air_conditioner_controller

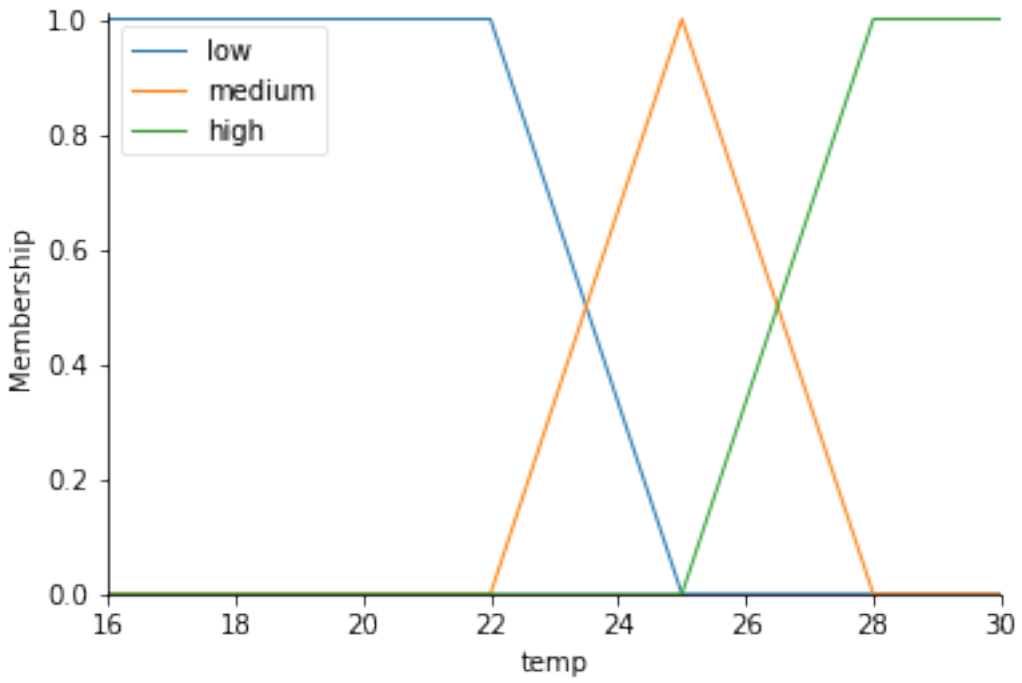
April 30, 2022

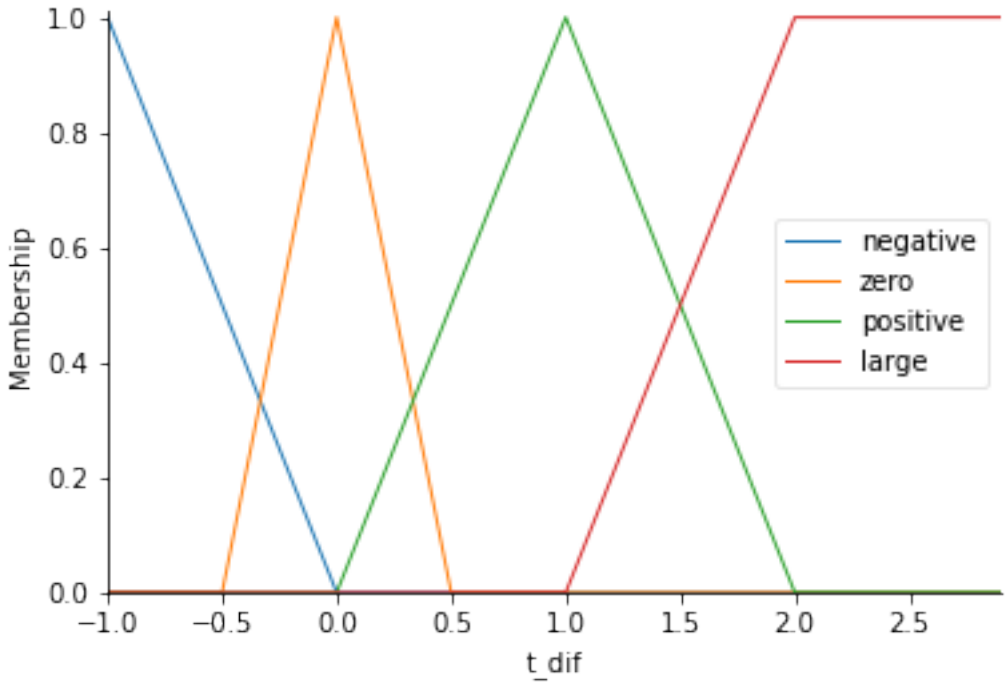
[1]:
```
import warnings
warnings.filterwarnings("ignore")

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

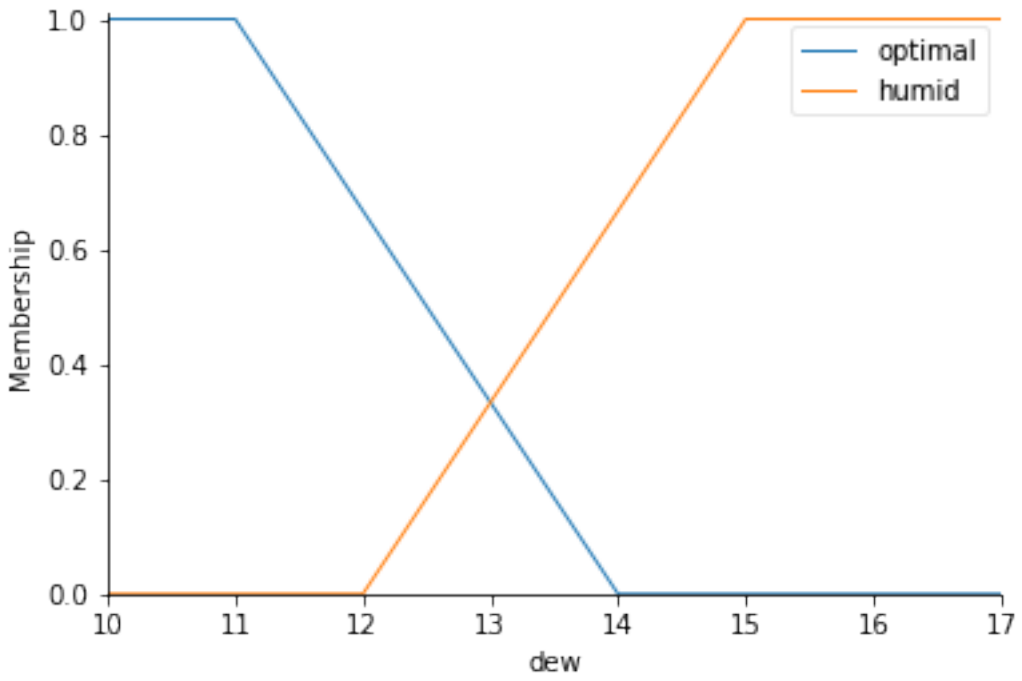
[2]:
```
#Setting Input Attributes
temp = ctrl.Antecedent(np.arange(16,31,1),'temp')
t_dif = ctrl.Antecedent(np.arange(-1,3,0.1),'t_dif')
d_point = ctrl.Antecedent(np.arange(10,18,1),'dew')
e_volt = ctrl.Antecedent(np.arange(130,220,10),'ev')
```

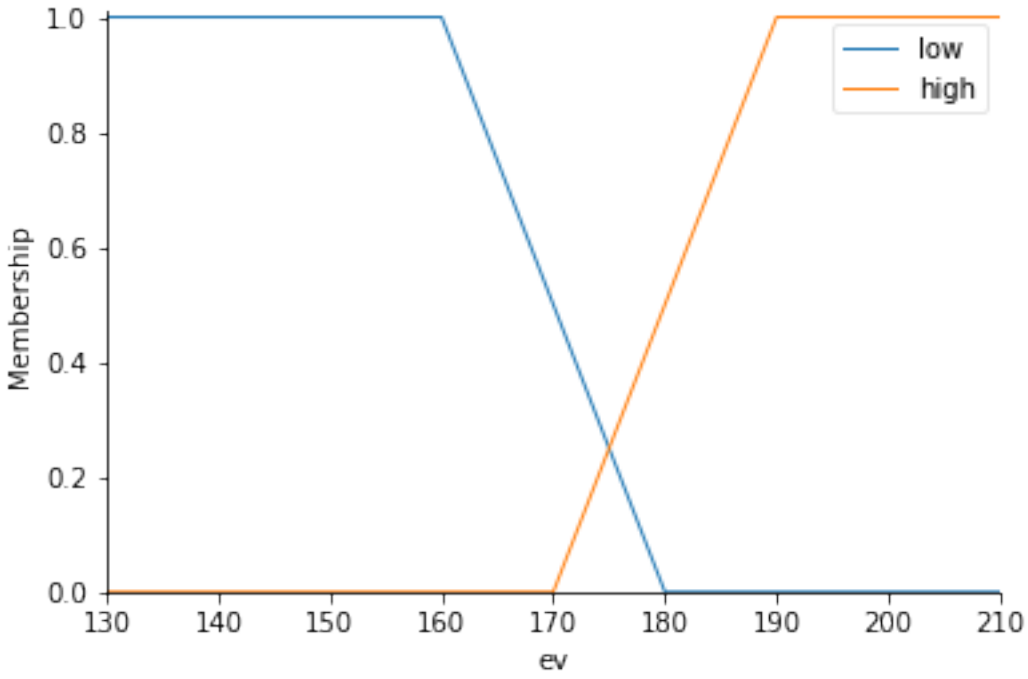
[3]:
```
#Setting Input Membership Functions
#Membership Function values for temperature
temp['low'] = fuzzy.trapmf(temp.universe,[16,16,22,25])
temp['medium'] = fuzzy.trimf(temp.universe,[22,25,28])
temp['high'] = fuzzy.trapmf(temp.universe,[25,28,30,30])
#Membership Function values for temperature difference
t_dif['negative'] = fuzzy.trimf(t_dif.universe,[-1,-1,0])
t_dif['zero'] = fuzzy.trimf(t_dif.universe,[-0.5,0,0.5])
t_dif['positive'] = fuzzy.trimf(t_dif.universe,[0,1,2])
t_dif['large'] = fuzzy.trapmf(t_dif.universe,[1,2,3,3])
#Membership Function values for Dew Point
d_point['optimal'] = fuzzy.trapmf(d_point.universe,[10,10,11,14])
d_point['humid'] = fuzzy.trapmf(d_point.universe,[12,15,18,18])
#Membership Function values for Electric Volt
e_volt['low'] = fuzzy.trapmf(e_volt.universe,[130,130,160,180])
e_volt['high'] = fuzzy.trapmf(e_volt.universe,[170,190,220,220])
```

[4]:
```
temp.view()
t_dif.view()
d_point.view()
e_volt.view()
```


low
medium
high
Membership
temp
1.0
0.8
0.6
0.4
0.2
0.0
16
18
20
22
24
26
28
30


negative
zero
positive
large
Membership
t_dif
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

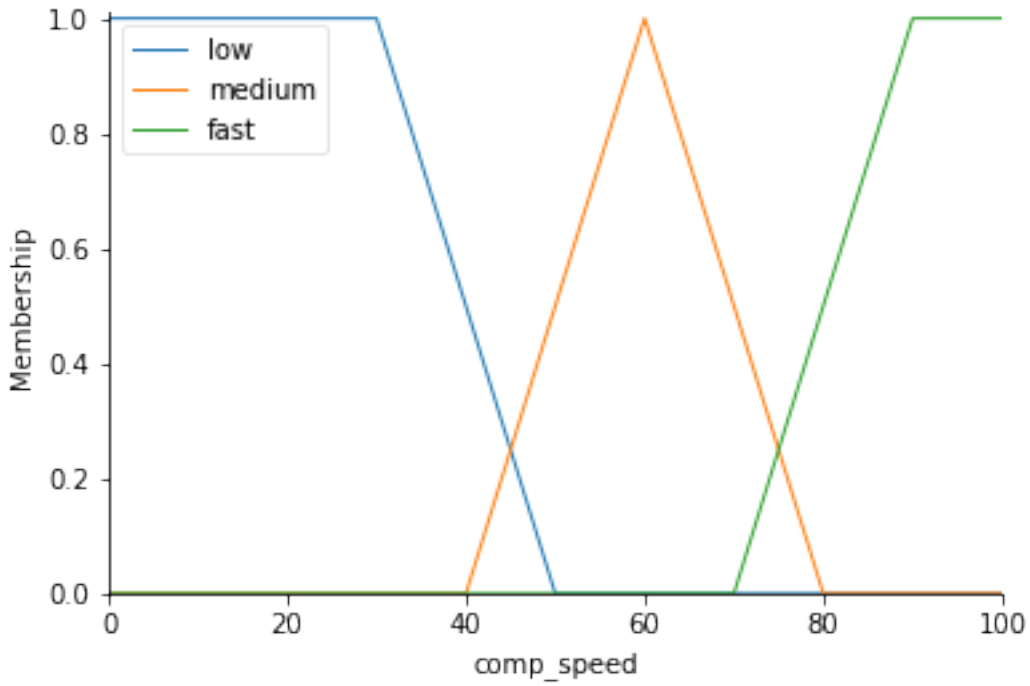

optimal
humid
Membership
dew
1.0
0.8
0.6
0.4
0.2
0.0
10
11
12
13
14
15
16
17

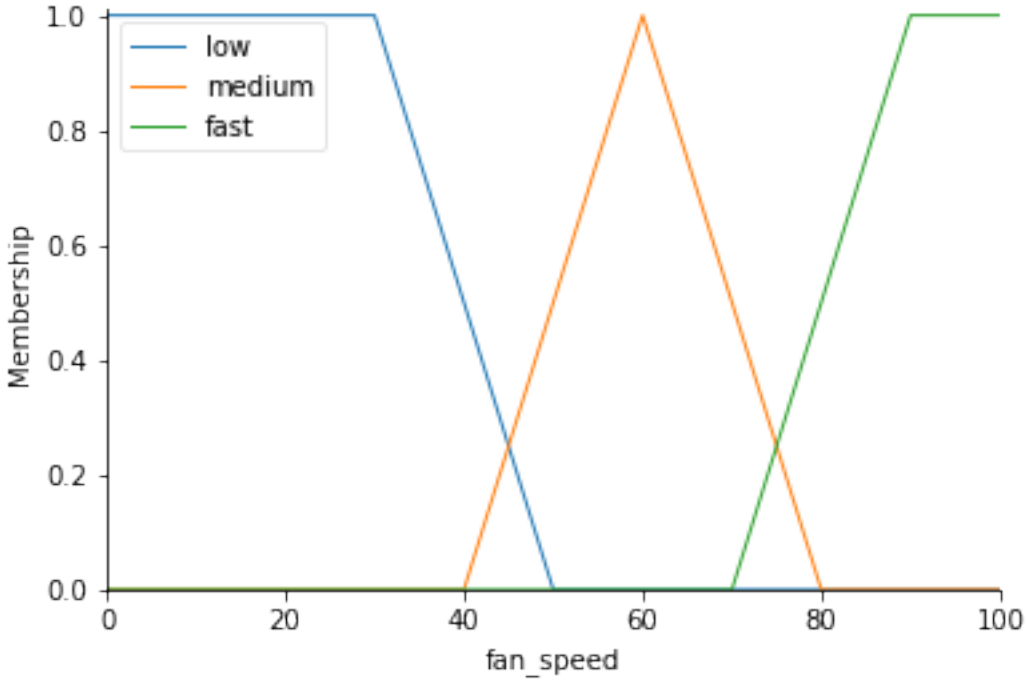

low
high
Membership
ev
1.0
0.8
0.6
0.4
0.2
0.0
130
140
150
160
170
180
190
200
210

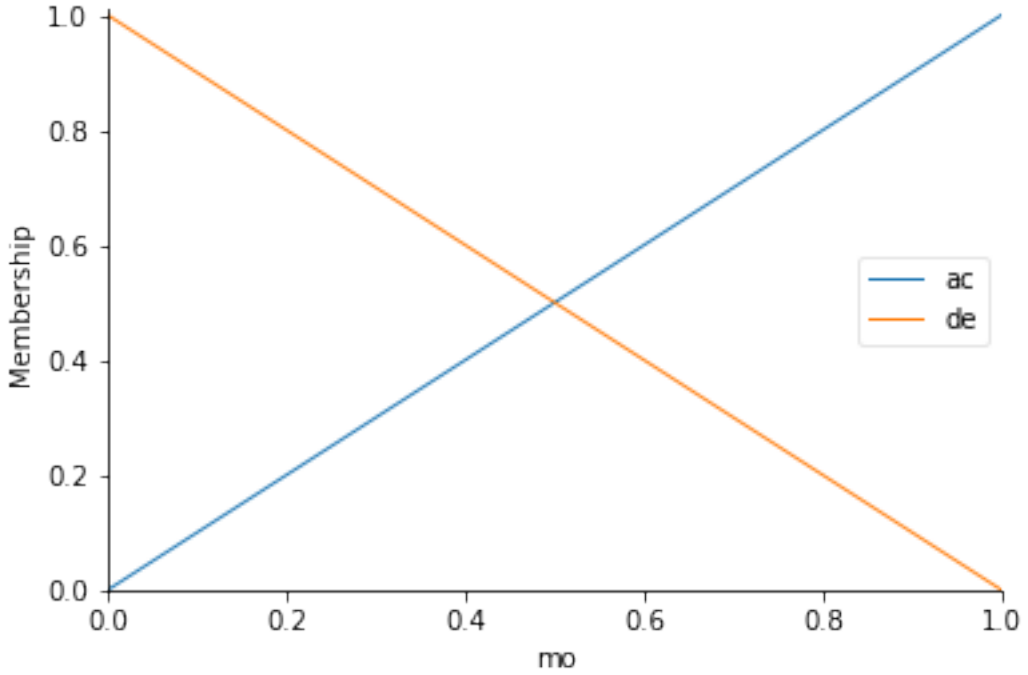
[5]:
```
#Setting Output Attributes
c_speed = ctrl.Consequent(np.arange(0,110,10),'comp_speed')
f_speed = ctrl.Consequent(np.arange(0,110,10),'fan_speed')
