ECE 449 - Intelligent Systems Engineering

Lab 3-D41: Fuzzy Logic Concepts

Lab date: *Thursday, October 21, 2021 -- 2:00 - 4:50 PM*
Room: *ETLC E5-013*
Lab report due: *Wednesday, November 3, 2021 -- 3:50 PM*

---

1. Objectives

The objectives of this lab are to become familiar with the basic concepts of fuzzy logic. These concepts include:

- defining membership functions and modifying them with linguistic terms
- performing various operations on fuzzy sets
- representing fuzzy sets using α-cuts
- constructing fuzzy relations, projections, and cylindrical extensions
- performing composition and using it in compositional rules of inference

2. Expectations

Complete the pre-lab, and hand it in before the lab starts. A formal lab report is required for this lab, which will be the completed version of this notebook. There is a marking guide at the end of the lab manual. If figures are required, label the axes and provide a legend when appropriate. An abstract, introduction, and conclusion are required as well, for which cells are provided at the end of the notebook. The abstract should be a brief description of the topic, the introduction a description of the goals of the lab, and the conclusion a summary of what you learned, what you found difficult, and your own ideas and observations.

3. Pre-lab

1. Why is defuzzification an important step when using fuzzy sets?

We also strongly recommend that you look over section 1 of the Python supplement to familiarize yourself with Jupyter notebooks and install the necessary libraries for future labs.

4. Introduction

Fuzzy logic is a form of logic in which the truth values of variables can range from the interval of 0 to 1, instead of exclusively 0 or 1. This can be used to solve problems in a more human-like reasoning way by allowing gradual membership in sets. These fuzzy sets form inputs and outputs to linguistic relations that can be easily constructed, such as:

IF *temp* **IS** HOT **THEN** *fan* **IS** HIGH

An important concept is the *linguistic variable*, which is a variable whose values are words. In the example above, the *linguistic variable* "temp" takes the value "HOT".
Employing fuzzy systems requires the user to first define membership functions that take values from 0 to 1, and are defined over the region of interest, called the *universe set*. One can apply linguistic modifiers (*hedges*) to modify the meaning of a fuzzy set, such as:

temp **IS** VERY HOT, rather than *temp* **IS** HOT

A *hedge* in this case is VERY.
Similar to crisp sets, the *union*, *intersection*, and *complement* operators can be performed on fuzzy sets. They may also be represented using a family of crisp sets, by using α*-cuts*.
Finally, similarly to crisp relations, which is the mapping between two crisp sets, fuzzy sets can form *relations* between two membership functions of different universes. These relations bring forth more operations, such as *cylindrical closure*, *sup-min composition*, *compositional rule of inference*, and *defuzzification*. *Cylindrical closure* is a fuzzy relation that corresponds to the cross-product domain of linguistic variables.

5. Background

Automatic monitoring stations are used to characterize the quality of the environment in the Arctic by collecting meteorological data at regular intervals. Because of how remote these locations are, the monitoring stations are designed to generate and store power from renewable resources, namely the sun and wind, to minimize the frequency of maintenance required. However, due to the polar nights and long winters, solar radiation reaching the ground during these times is very low or non-existent. Consequently, this can lead to long intervals during which there is no remaining power, and no data is collected. To avoid this, the duty cycle of the monitoring station can be adjusted in order to conserve power. A controller to determine the optimal duty cycle can be built using fuzzy logic based on two factors: *state of charge* (SOC) of the battery and *future average power* (P) from the renewable resources. For example, one such rule could be as follows:

IF *state of charge* **IS** LOW **AND** *future average power* **IS** MEDIUM **THEN** *duty cycle* ***IS*** *MEDIUM*

In the case where this rule would apply, the monitoring station could only take measurements for around half of its regular period to conserve power, and obtain data more frequently than what the previous method would offer. The next two labs will focus on this concept and work towards building a fuzzy controller to manage the power consumption of a monitoring station.

6. Experimental Procedure

If you have not yet installed the skfuzzy library, run the cell below.

