

Univerzitet u Tuzli  
Fakultet elektrotehnike  
Predmet: Inteligentni sistemi  
Akademska godina: 2021./2022.  
Student: Indir Karabegović  
Datum: 31.05.2022. god.

## Izvještaj zadatke broj 3

### Zadatak br. 1 (Metoda zaključivanja: Mamdani; Defuzifikacija: Centroid)

Kreirati Fuzzy sistem zaključivanja *Mamdani* tip za usmjeravanje mobilnog robota na stanicu za punjenje.

Ulazni parametri su:

- udaljenost od stanice (0-10m) i
- ugao konektora na robotu u odnosu na konektor na stanici ( $-90^{\circ} \dots + 90^{\circ}$ )

Izlazni parametri su:

- promjena ugla robota prema stanici kako bi se ostvario prilaz stanici pod uglom 0.

Ideja je da se robot približava najkraćim putem prema stanici, osim kada je jako blizu, tada mora napraviti manevar uključivanja pod uglom  $0^{\circ}$ .

Tip i broj membership funkcija odrediti proizvoljno, kao i fuzzy pravila. Kao rješenje, potrebno je:

- Priložiti parametre MF ulaza i izlaza, listu pravila
- Grafički prikazati (screenshot) 3 karakteristična slučaja iz Rule viewera i dati komentar
- Priložiti Surface viewer, površina ne bi smjela imati lokalnih minimuma/maksimuma.

### Rješenje:

Shodno postavci zadatka, imamo dva ulaza i jedan izlaz. Za ulaze smo uzeli:

- udaljenost
- ugao

Dok smo kao izlaz uzeli:

- promjena-ugla

Odgovarajuće MF (Membership functions) i njihovi parametri su prikazani u narednim tabelama:

### 1. Udaljenost

Br.	Naziv	Tip	Parametri:
1	skoro-nula	trapmf	[0 0 0.87 2.51]
2	mala	trapmf	[0.531 2.67 3.41 6.22]
3	srednja	trapmf	[2.8 6.16 6.67 9.81]
4.	velika	trapmf	[6.53 8.91 10.6 12.2]

### 2. Ugao

Br.	Naziv	Tip	Parametri
1.	neg-veliki	trapmf	[-140 -95 -85 -40]
2.	neg-mali	trapmf	[-85 -35 -20 5]
3.	nula	trimf	[-20 0 20]
4.	poz-mali	trapmf	[-5 20 35 85]
5.	poz-veliki	trapmf	[40 85 95 140]

### 3. Promjena – ugla

Br.	Naziv	Tip	Parametri
1.	neg-velika	trapmf	[-140 -110 -85 -60]
2.	neg-srednja	trapmf	[-110 -70 -60 -40]
3.	neg-mala	trapmf	[-70 -35 -25 10]
4.	nula	trimf	[-20 0 20]
5.	poz-mala	trapmf	[-10 25 35 70]
6.	poz-srednja	trapmf	[40 60 70 110]
7.	poz-velika	trapmf	[60 85 110 140]

Nakon što smo specificirali odgovarajuće MF, u narednoj tabeli smo kreirali listu pravila (u ovoj tabeli imamo 20 pravila iz razloga što imamo 20 mogućih kombinacija na ulazu):