mo = ctrl.Consequent(np.arange(0,1.1,0.1),'mo')
f_dir = ctrl.Consequent(np.arange(0,100,10),'f_dir')
```

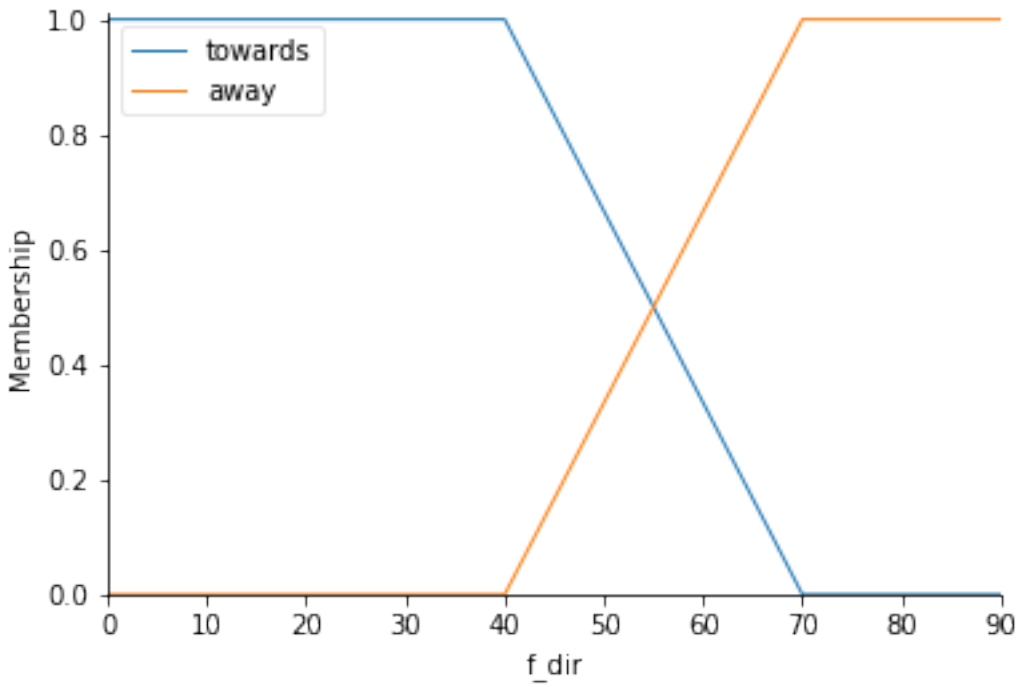
[6]:
```
#Setting Output Membership Functions
#Membership Function Values for Compressor Speed
c_speed['low'] = fuzzy.trapmf(c_speed.universe,[0,0,30,50])
c_speed['medium'] = fuzzy.trimf(c_speed.universe,[40,60,80])
c_speed['fast'] = fuzzy.trapmf(c_speed.universe,[70,90,100,100])
#Membership Function Values for Fan Speed
f_speed['low'] = fuzzy.trapmf(f_speed.universe,[0,0,30,50])
f_speed['medium'] = fuzzy.trimf(f_speed.universe,[40,60,80])
f_speed['fast'] = fuzzy.trapmf(f_speed.universe,[70,90,100,100])
#Membership Function for Mode of Operation
mo['ac'] = fuzzy.trimf(mo.universe,[0,1,1])
mo['de'] = fuzzy.trimf(mo.universe,[0,0,1])
#Membership Function Values for Fan Direction
f_dir['towards'] = fuzzy.trapmf(f_dir.universe,[0,0,40,70])
f_dir['away'] = fuzzy.trapmf(f_dir.universe,[40,70,90,90])
```

[7]:
```
c_speed.view()
f_speed.view()
mo.view()
f_dir.view()
```




low
medium
fast
Membership
fan_speed
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100


ac
de
Membership
mo
0.0
0.2
0.4
0.6
0.8
1.0



[8]:
```
#Rules for Fuzzy
csr_list = []
rule_1 = ctrl.
 ↪Rule(temp['low']&t_dif['negative']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed
csr_list.append(rule_1)
rule_2 = ctrl.
 ↪Rule(temp['medium']&t_dif['negative']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_sp
csr_list.append(rule_2)
rule_3 = ctrl.
 ↪Rule(temp['high']&t_dif['negative']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_spee
csr_list.append(rule_3)
rule_4 = ctrl.
 ↪Rule(temp['low']&t_dif['zero']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed['lo
csr_list.append(rule_4)
rule_5 = ctrl.
 ↪Rule(temp['medium']&t_dif['zero']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed[
csr_list.append(rule_5)
rule_6 = ctrl.
 ↪Rule(temp['high']&t_dif['zero']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed['l
csr_list.append(rule_6)
rule_7 = ctrl.
 ↪Rule(temp['low']&t_dif['positive']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed
csr_list.append(rule_7)
```

