Computer Assignment 2  
CPE 261456 (Introduction to Computational Intelligence)

โดย

นายธนาคม หัสแดง  
รหัสนักศึกษา 590610624

เสนอ  
ผศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล   
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

**Sous Vide Fuzzy Logic Simulator**

**ลักษณะการทำงาน :**

เริ่มต้นจากการกำหนด ข้อกำหนด ทั้ง 4 แบบ คือ

* Size โดยจะมี 3 ระดับ คือ
  + SMALL
  + MEDIUM
  + BIG
* Time โดยจะมี 3 ระดับ คือ
  + SHORT
  + MEDIUM
  + LONG
* Temperature โดยจะมี 4 ระดับ คือ
  + Moderate
  + WARM
  + Very Warm
  + Hot
  + Scorching
* Done โดยจะมี 4 ระดับ คือ
  + Rare
  + Medium Rare
  + Medium
  + Medium Well
  + Well Done

โดยเมื่อกำหนดข้อกำหนดแล้ว ก็จะนำ กฎที่ตั้งเข้าสู่ Fuzzy และ เริ่มคำนวณจาก Input โดยการที่จะเลือกตัวแปรที่มีค่า Input ตรงกับ กฎ นั้น ก็จะนำค่าสมาชิกที่ได้ มา และ หาค่าน้อยทีสุดของ แต่ละ กฎ และเมื่อได้ครบทุกกฎแล้ว จะนำค่าของ ทุกกฎมาหา ค่ามากที่สุด จากทั้งหมด และเมื่อได้ค่ามากที่สุดแล้ว จึงไปทำการ defuzzification โดยการหา Centroid ก็จะได้ผลลัพธ์ออกมาอยู่ในช่วงของ คำตอบที่ต้องการ

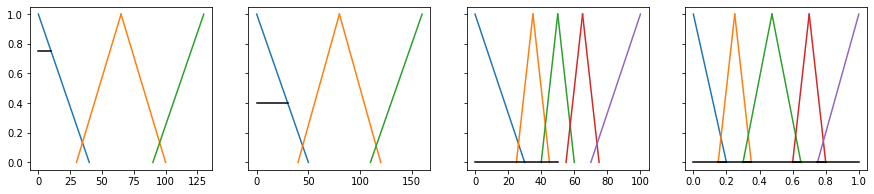
**Rule ที่ใช้:**

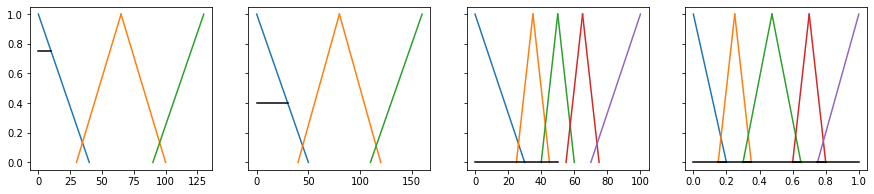
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **กฎข้อที่** | **If (ขนาด, Size)** | **If (เวลา, Time)** | **If (อุณหภูมิ, Temp)** | **Else (ความสุก, Done)** |
| 1 | SMALL | SHORT | Moderate | Rare |
| 2 | SMALL | SHORT | Warm | Medium Rare |
| 3 | SMALL | SHORT | Very Warm | Medium Rare |
| 4 | SMALL | SHORT | Hot | Medium |
| 5 | SMALL | SHORT | Scorching | Medium |
| 6 | SMALL | MEDUIM | Moderate | Medium Rare |
| 7 | SMALL | MEDUIM | Warm | Medium |
| 8 | SMALL | MEDUIM | Very Warm | Medium |
| 9 | SMALL | MEDUIM | Hot | Medium Well |
| 10 | SMALL | MEDUIM | Scorching | Medium Well |
| 11 | SMALL | LONG | Moderate | Medium |
| 12 | SMALL | LONG | Warm | Medium Well |
| 13 | SMALL | LONG | Very Warm | Medium Well |
| 14 | SMALL | LONG | Hot | Well Done |
| 15 | SMALL | LONG | Scorching | Well Done |
| 16 | MEDIUM | SHORT | Moderate | Rare |
| 17 | MEDIUM | SHORT | Warm | Rare |
| 18 | MEDIUM | SHORT | Very Warm | Medium Rare |
| 19 | MEDIUM | SHORT | Hot | Medium Rare |
| 20 | MEDIUM | SHORT | Scorching | Medium |
| 21 | MEDIUM | MEDUIM | Moderate | Medium Rare |
| **กฎข้อที่** | **If (ขนาด, Size)** | **If (เวลา, Time)** | **If (อุณหภูมิ, Temp)** | **Else (ความสุก, Done)** |
| 22 | MEDIUM | MEDUIM | Warm | Medium Rare |
| 23 | MEDIUM | MEDUIM | Very Warm | Medium |
| 24 | MEDIUM | MEDUIM | Hot | Medium |
| 25 | MEDIUM | MEDUIM | Scorching | Medium Well |
| 26 | MEDIUM | LONG | Moderate | Medium |
| 27 | MEDIUM | LONG | Warm | Medium |
| 28 | MEDIUM | LONG | Very Warm | Medium Well |
| 29 | MEDIUM | LONG | Hot | Medium Well |
| 30 | MEDIUM | LONG | Scorching | Well Done |
| 31 | BIG | SHORT | Moderate | Rare |
| 32 | BIG | SHORT | Warm | Rare |
| 33 | BIG | SHORT | Very Warm | Rare |
| 34 | BIG | SHORT | Hot | Medium Rare |
| 35 | BIG | SHORT | Scorching | Medium Rare |
| 36 | BIG | MEDUIM | Moderate | Medium Rare |
| 37 | BIG | MEDUIM | Warm | Medium Rare |
| 38 | BIG | MEDUIM | Very Warm | Medium Rare |
| 39 | BIG | MEDUIM | Hot | Medium |
| 40 | BIG | MEDUIM | Scorching | Medium |
| 41 | BIG | LONG | Moderate | Medium |
| 42 | BIG | LONG | Warm | Medium |
| 43 | BIG | LONG | Very Warm | Medium |
| 44 | BIG | LONG | Hot | Medium Well |
| 45 | BIG | LONG | Scorching | Medium Well |

