

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**

**Intelektikos pagrindai 2020**  
***Laboratorinio darbo nr. 3 ataskaita***

Atliko:

IFF-7/2 gr. studentas

Justas Milišiūnas

2020-02-29

Dėstytojai:

lekt. Audrius Nečiūnas

doc. Agnė Paulauskaitė-Tarasevičienė

## TURINYS

1. Užduotis.....	3
2. Sistemos įvesčių kiekis ir fuzzy aibės.....	3
3. Sistemos išvesčių kiekis ir fuzzy aibės.....	3
4. Taisyklės.....	3
5. Metodai naudoti implikacijai, agregacijai ir defuzifikacijai.....	3
6. Testavimo scenarijai.....	3
6.1. Scenarijus nr. 1.....	3
6.2. Scenarijus nr. 2.....	4
6.3. Scenarijus nr. 3.....	5
7. Rezultatų verifikavimas.....	6
7.1. Scenarijaus nr. 1 verifikavimas.....	6
7.2. Scenarijaus nr. 2 verifikavimas.....	8
7.3. Scenarijaus nr. 3 verifikavimas.....	9
8. Programos kodas.....	10

## 1. Užduotis

Benzininio automobilio ekonomiškumo nustatymas [3-25 l/100km]. Bus atsižvelgiama į:

- Galia [50-500 kw]
- Svorį [700-2700 kg]
- Pagaminimo metus [1970-2020 metai]

Kintamųjų matmenys sugalvoti

## 2. Sistemos įvesčių kiekis ir fuzzy aibės

Sistemoje 3 įvestys:

- Galia [50-500 kw], fuzzy aibės: maža, vidutinė, didelė
- Svoris [700-2700 kg], fuzzy aibės: lengvas, vidutinis, sunkus
- Pagaminimo metai [1970-2020 metai], fuzzy aibės: senas, vidutinis, naujas

## 3. Sistemos išvesčių kiekis ir fuzzy aibės

Sistemoje 1 išvestis:

- Ekonomiškumas [3-25 l/100km], fuzzy aibės: didelis, vidutinis, mažas

## 4. Taisyklės

Sistemos taisyklės:

1. Jeigu svoris yra lengvas ir pagaminimo metai yra naujas, tai ekonomiškumas yra didelis
2. Jeigu galia yra maža, tai ekonomiškumas yra didelis
3. Jeigu galia yra vidutinė ir svoris ne sunkus, tai ekonomiškumas yra vidutinis
4. Jeigu gali yra vidutinė ir pagaminimo metai yra vidutinis, tai ekonomiškumas yra vidutinis
5. Jeigu svoris yra sunkus ir pagaminimo metai yra ne nauja, tai ekonomiškumas yra mažas
6. Jeigu galia yra didelė, tai ekonomiškumas yra mažas

## 5. Metodai naudoti implikacijai, agregacijai ir defuzifikacijai

Metodai:

- Ir panaudota min
- Arba panaudota max
- Implikacijai panaudota min
- Agregacijai panaudota max
- Defuzifikacijai panaudota centroid ir MOM

## 6. Testavimo scenarijai

### 6.1. Scenarijus nr. 1

Įėjimo kintamieji:

- Galia = 275 kw
- Svoris = 1700 kg
- Pagaminimo metai = 1998 m.

Išėjimo kintamieji:

- Ekonomija = 12.07 l/100km (naudojant centroid metodą)
- Ekonomija = 12 l/100km (naudojant MOM metodą)

