

Logika rozmyta

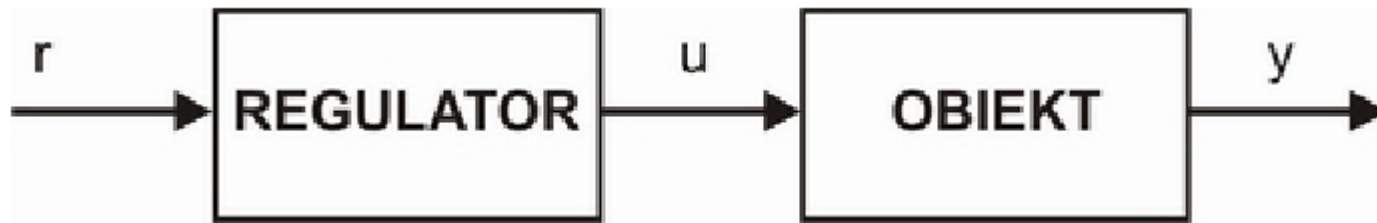
Oleksandr Sokolov

Wydział Fizyki, Astronomii i Informatyki Stosowanej
UMK

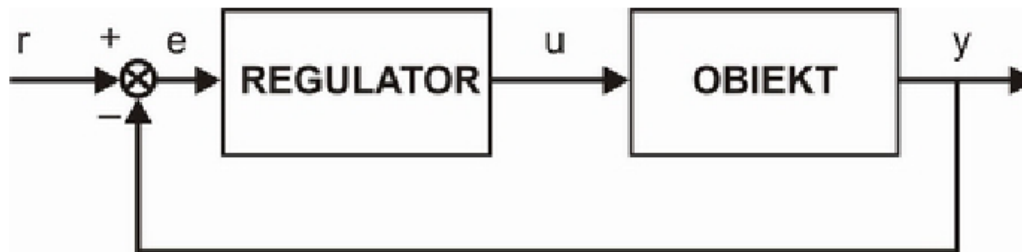
<http://fizyka.umk.pl/~osokolov/LR/>

Sterowanie z wykorzystaniem logiki rozmytej

Układ otwarty



Układ zamknięty



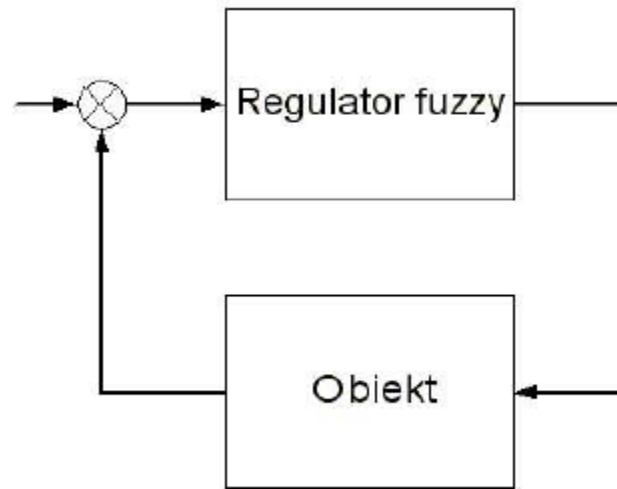
Projektowanie regulatorów rozmytych w środowisku MATLAB-Simulink

W *Fuzzy Logic Toolbox* są dostępne dwa podstawowe modele rozmyte: **Mamdani i Sugeno**. Są one opisane zbiorem reguł. Każda reguła składa się z *przesłanki* (część **Jeśli**) i konkluzji (**część to**). **Przesłanka zawiera zbiór warunków, a konkluzja wniosek.**

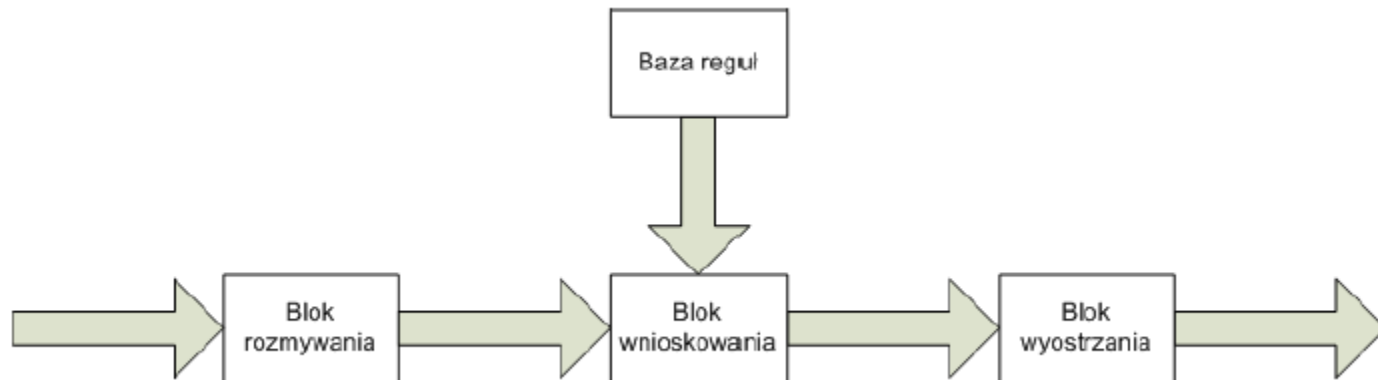
Reguły obu modeli mają identyczną postać przesłanek, różnią się konkluzjami. W modelu rozmytym typu Mamdani konkluzja reguł zawiera zbiór rozmyty. Model typu Sugeno w konkluzji ma funkcję.

Układ zamknięty

Układ sterowania z rozmytym regulatorem



Schemat regulatora rozmytego



Regulatory rozmyte PI typu Mamdani

Rozmyte regulatory PI typu Mamdani zwykle używają dwóch zmiennych wejściowych $E(\text{error})$ i $SE(\text{sum of errors} - \text{całka błędu})$ oraz jednej zmiennej wyjściowej $U(\text{output})$. Baza reguł regulatora rozmytego PI typu Mamdani przyjmuje postać jak poniżej

Jeśli (E jest A_j) i (SE jest B_j), to (U jest C_j)

gdzie: $j = 1, 2, \dots, K$, K – liczba reguł modelu, A_i , B_j , C_j – zbiory rozmyte jako wartości lingwistyczne (np. small, large itp.).

Model rozmyty typu Mamdani często jest stosowany w układach opartych na wiedzy eksperta. Jest to model:

- tworzony intuicyjnie

- powszechnie znany i akceptowany

- dobrze dostosowany do sposobu postrzegania zmiennych przez człowieka.

Rozmyty regulator PID

P

- Jeżeli błąd e jest negatywnie duży, to sterowanie u negatywnie duże;
- Jeżeli błąd e negatywnie mały, to sterowanie u negatywnie małe;
- Jeżeli błąd $e = 0$, to sterowanie $u = 0$;
- Jeżeli błąd e pozytywnie mały, to sterowanie u pozytywnie małe;
- Jeżeli błąd e pozytywnie duży, to sterowanie u pozytywnie duże.

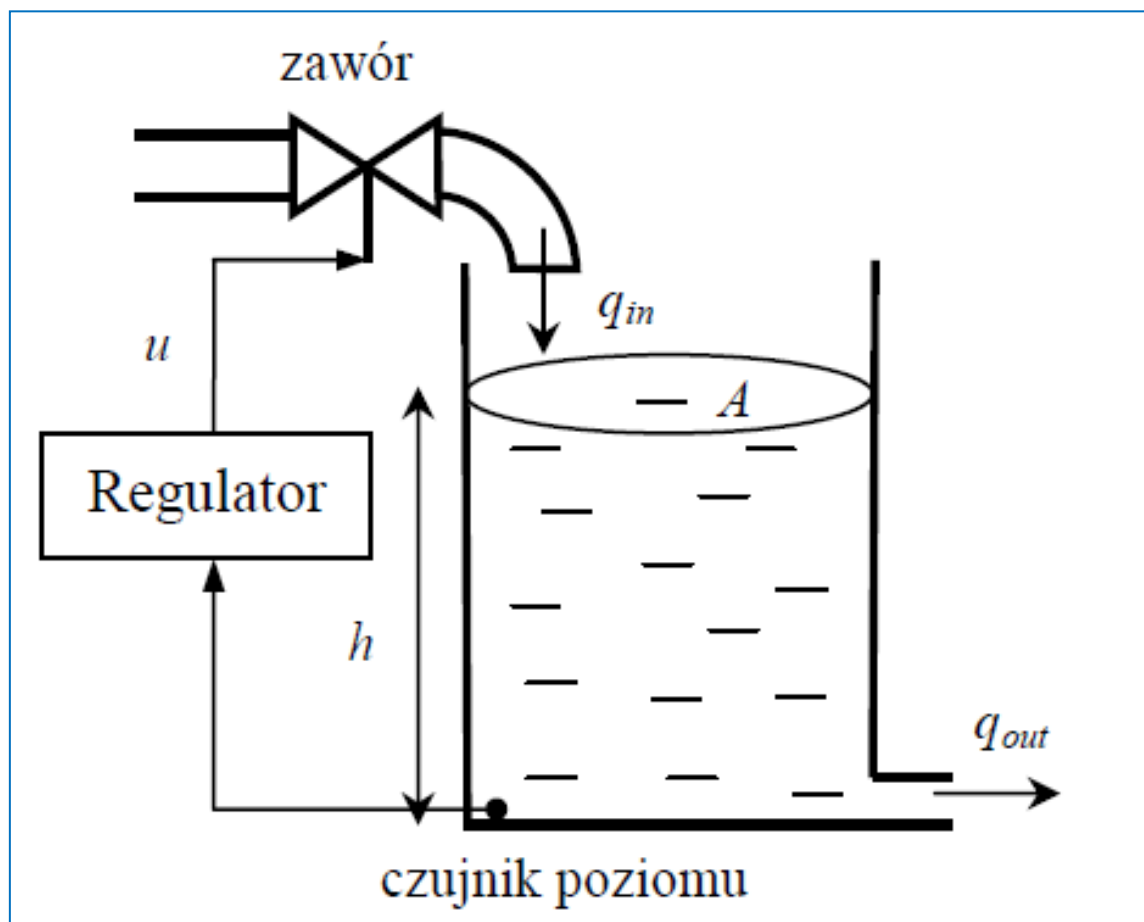
PD regulator

Lub, przedstawiając w formie tabelki:

- IF $e(t) = N$ AND $\dot{e}(t) = N$ THEN $u(t) = N$;
- IF $e(t) = N$ AND $\dot{e}(t) = Z$ THEN $u(t) = N$;
- IF $e(t) = Z$ AND $\dot{e}(t) = N$ THEN $u(t) = N$;
- IF $e(t) = N$ AND $\dot{e}(t) = P$ THEN $u(t) = Z$;
- IF $e(t) = Z$ AND $\dot{e}(t) = Z$ THEN $u(t) = Z$;
- IF $e(t) = P$ AND $\dot{e}(t) = N$ THEN $u(t) = Z$;
- IF $e(t) = Z$ AND $\dot{e}(t) = P$ THEN $u(t) = P$;
- IF $e(t) = P$ AND $\dot{e}(t) = Z$ THEN $u(t) = P$;
- IF $e(t) = P$ AND $\dot{e}(t) = P$ THEN $u(t) = P$;

sterowanie $u(t)$		$\dot{e}(t)$		
		N	Z	P
$e(t)$	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

Schemat układu sterowania poziomem cieczy w zbiorniku z wypływem



Model matematyczny

Zmiana objętości V (m^3) cieczy w zbiorniku o stałym polu przekroju A (m^2) jest opisana *nieliniowym* równaniem różniczkowym:

$$\frac{dV}{dt} = A \frac{dh}{dt} = q_{in} - q_{out} ,$$

gdzie: h jest wysokością słupa cieczy w zbiorniku (wielkość sterowana, m), q_{in} – objętościowym strumieniem wpływającym (m^3/s), q_{out} – objętościowym strumieniem wypływającym ze zbiornika (m^3/s).

Strumień wypływający zależy o ciśnienia słupa cieczy w zbiorniku $\Delta p = \rho g h$ (ρ – gęstość cieczy, kg/m^3 , g – przyspieszenie ziemskie, $9.8 \text{ m}/\text{s}^2$) i jest opisany równaniem Bernoulliego:

$$q_{out} = \sqrt{\frac{2 \cdot \Delta p}{\rho}} \cdot S_{out} = \sqrt{2gh} \cdot S_{out} ,$$

Model matematyczny

gdzie S_{out} jest polem przekroju wylotu zbiornika (m^2). Po przekształceniu otrzymujemy równanie dynamiki zmian poziomu zamodelowane w bloku FLUID TANK schematu Simulinka:

$$\frac{dh}{dt} = \frac{1}{A} (q_{in} - S_{out} \sqrt{2gh})$$

Równanie zaworu (blok VALVE) regulującego w sposób ciągły strumień wpływający:

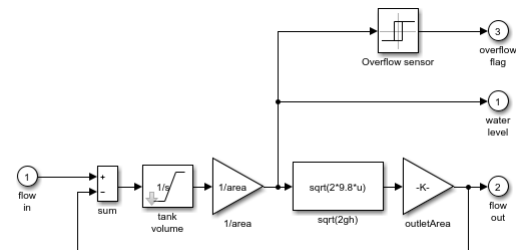
$$q_{in} = q_{in_max} \cdot \int_0^t u d\tau$$

gdzie $q_{in_max} = \text{const}$ to określony dopływ maksymalny, jest równaniem *układu całkującego z ograniczeniem całkowania* sygnału sterującego do zakresu $[0, 1]$. W związku z tym regulator jest *regulatorem przyrostowym*, tzn. dla sygnału sterującego $u=0$ strumień wpływający pozostaje stały: $q_{in} = \text{const}$.

Sygnałem wejściowym regulatora jest błąd poziomu cieczy

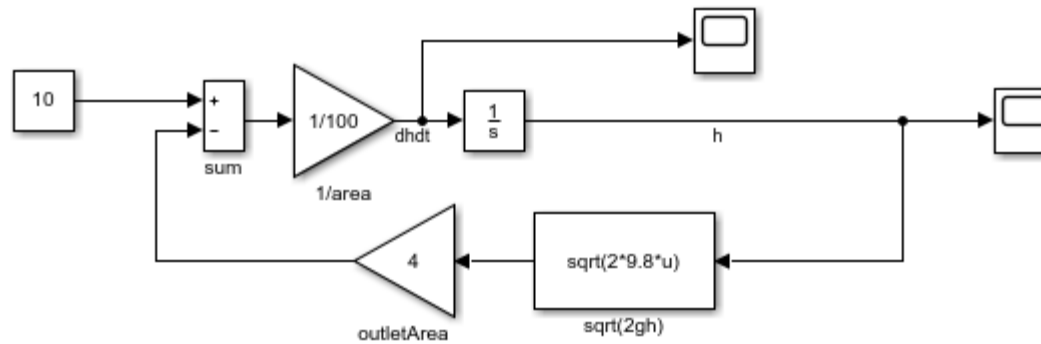
$$e = h_{zad} - h,$$

[\lit\control\ZNWEiAPG_51-17.pdf](#)



Symulacja

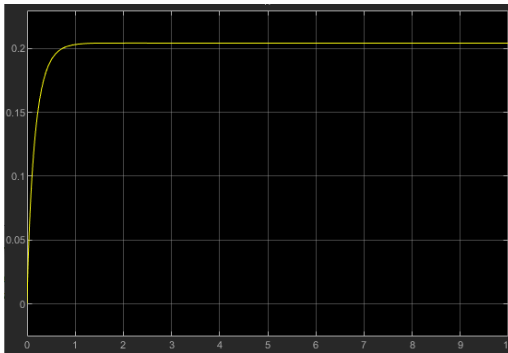
$$\frac{dh}{dt} = \frac{1}{A} \left(q_{in} - S_{out} \sqrt{2gh} \right)$$



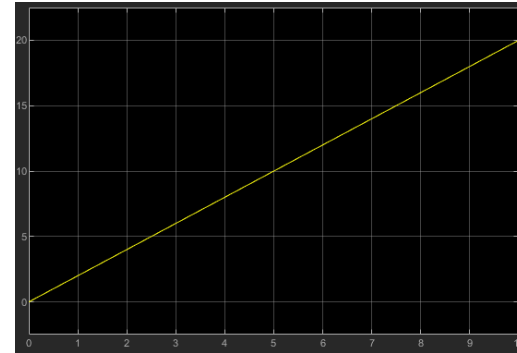
tankOS.slx

Symulacja (cd)_{Sout=0}

Sout=A

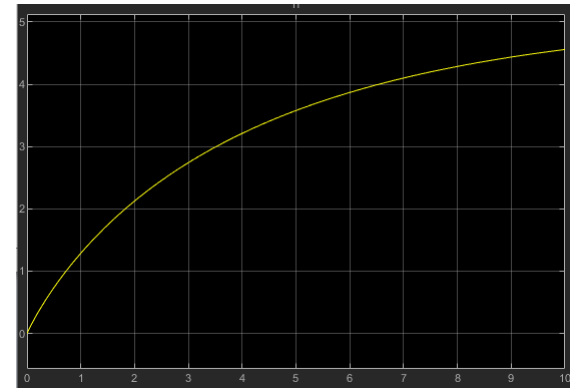
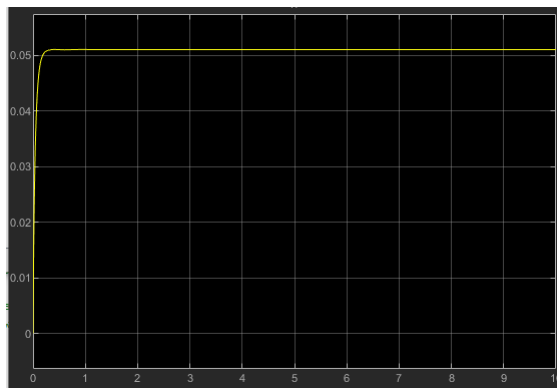


$$\frac{dh}{dt} = \frac{1}{A} (q_{in} - S_{out} \sqrt{2gh})$$



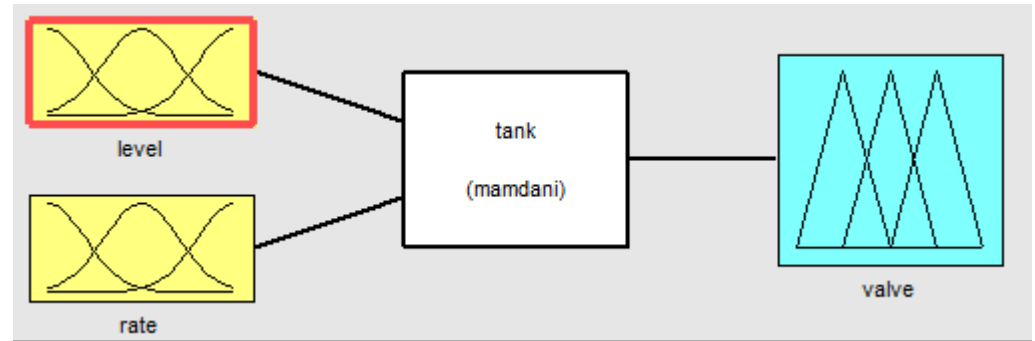
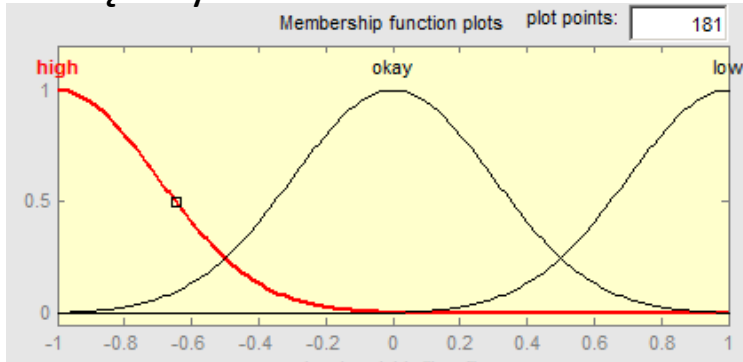
Sout<A

Sout>A

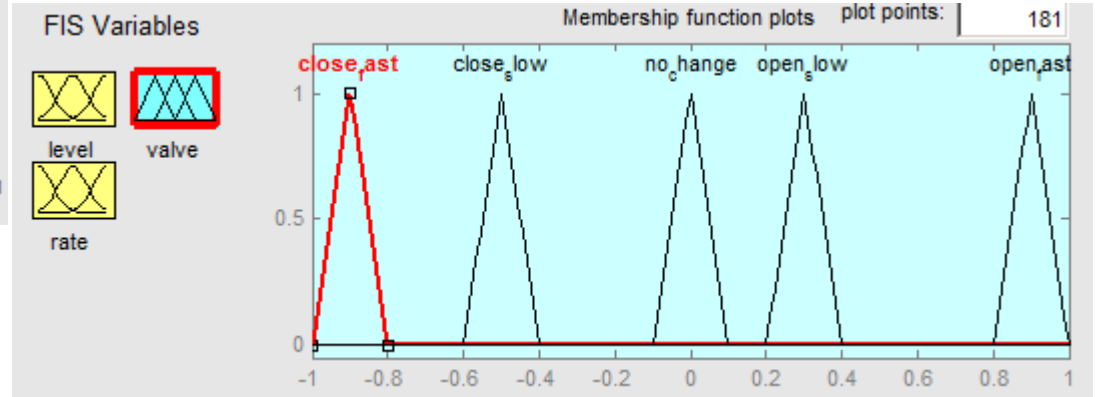
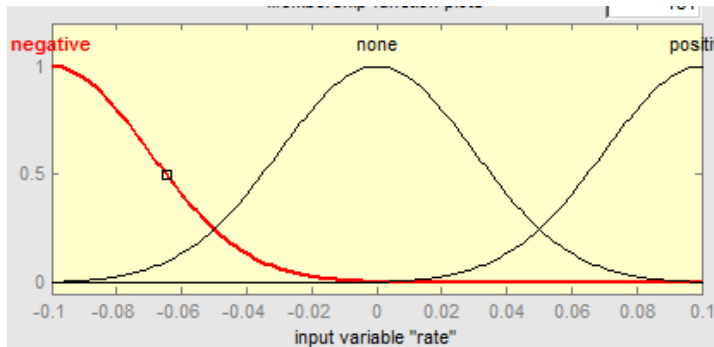


Sterownik rozmyty

Błąd wysokości



Prędkość błędu

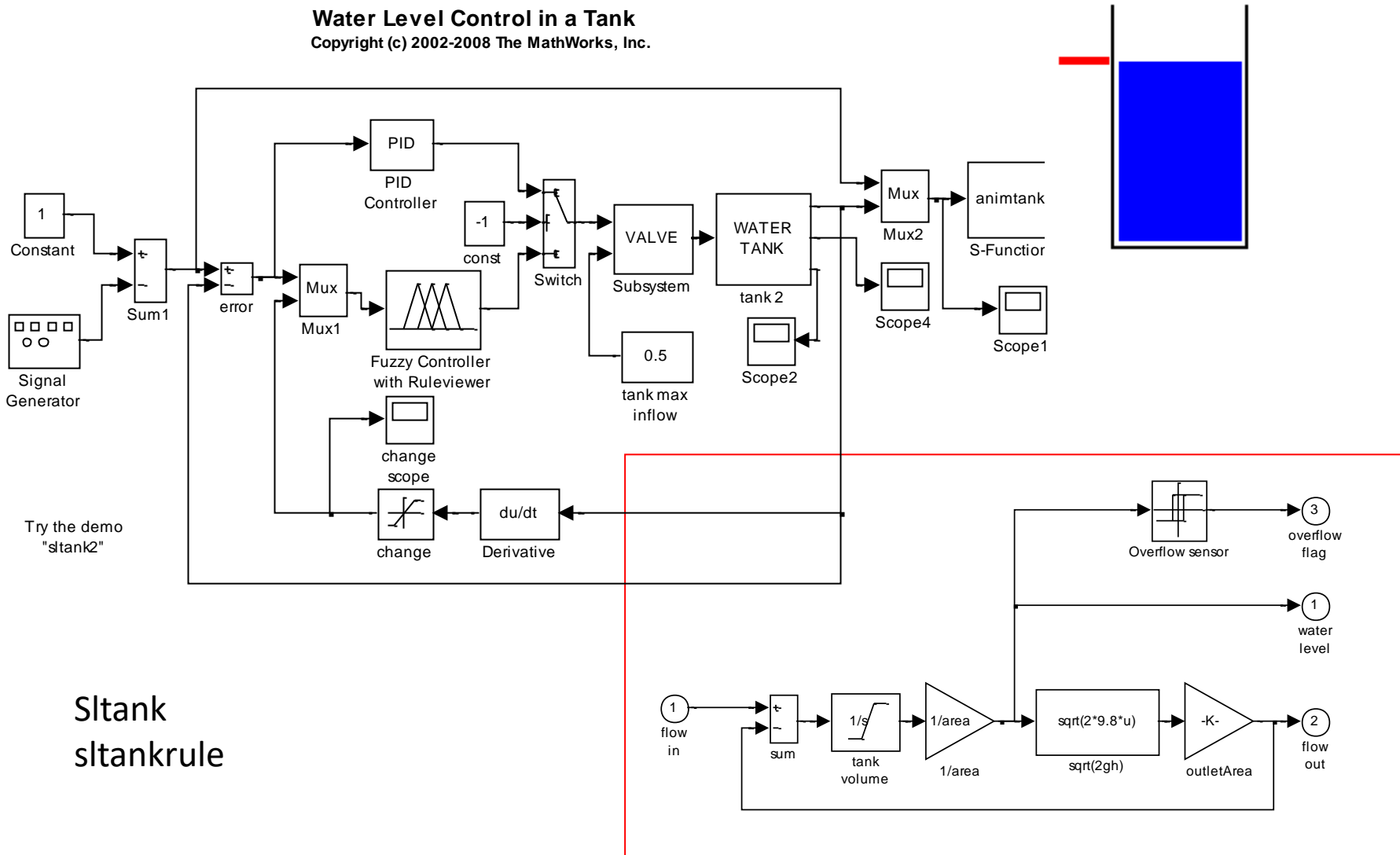


Otworzyć

Zamknąć

Sterowanie zbiornikiem z wypływem

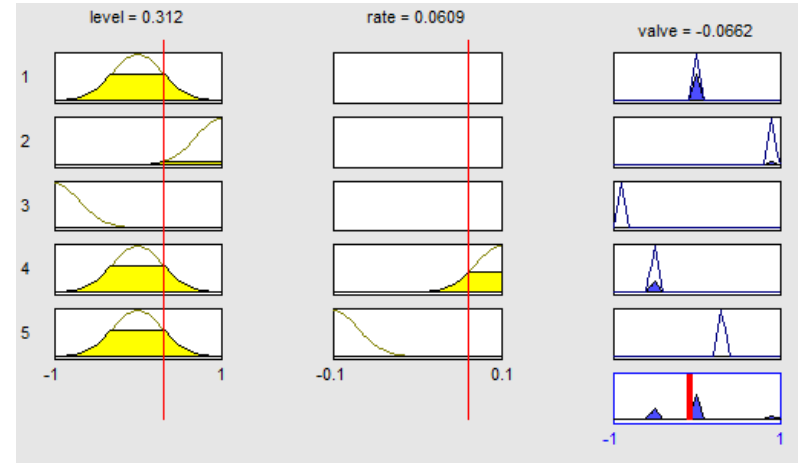
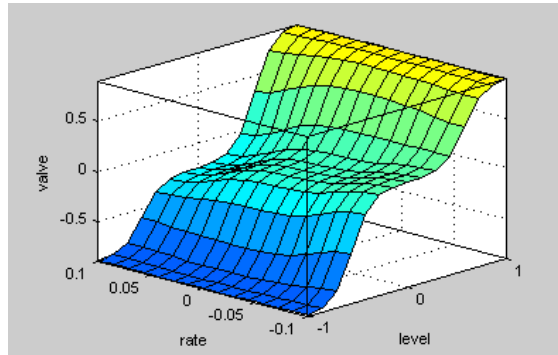
Water Level Control in a Tank
Copyright (c) 2002-2008 The MathWorks, Inc.



Try the demo "stank2"

Stank
sltankrule

Baza reguł



Ze względu na średnicę rury odpływowej zbiornik wody w tym systemie opróżnia się wolniej niż się napełnia. Aby zrekompensować ten brak równowagi, funkcje przynależności w zaworach **close_slow** i **open_slow** nie są symetryczne. **Regulator PID nie obsługuje takiej asymetrii.**

System rozmytych ma pięć reguł.

Pierwsze trzy reguły dostosowują zawór na podstawie **tylko błędu poziomu wody**.

Jeśli poziom wody jest prawidłowy, nie reguluj zaworu.

Jeśli poziom wody jest niski, szybko otwórz zawór.

Jeśli poziom wody jest wysoki, szybko zamknij zawór.

Pozostałe dwie reguły regulują zawór w oparciu o szybkość zmiany poziomu wody, gdy poziom wody jest bliski wartości zadanej.

Jeśli poziom wody jest w porządku i rośnie, powoli zamknij zawór.

Jeśli poziom wody jest w porządku i spada, powoli otwórz zawór.

Struktura fis pliku

```
% $Revision: 1.1 $  
[System]  
Name='tank'  
Type='mamdani'  
NumInputs=2  
NumOutputs=1  
NumRules=5  
AndMethod='prod'  
OrMethod='probor'  
ImpMethod='prod'  
AggMethod='max'  
DefuzzMethod='centroid'
```

```
[Input1]  
Name='level'  
Range=[-1 1]  
NumMFs=3  
MF1='high': 'gaussmf', [0.3 -1]  
MF2='okay': 'gaussmf', [0.3 0]  
MF3='low': 'gaussmf', [0.3 1]
```

```
[Input2]  
Name='rate'  
Range=[-0.1 0.1]  
NumMFs=3  
MF1='negative': 'gaussmf', [0.03 -0.1]  
MF2='none': 'gaussmf', [0.03 0]  
MF3='positive': 'gaussmf', [0.03 0.1]
```

```
[Output1]  
Name='valve'  
Range=[-1 1]  
NumMFs=5  
MF1='close_fast': 'trimf', [-1 -0.9 -0.8]  
MF2='close_slow': 'trimf', [-0.6 -0.5 -0.4]  
MF3='no_change': 'trimf', [-0.1 0 0.1]  
MF4='open_slow': 'trimf', [0.2 0.3 0.4]  
MF5='open_fast': 'trimf', [0.8 0.9 1]
```

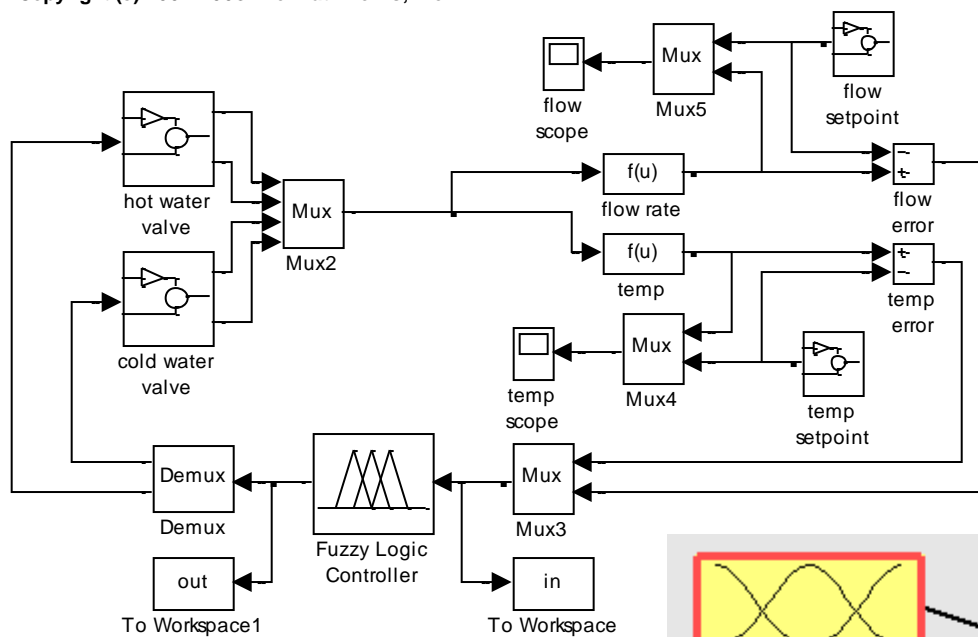
```
[Rules]  
2 0, 3 (1) : 1  
3 0, 5 (1) : 1  
1 0, 1 (1) : 1  
2 3, 2 (1) : 1  
2 1, 4 (1) : 1
```

Sterowanie prysznicem

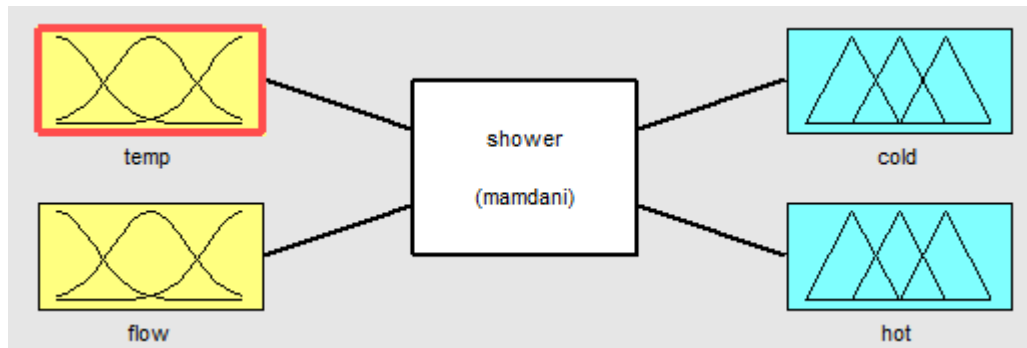
Temperature Control in a Shower

Temperature Control in a Shower
Copyright (c) 2002-2008 The MathWorks, Inc.

fuzzyshowerdemo.m



strumień



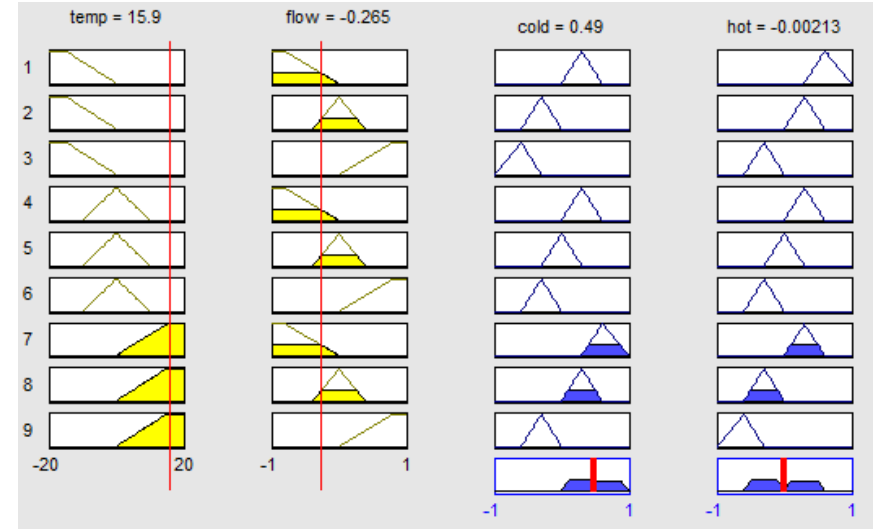
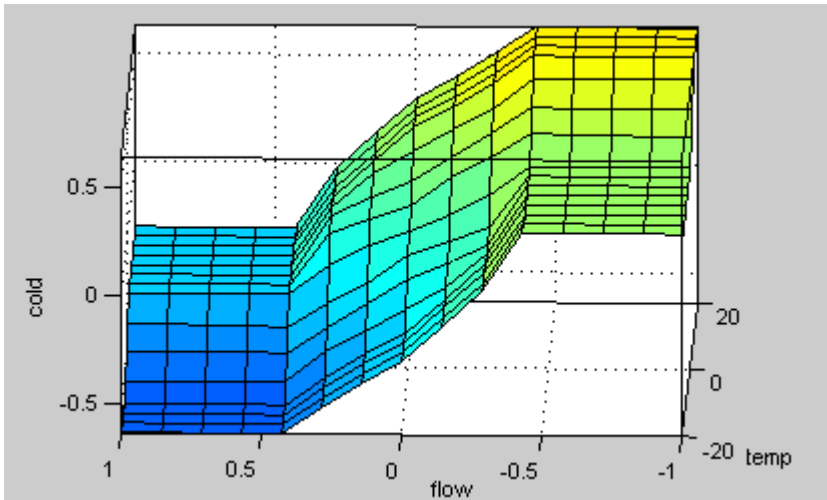
Opis

W tym systemie kontrolujesz natężenie przepływu i temperaturę prysznica poprzez regulację zaworów ciepłej i zimnej wody. Ponieważ istnieją dwa wejścia dla systemu rozmytego, model konkatenuje sygnały wejściowe za pomocą bloku Mux. Wyjście bloku Mux jest podłączone do wejścia bloku Fuzzy Logic Controller. Podobnie dwa sygnały wyjściowe są uzyskiwane za pomocą bloku Demux podłączonego do kontrolera.

W systemie rozmytym obowiązuje dziewięć zasad regulacji zaworów ciepłej i zimnej wody w oparciu o błędy przepływu i temperatury. Reguły dostosowują całkowite natężenie przepływu na podstawie błędu przepływu i dostosowują względne natężenia przepływu gorącego i zimnego na podstawie błędu temperatury.

Baza reguł

krany

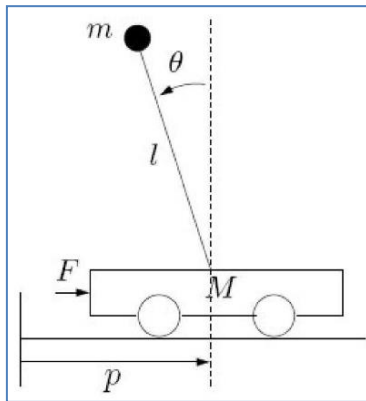


- 1 "temp==cold & flow==soft => cold=openSlow, hot=openFast (1)"
- 2 "temp==cold & flow==good => cold=closeSlow, hot=openSlow (1)"
- 3 "temp==cold & flow==hard => cold=closeFast, hot=closeSlow (1)"
- 4 "temp==good & flow==soft => cold=openSlow, hot=openSlow (1)"
- 5 "temp==good & flow==good => cold=steady, hot=steady (1)"
- 6 "temp==good & flow==hard => cold=closeSlow, hot=closeSlow (1)"
- 7 "temp==hot & flow==soft => cold=openFast, hot=openSlow (1)"
- 8 "temp==hot & flow==good => cold=openSlow, hot=closeSlow (1)"
- 9 "temp==hot & flow==hard => cold=closeSlow, hot=closeFast (1)"

Sterowanie odwróconym wahadłem

MatLab - slcp

Układ składa się z wózka i zamocowanego na elastycznym przegubie pionowego wahadła o masie m i długości l



Zachowanie systemu opisuje równanie różniczkowe

$$-ml^2 \frac{d^2\theta}{dt^2} + mlg \cdot \sin \theta = \tau = u(t)$$

gdzie $\tau = u(t)$ jest sterowaniem (moment obrotowy), które należy przyłożyć w kierunku przeciwnym do wychylenia θ aby utrzymać wahadło w pozycji pionowej.

Model obiektu

Oznaczając $x_1 = \theta$, $x_2 = d\theta/dt$ sprowadzamy powyższe równanie do układu równań opisującego zachowanie badanego układu w przestrzeni stanów

$$\begin{aligned} dx_1/dt &= x_2 \\ dx_2/dt &= (g/l) \sin x_1 - (1/ml^2)u(t) \end{aligned}$$

Dla małych wartości θ mamy $\sin \theta \approx \theta$, co pozwala zlinearyzować powyższy układ

$$\begin{aligned} dx_1/dt &= x_2 \\ dx_2/dt &= (g/l)x_1 - (1/ml^2)u(t) \end{aligned}$$

Jeżeli x_1 jest mierzone w stopniach, a x_2 w stopniach na sekundę (*sns*), to przyjmując $l = g$, $m = 180/(\pi \cdot g^2)$ uzyskujemy następujący różnicowy odpowiednik powyższego układu

$$\begin{aligned} x_1(t+1) &= x_1(t) + x_2(t) \\ x_2(t+1) &= x_1(t) + x_2(t) - u(t) \end{aligned}$$

Przestrzeni zmiennych

Przyjmijmy, że $-2^\circ \leq x_1 \leq 2^\circ$, $-5sns \leq x_2 \leq 5sns$. Konstrukcja sterownika składa się z następujących kroków.

1. Dziedzinę zmiennej wejściowej x_1 pokrywamy trzema zbiorami rozmytymi: $\mu_{NEG} = \Delta(-\infty, -2, 0)$, $\mu_{Zero} = \Delta(-2, 0, 2)$, $\mu_{POS} = \Delta(0, 2, \infty)$. Podobnie, trzema zbiorami rozmytymi, lecz o innych nośnikach, pokrywamy dziedzinę zmiennej x_2 : $\mu_{NEG} = \Delta(-\infty, -5, 0)$, $\mu_{Zero} = \Delta(-5, 0, 5)$, $\mu_{POS} = \Delta(0, 5, \infty)$.
2. Pokrywamy dziedzinę zmiennej sterującej u pięcioma funkcjami przynależności: $\mu_{NB} = \Delta(-\infty, -24, -16)$, $\mu_N = \Delta(-16, -8, 0)$, $\mu_Z = \Delta(-8, 0, 8)$, $\mu_P = \Delta(0, 8, 16)$, $\mu_{PB} = \Delta(16, 24, \infty)$.

Baza rozmytych reguł

	NEG	Zero	POS
NEG	NB	N	Z
Zero	N	Z	P
POS	Z	P	PB

Właściwe sterowanie. Przypuśćmy, że w chwili $t = 0$ wahadło znajduje się w punkcie $x_1 = 1^\circ$, $x_2 = -4stn$. Punkt ten uaktywnia następujące reguły

- if* x_1 is Pos and x_2 is Zero then u is P $\min(0.5, 0.2) = 0.2$ (P)
- if* x_1 is Pos and x_2 is Neg then u is Z $\min(0.5, 0.8) = 0.5$ (Z)
- if* x_1 is Zero and x_2 is Zero then u is Z $\min(0.5, 0.2) = 0.2$ (Z)
- if* x_1 is Zero and x_2 is Neg then u is N $\min(0.5, 0.8) = 0.5$ (N)

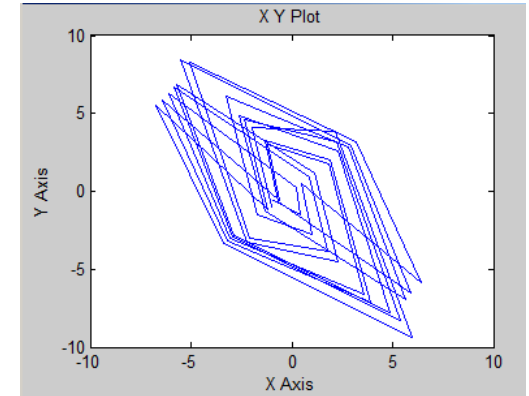
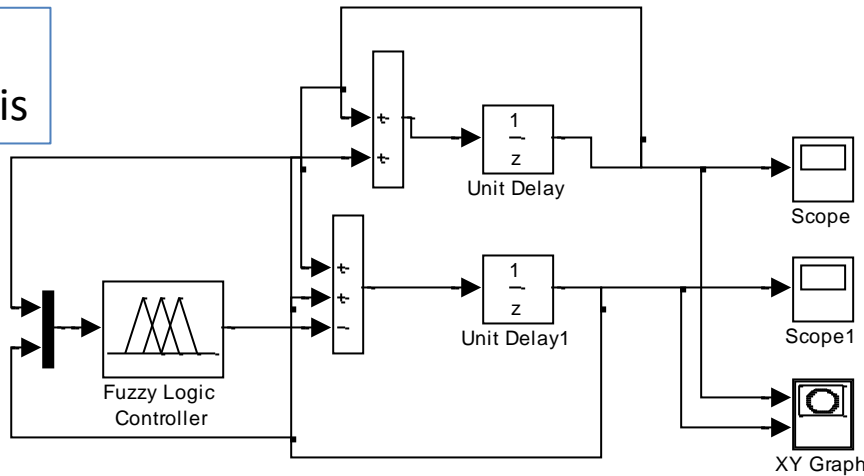
Ostatecznie konkluzji N przypisujemy stopień 0.5, konkluzji Z stopień $\max(0.5, 0.2) = 0.5$ i konkluzji P stopień 0.2. Wyostrzając wynikowy zbiór rozmyty stwierdzamy, że $u = -2$. Możemy więc wyznaczyć kolejne położenie wahadła w przestrzeni stanów

$$x_1(1) = x_1(0) + x_2(0) = 1 - 4 = -3$$

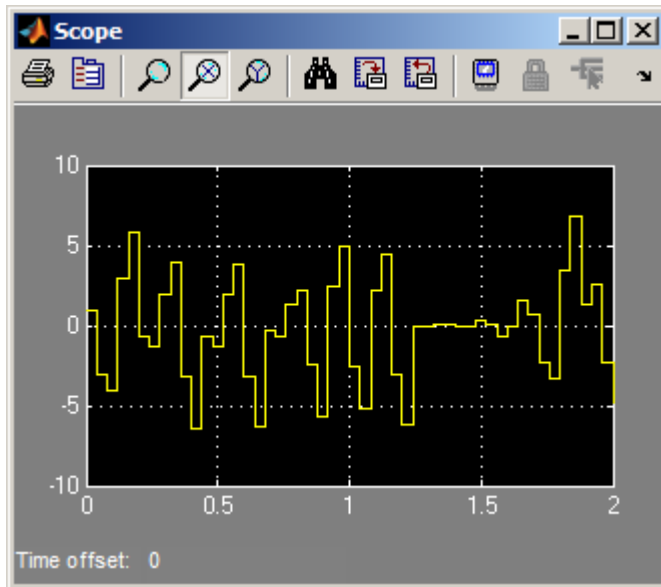
$$x_2(1) = x_1(0) + x_2(0) - u(0) = 1 - 4 - (-2) = -1$$

Realizacja w Matlabie

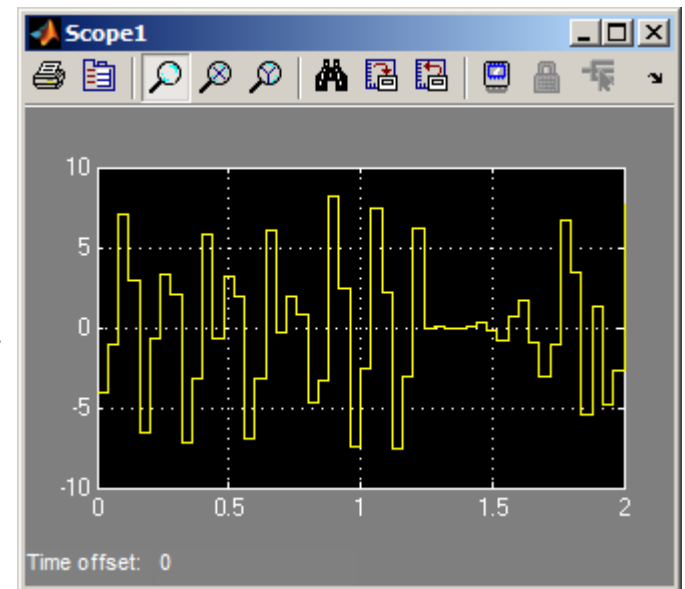
wah.m
Wahadlo.fis



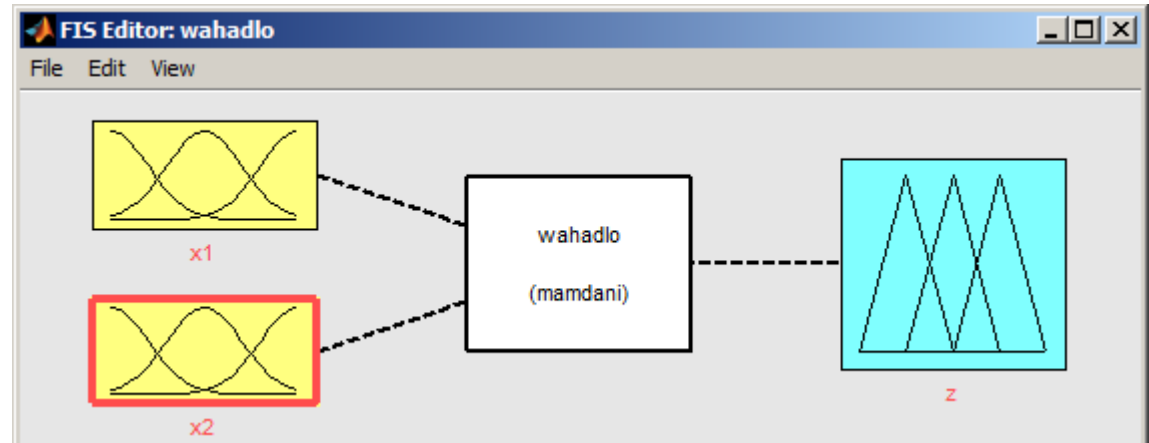
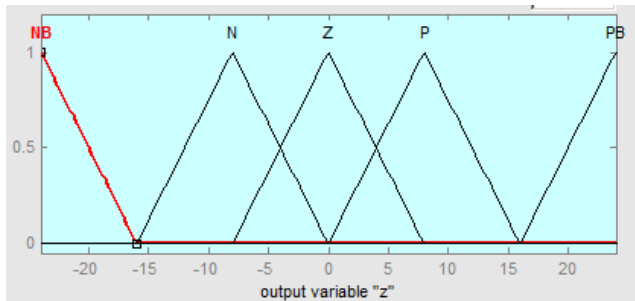
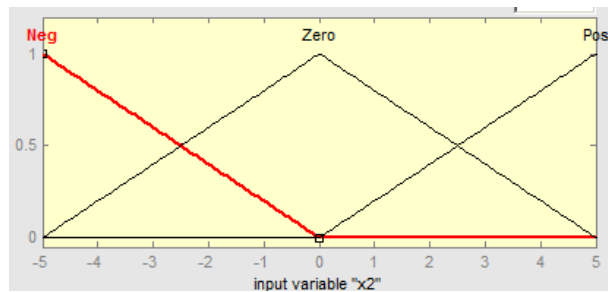
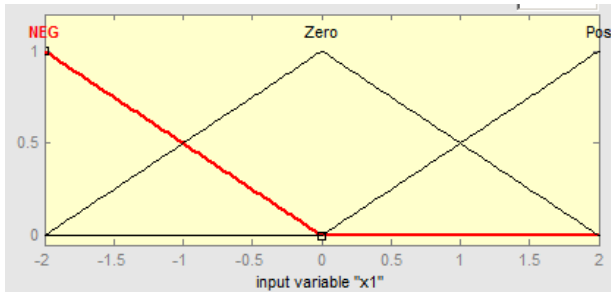
$x_1(0)=1$
 $x_1(1)=-3$



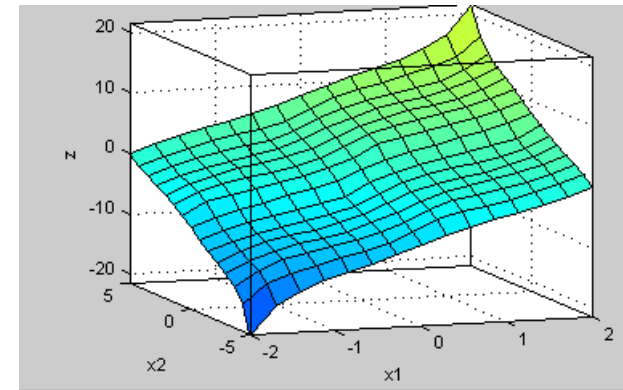
$x_2(0)=-4$
 $x_2(1)=-1$



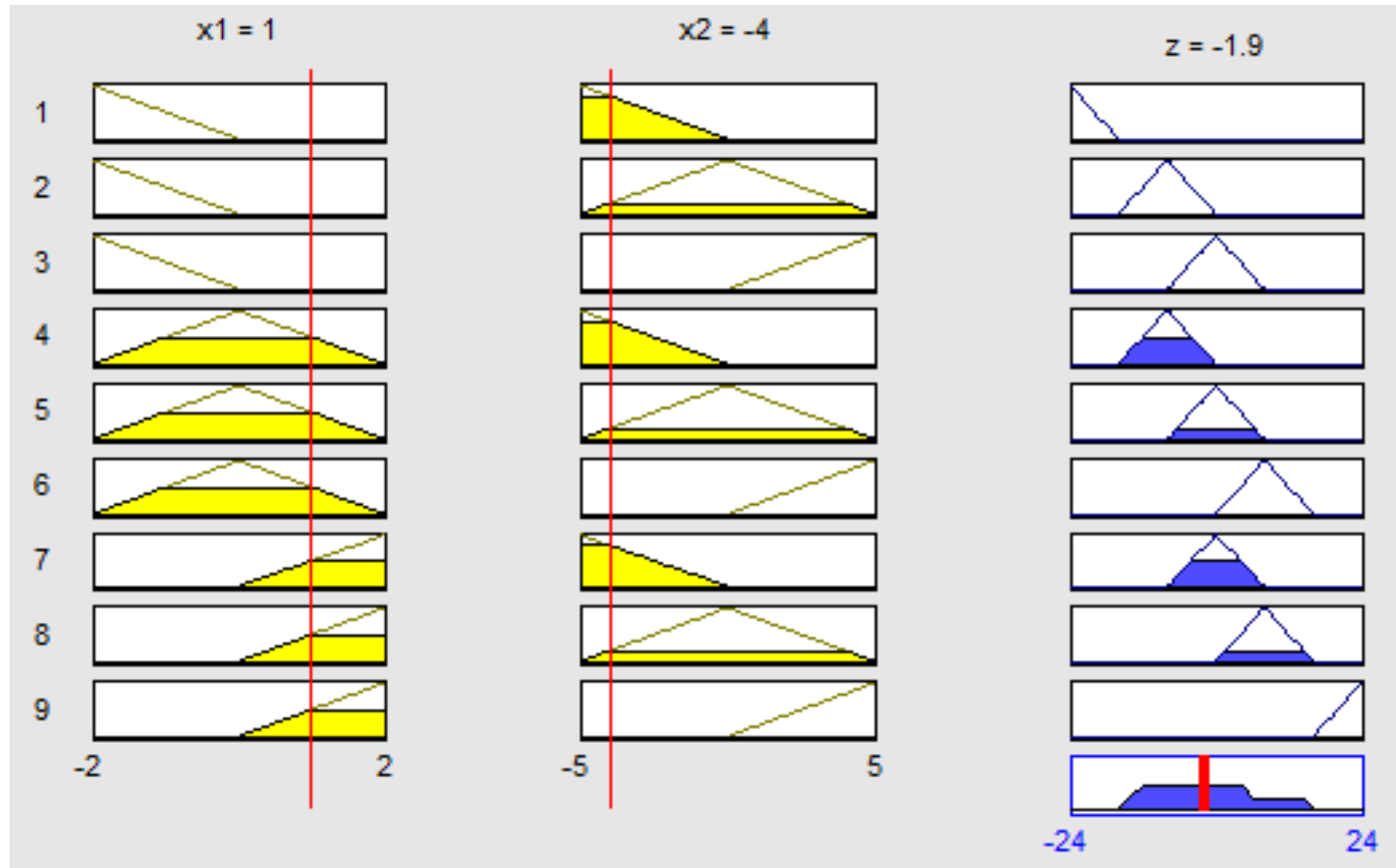
Sterownik rozmyty Mamdaniego



1. If (x1 is NEG) and (x2 is Neg) then (z is NB) (1)
2. If (x1 is NEG) and (x2 is Zero) then (z is N) (1)
3. If (x1 is NEG) and (x2 is Pos) then (z is Z) (1)
4. If (x1 is Zero) and (x2 is Neg) then (z is N) (1)
5. If (x1 is Zero) and (x2 is Zero) then (z is Z) (1)
6. If (x1 is Zero) and (x2 is Pos) then (z is P) (1)
7. If (x1 is Pos) and (x2 is Neg) then (z is Z) (1)
8. If (x1 is Pos) and (x2 is Zero) then (z is P) (1)
9. If (x1 is Pos) and (x2 is Pos) then (z is PB) (1)



Wnioskowanie rozmyte



Wykorzystanie poleceń

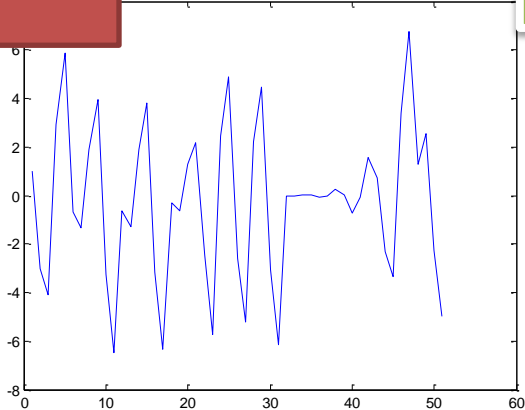
```
%twoz FIS
wah=newfis('wahControl');
wah=addvar(wah,'input','x1',[-2 2]);
    wah=addmf(wah,'input',1,'NEG','trimf',[-3.6 -2 0]);
    wah=addmf(wah,'input',1,'Zero','trimf',[-2 0 2]);
    wah=addmf(wah,'input',1,'Pos','trimf',[0 2 3.6] );

wah=addvar(wah,'input','x2',[-5 5]);
    wah=addmf(wah,'input',2,'NEG','trimf',[-9 -5 0]);
    wah=addmf(wah,'input',2,'Zero','trimf',[-5 0 5]);
    wah=addmf(wah,'input',2,'Pos','trimf',[0 5 9] );
wah=addvar(wah,'output','z',[-24 24]);
    wah=addmf(wah,'output',1,'NB','trimf',[-36 -24 -16] );
    wah=addmf(wah,'output',1,'N','trimf',[-16 -8 0]);
    wah=addmf(wah,'output',1,'Z','trimf',[-8 0 8]);
    wah=addmf(wah,'output',1,'P','trimf',[0 8 16] );
    wah=addmf(wah,'output',1,'PB','trimf',[16 24 36] );
als.values)
```

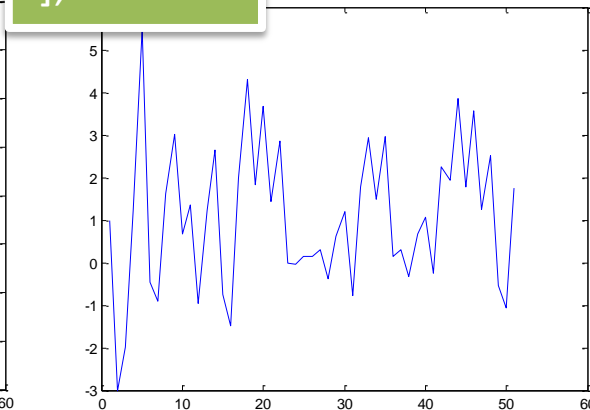
```
ruleList=[
1 1 1 1 1
1 2 2 1 1
1 3 3 1 1
2 1 2 1 1
2 2 3 1 1
2 3 4 1 1
3 1 3 1 1
3 2 4 1 1
3 3 5 1 1
];
wah = addrule(wah,ruleList);
sim('wah1');
plot(x1.signals.values)
```

Analiza wyników

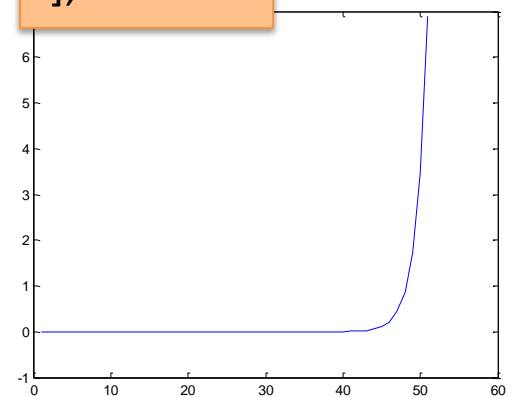
```
ruleList=[  
1 1 1 1 1  
1 2 2 1 1  
1 3 3 1 1  
2 1 2 1 1  
2 2 3 1 1  
2 3 4 1 1  
3 1 3 1 1  
  
3 2 4 1 1  
3 3 5 1 1  
];
```



```
ruleList=[  
1 1 1 1 1  
1 2 2 1 1  
1 3 3 1 1  
2 1 2 1 1  
2 2 3 1 1  
2 3 4 1 1  
3 1 3 1 1  
  
3 2 3 1 1  
3 3 5 1 1  
];
```



```
ruleList=[  
1 1 1 1 1  
1 2 2 1 1  
1 3 3 1 1  
2 1 2 1 1  
2 2 3 1 1  
2 3 4 1 1  
3 1 3 1 1  
  
3 2 2 1 1  
3 3 5 1 1  
];
```




```
for i1=1:5 for i2=1:5 for i3=1:5 for i4=1:5 for i5=1:5  
for i6=1:5 for i7=1:5 for i8=1:5 for i9=1:5
```

```
    ruleList=[
```

```
        1 1 i1 1 1
```

```
        1 2 i2 1 1
```

```
        1 3 i3 1 1
```

```
        2 1 i4 1 1
```

```
        2 2 i5 1 1
```

```
        2 3 i6 1 1
```

```
        3 1 i7 1 1
```

```
        3 2 i8 1 1
```

```
        3 3 i9 1 1
```

```
    ];
```

```
    wah = addrule(wah,ruleList);
```

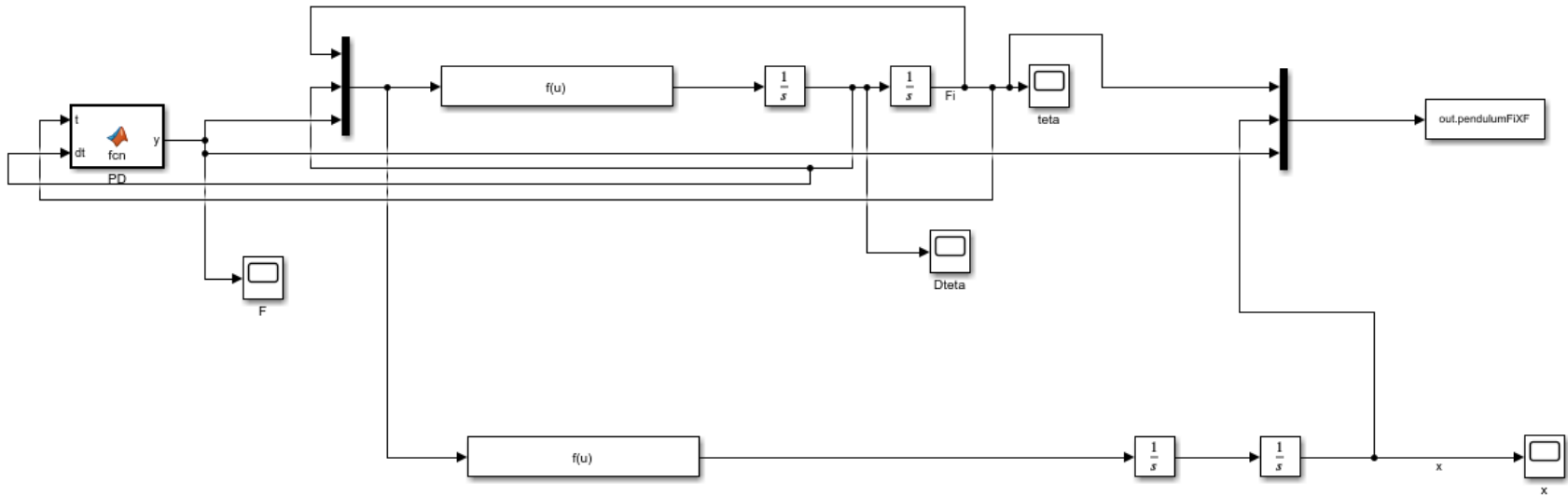
```
    sim('wah1');
```

```
    plot(x1.signals.values)
```

```
end; end; end; end; end; end; end; end;
```

