

Sterowanie układem dynamicznym za pomocą systemów rozmytych

Na podstawie materiałów opracowanych przez
dr. hab. inż. Mirosława Świercza, prof. nzw. PB

Białystok, wrzesień 2020 r.

Celem ćwiczenia jest zaprojektowanie i nastrojenie regulatora rozmytego (*FLC* – ang. *Fuzzy Logic Controller*) dla ciągłego, stacjonarnego, liniowego układu drugiego rzędu o parametrach skupionych. Wybrany obiekt regulacji jest z założenia stabilny i reprezentuje układ oscylacyjny lub dwuinercyjny. Obiekt taki można zatem opisać za pomocą liniowego równania różniczkowego zwyczajnego, które wiąże sygnał wyjściowy $y(t)$ z sygnałem pobudzającym wejście układu, $u(t)$:

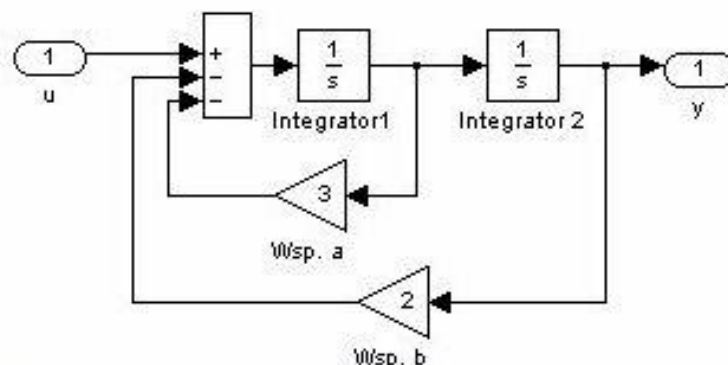
$$\frac{d^2 y(t)}{dt^2} + a \frac{dy(t)}{dt} + b y(t) = k u(t) \quad (1)$$

lub (przy zerowych warunkach początkowych w równaniu (1)) transmitancji operatorowej:

$$G_0(s) = \frac{k}{s^2 + a s + b} \quad (2)$$

Występujące w obydwu równaniach współczynniki liczbowe są, przy założeniu stabilności układu, dodatnie ($k > 0$, $a > 0$, $b > 0$). Dla układu dwuinercyjnego zachodzi zależność $a^2 \geq 4b$, zaś w przypadku przeciwnym odpowiedź układu ma charakter gasnących oscylacji.

Schemat wykorzystywanego w ćwiczeniu układu regulacji, w postaci bloków funkcjonalnych pakietu *MATLAB/SIMULINK*, przedstawiono na rysunku 1. Współczynniki wzmocnienia bloków wzmacniaczy ('Gain'), umieszczonych w pętli sprzężenia (oznaczonych na schemacie jako 'Wsp. a' i 'Wsp. b'), odpowiadają odpowiednio współczynnikom a i b w równaniach (1) i (2).



Rys. 1. Schemat obiektu regulacji (II-go rzędu) w postaci bloków funkcjonalnych pakietu *SIMULINK*

1. Układ regulacji z regulatorem rozmytym typu PD

1.1. Opis przykładowego układu regulacji z regulatorem rozmytym typu PD

W przypadku klasycznego liniowego regulatora typu *PD*, prawo sterowania (będące opisem relacji pomiędzy wejściem a wyjściem regulatora) można wyrazić następującym wzorem, odpowiadającym strukturze bloku regulatora *PID* pakietu *MATLAB*:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (3)$$

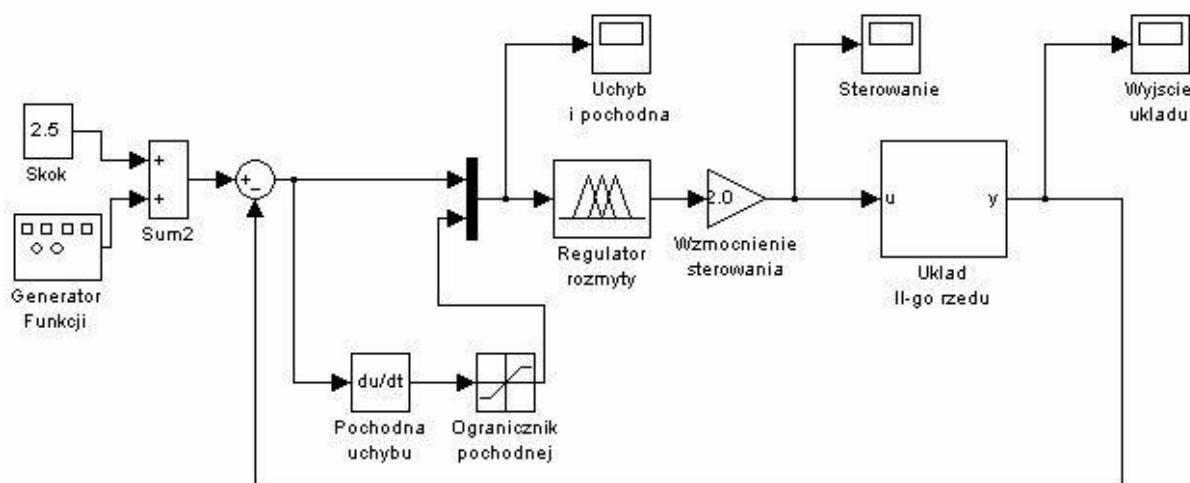
gdzie $e(t)$ jest uchybem regulacji (wejściem regulatora), zaś wyjście regulatora $u(t)$ pobudza wejście obiektu regulacji.