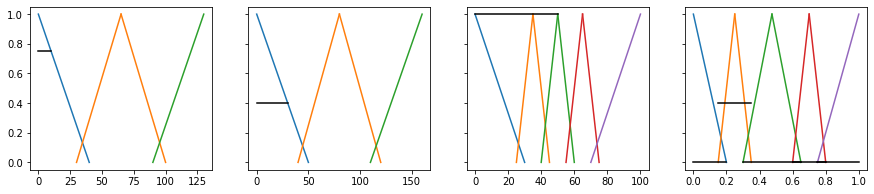
**Simulator**

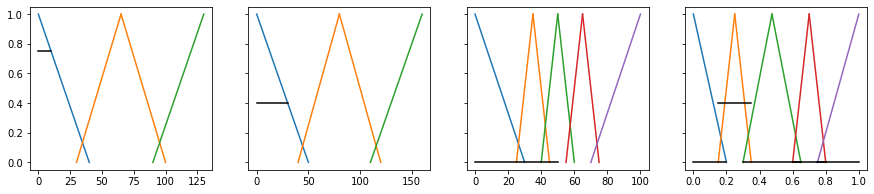
กำหนด Input คือ

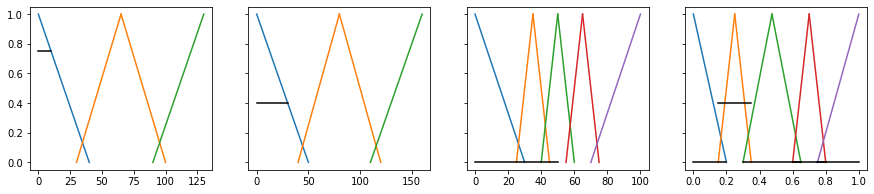
* Size = 10
* Time = 30
* Temp = 50

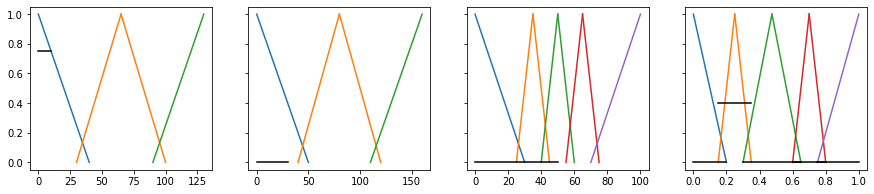
กฎข้อที่ 1 : 

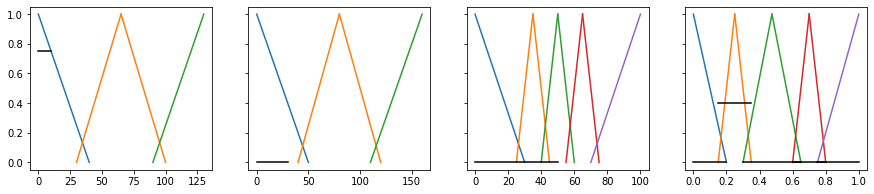
กฎข้อที่ 2 : 

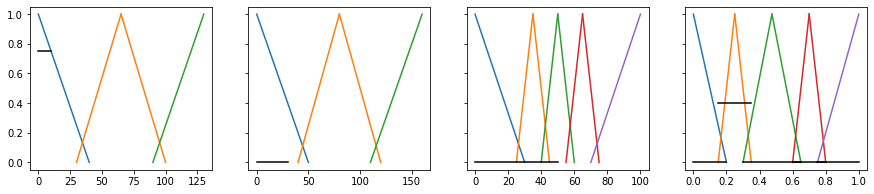
กฎข้อที่ 3 : 

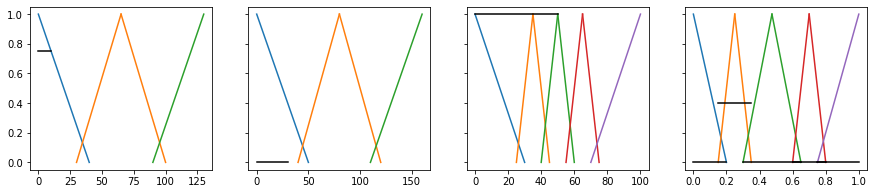
กฎข้อที่ 4 : 

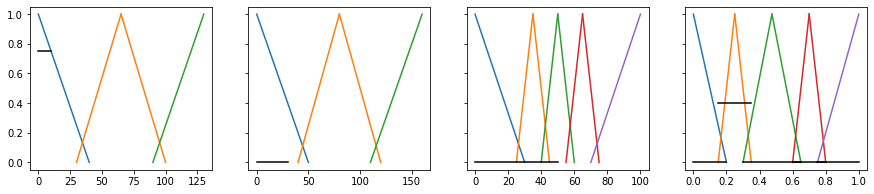
กฎข้อที่ 5 : 

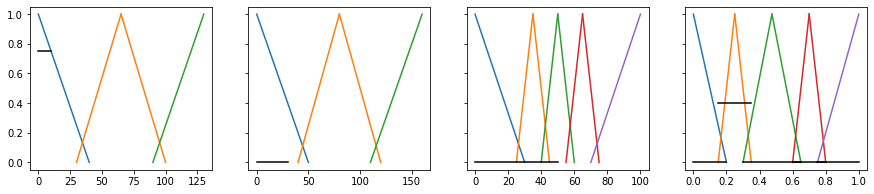
กฎข้อที่ 6 : 

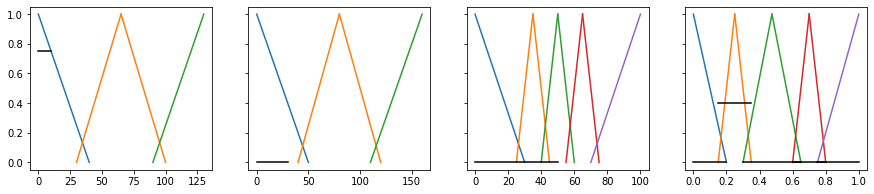
กฎข้อที่ 7 : 

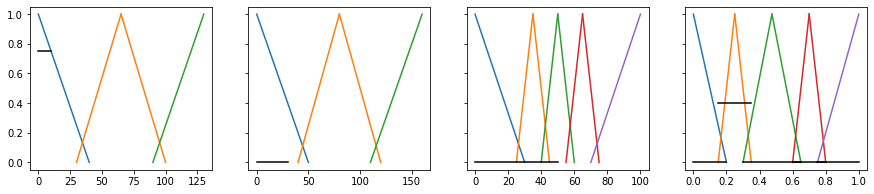
กฎข้อที่ 8 : 

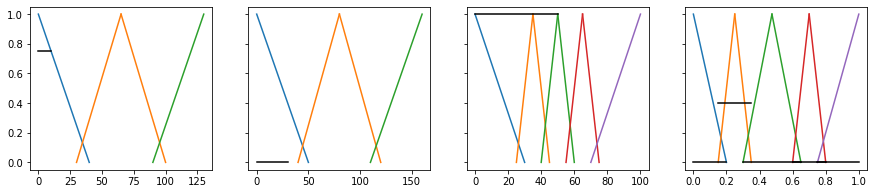
กฎข้อที่ 9 : 