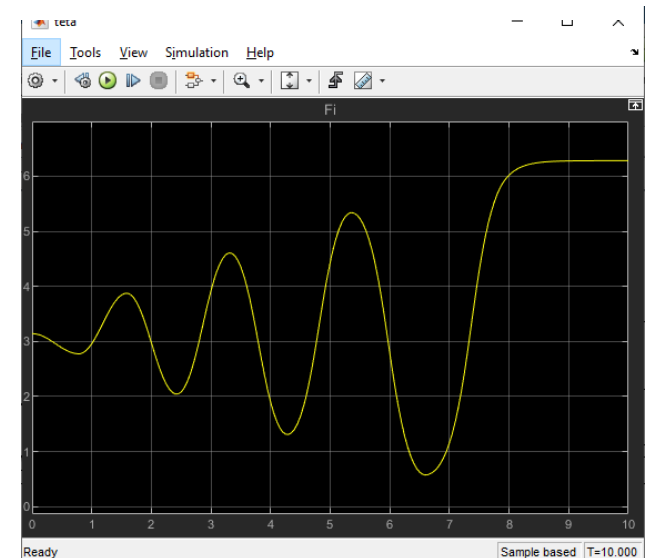
5⁹= 1953125 wariantów

InvPendulumPD

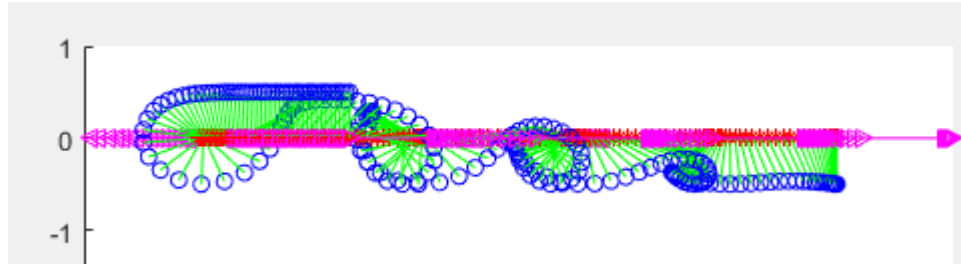


$$u = u(\theta, \dot{\theta}) = \begin{cases} K_1(0 - \theta) + K_2(0 - \dot{\theta}), & \text{if } 0 \leq |\theta| \leq \frac{\pi}{9} \\ K_1(2\pi - \theta) + K_2(0 - \dot{\theta}), & \text{if } \frac{17\pi}{9} \leq |\theta| \leq 2\pi \\ F_0 \operatorname{sgn}(\dot{\theta}), & \text{else} \end{cases}$$

\LR\lit\Neural_network_control_of_an_inverted_pendulum_on_.pdf

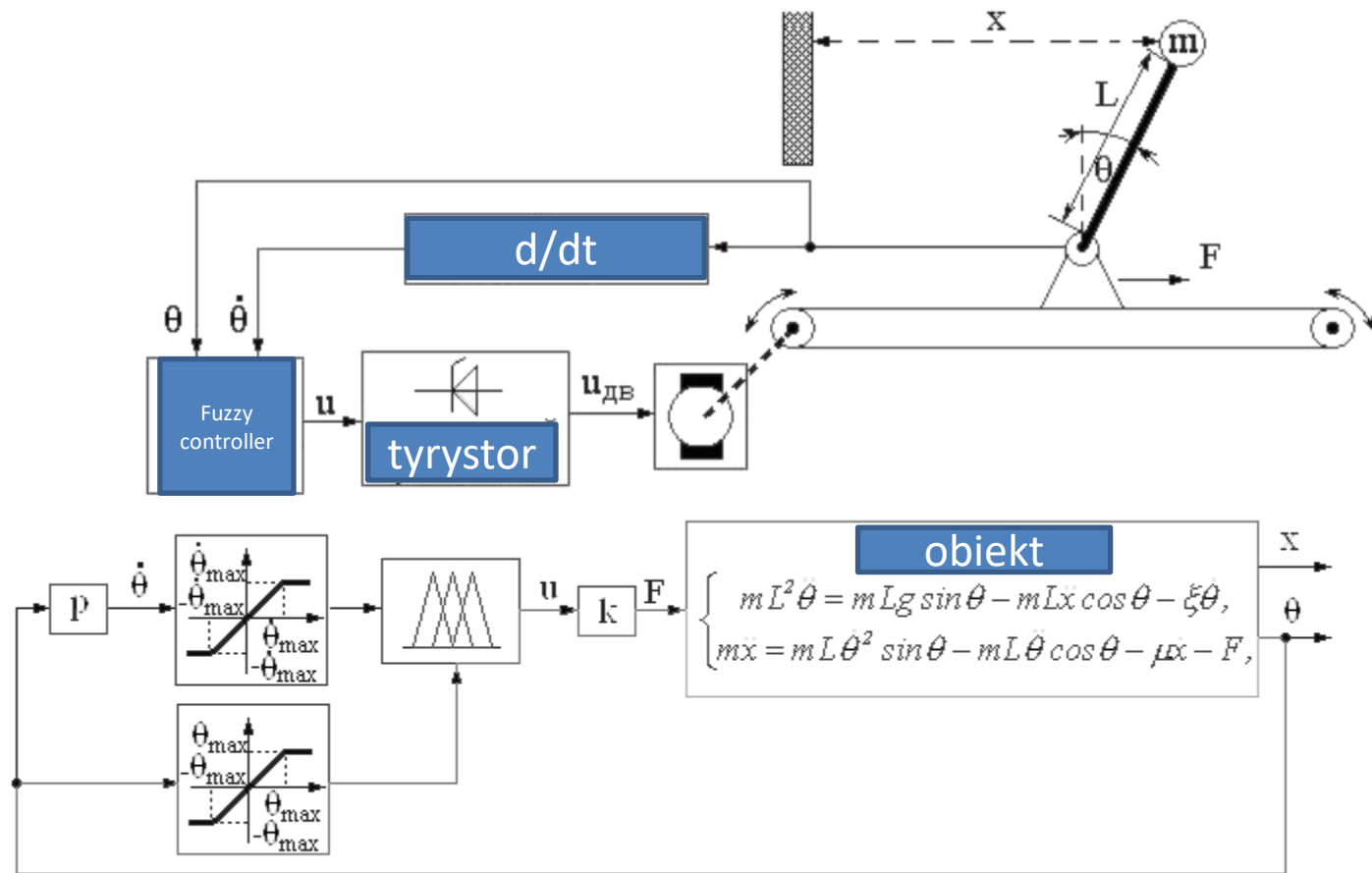


Animacja ruchu



```
animationPendulumOS.m  getpixelposition.m  Untitled6*  +
1  %animation
2  L=0.5;
3  D=out.pendulumFixF.Data;
4  clear M;
5  N=length(D);
6  for j=1:N
7      x=D(j,2);
8      y=0;
9      x1=x+L*sin(D(j,1));
10     y1=L*cos(D(j,1));
11     hold on;
12     if D(j,3)>0
13         plot(x,y,'x',x1,y1,'bo',[x x1],[y y1'],'-g',[x x+D(j,3)],[y y1'],'-m',x+D(j,3),y,'m>');
14     else
15         plot(x,y,'x',x1,y1,'bo',[x x1],[y y1'],'-g',[x x+D(j,3)],[y y1'],'-m',x+D(j,3),y,'m<');
16     end;
17     %axis([XMIN XMAX YMIN YMAX]);
18     %axis([min(D(:,2))-2*L max(D(:,2))+2*L -1 1]);
19     axis([min(D(:,2))-2*L max(D(:,2))+2*L min(D(:,2))-2*L max(D(:,2))+2*L]);
20
21     M(j) = getframe;
22
23
24
25
26 end;
```

Schemat urządzenia



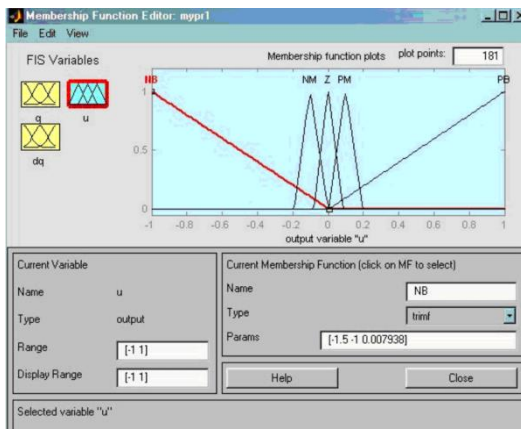
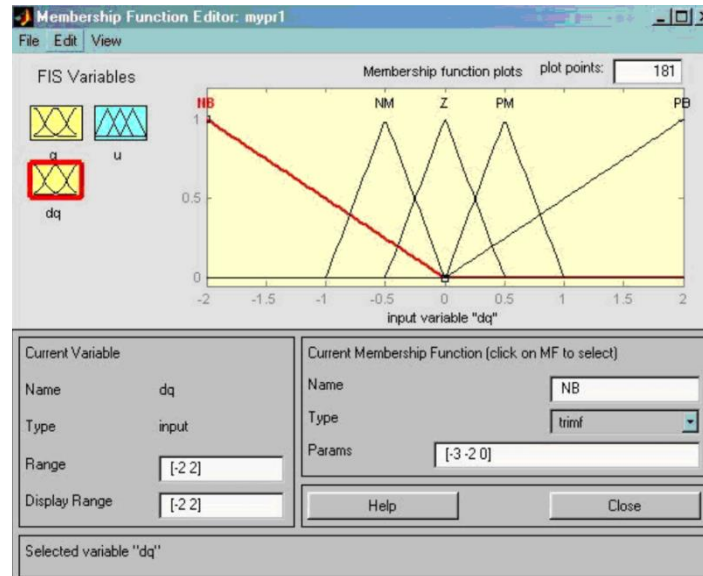
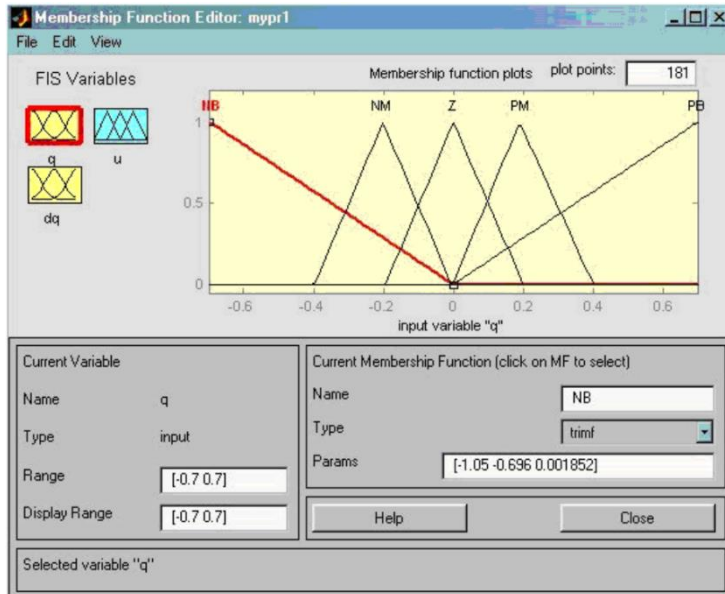
$m = 0,074 \text{ kg}, L = 0,5 \text{ m}, \xi = 0,002, \text{ kg} \cdot \text{m}^2 / \text{s}, \mu = 2 \text{ kg} / \text{s}, k = 5$

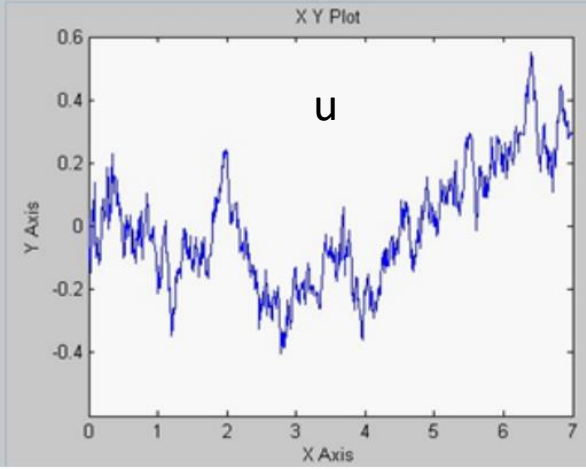
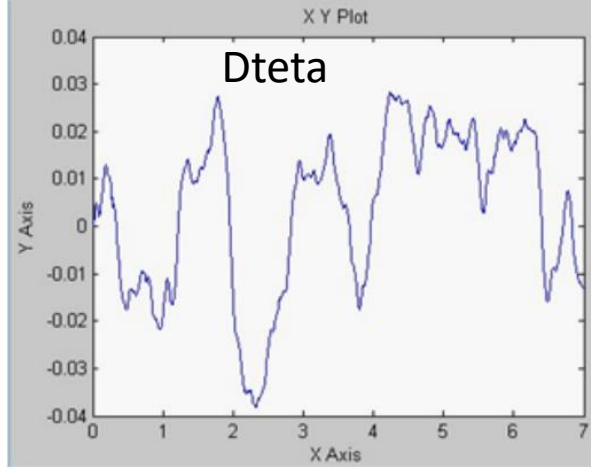
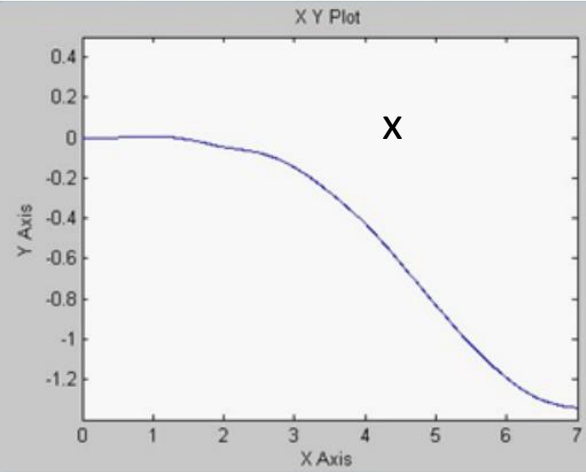
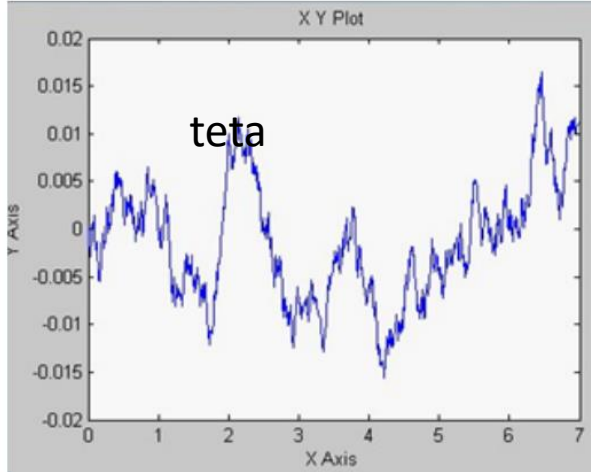
Model Mamdaniego

θ	NB	NM	Z	PM	PB
$\dot{\theta}$					
NB	-	-	NB	-	-
NM	-	-	NM	Z	-
Z	NB	NM	Z	PM	PB
PM	-	Z	PM	-	-
PB	-	-	PB	-	-

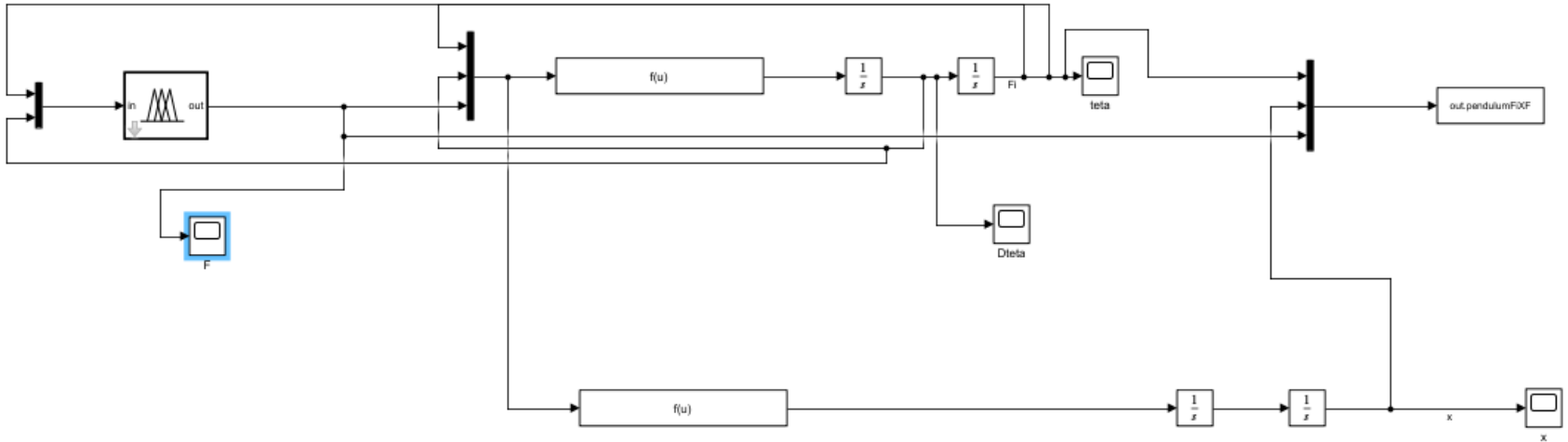
Rule Editor: mypr1
 File Edit View Options

1. If (q is NB) and (dq is Z) then (u is NB) (1)
2. If (q is NM) and (dq is Z) then (u is NM) (1)
3. If (q is Z) and (dq is Z) then (u is Z) (1)
4. If (q is PM) and (dq is Z) then (u is PM) (1)
5. If (q is PB) and (dq is Z) then (u is PB) (1)
6. If (q is Z) and (dq is NM) then (u is NM) (1)
7. If (q is Z) and (dq is NB) then (u is NB) (1)
8. If (q is Z) and (dq is PM) then (u is PM) (1)
9. If (q is Z) and (dq is PB) then (u is PB) (1)
10. If (q is NM) and (dq is PM) then (u is Z) (1)
11. If (q is PM) and (dq is NM) then (u is Z) (1)





InvPendulumMamdani



Fuzzy Logic Designer: wahadlo

File Edit View

The screenshot shows the Fuzzy Logic Designer interface for a Mamdani inference engine. The FIS Name is "wahadlo" and the FIS Type is "mamdani". The interface includes the following settings:

Parameter	Value
And method	min
Or method	max
Implication	min
Aggregation	max
Defuzzification	centroid

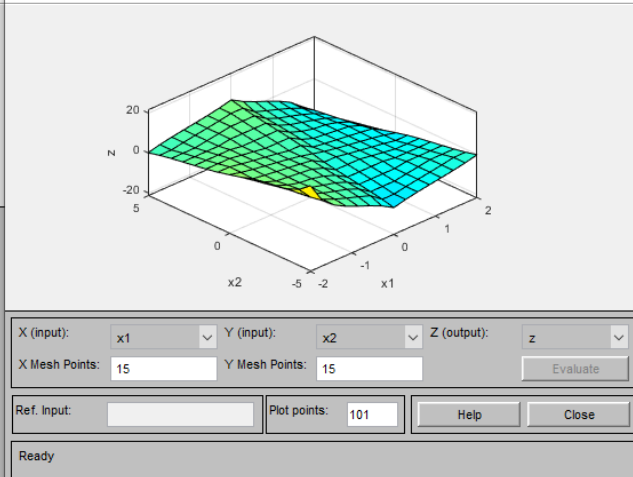
The Current Variable section shows:

Property	Value
Name	x1
Type	input
Range	[-2 2]

The interface also includes buttons for "Help" and "Close", and a "Ready" status indicator.

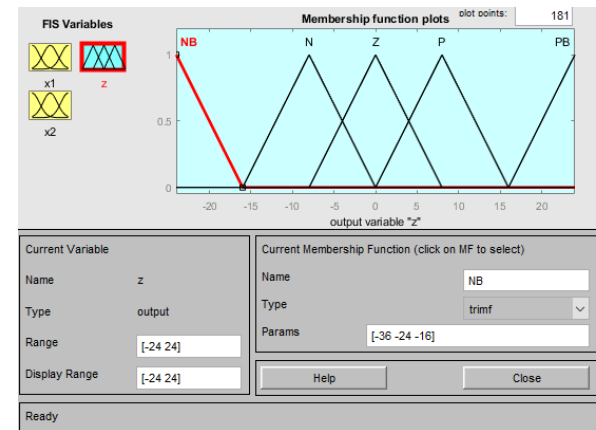
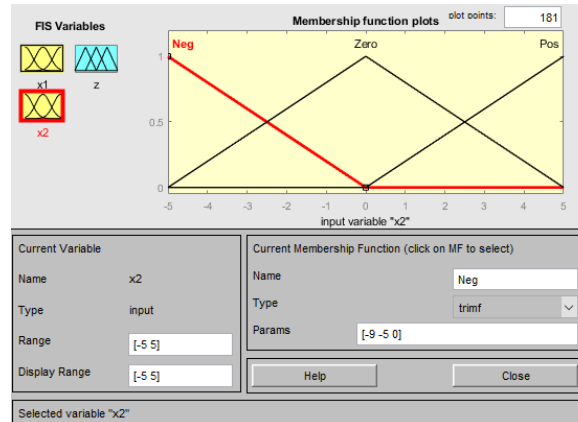
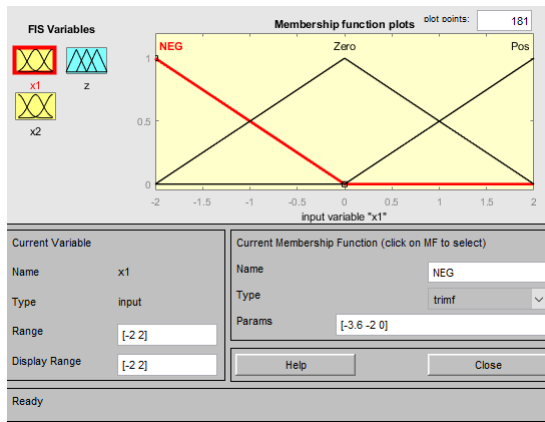
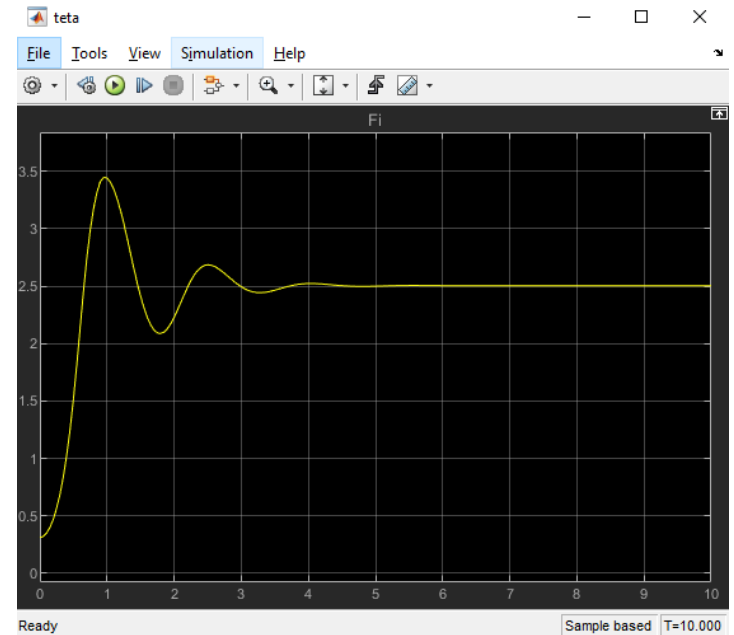
Surface Viewer: wahadlo

File Edit View Options



Wahadlo.fis

1. If (x1 is NEG) and (x2 is Neg) then (z is PB) (1)
2. If (x1 is NEG) and (x2 is Zero) then (z is P) (1)
3. If (x1 is NEG) and (x2 is Pos) then (z is Z) (1)
4. If (x1 is Zero) and (x2 is Neg) then (z is N) (1)
5. If (x1 is Zero) and (x2 is Zero) then (z is Z) (1)
6. If (x1 is Zero) and (x2 is Pos) then (z is P) (1)
7. If (x1 is Pos) and (x2 is Neg) then (z is Z) (1)
8. If (x1 is Pos) and (x2 is Zero) then (z is N) (1)
9. If (x1 is Pos) and (x2 is Pos) then (z is NB) (1)



Automatyczny pilot

Rozważmy teraz symulację procesu lądowania samolotu. Interesują nas dwie zmienne: wysokość samolotu h i jego prędkość v . Równania opisujące dynamikę procesu wyprowadza się stosując następujące rozumowanie. Ciało o masie m poruszające się z prędkością v posiada moment $p = mv$. Przy braku działania zewnętrznych sił, ciało to będzie poruszać się nadal ze stałą prędkością v . Dodatkowe oddziaływanie siłą f w przedziale czasu Δt skutkuje zmianą prędkości $\Delta v = f\Delta t/m$. Przyjmując $\Delta t = 1.0$ oraz $m = 1.0$ (i dobierając odpowiednio jednostki miary), otrzymujemy $\Delta v = f$, co oznacza, że zmiana prędkości jest proporcjonalna do użytej siły. Otrzymujemy więc następujący układ równań:

$$\begin{aligned}v(t + 1) &= v(t) + f(t) \\h(t + 1) &= h(t) + v(t)\end{aligned}$$

Zmienne lingwistyczne

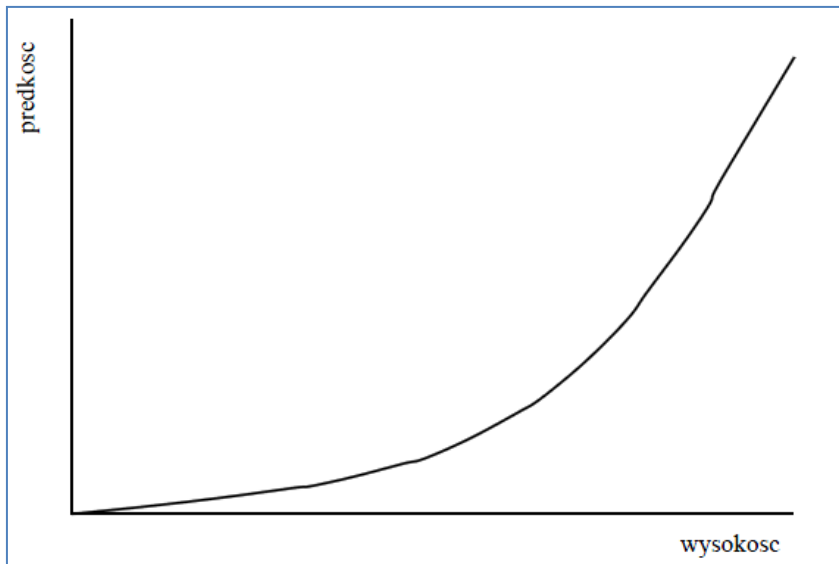
Przyjmijmy, że wysokość samolotu zmienia się w przedziale $[0, 1000]$. Pokrywamy dziedzinę zmiennej h czterema zbiorami rozmytymi: **Near Zero** o funkcji przynależności $\Delta(0, 0, 500)$, **Small** o funkcji przynależności $\Delta(-200, 300, 800) \cdot \mathbb{1}(x)$, gdzie $\mathbb{1}(x)$ jest funkcją przyjmującą wartość 0 dla $x \leq 0$ i 1 gdy $x > 0$, **Medium** o funkcji przynależności $\Delta(300, 800, 1300) \cdot (1 - \mathbb{1}(x - 1000))$ oraz **Large** o trójkątnej funkcji $\Delta(500, 1000, 1000)$.

Założmy dalej, że prędkość samolotu zmienia się w przedziale $[-30, 30]$ pokrytym pięcioma zbiorami rozmytymi. Podobnie zmienia się **wartości siły f** (sterowania).

etykieta	definicja
Down Large (DL)	$\Delta(-\infty, -20, -10)$
Down Small (DS)	$\Delta(-20, -10, 0)$
Zero (Z)	$\Delta(-10, 0, 10)$
Up Small (US)	$\Delta(0, 10, 20)$
Up Large (UL)	$\Delta(10, 20, \infty)$

Cel sterowania

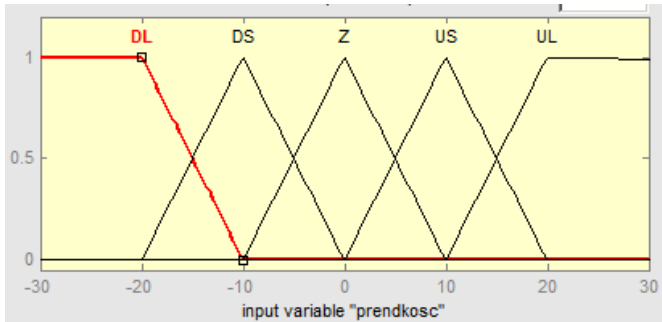
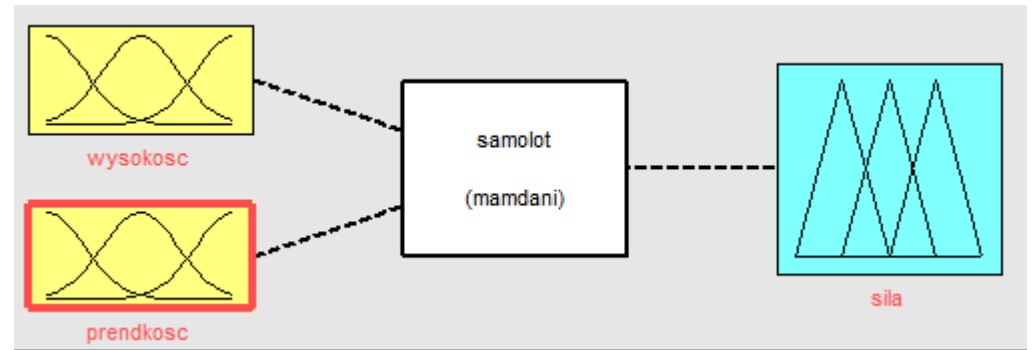
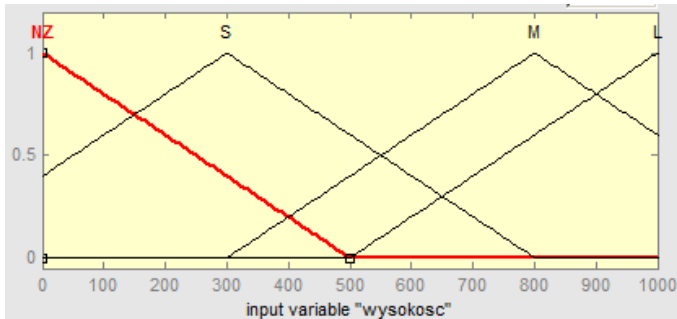
Należy tak sterować samolotem, aby jego trajektoria ruchu wyglądała jak na poniższym rysunku.



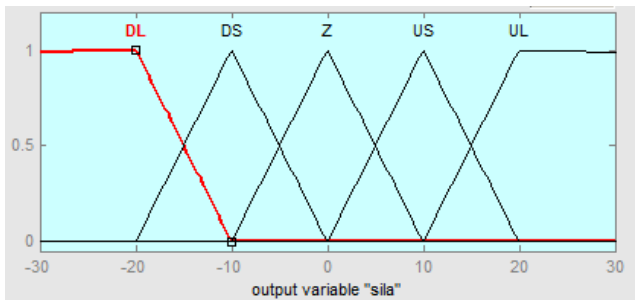
Baza reguł sterownika

	prędkość v				
wysokość h	DL	DS	Zero	US	UL
L	Z	DS	DL	DL	DL
M	US	Z	DS	DL	DL
S	UL	US	Z	DS	DL
NZ	UL	UL	Z	DS	DS

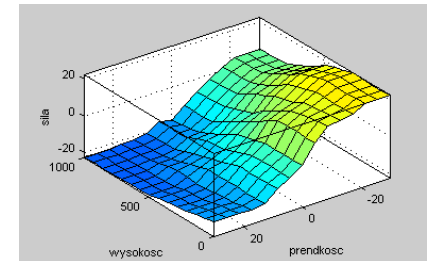
Sterownik rozmyty Mamdaniego



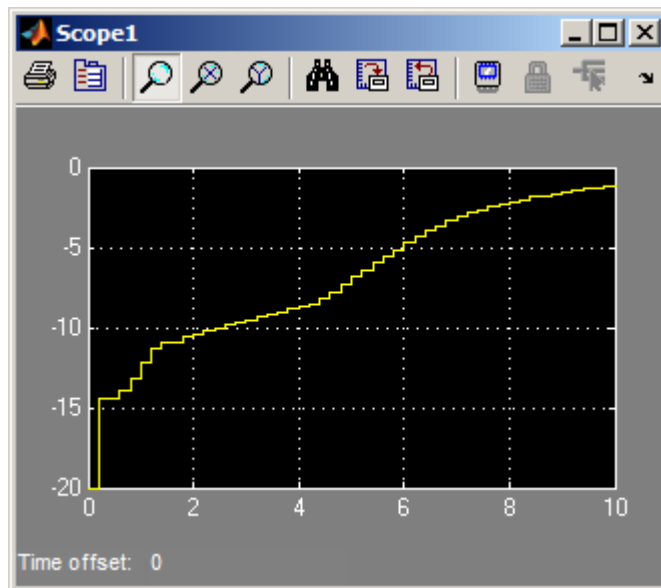
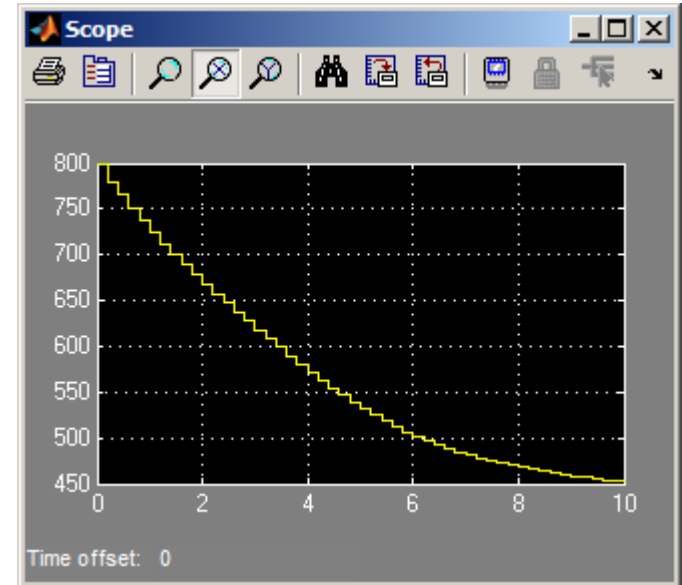
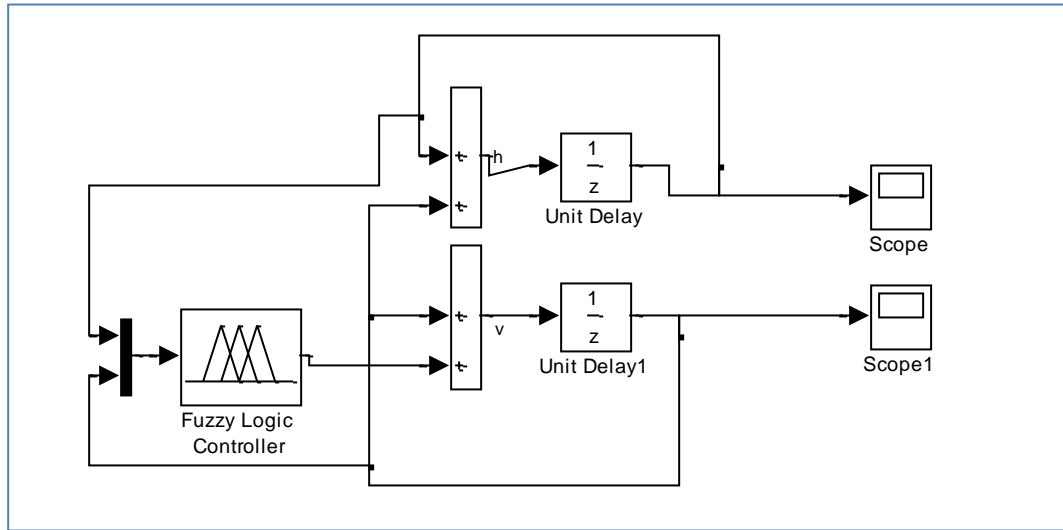
1. If (wysokosc is L) and (prendkosc is DL) then (sila is Z) (1)
2. If (wysokosc is L) and (prendkosc is DS) then (sila is DS) (1)
3. If (wysokosc is L) and (prendkosc is Z) then (sila is DL) (1)
4. If (wysokosc is L) and (prendkosc is US) then (sila is DL) (1)
5. If (wysokosc is L) and (prendkosc is UL) then (sila is DL) (1)
6. If (wysokosc is M) and (prendkosc is DL) then (sila is US) (1)
7. If (wysokosc is M) and (prendkosc is DS) then (sila is Z) (1)
8. If (wysokosc is M) and (prendkosc is Z) then (sila is DS) (1)
9. If (wysokosc is M) and (prendkosc is US) then (sila is DL) (1)
10. If (wysokosc is M) and (prendkosc is UL) then (sila is DL) (1)



11. If (wysokosc is S) and (prendkosc is DL) then (sila is UL) (1)
12. If (wysokosc is S) and (prendkosc is DS) then (sila is US) (1)
13. If (wysokosc is S) and (prendkosc is Z) then (sila is Z) (1)
14. If (wysokosc is S) and (prendkosc is US) then (sila is DS) (1)
15. If (wysokosc is S) and (prendkosc is UL) then (sila is DL) (1)
16. If (wysokosc is NZ) and (prendkosc is DL) then (sila is UL) (1)
17. If (wysokosc is NZ) and (prendkosc is DS) then (sila is UL) (1)
18. If (wysokosc is NZ) and (prendkosc is Z) then (sila is Z) (1)
19. If (wysokosc is NZ) and (prendkosc is US) then (sila is DS) (1)
20. If (wysokosc is NZ) and (prendkosc is UL) then (sila is DS) (1)



Realizacja w Matlabie

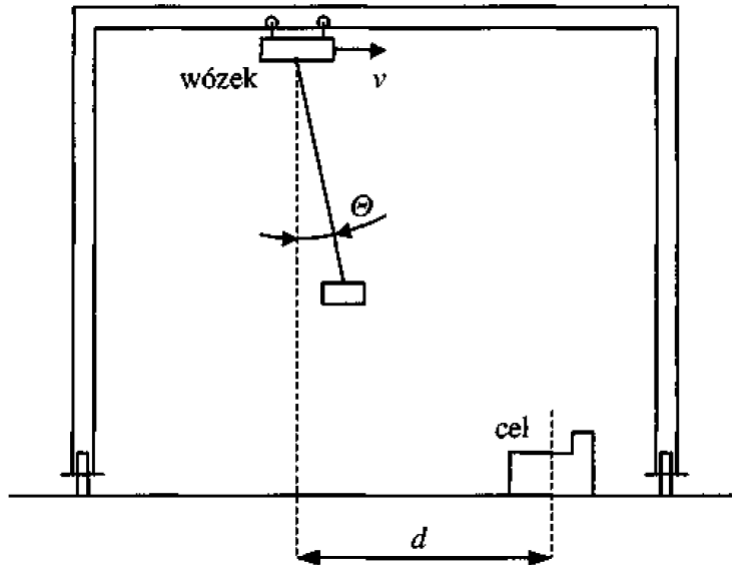


Sam.mdl

Samolot.fis

Sterowanie wózkiem

Sterowanie wózkiem suwnicy przenoszącej kontenery z jednego miejsca w inne

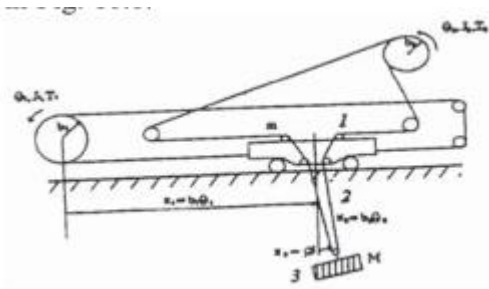


Podczas transportu kontenerów dochodzi często do dużego ich kołysania (duże odchylenie θ od pionu). Przy ustawianiu kontenera w punkcie docelowym, np. na statku, kołysania są niedopuszczalne. Na skutek uderzenia kołyszącego się kontenera w inne, już ustawione, może nastąpić ich uszkodzenie.

Operator suwnicy steruje prędkością v jej wózka przy pomocy dźwigni mającej dwa krańcowe położenia. Potrzebne jest tu **wycucie reakcji wózka**, jego bezwładności i transportowanego kontenera (**zmienny ciężar**) na zmiany położenia dźwigni. Sterowanie wózkiem suwnicy mogłoby się odbywać dwoma bardzo prostymi metodami **nie wymagającymi żadnej wiedzy eksperckiej**.

In an unmanned crane operational control system, the main problem is to transport the container to the target location without sway. To solve the problem several approaches have been investigated such as control methods with nonlinear feedback and the numerical methods for constrained nonlinear optimisation. However, all of them result in some performance decrease because of the sway rejection. Fuzzy control is supposed to be robust enough and can be applied here.

The mechanical model for no-sway crane operation is shown in Fig. 10.8.



$$\dot{x} = f(x, v, t), x = (x_1, \dots, x_6)^T, u = (v_1, v_2)^T$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = v_1 - \delta_1 x_3 v_2 + \delta_1 g x_3$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -(v_1 - \delta_1 x_3 v_2 + (1 + \delta_1) g x_3 + 2x_4 x_6) / x_5$$

$$\dot{x}_5 = x_6$$

$$\dot{x}_6 = -\delta_2 x_3 v_2 + v_2$$

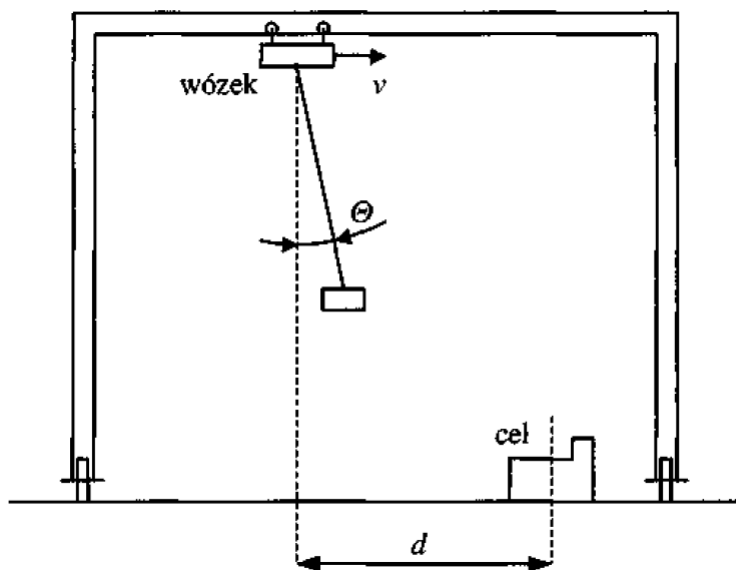
$$x_1 = b_1 \theta_1, x_2 = \dot{x}_1, x_3 = \psi, x_4 = \dot{x}_3, x_5 = b_2 \theta_2, x_6 = \dot{x}_5$$

$$v_1 = \frac{b_1 T_1}{J_1 + m b_1^2}, v_2 = \frac{b_2 (T_2 + M b_2 g)}{J_2 + M b_2^2}, \delta_1 = \frac{M b_1^2}{J_1 + m b_1^2}, \delta_2 = \frac{M b_2^2}{J_2 + M b_2^2}$$

where J_1 and J_2 are the momentum inertia for trolley and hoist motor, M and m are the masses of the load including the spreader and the trolley including the operation room. T_1 and T_2 are the torques, b_1 and b_2 are the drum radii, θ_1 and θ_2 are the rotation angles for the trolley and the hoist, ψ is the sway angle and g is the gravity acceleration constant.

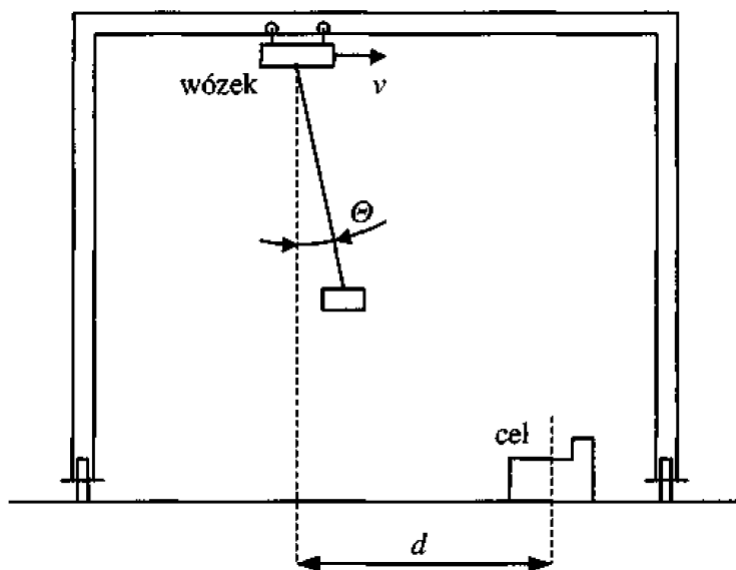
Design a fuzzy controller to keep the sway angle as small as possible. The sway angle is supposed to be measured.

Sterowanie wózkiem (cd)



Pierwsza metoda polegałaby na sterowaniu wózkiem z tak małą prędkością v , że nie wystąpiłyby żadne kołysania kontenera. W metodzie **drugiej** kontener byłby przewożony z dużą prędkością, co wiązałoby się oczywiście z wystąpieniem kołysań. Nad celem jednak operator czekałby aż kołysania ustaną i dopiero wtedy opuszczałby kontener na właściwe miejsce przeznaczenia. Obydwie te metody wiązałyby się jednak ze **zbyt długim czasem transportu**.

Sterowanie wózkiem (cd)

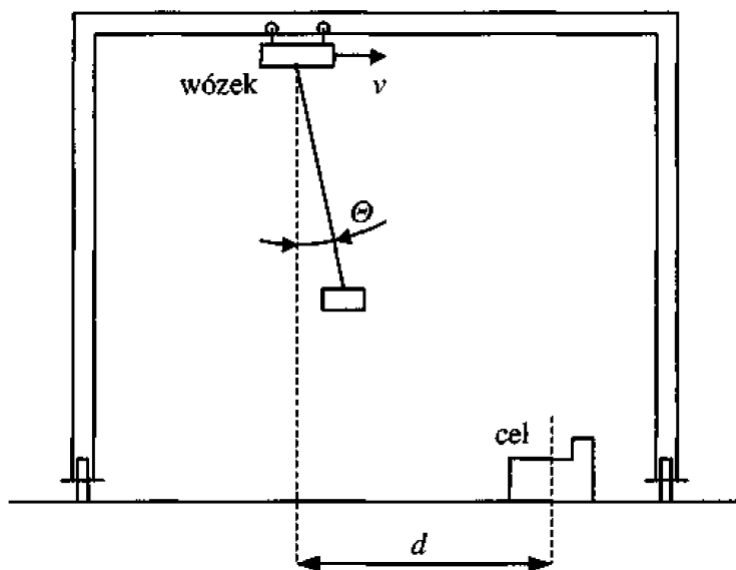


Doświadczony operator suwnicy potrafi zrealizować transport szybko i wytłumić kołysania nad miejscem ustawienia. Sterowanie suwnicą **nie może być zrealizowane przez konwencjonalny, liniowy regulator PID**, bowiem proces ten ma silnie nieliniowy charakter.

Przy dużej odległości d wózka od celu kołysania kontenera nie są istotne i wózek może poruszać się z dużą prędkością.

Przy zmniejszaniu się tej odległości operator musi jednak coraz bardziej zmniejszać amplitudę kołysań (wielkość odchylenia θ od pionu), po to, aby bezpośrednio nad celem kontener nie wykonywał żadnego ruchu i można go było szybko i bezpiecznie ustawić. Na kołysanie kontenerów negatywnie wpływają zakłócenia, np. wiatr wiejący z różnych kierunków i ze zróżnicowaną siłą.

Sterowanie wózkiem (cd)



Sterowanie suwnicą jest znacznie utrudnione istnieniem **tarcia spoczynkowego**. Aby wywołać ruch suwnicy należy doprowadzić do silnika taką moc elektryczną, która wytworzy w silniku moment większy od dużych oporów tarcia spoczynkowego. Z chwilą jednak, gdy wózek suwnicy zacznie się poruszać opory tarcia znacznie się zmniejszają i doprowadzona moc może okazać się zbyt dużą. Dotyczy to zwłaszcza przypadku, gdy transportowany kontener znajduje się nad celem i potrzebne są jedynie małe ruchy korekcyjne w prawo i w lewo. Niedoświadczony operator może mieć wtedy duże kłopoty i precyzyjnym ustawieniem kontenera, a jego sterowanie może zawierać dużą ilość przełączeń silnika i niepotrzebnych szarpnięć.

Sterowanie wózkiem (cd)

Mimo, że sterowanie transportem kontenerów nie jest łatwe, ludzie potrafią je bardzo dobrze opanować. Na podstawie wywiadu i obserwacji reakcji operatora pewnej konkretnej suwnicy zidentyfikowano następujące podstawowe reguły sterowania (d - odległość, θ - kąt odchylenia P – moc elektryczna doprowadzona do silnika napędowego przez odpowiednie ustawienie dźwigni sterującej):

R1: JEŚLI ($d = \text{duża}$) TO ($P = \text{duża}$),

R2: JEŚLI ($d = \text{mała}$) I ($\theta = \text{ujemny duży}$) TO ($P = \text{ujemna średnia}$),

*R3: JEŚLI ($d = \text{mała}$) I ($\theta = \text{ujemny mały LUB zero LUB dodatni mały}$)
TO ($P = \text{dodatnia średnia}$),*

R4: JEŚLI ($d = \text{mała}$) I ($\theta = \text{dodatni duży}$) TO ($P = \text{dodatnia duża}$),

R5: JEŚLI ($d = \text{zero}$) I ($\theta = \text{dodatni duży LUB mały}$) TO ($P = \text{ujemna średnia}$),

R6: JEŚLI ($d = \text{zero}$) I ($\theta = \text{zero}$) TO ($P = \text{zero}$),

R7: JEŚLI ($d = \text{zero}$) I ($\theta = \text{ujemny mały}$) TO ($P = \text{dodatnia średnia}$),

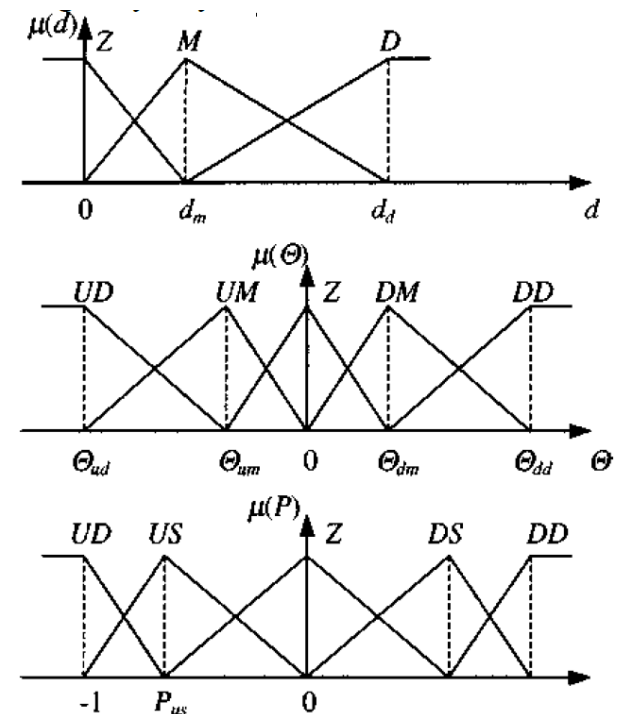
R8: JEŚLI ($d = \text{zero}$) I ($\theta = \text{ujemny duży}$) TO ($P = \text{dodatnia duża}$).

Sterowanie wózkiem (cd)

Należy zauważyć, że inny operator suwnicy może posługiwać się innymi regułami w tych samych sytuacjach. Zależy to od refleksu, temperamentu i doświadczenia operatora. Reguły sterowania mogą zależeć od rodzaju suwnicy, silnika napędowego, wielkości suwnicy, maksymalnej odległości transportu, wielkości tarcia w mechanizmach suwnicy. Z tego względu podane powyżej reguły mają jedynie charakter przykładowy i nie są ogólnie obowiązujące.

Do oceny poszczególnych wielkości operator użył następujących wartości lingwistycznych:

- d - duża (D), mała (M), zero (Z),
- Θ - dodatni duży (DD), dodatni mały (DM), zero (Z), ujemny mały (UM), ujemny duży (UD),
- P - ujemna duża (UD), ujemna średnia (US), zero (Z), dodatnia średnia (DS), dodatnia duża (DD).



Sterowanie wózkiem (cd)

W praktyce synteza regulatorów rozmytych na bazie wiedzy eksperta możliwa jest jedynie w przypadku prostszych obiektów, **maksymalnie z dwoma wejściami i raczej wolnozmiennych**. Człowiek nie jest bowiem w stanie opanować sterowania obiektów o większej ilości wejść (ograniczone możliwości percepcji i przetwarzania informacji) oraz obiektów szybkozmiennych (ograniczenie szybkości przetwarzania informacji). Jednak nawet w takich przypadkach człowiek może dostarczyć częściowej informacji o sterowaniu, którą można potem uzupełnić ucząc model próbkami pomiarowymi procesu sterowania. Wstępna wiedza o sterowaniu może być również źródłem pomysłów umożliwiających konstrukcję właściwej bazy reguł, doboru funkcji przynależności, itp.

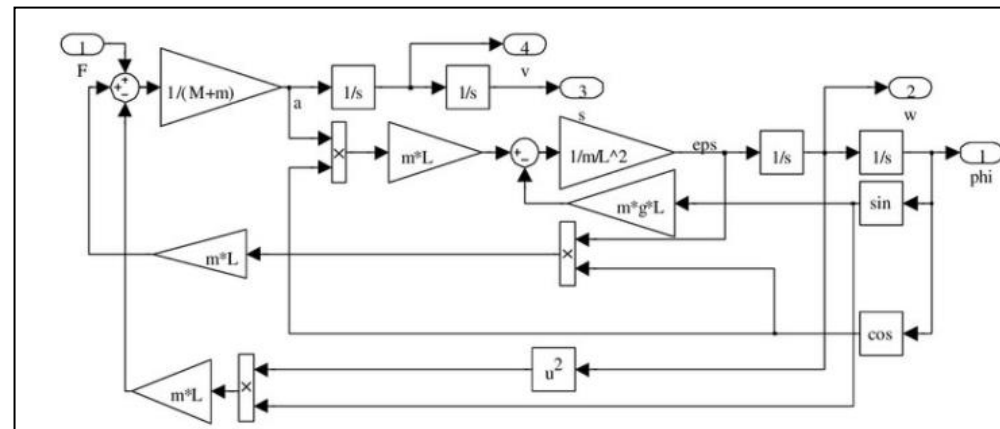
Model wózka

$$(M + m) \frac{d^2 s(t)}{dt^2} - mL \frac{d^2 \varphi(t)}{dt^2} \cos \varphi + mL \frac{d\varphi^2(t)}{dt} \sin \varphi(t) = F(t),$$

$$- mL \frac{d^2 s(t)}{dt^2} \cos \varphi(t) + mL^2 \frac{d^2 \varphi(t)}{dt^2} + mgL \sin \varphi(t) = 0,$$

s – odległość wózka od celu

M- masa wózka, m – masa ciężaru



Linearyzacja modelu

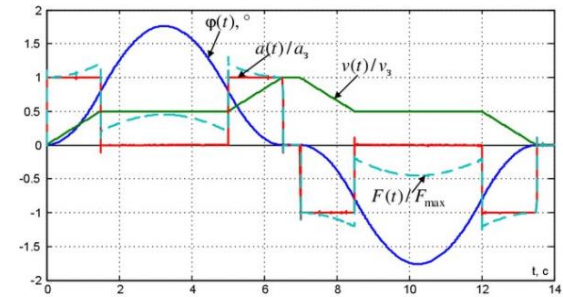
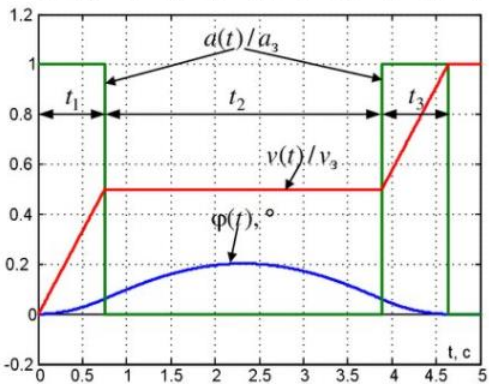
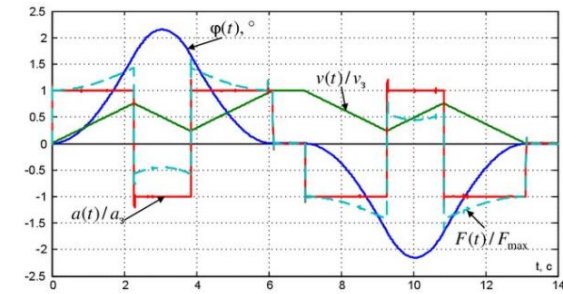
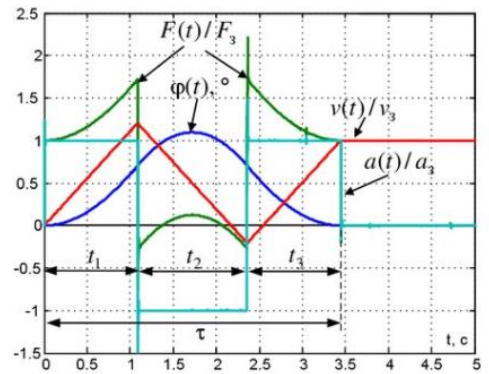
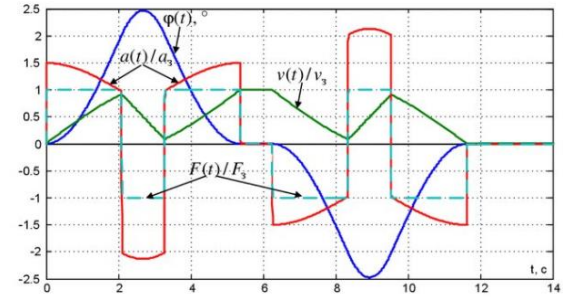
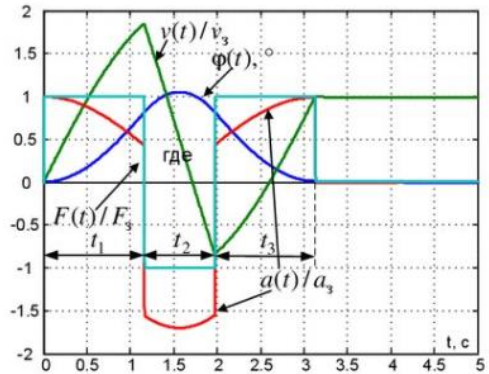
$$\frac{d^2 s(t)}{dt^2} = a(t), \quad \frac{ds(t)}{dt} = v(t), \quad \frac{d^2 \varphi(t)}{dt^2} = \varepsilon(t), \quad \frac{d\varphi(t)}{dt} = \omega(t)$$

$$\varphi \approx 0, \quad \cos \varphi \approx 1, \quad \sin \varphi \approx \varphi, \quad \omega^2 \approx 0$$

$$(M + m)a(t) - mL\varepsilon(t) = F(t),$$

$$- mL a(t) + mL^2 \varepsilon(t) + mLg \varphi(t) = 0.$$

Możliwe rozwiązania

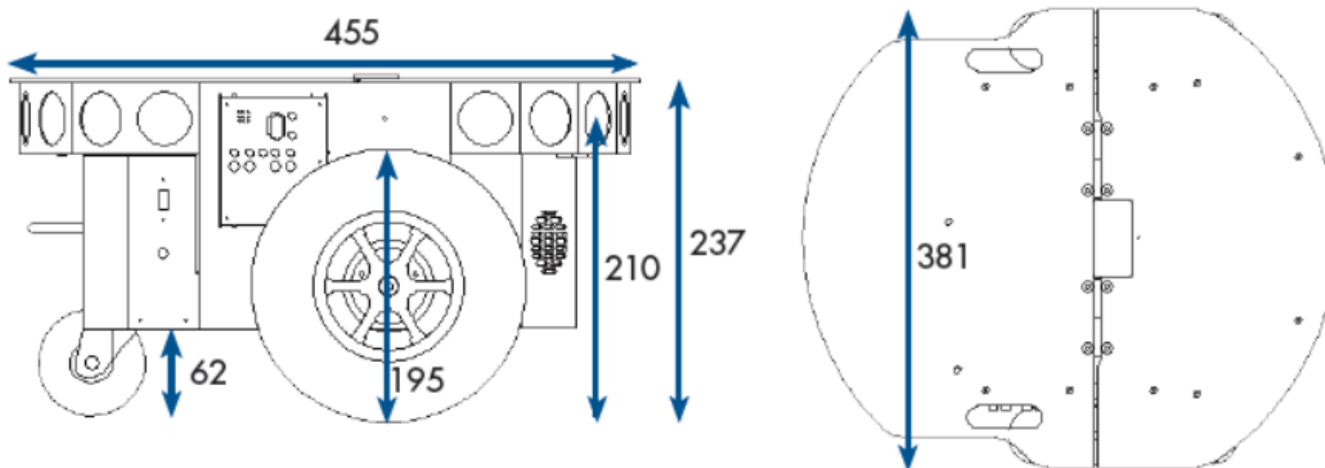


Sterowanie robotem

<http://www.mobilerobots.com/researchrobots/pioneer3dx.aspx>

Pioneer 3-DX

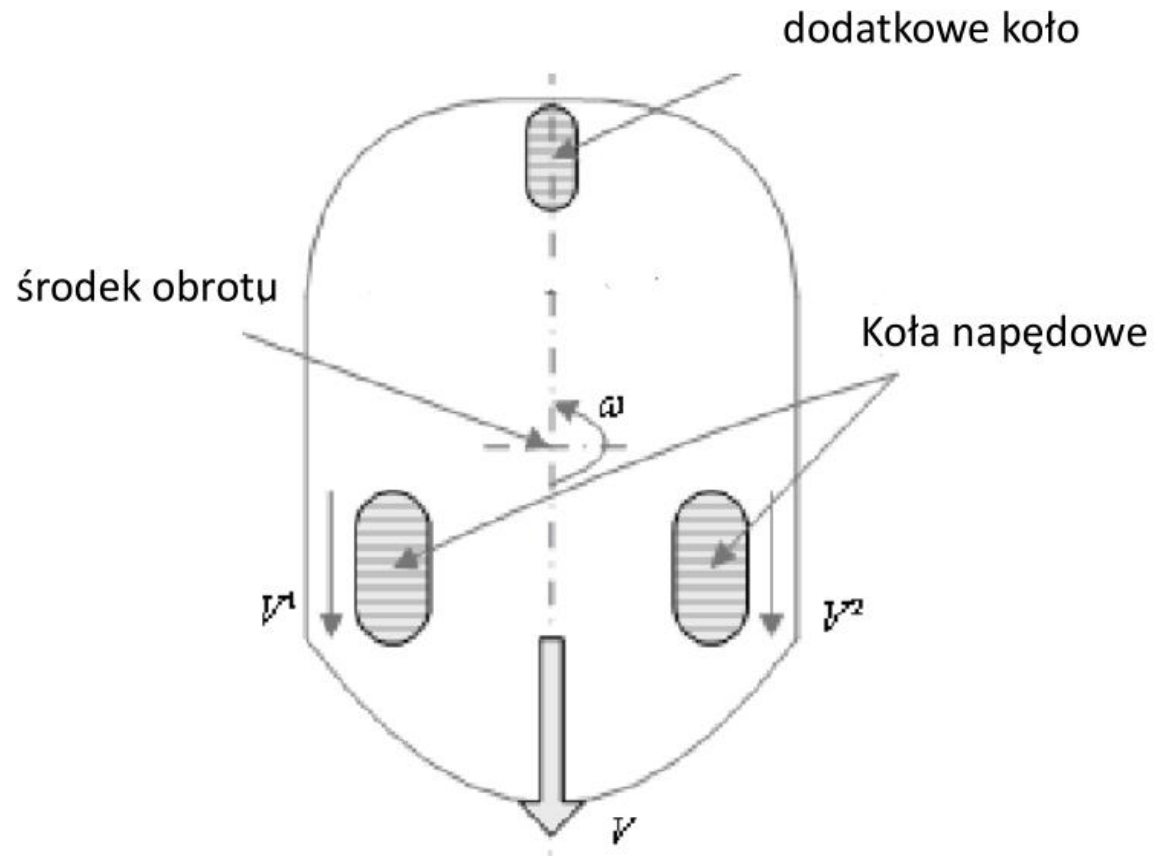
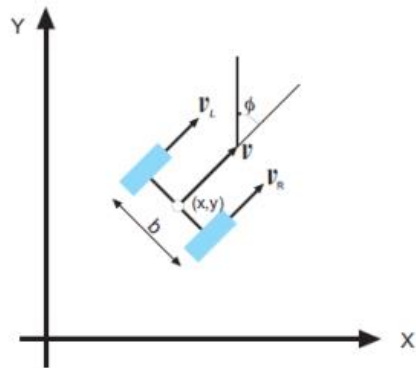
Dimensions (mm)



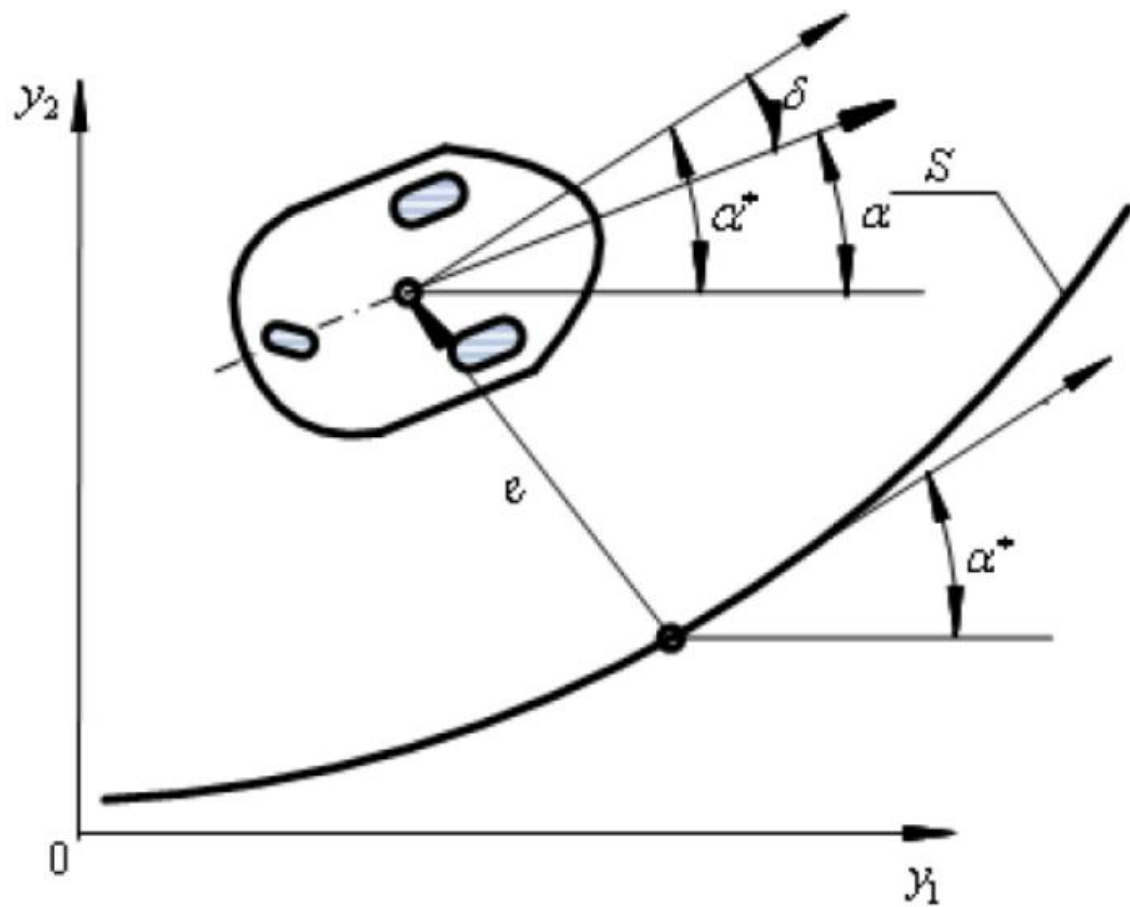
P3DX jest kołowy Robot mobilny (wheeled mobile robot - WMR), z dwoma **niezależnymi** kołami napędowymi, plus dodatkowe koła dla stabilności



Model kinematyczny



e - odchylenia od normalnej do krzywej S
 δ - błąd orientacji kątowej



Model Mamdani

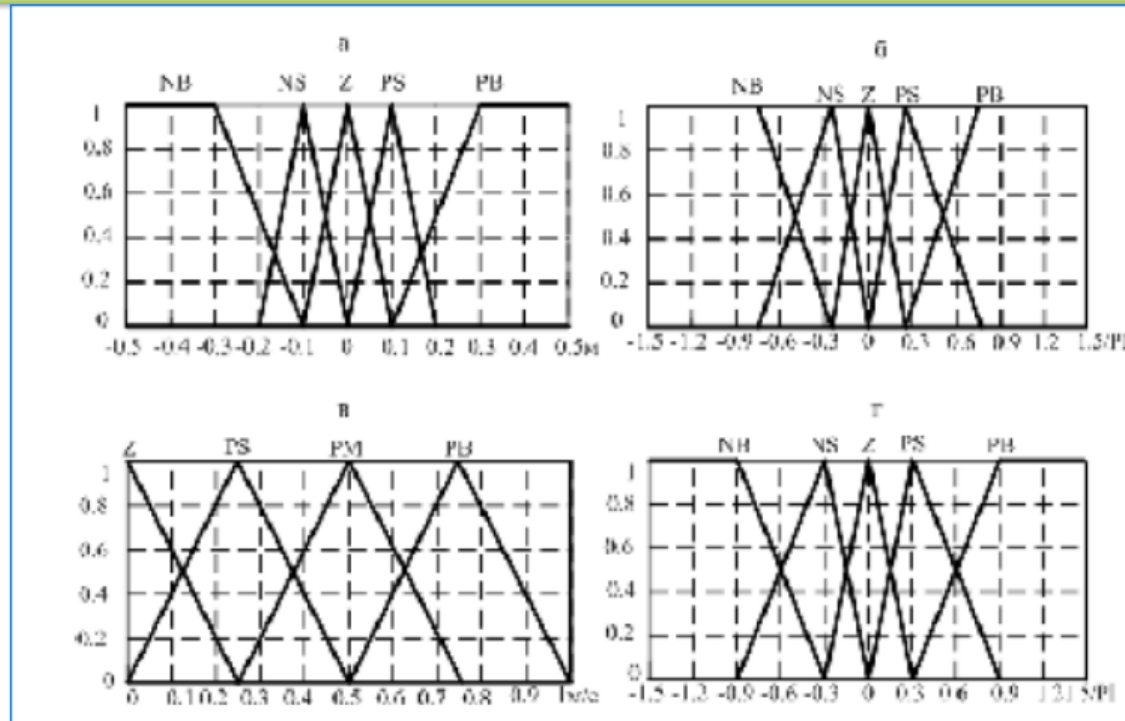
Y1 - odchylenia od normalnej do krzywej S

Y2 - błąd orientacji kątowej

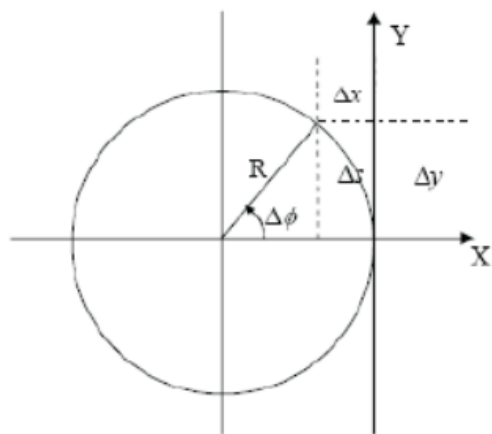
U1 – prędkość potrzebna

U2 - prędkość kątowa potrzebna

Jeśli " Y1 jest PB" i " Y2 jest NB" TO " U1 jest PS" i " U2 jest NM"



Ruch po okręgu



$$v = \frac{v_R + v_L}{2} = \frac{\omega_R + \omega_L}{2} r$$

$$\omega = \frac{v_R - v_L}{b} = \frac{\omega_R - \omega_L}{b} r$$


$$p' = \begin{bmatrix} -\sin \phi \\ \cos \phi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} -\sin \phi & 0 \\ \cos \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

\LR\lit\control\Robots\sys1-53.pdf

Model sterowania

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} -r(\sin \phi)/2 & -r(\sin \phi)/2 \\ r(\cos \phi)/2 & r(\cos \phi)/2 \\ -r/b & r/b \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} -(\sin \phi)/2 & -(\sin \phi)/2 \\ (\cos \phi)/2 & (\cos \phi)/2 \\ -1/b & 1/b \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix}$$


$$e = \sqrt{(x - x_c)^2 + (y - y_c)^2} - R$$

$$e' = \frac{\Delta e}{\Delta t}$$

Assume that the circular target trajectory is defined by the coordinates of the center (x_c, y_c) and the radius R .