Przykładowy rozmyty regulator PD , wykorzystywany w ćwiczeniu, realizuje nieliniowe prawo sterowania:

$$u(t) = f\left(e(t), \frac{de(t)}{dt}\right) \quad (4)$$

Prawo to, za pomocą rozmytych reguł wnioskowania typu **JEŻELI – TO**, opisuje relacje pomiędzy zmiennymi wejściowymi regulatora: uchybem $e(t)$ i pochodną uchybu $de(t)/dt$, a wyjściem regulatora – sterowaniem obiektu regulacji, $u(t)$.

Schemat blokowy układu regulacji PD (w symbolice pakietu *MATLAB/SIMULINK*) z regulatorem rozmytym przedstawia rysunek 2.



Rys. 2. Schemat układu regulacji z rozmytym regulatorem typu PD

Podsystem oznaczony jako ‘*Układ II-go rzędu*’ zawiera obiekt regulacji, przedstawiony na rys. 1. Zmiany wartości współczynników a i b modelu układu (równanie (1) lub (2)) można dokonywać po dwukrotnym naciśnięciu („kliknięciu”) myszką na symbol podsystemu, a po otwarciu okienka zawierającego model układu – kolejne dwukrotne „kliknięcie” myszką na symbol wzmacniacza i wpisanie odpowiedniej wartości wzmocnienia w okienku dialogowym.

Sygnał zadany, za którym powinna nadążać wielkość regulowana $y(t)$, jest sumą sygnału stałowartościowego (skoku jednostkowego) i sygnału zmiennego w czasie, np. sinusoidy lub sygnału losowego (modelującego szum biały). Zarówno postać sygnału zmiennego w czasie, jak i jego parametry (amplitudę i częstotliwość) można zmieniać po dwukrotnym naciśnięciu („kliknięciu”) myszką na symbol generatora funkcyjnego i wpisaniu odpowiednich wartości w okienku dialogowym. W układzie regulacji zastosowano również blok wzmacniacza z nasyceniem (‘*Ogranicznik pochodnej*’), który zapobiega nadmiernym zmianom wartości części różniczkującej wejścia regulatora, na przykład wówczas, gdy sygnał zadany zawiera składową losową (szum w paśmie wysokich częstotliwości). Wartość ograniczenia sygnału pochodnej można zmieniać w analogiczny sposób, jak w przypadku zmian parametrów innych bloków funkcjonalnych modelu.

W centralnej części schematu znajduje się opisany niżej podsystem, reprezentujący rozmyty regulator *PD*.

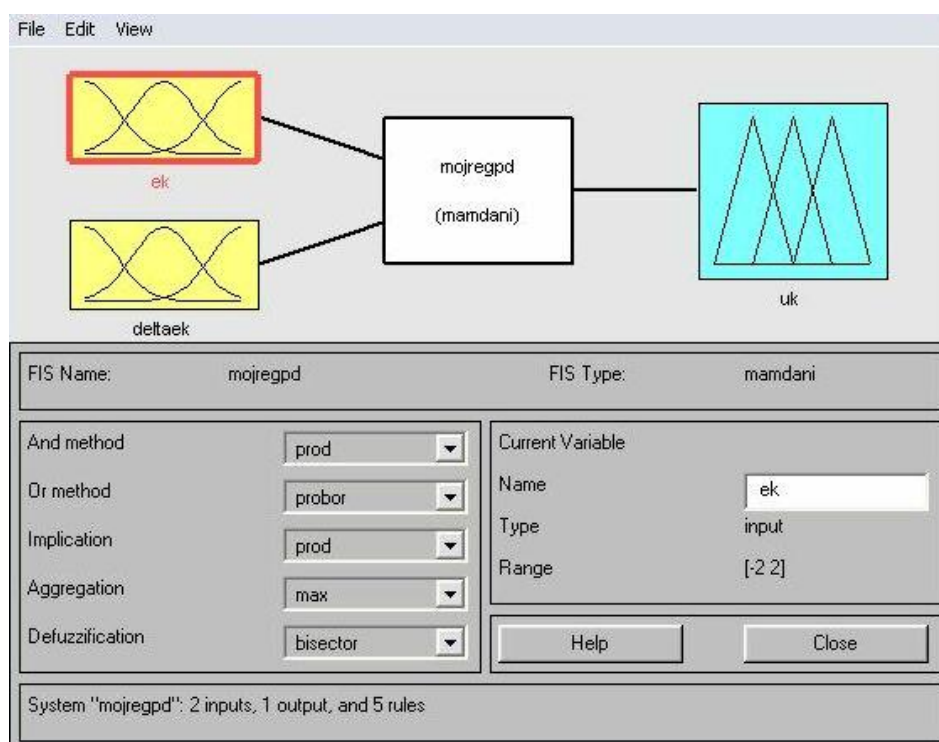
1.2. Opis regulatora rozmytego typu *PD*

