Fuzzy

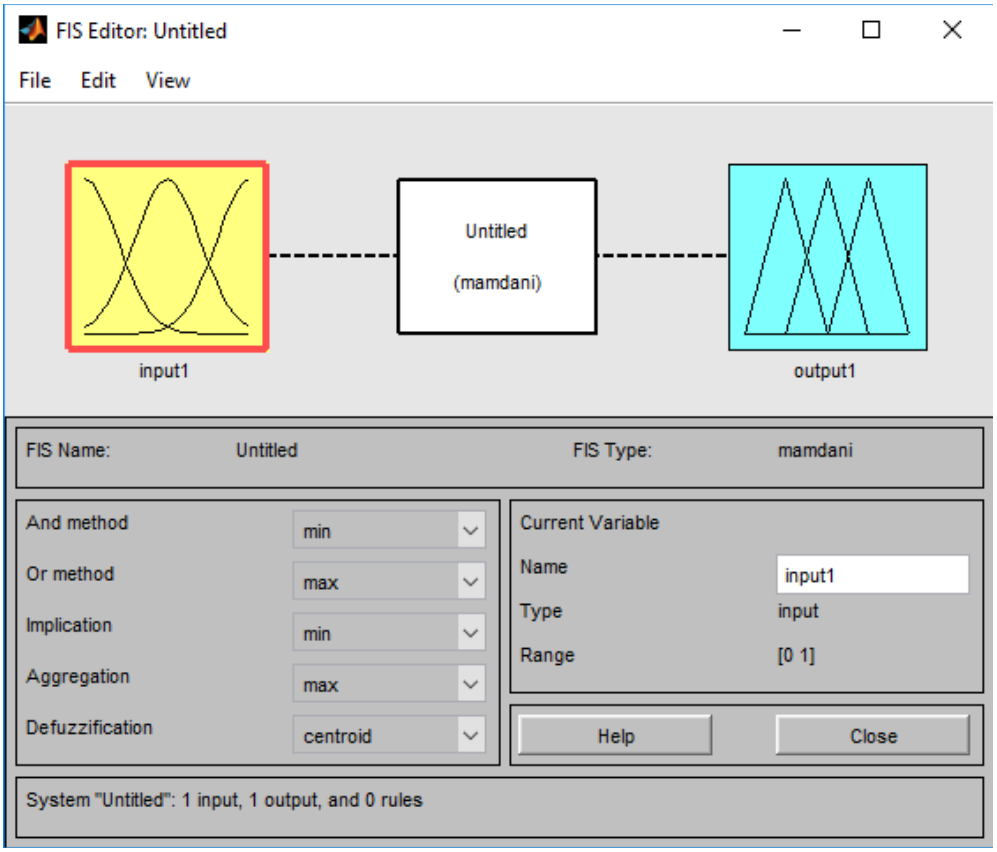
1. Upravlja se kočenjem automobila u odnosu na brzinu kretanja i udaljenost objekta ispred. Objekt ispred automobila može biti neki drugi automobil, neka prepreka na putu i sl. Ulazni skupovi imaju varijable: velika, srednja i mala, dok izlazni skup kočenja ima varijable: lagano, umjereno, snažno i jako snažno. Raspon udaljenosti je od 10 do 150 m. Raspon brzine je od 10 do 100 $\frac{km}{h}$. Kočenje na raspodijeljeno na rasponu od 0 do 10.

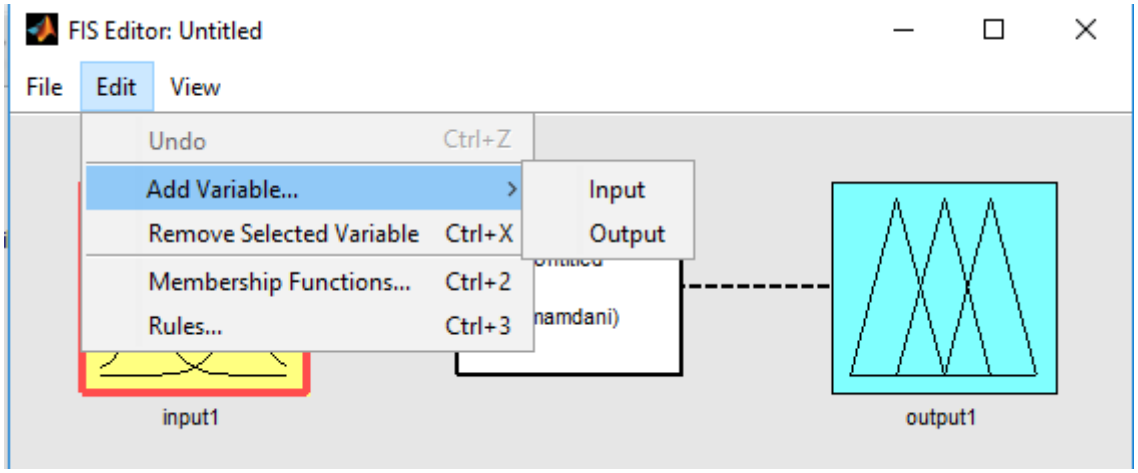
Tabela pravila:

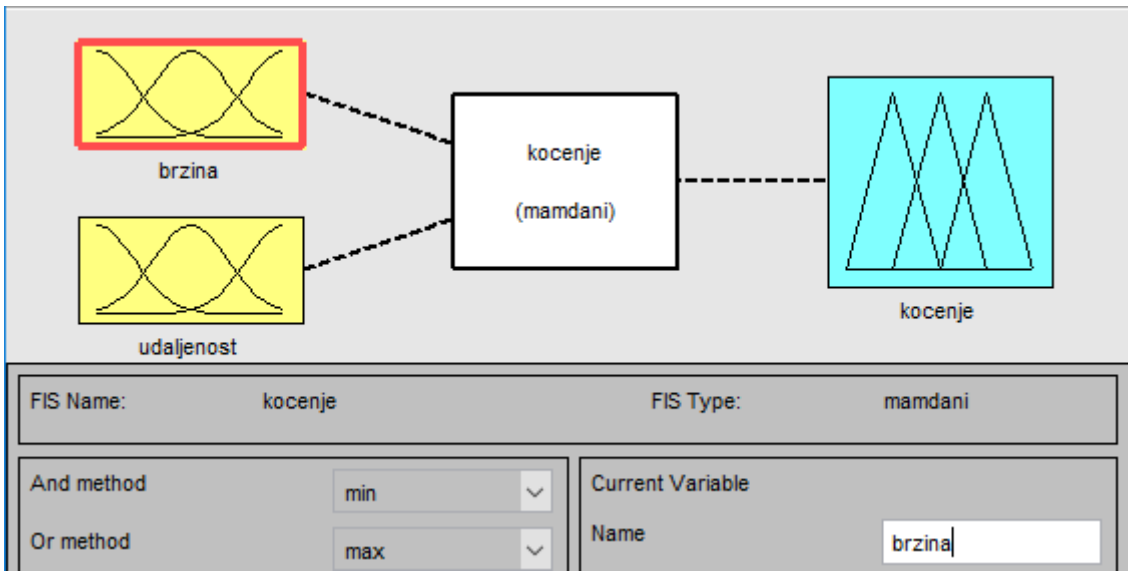
KOČENJE		**BRZINA**		
		V	S	M
UDALJENOST	V	U	U	L
	S	S	U	U
	M	JS	S	U

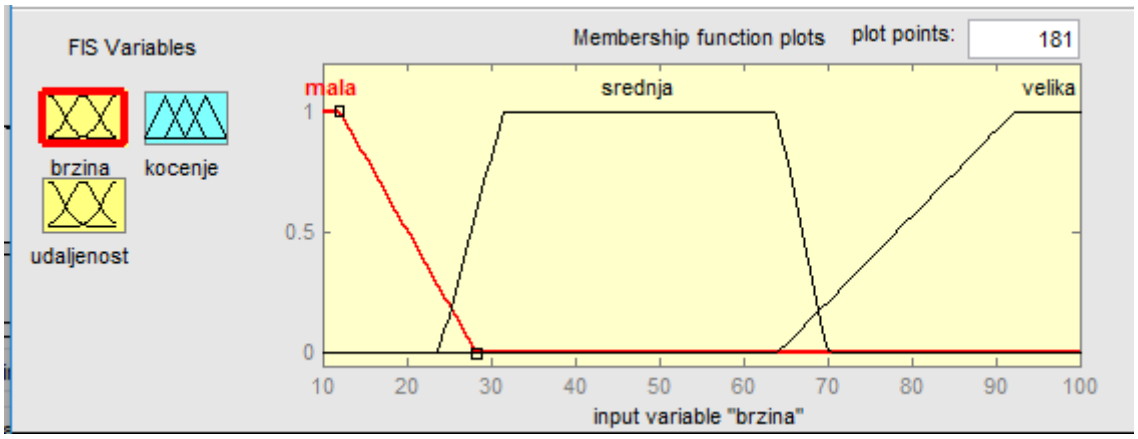
Kuca se u komadni prozor:

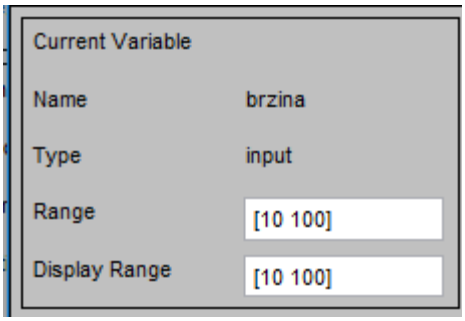
```
fuzzy
```

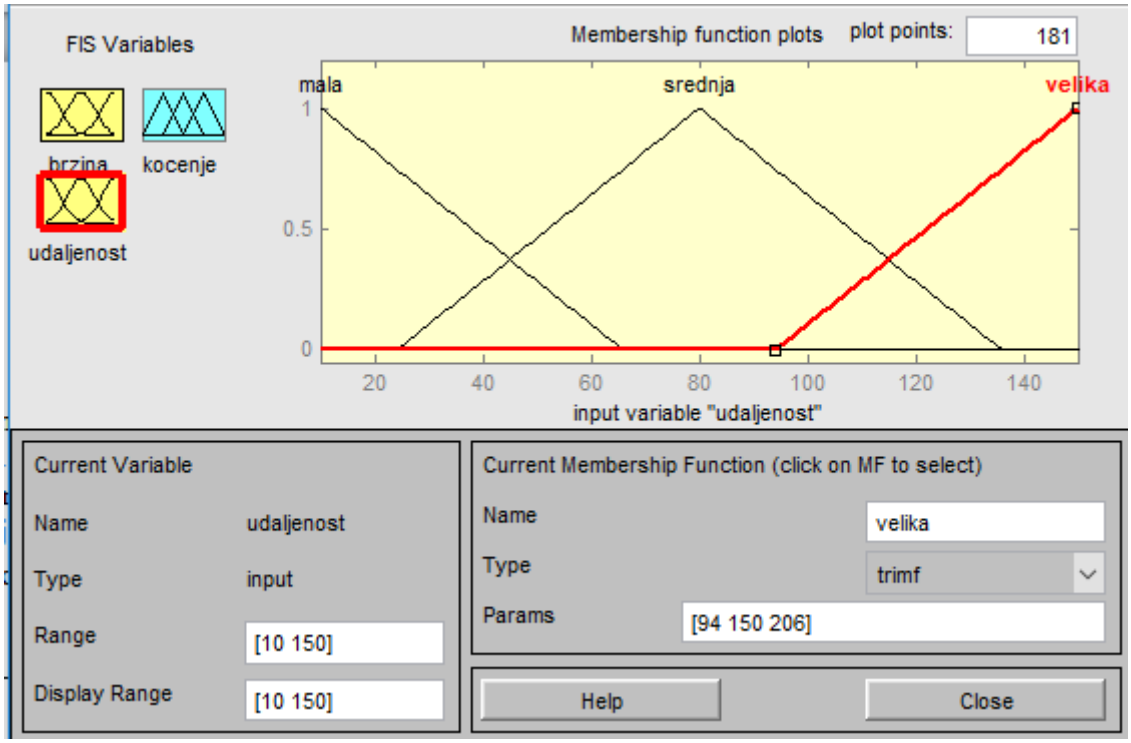


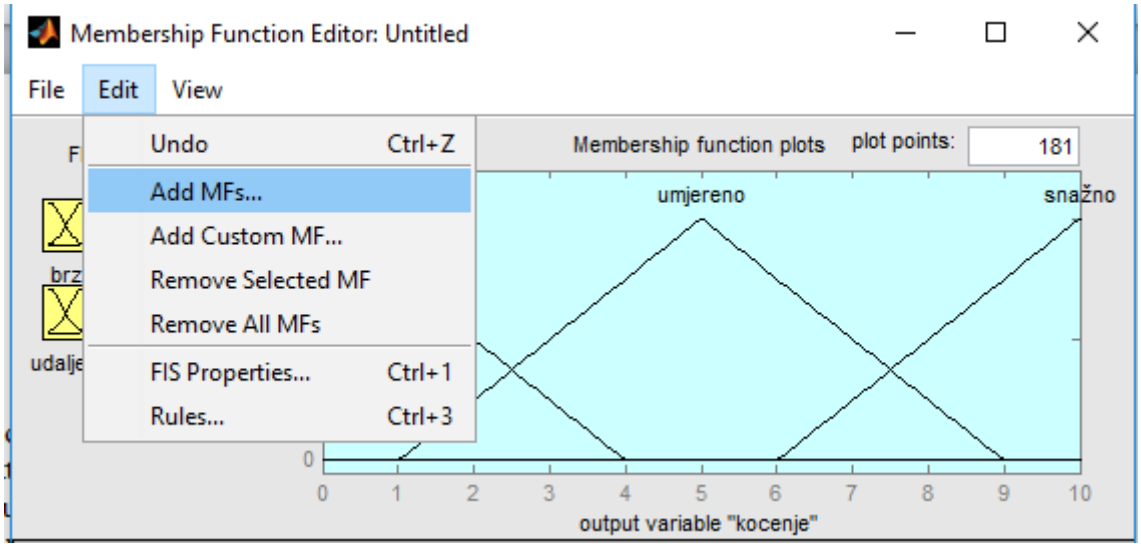

FIS Editor: Untitled
File
Edit
View
Undo Ctrl+Z
Add Variable...
Remove Selected Variable Ctrl+X
Membership Functions... Ctrl+2
Rules... Ctrl+3
Input
Output
input1
output1

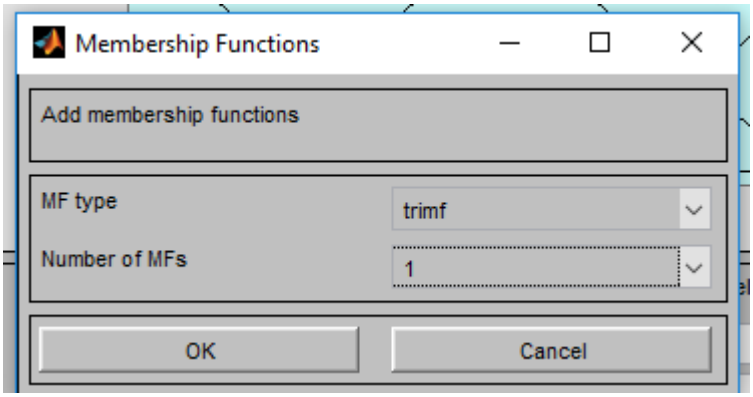

brzina
udaljenost
kocenje
(mamdani)
kocenje
FIS Name: kocenje
FIS Type: mamdani
And method min
Or method max
Current Variable
Name brzina

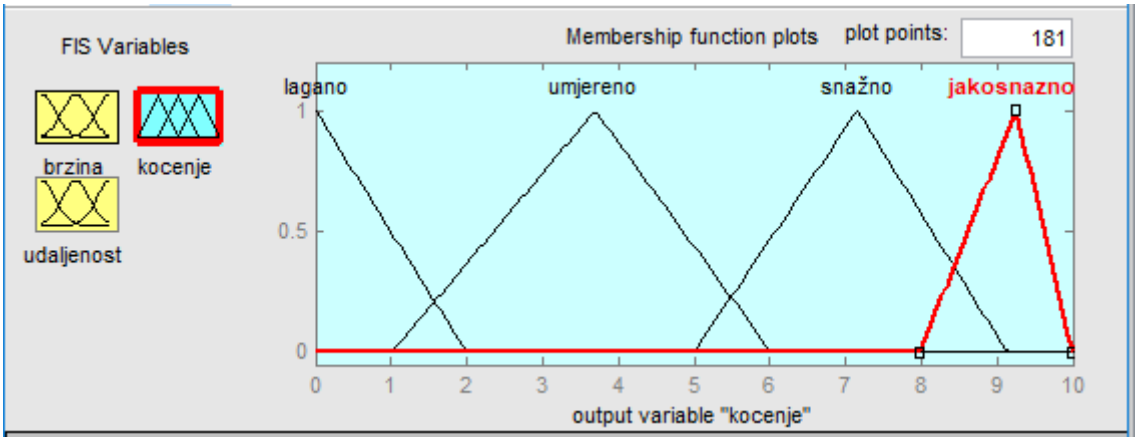

FIS Variables
brzina
kocenje
udaljenost
Membership function plots
plot points: 181
mala
srednja
velika
1
0.5
0
10 20 30 40 50 60 70 80 90 100
input variable "brzina"

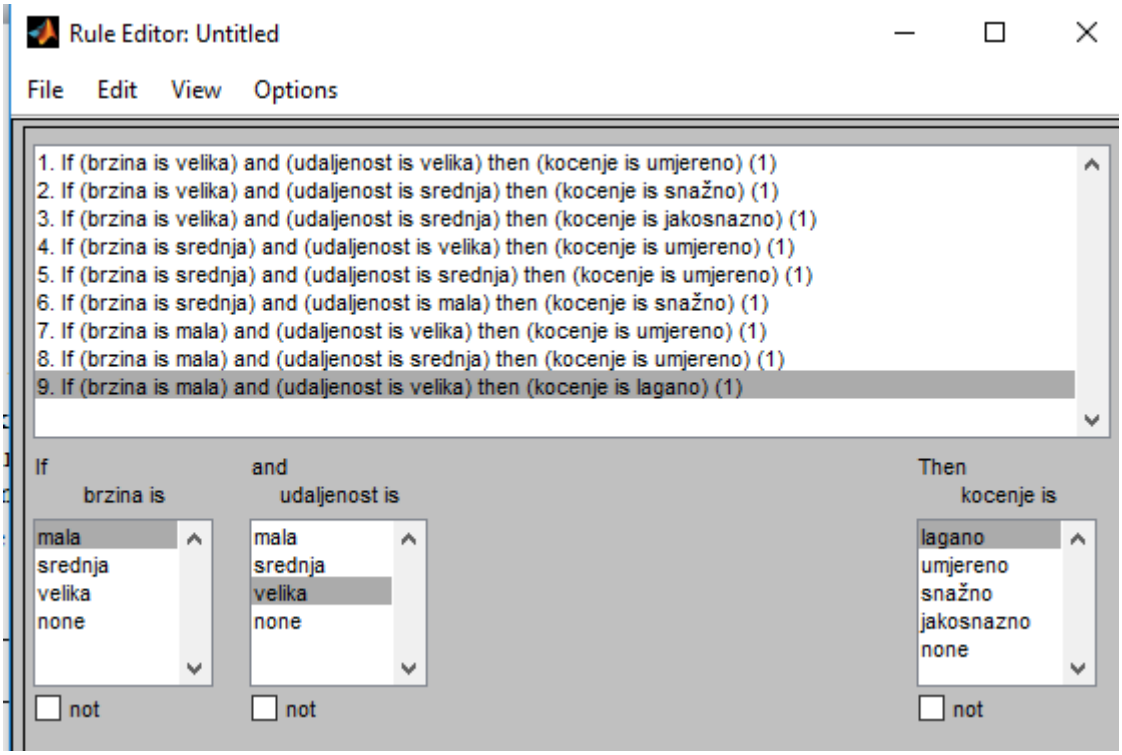

Current Variable
Name
brzina
Type
input
Range
[10 100]
Display Range
[10 100]