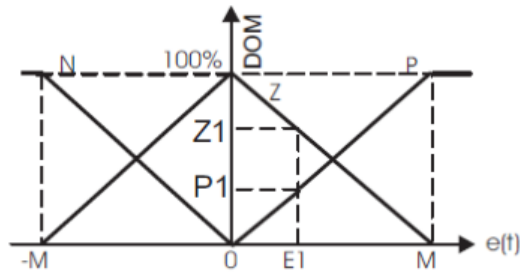
The WMR must be controlled so that the distance d from the current position to the center of the circle equals R .

error dot $e'(t)$	error $e(t)$		
	N	Z	P
N	HL	LH	LM
Z	HL	MM	MH
P	ML	MH	LH

Each cell of table 1 contains a logic sentence and should be read as: "If $e(t)$ is Negative AND $e'(t)$ is negative, THEN v_L must be HIGH and v_R must be LOW"

where H,M,L designate the singleton values for HIGH, MEDIUM and LOW fuzzy domains of the outputs v_L, v_R respectively.

The crisp output of the fuzzy controller is a combination of all the rules in the rule base as follows:



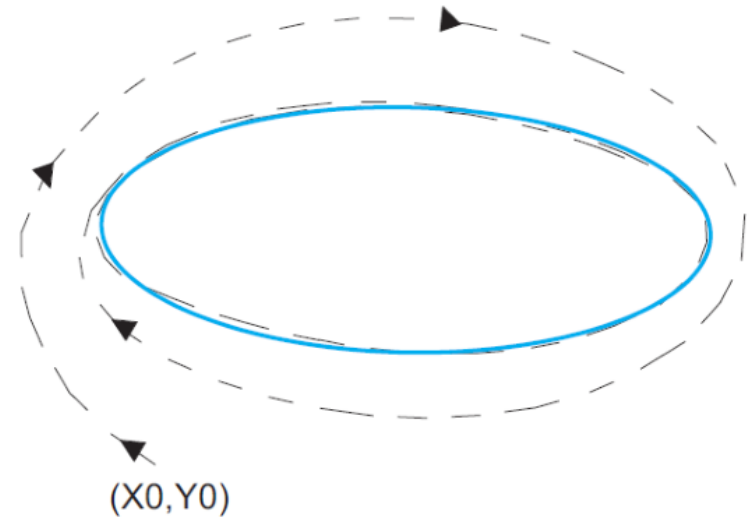
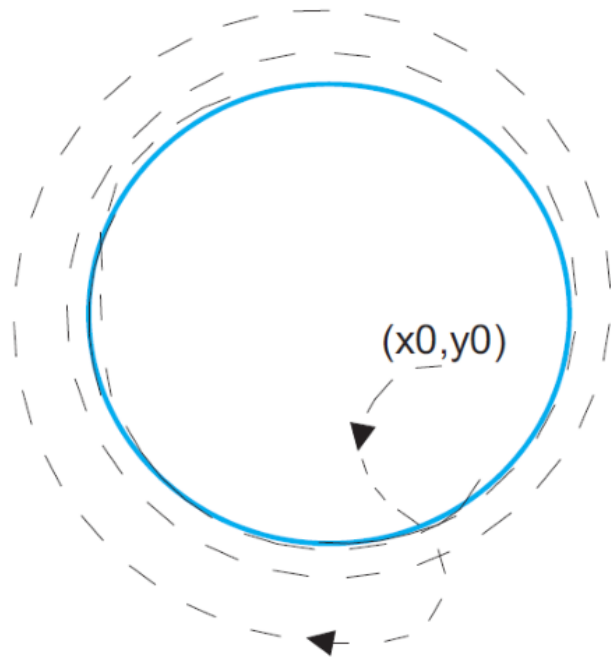
$$v_{out} = \frac{\sum_{i=1}^K z_i S_i}{\sum_{i=1}^K z_i} \quad (10)$$

where:

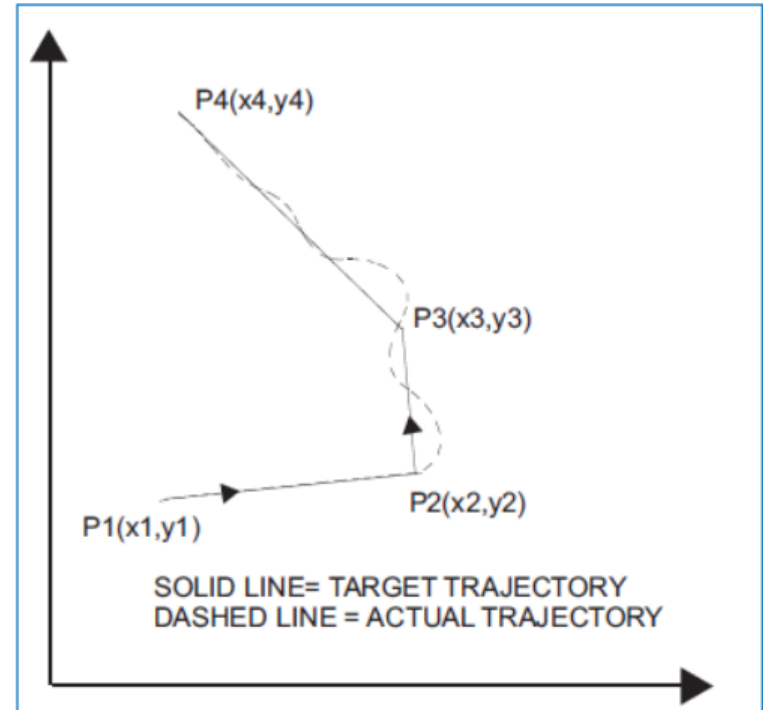
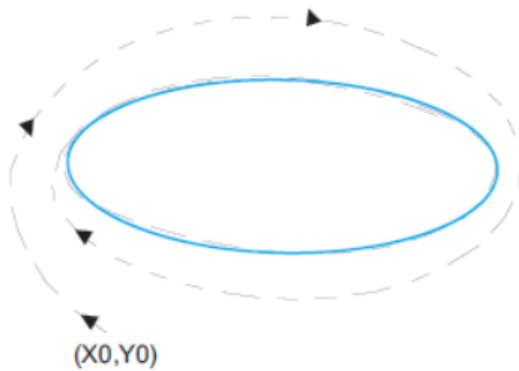
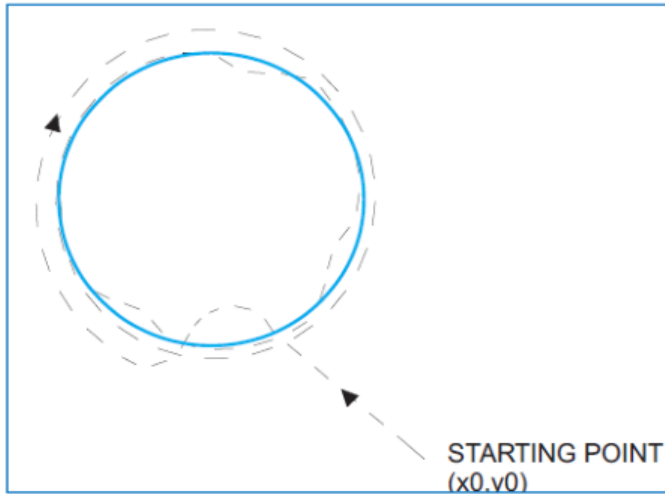
$$z_i = \min(E_i, E'_i) \quad (11)$$

S_i is the corresponding singleton value of the fuzzy output, and K is the total number of rules in the rule base. E_i, E'_i are the degrees of membership of $e(t)$ and $e'(t)$ to the domain corresponding to the cell i .

Testy jazdy



Zadania



$$Ax + By + C = 0$$

where

$$\begin{cases} A = y_{(i+1)} - y_i \\ B = (-x_{(i+1)} - x_i) \\ C = x_{(i+1)}y_i - x_iy_{(i+1)} \end{cases}$$

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

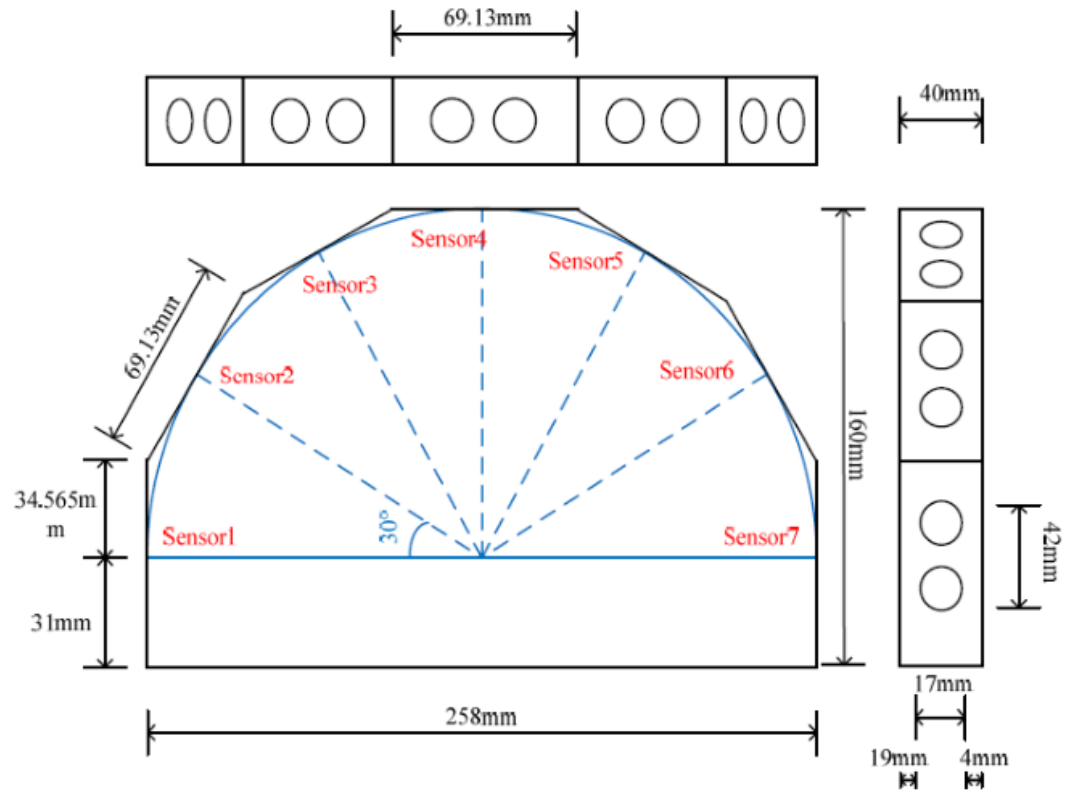
Unikanie przeszkód

Autonomiczne algorytmy nawigacyjne są stosowane do różnych zastosowań mobilnych robotów do jazdy w kierunku ich pozycji docelowej bez programu zdalnego sterowania. W etapie mapowania autonomicznego oprogramowania nawigacyjnego, **robot uzyskuje informacje o lokalizacji i konturu przeszkód z czujników i aktualizuje swoją mapę nawigacyjną**. Ścieżkę nawigacji z aktualnej pozycji do pozycji docelowej na mapie określa z aktualizowanych danych mapowych. Jeśli nie ma żadnych zmian w mapie podczas procesu nawigacji robota, unikanie przeszkody nie jest konieczne, ponieważ dobrze zdefiniowaną ścieżkę nawigacji doprowadzi robota do pozycji docelowej bezpiecznie. Nawet w przypadku, gdy niespodziewane przeszkody nagle pojawiają się na ścieżce nawigacji, robot jest wymagany do wykonywania unikania przeszkód, aby poruszać się po planowanej ścieżce bez kolizji z tymi rodzajami przeszkód. Aby uniknąć nieoczekiwanych przeszkód i dalej w kierunku pozycji docelowej w ramach nawigacji w czasie rzeczywistym, robot powinien określić jego kierunek skrętu i prędkości natychmiast.

`\LR\lit\control\Robots\InTech-`

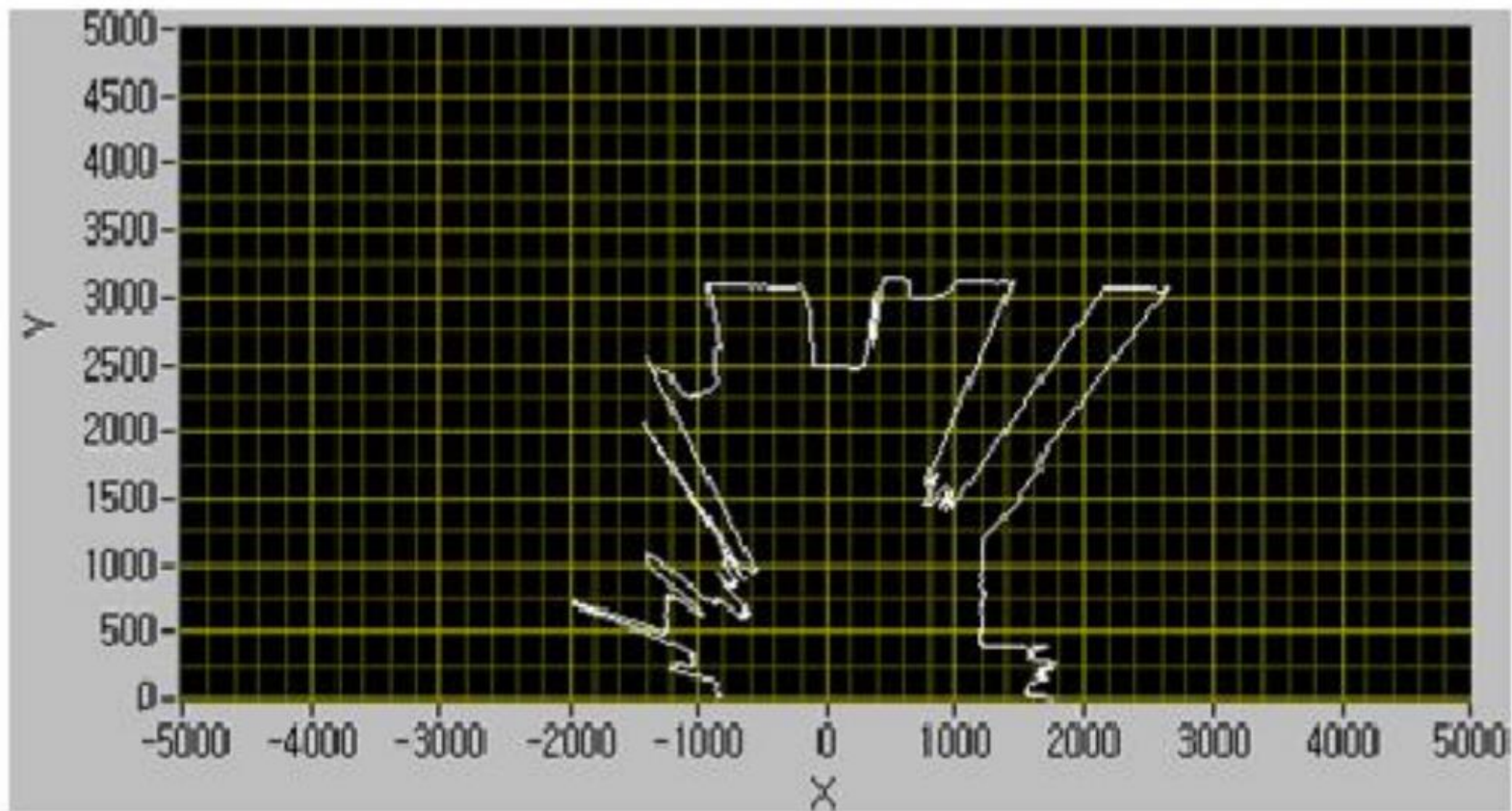
`A_fuzzy_obstacle_avoidance_controller_using_a_lookup_table_sharing_method_and_its_applications_for_mobile_robots.pdf`

Schemat układu sterowania



czujniki ultradźwiękowe

Mapa przeszkód



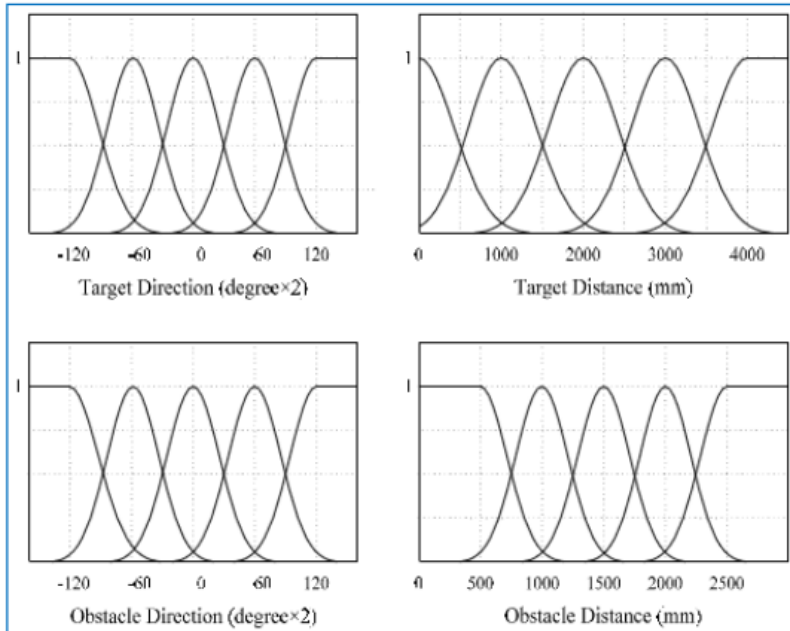
PN regule

Fuzzy rules of traditional fuzzy systems only describe the operations that are to be run. In Positive/Negative (P/N) rule fuzzy system theory, the concept of negative fuzzy rules was introduced to prescribe actions to avoid rather than execute. Two types of rules, positive rules and negative rules, are used in a P/N rule fuzzy controller. Positive-rules derive the proper output of the P/N-rule-based fuzzy system, comparable to typical fuzzy system rules. In contrast, negative rules cause the fuzzy system not

PN regule (cd)

to produce improper fuzzy output. The P/N rule fuzzy system can be applied to obstacle avoidance for the autonomous navigation of mobile robots. Positive rules direct the robot to a target place and negative rules prevent the robot from colliding with obstacles. The effect of positive rules is dominant in a non-obstacle environment because negative rules begin to affect navigation only when obstacles appear on the navigation path of the robot. If the detected obstacles are far from the robot, negative rules rarely affect the system. As the obstacles come closer to the robot, the negative rules dominate the system.

Funkcje przynależności



		Target		Obstacle	
		Center (c)	Variation (σ)	Center (c)	Variation (σ)
Direction (degree×2)	HL	-120	36	-120	36
	SL	-60	36	-60	36
	S	0	36	0	36
	SR	60	36	60	36
	HR	120	36	120	36
Distance (mm)	Z	0	6000	500	300
	VN	10000	6000	1000	300
	N	20000	6000	1500	300
	F	30000	6000	2000	300
	VF	40000	6000	2500	300

$$f(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

The fuzzy sets Hard Left (HL), Soft Left (SL), Straight (S), Soft Right (SR), and Hard Right (HR) are for the fuzzy inputs of the target or the obstacle direction. In addition, Zero (Z), Very Near (VN), Near (N), Far (Far), and Very Far (VF) are related to the fuzzy inputs of the target or the obstacle distance. The output fuzzy sets Hard Left (HL), Left (L), Soft Left (SL), Straight (S), Soft Right (SR), Right (R), and Hard Right (HR) are for positive rules, while \overline{HL} , \overline{L} , \overline{SL} , \overline{S} , \overline{SR} , \overline{R} , and \overline{HR} are for negative rules.

Regule pozytywne a negatywne

Rule 1P

If $x_{TargetDirection}$ is HL and $x_{TargetDistance}$ is Z then y is HL

Rule 2P

If $x_{TargetDirection}$ is HL and $x_{TargetDistance}$ is VN then y is HL

Rule 3P

If $x_{TargetDirection}$ is HL and $x_{TargetDistance}$ is N then y is L

·
·
·

Rule 25P

If $x_{TargetDirection}$ is HR and $x_{TargetDistance}$ is VF then y is SR.

Steering Direction		Target Direction				
		HL	SL	S	SR	HR
Target Distance	Z	HL	HL	S	HR	HR
	VN	HL	L	S	R	HR
	N	L	L	S	R	R
	F	L	SL	S	SR	R
	VF	SL	S	S	S	SR

Rule 1N

If $x_{ObstacleDirection}$ is HL and $x_{ObstacleDistance}$ is Z then y is not HL

Rule 2N

If $x_{ObstacleDirection}$ is HL and $x_{ObstacleDistance}$ is VN then y is not HL

Rule 3N

If $x_{ObstacleDirection}$ is HL and $x_{ObstacleDistance}$ is N then y is not L

·
·
·

Rule 25N

If $x_{ObstacleDirection}$ is HR and $x_{ObstacleDistance}$ is VF then y is not SR.

Steering Direction		Obstacle Direction				
		HL	SL	S	SR	HR
Obstacle Distance	Z	\overline{HL}	\overline{HL}	\overline{S}	\overline{HR}	\overline{HR}
	VN	\overline{HL}	\overline{L}	\overline{S}	\overline{R}	\overline{HR}
	N	\overline{L}	\overline{L}	\overline{S}	\overline{R}	\overline{R}
	F	\overline{L}	\overline{SL}	\overline{S}	\overline{SR}	\overline{R}
	VF	\overline{SL}	\overline{S}	\overline{S}	\overline{S}	\overline{SR}

Funkcje przynależności i obliczenia

Degrees of positive-rule fuzzy sets		Degree of negative-rule fuzzy sets	
HL	μ_{HL}	HL	$\mu_{\overline{HL}}$
SL	μ_{SL}	SL	$\mu_{\overline{SL}}$
S	μ_S	S	$\mu_{\overline{S}}$
SR	μ_{SR}	SR	$\mu_{\overline{SR}}$
HR	μ_{HR}	HR	$\mu_{\overline{HR}}$
Z	μ_Z	Z	$\mu_{\overline{Z}}$
VN	μ_{VN}	VN	$\mu_{\overline{VN}}$
N	μ_N	N	$\mu_{\overline{N}}$
F	μ_F	F	$\mu_{\overline{F}}$
VF	μ_{VF}	VF	$\mu_{\overline{VF}}$

$$\mu_{45} = (\text{offset} + \mu_S \cdot \mu_{VF}) \times \prod_{i=1}^{\text{#ofObstacle}} (1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{VF}_i}) (1 - \mu_{S_i} \cdot \mu_{Z_i}) (1 - \mu_{S_i} \cdot \mu_{\overline{VN}_i}) (1 - \mu_{S_i} \cdot \mu_{N_i}) (1 - \mu_{S_i} \cdot \mu_{F_i}) (1 - \mu_{S_i} \cdot \mu_{\overline{VF}_i}),$$

$$\mu_{46} = (\text{offset} + \mu_{SR} \cdot \mu_Z) \prod_{i=1}^{\text{#ofObstacles}} (1 - \mu_{\overline{SR}_i} \cdot \mu_{Z_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{Z_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{\overline{VN}_i}),$$

$$\mu_{47} = (\text{offset} + \mu_{SR} \cdot \mu_{VN}) \prod_{i=1}^{\text{#ofObstacle}} (1 - \mu_{\overline{SR}_i} \cdot \mu_{\overline{VN}_i}) (1 - \mu_{\overline{SR}_i} \cdot \mu_{N_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{N_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{F_i}),$$

$$\mu_{48} = (\text{offset} + \mu_{SR} \cdot \mu_N) \prod_{i=1}^{\text{#ofObstacle}} (1 - \mu_{\overline{SR}_i} \cdot \mu_{\overline{VN}_i}) (1 - \mu_{\overline{SR}_i} \cdot \mu_{N_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{N_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{F_i}),$$

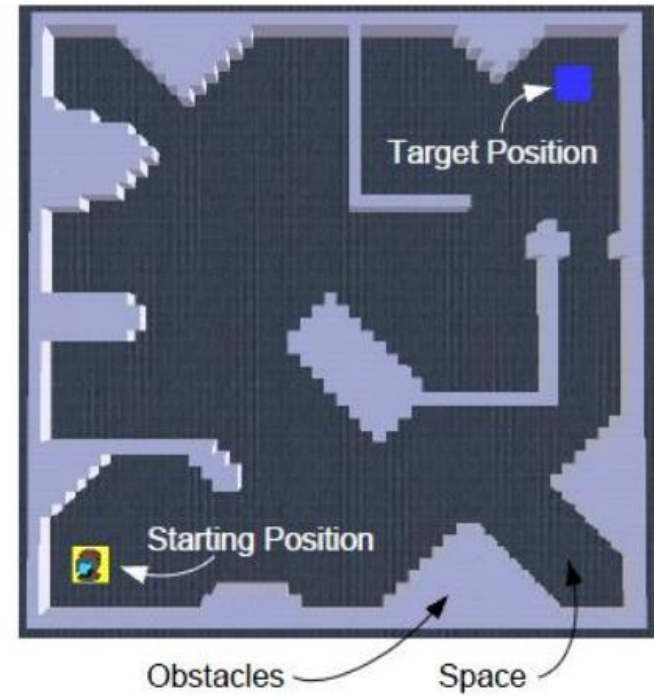
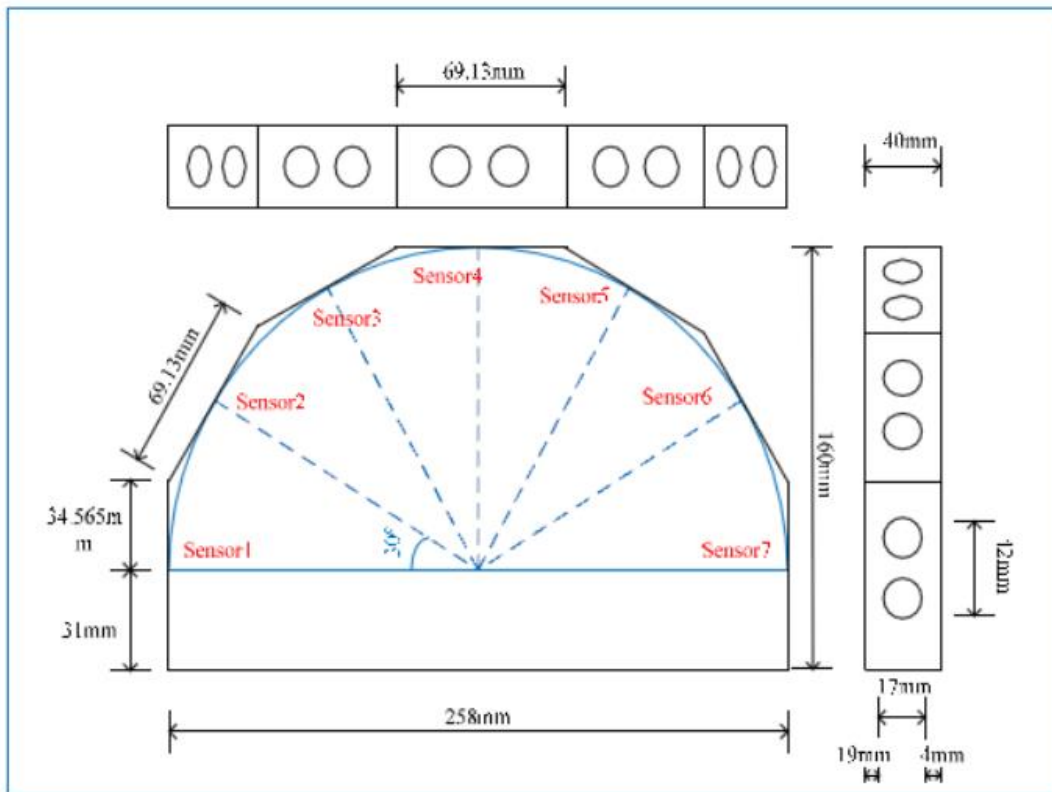
$$\mu_{49} = (\text{offset} + \mu_{SR} \cdot \mu_F) \prod_{i=1}^{\text{#ofObstacle}} (1 - \mu_{\overline{SR}_i} \cdot \mu_{F_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{\overline{VF}_i}),$$

$$\mu_{20} = (\text{offset} + \mu_S \cdot \mu_{VF}) \times \prod_{i=1}^{\text{#ofObstacle}} (1 - \mu_{\overline{SR}_i} \cdot \mu_{\overline{VF}_i}) (1 - \mu_{S_i} \cdot \mu_{Z_i}) (1 - \mu_{S_i} \cdot \mu_{\overline{VN}_i}) (1 - \mu_{S_i} \cdot \mu_{N_i}) (1 - \mu_{S_i} \cdot \mu_{F_i}) (1 - \mu_{S_i} \cdot \mu_{\overline{VF}_i}),$$

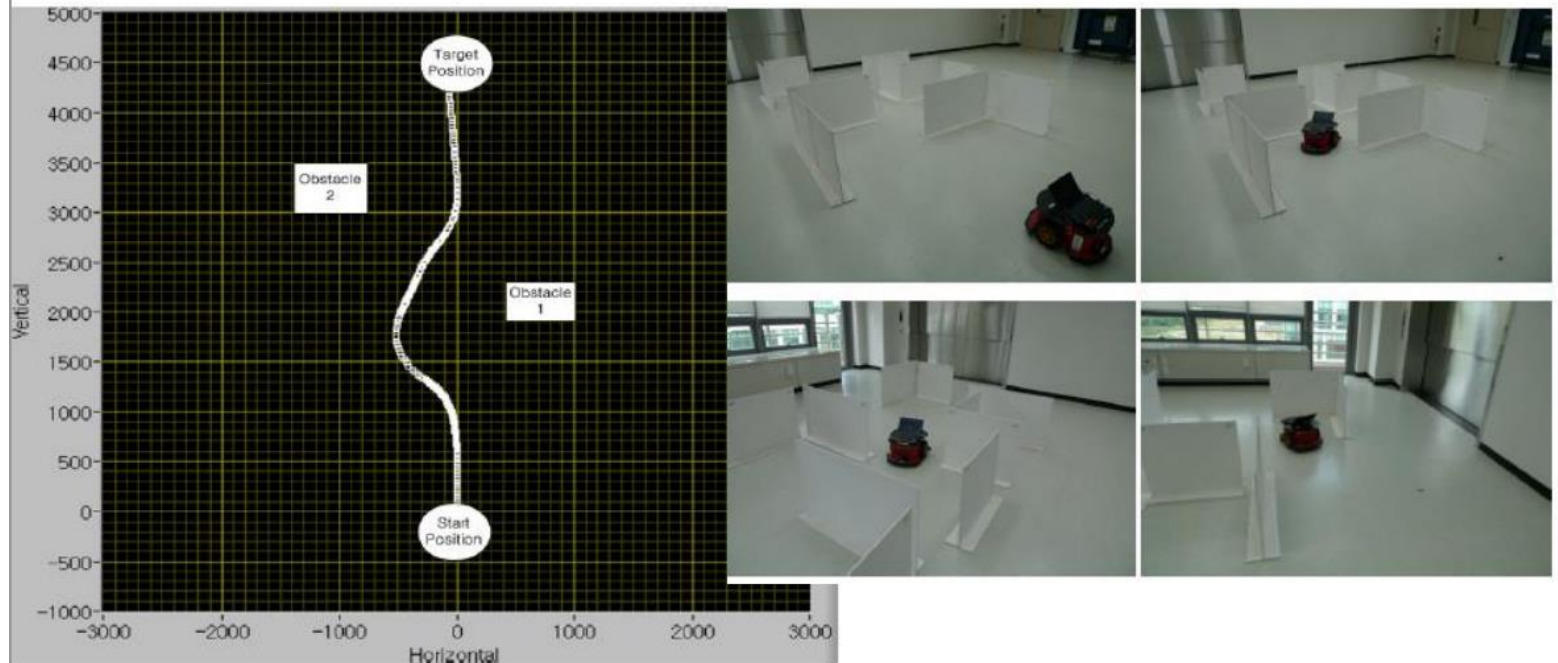
$$\mu_{21} = (\text{offset} + \mu_{HR} \cdot \mu_Z) \prod_{i=1}^{\text{#ofObstacle}} (1 - \mu_{\overline{SR}_i} \cdot \mu_{Z_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{Z_i}) (1 - \mu_{\overline{HR}_i} \cdot \mu_{\overline{VN}_i}),$$

... ..

Symulacja



Unikanie przeszkody

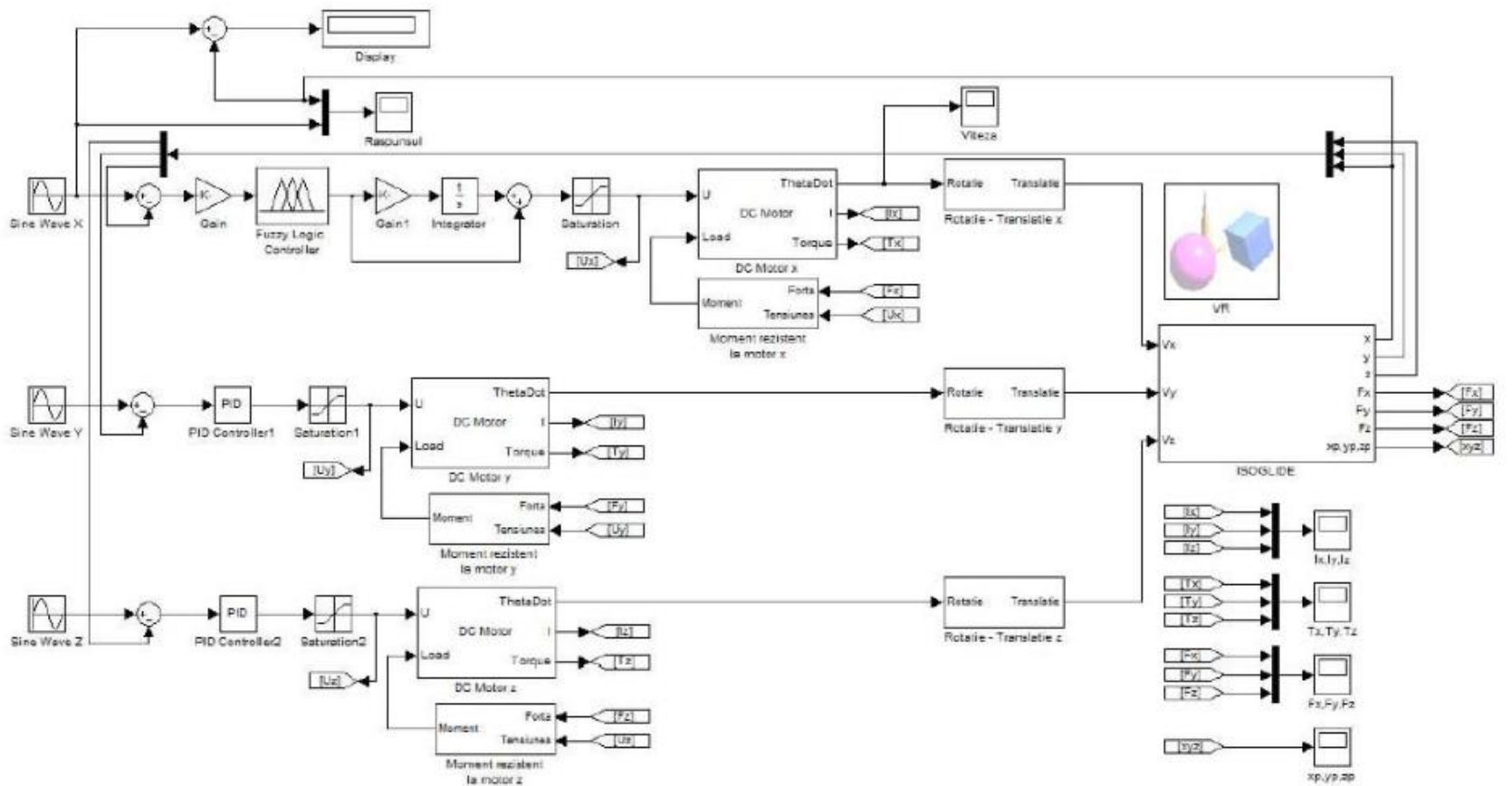


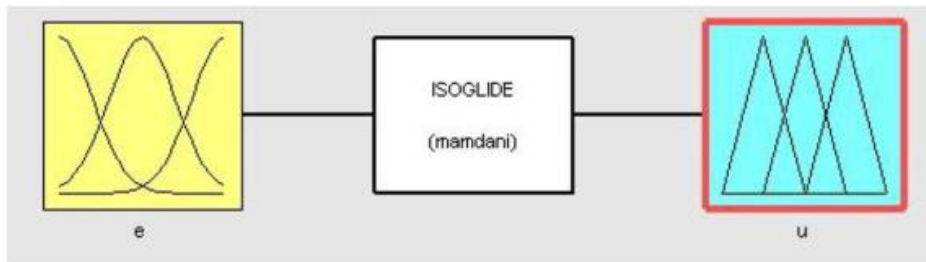
Kinematyka robota równoległego

3 ISOGLIDE3 DOF



[\LR\lit\control\Robots\Stan175.pdf](#)

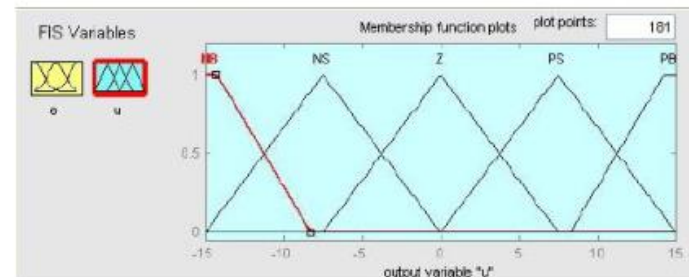
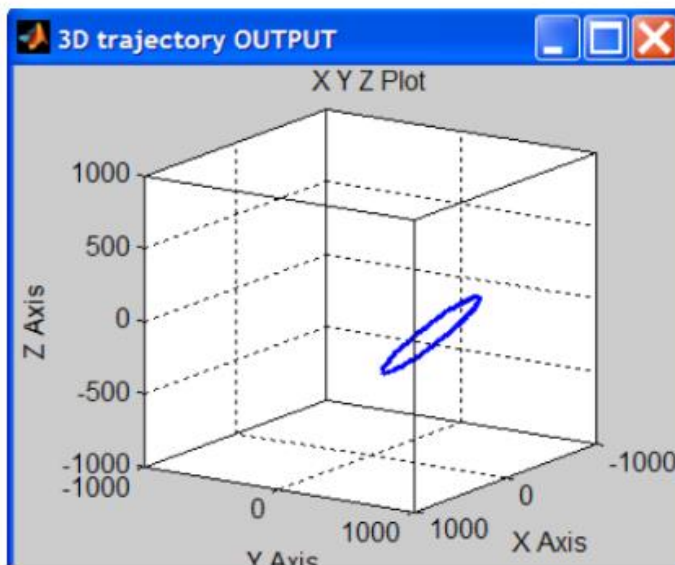




Rule Editor: ISOGLIDE

File Edit View Options

1. If (e is NB) then (u is NB) (1)
2. If (e is NS) then (u is NS) (1)
3. If (e is Z) then (u is Z) (1)
4. If (e is PS) then (u is PS) (1)
5. If (e is PB) then (u is PB) (1)



Analiza

The steps that were applied in the design procedure based on PID and fuzzy control were:

- tune a PID controller;
- replaced it with a linear fuzzy controller;
- fine-tune it.

The root of the problem is the conventional PID controller cannot adapt to the dynamics of the process. So we need some algorithm to adjust PID controller according to the dynamics of the process (Fig. 15).

So we need algorithm that can adapt with nonlinear behavior. Fuzzy logic matches to solve these problems because Fuzzy controller has some advantages:

- membership functions are simple triangular with fuzzy logic rules [6 - 9];
- stability of these fuzzy PID controllers is guaranteed [10 - 13].

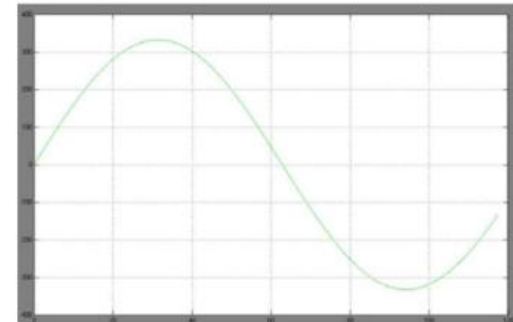


Fig. 14 Fuzzy control results for sinus tracking error

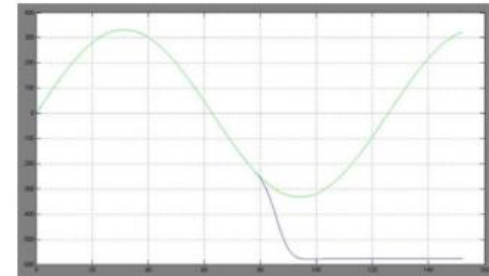


Fig. 15 PID control results for tracking error

Regulatory rozmyte typu Sugeno

Typowa reguła Sugeno dla modelu rozmytego z dwoma wejściami ma postać :

Jeśli (x jest A_j) i (y jest B_j), to $z_j = f(x,y)$

gdzie: x, y – zmienne wejściowe, A_j, B_j – zbiory rozmyte, $f(x,y)$ – funkcja, z – zmienna wyjściowa, określana ze wzoru defuzyfikacji (wyostrzania).

W *Fuzzy Logic Toolbox* funkcja $f(x,y)$ jest **wielomianem** rzędu pierwszego lub zerowego. Konkluzję j -tej reguły można zapisać w postaci:

$$***$z_j = f(x,y) = p_j x + q_j y + r_j$***$$

gdzie: p_j, q_j, r_j – współczynniki wielomianu, w modelu rzędu zerowego: $z_j = r_j$

Regulatory rozmyte PI typu Sugeno

Dla regulatora rozmytego PI typu Sugeno w postaci ogólnej – *j*-tą regułę można zapisać:

***Jeśli (w_1 jest A_j) i (w_2 jest B_j) i (w_3 jest C_j),
to $u_j = p1_j w1 + p2_j w2 + p3_j w3 + r_j$***

gdzie: A_j, B_j, C_j – zbiory rozmyte, $w1$ – zmienna wejściowa określająca nieliniowość regulatora,
 $w2 = E(\text{error})$, $w3 = SE(\text{sum of error})$, $p1_j, p2_j, p3_j, r_j$ – współczynniki w konkluzji.

Ważne zalety modelu typu Sugeno

- duża wydajność i skuteczność obliczeniowa
- dobre działanie w połączeniu z metodami stosowanymi w układach liniowych (np. regulator PID)
- dobra współpraca z metodami adaptacyjnymi i metodami optymalizacji
- struktura ułatwiająca przeprowadzenie dowodów stabilności układów sterowania.

Ważne zalety modelu typu Sugeno

Model Sugeno jest bardziej zwięzły i skuteczniejszy obliczeniowo w porównaniu z modelem Mamdani.

Ponadto model rozmyty typu Sugeno umożliwia łatwe **stosowanie metod adaptacyjnych** do doboru jego parametrów.

Kilka z tych metod jest zaimplementowanych w *Fuzzy Logic Toolbox*.

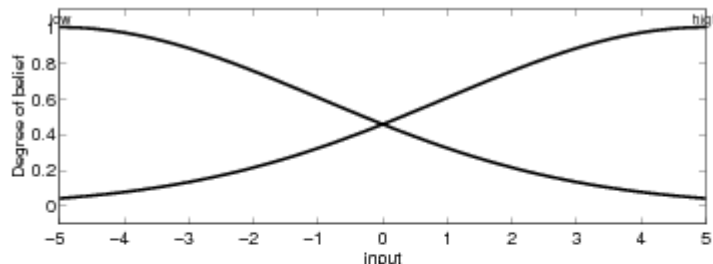
Na podstawie modelu Sugeno można zbudować **układ rozmyty, który dokonuje przełączeń pomiędzy kilkoma optymalnymi liniowymi regulatorami jako silnie nieliniowy układ** poruszający się wokół ich punktu pracy.

Aproksymacja (sugeno1.m)

07-12-2023

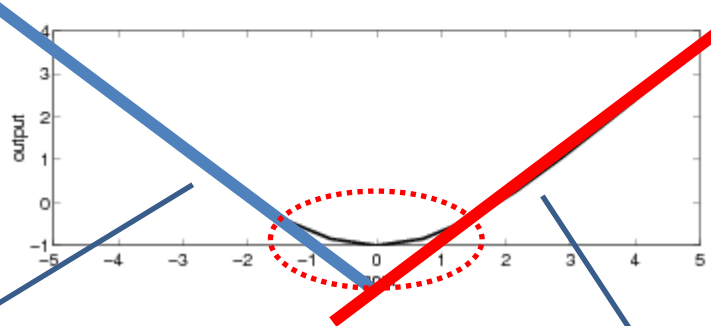
```
fismat = readfis('sugeno1');
```

1. If (input is **low**) then (output is line1) (1)
2. If (input is **high**) then (output is line2) (1)



$$output = (-1) \times input + (-1)$$

$$output = (1) \times input + (-1)$$

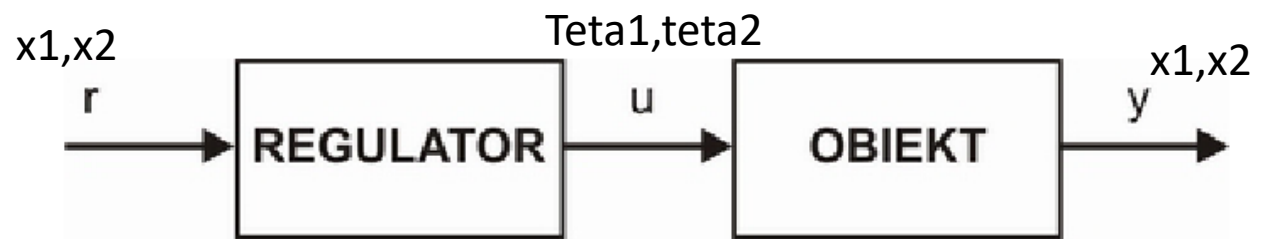
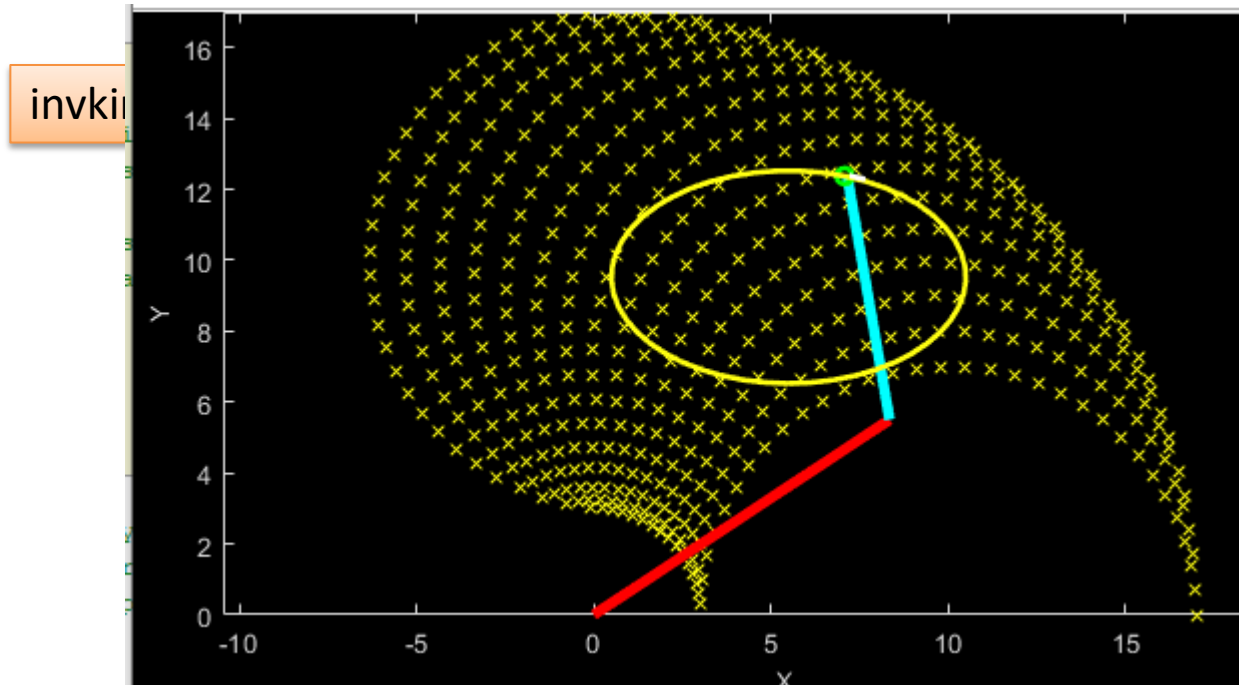


$$y(x) = -x - 1$$

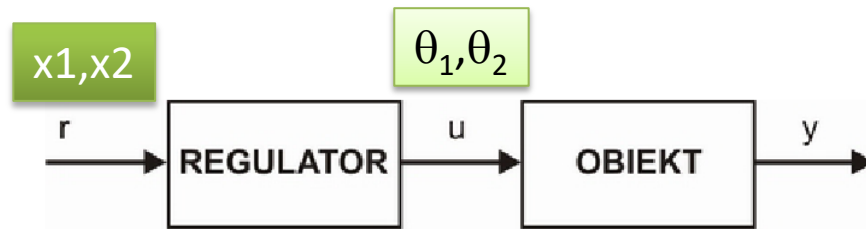
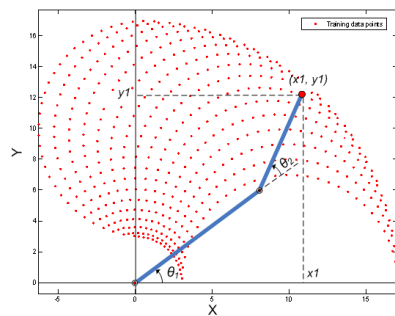
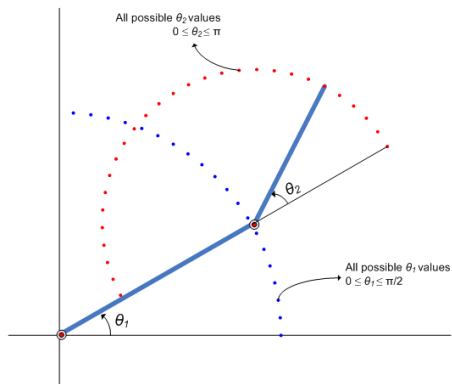
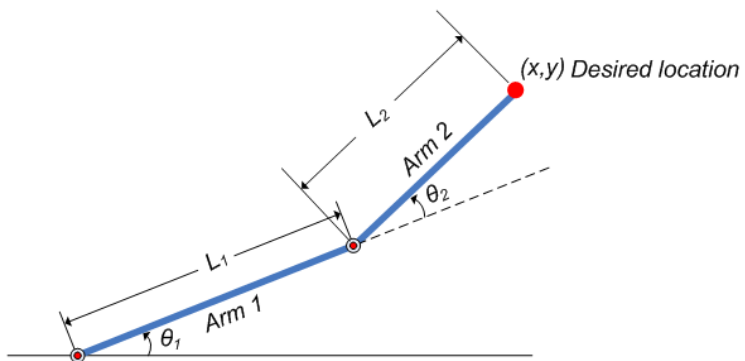
$$y(x) = x - 1$$

If x jest ujemny to $y = a_1 * x + a_2$
 If x jest dodatni to $y = b_1 * x + b_2$
 $y' = ((miUjemny(0) * (-1) + miDodatni(0) * (-1))) /$
 $' = ((miUjemny(0) + miDodatni(0)) =$
 $(0.25 * (-1) + 0.25 * (-1)) / (0.25 + 0.25) = -1$

Kinematyka robota



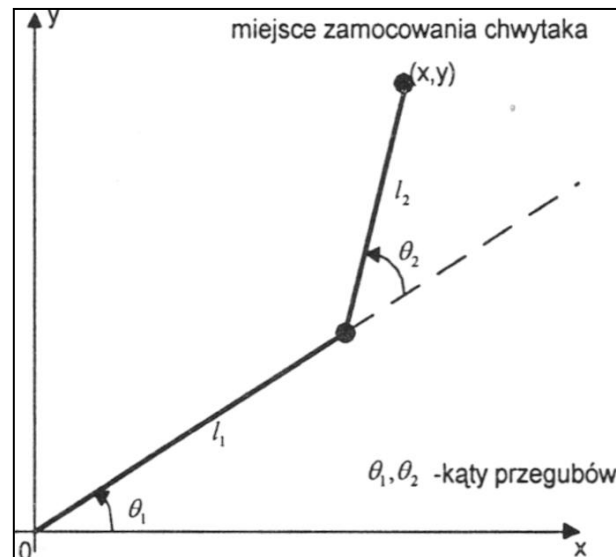
Układ otwarty



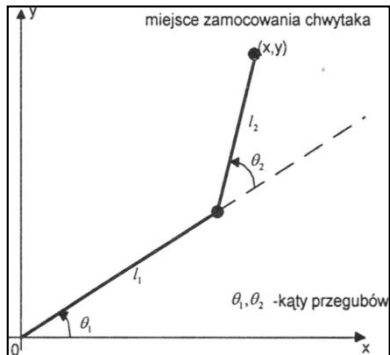
Sterowanie ruchem ramienia

W celu realizacji pożądanego ruchu ramienia należy rozwiązać dwa problemy:

- **odwrotne zadanie kinematyczne** - dane są współrzędne położenia chwytaka, a należy znaleźć odpowiadające im kąty przegubów,
- **sterowanie w celu uzyskania pozycji końcowej** - dane jest bieżące i końcowe położenie chwytaka, a należy znaleźć ciąg kątów przegubów zapewniających przejście od położenia początkowego do końcowego.



Równania kinematyczne



proste

$$x_1 = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

$$x_2 = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

odwrotne

$$\theta_2 = \arccos\left(\frac{x_1^2 + x_2^2 - l_1^2 - l_2^2}{2l_1 l_2}\right)$$

$$\theta_1 = \arctg\left(\frac{x_2}{x_1}\right) - \arctg\left(\frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2}\right)$$

θ_1, θ_2 - kąty przegubów, x_1, x_2 - współrzędne pozycji chwytaka, l_1, l_2 - długość ramion.

Ponieważ funkcja **arccos** jest różnowartościowa, wynikają **dwie możliwe wartości kątów dla tej samej pozycji chwytaka**. Stąd widać, że odwzorowanie przestrzeni kartezyjskiej w przestrzeń kątów przegubów nie jest jednoznaczne.

Sterowanie ramieniem robota

Sterowanie ramieniem robota wymaga rozwiązywania równań kinematyki w czasie rzeczywistym. Dla ramienia o n stopniach swobody zadania kinematyki prostej i odwrotnej można sformułować odpowiednio jako

$$\mathbf{x} = \mathbf{h}(\theta)$$

$$\theta = \mathbf{h}^{-1}(\mathbf{x})$$

gdzie θ i \mathbf{x} są odpowiednio kątami przegubów i położeniami chwytaka zdefiniowanymi następująco:

$$\theta \stackrel{\text{def}}{=} [\theta_1 \theta_2 \dots \theta_n]^t, \quad \mathbf{x} \stackrel{\text{def}}{=} [x_1 x_2 \dots x_n]^t$$

Uzyskanie rozwiązań numerycznych równania $\theta = \mathbf{h}^{-1}(\mathbf{x})$ w czasie rzeczywistym wymaga dużych mocy obliczeniowych. Zamiast wykonywania obliczeń numerycznych można zastosować sieci neuronowe zarówno do **sformułowania równań**, jak i do **ich rozwiązywania**.

Dobór struktury i parametrów początkowych regulatora rozmytego

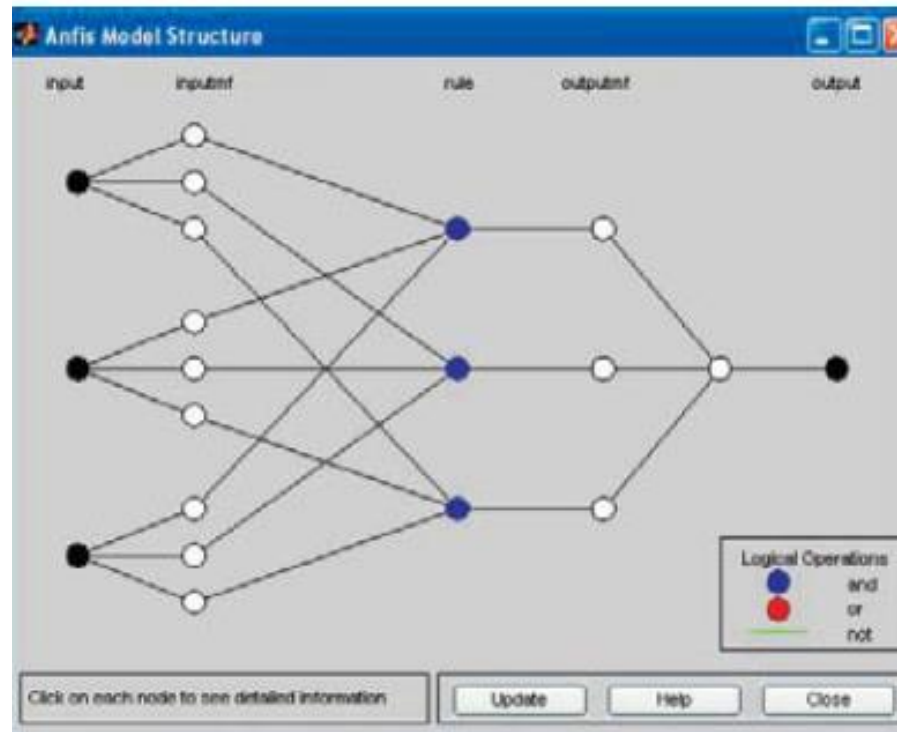
Określenie wstępnej struktury i parametrów rozmytego regulatora typu Sugeno można zrealizować na kilka sposobów:

- a. Na podstawie **wiedzy eksperta** można arbitralnie określić liczbę reguł oraz rodzaj, kształt (parametry) i liczbę funkcji przynależności dla każdej zmiennej wejściowej.
- b. Na podstawie dostępnych danych uczących – można użyć funkcji lub interfejsu **ANFIS**. Wygenerowany w ten sposób model typu Sugeno ma określoną liczbę reguł, parametry funkcji przynależności w przesłankach oraz wartości współczynników wielomianu w konkluzjach reguł.
- c. Na podstawie dostępnych danych uczących – można zastosować funkcje biblioteczne: **genfis1, genfis2 lub genfis3**. Funkcje te generują struktury reprezentujące układy rozmyte typu Sugeno z ustaloną liczbą reguł oraz z parametrami funkcji w przesłankach i konkluzjach.

