

Computer Homework 2
Auto Volume Headphone system

จัดทำโดย
นายสุริยา เตชะลือ
รหัสนักศึกษา 600610790

เสนอ
ผศ.ดร. ศันสนีย์ เอื้อพันธุ์วิริยะกุล

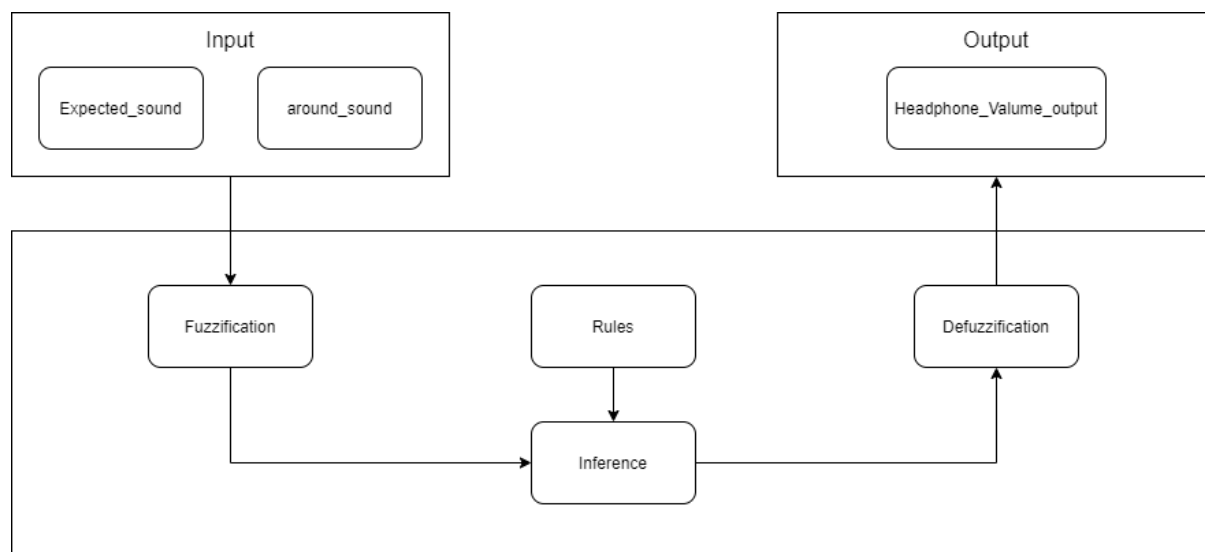
รายงานนี้เป็นส่วนหนึ่งของรายวิชา 261456
(Introduction to Computational Intelligence)

สารบัญ

สารบัญ	1
ลักษณะการทำงานของระบบ	2
การทำงานของระบบ	2
Input	2
Fuzzification	3
Inference	4
Defuzzification	4
ผลการทดลอง และวิเคราะห์	5
ผลการทดลอง	5
วิเคราะห์ผล	9

ลักษณะการทำงานของระบบ

ในรายงานเล่มนี้จัดทำเกี่ยวกับระบบควบคุมระดับเสียงของหูฟังอัตโนมัติ โดยใช้หลักการ mamdani fuzzy ซึ่งมีการรับอินพุต 2 ชนิดด้วยกันคือ ระดับเสียงโดยรอบ และ ระดับเสียงที่ผู้สวมใส่หูฟังต้องการรับฟัง ซึ่งผลลัพธ์ที่ได้คือ ระดับเสียงของหูฟัง โดยมีหลักการทำงานดังรูปต่อไปนี้



หลักการทำงานของระบบควบคุมระดับเสียงของหูฟัง

การทำงานของระบบ

1. Input

- ระดับเสียงโดยรอบ

เป็นระดับเสียงในสภาพแวดล้อมของผู้สวมใส่หูฟัง โดยมีหน่วยวัดระดับความดังตั้งแต่ 0 ถึง 140 Dasibel (db)

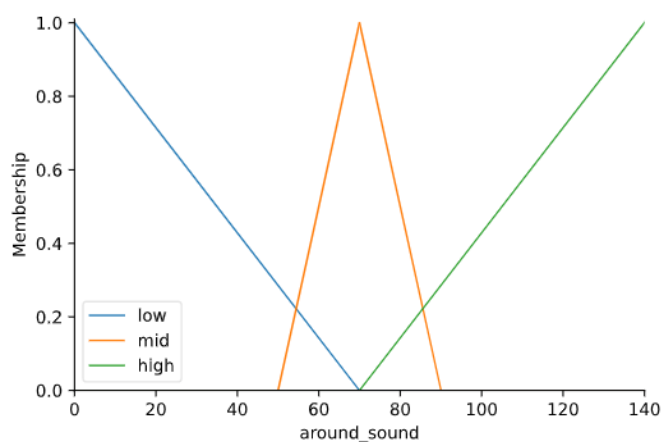
- ระดับเสียงที่ผู้สวมใส่หูฟังต้องการรับฟัง

เป็นอินพุตของผู้สวมใส่หูฟังที่ต้องการรับฟังเสียงในระดับเสียงใด เทียบกับสภาพแวดล้อมโดยรอบ โดยมีค่าตั้งแต่ 0 ถึง 100 เปอร์เซ็น

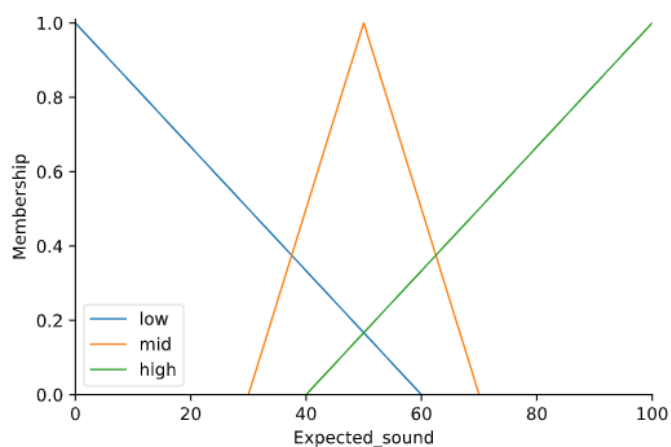
2. Fuzzification

ทำการแปลง Input ที่เข้ามาให้อยู่ในรูปของ fuzzy set เพื่อระบุค่าความเป็นสมาชิกของข้อมูลที่จะนำมาวิเคราะห์ต่อไป โดยกำหนดค่าดังนี้

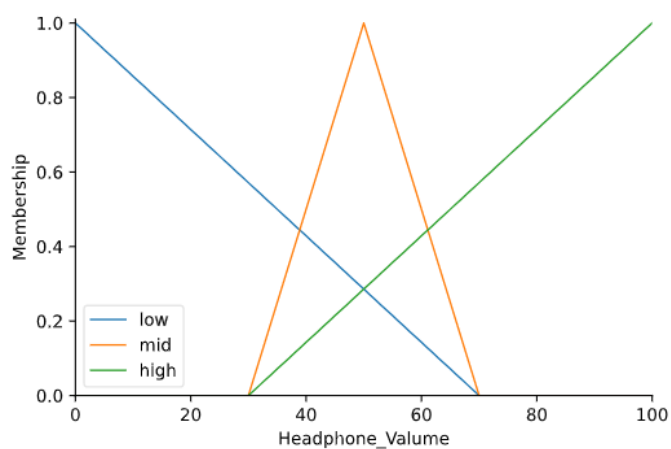
- ระดับเสียงโดยรอบ



- ระดับเสียงของผู้สวมใส่คาดหวังได้ยิน



- ระดับเสียงของหูฟัง



3. Inference

จากหัวข้อที่ผ่านมาได้มีการกำหนด fuzzy set เพื่อที่จะระบุถึงค่าความเป็นสมาชิกของข้อมูล ในหัวข้อนี้จึงทำการกำหนดกฎของระบบ เพื่อที่จะสามารถจัดการกับข้อมูลที่ได้รับเป็นผลลัพธ์ในรูปแบบของ fuzzy set ต่อไป โดยได้มีการกำหนดกฎดังนี้

		around_sound		
	level	low	mid	high
Expected_sound	low	low	low	mid
	mid	low	mid	high
	high	mid	high	high