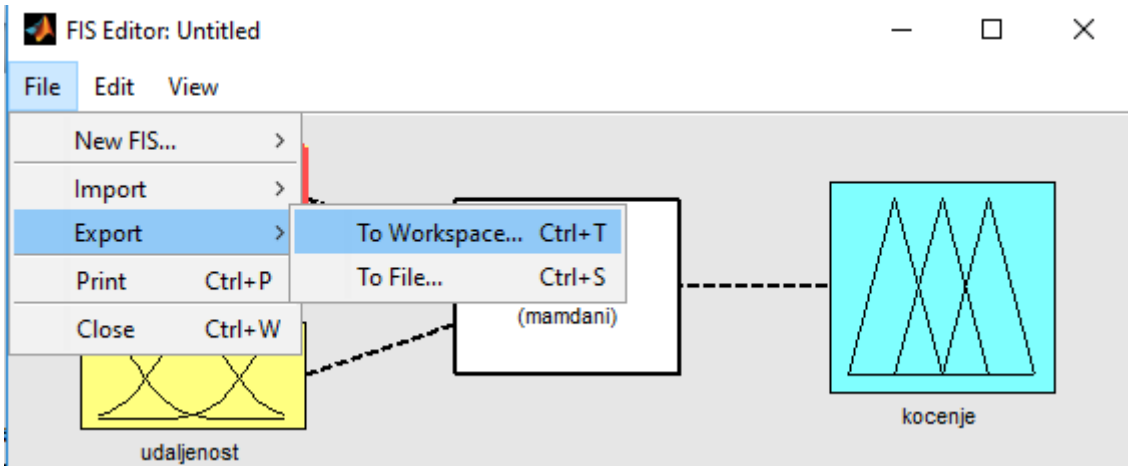

FIS Variables
brzina
kocenje
udaljenost
Membership function plots
plot points:
181
mala
srednja
velika
1
0.5
0
20
40
60
80
100
120
140
input variable "udaljenost"
Current Variable
Name
udaljenost
Type
input
Range
[10 150]
Display Range
[10 150]
Current Membership Function (click on MF to select)
Name
velika
Type
trimf
Params
[94 150 206]
Help
Close

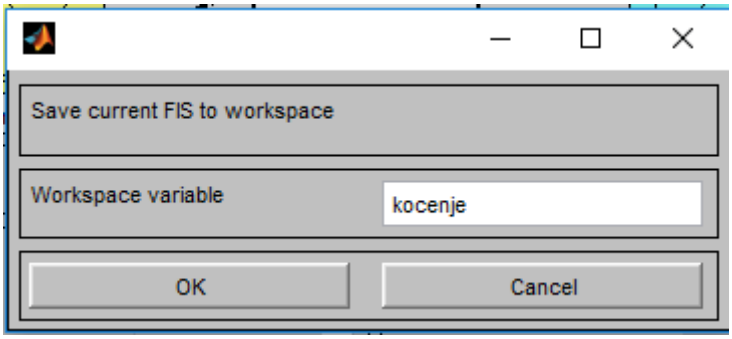

Membership Function Editor: Untitled
File
Edit
View
Undo Ctrl+Z
Add MFs...
Add Custom MF...
Remove Selected MF
Remove All MFs
FIS Properties... Ctrl+1
Rules... Ctrl+3
Membership function plots
plot points:
181
umjereno
snažno
0
0
1
2
3
4
5
6
7
8
9
10
output variable "kocenje"

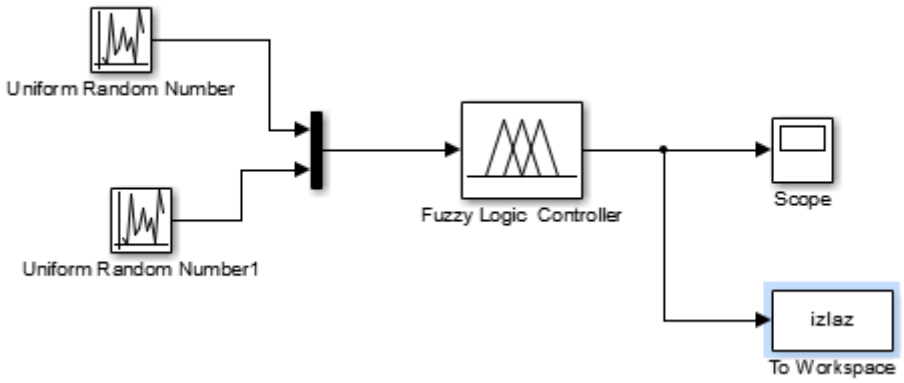

Membership Functions
Add membership functions
MF type
trimf
Number of MFs
1
OK
Cancel

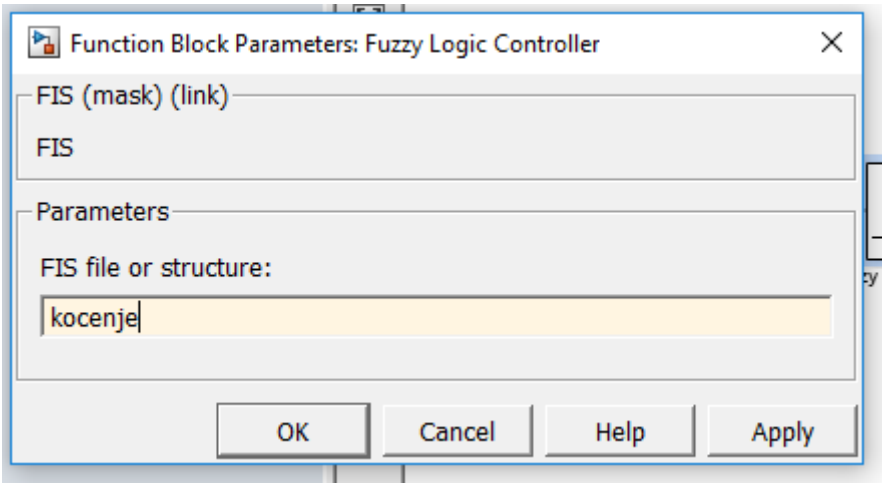

FIS Variables
brzina
kocenje
udaljenost
Membership function plots
plot points:
181
lagano
umjereno
snažno
jakosnazno
1
0.5
0
0 1 2 3 4 5 6 7 8 9 10
output variable "kocenje"

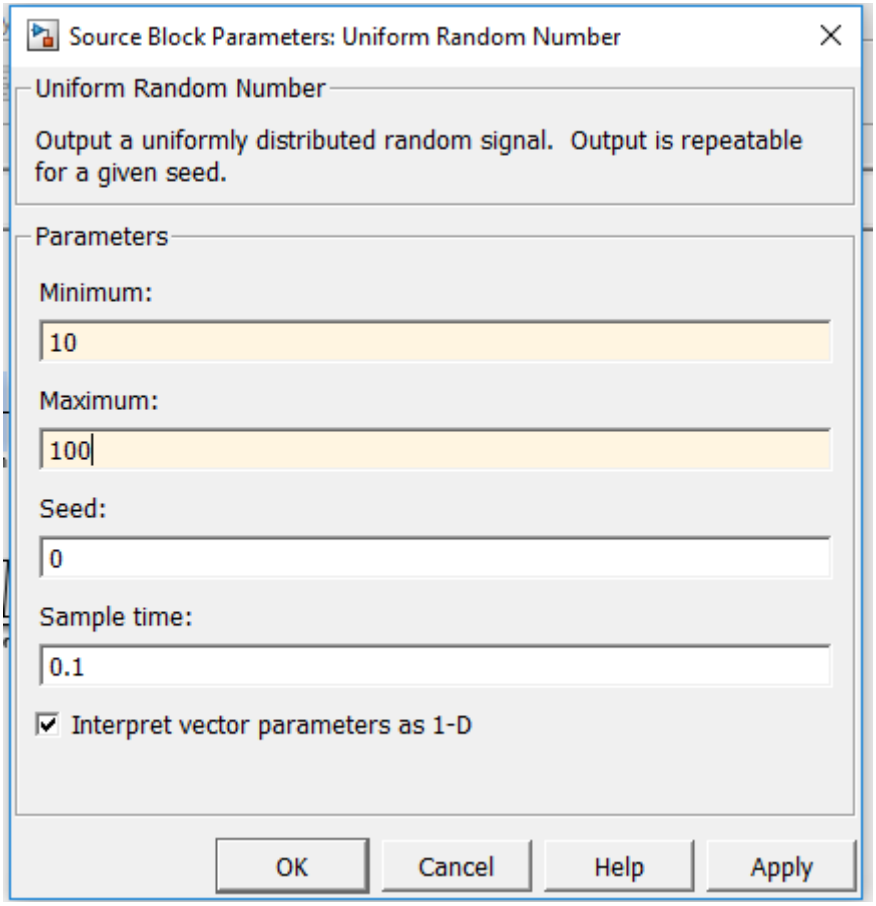

Rule Editor: Untitled
File Edit View Options
1. If (brzina is velika) and (udaljenost is velika) then (kocenje is umjereno) (1)
2. If (brzina is velika) and (udaljenost is srednja) then (kocenje is snažno) (1)
3. If (brzina is velika) and (udaljenost is srednja) then (kocenje is jakosnazno) (1)
4. If (brzina is srednja) and (udaljenost is velika) then (kocenje is umjereno) (1)
5. If (brzina is srednja) and (udaljenost is srednja) then (kocenje is umjereno) (1)
6. If (brzina is srednja) and (udaljenost is mala) then (kocenje is snažno) (1)
7. If (brzina is mala) and (udaljenost is velika) then (kocenje is umjereno) (1)
8. If (brzina is mala) and (udaljenost is srednja) then (kocenje is umjereno) (1)
9. If (brzina is mala) and (udaljenost is velika) then (kocenje is lagano) (1)
If
brzina is
mala
srednja
velika
none
not
and
udaljenost is
mala
srednja
velika
none
not
Then
kocenje is
lagano
umjereno
snažno
jakosnazno
none
not

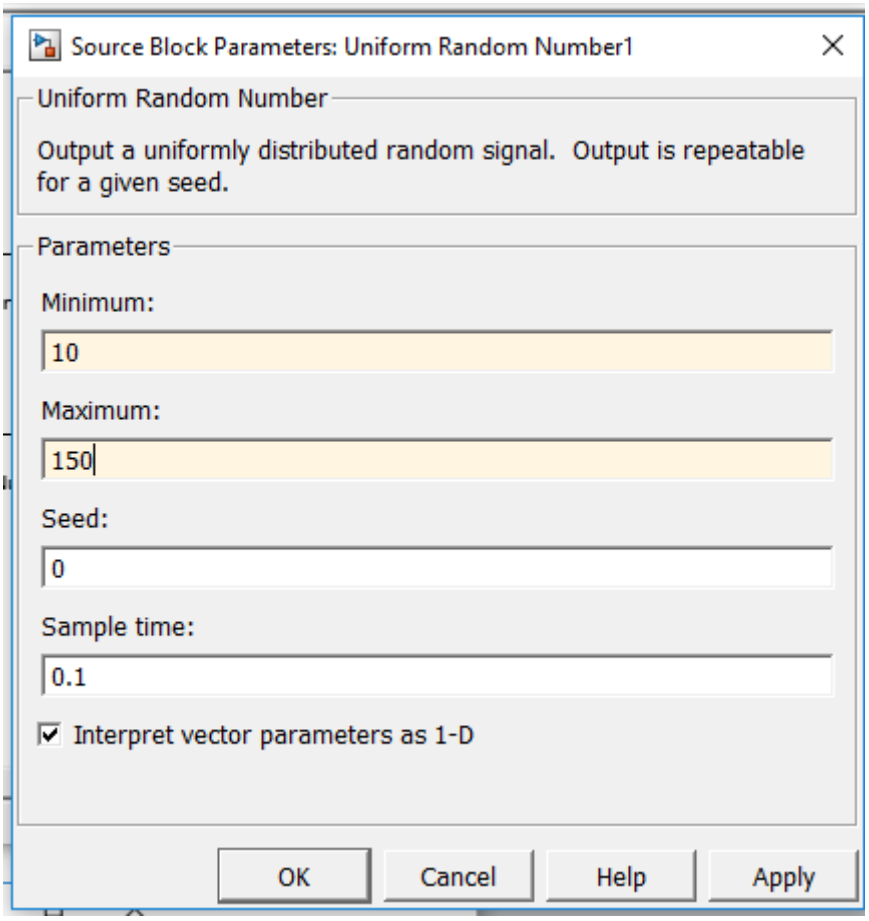

FIS Editor: Untitled
File
Edit
View
New FIS...
Import
Export
Print Ctrl+P
Close Ctrl+W
To Workspace... Ctrl+T
To File... Ctrl+S
(mamdani)
udaljenost
kocenje


Save current FIS to workspace
Workspace variable
kocenje
OK
Cancel


Uniform Random Number
Uniform Random Number1
Fuzzy Logic Controller
Scope
izlaz
To Workspace


Function Block Parameters: Fuzzy Logic Controller
FIS (mask) (link)
FIS
Parameters
FIS file or structure:
kocenje
OK
Cancel
Help
Apply

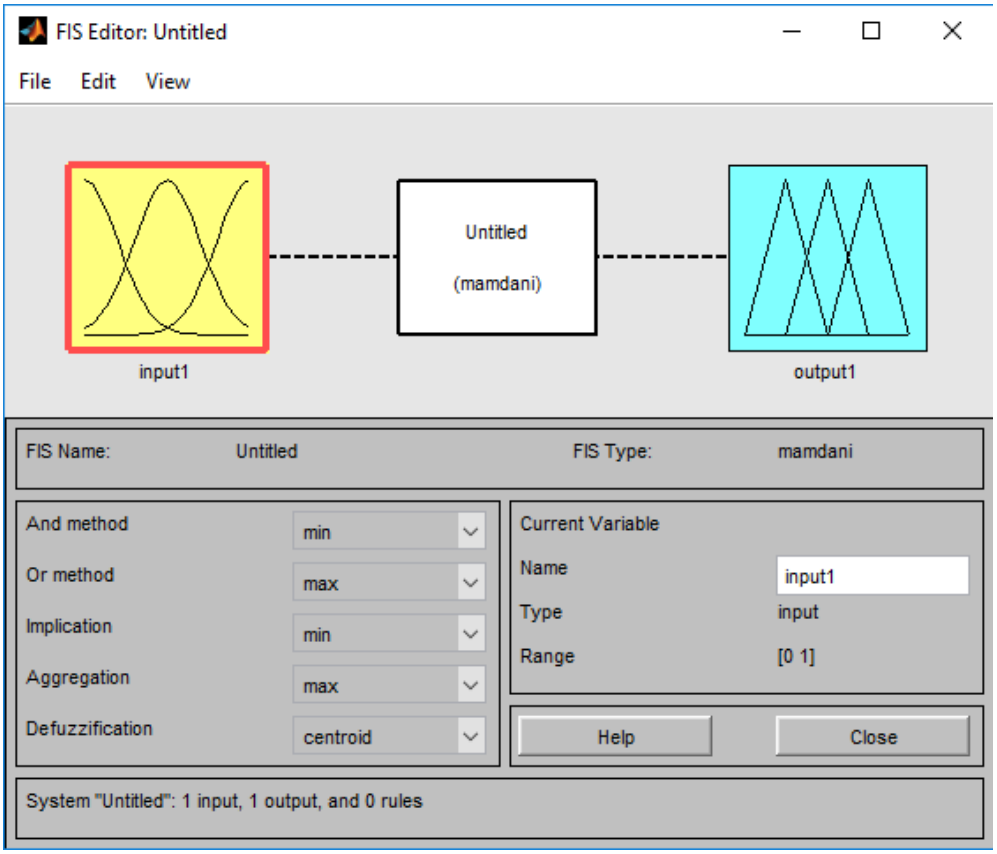

Source Block Parameters: Uniform Random Number
Uniform Random Number
Output a uniformly distributed random signal. Output is repeatable for a given seed.
Parameters
Minimum:
10
Maximum:
100
Seed:
0
Sample time:
0.1
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply

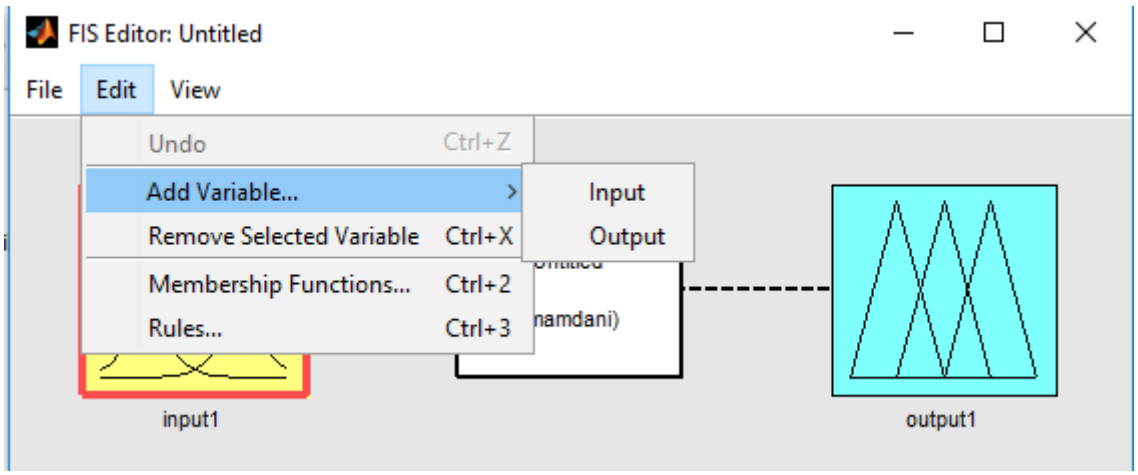

Source Block Parameters: Uniform Random Number1
Uniform Random Number
Output a uniformly distributed random signal. Output is repeatable for a given seed.
Parameters
Minimum:
10
Maximum:
150
Seed:
0
Sample time:
0.1
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply

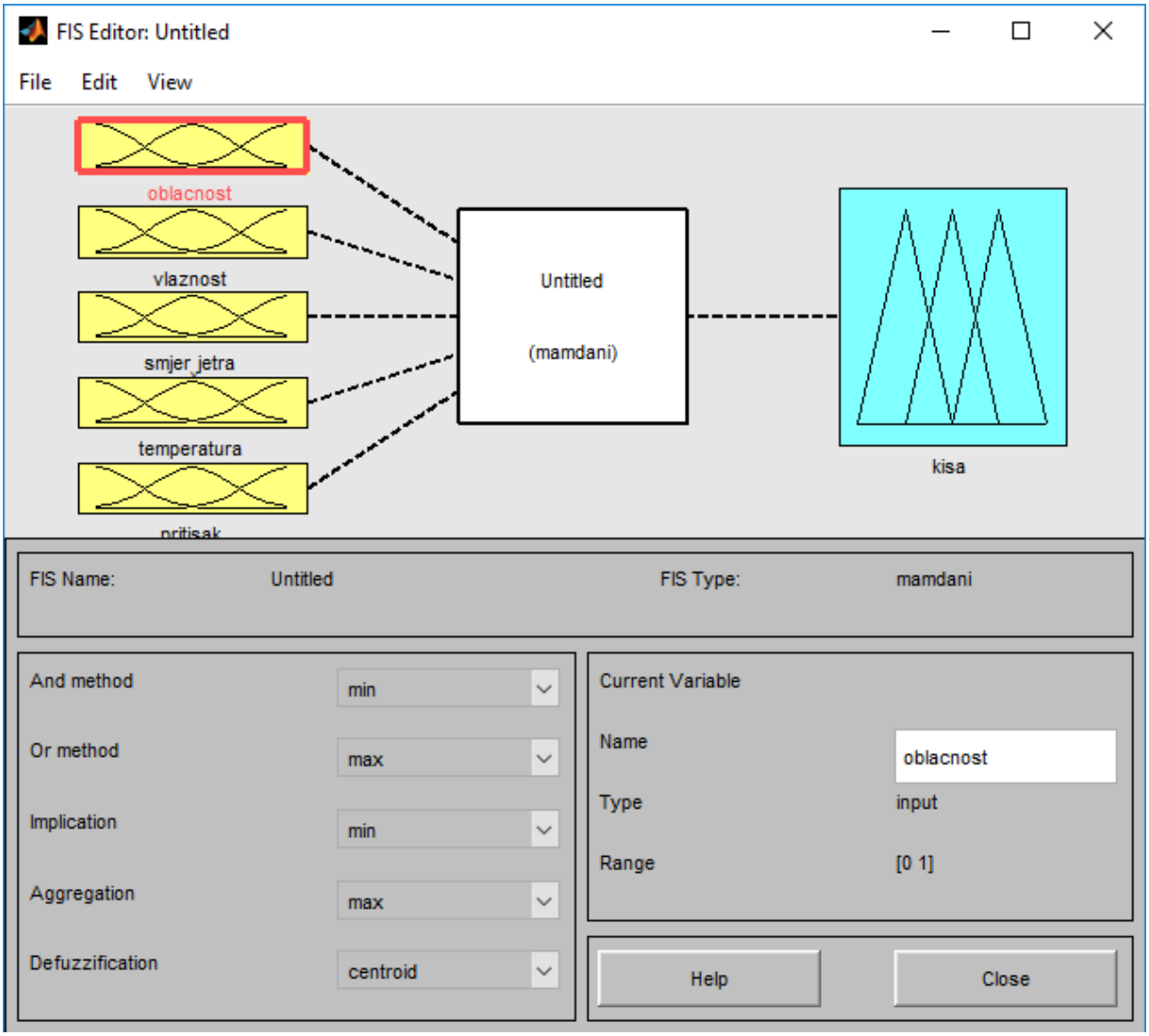
2. Na osnovu 5 ulaznih varijabli potrebno je odrediti da li će kiša padati. Za procjenu se koristi: relativna vlažnost, oblačnost, smjer vjetra, te temperatura i pritisak. Oblaci su u rasponu od 0 do 10, vlažnost od 0 do 100, smjer vjetra od 0 do 360, temperatura od 0 do 40, pritisak od 990 do 1030. Izlazna varijabla je u rasponu od 0 do 100. Napisati bazu pravila i rezultate provjeriti u Simulinku.

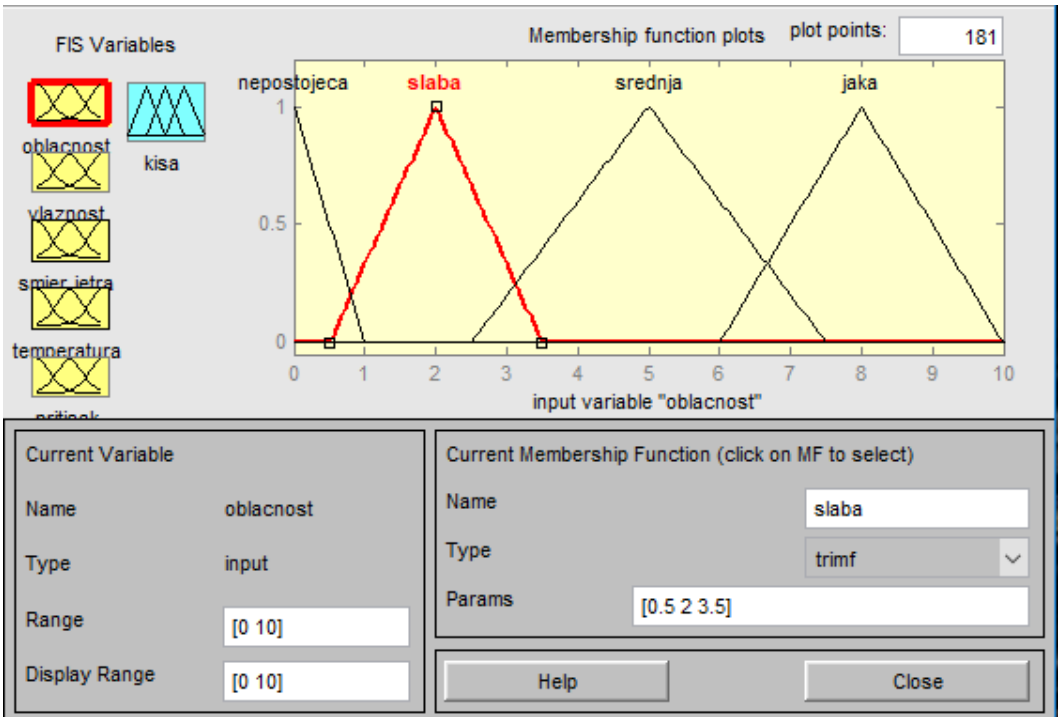
Komadni prozor:

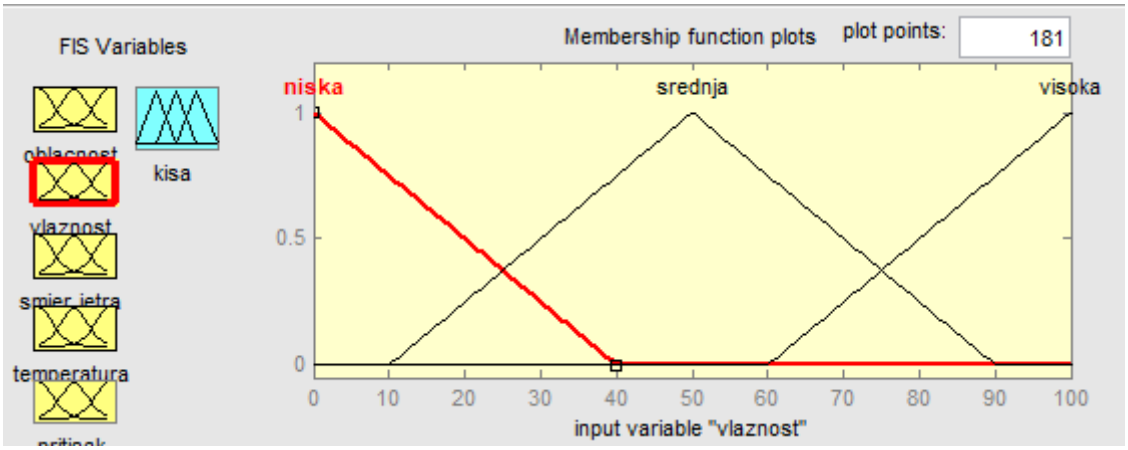
```
fuzzy
```

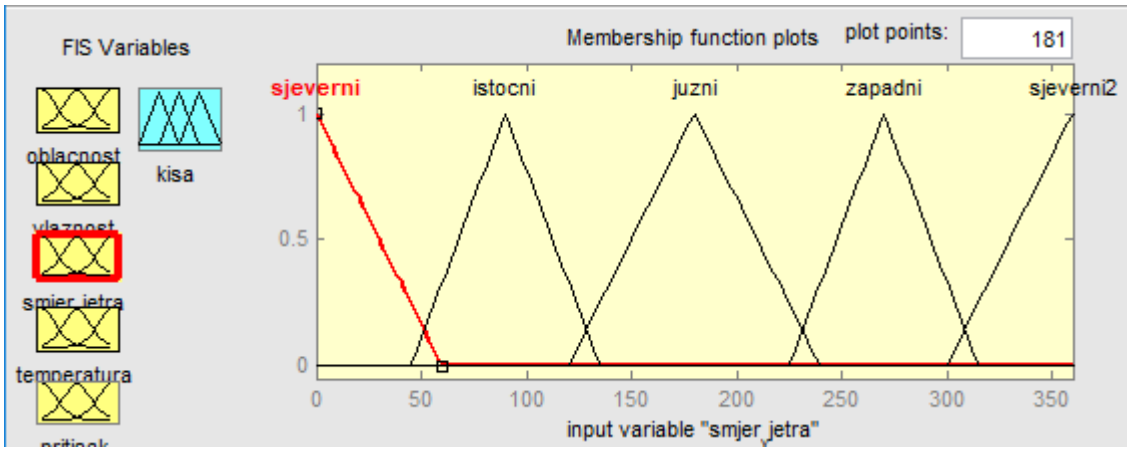


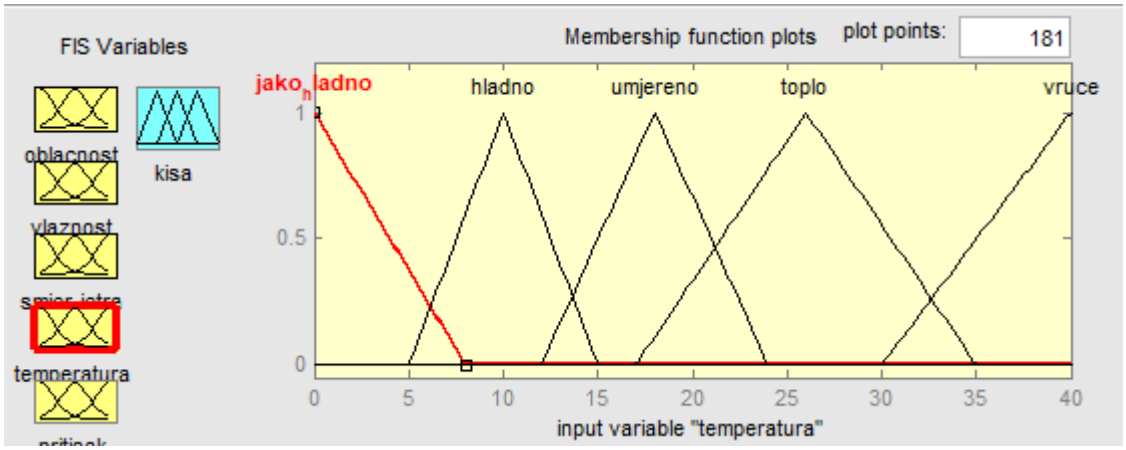


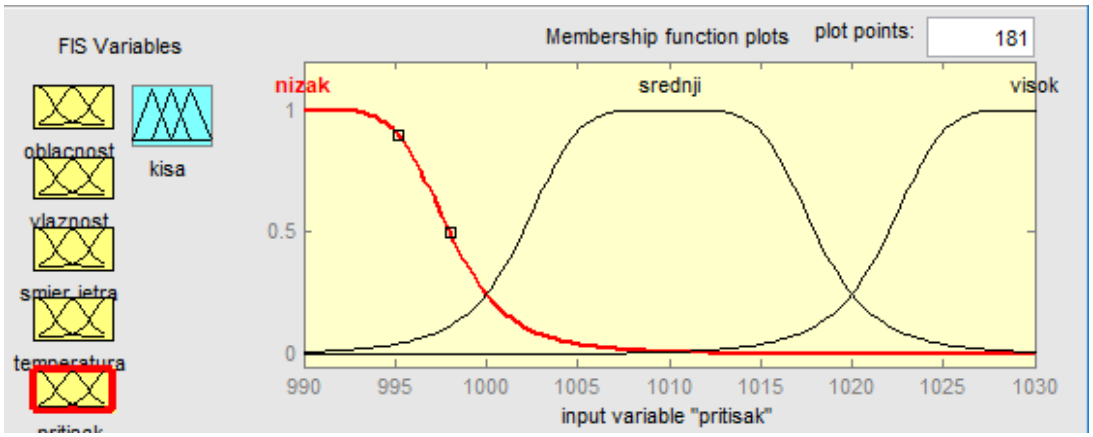

FIS Editor: Untitled
File
Edit
View
oblacnost
vlaznost
smjer_jetra
temperatura
pritisak
Untitled
(mamdani)
kisa
FIS Name:
Untitled
FIS Type:
mamdani
And method
min
Or method
max
Implication
min
Aggregation
max
Defuzzification
centroid
Current Variable
Name
oblacnost
Type
input
Range
[0 1]
Help
Close

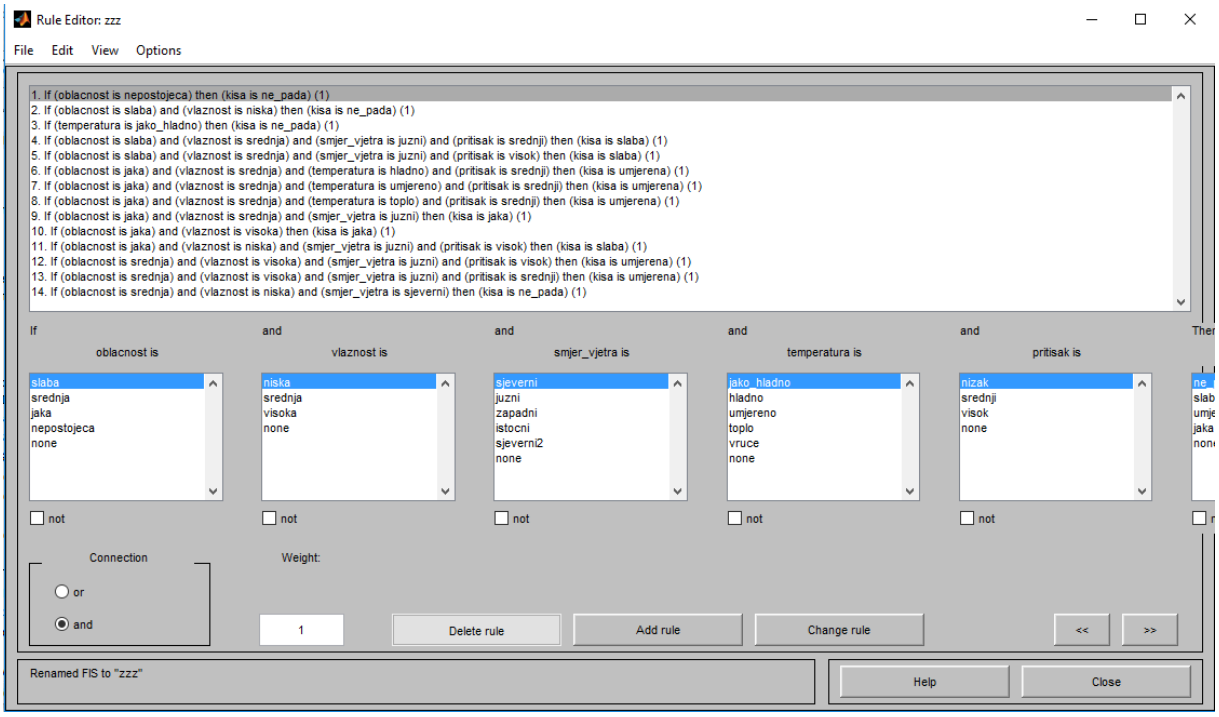

FIS Variables
oblacnost
kisa
vlaznost
smjer_jetra
temperatura
pritisak
Membership function plots
plot points:
181
nepostojeca
slaba
srednja
jaka
1
0.5
0
0
1
2
3
4
5
6
7
8
9
10
input variable "oblacnost"
Current Variable
Name
oblacnost
Type
input
Range
[0 10]
Display Range
[0 10]
Current Membership Function (click on MF to select)
Name
slaba
Type
trimf
Params
[0.5 2 3.5]
Help
Close


FIS Variables
oblacnost
kisa
vlaznost
smjer_jetra
temperatura
pritisak
Membership function plots
plot points:
181
niska
srednja
visoka
1
0.5
0
0
10
20
30
40
50
60
70
80
90
100
input variable "vlaznost"

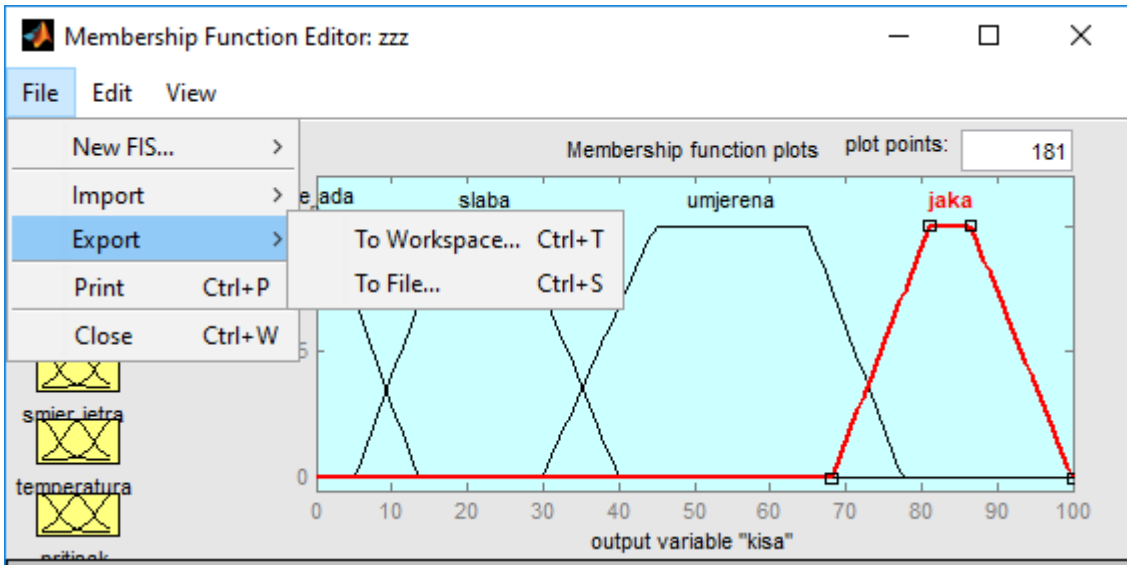

FIS Variables
oblacnost
kisa
vlaznost
smjer_jetra
temperatura
pritisak
Membership function plots
plot points:
181
sjeverni
istocni
juzni
zapadni
sjeverni2
1
0.5
0
0
50
100
150
200
250
300
350
input variable "smjer_jetra"

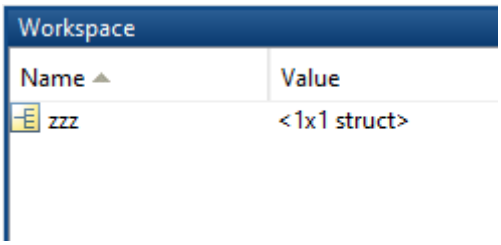

FIS Variables
oblacnost
kisa
vlaznost
smjer_jetra
temperatura
pritisak
Membership function plots
plot points:
181
jako_hladno
hladno
umjereno
toplo
vruce
1
0.5
0
0
5
10
15
20
25
30
35
40
input variable "temperatura"