ANFIS (anfisedit)

W bibliotece *Fuzzy Logic Toolbox* istnieje możliwość strojenia regulatorów rozmytych typu Sugeno **metodami uczenia stosowanymi w sieciach neuronowych**. W tym celu **model rozmyty** jest przekształcany w równoważną wielowarstwową sieć neuronową (**perceptron wielowarstwowy**).

Takie strojenie realizuje funkcja `anfis` lub interfejs **ANFIS Editor (anfisedit)**. Akronim ANFIS pochodzi od *Adaptive Neuro-Fuzzy Inference System*.



Struktura sieci



Jyh-Shing Roger
Jang [Tsing Hua](#)
[University](#), Taiwan

Algorytm strojenia zastosowany
w funkcji **anfis** zaproponował Jang.

Przekształcony w sieć neuronową model rozmyty jest reprezentowany
przez **5 warstw neuronów** następujących po sobie:

Warstwa 1 to wartości wejściowe (**input**).

Warstwa 2 odpowiada za fuzyfikację (rozmywanie) wartości
wejściowych

i reprezentuje funkcje przynależności w przesłankach
(**inputmf** – najczęściej są to funkcje Gaussa).

Warstwa 3 (rule) reprezentuje reguły i na wyjściu każdego jej neuronu
jest określany stopień aktywacji odpowiedniej reguły.

Pozostałe warstwy (outputmf i output) realizują wzór defuzyfikacji dla
modelu Sugeno.

Uczenie sieci

Uczenie sieci neuronowej polega na **stopniowej zmianie jej wag** w taki sposób, aby doprowadzić do minimalizacji kryterium nauczania. Jest nim zwykle **średni kwadratowy błąd** wyjścia sieci względem danych uczących.

Funkcja **anfis** przekształca model rozmyty w sieć neuronową, jak opisano powyżej. **Wagi tej sieci odpowiadają parametrom funkcji przynależności** w przesłankach i współczynnikom wielomianu w konkluzjach.

W ten sposób można uzyskać **samostrójący się regulator** rozmyty na bazie danych pomiarowych typu **wejście-wyjście**.

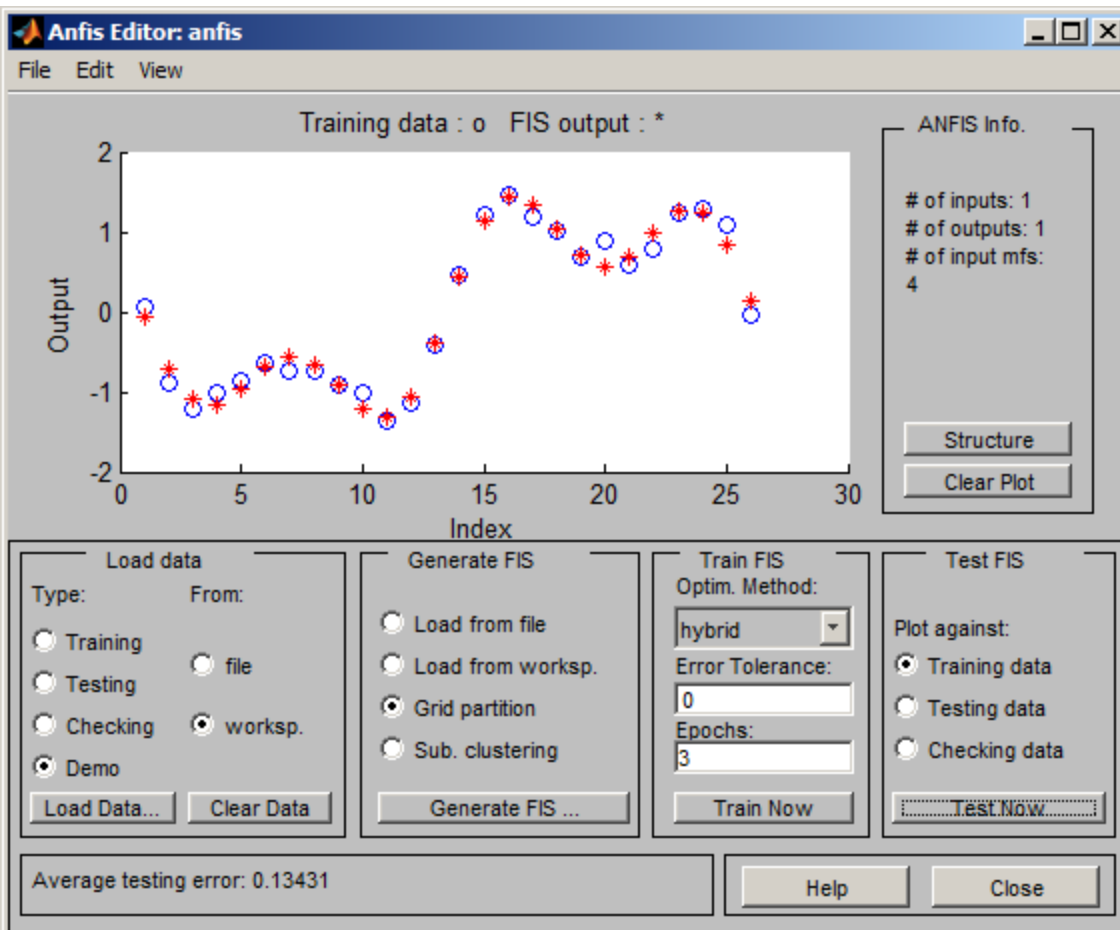
Za **samostrójący się regulator rozmyty** uznaje się regulator o

- ustalanej liczbie reguł,
- zbiorów rozmytych w przesłankach i
- wielomianów w konkluzjach.

Zmianie (strojeniu) podlegają **jedynie parametry funkcji przynależności w przesłankach i współczynniki wielomianów w konkluzjach**.

Ograniczenia funkcji i interfejsu ANFIS

Interfejs i funkcja anfis działa poprawnie tylko na modelach rozmytych typu **Sugeno rzędu zerowego lub pierwszego**. Oznacza to, że w konkluzjach modelu mogą występować tylko stałe lub funkcje liniowe.



Inne ograniczenia:
wszystkie funkcje wyjściowe w konkluzjach muszą być tego samego typu i muszą być albo stałymi albo funkcjami liniowymi
funkcje w konkluzjach nie mogą się powtarzać, ich liczba musi być równa liczbie reguł
należy określić funkcje przynależności dla wszystkich zmiennych wejściowych
układ rozmyty musi mieć wagę jedynkową dla każdej reguły.

Określenie reguł na podstawie danych pomiarowych

Określenie reguł (bazy reguł), liczby funkcji przynależności oraz ich parametrów zależy od umiejętności wykrywania klasterów (*cluster – grono, pęk, rój*) w próbkach pomiarowych i określania ich środków ciężkości.

Fuzzy Logic Toolbox zawiera funkcje **fcm** i **subclust**, które realizują algorytmy klasteryzacji odpowiednio metodą *c-środków* (*c-means clustering*) i klasteryzacji różnicowej (*subtractive clustering*).

Fuzzy Logic Toolbox zawiera też dwie funkcje (**genfis2** i **genfis3**), które **generują struktury reprezentujące układy rozmyte typu Sugeno**. Liczba i parametry funkcji przynależności w przesłankach są określane poprzez algorytmy odpowiednio:

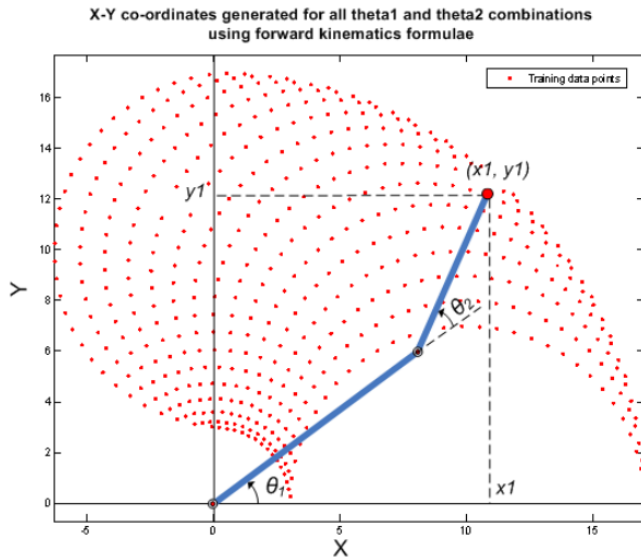
klasteryzacji różnicowej

i klasteryzacji metodą *c-środków*. Wartości współczynników wielomianu w konkluzjach są wyznaczone metodą najmniejszych kwadratów.

Funkcja **genfis1** generuje modele rozmyte typu Sugeno na podstawie siatki podziału (*grid partition*) – bez klasteryzacji. Wartości współczynników wielomianu w konkluzjach reguł mają wartości zerowe, a liczba reguł jest duża.

Wygenerowane za pomocą funkcji **genfis1**, **genfis2** i **genfis3** modele rozmyte są zazwyczaj stosowane **jako modele początkowe (wstępne)** dla funkcji lub interfejsu **ANFIS**.

Dane do uczenia



$$x_1 = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

$$x_2 = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

```
l1 = 10; % length of first arm  
l2 = 7; % length of second arm
```

```
theta1 = 0:0.1:pi/2; % all possible theta1 values  
theta2 = 0:0.1:pi; % all possible theta2 values
```

```
[THETA1,THETA2] = meshgrid(theta1,theta2); % generate a grid of  
theta1 and theta2 values
```

```
X = l1 * cos(THETA1) + l2 * cos(THETA1 + THETA2); % compute x  
coordinates
```

```
Y = l1 * sin(THETA1) + l2 * sin(THETA1 + THETA2); % compute y  
coordinates
```

```
data1 = [X(:) Y(:) THETA1(:)]; % create x-y-theta1 dataset
```

```
data2 = [X(:) Y(:) THETA2(:)]; % create x-y-theta2 dataset
```

```
plot(X(:),Y(:),'r');
```

```
axis equal;
```

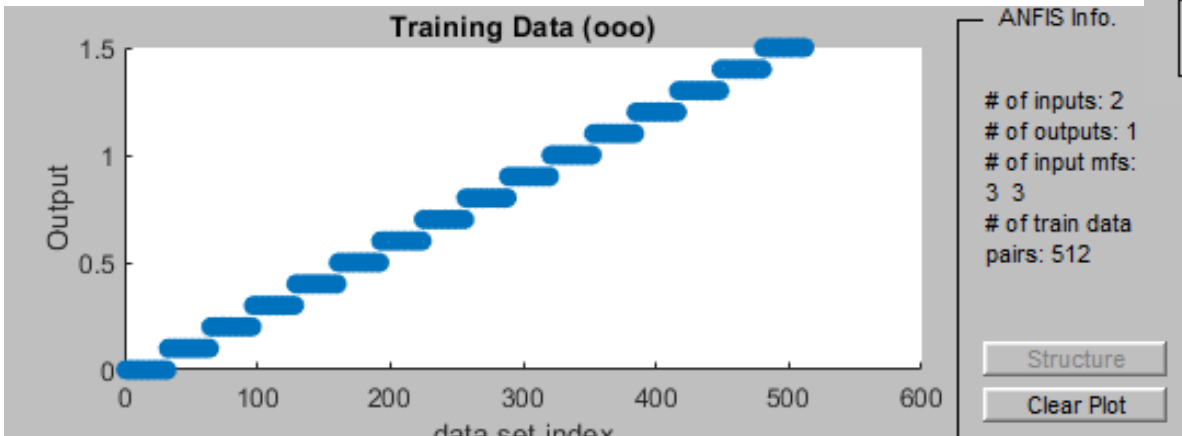
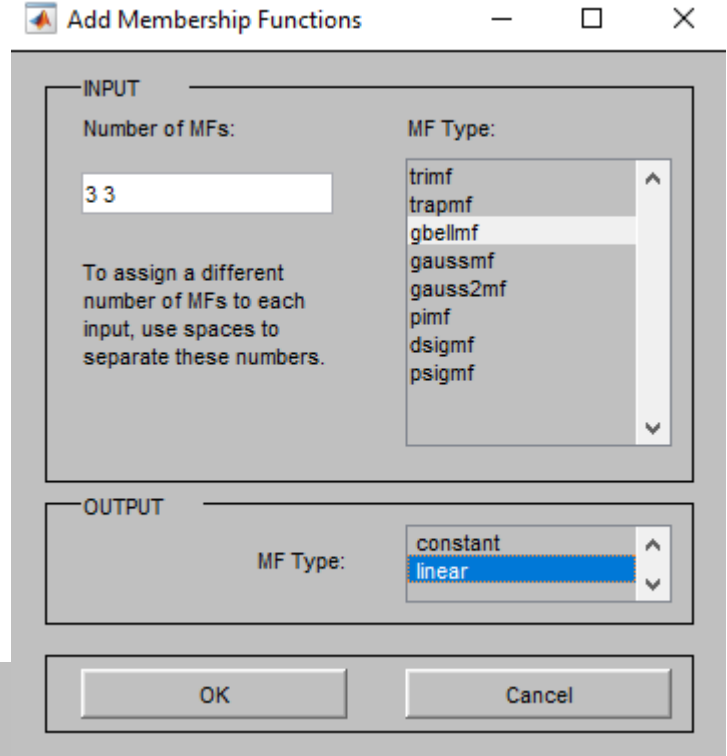
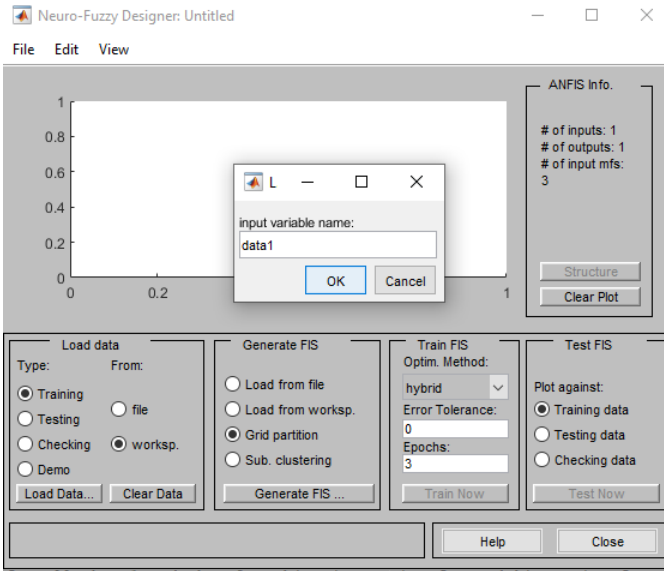
```
xlabel('X','fontsize',10)
```

```
ylabel('Y','fontsize',10)
```

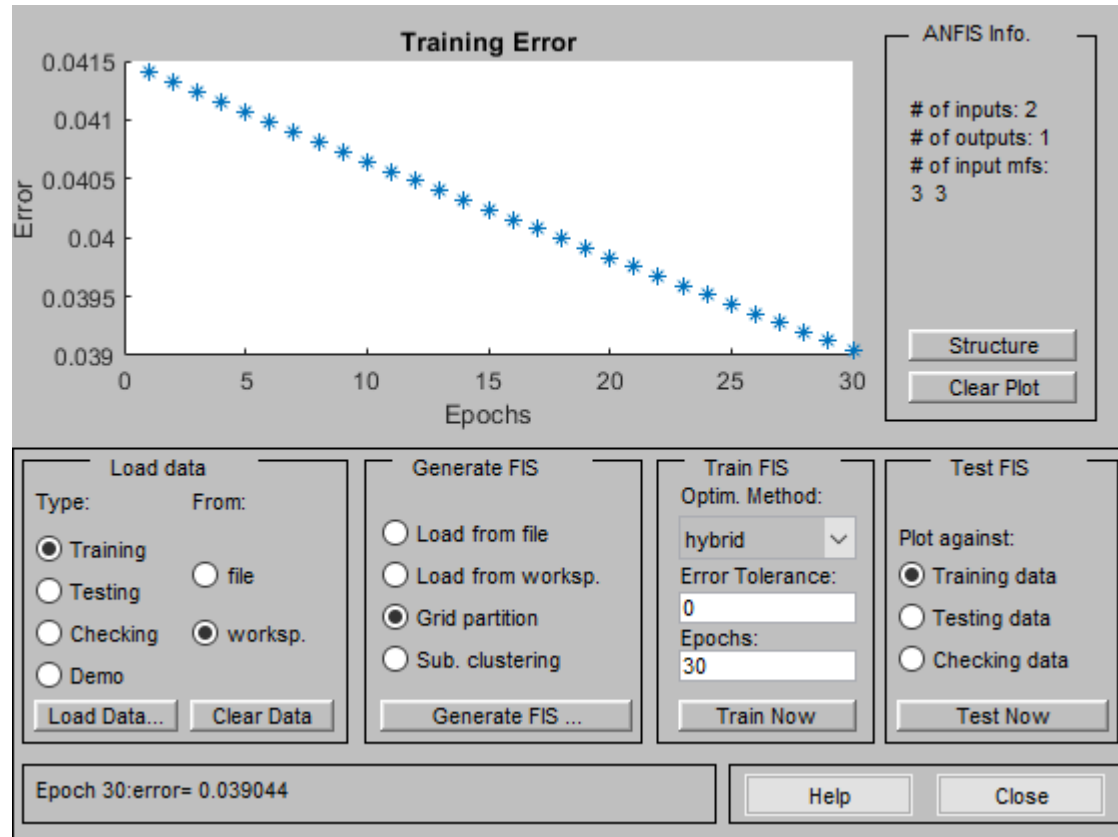
```
title('X-Y coordinates generated for all theta1 and theta2 combinations  
using forward kinematics formula','fontsize',10)
```

armOS.m

Zastosowanie ANFIS do zagadnienia kinematyki robota

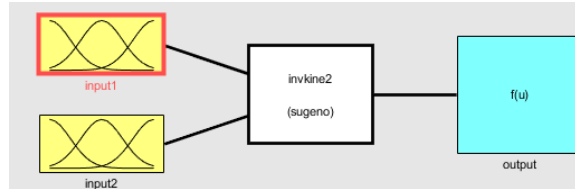


Uczenie modelu rozmytego

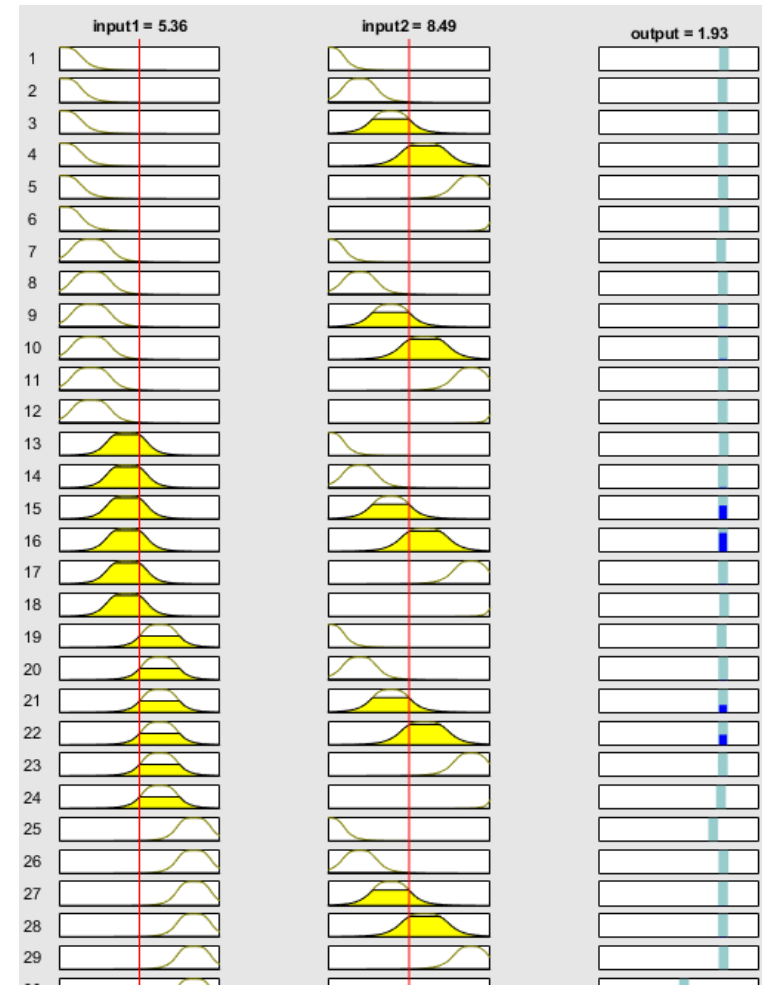
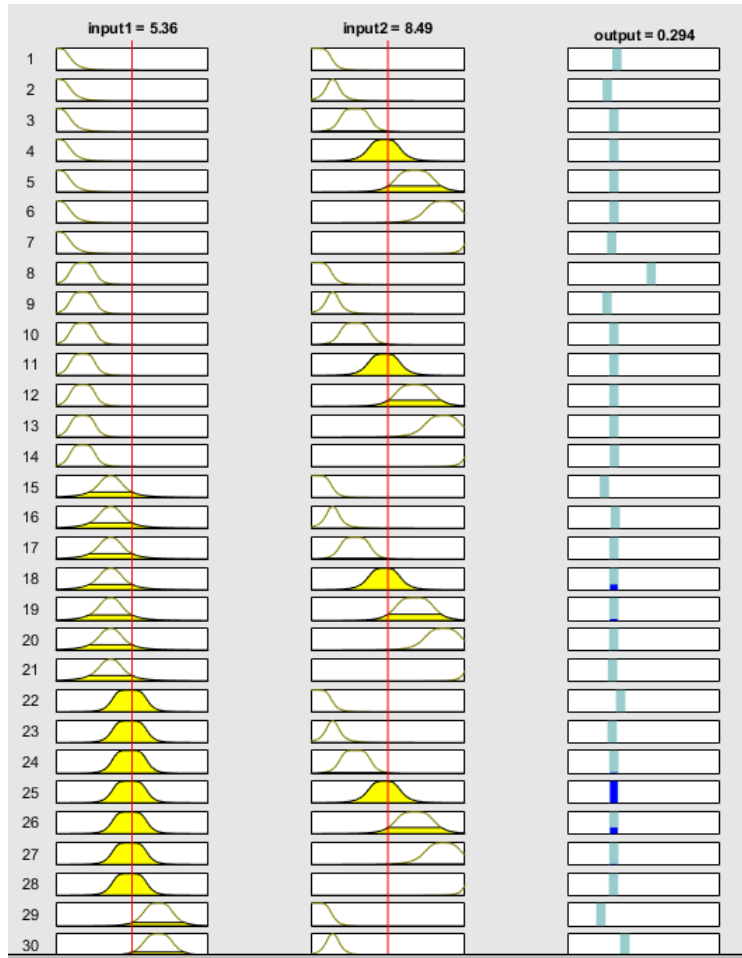


Model Sugeno 1 rzędu

Invike1.fis



Invike2.fis



Symulacja

```
%after ANFISEDIT!!!!!!  
Teta1=readfis('invkine1.fis');  
Teta2=readfis('invkine2.fis');  
x1=-0.194539443391643;  
x2=16.9166035392078;  
  
T1=evalfis(Teta1,[x1,x2]);  
T2=evalfis(Teta2,[x1,x2]);
```

```
>> T1  
T1 =  
    1.49812901716626  
  
>> T2  
T2 =  
    0.195470665968672
```

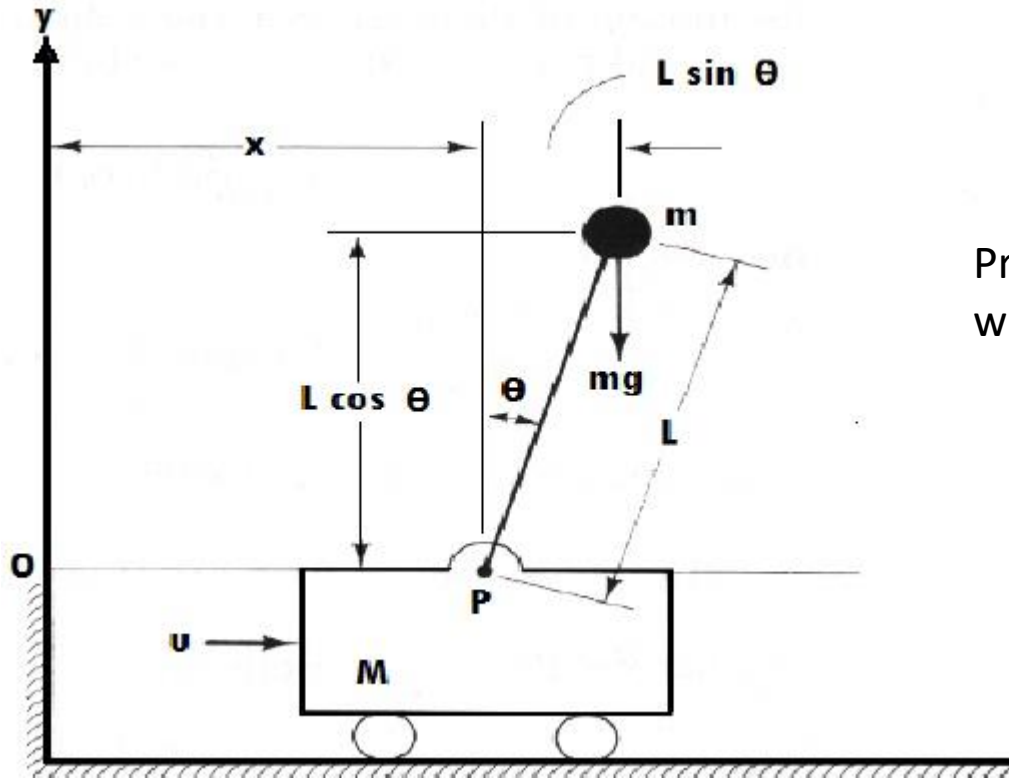
data1

482	0.5030	16.9720	1.5000
483	-0.1945	16.9166	1.5000

data2

482	0.5030	16.9720	0.1000
483	-0.1945	16.9166	0.2000

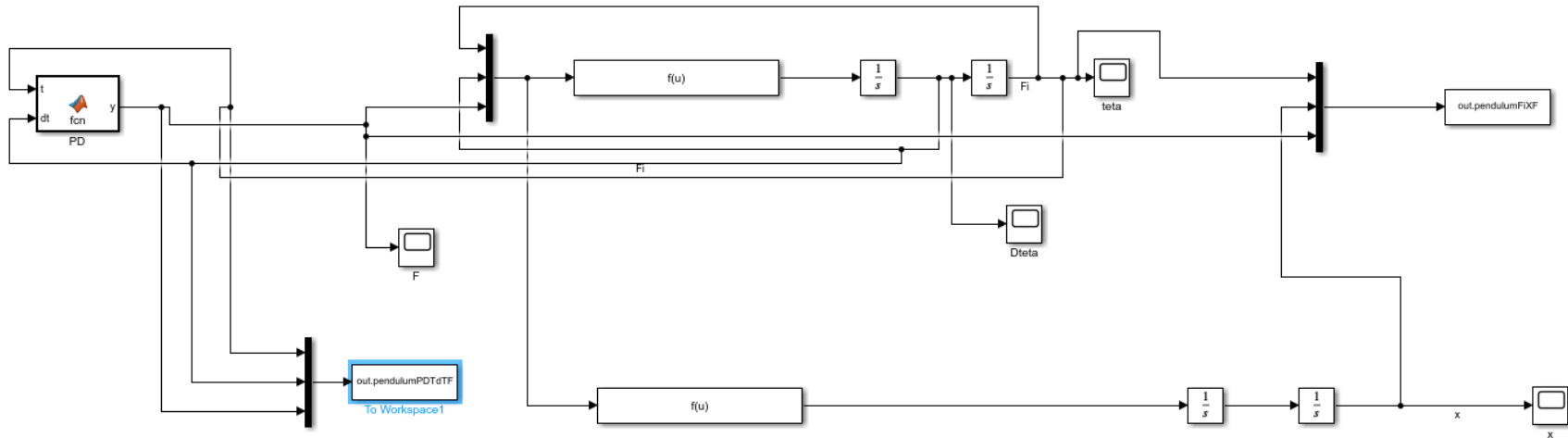
Wahadło (slcp)



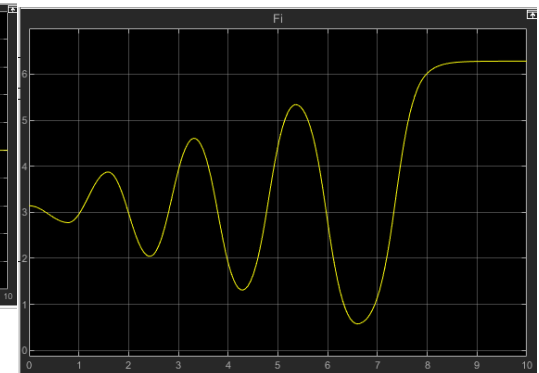
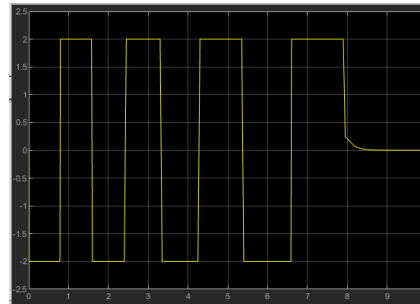
Przemieszczenie wahadła
w punkt docelowy

PD vs Sugeno

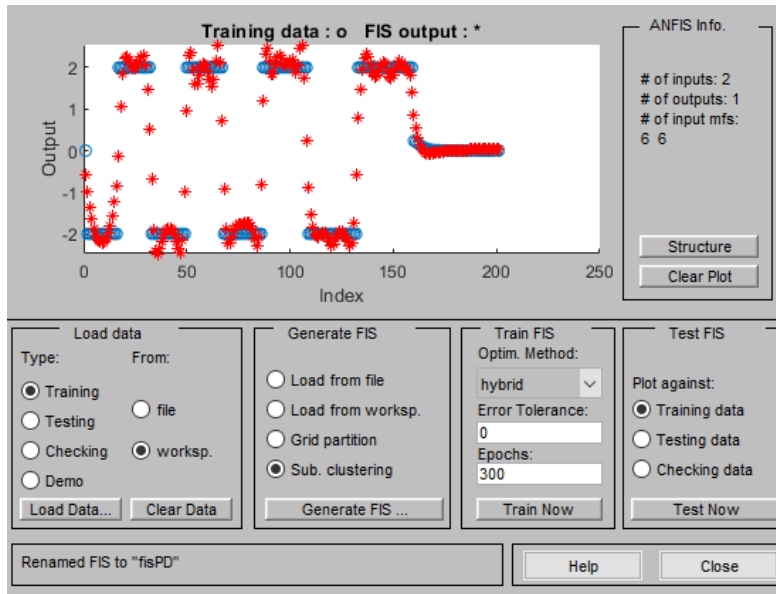
InvPendulumSugenoPD



```
function y = fcn(t,dt)
K1=-40;K2=-10;F0=2;
if (abs(t)>=0)*(abs(t)<=pi/9)
    y=K1*(0-t)+K2*(0-dt);
elseif (abs(t)>=17*pi/9)*(abs(t)<=2*pi)
    y=K1*(2*pi-t)+K2*(0-dt);
else
    y=F0*sign(dt);
end;
```



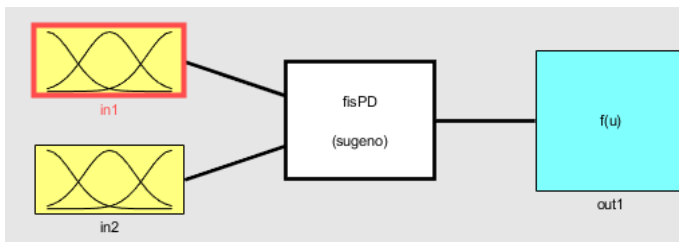
Nauczanie za pomocą ANFIS



out.pendulumFixF

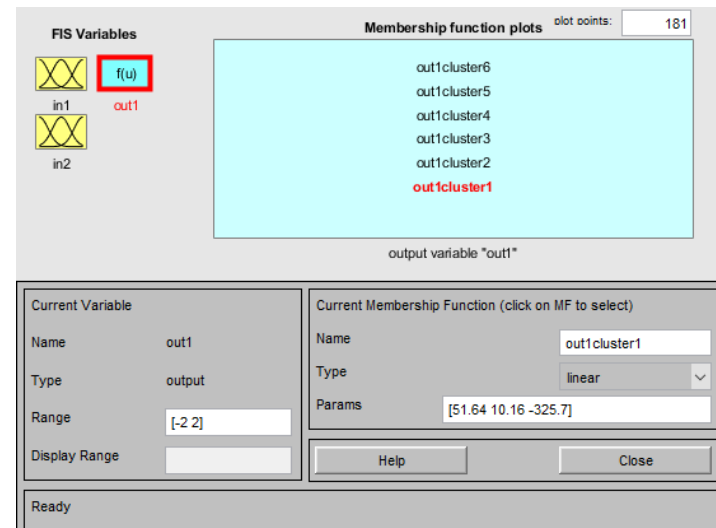
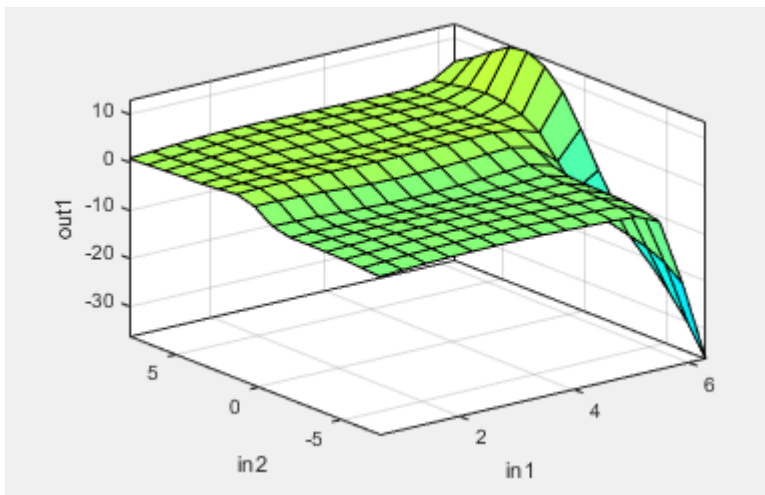
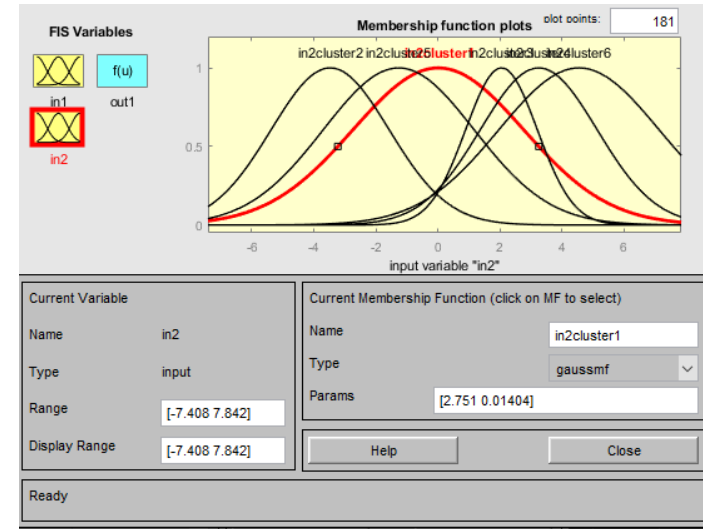
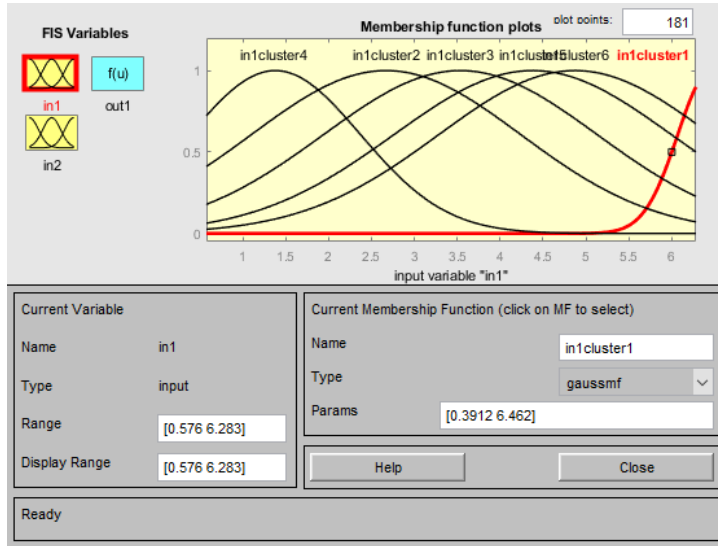
Time series name:

Time	Data:1	Data:2	Data:3
0	3.1416	0	-0.5744
0.0500	3.1405	-7.5289e-04	-0.7095
0.1000	3.1368	-0.0032	-0.8671
0.1500	3.1301	-0.0078	-1.0425
0.2000	3.1201	-0.0150	-1.2283
0.2500	3.1065	-0.0250	-1.4149
0.3000	3.0890	-0.0385	-1.5915
0.3500	3.0679	-0.0557	-1.7479
0.4000	3.0433	-0.0771	-1.8764
0.4500	3.0157	-0.1028	-1.9718
0.5000	2.9860	-0.1332	-2.0314
0.5500	2.9551	-0.1681	-2.0531
0.6000	2.9242	-0.2078	-2.0333
0.6500	2.8945	-0.2519	-1.9648
0.7000	2.8675	-0.3004	-1.8342



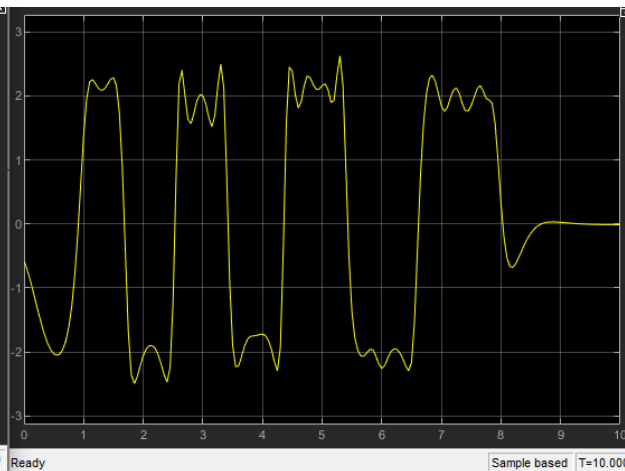
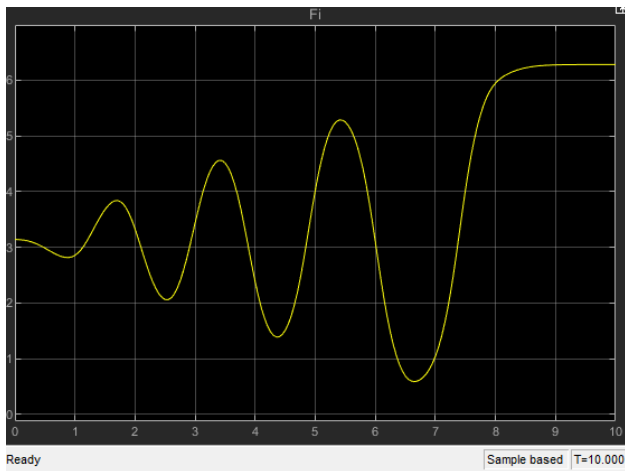
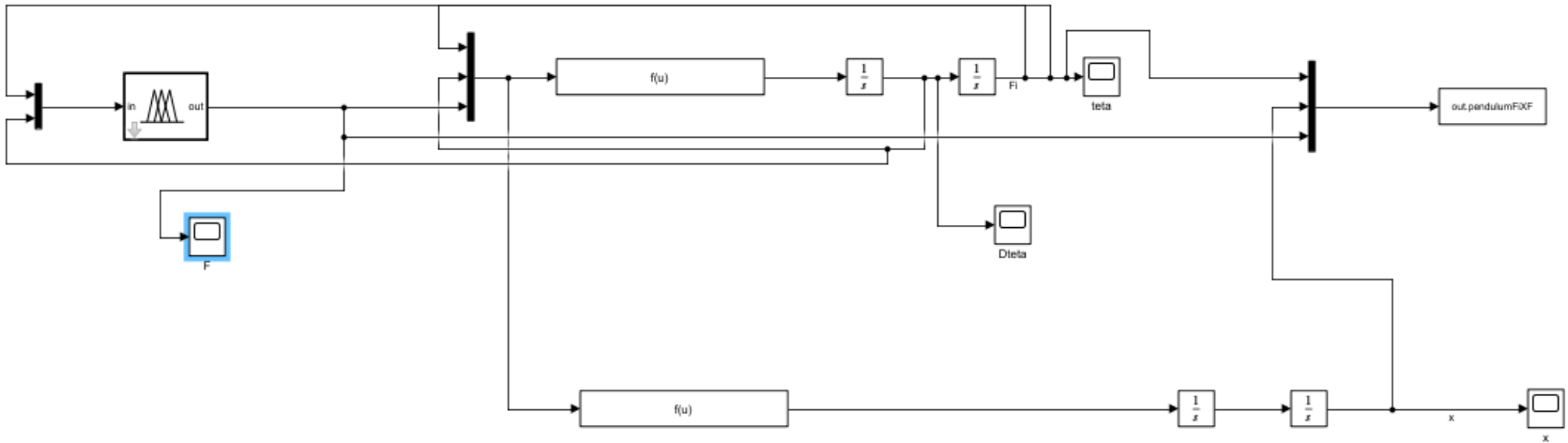
Subclustering

fisPD

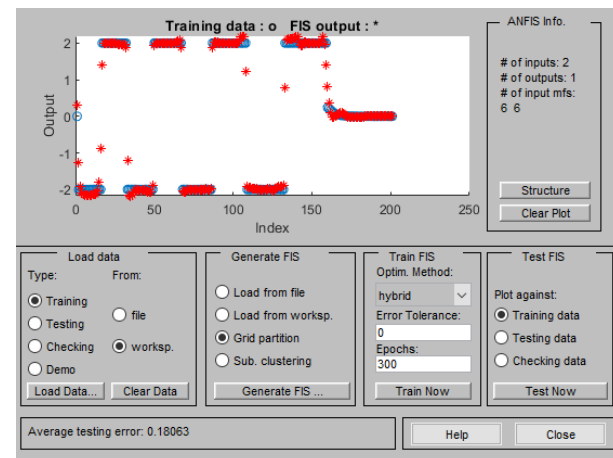
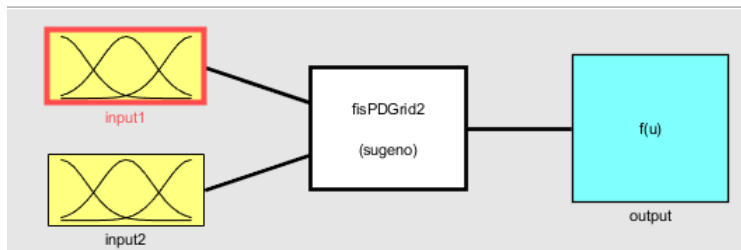
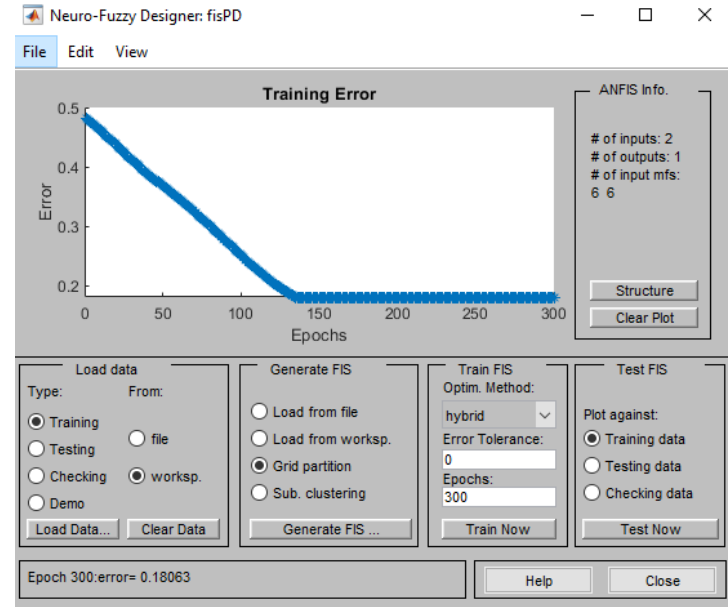
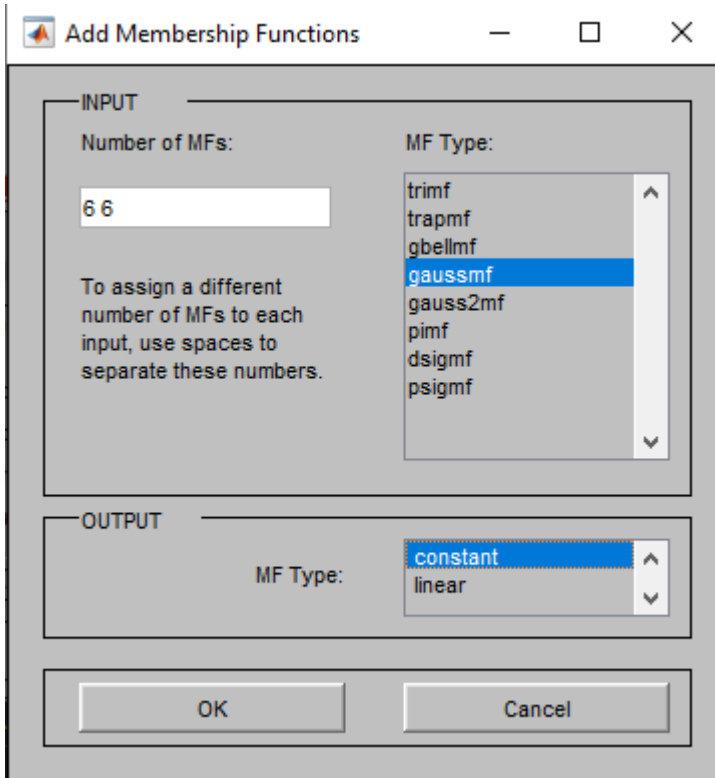


PD vs Sugeno

InvPendulumSugenoANFIS

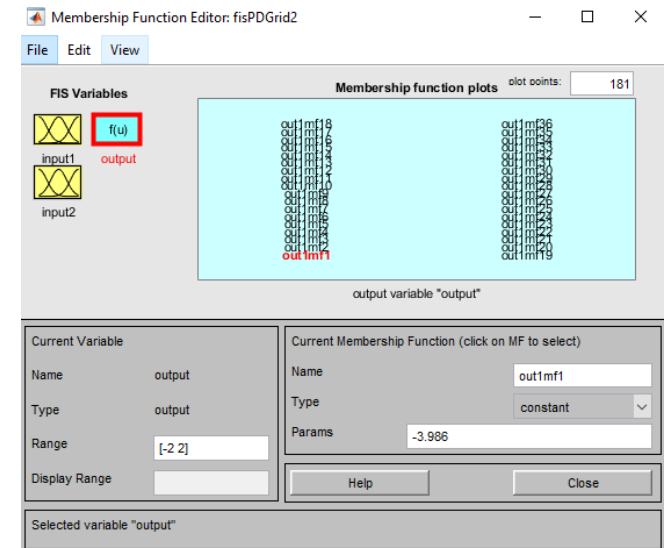
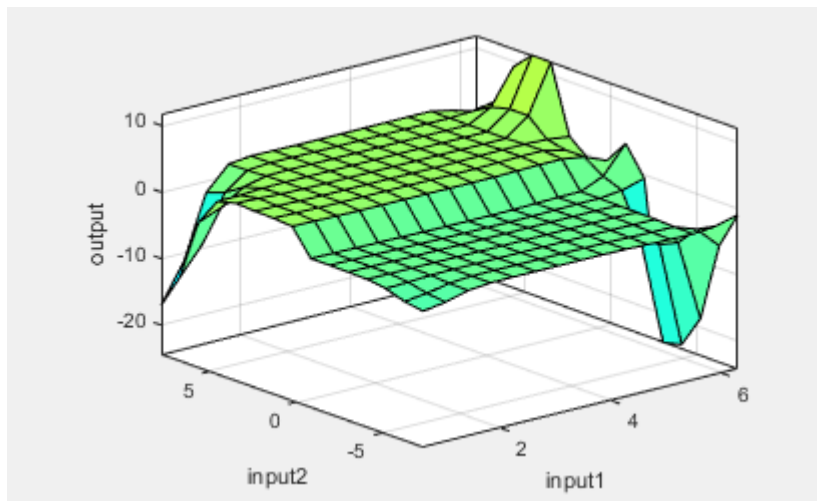
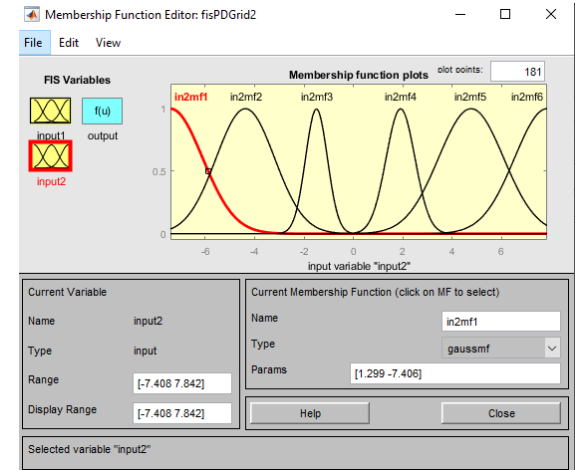
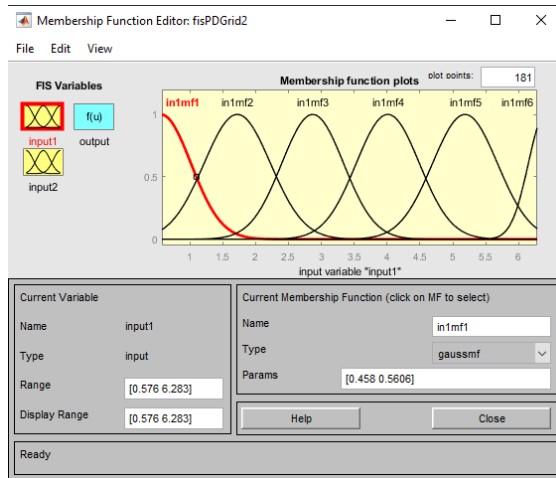


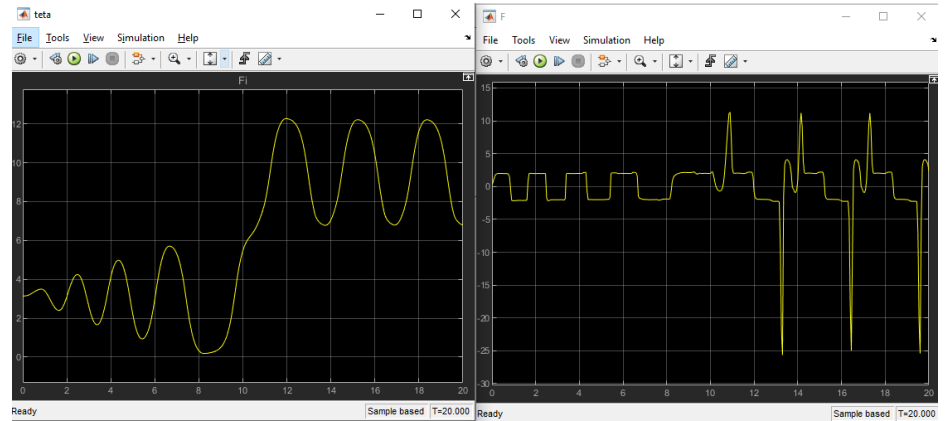
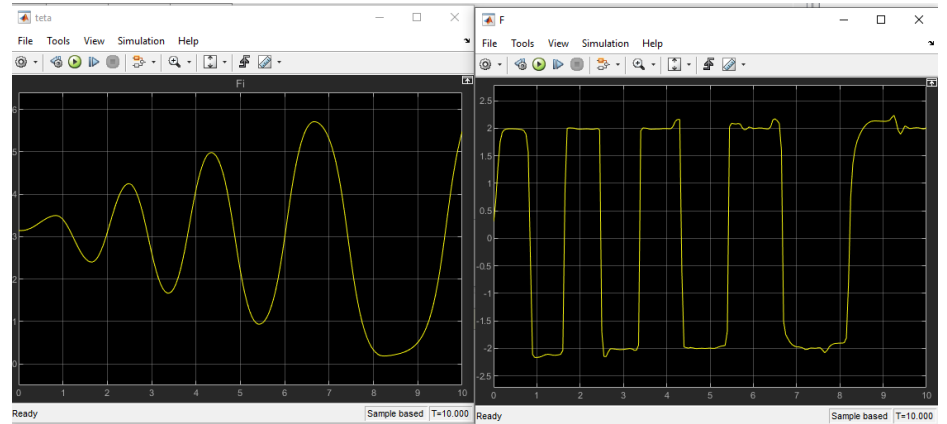
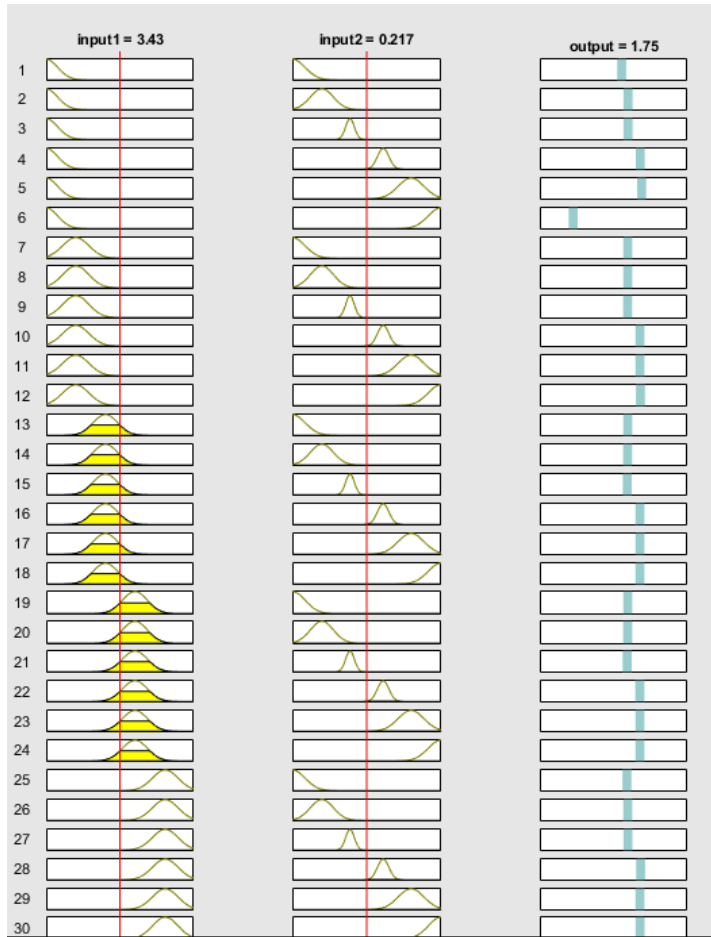
Grid partition



Grid partition

fisPDGrid2





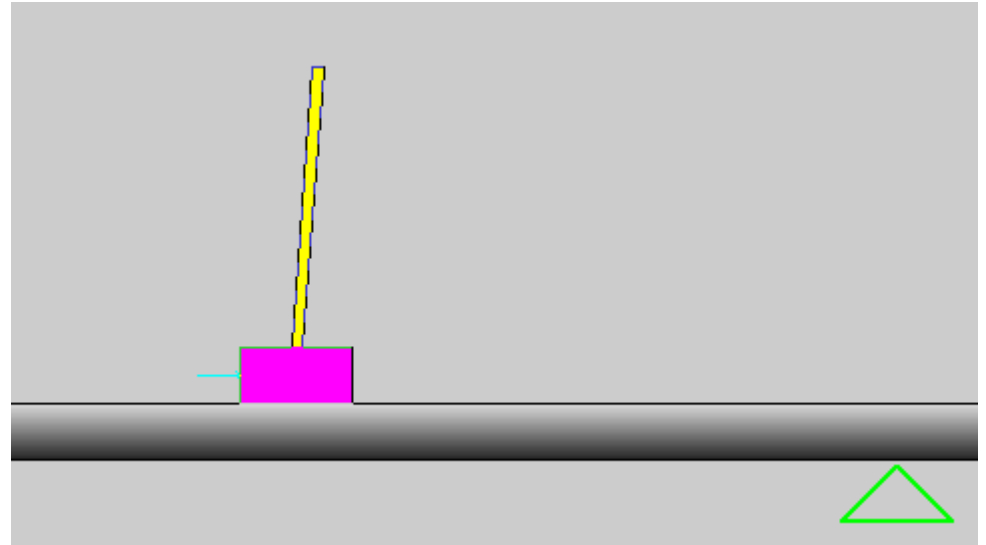
Wejście-wyjście model

- **Wejście**

- Kąt
- Prędkość kąta
- Pozycja
- Prędkość pozycji

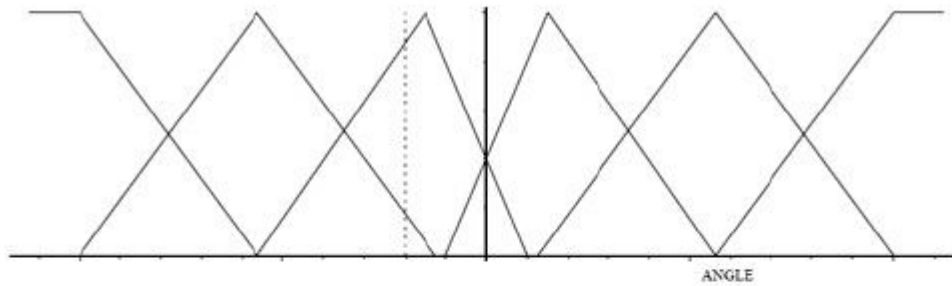
- **Wyjście**

- Siła

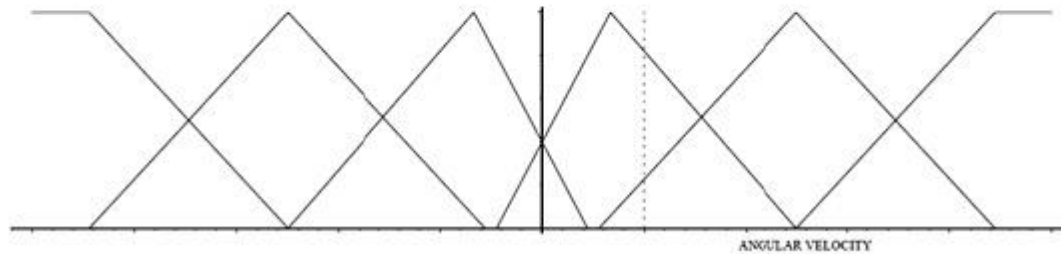


Kąt, prędkość kąta i siła

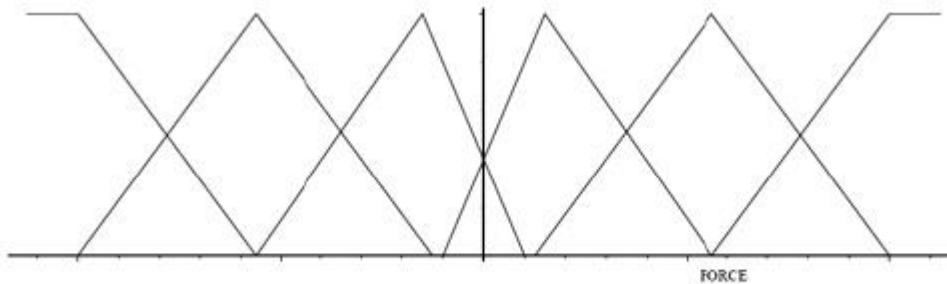
R_j : If θ is A_j and θ' is B_j then u is C_j



$A_1 = NB$ $A_2 = NM$ $A_3 = NS$ $A_4 = PS$ $A_5 = PM$ $A_6 = PB$

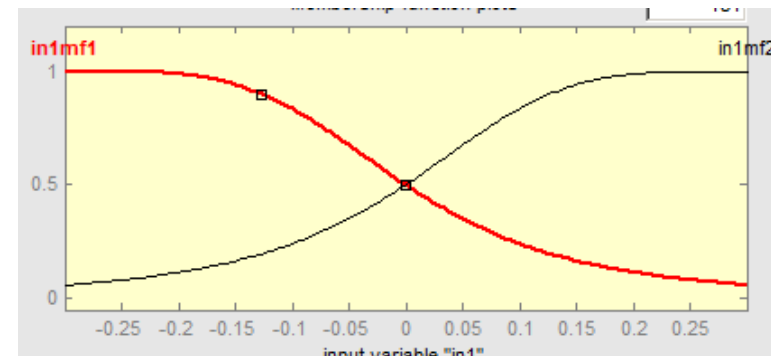
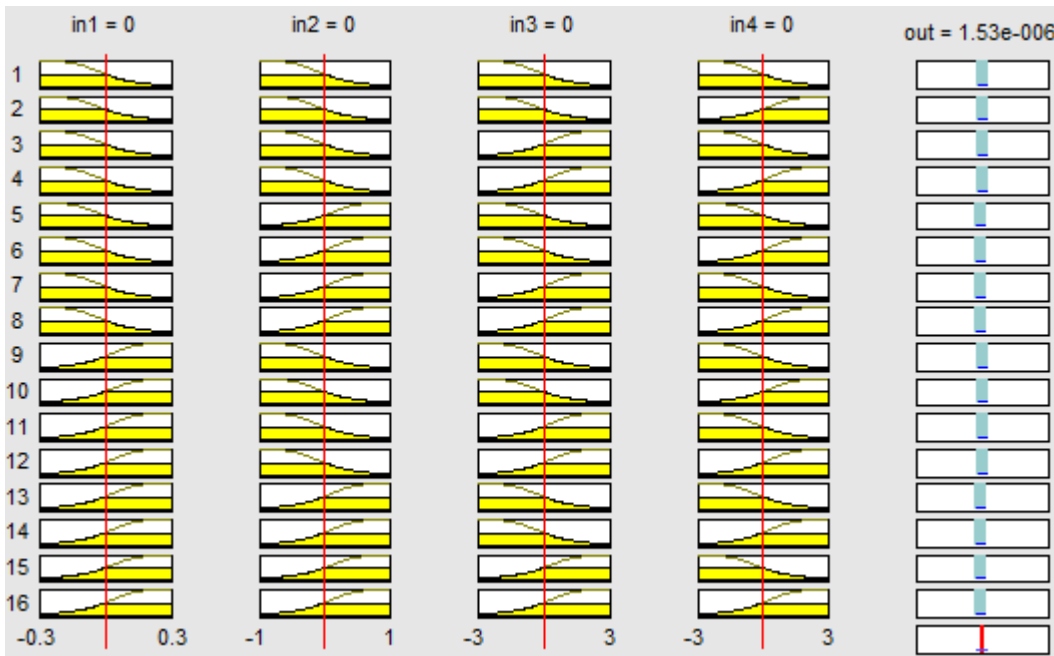
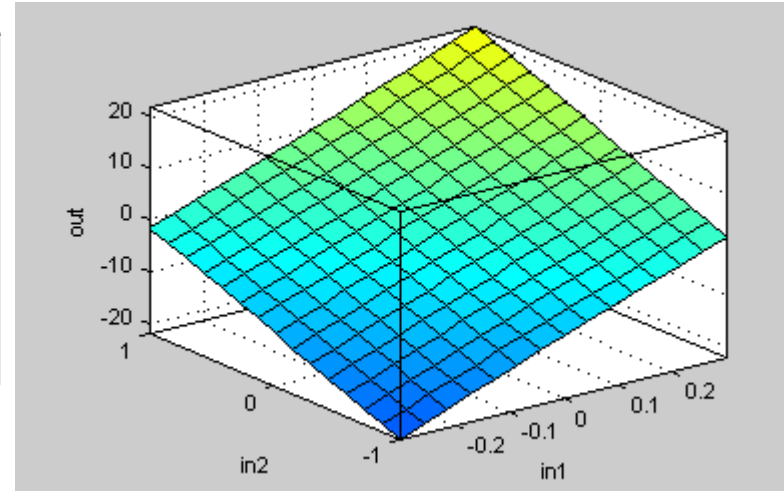
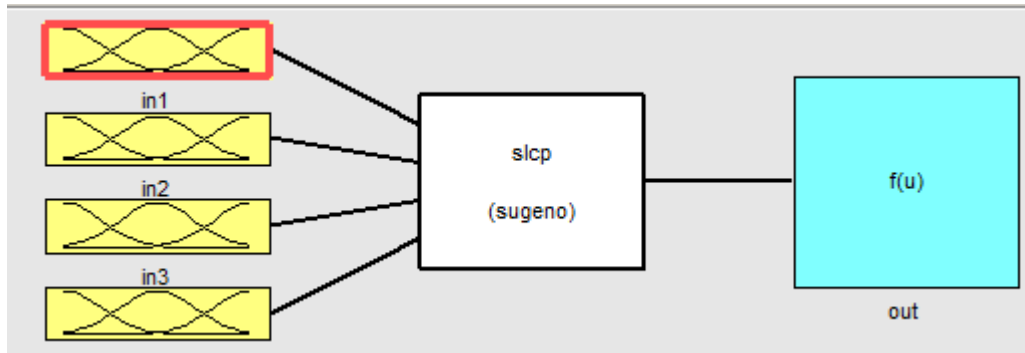


