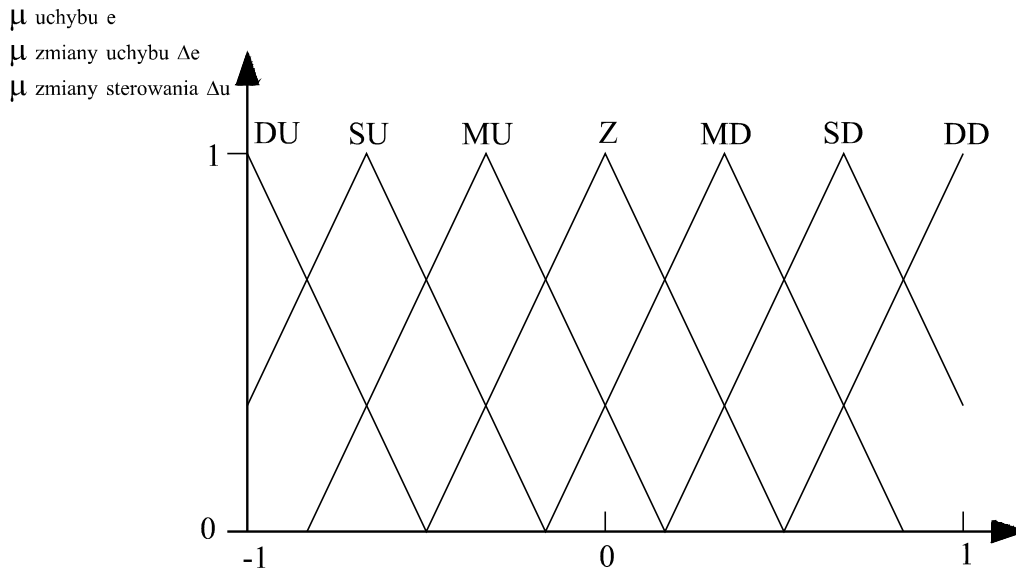
Prostym źródłem wyprowadzenia strategii sterowania lingwistycznego jest ludzkie doświadczenie i rozumowanie, które w istocie zawierają model układu sterowania w postaci raczej pośredniej. Przykładowo baza reguł z tabeli 3.1 może zostać znacząco

poprawiona, jeżeli uwzględnimy bardziej szczegółowe zadania logiczne. Baza reguł FLC z tabeli 3.2 jest rozszerzeniem bazy reguł z tabeli 3.1.

Zbiory terminu zmiennych wejście i wyjście danego FLC, uchybu  $e$ , zmiany uchybu  $\Delta e$  i zmiany sterowania  $\Delta u$  zawierają etykiety lingwistyczne *duży dodatni* (DD), *średni dodatni* (SD), *mały dodatni* (MD), *zero* (Z), *mały ujemny* (MU), *średni ujemny* (SU) i *duży ujemny* (DU). Funkcje przynależności odpowiednich zbiorów rozmytych odniesienia wykreślono na rysunku 3.3.

Tabela 3.2. Poprawiona baza reguł FLC do sterowania obiektów aproksymowanych przez układ pierwszego rzędu z czasem rzeczywistym

<b>JEŻELI</b>	$e(k)$	jest	DD	<b>I</b>	$\Delta e(k)$	jest	Z	<b>TO</b>	$\Delta u(k)$	jest	DD	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	SD	<b>I</b>	$\Delta e(k)$	jest	Z	<b>TO</b>	$\Delta u(k)$	jest	SD	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	MD	<b>I</b>	$\Delta e(k)$	jest	Z	<b>TO</b>	$\Delta u(k)$	jest	MD	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	MU	<b>I</b>	$\Delta e(k)$	jest	Z	<b>TO</b>	$\Delta u(k)$	jest	MU	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	SU	<b>I</b>	$\Delta e(k)$	jest	Z	<b>TO</b>	$\Delta u(k)$	jest	SU	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	DU	<b>I</b>	$\Delta e(k)$	jest	Z	<b>TO</b>	$\Delta u(k)$	jest	DU	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	Z	<b>I</b>	$\Delta e(k)$	jest	Z	<b>TO</b>	$\Delta u(k)$	jest	Z	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	Z	<b>I</b>	$\Delta e(k)$	jest	DD	<b>TO</b>	$\Delta u(k)$	jest	DD	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	Z	<b>I</b>	$\Delta e(k)$	jest	SD	<b>TO</b>	$\Delta u(k)$	jest	SD	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	Z	<b>I</b>	$\Delta e(k)$	jest	MD	<b>TO</b>	$\Delta u(k)$	jest	MD	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	Z	<b>I</b>	$\Delta e(k)$	jest	MU	<b>TO</b>	$\Delta u(k)$	jest	MU	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	Z	<b>I</b>	$\Delta e(k)$	jest	SU	<b>TO</b>	$\Delta u(k)$	jest	SU	<b>TAKŻE</b>
<b>JEŻELI</b>	$e(k)$	jest	Z	<b>I</b>	$\Delta e(k)$	jest	DU	<b>TO</b>	$\Delta u(k)$	jest	DU	



Rys. 3.3. Funkcje przynależności zbiorów rozmytych odniesienia związane ze zbiorami terminu uchybu  $e$ , zmiany uchybu  $\Delta e$  i zmiany sterowania  $\Delta u$  stosowanew bazie reguł w tabeli 3.2. Każdy zbiór terminu zawiera siedemetykiel lingwistycznych duży dodatni (DD), średni dodatni (SD), mały dodatni (MD), zero (Z), mały ujemny (MU), średni ujemny (SU) i duży ujemny (DU). Wszystkie zbiory rozmyte odniesienia są określone na przestrzeni  $[-1, 1]$ .

Postać FLC, która znacznie różni się od sformułowania stosowanego w klasycznych algorytmach sterowania, nasuwa pytanie o to, jak realizować FLC w praktyce. Inaczej mówiąc jak można obliczyć wyjście bazy reguł przyjmując, że dana jest wartość zmiennych wejściowych.

### **Wnioskowanie FLC**

Jak pokazano, FLC można uważać za złożoną relację funkcyjną między odpowiednimi zmiennymi. W dodatku postać użyta do reprezentowania tej relacji wymaga zastosowania reguł wykorzystujących nieokreślone predykaty. Zbiór tych reguł jest nazywany bazą wiedzy związanej z FLC. Terminem *wnioskowanie* będzie oznaczony proces, w którym na podstawie danych wartości wejści i zmiennych poprzedników (używanych w koniunkcji z bazą wiedzy), otrzymujemy wartość wyjściową.

Istnieją dwa rodzaje wnioskowania:

- wnioskowanie oparte na złożeniu - relacje rozmyte reprezentujące znaczenie każdej indywidualnej reguły zostają połączone w jedną relację rozmytą opisującą znaczenie całego zbioru reguł; potem odbywa się wnioskowanie lub odpalenie reguły przez operację złożenia między rozmytym wejściem a relacją rozmytą reprezentującą znaczenie całego zbioru reguł; wynikiem złożenia jest zbiór rozmyty opisujący wartość rozmytą ogólnego wyjścia sterującego.

- wnioskowanie oparte na pojedynczej regule - pojedyncza reguła jest odpalana.  
Formalnie bazę reguły można reprezentować w następującej formie:

**JEŻELI**  $U_1$  jest  $B_{11}$  **I**  $U_2$  jest  $B_{12}$  **TO**  $V$  jest  $D_1$

**TAKŻE**

(3.0)

...

**TAKŻE**

**JEŻELI**  $U_1$  jest  $B_{m1}$  **I**  $U_2$  jest  $B_{m2}$  **TO**  $V$  jest  $D_m$

W formacie (3.4)  $U_1$ ,  $U_2$  i  $V$  to zmienne,  $U_1$ ,  $U_2$  to zmienne wejściowe a  $V$  zmienna wyjściowa.  $B_{11}$ ,  $B_{12}$  i  $D_1$  są zmiennymi lingwistycznymi (etykietami) reprezentowanymi jako podzbiory rozmyte odpowiednich przestrzeni  $X_1$ ,  $X_2$  i  $Y$ . Funkcje przynależności tych wartości lingwistycznych zostaną oznaczone  $B_{11}(x_1)$ ,  $B_{12}(x_2)$  i  $D_1(y)$ . Jeżeli zastosujemy ten format do opisu reguł z tabeli 3.1 to  $U_1$  oznacza uchyb,  $U_2$  jest zmienną uchybu,  $V$  etykietą sterowania,  $B_{11}$  jest etykietą dodatni,  $B_{32}$  etykietą w przybliżeniu zero itd.

Jeżeli wejścia FLC mają wartości  $U_1 = x_1^*$  i  $U_2 = x_2^*$ , to pojawia się problem wyznaczenia właściwych wartości zmiennej wyjściowej  $V$ . Jest to problem zwany wnioskowaniem z FLC. Na ogół stosuje się następującą procedurę otrzymywania wyjścia rozmytego takiej bazy wiedzy:

1. Znalezienie poziomu zapłonu każdej reguły.
2. Znalezienie wyjścia każdej reguły.
3. Agregacja poszczególnych wyjść reguły w celu otrzymania całkowitego wyjścia systemu.

Najpierw należy rozważyć wyznaczenie poziomu zapłonu poszczególnych reguł. Poziom zapłonu reguły jest wyznaczony przez spełnienie każdego ze składowych poprzedników. Poziom dopasowania między etykietą lingwistyczną  $B_{i1}$  i wartością wejścia  $x_1^*$  jest wyznaczony jako stopień przynależności  $x_1^*$  do zbioru rozmytego reprezentującego  $B_{i1}$ . Zatem  $B_{i1}(x_1^*)$  jest to poziomy dopasowania pierwszego poprzednika. Analogicznie można znaleźć  $B_{i2}(x_2^*)$ . Aby otrzymać poziomy zapłonu reguły należy połączyć te dwie wartości.

Biorąc pod uwagę iloczyn logiczny między wejściami  $U_1$  i  $U_2$  w części poprzednika każdej reguły:

**JEŻELI**  $U_1$  jest  $B_{i1}$  **I**  $U_2$  jest  $B_{i2}$  **TO**  $V$  jest  $D_1$

łączy się te dwa poziomy dopasowania za pomocą agregacji typu iloczynowego. W szczególności stosujemy agregację typu  $\text{Min}(\wedge)$ . Dzięki temu otrzymujemy:



$$\tau_i = Bi1(x1^*) \wedge Bi2(x2^*) \quad (3.0)$$

przy czym  $\tau_i$  jest nazywane *poziomym zapłonem* (PZ)  $i$ -tej reguły względem wartości wejściowej  $U_1 = x_1^*$  i  $U_2 = x_2^*$ .

Poziom zapłon  $\tau_i$  przyjmuje wartość z przedziału jednostkowego i charakteryzuje przynależność (odpowiedniość) części poprzednikowej  $i$ -tej reguły.

W przypadku gdy zmienne wejściowe FLC przyjmują wartości rozmyte, to jest  $U_1 = A_1$  i  $U_2 = A_2$ , przy czym  $A_1$  i  $A_2$  są to podzbiory rozmytych zbiorów  $X_1$  i  $X_2$ , to poziom dopasowania pomiędzy wejściową wartością rozmytą  $A_1$  i etykietą lingwistyczną  $B_{i1}$  otrzymujemy z możliwości warunkowej:

$$Poss(B_{i1}|A_1) = \underset{x_1}{Max}[B_{i1}(x_2) \wedge A_1(x_1)]$$

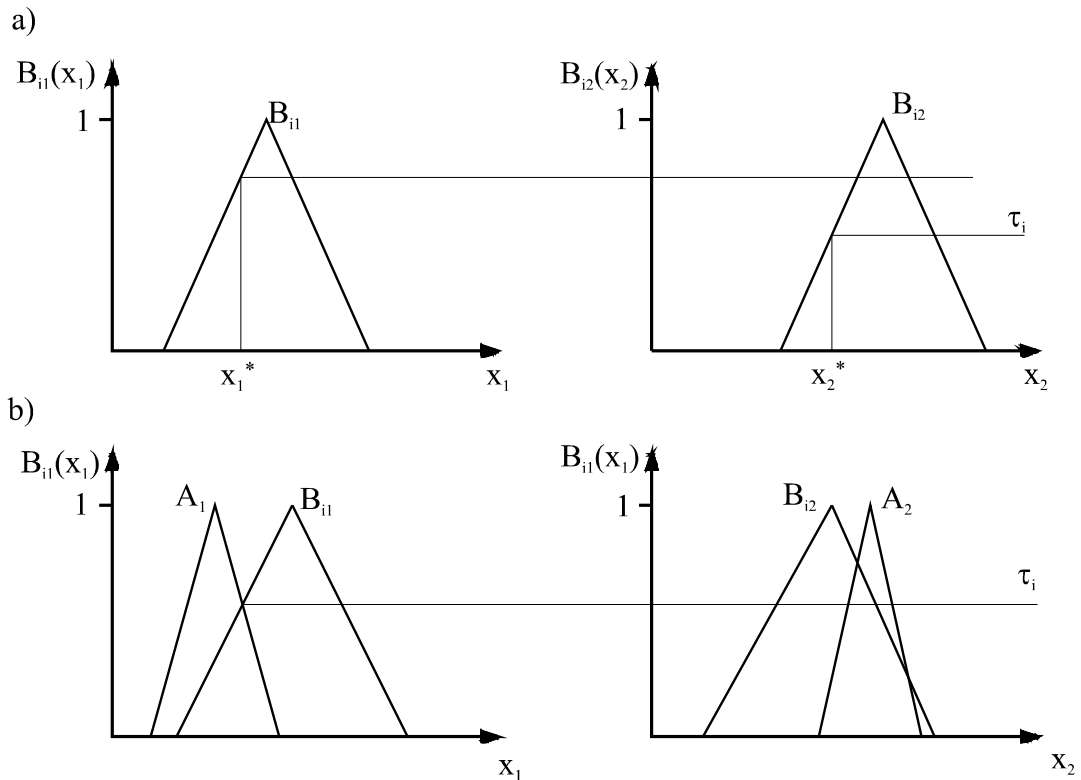
a nie ze stopnia przynależności. Podobnie dla drugiej zmiennej wyjściowej:

$$Poss(B_{i2}|A_2) = \underset{x_2}{Max}[B_{i2}(x_2) \wedge A_2(x_2)]$$

Poziom zapłon  $i$ -tej reguły jest w tym przypadku

$$\tau_i = Poss(B_{i1}|A_1) \wedge Poss(B_{i2}|A_2) \quad (3.0)$$

Oba wyrażenia (3.5) i (3.6) na obliczanie poziomu zapłonu dla nierozmytych i rozmytych wartości wejściowych pokazano na rysunku 3.4.



Rys. 3.4. Obliczanie poziomu zapłonu  $i$ -tej reguły: a) zmienne wejściowe FLC przyjmują wartości nierozmyte  $U_1=x_1^*$  i  $U_2=x_2^*$ ; stosuje się wyrażenie 3.5; b) zmienne wejściowe FLC przyjmują wartości rozmyte  $U_1=A_1$  i  $U_2=A_2$ ; stosuje się wyrażenie 3.6.

Następnym krokiem procesu jest wyznaczenie wyjścia pojedynczej reguły, które oznaczone będzie  $F_i$ . Poziom zapłonu reguły  $\tau_i$  oddziałuje z jej następnikiem  $D_i$ , aby dać wyjście reguły  $F_i$  w postaci podzbioru rozmytego nad przestrzenią wyjściową  $Y$ . Sformułowanie używane do wyznaczenia, jak oddziałują ze sobą  $\tau_i$  oraz zbiór rozmyty  $D_i$ , aby otrzymać wyjście reguły, nazywane jest *implikacją rozmytą*. W sterowaniu o logice rozmytej najpopularniejsze jest stosowanie metody wnioskowania o wyjściu reguły, zwanego metodą Mamdaniego. W metodzie Mamdaniego wyjściowy zbiór rozmyty  $F_i$  otrzymuje się jako iloczyn logiczny (obcinanie) poziomu zapłonu  $\tau_i$  oraz zbioru rozmytego następnika  $D_i$ :

$$F_i(y) = \tau_i \wedge D_i(y) \quad (3.0)$$

Trzecim krokiem procesu jest agregacja poszczególnych wyjść reguły, aby otrzymać całkowite wyjście systemu  $F$ , które również jest podzbiorem rozmytym nad  $Y$ .

Poszczególne wyjścia reguły są agregowane za pomocą spójnika alternatywnego, który przekształca się w agregację typu sumy logicznej wyjść  $F_i$ . Zatem wyjście rozmyte  $F$  wyprowadzone przez bazę reguł:

$$F(y) = \vee_i F_i(y) = \vee_i (\tau_i \vee D_i(y)) \quad (3.0)$$

Te trzy kroki stanowią tak zwane konstruktywne podejście do wnioskowania. Konstruktywna natura tego podejścia polega na tym, że budowane jest wyjście z reguł składowych. Należy zauważyć, że formułowanie określające wyjście rozmyte wyrażone wzorem (3.8) ma charakter agregacji ważonej następników poszczególnych reguł. Z tego punktu widzenia wagi są określone przez poziom zapłonu, czyli odpowiedniość danej reguły. Aby zastosować wnioskowanie w środowisku sterowania rozmytego, należy dodać jeszcze jeden krok. Wejściem układu sterowanego musi być pojedyncza wartość. Wyjściowy zbiór rozmyty  $F$  wyprowadzony przez bazę reguł nie może być użyty bezpośrednio do sterowania układem deterministycznym. Aby otrzymać wartość nierozmytą na wyjściu FLC, należy wybrać jeden element  $y^*$  z przestrzeni  $Y$ , który będzie reprezentować wartość do zrealizowania. Proces wyboru jednego reprezentatywnego elementu nierozmytego na podstawie wiedzy, że wartość rozmytej zmiennej wyjściowej  $V$  jest  $F$ , nazywa się wyostrzaniem.

Dwie często stosowane metody wyostrzania to metoda Środka Obszaru (*Center of Area* - COA) i metoda Średniej Maksymalnej (*Mean of Maximum* - MOM). Metoda COA określa wartość po wyostrzaniu zbioru rozmytego  $F$  jako jego centroidę:

$$y^* = \frac{\int_Y y F(y) dy}{\int_Y F(y) dy} \quad (3.0)$$

Obliczenia wartości po wyostrzaniu metodą COA upraszcza się do skończonej przestrzeni  $Y$ , czyli dyskretnej funkcji przynależności  $F(y)$ :

$$y^* = \frac{\sum_{j=1}^n F(y_j) y_j}{\sum_{j=1}^n F(y_j)} \quad (3.0)$$

Metoda MOM wyznacza wartość po wyostrzaniu jako średnią ze wszystkich wartości przestrzeni  $Y$  o maksymalnym stopniu przynależności:

$$y^* = \frac{1}{q} \sum_{j \in J} y_j \quad (3.0)$$

przy czym  $J^*$  jest zbiorem elementów przestrzeni  $Y$ , które osiągają wartość maksymalną  $F(y)$ , a  $y$  jest liczbą kardynalną zbioru  $J^*$ .

Inne znane metody wyostrzania to:

- metoda Środka Sum (*Center of Sums* - COS),
- metoda Środka Największego Obszaru (*Center of Largest Area* - COLA),
- metoda Pierwszy z Największych (*First of Maxima* - FOM),
- metoda wysokości (*Height* - HM).

Metoda Środka Obszaru jest inaczej nazywana metodą Środka Ciężkości - *Center of Gravity* (COG).

Podsumowując, procedura obliczania nierozmytego wyjścia bazy reguł FLC określonej wyrażeniem (3.5) dla pewnych wartości zmiennych wejściowych  $U_1$  i  $U_2$  jest oparta na następujących nierozmytych krokach. Pierwszy krok to wyznaczenie poziomu zapłonu reguł (wyrażenie (3.5)). W tym miejscu poziom zapłonu reguły otrzymuje się jako minimum stopnia dopasowania zbioru rozmytego poprzednika każdej reguły dla odpowiedniej wartości wejściowej. W drugim kroku, zwanym implikacją (wyrażenie (3.7)), funkcje przynależności zbiorów rozmytych następnika są obcinane od góry do poziomu odpowiedniego poziomu zapłonu; w ten sposób otrzymuje się zbiory rozmyte  $F_i$ , wnioskowane z poszczególnych reguł. W trzecim kroku, zwanym agregacją reguł (wyrażenie (3.8)), wyjścia poszczególnych reguł  $F_i$  są agregowane i tworzą całkowite wyjście bazy reguł. Czwarty krok to proces wyostrzania; jest to w istocie najczęściej obliczanie centroidy zbioru rozmytego  $F$  (wzór (3.10)). Jak widać ostatnie trzy kroki wymagają dyskretnej postaci funkcji przynależności następnika  $P_i$ , ponieważ agregacje te są wykonywane punktowo na każdym elemencie przestrzeni  $Y$ .

### **Ilustracja podstawowego algorytmu wnioskowania**

*Algorytm wnioskowania:*

1. Oblicz poziomy zapłonu reguł  $\tau_i$  według (3.5) dla zbiorów nierozmytych lub według (3.6) dla wejść w postaci zbiorów rozmytych.
2. Określ postać funkcji przynależności zbioru rozmytego wejścia  $F_i$  wyprowadzonego przez  $i$ -tą regułę zgodnie z wyrażeniem (3.7)
3. Utwórz funkcję przynależności zbioru rozmytego  $F$  na wyjściu wnioskowanego przez bazę reguł FLC przez agregowanie poszczególnych  $F_i$  według (3.8).