จากนั้นนำค่าที่ได้ไปสร้าง control system ของหุ่นยนต์

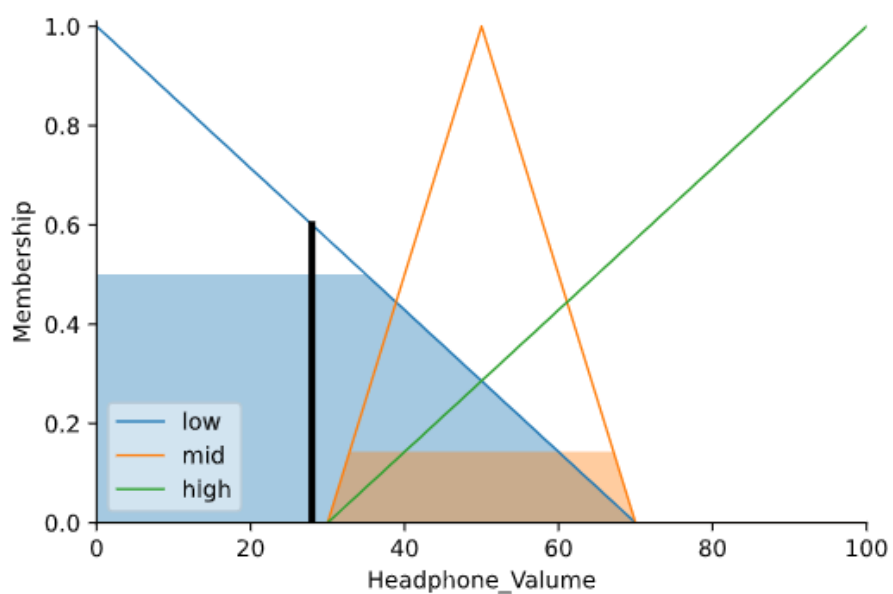
4. Defuzzification

ในส่วนของการทำ defuzzification ใช้กระบวนการ max membership defuzzification โดยเลือกจากฐานนิยม (mode) ให้กลายเป็นค่า output

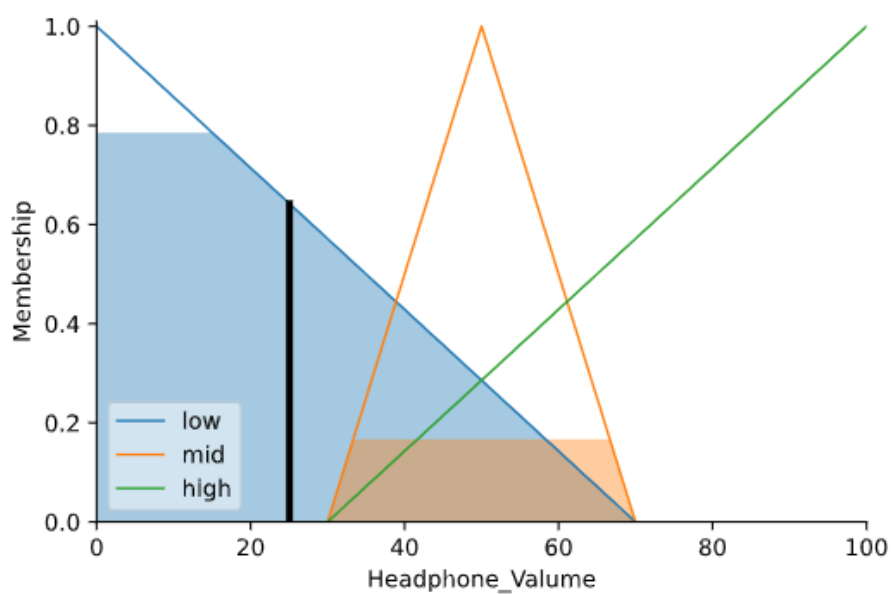
ผลการทดลอง และวิเคราะห์

1. ผลการทดลอง

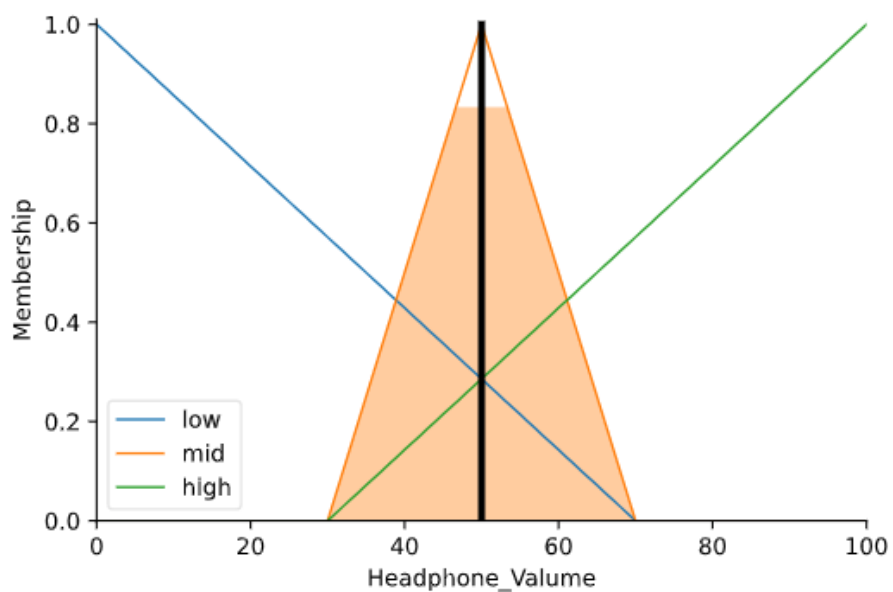
- ทดลอง เสียงรอบข้าง ต่ำ เสียงที่คาดหวัง ต่ำ ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง ต่ำ