```
rule_8 = ctrl.
 ↪Rule(temp['medium']&t_dif['positive']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_sp
csr_list.append(rule_8)
rule_9 = ctrl.
 ↪Rule(temp['high']&t_dif['positive']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_spee
csr_list.append(rule_9)
rule_10 = ctrl.
 ↪Rule(temp['low']&t_dif['large']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed['l
csr_list.append(rule_10)
rule_11 = ctrl.
 ↪Rule(temp['medium']&t_dif['large']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed
csr_list.append(rule_11)
rule_12 = ctrl.
 ↪Rule(temp['high']&t_dif['large']&d_point['optimal']&e_volt['low'],[c_speed['low'],f_speed['
csr_list.append(rule_12)
rule_13 = ctrl.
 ↪Rule(temp['low']&t_dif['negative']&d_point['optimal']&e_volt['high'],[c_speed['low'],f_spee
csr_list.append(rule_13)
rule_14 = ctrl.
 ↪Rule(temp['medium']&t_dif['negative']&d_point['optimal']&e_volt['high'],[c_speed['low'],f_s
csr_list.append(rule_14)
rule_15 = ctrl.
 ↪Rule(temp['high']&t_dif['negative']&d_point['optimal']&e_volt['high'],[c_speed['low'],f_spe
csr_list.append(rule_15)
rule_16 = ctrl.
 ↪Rule(temp['low']&t_dif['zero']&d_point['optimal']&e_volt['high'],[c_speed['low'],f_speed['f
csr_list.append(rule_16)
rule_17 = ctrl.
 ↪Rule(temp['medium']&t_dif['zero']&d_point['optimal']&e_volt['high'],[c_speed['low'],f_speed
csr_list.append(rule_17)
rule_18 = ctrl.
 ↪Rule(temp['high']&t_dif['zero']&d_point['optimal']&e_volt['high'],[c_speed['low'],f_speed['
csr_list.append(rule_18)
rule_19 = ctrl.
 ↪Rule(temp['low']&t_dif['positive']&d_point['optimal']&e_volt['high'],[c_speed['fast'],f_spe
csr_list.append(rule_19)
rule_20 = ctrl.
 ↪Rule(temp['medium']&t_dif['positive']&d_point['optimal']&e_volt['high'],[c_speed['medium'],
csr_list.append(rule_20)
rule_21 = ctrl.
 ↪Rule(temp['high']&t_dif['positive']&d_point['optimal']&e_volt['high'],[c_speed['medium'],f_
csr_list.append(rule_21)
rule_22 = ctrl.
 ↪Rule(temp['low']&t_dif['large']&d_point['optimal']&e_volt['high'],[c_speed['fast'],f_speed[
csr_list.append(rule_22)
```