4. Oblicz wyjście nierozmyte FLC przez wyostrenie zbioru rozmytego  $F$  za pomocą metody COA (3.9) lub MOM (3.11).

Na rysunku (3.5) zilustrowano algorytm prostego FLC typu Mamdaniego, którego baza reguł składa się z dwóch reguł logicznych:

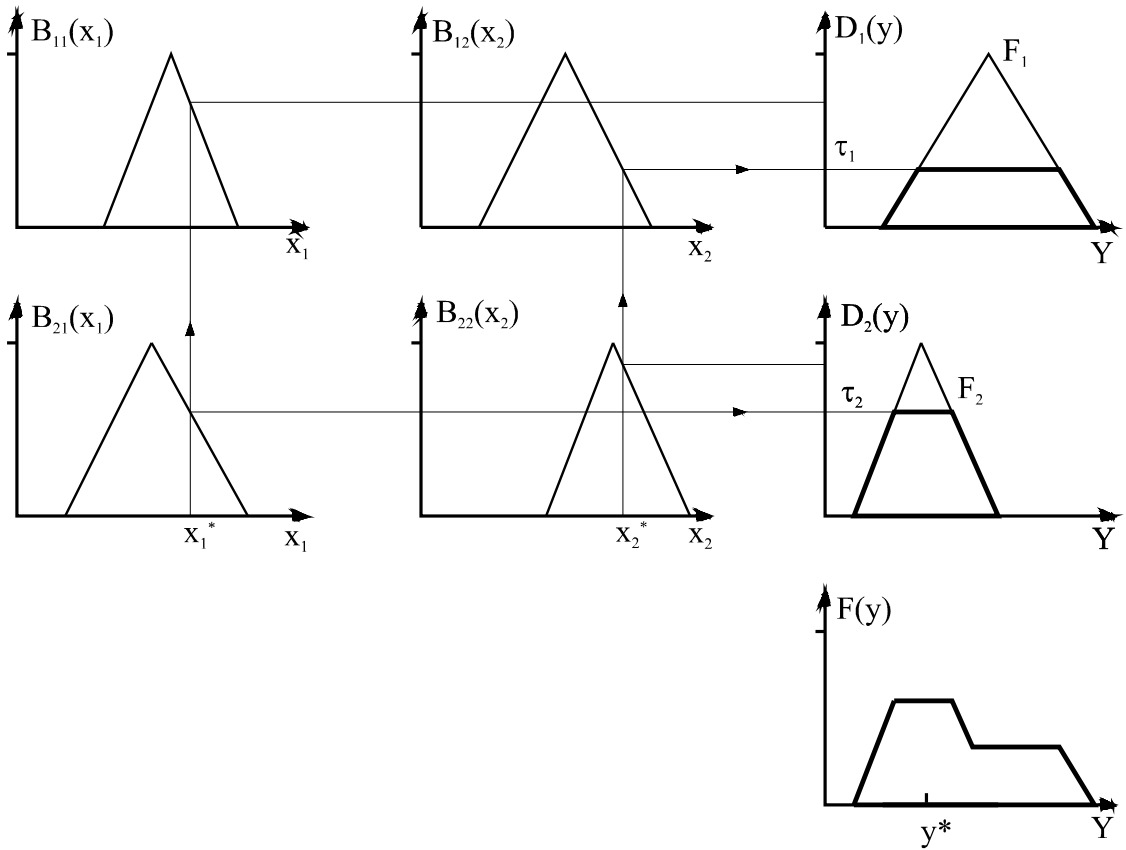
**JEŻELI**  $e(k)$  jest *średnie* **I**  $\Delta$   $e(k)$  jest *małe* **TO**  $\Delta$   $u(k)$  jest *średnie*  $\Delta$   $u$   
**TAKŻE**

**JEŻELI**  $e(k)$  jest *małe* **I**  $\Delta$   $e(k)$  jest *średnie* **TO**  $\Delta$   $u(k)$  jest *małe*  $\Delta$   $u$

Zbiory rozmyte reprezentujące etykiety *średnie*  $e$ , *małe*  $\Delta$   $e$ , *średnie*  $\Delta$   $u$ , *małe*  $e$ , *średnie*  $\Delta$   $e$ , *małe*  $\Delta$   $u$  oznaczone są odpowiednio  $B_{11}$ ,  $B_{12}$ ,  $D_1$ ,  $B_{21}$ ,  $B_{22}$ ,  $D_2$ . Zastosowano metodę wyostrenia MOM.

Algorytm FLC bazuje na trzech zasadach: spójnik I między poprzednikami reguł FLC był interpretowany przez operator min ( $\wedge$ ) we wzorze (3.5); spójnik TAKŻE między poszczególnymi regułami był interpretowany we wzorze (3.8) przez operator max ( $\vee$ ); implikacja rozmyta (3.7) była interpretowana jako iloczyn logiczny poziomów załpónu  $\tau_i$  i zbioru następnika  $D_i$  za pomocą operatora min ( $\wedge$ ). Teoria przybliżonego wnioskowania oferuje wiele alternatywnych interpretacji tych sądów w wyniku rozmaitych modyfikacji podstawowego FLC.

Przedstawione wyżej rozumowanie opisane jest dokładniej w [15,16].



Rys. 3.5. Reprezentacja graficzna algorytmu wnioskowania: wejście nierozmyte  $x_1^* = e(k)$  i  $x_2^* = \Delta e(k)$ ; metoda wyostrzenia MOM; wyjście nierozmyte jest wyprowadzone przez bazę reguł FLC i równa się  $y^* = \Delta u(k)$

## Modele systemu rozmytego

### Kategorie modeli systemów rozmytych

Złożoność świata rzeczywistego najostrzej odzwierciedla się w dziedzinach „nieścisłych” takich jak systemy społeczne, ekonomiczne, ekologiczne, biologiczne. Wielowymiarowość, struktury hierarchiczne, oddziaływania wzajemne, wewnętrzne sprzężenia zwrotne i nieprzewidywalna dynamika są tylko częścią charakterystyki takich systemów złożonych. Złożoność ta jest w pewnym stopniu przyczyną trudności w próbach przeniesienia na te dyscypliny skutecznych technik i systemów sterowania. Słabość tradycyjnych technik ilościowych przy opisywaniu zjawisk złożonych została podsumowana w zasadzie niespójności, sformułowanej przez L. Zadeha [3]. Zasada ta stwierdza, że „jeżeli rośnie złożoność systemu, to nasza zdolność do sformułowania dokładnych i w dodatku znaczących zdań o jego zachowaniu zmniejsza się aż do osiągnięcia wartości progowej, poza którą precyzja i znaczenie (lub ważność) stają się prawie wzajemnie wykluczającymi się cechami”.

Propozycja Zadeha modelowania mechanizmu ludzkiego myślenia za pomocą raczej rozmytych wartości lingwistycznych niż liczb doprowadziła do wprowadzenia rozmytości do teorii systemów i rozwoju nowej klasy systemów, zwanej *systemami rozmytymi*. Najważniejszą cechą systemów rozmytych jest rozmyte kodowanie (podział) informacji. Systemy rozmyte operują na zbiorach rozmytych zamiast na liczbach. Użycie zbiorów rozmytych umożliwia uogólnienie informacji. To uogólnienie związane jest z wprowadzeniem niedokładności. W wielu rzeczywistych problemach niedokładność jest dopuszczalna, a nawet pożyteczna, ponieważ kategorie ludzkiego myślenia są nieokreślonymi pojęciami, które trudną do ilościowo. W zasadzie, reprezentacja informacji w systemach rozmytych imituje mechanizm przybliżonego wnioskowania w wykonaniu ludzkiego umysłu.

Modele systemów rozmytych dzielą się na dwie podstawowe kategorie, które różnią się zasadniczo zdolnością reprezentowania różnych rodzajów informacji. Pierwszą klasą obejmuje Modele Lingwistyczne (LM), których podstawą jest zbiór reguł typu JEŻELI - TO o nieokreślonych predykatkach, i zastosowanie wnioskowania rozmytego. W tych modelach wielkości rozproszone są związane z etykietami lingwistycznymi i model rozmyty jest w istocie jakościowym opisem systemu. Modele tego typu tworzą podstawę modelowania jakościowego, które opisuje zachowanie systemu za pomocą

zyka naturalnego. Przykładem modelu lingwistycznego jest omówiony wcześniej regulator o logice rozmytej. Jak można było zauważyć, jego reguły dają wyrażenia lingwistycznej strategii sterowania na podstawie zdrowego rozsądku.

Drugą kategorię modeli na podstawie metody wnioskowania Takagi-Sugeno-Tanga (TSK) zaproponowali Sugeno i jego współpracownicy [17]. Modele te są tworzone przez reguły logiczne, które mają rozmytą część poprzednika i funkcyjny następnik; w istocie są one kombinacją modeli rozmytych i nierozmytych. Modele rozmyte na podstawie metody wnioskowania TSK integrują zdolność modeli lingwistycznych do reprezentowania wiedzy jakościowej z efektywną możliwością wyrażania informacji ilościowej. W dodatku model rozmyty tego typu umożliwia względnie łatwe zastosowanie potężnej techniki uczenia i identyfikacji modeli na podstawie danych.

### **Postępy modelowania rozmytego**

Podstawowe zasady modelowania rozmytego zostały sformułowane przez Zadeha w [3], w której podał on trzy cechy wyróżniające nowe podejście, dostarczające przybliżonych, ale skutecznych środków opisu zachowania systemów, zbyt złożonych, albo niedostatecznie ściśle zdefiniowanych, aby dopuścić zastosowanie precyzyjnej analizy matematycznej. Te trzy zasady były następujące:

1. Użycie *zmiennych lingwistycznych* zamiast lub w dodatku do *zmiennych liczbowych*.
2. Charakterystyka prostych relacji między zmiennymi przez *warunkowe zdania rozmyte*
3. Charakterystyka relacji złożonych przez *algorytmy rozmyte*

Chociaż Zadeh nie nazwał tego nowego podejścia do modelowania rozmytym, miał niewątpliwie na myśli aparat stosowany obecnie do modelowania systemów rozmytych.

Ważnym zagadnieniem przy projektowaniu takich modeli, tak jak w przypadku technik modelowania, jest podstawowe pytanie o metody ich wykorzystania, gdzie przez wykorzystanie modeli rozmytych rozumie się zbiór technik otrzymywania modelu rozmytego istniejącego systemu na podstawie wiedzy o tym systemie.

Wczesne przykłady modelowania rozmytego, zainspirowane przez zbliżone prace nad systemami ekspertowymi, realizowały idee Zadeha przez próby wyłonienia modelu rozmytego bezpośrednio z wiedzy eksperta. Ta pierwotna metoda konstruowania modeli rozmytych jest oparta na tak zwanym podejściu bezpośrednim. W podejściu bezpośrednim system jest najpierw opisany lingwistycznie przy użyciu terminów z ję-



zyka naturalnego i następnie przekładany na formalną strukturę modelu systemów rozmytych za pomocą teorii wnioskowania przybliżonego.

Opis lingwistyczny jest tworzony subiektywnie na podstawie wiedzy a priori o systemie. Zatem źródłem wprowadzenia reguł lingwistycznych jest bezpośrednio wiedza eksperta o systemie. Ta wiedza jest wyrażona w postaci reguł logicznych. Metodę tę można uważać za jakościową wersję tworzenia modeli tradycyjnych w nauce o systemach. Pierwsze ważne zastosowania logiki rozmytej do modelowania systemów złożonych, w szczególności do naśladowania pracy operatora w dziedzinie inżynierii sterowania wykazały wielką skuteczność tego nowego podejścia w warunkach złożonej rzeczywistości. To bezpośrednio podejście do modelowania rozmytego tylko na podstawie uzyskanego od eksperta opisu funkcjonowania systemu ma pewne swoiste ograniczenia. W bezpośrednim podejściu ilościowe obserwacje funkcjonowania systemu nie są właściwie wykorzystane do wyznaczania struktury bądź parametrów modelu. Jeżeli wiedza eksperta o systemie jest błędna, to można otrzymać zły model. Poszukując większego obiektywizmu przy konstruowaniu modeli rozmytych, naukowcy starali się lepiej wykorzystać techniki formalne, które mogły zrobić użytek z dostępnych danych w celu powiększenia ludzkiej wiedzy lub nawet w celu wykorzystania naszej wiedzy.

Drugi kierunek w zakresie wykorzystania modeli rozmytych, zainspirowany przez klasyczną teorię systemów i ostatnie osiągnięcia w dziedzinie sieci neuronowych, polega na użyciu danych wejście-wyjście. W języku teorii systemów można uważać to podejście za proces identyfikacji systemu. Identyfikacja modeli systemów rozmytych składa się z dwóch głównych faz. Faza pierwsza to *identyfikacja struktury* modelu rozmytego (*identyfikacja strukturalna*), a druga faza to *estymacja wartości parametrów* modelu rozmytego (*identyfikacja parametryczna*). Mówiąc inaczej, identyfikacja strukturalna obejmuje wyznaczanie zmiennych wejściowych i wyjściowych, relacji między zmiennymi (struktura reguł), liczby reguł w bazie reguł i podziału zmiennych wejściowych i wyjściowych na zbiory rozmyte. Identyfikacja strukturalna jest procesem trudnym, wyjątkowo źle zdefiniowanym (bardziej sztuką niż nauką) i nie przystosowanym do technik automatycznych.

Problem otrzymywania struktury modeli rozmytych z danych pojawił się jako jedno z klasycznych zagadnień teorii zbiorów rozmytych w pierwszych latach po opublikowaniu oryginalnych idei Zadeha o modelowaniu rozmytym. Pierwsze podstawowe idee wiążące właściwości reguł rozmytych i ich wyznaczanie na podstawie danych wejście-wyjście pochodzą od Zadeha [18]. Można je uważać za podstawę tak zwanych metod szablonowych. W tym podejściu, które łącznie wykorzystuje wiedzę eksperta i

dane, ekspert systemu dostarcza szablonowych wartości lingwistycznych, które wykorzystujemy do podziału przestrzeni wejście-wyjście; podzbiory rozmyte są dane a priori. Te szablonowe wartości są użyte do określenia potencjalnych reguł modelu rozmytego systemu. Dane wejście-wyjście są następnie używane do tworzenia wag lub prawdopodobieństw związanych z ważnością potencjalnych reguł.

W tym miejscu nacisk kładzie się na uczenie wag reguł (wiarygodność). Pojęcie uczenia wag reguł na podstawie danych wprowadzili Tong [19] i Kosko [20]. Wkład Tonga do problemu identyfikacji modeli rozmytych polegał na wprowadzeniu pojęcia badania logicznego. W osobnych i niezależnych badaniach Kosko rozwinął bardziej ogólną i efektywną obliczeniowo metodę modelowania systemów rozmytych.

Jeżeli informacja o szablonowych wartościach nie jest dostępna, a dostępne są tylko dane wejście-wyjście, to strukturę systemu (relacje między zmiennymi, zgrubne oceny funkcji przynależności poprzedników i następników zbiorów rozmytych i liczby reguł) można otrzymać przez grupowanie przestrzeni wejście-wyjście. Metoda takiego grupowania zapewnia systematyczne podejście do identyfikacji najważniejszych reguł z danych wejście-wyjście.

Duży sukces w identyfikacji parametrycznej modeli rozmytych osiągnięto w ostatnich latach, po opublikowaniu artykułu Takagi i Sugeno [21], w którym opisano nową metodę wnioskowania rozmytego, zwaną TSK (wspomnianą już wcześniej). Autorzy ci nie tylko pokazali wyjątkowo skuteczne podejście do identyfikacji modeli rozmytych na podstawie danych, które łączy pojęcie modelowania rozmytego i filtra Kalmana, ale również zademonstrowali sposób wykorzystania modeli rozmytych przez uproszczenie paradygmatu wnioskowania Mamdaniego, dostarczając bardziej formalnych reprezentacji.

### ***Bezpośrednie podejście do konstruowania modeli lingwistycznych***

Bezpośrednie podejście do modelowania rozmytego polega na wykazaniu kolejnych głównych kroków, które opisano w [3]:

1. Selekcja zmiennych wejścia, stanu i wyjścia
2. Określenie odpowiednich przestrzeni
3. Określenie etykiet lingwistycznych (zbiorów rozmytych odniesienia), na które będą podzielone zmienne.
4. Tworzenie zbioru reguł lingwistycznych reprezentujących relacje między zmiennymi systemu.

5. Wybór odpowiedniego mechanizmu wnioskowania w celu formalizacji modelu rozmytego.
6. Ocena adekwatności modelu.

Niestety, nie ma żadnej ogólnej metody realizacji powyższej procedury. Można spostrzec, że konstrukcja modelu rozmytego na podstawie podejścia bezpośredniego jest bardziej sztuką intuicji i doświadczenia niż ścisłą teorią. Ścisła teoria pojawia się tylko w mechanizmie wnioskowania, który formalizuje model rozmyty.

## Opis wybranego modelu

Analizie poddany został model silnika obcowzbudnego prądu stałego. Silnik taki składa się z dwóch zasadniczych części: nieruchomej - statora i wirującej - rotora. Zadaniem statora, zwanego inaczej magnesnicą, jest wytworzenie przez elektromagnesy napięcia magnetycznego powodującego powstanie strumienia magnetycznego. W statorze znajduje się uzwojenie, przez które płynie prąd.

Zadaniem rotora, zwanego inaczej twornikiem jest wytworzenie napięcia, warunkującego przepływ odpowiedniego prądu. Prąd twornika płynie przez miedziane pręty uzwojenia twornika. Stator i rotor mają rdzenie wykonane ze stali celem stworzenia dla strumienia magnetycznego drogi o dużej permeancji. Dokładniejszy opis można znaleźć w [22, 23].

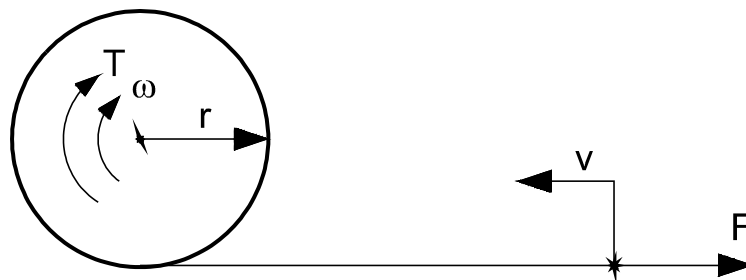
Przed podjęciem próby zaprojektowania takiego modelu należy podzielić pracę na kilka etapów. Podział może wyglądać następująco:

1. Wyprowadzenie matematycznego modelu części mechanicznej i elektrycznej silnika
2. Wyprowadzenie matematycznego modelu sterowania.

Opis procesu modelowania oraz jego sposobów można znaleźć w [24].

### Cześć mechaniczna

Analizowany silnik ma wszechstronne zastosowanie w wielu gałęziach przemysłu. Może być sprzężony z mechaniczną transmisją (przekładnią), która przykładowo zamienia ruch obrotowy silnika na ruch liniowy a moment obrotowy na siły liniowe.



Rys. 5.1- Zależność pomiędzy ruchem obrotowym a liniowym w silniku

Rysunek 5.1 pokazuje takie zależności. Prędkość liniowa jakiegokolwiek punktu oddległości  $r$  od centrum obrotu daną jest wzorem:

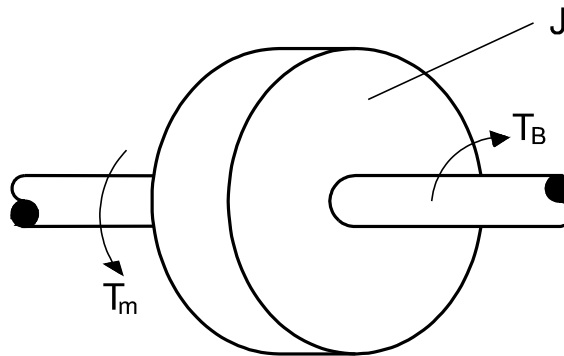
$$v = r\omega \quad (5.0)$$

gdzie:  $\omega$  - prędkość kątowa [rad/s].

Jeżeli nie ma przyspieszenia w elementach składowych przekładni to zależność pomiędzy momentem obrotowym  $T$  a siłą  $F$  wyrażona jest przez:

$$T = rF \quad (5.0)$$

Równania (5.1) i (5.2) są fundamentalne jeżeli chodzi o część mechaniczną. Wyznaczają matematyczną zależność pomiędzy prędkością liniową a kątową oraz siłą liniową a momentem obrotowym.



Rys. 5.2 - Schemat twornika silnika

Rysunek 5.2 przedstawia schematycznie twornik silnika. Twornik motoru posiada moment bezwładności  $J$  [kg\*m<sup>2</sup>]. Jedno z założeń jest takie, iż zachowanie w tworniku i przełożeniu wywołuje tłumienie wyrażone współczynnikiem tarcia lepkiego  $B$  [Nms]. Prąd przepływający przez twornik wytwarza moment elektromagnetyczny silnika  $T_m$ , pomniejszany przez reaktywny moment porowy typu generatorowego  $T_B$ . Z rysunku 5.2 można zatem zapisać:

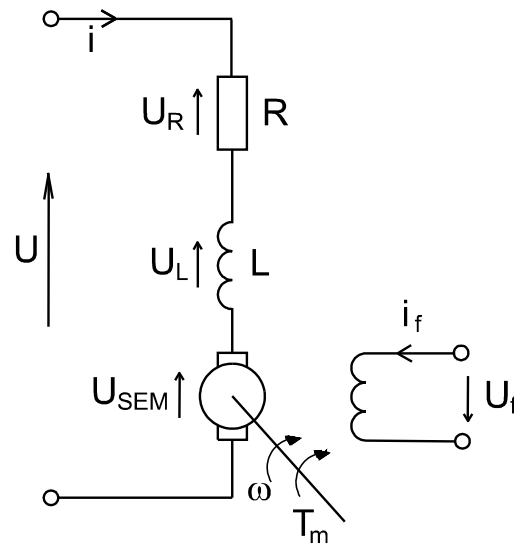
$$T_m - T_B = J \frac{d\omega}{dt} \quad (5.0)$$

gdzie:

$$T_B = B\omega \quad (5.0)$$

Podstawiając (5.4) do (5.3) otrzymujemy:

$$J \frac{d\omega}{dt} + B\omega = T_m \quad (5.0)$$

**Część elektryczna**

Rys. 5.3 - Schemat obwodu silnika

Rysunek 5.3 przedstawia schemat obwodu silnika obcowzbudnego prądu stałego. Wzbudzenie obce wytwarza pole magnetyczne tym silnika. Napięcie  $U$  [V] przyłożone wszerez twornika będzie pomniejszane przez spadek napięcia na oporności twornika  $R$  [ $\Omega$ ], spadek napięcia na indukcyjności obwodu twornika  $L$  [H] oraz wsteczną siłę elektromotoryczną  $U_{SEM}$  [V] indukowaną poprzez rotujący w polu magnetycznym twornik.

Moment elektromagnetyczny silnika jest proporcjonalny do prądu twornika:

$$T_m = M_a^f \cdot i_f \cdot i \quad (5.0)$$

gdzie:

$M_a^f$  - rotacyjna indukcyjność wzajemna (H/rad),

$i_f$  - prąd wzbudzenia [A],

$i$  - prąd twornika [A].

Siła elektromotoryczna generowana przez silnik jest proporcjonalna do jego prędkości kątowej:

$$U_{SEM} = M_a^f \cdot i_f \cdot \omega \quad (5.0)$$

Ze schematu przedstawionego na rysunku 5.3 i stosując napięciowe prawo Kirchhoffa można zapisać następujące równanie:

$$U = U_R + U_L + U_{SEM} \quad (5.0)$$

czyli inaczej:

$$U - Ri - L \frac{di}{dt} - U_{SEM} = 0 \quad (5.0)$$

Podstawiając (5.7) do (5.9) otrzymujemy:

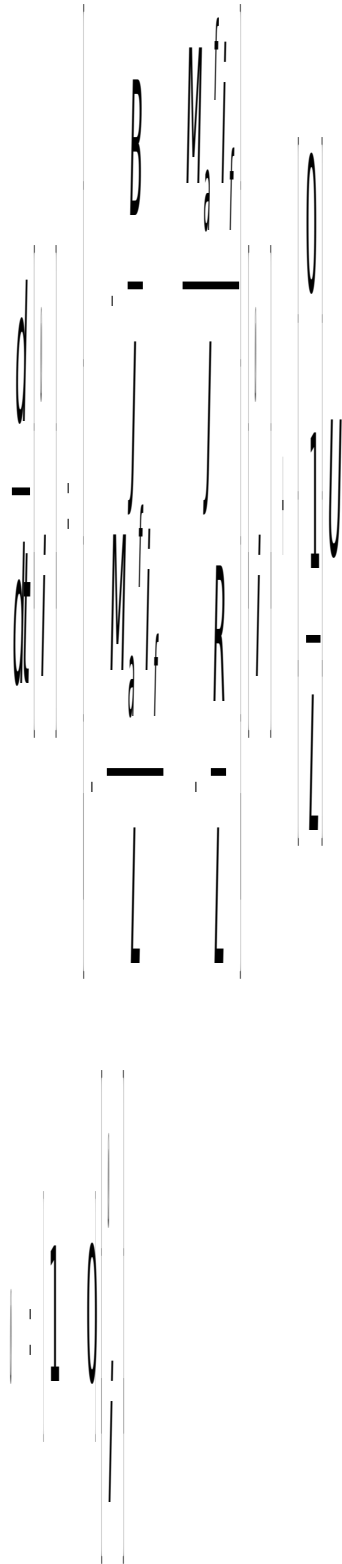
$$L \frac{di}{dt} + Ri = U - M_a^f i_f \omega \quad (5.0)$$

Jest to równanie różniczkowe pierwszego rzędu ze stałymi współczynnikami powiązanymi z prądem twomika jako funkcją napięcia sterowania i wstecznej siły elektromotorycznej.