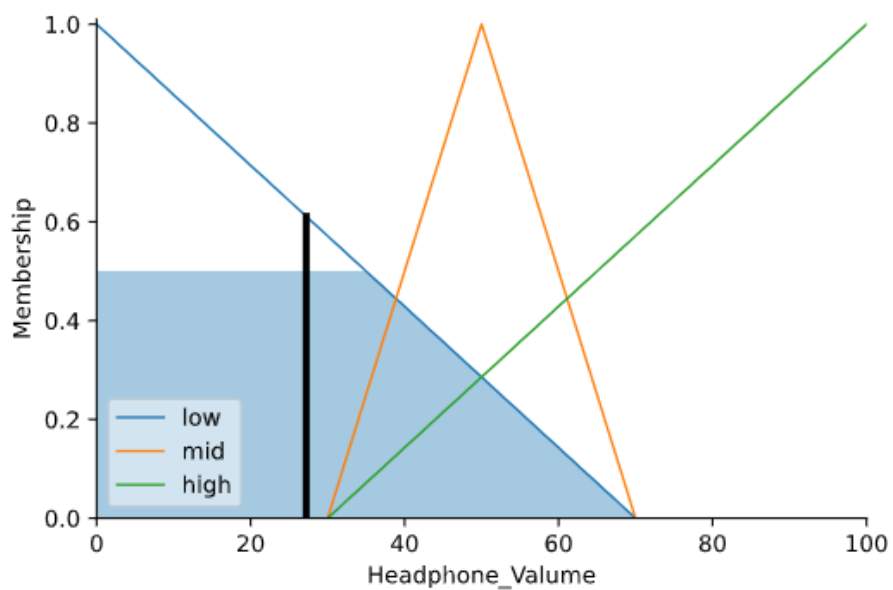
- ทดลอง เสียงรอบข้าง ต่ำ เสียงที่คาดหวัง กลาง ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง กลาง



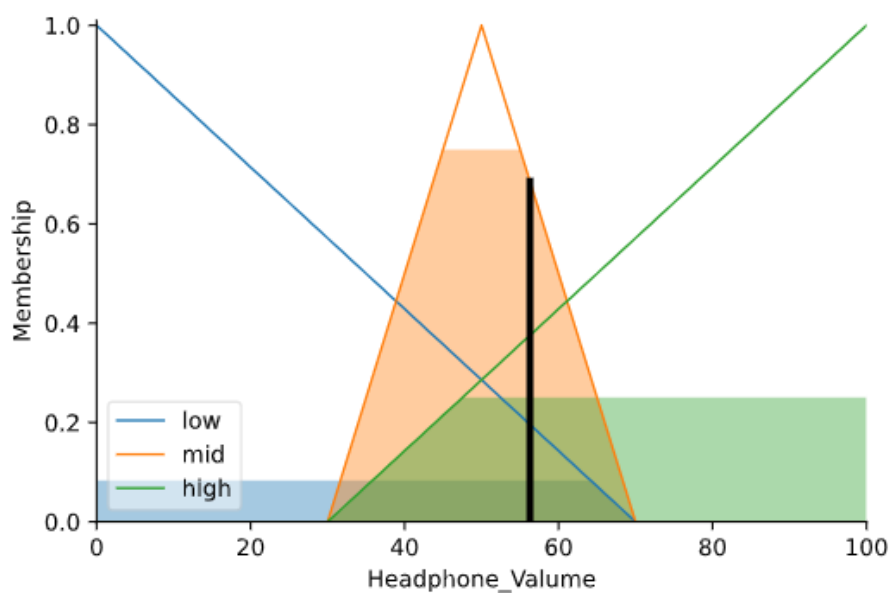
- ทดลอง เสียงรอบข้าง ต่ำ เสียงที่คาดหวัง สูง ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง สูง



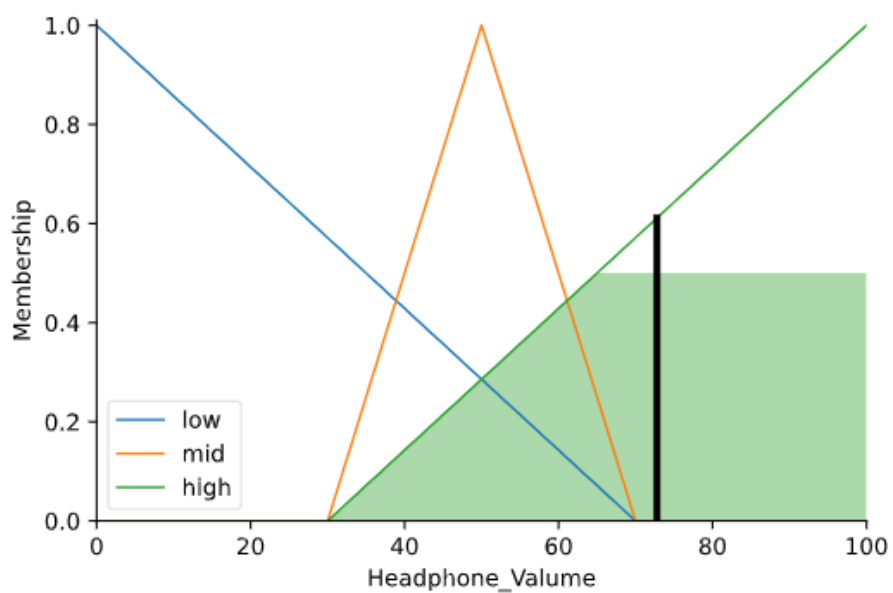
- ทดลอง เสียงรอบข้าง กลาง เสียงที่คาดหวัง ต่ำ ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง ต่ำ



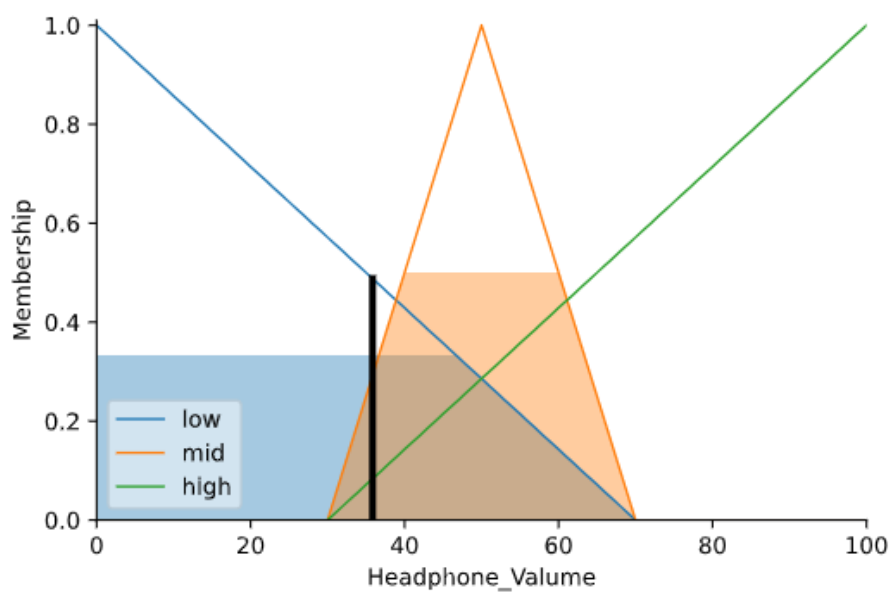
- ทดลอง เสียงรอบข้าง กลาง เสียงที่คาดหวัง กลาง ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง กลาง