```
rule_23 = ctrl.
 ↪Rule(temp['medium']&t_dif['large']&d_point['optimal']&e_volt['high'],[c_speed['fast'],f_spe
csr_list.append(rule_23)
rule_24 = ctrl.
 ↪Rule(temp['high']&t_dif['large']&d_point['optimal']&e_volt['high'],[c_speed['fast'],f_speed
csr_list.append(rule_24)
rule_25 = ctrl.
 ↪Rule(temp['low']&t_dif['negative']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['
csr_list.append(rule_25)
rule_26 = ctrl.
 ↪Rule(temp['medium']&t_dif['negative']&d_point['humid']&e_volt['low'],[c_speed['low'],f_spee
csr_list.append(rule_26)
rule_27 = ctrl.
 ↪Rule(temp['high']&t_dif['negative']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed[
csr_list.append(rule_27)
rule_28 = ctrl.
 ↪Rule(temp['low']&t_dif['zero']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['low'
csr_list.append(rule_28)
rule_29 = ctrl.
 ↪Rule(temp['medium']&t_dif['zero']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['l
csr_list.append(rule_29)
rule_30 = ctrl.
 ↪Rule(temp['high']&t_dif['zero']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['low
csr_list.append(rule_30)
rule_31 = ctrl.
 ↪Rule(temp['low']&t_dif['positive']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['
csr_list.append(rule_31)
rule_32 = ctrl.
 ↪Rule(temp['medium']&t_dif['positive']&d_point['humid']&e_volt['low'],[c_speed['low'],f_spee
csr_list.append(rule_32)
rule_33 = ctrl.
 ↪Rule(temp['high']&t_dif['positive']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed[
csr_list.append(rule_33)
rule_34 = ctrl.
 ↪Rule(temp['low']&t_dif['large']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['low
csr_list.append(rule_34)
rule_35 = ctrl.
 ↪Rule(temp['medium']&t_dif['large']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['
csr_list.append(rule_35)
rule_36 = ctrl.
 ↪Rule(temp['high']&t_dif['large']&d_point['humid']&e_volt['low'],[c_speed['low'],f_speed['lo
csr_list.append(rule_36)
rule_37 = ctrl.
 ↪Rule(temp['low']&t_dif['negative']&d_point['humid']&e_volt['high'],[c_speed['fast'],f_speed
csr_list.append(rule_37)
```