Rozmyty regulator *PD* został utworzony za pomocą edytora systemów rozmytych biblioteki narzędziowej „Fuzzy Logic Toolbox” pakietu *MATLAB*. Wszelkie zmiany elementów systemu rozmytego (który realizuje nieliniowe prawo sterowania (4)) mogą być dokonywane wyłącznie za pomocą tego edytora, po czym zaktualizowaną wersję systemu rozmytego należy powtórnie zapisać na dysku.

Uwaga!

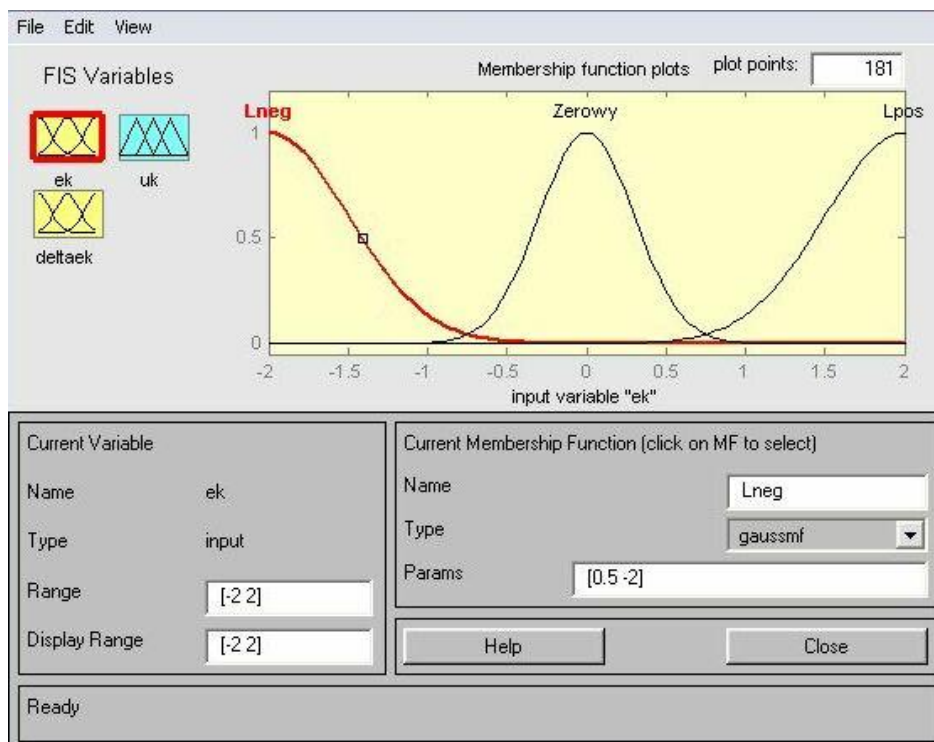
Model układu regulacji ‘fuz_pdreg’ zrealizowany w pakiecie *SIMULINK* (przedstawiony na rys. 2) wykorzystuje strukturę regulatora, ładowaną do przestrzeni roboczej pakietu *MATLAB* w momencie wywołania (otwarcia) tego modelu. Stąd też po każdorazowej aktualizacji systemu rozmytego ‘mojregpd’ powinno się również zamknąć model ‘fuz_pdreg’, albo też równolegle z zapisem modelu na dysk eksportować również uaktualnioną wersję systemu rozmytego do przestrzeni roboczej pakietu *MATLAB*.

Proponowany regulator rozmyty reprezentuje system wnioskowania typu Mamdaniego, o dwóch zmiennych wejściowych: ‘ek’ i ‘deltaek’, które są odpowiednio bieżącymi wartościami uchybu i jego pochodnej oraz o jednej zmiennej wyjściowej ‘uk’, która jest bieżącą wartością sterowania, działającego na obiekt regulacji. Na rysunku 3 przedstawiono wygląd głównego ekranu edytora systemu rozmytego, po załadowaniu pliku ‘mojregpd’, zawierającego opisywany regulator rozmyty typu *PD*. Z rysunku można również odczytać metody realizacji operacji logicznych (*AND*, *OR*, agregacji i implikacji) oraz wybraną metodę wyostrzania (defuzyfikacji).