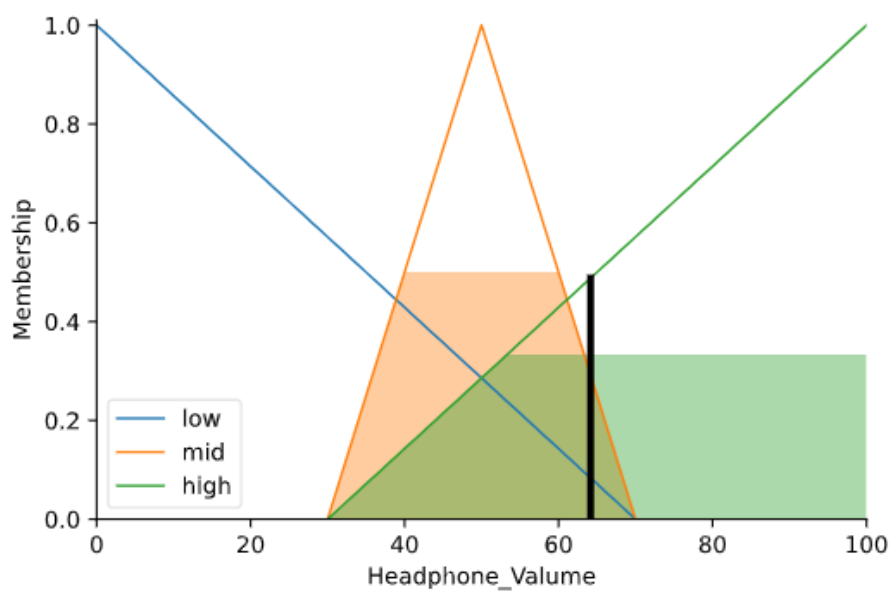
- ทดลอง เสียงรอบข้าง กลาง เสียงที่คาดหวัง สูง ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง สูง



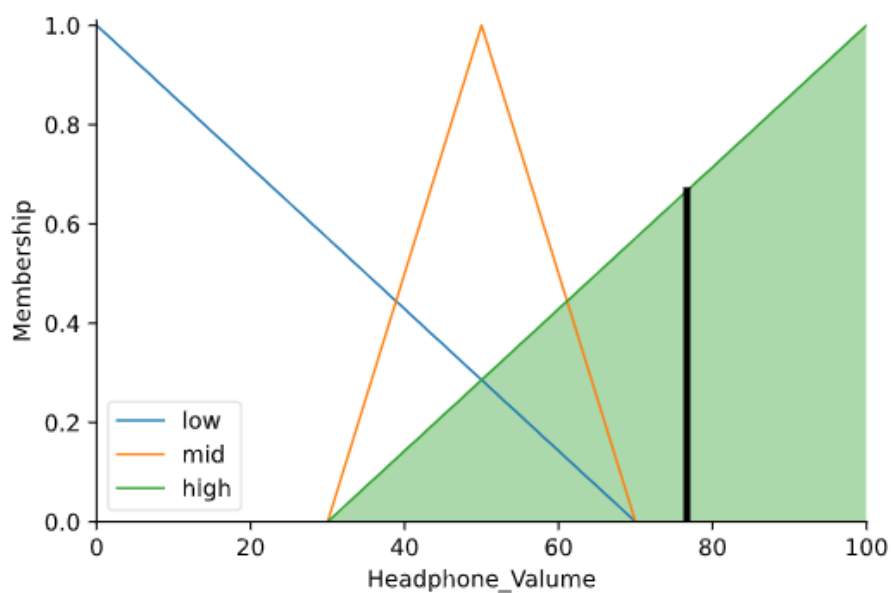
- ทดลอง เสียงรอบข้าง สูง เสียงที่คาดหวัง ต่ำ ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง กลาง



- ทดลอง เสียงรอบข้าง สูง เสียงที่คาดหวัง กลาง ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง สูง



- ทดลอง เสียงรอบข้าง สูง เสียงที่คาดหวัง สูง ผลลัพธ์ที่ได้ควรเป็น เสียงหูฟัง สูง



2. วิเคราะห์ผล

จากการสร้าง Auto Volume Headphone system โดยใช้ fuzzy ผลลัพธ์เป็นไปตามกฎที่ต้องการ แต่อาจจะมีการคลาดเคลื่อนบ้างเนื่องจากการ Defuzzification และผลลัพธ์จะดีหรือไม่ขึ้นอยู่กับผู้สร้างกฎ

```
In [1]: import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
```

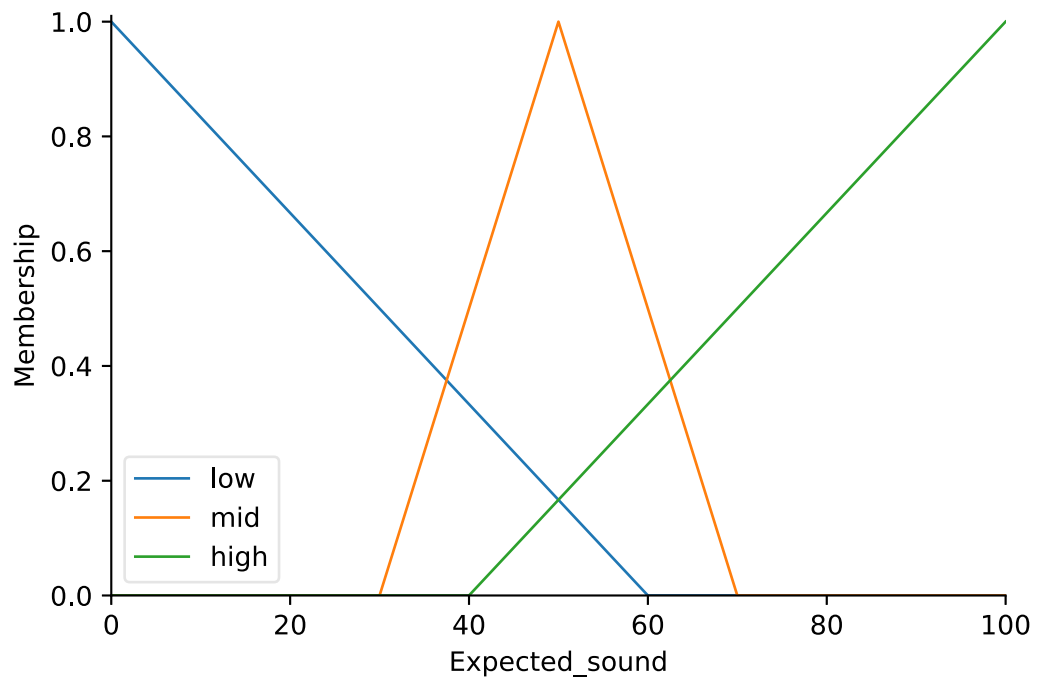
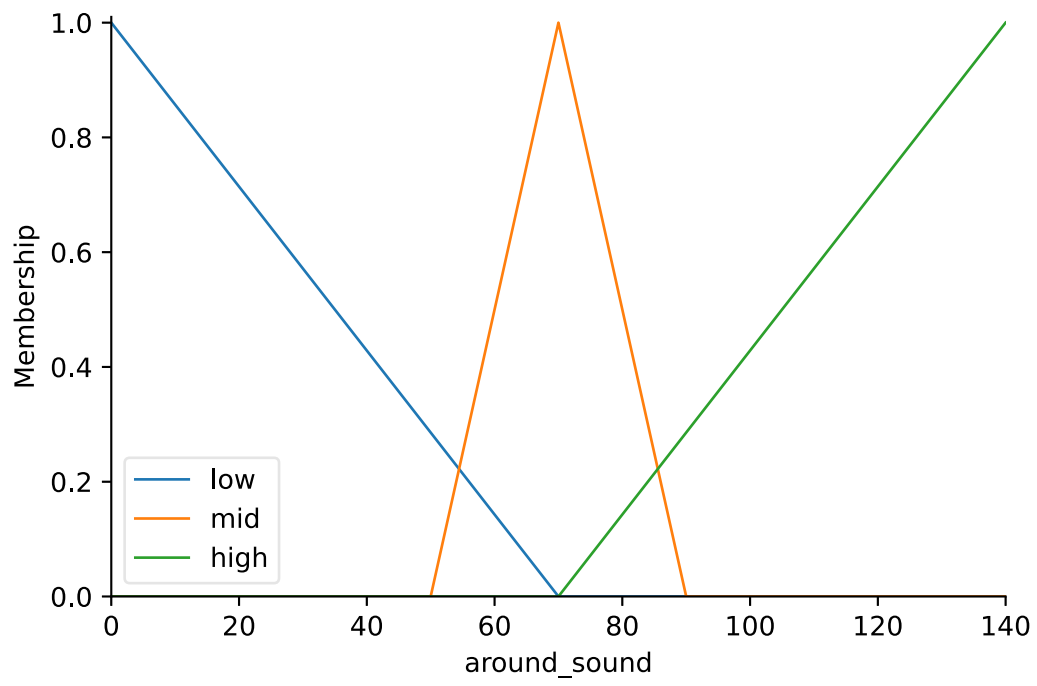
```
In [2]: # define fuzzy variables
around_sound = ctrl.Antecedent(np.arange(0, 141, 1), 'around_sound')
Expected_sound = ctrl.Antecedent(np.arange(0, 101, 1), 'Expected_sound')
Headphone_Valume = ctrl.Consequent(np.arange(0, 101, 1), 'Headphone_Valume')
```