```
rule_38 = ctrl.
 ↪Rule(temp['medium']&t_dif['negative']&d_point['humid']&e_volt['high'],[c_speed['low'],f_spe
csr_list.append(rule_38)
rule_39 = ctrl.
 ↪Rule(temp['high']&t_dif['negative']&d_point['humid']&e_volt['high'],[c_speed['low'],f_speed
csr_list.append(rule_39)
rule_40 = ctrl.
 ↪Rule(temp['low']&t_dif['zero']&d_point['humid']&e_volt['high'],[c_speed['fast'],f_speed['fa
csr_list.append(rule_40)
rule_41 = ctrl.
 ↪Rule(temp['medium']&t_dif['zero']&d_point['humid']&e_volt['high'],[c_speed['medium'],f_spee
csr_list.append(rule_41)
rule_42 = ctrl.
 ↪Rule(temp['high']&t_dif['zero']&d_point['humid']&e_volt['high'],[c_speed['medium'],f_speed[
csr_list.append(rule_42)
rule_43 = ctrl.
 ↪Rule(temp['low']&t_dif['positive']&d_point['humid']&e_volt['high'],[c_speed['fast'],f_speed
csr_list.append(rule_43)
rule_44 = ctrl.
 ↪Rule(temp['medium']&t_dif['positive']&d_point['humid']&e_volt['high'],[c_speed['fast'],f_sp
csr_list.append(rule_44)
rule_45 = ctrl.
 ↪Rule(temp['high']&t_dif['positive']&d_point['humid']&e_volt['high'],[c_speed['medium'],f_sp
csr_list.append(rule_45)
rule_46 = ctrl.
 ↪Rule(temp['low']&t_dif['large']&d_point['humid']&e_volt['high'],[c_speed['fast'],f_speed['f
csr_list.append(rule_46)
rule_47= ctrl.
 ↪Rule(temp['medium']&t_dif['large']&d_point['humid']&e_volt['high'],[c_speed['fast'],f_speed
csr_list.append(rule_47)
rule_48 = ctrl.
 ↪Rule(temp['high']&t_dif['large']&d_point['humid']&e_volt['high'],[c_speed['fast'],f_speed['
csr_list.append(rule_48)
```

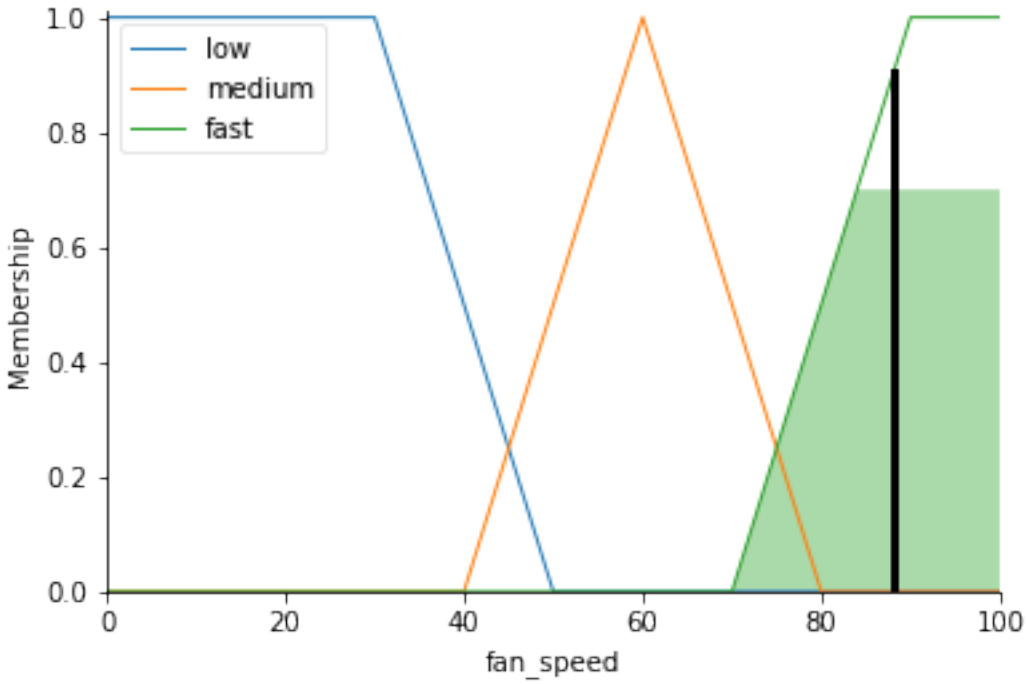
[9]:
```
ac_ctrl = ctrl.ControlSystem(csr_list)
```

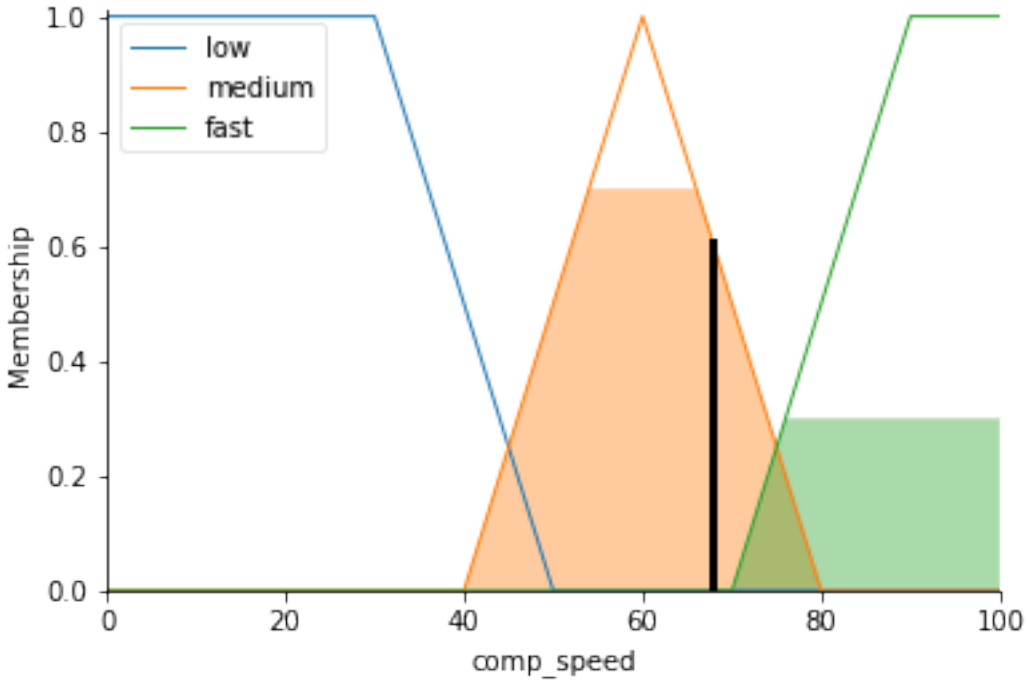
[10]:
```
ac = ctrl.ControlSystemSimulation(ac_ctrl)
```

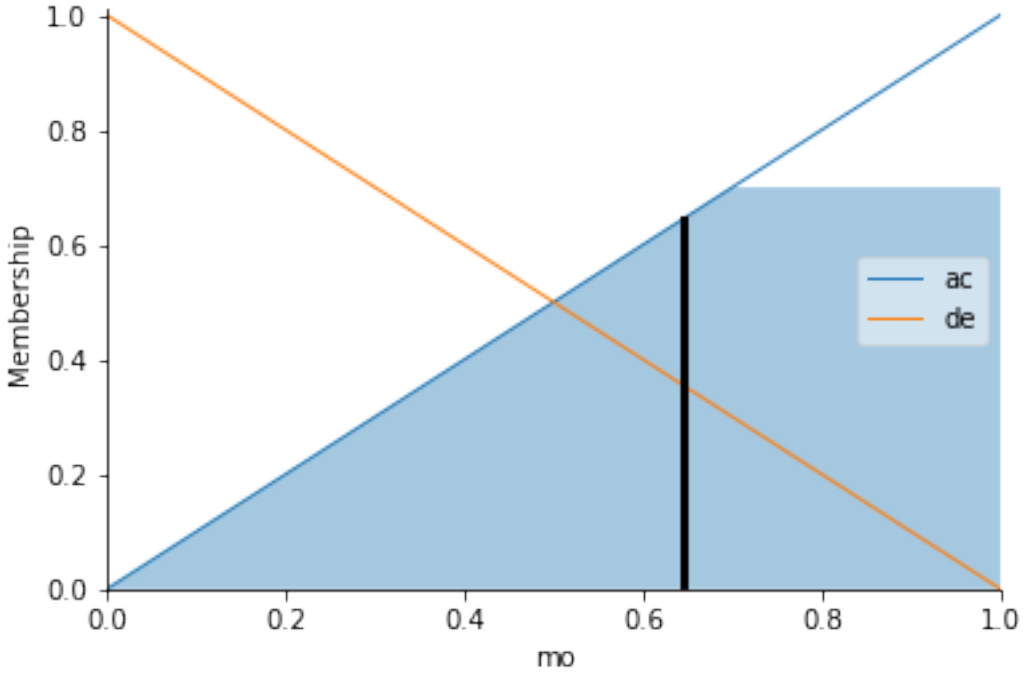
[11]:
```
ac.input['temp']  = 29
ac.input['t_dif'] = 1.3
ac.input['dew'] = 15
ac.input['ev'] = 209
ac.compute()
print(ac.output['comp_speed'],ac.output['fan_speed'],ac.output['mo'],ac.
 ↪output['f_dir'])
```

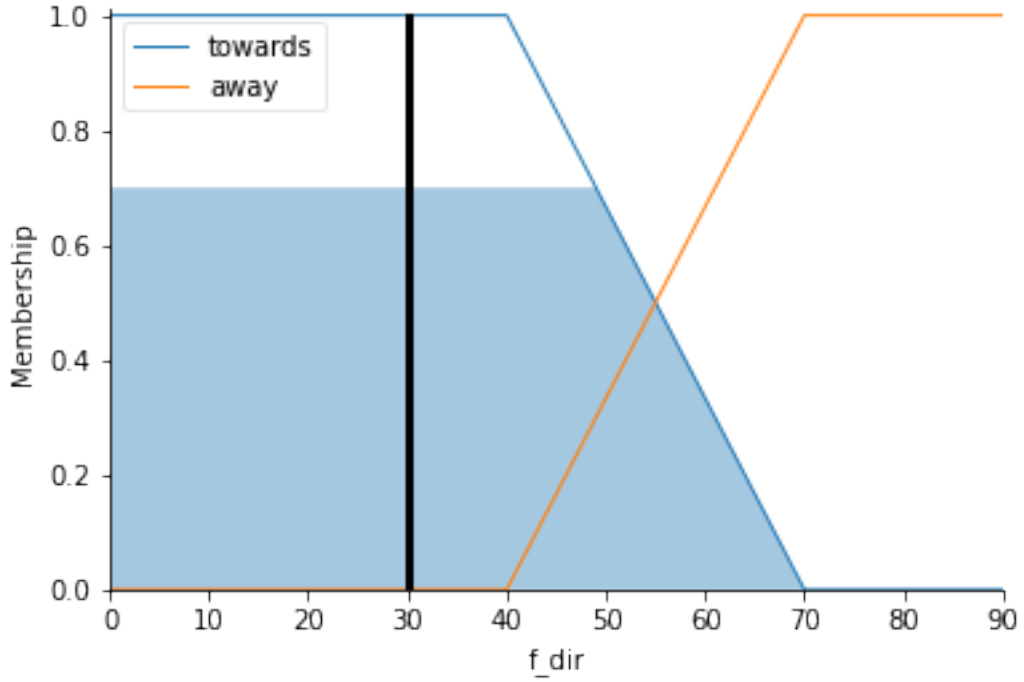
```
67.86034255599472 88.14492753623188 0.6435897435897436 30.05882352941177
```

[12]:
```
f_speed.view(sim=ac)
c_speed.view(sim=ac)
mo.view(sim=ac)
f_dir.view(sim=ac)
```



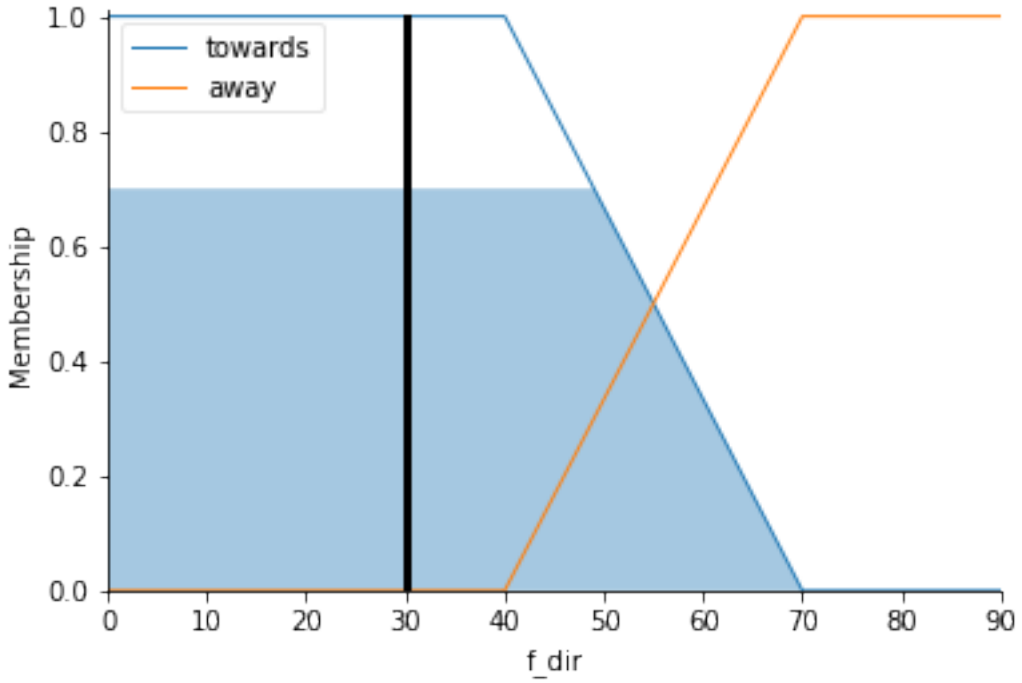



1.0
0.8
0.6
0.4
0.2
0.0
Membership
0.0
0.2
0.4
0.6
0.8
1.0
mo
ac
de


1.0
0.8
0.6
0.4
0.2
0.0
Membership
towards
away
0
10
20
30
40
50
60
70
80
90
f_dir

[13]:
```
print(f_dir.view(sim=ac))
```