$B_1 = NB$ $B_2 = NM$ $B_3 = NS$ $B_4 = PS$ $B_5 = PM$ $B_6 = PB$



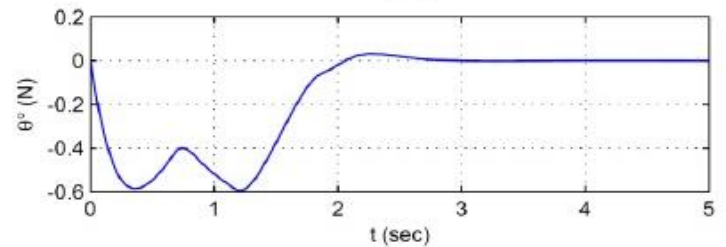
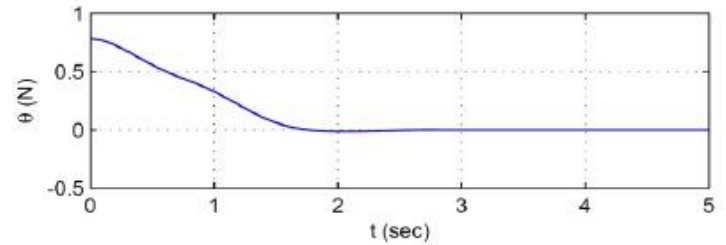
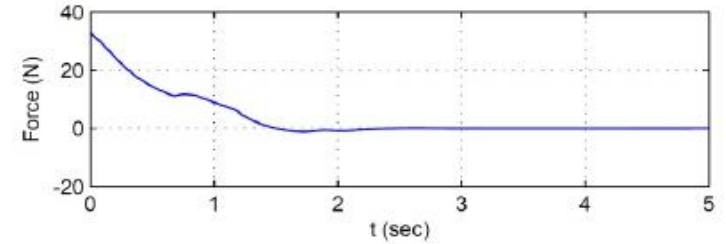
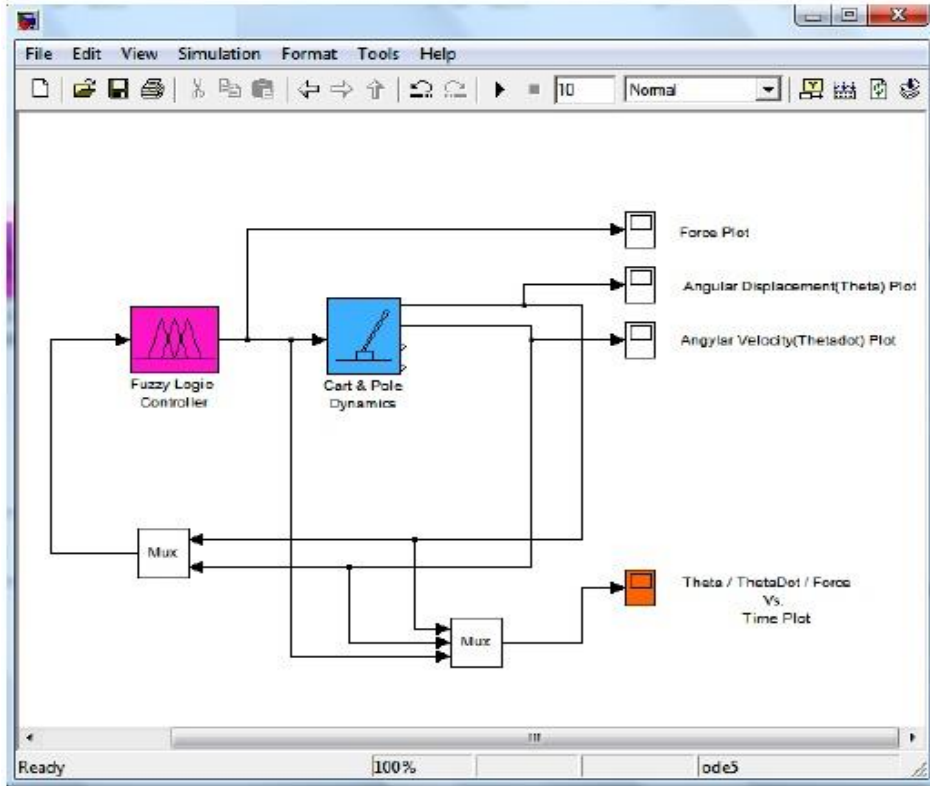
$C_1 = NB$ $C_2 = NM$ $C_3 = NS$ $C_4 = PS$ $C_5 = PM$ $C_6 = PB$

Baza reguł

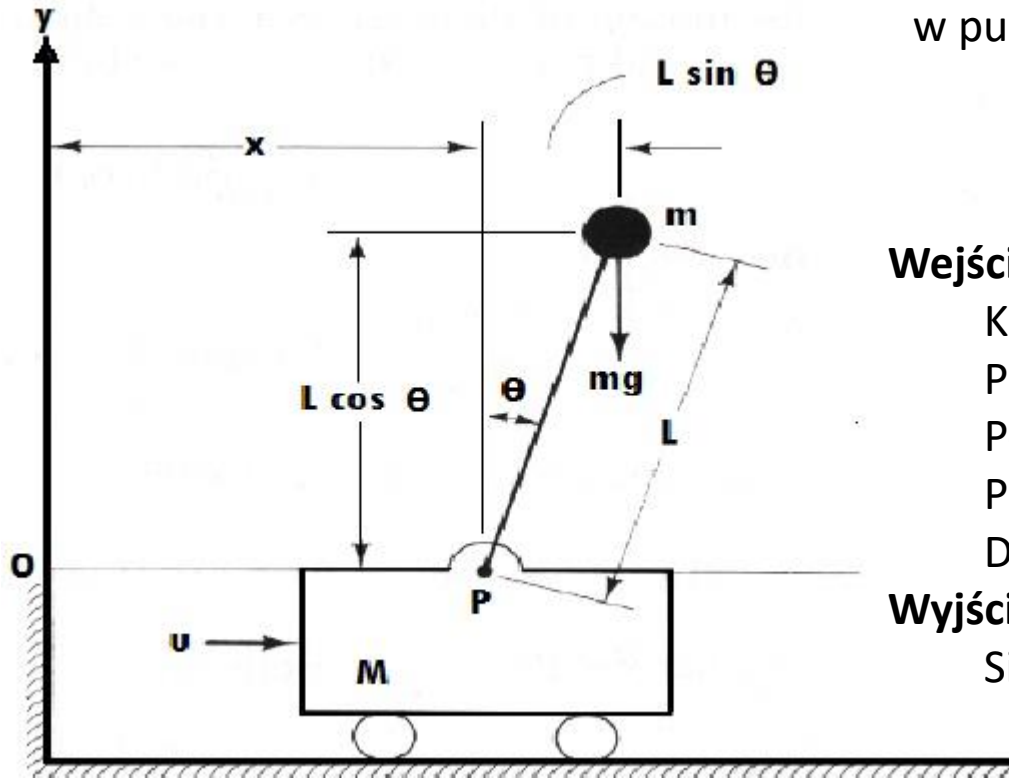


16 reguł Sugeno

Model i wyniki



Wahadło (slcp)



Przemieszczenie wahadła w punkt docelowy

Wejście

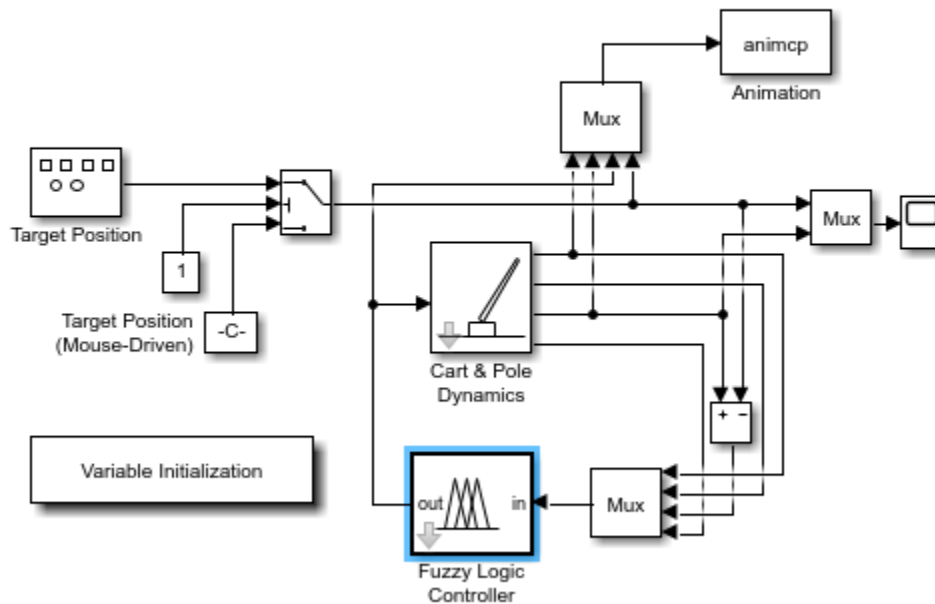
- Kąt
- Prędkość kąta
- Pozycja
- Prędkość pozycji
- Długość wahadła zmienna

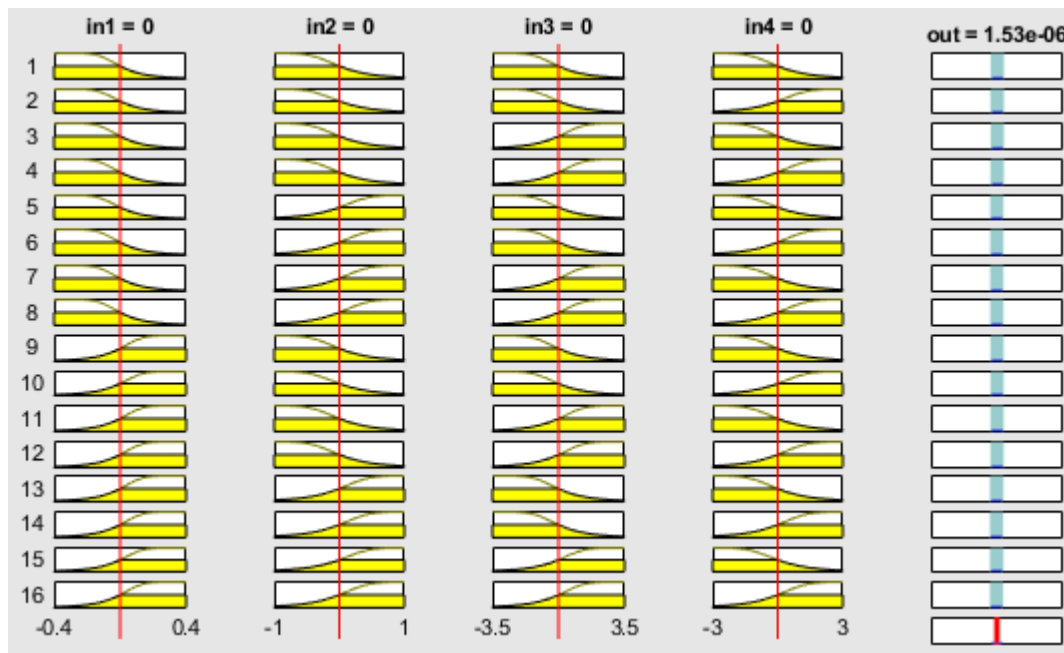
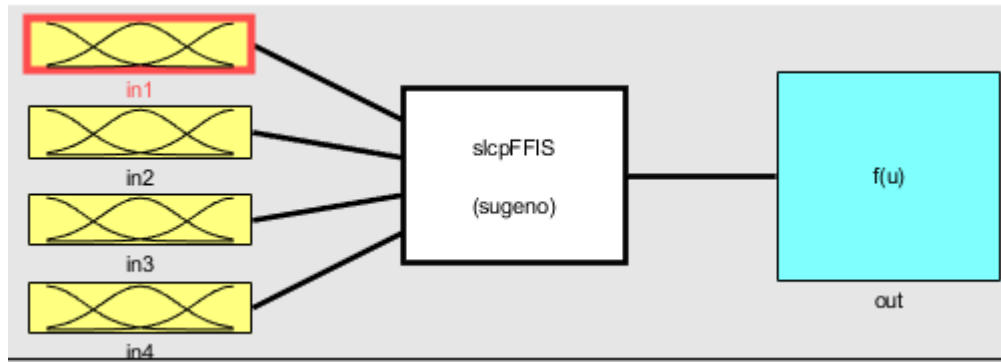
Wyjście

- Siła

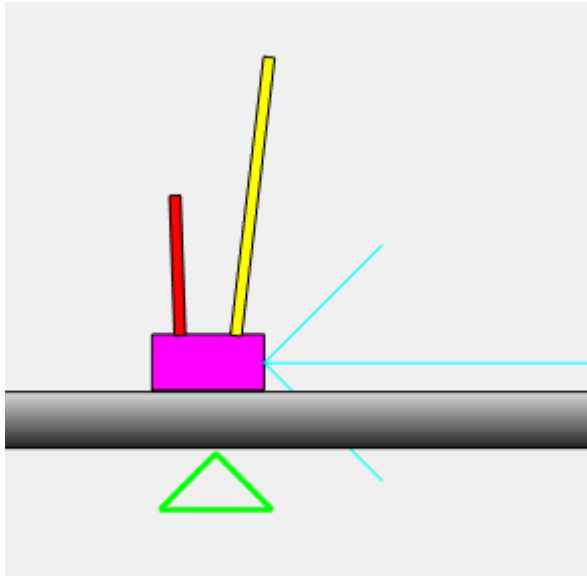
Cart and Pole System

Copyright (c) 2002-2018 The MathWorks, Inc.





Wahadło (slcpp1)



Przemieszczenie 2 wahadeł
w punk docelowy

Wejście

Kąt

Prędkość kąta

Pozycja

Prędkość pozycji

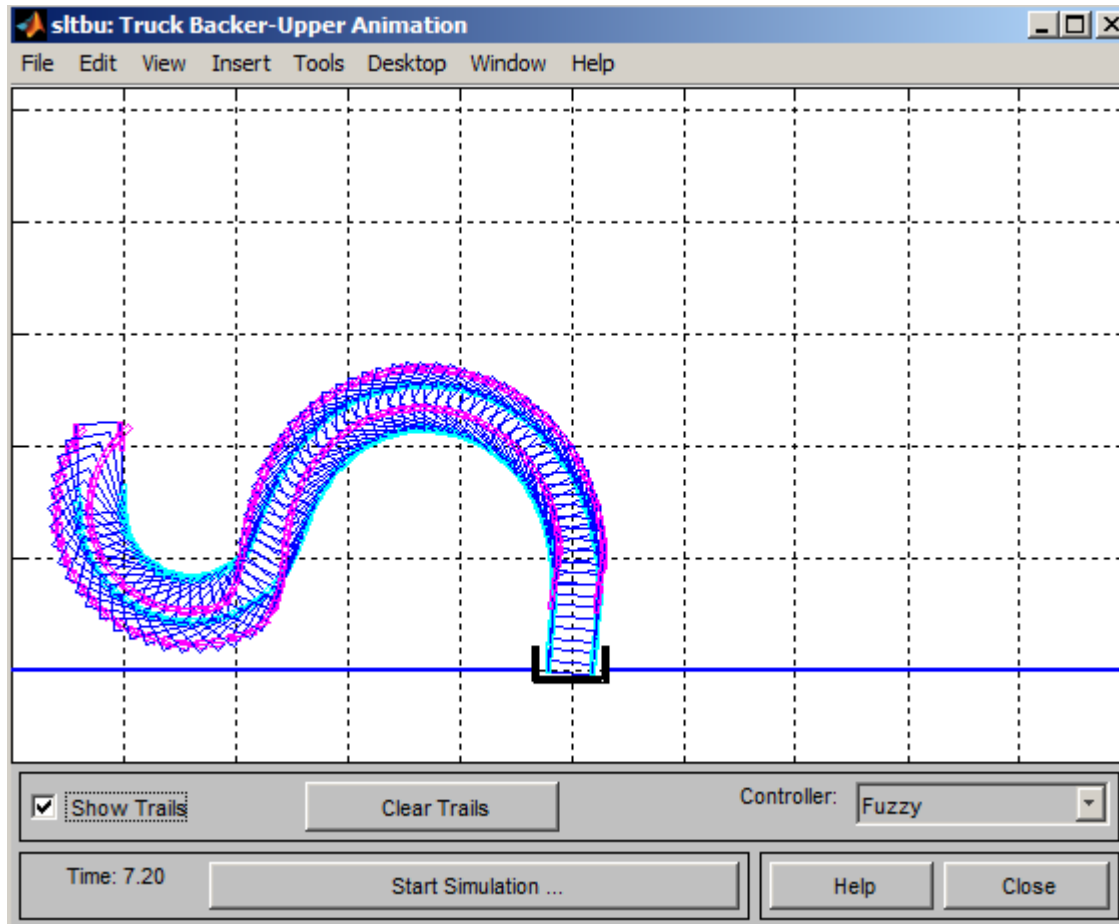
Długość wahadła 1 zmienna

Długość wahadła 2 zmienna

Wyjście

Siła

Parkowanie tyłem samochodu (sltbu)



Wybór jednego z dwóch algorytmów sterowania w zależności od odległości do punktu parkowania

Sterowanie parkowaniem

```
function out = ctrltbu1(x)
% CTRLTBU1 controller for the truck backer-upper when distance is far.

% Roger Jang, 10-21-93, 1-16-94, 11-7-94
% Copyright 1994-2002 The MathWorks, Inc.

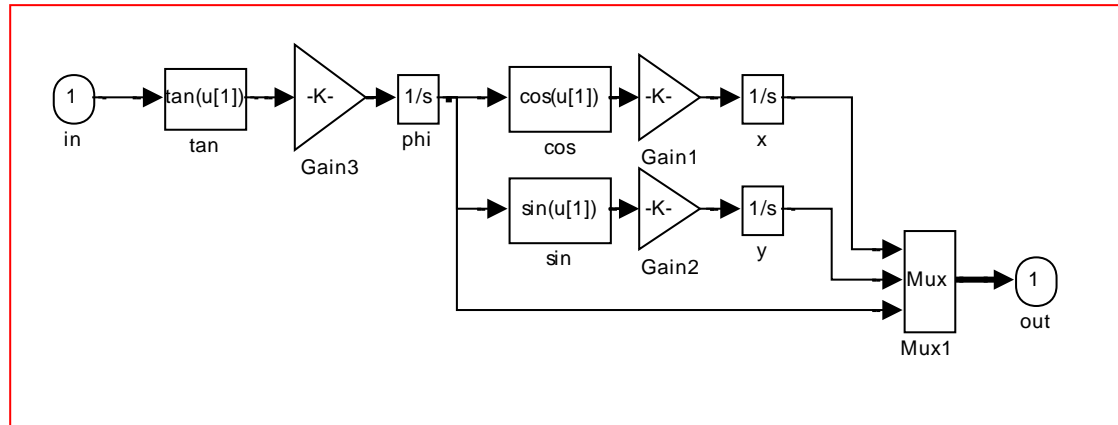
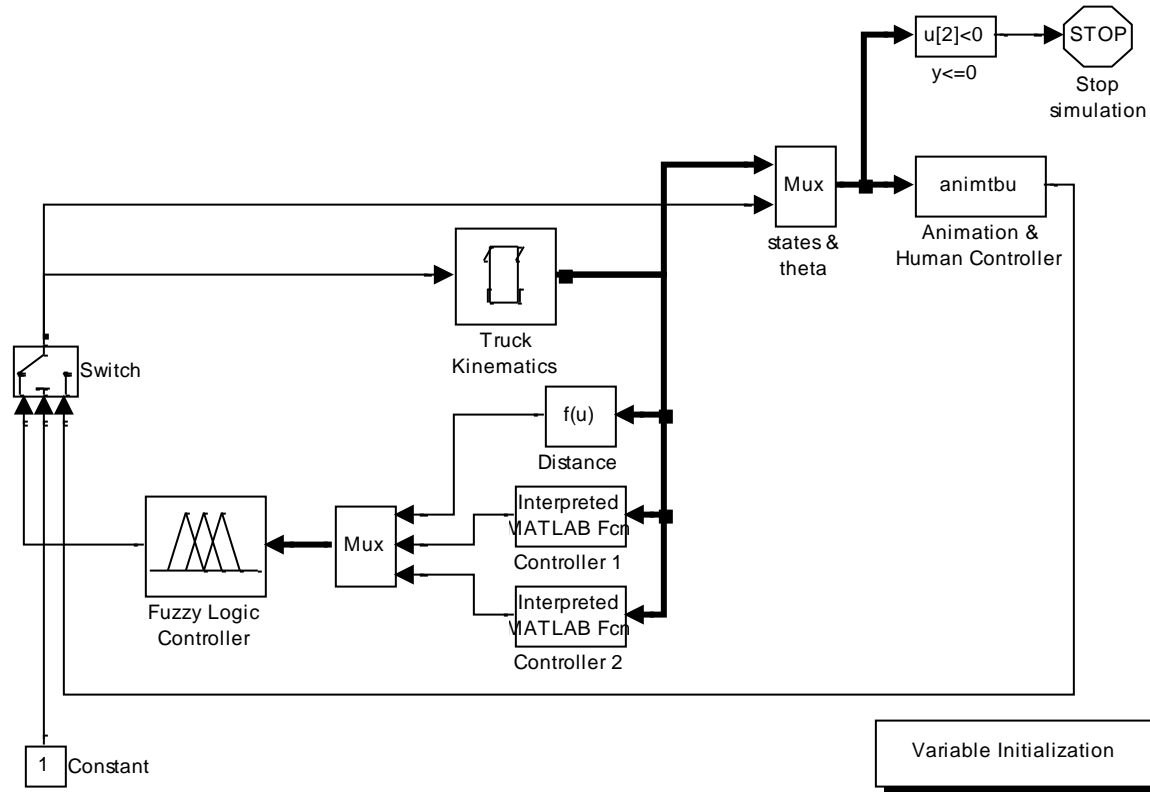
distance = norm(x(1:2));
alpha = acos(x(1)/distance) - pi/2; % abs(alpha) <= pi/2
tmp = x(3) - pi/2 - alpha;
beta = tmp - round(tmp/2/pi)*2*pi; % abs(beta) <= pi
out = - 3*alpha + 2*beta;
out = max(-pi/4, min(pi/4, out));
```

```
function out = ctrltbu2(x)
% CTRLTBU2 controller for the truck backer-upper when distance is near.

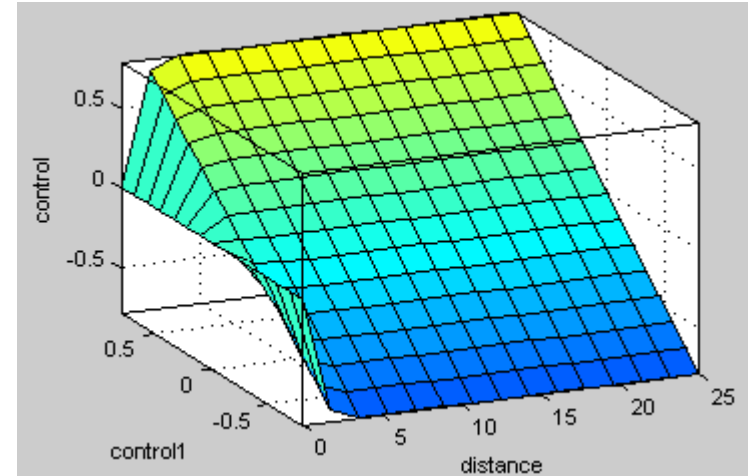
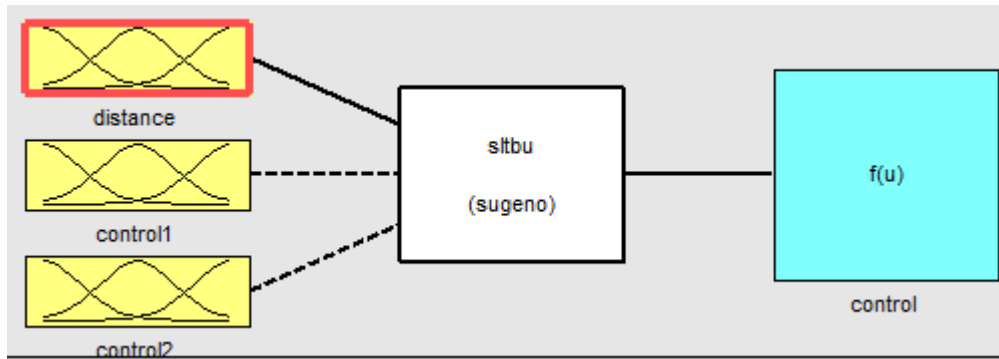
% Roger Jang, 10-21-93, 1-16-94, 11-7-94
% Copyright 1994-2002 The MathWorks, Inc.

tmp = (x(3) - pi/2) - round((x(3)-pi/2)/(2*pi))*(2*pi); % abs(tmp) < pi
out = x(1)/8 + tmp;
out = max(-pi/4, min(pi/4, out));
```

Model SS i samochodu



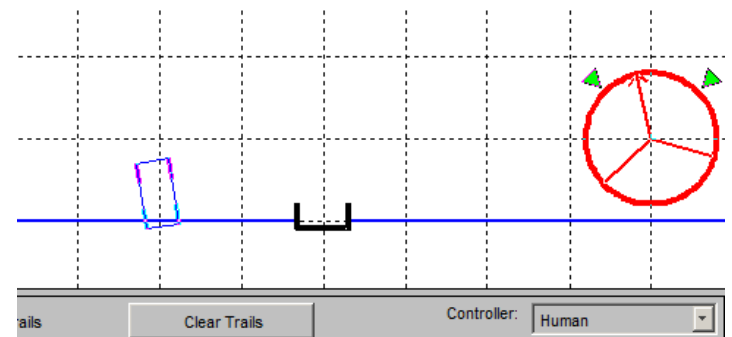
Sterownik



Rule Editor: sltbu

File Edit View Options

1. If (distance is near) then (control is out1mf1) (1)
2. If (distance is far) then (control is out1mf2) (1)



slbb

Stabilizacja piłki

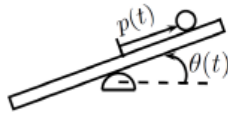


Figure 1: Ball and Beam System

The ball rolls on the beam without slipping under the action of the force of gravity. The beam is tilted from an external torque to control the position of the ball on the beam. We first define a set of generalized coordinates which fully describe the system. The generalized coordinates are defined as

$$q(t) = \begin{bmatrix} p(t) \\ \theta(t) \end{bmatrix} \quad (1)$$

where $p(t)$ is the position of the ball on the beam and $\theta(t)$ is the angle of the beam. The Lagrangian of a system is a quantity which is defined as (D'Souza, 1984)

$$L = K - U \quad (2)$$

where K is the kinetic energy and U is the potential energy of the system. To facilitate the evaluation of K and U , we define the Cartesian coordinates $x(t)$ and $y(t)$ as shown in Figure 2.

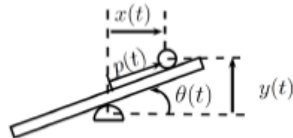


Figure 2: Cartesian Coordinates and Generalized Coordinates

The kinetic energy of the beam is

$$K_1 = \frac{1}{2} J \dot{\theta}^2 \quad (3)$$

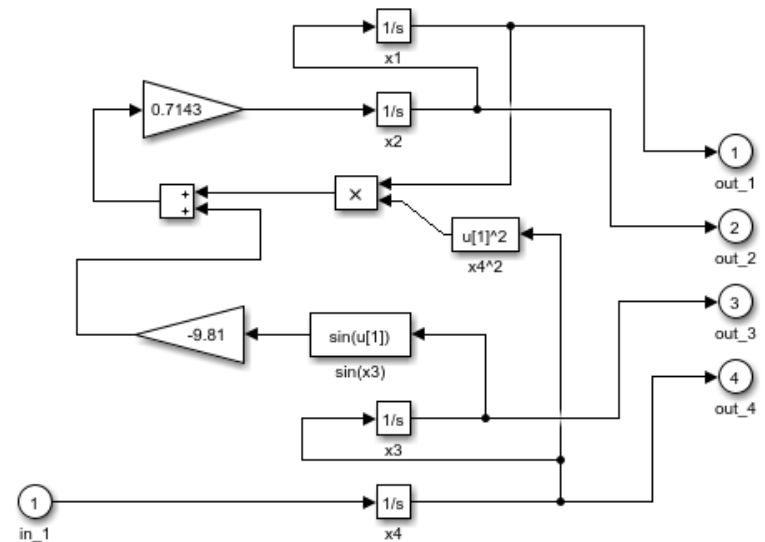
where J is the moment of inertia of the beam. The kinetic energy of the ball is

$$K_2 = \frac{1}{2} J_b \dot{\theta}_b^2 + \frac{1}{2} m v_b^2 \quad (4)$$

where $\dot{\theta}_b$ is the angular velocity of the ball and v_b is the linear velocity of the ball. The quantity $\dot{\theta}_b$ can be expressed in terms of the generalized coordinates as

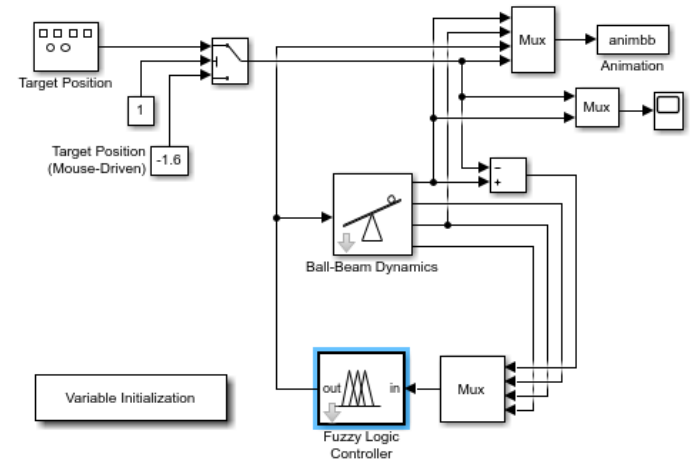
$$\dot{\theta}_b = \frac{p}{r} \quad (5)$$

where r is the radius of the ball. We can also express v_b in terms of the generalized coordinates.



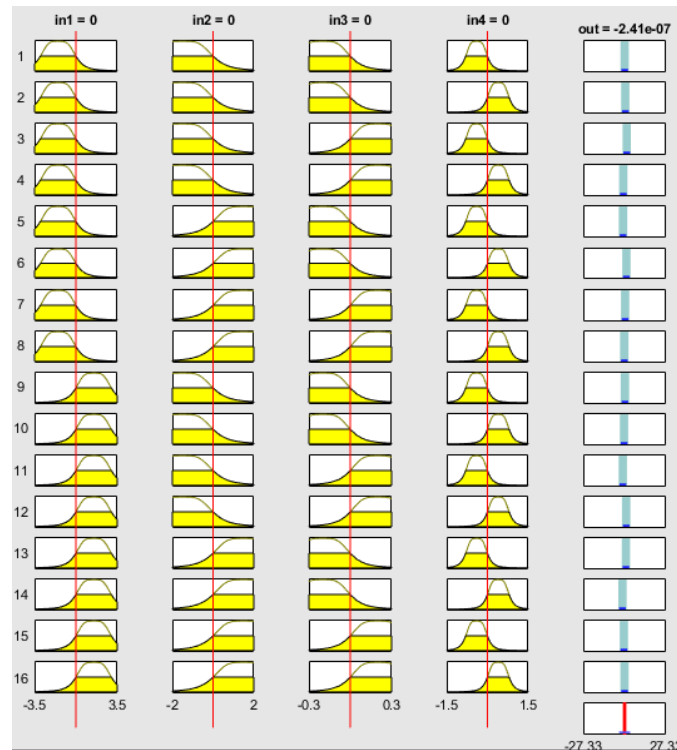
Ball and Beam System

Copyright (c) 2002-2018 The MathWorks, Inc.

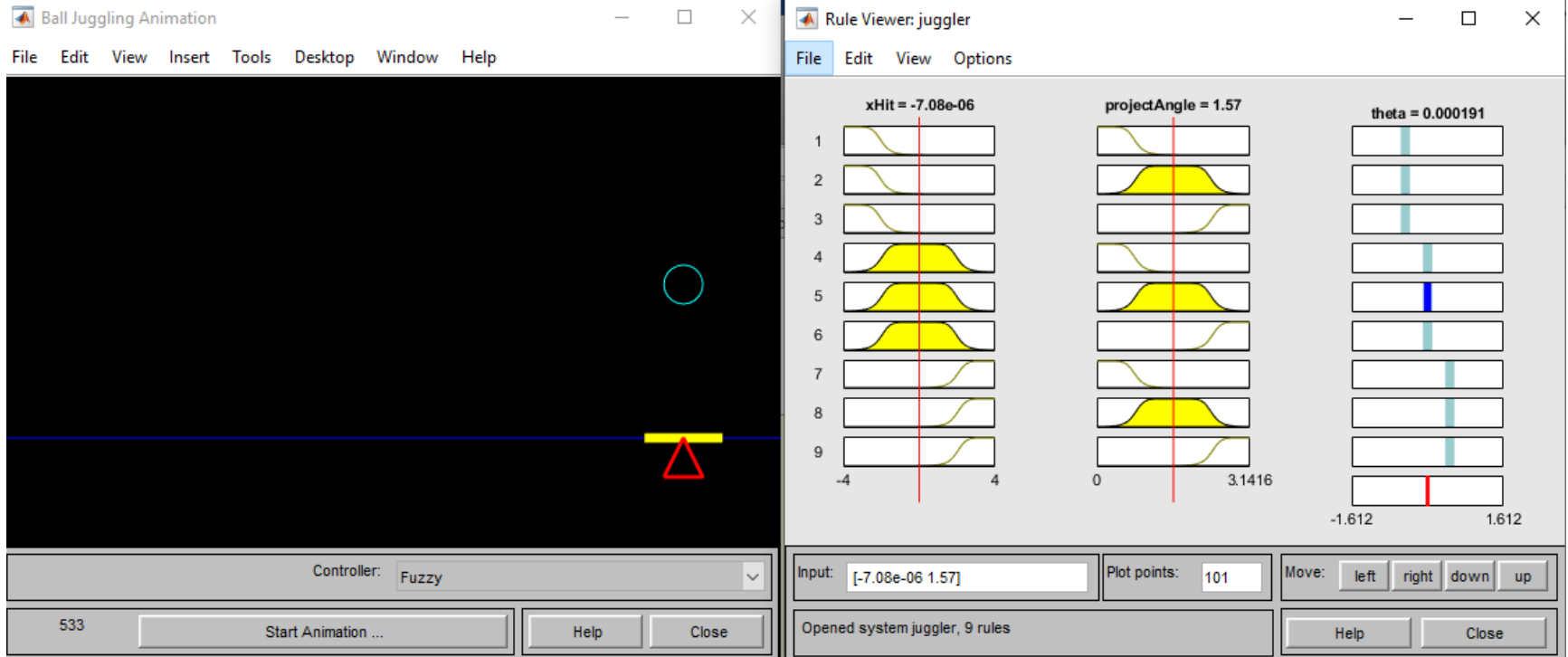


Ball and Beam (Simulink only)

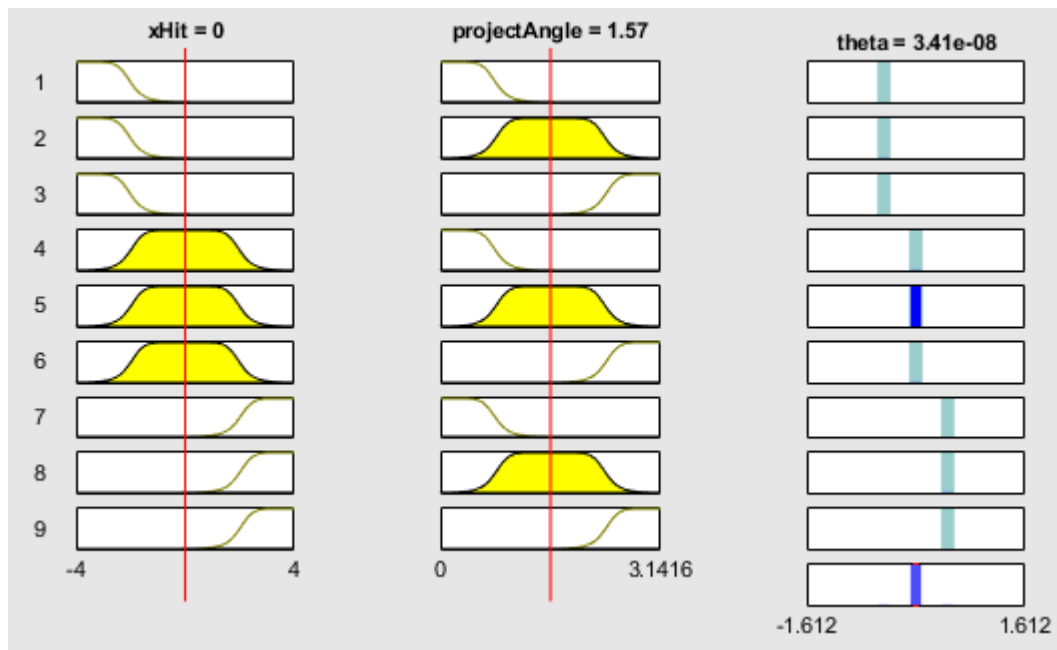
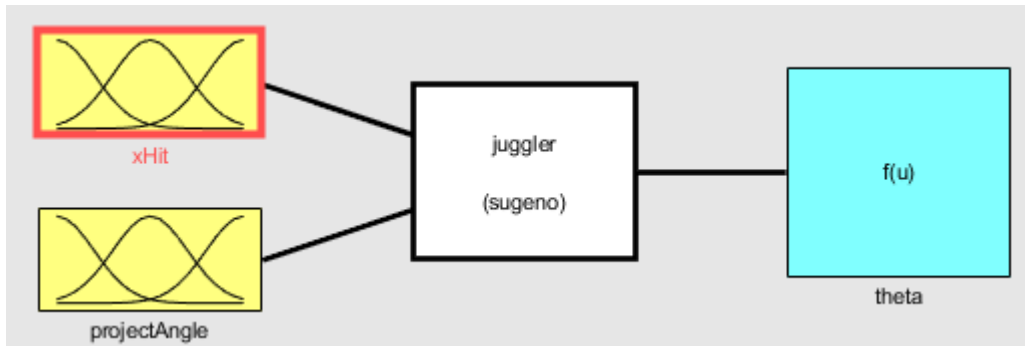
The ball-beam (BB) system consists of a ball rolling frictionlessly on a beam; a motor generates a torque to tilt the beam in order to send the ball to a desired location. A fuzzy controller is designed to generate an appropriate torque to achieve the control goal.



Juggler



juggler



Podstawowy algorytm generowania reguł rozmytych

Sterowanie rozmyte jest znakomita metoda pozwalająca korzystać **z opisowej wiedzy o problemie (model lingwistyczny)**.

Z drugiej strony podejścia bazujące na zastosowaniu **sieci neuronowych** w sterowaniu odwołują się do wiedzy reprezentowanej w postaci par (**wejście, wyjście**), gdzie *wejście to zbiór obserwowanych wartości wejściowych, a wyjście – odpowiadające im wartości zmiennych wyjściowych*.

A więc **sterowniki rozmyte korzystają z reguł lingwistycznych, a sterowniki neuronowe – z danych numerycznych**.

Chcemy utworzyć bazę reguł dla układu rozmytego o dwóch wejściach i jednym wyjściu. Dane uczące zadane są w postaci m trójek

$$\begin{aligned} &(x_1^{(1)}, x_2^{(1)}, u^{(1)}) \\ &(x_1^{(2)}, x_2^{(2)}, u^{(2)}) \\ &\dots \\ &(x_1^{(m)}, x_2^{(m)}, u^{(m)}) \end{aligned}$$

gdzie $x_1^{(i)}, x_2^{(i)}$ to wartości sygnałów wejściowych zarejestrowanych w i -tym przykładzie, a $u^{(i)}$ to odpowiadająca im wartość sygnału wyjściowego.

Podział wejściowej i wyjściowej przestrzeni na obszary

Dla każdego sygnału wejściowego i wyjściowego wyznaczamy jego dziedzinę $[x_{i,min}, x_{i,max}]$, $[u_{min}, u_{max}]$, $i = 1, 2$.

Każdy z wyznaczonych w ten sposób przedziałów dzielony jest na $2N + 1$ obszarów, oznaczających nośniki N zbiorów rozmytych. Liczba N może być różna dla każdego z sygnałów, a nośniki mogą być różnej długości. Z każdym zbiorem rozmytym kojarzymy etykietę oddającą intencję jego wprowadzenia. Przykładowe określenia to *NB* (*Negative Big*), *NM* (*Negative Medium*), *NS* (*Negative Small*), *Z*, *PS* (*Positive Small*), *PM* (*Positive Medium*), *PB* (*Positive Big*).

Tworzenie reguł rozmytych na podstawie przykładów

Przypuśćmy, że zakres zmienności sygnału x_1 pokryto zbiorami rozmytymi $A^j, j = 1, \dots, j_{max}$, sygnału x_2 – zbiorami $B^k, k = 1, \dots, k_{max}$, a dziedzinę sygnału wyjściowego pokryto zbiorami $C^l, l = 1, \dots, l_{max}$.

Dla każdego przykładu $(x_1^{(i)}, x_2^{(i)}, u^{(i)})$ wyznaczmy teraz stopnie przynależności do odpowiedniego obszaru: $\mu_{A^j}(x_1^{(i)}), \mu_{B^k}(x_2^{(i)}), \mu_{C^l}(u^{(i)})$. Ostatecznie konkretną wartość sygnału zaliczamy do obszaru o najwyższym stopniu przynależności. Jeżeli np. obserwacja $x_1^{(i)}$ należy do zbioru PB w stopniu 0.8, do zbioru PS w stopniu 0.3, a jej przynależność do pozostałych zbiorów rozmytych jest równa zeru, to zaobserwowaną wartość zaliczamy do zbioru PB .

Ogólnie, jeżeli obserwacja $x_1^{(i)}$ należy w maksymalnym stopniu do zbioru A^{j^*} , obserwacja $x_2^{(i)}$ należy w maksymalnym stopniu do zbioru B^{k^*} , a obserwacja $u^{(i)}$ należy w maksymalnym stopniu do zbioru C^{l^*} , to przykładowi $(x_1^{(i)}, x_2^{(i)}, u^{(i)})$ odpowiada reguła

if (x_1 *is* A^{j^*}) **and** (x_2 *is* B^{k^*}) **then** (u *is* C^{l^*})

Przyporządkowanie stopni prawdziwości regułom

W przypadku, gdy reguły o identycznych przesłankach mają różne konkluzje – mamy do czynienia z sytuacją konfliktową, a reguły o takiej własności tworzą zbiór sprzecznych reguł.

Najprostsza metoda rozwiązywania konfliktów polega na przypisaniu każdej regule (ze zbioru sprzecznych reguł) tzw. stopnia prawdziwości i wybrania tylko tej reguły, która ma najwyższy taki stopień. W ten sposób nie tylko rozwiązujemy konflikt, ale i redukujemy liczbę reguł. Dla reguły R stopień prawdziwości τ_R obliczamy następująco:

$$\tau_R = \mu_{A^{j*}}(x_1) \cdot \mu_{B^{k*}}(x_2) \cdot \mu_{C^{l*}}(u)$$

W praktyce posiadamy pewną *aprioryczną* wiedzę o rozwiązywanym problemie. Np. parze danych $(x_1^{(i)}, x_2^{(i)}, u^{(i)})$ można przypisać subiektywny stopień $\beta^{(i)}$ charakteryzujący jej wiarygodność, typowość czy też użyteczność w procesie formowania bazy wiedzy. W takim przypadku stopień prawdziwości przyznany tworzonej regule będzie miał postać

$$\tau_R = \mu_{A^{j*}}(x_1^{(i)}) \cdot \mu_{B^{k*}}(x_2^{(i)}) \cdot \mu_{C^{l*}}(u^{(i)})\beta^{(i)}$$

Utworzenie bazy rozmytych reguł

Bazę reguł (wiedzy) stanowi tablica tworzona w sposób następujący: tworzymy tablicę, której boki odpowiadają wartościom rozmytym sygnałów wejściowych. Dla reguły $\text{if } (x_1 \text{ is } A^{i*}) \text{ and } (x_2 \text{ is } B^{j*}) \text{ then } (u \text{ is } C^k)$ rozmytą wartość sygnału wyjściowego umieszczamy na przecięciu wiersza odpowiadającego rozmytej wartości A^{i*} sygnału x_1 i kolumny odpowiadającej rozmytej wartości B^{j*} sygnału x_2 . W ten sposób otrzymujemy tablicę. Reprezentuje ona regułę $\text{if } (x_1 \text{ is } NS) \text{ and } (x_2 \text{ is } PS) \text{ then } (u \text{ is } PB)$.

<i>PB</i>					
<i>PM</i>					
<i>PS</i>		<i>PB</i>			
<i>Z</i>					
<i>NS</i>					
<i>NM</i>					
<i>NB</i>					
	<i>NM</i>	<i>NS</i>	<i>Z</i>	<i>PM</i>	<i>PB</i>

Wyostrzanie

W oparciu o uzyskaną tablicę reguł poszukujemy teraz odwzorowania $g: (x_1, x_2) \rightarrow \bar{u}$, gdzie \bar{u} jest wartością wyjściową zwracaną przez sterownik.

Niech

$$\alpha^{(k)} = \mu_{A^{(k)}}(x_1) \cdot \mu_{B^{(k)}}(x_2)$$

oznacza stopień aktywności k -tej reguły.

Poszukiwana wartość \bar{u} wyznaczamy korzystając z jednej z metod wyostrzania, np. stosując metodę *center average defuzzification* otrzymujemy

$$\bar{u} = \frac{\sum_{k=1}^N \alpha^{(k)} \bar{u}^{(k)}}{\sum_{k=1}^N \alpha^{(k)}}$$

gdzie $\bar{u}^{(k)}$ jest punktem, w którym funkcja przynależności występująca w konkluzji k -tej reguły osiąga swoje maksimum.

W przypadku, gdy przesłanki reguł zawierają n przesłanek elementarnych, w których występują zbiory $A_1^{(k)}, \dots, A_n^{(k)}$, stopień aktywności k -tej reguły jest równy

$$\alpha^{(k)} = \prod_{i=1}^n \mu_{A_i^{(k)}}(\bar{x}_i)$$

Metody alternatywne. Generowanie uproszczonych reguł

$R^{(k)}$: if $[(x_1 \text{ is } A_1^{(k)}) \text{ and } \dots \text{ and } (x_n \text{ is } A_n^{(k)})]$ then $(u \text{ is } b^{(k)})$

gdzie $b^{(k)} \in \mathbb{R}$ jest pewną liczbą.

Wnioskowanie z użyciem prostych reguł rozmytych

Niech $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)^T$ będzie wektorem obserwacji. Odpowiadającą mu wartość $u(\bar{x})$ wyznaczamy z równania

$$u(\bar{x}) = \frac{\sum_{k=1}^N \prod_{j=1}^n \mu_{A_j^{(k)}}(\bar{x}_j) b^{(k)}}{\sum_{k=1}^N \prod_{j=1}^n \mu_{A_j^{(k)}}(\bar{x}_j)}$$

Tutaj, jak poprzednio, iloczyn $\prod_{j=1}^n \mu_{A_j^{(k)}}(\bar{x}_j)$ określa stopień zgodności obserwacji \bar{x} z przesłankami k -tej reguły. Jest to wariant wyostrzania metodą środków ciężkości.

Identyfikacja prostych reguł rozmytych

Zakładamy, że dane mają postać

$$\begin{aligned} & (x_1^{(1)}, x_2^{(1)}, u^{(1)}) \\ & (x_1^{(2)}, x_2^{(2)}, u^{(2)}) \\ & \dots \\ & (x_1^{(m)}, x_2^{(m)}, u^{(m)}) \end{aligned}$$

Zakładamy też, że dokonano odpowiedniego pokrycia dziedzin zmiennych X_1, \dots, X_n zbiorami rozmytymi. Jeżeli A_i^k oznacza k_i -ty rozmyty podzbiór dziedziny zmiennej X_i to przez

$$w_{k_1 \dots k_n}(\bar{x}) = \left(\prod_{j=1}^n \mu_{A_j^{k_j}}(\bar{x}_j) \right)^\alpha$$

oznacza stopień zgodności obserwacji \bar{x} z rozmytym obszarem wyznaczonym przez wybrana rodzinę podzbiorów. Zauważmy, że z równania tego wynika, że jeżeli dziedzinę zmiennej X_i pokryto K_i zbiorami rozmytymi, to dane uczące pozwalają wygenerować co najwyżej $K = K_1 \dots K_n$ reguł rozmytych. Jeżeli istotnie wygenerowano K reguł, mówimy wówczas o *kompletnej* bazie wiedzy.

Występujący w tym równaniu wykładnik $\alpha \in (0, 1)$ steruje stopniem dopasowania wynikowych reguł do danych trenujących.

Identyfikacja prostych reguł rozmytych

Dysponując wagami regule o indeksie $k = k_1 \cdots k_n$ przypisuje się następującą numeryczną konkluzję:

$$b_{k_1 \dots k_n} = \frac{\sum_{j=1}^m w_{k_1 \dots k_n}(\bar{\mathbf{x}}_j) \cdot u_j}{\sum_{j=1}^m w_{k_1 \dots k_n}(\bar{\mathbf{x}}_j)}$$

tzn. $b_{k_1 \dots k_n}$ jest ważoną sumą m obserwacji zmiennej zależnej.

Oczywiście, jeżeli $w_{k_1 \dots k_n} = 0$ oznacza to niemożność wygenerowania reguły dla wybranej kombinacji zbiorów rozmytych etykietujących elementarne przesłanki. Jest to sygnał wskazujący na konieczność redefinicji rozmytych pokryć.

Decydując się na wybór konkretnego pokrycia, jak również na wybór wartości α można kierować się wartościami wskaźnika dopasowania

$$e = \sqrt{\sum_{j=1}^m \left(u(\bar{\mathbf{x}}_j) - u_j \right)^2}$$

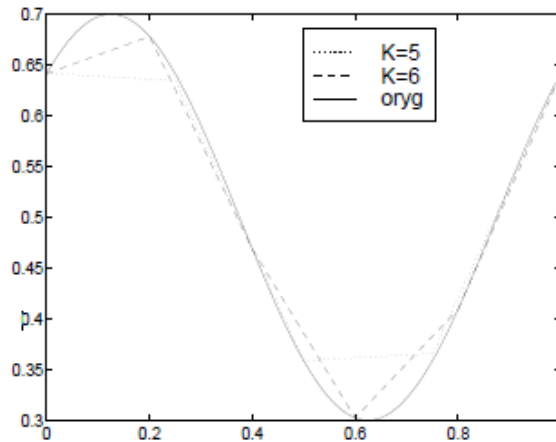
Przykład

Celem ilustracji omówionej procedury rozważmy problem aproksymacji statycznego nieliniowego systemu opisanego równaniem

$$u = 0.2 \sin(2\pi x + \pi/4) + 0.5, 0 \leq x \leq 1$$

Zbiór trenujący utworzono wybierając 11 punktów o odciętych $x_{i+1} = x_i + 0.1$, $x_0 = 0.0$. Rozważono też dwa alternatywne pokrycia zbioru $[0, 1]$. Pierwsze to pięć trójkątnych funkcji przynależności postaci $\Delta_1(-\infty, 0.0, 0.25)$, $\Delta_2(0.0, 0.25, 0.50)$, $\Delta_3(0.25, 0.50, 0.75)$, $\Delta_4(0.50, 0.75, 1.0)$, $\Delta_5(0.75, 1.0, \infty)$. Drugie pokrycie obejmuje sześć trójkątnych funkcji przynależności postaci $\Delta_1(-\infty, 0.0, 0.2)$, $\Delta_2(0.0, 0.2, 0.4)$, $\Delta_3(0.2, 0.4, 0.6)$, $\Delta_4(0.4, 0.6, 0.8)$, $\Delta_5(0.6, 0.8, 1.0)$, $\Delta_6(0.8, 1.0, \infty)$.

Porównanie jakości aproksymacji zadanej funkcji (przedstawionej linia ciągła) dla dwóch różnych pokryć dziedziny $[0, 1]$.



W pierwszym przypadku jakość rekonstrukcji oryginalnej funkcji mierzona błędem średniokwadratowym jest równa $e = 0.003268$, a w drugim $e = 0.001972$. Zrekonstruowane krzywe przedstawiono na rysunku. Jeżeli w danych występują zakłócenia, np. *outliery*, to sensowny wybór wartości α pozwala zredukować ich wpływ na jakość rekonstrukcji.

Konwersja prostych reguł rozmytych

Przypuśćmy, że dany jest zestaw prostych reguł, a na dziedzinie zmiennej u określono rodzinę rozmytych podzbiorów $\mathbb{B} = \{B_1, \dots, B_N\}$. Niech $B_{j^*}^{(k)} \in \mathbb{B}$ będzie zbiorem rozmytym o najwyższym stopniu zgodności z konkluzją $b^{(k)}$ tej reguły. Podobnie, niech $B_{j^{**}}^{(k)} \in \mathbb{B} \setminus \{B_{j^*}^{(k)}\}$ będzie zbiorem rozmytym o najwyższym (w zbiorze $\mathbb{B} \setminus \{B_{j^*}^{(k)}\}$) stopniu zgodności z konkluzją $b^{(k)}$ tejże reguły. Wówczas regułę można zastąpić dwiema regułami lingwistycznymi

$$R^{(k*)}: \quad \text{if } [(x_1 \text{ is } A_1^{(k)}) \text{ and } \dots \text{ and } (x_n \text{ is } A_n^{(k)})] \text{ then } (u \text{ is } B_{j^*}^{(k)})$$
$$R^{(k^{**})}: \quad \text{if } [(x_1 \text{ is } A_1^{(k)}) \text{ and } \dots \text{ and } (x_n \text{ is } A_n^{(k)})] \text{ then } (u \text{ is } B_{j^{**}}^{(k)})$$

Konwersja prostych reguł rozmytych

W szczególnym przypadku, gdy stosujemy zbiory rozmyte o trójkątnych funkcjach przynależności przecinających się parami na wysokości $\mu = 0.5$ zauważamy co następuje:

- dowolna konkluzja $b^{(k)}$ jest pokrywana przez dwa sąsiadujące ze sobą zbiory $B_{j^*}^{(k)}$ oraz $B_{j^{**}}^{(k)}$;
- $\mu_{B_{j^*}^{(k)}}(u) + \mu_{B_{j^{**}}^{(k)}}(u) = 1$, $u \in [\hat{B}_{j^*}^{(k)}, \hat{B}_{j^{**}}^{(k)}]$, przy czym $\hat{B}_{j^*}^{(k)}$ oznacza środek zbioru $B_{j^*}^{(k)}$, a ponadto $B_{j^*}^{(k)} < B_{j^{**}}^{(k)}$;
- dowolną wartość $u \in [\hat{B}_{j^*}^{(k)}, \hat{B}_{j^{**}}^{(k)}]$ można przedstawić w postaci sumy ważonej $u = \mu_{B_{j^*}^{(k)}}(u) \cdot \hat{B}_{j^*}^{(k)} + \mu_{B_{j^{**}}^{(k)}}(u) \cdot \hat{B}_{j^{**}}^{(k)}$, tzn. dowolną wartość $u \in [\hat{B}_{j^*}^{(k)}, \hat{B}_{j^{**}}^{(k)}]$ można przedstawić jako kombinację wypukłą zbiorów B_{j^*} oraz $B_{j^{**}}$.

Ta ostatnia własność oznacza, że istotnie liczbę $b^{(k)}$ można przedstawić lingwistycznie za pomocą dwóch zbiorów rozmytych.

Funkcje Matlab

```
a=newfis('tipper');  
a=addvar(a,'input','service',[0 10]);  
getfis(a,'input',1)
```

```
ruleList=[  
1 1 1 1 1  
1 2 2 1 1];  
a = addrule(a,ruleList);
```

```
a = newfis('tipper');  
a = addvar(a,'input','service',[0 10]);  
a = addmf(a,'input',1,'poor','gaussmf',[1.5 0]);  
a = addmf(a,'input',1,'good','gaussmf',[1.5 5]);  
a = addmf(a,'input',1,'excellent','gaussmf',[1.5  
10]);
```

opis

addrule ma dwa argumenty. Pierwszym argumentem jest nazwa zmiennej MATLAB FIS. Drugim jest macierz z jednego lub więcej rzędów, z których każdy reprezentuje daną regułę. Jeśli istnieje m wejść do systemu i n wyjść, musi być dokładnie $m + n + 2$ kolumny na liście reguł.

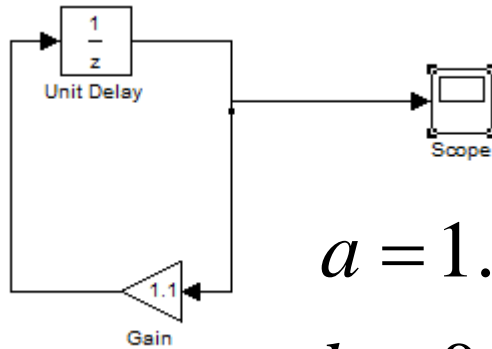
Pierwsze m kolumny to jest wejść układu. Każda kolumna zawiera numer, który odnosi się do indeksu o funkcji przynależności do tej zmiennej.

Kolejne kolumny dotyczą n wyjść. Każda kolumna zawiera numer, który odnosi się do indeksu o funkcji przynależności do tej zmiennej.

$m + n + 1$ kolumna zawiera wagę, która ma być zastosowana. Waga musi być liczbą pomiędzy zero a jeden i na ogół pozostaje jako jeden.

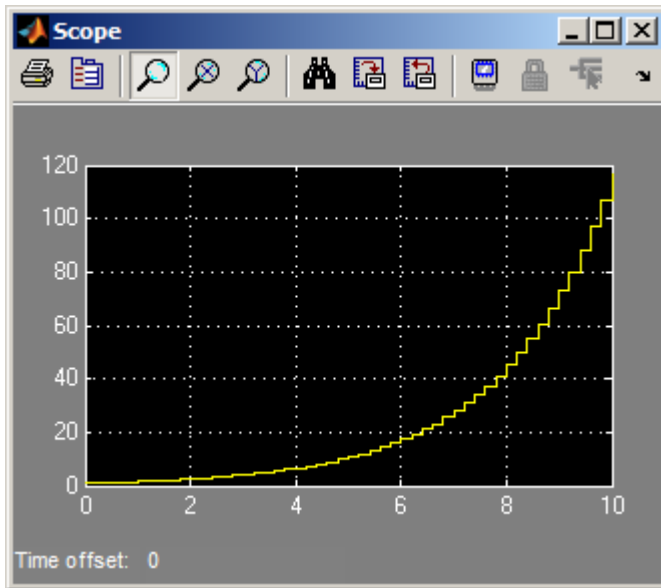
$m + n + 2$ kolumna zawiera 1 jeśli operator rozmytej dla rzędów w poprzedniku jest AND. Zawiera 2, jeśli operator jest OR.