Kompletny opis modelu stanowią wspólnie równania (5.5) i (5.10) (po wstawieniu (5.6) do (5.5)):

$$\begin{cases} J \frac{d\omega}{dt} + B\omega = M_a^f i_f i \\ L \frac{di}{dt} + Ri = U - M_a^f \omega \end{cases} \quad (5.0)$$

Układ równań (5.11) można zapisać w formie macierzowej. Prędkość kątową  $\omega$  oraz prąd twomika  $i$  przyjmuje się jako zmienną stanu a napięcie sterowania  $U$  jako wejście. Za wyjście przyjmuje się prędkość kątową. Reprezentacja macierzowa ma następującą postać:



**Parametry wewnętrzne wybranego silnika**

Do sterowania wybrano jeden z silników produkowanych przez Zakład Specjalnych Maszyn Elektrycznych i Urządzeń Technologicznych „Komel” w Katowicach.



Z katalogu wybrano silnik typ PZb32b (numer katalogowy II-1/81). Silnik ten posiada zgodność PN-72/E-06000. W katalogu zostały zawarte tylko niektóre parametry:

$$\text{Moc znamionowa } P_N = 1,5 \text{ kW}$$

$$\text{Prąd znamionowy } I_N = 8,4 \text{ A}$$

$$\text{Prąd wzbudzenia } i_f = 0,5 \text{ A}$$

$$\text{Maksymalny moment obrotowy } T = 4,78 \text{ Nm}$$

$$\text{Znamionowa prędkość obrotowa } n_N = 3000 \text{ obr/min}$$

$$\text{Moment bezwładności twornika } J = 0,05 \text{ kg}\cdot\text{m}^2$$

$$\text{Napięcie znamionowe } U_N = 220 \text{ V}$$

$$\text{Napięcie wzbudzenia } U_f = 220 \text{ V}$$

$$\text{Współczynnik tarcia lepkiego } B = 0,0098 \text{ Nms}$$

Pozostałe wartości należało wyznaczyć<sup>1</sup>.

Rotacyjną indukcyjność wzajemną  $M_{af}^f$  można obliczyć znając wartości siły elektromotorycznej przy ustalonej pracy znamionowej:

$$U_{SEM} = U_N - I_N R \quad \text{gdzie } R - \text{rezystancja obwodu twornika.}$$

$$R = \frac{P_{dcuN}}{I_N} \quad \text{gdzie } P_{dcuN} \approx (5 \div 10)\% P_N$$

$$\text{Wtedy } M_{af}^f = \frac{U_{SEM}}{\omega_N i_f} \quad \text{gdzie } \omega_N - \text{nominalna prędkość obrotowa ale w rad/s.}$$

Kolejną wielkością do wyznaczenia jest indukcyjność obwodu twornika - L. Wielkość ta jest trudna do bezpośredniego obliczenia. Na podstawie podawanych w literaturze przybliżonych wzorów częściowo doświadczalnych można przyjąć dla silnika obcowzbudnego mocy około 1 kW:

$$L = 0,08 \cdot 60 \frac{U_N}{I_N} p \cdot n_N$$

gdzie: p - liczba par biegunów komutacyjnych (w małych silnikach zwykłe p = 1).

Wyznaczono w ten sposób wartości wynoszące odpowiednio:

$$\text{Oporność twornika } R = 12,4 \Omega$$

$$\text{Rotacyjna indukcyjność wzajemna } M_{af}^f = 0,73 \text{ H/rad}$$

$$\text{Indukcyjność } L = 0,377136 \text{ H.}$$

<sup>1</sup> Wzory szacunkowe i parametry silnika uzyskano dzięki pomocy dr hab. inż. Zbigniewa Tertila.

## Model sterowania dla silnika obcowzbudnego prądu stałego

### Wybór platformy sprzętowej

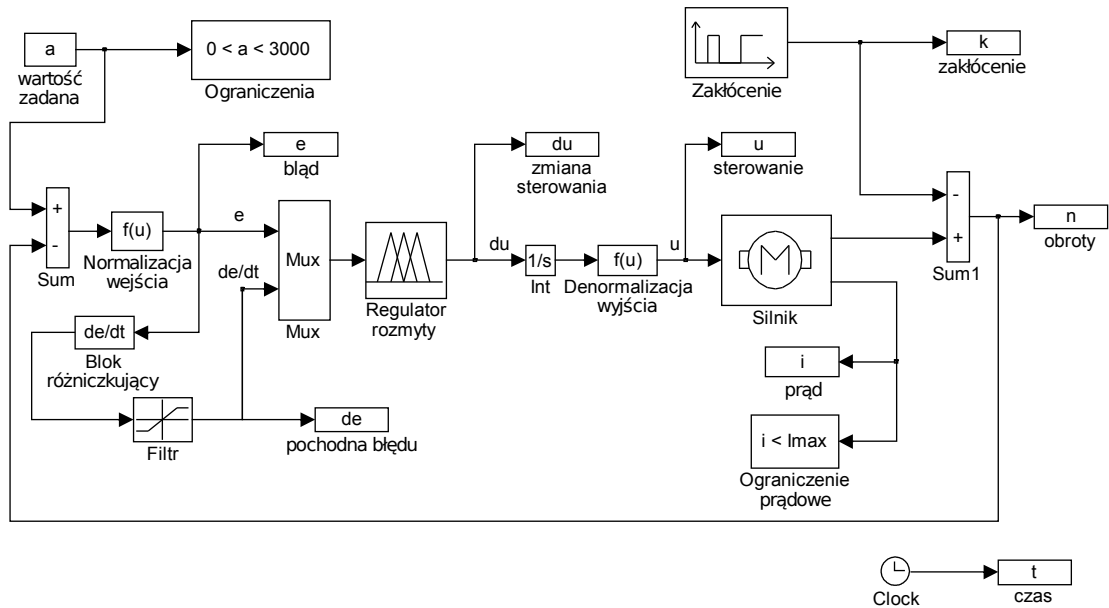
Na rynku nie ma zbyt wielu dostępnych programów umożliwiających przeprowadzanie badań testów zaprojektowanych systemów, opisanych w sposób analityczny. Tym gorzej sprawa wygląda dla zagadnień sterowania rozmytego. Do stworzenia obiektu symulowanego - silnika obcowzbudnego prądu stałego jak i reszty komponentów testowanego systemu wykorzystano popularny w środowisku studenckim, dostępny na uczelni, program *Matlab* oraz dodatkowy pakiet do przeprowadzania symulacji - *Simulink* (*Matlab* i *Simulink* są zarejestrowanymi nazwami i znakami towarowymi firmy The Math Works, Inc.) Wersja ta posiada licencję na Akademii Górniczo - Hutniczej. Do zaprojektowania samego regulatora rozmytego FLC wykorzystano zaś pakiet *Fuzzy Logic Toolbox*, również dostępny na uczelni (zostanie on opisany w kolejnym rozdziale).

Wybór tej platformy sprzętowej pozwolił na stosunkowo łatwiej czytelne zaprojektowanie całego układu. Program umożliwia przeprowadzanie symulacji, a w związku z tym obserwację wpływu poszczególnych elementów składowych na zachowanie się sterowanego obiektu. Wykresy umożliwiają zilustrowanie efektu sterowania i zależności pomiędzy wszystkimi zmiennymi.

Przy pomocy pakietu *Simulink* zostały utworzone dwa oddzielne programy ze względu na dwa rodzaje regulatorów FLC zastosowanych do sterowania. W tym rozdziale uwaga zostanie poświęcona całemu układowi niezbędnemu do przeprowadzania testowania. Zostanie pominięty szczegółowy opis samego sterowania.

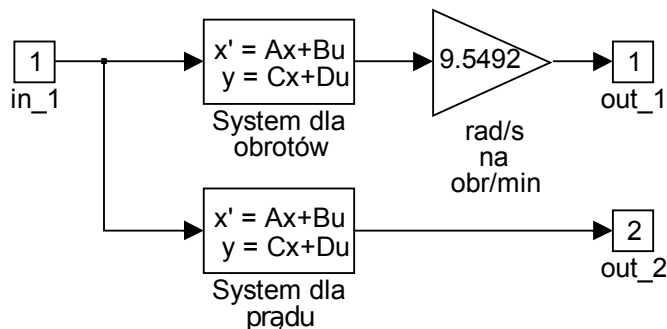
### Model PI

Pierwszy z modeli reprezentuje regulację rozmytą typu PI (opisaną dokładniej w następnym rozdziale 7). Zawarty jest on w pliku *dcmam1.m* (uruchamianym z linii *Matlaba*). Schemat przedstawia rysunek 6.1.



Rys. 6.1 - Schemat systemu sterowania i symulacji dla regulatora FLC typu PI

Obiekt sterowany zawarty jest w bloku *Silnik*. Jest to zamaskowany podsystem, który zawiera model silnika obcowzbudnego prądu stałego zaprojektowany przy pomocy bloków *Statespace Matlab* (są one matlabowską reprezentacją układu równań macierzowych 5.11). Przedstawia to rysunek 6.2.



Rys. 6.2 - Matlabowski schemat modelu silnika

Po otwarciu blocka pojawia się okno dialogowe, w którym można wpisać wewnętrzne parametry silnika niezbędne do przeprowadzenia symulacji. Domyślnie wpisane są katalogowe parametry silnika opisanego szczegółowo w punkcie 5.3.

Wejściem obiektu sterowanego jest napięcie twornika  $u$  a wyjściami obroty  $n$  i prąd  $i$ . Drugie wyjście uzyskano przez skopiowanie obiektu i zmianę wektora  $C$  (z  $[1 \ 0]$  na  $[0 \ 1]$ ). Obroty będące bezpośrednim wyjściem *Systemu dla obrotów* wyrażone są

w rad/s, dlatego też celem uzyskania bardziej obrazowych obr/min przeskalowuje się wyjście przez współczynnik 9.5492 ( $1 \text{ rad/s} = 9.5492 \text{ obr/min}$ ).

Sterowany silnik ma stabilizować obroty na poziomie wartości zadanej  $a$ . Wartość tę podaje się w linii *Matlaba* przed rozpoczęciem symulacji. Założono, że wartość zadana obrotów mieści się w przedziale  $[0, 3000]$ . Wartość 3000 obr/min odpowiada obrotom nominalnym silnika (obroty silnika pracującego bez obciążenia). Zrealizowane jest to w bločku *Ograniczenia*. W przypadku przekroczenia tych ograniczeń (podania liczby spoza tego zakresu) następuje zakończenie symulacji.

Różnica pomiędzy wartością zadaną obrotów ( $a$ ) a wartością aktualną obrotów ( $n$ ) stanowi wejście kolejnego bloku: *Normalizacja wejścia*. Następuje w nim przeskalowanie sygnału wchodzącego na bloček do znormalizowanej dziedziny  $[-1, 1]$  (szerzej zostanie to opisane w rozdziale 7). Współczynnik normalizacji wynosi *sygnał wejściowy/wartość zadana  $a$* . Uzyskuje się w ten sposób jedno z wejść regulatora rozmytego - *błąd*. Drugim wejściem regulatora jest *pochoďna błędu  $du/dt$* . Oba wejścia podłączone są do bločka *Regulatora rozmytego* poprzez multiplexer. Dodatkowo na linii pochodnej błędu podpięty jest bloček saturacji - *Filtr*. Jego zadaniem jest ograniczenie przepuszczanych sygnałów do tych należących jedynie do wspomnianej wyżej dziedziny znormalizowanej. Eliminuje się w ten sposób ewentualne przekroczenia wynikłe z wpływu dużego zakłócenia, którego efektem może być wartość wejścia nie mieszcząca się w przestrzeni rozważań. W bločku *Regulator rozmyty* (zamaskowany podsystem) wpisana jest nazwa zmiennej, w której mieszczą się parametry regulatora utworzone za pomocą edytora *FIS* (patrz rozdział 7). W tym przypadku jest to zmienna *dcmam2*. Ze względu na charakter regulatora (PI), jego wyjściem jest *zmiana sterowania  $du$* . Dlatego też celem uzyskania sterowania  $u$  (bezpośredniego wejścia sterowanego silnika) zastosowano blok całkujący *Int*. Wyjście regulatora rozmytego opisane jest, podobnie jak wejścia, na dziedzinie znormalizowanej, dlatego też po całkowaniu następuje *Denormalizacja wyjścia*. Współczynnik denormalizacji wynosi *sygnał wejściowy \* 220 \* wartość zadana  $a$  / 3000*. Należy nadmienić, że normalizacja wejścia i denormalizacja wyjścia przeprowadzane są celem zachowania stałych parametrów sterowania dla różnych nastaw wartości zadanej obrotów. W wyżej wspomnianych bločkach wykorzystywana jest wartość zadana  $a$ . Przez to przed rozpoczęciem symulacji należy ją podać w linii *Matlaba*. Nie można jej podać w oknie symulacji, gdyż będzie nieznaną w pierwszym kroku symulacji - pojawi się wtedy informacja o błędzie.

Do systemu podłączony jest też blok *Zakłócenie*. Stanowi on również zamaskowany podsystem. Po uruchomieniu otwiera się okno dialogowe, w którym wpisuje się

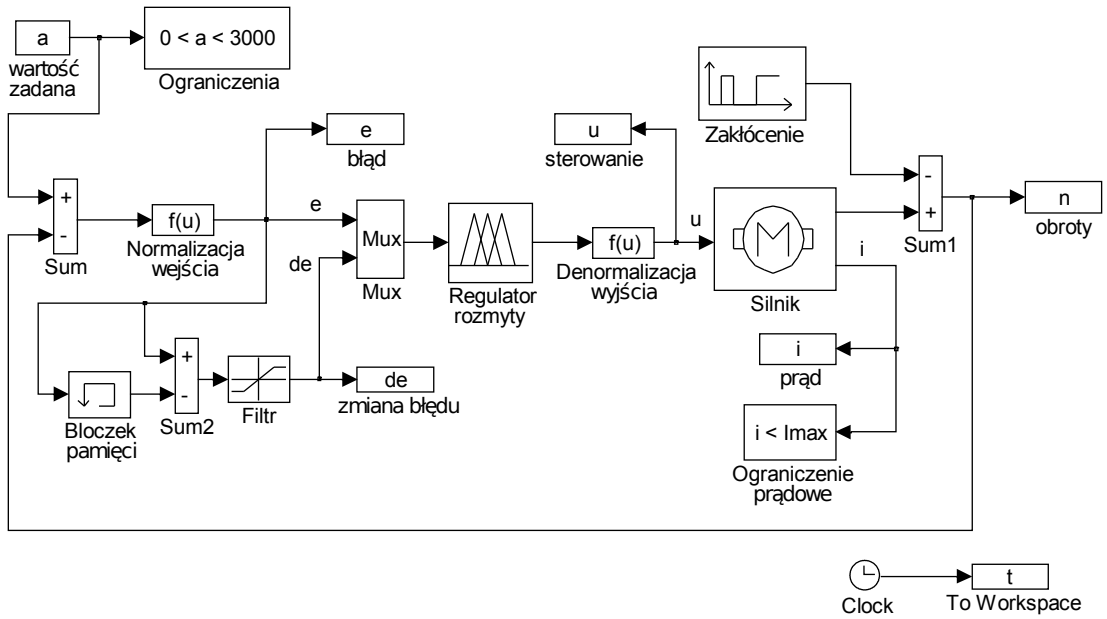
podstawowe parametry zakłócenia. Do wyboru mamy skok jednostkowy, falę prostokątną lub kombinację obydwu. Zakłócenie wpływa na wartość wyjściową obrotów. Jest ono w rzeczywistości symulowanym obciążeniem zewnętrznym przyłożonym do silnika.

W systemie istnieje jeszcze jedno ograniczenie. Jest to *Ograniczenie prądowe*. W bločku podaje się maksymalną wartość prądu płynącego przez twornik. Domyślnie ustawiona jest wartość 20A co odpowiada ok. 120% wartości maksymalnego prądu uzyskiwanego przy rozruchu silnika po podłączeniu do napięcia znamionowego (220V). W przypadku przekroczenia wpisanej wartości, w trakcie trwania symulacji, następuje jej zatrzymanie.

### **Model PD**

Drugi z modeli wykorzystuje regulację rozmytą typu PD. Schemat przedstawia rysunek 6.3. W samym modelu systemu widoczne są tylko niewielkie różnice opisane poniżej. Główną różnicą tkwi w samej idei regulatora rozmytego. W bločku *Regulator rozmyty* znajduje się nazwa innej zmiennej: *dcmamdan*, odpowiadającej innym nastawom parametrów regulatora.

Kolejna różnica jest taka, że na regulator nie wchodzi już pochodna błędu lecz jego zmiana -  $de$ . W związku z tym zamiast bločku różniczkującego jest bloczek pamięci, który zapamiętuje wartość sygnału z poprzedniego kroku symulacji. Dzięki takim rozwiązaniom znacznie polepszyła się efektywność sterowania.



Rys. 6.3- Schemat systemu sterowania i symulacji dla regulatora FLC typu PD

Ostatnia różnica jest na wyjściu z bloku regulacji. Jest tam sterowanie  $u$  a nie jego zmian jak ma to miejsce w przypadku regulatora PI. Reszta układu, czyli ograniczenia, zakłócenie, normalizacja i denormalizacja wraz ze współczynnikami są takie same jak w przypadku regulatora FLC typu PI.

## Regulator rozmyty dla silnika obcowzbudnego prądu stałego

### Wstęp

Jak już wspomnianow poprzednimrozdziale częśćregulatorowasystemustero-  
wania silnika zawierasię w bloczku *Regulator rozmyty*. Wewnątrz bloczkazawartajest  
nazwa zmiennej zawierającej się w przestrzeni *Matlaba* (*workspace*) i reprezentująca  
działanie regulatora. W niej zawarte są wszystkie parametry: rodzaje i zakres funkcji  
przynależności, zbór reguł, rodzajeoperatorów.

Regulator rozmyty został stworzony za pomocą pakietu *Fuzzy Logic Toolbox*,  
będącego dodatkiem do programu Matlab. Dokładny opis pakietu można znaleźć  
w [25].

Zmienną reprezentującą dany regulator uzyskano przy pomocy *FIS Editor'a*  
(*Fuzzy Inference System*- uruchamiany komendą *fuzzy* w linii *Matlaba*). Edytor jak i  
cały pakiet wykorzystuje graficzny interfejs użytkownika (*GUI Tools*), który musi być  
zainstalowany przed rozpoczęciem pracy projektowej. Dzięki temu projektowanie sys-  
temów opartych na logice rozmytej jest wygodniejsze w porównaniu z pracą w oknie  
*Matlaba*.

W pracy do sterowania silnikiem wykorzystano dwa regulatory rozmyte: pierw-  
szy typu PI i drugi typu PD. W tym rozdziale zostaną przedstawione procedury projek-  
towania regulatorów dla silnika obcowzbudnego prądu stałego, opis parametrów tych  
regulatorów, różnice między nimi oraz proces strojenia. Strojenie regulatora FLC jest  
procedurą trudniejszą i bardziej skomplikowaną niż strojenie regulatora konwencjonal-  
nego. Przyczyną tego jest fakt, że FLC jest układem elastycznym, którego zachowanie  
jest określone dużą liczbą parametrów definiujących funkcje przynależności i mecha-  
nizm wnioskowania. Najbardziej skutecznwyniki strojenia otrzymujemy na podstawie  
kombinacji dobrego zrozumienia obiektu sterowanego przez eksperta i zastosowanie  
analogii międzyregulatorami FLC i konwencjonalnymi PID.

### Konstruowanie regulatora FLC typu PI

Regulator tego typu opisuje za pomocą reguł rozmytych JEŻELI - TO relacje  
między zmienną sterowania  $\Delta u(k)$  z jednej strony i uchybem  $e(k)$  oraz jego zmianą

$\Delta e(k)$  z drugiej strony. Wewnętrznym mechanizmem regulatora FLC przekładano w odwzorowanie:  $\Delta u(k) = F(e(k), \Delta e(k))$  (patrz wzór 3.2).

Równanie opisujące konwencjonalny regulator PI ma postać:

$$u = K_p \cdot e + K_i \cdot \int e dt \quad (7.0)$$

przy czym  $K_p$  i  $K_i$  są współczynnikami członów proporcjonalnego i całkowania regulatora PI. Jeżeli zróżniczkujemy wyrażenie (7.1) po czasie  $t$ , to można je przekształcić do równoważnego wyrażenia postaci:

$$\dot{u} = K_p \cdot \dot{e} + K_i \cdot e \quad (7.0)$$

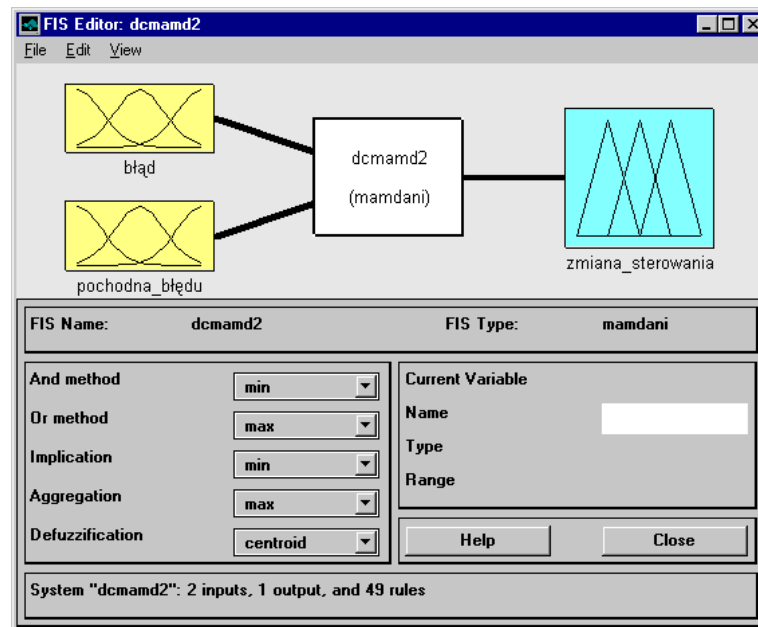
Aby dopełnić całości regulacji rozmytej PI należy dodać element całkujący po regulatorze (patrz rysunek 6.1). Dzieje się to poza regulatorem FLC typu PI i nie oddziałuje na reguły, które ostatecznie mają postać:

**JEŻELI**  $e(k)$  jest <symbol własności> I  $\Delta e(k)$  jest <symbol własności>

**TO**  $\Delta u(k)$  jest <symbol własności>

gdzie <symbol własności> jest nazwą symboliczną zmiennej lingwistycznej, a  $k$  chwilą próbkowania.

Po uruchomieniu edytora FIS pojawi się okno przedstawione na rysunku 7.1.



