

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU ELEKTROTEHNIČKI
FAKULTET

Diplomski studij računarstva

Laboratorijska vježba 7

Realiziranje regulatora temperature pomoću neizrazitih sustava

Ivan Budoš, DRB

Osijek, 2023.

SADRŽAJ

UVOD	3
NEIZRAZITA LOGIKA	4
REGULATOR TEMPERATURE	7
Zadatak.....	7
Programski kod	7
REZULTATI.....	9
Trokutasta funkcija	9
Trapezasta funkcija	9
Gaussova funkcija.....	10
ZAKLJUČAK	11

UVOD

U sedmoj laboratorijskoj vježbi potrebno je izraditi neizrazite regulatore temperature toplinskog procesa. Proces sadrži senzor temperature i grijač koji je moguće kontinuirano regulirati. Koristeći Python programski jezik i scikit-fuzzy biblioteku potrebno je realizirati neizraziti regulator temperature sa zadanim svojstvima.

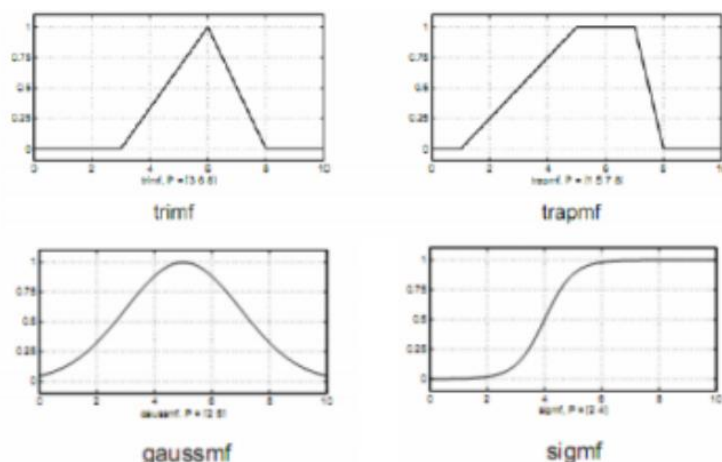
NEIZRAZITA LOGIKA

Neizrazita logika (engl. Fuzzy logic) omogućuje donošenje odluka na temelju nepreciznih informacija prisutnih u stvarnom svijetu. Neizraziti sustav zaključivanja moguće je izgraditi na temelju podataka dobivenih od eksperta. Razvoj im započinje Lotfi Zadeh 1965., a usavršava Ebrahim H. Mamdani 1975. i Michio Sugeno 1985. Za razliku od neuronske mreže, proces zaključivanja je transparentan. Također je omogućeno kombiniranje s klasičnim sustavima upravljanja. Zasnovana je na prirodnom jeziku, što znači da je neizraziti sustav odlučivanja temeljen na pravilima razumljivim i jasnim čovjeku (npr. “Ako je temperatura visoka, smanji grijanje”). I pored svih prednosti neizrazite logike, nije ju preporučljivo koristiti tamo gdje postoje dobra i provjerena “klasična” rješenja. Bitan dio neizrazite logike jeste neizraziti skup, koji se razlikuje od klasičnih skupova. Tako kod klasičnih skupova određeni element pripada nekom skupu, ili mu ne pripada (crno - bijelo), dok je kod neizrazitih skupova definirana pripadnost elementa skupu. Drugim riječima u neizrazitoj logici istinitost neke tvrdnje postaje stvar mjere! Ta mjera odnosno pripadnost definirana je kao broj, najčešće u intervalu [0, 1].

Neizrazit skup se može prikazati izrazom:

$$A = \{(\mu_A(x), x) | x \in X\},$$

gdje je μ_A funkcija pripadnosti, a X neki skup. Funkcija pripadnosti (engl. membership function) definira pripadnost skupu. Postoji mnogo oblika funkcije pripadnosti (npr. trokut, trapez, gauss, zvono, itd.). Najčešće je jedini uvjet za funkciju pripadnosti da poprima vrijednosti u intervalu [0, 1].