Przykład



$$a = 1.1$$

$$b = 0$$



$$x_{k+1} = a \cdot x_k + b \cdot u_k$$

$$\varepsilon = g - x,$$

$$g = 0,$$

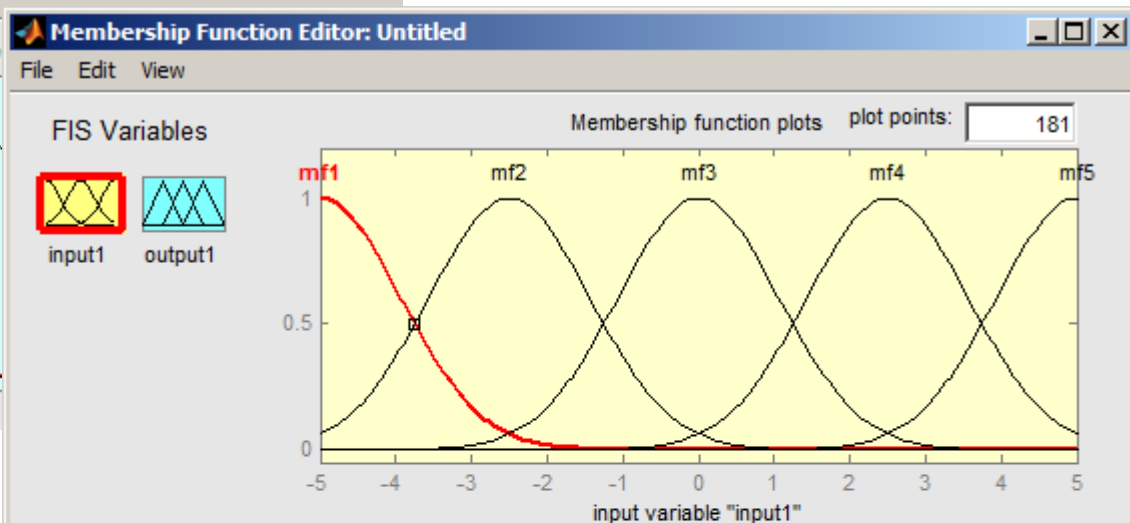
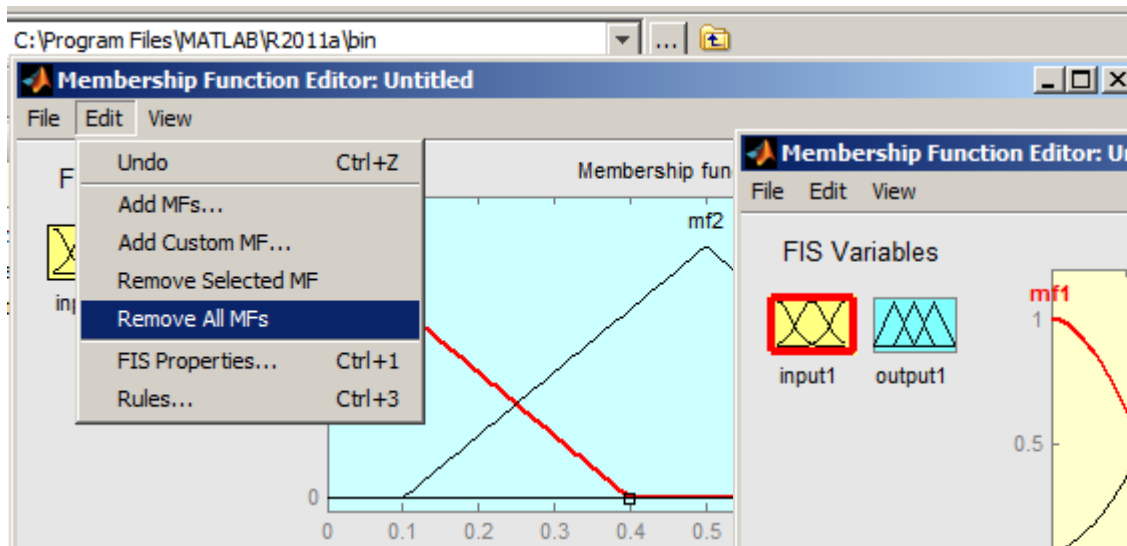
$$\varepsilon_{k+1} = -a \cdot \varepsilon_k + b \cdot u_k,$$

$$\varepsilon_{k+1} = 0 \Rightarrow u_k = \frac{a}{b} \varepsilon_k,$$

$$u_k = -\frac{a}{b} x_k,$$

Tworzenie modelu rozmytego

$$x \in [-5, 5], u \in [-7, 7]$$



Membership Functions

Add membership functions

MF type: gaussmf

Number of MFs: 5

OK Cancel

Current Variable

Name: input1

Type: input

Range: [-5 5]

Display Range: [-5 5]

Current Membership Function (click on MF to select)

Name: mf1

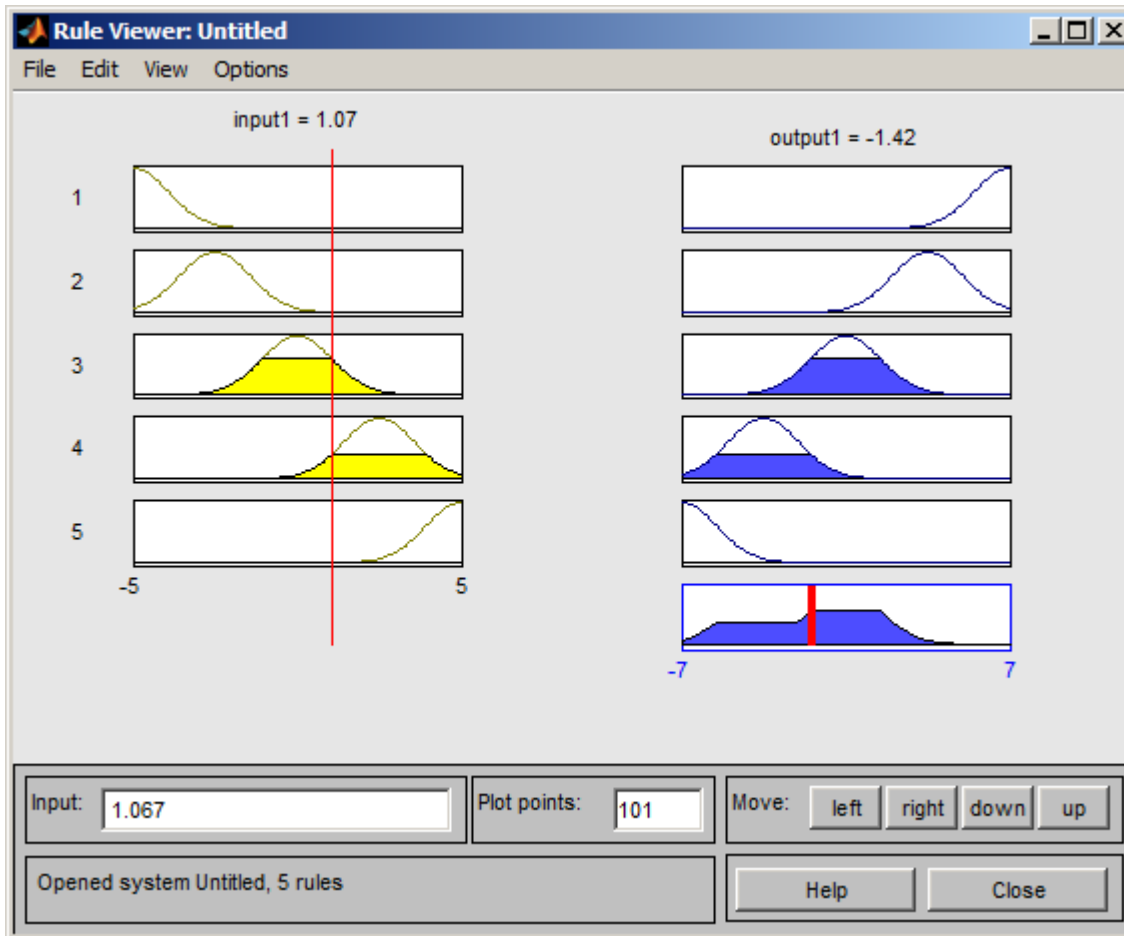
Type: gaussmf

Params: [1.062 -5]

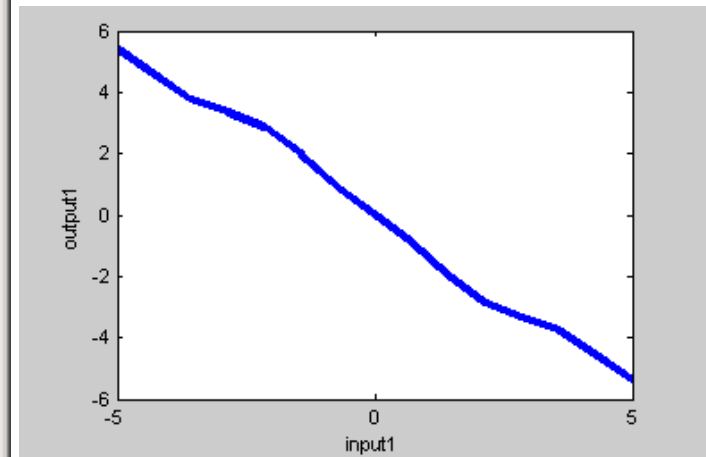
Help Close

Ready

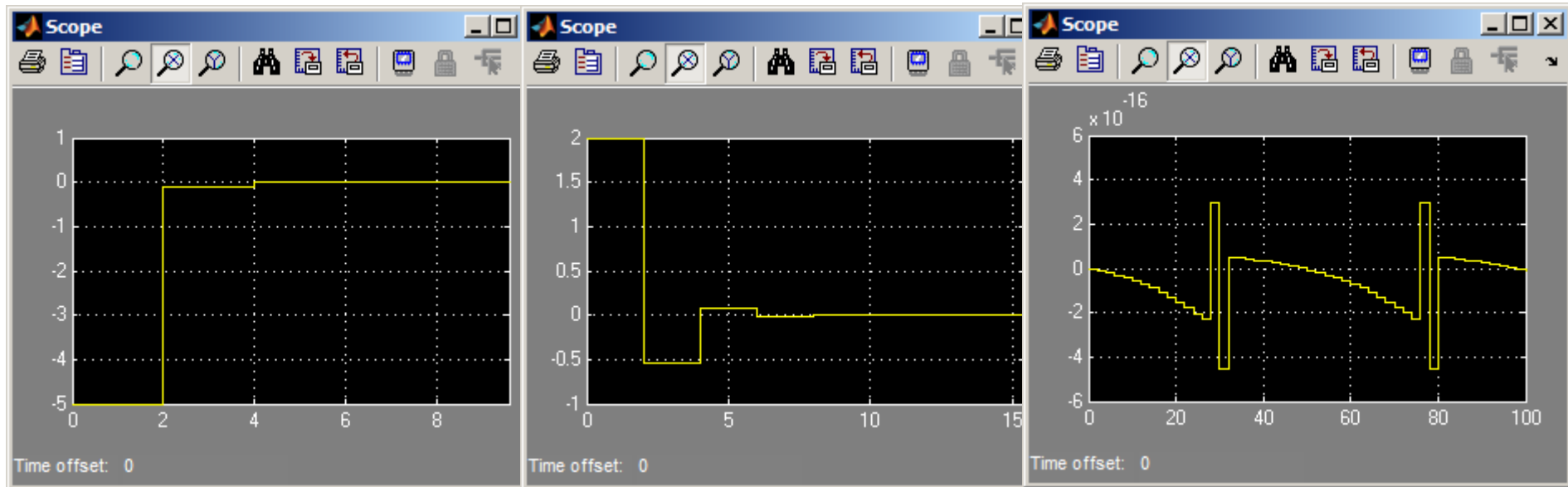
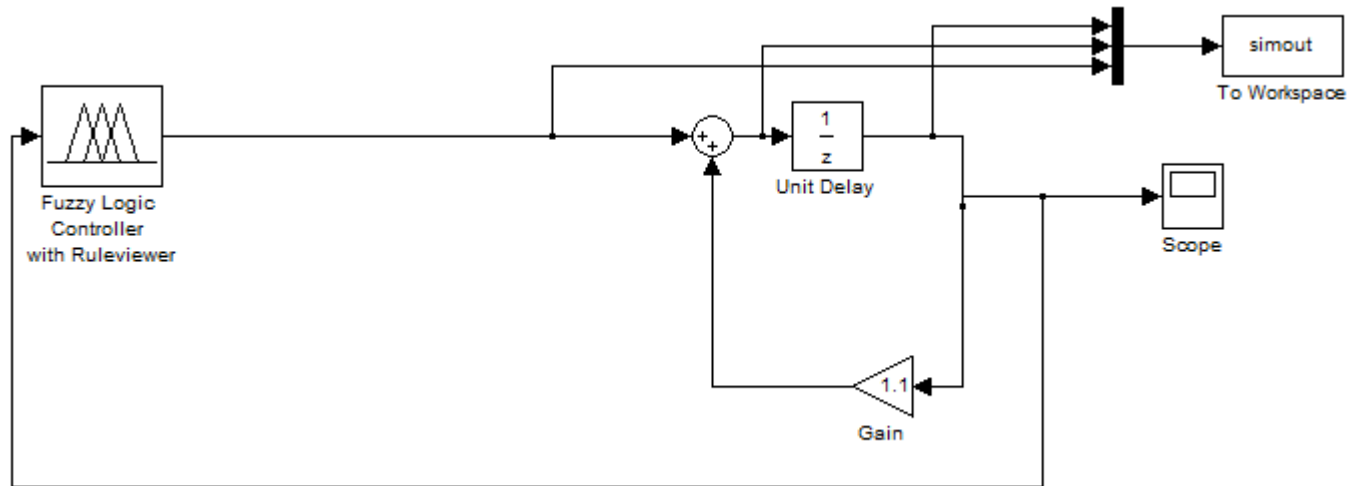
Tworzenie reguł



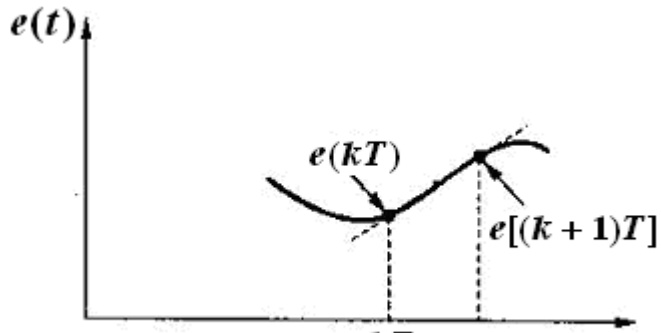
1. If (input1 is mf1) then (output1 is mf5) (1)
2. If (input1 is mf2) then (output1 is mf4) (1)
3. If (input1 is mf3) then (output1 is mf3) (1)
4. If (input1 is mf4) then (output1 is mf2) (1)
5. If (input1 is mf5) then (output1 is mf1) (1)



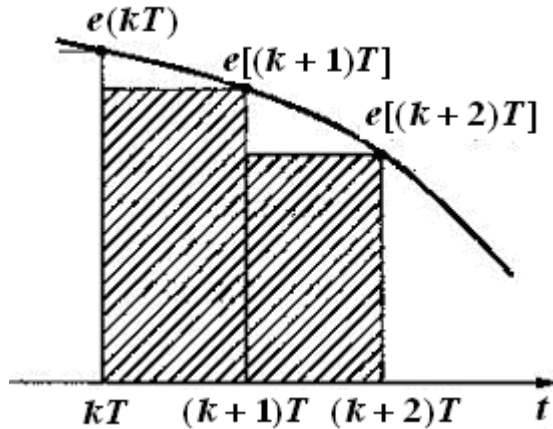
Sterownik rozmyty



PID



$$u[(k+1)T] = \frac{e[(k+1)T] - e(kT)}{T}$$

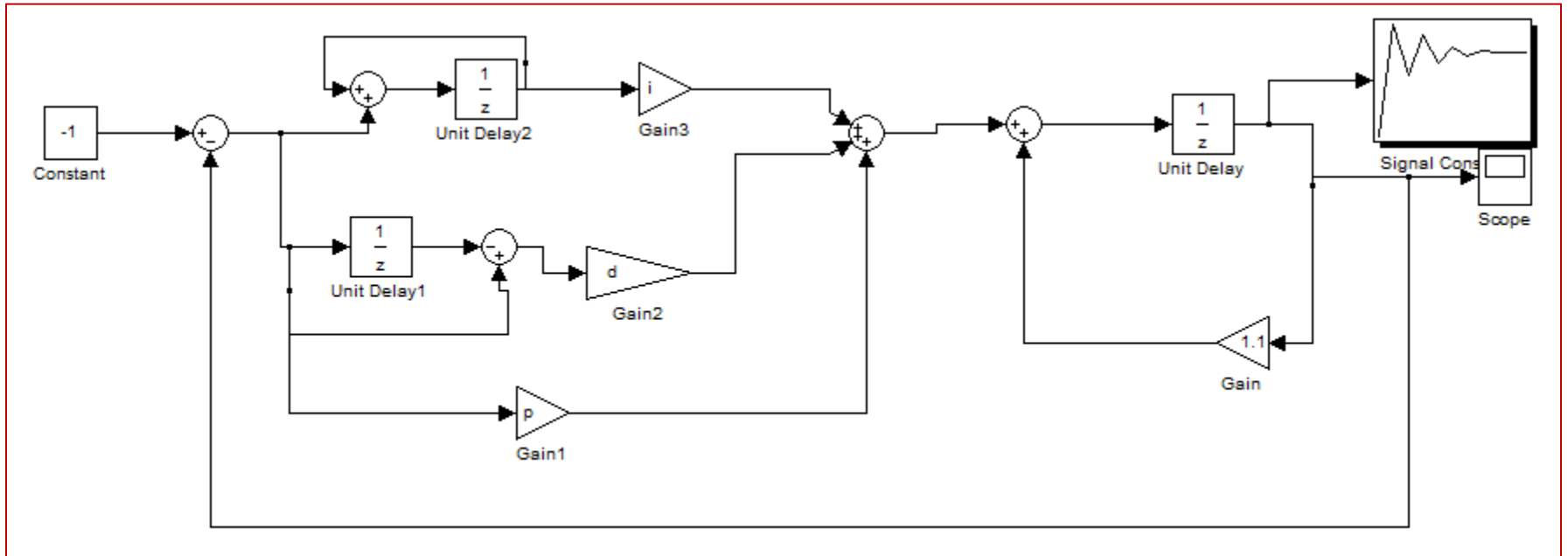


$$u[(k+1)T] = u(kT) + Te[(k+1)T].$$

$$u(k+1) = K_1 e(k+1) + K_2 u_i(k+1) + \frac{K_3 [e(k+1) - e(k)]}{T}$$

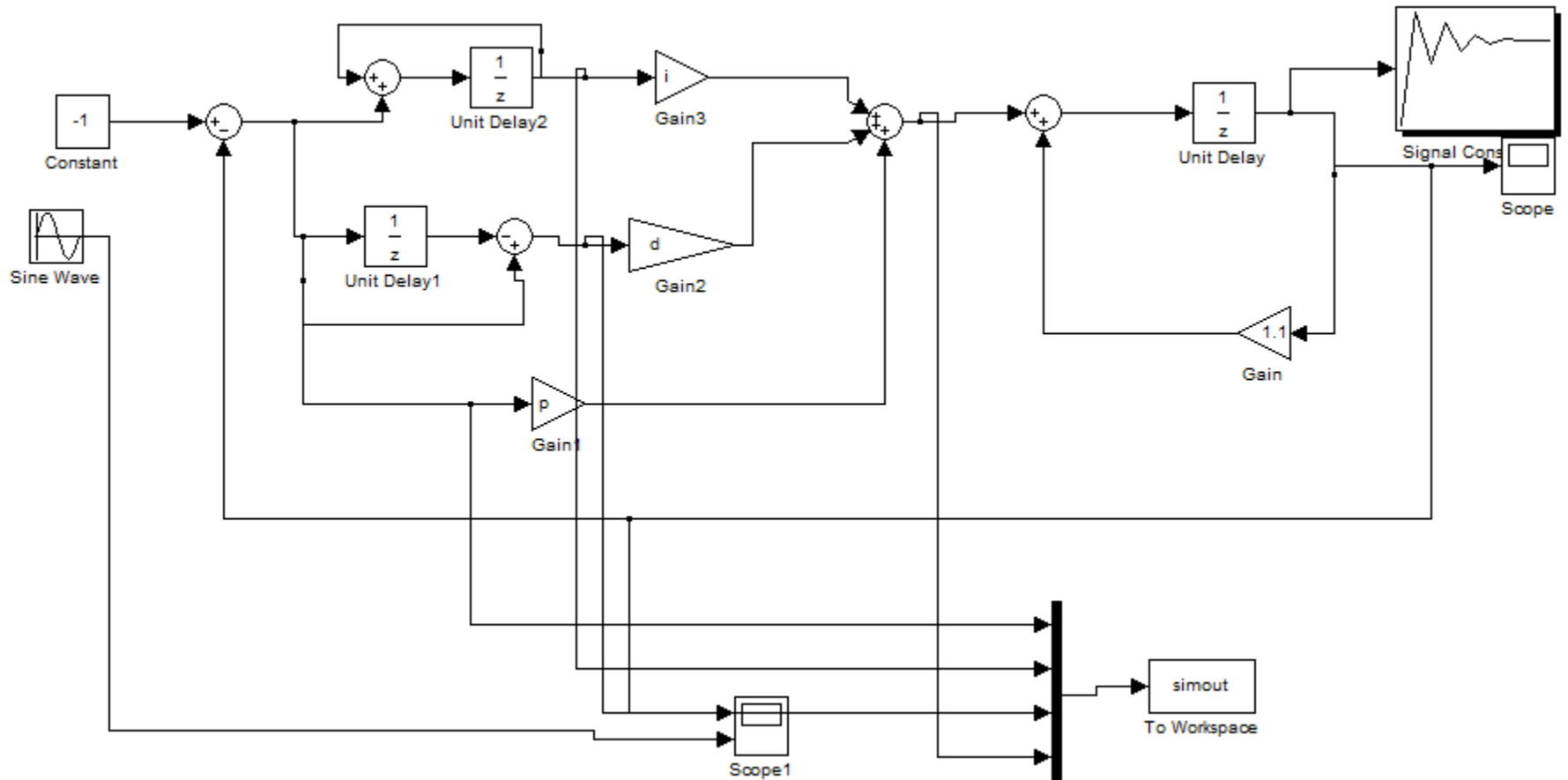
$$u_i(k+1) = u(k) + Te(k+1).$$

PID

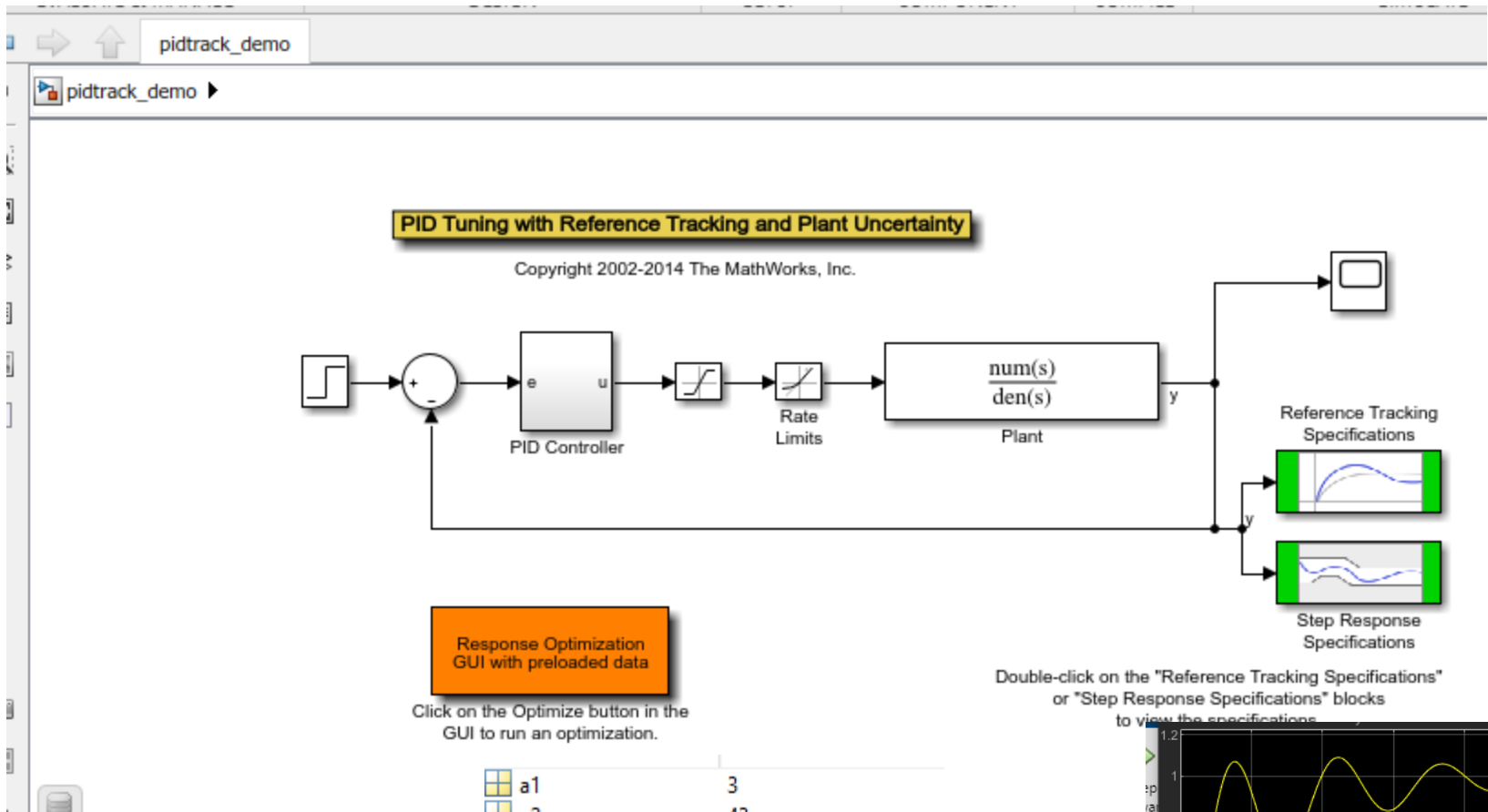


```
>> [i p d]
ans =
    0.7937    1.7601   -0.0828
```

Eksport danych

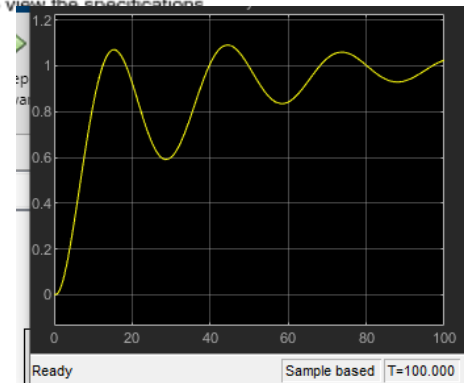


open_system('pidtrack_demo')



Double-click on the "Reference Tracking Specifications" or "Step Response Specifications" blocks to view the specifications.

a1	3
a2	43
Kd	1.9688
Ki	0.0504
Kp	0.6300
pidtrack_optim	1x1 SimProjectForm
t	150x1 double
y	150x1 double



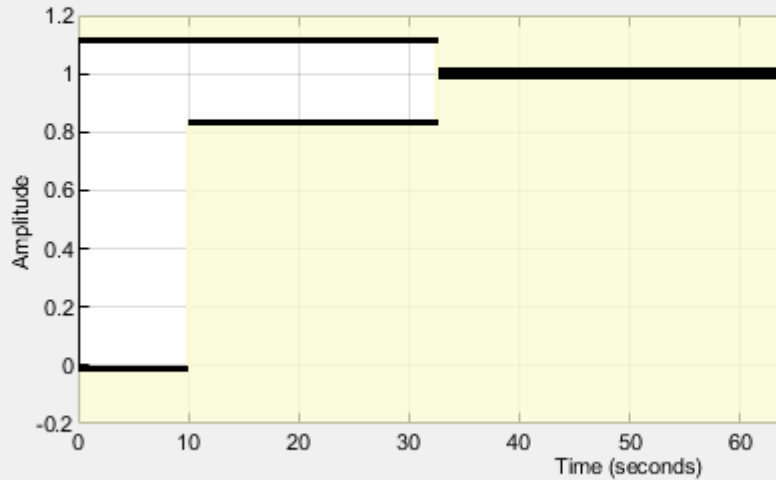
load pidtrack_demo

Check Custom Bounds [6] - Step Response Specifications

File Edit Tools Help



Update block



Ready

Block Parameters: Step Response Specifications

Check Custom Bounds

Assert that the input signal satisfies the specified bounds.

Bounds Assertion

Include upper bound in assertion

Times (seconds): [0 32.6127819548872;32.6127819548872 100]

Amplitudes: [1.11555555555556 1.11555555555556;1.01 1.01]

Include lower bound in assertion

Times (seconds): 74436 32.6127819548872;32.6127819548872 100]

Amplitudes: [.833333333333333 0.833333333333333;0.99 0.99]

Enable zero-crossing detection

Show Plot

Show plot on block open

Response Optimization...



OK

Cancel

Help

Apply

Response Optimization* - pidtrack_demo - Time plot 2

RESPONSE OPTIMIZATION | TIME PLOT | VIEW

Design Variables Set: None | Uncertain Variables Set: None

Open Session | Save Session | New | Evaluate Requirements | Select | Sensitivity Analysis

FILE | REQUIREMENTS | VARIABLES | PLOTS | OPTIONS | OPTIMIZE

Data Browser: Search workspace variables

MATLAB Workspace:

Name	Value
Kd	1.9688
Ki	0.0504
Kp	0.6300
a1	3

Model Workspace (pidtrack_demo):

Name	Value
pidtrack_optim	1x1 Sessio...

Data:

Name	Value
BlockReq	1x1 BlockR...
BlockReq_1	1x1 BlockR...

Variable Preview:

Time plot 1 | Time plot 2

There is no data for pidtrack_demo/Step Response Specifications, use "Plot Model Response" or run the optimization to update the plot.

pidtrack_demo/Step Response Specifications

Amplitude

Time (seconds)

Create Design Variables Set

Create Design Variables set: DesignVars

Variable	Value	Minimum	Maximum	Scale
Add model variables to optimize.				

Update model variables

Variable Detail

Variable	Current value	Used By
Kd	1.9688	pidtrack_demo/PID Controller/Derivative
Ki	0.0504	pidtrack_demo/PID Controller/Integral
Kp	0.63	pidtrack_demo/PID Controller/Proportional
a1	3	pidtrack_demo/Plant
a2	43	pidtrack_demo/Plant

Specify expression indexing if necessary (e.g., a(3) or s.x)

OK Cancel Help

Workspace:

Name	Value
a1	3
a2	43
Kd	1.9688
Ki	0.0504
Kp	0.6300
pidtrack_optim	1x1 SimProjectF
t	150x1 double
tout	655x1 double
y	150x1 double

Create Design Variables Set

Create Design Variables set: DesignVars

Variable	Value	Minimum	Maximum	Scale
<input checked="" type="checkbox"/> Kd	1.9688	-Inf	Inf	2
<input checked="" type="checkbox"/> Ki	0.0504	-Inf	Inf	0.0625
<input checked="" type="checkbox"/> Kp	0.63	-Inf	Inf	1

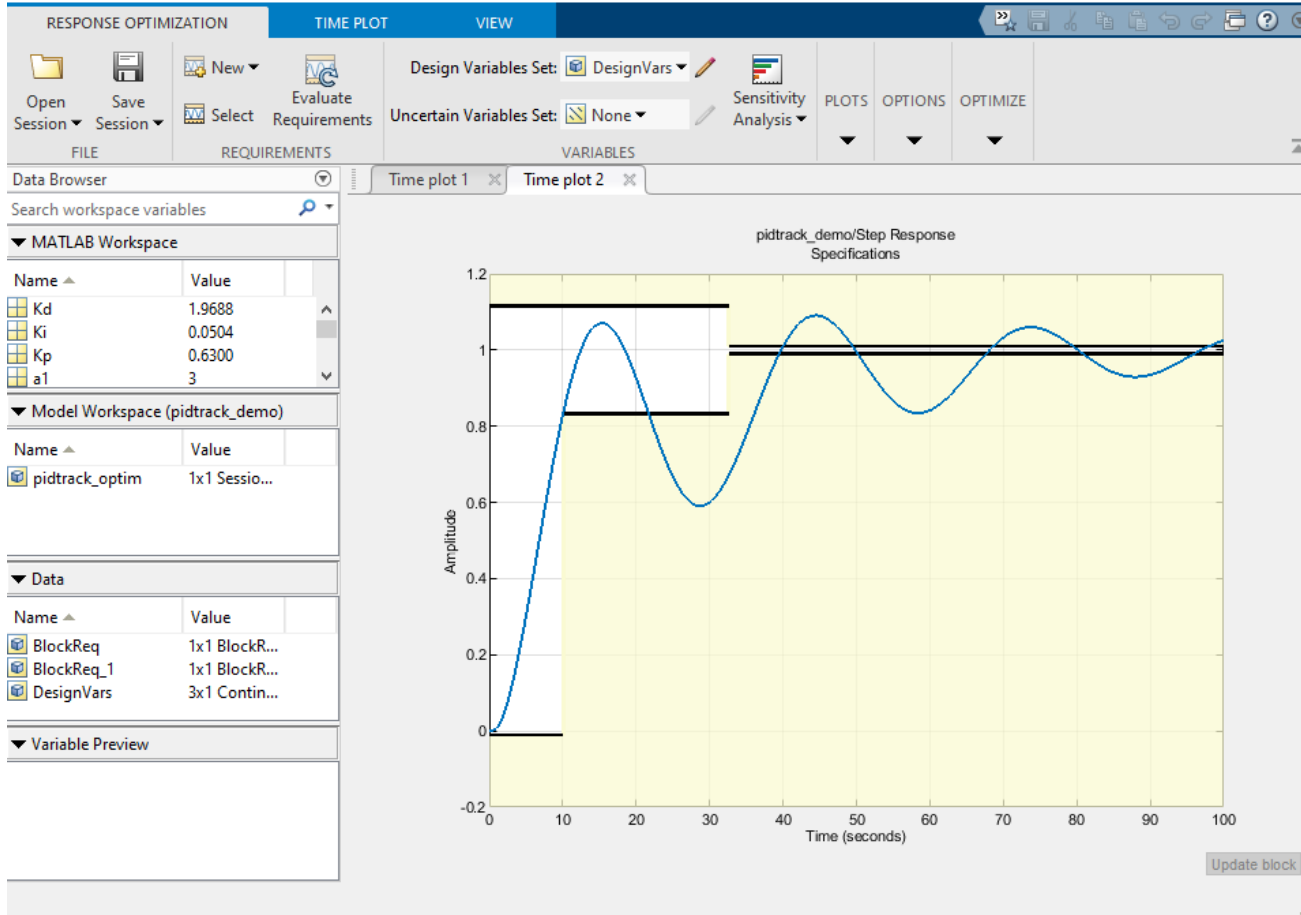
Update model variables

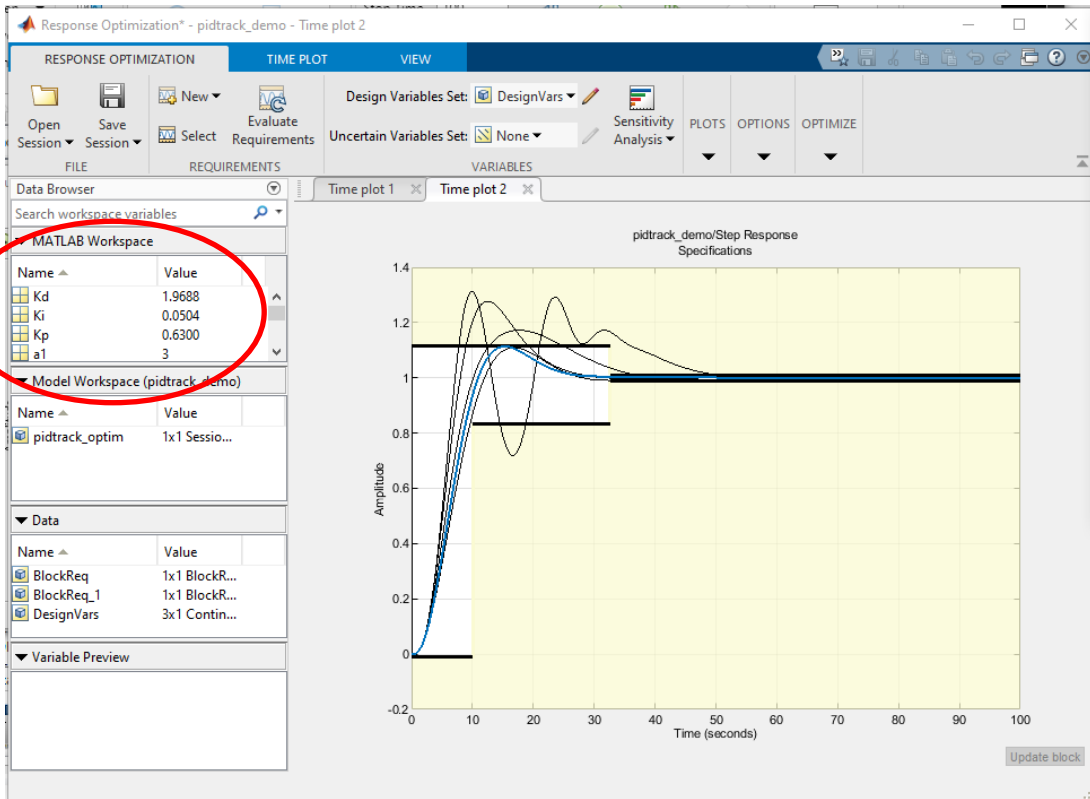
Variable Detail

Variable	Current value	Used By
a1	3	pidtrack_demo/Plant
a2	43	pidtrack_demo/Plant

Specify expression indexing if necessary (e.g., a(3) or s.x)

OK Cancel Help





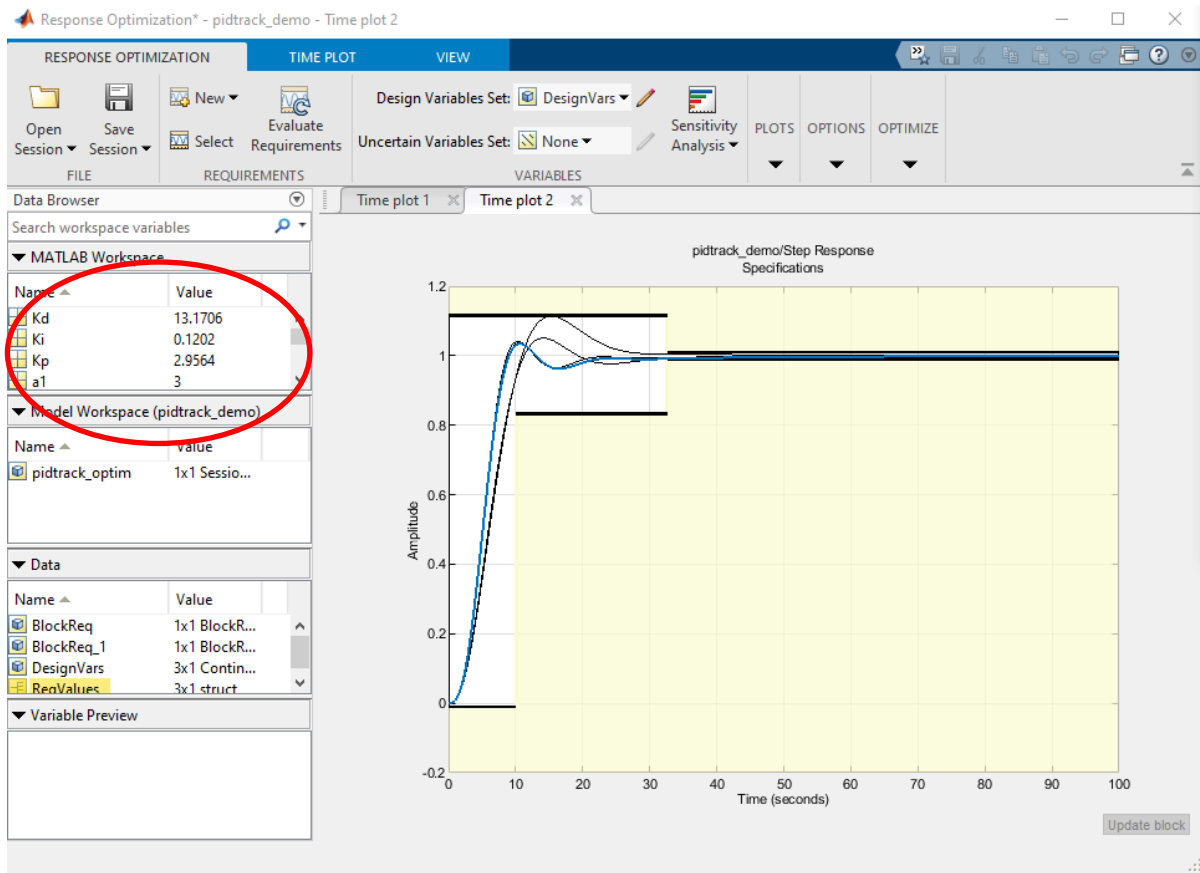
Optimization Progress Report

Iteration	F-count	Reference Tracking Spe... (Minimize)	Step Response Specific... (<=0)	Step Response Specific... (>=0)
0	7	24.4128	0.0800	-0.3218
1	23	22.9700	0.1342	-0.3450
2	32	14.7716	0.1784	-0.1371
3	45	10.1806	0.1448	0.0063
4	52	9.2197	0.0510	0.0022
5	61	9.8633	-0.0057	7.9333e-04
6	68	8.7471	-0.0018	0.0101

Optimization started 21-Dec-2022 19:17:18

Optimization terminated by user, 21-Dec-2022 19:23:37

Save Iteration... Display Options... Optimize



Optimization Progress Report

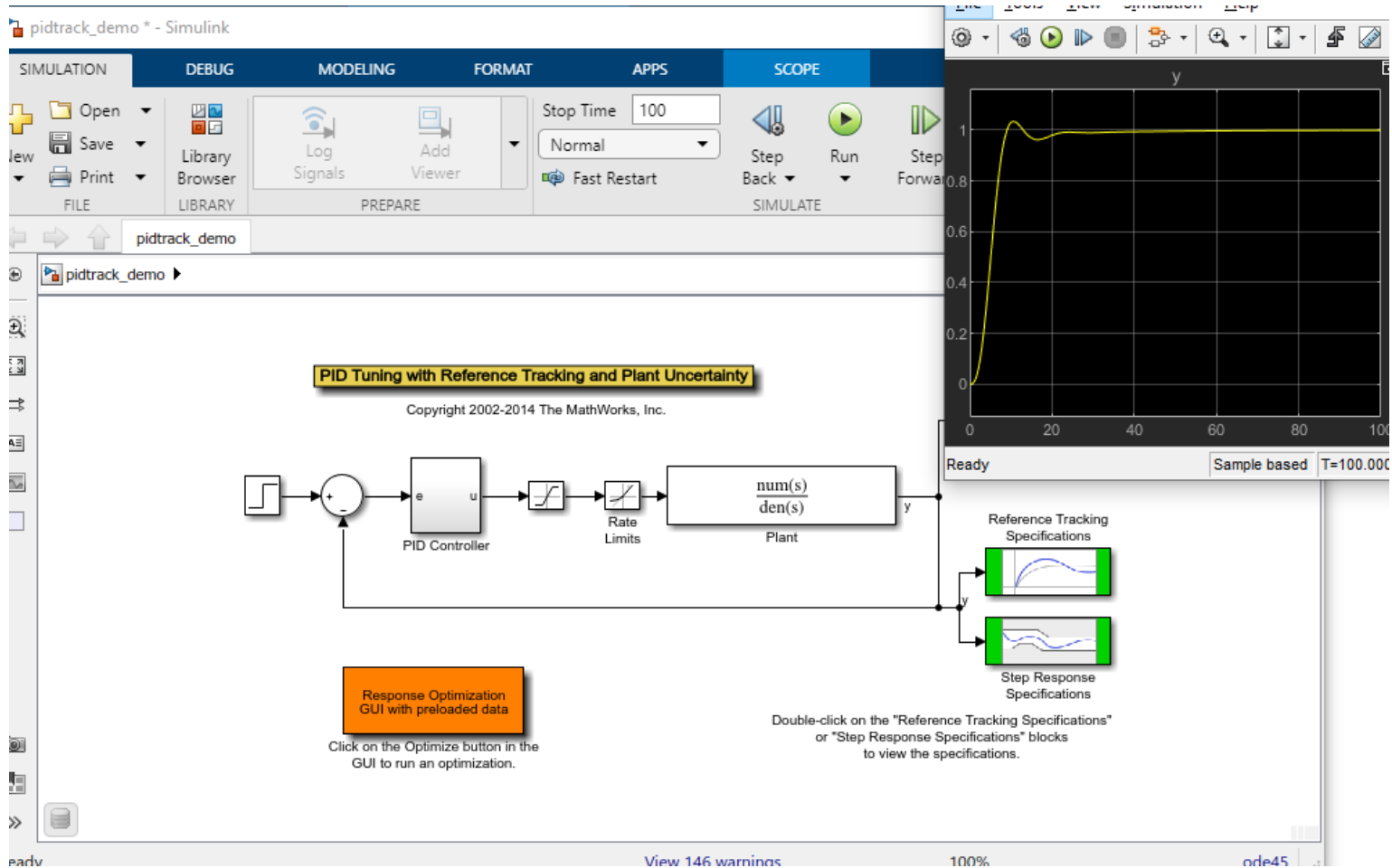
Iteration	F-count	Reference Tracking Spe... (Minimize)	Step Response Specific... (<=0)	Step Response Specific... (>=0)
0	7	24.4128	0.0800	-0.3218
1	23	22.9700	0.1342	-0.3450
2	32	14.7716	0.1764	-0.1371
3	45	10.1806	0.1448	0.0063
4	52	9.2197	0.0510	0.0022
5	61	9.8633	-0.0057	7.9333e-04
6	68	8.7471	-0.0018	0.0101
7	76	7.8919	-0.0100	0.0032
8	87	5.5772	-0.0103	0.0046
9	94	5.5444	-0.0105	0.0012
10	111	5.4806	-0.0105	0.0013

Optimization started 21-Dec-2022 19:26:27

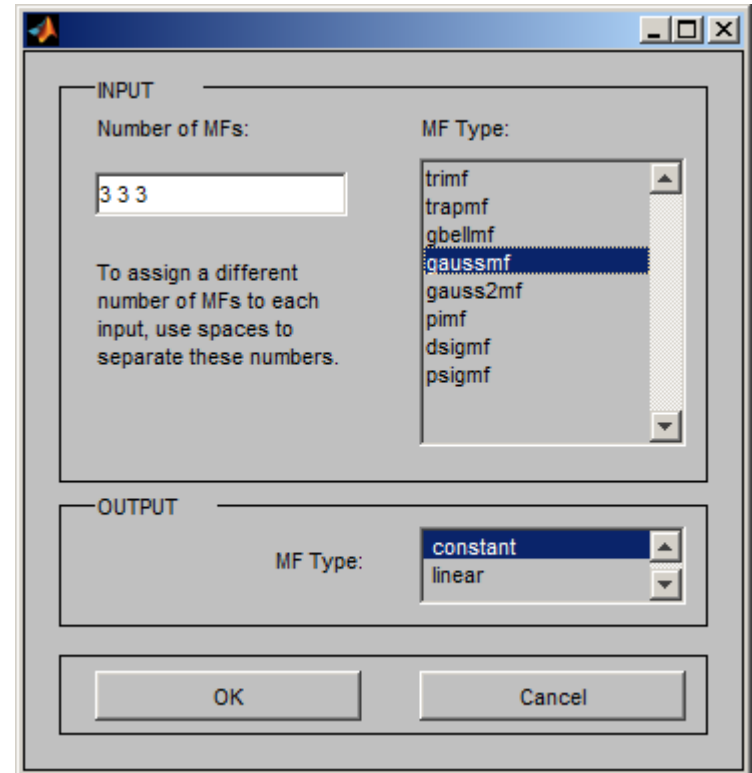
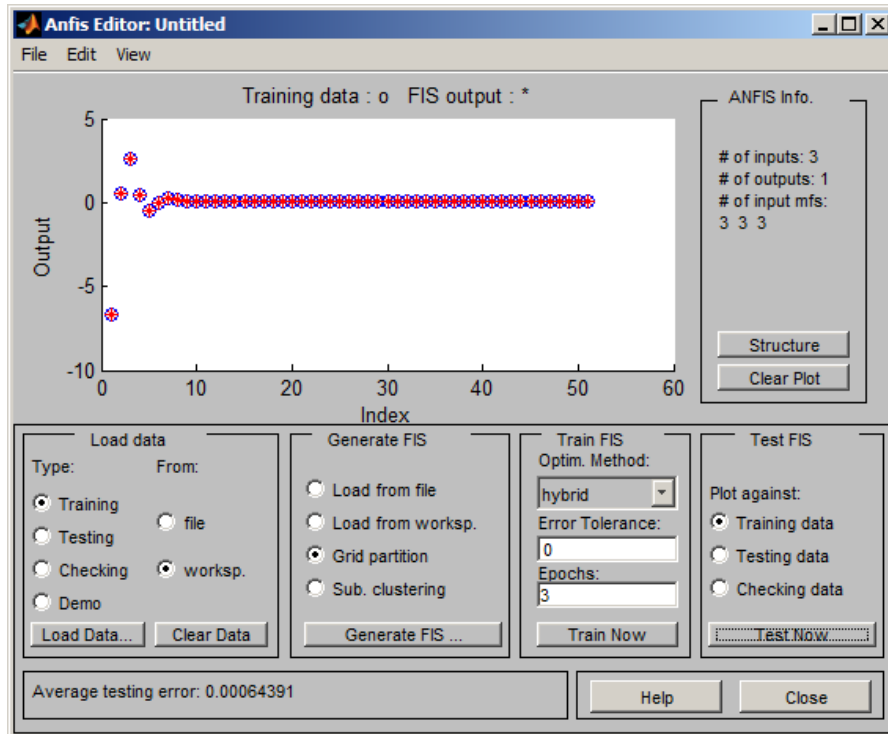
Optimization failed to converge, 21-Dec-2022 19:40:19

Optimized variable values written to 'DesignVars' in the Design Optimization workspace

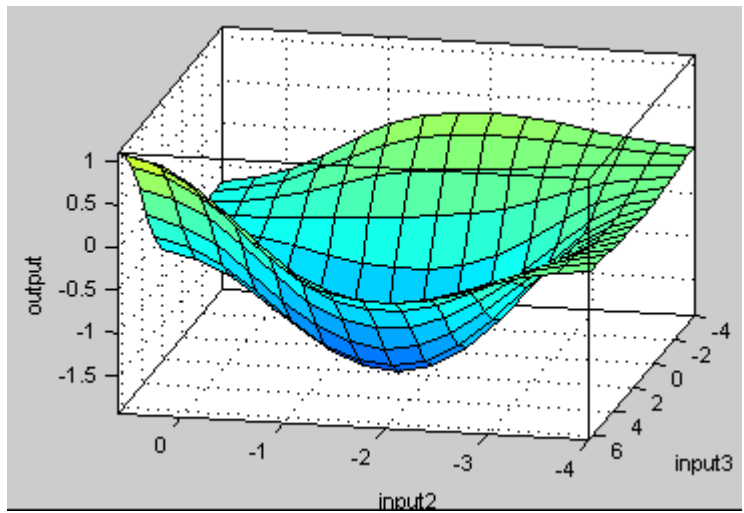
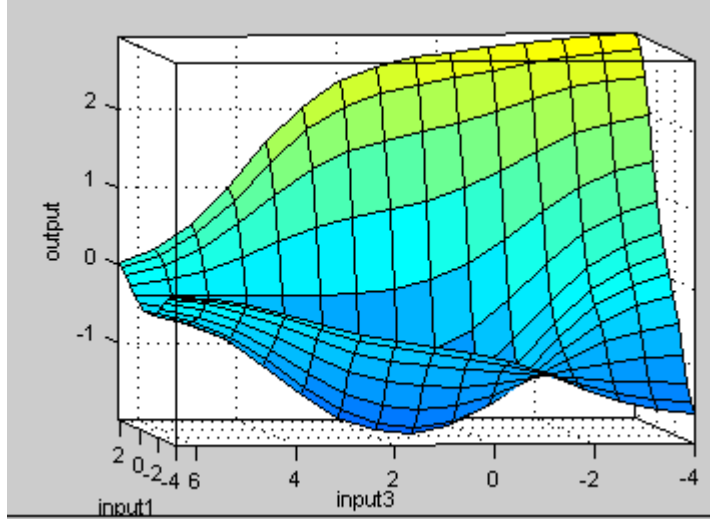
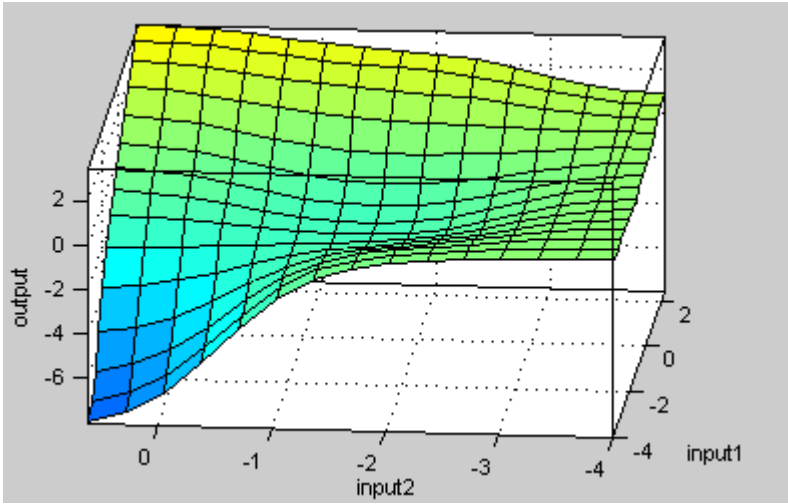
Save Iteration... | Display Options... | Optimize



Nauczanie ANFIS



FisPID



Rule Viewer: FisPID

File Edit View Options

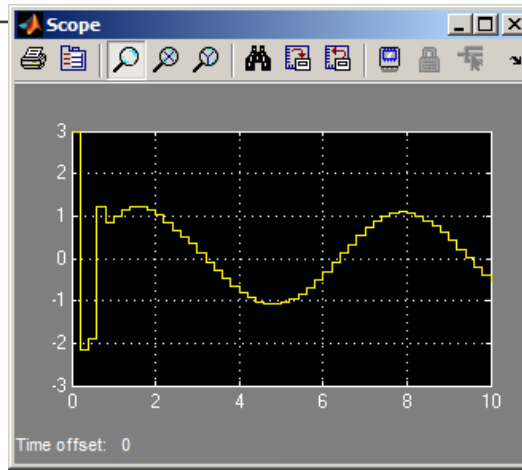
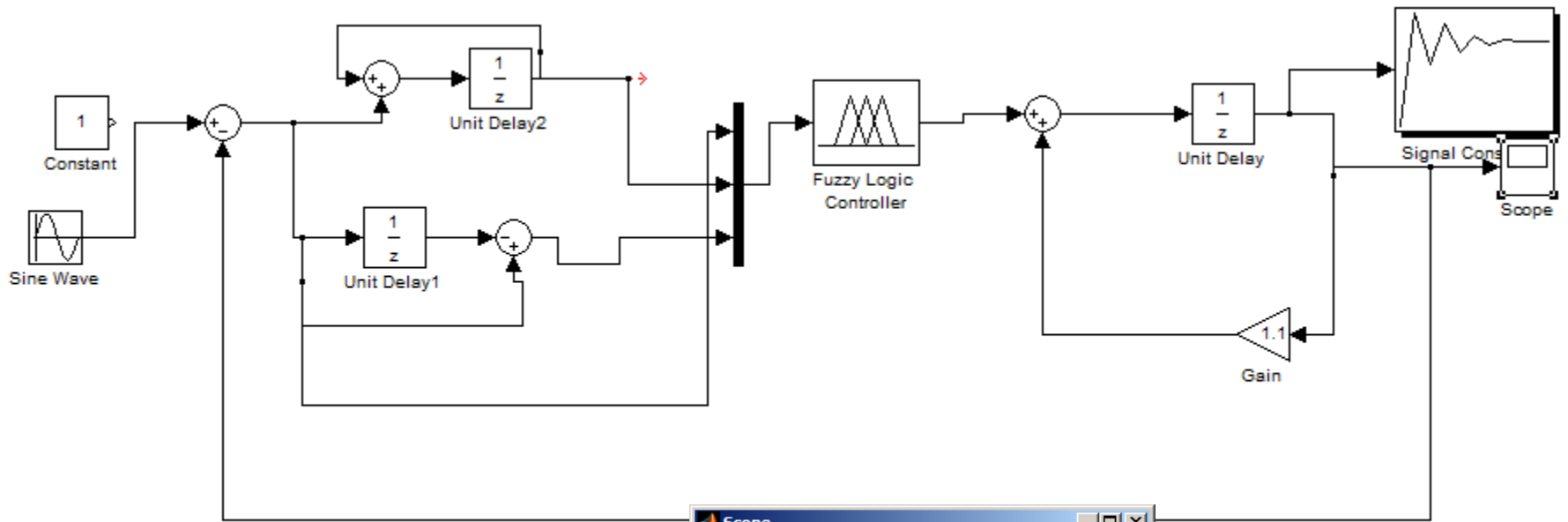
input1 = -1.64 input2 = -1.02 input3 = 0.076 output = -1.93

1	[rule 1]	[rule 1]	[rule 1]	[rule 1]
2	[rule 2]	[rule 2]	[rule 2]	[rule 2]
3	[rule 3]	[rule 3]	[rule 3]	[rule 3]
4	[rule 4]	[rule 4]	[rule 4]	[rule 4]
5	[rule 5]	[rule 5]	[rule 5]	[rule 5]
6	[rule 6]	[rule 6]	[rule 6]	[rule 6]
7	[rule 7]	[rule 7]	[rule 7]	[rule 7]
8	[rule 8]	[rule 8]	[rule 8]	[rule 8]
9	[rule 9]	[rule 9]	[rule 9]	[rule 9]
10	[rule 10]	[rule 10]	[rule 10]	[rule 10]
11	[rule 11]	[rule 11]	[rule 11]	[rule 11]
12	[rule 12]	[rule 12]	[rule 12]	[rule 12]
13	[rule 13]	[rule 13]	[rule 13]	[rule 13]
14	[rule 14]	[rule 14]	[rule 14]	[rule 14]
15	[rule 15]	[rule 15]	[rule 15]	[rule 15]
16	[rule 16]	[rule 16]	[rule 16]	[rule 16]
17	[rule 17]	[rule 17]	[rule 17]	[rule 17]
18	[rule 18]	[rule 18]	[rule 18]	[rule 18]
19	[rule 19]	[rule 19]	[rule 19]	[rule 19]
20	[rule 20]	[rule 20]	[rule 20]	[rule 20]
21	[rule 21]	[rule 21]	[rule 21]	[rule 21]
22	[rule 22]	[rule 22]	[rule 22]	[rule 22]
23	[rule 23]	[rule 23]	[rule 23]	[rule 23]
24	[rule 24]	[rule 24]	[rule 24]	[rule 24]
25	[rule 25]	[rule 25]	[rule 25]	[rule 25]
26	[rule 26]	[rule 26]	[rule 26]	[rule 26]
27	[rule 27]	[rule 27]	[rule 27]	[rule 27]

Input: [-1.645; -1.019; 0.07595] Plot points: 101 Move: left right down up

Opened system FisPID, 27 rules

Help Close



m3

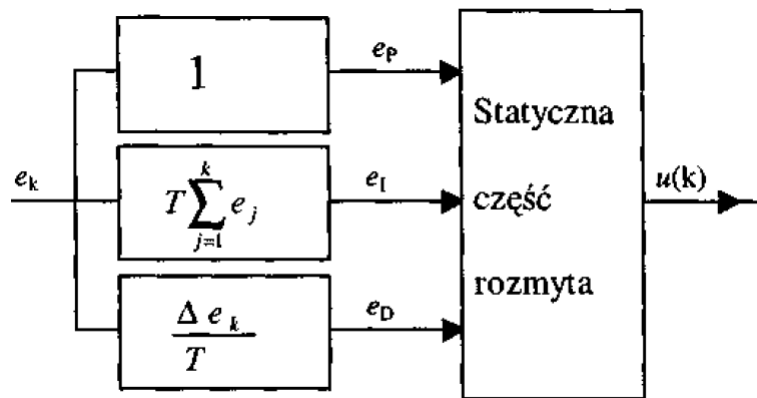
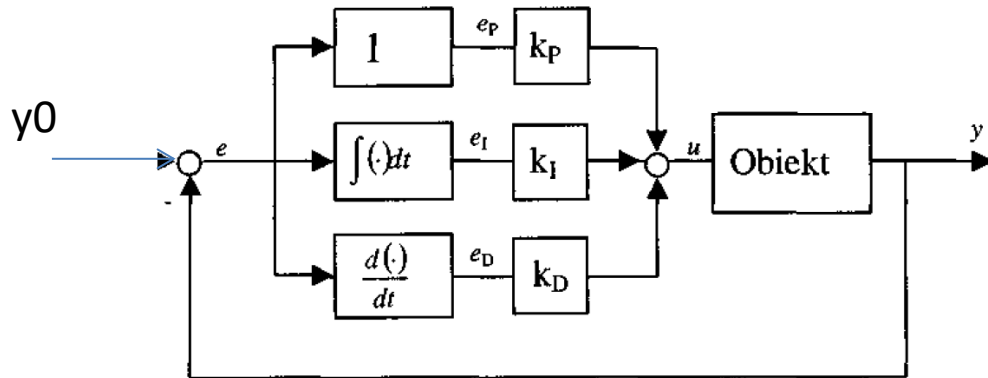
Sterownik Takagi-Sugeno

W przypadku tego sterownika konkluzja k -tej reguły ma postać $u = f^{(k)}(x_1, \dots, x_n)$.
W celu wyznaczenia wartości sterowania \bar{u} postępujemy następująco:

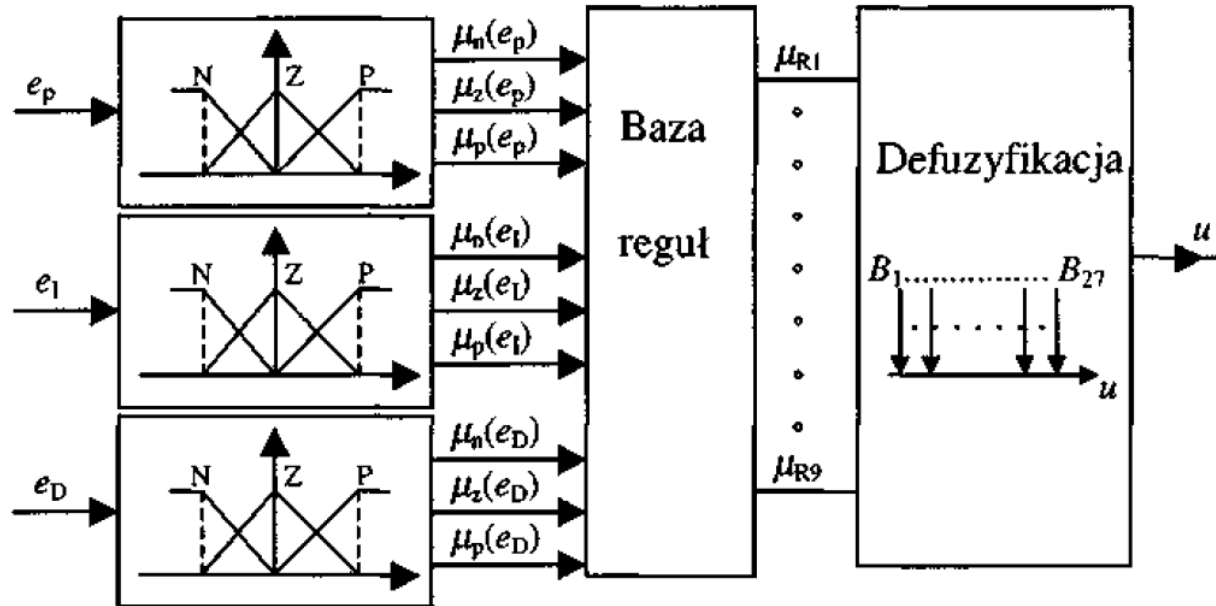
1. Dla każdej reguły wyznaczamy stopień zgodności $w^{(k)}$ obserwacji $(\bar{x}_1, \dots, \bar{x}_n)$ z przesłankami k -tej reguły, $w^{(k)} = \mu_{A_1^{(k)}}(\bar{x}_1) * \dots * \mu_{A_n^{(k)}}(\bar{x}_n)$, gdzie $*$ jest symbolem T-normy (jest to zazwyczaj min lub iloczyn). Obliczamy także wartość $\bar{u}^{(k)} = f^{(k)}(x_1, \dots, x_n)$.
2. Poszukiwaną wartość \bar{u} wyznaczamy z równania

$$\bar{u} = \frac{\sum_{k=1}^N w^{(k)} \bar{u}^{(k)}}{\sum_{k=1}^N w^{(k)}}$$

Regulator PID vs FuzzyPID



Statyczna część rozmyta Mamdaniego



- R1: JEŚLI ($e_p = N$) I ($e_l = N$) I ($e_D = N$) TO ($u = B_1$)*
R2: JEŚLI ($e_p = N$) I ($e_l = N$) I ($e_D = Z$) TO ($u = B_2$)
⋮
R27: JEŚLI ($e_p = P$) I ($e_l = P$) I ($e_D = P$) TO ($u = B_{27}$)

Pytania

- Jaką ilość zbiorów rozmytych wejść i jaką ilość reguł stosować w regulatorze?
- Jaka jest różnica między konwencjonalnym i rozmytym regulatorem PID?
- Czy stosowanie rozmytych regulatorów PID ma sens jeśli istnieją konwencjonalne regulatory PID?

Konwencjonalny a rozmyty PID

$$u = k_p e_p + k_I e_I + k_D e_D$$

$$u = \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^4 v_i w_j q_k (a_{0ijk} + a_{1ijk} e_p + a_{2ijk} e_I + a_{3ijk} e_D + a_{4ijk} e_p e_I + a_{5ijk} e_p e_D + a_{6ijk} e_I e_D + a_{7ijk} e_p e_I e_D)$$

Zmienne logiczne v, w, q informują o przynależności aktualnej wartości sygnału wejściowego do **określonego sektora** jego przestrzeni rozważań

$$v_1 = \begin{cases} 1 & \text{dla } e_p \leq e_{p1} \\ 0 & \text{poza tym} \end{cases}$$

$$v_2 = \begin{cases} 1 & \text{dla } e_{p1} < e_p \leq e_{p2} \\ 0 & \text{poza tym} \end{cases}$$

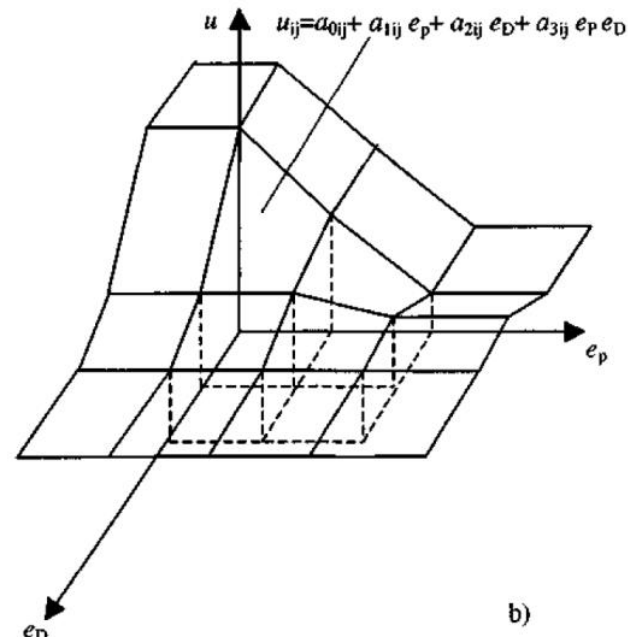
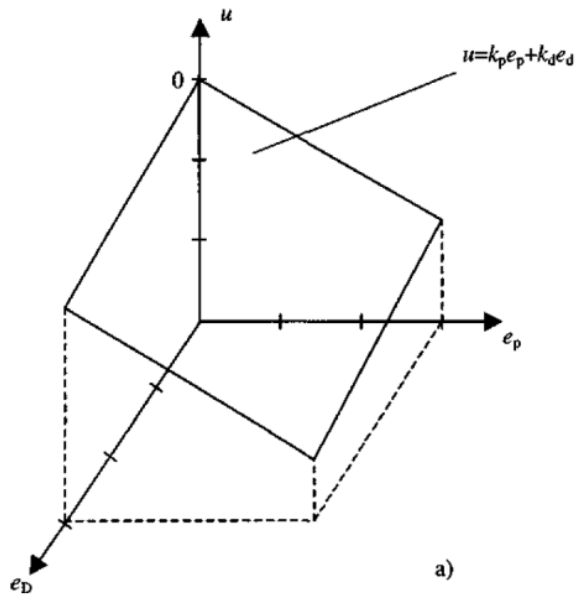
$$v_3 = \begin{cases} 1 & \text{dla } e_{p2} < e_p \leq e_{p3} \\ 0 & \text{poza tym} \end{cases}$$

$$v_4 = \begin{cases} 1 & \text{dla } e_{p3} < e_p \\ 0 & \text{poza tym} \end{cases}$$

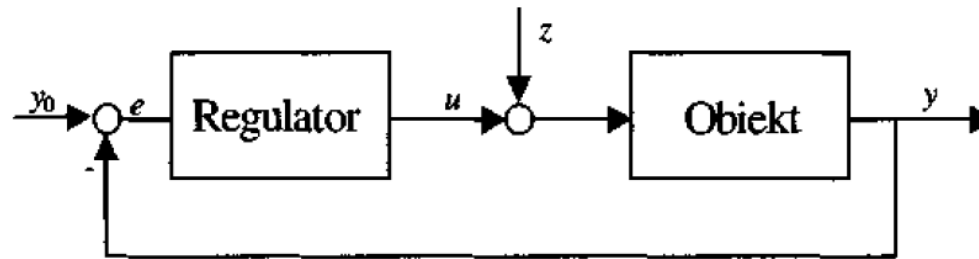
K.PD a R.PD

Powierzchnia regulatora K.PD jest płaszczyzną a 2 stopnie swobody k_P , k_D regulatora pozwalają jedynie na zmianę jej pochylenia.