```
None
```



[14]:
```
ac.output
```

[14]:
```
OrderedDict([('comp_speed', 67.86034255599472),
             ('fan_speed', 88.14492753623188),
             ('mo', 0.6435897435897436),
             ('f_dir', 30.05882352941177)])
```

[15]:
```
import skfuzzy

def get_linguist(valmf,val,val_range):
    mf_keys = list(valmf.terms.keys())
    mf_value_list = []
    for i in range(len(mf_keys)):
        mf_value_list.append(skfuzzy.
 ↪interp_membership(val_range,valmf[mf_keys[i]].mf,val))
    max_mf = max(mf_value_list)
    #print(mf_keys)
    #print(mf_value_list)
    ling_key = ''
    for i in range(len(mf_keys)):
        if max_mf == mf_value_list[i]:
```

```
            ling_key = mf_keys[i]
            break
    return ling_key
```

[16]:
```
print(get_linguist(f_speed,ac.output['fan_speed'],np.arange(0,110,10)))
```

```
fast
```

[17]:
```
temp_range = np.arange(16,31,1)
tdif_range = np.arange(-1,3,0.1)
dew_range = np.arange(10,18,1)
ev_range = np.arange(130,220,10)
c_speed_range = np.arange(0,110,10)
f_speed_range = np.arange(0,110,10)
mo_range = np.arange(0,1.1,0.1)
dir_range = np.arange(0,100,10)
```

[18]:
```
temp_list = []
tdif_list = []
dew_list = []
ev_list = []
temp_list.append(np.random.randint(low=16,high = 22))
temp_list.append(np.random.randint(low=23,high = 27))
temp_list.append(np.random.randint(low=27,high = 30))
tdif_list.append(float(np.random.uniform(low = -1,high = -0.6,size=(1,1))))
tdif_list.append(float(np.random.uniform(low = -0.5,high = 0.5,size=(1,1))))
tdif_list.append(float(np.random.uniform(low = 0.6,high = 1,size=(1,1))))
tdif_list.append(float(np.random.uniform(low = 1.1,high = 3,size=(1,1))))
dew_list.append(np.random.randint(low=10,high=13))
dew_list.append(np.random.randint(low=14,high=18))
ev_list.append(np.random.randint(low=130,high=160))
ev_list.append(np.random.randint(low=170,high=220))
print(temp_list)
print(tdif_list)
print(dew_list)
print(ev_list)
```

```
[18, 25, 28]
[-0.946481115184088, -0.4880827292945036, 0.9385890639551544,
2.8913008330498697]
[10, 17]
[140, 215]
```

[19]:
```
print('temp','   t_dif   ','   dew   ',' ev   ','   cs   ','   fs'   ,'     mo ␣
 ↪ ','     f_dir   ')
for i in range(len(temp_list)):
    for j in range(len(tdif_list)):
        for k in range(len(dew_list)):
```

```
            for p in range(len(ev_list)):
                ac.input['temp'] = temp_list[i]
                ac.input['t_dif'] = tdif_list[j]
                ac.input['dew'] = dew_list[k]
                ac.input['ev'] = ev_list[p]
                ac.compute()
                print(temp_list[i],tdif_list[j],dew_list[k],ev_list[p],ac.
↪output['comp_speed'],ac.output['fan_speed'],ac.output['mo'],ac.
↪output['f_dir'])
```