#### 4. Logika

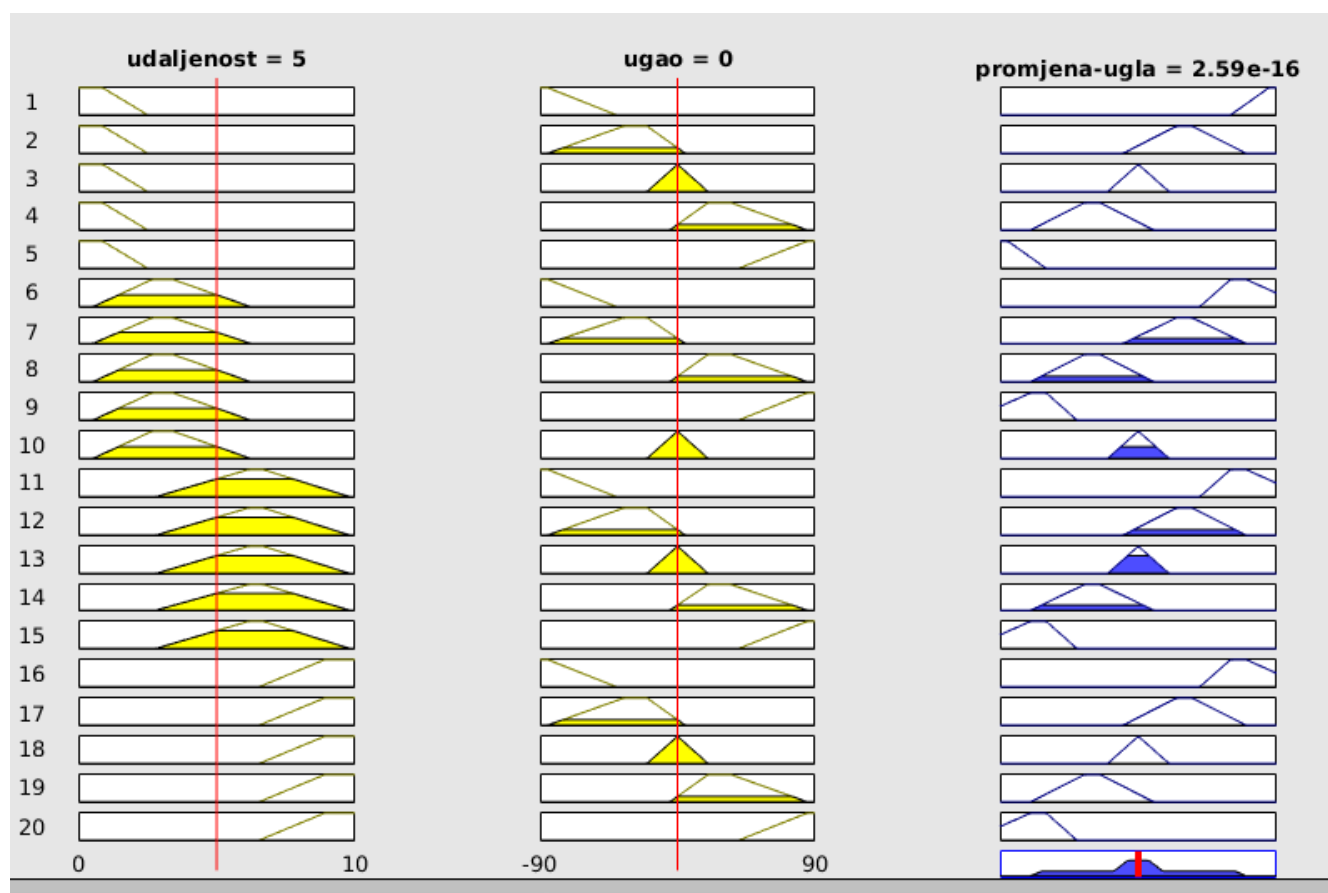
Br.	Udaljenost	Ugao	Promjena-ugla
1.	skoro-nula	neg-veliki	poz-velika
2.	skoro-nula	neg-mali	poz-mala
3.	skoro-nula	nula	nula
4.	skoro-nula	poz-mali	neg-mala
5.	skoro-nula	poz-veliki	neg-velika
6.	mala	neg-veliki	poz-srednja
7.	mala	neg-mali	poz-mala
8.	mala	nula	nula
9.	mala	poz-mali	neg-mala
10.	mala	poz-veliki	neg-srednja
11.	srednja	neg-veliki	poz-srednja
12.	srednja	neg-mali	poz-mala
13.	srednja	nula	nula
14.	srednja	poz-mali	neg-mala
15.	srednja	poz-veliki	neg-srednja
16.	velika	neg-veliki	poz-srednja
17.	velika	neg-mali	poz-mala
18.	velika	nula	nula
19.	velika	poz-mali	neg-mala
20.	velika	poz-veliki	neg-srednja

U prethodnoj tabeli je naznačena logika ponašanja za ovaj zadatak. Vodili smo se sljedećim zaključivanjem: Ako smo na veoma maloj udaljenosti od stanice (punjača), potrebno je da mobilni robot vrši manevar naglog

zakretanja (odnosno zakretanja za ugao koji ga vodi direktno na stanicu). Zbog toga smo u ovom slučaju robotu omogućili da vrši i pozitivno-veliku, odnosno negativno-veliku promjenu ugla. Na svim ostalim udaljenostima, robot može da vrši nultu promjenu ugla (ako mu je ugao u odnosu na stanicu nula), odnosno malu promjenu ugla ako mu je ugao u odnosu na stanicu mali, te srednju promjenu ugla, ako nam je robot pod znatnom većim uglom u odnosu na stanicu. Razlog zašto smo se vodili ovime jeste da nam senzori mogu dati pogrešnu informaciju, naročito na većim udaljenostima, te kako bismo spriječili da robot stalno vrši naglo manevarisanje, odnosno nagle pokrete, čime bi se i vrijeme i distanca povećali. Shodno tome, robot ima 7 stanja za promjenu ugla, koja su navedena u prethodnim tabelama. Također, ono što je bitno napomenuti, jeste to da, ukoliko je robot u odnosu na stanicu pod uglom npr od 60 stepeni, promjena ugla mora biti negativna, odnosno oko -60 stepeni, kako bi nam konačni ugao bio nula.