In [37]:
```
%%bash
# "--user" is essential to install in local environment"
pip install --user -U scikit-fuzzy
```

```
Requirement already satisfied: scikit-fuzzy in /home/jupyter/.local/lib/python3.9/site-packages (0.4.2)
Requirement already satisfied: networkx>=1.9.0 in /opt/conda/lib/python3.9/site-packages (from scikit-fuzzy) (2.
3)
Requirement already satisfied: scipy>=0.9.0 in /opt/conda/lib/python3.9/site-packages (from scikit-fuzzy) (1.7.1)
Requirement already satisfied: numpy>=1.6.0 in /opt/conda/lib/python3.9/site-packages (from scikit-fuzzy) (1.21.
2)
Requirement already satisfied: decorator>=4.3.0 in /opt/conda/lib/python3.9/site-packages (from networkx>=1.9.0->
scikit-fuzzy) (5.0.9)
```

Run the cell below to import the libraries required to complete this lab.

In [38]:
```
%matplotlib inline

import numpy as np                              # General math operations
import matplotlib.pyplot as plt                 # Data visualization
from mpl_toolkits.mplot3d import Axes3D         # 3D data visualization
import skfuzzy as fuzz                          # Fuzzy toolbox
```

Exercise 1: Membership functions
Consider a weather station with a battery that has a minimum SOC of 20% and a maximum SOC of 100%.

1. Define the universe set for SOC from 20 to 100, using 81 discrete elements.

In [39]:
```
universe_set = np.arange(20,101)
print(universe_set)
```

```
[ 20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73
  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91
  92  93  94  95  96  97  98  99 100]
```

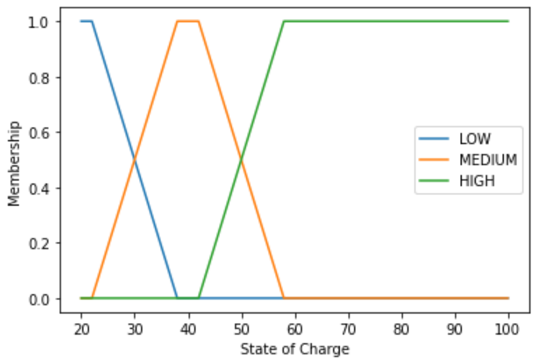
2. Plot the trapezoidal membership functions, LOW, MEDIUM, and HIGH, on one figure according to the parameters given below.

Fuzzy set	*a*	*b*	*c*	*d*
LOW	20	20	22	38
MEDIUM	22	38	42	58
HIGH	42	58	100	100

In [40]:
```
low = [20,20,22,38]
medium = [22,38,42,58]
high = [42,58,100,100]
y_low = fuzz.trapmf(universe_set,low)
y_medium = fuzz.trapmf(universe_set,medium)
y_high = fuzz.trapmf(universe_set,high)

plt.plot(universe_set,y_low, label = "LOW")
plt.plot(universe_set,y_medium, label = "MEDIUM")
plt.plot(universe_set,y_high, label = "HIGH")
plt.xlabel("State of Charge")
plt.ylabel("Membership")
plt.legend()
```

Out[40]: <matplotlib.legend.Legend at 0x7fc2777d2370>



Exercise 2: Linguistic modifiers

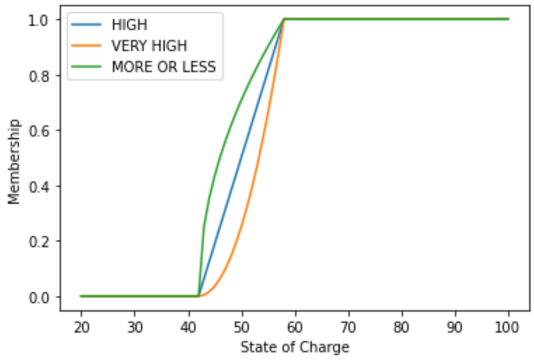
Modify the fuzzy set HIGH SOC to VERY HIGH SOC and MORE OR LESS HIGH SOC.

1. Plot HIGH, VERY HIGH, and MORE OR LESS HIGH on the same figure.

In [41]:
```
y_very_high = y_high**2
y_more_or_less_high = y_high**0.5

plt.plot(universe_set,y_high, label = "HIGH")
plt.plot(universe_set, y_very_high, label = "VERY HIGH")
plt.plot(universe_set, y_more_or_less_high, label = "MORE OR LESS")
plt.ylabel("Membership")
plt.xlabel("State of Charge")
plt.legend()
```