Rys. 3. Główny ekran edytora systemu rozmytego ‘mojregpd’, reprezentującego regulator typu *PD*

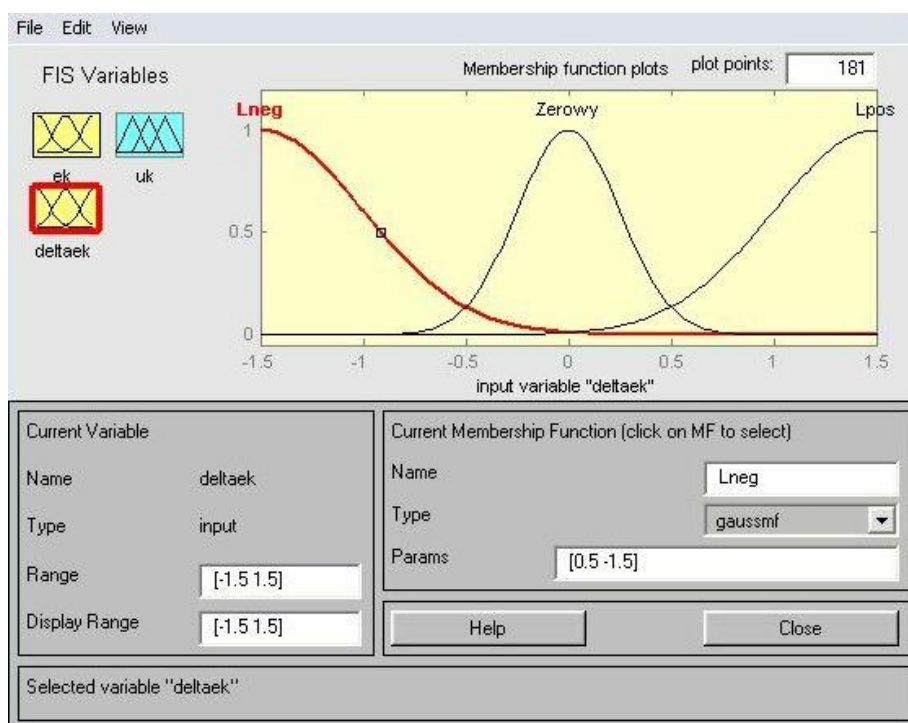
Zmiennej wejściowej ‘ek’ przyporządkowano trzy etykiety (kategorie) lingwistyczne: „duży ujemny” (‘Lneg’), „zerowy” (‘Zerowy’) i „duży dodatni” (‘Lpos’). Dla wszystkich kategorii wybrano gaussowskie funkcje przynależności, pokrywające symetrycznie założony zakres zmienności uchybu ($[-2, 2]$). Wykresy funkcji przynależności dla zmiennej wejściowej systemu rozmytego ‘ek’ przedstawia rysunek 4.



Rys. 4. Funkcje przynależności dla zmiennej wejściowej ‘ek’ rozmytego regulatora PD

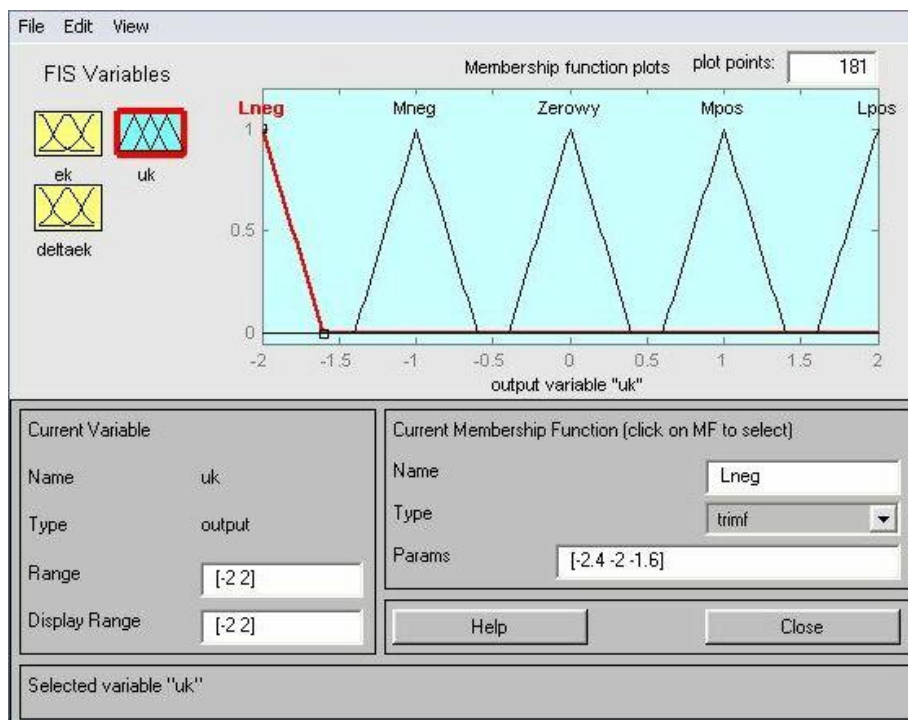
W większości opracowań, dotyczących projektowania regulatorów rozmytych, zaleca się normalizację zakresu wszystkich funkcji przynależności do przedziału $[-1, 1]$. Wymaga to jednak stosowania w modelu symulacyjnym z rys. 2 wzmacniaczy skalujących dla każdej zmiennej wejściowej regulatora rozmytego. Wzmocnienie każdego ze wzmacniaczy (zapewniającego normalizację do przedziału $[-1, 1]$) jest równe odwrotności maksymalnej założonej wartości danej zmiennej wejściowej. Należy zwrócić uwagę, że przy normalizacji zakresu definiowania funkcji przynależności do przedziału $[-1, 1]$, współczynniki skalujące mają zasadniczy wpływ na jakość sterowania.