```
temp    t_dif       dew     ev       cs       fs      mo          f_dir
18 -0.946481115184088 10 140 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
18 -0.946481115184088 10 215 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
18 -0.946481115184088 17 140 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
18 -0.946481115184088 17 215 89.00533937748418 89.00533937748418
0.3342395885741334 28.503395547619874
18 -0.4880827292945036 10 140 22.64758469818623 22.64758469818623
0.6088902928918676 68.4513025714036
18 -0.4880827292945036 10 215 22.64758469818623 24.59707462710421
0.6088902928918676 66.20570873172238
18 -0.4880827292945036 17 140 22.64758469818623 22.64758469818623
0.6088902928918676 68.4513025714036
18 -0.4880827292945036 17 215 87.2823506180083 87.2823506180083
0.39110970710813237 31.481906732213
18 0.9385890639551544 10 140 20.668566744565315 20.668566744565315
0.665482298784937 71.11974834913045
18 0.9385890639551544 10 215 88.9806908992942 88.9806908992942 0.665482298784937
28.551335167706107
18 0.9385890639551544 17 140 20.668566744565315 20.668566744565315
0.665482298784937 71.11974834913045
18 0.9385890639551544 17 215 88.9806908992942 88.9806908992942 0.665482298784937
28.551335167706107
18 2.8913008330498697 10 140 20.416666666666664 20.416666666666664
0.6666666666666666 71.42857142857143
18 2.8913008330498697 10 215 89.16666666666666 89.16666666666666
0.6666666666666666 28.181818181818187
18 2.8913008330498697 17 140 20.416666666666664 20.416666666666664
0.6666666666666666 71.42857142857143
18 2.8913008330498697 17 215 89.16666666666666 89.16666666666666
0.6666666666666666 28.181818181818187
25 -0.946481115184088 10 140 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
25 -0.946481115184088 10 215 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
25 -0.946481115184088 17 140 20.63592727546777 20.63592727546777
```

0.6657604114258666 71.16031556281365
25 -0.946481115184088 17 215 20.63592727546777 20.63592727546777
0.3342395885741334 71.16031556281365
25 -0.4880827292945036 10 140 22.64758469818623 22.64758469818623
0.6088902928918676 68.4513025714036
25 -0.4880827292945036 10 215 22.64758469818623 24.07910079166665
0.6088902928918676 66.20570873172238
25 -0.4880827292945036 17 140 22.64758469818623 22.64758469818623
0.6088902928918676 68.4513025714036
25 -0.4880827292945036 17 215 24.07910079166665 24.59707462710421
0.39110970710813237 66.20570873172238
25 0.9385890639551544 10 140 20.668566744565315 20.668566744565315
0.665482298784937 71.11974834913045
25 0.9385890639551544 10 215 60.0 60.0 0.665482298784937 28.551335167706107
25 0.9385890639551544 17 140 20.668566744565315 20.668566744565315
0.665482298784937 71.11974834913045
25 0.9385890639551544 17 215 88.9806908992942 88.9806908992942 0.665482298784937
28.551335167706107
25 2.8913008330498697 10 140 20.416666666666664 20.416666666666664
0.6666666666666666 71.42857142857143
25 2.8913008330498697 10 215 89.16666666666666 89.16666666666666
0.6666666666666666 28.181818181818187
25 2.8913008330498697 17 140 20.416666666666664 20.416666666666664
0.6666666666666666 71.42857142857143
25 2.8913008330498697 17 215 89.16666666666666 89.16666666666666
0.6666666666666666 28.181818181818187
28 -0.946481115184088 10 140 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
28 -0.946481115184088 10 215 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
28 -0.946481115184088 17 140 20.63592727546777 20.63592727546777
0.6657604114258666 71.16031556281365
28 -0.946481115184088 17 215 20.63592727546777 20.63592727546777
0.3342395885741334 71.16031556281365
28 -0.4880827292945036 10 140 22.64758469818623 22.64758469818623
0.6088902928918676 68.4513025714036
28 -0.4880827292945036 10 215 22.64758469818623 22.64758469818623
0.6088902928918676 68.4513025714036
28 -0.4880827292945036 17 140 22.64758469818623 22.64758469818623
0.6088902928918676 68.4513025714036
28 -0.4880827292945036 17 215 24.07910079166665 24.07910079166665
0.39110970710813237 66.20570873172238
28 0.9385890639551544 10 140 20.668566744565315 20.668566744565315
0.665482298784937 71.11974834913045
28 0.9385890639551544 10 215 60.0 60.0 0.665482298784937 28.551335167706107
28 0.9385890639551544 17 140 20.668566744565315 20.668566744565315
0.665482298784937 71.11974834913045
28 0.9385890639551544 17 215 60.0 88.9806908992942 0.665482298784937

```
28.551335167706107
28 2.8913008330498697 10 140 20.416666666666664 20.416666666666664
0.6666666666666666 71.42857142857143
28 2.8913008330498697 10 215 89.16666666666666 89.16666666666666
0.6666666666666666 28.181818181818187
28 2.8913008330498697 17 140 20.416666666666664 20.416666666666664
0.6666666666666666 71.42857142857143
28 2.8913008330498697 17 215 89.16666666666666 89.16666666666666
0.6666666666666666 28.181818181818187
```

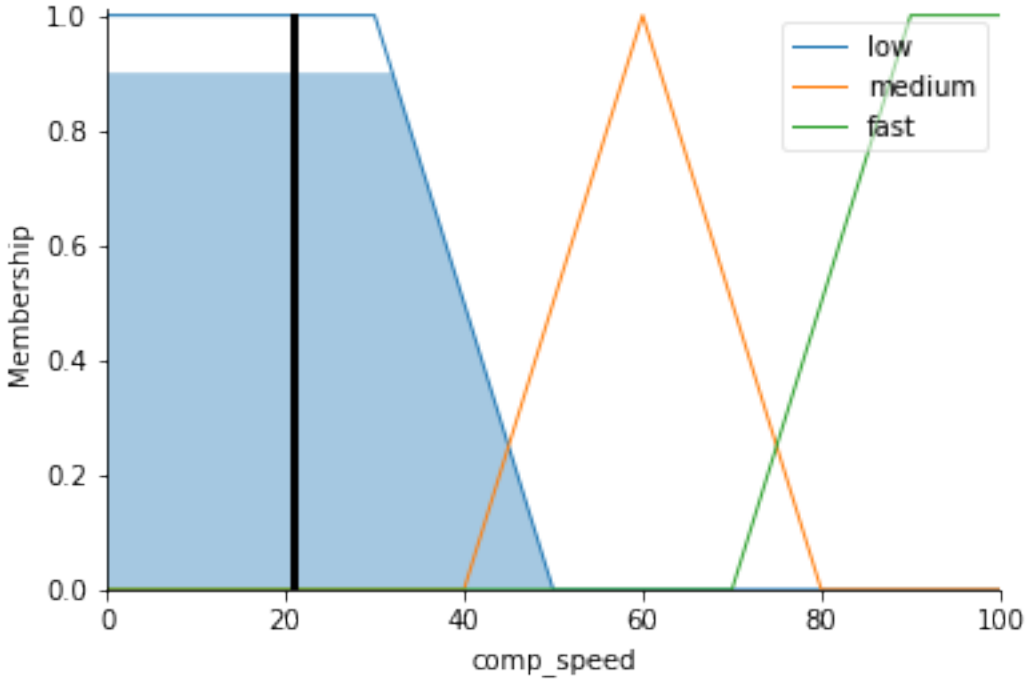
[20]:
```
print('Linguistic Values\n')
print('TEMP','T_DIF',' DEW ','EV',' CS ','FS',' MO',' F_DIR')
print('\n')
for i in range(len(temp_list)):
    for j in range(len(tdif_list)):
        for k in range(len(dew_list)):
            for p in range(len(ev_list)):
                ac.input['temp'] = temp_list[i]
                ac.input['t_dif'] = tdif_list[j]
                ac.input['dew'] = dew_list[k]
                ac.input['ev'] = ev_list[p]
                ac.compute()
                temp_ling = get_linguist(temp,temp_list[i],temp_range)
                tdif_ling = get_linguist(t_dif,tdif_list[j],tdif_range)
                dew_ling = get_linguist(d_point,dew_list[k],dew_range)
                ev_ling = get_linguist(e_volt,ev_list[p],ev_range)
                cs_ling = get_linguist(c_speed,ac.
 ↪output['comp_speed'],c_speed_range)
                fs_ling = get_linguist(f_speed,ac.
 ↪output['fan_speed'],f_speed_range)
                mo_ling = get_linguist(mo,ac.output['mo'],mo_range)
                dir_ling = get_linguist(f_dir,ac.output['f_dir'],dir_range)
                ␣
 ↪print(temp_ling,tdif_ling,dew_ling,ev_ling,cs_ling,fs_ling,mo_ling,dir_ling)
```

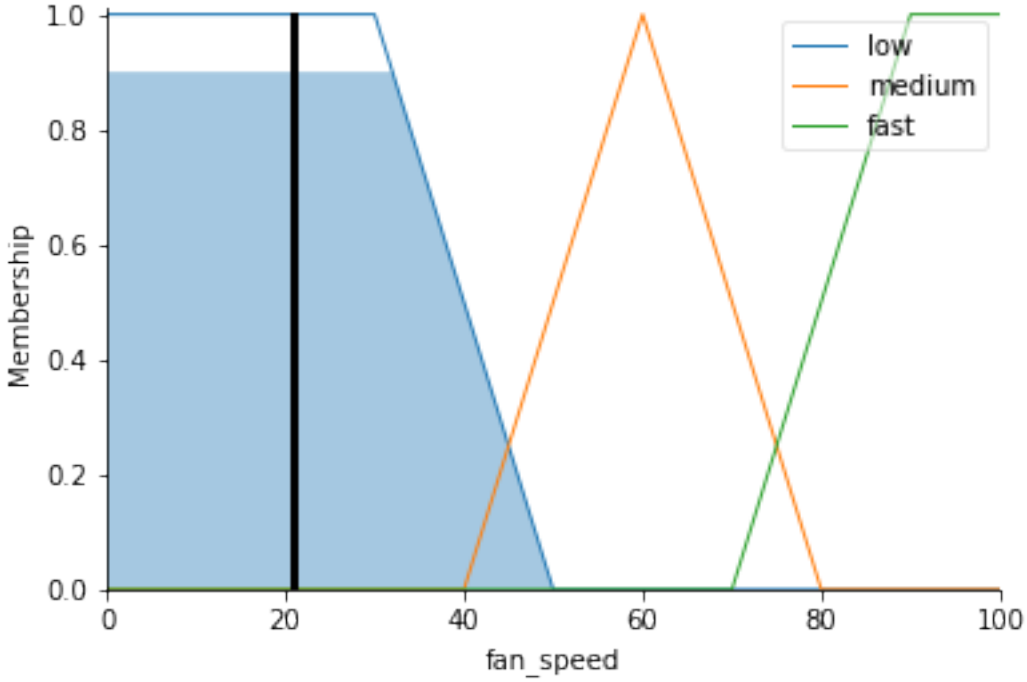
```
Linguistic Values