Powierzchnia regulatora R.PD składa się z 16 segmentów nie będących płaszczyznami (powierzchnie wieloliniowe o prostych krawędziach). Segmenty te mogą zajmować różne położenie w przestrzeni i posiadać różny stopień wypukłości, zależnie od wartości 15 stopni swobody (wartości modalne funkcji przynależności wejść e_P , e_D i wyjścia u).



Kryteria jakości sterowania



błąd

$$K_{1y_0}^c = \min \int_0^{\infty} e_{y_0}^2(t) dt,$$

$$K_{1y_0}^d = \min \sum_{k=1}^{\infty} e_{y_0}^2(k) \cdot T,$$

$$K_{1z}^c = \min \int_0^{\infty} e_z^2(t) dt,$$

$$K_{1z}^d = \min \sum_{k=1}^{\infty} e_z^2(k) \cdot T.$$

energia

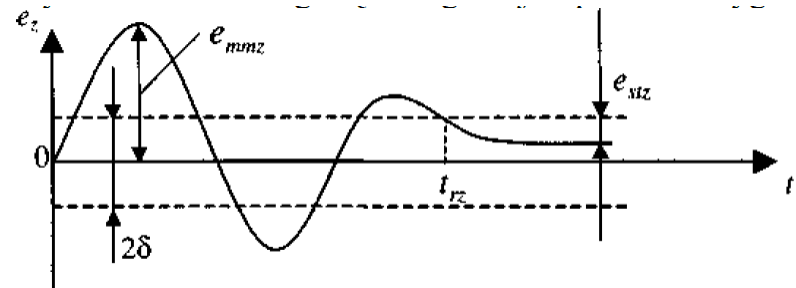
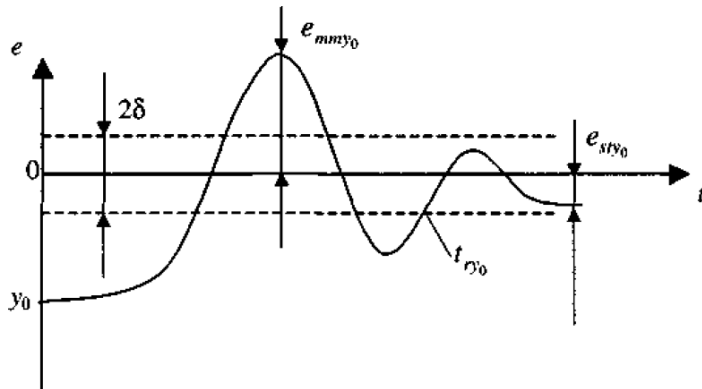
$$K_{2y_0}^c = \min \int_0^{\infty} u_{y_0}^2(t) dt,$$

$$K_{2y_0}^d = \min \sum_{k=1}^{\infty} u_{y_0}^2(k) \cdot T,$$

$$K_{2z}^c = \min \int_0^{\infty} u_z^2(t) dt,$$

$$K_{2z}^d = \min \sum_{k=1}^{\infty} u_z^2(k) \cdot T.$$

Kryteria jakości sterowania (cd)



kryteria odcinkowe błędu śledzenia skoku sygnału zadającego

kryteria zapewnienia możliwie spokojnej pracy urządzenia nastawczego i zminimalizowania ilości jego przełączeń

w przypadku śledzenia skoku sygnału zadającego dąży się do minimalizacji lub utrzymania w pewnych granicach czasu regulacji, przeregulowania i błędu statycznego

Kryteria jakości sterowania (cd)

$$K = \min \int_0^t [\alpha e^2(t) + (1 - \alpha) u^2(t)] dt$$

$$K_{3y0} = \min t_{ry0},$$

$$K_{3z} = \min t_{rz},$$

$$K_{4y0} = \min p_{ry0}$$

$$K_{4z} = \min \frac{e_{mmz}}{z},$$

$$K_{5y0} = \min e_{scy0},$$

$$K_{5z} = \min e_{stz}.$$

$$K = \min(\alpha_1 K_1 + \alpha_2 K_2 + \dots + \alpha_n K_n),$$

Większa ilość stopni swobody regulatora rozmytego ułatwia minimalizację większej ilości kryteriów składowych problemu wielokryterialnego!

Mała ilość stopni swobody regulatora liniowego K.PID znacznie tą minimalizację utrudnia.

Stosowanie regulatorów rozmytych do sterowania obiektów liniowych jest więc jak najbardziej wskazane.

Określenie struktury i parametrów regulatorów rozmytych

Syntezę regulatora rozmytego można przeprowadzić następującymi metodami:

I. na bazie **wiedzy eksperta**,

II. przez **modelowanie eksperta - regulatora** na bazie generowanych przez niego sygnałów sterujących,

III. na bazie **modelu sterowanego obiektu**.

Metody I i II, w których **projektowanie odbywa się bez modelu obiektu** opierają się na umiejętności prawidłowego sterowania obiektem przez eksperta, która nabyta została w drodze często **żmudnych i długotrwałych doświadczeń** i która w sposób utajony zawiera wiedzę o dynamice i innych cechach obiektu.

Np. prawidłowe kierowanie samochodem wymaga zdobycia wyczucia posługiwania się pedałem gazu i sprzęgła, umożliwiającego **płynny start samochodu, wyczucia możliwości przyspieszania, zachowania samochodu przy różnych prędkościach, możliwości szybkiego hamowania**, itd.

Po zmianie samochodu wyczucie to musimy zdobywać od nowa. Kierowania samochodem możemy nauczyć się nie znając jego modelu matematycznego. Podobnie jest w przypadku innych urządzeń technicznych (dźwigów, samolotów, statków, procesów produkcyjnych).

Projektowanie regulatora rozmytego na bazie wiedzy eksperta sterowanego obiektu

Wiedza eksperta ma postać lingwistycznych reguł typu

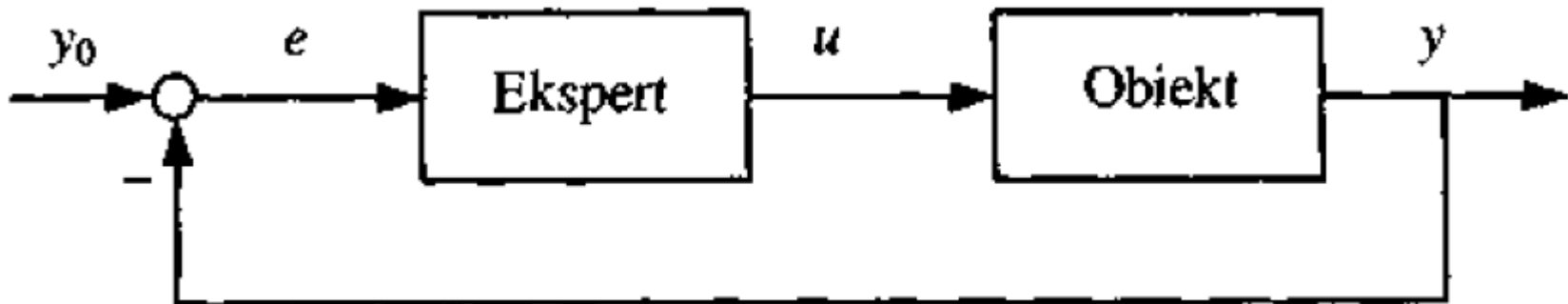
JEŻELI (x = A) TO (y = B)

gdzie **y** jest obserwowaną wielkością sterowaną, a **x** wielkością nastawianą przez eksperta. A i B są ocenami lingwistycznymi np.: mały, duży, wolny, szybki, itd.

Poza regułami lingwistycznymi dotyczącymi sterowania obiektem należy również uzyskać od eksperta informacje o stosowanych przez niego **ocenach lingwistycznych** w celu określenia istotnych **parametrów funkcji przynależności** - głównie wartości modalnych.

Projektowanie regulatora rozmytego metodą modelowana eksperta sterującego obiektem

Ponieważ ekspert nie potrafi w wielu przypadkach sformułować swej mentalnej wiedzy o sterowaniu obiektem, możemy wykorzystać sygnały generowane przez niego w procesie sterowania do stworzenia jego modelu



W tym celu należy pomierzyć i zarejestrować przebieg błędu regulacji $e(t)$ będący podstawą podejmowania decyzji przez eksperta oraz wygenerowany przez niego sygnał sterujący $u(t)$.

Następnie stosując odpowiednie metody modelowania, np. metody oparte na logice rozmytej, sieciach neuronowych, klasteryzacji lub metody konwencjonalne, należy opracować **model matematyczny eksperta -regulatora**.

Ekspert–regulator

Człowiek realizuje sterowanie nieliniowe i adaptacyjne. **Przy małych wartościach błędu regulacji posiada on dynamikę zbliżoną do regulatora PI**, dzięki czemu potrafi sterować bez błędu statycznego.

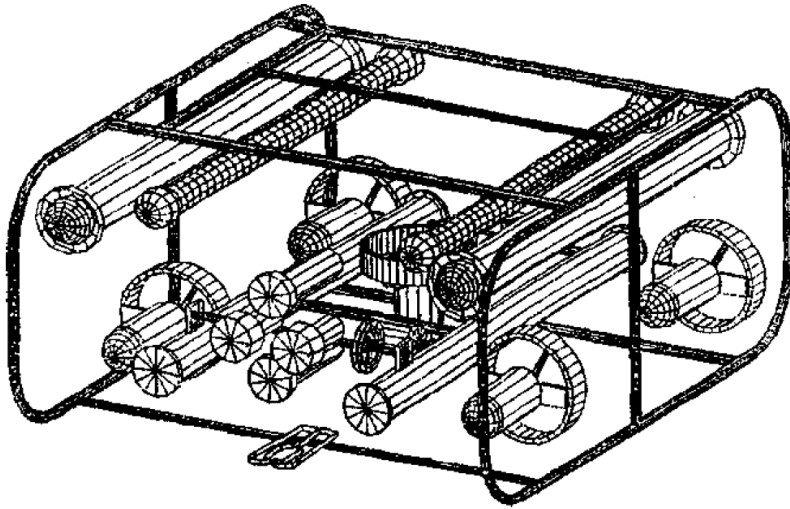
Jest to szczególnie ważne np. przy stabilizacji kursu statków lub samolotów. **Przy większych wartościach błędu człowiek działa jak regulator PD** z tendencją do sterowania typu bang-bang, tzn. sterowania dwustanowego polegającego na dość gwałtownym przełączeniu sygnału sterującego u z wartości maksymalnej (nasycenie górne) na minimalną (nasycenie dolne).

Człowiek dostosowuje swój sposób sterowania do rodzaju obiektu. I tak np. przy sterowaniu modelem helikoptera, którego transmitancja zawiera pojedyncze i podwójne całkowanie (w związku z czym całkowanie w regulatorze w celu usunięcia błędu statycznego jest zbędne) człowiek steruje w sposób zbliżony do regulatora PD. Jego transmitancja ma postać

$$G(s) = \frac{u(s)}{e(s)} = \frac{k_r e^{-sT_0} (1 + sT_d)}{(1 + sT_{nm})(1 + sT_r)}$$

Czas opóźnienia T_0 w przypadku sterowania modelem helikoptera waha się zwykle w zakresie 0.1-0.2 s. Stała czasowa T_{nm} charakteryzuje inercję układu nerwowego (n) i mięśniowego (m) człowieka i waha się w zakresie 0.1-0.6s. Wzmocnienie k_r , czas różniczkowania T_d oraz druga stała czasowa inercji T_r są zmienne i zależą od sterowania obiektu.

Sterowanie pojazdem podwodnym przez operatora



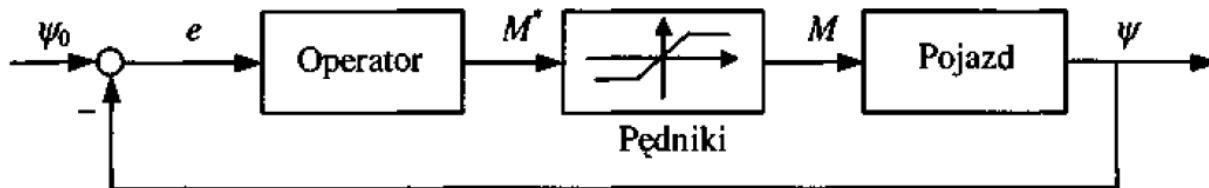
Zadanie sterowania polegało na zmianie kursu modelu pojazdu podwodnego.

Pojazd zmienia swój kurs ψ pod wpływem momentu obrotowego M generowanego przez parę bocznych śrub napędowych zwanych pędnikami. Moment ten nie może być dowolnie duży. Jest on ograniczony do zakresu $[-18.3, 28.3]$. Górna i dolna wartość ograniczenia jest różna, co utrudnia sterowanie. Operator obserwuje rzeczywisty kurs pojazdu $\psi(t)$ porównuje go z kursem zadany ψ_0 i generował, przy pomocy joysticka (dźwigni sterującej), elektryczny sygnał M^* sterujący pędnikami, które z kolei wytwarzały moment M pędników, obracający pojazd.

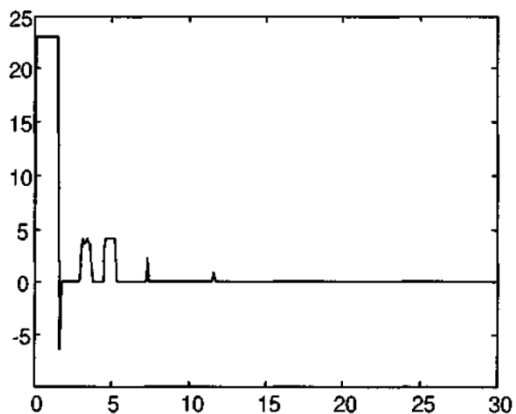
Podczas eksperymentu zarejestrowano przebiegi momentu $M(t)$ i kąta kursowego $\psi(t)$ pojazdu dla różnych wartości zadanych kursu

ψ_0

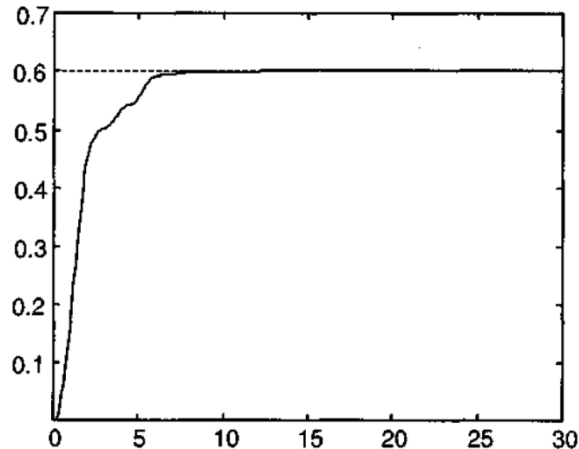
Sterowanie pojazdem podwodnym przez operatora (cd)



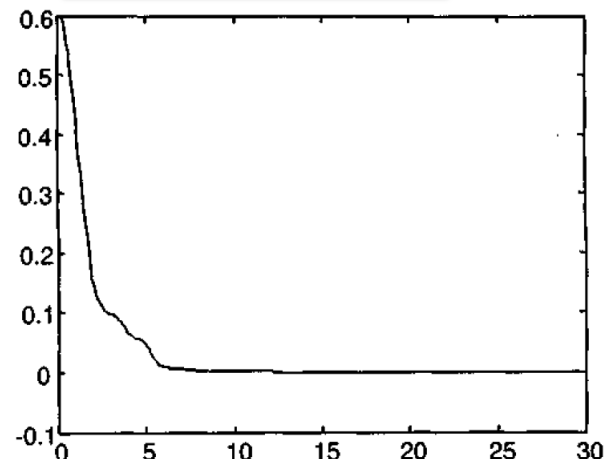
sygnał sterujący $M(t)$



kurs pojazdu $\psi(t)$



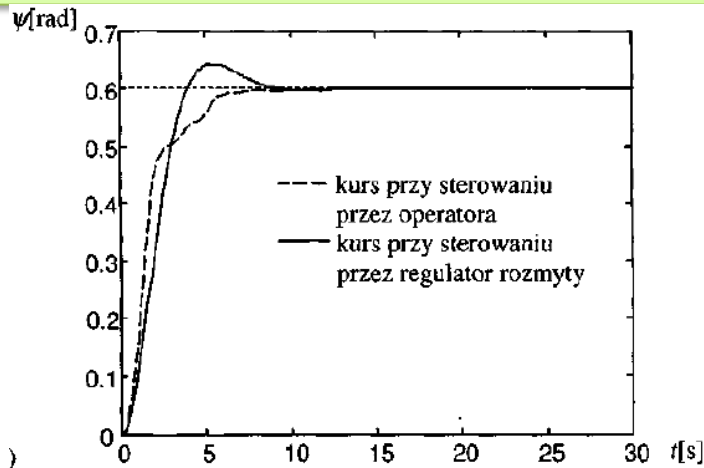
błąd regulacji $e(t)$



Operator nie działa jak regulator liniowy (gwałtowne zmiany sygnału sterującego, nieregularności w zmianach kąta kursowego).

Sterowanie pojazdem podwodnym przez operatora (cd)

Sygnaly $e(t)$ i $M(t)$ ze sterowania ręcznego zostały użyte do nastrojenia rozmytego regulatora neuronowego PID z 27 regułami.

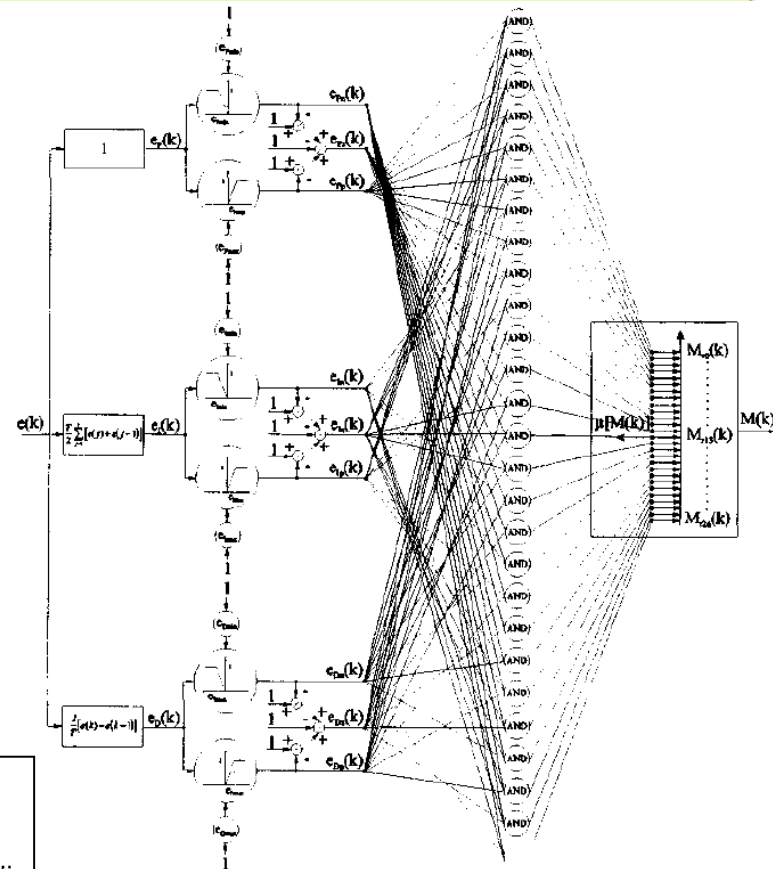


Działanie regulatora rozmytego jest daleko bardziej „delikatne” niż człowieka, który generował dość gwałtowne bądź też niekiedy zupełnie przypadkowe i niepotrzebne ruchy dźwigni sterującej.

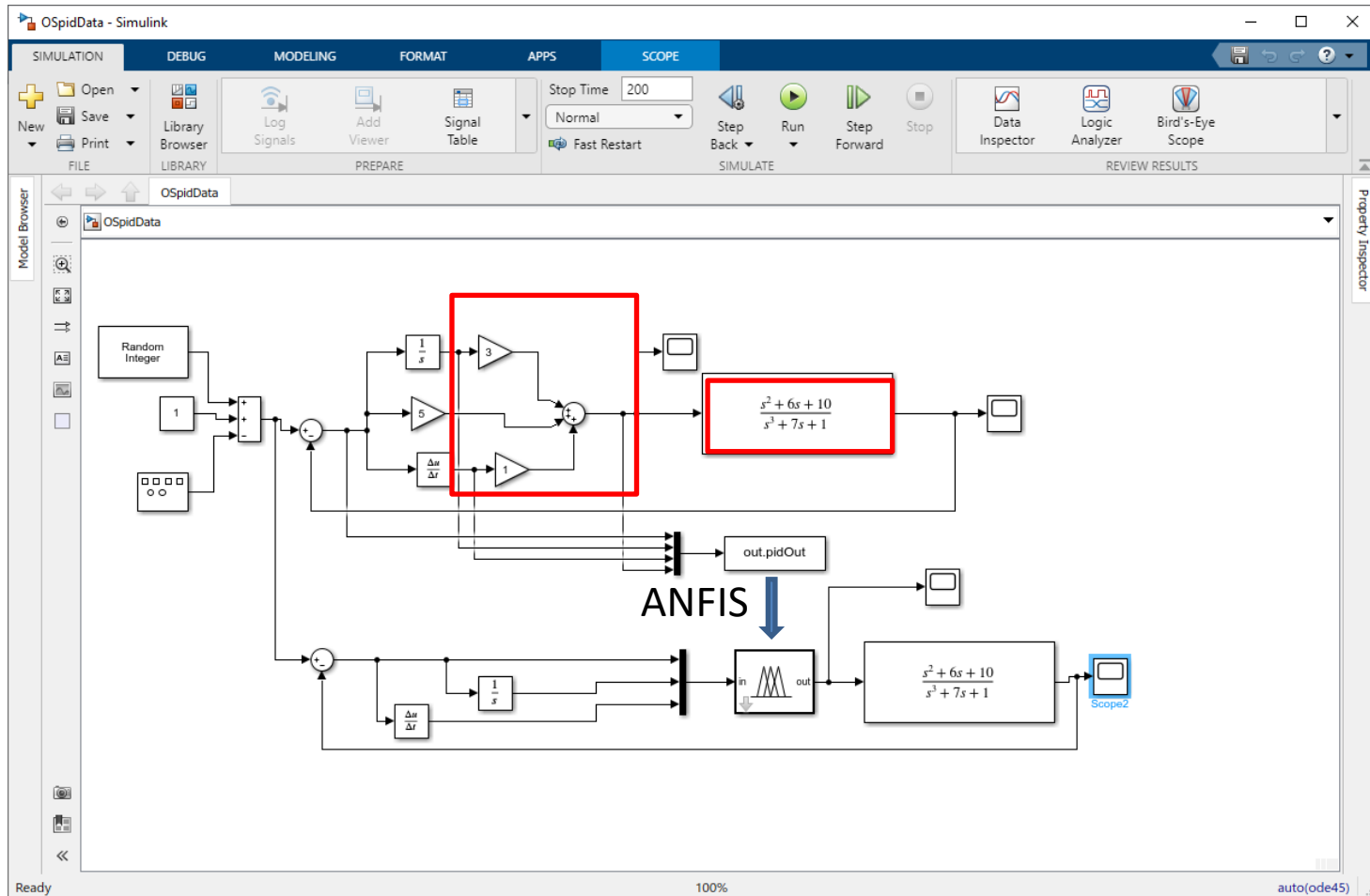
$$\int_0^{\infty} |e| dt = 1.07, \quad \int_0^{\infty} |u| dt = 41.30$$

Przy sterowaniu przez regulator rozmyty:

$$\int_0^{30s} |e| dt = 0.95, \quad \int_0^{30s} |u| dt = 45.86$$



Przykład



ANFIS

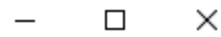
The image shows two windows from the Neuro-Fuzzy Designer software. The left window is titled "Add Membership Functions" and has two sections: "INPUT" and "OUTPUT".

- INPUT Section:** "Number of MFs:" is set to "3 3 3". "MF Type:" is a list box with "trimf" selected. Other options include trapmf, gbellmf, gaussmf, gauss2mf, pimf, dsigmf, and psigmf.
- OUTPUT Section:** "MF Type:" is a list box with "linear" selected. Other options include "constant".
- Buttons: "OK" and "Cancel".

The right window is titled "Neuro-Fuzzy Designer: Untitled" and contains the main interface:

- Menu:** File, Edit, View.
- Plot:** "Training Data (ooo)" showing Output vs. data set index. The plot shows a horizontal blue line at Output = 0 and scattered blue data points.
- ANFIS Info:** # of inputs: 3, # of outputs: 1, # of input mfs: 3 3 3, # of train data pairs: 1978. Buttons: "Structure", "Clear Plot".
- Load data:** Type: Training, Testing, Checking, Demo. From: file, worksp. Buttons: "Load Data...", "Clear Data".
- Generate FIS:** Load from file, Load from worksp., Grid partition, Sub. clustering. Button: "Generate FIS ...".
- Train FIS:** Optim. Method: hybrid, Error Tolerance: 0, Epochs: 3. Button: "Train Now".
- Test FIS:** Plot against: Training data, Testing data, Checking data. Button: "Test Now".
- Status:** "train data loaded". Buttons: "Help", "Close".

Fuzzy Logic Designer: FuzzyPID2



File Edit View



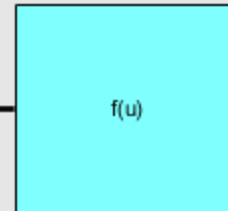
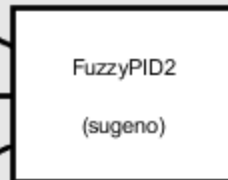
input1



input2

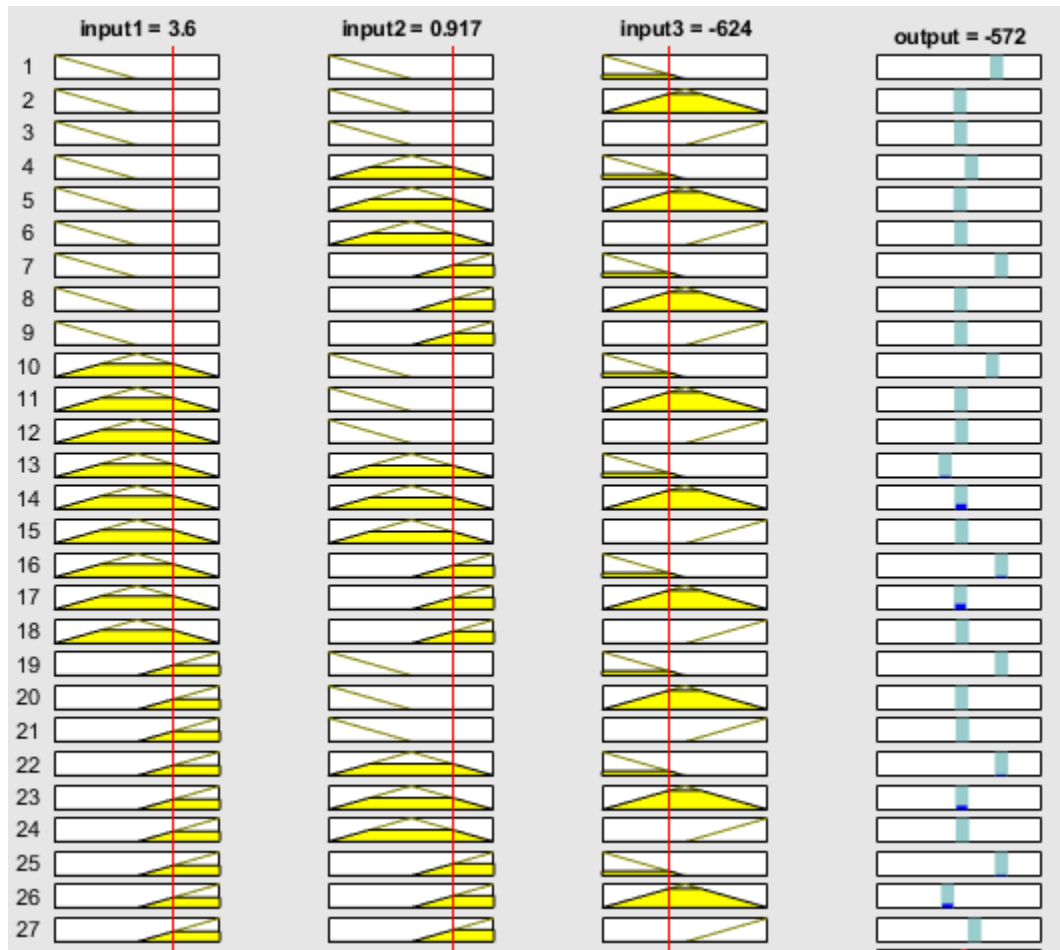


input3



output

FIS Name:	FuzzyPID2	FIS Type:	sugeno
And method	prod	Current Variable	
Or method	probor	Name	input1
Implication	min	Type	input
Aggregation	max	Range	[-4.9904000498858 6.999787853187381]
Defuzzification	wtaver	Help Close	
Saved FIS "FuzzyPID2" to file			



```
[Input1]
Name='input1'
Range=[-4.9904000498858 6.99978785318738]
NumMFs=3
MF1='in1mf1':'trimf',[-10.9854940014224 -4.99004793063842 1.0053556880085]
MF2='in1mf2':'trimf',[-4.99021269462818 1.00504602080603 6.99978785306744]
MF3='in1mf3':'trimf',[1.00469389668743 6.99978785309523 12.994881804724]

[Input2]
Name='input2'
Range=[-0.930551041307905 1.52319455614455]
NumMFs=3
MF1='in2mf1':'trimf',[-2.15742384003413 -0.929053917935743 0.306055368512293]
MF2='in2mf2':'trimf',[-0.930320767242993 0.297818878486571 1.52319455526083]
MF3='in2mf3':'trimf',[0.296321778037389 1.52319455384063 2.75006735487077]

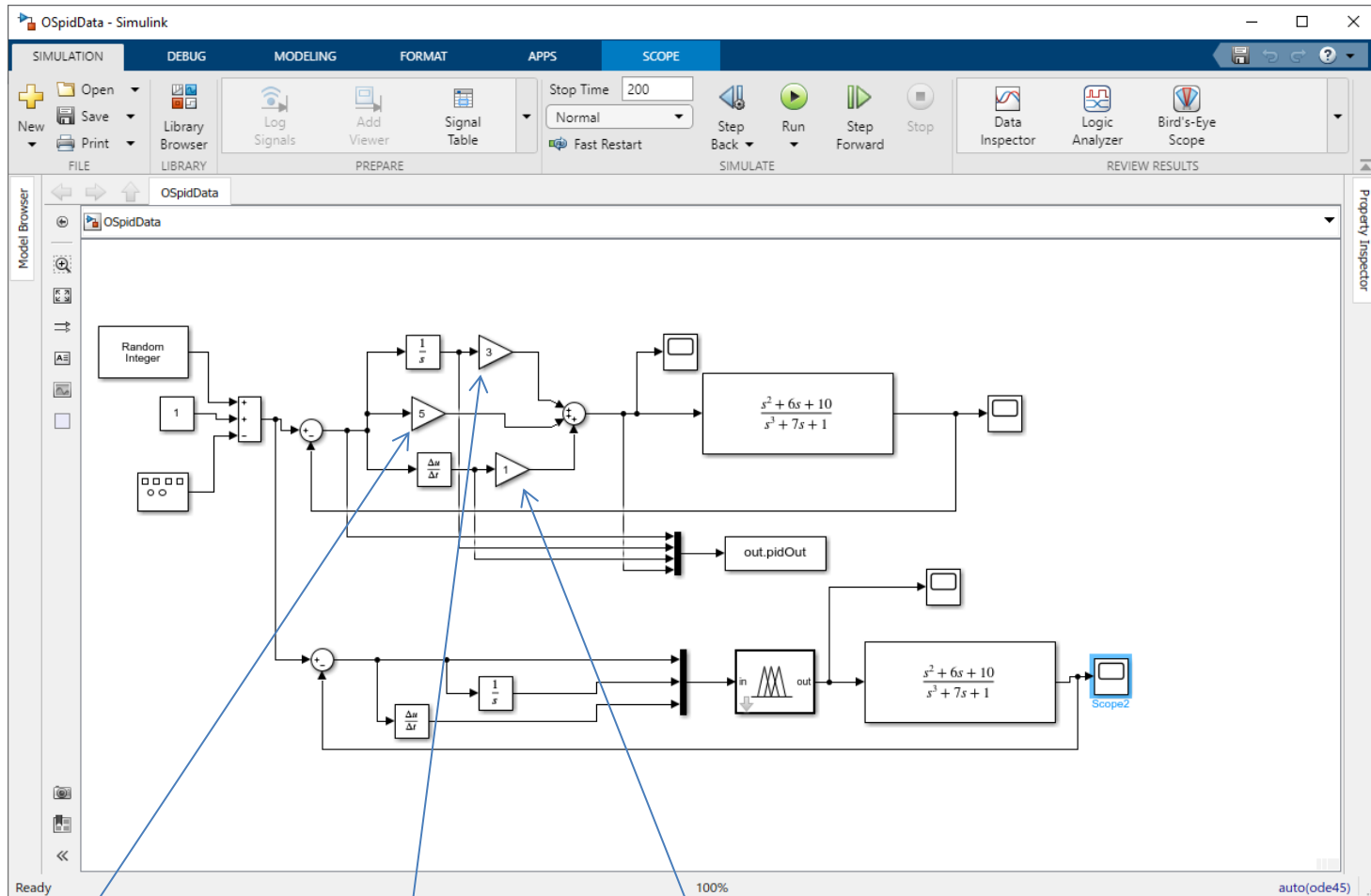
[Input3]
Name='input3'
Range=[-1237.84997148908 281.922287498847]
NumMFs=3
MF1='in3mf1':'trimf',[-1997.73610098304 -1237.84997148908 -477.963841995116]
MF2='in3mf2':'trimf',[-1237.84997144768 -477.963841995129 281.922287498824]
MF3='in3mf3':'trimf',[-477.963841995123 281.922287498834 1041.80841699281]
```

[Output1]

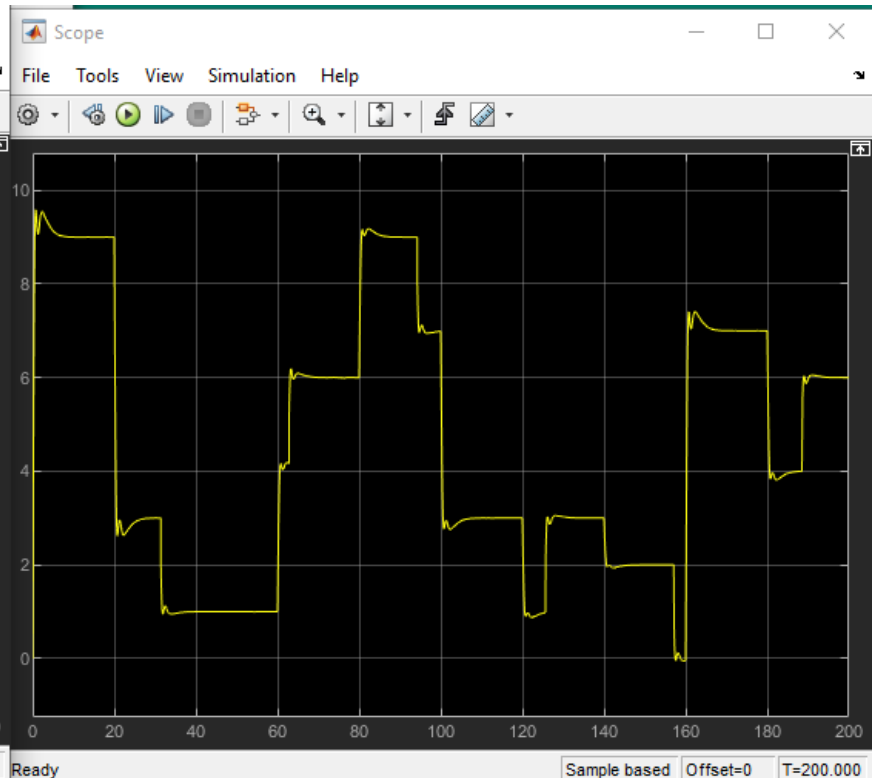
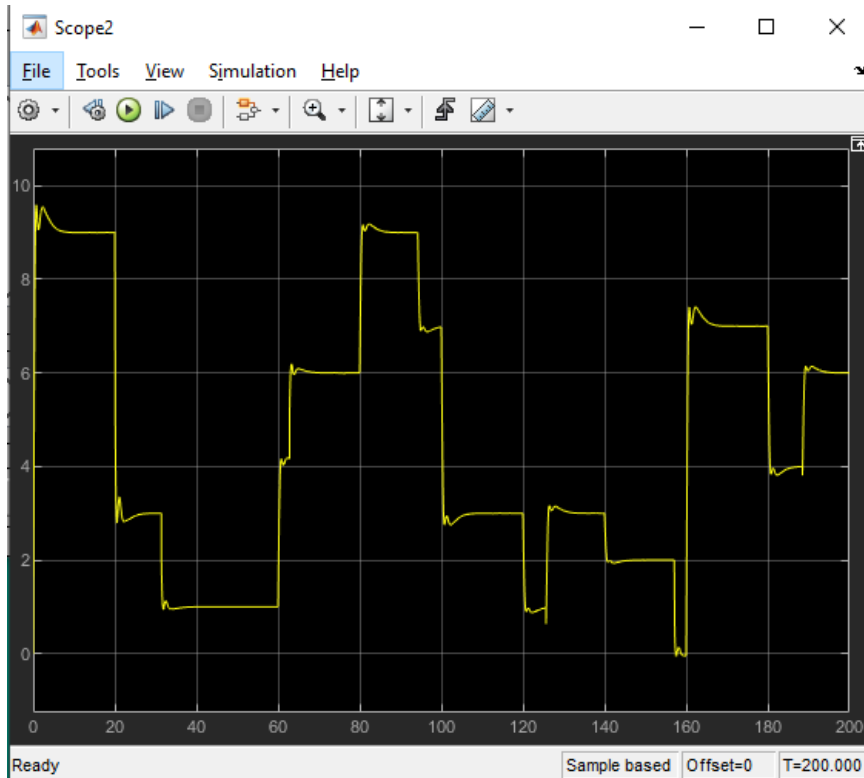
Name='output'
Range=[-1242.8384230369 287.358336077386]
NumMFs=27
MF1='out1mf1': 'linear', [0.000101914959891758 -1.25612855659263e-05
0.11708037416247 -9.46063433708844e-05]
MF2='out1mf2': 'linear', [3.34979198465795 1.75683809343899 1.01820002059126
-1.30013654929449]
MF3='out1mf3': 'linear', [5.63923871203552 2.3491047391658 1.01695957377993
-2.19699941423392]
MF4='out1mf4': 'linear', [0.000628800257345574 -7.74999395645182e-05
0.722588704345981 -0.000583706006352713]
MF5='out1mf5': 'linear', [3.35021839525185 1.47253256417681 1.01515847552451
-0.438801435087843]
MF6='out1mf6': 'linear', [5.6886582461767 2.51386876647213 1.01548440331765
-0.736729892848805]
MF7='out1mf7': 'linear', [0 0 0]
MF8='out1mf8': 'linear', [3.35008793734944 1.197431821767 1.00587926184732
0.334888745221749]
MF9='out1mf9': 'linear', [5.66740761624312 2.67786540389339 1.01772928708778
0.57493971592297]
MF10='out1mf10': 'linear', [0.000191504923812147 -2.35972969431231e-05
0.21998765389435 -0.000177730320878311]
MF11='out1mf11': 'linear', [3.31771674641097 2.15877816101763 0.998420457279278
0.254366455686917]
MF12='out1mf12': 'linear', [5.65670311053534 3.76575592500541 0.99873066953571
0.438330437385945]
MF13='out1mf13': 'linear', [0.00118159017245034 -0.000145577265421082
1.35738593770657 -0.00109672546386719]
MF14='out1mf14': 'linear', [3.36070857553286 2.15365168154732 0.996208581654961
0.0859351917697351]
MF15='out1mf15': 'linear', [5.68282571357321 3.76902000228647 0.996182894728844
0.152759494341996]
MF16='out1mf16': 'linear', [0 0 0]
MF17='out1mf17': 'linear', [3.02092274464107 2.25363721166654 1.00900612677583
-0.0653360490476449]
MF18='out1mf18': 'linear', [5.91352004220528 3.71123246570961 0.984747212719096
-0.103697069858556]
MF19='out1mf19': 'linear', [0 0 0]
MF20='out1mf20': 'linear', [3.32181639513016 1.53431482989023 0.983182165117317
0.441239518384574]
MF21='out1mf21': 'linear', [5.65139560812564 2.50427600118012 0.976691674999822
0.755018028783216]
MF22='out1mf22': 'linear', [0 0 0]
MF23='out1mf23': 'linear', [3.35259393256172 1.59639177175335 0.976439752814321
1.0573831301284]
MF24='out1mf24': 'linear', [5.6889485308561 2.37164761020652 0.977703530072551
1.79959183522747]
MF25='out1mf25': 'linear', [0 0 0]
MF26='out1mf26': 'linear', [3.62360508111227 1.22287434444501 1.32121484862684
1.68928439122316]
MF27='out1mf27': 'linear', [5.39377304008285 2.33055054221945 0.689784519877834
2.871102759715431]

[Rules]

1 1 1, 1 (1) : 1
1 1 2, 2 (1) : 1
1 1 3, 3 (1) : 1
1 2 1, 4 (1) : 1
1 2 2, 5 (1) : 1
1 2 3, 6 (1) : 1
1 3 1, 7 (1) : 1
1 3 2, 8 (1) : 1
1 3 3, 9 (1) : 1
2 1 1, 10 (1) : 1
2 1 2, 11 (1) : 1
2 1 3, 12 (1) : 1
2 2 1, 13 (1) : 1
2 2 2, 14 (1) : 1
2 2 3, 15 (1) : 1
2 3 1, 16 (1) : 1
2 3 2, 17 (1) : 1
2 3 3, 18 (1) : 1
3 1 1, 19 (1) : 1
3 1 2, 20 (1) : 1
3 1 3, 21 (1) : 1
3 2 1, 22 (1) : 1
3 2 2, 23 (1) : 1
3 2 3, 24 (1) : 1
3 3 1, 25 (1) : 1
3 3 2, 26 (1) : 1
3 3 3, 27 (1) : 1



MF9='out1mf9':'linear',
 [5.66740761624312 2.67786540389339 1.01772928708778 0.57493971592297]



Projektowanie regulatora rozmytego na bazie modelu sterowanego obiektu

Do zaprojektowania regulatora rozmytego powyższą metodą konieczne jest posiadanie modelu obiektu. Obiektywną identyfikację modeli można przeprowadzić wykorzystując czasowe **szeregi pomiarowe wejść i wyjścia obiektu**.

Identyfikacja składa się z dwóch etapów:

- I. **określenie istotnych wejść obiektu,**
- II. **określenie struktury i parametrów modelu.**

Identyfikacja modeli obiektów dynamicznych

Tylko w nielicznych przypadkach sterowany obiekt można uznać za statyczny i opisać zależnością

$$y(k) = F[u(k)]$$

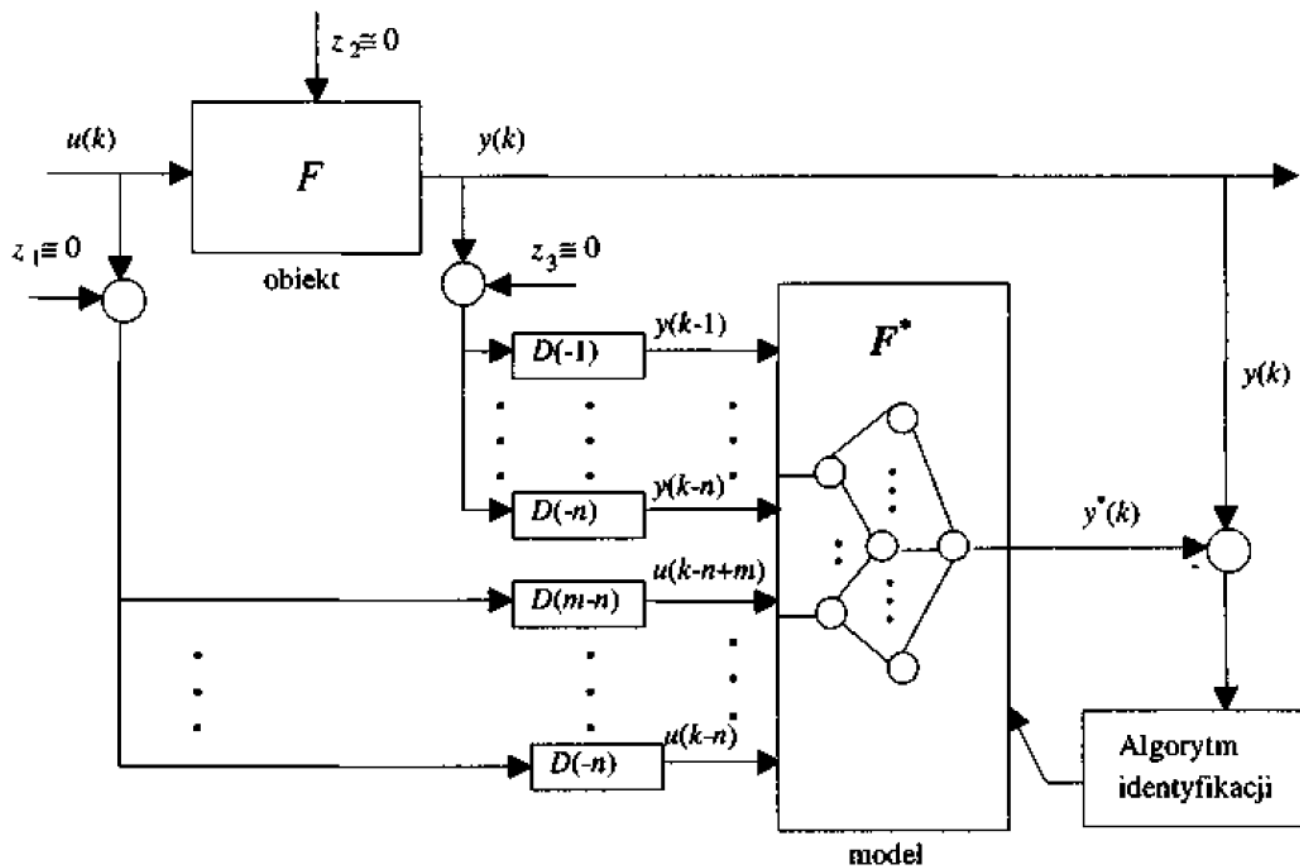
gdzie : u - wejście, y - wyjście obiektu, k - krok próbkowania.

W większości przypadków mamy do czynienia z obiektami, w których zależność wejście/wyjście ma formę nieliniowego odwzorowania F określonego wzorem

$$y(k+1) = F[y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)]$$

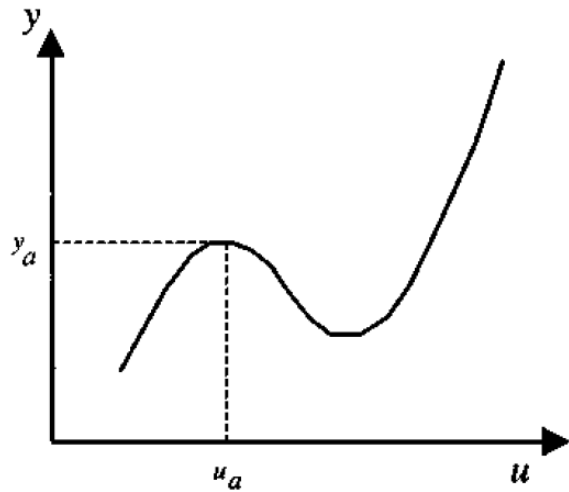
Wyjście y obiektu w chwili $(k+1)$ zależy od stanu wejścia u w „ m ” oraz stanu wyjścia w „ n ” poprzednich chwilach.

Schemat identyfikacji obiektu F

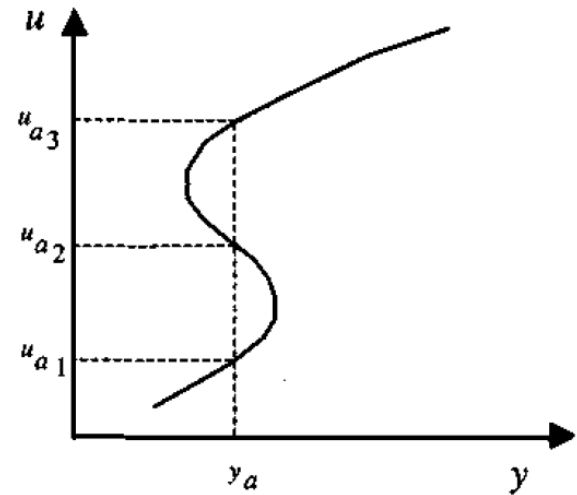


D oznacza tu operator opóźnienia

Problem jednoznaczności modelu inwersji obiektu



model F^*

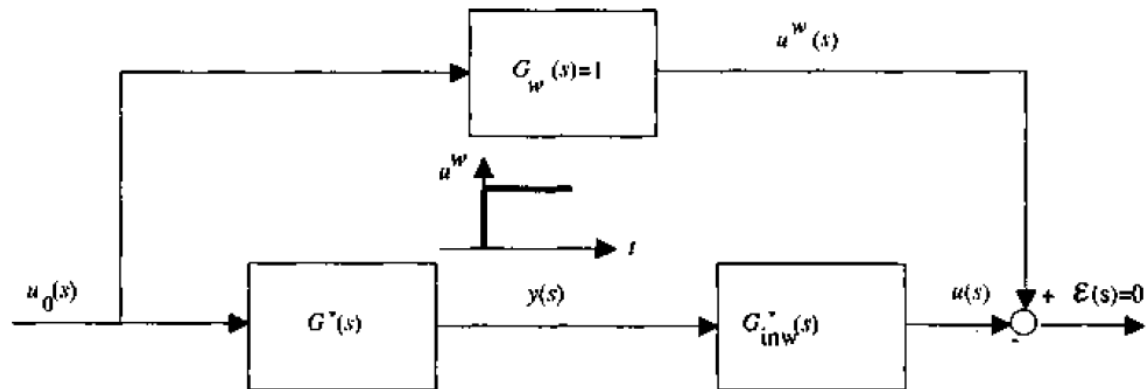


model $(F^*)^{-1}$



cel

$$G^*(s) G_{inw}^*(s) = 1$$



System z modelem wzorcowym

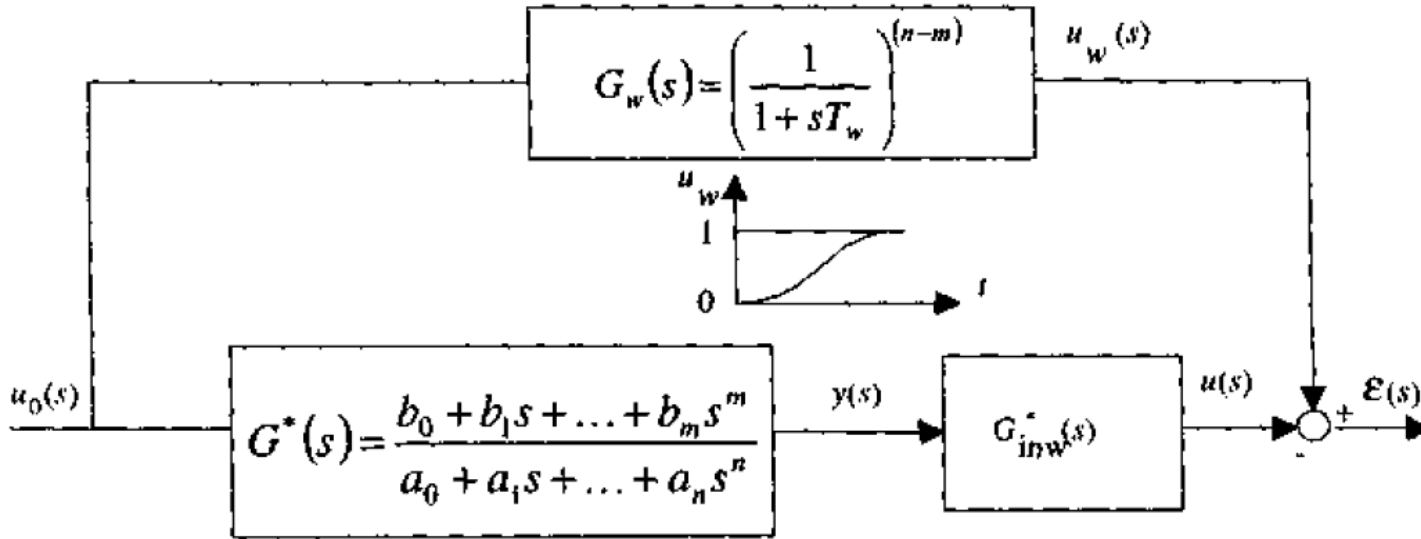
$$G^*(s) = \frac{b_0 + b_1s + \dots + b_ms^n}{a_0 + a_1s + \dots + a_ns^n} \quad \lim_{s \rightarrow +\infty} |G(s)| = 0$$



$$G_{inw}^*(s) = G_w(s) / G^*(s) = \frac{a_0 + a_1s + \dots + a_ns^n}{b_0 + b_1s + \dots + b_ms^n}$$

Inwersja taka jest nierealizowalna, bowiem obliczenie aktualnego wyjścia w(f) wymagałoby znajomości przyszłych stanów wejścia

Model właściwy



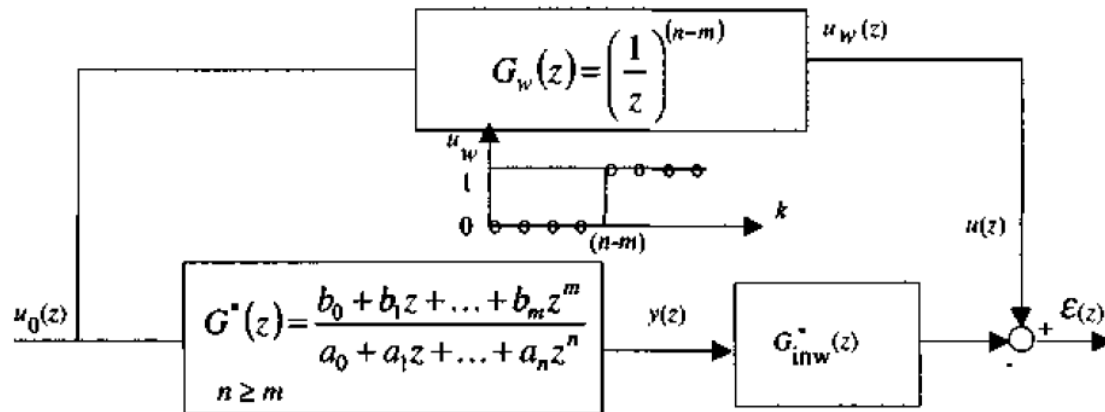
$$G_{inw}^*(s) = \frac{a_0 + a_1s + \dots + a_ns^n}{(b_0 + b_1s + \dots + b_ms^m)(1 + sT_w)^{(n-m)},}$$

$$n \geq m.$$

$$G_{inw}(s) = G_w(s) / G^*(s) = \frac{u(s)}{y(s)} = \frac{e^{sT_0}}{G_0^*(s)}$$

$$G_w(s) = \frac{e^{-sT_0}}{(1 + sT_w)^{(n-m)}}$$

Układy dyskretne



$$G_w(z) = \left(\frac{1}{z}\right)^{(n-m)}$$

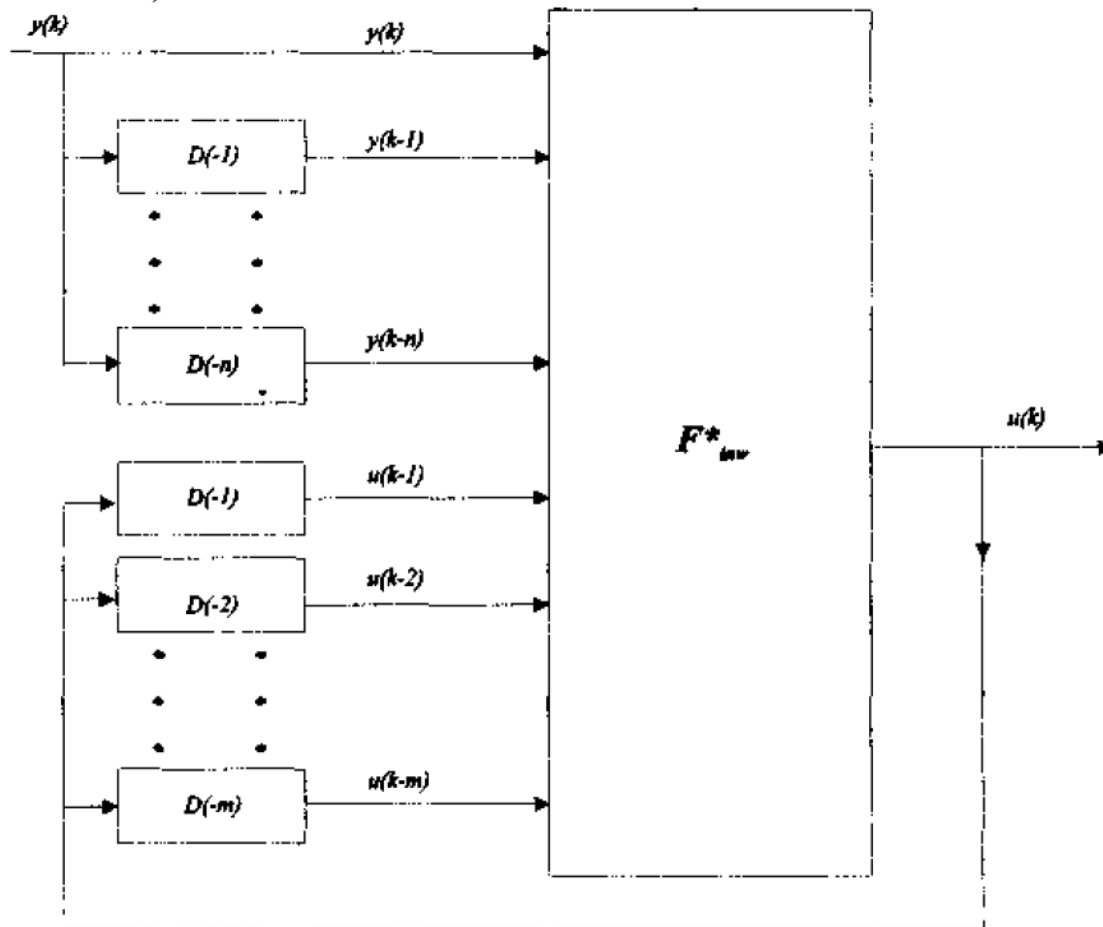
$$G^*(z) = \frac{y(z)}{u_0(z)} = \frac{b_0 + b_1 z + \dots + b_m z^m}{a_0 + a_1 z + \dots + a_n z^n} = \frac{b_0 z^{-n} + \dots + b_{m-1} z^{-n+m+1} + b_m z^{-n+m}}{a_0 z^{-n} + \dots + a_{n-1} z^{-1} + a_n}$$

$$a_n y(k) + \dots + a_1 y(k-n+1) + a_0 y(k-n) = b_m u_0(k-n+m) + \dots + b_1 u_0(k-n+1) + b_0 u_0(k-n)$$

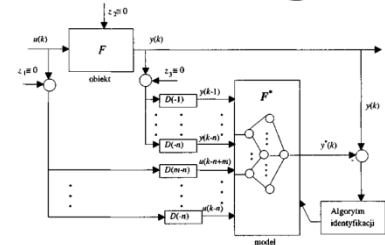
$$G_{inw}^*(z) = \frac{u(z)}{y(z)} = \frac{a_0 + a_1 z + \dots + a_n z^n}{b_0 z^{n-m} + b_1 z^{n-m+1} + \dots + b_m z^m} = \frac{a_0 z^{-n} + \dots + a_{n-1} z^{-1} + a_n}{b_0 z^{-m} + \dots + b_{m-1} z^{-1} + b_m}$$

$$b_0 u(k-m) + \dots + b_{m-1} u(k-1) + b_m u(k) = a_0 y(k-n) + \dots + a_{n-1} y(k-1) + a_n y(k)$$

Schemat inwersji modelu



Model inwersji Sugeno zerowego rzędu



Model obiektu

JEŚLI $(u_0(k) = A_1) I, \dots, I (u_0(k-n) = A_{n+1}) I (y(k-1) = B_1) I, \dots, I (y(k-n) = B_n)$
TO $y(k) = c$

gdzie A_1, \dots, A_{n+1} i B_1, \dots, B_n są zbiorami rozmytymi, a c jest singletonem

Model inwersji

JEŚLI $(u_0(k-1) = A_2) I, \dots, I (u_0(k-n) = A_{n+1}) I (y(k) = C) I (y(k-1) = B_1)$
 $I, \dots, I (y(k-n) = B_n)$ *TO* $u_0(k) = a_1,$

Model skokowy

Jeżeli zmienna wyjściowa $y(k)$ zależna jest bezpośrednio od zmiennej $u_0(k)$, oznacza to, że wejście u_0 w chwili k oddziałuje na wyjście y bezpośrednio, bez żadnego opóźnienia. Modele te nazywane będą dalej, **modelami skokowymi**, bowiem zdolne są one do natychmiastowego przeniesienia skoku wejścia na wyjście.

Jeżeli rozmyty model obiektu jest standardowym modelem Sugeno zerowego rzędu, trójkątnymi funkcjami przynależności zbiorów rozmytych w przesłankach pokrywającymi się do jedności ($\sum \mu = 1$), reguły mają formę koniunkcyjną, do operacji I (AND) stosowany jest operator PROD, defuzyfikację realizujemy metodą singletonów,

$$y = \frac{\sum_{j=1}^K \beta_j c_j}{\sum_{j=1}^K \beta_j}$$

gdzie β_j oznacza stopień spełnienia przesłanki j -tej reguły, K - jest numerem reguły w bazie,

Model skokowy (cd)

model rozmyty jest jednoznaczny względem zmiennej wyjściowej $y(k)$ i jest modelem skokowym zawierającym reguły typu

$$\text{JEŚLI } (u_0(k) = A_1) I, \dots, I (u_0(k-n) = A_{n+1}) I (y(k-1) = B_1) I, \dots, I (y(k-n) = B_n)$$

$$\text{TO } y(k) = c$$

W rezultacie otrzymujemy model rozmyty z regułami typu

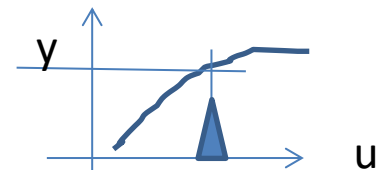
$$\text{JEŚLI } (u_0(k-1) = A_2) I, \dots, I (u_0(k-n) = A_{n+1}) I (y(k) = C) I (y(k-1) = B_1)$$

$$I, \dots, I (y(k-n) = B_n) \text{ TO } u_0(k) = a_1,$$

gdzie: a - singleton o wartości modalnej równej wartości modalnej zbioru A_1 ,
 C - trójkątny zbiór rozmyty o wartości modalnej równej wartości modalnej singletona c .