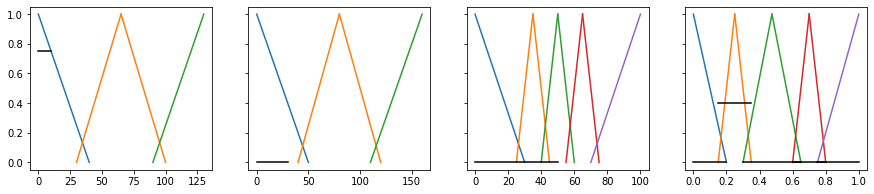
กฎข้อที่ 10 : 

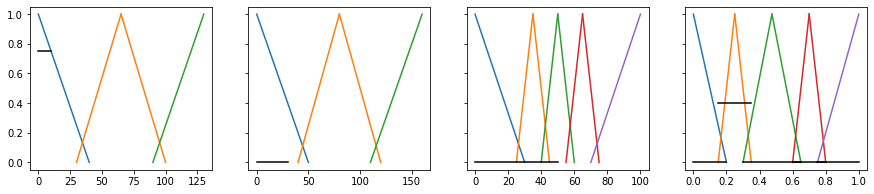
กฎข้อที่ 11 : 

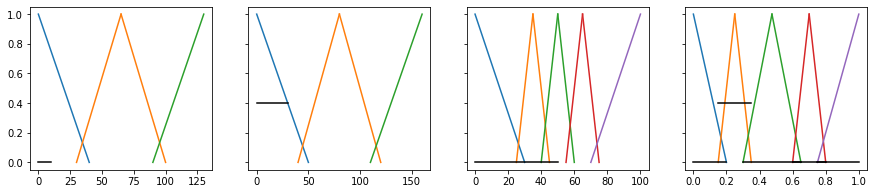
กฎข้อที่ 12 : 

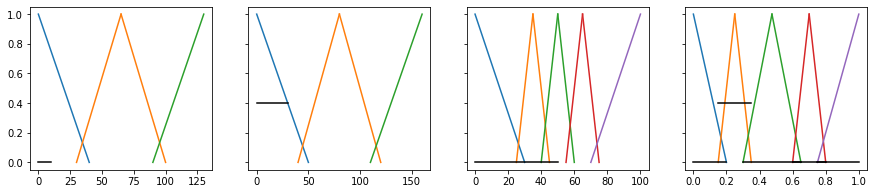
กฎข้อที่ 13 : 

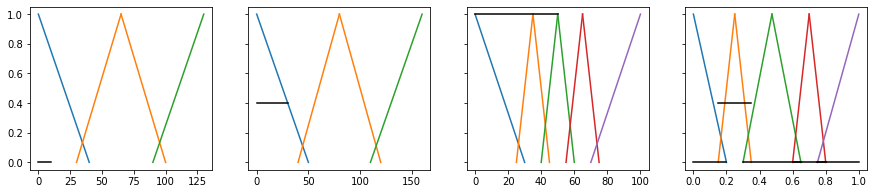
กฎข้อที่ 14 : 

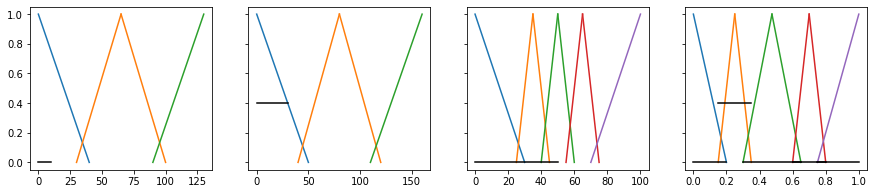
กฎข้อที่ 15 : 

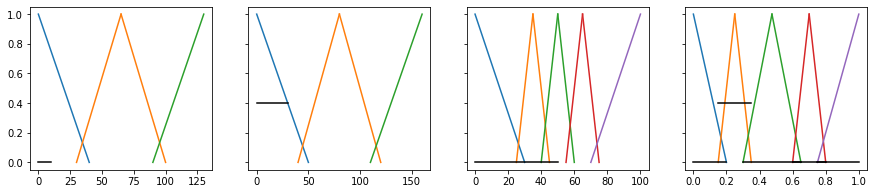
กฎข้อที่ 16 : 

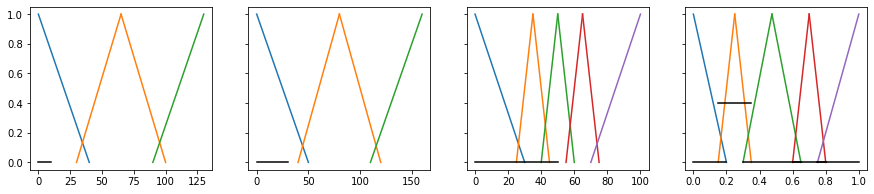
กฎข้อที่ 17 : 

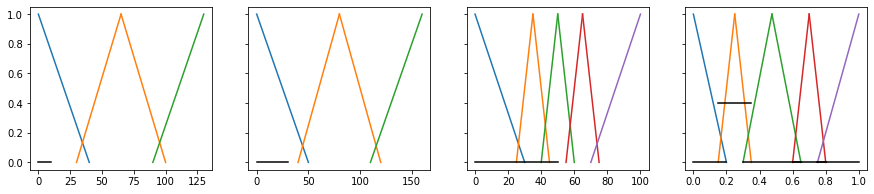
กฎข้อที่ 18 : 

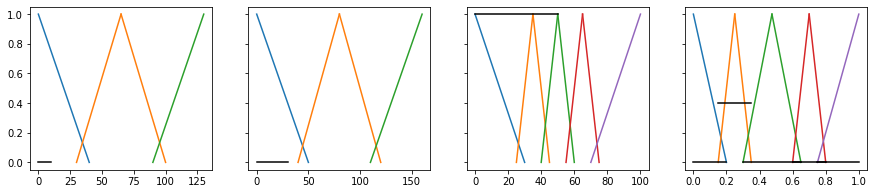
กฎข้อที่ 19 : 

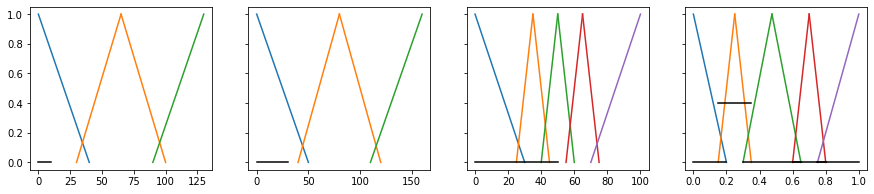
กฎข้อที่ 20 : 

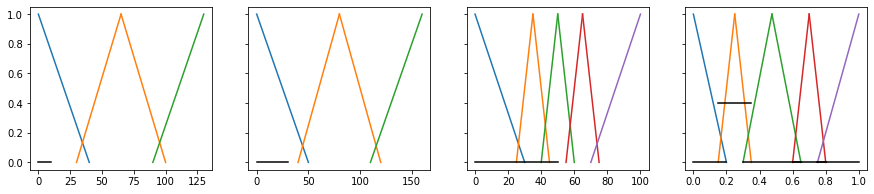
กฎข้อที่ 21 : 