Out[41]: `<matplotlib.legend.Legend at 0x7fc2776eb8e0>`

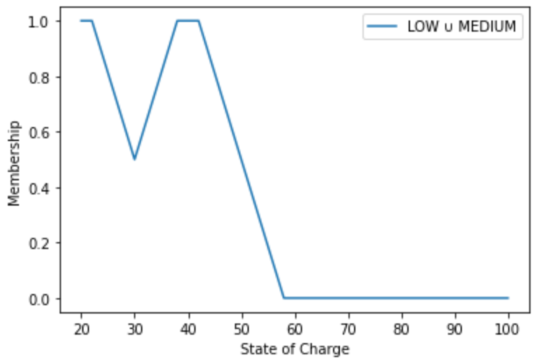


Exercise 3: Fuzzy set operations
On separate figures, plot the following fuzzy sets:

1. Union of LOW and MEDIUM

In [42]:
```
_, low_union_medium = fuzz.fuzzy_or(universe_set, y_low, universe_set, y_medium);
plt.plot(universe_set, low_union_medium, label = "LOW ∪ MEDIUM")
plt.xlabel("State of Charge")
plt.ylabel("Membership")
plt.legend()
```

Out[42]: `<matplotlib.legend.Legend at 0x7fc2778c5130>`

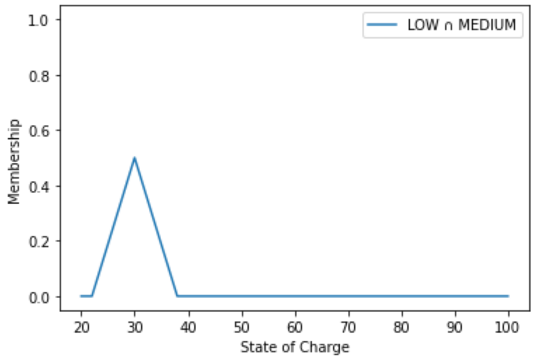


2. Intersection of LOW and MEDIUM

In [43]:

```
_, low_intersect_medium = fuzz.fuzzy_and(universe_set, y_low, universe_set, y_medium);
plt.plot(universe_set, low_intersect_medium, label = "LOW ∩ MEDIUM")
plt.xlabel("State of Charge")
plt.ylabel("Membership")
ax = plt.gca()
# Set the ylim of y-axis to 1 to keep it normalized
ax.set_ylim([-0.05, 1.05])
plt.legend()
```

Out[43]: <matplotlib.legend.Legend at 0x7fc277882a90>

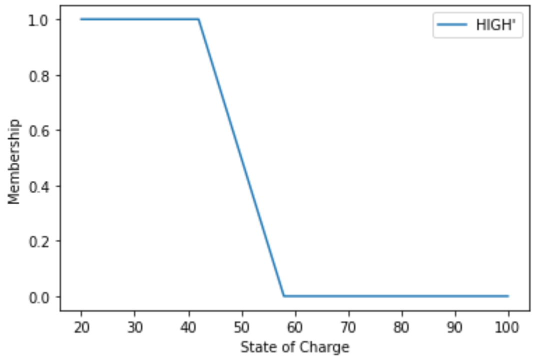


3. Complement of HIGH

In [44]:

```
not_high = fuzz.fuzzy_not(y_high);
plt.plot(universe_set, not_high, label = "HIGH'")
plt.xlabel("State of Charge")
plt.ylabel("Membership")
ax = plt.gca()
# Set the ylim of y-axis to 1.0 to keep it normalized
ax.set_ylim([-0.05, 1.05])
plt.legend()
```

Out[44]: <matplotlib.legend.Legend at 0x7fc27793d340>



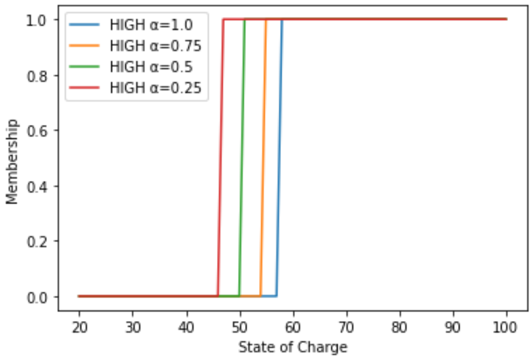
Exercise 4: α**-cuts**
Using the HIGH SOC fuzzy set,

1. Plot the individual α-cuts for α = {1.0, 0.75, 0.50, 0.25} on the same figure.

In [45]:

```
alpha_cuts = [1.0,0.75,0.50,0.25]
for alphas in alpha_cuts:
    plt.plot(universe_set, fuzz.defuzzify.lambda_cut(y_high, alphas), label="HIGH α={}".format(alphas))

#plt.plot(universe_set, y_high)
plt.xlabel("State of Charge")
plt.ylabel("Membership")
plt.legend();
```



2. Plot the original fuzzy set and its α-cut reconstruction on the same figure.

HINT: The **np.amax()** function is helpful in reconstructing the fuzzy set.

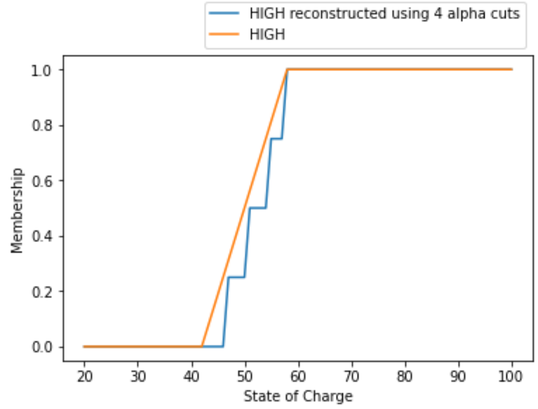
In [46]:

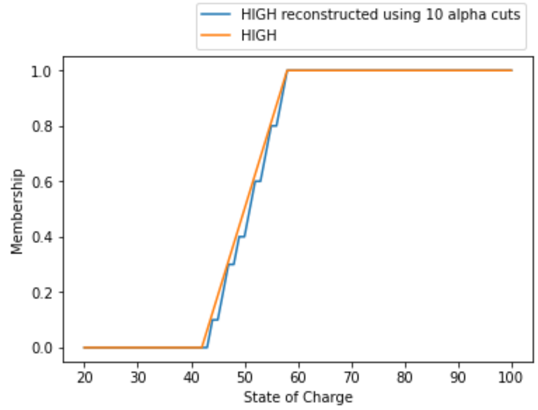
```
def alpha_reconstruction(alpha_cuts):
    matrix = [[]]
    # Find all alpha cut membership functions
    for i in range(len(alpha_cuts)):
        matrix[0].append(fuzz.defuzzify.lambda_cut(y_high, alpha_cuts[i]))
        # Multiply the membership values by alpha.
        matrix[0][i] = matrix[0][i] * alpha_cuts[i]

    reconstructed_plot = matrix[0][0]
    # Apply OR to all the alpha cuts membership functions (which is basically max)
    for i in range(len(matrix[0])):
       _,reconstructed_plot = fuzz.fuzzy_or(universe_set, reconstructed_plot, universe_set, matrix[0][i])

    plt.plot(universe_set, reconstructed_plot, label = "HIGH reconstructed using {} alpha cuts".format(len(alpha_c
    plt.plot(universe_set, y_high, label = "HIGH")
    plt.xlabel("State of Charge")
    plt.ylabel("Membership")
    plt.legend(bbox_to_anchor=(1.0, 1.2))
    plt.show()

alpha_reconstruction(alpha_cuts)
alpha_reconstruction([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0])
```





3. Comment on the quality of the α-cut reconstruction.

For the given 4 alpha cuts α = {1.0, 0.75, 0.50, 0.25}, we can see that the reconstructed HIGH membership function has lost some precision as seen from the first graph. However if we use a higher number of alpha cuts, for example, α = {0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0} we would obtain a higher precision in terms of the reconstructed HIGH membership function as seen from the second graph.

Exercise 5: Relations - Cylindrical closure

Based off of typical meteorological data, the locations in which the monitoring stations are situated can only provide future average power from 0W to 100W.

1. Define the universe set for future average power from 0 to 100, using 101 discrete elements.

In [47]:
```
universe_set2 = np.arange(0,101)
print(universe_set2)
```

```
[  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
  90  91  92  93  94  95  96  97  98  99 100]
```

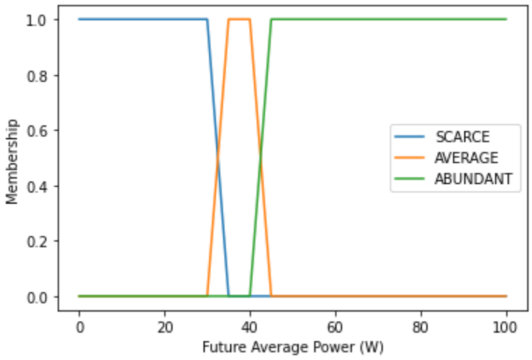
2. Plot the trapezoidal membership functions, SCARCE, AVERAGE, and ABUNDANT in one figure, according to the parameters given below.

$Fuzzy\ set$	a	b	c	d
$SCARCE$	0	0	30	35
$AVERAGE$	30	35	40	45
$ABUNDANT$	40	45	100	100

In [48]:
```
scarce = [0,0,30,35]
average = [30,35,40,45]
abundant = [40,45,100,100]
y_scarce = fuzz.trapmf(universe_set2,scarce)
y_average = fuzz.trapmf(universe_set2,average)
y_abundant = fuzz.trapmf(universe_set2,abundant)

plt.plot(universe_set2,y_scarce, label = "SCARCE")
plt.plot(universe_set2,y_average, label = "AVERAGE")
plt.plot(universe_set2,y_abundant, label = "ABUNDANT")
plt.xlabel("Future Average Power (W)")
plt.ylabel("Membership")
plt.legend()
```

Out[48]: `<matplotlib.legend.Legend at 0x7fc27762fee0>`

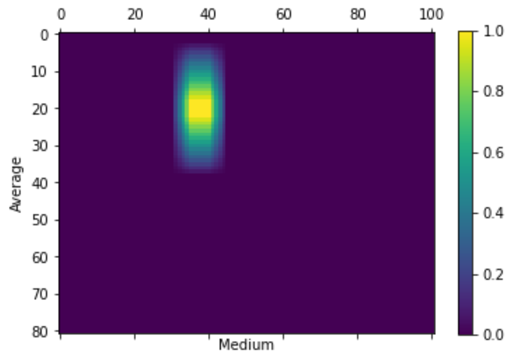


3. Using *Larsen implication*, define the relation $R(MEDIUM,\ AVERAGE)$. Plot the relation matrix.

In [49]:

```python
# Larsen Implication uses product operator for implication
res = fuzz.relation_product(y_medium,y_average)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(res)
fig.colorbar(cax)
plt.xlabel("Medium")
plt.ylabel("Average")
print(res[0])
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.]
```



4. What is the meaning of the individual rows of the relation matrix? What does the first row mean?

```
Each row in the matrix represents the extent of how related future power in the AVERAGE fuzzy set to a single value
battery level in the MEDIUM fuzzy set. The first row is filled with zeros, it means that there is no relation
between all values of the AVERAGE power fuzzy set and when the MEDIUM battery level is zero.
```

Exercise 6: Sup-min composition

Three monitoring stations positioned at different locations are checked, accordingly, the membership values of SOC fuzzy sets are assigned. The findings are expressed as a relation, $LocationSOC(location, state\ of\ charge)$, and defined using the following matrix:

$$LocationSOC = \begin{bmatrix} 0.84 & 0.08 & 0 \\ 0.03 & 0.5 & 0.08 \\ 0 & 0.1 & 0.8 \end{bmatrix}$$

Each row in the matrix corresponds to a monitoring station, and the columns give the membership values in the fuzzy sets, LOW, MEDIUM, and HIGH, respectively. For example, the last monitoring station has a 0 LOW, 0.1 MEDIUM, and a 0.8 HIGH SOC.
Additionally, it was determined how the SOC of the monitoring station corresponds to the future average power of its location. This is represented by the relation, $SOCPower(state\ of\ charge, future\ average\ power)$, found below.

$$SOCPower = \begin{bmatrix} 1 & 0.3 & 0 \\ 0.2 & 0.5 & 0.3 \\ 0 & 0.5 & 1 \end{bmatrix}$$

1. Determine the max-min composition $c_1 = LocationSOC \circ SOCPower$ and $c_2 = SOCPower^T \circ LocationSOC^T$ and print the resulting matrices.

In [50]:
```python
location_soc = np.array([
    [0.84, 0.08, 0],
    [0.03, 0.5, 0.08],
    [0, 0.1, 0.8]]);

soc_power = np.array([
    [1, 0.3, 0],
    [0.2, 0.5, 0.3],
    [0, 0.5, 1]
]);

c1 = fuzz.maxmin_composition(location_soc, soc_power)
c2 = fuzz.maxmin_composition(np.transpose(soc_power), np.transpose(location_soc))

print("LocationSOC ∘ SOCPower:")
print(c1)
print("SOCPower' ∘ LocationSOC' :")
print(c2)

print("Is c1ᵀ = c2? {}".format(np.array_equal(np.transpose(c1), c2)))
print("Is c2ᵀ = c1? {}".format(np.array_equal(np.transpose(c2), c1)))
```

```
LocationSOC ∘ SOCPower:
[[0.84 0.3  0.08]
 [0.2  0.5  0.3 ]
 [0.1  0.5  0.8 ]]
SOCPower' ∘ LocationSOC' :
[[0.84 0.2  0.1 ]
 [0.3  0.5  0.5 ]
 [0.08 0.3  0.8 ]]
Is c1ᵀ = c2? True
Is c2ᵀ = c1? True
```