Slika 1. Najčešće funkcije pripadnosti

Kao i kod klasične logike i u neizrazitoj logici postoje operatori I, ILI, NE. Međutim, u odnosu na klasičnu logiku ovi operatori nisu jednoznačno određeni, već su prošireni na cijeli interval $[0, 1]$, time da na granicama prelaze u operatore klasične logike odnosno u tzv. Boolovu logiku. U neizrazitoj logici se za pojedine operatore koriste razne funkcije. Tako se za I operator koriste funkcije:

$$- y = \min(a, b)$$

$$- y = \text{prod}(a, b) = a * b$$

Za operator ILI se koriste funkcije:

$$- y = \max(a, b) - y = \text{probor}(a, b) = a + b - a * b$$

Za operator NE se koristi funkcija:

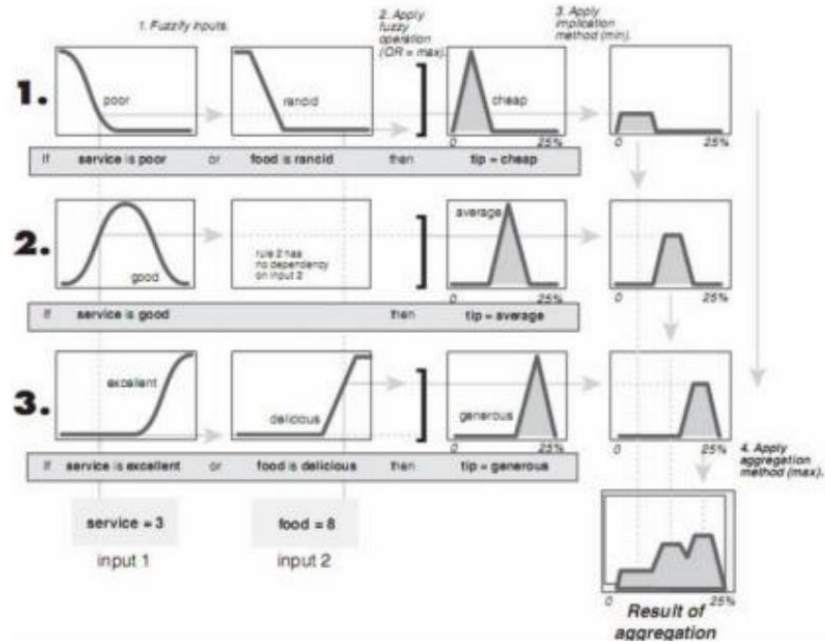
$$- y = 1 - a$$

Baza neizrazitih pravila (engl. fuzzy rule base) predstavlja središnji dio neizrazitog sustava. Neizrazito pravilo ima standardni if...then oblik:

ako je temperatura visoka **onda** smanji grijanje,

gdje je „**ako** je temperatura visoka“ polazni dio pravila, a „**onda** smanji grijanje“ posljedični dio pravila. Pravila povezuju područja vrijednosti ulaznih i izlaznih varijabli. Neizrazito zaključivanje se sastoji od 5 koraka kao što se vidi na slici 2, a ti koraci su slijedeći:

1. Fuzifikacija ulaza
2. Primjena neizrazitih operatora (pretpostavka pravila)
3. Implikacija (zaključak pravila)
4. Agregacija izlaza
5. Defuzifikacija



Slika 3. Koraci rada neizrazitog sustava

Postoje dva tipa neizrazitih sustava: Takagi-Sugeno-Kang (Sugeno) i Mamdani. Sve prethodno rečeno odnosilo se na Mamdani neizraziti sustav.

Uporaba neizrazitih sustava danas ima implementacije u svim granama tehničkih znanosti, počevši sa jednostavnim upravljačkim sustavima pa do ekspertnih sustava i sofisticiranih umjetnih inteligencija

REGULATOR TEMPERATURE

Regulator temperature se koristi kako bi regulirao temperaturu koristeći povratne informacije koje dobije. Cilj je da održi zadanu temperaturu pri utjecaju smetnji.

Zadatak

Potrebno je izraditi neizrazite regulatore temperature toplinskog procesa. Proces sadrži senzor temperature i grijač koji je moguće kontinuirano regulirati. Koristeći Python programski jezik i scikitfuzzy biblioteku realizirati neizraziti regulator temperature sljedećih svojstava:

Tip regulatora: Mamdani

Ulazna varijabla: Greska_temp - pogreška temperature (razlika između zadane i trenutne temperature), u intervalu [0,30] C, neizraziti skupovi = Niska, Srednja, Visoka

Izlazna varijabla: Grijac - snaga grijača u intervalu [0,10], neizraziti skupovi = Slabo, Srednje, Jako

Pravila:

If Greska_temp is Niska then Grijac is Slabo

If Greska_temp is Srednja then Grijac is Srednje

If Greska_temp is Visoka then Grijac is Jako

Potrebno je realizirati tri različita regulatora temperature korištenjem trokutastih, trapezastih i gausovih funkcija pripadnosti. Parametre neizrazitih skupova na ulazu i izlazu podesiti samostalno

Programski kod

Prva stvar koju je bilo potrebno napraviti je uključiti potrebne biblioteke među kojima je i skfuzzy. Nakon toga su definirane ulazne i izlazne varijable regulatora sa svojim radnim prostorima. Zadano je da pogreška mora biti u rasponu od 0 do 30 °C i za metodu defuzifikacije izlazne varijable odabran je 'centroid'.

```
#Ulazna varijabla
Greska_temp = ctrl.Antecedent(np.linspace(0, 30, 100), 'Greska_temp')
#Izlazna varijabla
Grijac = ctrl.Consequent(np.linspace(0, 10, 100), 'Grijac', defuzzify_method='centroid')
```

Slika 3. Definiranje ulazne i izlazne varijable

Koristeći primjere iz pdf-a realizirana su tri različita regulatora temperature od kojih svaki ima drugačiju funkciju pripadnosti i sadrži parametre koji su navedeni. Dva pravila su uvijek zakomentirana kako bi se moglo prikazati ponašanje jednog.

```
#trokutasta funkcija
Greska_temp['Niska'] = fuzz.trimf(Greska_temp.universe, [0, 0, 15]) #Trokut, lomi se u 3 točke
Greska_temp['Srednja'] = fuzz.trimf(Greska_temp.universe, [0, 15, 30]) #Trokut, lomi se u 3 točke
Greska_temp['Visoka'] = fuzz.trimf(Greska_temp.universe, [15, 30, 30]) #Trokut, lomi se u 3 točke

Grijac['Slabo'] = fuzz.trimf(Grijac.universe, [0, 0, 5]) #Gaussova funkcija definirana s mi i sigma
Grijac['Srednje'] = fuzz.trimf(Grijac.universe, [0, 5, 10]) #Gaussova funkcija definirana s mi i sigma
Grijac['Jako'] = fuzz.trimf(Grijac.universe, [5, 10, 10]) #Gaussova funkcija definirana s mi i sigma

"""
#trapezasta funkcija
Greska_temp['Niska'] = fuzz.trapmf(Greska_temp.universe, [0, 0, 5, 15]) #Trapez, lomi se u 4 točke
Greska_temp['Srednja'] = fuzz.trapmf(Greska_temp.universe, [0, 10, 20, 30]) #Trapez, lomi se u 4 točke
Greska_temp['Visoka'] = fuzz.trapmf(Greska_temp.universe, [15, 25, 30, 30]) #Trapez, lomi se u 4 točke

Grijac['Slabo'] = fuzz.trapmf(Grijac.universe, [0, 0, 2.5, 5]) #Gaussova funkcija definirana s mi i sigma
Grijac['Srednje'] = fuzz.trapmf(Grijac.universe, [0, 2.5, 7.5, 10]) #Gaussova funkcija definirana s mi i sigma
Grijac['Jako'] = fuzz.trapmf(Grijac.universe, [5, 7.5, 10, 10]) #Gaussova funkcija definirana s mi i sigma

"""
#gaussova funkcija
Greska_temp['Niska'] = fuzz.gaussmf(Greska_temp.universe, 0, 7.5) #Trokut, lomi se u 3 točke
Greska_temp['Srednja'] = fuzz.gaussmf(Greska_temp.universe, 15, 7.5) #Trokut, lomi se u 3 točke
Greska_temp['Visoka'] = fuzz.gaussmf(Greska_temp.universe, 30, 7.5) #Trokut, lomi se u 3 točke

Grijac['Slabo'] = fuzz.gaussmf(Grijac.universe, 0, 2.5) #Gaussova funkcija definirana s mi i sigma
Grijac['Srednje'] = fuzz.gaussmf(Grijac.universe, 5, 2.5) #Gaussova funkcija definirana s mi i sigma
Grijac['Jako'] = fuzz.gaussmf(Grijac.universe, 10, 2.5) #Gaussova funkcija definirana s mi i sigma

"""
```

Slika 4. Kreiranje regulatora sa drugačijim funkcijama pripadnosti

Zatim su definirana pravila koja kažu:

-ako je temperatura niska onda je grijač slab

```
pravilo1=ctrl.Rule(Greska_temp['Niska'],Grijac['Slabo'])
```

-ako je temperatura srednja onda je grijač srednji

```
pravilo2=ctrl.Rule(Greska_temp['Srednja'],Grijac['Srednje'])
```

-ako je temperatura visoka onda je grijač jak

```
pravilo3=ctrl.Rule(Greska_temp['Visoka'],Grijac['Jako'])
```

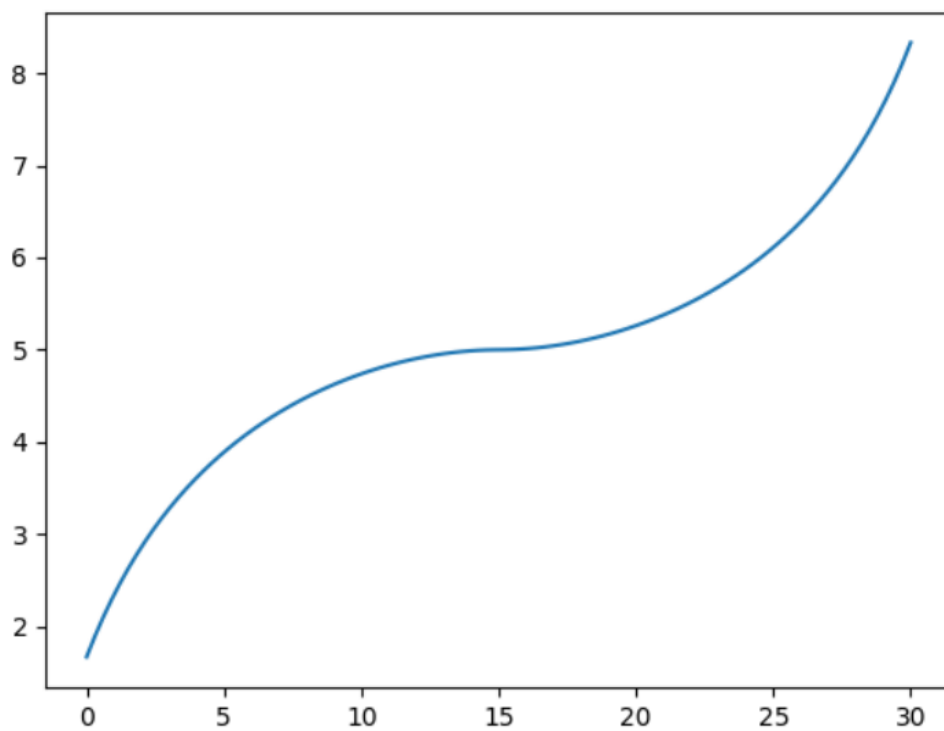
Greska_temp „Niska“, „Srednja“ i „Visoka“ se odnose na razliku temperature dok se Grijac „Slabo“, „Srednje“, „Jako“ odnose na snagu grijača koju je potrebno uključiti.

```
regulator=ctrl.ControlSystem([pravilo1,pravilo2,pravilo3])
regulator_simulacija=ctrl.ControlSystemSimulation(regulator)
```