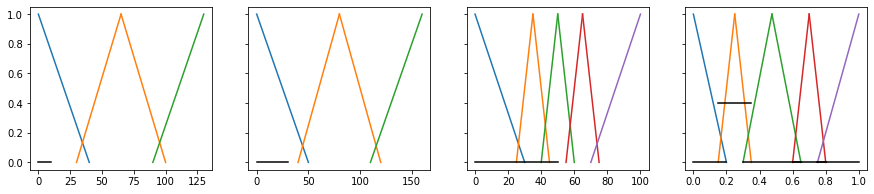
กฎข้อที่ 22 : 

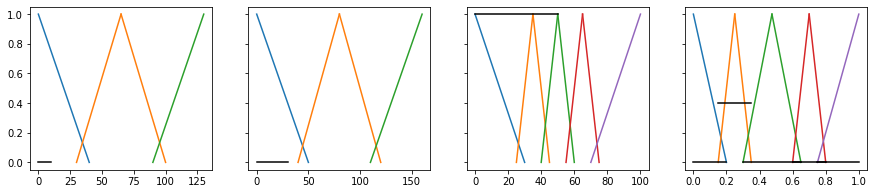
กฎข้อที่ 23 : 

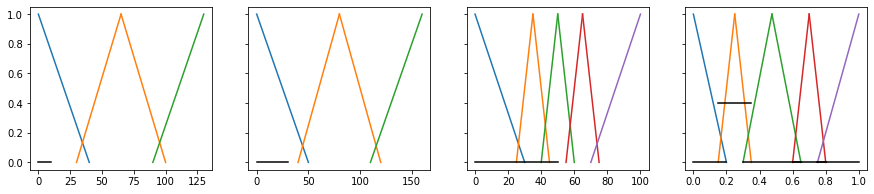
กฎข้อที่ 24 : 

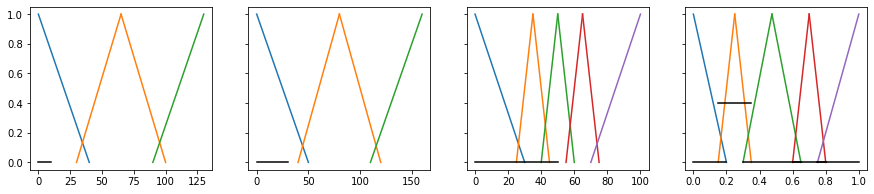
กฎข้อที่ 25 : 

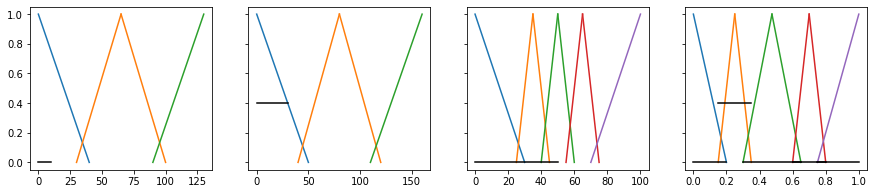
กฎข้อที่ 26 : 

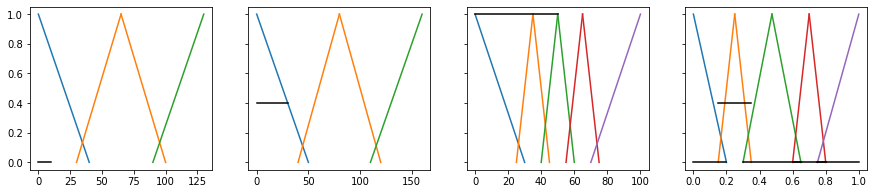
กฎข้อที่ 27 : 

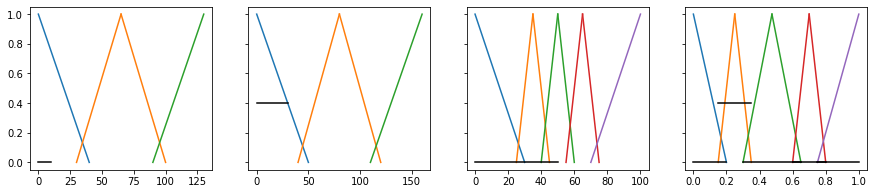
กฎข้อที่ 28 : 

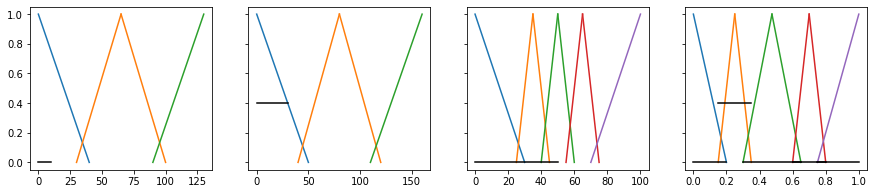
กฎข้อที่ 29 : 

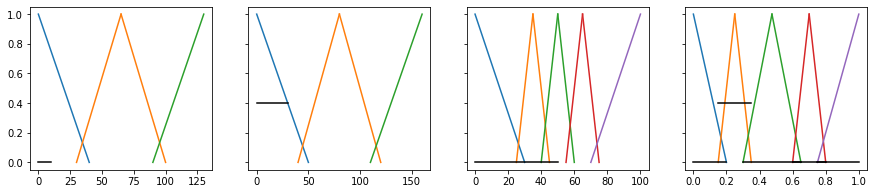
กฎข้อที่ 30 : 

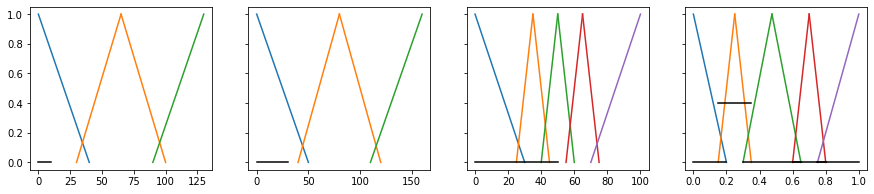
กฎข้อที่ 31 : 

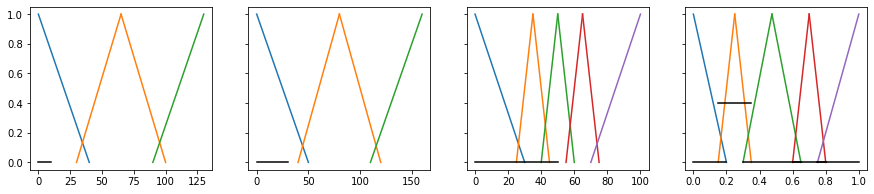
กฎข้อที่ 32 : 

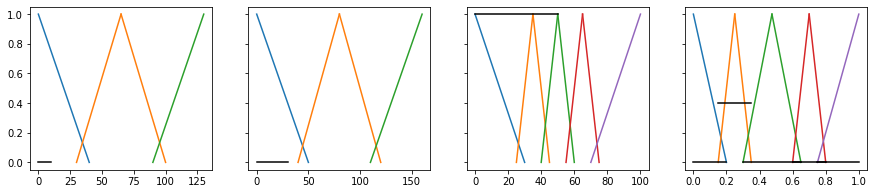
กฎข้อที่ 33 : 

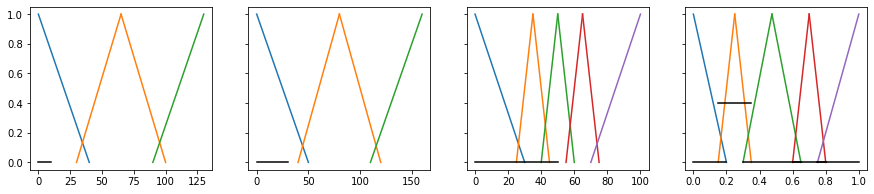
กฎข้อที่ 34 : 

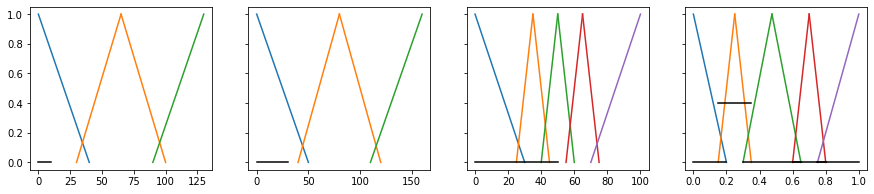
กฎข้อที่ 35 : 

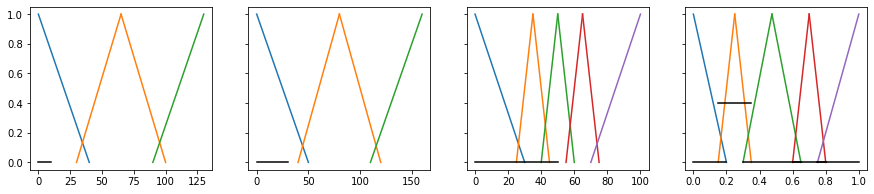
กฎข้อที่ 36 : 

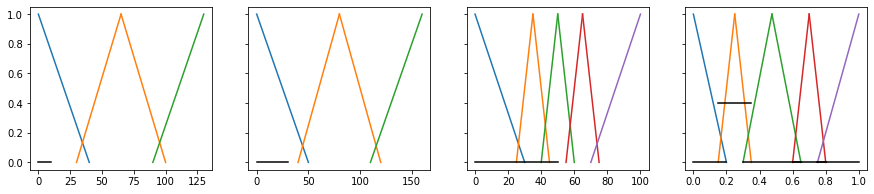
กฎข้อที่ 37 : 

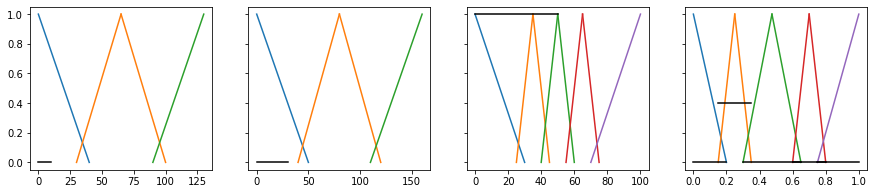
กฎข้อที่ 38 : 

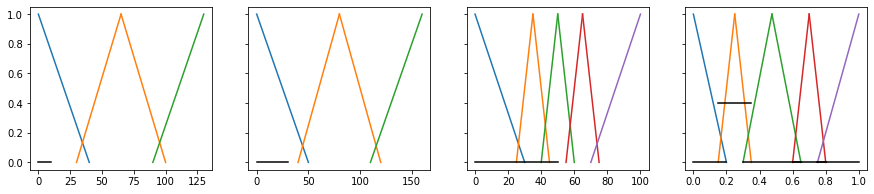
กฎข้อที่ 39 : 

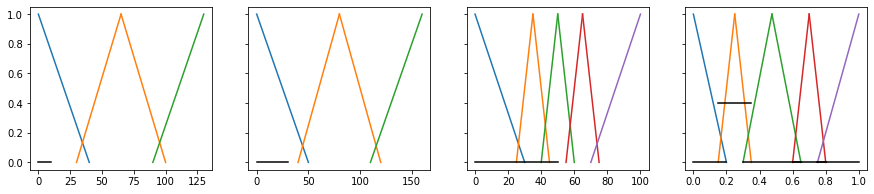
กฎข้อที่ 40 : 

กฎข้อที่ 41 : 

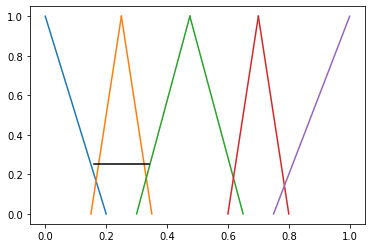
กฎข้อที่ 42 : 

กฎข้อที่ 43 : 

กฎข้อที่ 44 : 

กฎข้อที่ 45 : 

Output:



โดยเมื่อได้ Output แล้วจะนำไป Defuzzification โดยจะได้ค่าระดับความสุข = 0.25

**ผลการทดลอง**

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | | | **OUTPUT** |
| **If (ขนาด, Size)** | **If (เวลา, Time)** | **If (อุณหภูมิ, Temp)** | **Else (ความสุก, Done)** |
| 10 | 10 | 10 | 0.16342960288807 |
| 34 | 67 | 25 | 0.25000000000000 |
| 34 | 120 | 65 | 0.44273615941347 |
| 57 | 150 | 74 | 0.42808481532147 |
| 68 | 20 | 100 | 0.38489098408956 |
| 120 | 87 | 78 | 0.35111531190926 |
| 120 | 46 | 36 | 0.23071428571428 |
| 130 | 39 | 49 | 0.20533834586466 |

**วิเคราะห์ผลการทดลอง**

จะเห็นว่าเมื่อนำค่า Input ทั้งหมด 3 ค่า (Size , Time , Temp) มาใส่เข้าไปยัง Fuzzy Simulator จะได้ผลลัพธ์ที่ออกมาอยู่ในช่วงที่ตรงกับกฎที่ตั้งไว้ โดยเมื่อได้ผลลัพธ์ที่ตรงแม้จะเปลี่ยนแปลงค่า Input ทั้งหมด จึงสรุปได้ว่า Fuzzy Simulator นั้นทำงานได้อยากถูกต้อง แต่ข้อจำกัดของ Simulator นี้ก็คือ เมื่อเราใส่ Input ค่าอยู่เกิน หรือ อยู่นอกช่วงของขอบเขต ที่เรากำหนดไว้ จะทำให้เกิด Error ขึ้น

**Code :**

1. **import** numpy **as** np
2. **import** matplotlib.pyplot **as** plt
3. **from** enum **import** Enum, auto
4. **import** matplotlib.pyplot **as** plt
5. **class** Fuzzy(Enum):
6. SIZE = auto()
7. TIME = auto()
8. TEMPURETURE = auto()
9. DONENESS = auto()
11. **class** SIZELVL(Enum):
12. SMALL = auto()
13. MEDIUM = auto()
14. BIG = auto()
16. **class** TIMELVL(Enum):
17. SHORT = auto()
18. MEDIUM = auto()
19. LONG = auto()
21. **class** TEMPLVL(Enum):
22. MODERATE = auto()
23. WARM = auto()
24. VERY\_WARM = auto()
25. HOT = auto()
26. SCORCHING = auto()
28. **class** DONELVL(Enum):
29. RARE = auto()
30. MEDIUM\_RARE = auto()
31. MEDIUM = auto()
32. MEDIUM\_WELL = auto()
33. WELL\_DONE = auto()
35. **class** range\_step :
36. **def** \_\_init\_\_(self,start,end,step):
37. self.start = start
38. self.end = end
39. self.step =step
41. **class** Rule :
42. **def** \_\_init\_\_(self,ifRule,thenRule):
43. self.ifRule = ifRule
44. self.thenRule = thenRule
46. **class** RuleData :
47. **def** \_\_init\_\_(self,fuzzy,level) :
48. self.fuzzy = fuzzy
49. self.level = level
51. **class** Data :
52. **def** \_\_init\_\_(self,name,RangeDat):
53. self.name = name
54. self.RangeDat = RangeDat
56. **class** Graph :
58. **def** \_\_init\_\_(self,start,end) :
59. self.x\_start = start
60. self.x\_end = end
62. **def** getFuzzyleft(self,x) :
63. m = 1/(self.x\_start-self.x\_end)
64. c = -m\*self.x\_end
65. y = m\*x+c
66. **if**(y > 1) :
67. **return** 1
68. **elif**(y < 0):
69. **return** 0
70. **else**:
71. **return** y
73. **def** getFuzzymid(self,x):
74. center = self.x\_start + (self.x\_end - self.x\_start)/2
75. **if**(x < center) :
76. x\_end = center
77. m = -1/(self.x\_start-x\_end)
78. c = -m\*self.x\_start
79. y = m\*x+c
80. **if**(y > 1) :
81. **return** 1
82. **elif**(y < 0):
83. **return** 0
84. **else**:
85. **return** y
86. **else** :
87. x\_start = center
88. m = 1/(x\_start-self.x\_end)
89. c = -m\*self.x\_end
90. y = m\*x+c
91. **if**(y > 1) :
92. **return** 1
93. **elif**(y < 0):
94. **return** 0
95. **else**:
96. **return** y
98. **def** getFuzzyright(self,x) :
99. m = -1/(self.x\_start-self.x\_end)
100. c = -m\*self.x\_start
101. y = m\*x+c
102. **if**(y > 1) :
103. **return** 1
104. **elif**(y < 0):
105. **return** 0
106. **else**:
107. **return** y
108. **def** newRule(sizelvl,timelvl,templvl,donenesslvl):
109. ifRule = []
111. ifRule.append(RuleData(Fuzzy.SIZE,sizelvl))
112. ifRule.append(RuleData(Fuzzy.TIME,timelvl))
113. ifRule.append(RuleData(Fuzzy.TEMPURETURE,templvl))
115. thenRule = RuleData(Fuzzy.DONENESS,donenesslvl)
117. **return** Rule(ifRule,thenRule)
119. c = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b", "#e377c2", "#7f7f7f", "#bcbd22", "#17becf"]
121. **def** showSimulator(s,t,m,plot,maxinrule):
122. fig, axs = plt.subplots(1, 4, figsize=(15, 3), sharey=True)
123. axs[0].plot([0,40], [1,0],color=c[0])
124. axs[0].plot([30,65], [0,1],color=c[1])
125. axs[0].plot([65,100], [1,0],color=c[1])
126. axs[0].plot([90,130], [0,1],color=c[2])
127. axs[0].plot([0,s],[plot[0],plot[0]],color='black')
129. axs[1].plot([0,50], [1,0],color=c[0])
130. axs[1].plot([40,80], [0,1],color=c[1])
131. axs[1].plot([80,120], [1,0],color=c[1])
132. axs[1].plot([110,160], [0,1],color=c[2])
133. axs[1].plot([0,t],[plot[1],plot[1]],color='black')
135. axs[2].plot([0,30], [1,0],color=c[0])
136. axs[2].plot([25,35], [0,1],color=c[1])
137. axs[2].plot([35,45], [1,0],color=c[1])
138. axs[2].plot([40,50], [0,1],color=c[2])
139. axs[2].plot([50,60], [1,0],color=c[2])
140. axs[2].plot([55,65], [0,1],color=c[3])
141. axs[2].plot([65,75], [1,0],color=c[3])
142. axs[2].plot([70,100], [0,1],color=c[4])
143. axs[2].plot([0,m],[plot[2],plot[2]],color='black')
145. levelofoutput\_split = [[0,0.2],[0.15,0.25],[0.25,0.35],[0.3,0.475],[0.475,0.65],[0.6,0.7],[0.7,0.8],[0.75,1]]
146. levelofoutput= [[0,0.2],[0.15,0.35],[0.3,0.65],[0.6,0.8],[0.75,1]]
147. axs[3].plot(levelofoutput\_split[0], [1,0],color=c[0])
148. axs[3].plot(levelofoutput\_split[1], [0,1],color=c[1])
149. axs[3].plot(levelofoutput\_split[2], [1,0],color=c[1])
150. axs[3].plot(levelofoutput\_split[3], [0,1],color=c[2])
151. axs[3].plot(levelofoutput\_split[4], [1,0],color=c[2])
152. axs[3].plot(levelofoutput\_split[5], [0,1],color=c[3])
153. axs[3].plot(levelofoutput\_split[6], [1,0],color=c[3])
154. axs[3].plot(levelofoutput\_split[7], [0,1],color=c[4])
155. **for** idx,val **in** enumerate(maxinrule):
156. axs[3].plot(levelofoutput[idx],[val,val],color='black')
158. plt.show()
160. **def** showSimulator\_result(use\_x,defuz):
162. plt.plot([0,0.2], [1,0],color=c[0])
163. plt.plot([0.15,0.25], [0,1],color=c[1])
164. plt.plot([0.25,0.35], [1,0],color=c[1])
165. plt.plot([0.3,0.475], [0,1],color=c[2])
166. plt.plot([0.475,0.65], [1,0],color=c[2])
167. plt.plot([0.6,0.7], [0,1],color=c[3])
168. plt.plot([0.7,0.8], [1,0],color=c[3])
169. plt.plot([0.75,1], [0,1],color=c[4])
170. plt.plot([use\_x[0],use\_x[-1]],[defuz,defuz],color='black')
171. plt.show()
173. Size = {SIZELVL.SMALL : Graph(0,40),
174. SIZELVL.MEDIUM : Graph(30,100),
175. SIZELVL.BIG : Graph(90,130)}
177. Time = {TIMELVL.SHORT : Graph(0,50),
178. TIMELVL.MEDIUM : Graph(40,120),
179. TIMELVL.LONG : Graph(110,160)}
181. Tempureture = {TEMPLVL.MODERATE : Graph(0,30),
182. TEMPLVL.WARM : Graph(25,45),
183. TEMPLVL.VERY\_WARM : Graph(40,60),
184. TEMPLVL.HOT : Graph(55,75),
185. TEMPLVL.SCORCHING : Graph(70,100)}
187. Doneness = {DONELVL.RARE : Graph(0,0.2),
188. DONELVL.MEDIUM\_RARE : Graph(0.15,0.35),
189. DONELVL.MEDIUM : Graph(0.3,0.65),
190. DONELVL.MEDIUM\_WELL : Graph(0.6,0.8),
191. DONELVL.WELL\_DONE : Graph(0.75,1)}
193. Input = {Fuzzy.SIZE : Data(Size,range\_step(0,130,1)),
194. Fuzzy.TIME : Data(Time,range\_step(0,160,1)),
195. Fuzzy.TEMPURETURE : Data(Tempureture,range\_step(0,100,1))}
197. Output = {Fuzzy.DONENESS : Data(Doneness,range\_step(0,1,0.01))}
199. Rules = []
201. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.SHORT,TEMPLVL.MODERATE,DONELVL.RARE))
202. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.SHORT,TEMPLVL.WARM,DONELVL.MEDIUM\_RARE))
203. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.SHORT,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM\_RARE))
204. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.SHORT,TEMPLVL.HOT,DONELVL.MEDIUM))
205. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.SHORT,TEMPLVL.SCORCHING,DONELVL.MEDIUM))
206. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.MEDIUM,TEMPLVL.MODERATE,DONELVL.MEDIUM\_RARE))
207. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.MEDIUM,TEMPLVL.WARM,DONELVL.MEDIUM))
208. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.MEDIUM,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM))
209. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.MEDIUM,TEMPLVL.HOT,DONELVL.MEDIUM\_WELL))
210. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.MEDIUM,TEMPLVL.SCORCHING,DONELVL.MEDIUM\_WELL))
211. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.LONG,TEMPLVL.MODERATE,DONELVL.MEDIUM))
212. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.LONG,TEMPLVL.WARM,DONELVL.MEDIUM\_WELL))
213. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.LONG,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM\_WELL))
214. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.LONG,TEMPLVL.HOT,DONELVL.WELL\_DONE))
215. Rules.append(newRule(SIZELVL.SMALL,TIMELVL.LONG,TEMPLVL.SCORCHING,DONELVL.WELL\_DONE))
217. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.SHORT,TEMPLVL.MODERATE,DONELVL.RARE))
218. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.SHORT,TEMPLVL.WARM,DONELVL.RARE))
219. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.SHORT,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM\_RARE))
220. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.SHORT,TEMPLVL.HOT,DONELVL.MEDIUM\_RARE))
221. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.SHORT,TEMPLVL.SCORCHING,DONELVL.MEDIUM))
222. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.MEDIUM,TEMPLVL.MODERATE,DONELVL.MEDIUM\_RARE))
223. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.MEDIUM,TEMPLVL.WARM,DONELVL.MEDIUM\_RARE))
224. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.MEDIUM,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM))
225. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.MEDIUM,TEMPLVL.HOT,DONELVL.MEDIUM))
226. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.MEDIUM,TEMPLVL.SCORCHING,DONELVL.MEDIUM\_WELL))
227. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.LONG,TEMPLVL.MODERATE,DONELVL.MEDIUM))
228. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.LONG,TEMPLVL.WARM,DONELVL.MEDIUM))
229. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.LONG,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM\_WELL))
230. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.LONG,TEMPLVL.HOT,DONELVL.MEDIUM\_WELL))
231. Rules.append(newRule(SIZELVL.MEDIUM,TIMELVL.LONG,TEMPLVL.SCORCHING,DONELVL.WELL\_DONE))
233. Rules.append(newRule(SIZELVL.BIG,TIMELVL.SHORT,TEMPLVL.MODERATE,DONELVL.RARE))
234. Rules.append(newRule(SIZELVL.BIG,TIMELVL.SHORT,TEMPLVL.WARM,DONELVL.RARE))
235. Rules.append(newRule(SIZELVL.BIG,TIMELVL.SHORT,TEMPLVL.VERY\_WARM,DONELVL.RARE))
236. Rules.append(newRule(SIZELVL.BIG,TIMELVL.SHORT,TEMPLVL.HOT,DONELVL.MEDIUM\_RARE))
237. Rules.append(newRule(SIZELVL.BIG,TIMELVL.SHORT,TEMPLVL.SCORCHING,DONELVL.MEDIUM\_RARE))
238. Rules.append(newRule(SIZELVL.BIG,TIMELVL.MEDIUM,TEMPLVL.MODERATE,DONELVL.MEDIUM\_RARE))
239. Rules.append(newRule(SIZELVL.BIG,TIMELVL.MEDIUM,TEMPLVL.WARM,DONELVL.MEDIUM\_RARE))
240. Rules.append(newRule(SIZELVL.BIG,TIMELVL.MEDIUM,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM\_RARE))
241. Rules.append(newRule(SIZELVL.BIG,TIMELVL.MEDIUM,TEMPLVL.HOT,DONELVL.MEDIUM))
242. Rules.append(newRule(SIZELVL.BIG,TIMELVL.MEDIUM,TEMPLVL.SCORCHING,DONELVL.MEDIUM))
243. Rules.append(newRule(SIZELVL.BIG,TIMELVL.LONG,TEMPLVL.MODERATE,DONELVL.MEDIUM))
244. Rules.append(newRule(SIZELVL.BIG,TIMELVL.LONG,TEMPLVL.WARM,DONELVL.MEDIUM))