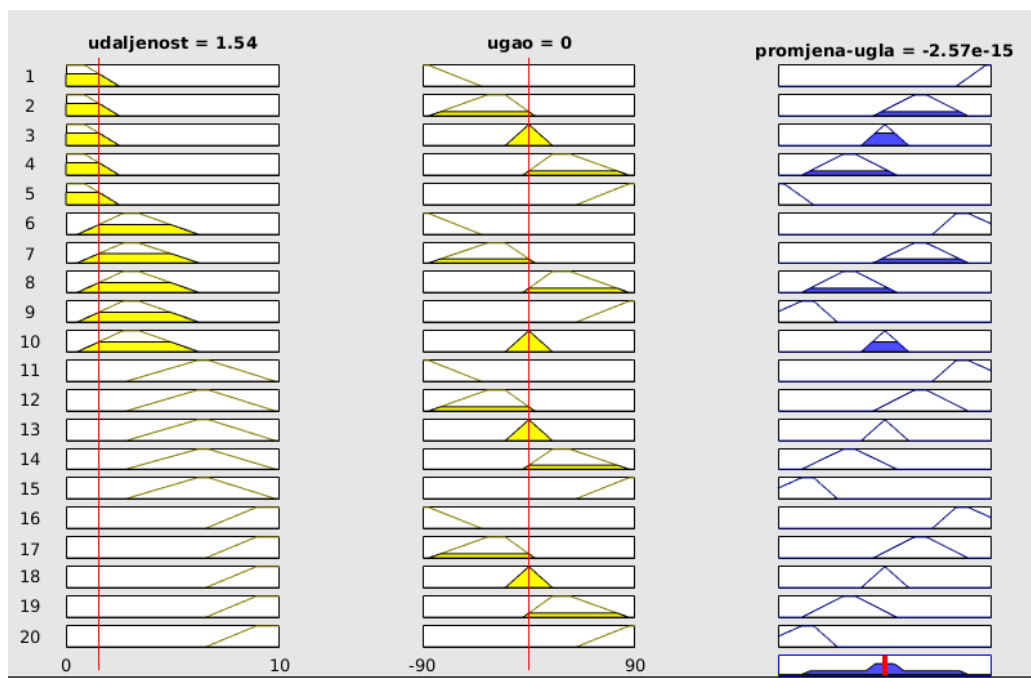
Na narednim slikama su prikazani slučajevi iz Rule viewer-a:

### 1. slučaj:



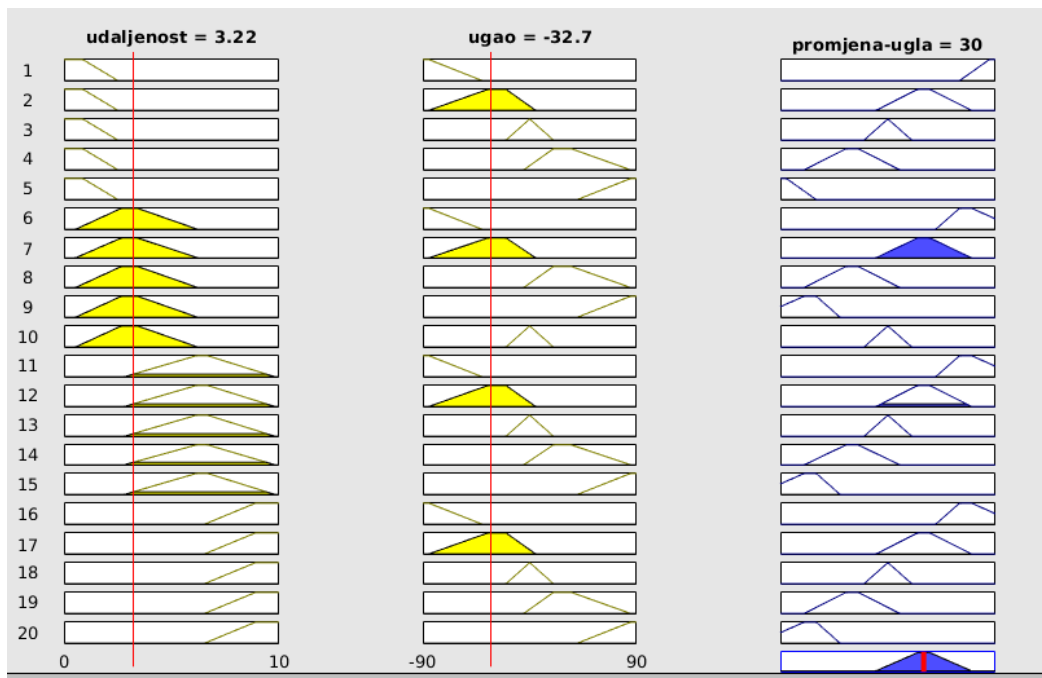
Kao što možemo vidjeti iz ovog slučaja, udaljenost u odnosu na stanicu (konektor) je 5 m, te je ugao u odnosu na stanicu jednak nuli, što znači da moramo imati izuzetno malu promjenu ugla (vidimo da kod nas ova promjena nije baš jednaka nuli, ali je izuzetno mala i iznosi  $2.59 \cdot 10^{-16}$ , što je u realnim uslovima skoro zanemarivo mali broj).

2. slučaj:



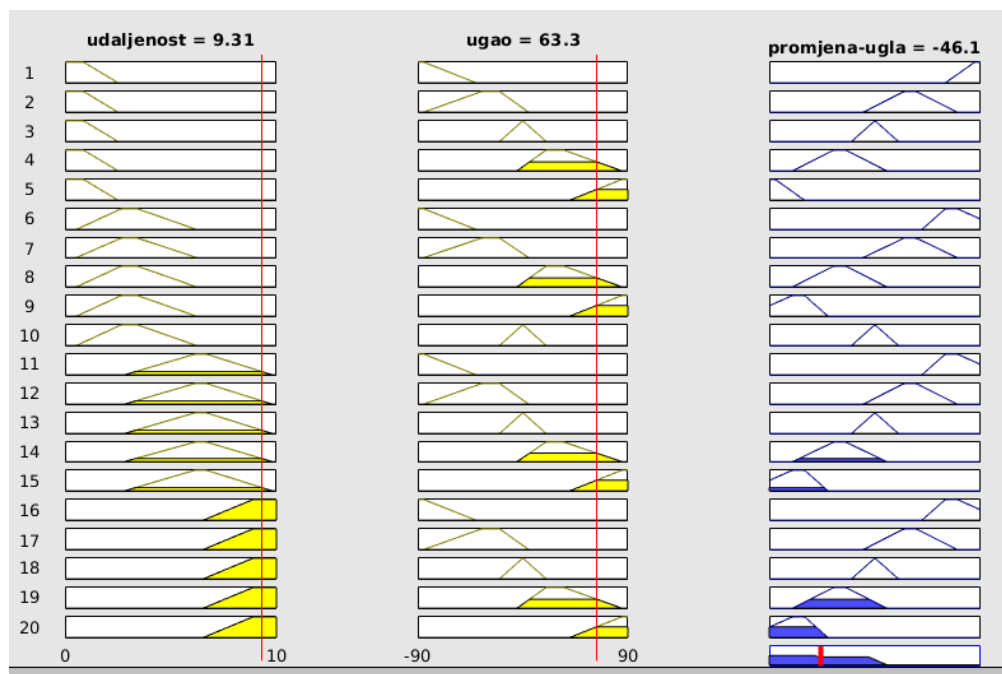
Također, kao i u prethodnom slučaju, vidimo da je udaljenost 1.54 te ugao u odnosu na stanicu je 0. Sama promjena ugla je i ovom slučaju znatno mala, skoro jednaka nuli.

3. slučaj:



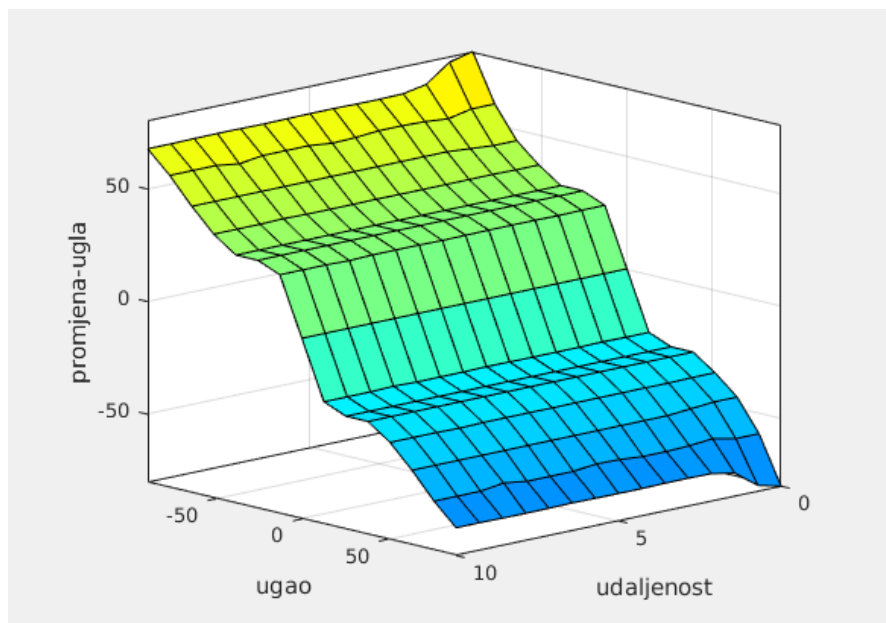
U ovom slučaju je udaljenost u odnosu na stanicu 3.22 m, dok je ugao mobilnog robota u odnosu na stanicu jednak -32.7 stepeni. Kao rezultat vidimo da je promjena ugla 30 stepeni, odnosno robot se kreće tako da mu se ugao u odnosu na stanicu cijelo vrijeme smanjuje, te da je cilj da ugao na kraju bude skoro nula. Vodili smo se time da će robot obrađivati podatke i mijenjati ugao kako se približava stanici. Shodno tome, analizirati ćemo još jedan slučaj kada je udaljenost znatno veća.

4. slučaj:



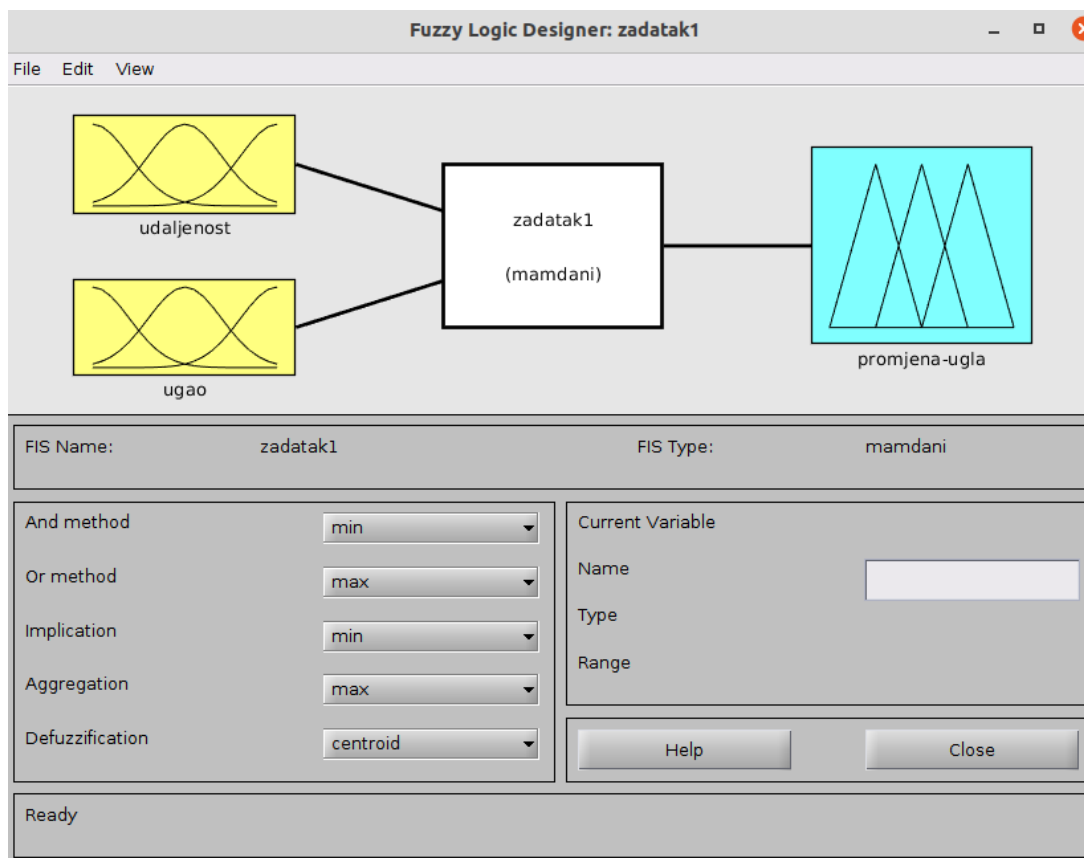
U ovom slučaju vidimo da nam logika ne daje najbolje rezultate, što smo i očekivali. Samom logikom, te izborom membership funkcija smo onemogućili nagle promjene uglova na velikim udaljenostima, te vidimo da na velikoj udaljenosti, od skoro 9 metara, imamo ugao od 63.3 stepeni a promjena ugla je -46.1 stepeni. Kada uradimo ovaku analizu, osobine koje nam se ne sviđaju možemo ispraviti promjenom uslova, membership funkcija te samom promjenom parametara membership funkcija.

Surface viewer je prikazan na narednoj slici:



Sa slike možemo vidjeti razliku u gornjem desnom, odnosno donjem desnom uglu, a ta dva specifična regiona predstavljaju mogućnost nagle (velike) promjene ugla, koja je ranije objašnjena.

**Napomena:** Kao defuzzifikacija je korištena opcija centroid što se može vidjeti na narednoj slici:



## Zadatak br. 2 (ANFIS)

Kreirati ANFIS sistem za aproksimaciju funkcije na intervalu 0-10 prema tabeli. Funkcija:

$$Y = \cos^2(x) + 3 \sin(3x)$$

Potrebno je:

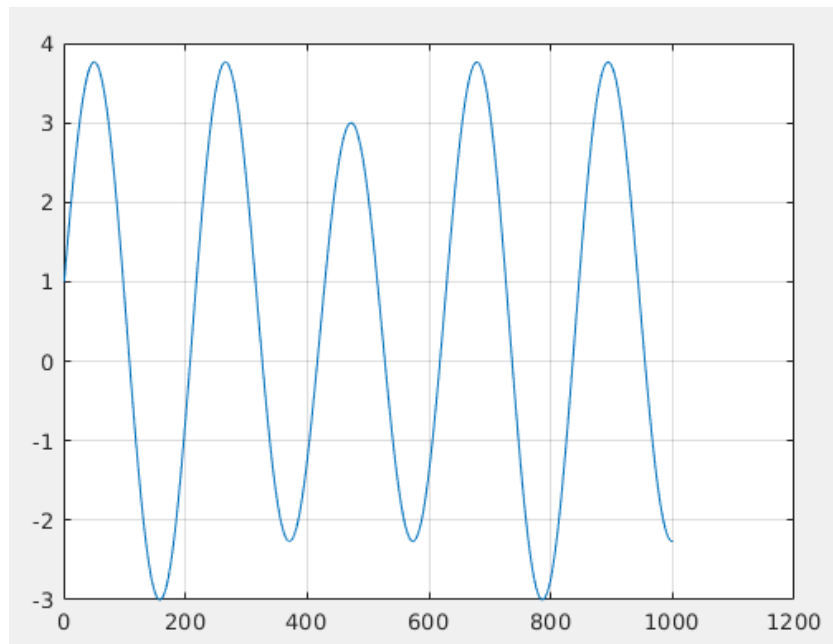
- Isprobati različit tip i broj membership funkcija (6 kombinacija), odrediti optimalnu kombinaciju. Obrazložiti odabir.
- Tabelarno prikazati grešku za svaku isprobanu kombinaciju (za trening i checking podatke).
- Prikazati grafički funkciju (test i trening podatke) za najbolje rješenje, kao i mf prije i nakon treniranja.
- Za odabrano rješenje napisati matlab skriptu koji se pokreće i iscrtava dijagrame, tako da je rezultate moguće prikazati bez korištenja ANFIS GUI-a.