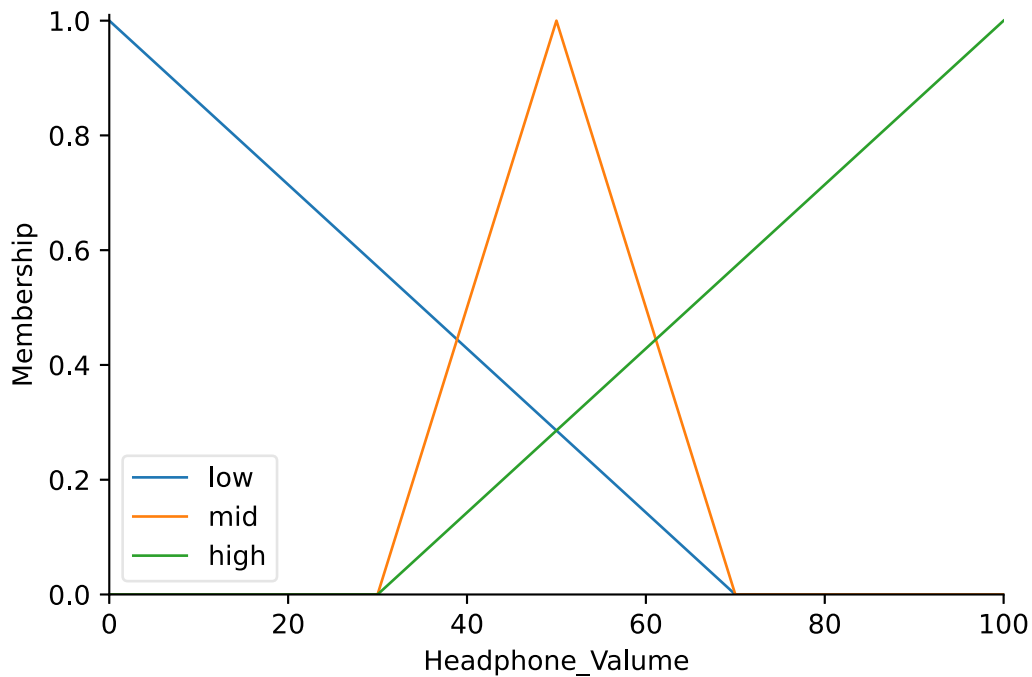
```
In [3]: # add fuzzy membership function
around_sound['low'] = fuzz.trimf(around_sound.universe, [0, 0, 70])
around_sound['mid'] = fuzz.trimf(around_sound.universe, [50, 70, 90])
around_sound['high'] = fuzz.trimf(around_sound.universe, [70, 140, 140])

Expected_sound['low'] = fuzz.trimf(Expected_sound.universe, [0, 0, 60])
Expected_sound['mid'] = fuzz.trimf(Expected_sound.universe, [30, 50, 70])
Expected_sound['high'] = fuzz.trimf(Expected_sound.universe, [40, 100, 100])

Headphone_Valume['low'] = fuzz.trimf(Headphone_Valume.universe, [0, 0, 70])
Headphone_Valume['mid'] = fuzz.trimf(Headphone_Valume.universe, [30, 50, 70])
Headphone_Valume['high'] = fuzz.trimf(Headphone_Valume.universe, [30, 100, 100])
```

```
In [4]: around_sound.view()  
Expected_sound.view()  
Headphone_Valume.view()  
plt.show()
```





```
In [5]: # fuzzy rules
rule1 = ctrl.Rule(around_sound['low'] & Expected_sound['low'], Headphone_Volume['low'])
rule2 = ctrl.Rule(around_sound['low'] & Expected_sound['mid'], Headphone_Volume['low'])
rule3 = ctrl.Rule(around_sound['low'] & Expected_sound['high'], Headphone_Volume['mid'])
rule4 = ctrl.Rule(around_sound['mid'] & Expected_sound['low'], Headphone_Volume['low'])
rule5 = ctrl.Rule(around_sound['mid'] & Expected_sound['mid'], Headphone_Volume['mid'])
rule6 = ctrl.Rule(around_sound['mid'] & Expected_sound['high'], Headphone_Volume['high'])
rule7 = ctrl.Rule(around_sound['high'] & Expected_sound['low'], Headphone_Volume['mid'])
rule8 = ctrl.Rule(around_sound['high'] & Expected_sound['mid'], Headphone_Volume['high'])
rule9 = ctrl.Rule(around_sound['high'] & Expected_sound['high'], Headphone_Volume['high'])
```