245. Rules.append(newRule(SIZELVL.BIG,TIMELVL.LONG,TEMPLVL.VERY\_WARM,DONELVL.MEDIUM))
246. Rules.append(newRule(SIZELVL.BIG,TIMELVL.LONG,TEMPLVL.HOT,DONELVL.MEDIUM\_WELL))
247. Rules.append(newRule(SIZELVL.BIG,TIMELVL.LONG,TEMPLVL.SCORCHING,DONELVL.MEDIUM\_WELL))
249. **class** SousVideFuzzyLogic :
250. **def** \_\_init\_\_(self,input\_data,output\_data,rules):
251. self.input\_data = input\_data
252. self.output\_data = output\_data
253. self.rules = rules
255. **def** defuzzifier(self,size,time,temp):
257. maxInRule = np.zeros(5)
258. plot = np.zeros(3)
260. **for** rule **in** self.rules:
261. minInRule = 1
262. **for** ruleData **in** rule.ifRule :
263. done = 0
264. **if**(ruleData.fuzzy == Fuzzy.SIZE):
265. **if**(ruleData.level == SIZELVL.SMALL):
266. done = (self.input\_data[Fuzzy.SIZE]).name[SIZELVL.SMALL].getFuzzyleft(size)
267. **if**(ruleData.level == SIZELVL.MEDIUM):
268. done = (self.input\_data[Fuzzy.SIZE]).name[SIZELVL.MEDIUM].getFuzzymid(size)
269. **if**(ruleData.level == SIZELVL.BIG):
270. done = (self.input\_data[Fuzzy.SIZE]).name[SIZELVL.BIG].getFuzzyright(size)
271. plot[0] = done
273. **elif**(ruleData.fuzzy == Fuzzy.TIME):
274. **if**(ruleData.level == TIMELVL.SHORT):
275. done = (self.input\_data[Fuzzy.TIME]).name[TIMELVL.SHORT].getFuzzyleft(time)
276. **if**(ruleData.level == TIMELVL.MEDIUM):
277. done = (self.input\_data[Fuzzy.TIME]).name[TIMELVL.MEDIUM].getFuzzymid(time)
278. **if**(ruleData.level == TIMELVL.LONG):
279. done = (self.input\_data[Fuzzy.TIME]).name[TIMELVL.LONG].getFuzzyright(time)
280. plot[1] = done
282. **elif**(ruleData.fuzzy == Fuzzy.TEMPURETURE):
283. **if**(ruleData.level == TEMPLVL.MODERATE):
284. done = (self.input\_data[Fuzzy.TEMPURETURE]).name[TEMPLVL.MODERATE].getFuzzyleft(temp)
285. **if**(ruleData.level == TEMPLVL.WARM):
286. done = (self.input\_data[Fuzzy.TEMPURETURE]).name[TEMPLVL.WARM].getFuzzymid(temp)
287. **if**(ruleData.level == TEMPLVL.VERY\_WARM):
288. done = (self.input\_data[Fuzzy.TEMPURETURE]).name[TEMPLVL.VERY\_WARM].getFuzzymid(temp)
289. **if**(ruleData.level == TEMPLVL.HOT):
290. done = (self.input\_data[Fuzzy.TEMPURETURE]).name[TEMPLVL.HOT].getFuzzymid(temp)
291. **if**(ruleData.level == TEMPLVL.SCORCHING):
292. done = (self.input\_data[Fuzzy.TEMPURETURE]).name[TEMPLVL.SCORCHING].getFuzzyright(temp)
293. plot[2] = done
295. **if**(minInRule > done **and** done >= 0):
296. minInRule = done
298. **if**(rule.thenRule.level == DONELVL.RARE):
299. **if**(maxInRule[0] < minInRule):
300. maxInRule[0] = minInRule
301. **elif**(rule.thenRule.level == DONELVL.MEDIUM\_RARE):
302. **if**(maxInRule[1] < minInRule):
303. maxInRule[1] = minInRule
304. **elif**(rule.thenRule.level == DONELVL.MEDIUM):
305. **if**(maxInRule[2] < minInRule):
306. maxInRule[2] = minInRule
307. **elif**(rule.thenRule.level == DONELVL.MEDIUM\_WELL):
308. **if**(maxInRule[3] < minInRule):
309. maxInRule[3] = minInRule
310. **elif**(rule.thenRule.level == DONELVL.WELL\_DONE):
311. **if**(maxInRule[4] < minInRule):
312. maxInRule[4] = minInRule
314. *#             showSimulator(size,time,temp,plot,maxInRule)*
316. start\_out = self.output\_data[Fuzzy.DONENESS].RangeDat.start
317. end\_out = self.output\_data[Fuzzy.DONENESS].RangeDat.end
318. step\_out = self.output\_data[Fuzzy.DONENESS].RangeDat.step
320. start = []
321. end =[]
322. **for** i **in** DONELVL :
323. start.append(self.output\_data[Fuzzy.DONENESS].name[i].x\_start)
324. end.append(self.output\_data[Fuzzy.DONENESS].name[i].x\_end)
326. defuzzi = np.zeros(2)
327. use\_x = []
328. **for** x **in** range(start\_out\*100,end\_out\*100,int(step\_out\*100)):
330. x = x/100
331. y = np.zeros(5)
333. **if**(x <= end[0]):
334. y[0] = self.output\_data[Fuzzy.DONENESS].name[DONELVL.RARE].getFuzzyleft(x)
335. **if**(y[0] > maxInRule[0]):
336. y[0] = maxInRule[0]
337. **if**(x >= start[1] **and** x <= end[1]):
338. y[0] = self.output\_data[Fuzzy.DONENESS].name[DONELVL.MEDIUM\_RARE].getFuzzymid(x)
339. **if**(y[1] > maxInRule[1]):
340. y[1] = maxInRule[1]
341. **if**(x >= start[2] **and** x <= end[2]):
342. y[2] = self.output\_data[Fuzzy.DONENESS].name[DONELVL.MEDIUM].getFuzzymid(x)
343. **if**(y[2] > maxInRule[2]):
344. y[2] = maxInRule[2]
345. **if**(x >= start[3] **and** x <= end[3]):
346. y[3] = self.output\_data[Fuzzy.DONENESS].name[DONELVL.MEDIUM\_WELL].getFuzzymid(x)
347. **if**(y[3] > maxInRule[3]):
348. y[3] = maxInRule[3]
349. **if**(x >= start[4]):
350. y[4] = self.output\_data[Fuzzy.DONENESS].name[DONELVL.WELL\_DONE].getFuzzyright(x)
351. **if**(y[4] > maxInRule[4]):
352. y[4] = maxInRule[4]
354. maximum = np.max(y)
355. **if**(maximum > 0):
356. use\_x.append([x,maximum])
358. defuzzi[0] += maximum\*x
359. defuzzi[1] += maximum
361. **if**(defuzzi[1] == 0):
362. defuzzi[1] = 1
363. *#         print(use\_x)*
364. *#         showSimulator\_result(use\_x,defuzzi[0]/defuzzi[1])*
366. **return** defuzzi[0]/defuzzi[1]
368. souvidefuzzy = SousVideFuzzyLogic(Input,Output,Rules)
369. souvidefuzzy.defuzzifier(130,39,49)