Zmiennej wejściowej ‘deltaek’ przyporządkowano również trzy etykiety (kategorie) lingwistyczne, identyczne jak dla zmiennej ‘ek’: „duży ujemny” (‘Lneg’), „zerowy” (‘Zerowy’) i „duży dodatni” (‘Lpos’). Dla wszystkich kategorii wybrano gaussowskie funkcje przynależności, pokrywające symetrycznie założony zakres zmienności pochodnej uchybu ($[-1.5, 1.5]$). Parametry tych funkcji przynależności (w szczególności – wartości ich dyspersji) różnią się jednak od parametrów funkcji przynależności zdefiniowanych dla zmiennej ‘ek’. Wykresy funkcji przynależności dla zmiennej wejściowej systemu rozmytego ‘deltaek’ przedstawia rysunek 5.



Rys. 5. Funkcje przynależności dla zmiennej wejściowej 'deltaek' rozmytego regulatora PD

Zmiennej wyjściowej systemu ('uk') przypisano pięć etykiet (kategorii) lingwistycznych: „duży ujemny” ('Lneg'), „średni ujemny” ('Mneg'), „zerowy” ('Zero'), „średni dodatni” ('Mpos') i „duży dodatni” ('Lpos'). Dla wszystkich kategorii wybrano trójkątne funkcje przynależności, pokrywające symetrycznie założony zakres zmienności sterowania 'uk' $[-2, 2]$. Wykresy funkcji przynależności dla zmiennej wyjściowej systemu rozmytego 'uk' przedstawia rysunek 6.



Rys. 6. Funkcje przynależności dla zmiennej wyjściowej 'uk' rozmytego regulatora PD

Zaproponowano pięć reguł wnioskowania, wiążących kategorie lingwistyczne zmiennych wejściowych ('ek' i 'deltaek') z etykietami lingwistycznymi zmiennej wyjściowej ('uk'):

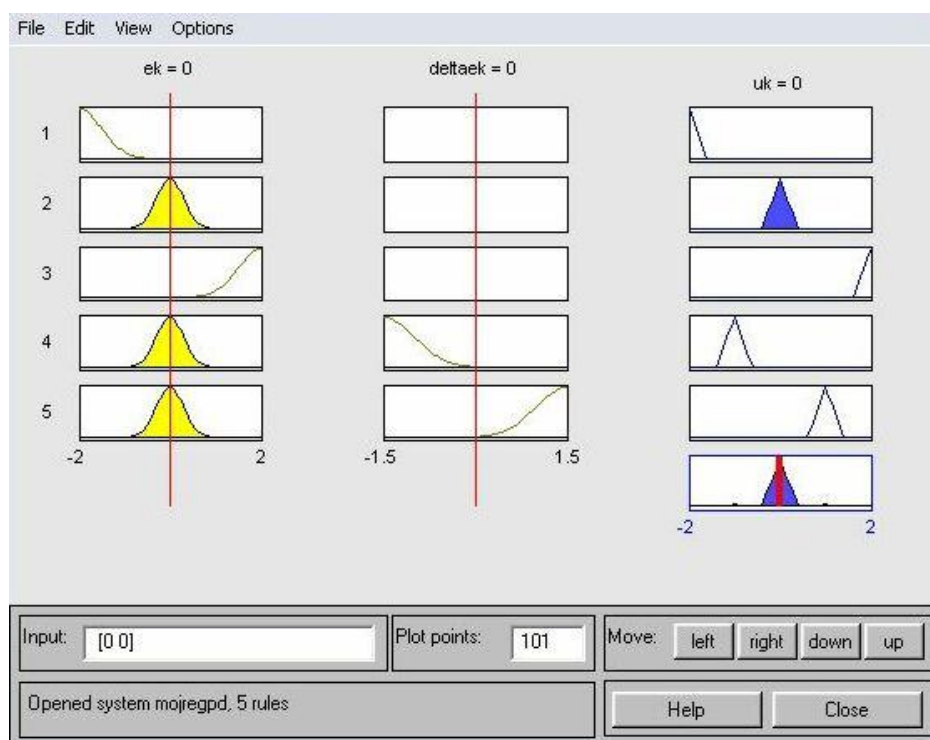
IF (ek IS Lneg) THEN (uk IS Lneg)
IF (ek IS Zerowy) THEN (uk IS Zerowy)
IF (ek IS Lpos) THEN (uk IS Lpos)
IF (ek IS Zerowy) AND (deltaek IS Lneg) THEN (uk IS Mneg)
IF (ek IS Zerowy) AND (deltaek IS Lpos) THEN (uk IS Mpos)