### Rješenje:

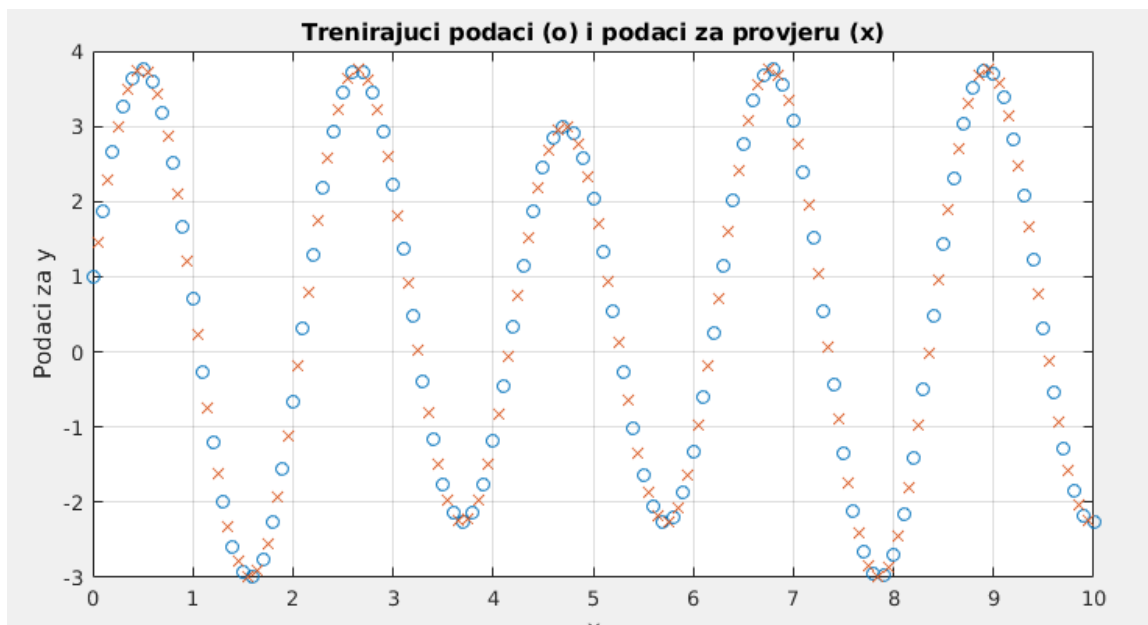
U ovom zadatku potrebno je kreirati ANFIS sistem za aproksimaciju funkcije:

$$Y = \cos^2(x) + 3 \sin(3x)$$

i to na intervalu od 0 do 10. Prvo je potrebno provjeriti kako zadana funkcija izgleda, odnosno kakav je njen oblik. Plot ove funkcije je prikazan na narednoj slici.



Kada smo dobili izgled funkcije, naredni korak nam je formiranje odgovarajućeg seta podataka, te raspodjela podataka na trening podatke i podatke za provjeru. Kada smo navedene podatke podijelili, dobili smo sljedeći dijagram (grafički prikaz):



Vidimo da su podaci prikazani na intervalu od 0 do 10. Naredni korak jeste treniranje koje ćemo vršiti preko GUI-a, Potrebno je isprobati nekoliko slučajeva, te specifične stvari unijeti u tabelu. Potrebno je mijenjati broj MF-ova, njihov tip, te broj epoha.

Slučajevi koje smo testirali su prikazani u narednoj tabeli:



Br.	Tip MF	Broj MF	Broj epoha	Greška trndata	Greška chkdata
1.	trimf (const.)	3	100	2.196643	2.19844
2.	trimf (const.)	10	100	0.798582	0.759033
3.	trapmf(const.)	15	100	0.296796	0.305302
4.	gaussmf(const.)	10	100	0.337859	0.294405
5.	pimf(const.)	10	100	0.904881	0.89225
6.	psigmf(const.)	10	100	0.528611	0.50224

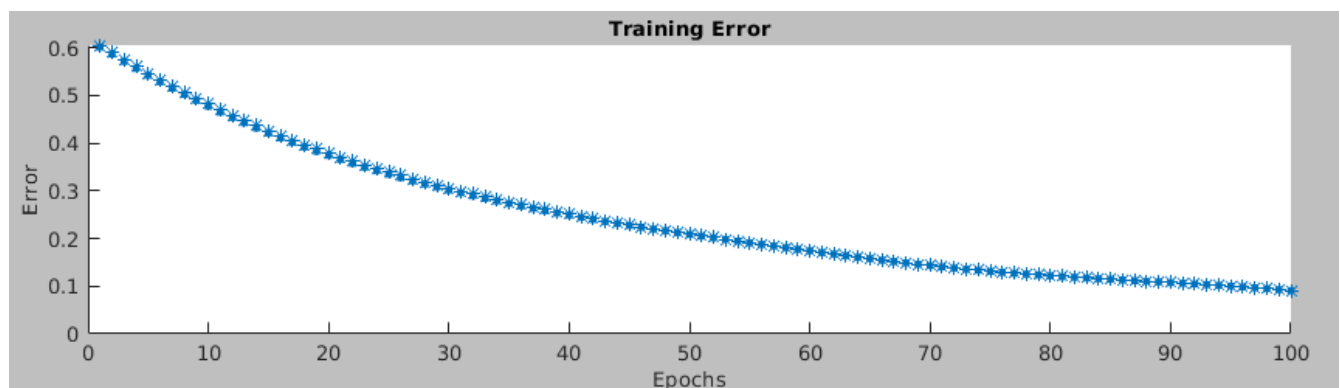
U prethodnoj tabeli su prikazani samo neki od testiranih slučajeva (neke od kombinacija na osnovu kojih smo donosili zaključke). Bitno je napomenuti da smo u ovom slučaju toleranciju na grešku stavili 0.01 (**Error Tolerance = 0.01**). Iz testiranja mnogobrojnih kombinacija došli smo do zaključka da nam veliku ulogu igraju **broj iteracija** kao i **broj epoha**. Izuzetno malen broj epoha (npr 3.) nam ne može dati dovoljno dobru aproksimaciju. Broj epoha koje smo koristili je 100, te nam se ovaj broj pokazao kao izuzetno dobar. Dalje, znatno malan broj MF-ova također ne daje dobru aproksimaciju, ovdje se pokazalo da nam broj MF-ova iznad 10 daje relativno dobre rezultate. Korisno je iskombinovati broj epoha i membership funkcija kako bismo dobili što bolji rezultat, odnosno što manju grešku.

Kao najbolje rješenje smo uzeli sljedeći slučaj: (**razlog je izuzetno mala greška**)

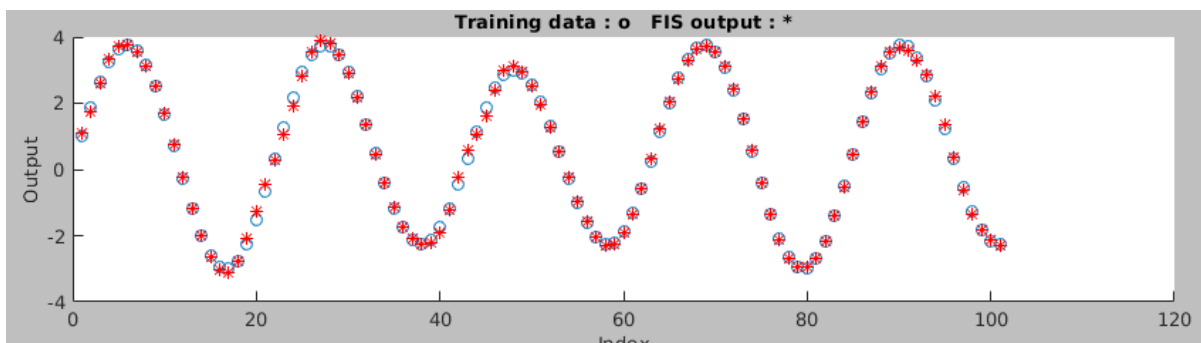
Br.	Tip MF	Broj MF	Broj epoha	Greška trndata	Greška chkdata
1.	gaussmf (const.)	15	100	0.088805	0.0885243

Odgovarajući dijagrami koji su potrebni za ovaj zadatak su prikazani u nastavku:

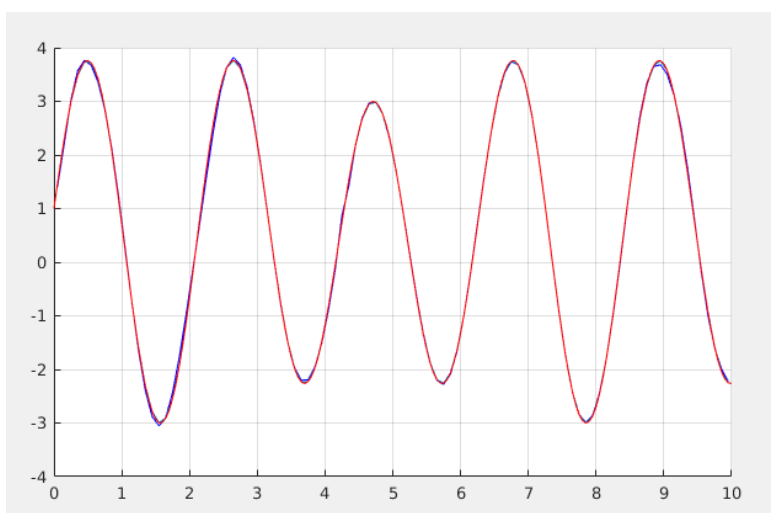
1. Na sljedećem grafiku prikazana je greška između training i checking podataka za najbolji slučaj:



2. Na sljedećoj slici prikazana je predikcija funkcije (najbolji slučaj) odrađena preko Fuzzy GUI-a:

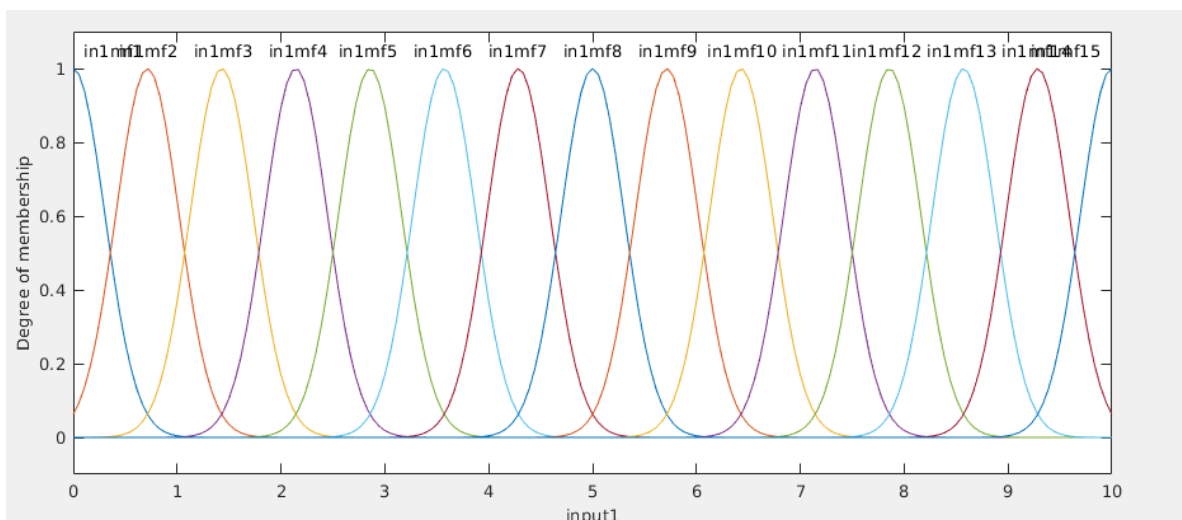


3. Na sljedećoj slici prikazana je predikcija funkcije (najbolji slučaj) odrađena preko MATLAB koda prikazanog na kraju dokumenta. Crveni grafik predstavlja izlaz iz ANFISA, a plavi grafik predstavlja ulazne podatke.

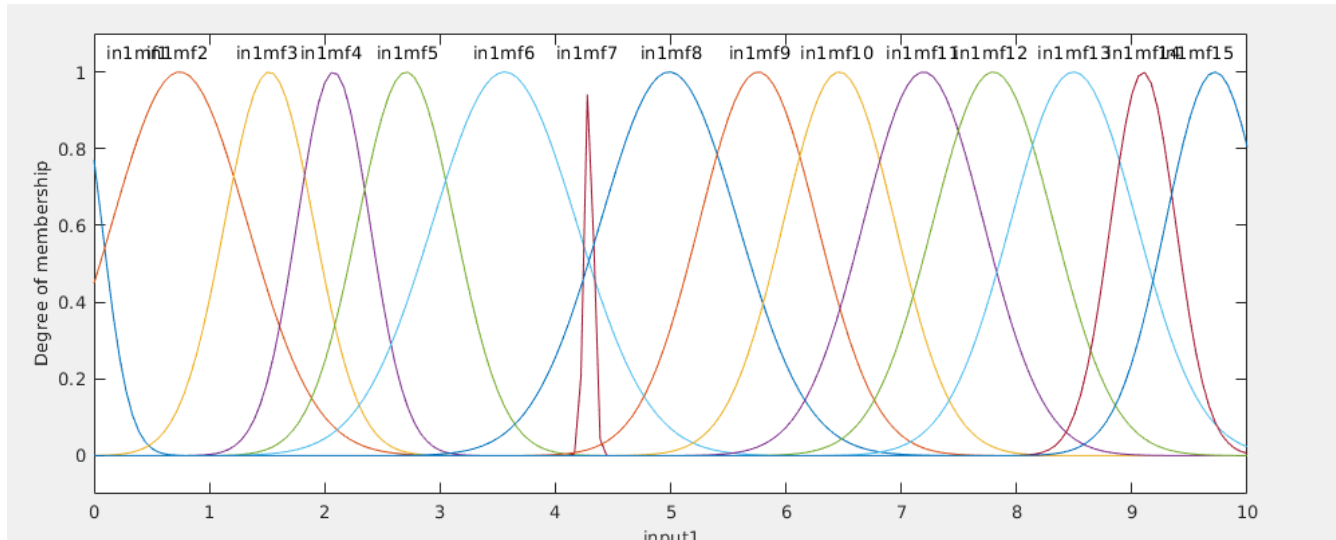


Sa ove slike možemo vidjeti da je aproksimacija skoro ista kao originalna funkcija, te ima izuzetno malo odstupanje u odnosu na stvarnu funkciju.

4. MF funkcije prije treniranja:



## 5. MF funkcije poslije treniranja:



Matlab kod je predstavljen u nastavku:

```
clear all
close all
clc
% Plot za izgled signala:
%x=0:0.01:10;
%x=x';
%y=(cos(x)).^2 + 3*sin(3*x);
%plot(y)
%grid
%Kod sa vjezbi
x=0:0.05:10;
x=x';
y=(cos(x)).^2 + 3*sin(3*x);
numPts=size(x,1);
data =[x y];
trndata=data(1:2:numPts,:);
chkdata=data(2:2:numPts,:);
plot(trndata(:,1),trndata(:,2),'o',chkdata(:,1),chkdata(:,2),'x')
grid
title('Trenirajuci podaci (o) i podaci za provjeru (x)')
xlabel('x');
ylabel('Podaci za y');
hold on
% Dodatni kod:
% gaussmf, 15 mfs, 100 epoch
options =
genfisOptions('GridPartition','OutputMembershipFunctionType','constant','InputMembershipFunctionType','gaussmf','
NumMembershipFunctions',15);
```

```
fismat = genfis(x,y,options);
%plotmf(fismat,'input',1)
options = anfisOptions;
options.InitialFIS = fismat;
options.EpochNumber = 100;
options.ValidationData = chkdata;
[fismat1, trnerr, ss, fismat2, chkerr] = anfis(trndata, options);
figure(2)
plotmf(fismat1,'input',1)

figure(3)
out = evalfis(chkdata(:,1),fismat1);
hold;
plot(chkdata(:,1),out,'b');
hold on;
plot(chkdata(:,1),chkdata(:,2));
plot(x,y,'r');
grid
```