### Defuzifikacijos grafikai:

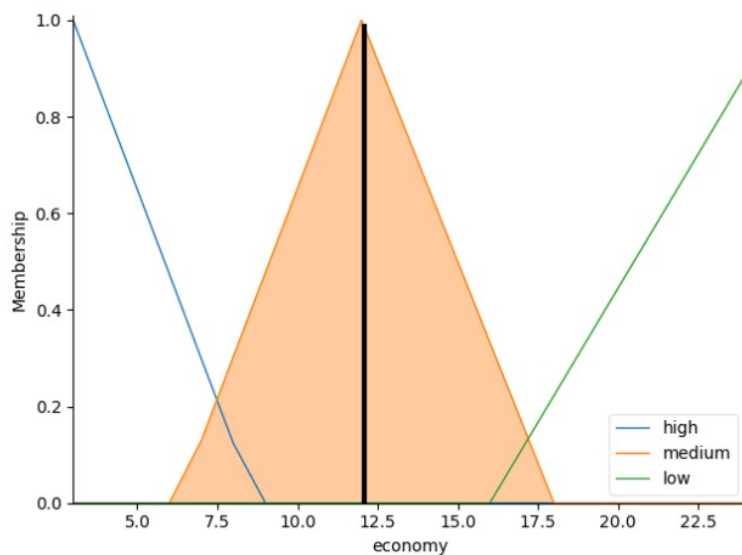


Diagrama 1: Defuzifikacija naudojant centroid metodą

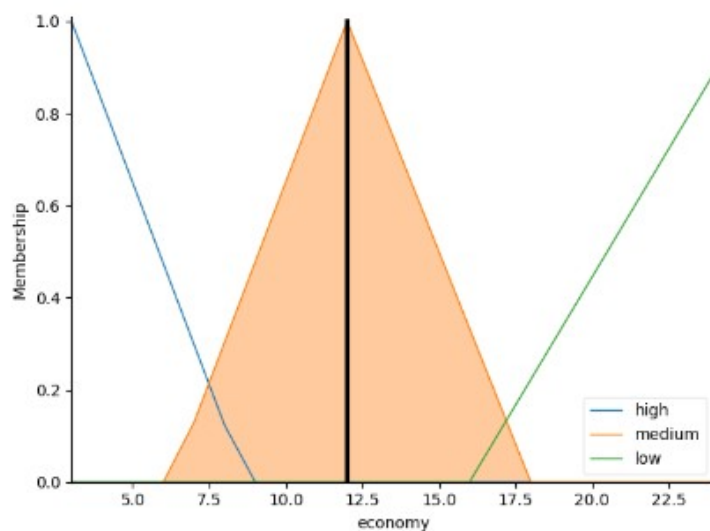


Diagrama 2: Defuzifikacija naudojant MOM metodą

## 6.2. Scenarijus nr. 2

### Iėjimo kintamieji:

- Galia = 100 kw
- Svoris = 1200 kg
- Pagaminimo metai = 2018 m.

### Išėjimo kintamieji:

- Ekonomija = 5.12 l/100km (naudojant centroid metodą)
- Ekonomija = 4.28 l/100km (naudojant MOM metodą)