Reguły te w dość uproszczony sposób opisują działanie rozmytego regulatora *PD*. Konstrukcja bazy reguł systemu rozmytego jest bardzo ważnym etapem projektowania i często decyduje o powodzeniu procesu regulacji za pomocą takiego regulatora. Zwykle stosuje się tzw. podejście eksperckie, gdzie źródłem informacji do stworzenia bazy reguł (bazy wiedzy) jest wiedza i doświadczenie operatora, sterującego danym procesem. Tabela 1 przedstawia powyższe reguły w postaci zbliżonej do koncepcji standardowej bazy reguł.

Tabela 1. Baza reguł rozmytego regulatora typu PD

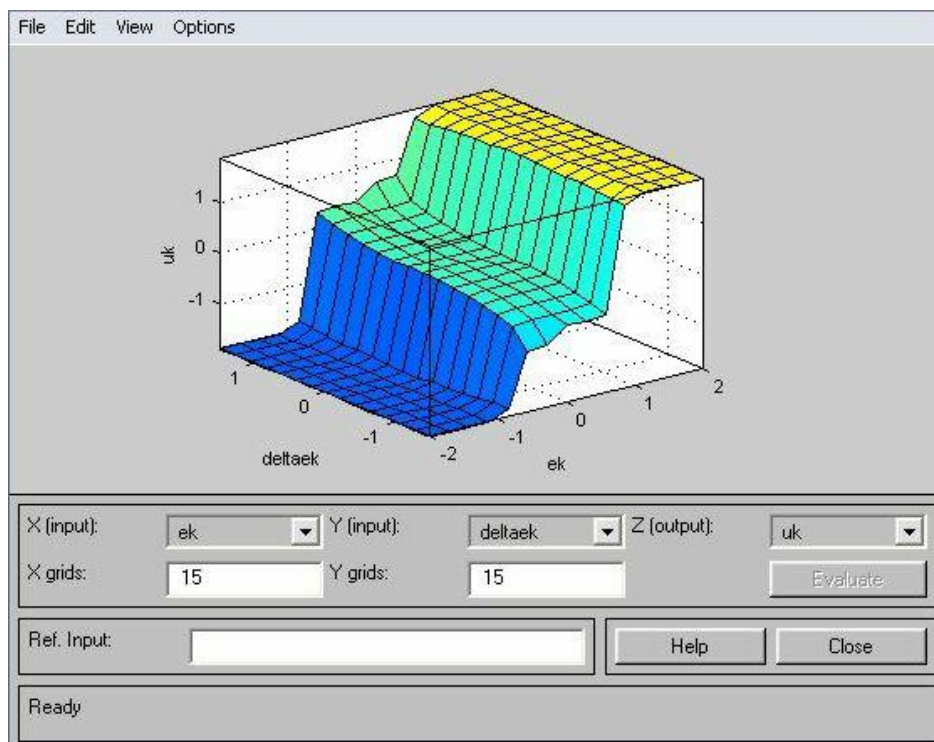
		ek →		
		Lneg	Zerowy	Lpos
deltaek →	Lneg	Lneg	Mneg	Lpos
	Zerowy	Lneg	Zerowy	Lpos
	Lpos	Lneg	Mpos	Lpos

Działanie reguł decyzyjnych można zaobserwować, wywołując okno monitora działania reguł wnioskowania. Dla zaprezentowanego wyżej systemu reprezentującego regulator *PD*, obraz działania reguł wnioskowania przedstawiono na rysunku 7.



Rys. 7. Okno monitora działania reguł w systemie rozmytego regulatora *PD*

Widoczny w prawym dolnym rogu okna rezultat – uaktywniane funkcje przynależności zmiennej wyjściowej ‘*uk*’ sugeruje daleką od ideału pracę regulatora rozmytego. Na rysunku 8 można zaobserwować powierzchnię odwzorowania regulatora rozmytego, zaprojektowanego na podstawie zaprezentowanej wyżej procedury. W przypadku liniowego („konwencjonalnego”) regulatora *PD* powierzchnia ta jest płaszczyzną.



Rys. 8. Powierzchnia odwzorowania rozmytego regulatora *PD*

Działanie zaprojektowanego w powyższy sposób układu regulacji z rozmytym regulatorem typu *PD* jest niezadowalające, co można stwierdzić na podstawie obserwacji przebiegów przejściowych oraz wartości ilościowych wskaźników regulacji!

Kształt powierzchni odwzorowania przypomina działanie regulatora trójpołożeniowego, w dodatku – wpływ zmiennej ‘*deltaek*’ (pochodnej uchybu) na sterowanie jest widoczny jedynie w obszarze małych uchybów, odpowiadających kategorii lingwistycznej ‘*Zerowy*’ dla zmiennej ‘*ek*’. W odpowiedzi na skok wartości zadanej (tj. przy zerowej wartości amplitudy sygnału z generatora funkcyjnego na rys. 2), układ regulacji daje zbyt duże wartości uchybu ustalonego. W przypadku dodania do sygnału zadanego szumu (tj. sygnału losowego z generatora funkcyjnego), oprócz niezerowej wartości uchybu ustalonego innym niekorzystnym efektem jest „nerwowy” przebieg procesu regulacji.

Zachodzi zatem potrzeba starannego dostrojenia regulatora.

2. Układ regulacji z regulatorem rozmytym typu *PI*

2.1. Opis przykładowego układu regulacji z regulatorem rozmytym typu *PI*

W przypadku klasycznego liniowego regulatora typu *PI*, prawo sterowania, które opisuje relację pomiędzy wejściem a wyjściem regulatora, można wyrazić wzorem (odpowiadającym strukturze bloku regulatora *PID* pakietu *MATLAB*):

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau \quad (5)$$

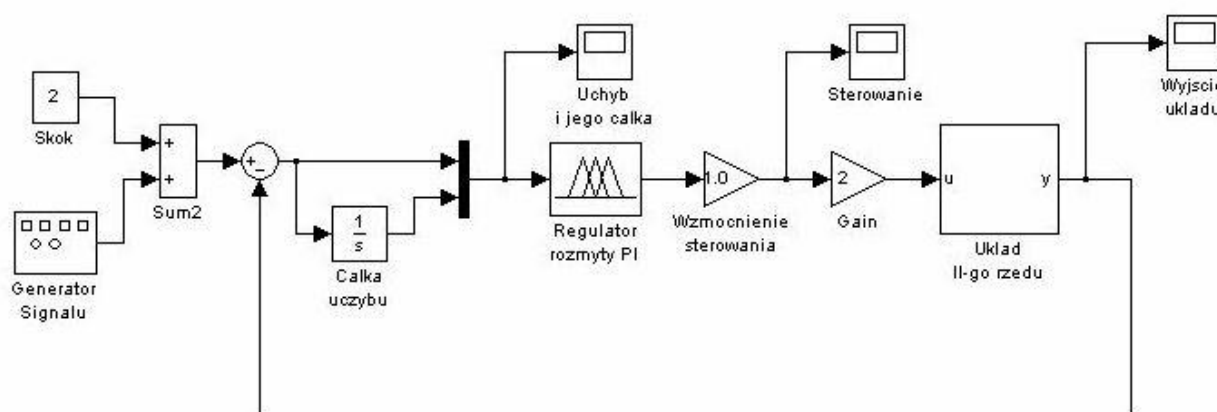
gdzie $e(t)$ jest uchybem regulacji (wejściem regulatora), zaś wyjście regulatora $u(t)$ pobudza wejście obiektu regulacji.

Przykładowy rozmyty regulator *PI*, wykorzystywany w ćwiczeniu, realizuje nieliniowe prawo sterowania:

$$u(t) = f(e(t), \int e(\tau) d\tau) \quad (6)$$

Za pomocą rozmytych reguł wnioskowania **JEŻELI - TO**, prawo sterowania opisuje relacje pomiędzy zmiennymi wejściowymi regulatora: uchybem $e(t)$ i jego całką $\int e(\tau) d\tau$, a wyjściem regulatora – sterowaniem obiektu regulacji, $u(t)$.

Schemat blokowy układu regulacji *PI* z regulatorem rozmytym (w symbolice pakietu *MATLAB/SIMULINK*) przedstawia rysunek 9.