TEMP T_DIF  DEW  EV  CS  FS  MO  F_DIR


low negative optimal low low low ac away
low negative optimal low low low ac away
low negative humid low low low ac away
low negative humid low fast fast de towards
low negative optimal low low low ac away
low negative optimal low low low ac away
low negative humid low low low ac away
low negative humid low fast fast de towards
```

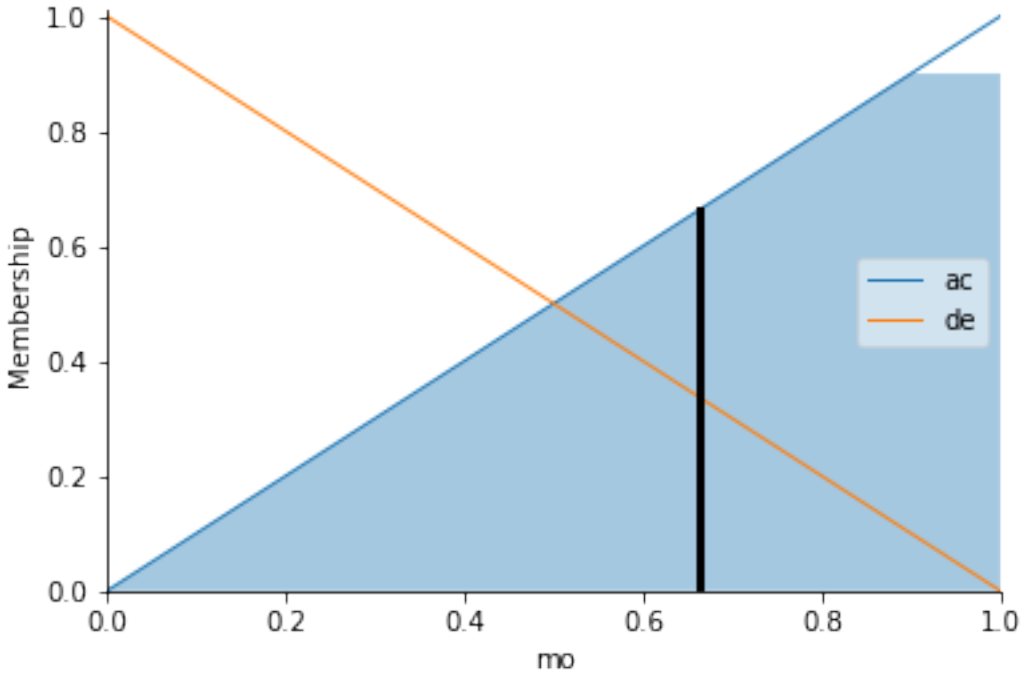
```
low positive optimal low low low ac away
low positive optimal low fast fast ac towards
low positive humid low low low ac away
low positive humid low fast fast ac towards
low large optimal low low low ac away
low large optimal low fast fast ac towards
low large humid low low low ac away
low large humid low fast fast ac towards
medium negative optimal low low low ac away
medium negative optimal low low low ac away
medium negative humid low low low ac away
medium negative humid low low low de away
medium negative optimal low low low ac away
medium negative optimal low low low ac away
medium negative humid low low low ac away
medium negative humid low low low de away
medium positive optimal low low low ac away
medium positive optimal low medium medium ac towards
medium positive humid low low low ac away
medium positive humid low fast fast ac towards
medium large optimal low low low ac away
medium large optimal low fast fast ac towards
medium large humid low low low ac away
medium large humid low fast fast ac towards
high negative optimal low low low ac away
high negative optimal low low low ac away
high negative humid low low low ac away
high negative humid low low low de away
high negative optimal low low low ac away
high negative optimal low low low ac away
high negative humid low low low ac away
high negative humid low low low de away
high positive optimal low low low ac away
high positive optimal low medium medium ac towards
high positive humid low low low ac away
high positive humid low medium fast ac towards
high large optimal low low low ac away
high large optimal low fast fast ac towards
high large humid low low low ac away
high large humid low fast fast ac towards
```

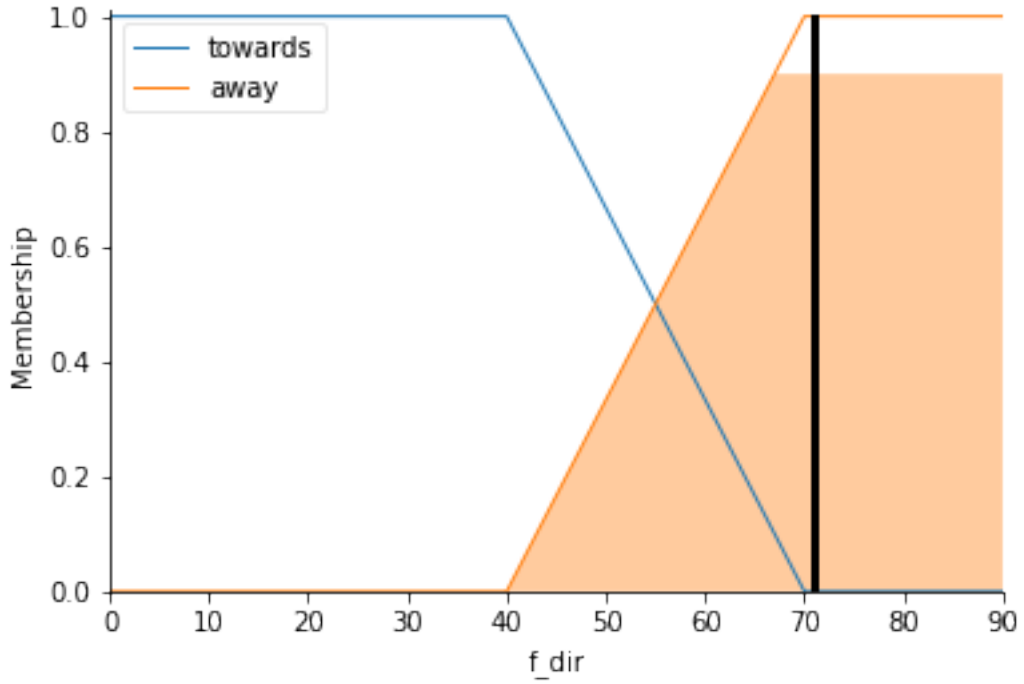
[21]:
```
ac.input['temp'] = 17
ac.input['t_dif'] = -0.9
ac.input['dew'] = 11
ac.input['ev'] = 159
ac.compute()
c_speed.view(sim = ac)
```

```
f_speed.view(sim = ac)
mo.view(sim = ac)
f_dir.view(sim=ac)
```






1.0
0.8
0.6
0.4
0.2
0.0
Membership
0.0
0.2
0.4
0.6
0.8
1.0
mo
ac
de


1.0
0.8
0.6
0.4
0.2
0.0
Membership
towards
away
0
10
20
30
40
50
60
70
80
90
f_dir

[22]:
```
get_linguist(f_speed,ac.output['fan_speed'],f_speed_range)
```

[22]:
```
'low'
```