2. How can you interpret these relations?

```
Each column in the relation corresponds to the future average power membership and each row in the relation corresponds to a monitoring station. The values obtained inside the matrix after composition indicate the degree of relation between the two (how related they are). The results obtained from the max-min composition c1 and c2 show an interesting result. c1 transposed is equivalent to c2 and vice versa.
```

Exercise 7: Compositional rule of inference
Another monitoring station was checked and found to have a SOC of 28%. Use a compositional rule of inference to determine the future average power fuzzy set based on the knowledge of a monitoring station with LOW SOC in a location with SCARCE future average power.

1. Express the item as a fuzzy singleton on the SOC universe set.

In [51]:
```python
universe_set3 = np.zeros(81)
# 28-20 since array starts at 20% we want 28% so 8th index would be 28
universe_set3[28-20] = 1
print(universe_set3)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

2. Use *Mamdani implication* to define the relation between LOW and SCARCE.

In [52]:
```python
relation_low_scarce = fuzz.relation_min(y_low,y_scarce)
print(relation_low_scarce)
```

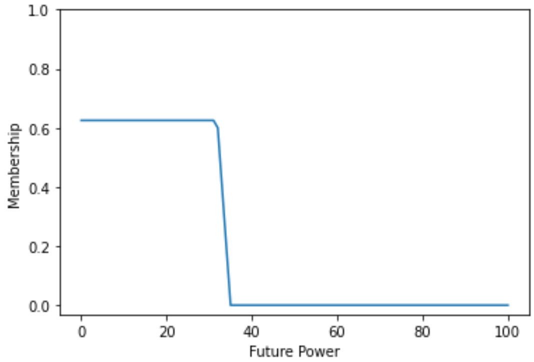
```
[[1. 1. 1. ... 0. 0. 0.]
 [1. 1. 1. ... 0. 0. 0.]
 [1. 1. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

3. Use the relation from exercise 9 to derive the associated fuzzy set. Print this fuzzy set as a vector.

In [63]:

```
fuzzy_set = fuzz.maxprod_composition(universe_set3, relation_low_scarce);
fuzzy_set = np.transpose(fuzzy_set)
print(np.transpose(fuzzy_set))
plt.plot(np.arange(0,101),fuzzy_set);
plt.ylim(top=1);
plt.xlabel("Future Power")
plt.ylabel("Membership")
plt.show()
```

```
[[0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625
  0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625
  0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.625 0.6   0.4   0.2   0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    ]]
```



Exercise 8: Defuzzification
Determine the crisp value of the fuzzy set obtained from the compositional rule of inference applied in the previous exercise. Use the Mean of Maxima (MOM) defuzzification method.

1. Print the resulting future average power of the location.

In [66]:

```
defuzz_val = fuzz.defuzz(np.arange(0,101), fuzzy_set, 'mom');
print("Future average power of the location = {}".format(defuzz_val))
```

```
Future average power of the location = 15.5
```

Abstract

The purpose of this lab was to get familiarized with the basics of fuzzy sets including the (binary/unary) operations that can be performed on them and how they can form relations between two membership functions of different universes. The case provided to us was to help a station conserve power during polar nights and long winters. By providing a controller that is built using state of charge (SOC) of the battery and future average power (P), we can use it to determine the optimal duty cycle. We began by making plotting the trapezoidal membership functions of the LOW, MEDIUM and HIGH fuzzy sets. Afterwards, we applied linguistic hedges "VERY" and "MORE OR LESS" to the HIGH membership function and plotted it. From this, we saw that the "VERY" linguistic hedge resulted in concentration of the membership function while the "MORE OR LESS" linguistic hedge resulted in dilation of the membership function. Then we performed intersection and union operations on the LOW and MEDIUM fuzzy sets and plotted our results and performed the complement operation on the HIGH membership function and plotted all of the results. Later, we plotted the individual alpha cuts 1.0, 0.75, 0.50 and 0.25 of the HIGH membership function, and then proceeded to reconstruct the HIGH membership function using these alpha cuts. The result was imprecise when compared to the original HIGH membership function, however, if we use more alpha cuts then we would obtain a more accurate reconstruction. Afterwards, we plotted the SCARCE, AVERAGE and ABUNDANT trapezoidal membership functions and proceed to find a relation between the MEDIUM and AVERAGE fuzzy sets. This relation matrix was plotted. Given the SOCPower and LocationSOC matrices, we performed max-min composition twice, once on the original matrices and once on the transpose of the matrices. Using Mamdami implication, we found the relation between the LOW and SCARCE and then proceeded to find the associated fuzzy set. Finally, we found the crisp value of the fuzzy set by using defuzzification via the mean of maxima method and obtained 15.5 for the future average power of the location.

Introduction

In this lab we explored the basic concepts of fuzzy logic by utilizing the skfuzzy library to solve a case. Fuzzy logic is the approach where there is multiple "degrees of truth (fulfilment)" rather than the Boolean Logic which is just 1 or 0 (True or False). This is helpful for universes of discourse where their outcomes cannot be translated to a 0 or 1 but instead require a value ranging from 0 to 1 representing the degree of fulfilment. In this case, a controller is to be built to help a station conserve power during polar nights and long winters. This controller's design is going to be based on future average power (P) and state of charge (C) to determine the optimal duty cycle. We plotted the HIGH, MEDIUM and LOW membership functions (fuzzy sets) for the state of charge (SOC) variable and performed operations on them. Afterwards, the SCARCE, AVERAGE and ABUNDANT membership functions of the future average power (P) variable were plotted and operations were performed on them. Finally, by utilizing defuzzification via the Mean of Maxima method, the future avearge power crisp value of the location will be found.

Conclusion

In conclusion this lab served as a great introduction to putting our fuzzy logic knowledge to practical use by utilizing the skfuzzy library to perform operations. We started by plotting the LOW, MEDIUM and HIGH trapezoidal membership functions (fuzzy sets) for the state of charge (SOC) variable and performed operations on them. Firstly, we applied linguistic hedges "VERY" and "MORE OR LESS" to the HIGH membership function. As expected, the "VERY" linguistic hedge lead to a concentration (concavity) of the membership function and the "MORE OR LESS" linguistic hedge lead to a dilation (convexity) of the membership function. Then we used the binary operations intersection and union on the LOW and MEDIUM membership functions. As expected the union of these two was the line representing their maximum(MEDIUM(x),LOW(x)) at each input value (overlap) and the intersection was the line representing thier minimum(MEDIUM(x),LOW(x)) (underlap). The complement of the HIGH membership function was also found and plotted and as expected, the result was 1-HIGH(x) at each input value (state of charge). Then we reconstructed the HIGH membership function after finding alpha cuts at levels 1.0, 0.75, 0.50 and 0.25. The reconstruction was imprecise when compared to the original HIGH membership function, however, if we use more alpha cuts we can obtain a more accurate reconstruction. Then, we plotted the SCARCE, AVERAGE and ABUNDANT trapezoidal membership functions. Afterwards, we found a relation between the MEDIUM and AVERAGE fuzzy sets and plotted the relation matrix. The two were related more in values of 30-40 in the MEDIUM fuzzy set and 5-35 in the AVERAGE fuzzy set. Using the provided SOCPower and LocationSOC matrices we found the max-min composition twice, once on the original matrices and once on the transpose of both of them. Looking at the results, we could see that the resulting matrices were the transpose of each other. By utilizing Mamdami implication, we found the relation between the LOW and SCARCE fuzzy sets and the associated fuzzy set. The last step of this lab was to use defuzzification with the Mean of Maxima method to obtain the crisp value of the future average power of the location and we obtained 15.5.

Lab 1 Marking Guide

Exercise	Item	Total Marks	Earned Marks
	$Pre-lab$	10	
	$Abstract$	3	
	$Introduction$	3	
	$Conclusion$	4	
1	$Membership\ functions$	10	
2	$Linguistic\ modifiers$	5	
3	$Fuzzy\ operations$	10	
4	$Alpha\ cuts$	10	
5	$Fuzzy\ relations$	15	
6	$Sup-min\ composition$	15	
7	CRI	5	
8	$Defuzzification$	10	
	TOTAL	100	