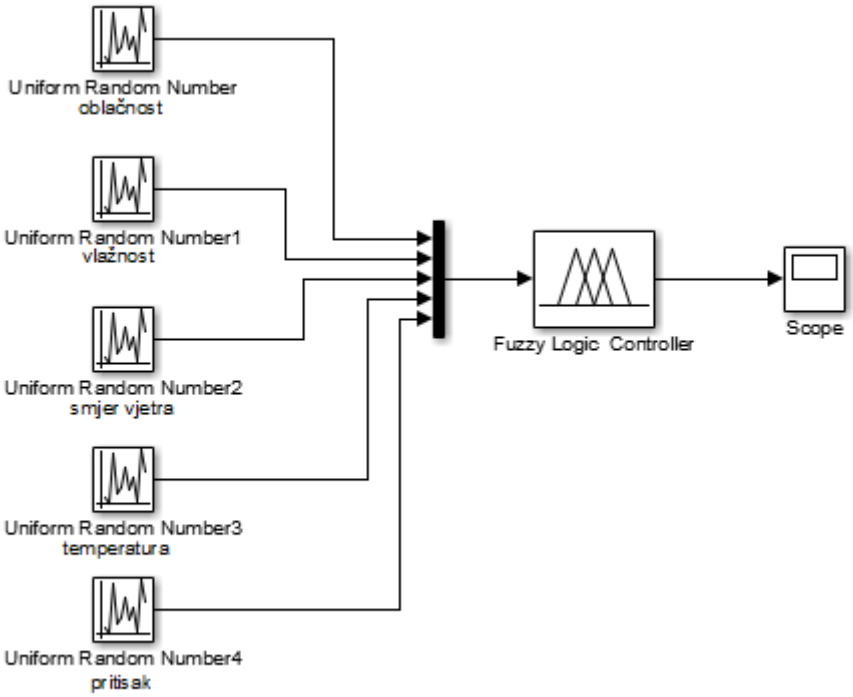

FIS Variables
oblacnost
kisa
vlaznost
smjer_jetra
temperatura
pritisak
Membership function plots
plot points:
181
nizak
srednji
visok
1
0.5
0
990
995
1000
1005
1010
1015
1020
1025
1030
input variable "pritisak"

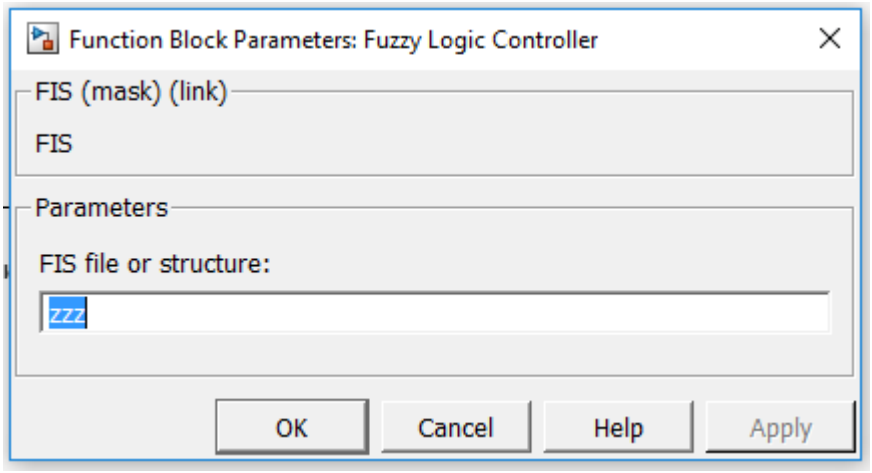


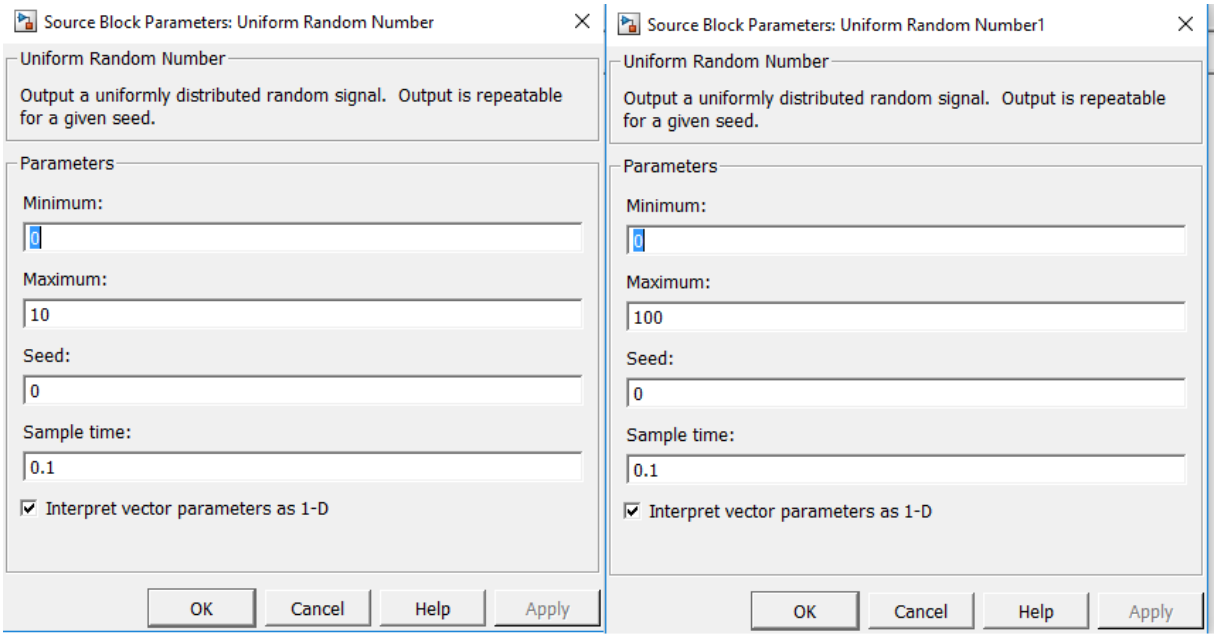
1. If (oblacnost is nepostojeca) then (kisa is ne_pada) (1)
2. If (oblacnost is slaba) and (vlaznost is niska) then (kisa is ne_pada) (1)
3. If (temperatura is jako_hladno) then (kisa is ne_pada) (1)
4. If (oblacnost is slaba) and (vlaznost is srednja) and (smjer_vjetra is juzni) and (pritisak is srednji) then (kisa is slaba) (1)
5. If (oblacnost is slaba) and (vlaznost is srednja) and (smjer_vjetra is juzni) and (pritisak is visok) then (kisa is slaba) (1)
6. If (oblacnost is jaka) and (vlaznost is srednja) and (temperatura is hladno) and (pritisak is srednji) then (kisa is umjerena) (1)
7. If (oblacnost is jaka) and (vlaznost is srednja) and (temperatura is umjereno) and (pritisak is srednji) then (kisa is umjerena) (1)
8. If (oblacnost is jaka) and (vlaznost is srednja) and (temperatura is toplo) and (pritisak is srednji) then (kisa is umjerena) (1)
9. If (oblacnost is jaka) and (vlaznost is srednja) and (smjer_vjetra is juzni) then (kisa is jaka) (1)
10. If (oblacnost is jaka) and (vlaznost is visoka) then (kisa is jaka) (1)
11. If (oblacnost is jaka) and (vlaznost is niska) and (smjer_vjetra is juzni) and (pritisak is visok) then (kisa is slaba) (1)
12. If (oblacnost is srednja) and (vlaznost is visoka) and (smjer_vjetra is juzni) and (pritisak is visok) then (kisa is umjerena) (1)
13. If (oblacnost is srednja) and (vlaznost is visoka) and (smjer_vjetra is juzni) and (pritisak is srednji) then (kisa is umjerena) (1)
14. If (oblacnost is srednja) and (vlaznost is niska) and (smjer_vjetra is sjeverni) then (kisa is ne_pada) (1)

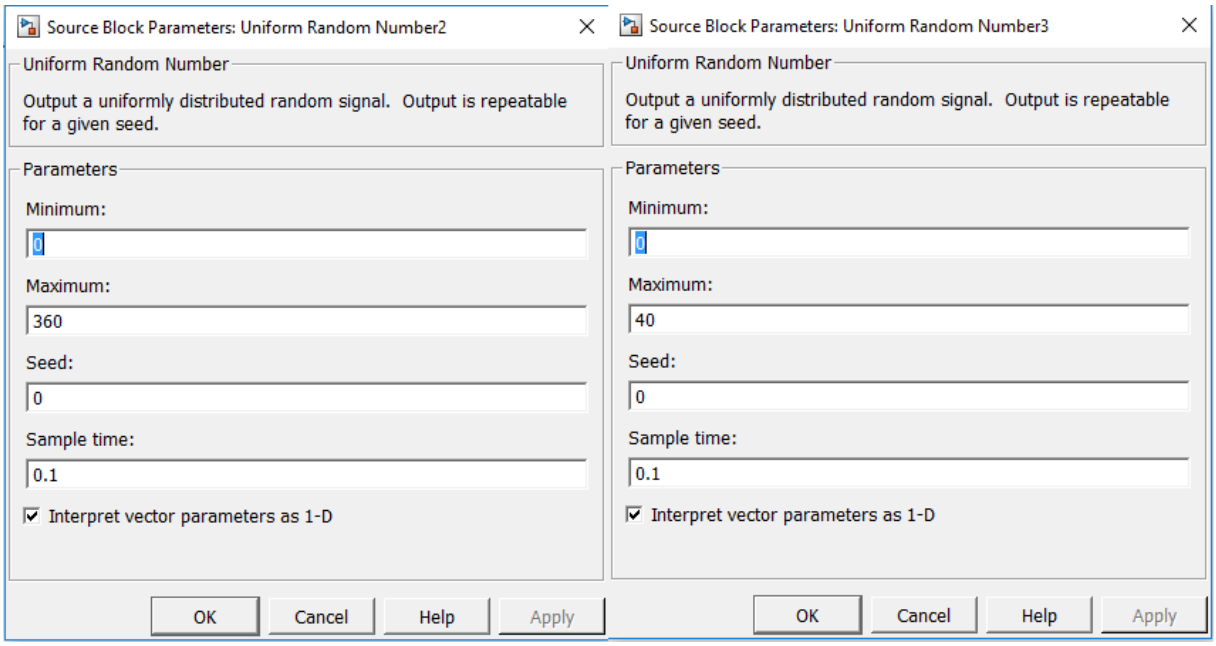


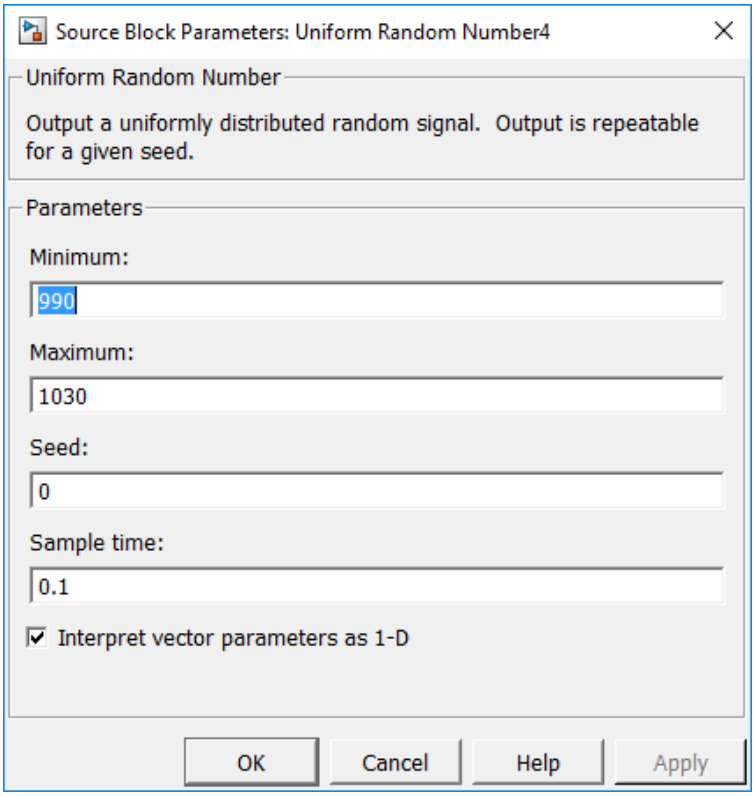

Workspace
Name
Value
zzz
<1x1 struct>

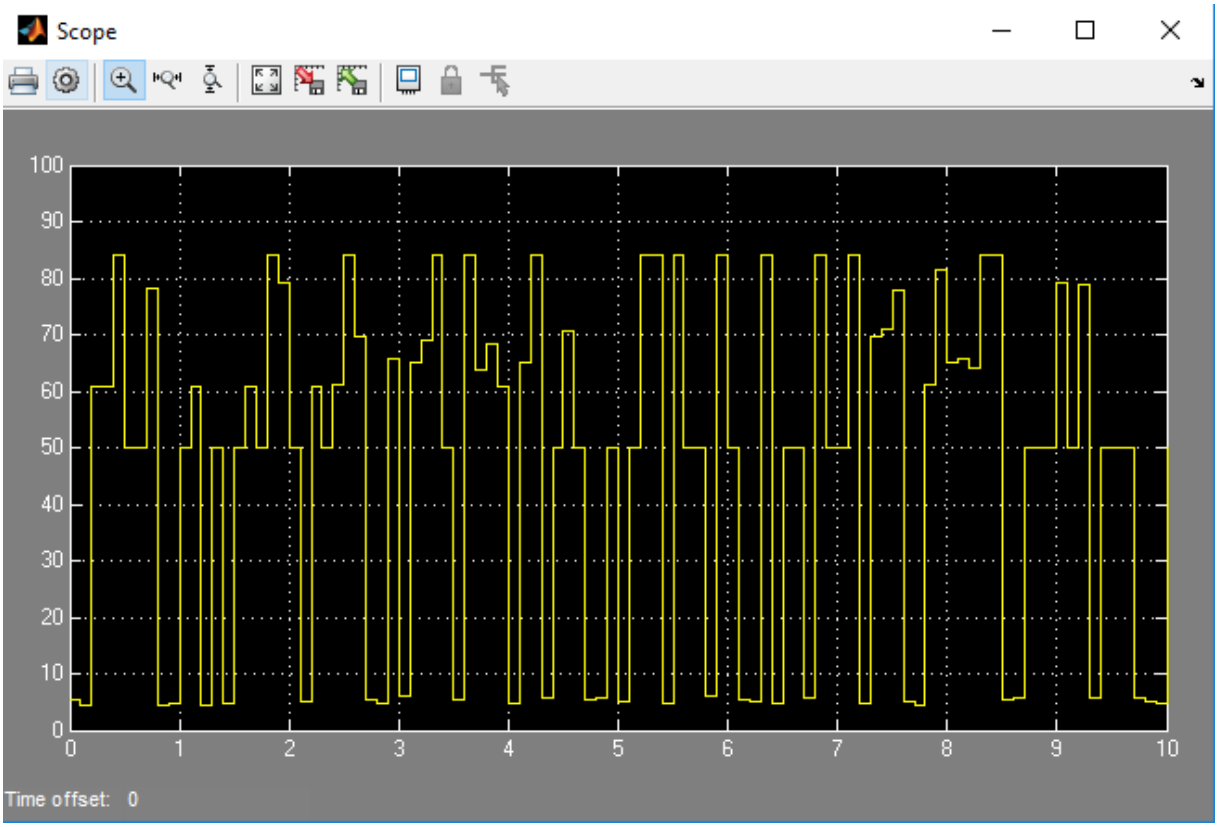

Uniform Random Number
oblačnost
Uniform Random Number1
vlažnost
Uniform Random Number2
smjer vjetra
Uniform Random Number3
temperatura
Uniform Random Number4
pritisak
Fuzzy Logic Controller
Scope


Function Block Parameters: Fuzzy Logic Controller
FIS (mask) (link)
FIS
Parameters
FIS file or structure:
zzz
OK
Cancel
Help
Apply


Source Block Parameters: Uniform Random Number
Uniform Random Number
Output a uniformly distributed random signal. Output is repeatable for a given seed.
Parameters
Minimum:
0
Maximum:
10
Seed:
0
Sample time:
0.1
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply
Source Block Parameters: Uniform Random Number1
Uniform Random Number
Output a uniformly distributed random signal. Output is repeatable for a given seed.
Parameters
Minimum:
0
Maximum:
100
Seed:
0
Sample time:
0.1
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply


Source Block Parameters: Uniform Random Number2
Uniform Random Number
Output a uniformly distributed random signal. Output is repeatable for a given seed.
Parameters
Minimum:
0
Maximum:
360
Seed:
0
Sample time:
0.1
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply
Source Block Parameters: Uniform Random Number3
Uniform Random Number
Output a uniformly distributed random signal. Output is repeatable for a given seed.
Parameters
Minimum:
0
Maximum:
40
Seed:
0
Sample time:
0.1
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply

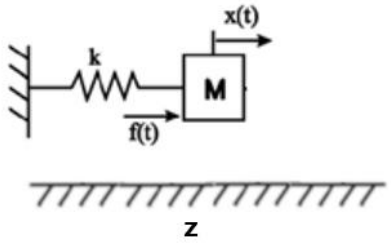

Source Block Parameters: Uniform Random Number4
Uniform Random Number
Output a uniformly distributed random signal. Output is repeatable for a given seed.
Parameters
Minimum:
990
Maximum:
1030
Seed:
0
Sample time:
0.1
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply

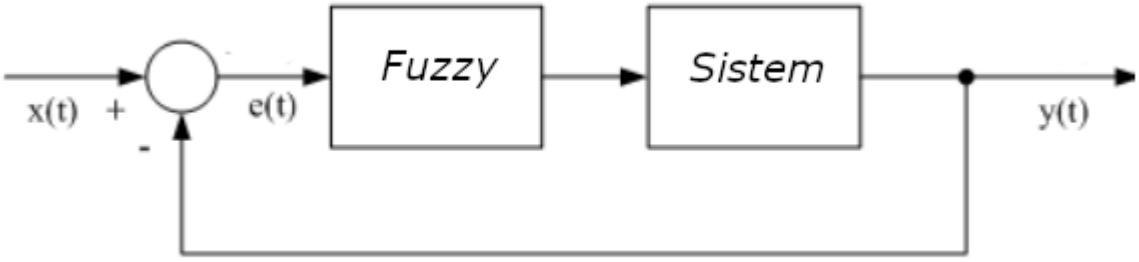

Scope
100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
9
10
Time offset: 0

Fuzzy 2

1. Za sistem fuzzy upravljanja mehaničkim sistemom na slici i za dani mehanički sistem na drugoj slici i na osnovu fuzzy modela iz tabele prikazati izlaze sistema ako se na ulaze dovodi proizvoljni ulazni signal koji odgovara danom fuzzy sistemu.

$$m = 10$$
$$k = 1$$
$$z = 0{,}5$$





$$m \cdot x''(t) + z \cdot x'(t) + kx(t) = f(t)$$

ULAZ1 → GREŠKA

NAZIV	TIP	PARAMETRI	OPSEG
mala	trimf	-3.5 0 3.5	0-100
srednja	trimf	5 6.5 10.5	
velika	trapmf	4 8 10 100	

ULAZ2 → PREKORAČENJE

NAZIV	TIP	PARAMETRI	OPSEG
nisko	trimf	3 8 10	0-100
prosječno	trimf	5.5 10 20	
visoko	trapmf	8 17.5 10 100	

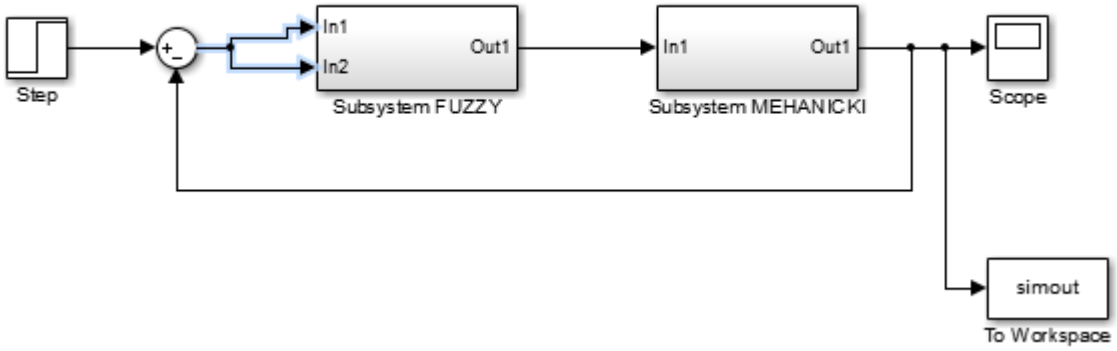
IZLAZ → SILA

NAZIV	TIP	PARAMETRI	OPSEG

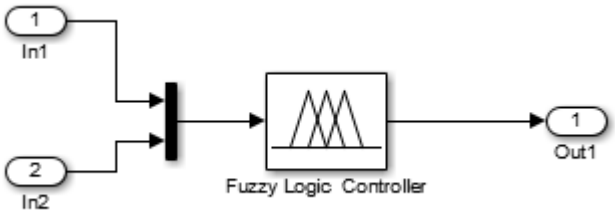
slaba	trimf	-60 10 60	0-200
normalna	trimf	60 85 100	
jaka	trapmf	100 125 200 200	

PRAVILA

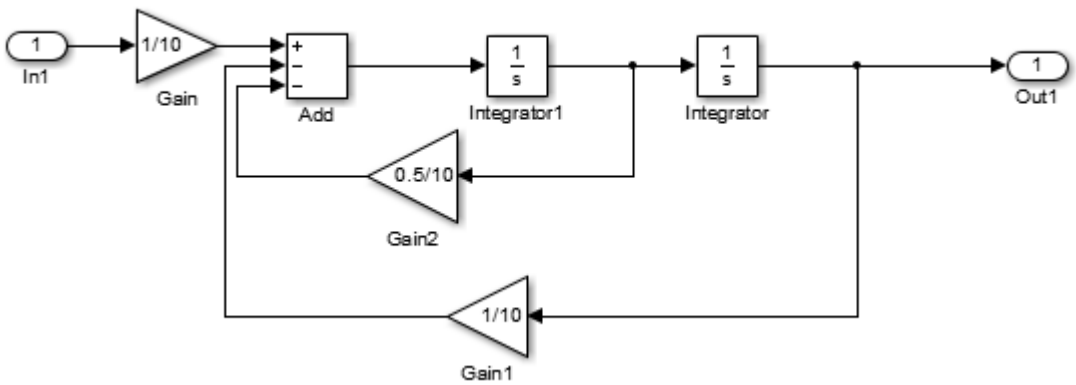
1.	IF GREŠKA	mala	OR	PREKORAČENJE	veliko	THEN SILA IS	slaba
2.		srednja	AND		prosječno		normalna
3.		velika	OR		visoko		jaka



Fuzzy podsistem (može i bez podsistema sa jednim ulazom):

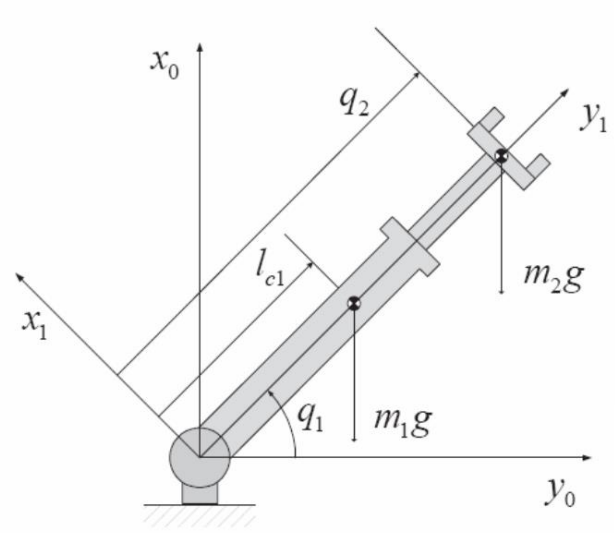


Mehanički podsistem:



2. Izvršiti upravljanje RT robotom koje je prikazano na slici koristeći fuzzy logiku. Potrebno je prikazati poziciju 1, poziciju 2, brzinu 1, brzinu 2, moment 1 i silu 2. *(sa infoservisa)*

RT robot



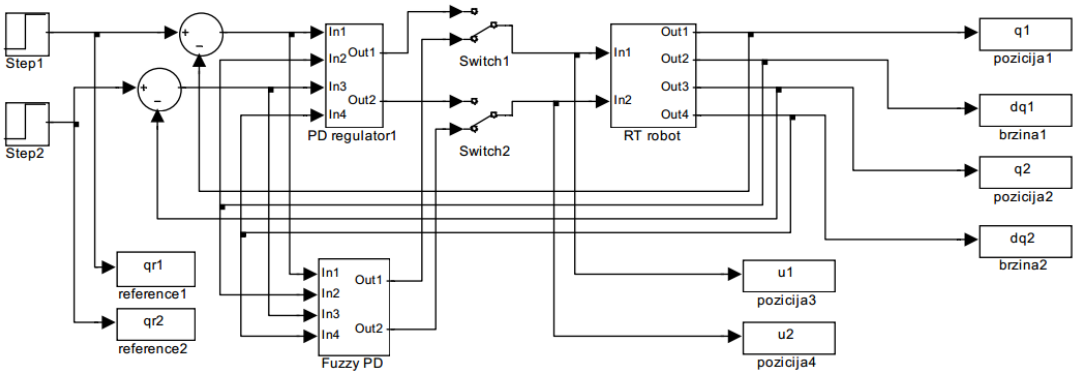
$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u,$$

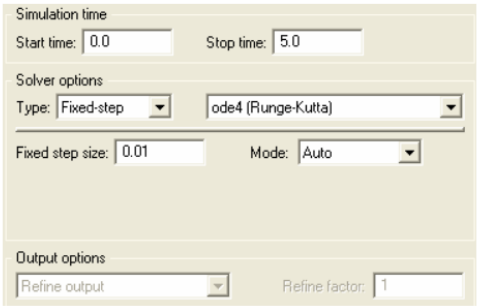
$$M(q) = \begin{bmatrix} m_1 l_{c1}^2 + m_2 q_2^2 + I_1 + I_2 & 0 \\ 0 & m_2 \end{bmatrix},$$

$$C(\dot{q},q) = \begin{bmatrix} m_2\dot{q}_2 q_2 & m_2\dot{q}_1 q_2 \\ -m_2\dot{q}_1 q_2 & 0 \end{bmatrix},$$

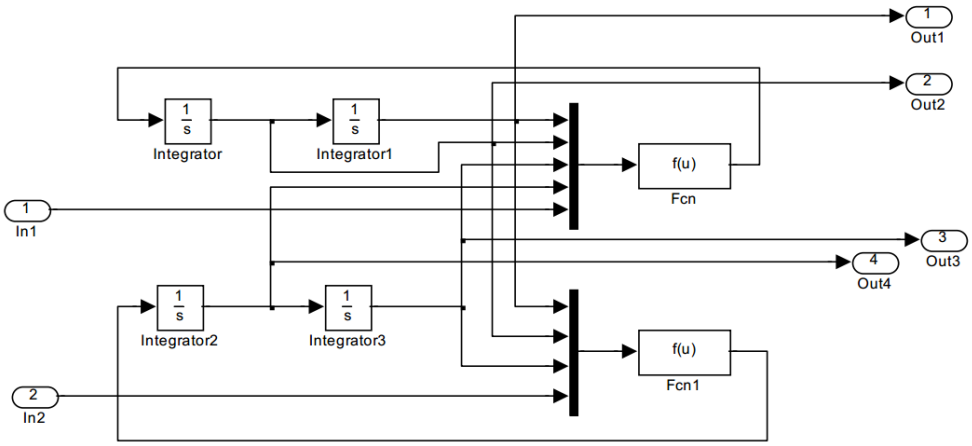
$$g(q) = \begin{bmatrix} m_1 l_{c1} g\cos q_1 + m_2 g q_2 \cos q_1 \\ m_2 g \sin q_1 \end{bmatrix},$$

Simulink model





RT robot blok



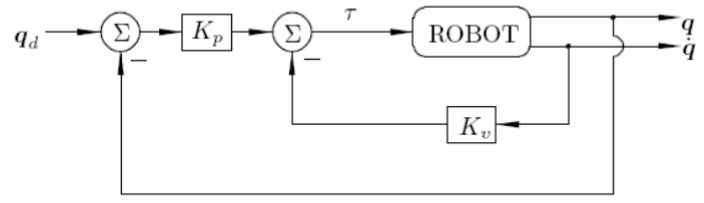
(-2*m2*u(3)*u(2)*u(4)-m1*lc1*g*cos(u(1))-m2*g*u(3)*cos(u(1))+u(5))/(m0+m2*u(3)*u(3))

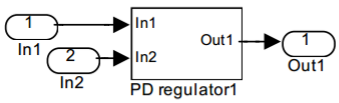
u(3)*u(2)*u(2)-g*sin(u(1))+u(4)/m2

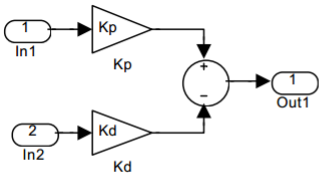
PD regulator blok

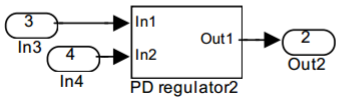
$$\tilde{\boldsymbol{q}}(t) := \boldsymbol{q}_d - \boldsymbol{q}(t)\,.$$

$$\boldsymbol{\tau} = K_p\tilde{\boldsymbol{q}} - K_v\dot{\boldsymbol{q}},$$

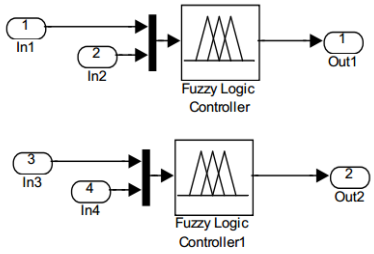


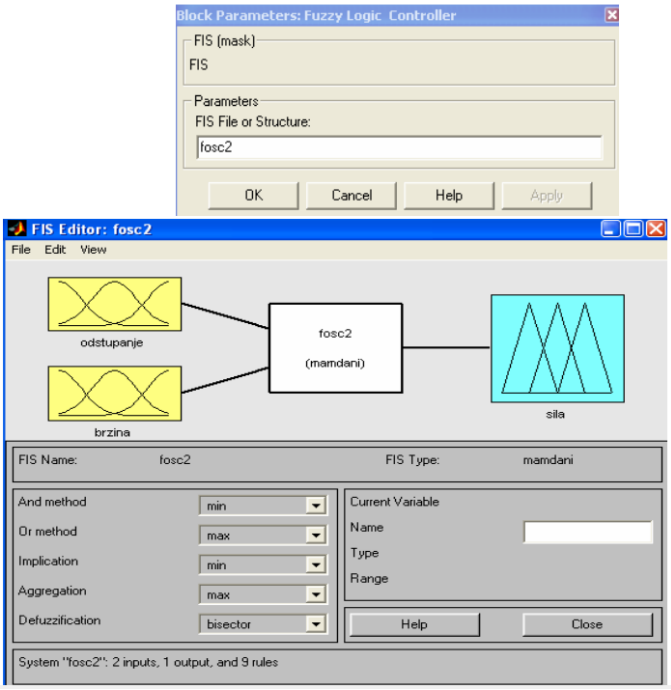




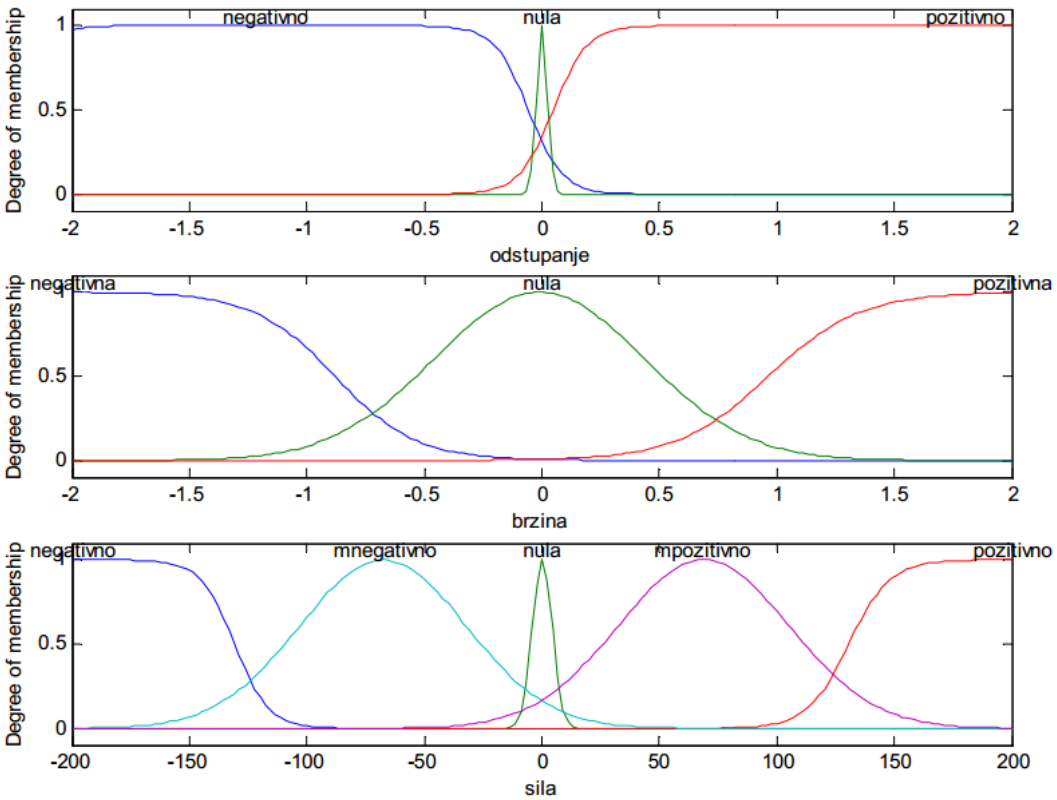


Fuzzy PD regulator blok





Funkcije pripadnosti



Baza pravila ponašanja

1. If (odstupanje is nula) and (brzina is nula) then (sila is nula) (1)
2. If (odstupanje is nula) and (brzina is negativna) then (sila is mpozitivno) (1)
3. If (odstupanje is nula) and (brzina is pozitivna) then (sila is mnegativno) (1)
4. If (odstupanje is negativno) and (brzina is nula) then (sila is negativno) (1)
5. If (odstupanje is pozitivno) and (brzina is nula) then (sila is pozitivno) (1)
6. If (odstupanje is pozitivno) and (brzina is negativna) then (sila is pozitivno) (1)
7. If (odstupanje is pozitivno) and (brzina is pozitivna) then (sila is negativno) (1)
8. If (odstupanje is negativno) and (brzina is pozitivna) then (sila is negativno) (1)
9. If (odstupanje is negativno) and (brzina is negativna) then (sila is pozitivno) (1)

de \ *e*	N	Z	P
N	P	MP	P
Z	N	Z	P
P	N	MN	N

MATLAB KOD:

```
m2 = 10;
m1 = 10;
lc1 = 0.8;
Il2 = 0.1;
m0 = m1*lc1*lc1+Il2;
g = 9.81;
q10 = 0;
dq10 = 0;
q20 = 0;
dq20 = 0;
Kp = 900; Kd = 200;

figure(1)
subplot(221); plot(tout, q1,'b',tout,qr1,'r');
xlabel('vrijeme'); ylabel('pozicija 1');
legend('odziv','referentni signal');

subplot(222); plot(tout, q2, 'b', tout, qr2, 'r');
xlabel('vrijeme'); ylabel('pozicija 2');
legend('odziv','referentni signal');

subplot(223); plot(tout, dq1, 'b');
```

```
xlabel('vrijeme'); ylabel('brzina 1');

subplot(224); plot(tout, dq2, 'b');
xlabel('vrijeme'); ylabel('brzina 2');

figure(2)
subplot(221); plot(tout, u1, 'b');
xlabel('vrijeme'); ylabel('moment 1');

subplot(222); plot(tout, u2, 'b');
xlabel('vrijeme'); ylabel('sila 2');

% (-2*m2*u(3)*u(2)*u(4)-m1*lc1*g*cos(u(1))-
m2*g*u(3)*cos(u(1))+u(5))/(m0+m2*u(3)*u(3))

% u(3)*u(2)*u(2)-g*sin(u(1))+u(4)/m2
```

Fuzzy – zadaća

Na slici 1 je prikazan model dva međusobno povezana rezervoara, gdje su prikazani ventili (mu_1, mi_1, mu_2 i mi_2) i površine A_1 i A_2, kao i nivoi h_1 i h_2. Potrebno je prikazati nivoe rezervoara 1 i 2 (h_1 i h_2) koristeći:

a) Direktan način (Jednačina kola za nivoe h_1 i h_2)

b) Korištenjem PID regulatora

c) Korištenjem Fuzzy regulatora

d) Na jednom modelu i jednom dijagramu prikazati razlike dobijenih rezultata

e) Napraviti GUI aplikaciju u Matlabu koja izgleda kao na slici 2

Poznato je:

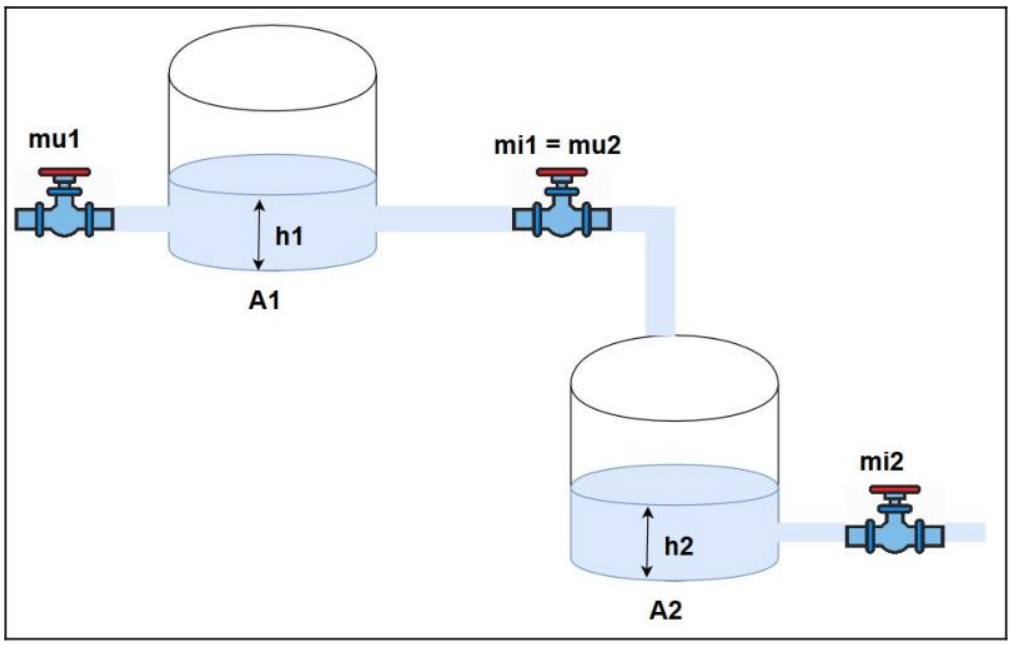
Pumpa: H=12 [m]

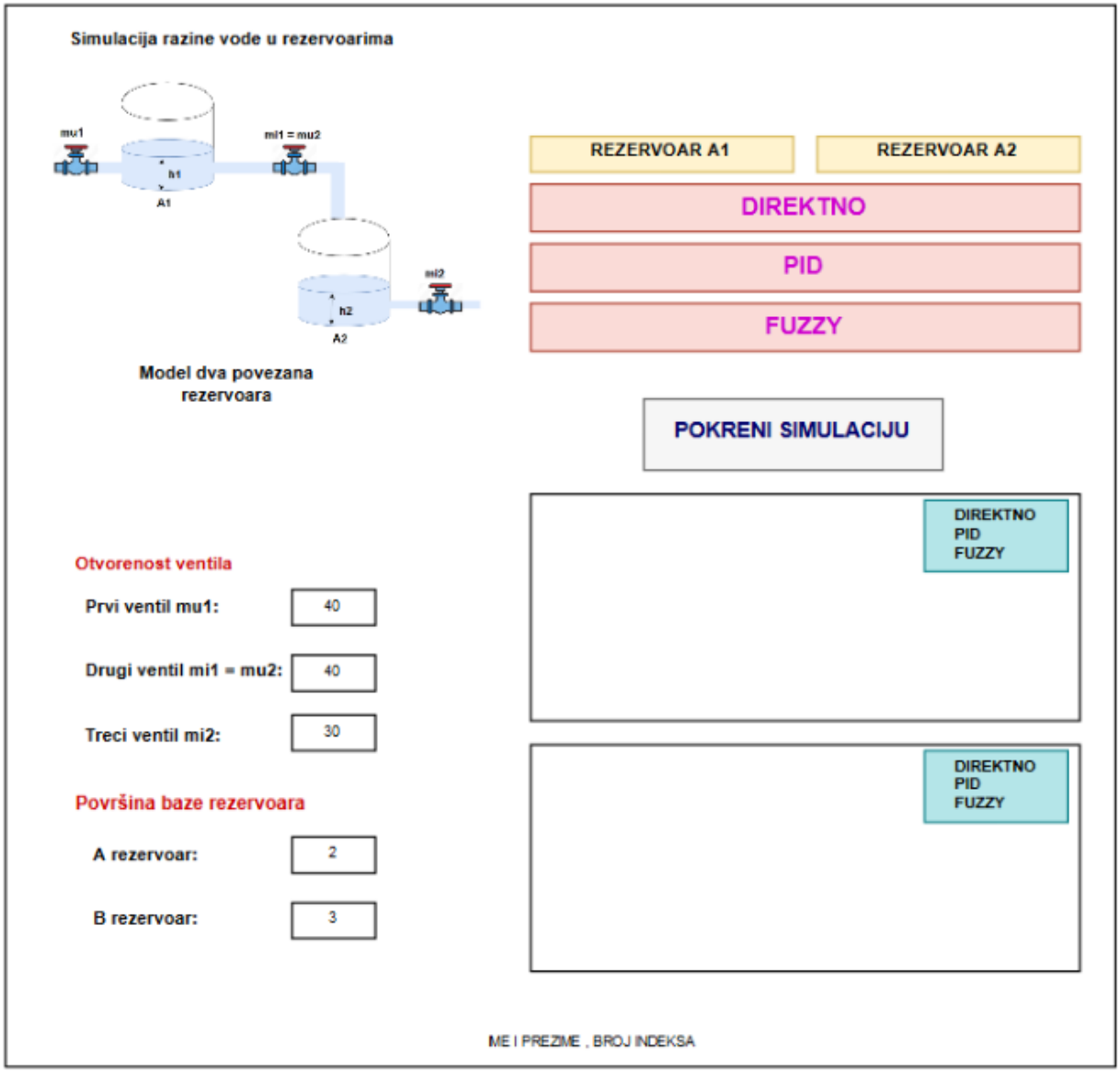
Površina prvog rezervoara: $A_1 = 2\ m^2$

Površina drugog rezervoara: $A_2 = 3\ m^2$

Ventili: mu_1= $mi_1 = mu_2 = 40$ [m m2s-1]

mi_2 =30 [m m2s-1]





$$A1 = \frac{dh_1(t)}{dt} = \left[\frac{(\Delta H - h_1(t))}{mv_1} - \frac{h_1(t)}{mv_1}\right]$$

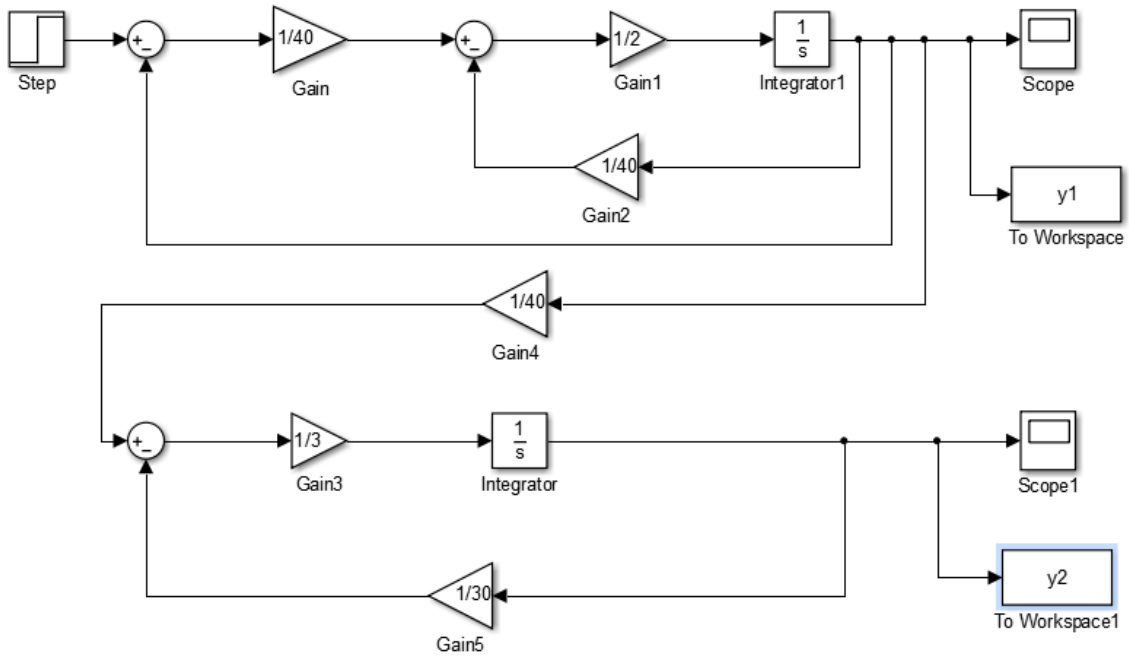
$$A2 = \frac{dh_2(t)}{dt} = \left[\frac{(h_1(t))}{mi_1} - \frac{h_2(t)}{mi_2}\right]$$

$$\frac{dh_1(t)}{dt} = \frac{1}{A_1} \cdot \left[\frac{(\Delta H - h_1(t))}{mv_1} - \frac{h_1(t)}{mi_1}\right]$$

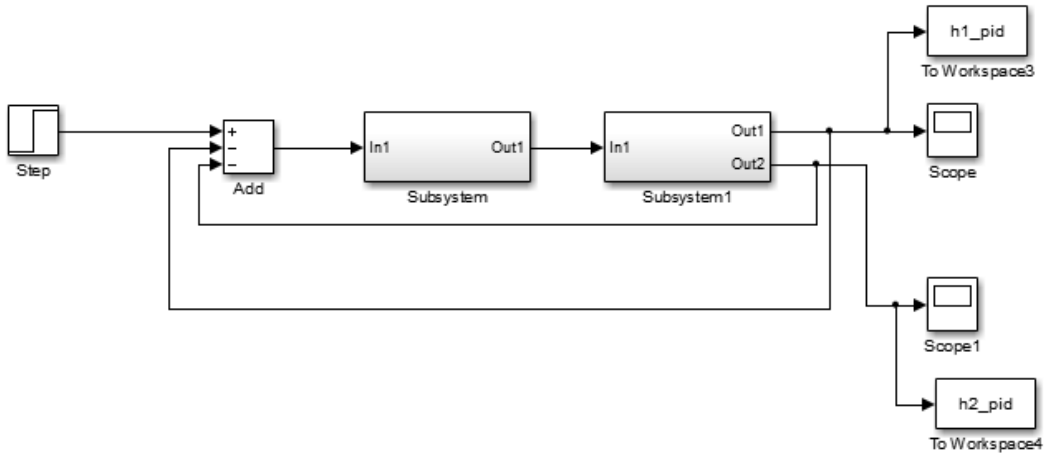
$$\frac{dh_2(t)}{dt} = \frac{1}{A_2} \cdot \left[\frac{(h_1(t))}{mi_1} - \frac{h_2(t)}{mi_2}\right]$$

$$G(s) = \frac{1}{9600s^2 + 316.6s + 2.66}$$

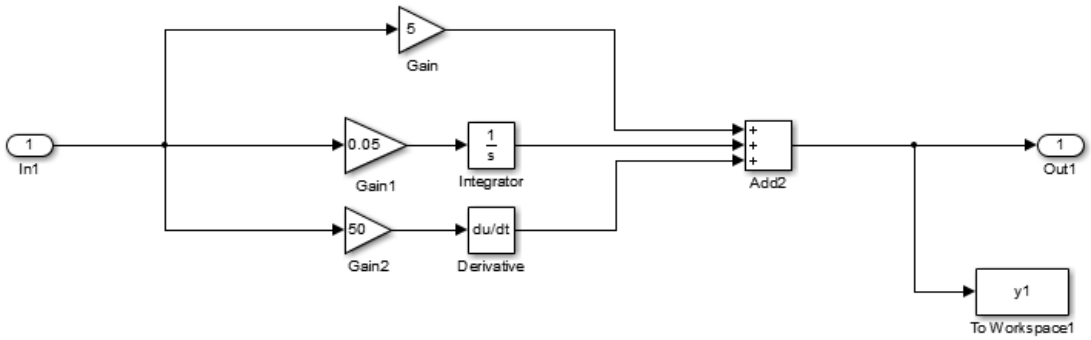
a)



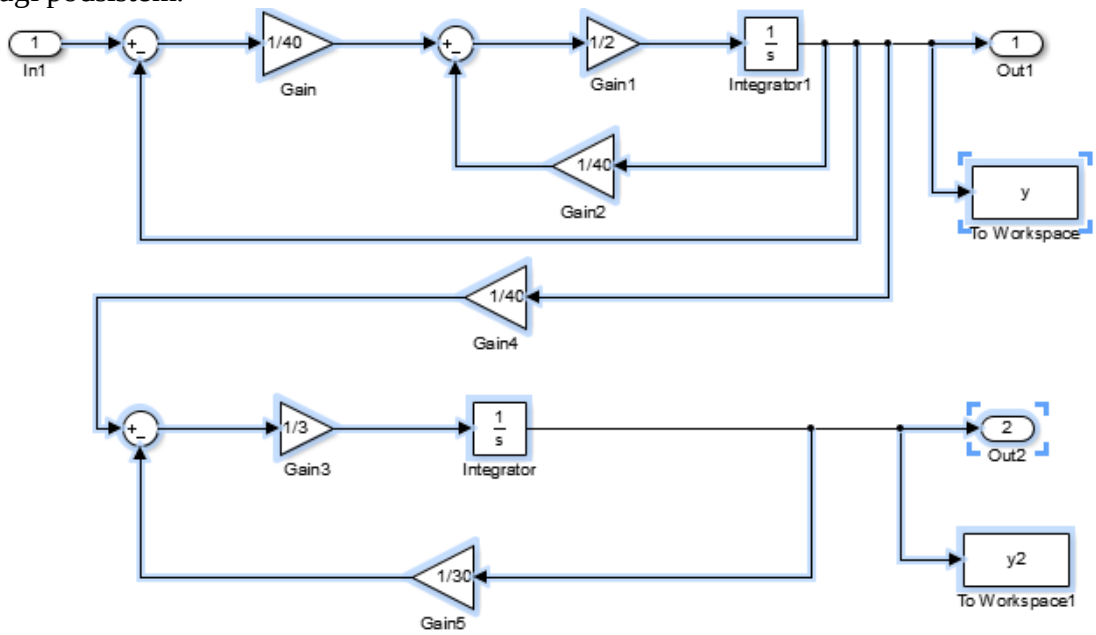
b)



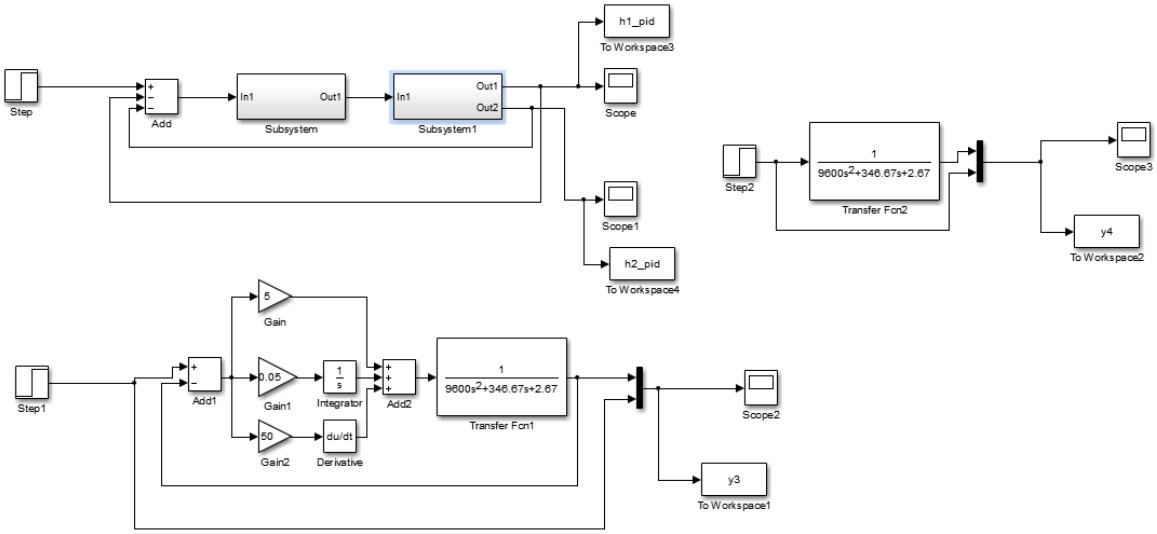
Prvi podsistem:



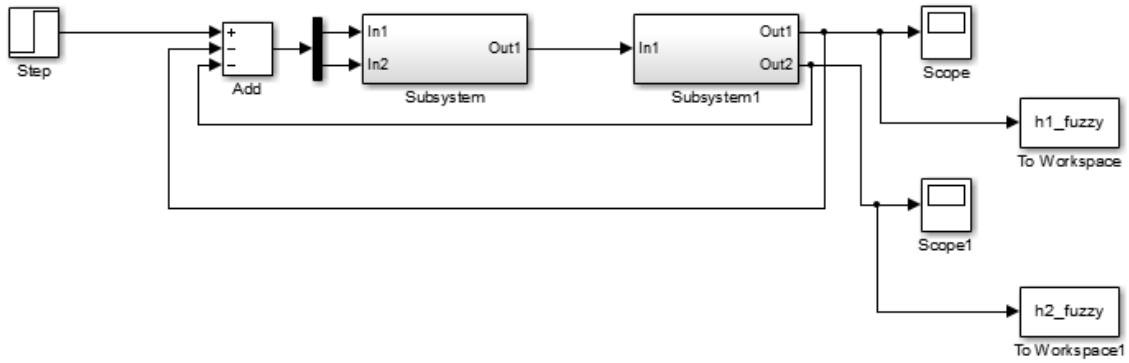
Drugi podsistem:



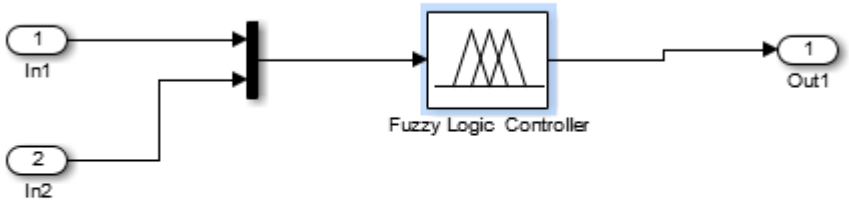
Rješavanje PID-a:



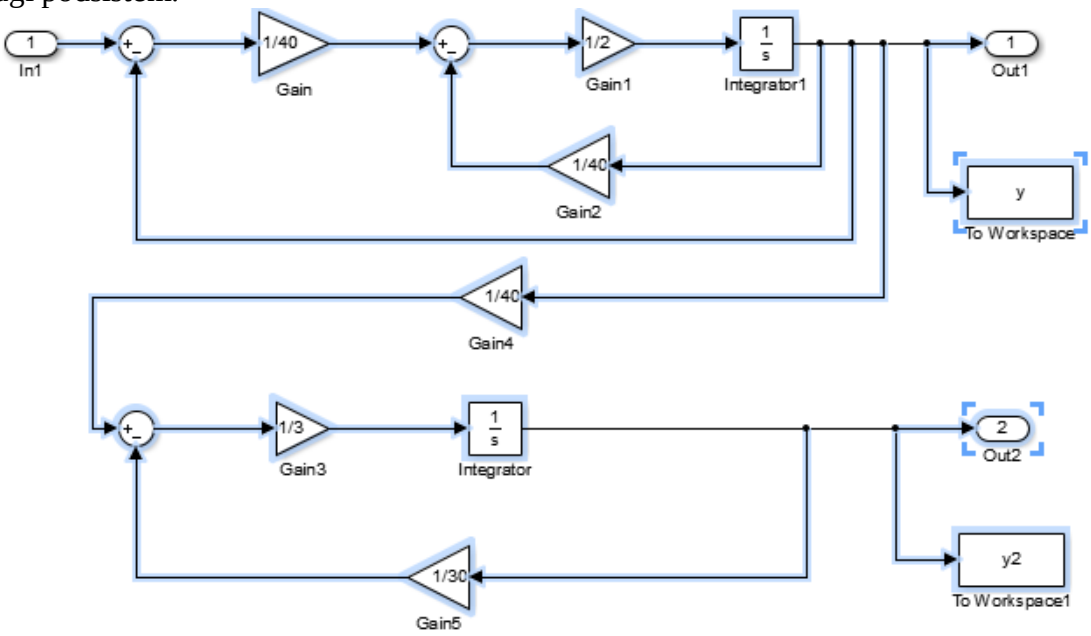
c)



Prvi podsistem:



Drugi podsistem:



Tabele za fuzzy:

NIVO

Nizak	Trapezoidal	-2 -1 -0.9 0	-1 1
Malo_nizak	Trapezoidal	-0.8 -0.6 -0.4 -0.2	
Dobar	Triangular	-0.3 0 0.3	
Malo_visok	Triangular	0.2 0.4 0.6 0.8	
Visok	Triangular	0 0.9 1 0	

Promjena

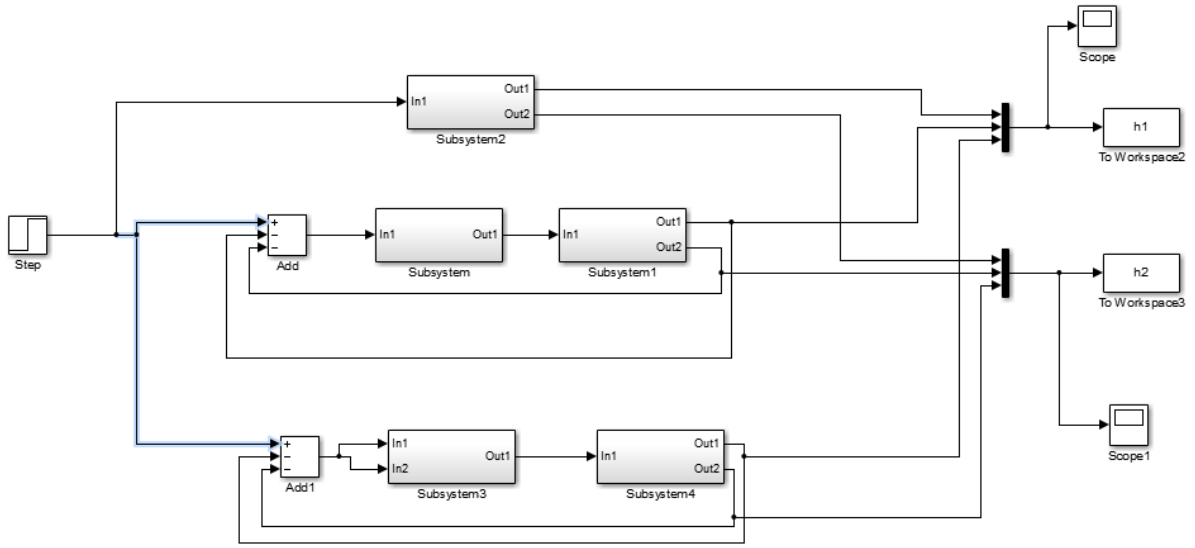
Negativna	Trapezoidal	-0.3987 -0.27 -0.2077 0.1	[-0.27 0.27]
Neznatna	Triangular	-0.1 0 0.1	
Pozitivna	Trapezoidal	0.1 0.2077 0.27 0.3738	

Ventil

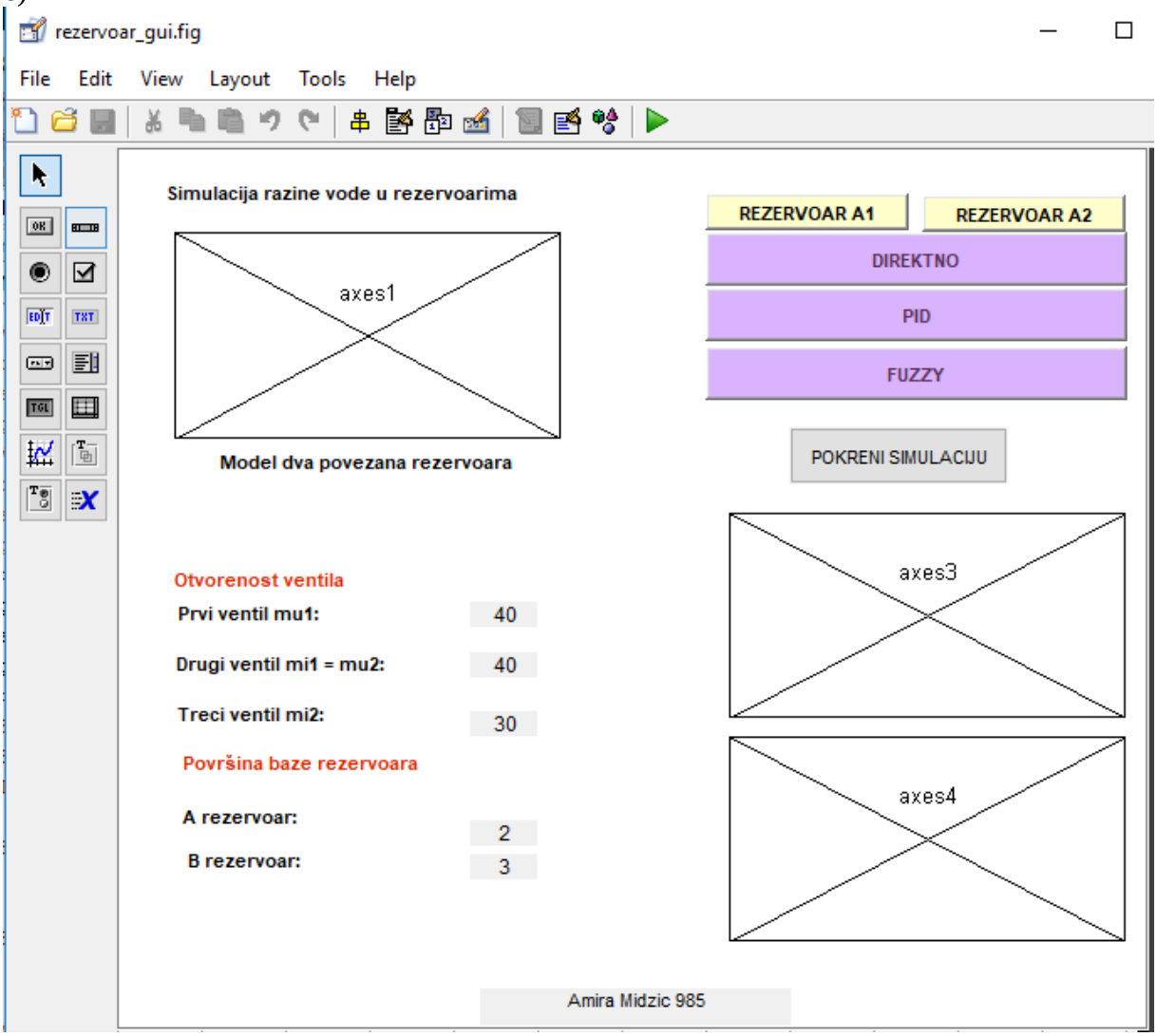
Zatvori brzo	Triangular	-0.825 -0.55 -0.275	[-0.55 0.55]
Zatvori polako		-0.55 -0.4 -0.25	
Malo_zatvori		-0.35 -0.2 -0.05	
Bez promjene		-0.1375 0 0.1375	
Malo otvori		0.05 0.2 0.35	
Otvori polako		0.25 0.4 0.55	
Otvori brzo		0.275 0.55 0.825	

IF NIVO	VISOK	THEN VENTIL IS OTVORI BRZO	
	NIZAK		THEN VENTIL ZATVOIR BRZO
	DOBAR	AND PROMJENA IS NEGATIVNA	OTVRI POLAKO
	DOBAR	AND PROMJENA IS POZITIVNA	ZATVORI POLAKO
	DOBAR	AND PROMJENA IZ NEZNATNA	BEZ PROMJENE
	MALO NIZAK	AND PROMJENA IZ NEZNATNA	MALO ZATVORI
	MALO VISOK	AND PROMJENA IZ NEZNATNA	MALO OTVORI

d)


Scope
Out1
In1
Out2
Subsystem2
h1
To Workspace2
Step
Add
In1
Out1
Subsystem
In1
Out1
Out2
Subsystem1
h2
To Workspace3
Add1
In1
In2
Out1
Subsystem3
In1
Out1
Out2
Subsystem4
Scope1

e)


rezervoar_gui.fig
File Edit View Layout Tools Help
Simulacija razine vode u rezervoarima
axes1
Model dva povezana rezervoara
REZERVOAR A1
REZERVOAR A2
DIREKTNO
PID
FUZZY
POKRENI SIMULACIJU
axes3
axes4
Otvorenost ventila
Prvi ventil mu1:
40
Drugi ventil mi1 = mu2:
40
Treci ventil mi2:
30
Površina baze rezervoara
A rezervoar:
2
B rezervoar:
3
Amira Midzic 985

```
function pokreni_Callback(hObject, eventdata, handles)
% hObject    handle to pokreni (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of pokreni

set_param('rezervoar_a','SimulationCommand','Start')
set_param('rezervoar_b_pid','SimulationCommand','Start')
set_param('rezervoar_c_fuzzy','SimulationCommand','Start')
set_param('rezervoar_d_sve','SimulationCommand','Start')

r1 = get(handles.rezervoar1, 'Value');
r2 = get(handles.rezervoar2, 'Value');
dir = get(handles.direktno, 'Value');
p = get(handles.pid, 'Value');
f = get(handles.fuzzy, 'Value');
prvi = evalin('base', 'h1');
drugi = evalin('base', 'h2');
 y1 = evalin('base', 'y1');
 h1_pid = evalin('base', 'h1_pid');
 h1_fuzzy = evalin('base', 'h1_fuzzy');
  y2 = evalin('base', 'y2');
 h2_pid = evalin('base', 'h2_pid');
 h2_fuzzy = evalin('base', 'h2_fuzzy');

if(r1==1)
axes(handles.axes3)
if(dir==1 && p ==0 && f==0)
    plot(y1), legend('Jednacine'), grid on;
end
if(dir==0 && p ==1 && f==0)
    plot(h1_pid), legend('PID'), grid on;
end
if(dir==0 && p ==0 && f==1)
    plot(h1_fuzzy), legend('Fuzzy'), grid on;
end
if(dir==1 && p ==1 && f==0)
    plot(0:length(y1)-1, y1, 0:length(h1_pid)-1, h1_pid),
    legend('Jednacine', 'PID'), grid on;
end
if(dir==1 && p ==0 && f==1)
    plot(0:length(y1)-1, y1, 0:length(h1_fuzzy)-1, h1_fuzzy),
    legend('Jednacine', 'Fuzzy'), grid on;
end
if(dir==0 && p ==1 && f==1)
    plot(0:length(h1_pid)-1, h1_pid, 0:length(h1_fuzzy)-1, h1_fuzzy),
    legend('PID', 'Fuzzy'), grid on;
end
if(dir==1 && p ==1 && f==1)
    plot(0:length(prvi)-1, prvi), legend('Jednacine', 'PID', 'Fuzzy'),
    grid on;
end
end

if(r2==1)
axes(handles.axes4)

if(dir==1 && p ==0 && f==0)
    plot(y2), legend('Jednacine'), grid on;
```

```
end
if(dir==0 && p ==1 && f==0)
    plot(h2_pid), legend('PID'), grid on;
end
if(dir==0 && p ==0 && f==1)
    plot(h2_fuzzy), legend('Fuzzy'), grid on;
end
if(dir==1 && p ==1 && f==0)
    plot(0:length(y2)-1, y2, 0:length(h2_pid)-1, h2_pid),
    legend('Jednacine', 'PID'), grid on;
end
if(dir==1 && p ==0 && f==1)
    plot(0:length(y2)-1, y2, 0:length(h2_fuzzy)-1, h2_fuzzy),
    legend('Jednacine', 'Fuzzy'), grid on;
end
if(dir==0 && p ==1 && f==1)
    plot(0:length(h2_pid)-1, h2_pid, 0:length(h2_fuzzy)-1, h2_fuzzy),
    legend('PID', 'Fuzzy'), grid on;
end
if(dir==1 && p ==1 && f==1)
    plot(0:length(drugi)-1, drugi), legend('Jednacine', 'PID', 'Fuzzy'),
    grid on;
end
end
```

Genetski algoritam 1

1. Napisati program koji će predstavljati određivanje optimuma funkcije pod a, b i c upotrebom integrisane funkcije genetskog algoritma u MATLAB-u.

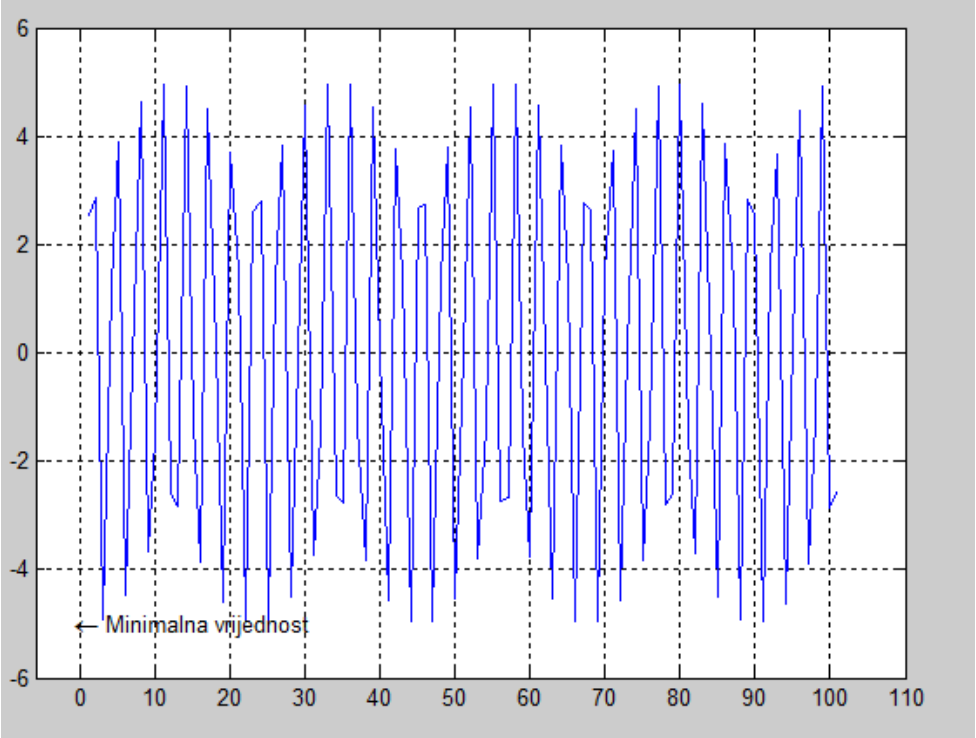
a) $z = 5\sin(2x)$

b) $z = y^2 + x^2$

c) $z = \sin(x^2) + y^2$

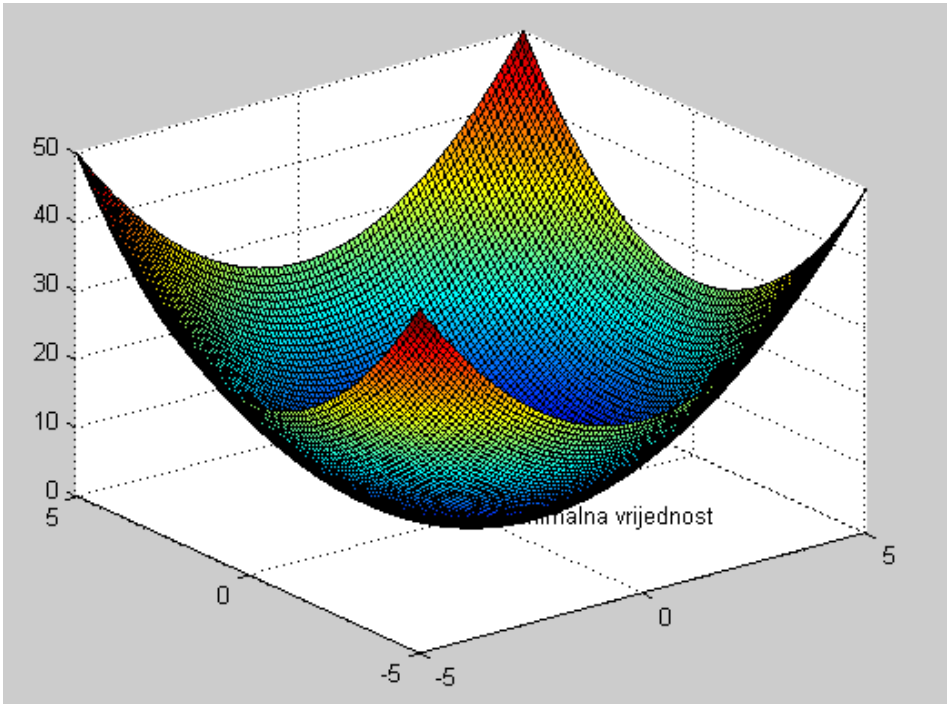
a)

```
x = -50:1:50;
z = 5*sin(2*x); %FUNKCIJA
plot(z);
[x, f, m, n, k, v] = ga(@(x) 5*sin(2*x(1)), 1); %NAREDBA ZA POKRETANJE
text(x, f, '\leftarrow Minimalna vrijednost');
axis([-6 110 -6 6]);
grid ON
```



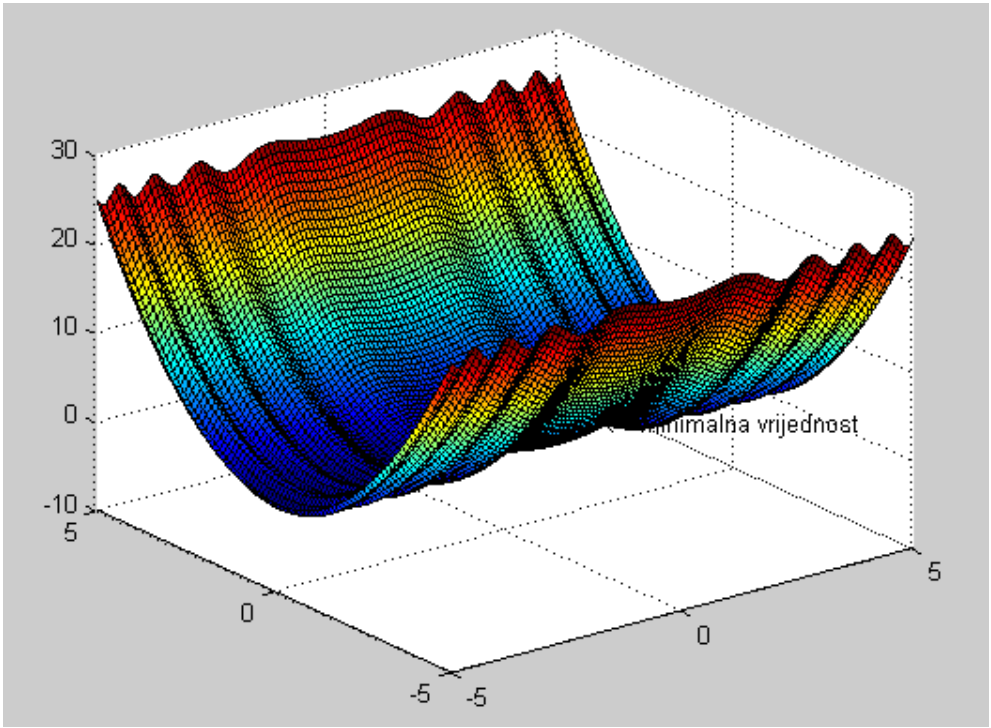
b)

```
[x, y] = meshgrid(-5:0.1:5, -5:0.1:5);
z = y.^2+x.^2;
surf(x, y, z);
[x, f, m, n, k, v] = ga(@(x) x(2).^2+x(1).^2,2);
text(x(1), x(2), f, '\leftarrow Minimalna vrijednost');

%KLASI?AN NA?IN TRAŽENJA MINIMUMA FUNKCIJE
r = min(z)
disp('Minimum funkcije');
h = min(r)
```



c)

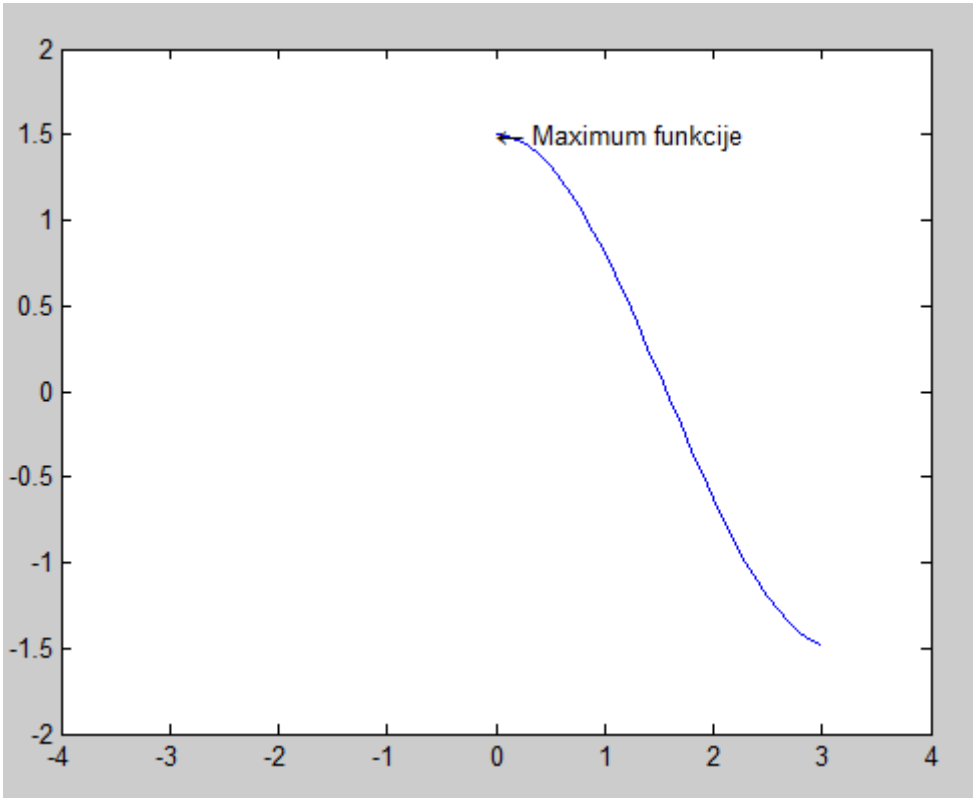
```
[x, y] = meshgrid(-5:0.1:5, -5:0.1:5);
z = sin(x.^2)+y.^2;
surf(x, y, z);
[x, f, m, n, k, v] = ga(@(x) sin(x(1).^2)+x(2).^2,2);
text(x(1), x(2), f, '\leftarrow Minimalna vrijednost');

%KLASI?AN NA?IN TRAŽENJA MINIMUMA FUNKCIJE
r = min(z)
disp('Minimum funkcije');
h = min(r)
```



2. Napisati programski kod u MATLAB-u koji će pronaći maksimum funkcije $\frac{3}{2}\cos(t)$ koristeći mutacijsko-selekcijski algoritam (simulirano kaljenje)

```
period = 0:0.1:3;
funkcija = (3/2)*cos(period); %FUNKCIJA U KOJOJ SE TRAŽI MAKSIMUM
plot(period, funkcija);
axis([-4 4 -2 2]);

%PRVI KORAK: IZBOR SLU?AJNOG HROMOSOMA X
xx = randi([1 30]); %NASUMICNI INTEGER

%DRUGI KORAK: ODREDIVANJE FUNKCIJE SPOSOBNOSTI POCETNOG HROMOSOMA
funkcija_sposobnosti_xx = funkcija(xx)

uslov = 1;
zastavica = true;

%PETLJA KOJA OMOGU?AVA POVRATAK SA 7. NA 3. KORAK
while zastavica == true
    % TRECI KORAK: PROMJENA X U X' (MUTACIJA)
    ni = 0.01;
    z = randi([-1 1]);
    delta = ni*z;
    xPrim = int8(xx+delta);

    %CETVRTI KORAK: ODREDIVANJE FUNKCIJE SPOSOBNOSTI OD X'
    funkcija_sposobnosti_xPrim = funkcija(xPrim);

    %PETI KORAK: ODREDIVANJE d
    d = funkcija_sposobnosti_xPrim-funkcija_sposobnosti_xx;

    %SESTI KORAK: a) racunanje p(d)
    T = 0.1;
    p = 1/1+exp(-d/T);

    %SESTI KORAK: b) NOVO z
    z_novo = randi([0 1]);

    %SESTI KORAK: c) z_novo<p
    if z_novo<p
        xx = xPrim;
    end

    %SESTI KORAK: d) T-ni
    T = 0.1-0.01;

    %SEDMI KORAK: USLOV PREKIDA
    if uslov == 1000
        zastavica =false;
    end

    uslov = uslov+1;
end
koordinata1 = double(xx)/1000;
```

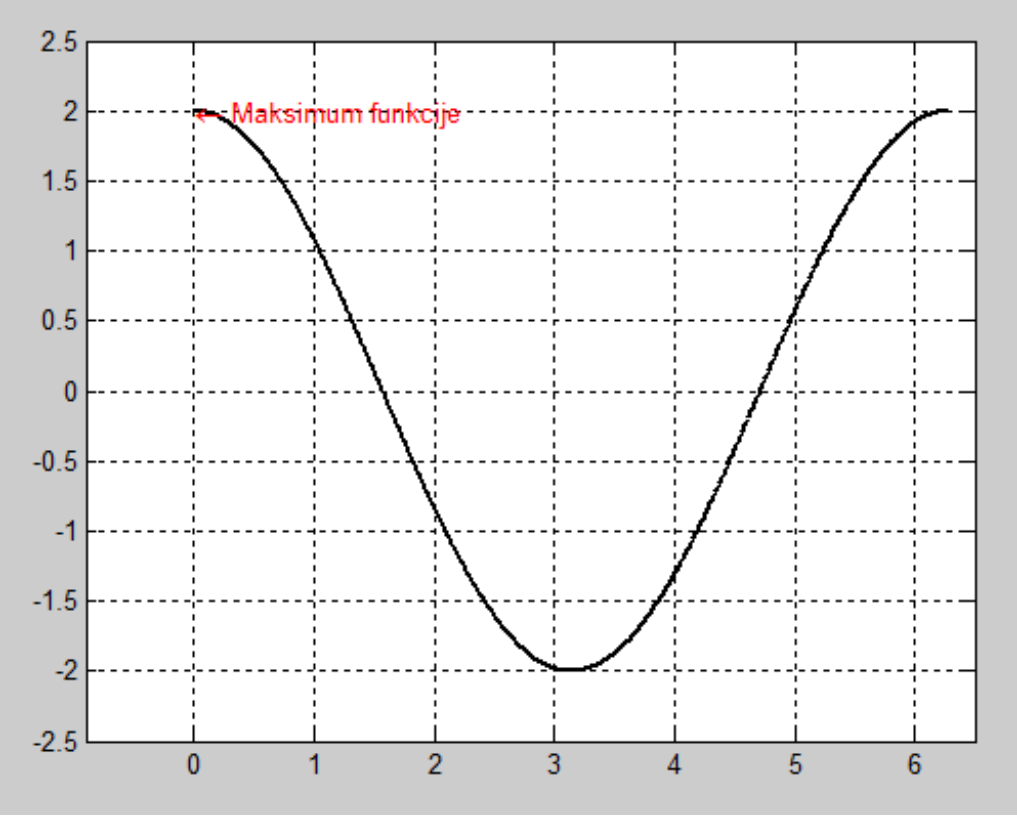
```
koordinata2 = 3/2;
text(koordinata1, koordinata2, '\leftarrow Maximum funkcije');
```



3. Za funkciju $2\cos(t)$ napisati program koji će predstavljati određivanje maksimuma funkcije upotrebom mutacijsko-selekcijskog algoritma

```
clc
clear all
t = 0:0.01:2*pi;
f = 2*cos(t);
plot(t,f,'k','linewidth',2); grid on;
axis([-0.9 6.5 -2.5 2.5]);

% 1. Izbor slučajnog hromosoma x
x = randi([1 length(f)]);

% 2. Funkcija sposobnosti x
fx = f(x);

uslov = 1;
zast = true;

while zast == true

% 3. Mutacija (x->x')
ni = length(f)*rand(1);
z = randi([-1 1]);
delta = ni*z;
xp = int8(x+delta);

zast1 = true;
while zast1 == true
```

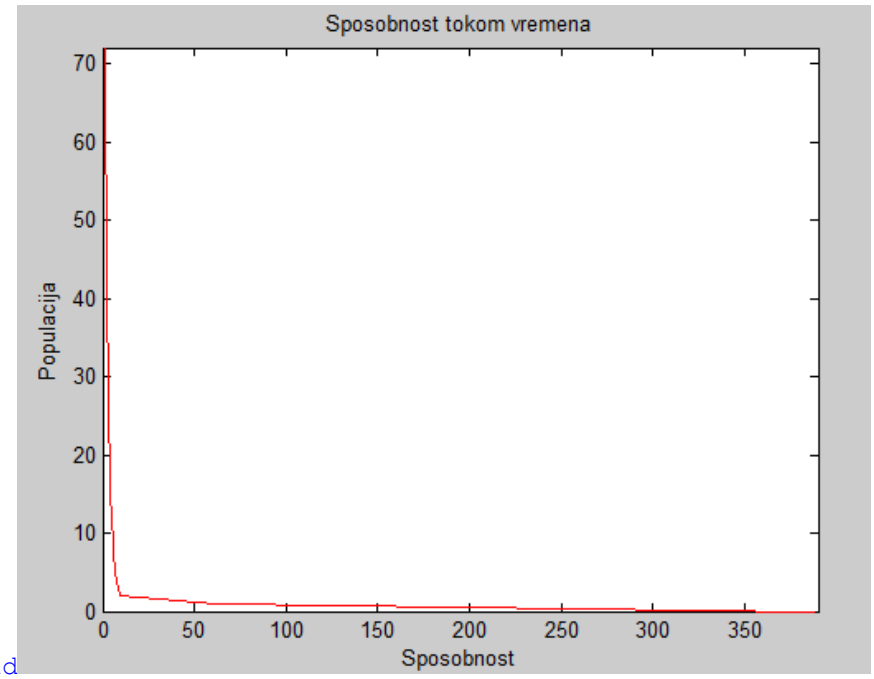
```
    zast1 = false;
    if (xp>length(f) || xp<=0)
        ni = length(f)*rand(1)
        delta = ni*z;
        xp = int8(x+delta);
        zast1 = true;
    end
end

% 4. Funkcija sposobnosti x'
fxp = f(xp);

% 5. Razlika funkcija sposobnosti (određivanje boljeg hromosoma)
d = fxp - fx;

% 6. Odabiranje hromosoma
if d>0
    x = xp;
end

% 7. Izlaz iz petlje
if uslov == 50
    zast = false;
end
uslov = uslov+1;
end

k1 = double(x)/50;
k2 = 2;
text(k1, k2, '\leftarrow Maksimum funkcije','color','r');
```



4. Napisati programski kod kojim će se odrediti maksimum funkcije od kreirane slučajne populacije upotrebom genetskog algoritma

```
clear all
clc
tic %POCETAK TAJMERA
tekst = 'Tekst';

popDuzina = 1000;
genome = length(tekst);
stopaMutacije = 0.001;
brojac = 1;
S = 4;
najbolji = Inf;
maxVrijednost = max(double(tekst));
idealno = double(tekst);

selekcija=0;
krizanje = 1;
iscrtavanje = 1;

%POSTAVLJANJE POCETKA SLUCAJNE POPULACIJE
Pop = round(rand(popDuzina, genome)*(maxVrijednost-1)+1);

initF = min(sum(abs(bsxfun(@minus, Pop, idealno)), 2);

for Gen = 1:1e6
    F = sum(abs(bsxfun(@minus, Pop, idealno)), 2);
    [trenutni, trenutniGen] = min(F);

    if trenutni < najbolji
        najbolji=trenutni;
        najboljiGen = Pop(trenutniGen);

        if iscrtavanje == 1
            B(brojac) = najbolji;
            G(brojac) = Gen;
            brojac = brojac+1;
        end

        fprintf('Gen: %d Sposobnost: %d', Gen, najbolji);
        disp(char(najboljiGen));
    elseif najbolji == 0
        break
    end

    %ODABIR PAROVA RODITELJA
    if selekcija == 0
        T = round(rand(2*popDuzina, S)*(popDuzina-1)+1);
        [~, idx] = min(F(T), [], 2);
        W = T(sub2ind(size(T), (1:2*popDuzina)', idx));
    elseif selekcija == 1
        [~, V] = sort(F, 'descend');
        V = V(popDuzina/2+1:end);
        W = V(round(rand(2*popDuzina, 1)*(popDuzina/2-1)+1))';
    end

    %KRIZANJE

    if krizanje == 0
```

```
        idx = logical(round(rand(size(P))));
        Pop2 = Pop(W(1:2:end), :);
        P2A = Pop(W(2:2:end), :);
        Pop2(idx) = P2A(idx);

    elseif krizanje==1
        Pop2 = Pop(W(1:2:end), :);
        P2A = Pop(W(2:2:end), :);
        Ref = ones(popDuzina, 1)*(1:genome);
        idx = (round(rand(popDuzina, 1) *(genome-1)+1)*ones(1, genome)) >
Ref;
        Pop2(idx) = P2A(idx);
    elseif krizanje == 2
        Pop2 = Pop(W(1:2:end), :);
        P2A = Pop(W(2:2:end), :);
        Ref = ones(popDuzina, 1)*(1:genome);
        CP = sort(round(rand(popDuzina,2) * (genome-1)+1), 2);
        idx = CP(:, 1)*ones(1, genome)<Ref&CP(:, 2)*ones(1, genome)>Ref;
    end

    %MUTACIJA
    idx = rand(size(Pop2))<stopaMutacije;
    Pop2(idx) = round(rand([1, sum(sum(idx))])*(maxVrijednost-1)+1);

    %RESETOVANJE POPULACIJE
    Pop = Pop2;
end

toc

if iscrtavanje ==1
    figure(1)
    plot(G(:), B(:), '-r');
    axis([0 Gen 0 initF]);
    xlabel('Sposobnost'); ylabel('Populacija'), title('Sposobnost tokom
vremena');
```

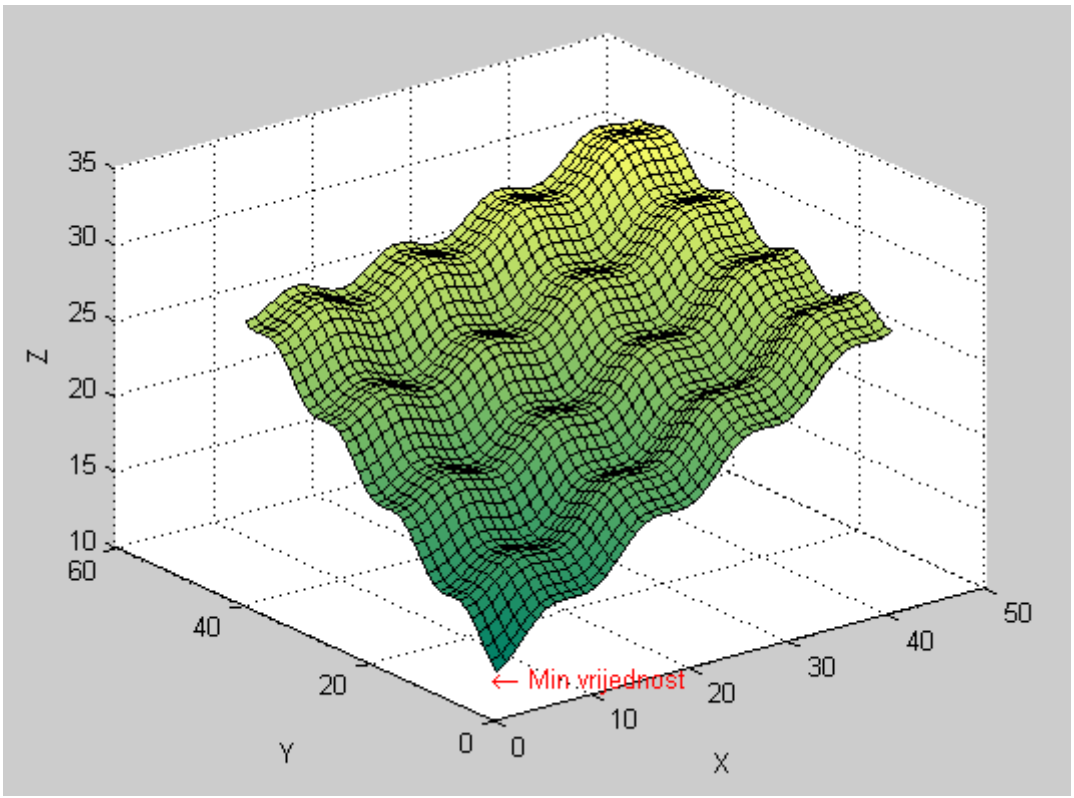



```
end
```

Genetski algoritam 2

1. Nađi minimum funkcije na intervalu 0-5, 0-5.

```
close all
%zadatak 1
%naciminimum funkcije naintervalu 0-5, 0-5
[x,y] = meshgrid (0:0.1:4, 0:0.1:4);
c=2.*pi;
z=1/2.*(-20.*exp(1).*(-0.20.*sqrt(1/2.*(x.^2+y.^2)))-
exp(1).*(1/2.*(cos(c.*x)+cos(c.*y)))+20+exp(1)+5.7);
disp('Minimalna vrijednost rucno: ')
b=min(min(z));
[x,f]=ga(@(x) 1/2.*(-20.*exp(1).*(-0.20.*sqrt(1/2.*(x(1).^2+x(2).^2)))-
exp(1).*(1/2.*(cos(c.*x(1))+cos(c.*x(2))))+20+exp(1)+5.7),2,[],[],[],...
 [],[0 0], [4 4]);
x
x(1)
x(2)
disp('Minimalna vrijednost ga: ')
f
figure

surf(z)
colormap summer
xlabel('X');
ylabel('Y');
zlabel('Z');
text(x(1),x(2),f,'\leftarrow Min
vrijednost','FontSize',10,'Color','red','linewidth',2.5);
```



2. Nađi minimum funkcije:

$$f(x) = \begin{cases} -\exp\left(-\left(\frac{x}{20}\right)^2\right) & za\ x \leq 20 \\ -\exp(-1) + (x-20)(x+22) & za\ x > 20 \end{cases}$$

```
close all
clc
clear
x=0:0.1:22; %proizvoljno stavljamo

if max(x)<=20
    y1=-exp(-(x./20).^2);
    plot(x,y1);
a=min(y1)

[x,f]=ga(@(x) -exp(-x(1)./20).^2,1,[],[],[], ...
    [],[0],[20]);
x
f
elseif max(x)>20
    y=-exp(-1)+(x-20).*(x+22);

    plot(x,y);
    b=min(y)
    [x,f]=ga(@(x) -exp(-1)+(x(1)-20).*(x(1)+22),1,[],[],[], ...
    [],[0],[21]);
    x
    f
end
text(x(1),f,'\leftarrow Min
vrijednost','FontSize',10,'Color','red','linewidth',2.5)
```