Jeżeli u jest (około 5) to $y = 15$

Jeżeli y jest (singleton 15) to $u = 5$

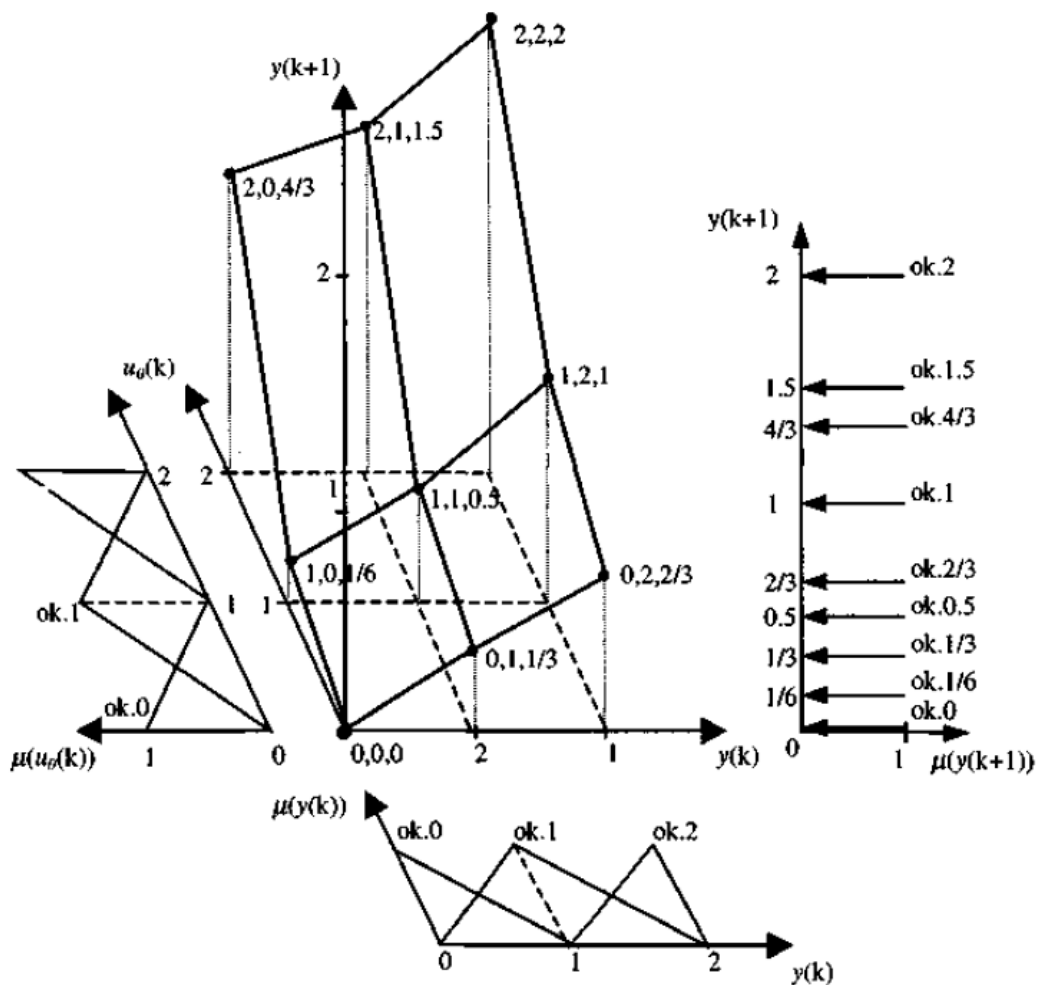


Przykład

Dany jest rozmyty model obiektu $y(k+1)=F*[u_0(k),y(k)]$ z bazą reguł

- R1: JEŚLI ($u_0(k)$ około 0) I ($y(k)$ około 0) TO ($y(k+1)$ około 0)*
- R2: JEŚLI ($u_0(k)$ około 1) I ($y(k)$ około 0) TO ($y(k+1)$ około 1/3)*
- R3: JEŚLI ($u_0(k)$ około 2) I ($y(k)$ około 0) TO ($y(k+1)$ około 4/3)*
- R4: JEŚLI ($u_0(k)$ około 0) I ($y(k)$ około 1) TO ($y(k+1)$ około 1/6)*
- R5: JEŚLI ($u_0(k)$ około 1) I ($y(k)$ około 1) TO ($y(k+1)$ około 0.5)*
- R6: JEŚLI ($u_0(k)$ około 2) I ($y(k)$ około 1) TO ($y(k+1)$ około 1.5)*
- R7: JEŚLI ($u_0(k)$ około 0) I ($y(k)$ około 2) TO ($y(k+1)$ około 2/3)*
- R8: JEŚLI ($u_0(k)$ około 1) I ($y(k)$ około 2) TO ($y(k+1)$ około 1)*
- R9: JEŚLI ($u_0(k)$ około 2) I ($y(k)$ około 2) TO ($y(k+1)$ około 2)*

Przykład (cd)




Funkcje przynależności wejść i wyjść oraz powierzchnia modelu rozmytego z bazą reguł. Liczby przy węzłach powierzchni podają ich współrzędne

Przykład (cd)

W drugim etapie tworzenia inwersji przesuwamy wektor $u_0(k) = u_0(k)$ o wartość $(n-m)$ równą w tym wypadku 1. W rezultacie model o charakterze nieskokowym przekształcony zostaje w model skokowy

R1: JEŚLI ($u_0(k+1)$ około 0) I ($y(k)$ około 0) TO ($y(k+1)$ około 0)
⋮
R9: JEŚLI ($u_0(k+1)$ około 2) I ($y(k)$ około 2) TO ($y(k+1)$ około 2)



W trzecim etapie, w skokowym modelu zamieniamy miejscem konkluzje zawierające zmienną $y(k+1)$ z przesłankami składowymi zawierającymi $u_0(k+1)$. W rezultacie otrzymujemy bazę reguł

- R1: JEŚLI*** ($y(k+1)$ około 0) ***I*** ($y(k)$ około 0) ***TO*** ($u(k+1)$ około 0)
R2: JEŚLI ($y(k+1)$ około 1/3) ***I*** ($y(k)$ około 0) ***TO*** ($u(k+1)$ około 1)
R3: JEŚLI ($y(k+1)$ około 4/3) ***I*** ($y(k)$ około 0) ***TO*** ($u(k+1)$ około 2)
R4: JEŚLI ($y(k+1)$ około 1/6) ***I*** ($y(k)$ około 1) ***TO*** ($u(k+1)$ około 0)
R5: JEŚLI ($y(k+1)$ około 0.5) ***I*** ($y(k)$ około 1) ***TO*** ($u(k+1)$ około 1)
R6: JEŚLI ($y(k+1)$ około 1.5) ***I*** ($y(k)$ około 1) ***TO*** ($u(k+1)$ około 2)
R7: JEŚLI ($y(k+1)$ około 2/3) ***I*** ($y(k)$ około 2) ***TO*** ($u(k+1)$ około 0)
R8: JEŚLI ($y(k+1)$ około 1) ***I*** ($y(k)$ około 2) ***TO*** ($u(k+1)$ około 1)
R9: JEŚLI ($y(k+1)$ około 2) ***I*** ($y(k)$ około 2) ***TO*** ($u(k+1)$ około 2)

baza reguł bezpośrednio inwersji modelu jest niekompletna. Zawiera ona bowiem jedynie informacje o wartościach wyjść dla 9 węzłów przestrzeni wejść - taka była bowiem ilość reguł modelu pierwotnego. Niekompletna baza reguł uniemożliwia obliczenie wyjść w sektorach przestrzeni wejść, których węzły nie są w pełni pokryte regułami. Dlatego konieczne jest określenie stanów wyjścia $u(k+1)$ dla niepokrytych węzłów, a więc rozszerzenie bazy reguł do bazy kompletnej zawierającej $9 \times 3 = 27$ reguł.

Przykład (cd)

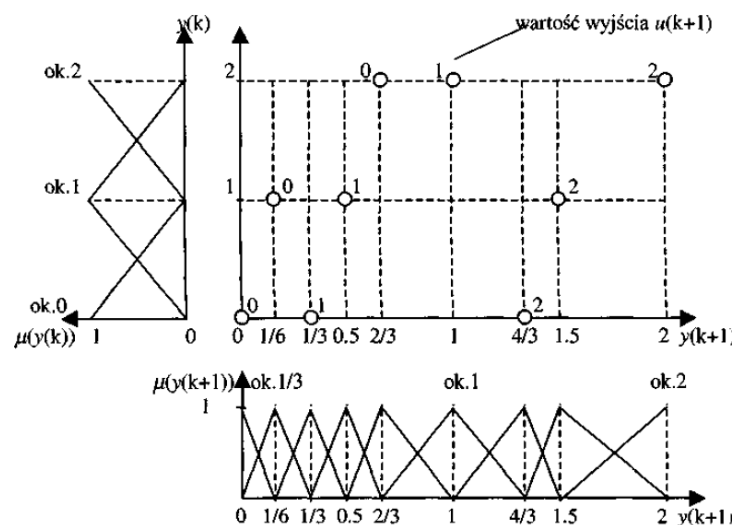
R1: *JESLI* ($\tilde{y}(k+1)$ około 0) *I* ($\tilde{y}(k)$ około 0) *TO* ($\tilde{u}(k+1)$ około a_1)

R2: *JESLI* ($y(k+1)$ około 0.5) *I* ($y(k)$ około 0) *TO* ($u(k+1)$ około a_2)

R3: *JESLI* ($y(k+1)$ około 1) *I* ($y(k)$ około 0) *TO* ($u(k+1)$ około a_3)

⋮

R27: *JESLI* ($y(k+1)$ około 6) *I* ($y(k)$ około 2) *TO* ($u(k+1)$ około a_{27})



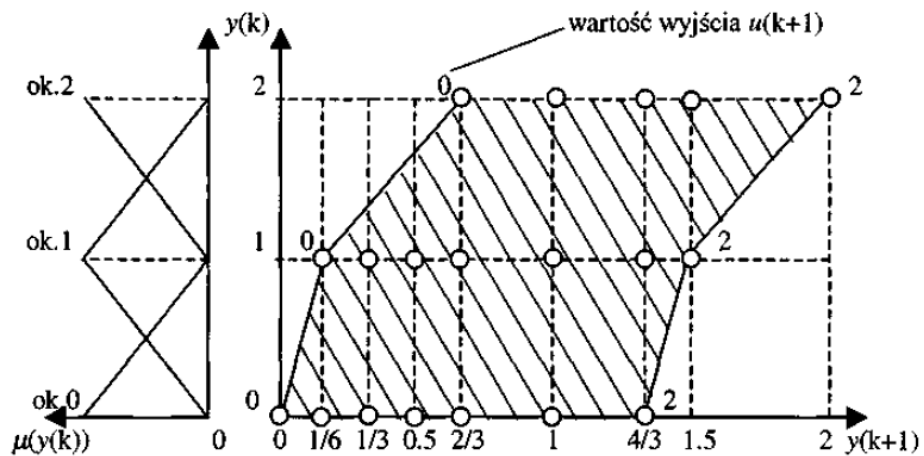
Przykład (cd)

Przesłanki bazy reguł zawierają wszystkie 27 kombinacji zbiorów rozmytych wejść. Parametry a , w konkluzjach są wartościami singletonów.

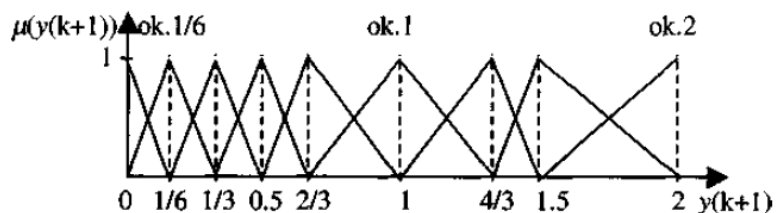
singletonów jest znanych dzięki bazie reguł bezpośredniej inwersji modelu. Np. wartość $a_1 = 0$, $a_{27} = 2$. Wartości pozostałych singletonów mogą być określone dwoma metodami:

- I. przez przekształcenie kompletnej bazy reguł w rozmytą sieć neuronową z funkcjami przynależności wejść jak na rysunku oraz z singletonowymi funkcjami przynależności wyjścia i nastrojenie tej sieci metodą wstecznej propagacji błędów na bazie próbek pomiarowych wejść i wyjść obiektu lub też obliczenie wartości a , metodą najmniejszych kwadratów,
- II. jak wyżej, z tą różnicą, że strojenie przeprowadzane jest próbkami wejść wyjścia wygenerowanymi przez pierwotny model rozmyty.

Przykład (cd)



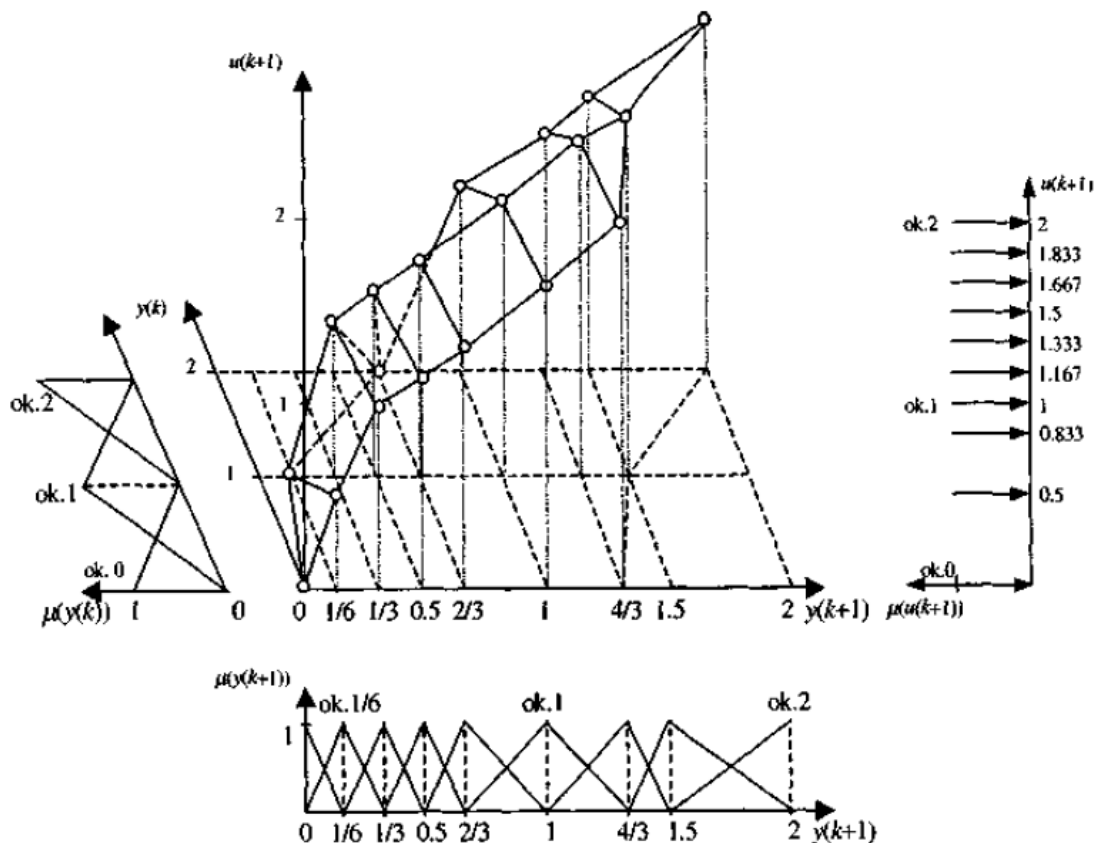
Realna (zakreskowana) przestrzeń wejść inwersji modelu rozmytego, dla której możliwe jest obliczenie wyjścia na bazie próbek pomiarowych obiektu.



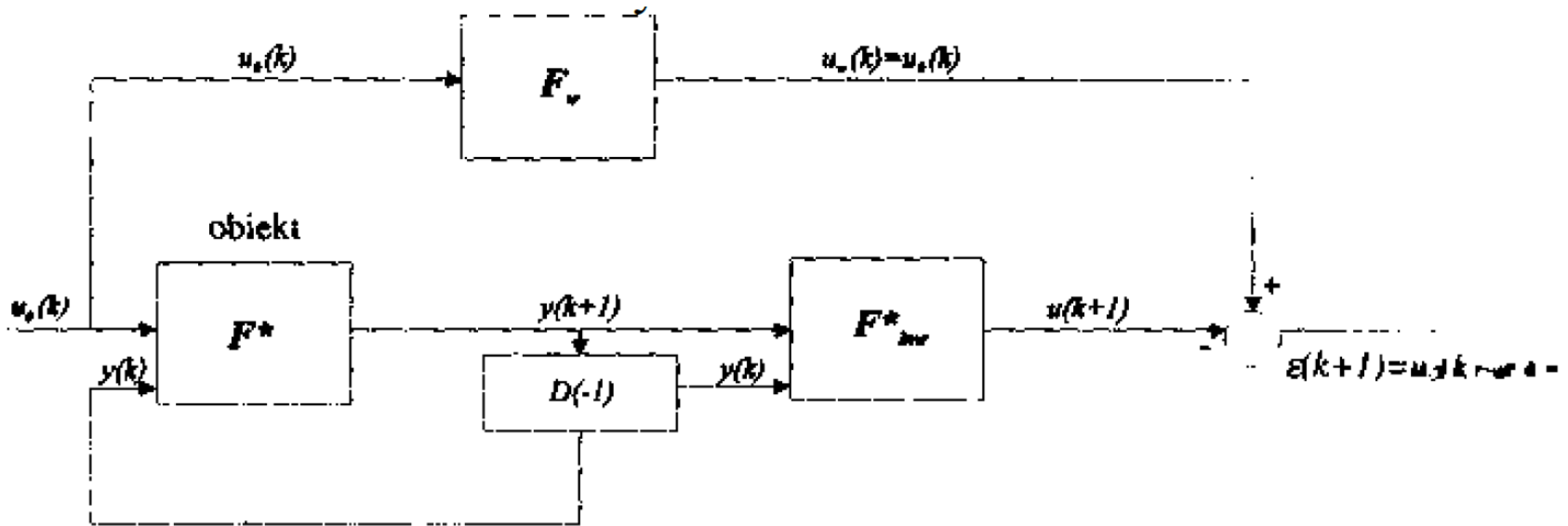
$$0 \leq u_0(k) \leq 2, \quad 0 \leq y(k) \leq 2, \quad 0 \leq y(k+1) \leq 2$$

Przykład (cd)

Powierzchnia i funkcje przynależności inwersji modelu



Schemat układu z modelem inwersji



Układ złożony z modelu F^* , inwersji F_{inv} , modelu wzorcowego F_w

$$u(k) = F_{inv}^* [y(k), y(k-1), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-n)]$$

Strojenie parametrów regulatora rozmytego PID

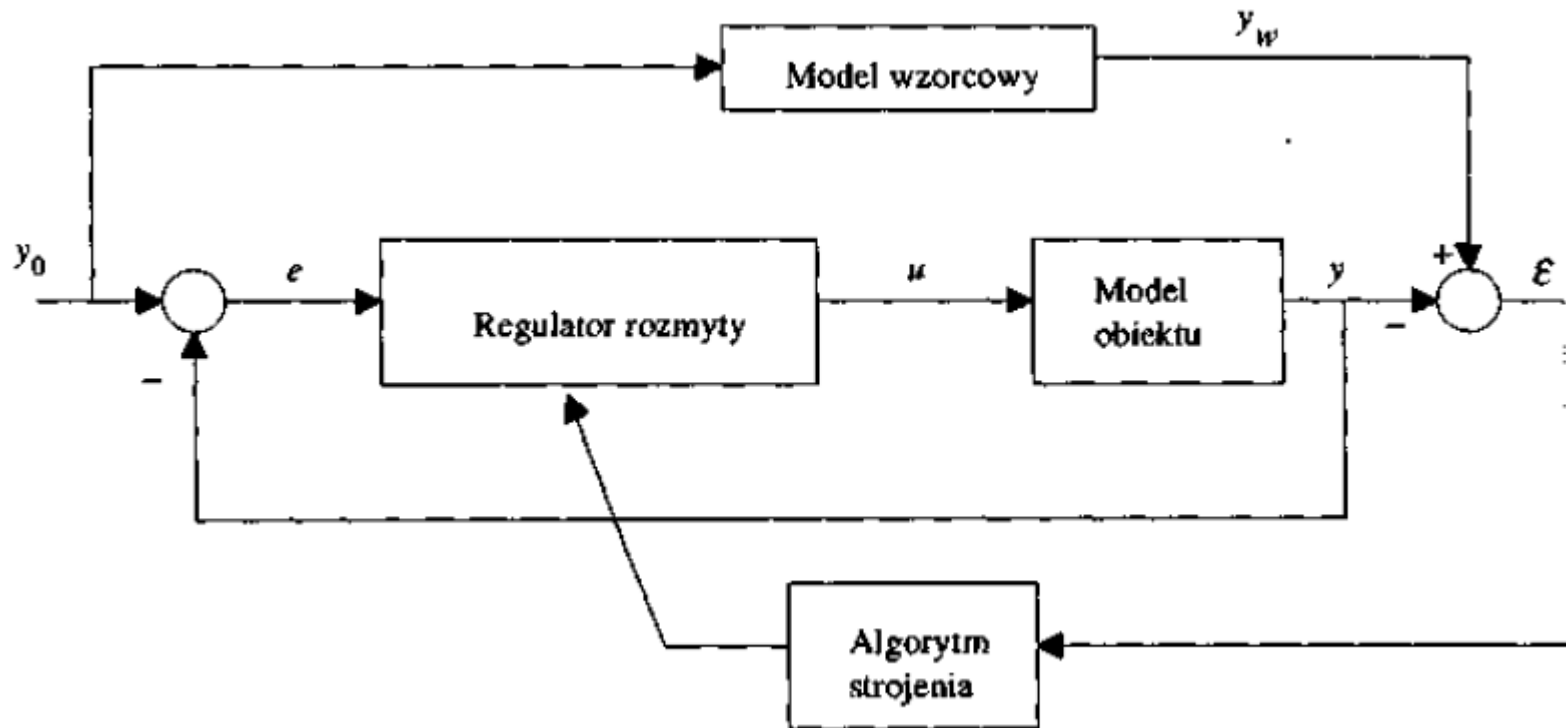
W przypadku wielu obiektów znane są rodzaje regulatorów stosowane do ich sterowania i sprawdzone w praktyce (PID, PI, PD, P, PDD).

Ponieważ kryteria jakości sterowania zależne są w większości przypadków nieliniowo od parametrów regulatora, zwykle możemy podwyższać ich wartość stosując zamiast regulatorów liniowych rozmyte regulatory (PID, PI, PD, P, PDD) o zmiennych wzmocnieniach, realizujące nieliniowe odwzorowania wejścia/wyjście. Rozmyte regulatory PID są często stosowane w praktyce.

Jeżeli typ regulatora rozmytego jest znany (np.: PID), można przyjąć odpowiednią ilość zbiorów rozmytych dla każdego wejścia i wyjścia regulatora (przy wyższej ilości zbiorów można uzyskać większą krzywiznę powierzchni sterowania) i nastroić jego parametry. Jeżeli nie wiemy jaki typ regulatora byłby dla danego obiektu odpowiedni, można zastosować eksperymentalną metodę prób i błędów.

Strojenie parametrów regulatora rozmytego na podstawie modelu

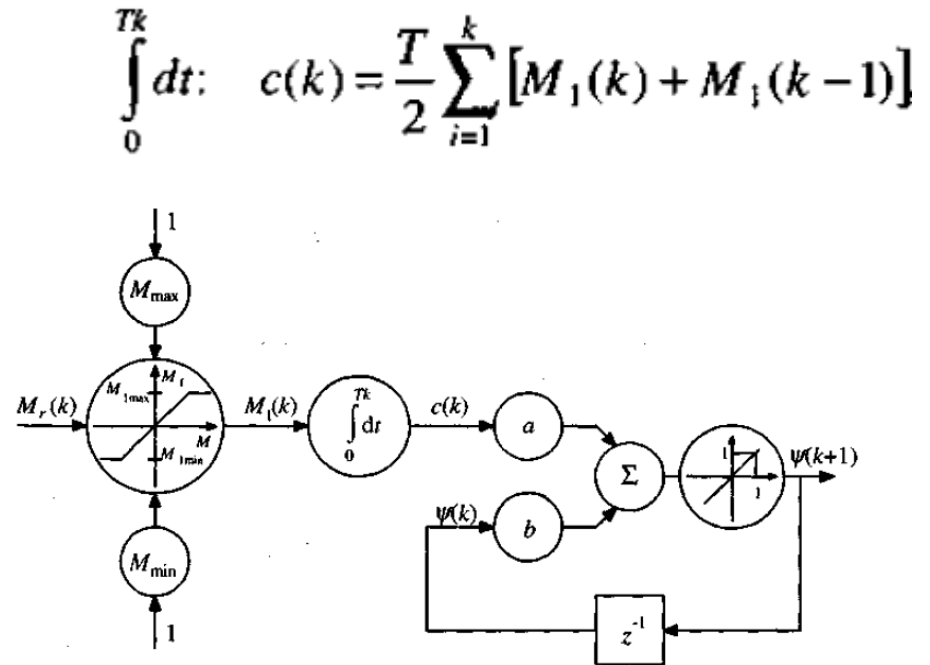
Dysponując modelem matematycznym obiektu, który niekoniecznie musi być modelem rozmytym, możemy przeprowadzić strojenie regulatora rozmytego metodą wstecznej propagacji błędu w układzie z modelem odniesienia (wzorcowym), określającym nasze wymagania (życzenia) dotyczące jakości sterowania.



Przykład

Model matematyczny ruchu obrotowego pojazdu podwodnego, określający zależności między kątem kursowym ψ pojazdu a sygnałem sterującym M_r (wartość momentu obrotowego generowanego przez pędniki boczne pojazdu), obliczanym przez regulator rozmyty, ma formę rekurencyjnej sieci neuronowej

$$G^*(s) = \frac{\psi(s)}{M_r(s)} = \frac{k_o}{s(1 + sT_o)}$$



Przykład (cd)

R1: JEŚLI ($e_p = p$) I ($e_l = p$) I ($e_D = p$) TO [$M_r = M_{max}(k_{1p} \text{ LUB } - k_{1n})$]

R2: JEŚLI ($e_p = p$) I ($e_l = p$) I ($e_D = n$) TO [$M_r = M_{max}(k_{2p} \text{ LUB } - k_{2n})$]

R3: JEŚLI ($e_p = p$) I ($e_l = n$) I ($e_D = p$) TO [$M_r = M_{max}(k_{3p} \text{ LUB } - k_{3n})$]

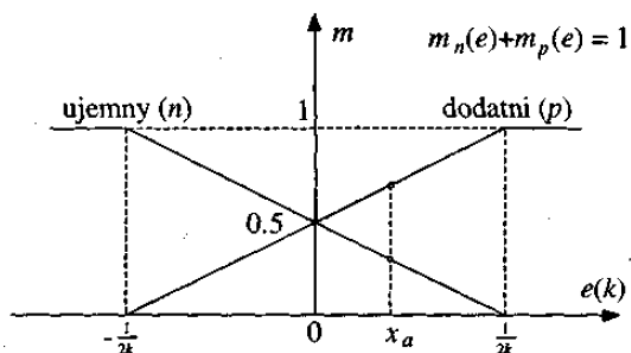
R4: JEŚLI ($e_p = p$) I ($e_l = n$) I ($e_D = n$) TO [$M_r = M_{max}(k_{4p} \text{ LUB } - k_{4n})$]

R5: JEŚLI ($e_p = n$) I ($e_l = p$) I ($e_D = p$) TO [$M_r = M_{max}(k_{5p} \text{ LUB } - k_{5n})$]

R6: JEŚLI ($e_p = n$) I ($e_l = p$) I ($e_D = n$) TO [$M_r = M_{max}(k_{6p} \text{ LUB } - k_{6n})$]

R7: JEŚLI ($e_p = n$) I ($e_l = n$) I ($e_D = p$) TO [$M_r = M_{max}(k_{7p} \text{ LUB } - k_{7n})$]

R8: JEŚLI ($e_p = n$) I ($e_l = n$) I ($e_D = n$) TO [$M_r = M_{max}(k_{8p} \text{ LUB } - k_{8n})$]



Defuzyfikacja

$$M_r(k) = \left(\frac{f_p - f_n}{f_p + f_n} \right) M_{max}$$

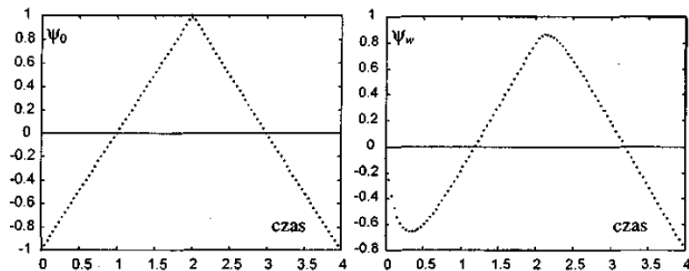
gdzie: f_p - wynikowy stopień aktywizacji konkluzji $M_r = M_{max}$

f_n - wynikowy stopień aktywizacji konkluzji $M_r = -M_{max}$

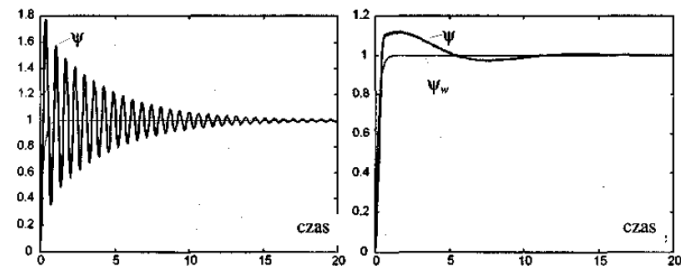
Przykład (cd)

Regulator rozmyty posiada ogółem 11 parametrów, które mogą podlegać strojeniu (K_p , K_I , K_D , k_{1p} - k_{8p}).

Do strojenia regulatora użyto trójkątnego sygnału $\psi_0(t)$ i odpowiadającego mu sygnału wzorcowego $\psi_w(t)$. Strojenie regulatora odbywało się metodą wstecznej propagacji błędów $e = \psi_w(t) - \psi(t)$ według struktury z modelem wzorcowym.



Sygnały zadający $\psi_0(t)$ i wzorcowy $\psi_w(t)$ użyte do strojenia regulatora PID.



a) przed strojeniem

b) po strojeniu

Odpowiedź skokowa ψ układu sterowania z rozmytym regulatorem neuronowym oraz odpowiedź ψ_w modelu wzorcowego.

Przykład 2

Obiekt właściwy

$$y(k+1) = 0.83333 y(k) + 1.5 u(k) - 1.33333 u(k-1)$$

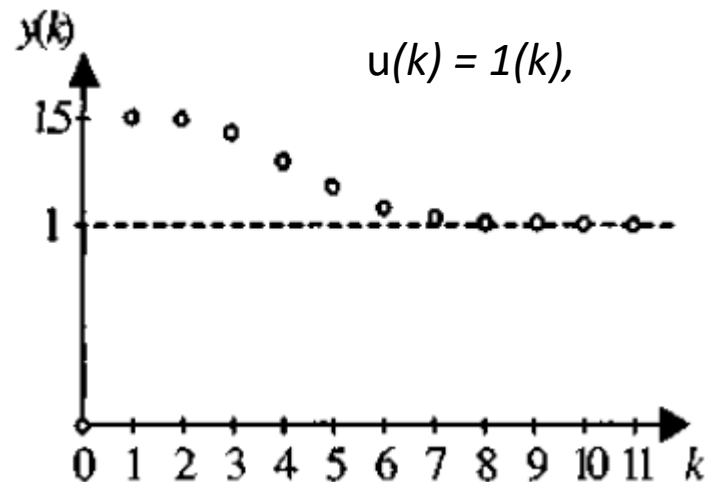
dla $-1 \leq u \leq 1$,

$$y(k+1) = 0.83333 y(k) + 0.166667$$

dla $u > 1$,

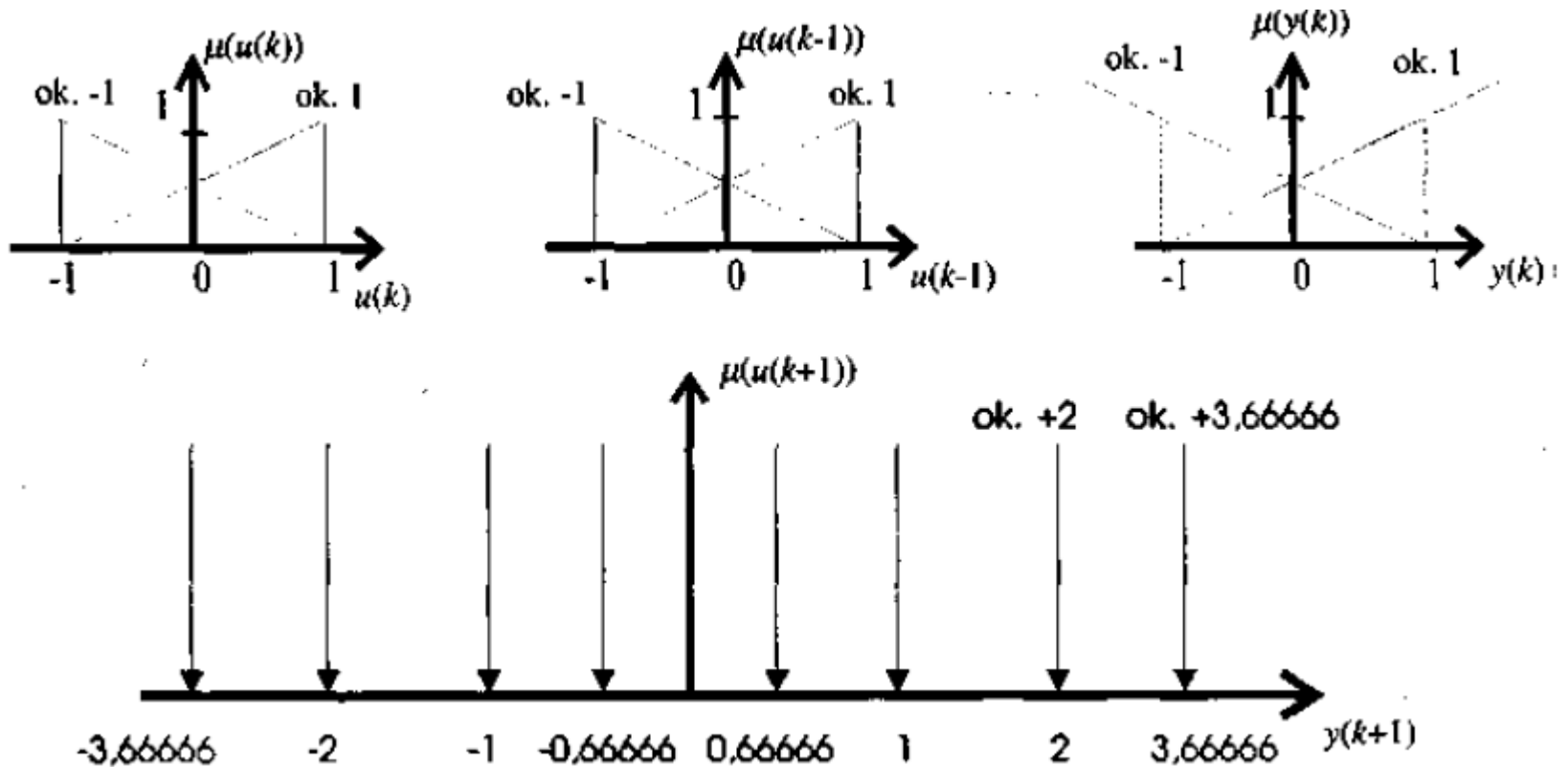
$$y(k+1) = 0.83333 y(k) - 2.83333$$

dla $u < -1$.



Przykład 2 (cd)

Model F* rozmyty



Przykład 2 (cd)

R1÷4: *JESLI* [$u(k-1) = \text{ok. } -1$] *I* [$u(k) = \text{ok. } \dots$] *I* [$y(k) = \text{ok. } \dots$]
TO [$y(k+1) = \text{ok. } \dots$]

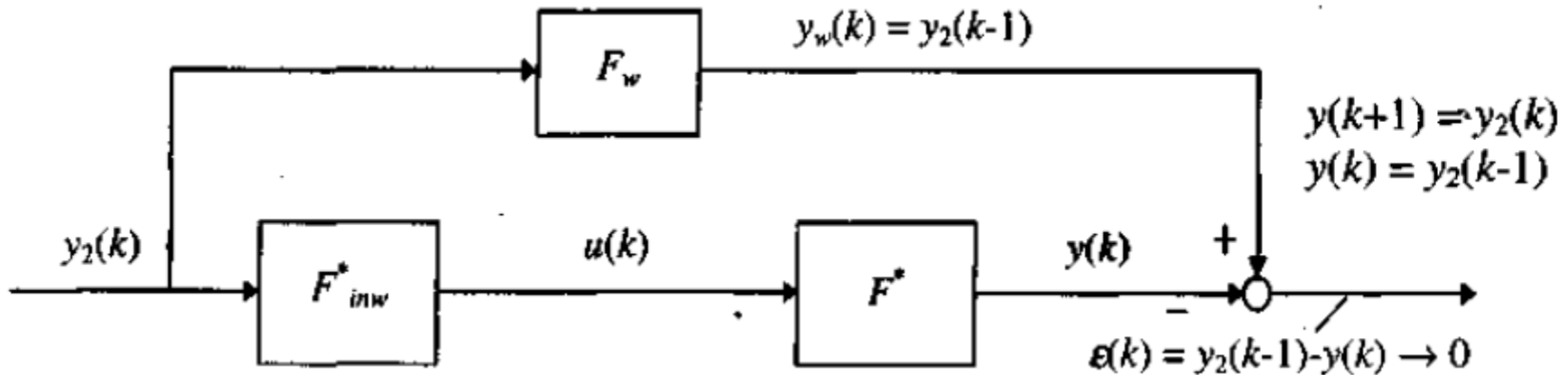
$y(k)$ $u(k)$	około -1	około +1
około -1	około -1	około +0.66666
około +1	około +2	około +3.66666

R5÷6: *JESLI* [$u(k-1) = \text{ok. } 1$] *I* [$u(k) = \text{ok. } \dots$] *I* [$y(k) = \text{ok. } \dots$]
TO [$y(k+1) = \text{ok. } \dots$]

$y(k)$ $u(k)$	około -1	około +1
około -1	około -3.66666	około -2
około +1	około -0.66666	około +1

Przykład 2 (cd)

Schemat układu strojenia inwersji F^*_{inw} modelu



Reguła pierwotna:

R2: JEŚLI [$u(k-1) = \text{ok. -1}$] I [$u(k) = \text{ok. -1}$] I [$y(k) = \text{ok. +1}$]
TO [$y(k+1) = 0.66666$].

Reguła po zamianie zmiennych:

R2*: JEŚLI [$u(k-1) = \text{ok. -1}$] I [$u(k) = \text{ok. -1}$] I [$y_2(k-1) = \text{ok. +1}$]
TO [$y_2(k) = 0.66666$].

Reguła inwersji modelu odpowiadająca regule R2:

R*_{inw}: JEŚLI [$u(k-1) = \text{ok. -1}$] I [$y_2(k-1) = \text{ok. +1}$] I [$y_2(k) = 0.66666$]
TO [$u(k) = \text{ok. -1}$].

Przykład 2 (cd)

W wyniku bezpośredniego odwracania 8 reguł modelu F^* uzyskujemy 8 reguł inwersji przedstawionych w tabeli

$R1 \div 8$: *JESLI* [$u(k-1) = \text{ok. } -1$] *I* [$y_2(k-1) = \text{ok. } \dots$] *I* [$y_2(k) = \text{ok. } \dots$]
TO [$u(k) = \text{ok. } \dots$]

$y_2(k)$	ok.	ok.	ok.	ok.	ok.	ok.	ok.	ok.
$y_2(k-1)$	-3.6666	-2	-1	-0.6666	+0.6666	+1	+2	+3.6666
ok. -1			ok. -1				ok. +1	
ok. +1					ok. -1			ok. +1

Reguły $R1 \div R8$ inwersji modelu.

$u(k)$

$R9 \div 16$: *JESLI* [$u(k-1) = \text{ok. } +1$] *I* [$y_2(k-1) = \text{ok. } \dots$] *I* [$y_2(k) = \text{ok. } \dots$]
TO [$u(k) = \text{ok. } \dots$]

$y_2(k)$	ok.	ok.	ok.	ok.	ok.	ok.	ok.	ok.
$y_2(k-1)$	-3.6666	-2	-1	-0.6666	+0.6666	+1	+2	+3.6666
ok. -1	ok. -1			ok. +1				
ok. +1		ok. -1			ok. +1			

$u(k)$

Reguły $R9 \div R16$ inwersji modelu.

Przykład 2 (cd)

Model uzupełniony

$R1 \div 16$: *JESLI* [$u(k-1) = \text{ok. } -1$] *I* [$y_2(k-1) = \text{ok. } \dots$] *I* [$y_2(k) = \text{ok. } \dots$]
TO [$u(k) = \text{ok. } \dots$]

$y_2(k)$	ok.	ok.	ok.	ok.	ok.	ok.	ok.	ok.
$y_2(k-1)$	-3.6666	-2	-1	-0.6666	+0.6666	+1	+2	+3.6666
ok. -1	ok. -1	ok. -1	ok. -1	ok.	ok.	ok.	ok. +1	ok. +1
				-0.7777	+0.1111	+0.3333		
ok. +1	ok. -1	ok. -1	ok. -1	ok. -1	ok. -1	ok.	ok.	ok. +1
						-0.7777	-0.1111	

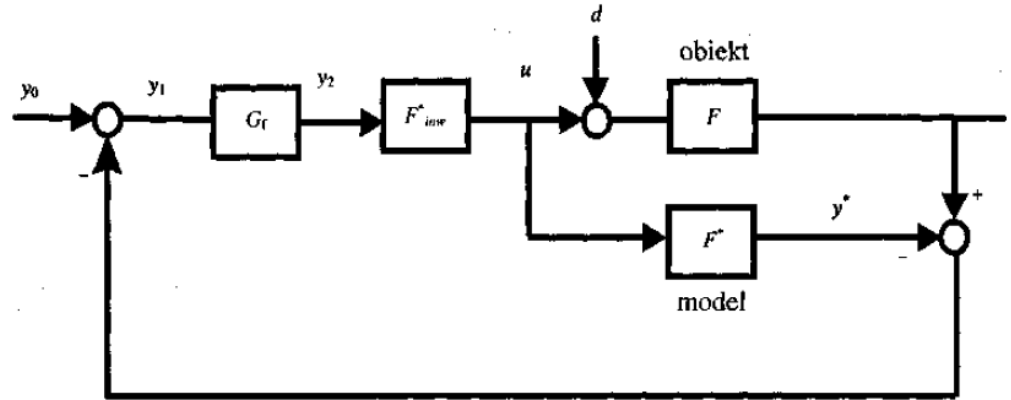
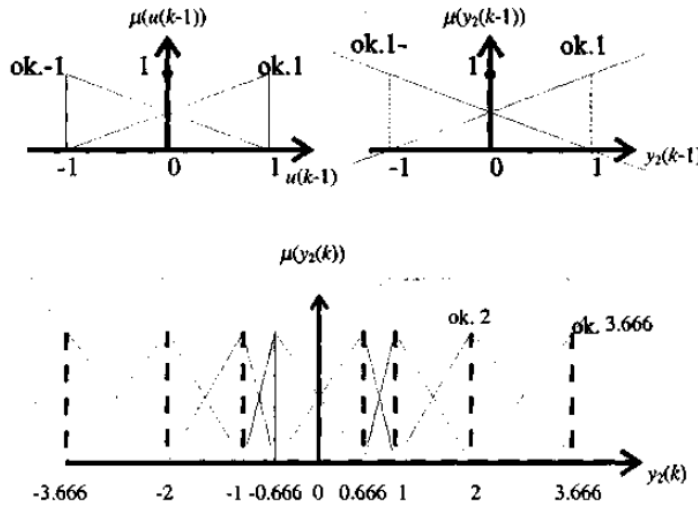
Reguły $R1 \div R16$ inwersji modelu F_{inw}^* .

$R17 \div 32$: *JESLI* [$u(k-1) = \text{ok. } +1$] *I* [$y_2(k-1) = \text{ok. } \dots$] *I* [$y_2(k) = \text{ok. } \dots$]
TO [$u(k) = \text{ok. } \dots$]

$y_2(k)$	ok.	ok.	ok.	ok.	ok.	ok.	ok.	ok.
$y_2(k-1)$	-3.6666	-2	-1	-0.6666	+0.6666	+1	+2	+3.6666
ok. -1	ok. -1	ok.	ok.	ok. +1	ok. +1	ok. +1	ok. +1	ok. +1
		+0.1111	+0.7777					
ok. +1	ok. -1	ok. -1	ok.	ok.	ok.	ok. +1	ok. +1	ok. +1
			-0.3333	-0.1111	+0.7777			

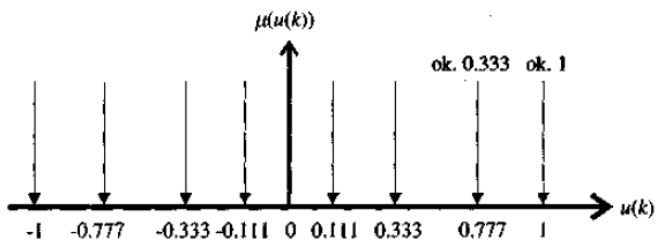
Reguły $R17 \div R32$ inwersji modelu F_{inw}^* .

Przykład 2 (cd)



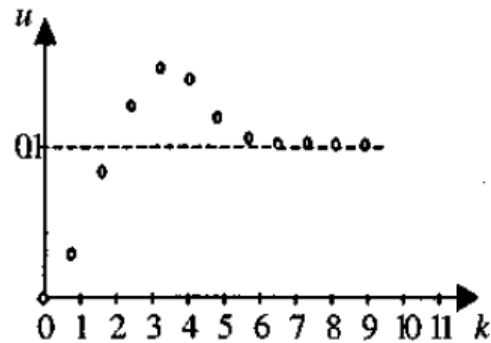
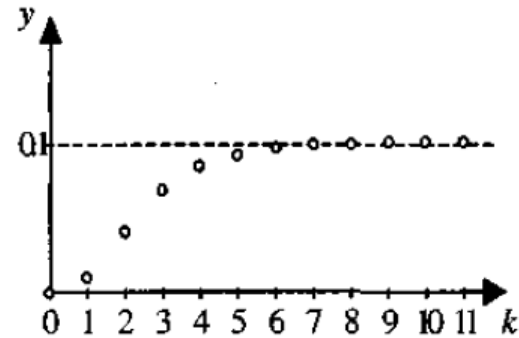
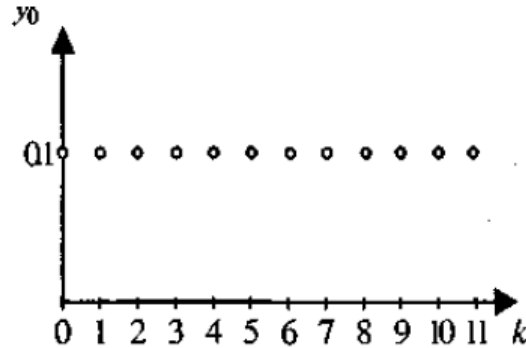
Dobrym zostaje filtr inercyjny o stałej czasowej $T_f=0.3$,

$$G_f(z) = \frac{0.28347}{z - 0.71653} = \frac{0.28347 z^{-1}}{1 - 0.71653 z^{-1}}$$



Przykład 2(cd)

Odpowiedź y układu na mały skok wartości zadanej $y_0 = 0.1$ (k), przy zerowych warunkach początkowych, oraz sygnał sterujący $u(k)$ wygenerowany przez inwersję F_{inv} .



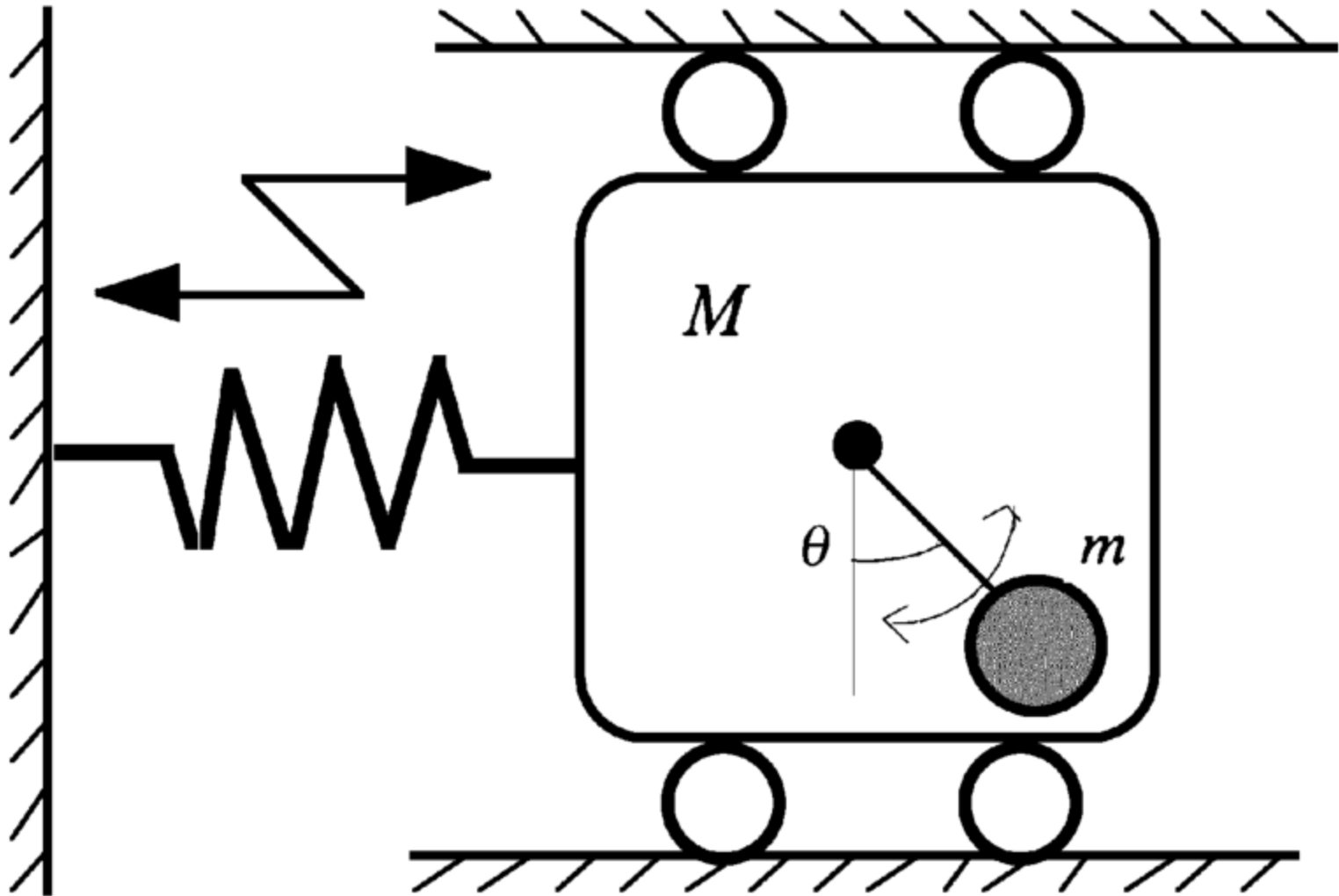


Fig. 7.1 TORA system.

Consider the system shown in Figure 7.1, which represents a translational oscillator with an eccentric rotational proof mass actuator (TORA) [4–6]. The nonlinear coupling between the rotational motion of the actuator and the translational motion of the oscillator provides the mechanism for control.

Let x_1 and x_2 denote the translational position and velocity of the cart with $x_2 = \dot{x}_1$. Let $x_3 = \theta$ and $x_4 = \dot{x}_3$ denote the angular position and velocity of the rotational proof mass. Then the system dynamics can be described by the equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u + \mathbf{d}, \quad (7.2)$$

where u is the torque applied to the eccentric mass, \mathbf{d} is the disturbance, and

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_2 \\ \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon \cos x_3 (x_1 - \varepsilon x_4^2 \sin x_3)}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix},$$

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix},$$

$$\varepsilon = 0.1.$$

Consider the case of no disturbance, as in [4–6], introduce new state variables $z_1 = x_1 + \varepsilon \sin x_3$, $z_2 = x_2 + \varepsilon x_4 \cos x_3$, $y_1 = x_3$, $y_2 = x_4$, and employ the feedback transformation

$$\begin{aligned} \nu &= \frac{1}{1 - \varepsilon^2 \cos^2 y_1} \left[\varepsilon \cos y_1 (z_1 - (1 + y_2^2) \varepsilon \sin y_1) + u \right] \\ &= \alpha(z_1, y_1) + \beta(y_1)u \end{aligned}$$

to bring the system into the following form:

$$\dot{z}_1 = z_2, \quad (7.3)$$

$$\dot{z}_2 = -z_1 + \varepsilon \sin y_1, \quad (7.4)$$

$$\dot{y}_1 = y_2, \quad (7.5)$$

$$\dot{y}_2 = \nu. \quad (7.6)$$

The equilibrium point of system (7.2) can be any point $[0, 0, x_3^0, 0]$, where x_3^0 is an arbitrary constant. Consider $[0, 0, 0, 0]$ as the desired equilibrium point. The linearization around this point has a pair of nonzero imaginary eigenvalues and two zero eigenvalues. Hence the system (7.2) at the origin is an example of a critical nonlinear system. This control problem is interpreted as a regulator problem of $z_1 \rightarrow 0$, $z_2 \rightarrow 0$, $y_1 \rightarrow 0$, and $y_2 \rightarrow 0$.

The T-S model of the TORA system can be constructed from (7.3)–(7.6) by using the fuzzy model construction described in Chapter 2:

Rule 1

IF $y_1(t)$ is “about $-\pi$ or π rad,”

THEN

$$\dot{\mathbf{x}}(t) = \mathbf{A}_1\mathbf{x}(t) + \mathbf{B}_1u(t),$$

$$\mathbf{y}(t) = \mathbf{C}_1\mathbf{x}(t).$$

Rule 2

IF $y_1(t)$ is “about $-\frac{\pi}{2}$ or $\frac{\pi}{2}$ rad,”

THEN

$$\dot{\mathbf{x}}(t) = \mathbf{A}_2\mathbf{x}(t) + \mathbf{B}_2u(t),$$

$$\mathbf{y}(t) = \mathbf{C}_2\mathbf{x}(t).$$

Rule 3

IF $y_1(t)$ is “about 0 rad” and $y_2(t)$ is “about 0,”

THEN

$$\dot{\mathbf{x}}(t) = \mathbf{A}_3 \mathbf{x}(t) + \mathbf{B}_3 u(t),$$

$$\mathbf{y}(t) = \mathbf{C}_3 \mathbf{x}(t).$$

Rule 4

IF $y_1(t)$ is “about 0 rad” and $y_2(t)$ is “about $-a$ or a ,”

THEN

$$\dot{\mathbf{x}}(t) = \mathbf{A}_4 \mathbf{x}(t) + \mathbf{B}_4 u(t),$$

$$\mathbf{y}(t) = \mathbf{C}_4 \mathbf{x}(t),$$

Here, $\mathbf{x}^T(t) = [z_1(t), z_2(t), y_1(t), y_2(t)]$,

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & \varepsilon \frac{\sin(\alpha\pi)}{\alpha\pi} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-\varepsilon}{1-\varepsilon^2} & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{1-\varepsilon^2} \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & \varepsilon \frac{2}{\pi} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & \varepsilon & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\varepsilon}{1-\varepsilon^2} & 0 & \frac{-\varepsilon^2}{1-\varepsilon^2} & 0 \end{bmatrix}, \quad \mathbf{B}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{1-\varepsilon^2} \end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & \varepsilon & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\varepsilon}{1-\varepsilon^2} & 0 & \frac{-\varepsilon^2(1+a^2)}{1-\varepsilon^2} & 0 \end{bmatrix}, \quad \mathbf{B}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{1-\varepsilon^2} \end{bmatrix},$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}_3 = \mathbf{C}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In this simulation, $x_4 \in [-a, a]$ ($a = 4$) and $0 < \alpha < 1$ instead of $\alpha = 1$ (e.g., $\alpha = 0.99$) is used to maintain the controllability of the subsystem $(\mathbf{A}_1, \mathbf{B}_1)$ in Rule 1.

The above fuzzy model is represented as

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^r h_i(\mathbf{z}(t)) \{\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t)\}, \quad (7.7)$$

$$\mathbf{y}(t) = \sum_{i=1}^r h_i(\mathbf{z}(t)) \mathbf{C}_i \mathbf{x}(t), \quad (7.8)$$

where $r = 4$ and $\mathbf{z}(t) = [y_1(t) \ y_2(t)]$. Here, $h_i(\mathbf{z}(t))$ is the weight of the i th rules calculated by the membership values. Figure 7.2 shows the membership functions.

The PDC fuzzy controller is designed as follows:

Control Rule 1

IF $y_1(t)$ is “about $-\pi$ or π rad,”

THEN $u(t) = -F_1 x(t)$.

Control Rule 2

IF $y_1(t)$ is “about $-\frac{\pi}{2}$ or $\frac{\pi}{2}$ rad,”

THEN $u(t) = -F_2 x(t)$.

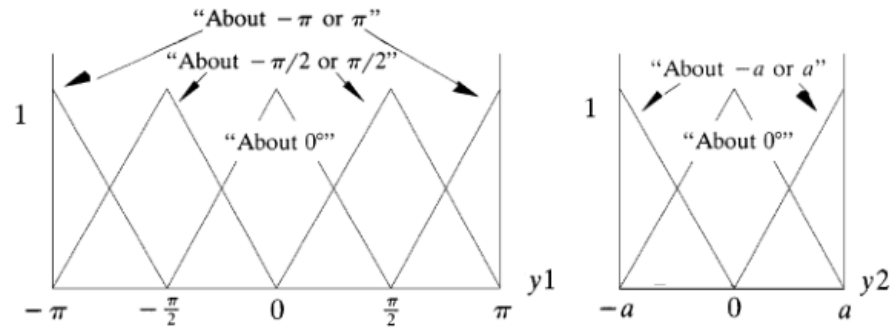


Fig. 7.2 Membership functions.

$K = \text{acker}(A, B, P), \quad K = \text{place}(A, B, P)$

$$P = A - B * K$$

$$u(t) = - \sum_{i=1}^r h_i(z(t)) F_i x(t).$$

c:\Users\osokolov\OneDrive\saszasokolov\Nauka\LR\lit\1-s2.0-S1474667017569165-main.pdf

Control Rule 3

IF $y_1(t)$ is “about 0 rad” and $y_2(t)$ is “about 0,”

THEN $u(t) = -F_3 x(t).$

Control Rule 4

IF $y_1(t)$ is “about 0 rad” and $y_2(t)$ is “about $-a$ or a ,”

THEN $u(t) = -F_4 x(t).$

Figure 7.3 shows the comparison between a stable fuzzy controller [satisfying (3.23) and (3.24)] and a robust fuzzy controller (satisfying the conditions in Theorem 23) for the TORA system with parameter change $\varepsilon = 0.05$. Figure 7.4 compares the performance of the stable fuzzy controller and an optimal fuzzy controller (satisfying the conditions in Theorem 25) for the nominal TORA system. Figure 7.5 shows the control results of the robust

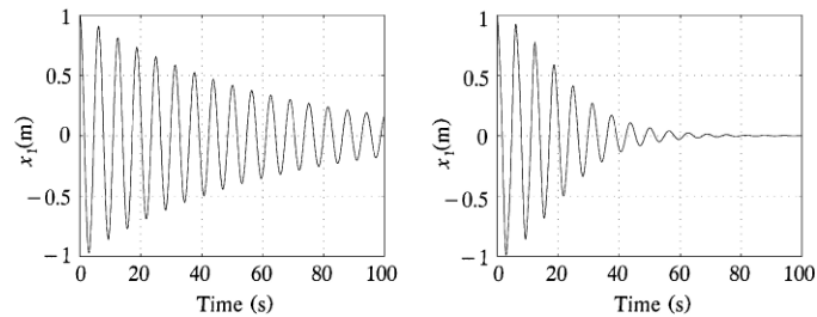


Fig. 7.3 Control results for TORA with parameter change ($\varepsilon = 0.05$).

Fig. 7.3 Control results for TORA with parameter change ($\varepsilon = 0.05$).

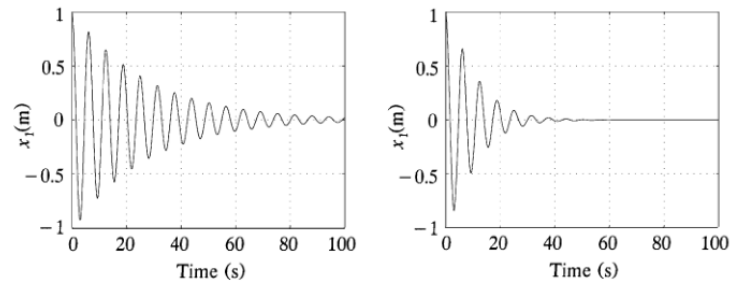
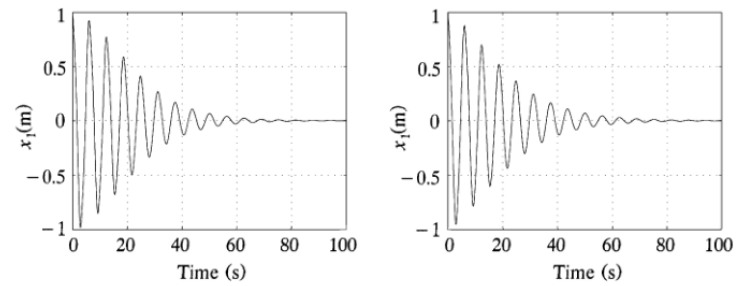


Fig. 7.4 Control results for the nominal TORA.

130 ROBUST-OPTIMAL FUZZY CONTROL



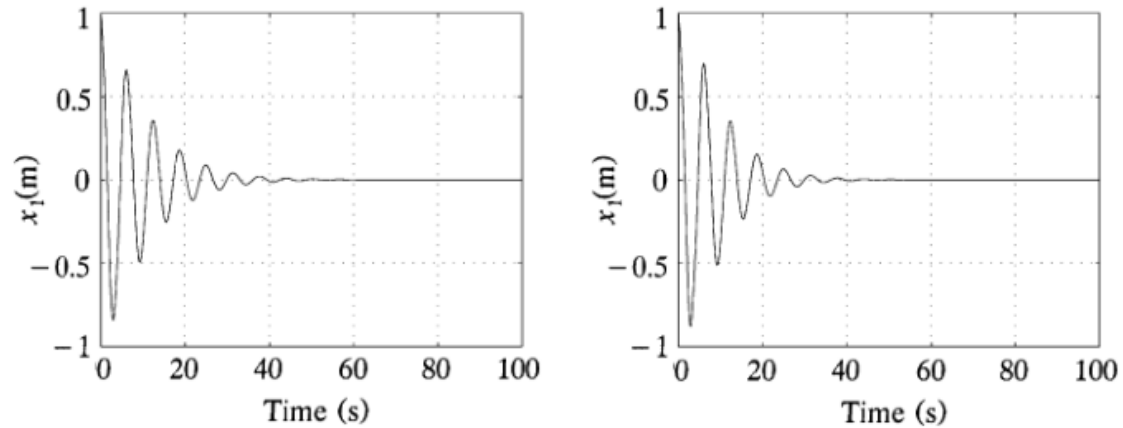


Fig. 7.6 Control results for the nominal TORA.

fuzzy controller and the robust-optimal fuzzy controller (satisfying the conditions in Theorem 28) for the TORA with the parameter change. Figure 7.6 compares the control results of the optimal fuzzy controller and the robust-optimal fuzzy controller for the nominal TORA. In all cases, the fuzzy control designs get the job done but with different performance characteristics. The robust-optimal fuzzy controller is the most versatile in that it addresses both the robustness and the optimality.

FUZZY MODELING OF A VEHICLE WITH TRIPLE TRAILERS

$$x_0(t+1) = x_0(t) + \frac{v \cdot \Delta t}{l} \tan(u(t)), \quad (8.1)$$

$$x_1(t) = x_0(t) - x_2(t), \quad (8.2)$$

$$x_2(t+1) = x_2(t) + \frac{v \cdot \Delta t}{L} \sin(x_1(t)), \quad (8.3)$$

$$x_3(t) = x_2(t) - x_4(t), \quad (8.4)$$

$$x_4(t+1) = x_4(t) + \frac{v \cdot \Delta t}{L} \sin(x_3(t)), \quad (8.5)$$

$$x_5(t) = x_4(t) - x_6(t), \quad (8.6)$$

$$x_6(t+1) = x_6(t) + \frac{v \cdot \Delta t}{L} \sin(x_5(t)), \quad (8.7)$$

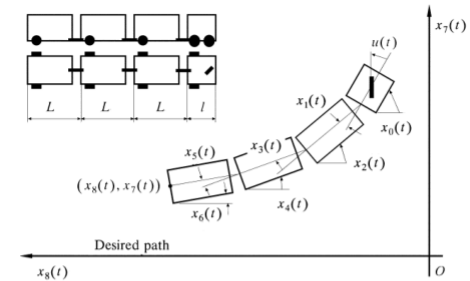


Fig. 8.1 Vehicle model with triple trailers.

$$x_7(t+1) = x_7(t) + v \cdot \Delta t \cos(x_5(t)) \sin\left(\frac{x_6(t+1) + x_6(t)}{2}\right), \quad (8.8)$$

$$x_6(t+1) = x_6(t) + v \cdot \Delta t \cos(x_5(t)) \cos\left(\frac{x_6(t+1) + x_6(t)}{2}\right), \quad (8.9)$$

where

$x_0(t)$ = angle of vehicle,
 $x_1(t)$ = angle difference between vehicle and first trailer,
 $x_2(t)$ = angle of first trailer,
 $x_3(t)$ = angle difference between first trailer and second trailer,
 $x_4(t)$ = angle of second trailer,
 $x_5(t)$ = angle difference between second trailer and third trailer,
 $x_6(t)$ = angle of third trailer,
 $x_7(t)$ = vertical position of rear end of third trailer,
 $x_6(t)$ = horizontal position of rear end of third trailer,
 $u(t)$ = steering angle.