### Defuzifikacijos grafikai:

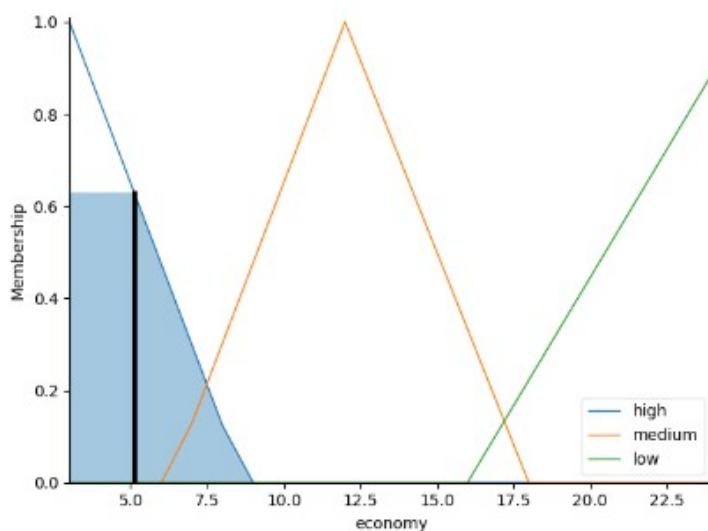


Diagrama 3: Difuzifikacija naudojant centroid metodą

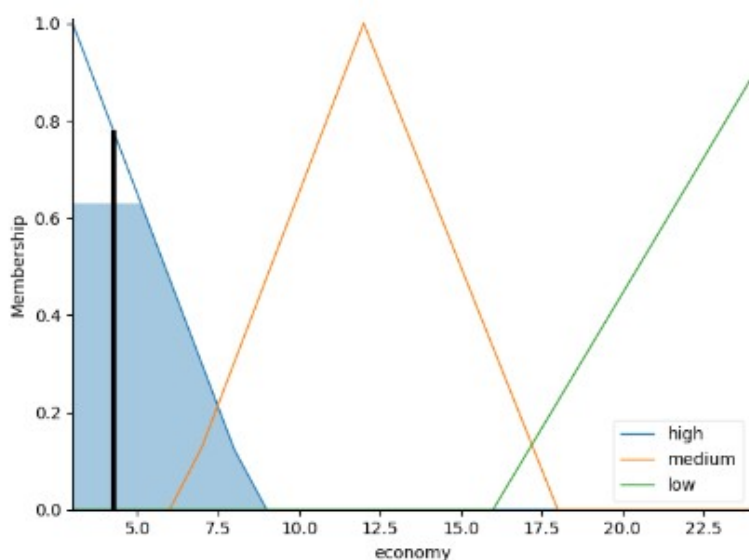


Diagrama 4: Defuzifikacija naudojant MOM metodą

### 6.3. Scenarijus nr. 3

#### Iėjimo kintamieji:

- Galia = 350 kw
- Svoris = 2200 kg
- Pagaminimo metai = 1981 m.

#### Išėjimo kintamieji:

- Ekonomija = 14.62 l/100km (naudojant centroid metodą)
- Ekonomija = 12.02 l/100km (naudojant MOM metodą)

## Defuzifikacijos grafikai:

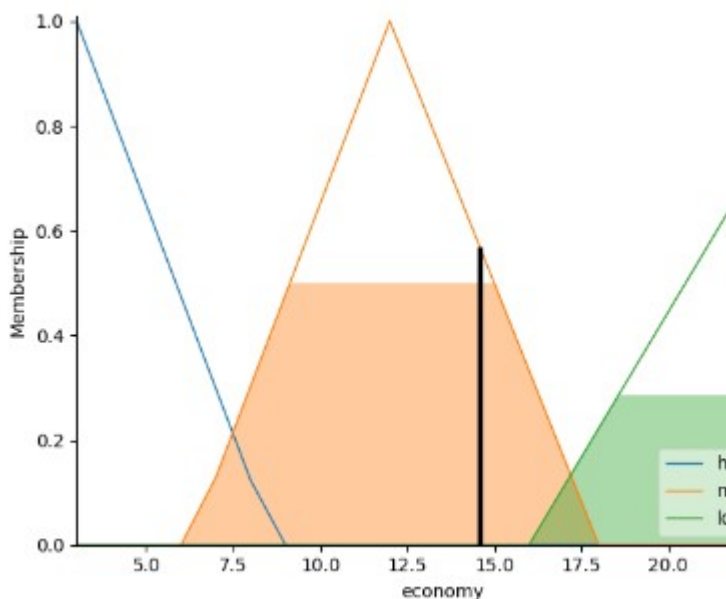


Diagrama 5: Defuzifikacija naudojant centroid metodą

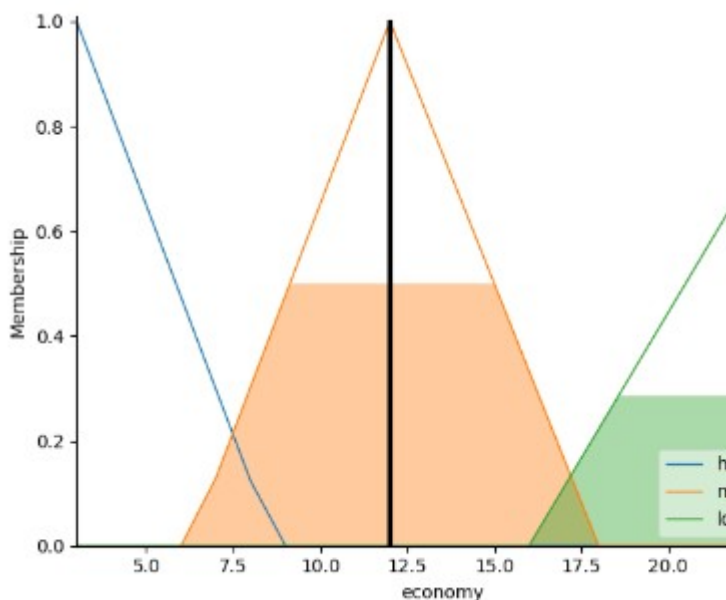


Diagrama 6: Defuzifikacija naudojant MOM metodą

## 7. Rezultatų verifikavimas

### 7.1. Scenarijaus nr. 1 verifikavimas

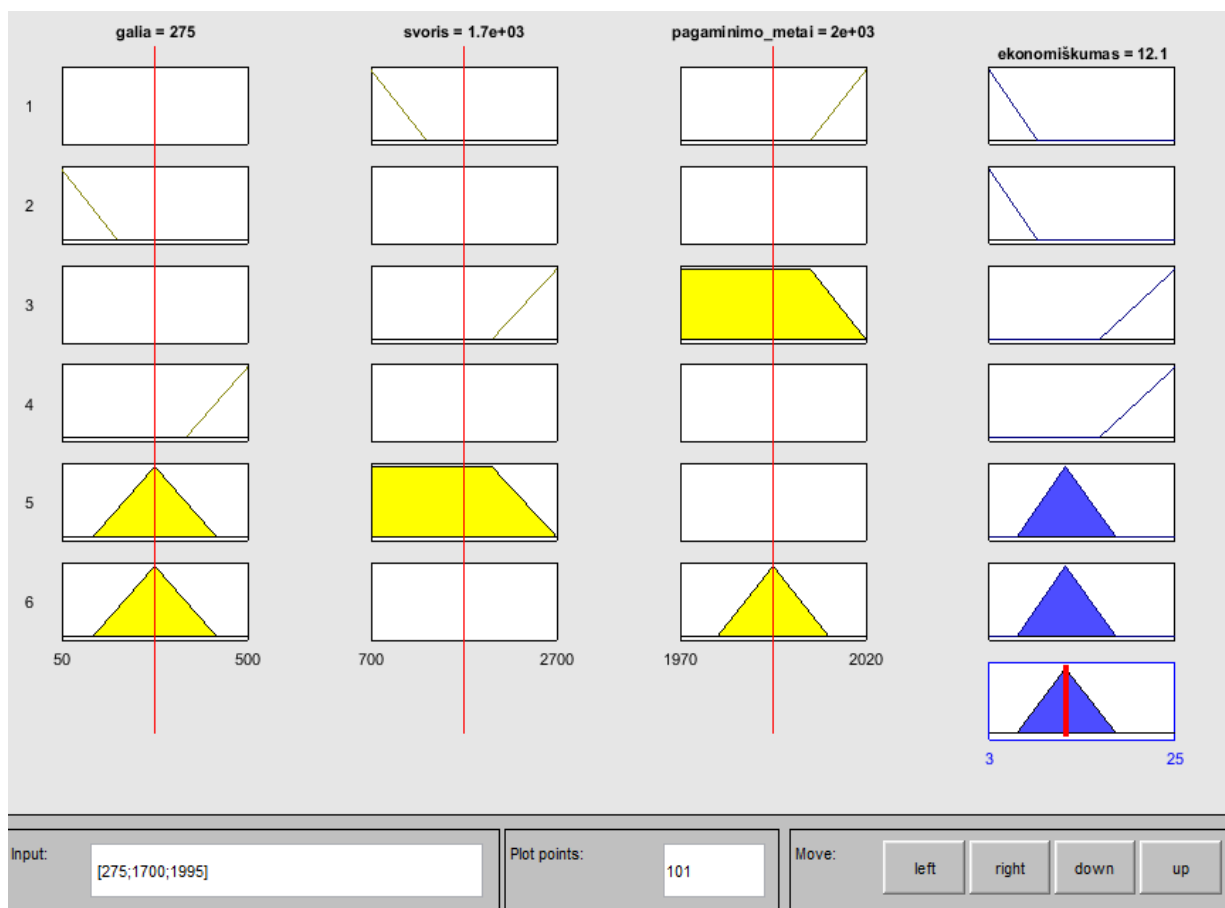
Gautos išėjimo reikšmės su python:

- 12.07 l/100km (centroid)
- 12 l/100km (MOM)

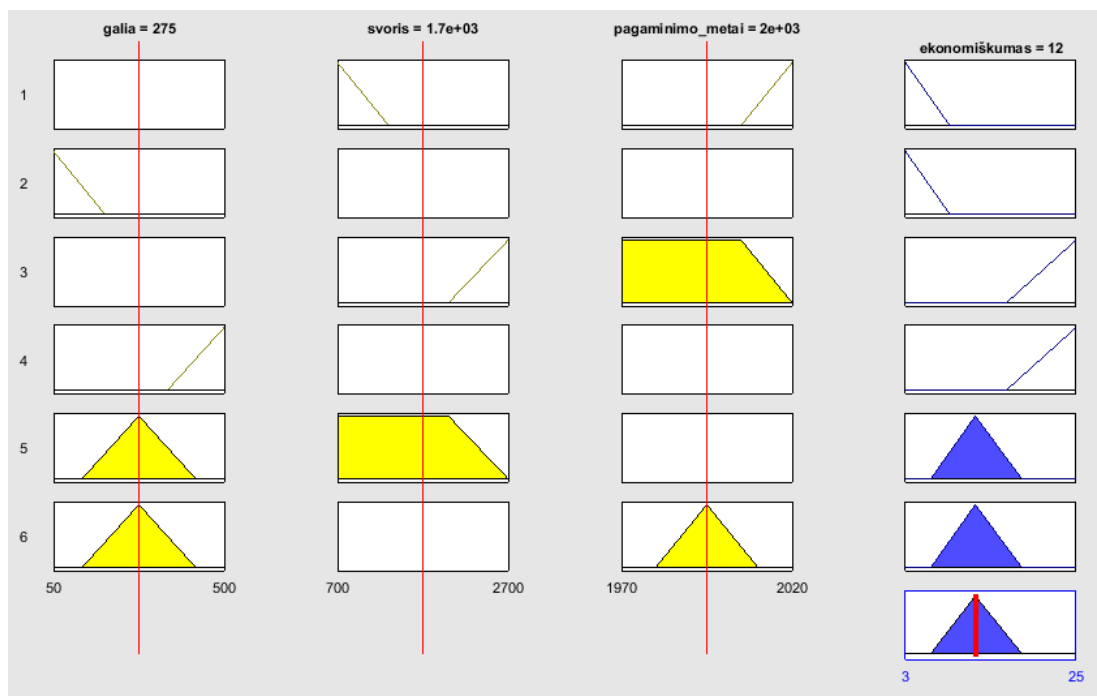
Gautos išėjimo reikšmės naudojant Matlab Fuzzy Logic Toolbox:

- 12.1 l/100km (centroid)
- 12 l/100km (MOM)

## Matlab Fuzzy Logic Toolbox rezultatai:



Pav 1: Matlab rezultatas (centroid)