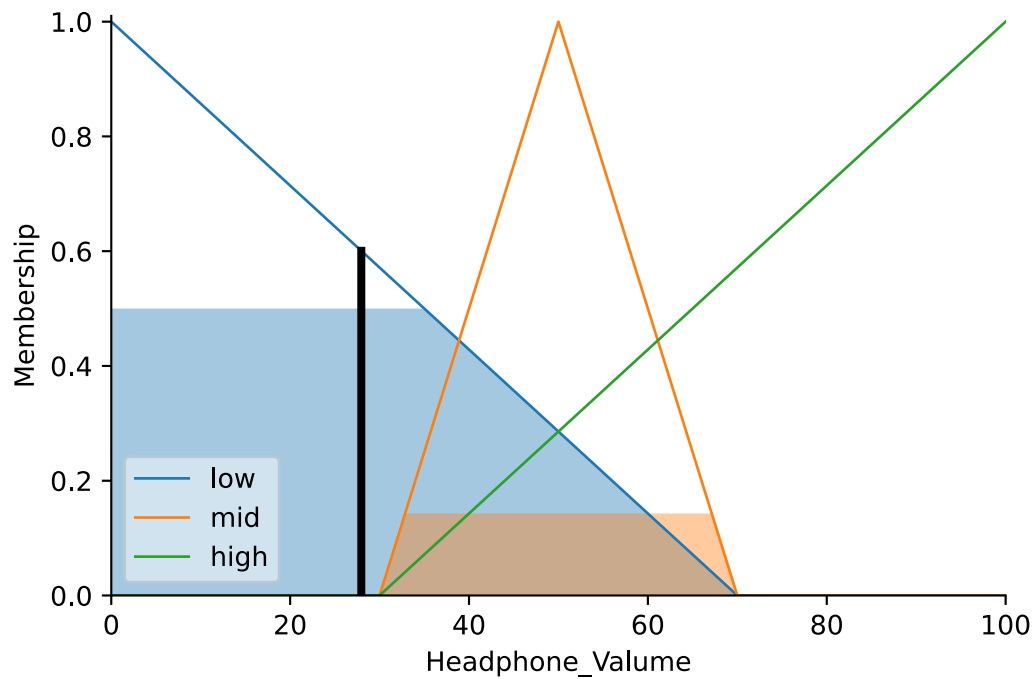
```
In [6]: # add rules to control system
Headphone_Volume_rule = ctrl.ControlSystem(
    [rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9])

Headphone_Volume_output = ctrl.ControlSystemSimulation(Headphone_Volume_rule)
```

```
In [7]: def simulate(around_sound, Expected_sound):
    Headphone_Volume_output.input['around_sound'] = around_sound
    Headphone_Volume_output.input['Expected_sound'] = Expected_sound
    Headphone_Volume_output.compute()
    print(
        f'input : around_sound = {around_sound} db,Expected_sound = {Expected_sound} %')
    print(
        f'output : Headphone_Volume_output = {Headphone_Volume_output.output["Headphon"
e_Volume"]} %')
    Headphone_Volume.view(sim = Headphone_Volume_output)
    plt.show()
```

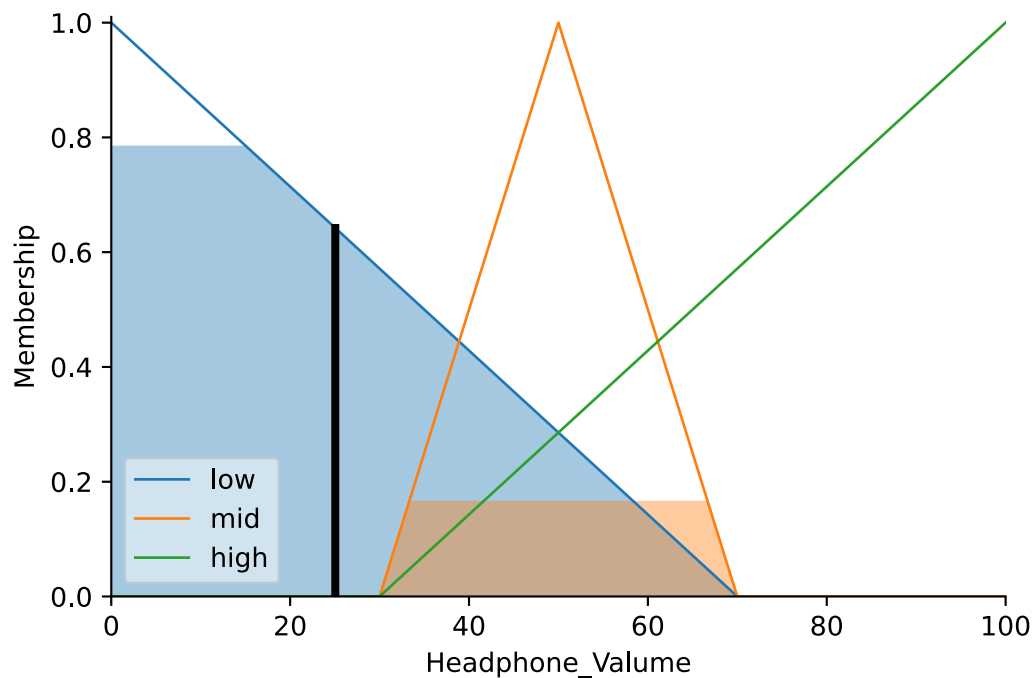
```
In [8]: # simulation 1
simulate(around_sound=20, Expected_sound=2)
```

input : around_sound = 80 db,Expected_sound = 2 %
output : Headphone_Volume_output = 27.95610331835309 %



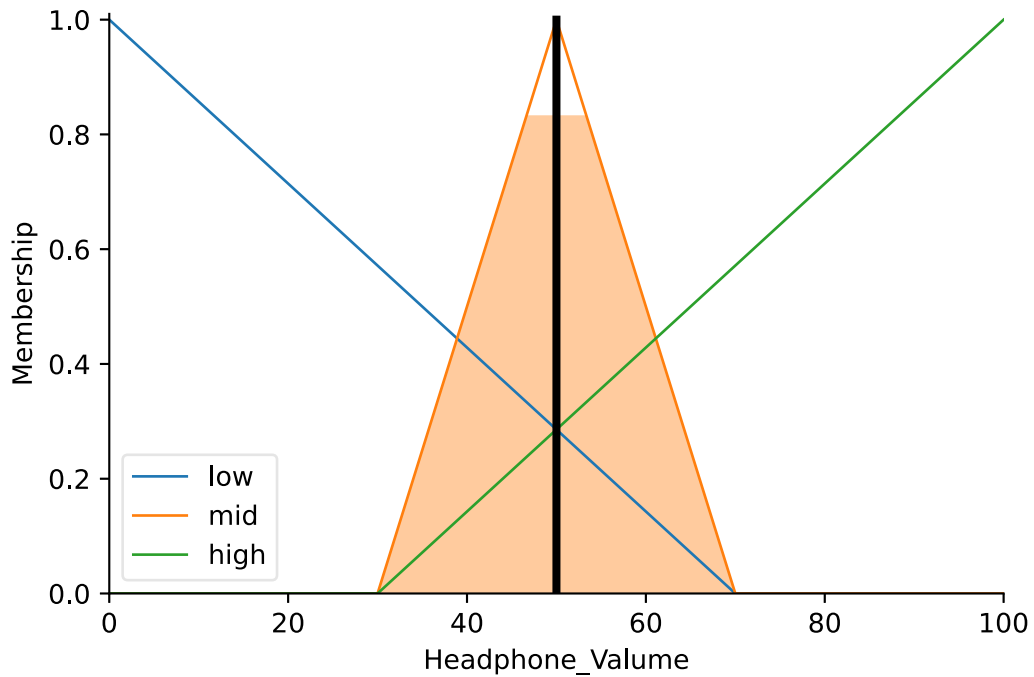
```
In [9]: # simulation 2
simulate(around_sound=15, Expected_sound=50)
```

input : around_sound = 15 db,Expected_sound = 50 %
output : Headphone_Volume_output = 25.048120899815828 %