Rys. 7.1- Ogólny wygląd edytora FIS dla zmiennej dcmamd2

W oknie tym definiuje się podstawowe parametry regulatora. Mianowicie określa się czy system ma być typu Mamdaniego czy też Sugeno (TSK). System *dcmamd2* jest typu



Mamdaniego (konstruktywny) - najczęściej stosowany. Oznaczano, że jest to regulator oparty na modelu lingwistycznym, w którym wyjście jest konstruowane przez superpozycję wyjść poszczególnych reguł. Stosuje się wtedy oryginalną metodę wnioskowania typu Mamdaniego. W tym podejściu każda reguła:

**JEŻELI**  $U$  jest  $B_i$  **TO**  $V$  jest  $D_i$

wyraża się jako relacja rozmyta  $R_i$ , interpretowana jako iloczyn rozmyty zbiorów rozmytych  $B_i$  i  $D_i$ :

$$R_i = B_i \cap D_i \quad (7.0)$$

$R_i$  jest określona na iloczynie kartezjańskim przestrzeni  $X \times Y$  i ma funkcję przynależności:

$$R_i(x, y) = B_i(x) \wedge D_i(y) \quad (7.0)$$

W modelu Mamdaniego agregacja reguł (spójnik TAKŻE) jest wykonywana przez sumę poszczególnych relacji rozmytych:

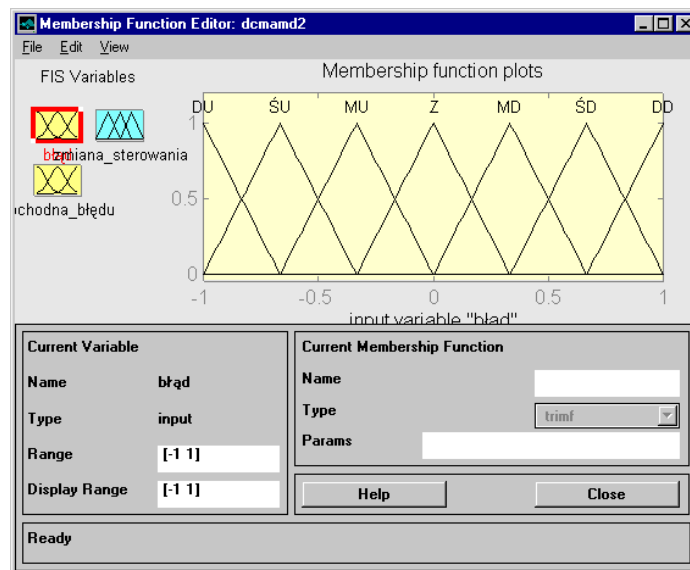
$$R = \bigcup_{i=1}^m R_i. \quad (7.0)$$

Taka reprezentacja umożliwia otrzymanie modelu konstruktywnego. Dodatkowo informacja na temat można również znaleźć w innych punktach pracy (2.3, 3.2, 3.3).

Ponadto, w oknie edytora FIS, określa się liczbę wejść i wyjść oraz nadaje się im nazwę. W tym przypadku określono dwa wejścia: *błąd*, *pochodna błędu* i jedno wyjście *zmiana sterowania*. Określa się również metodę logicznego I (*And*), logicznego LUB (*Or*), rodzaj implikacji (*Implication*), rodzaj agregacji (*Aggregation*), metodę wyostrzania (*Defuzzification*). W każdym z tych przypadków istnieje możliwość zadeklarowania własnego rozwiązania (*custom*).

Po selekcji zmiennych wejścia i wyjścia należy wyznaczyć przestrzenie, na których będą określone. Do tego służy kolejny składnik edytora FIS - *Membership Function Editor* przedstawiony na rysunku 7.2. Określa się to w polu *Range* (z ang. zakres). Jak już wspomniano w rozdziale 6, celem uzyskania płynnej regulacji obrotów w całym zakresie, wejścia i wyjście regulatora FLC określono na dziedzinach znormalizowanych. Wymagano transformacji skali, która przekształca wartości fizyczne zmiennych stanu procesu dziedzinie znormalizowane. Nazywa się to normalizacją wejścia. Poza tym denormalizacja wyjścia przekształca znormalizowane wartości zmiennych sterujących w odpowiednie dziedziny fizyczne. W związku z tym dla wszystkich tych zmiennych przestrzeń jest przedziałem  $[-1, 1]$ . Mimo, że niektóre funkcje przynależności

zmiennych lingwistycznych (opisanych w dalszej części) wykraczają poza ten obszar, w regulacji pod uwagę brana jest tylko ta właśnie przestrzeń.



Rys. 7.2 - Edytor funkcji przynależności - MembershipFunctionEditor

Czynniki skalujące, które opisują poszczególne normalizacje wejścia i denormalizacje wyjścia odgrywają podobną rolę, jak współczynniki wzmocnienia w konwencjonalnym regulatorze. Innymi słowy, mają one bezpośredni wpływ na działanie regulatora i jego własności stabilności. W przypadku opisywanego regulatora do określenia tych czynników wykorzystana została metoda heurystyczna oparta na metodzie prób i błędów. Trzeba zdawać sobie sprawę, że zmiana czynników skalujących dla każdego z  $e$  i  $\Delta e$ , zmienia wagi poszczególnych zmiennych stanu procesu. Efektywność sterowania na podstawie współczynników skalowania jest ograniczona przez sprzeczne wymagania co do tych współczynników, wynikających z różnych miar jakości.

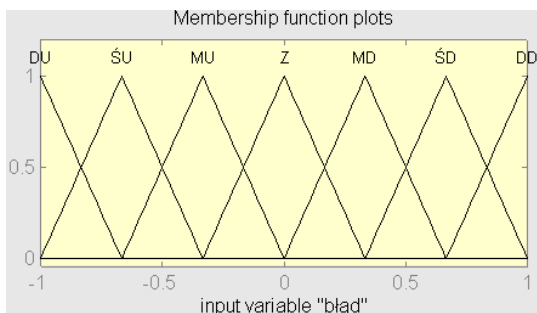
Po określeniu przestrzeni dla zmiennych należy określić etykiety lingwistyczne (zbiory rozmyte odniesienia), na które będą podzielone te zmienne. Wartości lingwistyczne, wchodzące w skład zbioru słów, są wyrażone jako  $n$ -tki o postaci <znak wartości, wielkość wartości>. Część znakowa wartości takiej  $n$ -tki przybiera jedną z dwóch wartości: dodatnią lub ujemną. Wielkość wartości może przybierać dowolną liczbę wielkości wyrażanych lingwistycznie. Dla wszystkich zmiennych: *bład*,  *pochodna\_błędu*, *zmiana\_sterowania* określono następujące etykiety:

1. DU - **D**uży **U**jemny,
2. ŚU - **Ś**redni **U**jemny,
3. MU - **M**ały **U**jemny,

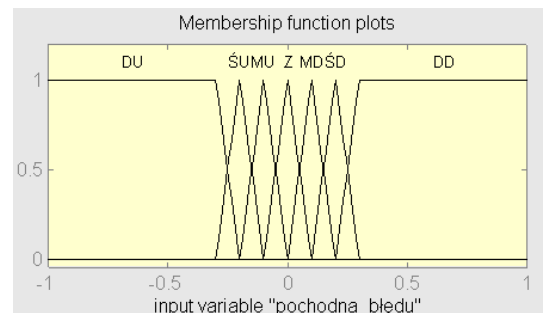
4. Z - **Z**erowy,
5. MD - **M**ały **D**odatni,
6. ŚD - **Ś**redni **D**odatni,
7. DD - **D**uży **D**odatni.

Zmienna  $\Delta e$  określona jest jako różnica pomiędzy wartością zadaną obrotów a wartością aktualną. Dlatego wartości lingwistyczne  $e$  ze znakiem dodatnim oznaczają, że bieżąca prędkość obrotowa przyjmuje wartość poniżej wartości zadanej. Wartości  $e$  ze znakiem ujemnym oznaczają, że aktualna prędkość obrotowa jest za duża. Wartości lingwistyczne  $\Delta e$  z ujemnym znakiem oznaczają, że wartość bieżących obrotów wzrosła w porównaniu z poprzednią wartością (dokładne rozumowanie znajduje się w punkcie 3.2). Analogicznie dla  $\Delta e$  ze znakiem dodatnim. Wartość lingwistyczna  $Z$  dla zmiennej  $e$  i  $\Delta e$  oznacza odpowiednio, że prędkość jest bliska wartości zadanej i nie zmieniła się w ostatnim kroku. Jeżeli chodzi o zmienną wyjścia  $\Delta u$ , to jej wartości dodatnie oznaczają, że wartość zmiennej sterującej musi wzrosnąć dla bieżącej chwili próbkowania. Analogicznie dla ujemnych wartości.

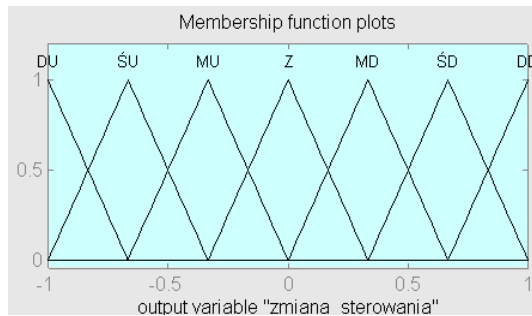
Zbiory przynależności dla wyżej wymienionych etykiet różnią się między sobą. Na początku projektowania wszystkie funkcje przynależności wszystkich zmiennych były równomiernie rozłożone w swoich przestrzeniach (jest to domyślna opcja edytora). Jednak regulator nie działał wtedy prawidłowo. Pojawiały się oscylacje, przeregulowania. Innymi słowy nie osiągnięto założonego rezultatu stabilizacji obrotów i odporności na zakłócenie. Rysunek 7.3 przedstawia rozłożenie funkcji dla poszczególnych zmiennych. Dobór odpowiednich funkcji przynależności, ich kształtu, rozmieszczenia w przestrzeni stanowi część procesu dostrajania regulatora. Ścisnięcie zbiorów funkcji przynależności dla pochodnej błędu wynikało z wielu przeprowadzonych prób doświadczalnych. Przy takim rozłożeniu uzyskano najlepsze wyniki sterowania procesem stabilizacji obrotów.



a)



b)



c)

Rys. 7.3 - Zbiory funkcji przynależności dla FLC typu PI: a) błąd, b) pochodna błędu, c) zmiana sterowania

Do przeprowadzenia obliczeń, ze względu na efektywność pamięci komputera i wymagania analizy, jest wskazana jednolita reprezentacja funkcji przynależności. Można ją uzyskać stosując funkcję przynależności o ujednoliconym kształcie i parametryczną definicję funkcji.

W przypadku regulatora FLC typu PI zastosowano funkcje trójkątnej i trapezoidalnej (tylko w dwóch przypadkach dla zmiennej *pochodna błędu*). Uzasadnione jest to łatwością uzyskania parametrycznego opisu funkcji przynależności, minimalną ilością pamięci potrzebną do jego przechowywania i efektywnością przetwarzania przez maszynę wnioskującą. Należy dodać, że parametryczny opis trójkątnej funkcji przynależności jest najbardziej ekonomiczny - wymaga jedynie podania trzech parametrów. Dla funkcji trapezoidalnej już czterech. W edytorze służy do tego pole *Params* (patrz rysunek 7.2). Oczywiście edytor pozwala na wybór jeszcze innych rodzajów funkcji przynależności (oprócz zastosowanych trójkątnej i trapezoidalnej) - gaussowskiej (*gaussmf*), dzwonowej (*gbellmf*) i innych. Jednak przeprowadzone symulacje wykazały, że kształt funkcji przynależności nie ma większego wpływu na zachowanie się obiektu i stabilizację obrotów silnika.

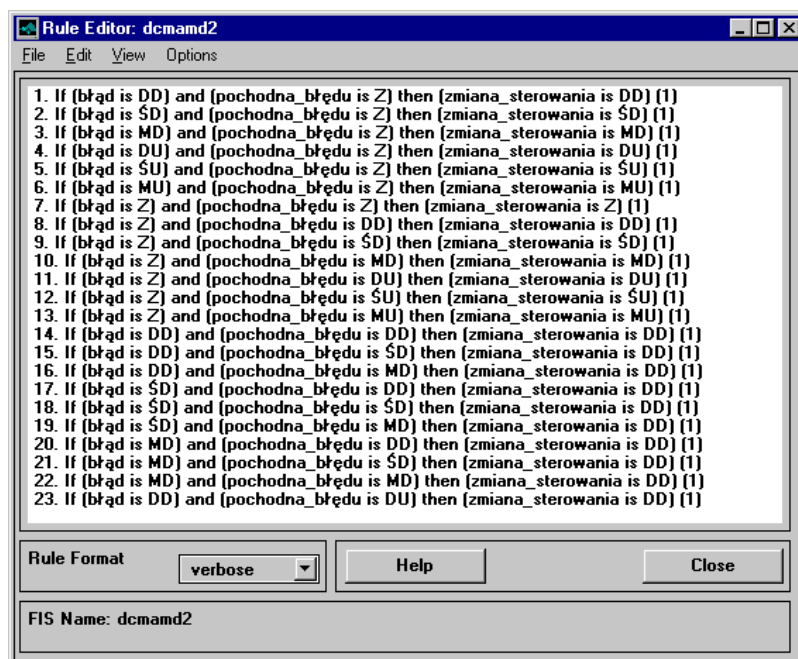
Wszystkie trójkątne funkcje przynależności w regulatorze są symetryczne. Oznaczano, że ich lewa i prawa szerokości są sobie równe. Lewa szerokość jest to odległość od punktu na lewo od wartości szczytowej, dla którego stopień przynależności wynosi zero, do wartości szczytowej (punkt w którym stopień przynależności wynosi 1). Analogicznie definiuje się prawą szerokość. Suma lewej i prawej szerokości definiuje długość przedziału, który nazywany jest nośnikiem funkcji przynależności.

Punktem przecięcia dwóch funkcji przynależności określa się taką wartość  $x$  należącą do przestrzeni zdyskretyzowanej, dla której  $\mu_1(x) = \mu_2(x) > 0$ . Punkt ten oznacza poziom przecięcia definiowany za pomocą stopnia przynależności. Natomiast

współczynnikiem przecięcia określa się liczbę punktów przecięcia. Wszystkie funkcje przynależności mają współczynnik przecięcia równy 1. Dzięki takiej sytuacji wykluczony jest przypadek odpalenia w danej chwili tylko jednej reguły (tak się dzieje jeżeli współczynnik przecięcia wynosi zero). Natomiast poziom przecięcia dla każdych dwóch funkcji przynależności wynosi 0.5. Oznaczano, że nie istnieje wejściowa ostra wartość, która nie będzie mogła być porównana w fazie rozmywania z poprzednikiem reguły. Przez to nie jest możliwy przypadek nie odpalenia żadnej z reguł i w rezultacie brak zmiennej sterującej. Ponadto jak pokazano w artykule [26] dla symetrycznych funkcji przynależności istnieją optymalne wartości poziomu przecięcia i współczynnika przecięcia (odpowiednio 0.5 i 1). Dzięki takim parametrom otrzymuje się znacznie mniejsze przeregulowanie, szybszy czas narastania i mniejsze niedoregulowanie. Są to parametry najczęściej podawane w literaturze jako zalecane.

Po określeniu zbiorów przynależności należy przejść do kolejnego kroku kreowania regulatora rozmytego. Jest nim tworzenie zbioru reguł lingwistycznych reprezentujących relacje między zmiennymi systemu. To właśnie reguły są sercem całego regulatora.

W *Matlabie* służy do tego kolejna część edytora FIS - *Rule Editor*. Wpisywane reguły mogą być edytowane na kilka różnych sposobów. Rysunek 7.4 przedstawia sposób *verbose* wykorzystujący w zapisie słowa *if*, *and*, *or*, *then* - najbardziej zbliżające zapis do języka naturalnego.



Rys. 7.4 - Edytor zbioru reguł - Rule Editor

Pełny zbiór reguł dla regulatora FLC typu PI przedstawiony jest w tabeli 7.1.

Tabela 7.1 - Zbiór reguł dla regulatora FLC typu PI

e \ de	DU	ŚU	MU	Z	MD	ŚD	DD
DU	DU <sup>38</sup>	DU <sup>39</sup>	DU <sup>40</sup>	DU <sup>4</sup>	DU <sup>47</sup>	DU <sup>48</sup>	DU <sup>49</sup>
ŚU	DU <sup>35</sup>	DU <sup>36</sup>	DU <sup>37</sup>	ŚU <sup>5</sup>	DU <sup>44</sup>	DU <sup>45</sup>	DU <sup>46</sup>
MU	DU <sup>32</sup>	DU <sup>33</sup>	DU <sup>34</sup>	MU <sup>6</sup>	DU <sup>41</sup>	DU <sup>42</sup>	DU <sup>43</sup>
Z	DU <sup>11</sup>	ŚU <sup>12</sup>	MU <sup>13</sup>	Z <sup>7</sup>	MD <sup>10</sup>	ŚD <sup>9</sup>	DD <sup>8</sup>
MD	Z <sup>29</sup>	MD <sup>30</sup>	SD <sup>31</sup>	MD <sup>3</sup>	DD <sup>22</sup>	DD <sup>21</sup>	DD <sup>20</sup>
ŚD	MD <sup>26</sup>	SD <sup>27</sup>	DD <sup>28</sup>	SD <sup>2</sup>	DD <sup>19</sup>	DD <sup>18</sup>	DD <sup>17</sup>
DD	DD <sup>23</sup>	DD <sup>24</sup>	DD <sup>25</sup>	DD <sup>1</sup>	DD <sup>16</sup>	DD <sup>15</sup>	DD <sup>14</sup>

GRUPA 1	GRUPA 2	GRUPA 3	GRUPA 4	GRUPA 5

W tabeli 7.1 oprócz etykiet dla zmiennej *zmiana\_sterowania* podane są numery odpowiadające regułom w edytorze *Rule Editor*. Na zbiór składa się 49 reguł. Jest to zbiór zupełny, gdyż wykorzystane są wszystkie możliwości. Na początku projektowania regulatora zbiór składał się z 13 podstawowych reguł. Jednak w trakcie pracy okazało się, że zachowanie się obiektu w znacznym stopniu zależy od ilości reguł. Jednak przede wszystkim zależy od odpowiedniego doboru następników reguł (etykiet wyjścia). Jest to najistotniejszy element procesu strojenia.

Zbiór reguł można podzielić na pięć grup.

- **GRUPA 1** W tej grupie zarówno  $e$  jak i  $\Delta e$  są (dodatnie lub ujemne) małe albo zerowe. Oznacza to, że aktualna prędkość obrotowa wykazuje małe odchylenie od wartości zadanej. Jednak doświadczenia wykazały, że podanie dla tych reguł również małych zmian sterowania nie wpływa korzystnie na sterowanie. Znaczenie lepszy efekt uzyskano, gdy dla reguł 22, 34 i 41 wprowadzono dużą zmianę (DD). Bo skoro (dla przypadku reguły 22) prędkość jest za mała i w dodatku maleje w czasie (choć wolno) to rozsądnie jest jak najszybciej doprowadzić do narastania prędkości.
- **GRUPA 2** Dla tej grupy  $e$  jest albo bliskie wartości zadanej (Z, MD) albo znacznie poniżej tej wartości przy jednoczesnym odchodzeniu od wartości zadanej ( $\Delta e$  do-

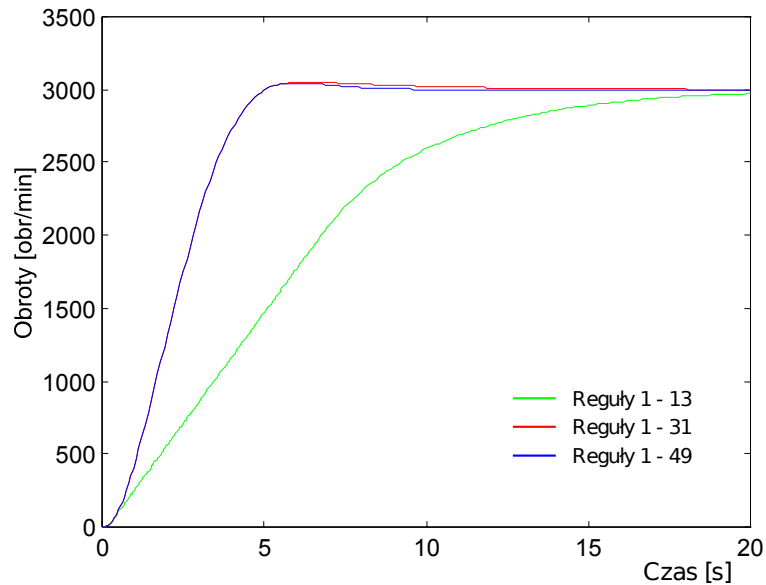
datnie). Tak więc dodatnia zmiana  $\Delta u$  zmierza do odwrócenia tego procesu i powoduje, że prędkość zaczyna się zbliżać do wartości zadanej.

- **GRUPA 3.** W tym przypadku prędkość jest zdecydowanie za duża ale za to maleje w czasie. Reguły odpowiadają za przyspieszenie do wartości zadanej.
- **GRUPA 4.** Dla tej grupy  $e$  jest dodatnie (prędkość jest mniejsza) i zmierza do wartości zadanej. Reguły te przyspieszają doświadczenie do wartości zadanej podobnie jak w grupie 3 z tym, że z drugiej strony.
- **GRUPA 5.** Dla tej grupy prędkość jest blisko wartości zadanej albo znacznie powyżej. Jednocześnie ponieważ  $\Delta e$  jest ujemne, wciąż rośnie. Tak więc ujemna zmiana  $\Delta u$  ma na celu odwrócenie tego procesu i spowodowanie aby prędkość zaczęła się zbliżać do wartości zadanej.

Dostrajanie zbioru reguł jest procesem bardzo czasochłonnym. Powodem tego jest fakt, że zmiana etykiety dla jednej z reguł pociąga za sobą najczęściej zmianę w jednej lub w wielu innych regułach. Radykalna zmiana w jednej regule zazwyczaj pogarsza sterowanie zamiast go polepszyć. Dlatego zmiany należy przeprowadzać w kilku regułach jednocześnie. Wiąże się to z tym, że w jednej chwili odpala najczęściej kilka reguł. Dlatego należy je tak dostroić aby wspólnie dawały jak najlepszy wynik a nie przypadkiem likwidowały się nawzajem. Podczas strojenia reguł zastosowano metodę „zdroworozsądkową”. Jest ona jedną z najczęściej stosowanych jeżeli nie ma informacji o obiekcie pochodzącej od eksperta. Niemniej jednak jest ona bardzo skuteczna, choć wymaga przeprowadzenia wielu prób.

W przypadku przyłożonego do obiektu zakłócenia (obciążenia) najistotniejszą rolę miały reguły 26 - 31. Odpowiednie ich dostrojenie pozwoliło na uzyskanie najbardziej optymalnych rezultatów.

Wpływ liczby reguł na zachowanie się silnika przy rozruchu przedstawia rysunek 7.5. Jednocześnie ilustruje zachowanie się silnika przy rozruchu. Przed osiągnięciem wartości zadanej obrotów zauważalne jest lekkie przesterowanie. Jest to właśnie efekt doboru reguł 26 - 31. Jednak dzięki temu uzyskuje się lepsze wyniki i szybszy czas narastania. Dokładniej zostanie to opisane w kolejnym rozdziale. Najlepsze zachowanie się obiektu występuje przy wszystkich 49 regułach. Wtedy regulator w najkrótszym czasie doprowadza silnik do obrotów znamionowych.

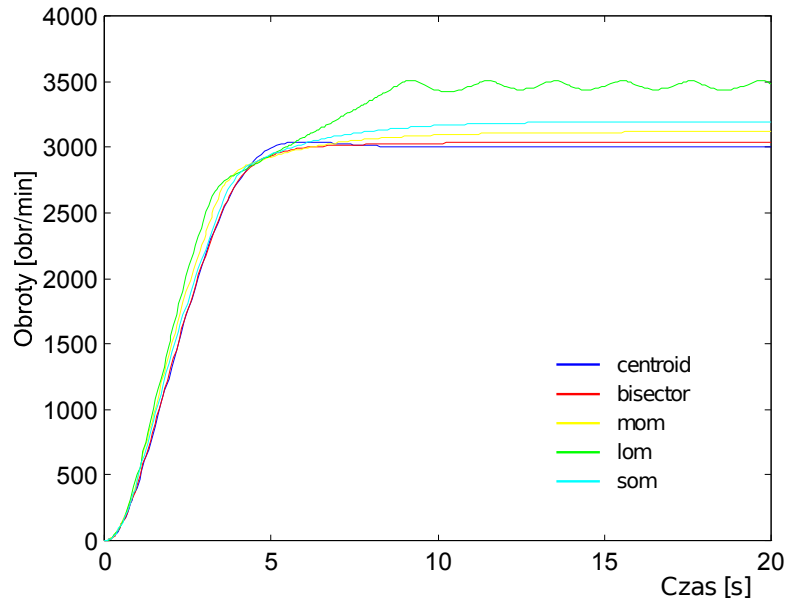


Rys. 7.5 - Wpływu reguła na obroty silnika przy rozruchu dla FLC typu PI

Należy tutaj wspomnieć o innych parametrach regulatora ustawianych w edytorze *FIS* (oprócz funkcji przynależności i zbioru reguł) a wchodzących w skład procesu strojenia. Wybór rodzaju logicznego (*min* lub *prod*) nie wpływa w zauważalny czy też istotny sposób na zachowanie się układu. Podobnie sprawa wygląda dla logicznego LUB (*max* lub *probo*), rodzaju implikacji (*min* lub *prod*), czy też agregacji (*max*, *sum*, *probo*).

Natomiast wybór odpowiedniej metody wyostrzania jest już istotny dla zachowania się sterowanego obiektu. Rysunek 7.6 pokazuje zależności. Najlepszą jest metoda *centroid* (*centroid of area method* - metoda środka obszaru - wzór 3.10). Kolejne metody to: *bisector* (*bisector of area method* - metoda symetralnej obszaru), *mom* (*mean of maximum method* - metoda średnia z największych - wzór 3.11), *som* (*smallest of maximum method* - metoda najmniejszy z największych), *lom* (*largest of maximum method* - metoda największy z największych). Nie trudno zauważyć, że zdecydowanie najgorszą metodą jest *lom*.





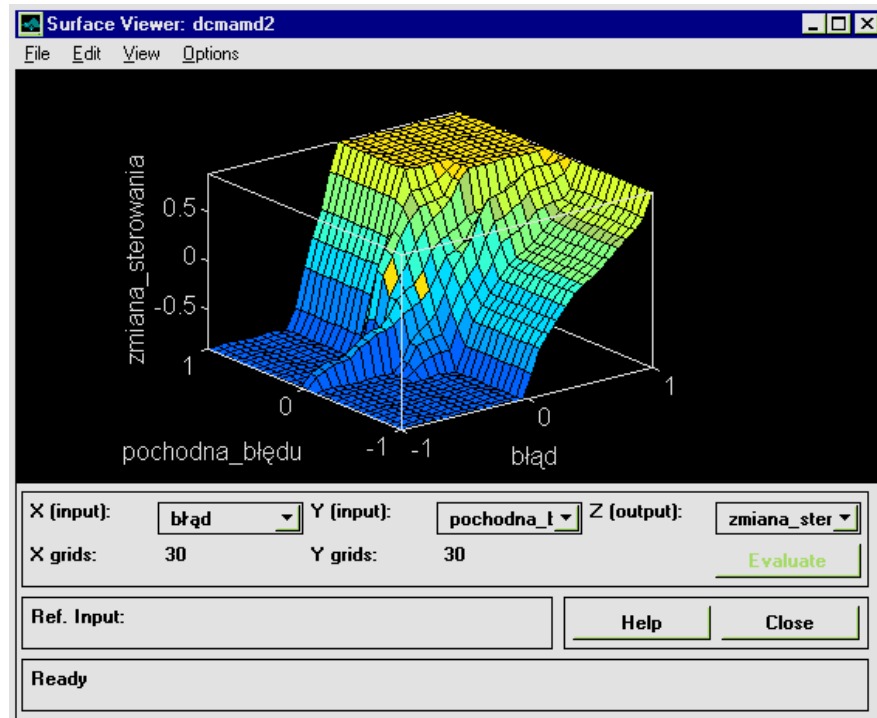
Rys. 7.6 - Wpływ metody wyostrzania na stabilizację obrotów dla FLC typu PI

Bardzo przydatnym elementem w procesie strojenia regulatora jest jeden inny z elementów edytora FIS. Jest to *Rule Viewer*, umożliwiający graficzną interpretację zależności regulatora FLC. Wyświetla on na ekranie części składowe systemu. Wpisując w polu *Input* wartości ostrej wejści (lub przesuwając pionowymi żółtymi liniami) można obserwować, które reguły odpalają i jaka wartość ostry wyjścia jest obliczana.

Innym elementem edytora FIS jest *Surface Viewer*. Wyświetla on zależność wyjścia od składowych wejściowych. Dla obiektu drugiego rzędu tworzona jest powierzchnia pokazana na rysunku 7.8.



Rys. 7.7 - RuleViewer



Rys. 7.8 - Powierzchnia zależności zmiennej  $zmiana\_sterowania$  od zmiennych wejściowych dla FLC typu PI

## Konstruowanie regulatora FLC typu PD

Jeżeli zmienną wyjściową regulatora FLC nie jest zmiana sterowania  $\Delta u$ , ale sama wartość sterowania  $u$ , to otrzymujemy FLC typu alternatywnego, który realizuje prawo sterowania:  $u(k) = F(e(k), \Delta e(k))$ . Reguły tego FLC mają jako wejścia uchyb  $e$  oraz jego zmianę  $\Delta e$ .

Równanie regulatora PD ma postać:

$$u = K_P \cdot e + K_D \cdot \dot{e} \quad (7.0)$$

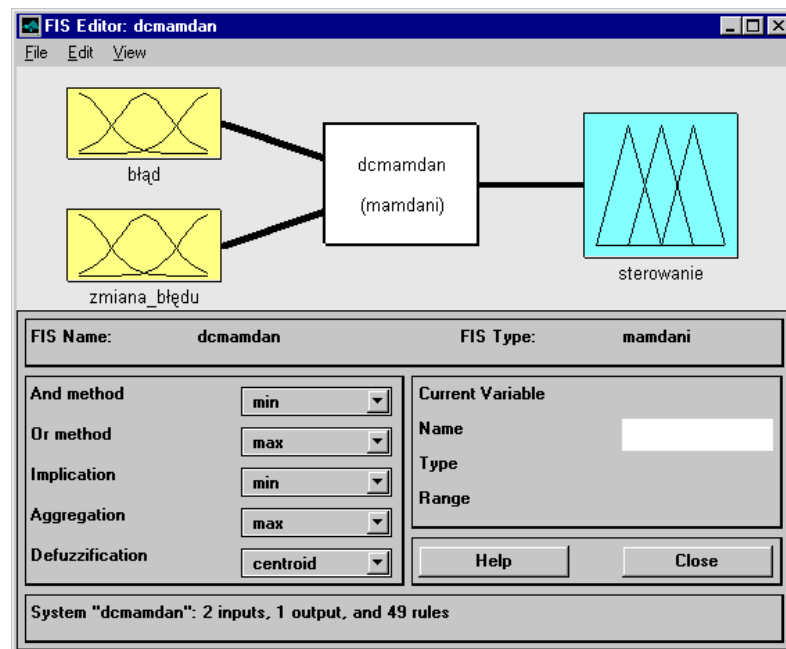
gdzie  $K_P$  i  $K_D$  są współczynnikami członów proporcjonalnego i różniczkującego regulatora PD.

Regulator FLC typu PD składa się z reguł, przy czym dla każdej z tych reguł jest dany symboliczny opis:

**JEŻELI**  $e(k)$  jest <symbol własności> I  $\Delta e(k)$  jest <symbol własności>

**TO**  $u(k)$  jest <symbol własności>

Zaprojektowany regulator FLC typu PD jest również regulatorem Mamdaniego. Istnieją w nim jednak dość istotne różnice w porównaniu do FLC typu PI.



Rys. 7.9 - Ogólny wygląd edytora FIS dla zmiennej dcmamdan

Przedewszystkim określone są inne wejścia i inne wyjście. W tym przypadku wejściami są  $błąd$  i  $zmiana\_błądu$ , natomiast wyjściem  $sterowanie$ . Początkowo jednym z wejść była pochodna błędu, tak jak ma to miejsce w przypadku FLC typu PI. (uzyskano

to wprowadzając do systemu bloček pamięci zamiast bločku różniczkowania - rysunek 6.3). Jednak wprowadzenie zmiany błęd znacznie poprawiło sterowanie. Przebiegi stały się znacznie łagodniejsze. Z rysunku 7.9 można odczytać w jaki sposób zdefiniowano operatory I, LUB, rodzaj implikacji, metodę agregacji czy metodę wyostrzenia.

Czynniki skalujące odpowiedzialne za przeprowadzanie procesu normalizacji i denormalizacji są takie same. Tutaj czynniki skalujące jednak częściowo odgrywają inną rolę. Ich zadaniem jest nie tylko przekształcanie wartości ostrych wejść w dziedzinie znormalizowane. Ważniejszą rolę jest uzyskanie odpowiedniej wartości wyjściowej dla wprowadzonej wartości zadanej obrotów. Chodzi tutaj o rozróżnienie sterowania dla dwóch przypadków o tych samych wejściach ale innej wartości zadanej. Przykładowo dla żądanych obrotów wartości 3000 i 2000 błęd i jego zmian w czasie wynoszą 500. Dla regulatora jest to identyczna sytuacja. Dlatego podaje takie same sterowanie w obu przypadkach. Dopiero czynnik skalujący w bloku denormalizacji podaje odpowiednią wartość napięcia. Jeżeli nie byłoby bloku denormalizacji nie byłaby możliwa nastawa różnych wartości zadanych.

Przestrzenie, na których określone są zmienne wejścia i wyjścia, również są odmienną porównaniu do FLC typu PI. Przestrzenia dla zmiennych wejścia jest, podobnie jak w przypadku regulatora FLC typu PI, przedział  $[-1, 1]$ . Jednak w przypadku zmiennej wyjściowej *sterowanie* jest to już przedział  $[0, 1.3]$ . Zmienna *sterowanie* w przypadku tego regulatora to bezpośrednio napięcie podane na wejście obiektu - sterowanego silnika (oczywiście pomijając blok denormalizacji). Początkowo był to przedział  $[0, 2]$ , co odpowiadałoby maksymalnemu napięciu zasilającemu ok. 440V. Przy podanym takim napięciu podczas rozruchu prąd w mika wzrastał do wartości powyżej 35A. Odpowiadało to około dwukrotnej wartości maksymalnego prądu rozruchowego przyłożeniu napięcia znamionowego. Tak więc jest to wartość nie do przyjęcia. Jednak należy zauważyć, że dla takich parametrów zmiennej *sterowanie*, osiągnęto mocno zadowalające wyniki prób doświadczeń zarówno bez jak i z obciążeniem.

Przedział zaczyna się od zera gdyż założono, że minimalne napięcie zasilające może wynosić 0V (nie dopuszcza się wartości napięcia z przeciwnym znakiem). Natomiast wartość 1.3 odpowiada maksymalnemu napięciu około 280V. Wartość napięcia nie jest tak istotna, gdyż nie jest ona szkodliwa dla samego silnika. Istotny jest natomiast prąd. Przy wartości wyjściowej około 1.3 prąd nie przekracza założonej w bloku *Ograniczenia* (rysunek 6.3) dwudziestoprocentowej rezerwy.

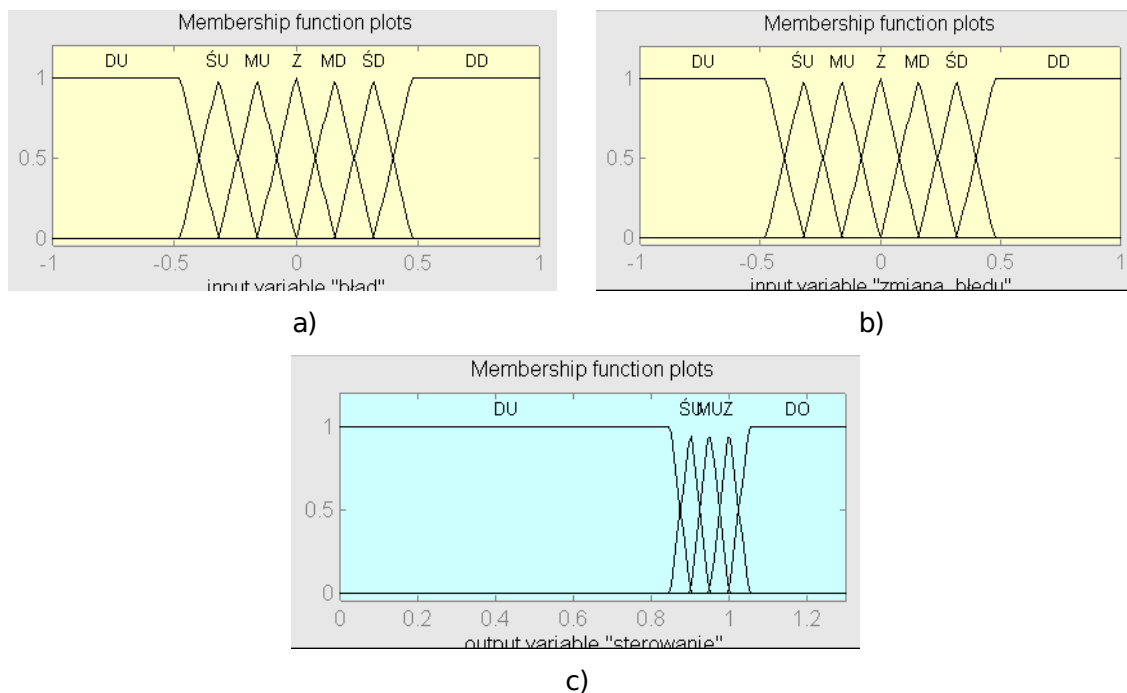
Etykiety lingwistyczne dla zmiennych wejścia są takie same jak dla FLC typu PI. (DU, ŚU, MU, Z, MD, ŚD, DD). Natomiast w przypadku zmiennej sterowanie określono następujące etykiety:

1. DU - **D**uży **U**jemny,
2. ŚU - **Ś**redni **U**jemny,
3. MU - **M**ały **U**jemny,
4. Z - **Z**erowy,
5. DO - **D**odatni.

Przez zawężenie zakresu z  $[0,2]$  do  $[0,1.3]$  zrezygnowano z trzech etykiet ze znakiem dodatnim. Ograniczono się jedynie do jednej (DO). Nie zmieniło to właściwości sterowania.

Znak dla zmiennych *błąd* i *zmiana błędu* interpretuje się w ten sam sposób jak w przypadku FLC typu PI (punkt 7.1). Nazwy etykiet dla zmiennej sterowanie mogą być trochę mylące. Ujemne etykiety dla sterowania  $u$  (DU, ŚU, MU) oznaczają nie ujemne wartości napięcia lecz to, że wyjście przyjmuje wartości poniżej nominalnego napięcia. Omawiana sytuacja dotyczy oczywiście przypadku, gdy wartość zadana równa jest obrotom nominalnym - 3000 obr/min. W innym przypadku po procesie denormalizacji wartość napięcia wchodzącego na obiekt sterowany będzie inna - niższa (można powiedzieć, że zmieni się wartość napięcia znamionowego). Etykieta Z oznacza, że wyjście jest bliskie wartości nominalnej. Dodatnia etykieta  $u$  oznacza, że sterowanie przyjmuje wartości powyżej znamionowej.

Rozkład zmiennych przynależności w ich przestrzeniach pokazany jest na rysunku 7.10. Rozkład funkcji przynależności dla zmiennej sterowanie taki jak na rysunku okazał się najbardziej korzystny dla sterowania. Sterowanie dla równomiernie rozłożonych trzech funkcji powyżej wartości 1 (MD, ŚD, DD), nie było skuteczniejsze. Dlatego pozostała tylko jedna trapezoidalna funkcja przynależności - DO. Idąc tą drogą próbowano wprowadzić jedną funkcję przynależności dla wartości poniżej 1. Ograniczyłyby to etykiety do trzech: U (ujemne), Z (zerowe), D (dodatnie), w tym dwie trapezoidalne a jedną trójkątną. Jednak wyniki jakie otrzymano okazały się znacznie gorsze w porównaniu z tymi uzyskanymi dla 5 etykiet, które ostatecznie zostały.



Rys. 7.10- Zbiory funkcji przynależności dla FLC typu PD: a) bład, b) zmiana błędu, c) sterowanie

Funkcje przynależności dla zmiennej *sterowanie* mają współczynnik przecięcia równy jeden. Poziom przecięcia dla każdego dwóch wynosi 0.5. Funkcje dla etykiet ŚU, MU, Z są symetryczne. Ścisniętych funkcji przynależności pozwoliło na optymalne sterowanie. Odbyło się to oczywiście w drodze doświadczeń.

Podobnie w drodze przeprowadzonych testów uzyskano ostateczny wygląd i zakres funkcji przynależności dla obu zmiennych wejść, których współczynnik przecięcia i poziomy przecięcia jest taki sam jak dla zmiennej wyjściowej.

Po określeniu zbiorów funkcji przynależności został utworzony odpowiedni zbiór reguł. Przedstawiony jest on w tabeli 7.2. Na pełny zbiór reguł składa się 49 reguł. Jest to zupełny zbiór reguł. W przypadku tego typu regulatora ilość reguł nie gra takiej roli jak w przypadku regulatora FLC typu PI. Wiąże się to między innymi z faktem obniżenia ilości etykiet dla zmiennej *sterowanie* z siedmiu do pięciu. Jedna etykieta zwiększająca szybkość narastania prędkości do wartości zadanej (DO) to nie za duży wybór dla regulatora. Niemniej jednak zostawiono wszystkie 49 reguł.

Tabela 7.2 - Zbiór reguł dla regulatora FLC typu PD

<b>de</b>	<b>DU</b>	<b>ŚU</b>	<b>MU</b>	<b>Z</b>	<b>MD</b>	<b>ŚD</b>	<b>DD</b>
-----------	-----------	-----------	-----------	----------	-----------	-----------	-----------

<b>e</b>							
<b>DU</b>	DU <sup>38</sup>	DU <sup>39</sup>	DU <sup>40</sup>	DU <sup>4</sup>	DU <sup>47</sup>	DU <sup>48</sup>	DU <sup>49</sup>
<b>ŚU</b>	DU <sup>35</sup>	DU <sup>36</sup>	DU <sup>37</sup>	ŚU <sup>5</sup>	ŚU <sup>44</sup>	ŚU <sup>45</sup>	MU <sup>46</sup>
<b>MU</b>	DU <sup>32</sup>	DU <sup>33</sup>	DU <sup>34</sup>	MU <sup>6</sup>	DU <sup>41</sup>	MU <sup>42</sup>	MU <sup>43</sup>
<b>Z</b>	DU <sup>11</sup>	ŚU <sup>12</sup>	MU <sup>13</sup>	<b>Z</b> <sup>7</sup>	DO <sup>10</sup>	DO <sup>9</sup>	DO <sup>8</sup>
<b>MD</b>	DO <sup>29</sup>	DO <sup>30</sup>	DO <sup>31</sup>	DO <sup>3</sup>	DO <sup>22</sup>	DO <sup>21</sup>	DO <sup>20</sup>
<b>SD</b>	DO <sup>26</sup>	DO <sup>27</sup>	DO <sup>28</sup>	DO <sup>2</sup>	DO <sup>19</sup>	DO <sup>18</sup>	DO <sup>17</sup>
<b>DD</b>	DO <sup>23</sup>	DO <sup>24</sup>	DO <sup>25</sup>	DO <sup>1</sup>	DO <sup>16</sup>	DO <sup>15</sup>	DO <sup>14</sup>

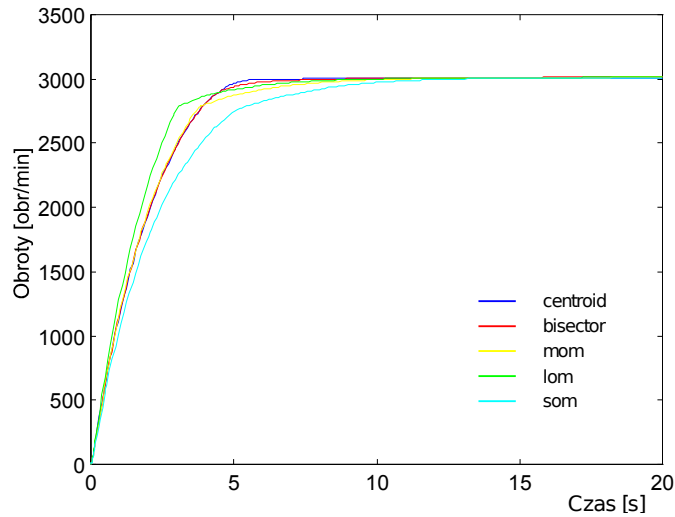
GRUPA 1	GRUPA 2	GRUPA 3
---------	---------	---------

Zbiór reguł dla tego przypadku można podzielić na trzy grupy.

- **GRUPA 1** Do tej grupy należy tylko jedna reguła. Jeżeli prędkość jest bliska wartości zadanej nie zmienia się w czasie to podane sterowanie ma mieć wartość nominalną.
- **GRUPA 2** Do tej grupy należą wszystkie reguły dla których sterowanie przyjmuje wartość DO (jest większe od wartości nominalnej). Ma to miejsce gdy aktualna prędkość obrotowa jest niższa od wartości zadanej, bez względu na to czy rośnie czy też maleje w czasie.
- **GRUPA 3.** Ostatnia grupa to reguły, dla których napięcie przyjmuje wartości poniżej nominalnego. Dotyczy to sytuacji gdy bieżące obroty są większe od wartości zadanej i rosną bądź maleją w czasie. Ta grupa reguł wymagała dokładniejszego dostrajania. Ze względu na większą ilość etykiet (DU, ŚU, MU) było większe pole manewru.

Proces strojenia reguł dla regulatora FLC typu PD przebiegał znacznie krócej niż w przypadku PI. Lecz i tutaj zastosowano metodę „zdroworozsądkową”.

Inne parametry regulatora ustawiane w edytorze FIS właściwie nie mają wpływu na zachowanie się regulatora. Chodzi tutaj o rodzaj logicznego I, LUB, zastosowanej implikacji, agregacji. Jedyne różnice widoczne są w metodzie wyostrażania (podobnie jak ma to miejsce w FLC typu PI).

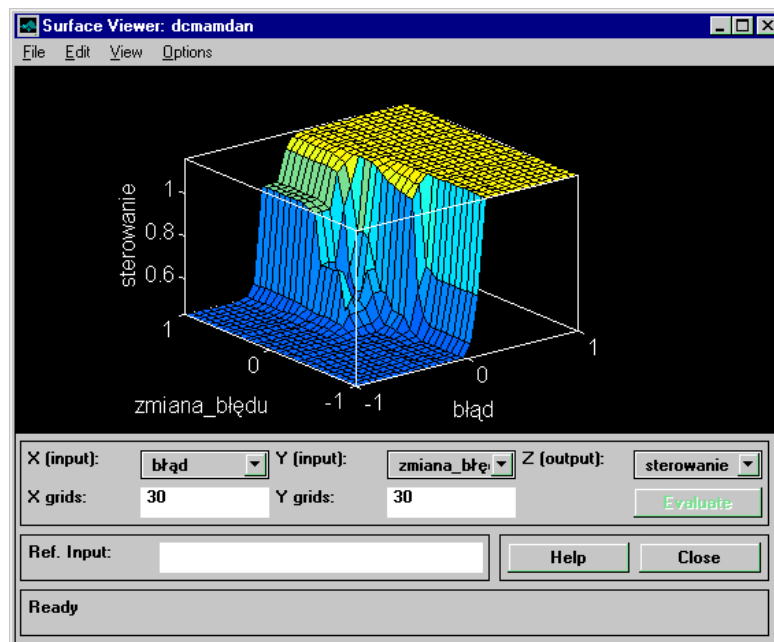


Rys. 7.11 - Wpływ metody wyostrzania na zachowanie się obiektu dla FLC typu PD

Jednak jak widać na rysunku 7.11 nie są to różnice aż tak mocne jak dla FLC typu PI. Niemniej jednak najkorzystniejszą jest metoda *centroid*

Podczas strojenie regulatora również korzystano z bardzo pomocnego edytora *RuleViewer*:

Rysunek 7.12 przedstawia wygenerowaną powierzchnię przez edytor *Surf Viewer* i przedstawiającą zależność zmiennej sterowanie od zmiennych wejściowych. Porównując tę powierzchnię z rysunkiem 7.8 widać różnice szczególnie w części środkowej.

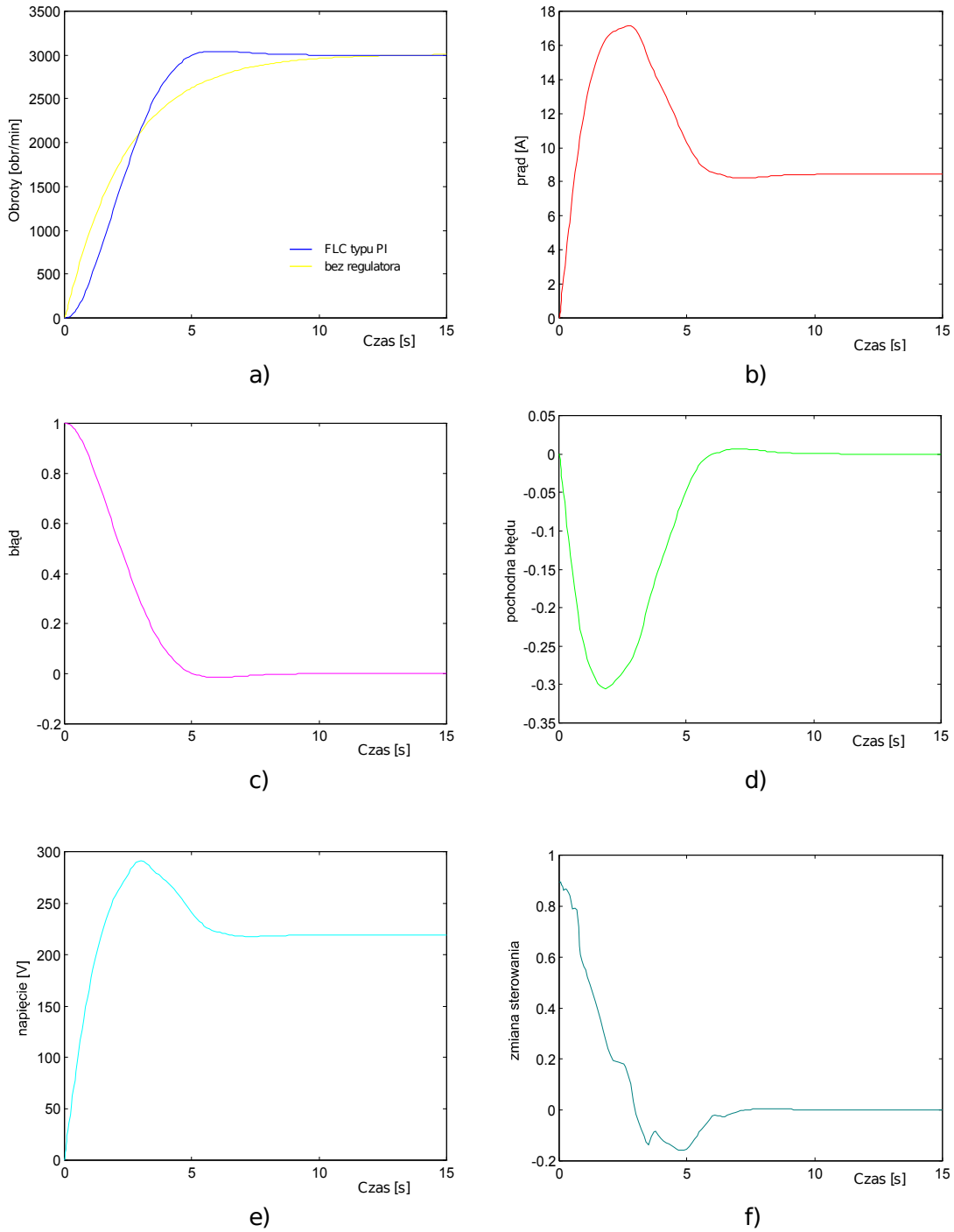


Rys. 7.12 - Zależność zmiennej wyjściowej od zmiennych wejściowych dla FLC typu PD



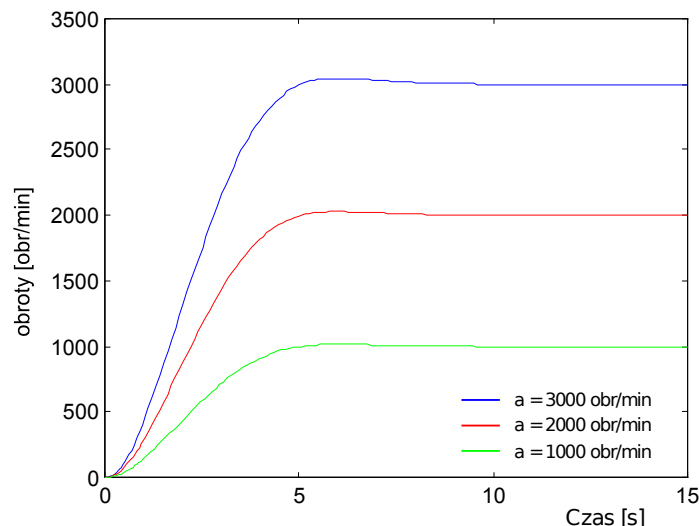
## Testowanie zaprojektowanego regulatora rozmytego

### Regulator FLC typu PI



Rys. 8.1 - Przebiegi dla prac silnika bez obciążenia przy sterowaniu FLC typu PI: a) prędkość, b) prąd, c) błąd, d) pochodna błędów, e) napięcie, f) zmiana sterowania

Na rysunku 8.1 pokazano przebiegi uzyskane w przeprowadzonej symulacji rozruchu silnika bez obciążenia, dla wartości zadanej obrotów  $a=3000$  obr/min. Rysunek a) dowodzi, iż stabilizacja obrotów na zamierzonym poziomie następuje dużo wcześniej niż po podłączeniu obiektu do napięcia znamionowego. Występuje lekkie przesterowanie, które jest efektem odpowiedniego doboru następników reguł, opisanych już wcześniej. Prąd narasta stopniowo ale nie przekracza wartości 17A, mimo wartości napięcia sięgającej 280V. Po ustabilizowaniu się obrotów wartość prądu ustala się na poziomie 8.4A, a wartość napięcia na 220V, co odpowiada wartościom znamionowym. Przebiegi zmiennych: błąd, pochodną błędu i zmian sterowania w czasie schodzą do zera. Innymi słowy dzięki regulatorowi FLC uzyskano znacznie korzystniejszy rozruch, nie przekraczając przy tym przyjętych ograniczeń w symulacji ograniczeń.

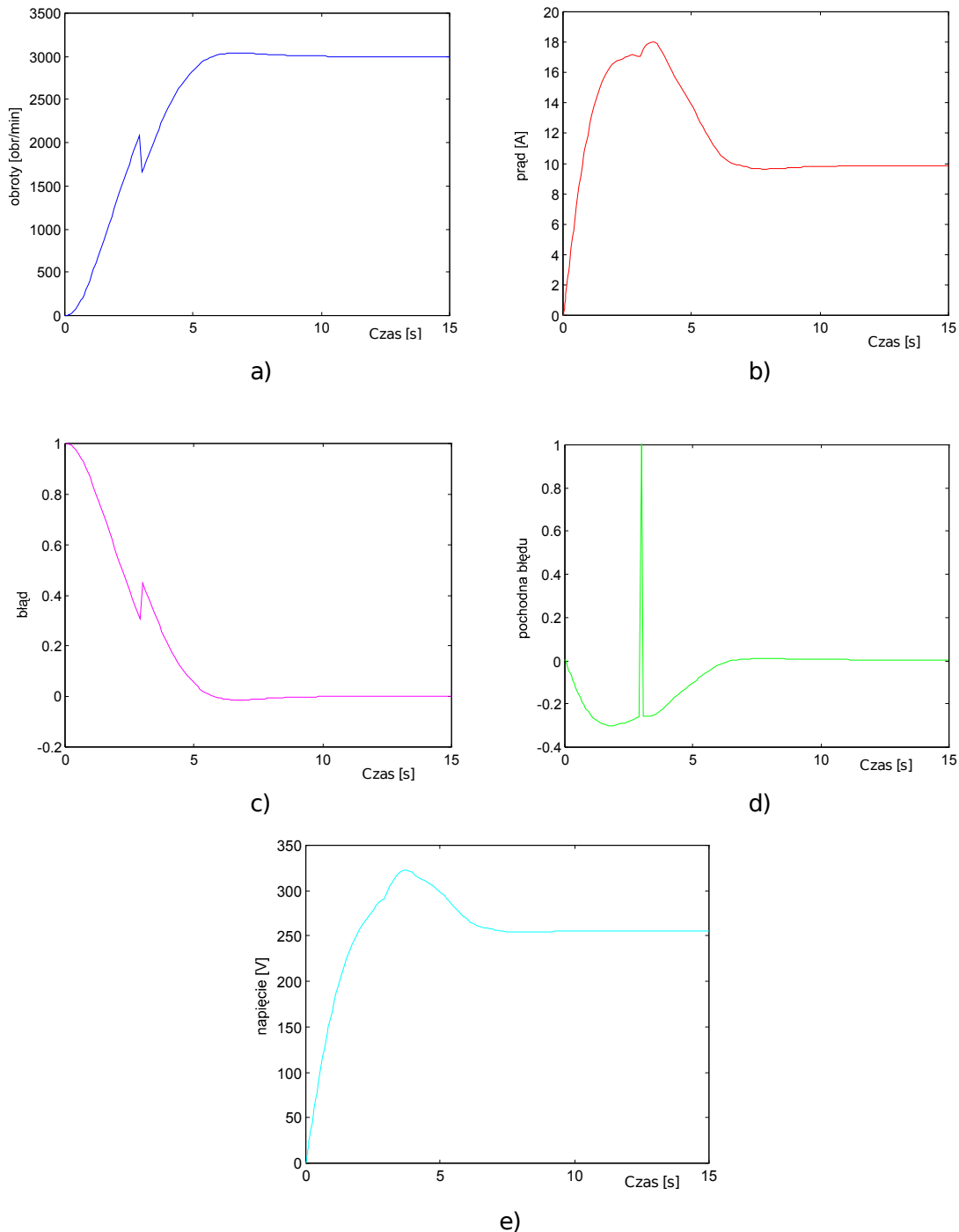


Rys. 8.2 - Przebieg rozruchu silnika dla różnych wartości zadanych obrotów przy sterowaniu FLC typu PI.

Rysunek 8.2 przedstawia rozruch silnika dla różnych nastaw wartości zadanych. Widoczny jest sposób działania skalowania wejścia i wyjścia. Dla różnych  $a$  prędkość stabilizuje się na zadanym poziomie przy takiej samej zmianie sterowania podawanej przez regulator. Oczywiście dla niższych nastaw prędkości wartości prądu rozruchu i podawanego napięcia sterowania są również niższe.

Rysunek 8.3 przedstawia wpływ przyłożonego w chwili  $t = 3$  s obciążenia stałego na wyjście obiektu. Zakłócenie wprowadzono jeszcze podczas rozruchu silnika. Obciążenie spowodowało spadek prędkości obrotowej do wartości 2500 obr/min. Wywołało to wzrost błędu, a co za tym idzie dodatnią wartość jego pochodnej w czasie (prędkość zmalała). Regulator bez większego problemu poradził sobie z tą sytuacją i prowa

dzając dodatnią zmianę sterowania - wzrost napięcia twornika. Prąd twornika nieznacznie wzrósł ale nie przekroczył założonego ograniczenia. Po osiągnięciu przez silnik wartości zadanej obrotów prąd i napięcie twornika są wyższe od nominalnych co jest zgodne z oczekiwaniami.



Rys. 8.3 - Przebiegi pracy silnika dla obciążenia skokiem jednostkowym w chwili  $t=3$  przy sterowaniu FLC typu PI: a) prędkość, b) prąd, c) błąd, d) pochodna błęd, e) napięcie

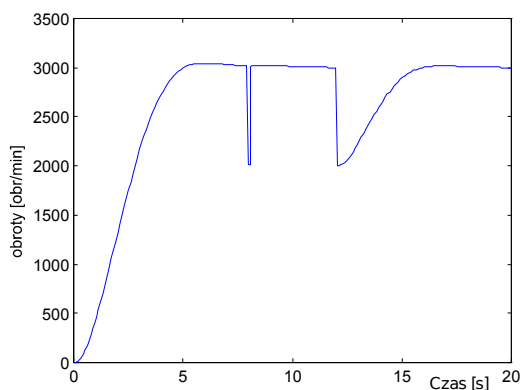
Rysunek 8.4 obrazuje wyniki symulacji, w której na wyjście przyłożono dwa różne obciążenia:

- pierwsze w chwili  $t = 8\text{s}$  - impuls wartości  $1/3$  momentu (fala prostokątna o czasie trwania równym  $0.1\text{s}$ ),
- drugie w chwili  $t = 12\text{s}$  - stałe obciążenie o takiej samej wartości (skok jednostkowy).

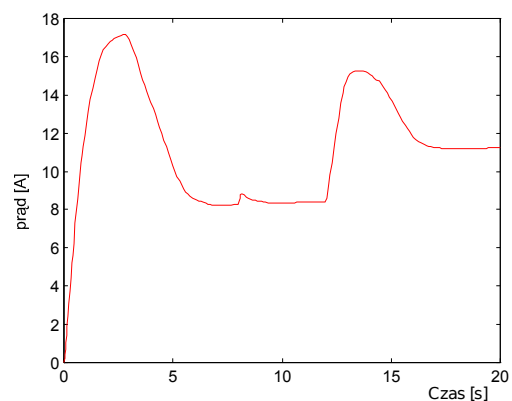
Na pierwsze zakłócenie regulator odpowiedział bardzo szybko. Dodatnie wartości błęd i pochodnej błędu spowodowały reakcję w formie dodatniej zmiany sterowania (wzrost napięcia). Po ustaniu impulsu ujemna wartość pochodnej zaowocowała ujemną zmianą sterowania (nagły spadek napięcia). Prędkość powróciła do wartości zadanej.

W przypadku drugiego zakłócenia regulator w momencie pojawienia się niezerowych wartości wejściowych (dodatni błąd i dodatnia pochodna błędu) zareagował odpowiednią (dodatnią) zmianą sterowania. W miarę osiągania przez wejścia wartości zerowych zmian sterowania była coraz mniejsza. Ostatecznie doprowadziło to do ustalenia prędkości obrotowej na poziomie  $3000\text{ obr/min}$ . Oczywiście aby taka sytuacja była możliwa zarówno napięcie jak i prąd twornika mają wartości wyższe od znamionowych.

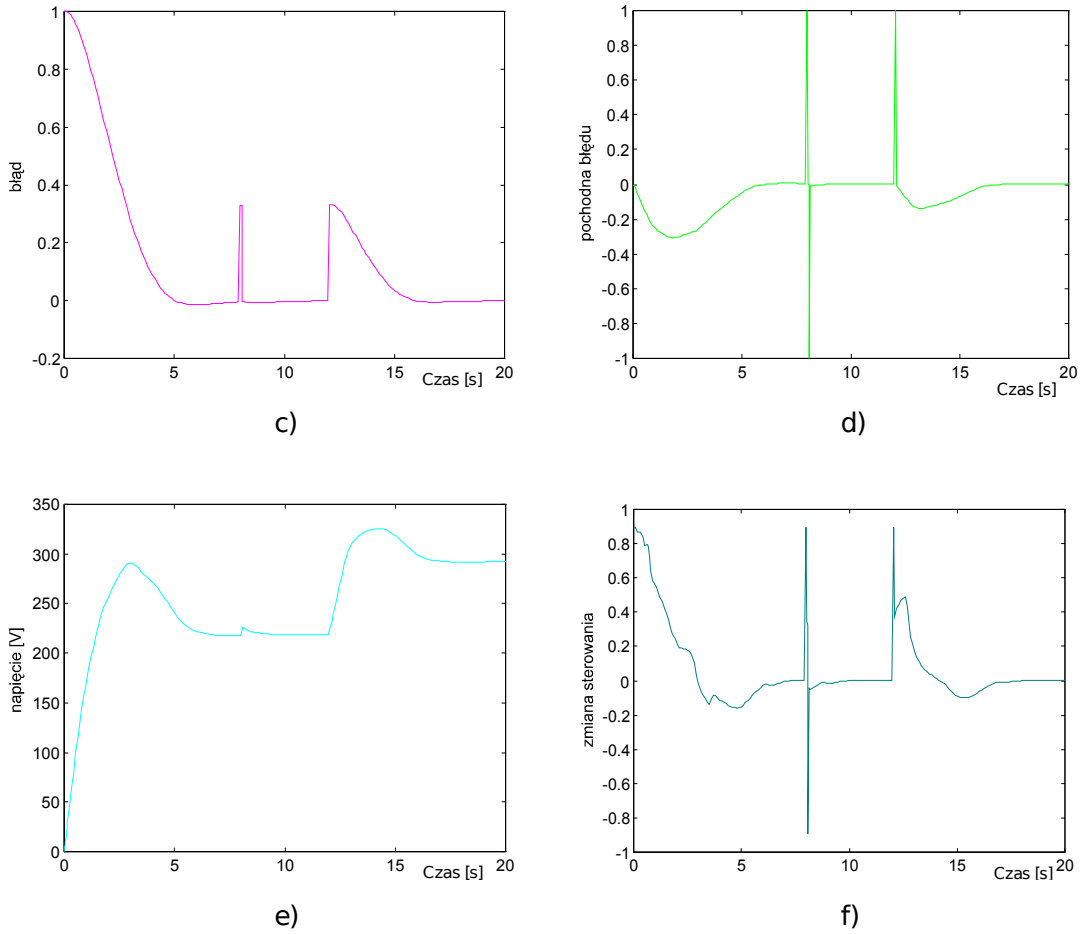
Dosyć ciekawym zakłóceniem jest to, które obrazuje rysunek 8.5 - po czasie  $10$  sekund zostaje na wyjście podana fala prostokątna trwająca  $3$  sekundy. Efektem takiego obciążenia jest obniżenie prędkości obrotowej do  $2500\text{ obr/min}$ . Przedstawione przebiegi zgadzają się z rysunkami przedstawionymi w artykule [27]. W chwili odłączenia obciążenia, czyli po  $3$  sekundach regulator odpowiednim sterowaniem prawie całkowicie wyeliminował odchyłkę prędkości od wartości zadanej (napięcie było wyższe od nominalnego). Dlatego też przy wysokiej wartości napięcia obroty nagle wzrosły, a prąd zmalał (silnik zaczął pracować bez obciążenia). Po chwili regulator ustalił optymalne sterowanie i nastąpiła stabilizacja obrotów na poziomie równym wartości nominalnej.



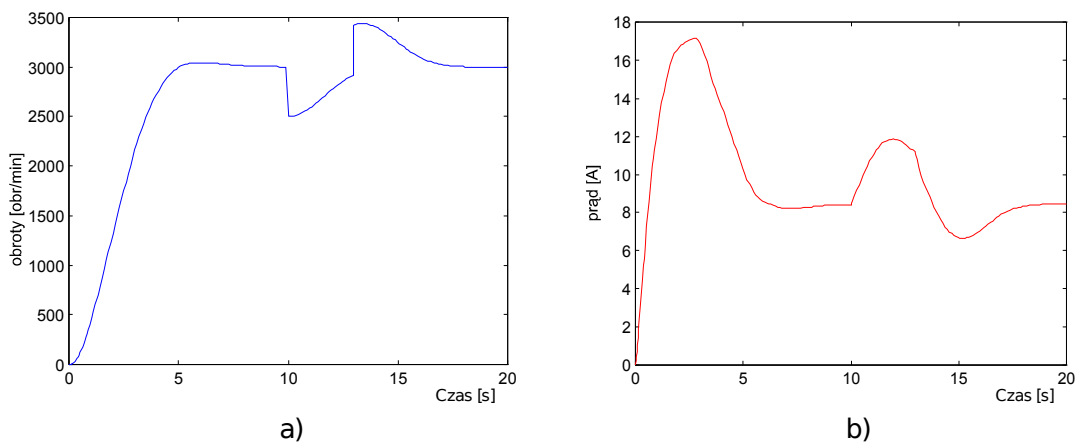
a)

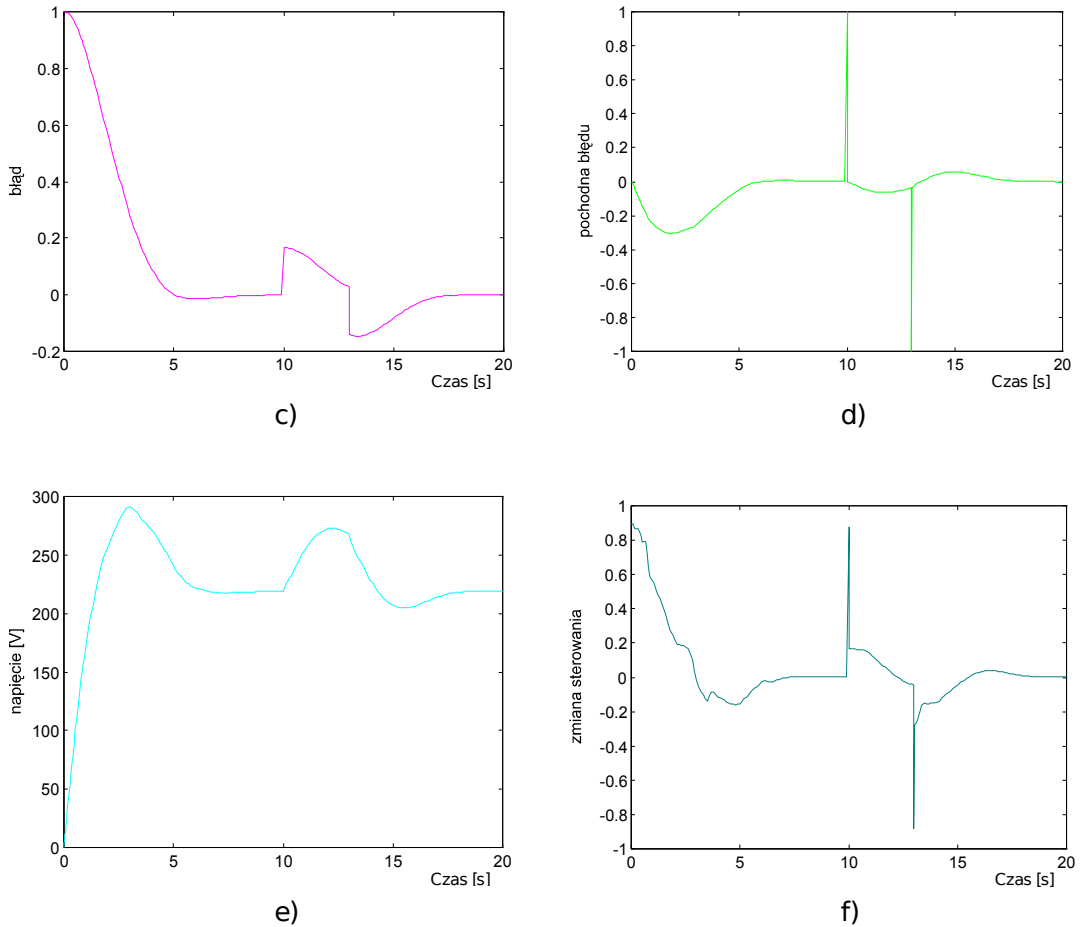


b)



Rys. 8.4 - Przebiegi pracy silnika dla przyłożonego obciążenia skokiem jednostkowymi falą prostokątną przy sterowaniu FLC typu PI: a) prędkość, b) prąd, c) błąd, d) pochodna błędu, e) napięcie, f) zmiana sterowania





Rys. 8.5 - Przebiegi pracy silnika dla obciążenia falą prostokątną przy sterowaniu FLC typu PI: a) prędkość, b) prąd, c) błąd, d) pochodna błęd, e) napięcie, f) zmian sterowania

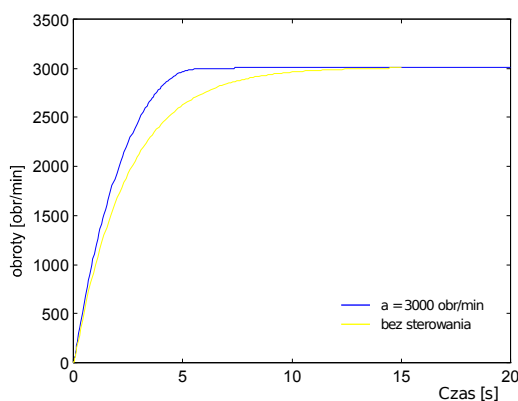
### Regulator FLC typu PD

Rysunek 8.6 przedstawia wyniki przeprowadzonej symulacji rozruchu silnika przy zastosowaniu regulacji FLC typu PD. Już na pierwszy rzut oka widoczne jest, że jej działanie jest odmienne od regulacji FLC typu PI. Wartość zadana obrotów uzyskiwana jest bardzo szybko. Od samego początku, kiedy błąd jest duży, napięcie podawane na twornik silnika jest wysokie. Ma to miejsce od pierwszego kroku przeprowadzanej symulacji (nie narasta stopniowo tak jak ma to miejsce w przypadku FLC typu PI - różnica wynika z charakteru sterowania). Później, w miarę zmniejszania się błędów i osiągnięcia przez jego zmiany wartości bliskich zera, napięcie zaczyna zbliżać się do

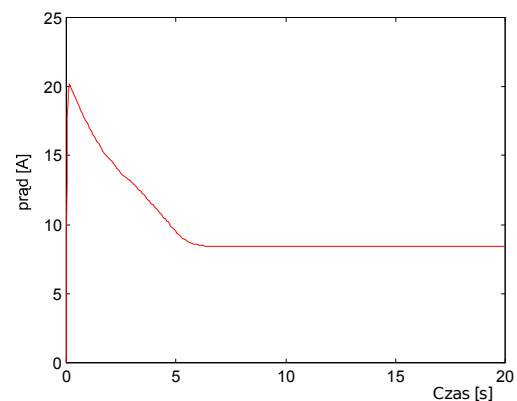
wartości nominalnej aby ostatecznie ją osiągnąć. Analogiczna jest sytuacja dla prądu twornika. Należy zauważyć, że przebieg prądu jest bardzo zbliżony do przebiegu uzyskiwanego przy rozruchu silnika podłączonego bezpośrednio do napięcia znamionowego.

Rysunek 8.7 natomiast przedstawia rozruch silnika dla różnych wartości zadanych. Podobnie jak w przypadku zastosowania regulatora FLC typu PI ma to miejsce dla takiego samego sygnału sterowania podawanego przez regulator. We wszystkich przypadkach zmiennej wejściowej wyjściowej regulator miały takie same wartości. Dopiero po denormalizacji sygnału wyjściowego regulatora otrzymano odpowiednią wartość napięcia twornika, wchodzącego na obiekt silnika.

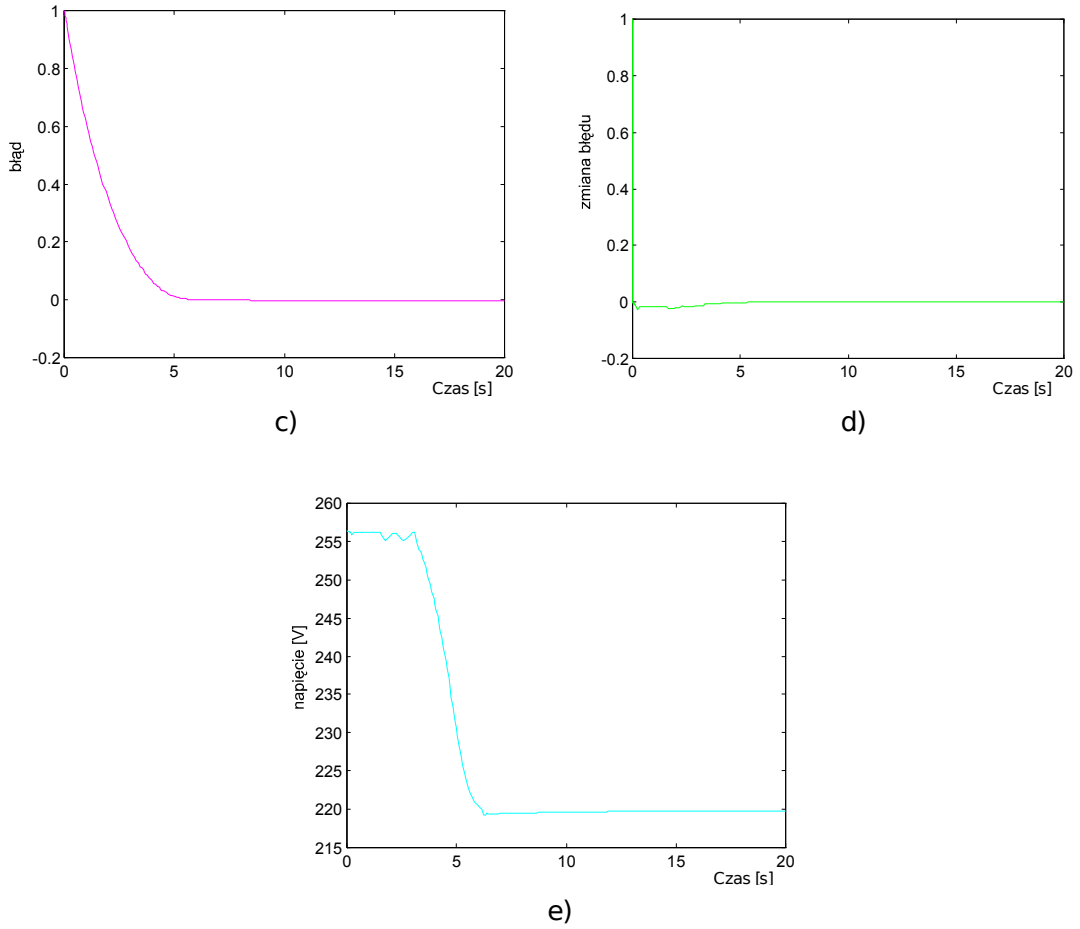
Rysunek 8.8 przedstawia wyniki symulacji sterowania obrotami silnika po przyłożeniu zakłócenia w formie skoku jednostkowego po czasie  $t = 3s$  (jeszcze w trakcie trwania rozruchu). Efekt takiego zakłócenia to obniżenie prędkości obrotowej o 500 obr/min. W całym czasie trwania symulacji błąd nie osiągnął wartości zerowej. Ustalił się (zmiana błędu zerowa) na poziomie około 0.05. Dlatego też silnik nie osiągnął wartości zadanej obrotów. Zakłócenie nie wywarło większego wpływu na sam wygląd przebiegu prądu twornika. Ale oczywiście jego wartość końcową uzyskaną w symulacji jest wyższa od wartości nominalnej (podobnie maksymalną wartość przy rozruchu). Ta sama sytuacja dotyczy wartości napięcia twornika. W chwili  $t = 3s$  nie ma widocznych odchyłów (wiąże się to z tym, że i tak napięcie było wysokie ze względu na trwanie fazy rozruchu). Na początku symulacji pojawiły się pewne skoki napięcia, ale oscylują one w granicach 1V, co przy rzeczywistej wartości napięcia jest odchyleniem dopuszczalnym.



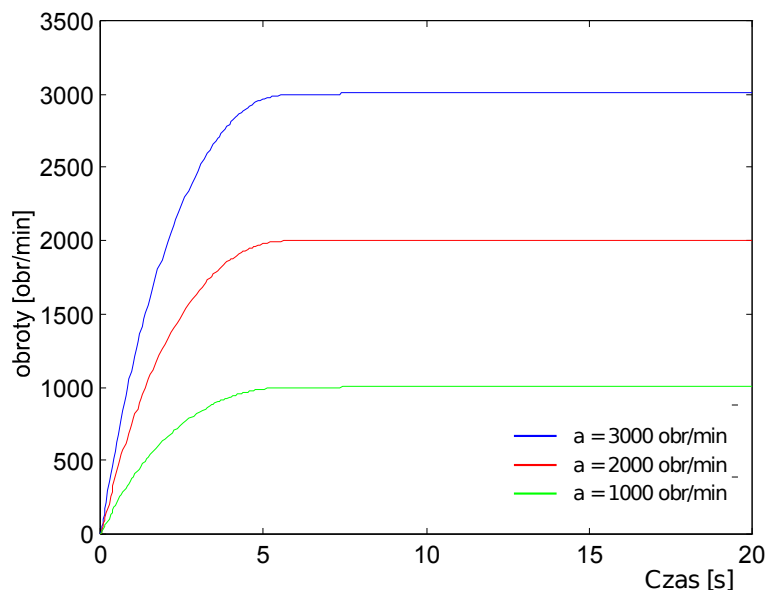
a)



b)



Rys. 8.6 - Przebiegi dla rozruchu silnika bez obciążenia przy sterowaniu regulatorem FLC typu PD:  
 a) obroty, b) prąd, c) błąd, d) zmianą błęd, e) napięcie



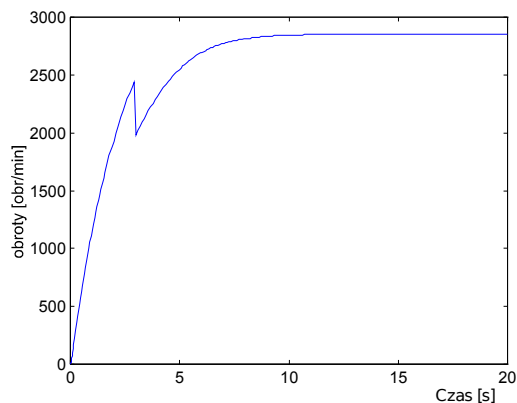


Rys. 8.7 - Przebieg ruchu silnika dla różnych wartości zadanych obrotów przy sterowaniu FLC typu PD.

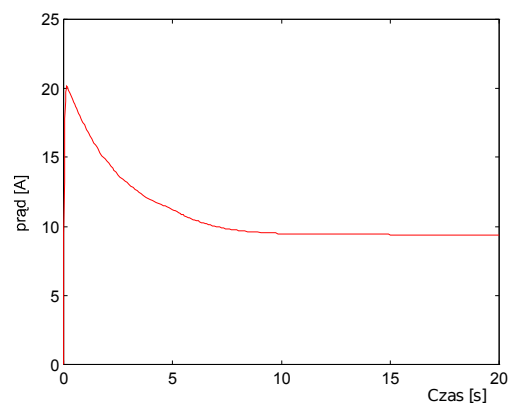
Rysunek 8.9 odpowiada symulacji, w której silnik obciążono dwoma różnymi zakłóceniami:

- pierwsze dla  $t = 8s$  - obciążenie impulsem zewnętrznym zmniejszające prędkość obrotową 1000 obr/min,
- drugie dla  $t = 12s$  - obciążenie stałe (skokiem jednostkowym), zmniejszające prędkość o tę samą wartość.

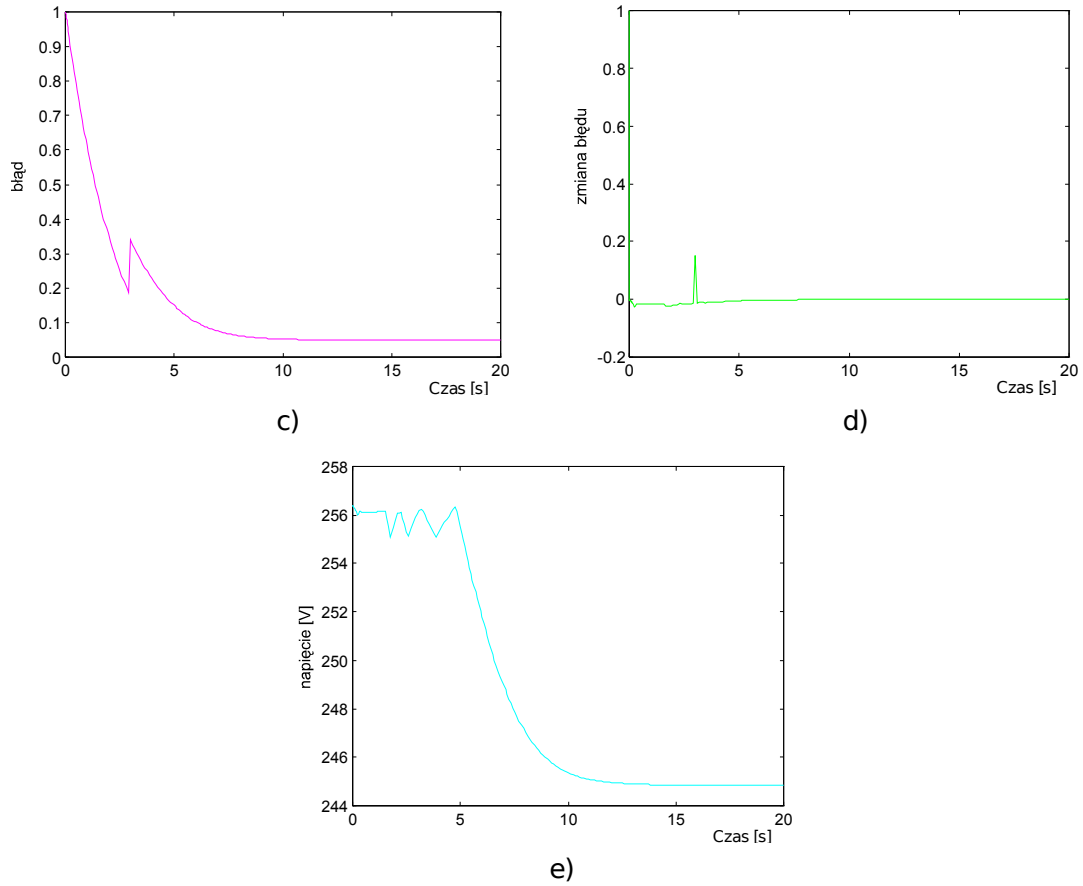
W przypadku pierwszego zakłócenia regulator, na dodatnią wartość błędu i dodatnią zmianę, zareagował zwiększając napięcie sterowania. Po ustaniu zakłócenia - pracą bez obciążenia - (błąd zerowy a zmiana ujemna), aby nie doszło do nadmiernego wzrostu prędkości obrotowej podawany sygnał sterowania został odpowiednio zmniejszony. Po chwili pracę ustabilizowała i silnik powrócił do obrotów nominalnych. Po pojawieniu się stałego obciążenia skokiem jednostkowym wzrasta napięcie i prąd twornika. Regulator podaje maksymalną możliwą numeryczną wartość napięcia wyliczoną w fazie wyostrażania. Błąd jednak pozostaje niezerowy. W związku z tym dla tego obciążenia silnik nie powraca do wartości zadanej obrotów.



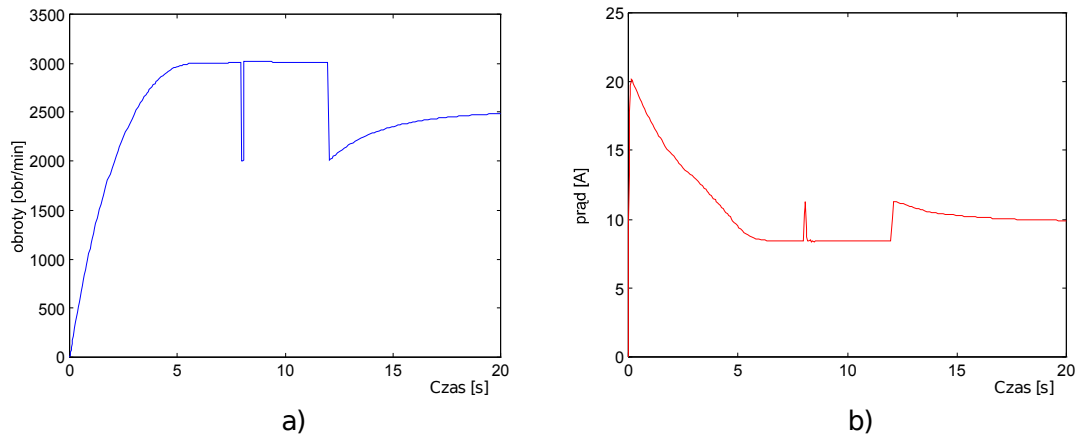
a)

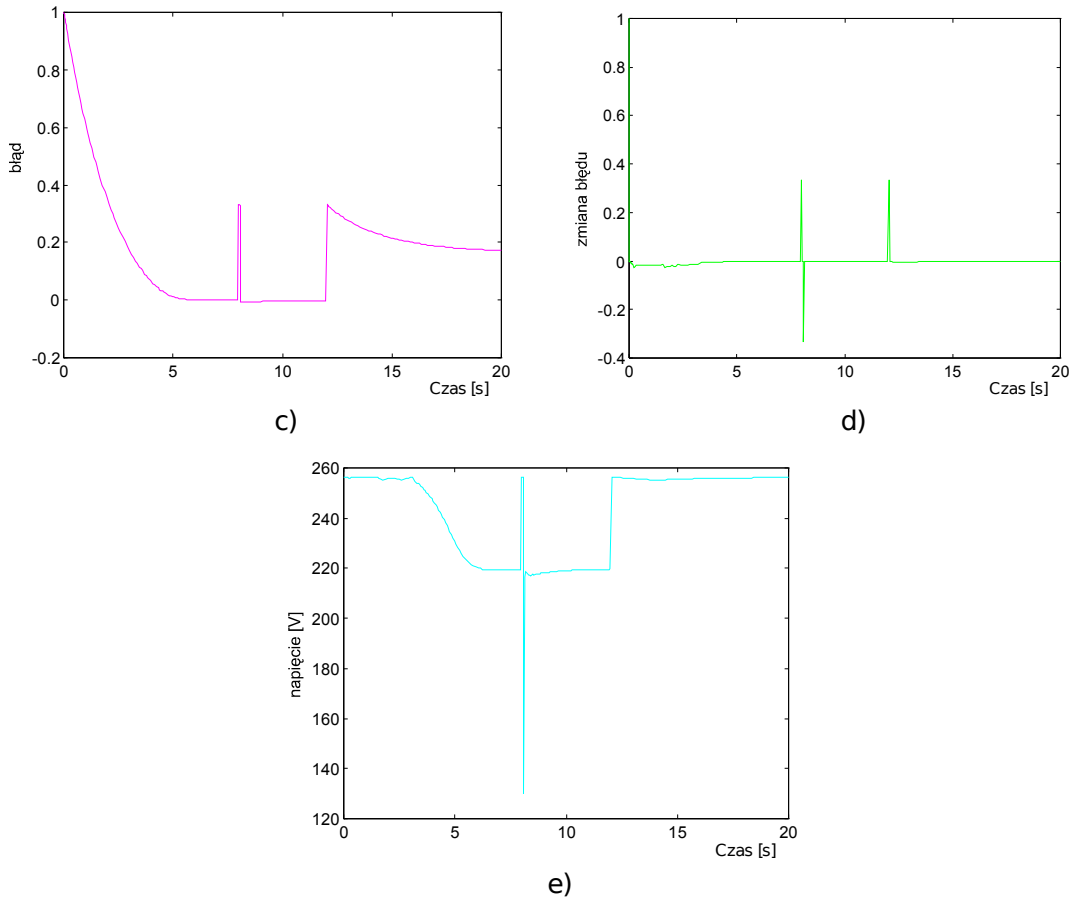


b)



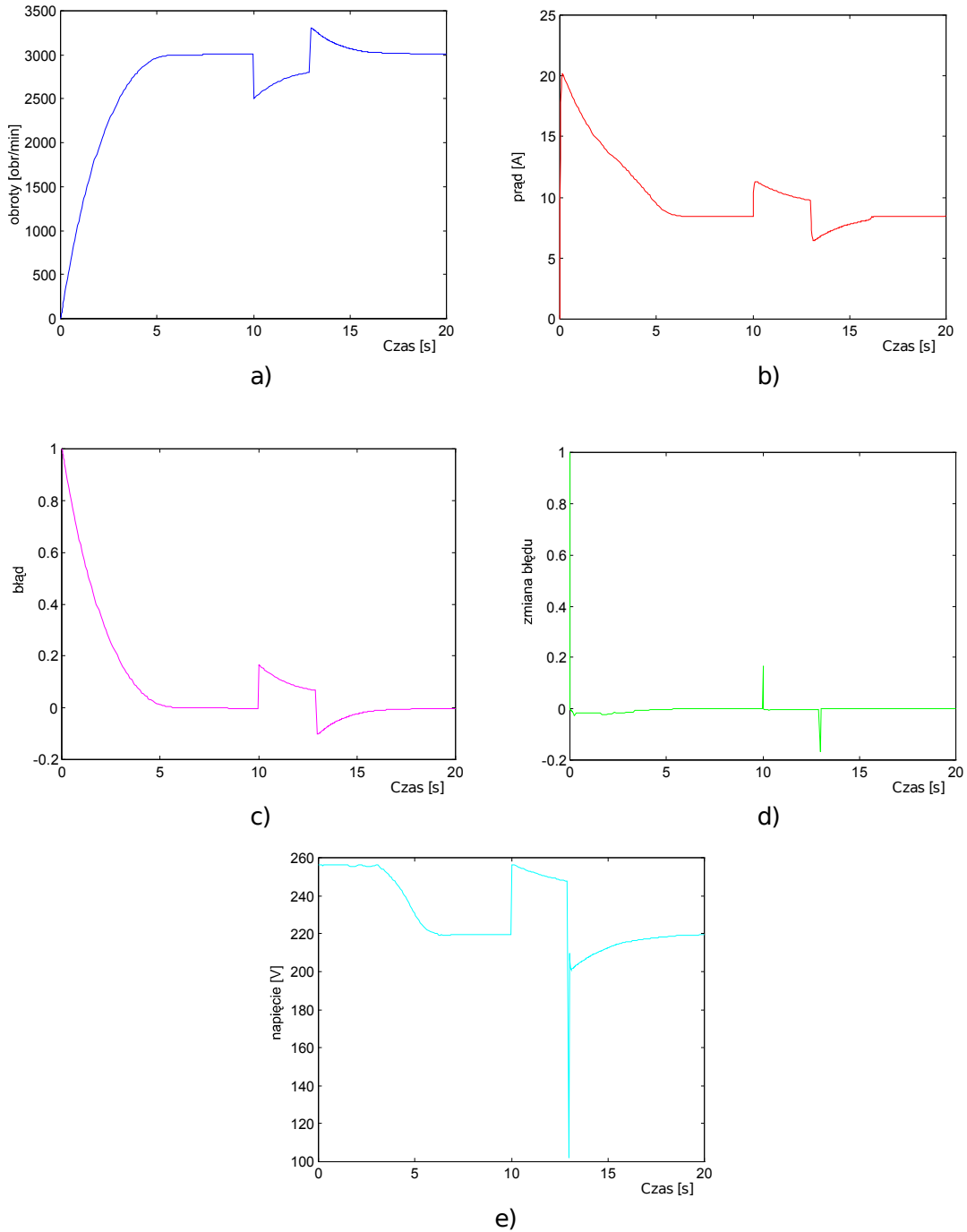
Rys. 8.8 - Przebiegi dla silnika przy obciążeniu przyłożonym w trakcie rozruchu dla sterowania FLC typu PD: a) obroty, b) prąd, c) błąd, d) zmiana błęd, e) napięcie





Rys. 8.9 - Przebiegi dla silnika przy obciążeniu falą prostokątną i skokiem jednostkowym dla sterowania FLC typu PD: a) obroty, b) prąd, c) błąd, d) zmiana błęd, e) napięcie

Rysunek 8.10 odpowiada symulacji, w której zadaniem regulatora była eliminacja wpływu obciążenia podanego na wyjście obiektu sterowanego po czasie 10 sekund od rozpoczęcia rozruchu. Zakłócenie miało postać fali prostokątnej trwającej 3 sekundy. W trakcie czasu trwania zakłócenia sygnał sterujący wzrósł starając się doprowadzić obroty do wartości zadanej. Nagle po ustaniu zakłócenia (powrót do pracy bez obciążenia) obroty wzrosły ale bardzo szybko wróciły do 3000 obr/min. Aby do tego doszło napięcie zostało chwilowo zmniejszone. Na końcu symulacji błąd i zmiana błęd osiągnęły zerowe wartości a prąd, napięcie twornika i prędkość osiągnęły wartości nominalne.



Rys. 8.10- Przebiegi pracy silnika dla obciążenia falą prostokątną przysterowaniu FLC typu PD:  
 a) prędkość, b) prąd, c) błąd, d) zmiana błędu, e) napięcie

### Porównanie sterowania rozmytego z konwencjonalnym oraz zebranie uzyskanych doświadczeń

Po przeprowadzonych symulacjach działania regulatora rozmytego pod wpływem różnych obciążeń postanowiono porównać go z regulatorem konwencjonalnym.

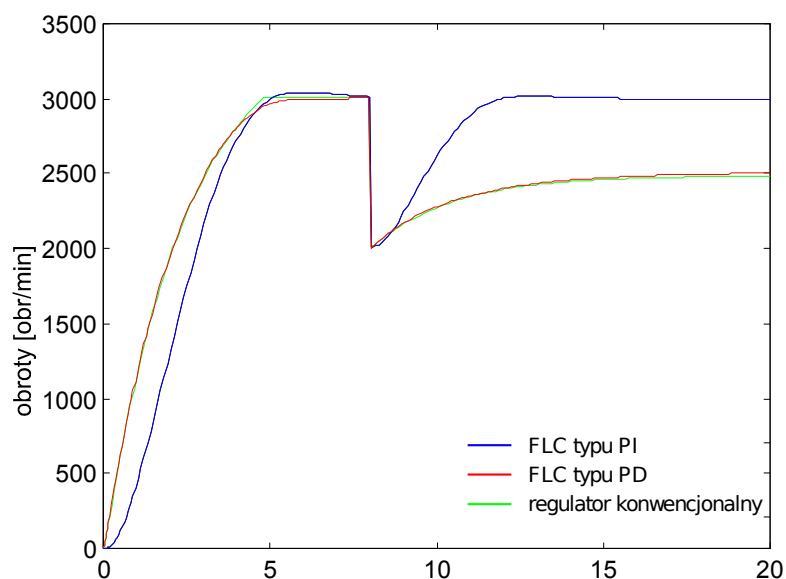
Do tego celu wybrano prosty regulator trójpołożeniowy. Sygnałem wejściowym tego regulatora był błąd - różnica pomiędzy wartością zadaną obrotów a wartością aktualną. Wyjściem natomiast napięcie sterowania podawane na obiekt silnika. Regulator ten posiadał następującą nastawę napięcia:

- 255V; w zakresie [0.01,1] w przypadku narastania błędu, w zakresie [0.005,1] w przypadku pomniejszania się błędu,
- 220V; w zakresie [-0.01,0.01] błędu,
- 70V; w zakresie [-1,-0.01] w przypadku zmniejszania się błędu, w zakresie [-1,-0.005] w przypadku narastania błędu.

Należy tutaj wspomnieć, że przedział, na którym określony jest błąd dla regulatora trójpołożeniowego pokrywa się z dziedziną znormalizowaną zmiennej *błąd* dla regulatorów rozmytych.

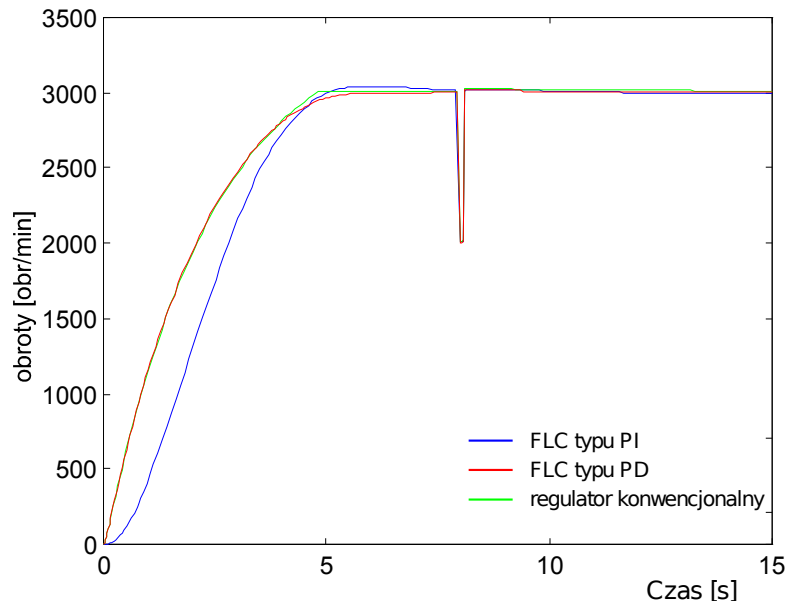
Przy maksymalnej nastawie 255V prąd twornika nie przekraczał wymaganego ograniczenia 20A. Napięcie 220V odpowiada wartości nominalnej dla silnika.

Rysunek 8.11 obrazuje skuteczność sterowania regulatorów przy obciążeniu silnika stałym momentem. Zdecydowanie najlepsze rezultaty uzyskano przy sterowaniu rozmytym regulatorem FLC typu PI. Tylko takie sterowanie doprowadziło do powrotu obrotów do wartości zadanej pomimo przyłożonego obciążenia. Regulator FLC typu PD okazał się niewiele lepszy od regulatora konwencjonalnego (odchyłka od prędkości zadanej jest mniejsza). Da się jednak zauważyć, że czas rozruchu jest najkrótszy dla regulatora trójpołożeniowego. Następuje jednak gwałtowne przełączenie napięcia w momencie przejścia w inny zakres błędu.



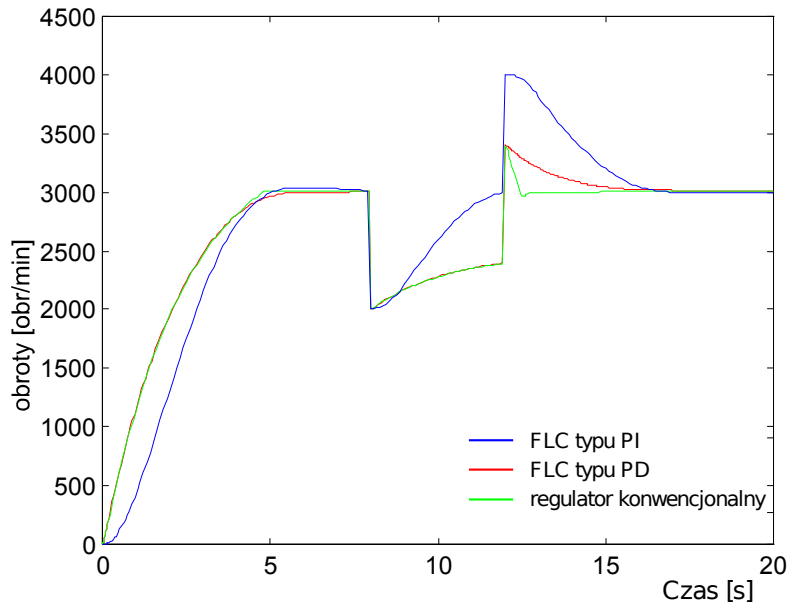
Rys. 8.11 - Porównanie działania regulatora rozmytego z regulatorem konwencjonalnym przy stałym obciążeniu

Rysunek 8.12 to wyniki przeprowadzonej symulacji, w której na wyjście podano obciążenie impulsowe. W zasadzie każdy z badanych regulatorów skutecznie poradził sobie z tym zakłóceniem.



Rys. 8.12 - Porównanie działania regulatora rozmytego z regulatorem konwencjonalnym przy obciążeniu impulsowym

Rysunek 8.13 przedstawia skuteczność działania regulatorów kiedy obciążenie miało formę fali prostokątnej trwającej 4 sekundy. Okazało się, że regulator FLC typu PI tym razem najgorzej radzi sobie z takim zakłóceniem. Najlepszy okazał się najprostszy regulator konwencjonalny. Można to jednak wytłumaczyć. Dolna nastawa regulatora ma wartość 70V. Żaden z regulatorów rozmytych nie osiągnął tak niskiego napięcia przy ujemnym błędzie i ujemnej zmianie. Wynika to z tego, że regulatory rozmyte mają bardziej płynny sposób działania. Nie ma możliwości tak radykalnej, nagłej zmiany wartości. Należy jednak dodać, że tylko FLC typu PI w czasie, w którym obciążenie było przyłożone wyeliminował błąd prędkości, mimo faktu, że w całości zakłócenia wypadł najgorzej.



Rys. 8.13 - - Porównanie działania regulatora rozmytego z regulatorem konwencjonalnym przy obciążeniu w formie fali prostokątnej

Podsumowując, okazuje się, że sterowanie silnika regulatorem rozmytym nadaje się jeżeli silnik pracuje przy stałe przyłożonym obciążeniu - najczęściej realizowane w rzeczywistości. Wtedy najlepszym rozwiązaniem jest FLC typu PI. W innym przypadku wystarczy prosty regulator konwencjonalny. Stosowanie regulatora FLC typu PD nie ma większego sensu, gdyż równie dobre wyniki można osiągnąć przy pomocy regulatora trójpołożeniowego, a sama faza dostrajania FLC jest dużo dłuższa i pracochłonna.

Dobrym rozwiązaniem w tym przypadku może też być regulator rozmyty typu TSK (typu Sugeno dla edytora FIS). Jest on właściwie połączeniem regulatora rozmytego i konwencjonalnego. Główną różnicą polega na tym, że wyjście (sterowanie) nie jest określone za pomocą funkcji przynależności (zbiorów rozmytych). Dlatego nie ma potrzeby wyliczania wartości numerycznej wyjścia (wyostrzania). Jest ono już w sposób numeryczny określone wewnątrz regulatora.

Metoda wnioskowania typu TSK jest związana z bazą reguł specjalnego formatu, który odznacza się następnikami typu funkcyjnego stosowanymi zamiast następników rozmytych używanych w przedstawionych wcześniej regulatorach FLC typu PI i PD. Reguły w tym przypadku miałyby następującą postać:

**JEŻELI**  $e$  jest  $B_{i1}$  **I**  $\Delta e$  jest  $B_{i2}$  **TO**  $u_i = f(e, \Delta e)$

przy czym  $B_{ij}$  są etykietami lingwistycznymi określonymi jako zbiory rozmyte odniesienia na przestrzeniach wejściowych. Każda funkcja w następniku reguły może być uwa-

żana za model o wejściach nierozmytych  $e$ , wyjściu nierozmytym  $u_i$  i odpowiednich parametrach. Wyjście nierozmyte wyprowadzone przez model metodą TSK jest określone przez średnią ważoną wyjść nierozmytych  $u_i$  poszczególnych podsystemów. Strukturą modelu TSK jest określona przez liczbę reguł, które wyznaczają podział przestrzeni wejściowej i zależy od struktury poszczególnych podsystemów. Podobnie jak w przypadku FLC typu Mamdaniego wejście regulatora jest normalizowane, dzięki czemu zyskuje się dużą elastyczność ze względu na użyte funkcje przynależności. Jednak procedura normalizacji dokonywana jest tylko w poprzedniku reguły. Następnik reguły pracuje ze stanami nieznormalizowanymi tak, że normalizacja nie jest potrzebna - wynik każdej reguły jest już ostry.

Regulator FLC typu TSK umożliwia zastąpienie dekompozycji nierozmytej dekompozycją rozmytą oraz przełączającymi funkcjami boolowskimi przez interpolacyjny mechanizm wnioskowania TSK. Umożliwia wprowadzenie wiedzy eksperta do podziału wejścia i stanu; jest to szczególnie użyteczne w przypadkach, gdy różne obszary związane z różnymi warunkami działania można scharakteryzować za pomocą etykiet lingwistycznych.

W przypadku zaprojektowanego FLC typu Sugeno w edytorze FIS wyjście projektuje się jako funkcję liniową wejść. W szczególnym przypadku może przyjmować określone stałe wartości. Są one oczywiście określone w sposób lingwistyczny (etykiety), lecz pod nazwą kryje się już tylko liczba. Przy odpowiednim doborze liczby etykiet, reguły całym procesie sterowania można osiągnąć zadowalające rezultaty.

Jednak w pracy przedstawione są tylko regulatory typu Mamdaniego. Ten typ regulatora jest bardziej popularny i zazwyczaj podawany w literaturze jako zalecany. Ponadto tematem pracy jest zaprojektowanie regulatora rozmytego, a jak wspomniano wcześniej regulator TSK jest swego rodzaju połączeniem regulatora rozmytego i konwencjonalnego. Ponadto regulator FLC typu PI, który okazał się najlepszy w porównaniu idealnie pasuje jako model Mamdaniego. Jego wyjściem jest zmiana sterowania, która w łatwy sposób daje się opisać w sposób lingwistyczny. Podając jedynie zakres działania tej zmiennej regulator już sam dobiera odpowiednią wartość wyjścia. Trudno byłoby określić stałą wartość zmiany sterowania a tym bardziej określić takie wyjście jako funkcję liniową błędów i jego pochodnej. Dlatego że między innymi na swą prostotę właśnie FLC typu Mamdaniego znalazł zastosowanie w przypadku sterowania obiektem silnika obcowzbudnego prądu stałego.

Poniższa tabela zawiera podsumowanie i wpływ poszczególnych czynników należących do procesu sterowania na zachowanie się sterowanego obiektu.



Tabela 8.1 - Wpływ czynników składowych regulatora na zachowanie się obiektu i efektywność sterowania

<b>CZYNNIK WYBORU</b>	<b>WPŁYW NA ZACHOWANIE</b>
<i>Rodzaj FLC: typu PI lub PD</i>	Dla przypadku sterowania silnikiem obciążonym prądem stałym zdecydowanie lepszy jest FLC typu PI. Dzięki obecności czynnika całkującego uzyskuje się dużą odporność na przykładne obciążenie.
<i>Normalizacja przestrzeni (czynnik skalujący)</i>	Jej rola jest analogiczna do współczynników wzmocnienia w regulatorze konwencjonalnym. Ma bezpośredni wpływ na działanie regulatora i jego własności stabilności.
<i>Zakres przestrzeni funkcji przynależności dla zmierzonych</i>	Jeżeli są przestrzeniami znormalizowanymi to ich zakres nie jest bardzo istotny. W przeciwnym razie mają takie znaczenie jak czynniki skalujące
<i>Ilość etykiet lingwistycznych</i>	Mają wpływ na ilość reguł. Odpowiedni ich dobór wpływa na precyzję działania regulatora - szybkość odpowiedzi na zakłócenie
<i>Kształt funkcji przynależności</i>	Nie gra większej roli. W zaprojektowanych regulatorach zastosowano jedynie funkcje w kształcie trójkąta i trapezu - prosty opis parametryczny (wpływ na szybkość obliczeń)
<i>Ilość reguł</i>	Zupełność zbioru reguł wyklucza sytuację kiedy regulator nie posiada odpowiedniej reguły dla danych wejściowych, aczkolwiek nie jest to konieczne. Większa ilość reguł - lepsze sterowanie
<i>Proces strojenia reguł</i>	Najważniejszy element strojenia regulatora. Odpowiedni dobór następników reguł przy różnych wejściach ma bezpośredni wpływ na efektywność właściwego działania regulatora.
<i>Parametr logicznego</i>	W przypadku zamodelowanego obiektu bez znaczenia
<i>Parametr logicznego LUB</i>	W przypadku zamodelowanego obiektu bez znaczenia
<i>Metoda implikacji</i>	W przypadku zamodelowanego obiektu bez znaczenia
<i>Metoda agregacji</i>	W przypadku zamodelowanego obiektu bez znaczenia
<i>Metoda wyostrzenia</i>	Najlepszą okazała się metoda środka obszaru - najczęściej zalecana w literaturze. Uzyskała dla niej najlepsze wyniki symulacji

## Zakończenie i wnioski

Celem pracy było zaprojektowanie, zaimplementowanie i przetestowanie wybranego regulatora rozmytego. W pracy dość dokładnie przeanalizowano teorię logiki rozmytej i sterowania rozmytego. Przedstawiono proces projektowania oraz strojenia regulatora rozmytego. Zbadano wpływ zmiany poszczególnych składników regulatora oraz jego rodzaj na pracę badanego (sterowanego) układu. Pokazano wyniki przeprowadzonych testów dla różnych nastaw obciążenia.

Okazało się, że zastosowanie regulatora rozmytego do procesu stabilizacji obrotów silnika obcowzbudnego prądu stałego dało zadowalające wyniki. Dzięki temu regulator rozmyty może stanowić jedną z opcji wyboru dla sterowania obiektem tego typu. Należy jednak zastosować regulator FLC typu PI, gdyż tylko on wykazuje lepsze działanie w porównaniu z regulatorem konwencjonalnym.

Reasumując, sterowanie rozmyte stanowi ciekawe rozwiązanie w dzisiejszej teorii sterowania. Ma praktyczne zastosowanie w przemyśle, tak więc ma szansę dalszego rozwoju. Informacja w regulatorze zawarta jest w formie lingwistycznej, dzięki czemu nie ma problemu z jej zrozumieniem. W przedstawionej pracy baza wiedzy została zbudowana po zastosowaniu metody zdroworozsądkowej. Takie rozwiązanie sprawdziło się w praktyce, choć sterowanie rozmyte ma większe zastosowanie w sytuacji, gdzie istnieje wiedza eksperta o danym obiekcie sterowanym.

## Literatura

1. Zadeh L. A. : Fuzzy Sets; Information and Control, **Vol. 8, pp. 338-353 1965**
2. Zadeh L. A. : Probability Measures of Fuzzy events; J. Of Mathematical Analysis and Applications, **Vol. 23, pp. 421-427, 1968**
3. Zadeh L. A. : Outline of a new approach to the analysis of complex systems and decision processes; IEEE Transactions on Systems, Man, and Cybernetics, **Vol. 3, No. 1, pp. 28-44 Jan. 1973**
4. Mamdani E. H. : Application of Fuzzy Algorithm for Control of Simple Dynamic Plant; Proc. IEEE, 121(12), **pp. 1585-1888, 1974**
5. Kickert W. J. M.; Van Nauta Lemke H. R. : Application of a fuzzy controller in a warmwater plant; Automat, **Vol. 12 No. 4, pp. 301-308 1976**
6. Pappis C. P.; Mamdani E. H. : A fuzzy logic controller for a traffic junction; IEEE Transactions on Systems, Man and Cybernetics 7, **pp. 707-712, 1977**
7. Tong R. M.; Beck M. B.; Latten A. : Fuzzy control of the activated sludge wastewater treatment process; Automatica 16, **pp. 695-702, 1980**
8. Yagishita O.; Itoh O.; Sugeno M. : Application of fuzzy reasoning to the water purification process; Industrial Applications of Fuzzy Control, **pp. 19-40** Amsterdam: North Holland, **1985**
9. Pedrycz W. : An approach to the analysis of fuzzy systems; Int. J. Control, **Vol. 34 No. 3, pp. 403-421 1981**
10. Larsen P. M. : Industrial applications of fuzzy logic control; Int. J. Man Mach. Studies. **Vol. 12, No. 1, pp. 3-10 1980**
11. Rescher N. : Many-Valued Logic; New York, McGraw-Hill, **1969**
12. Zadeh L. A. : Fuzzy algorithm, Inform. And Control 12, **pp. 94-102, 1968**
13. Driankov D.; Hellendoom H.; Reinfrank M. : Wprowadzenie do sterowania rozmytego; WNT, Warszawa **1996**
14. Mamdani E. H.; Assilian S. : An experiment in linguistic synthesis with a fuzzy logic controller; International Journal of Man-Machine Studies, **Vol. 7, No. 1, pp. 1-13 1975**
15. Yager R. R.; Filev D. P. : Podstawy modelowania i sterowania rozmytego; WNT, Warszawa **1995**
16. Lee C. C. : Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part II; IEEE transactions on systems, **Vol. 20, March/April 1990**
17. Sugeno M. : Industrial applications of fuzzy control; Elsevier Science Pub. Co., **1985**
18. Zadeh L. A. : Towards a theory of fuzzy systems; Aspects of Network and System Theory, **p. 469-490** New York **1971**
19. Tong R. M. : Synthesis of fuzzy models for industrial processes - Some recent results; International Journal of General Systems 4, **pp. 143-163, 1978,**
20. Kosko B. : Neural Networks and Fuzzy Systems; Prentice Hall: Englewood Cliffs, NJ, **1991,**
21. Takagi T.; Sugeno M. : Derivation of fuzzy control rules from human operator's control actions; Proc. Of the IFAC Symp. On Fuzzy Information, Knowledge Representation and Decision Analysis, **pp. 55-60,** Marseilles, France **July 1983,**
22. Latek W. : Zarys maszyn elektrycznych; Warszawa, **1974,**
23. Skwarczyński J.; Tertil Z. : Maszyny elektryczne. Teoria część druga; Wydawnictwa AGH, Kraków **1997,**
24. Lewis J. W. : Modeling Engineering Systems; High Text Publications, Inc., **1994**
25. Gulley N.; Roger Jang J. S. : Fuzzy Logic Toolbox; The Math Works, Inc., **1995**

26. Boverie S. Et al. : Fuzzy Logic Control Compared with other Automatic Control Approaches; Proceedings 30<sup>th</sup> IEEE-CDC Conf. Decision and Control, Brighton, UK, **December 11-13, 1991**
27. Devillard F.; Viard P.; Cecchin T. : Fuzzy Logic Speed Control of DC Motor.