Pav 2: Matlab rezultatas (MOM)

## 7.2. Scenarijaus nr. 2 verifikavimas

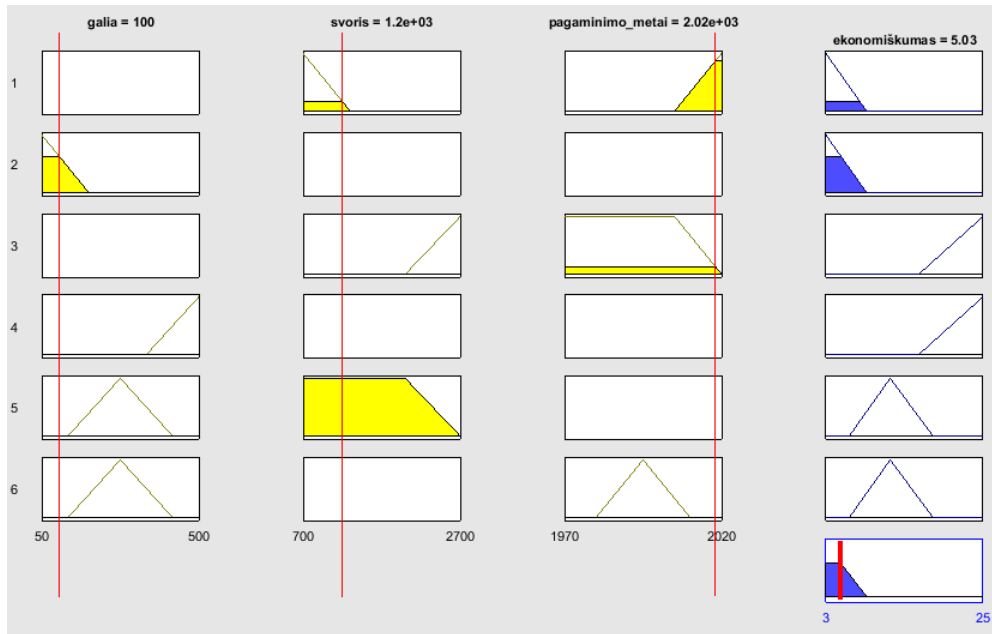
Gautos išėjimo reikšmės su python:

- 5.12 l/100km (centroid)
- 4.28 l/100km (MOM)

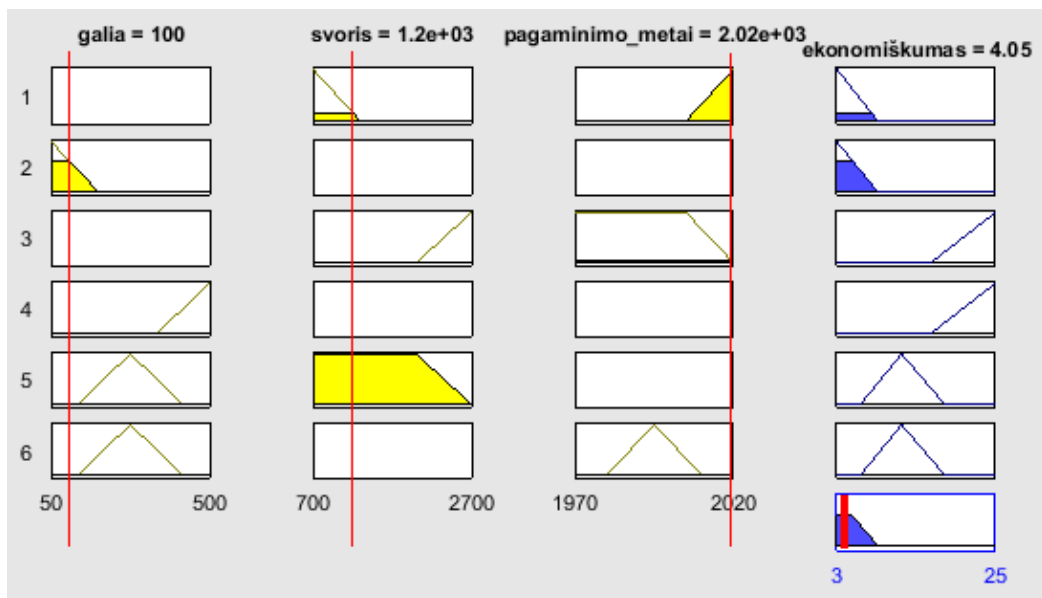
Gautos išėjimo reikšmės naudojant Matlab Fuzzy Logic Toolbox:

- 5.03 l/100km (centroid)
- 4.05 l/100km (MOM)

Matlab Fuzzy Logic Toolbox rezultatai:



Pav 3: Matlab rezultatas (centroid)



Pav 4: Matlab rezultatas (MOM)



### 7.3. Scenarijaus nr. 3 verifikavimas

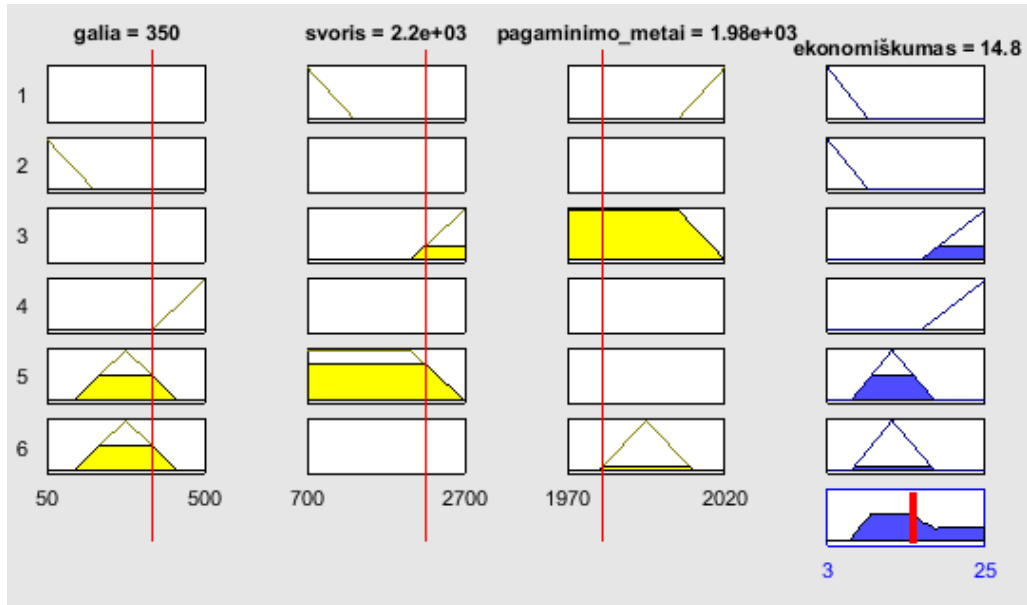
Gautos išėjimo reikšmės su python:

- 14.62 l/100km (centroid)
- 12.02 l/100km (MOM)

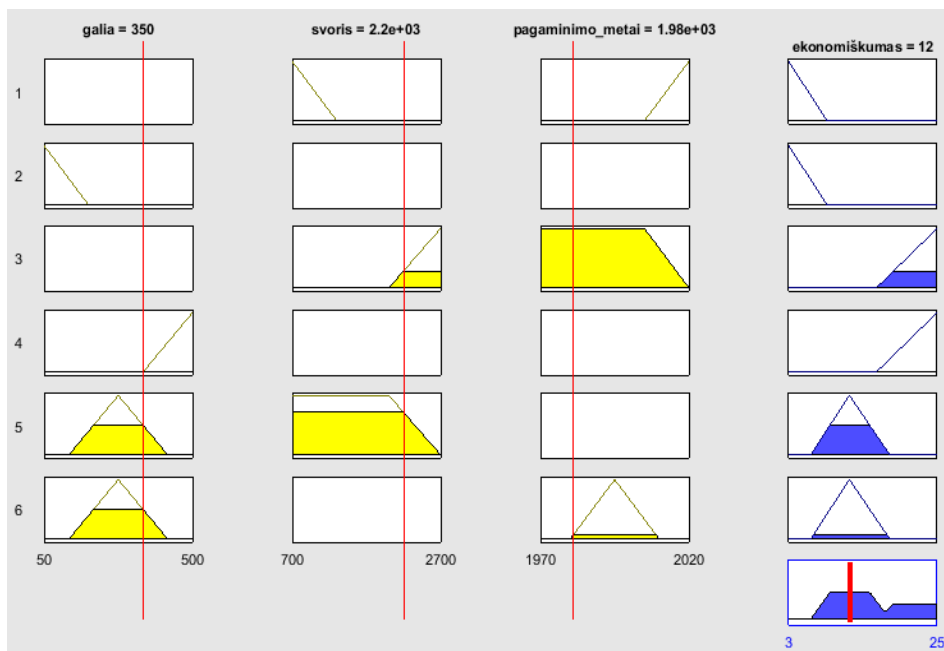
Gautos išėjimo reikšmės naudojant Matlab Fuzzy Logic Toolbox:

- 14.8 l/100km (centroid)
- 12 l/100km (MOM)

Matlab Fuzzy Logic Toolbox rezultatai:



Pav 5: Matlab rezultatas (centroid)



Pav 6: Matlab rezultatas (MOM)

## 8. Programas kodas

```
from skfuzzy import control as ctrl
import skfuzzy as fuzz
import numpy as np

power = ctrl.Antecedent(np.arange(50, 500, 1), 'power')
weight = ctrl.Antecedent(np.arange(700, 2700, 1), 'weight')
year = ctrl.Antecedent(np.arange(1970, 2020, 1), 'year')

economy = ctrl.Consequent(np.arange(3, 25, 1), 'economy')

power['low'] = fuzz.trimf(power.universe, [50, 50, 185])
power['medium'] = fuzz.trimf(power.universe, [125, 275, 425])
power['high'] = fuzz.trimf(power.universe, [350, 500, 500])

weight['light'] = fuzz.trimf(weight.universe, [700, 700, 1300])
weight['medium'] = fuzz.trimf(weight.universe, [1100, 1700, 2200])
weight['heavy'] = fuzz.trimf(weight.universe, [2000, 2700, 3700])

year['old'] = fuzz.trimf(year.universe, [1970, 1970, 1985])
year['normal'] = fuzz.trimf(year.universe, [1980, 1998, 2010])
year['new'] = fuzz.trimf(year.universe, [2005, 2020, 2020])

economy['high'] = fuzz.trimf(economy.universe, [3, 3, 8.704])
economy['medium'] = fuzz.trimf(economy.universe, [6.259, 12, 18])
economy['low'] = fuzz.trimf(economy.universe, [16, 25, 25])

rules = [ctrl.Rule(weight['light'] & year['new'], economy['high']),
         ctrl.Rule(power['low'], economy['high']),
         ctrl.Rule(power['medium'] & ~weight['heavy'], economy['medium']),
         ctrl.Rule(power['medium'] & year['normal'], economy['medium']),
         ctrl.Rule(weight['heavy'] & ~year['new'], economy['low']),
         ctrl.Rule(power['high'], economy['low'])]

economy_ctrl = ctrl.ControlSystem(rules)
economy_prediction = ctrl.ControlSystemSimulation(economy_ctrl)

economy_prediction.input['power'] = 275
economy_prediction.input['weight'] = 1700
economy_prediction.input['year'] = 1995

economy_prediction.compute()
print(economy_prediction.output['economy'])
economy.view(sim=economy_prediction)
```