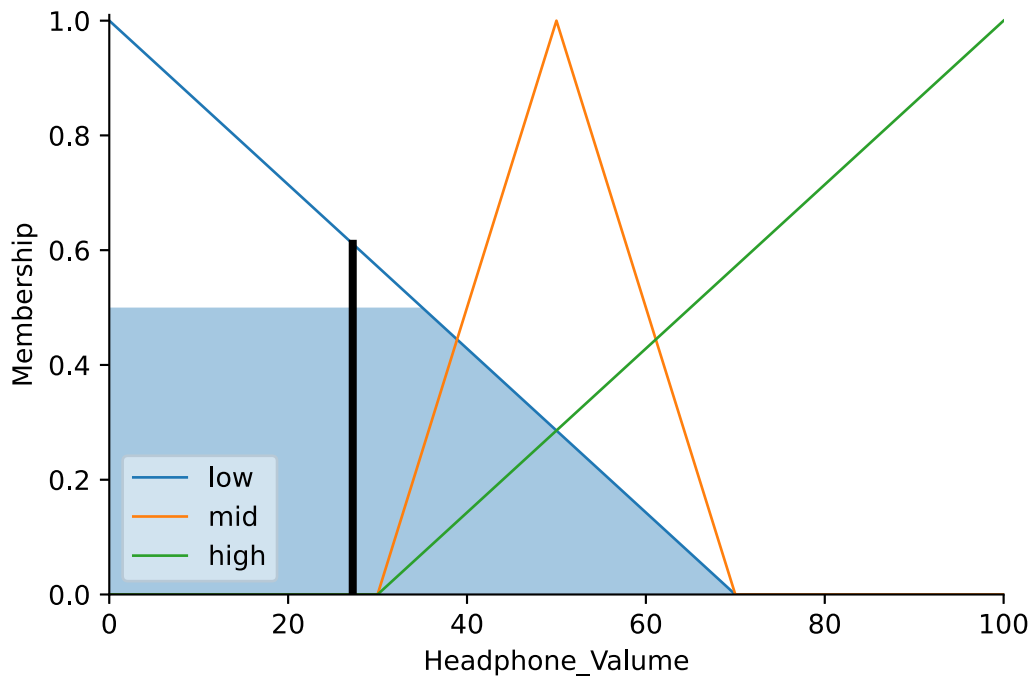
```
In [10]: # simulation 3
simulate(around_sound=10, Expected_sound=90)
```

input : around_sound = 10 db,Expected_sound = 90 %
output : Headphone_Volume_output = 49.99999999999999 %



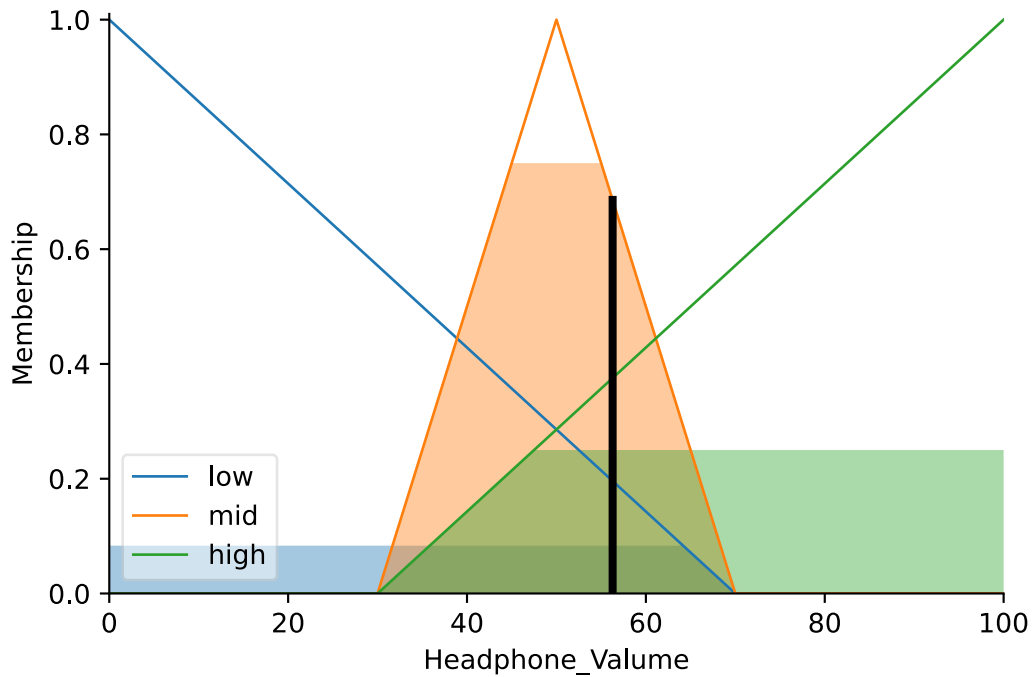
```
In [11]: # simulation 4
simulate(around_sound=60, Expected_sound=10)
```

input : around_sound = 60 db,Expected_sound = 10 %
output : Headphone_Volume_output = 27.22222222222232 %



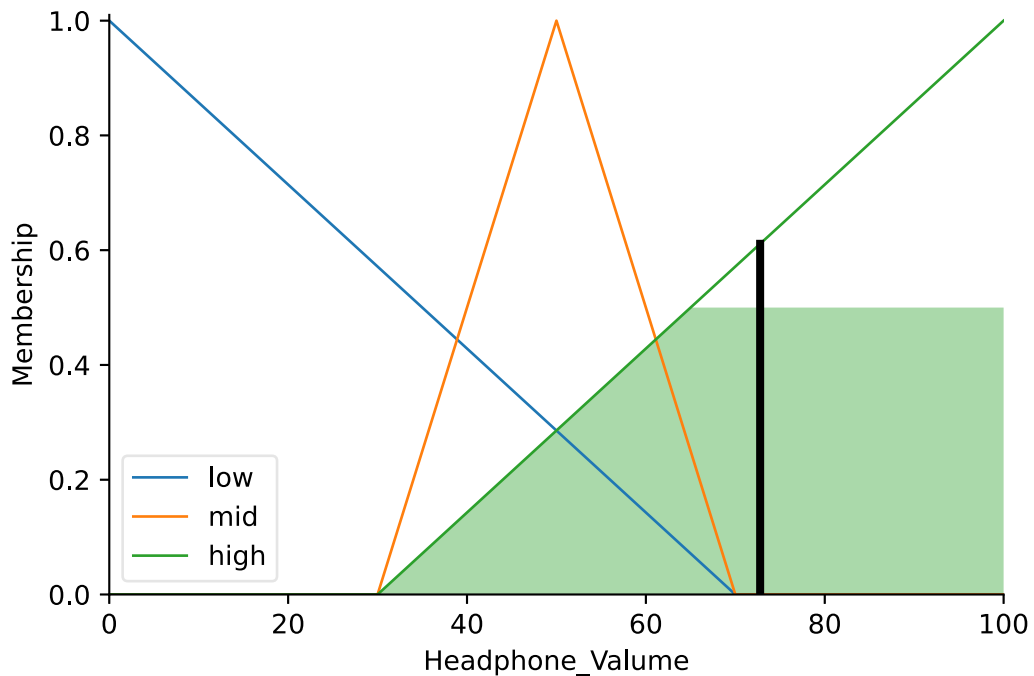

```
In [12]: # simulation 5
simulate(around_sound=70, Expected_sound=55)
```

input : around_sound = 70 db,Expected_sound = 55 %
output : Headphone_Volume_output = 56.28202225995095 %