Rys. 9. Schemat układu regulacji z rozmytym regulatorem typu *PI*

Podsystem oznaczony jako ‘*Układ II-go rzędu*’ zawiera obiekt regulacji, przedstawiony na rys. 1. Zmian wartości współczynników a i b modelu układu (równanie (1) lub (2)) można dokonywać po dwukrotnym naciśnięciu („kliknięciu”) myszką na symbol podsystemu, a po otwarciu okienka zawierającego model układu – kolejne dwukrotne „kliknięcie” myszką na symbol wzmacniacza i wpisanie odpowiedniej wartości wzmacnienia w okienku dialogowym.

2.2. Opis regulatora rozmytego typu *PI*

Rozmyty regulator *PI* został utworzony za pomocą edytora systemów rozmytych biblioteki narzędziowej „Fuzzy Logic Toolbox” pakietu *MATLAB*. Wszelkie zmiany elementów systemu rozmytego (realizującego nieliniowe prawo sterowania (4)) mogą być dokonywane wyłącznie za pomocą tego edytora, po czym zaktualizowaną wersję systemu rozmytego należy powtórnie zapisać na dysku.

Uwaga!

Model układu regulacji ‘fuz_pireg’ zrealizowany w pakiecie *SIMULINK* (przedstawiony na rys. 9) wykorzystuje strukturę regulatora, ładowaną do przestrzeni roboczej pakietu *MATLAB* w momencie wywołania (otwarcia) tego modelu. Stąd też po każdorazowej aktualizacji systemu rozmytego ‘mojregpi’ powinno się również zamknąć model ‘fuz_pireg’, albo też równolegle z zapisem modelu na dysk eksportować również uaktualnioną wersję systemu rozmytego do przestrzeni roboczej pakietu *MATLAB*.

Proponowany regulator rozmyty reprezentuje system wnioskowania typu Mamdaniego, o dwóch zmiennych wejściowych: ‘ek’ i ‘calkaek’, które są odpowiednio bieżącymi wartościami uchybu i jego całki oraz o jednej zmiennej wyjściowej ‘uk’, która jest bieżącą wartością sterowania, działającego na obiekt regulacji.

Konstrukcja systemu rozmytego reprezentującego regulator *PI* jest w zasadzie powieleniem systemu opisującego regulator *PD*, z tą różnicą, że zamiast zmiennej wejściowej ‘deltaek’ (występującej w definicji rozmytego regulatora *PD*) w tym regulatorze występuje zmienna wejściowa ‘calkaek’. Z tego względu szczegółowy opis wewnętrznej struktury rozmytego regulatora *PI* zostanie tutaj pominięty.

Jednakże, podobnie jak w przypadku systemu z rozmytym regulatorem *PD*, działanie zaprojektowanego w powyższy sposób układu regulacji z rozmytym regulatorem typu *PI* jest niezadowolające!

Zachodzi zatem potrzeba starannego dostrojenia regulatora.

3. Układ regulacji z regulatorem rozmytym typu *PID*

3.1. Opis przykładowego układu regulacji z regulatorem rozmytym typu *PID*

W przypadku klasycznego liniowego regulatora typu *PID*, prawo sterowania, które opisuje relację pomiędzy wejściem a wyjściem regulatora, można wyrazić wzorem (odpowiadającym strukturze bloku regulatora *PID* pakietu *MATLAB*):

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (7)$$

gdzie $e(t)$ jest uchybem regulacji (wejściem regulatora), zaś wyjście regulatora $u(t)$ pobudza wejście obiektu regulacji.

Przykładowy rozmyty regulator *PID*, wykorzystywany w ćwiczeniu, realizuje nieliniowe prawo sterowania:

$$u(t) = f\left(e(t), \frac{de(t)}{dt}, \int e(\tau) d\tau\right) \quad (8)$$

Za pomocą rozmytych reguł wnioskowania **JEŻELI - TO**, powyższe prawo sterowania opisuje relacje pomiędzy zmiennymi wejściowymi regulatora: uchybem $e(t)$, jego pochodną $de(t)/dt$ i jego całką $\int e(\tau) d\tau$, a wyjściem regulatora – sterowaniem obiektu regulacji, $u(t)$.

3.2. Opis regulatora rozmytego typu *PID*

Rozmyty regulator *PID* może być utworzony za pomocą edytora systemów rozmytych biblioteki narzędziowej „Fuzzy Logic Toolbox” pakietu *MATLAB*. W tym celu należy zbudować odpowiednią bazę reguł decyzyjnych, uwzględniającą trzy (a nie jak w poprzednich przypadkach – dwie) zmienne wejściowe systemu.

Tworzenie, aktualizacja parametrów i stosowanie rozmytego regulatora *PID* podlega tym wszystkim zasadom, które zostały podane w punktach 1, 2. i 2. 2. w odniesieniu do systemów rozmytych z regulatorami typu *PD* i *PI*.