Slika 5. Simulacija sustava

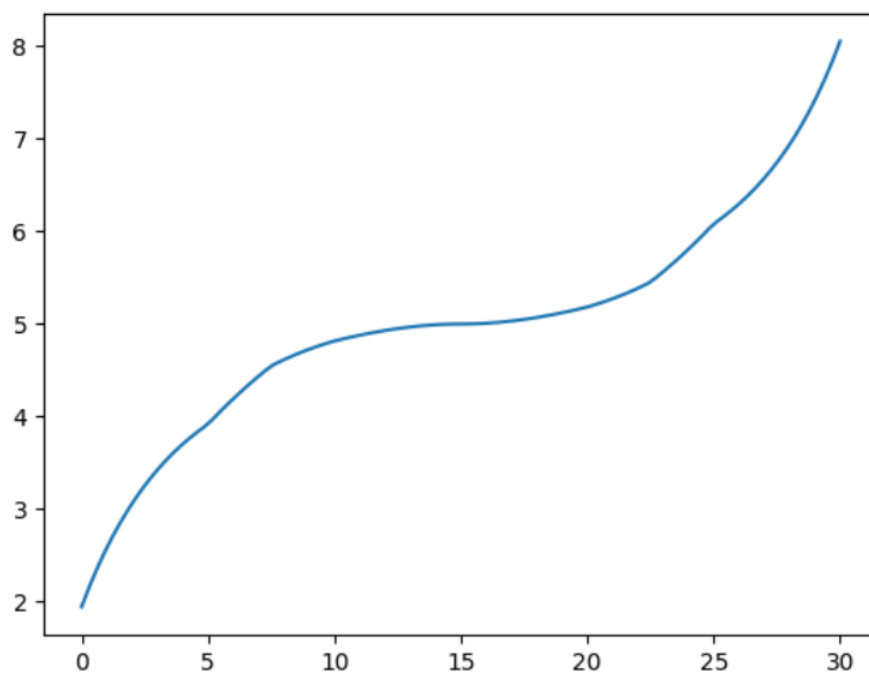
REZULTATI

Trokutasta funkcija



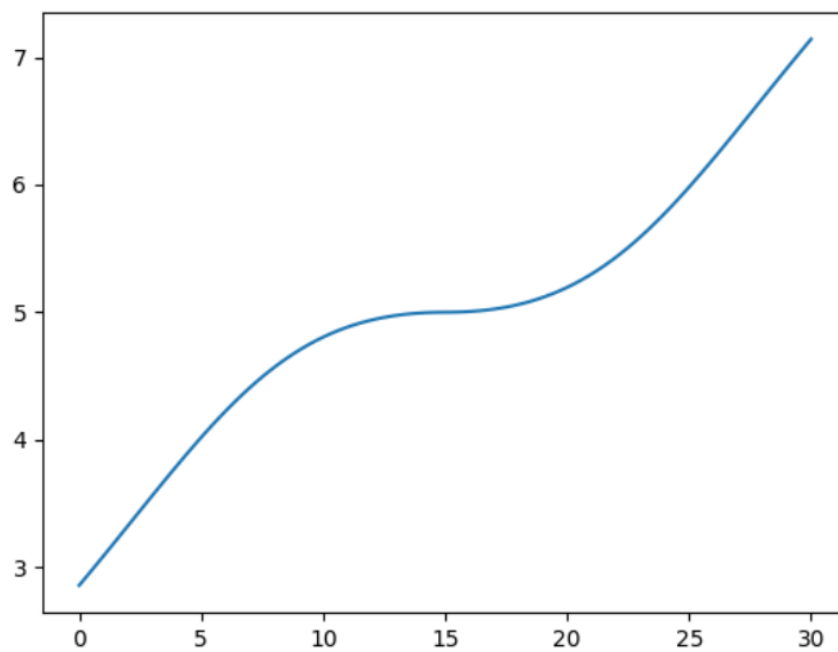
Slika 6. Trokutasta funkcija

Trapezasta funkcija



Slika 7. Trapezasta funkcija

Gaussova funkcija



Slika 8. Gaussova funkcija

ZAKLJUČAK

Neizrazita logika omogućuje donošenje odluka na temelju nepreciznih informacija prisutnih u stvarnom svijetu. Kao i kod klasične logike i u neizrazitoj logici postoje operatori I, ILI, NE no u odnosu na klasičnu logiku ovi operatori nisu jednoznačno određeni, već su prošireni na cijeli interval $[0, 1]$, time da na granicama prelaze u operatore klasične logike odnosno u tzv. Boolovu logiku. Neizrazito zaključivanje se sastoji od 5 koraka, a to su fuzifikacija ulaza, primjena neizrazitih operatora (pretpostavka pravila), implikacija (zaključak pravila), agregacija izlaza, defuzifikacija. Postoje dva tipa neizrazitih sustava: Takagi-Sugeno-Kang (Sugeno) i Mamdani. Sve prethodno rečeno odnosilo se na Mamdani i u ovoj vježbi je također korišten Mamdani.

Uspješnost odnosno performanse regulatora sa zadanom funkcijom se mogu odrediti tako da se pogleda njegov odziv. Što manje oscilacija ima i što je sustav bliži linearnoj funkciji to je bolji. U ovom slučaju najlošiji rezultat za naš sustav je dobiven koristeći trapezastu funkciju jer ima previše oscilacija dok je najbolji odziv dobiven koristeći Gaussovu funkciju. Trokutasta funkcija također dala dobar rezultat.