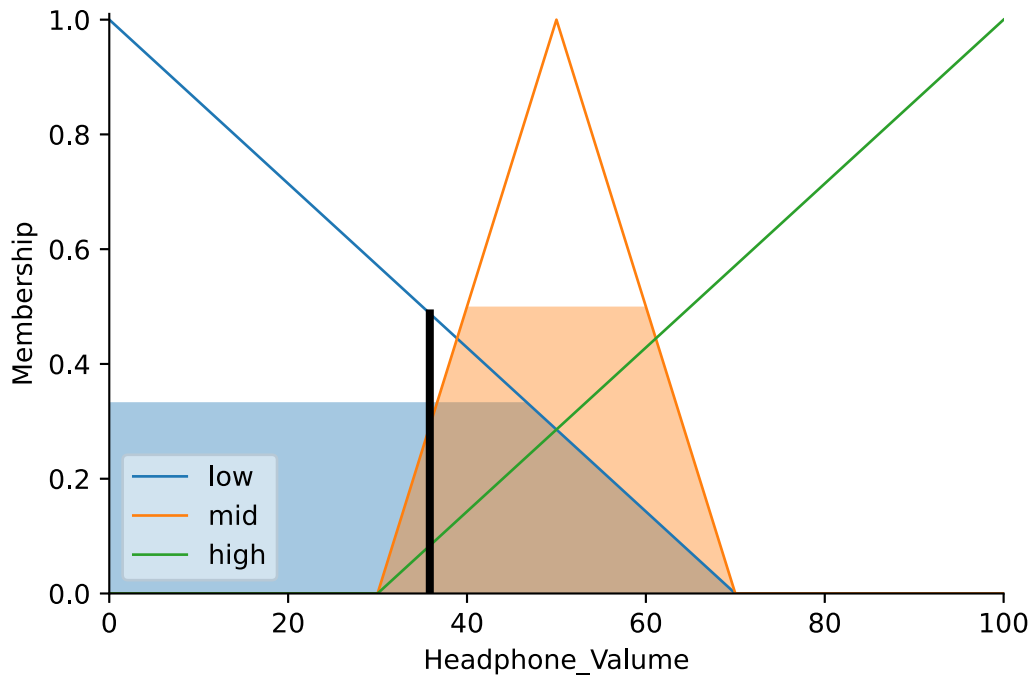
```
In [13]: # simulation 6
simulate(around_sound=80, Expected_sound=100)
```

input : around_sound = 80 db,Expected_sound = 100 %
output : Headphone_Volume_output = 72.77777777777779 %



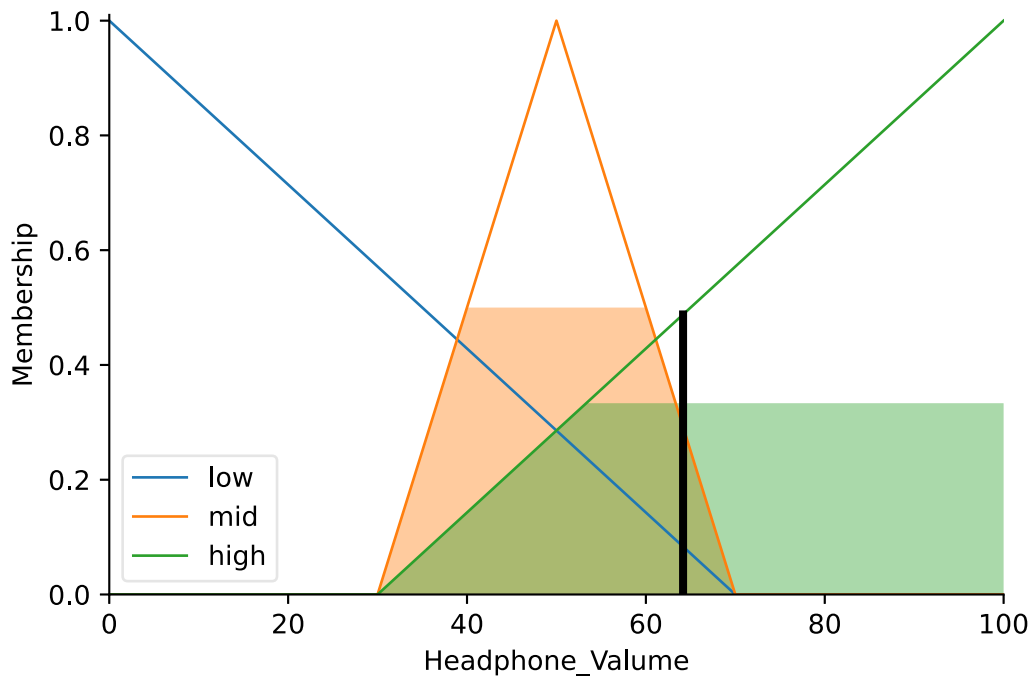
```
In [19]: # simulation 7
simulate(around_sound=70, Expected_sound=40)
```

input : around_sound = 70 db,Expected_sound = 40 %
output : Headphone_Volume_output = 35.83939587321846 %



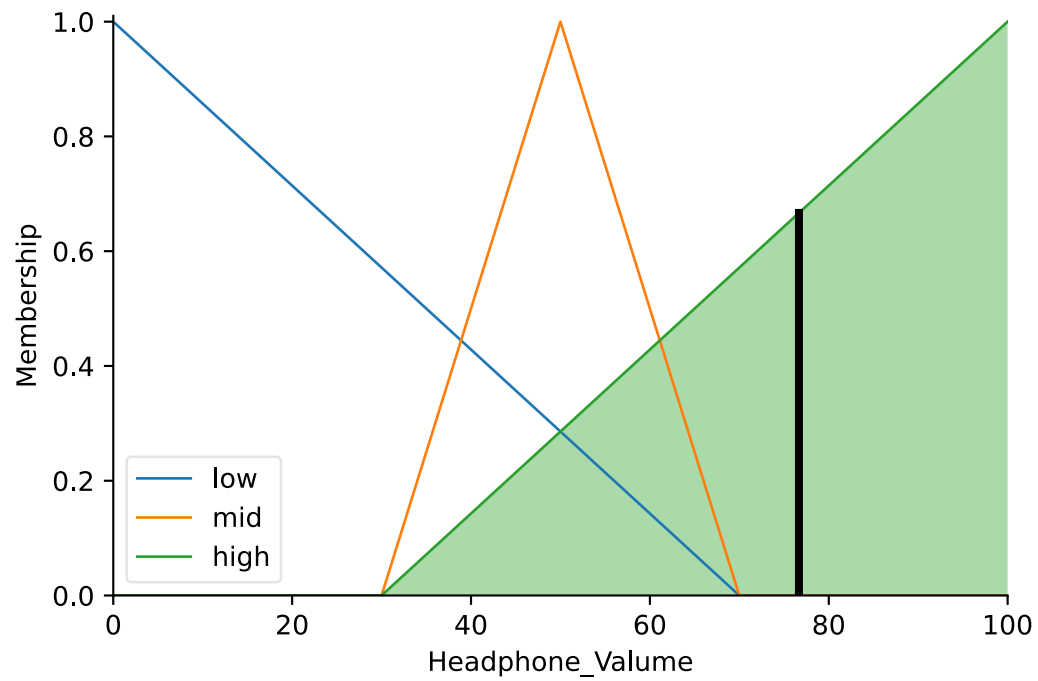
```
In [15]: # simulation 8
simulate(around_sound=80, Expected_sound=60)
```

input : around_sound = 80 db,Expected_sound = 60 %
output : Headphone_Volume_output = 64.16060412678162 %



```
In [16]: # simulation 9
simulate(around_sound=140, Expected_sound=100)
```

input : around_sound = 140 db,Expected_sound = 100 %
output : Headphone_Volume_output = 76.66666666666669 %



In []: