Examples-UPAFuzzySystems

November 8, 2022

1 Examples for using the UPAFuzzySystems library

Developed at the Universidad Politécnica de Aguascalientes by Dr. Martín Montes Rivera

1.1 Required libraries

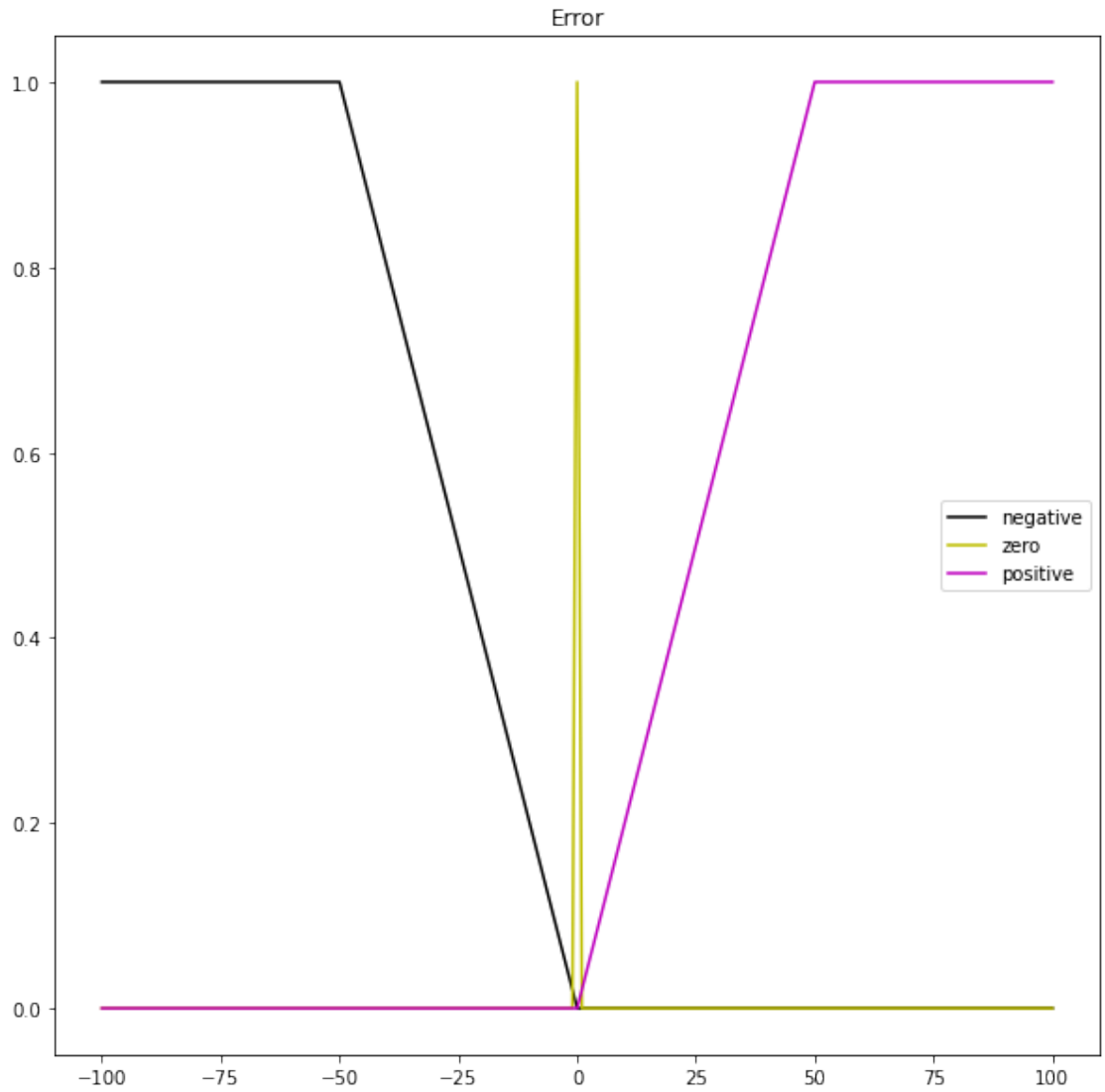
[1]:
```
import control
import numpy as np
import matplotlib.pyplot as plt
from pytictoc import TicToc
from UPAFuzzySystems import fuzzy_universe, inference_system, fuzzy_controller
tt = TicToc()
t = TicToc()
```

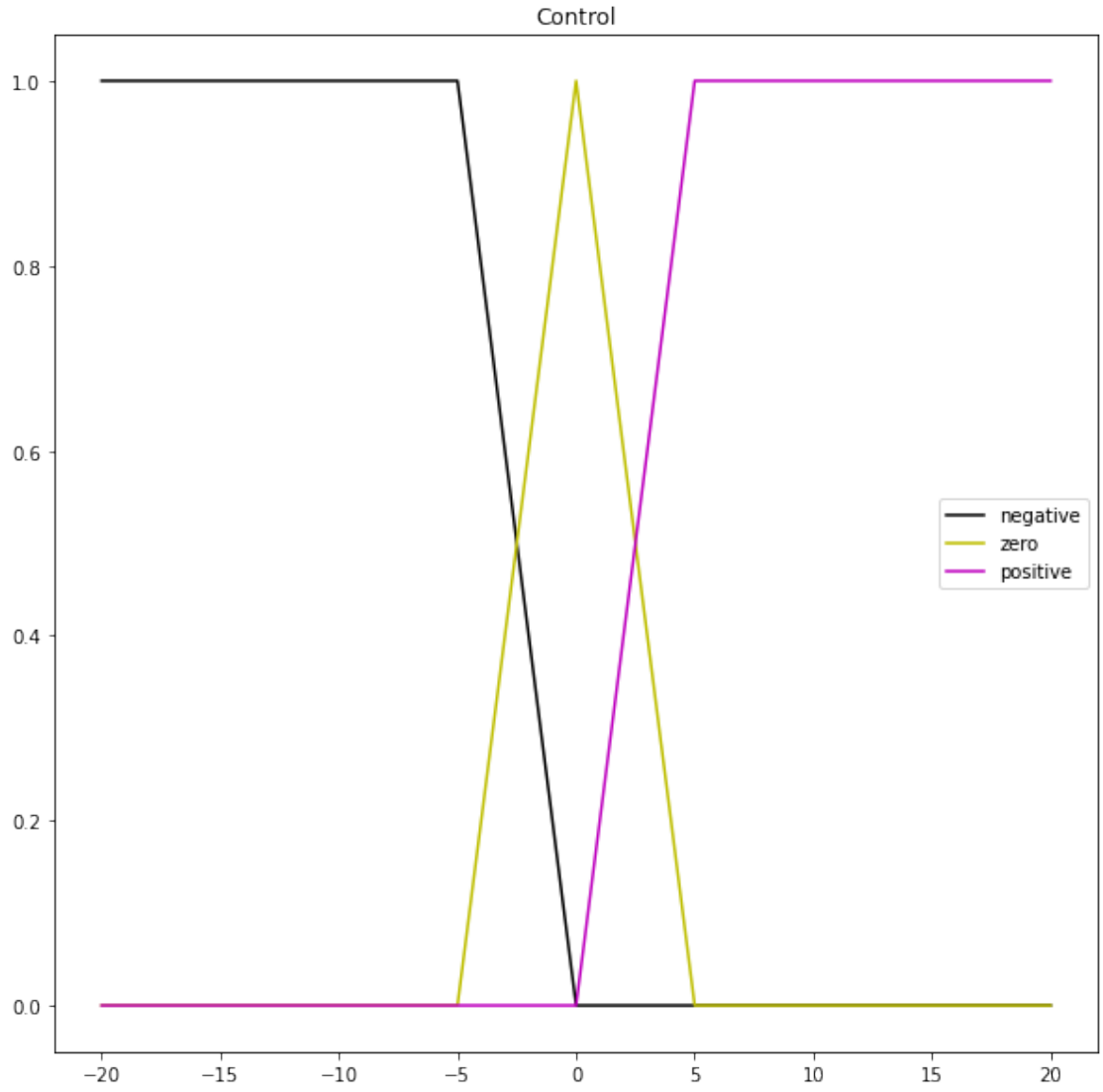
1.2 Inference Systems

1.2.1 Mamdani Inference System One Input

[2]:
```
tt.tic()
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trapmf',[-100,-100,-50,0])
Error_universe.add_fuzzyset('zero','trimf',[-1,0,1])
Error_universe.add_fuzzyset('positive','trapmf',[0,50,100,100])
Error_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-20,21,1), 'continuous')
Control_universe.add_fuzzyset('negative','trapmf',[-20,-20,-5,0])
Control_universe.add_fuzzyset('zero','trimf',[-5,-0,5])
Control_universe.add_fuzzyset('positive','trapmf',[0,5,20,20])
Control_universe.view_fuzzy()

Mamdani1 = inference_system('Mamdani')
Mamdani1.add_premise(Error_universe)
Mamdani1.add_consequence(Control_universe)
Mamdani1.add_rule([['Error','negative']],[],[['Control','negative']])
Mamdani1.add_rule([['Error','zero']],[],[['Control','zero']])
```

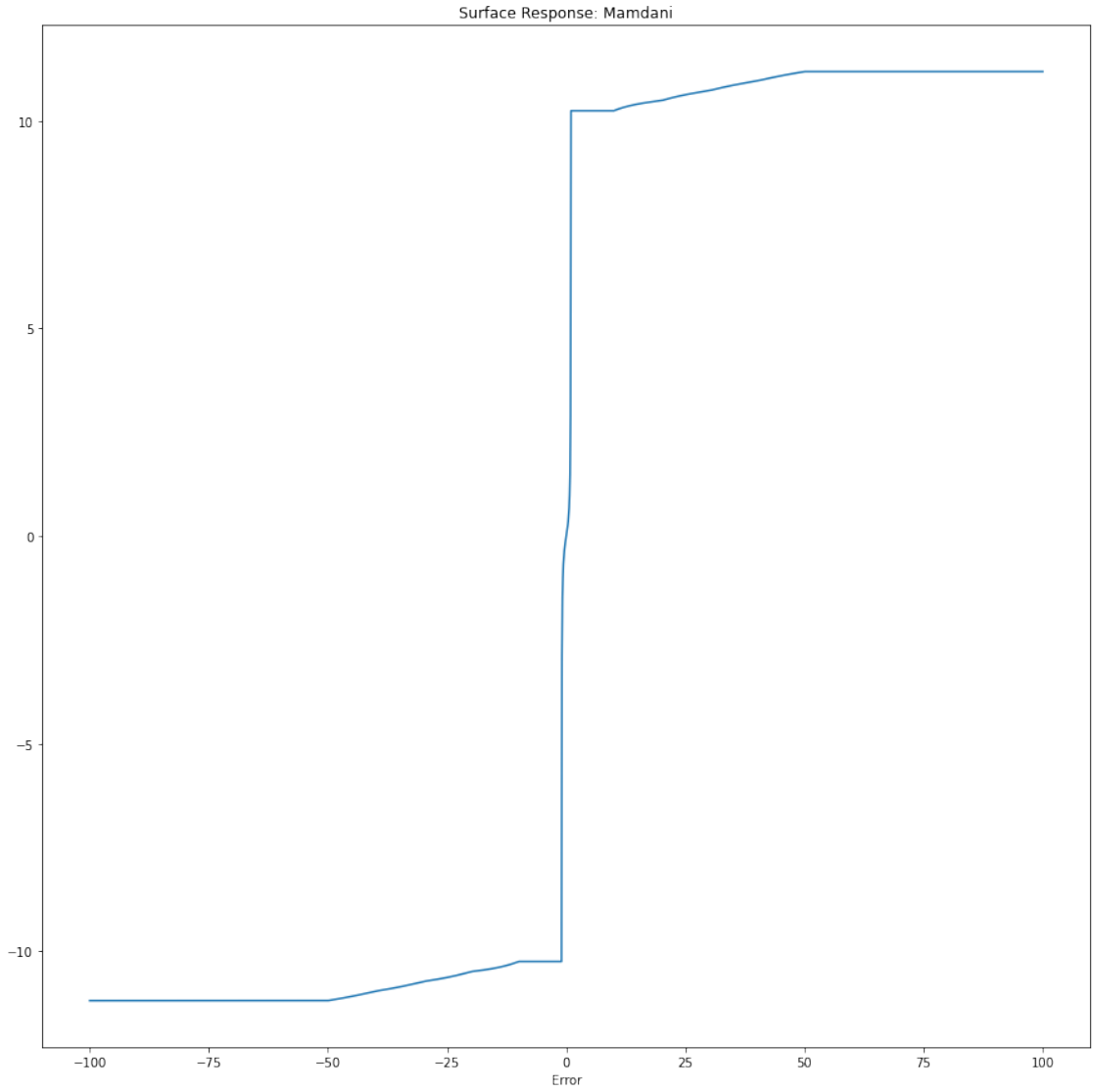
```
Mamdani1.add_rule([['Error','positive']],[],[['Control','positive']])

Mamdani1.configure('Mamdani')
Mamdani1.build()
t.toc()
del Error_universe
del Control_universe
```






```
Elapsed time is 0.641809 seconds.
```

[3]:
```
t.tic()
error_values = np.arange(-100,100.1,0.1)
Mamdani1.surface_fuzzy_system([error_values])
t.toc()
```

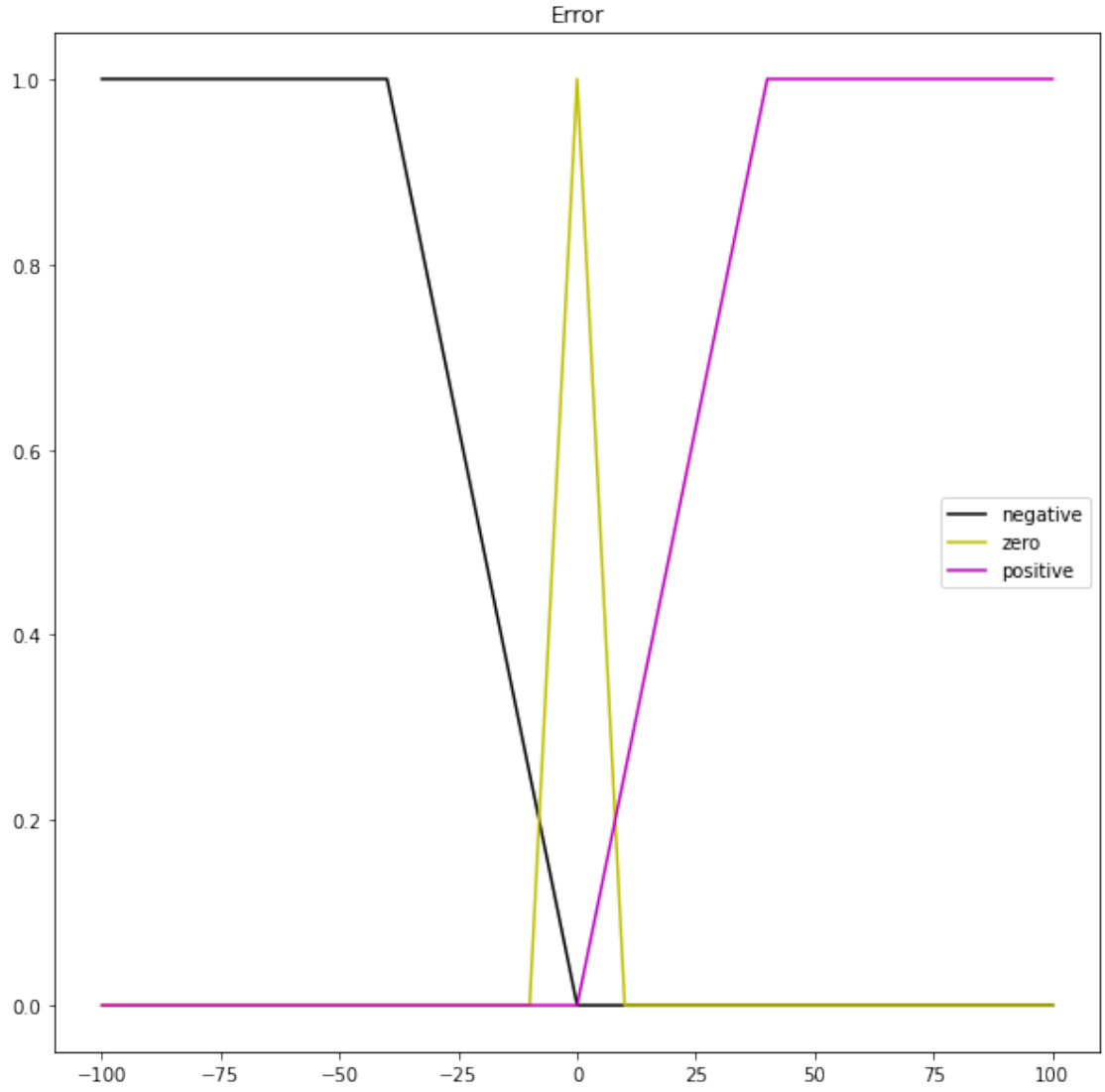


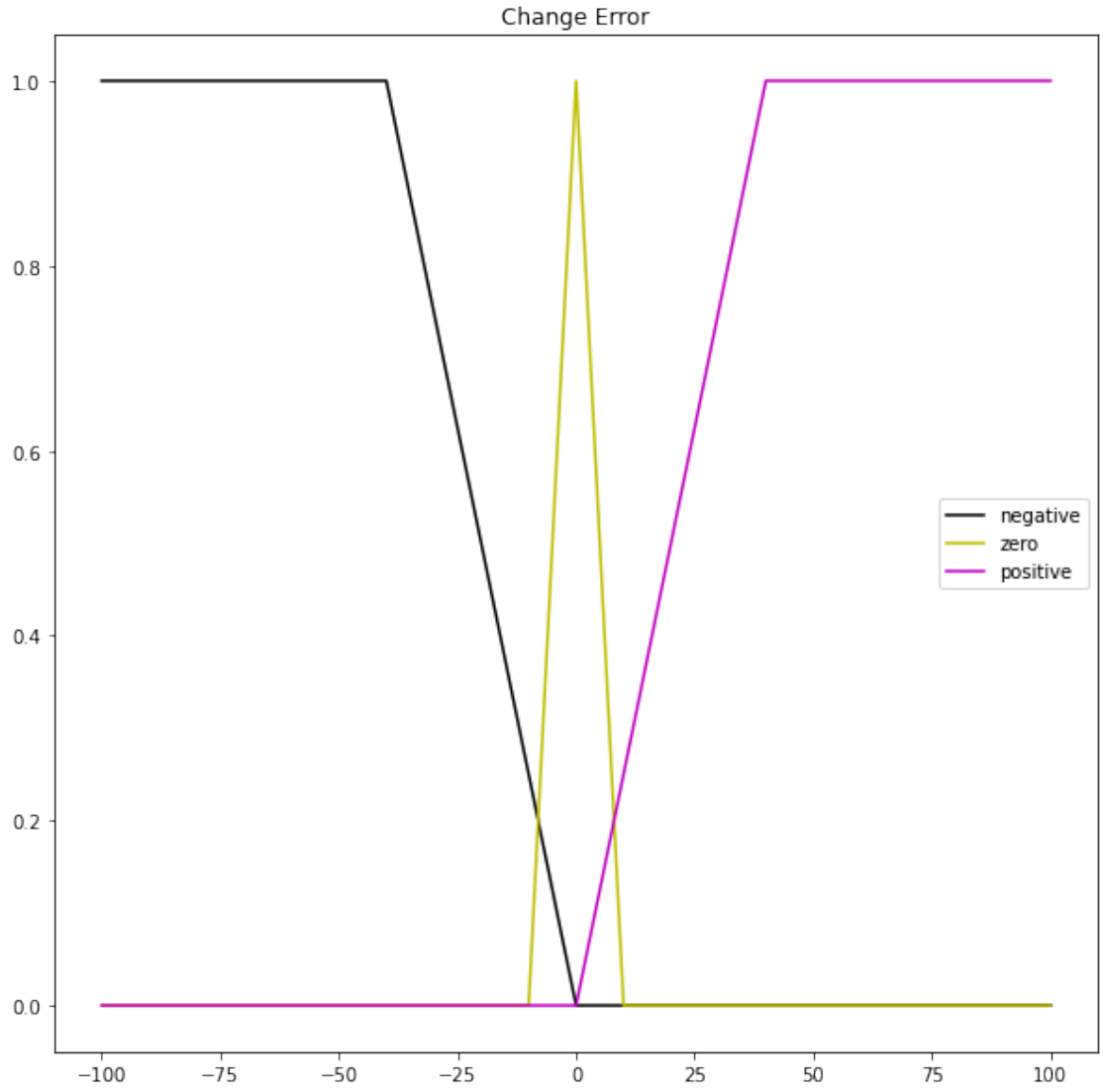

```
Elapsed time is 2.155710 seconds.
```

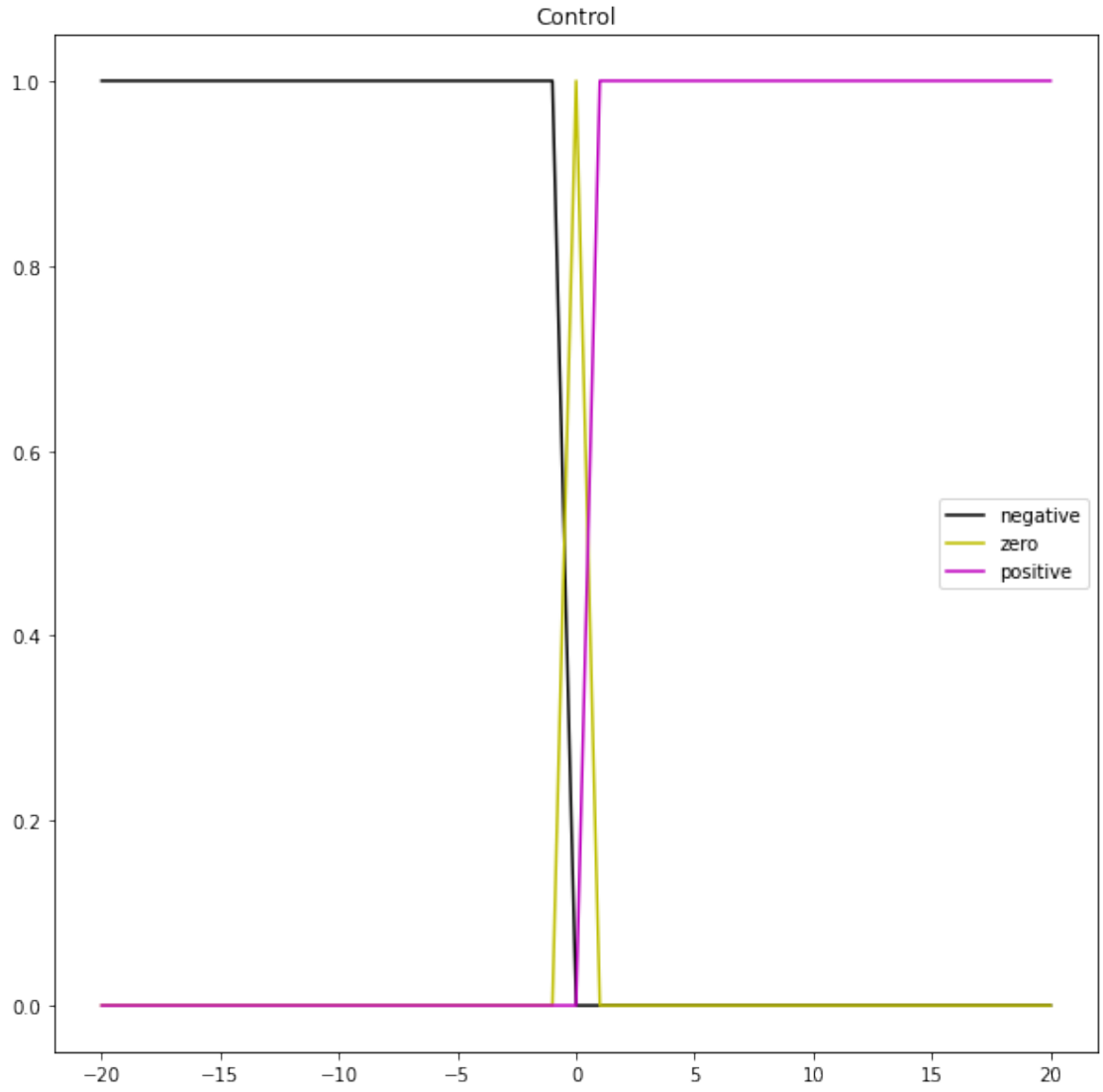
1.2.2 Mamdani Inference System Two Inputs

[4]:
```
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trapmf',[-100,-100,-40,0])
Error_universe.add_fuzzyset('zero','trimf',[-10,0,10])
Error_universe.add_fuzzyset('positive','trapmf',[0,40,100,100])
Error_universe.view_fuzzy()

ChError_universe = fuzzy_universe('Change Error', np.arange(-100,101,1),␣
↪'continuous')
```

```
ChError_universe.add_fuzzyset('negative','trapmf',[-100,-100,-40,0])
ChError_universe.add_fuzzyset('zero','trimf',[-10,0,10])
ChError_universe.add_fuzzyset('positive','trapmf',[0,40,100,100])
ChError_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-20,21,1), 'continuous')
Control_universe.add_fuzzyset('negative','trapmf',[-20,-20,-0.5,0])
Control_universe.add_fuzzyset('zero','trimf',[-0.01,0,0.01])
Control_universe.add_fuzzyset('positive','trapmf',[0,0.5,20,20])
Control_universe.view_fuzzy()

Mamdani2 = inference_system('Mamdani')
Mamdani2.add_premise(Error_universe)
Mamdani2.add_premise(ChError_universe)
Mamdani2.add_consequence(Control_universe)

Mamdani2.add_rule([['Error','negative'],['Change␣
 ↪Error','negative']],['and'],[['Control','negative']])
Mamdani2.add_rule([['Error','negative'],['Change␣
 ↪Error','zero']],['and'],[['Control','negative']])
Mamdani2.add_rule([['Error','zero'],['Change␣
 ↪Error','negative']],['and'],[['Control','zero']])
Mamdani2.add_rule([['Error','negative'],['Change␣
 ↪Error','positive']],['and'],[['Control','zero']])
Mamdani2.add_rule([['Error','zero'],['Change␣
 ↪Error','zero']],['and'],[['Control','zero']])
Mamdani2.add_rule([['Error','positive'],['Change␣
 ↪Error','negative']],['and'],[['Control','zero']])
Mamdani2.add_rule([['Error','zero'],['Change␣
 ↪Error','positive']],['and'],[['Control','zero']])
Mamdani2.add_rule([['Error','positive'],['Change␣
 ↪Error','zero']],['and'],[['Control','positive']])
Mamdani2.add_rule([['Error','positive'],['Change␣
 ↪Error','positive']],['and'],[['Control','positive']])


Mamdani2.configure('Mamdani')

Mamdani2.build()
t.toc()
del Error_universe
del ChError_universe
del Control_universe
```


Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
zero
positive

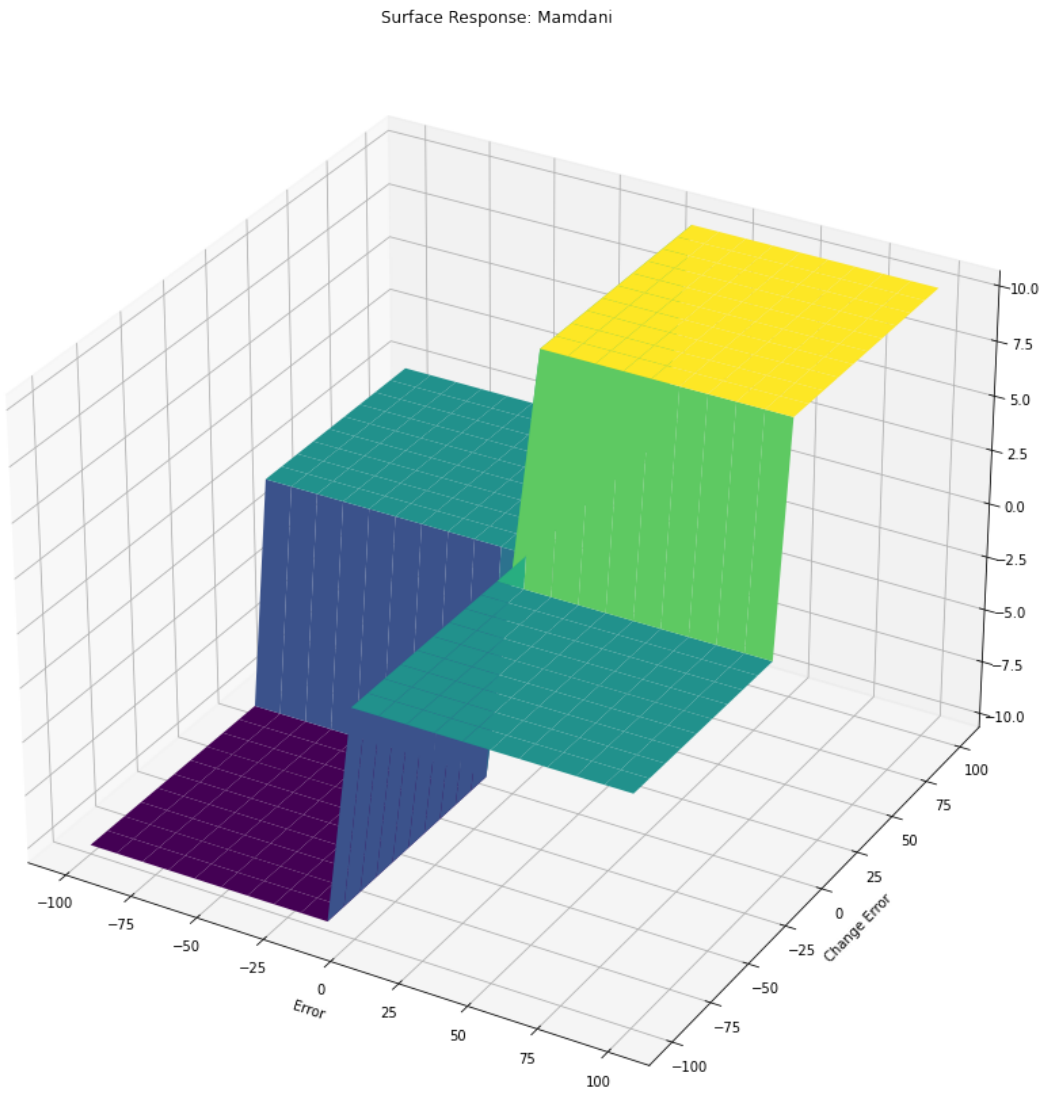

Change Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
zero
positive




```
Elapsed time is 0.697659 seconds.
```

[5]:
```
t.tic()
error_values = np.arange(-100,110,10)
change_error = np.arange(-100,110,10)

Mamdani2.surface_fuzzy_system([error_values,change_error])
t.toc()
```

```
(21, 21)
(21, 21)
(21, 21)
```

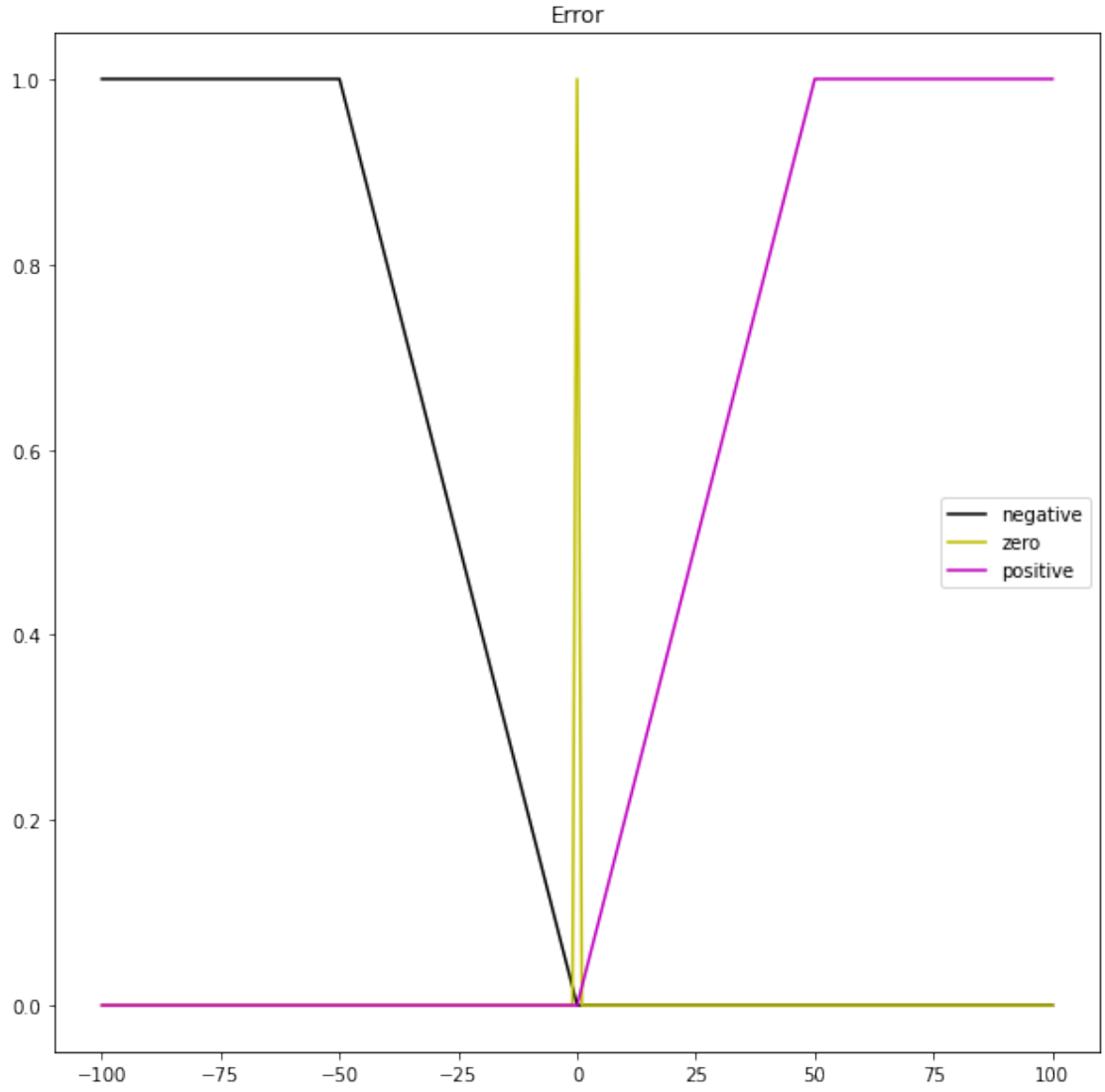


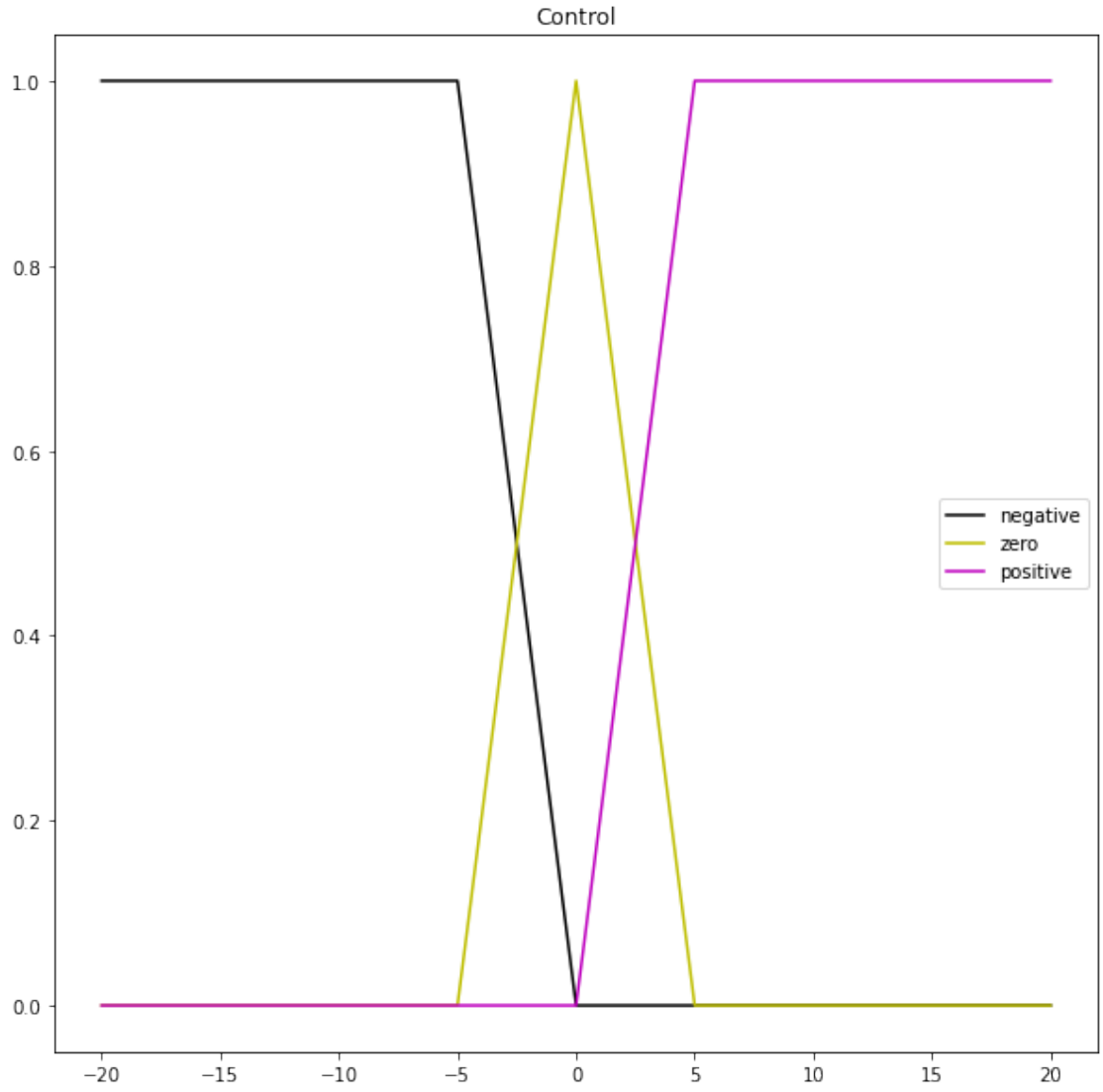

```
Elapsed time is 1.896674 seconds.
```

1.2.3 F.L. Smidth Inference System One Input

[6]:
```
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trapmf',[-100,-100,-50,0])
Error_universe.add_fuzzyset('zero','trimf',[-1,0,1])
Error_universe.add_fuzzyset('positive','trapmf',[0,50,100,100])
Error_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-20,21,1), 'continuous')
```

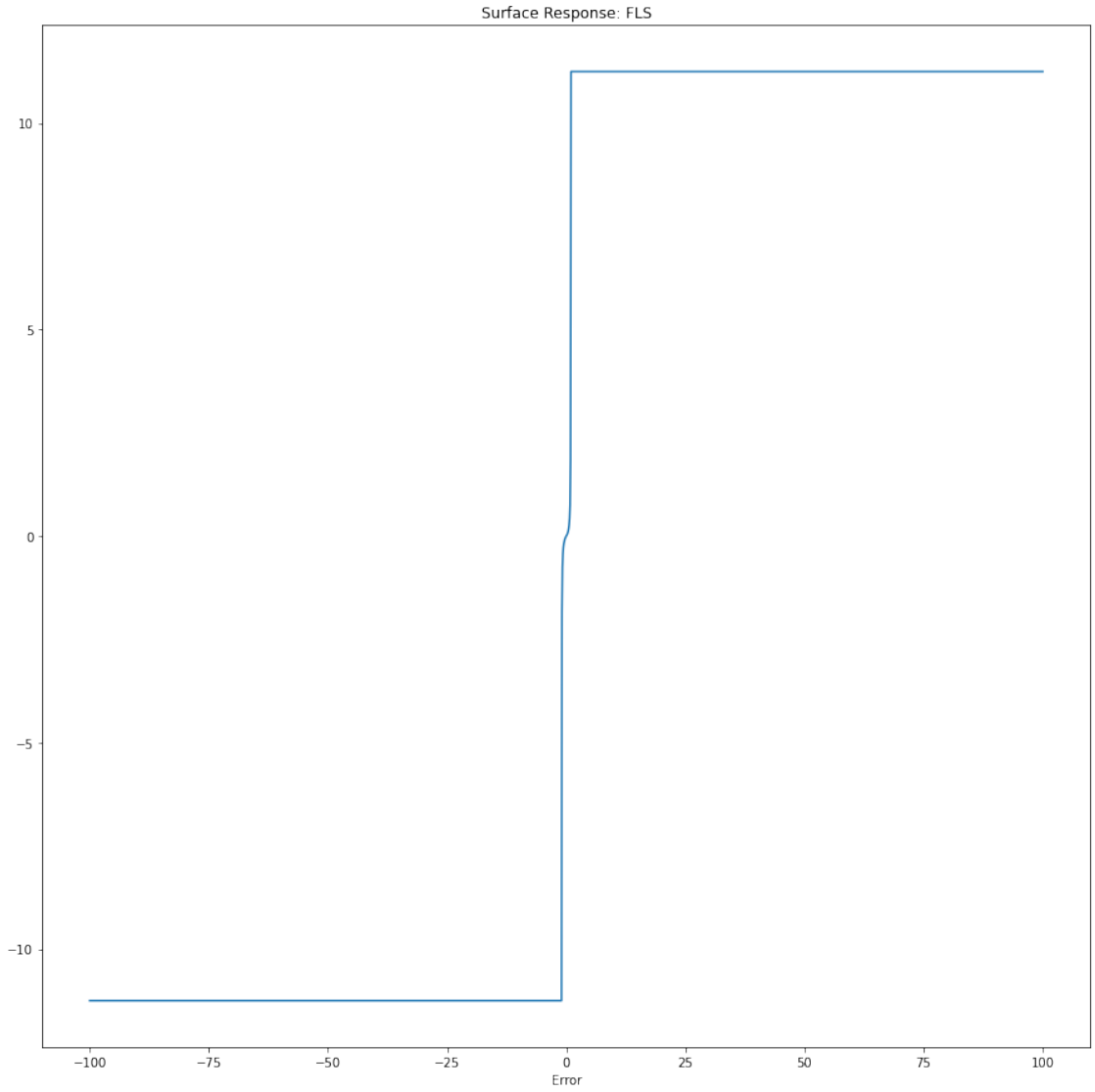
```
Control_universe.add_fuzzyset('negative','trapmf',[-20,-20,-5,0])
Control_universe.add_fuzzyset('zero','trimf',[-5,-0,5])
Control_universe.add_fuzzyset('positive','trapmf',[0,5,20,20])
Control_universe.view_fuzzy()

FLS1 = inference_system('FLS')
FLS1.add_premise(Error_universe)
FLS1.add_consequence(Control_universe)
FLS1.add_rule([['Error','negative']],[],[['Control','negative']])
FLS1.add_rule([['Error','zero']],[],[['Control','zero']])
FLS1.add_rule([['Error','positive']],[],[['Control','positive']])

FLS1.configure('FLSmidth')

FLS1.build()
t.toc()

del Error_universe
del Control_universe
```


Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
zero
positive




```
Elapsed time is 0.453422 seconds.
```

[7]:
```
t.tic()
error_values = np.arange(-100,100.1,0.1)
FLS1.surface_fuzzy_system([error_values])
t.toc()
```

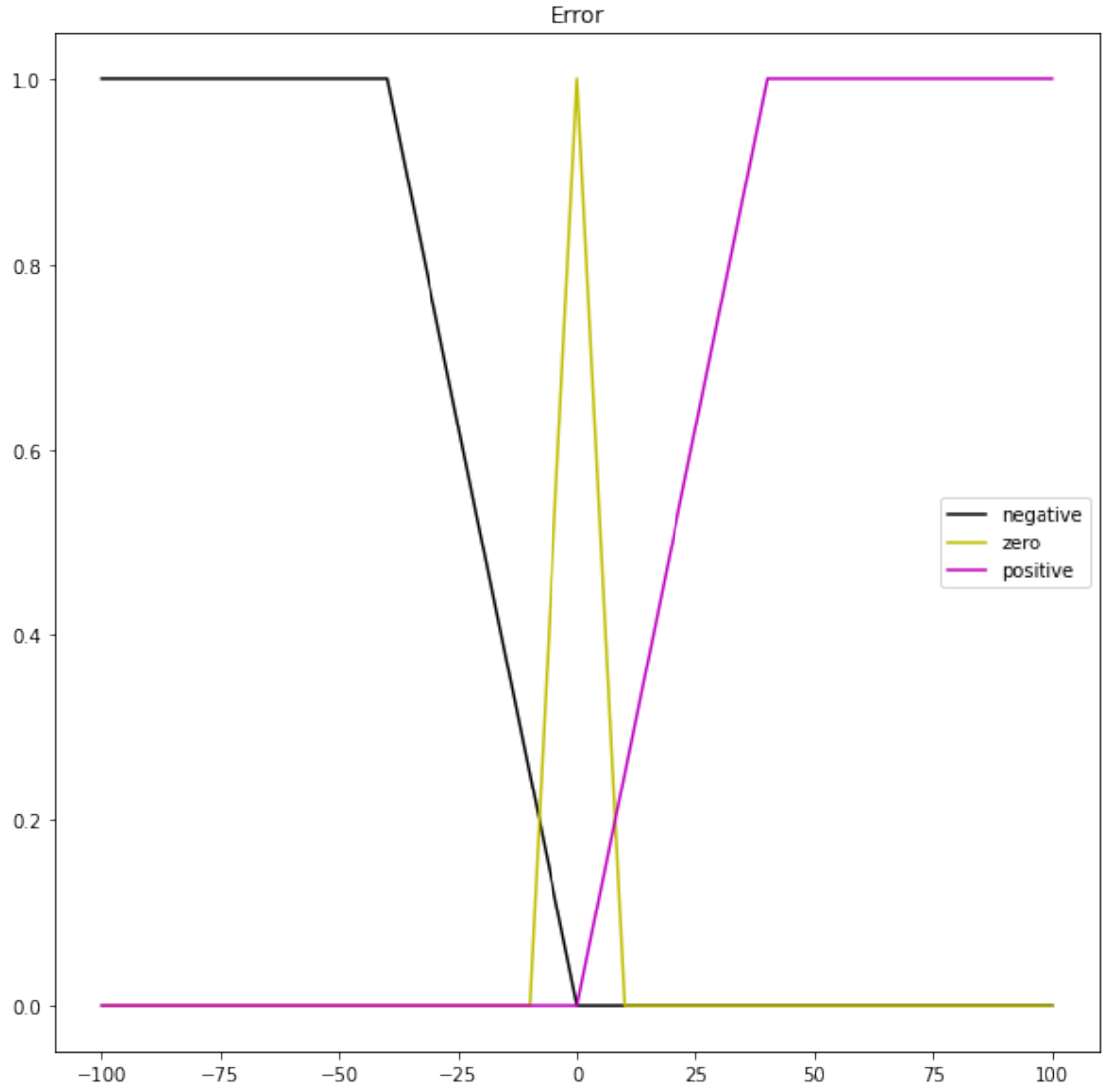


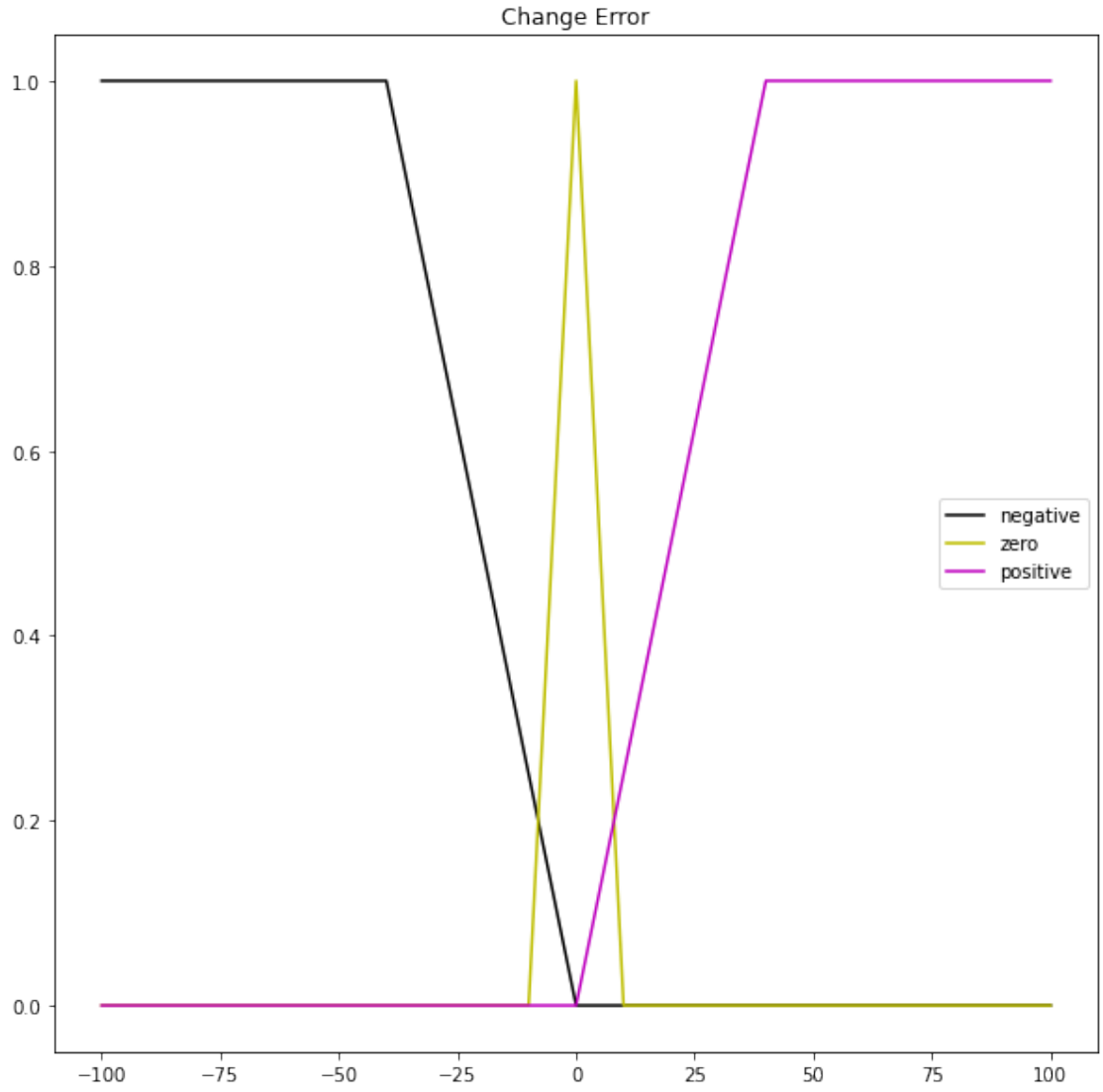

```
Elapsed time is 1.824471 seconds.
```

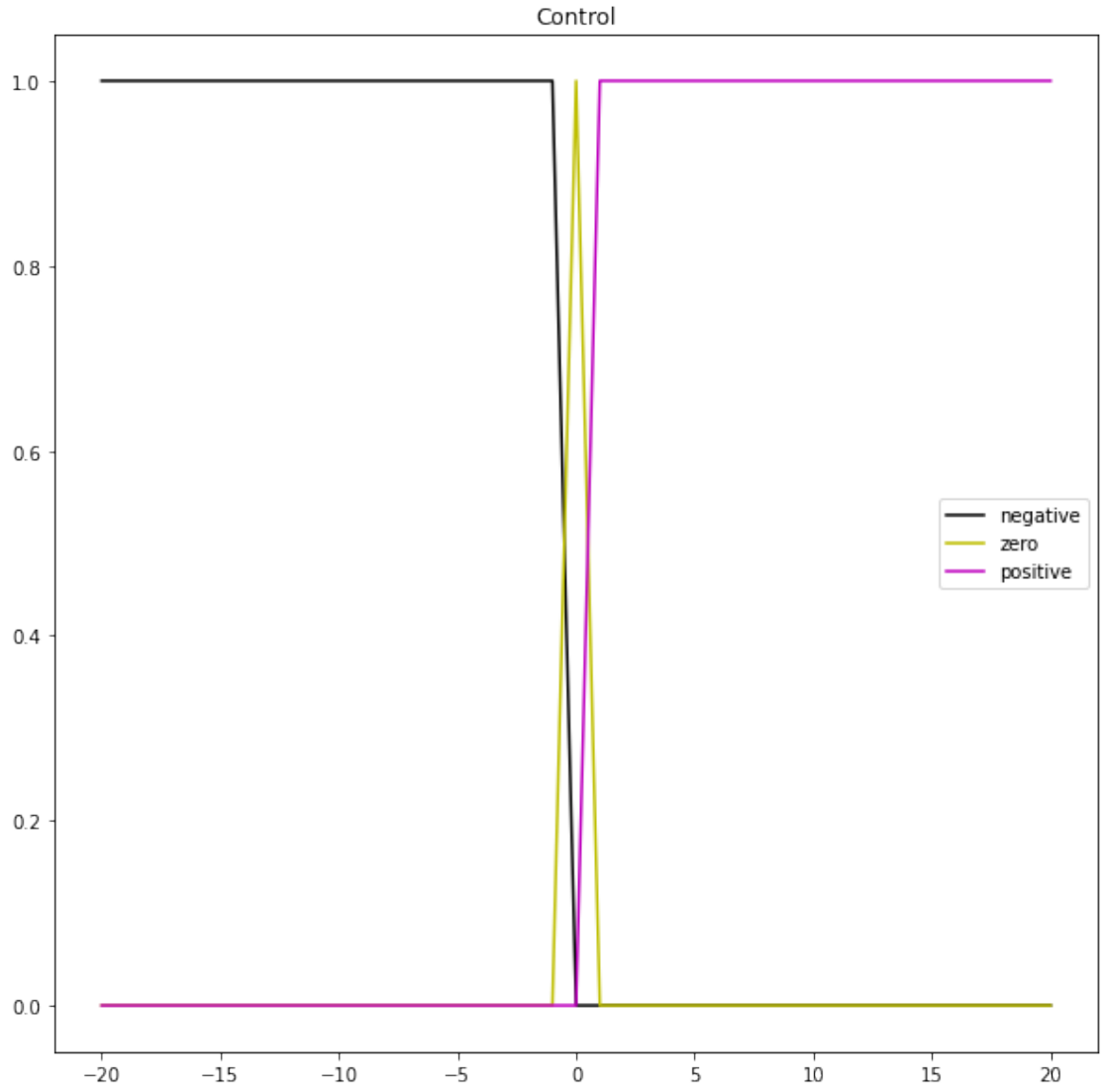
1.2.4 F.L. Smidth Inference System Two Inputs

[8]:
```
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trapmf',[-100,-100,-40,0])
Error_universe.add_fuzzyset('zero','trimf',[-10,0,10])
Error_universe.add_fuzzyset('positive','trapmf',[0,40,100,100])
Error_universe.view_fuzzy()

ChError_universe = fuzzy_universe('Change Error', np.arange(-100,101,1),␣
↪'continuous')
```

```
ChError_universe.add_fuzzyset('negative','trapmf',[-100,-100,-40,0])
ChError_universe.add_fuzzyset('zero','trimf',[-10,0,10])
ChError_universe.add_fuzzyset('positive','trapmf',[0,40,100,100])
ChError_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-20,21,1), 'continuous')
Control_universe.add_fuzzyset('negative','trapmf',[-20,-20,-0.5,0])
Control_universe.add_fuzzyset('zero','trimf',[-0.01,0,0.01])
Control_universe.add_fuzzyset('positive','trapmf',[0,0.5,20,20])
Control_universe.view_fuzzy()

FLS2 = inference_system('FLSmidth')
FLS2.add_premise(Error_universe)
FLS2.add_premise(ChError_universe)
FLS2.add_consequence(Control_universe)

FLS2.add_rule([['Error','negative'],['Change␣
 ↪Error','negative']],['and'],[['Control','negative']])
FLS2.add_rule([['Error','negative'],['Change␣
 ↪Error','zero']],['and'],[['Control','negative']])
FLS2.add_rule([['Error','zero'],['Change␣
 ↪Error','negative']],['and'],[['Control','zero']])
FLS2.add_rule([['Error','negative'],['Change␣
 ↪Error','positive']],['and'],[['Control','zero']])
FLS2.add_rule([['Error','zero'],['Change␣
 ↪Error','zero']],['and'],[['Control','zero']])
FLS2.add_rule([['Error','positive'],['Change␣
 ↪Error','negative']],['and'],[['Control','zero']])
FLS2.add_rule([['Error','zero'],['Change␣
 ↪Error','positive']],['and'],[['Control','zero']])
FLS2.add_rule([['Error','positive'],['Change␣
 ↪Error','zero']],['and'],[['Control','positive']])
FLS2.add_rule([['Error','positive'],['Change␣
 ↪Error','positive']],['and'],[['Control','positive']])


FLS2.configure('FLSmidth')

FLS2.build()
t.toc()

del Error_universe
del ChError_universe
del Control_universe
```


Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
zero
positive

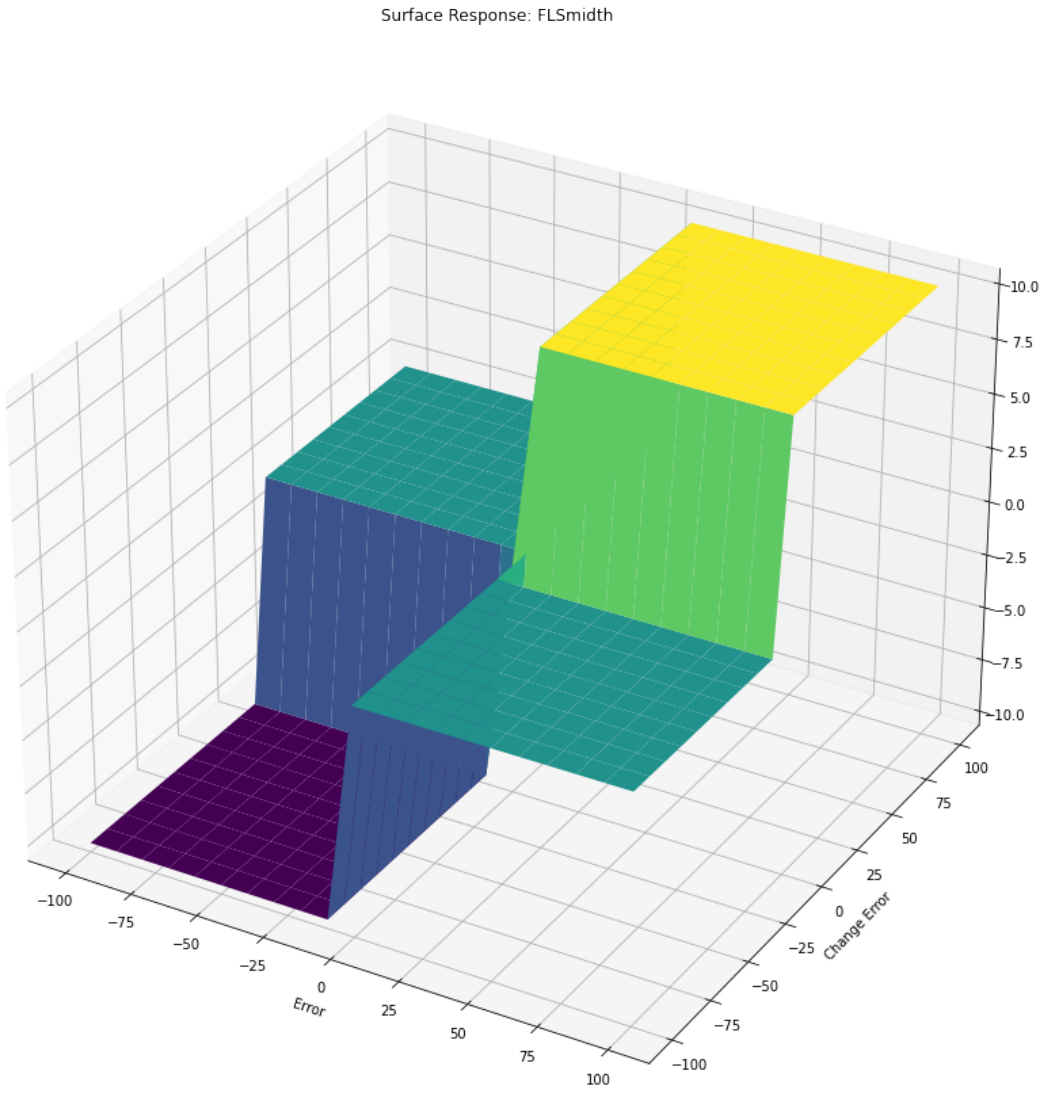

Change Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
zero
positive




```
Elapsed time is 0.734104 seconds.
```

[9]:
```
t.tic()
error_values = np.arange(-100,110,10)
change_error = np.arange(-100,110,10)

FLS2.surface_fuzzy_system([error_values,change_error])
t.toc()
```

```
(21, 21)
(21, 21)
(21, 21)
```

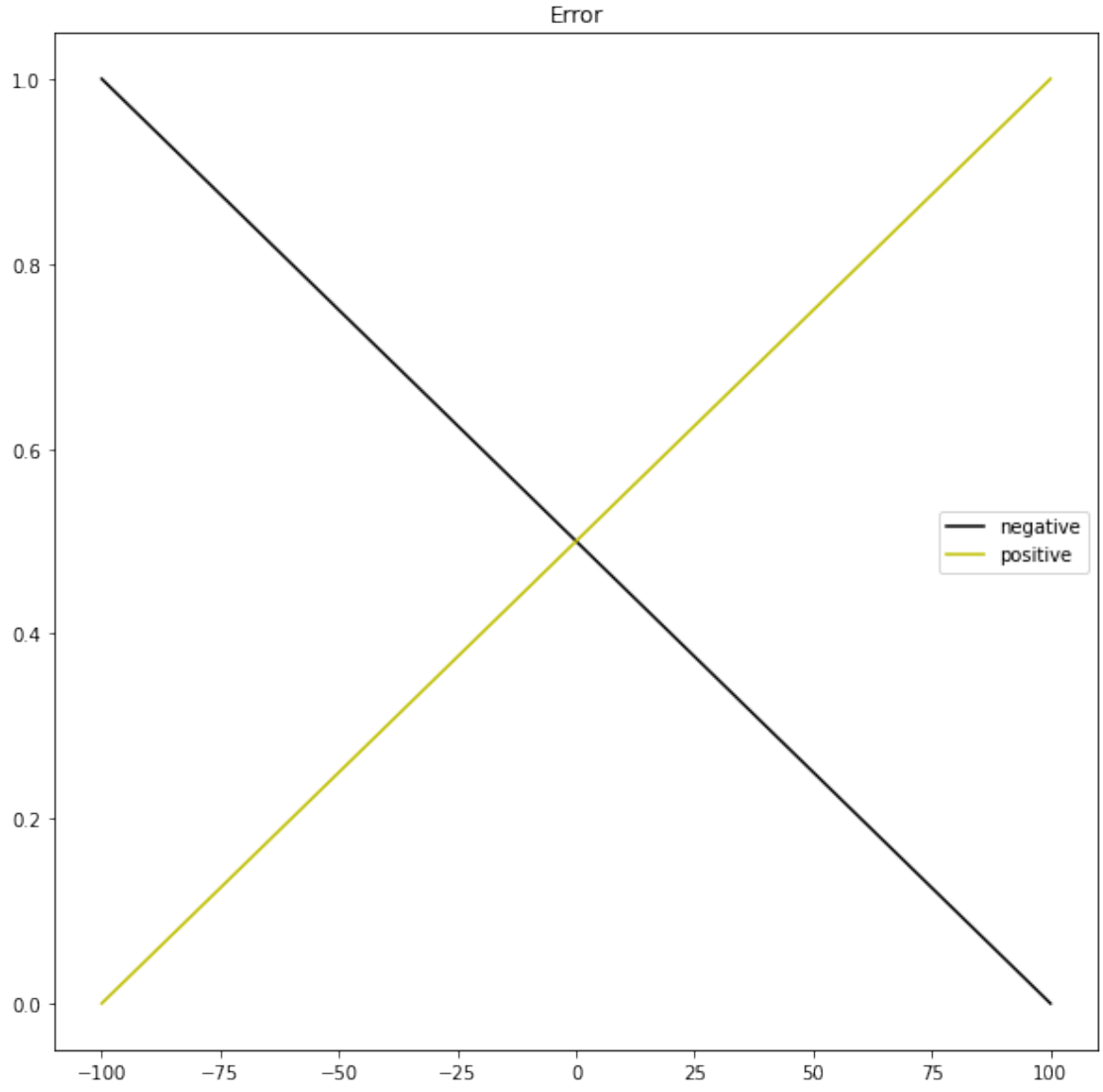


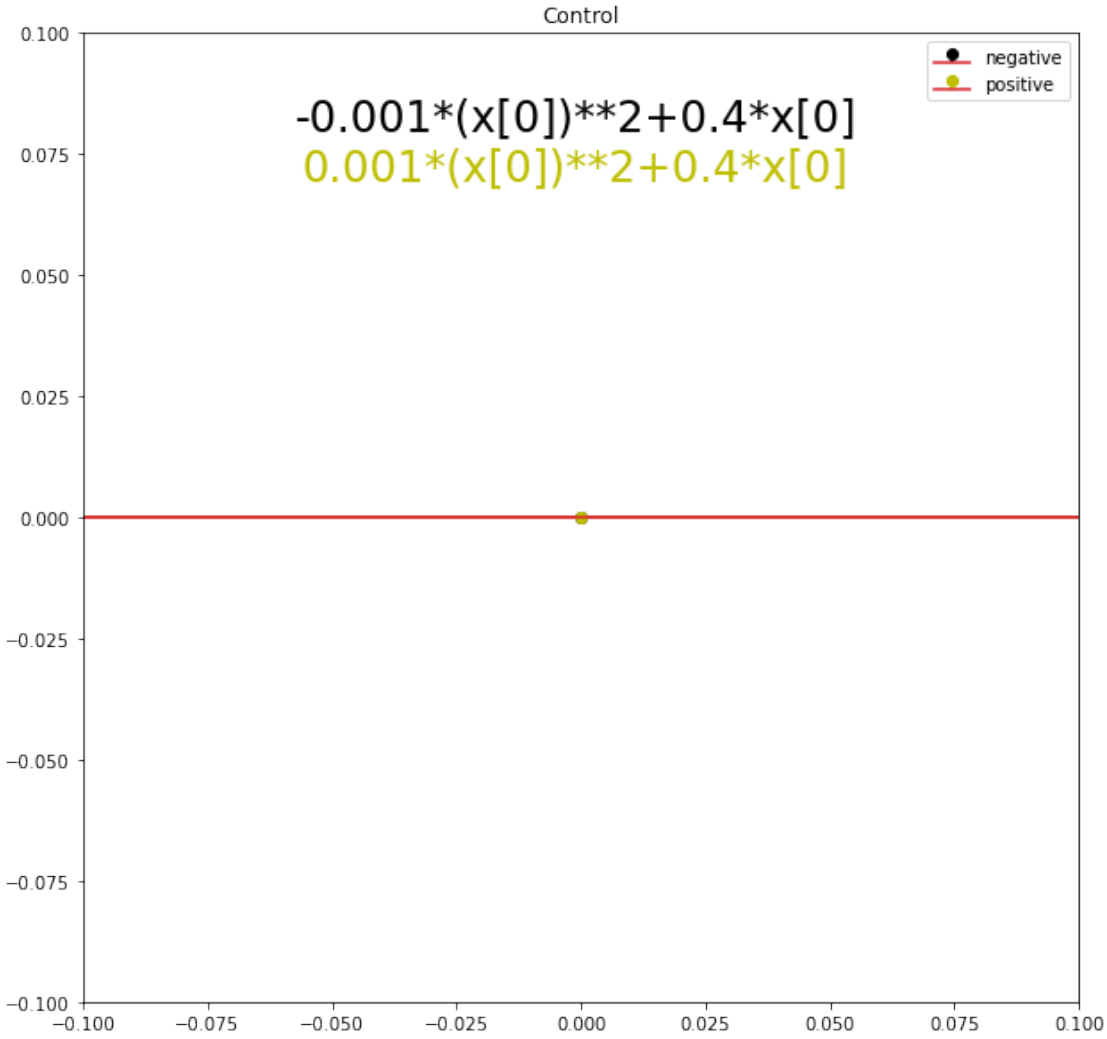

```
Elapsed time is 1.713117 seconds.
```

1.2.5 Takagi-Sugeno Inference System One Entry

[10]:
```
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trimf',[-200,-100,100])
Error_universe.add_fuzzyset('positive','trimf',[-100,100,200])
Error_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-20,22,2), 'continuous')
Control_universe.add_fuzzyset('negative','eq','-0.001*(x[0])**2+0.4*x[0]')
```

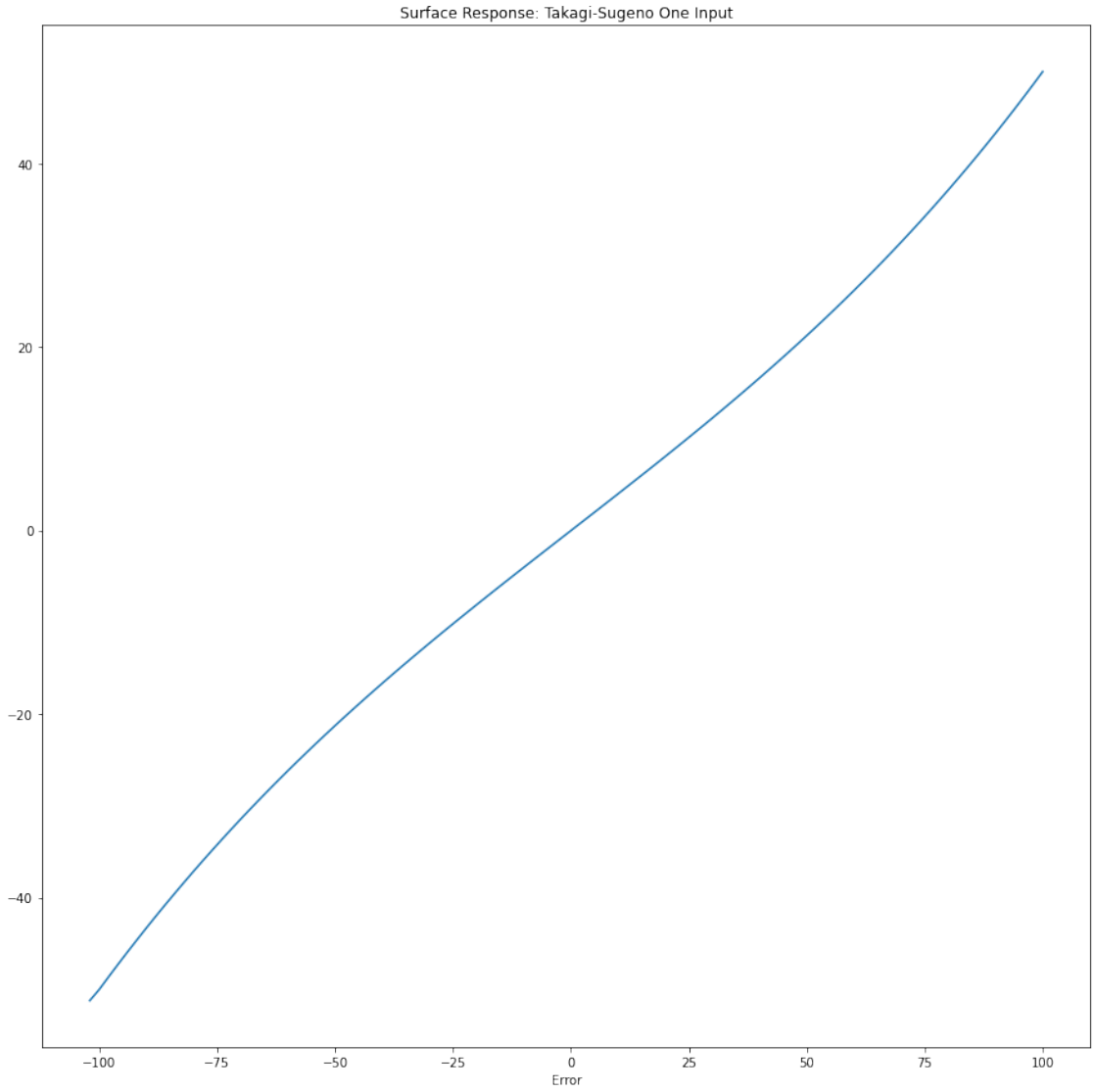
```
Control_universe.add_fuzzyset('positive','eq','0.001*(x[0])**2+0.4*x[0]')
Control_universe.view_fuzzy()

TSG1 = inference_system('Takagi-Sugeno One Input')
TSG1.add_premise(Error_universe)
TSG1.add_consequence(Control_universe)
TSG1.add_rule([['Error','negative']],[],[['Control','negative']])
TSG1.add_rule([['Error','positive']],[],[['Control','positive']])

TSG1.configure('Sugeno')

TSG1.build()
t.toc()
del Error_universe
del Control_universe
```






```
Elapsed time is 0.512254 seconds.
```

[11]:
```
t.tic()
error_values = np.arange(-102,102,2)
TSG1.surface_fuzzy_system([error_values])
t.toc()
```

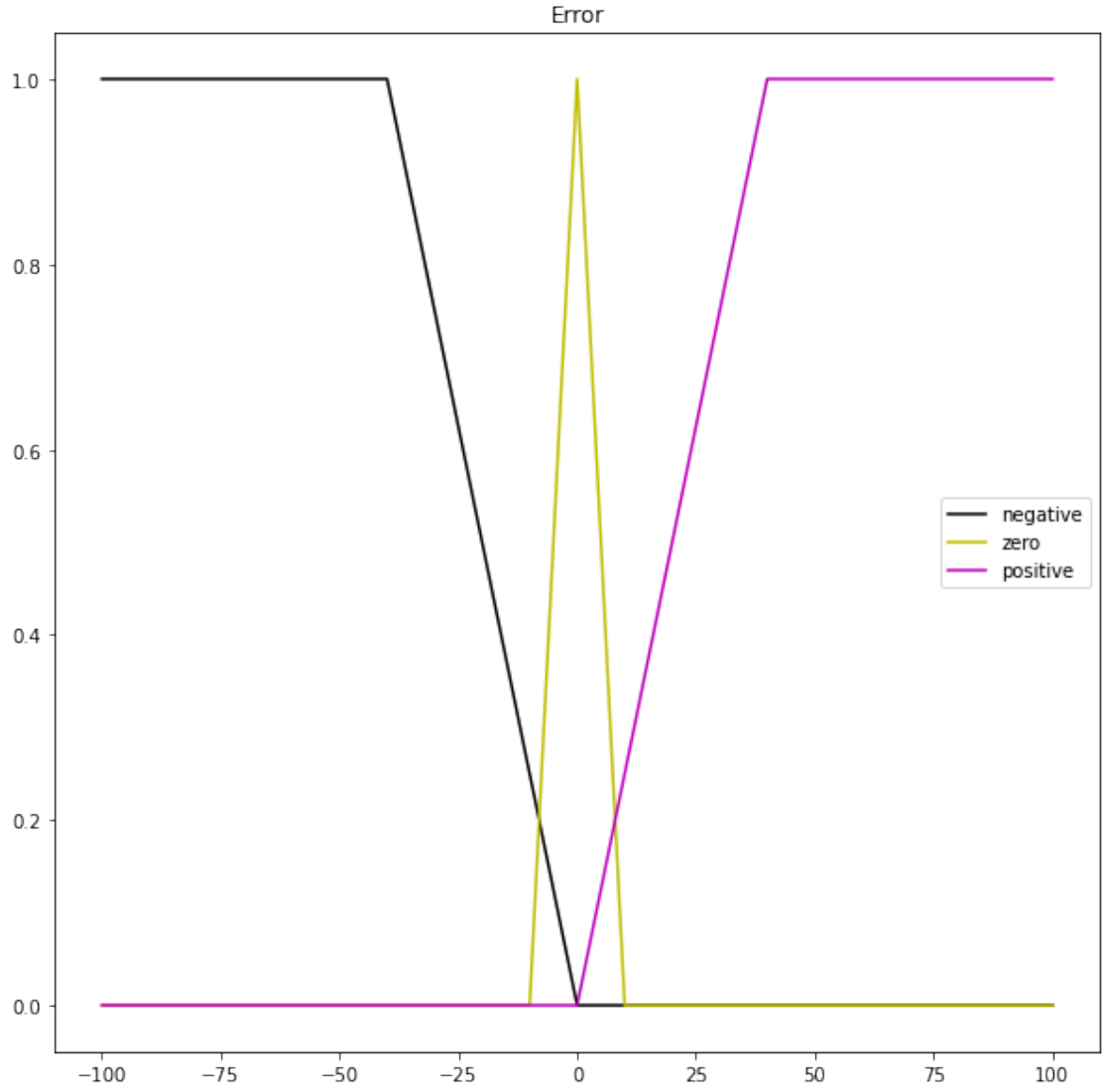


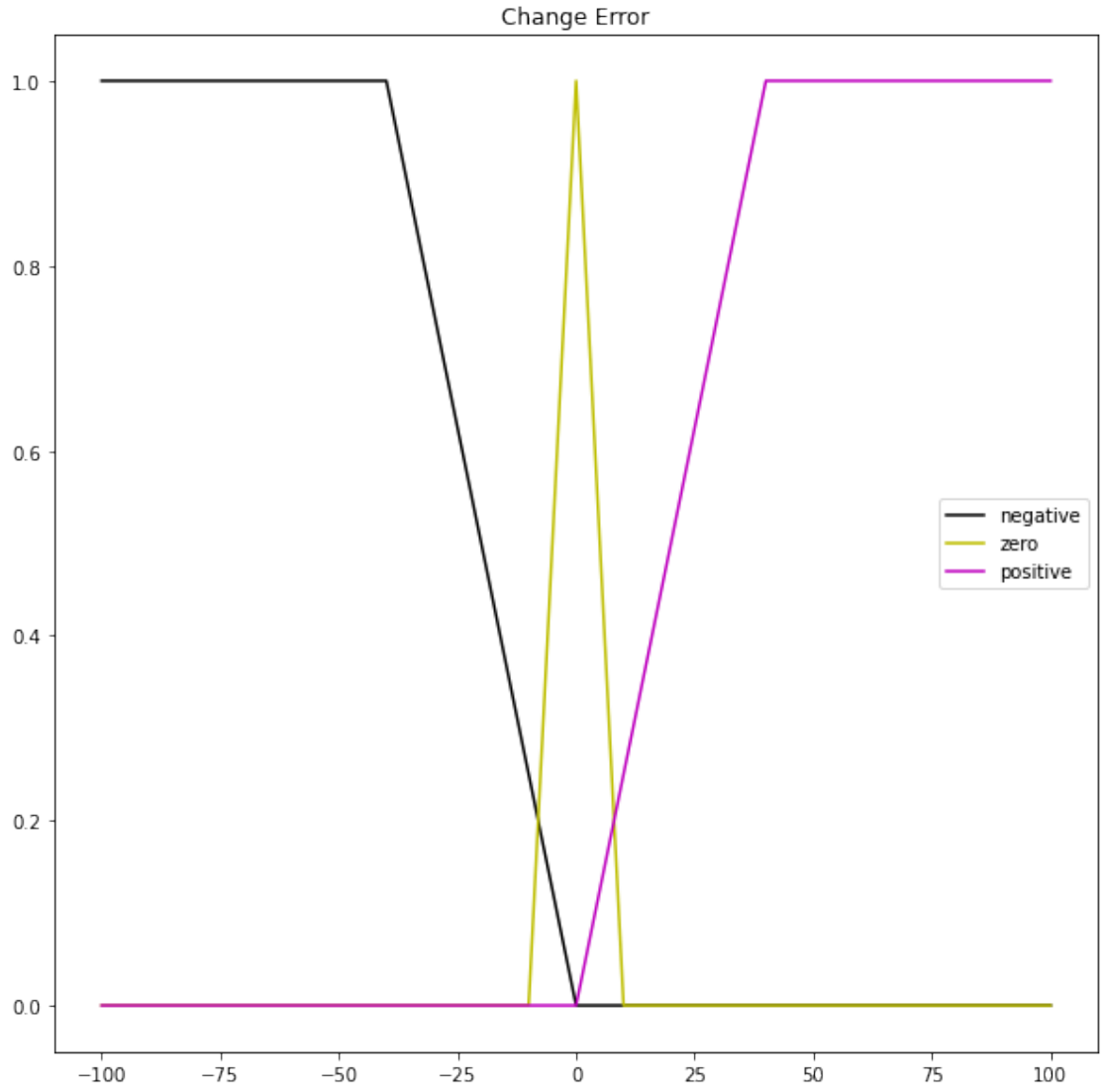

```
Elapsed time is 0.269573 seconds.
```

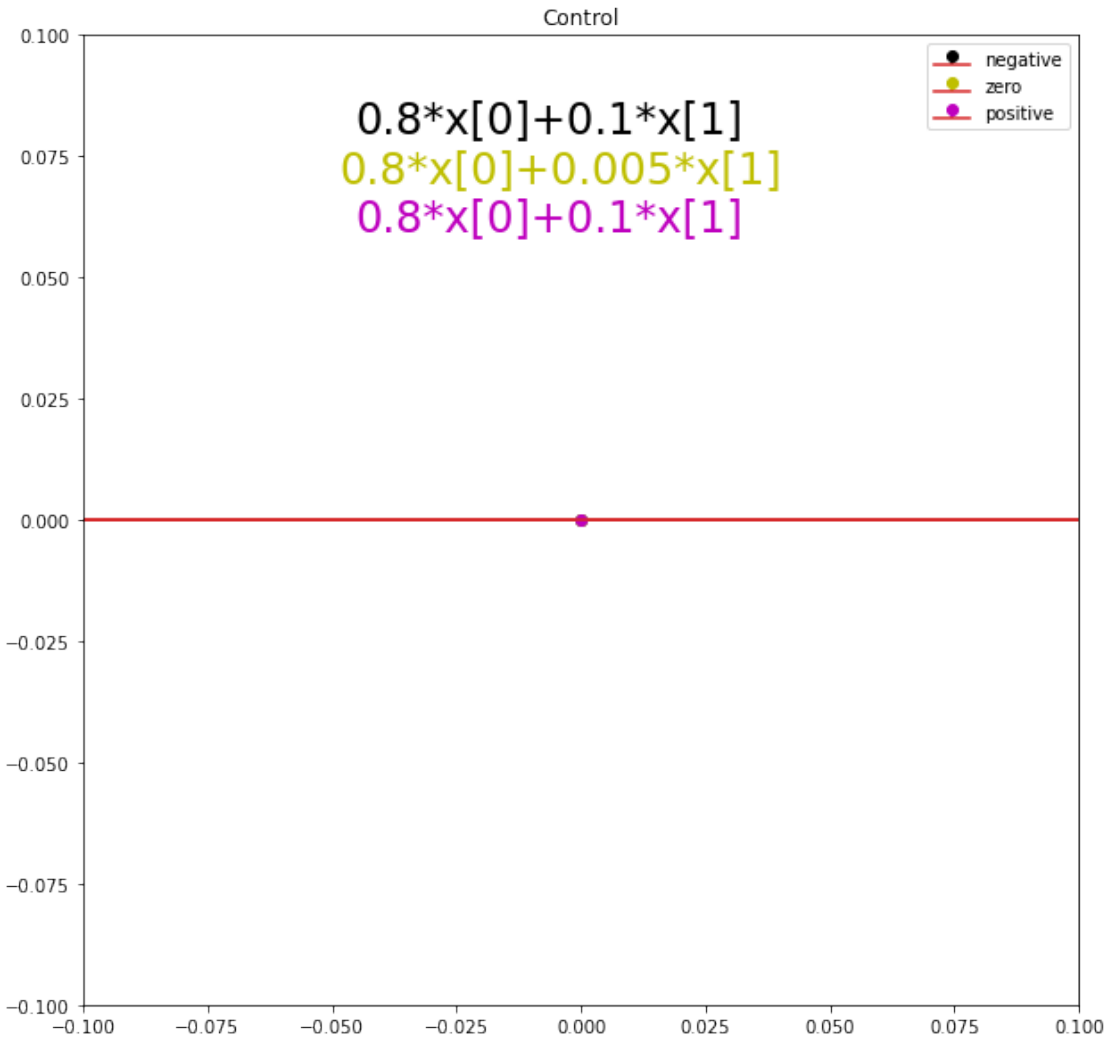
1.2.6 Takagi-Sugeno Inference System Two Inputs

[12]:
```
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trapmf',[-100,-100,-40,0])
Error_universe.add_fuzzyset('zero','trimf',[-10,0,10])
Error_universe.add_fuzzyset('positive','trapmf',[0,40,100,100])
Error_universe.view_fuzzy()

ChError_universe = fuzzy_universe('Change Error', np.arange(-100,101,1),␣
↪'continuous')
```

```
ChError_universe.add_fuzzyset('negative','trapmf',[-100,-100,-40,0])
ChError_universe.add_fuzzyset('zero','trimf',[-10,0,10])
ChError_universe.add_fuzzyset('positive','trapmf',[0,40,100,100])
ChError_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-20,22,2), 'continuous')
Control_universe.add_fuzzyset('negative','eq','0.8*x[0]+0.1*x[1]')
Control_universe.add_fuzzyset('zero','eq','0.8*x[0]+0.005*x[1]')
Control_universe.add_fuzzyset('positive','eq','0.8*x[0]+0.1*x[1]')
Control_universe.view_fuzzy()

TSG2 = inference_system('Takagi-Sugeno Two Inputs')
TSG2.add_premise(Error_universe)
TSG2.add_premise(ChError_universe)
TSG2.add_consequence(Control_universe)

TSG2.add_rule([['Error','negative'],['Change␣
 ↪Error','negative']],['and'],[['Control','negative']])
TSG2.add_rule([['Error','negative'],['Change␣
 ↪Error','zero']],['and'],[['Control','negative']])
TSG2.add_rule([['Error','zero'],['Change␣
 ↪Error','negative']],['and'],[['Control','zero']])
TSG2.add_rule([['Error','negative'],['Change␣
 ↪Error','positive']],['and'],[['Control','zero']])
TSG2.add_rule([['Error','zero'],['Change␣
 ↪Error','zero']],['and'],[['Control','zero']])
TSG2.add_rule([['Error','positive'],['Change␣
 ↪Error','negative']],['and'],[['Control','zero']])
TSG2.add_rule([['Error','zero'],['Change␣
 ↪Error','positive']],['and'],[['Control','zero']])
TSG2.add_rule([['Error','positive'],['Change␣
 ↪Error','zero']],['and'],[['Control','positive']])
TSG2.add_rule([['Error','positive'],['Change␣
 ↪Error','positive']],['and'],[['Control','positive']])


TSG2.configure('Sugeno')

TSG2.build()

t.toc()

del Error_universe
del Control_universe
```


Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
zero
positive

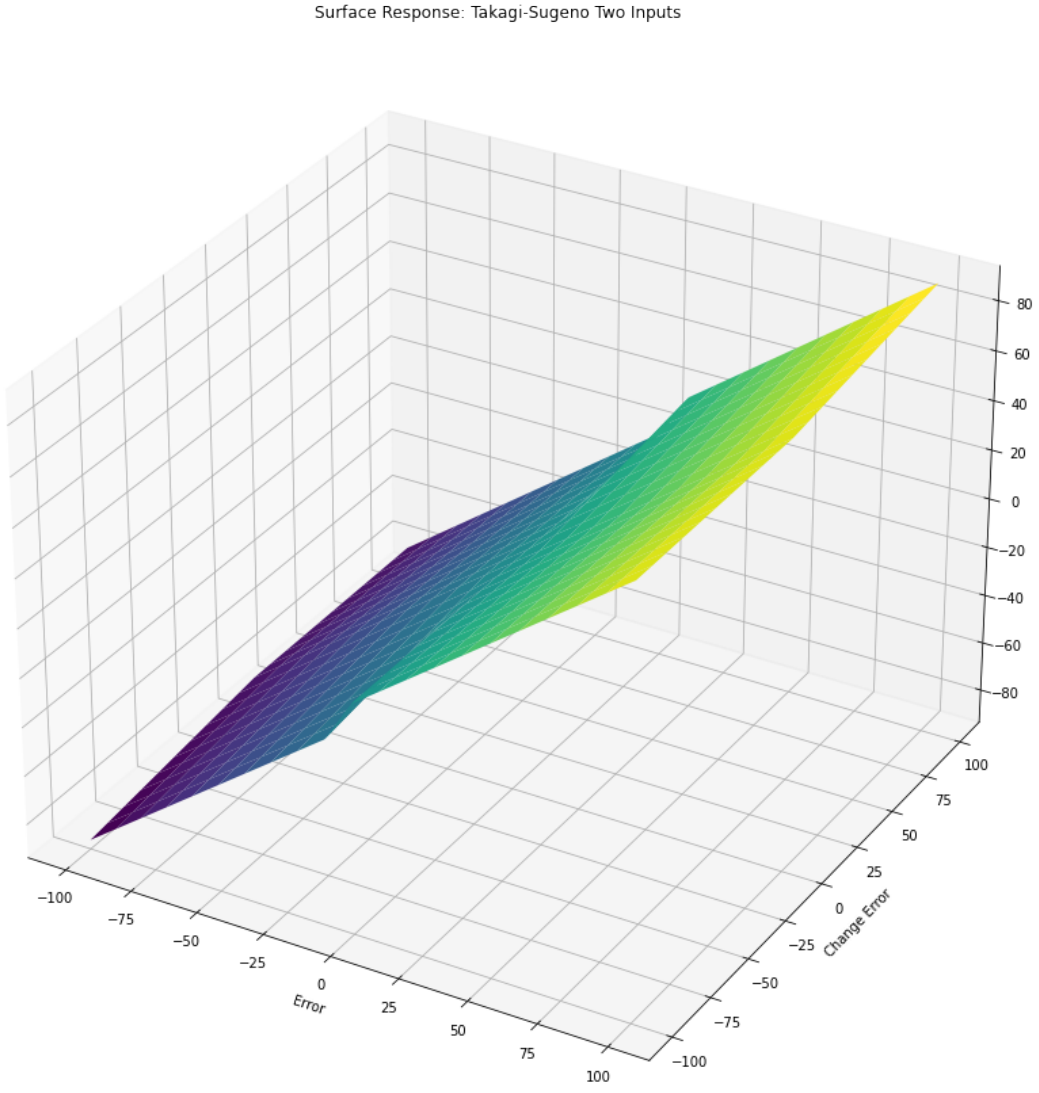

Change Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
zero
positive




```
Elapsed time is 0.810194 seconds.
```

[13]:
```
t.tic()
error_values = np.arange(-100,110,10)
change_error = np.arange(-100,110,10)

TSG2.surface_fuzzy_system([error_values,change_error])
t.toc()
```

```
(21, 21)
(21, 21)
(21, 21)
```

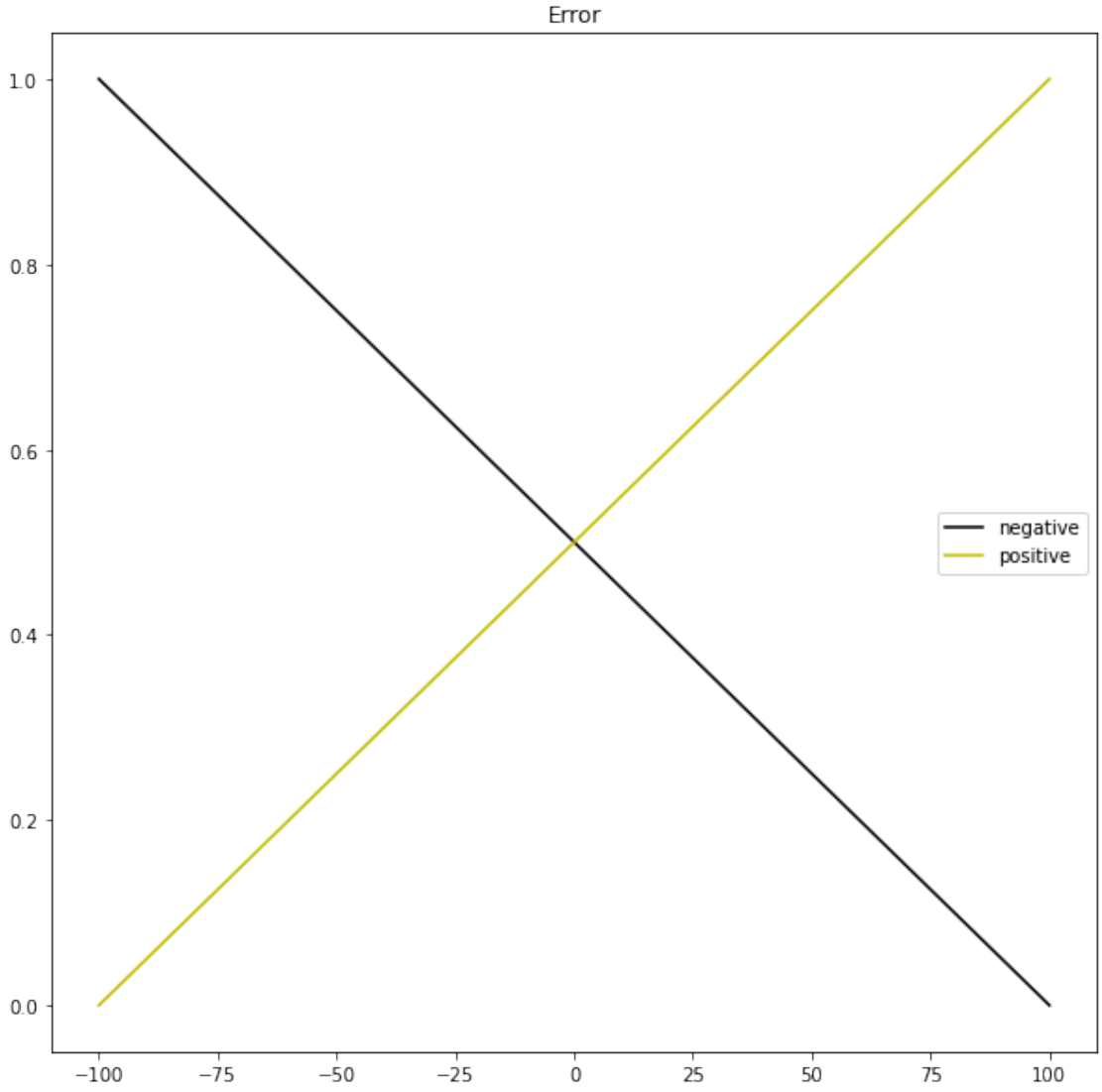


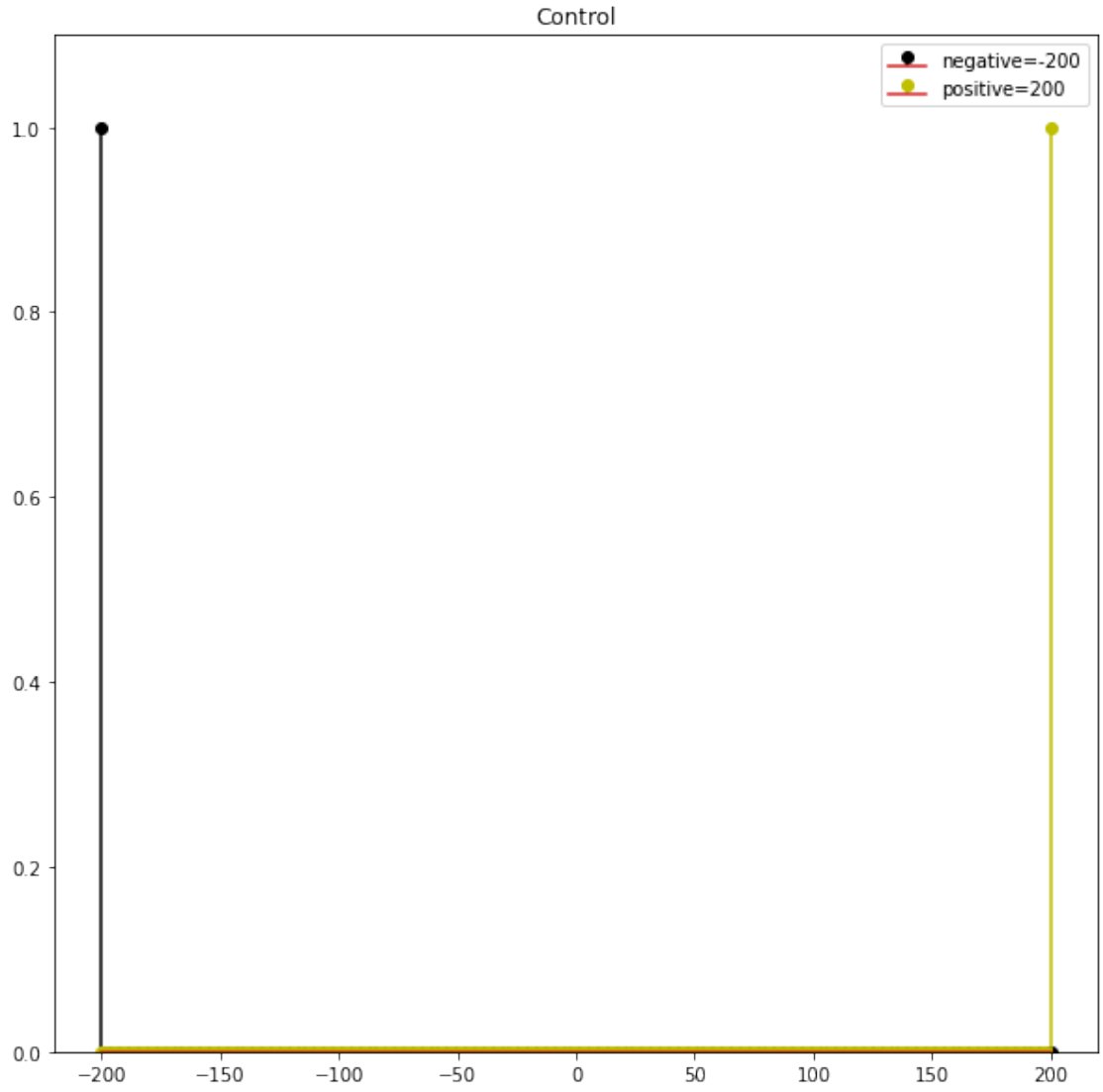

```
Elapsed time is 2.208288 seconds.
```

1.2.7 Sistema Difuso Lineal Una Entrada

[14]:
```
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trimf',[-200,-100,100])
Error_universe.add_fuzzyset('positive','trimf',[-100,100,200])
Error_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-200,202,2),␣
 ↪'continuous')
Control_universe.add_fuzzyset('negative','eq','-200')
```

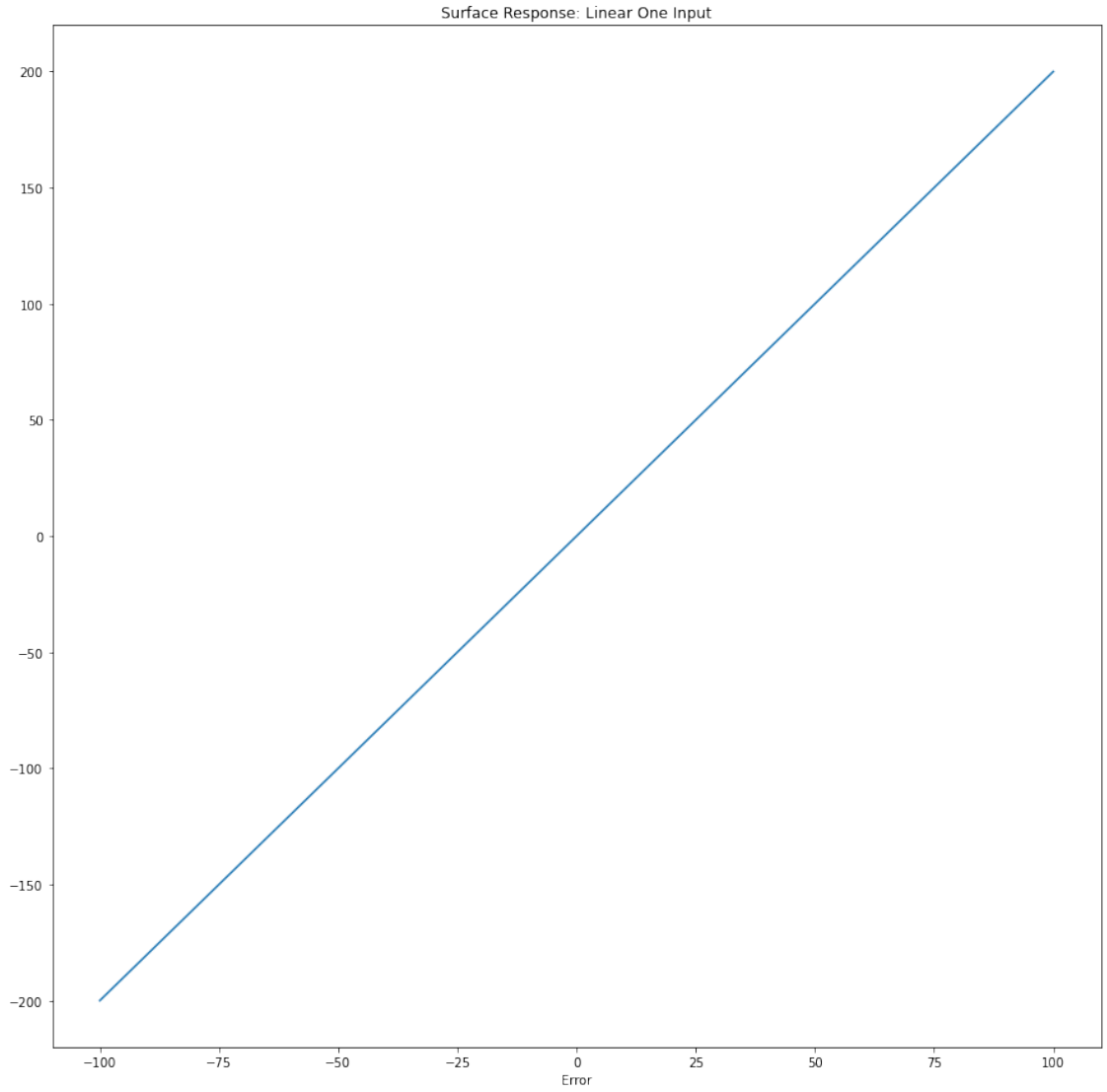
```
Control_universe.add_fuzzyset('positive','eq','200')
Control_universe.view_fuzzy()

LinearP = inference_system('Linear One Input')
LinearP.add_premise(Error_universe)
LinearP.add_consequence(Control_universe)
LinearP.add_rule([['Error','negative']],[],[['Control','negative']])
LinearP.add_rule([['Error','positive']],[],[['Control','positive']])

LinearP.configure('Linear')

LinearP.build()
t.toc()

del Error_universe
del Control_universe
```


Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
positive




```
Elapsed time is 0.429529 seconds.
```

[15]:
```
t.tic()
error_values = np.arange(-100,102,2)
LinearP.surface_fuzzy_system([error_values])
t.toc()
```

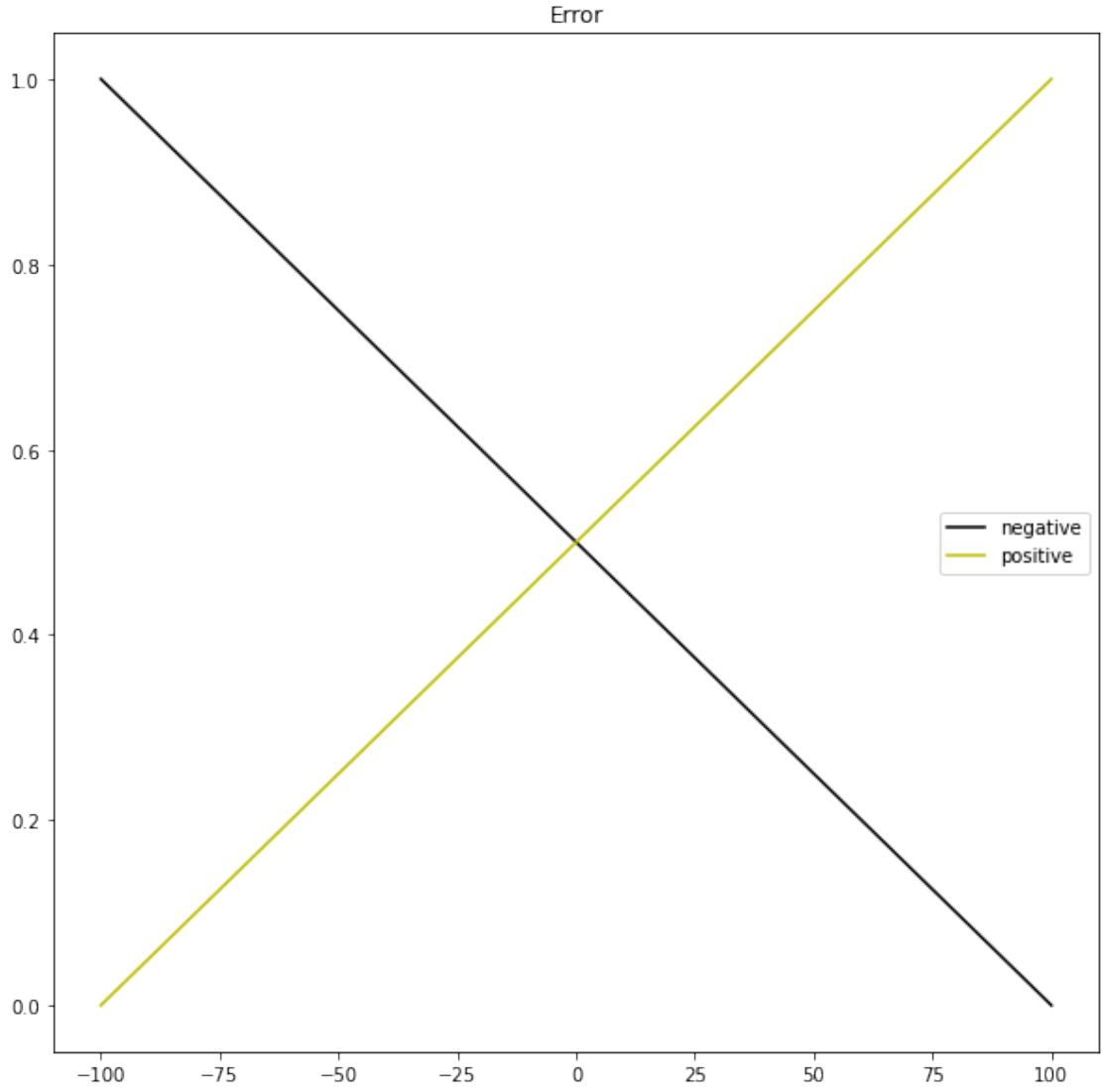


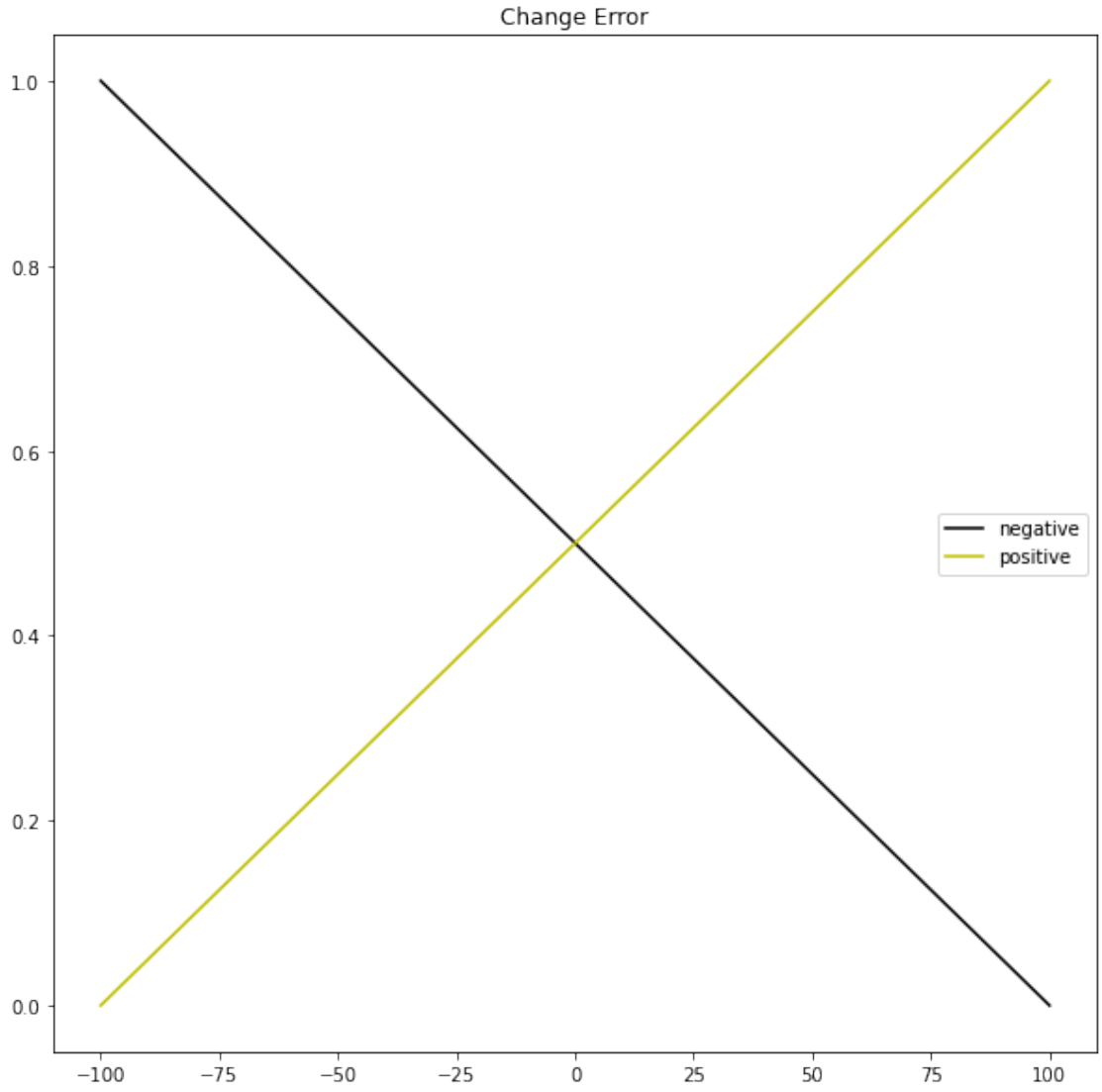

```
Elapsed time is 0.274371 seconds.
```

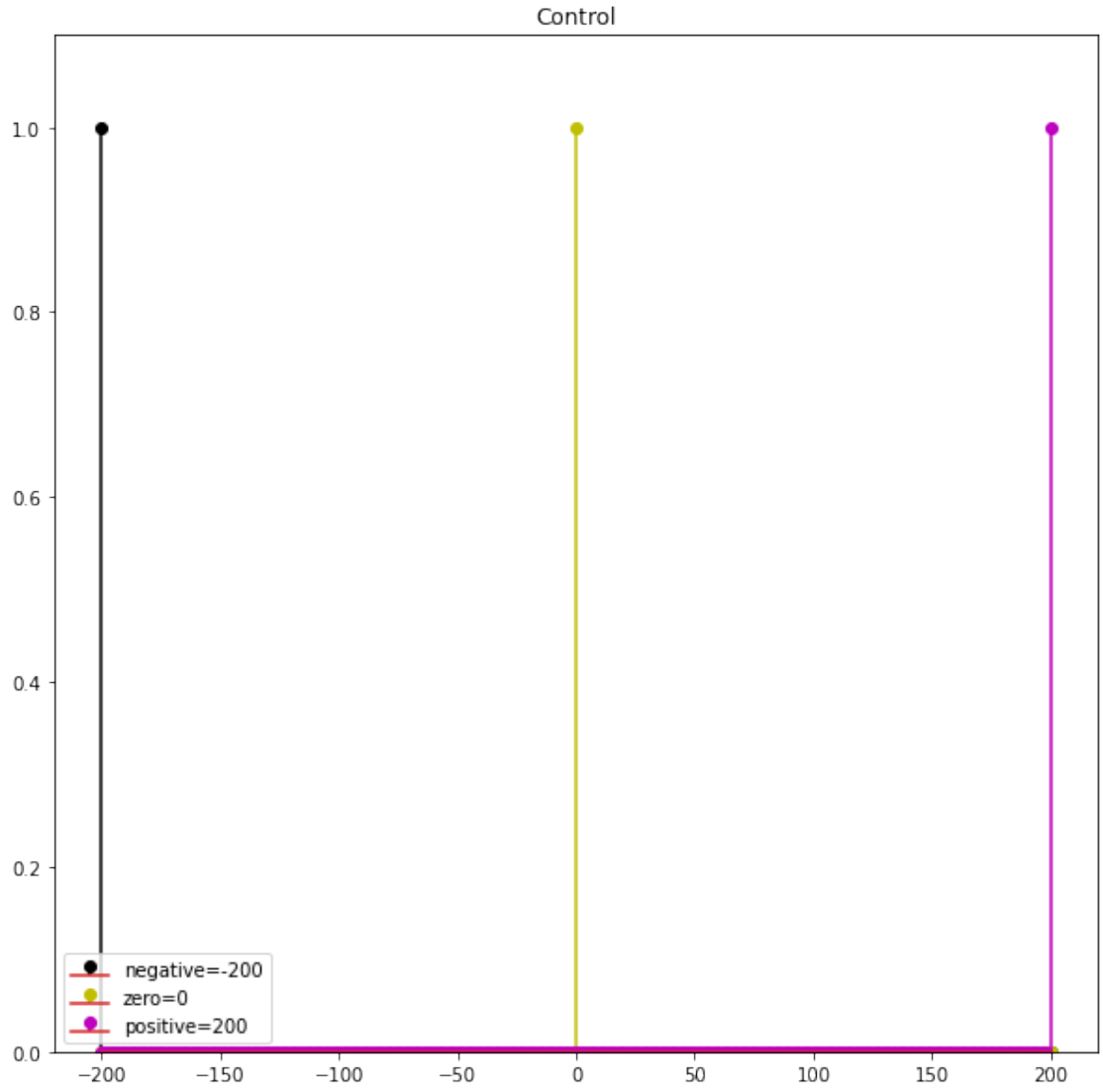
1.2.8 Linear Fuzzy System Two Inputs

[16]:
```
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trimf',[-200,-100,100])
Error_universe.add_fuzzyset('positive','trimf',[-100,100,200])
Error_universe.view_fuzzy()

ChError_universe = fuzzy_universe('Change Error', np.arange(-100,101,1),␣
↪'continuous')
ChError_universe.add_fuzzyset('negative','trimf',[-200,-100,100])
```

```
ChError_universe.add_fuzzyset('positive','trimf',[-100,100,200])
ChError_universe.view_fuzzy()

Control_universe = fuzzy_universe('Control', np.arange(-200,202,2),␣
 ↪'continuous')
Control_universe.add_fuzzyset('negative','eq','-200')
Control_universe.add_fuzzyset('zero','eq','0')
Control_universe.add_fuzzyset('positive','eq','200')
Control_universe.view_fuzzy()

Linear = inference_system('Linear')
Linear.add_premise(Error_universe)
Linear.add_premise(ChError_universe)
Linear.add_consequence(Control_universe)

Linear.add_rule([['Error','negative'],['Change␣
 ↪Error','negative']],['and'],[['Control','negative']])
Linear.add_rule([['Error','negative'],['Change␣
 ↪Error','positive']],['and'],[['Control','zero']])
Linear.add_rule([['Error','positive'],['Change␣
 ↪Error','negative']],['and'],[['Control','zero']])
Linear.add_rule([['Error','positive'],['Change␣
 ↪Error','positive']],['and'],[['Control','positive']])

Linear.configure('Linear')

Linear.build()

t.toc()
del Error_universe
del ChError_universe
del Control_universe
```


Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
positive

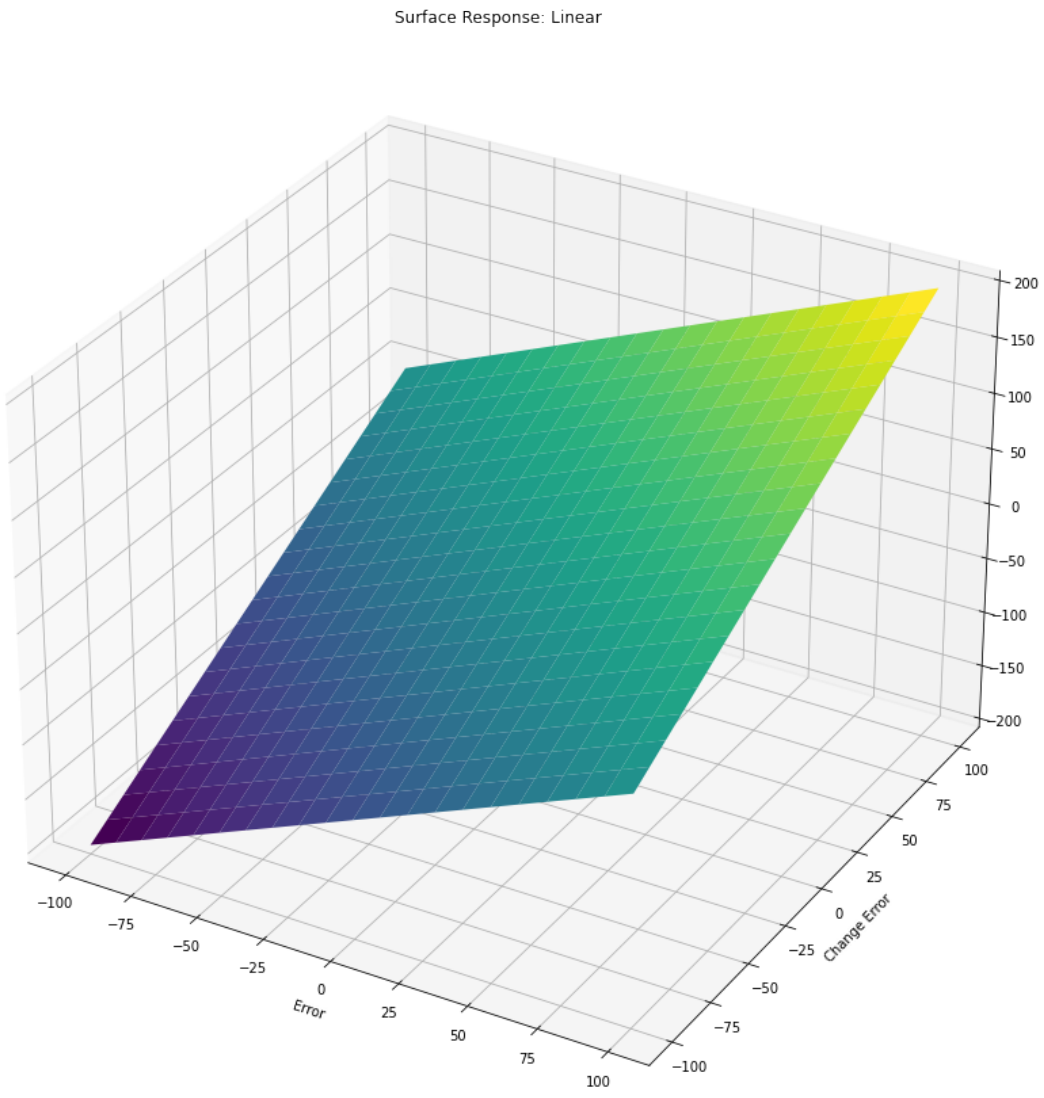

Change Error
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
negative
positive




```
Elapsed time is 0.666011 seconds.
```

[17]:
```
t.tic()
error_values = np.arange(-100,110,10)
change_error = np.arange(-100,110,10)

Linear.surface_fuzzy_system([error_values,change_error])
t.toc()
```

```
(21, 21)
(21, 21)
(21, 21)
```




```
Elapsed time is 0.644476 seconds.
```

1.3 Fuzzy Controllers

1.3.1 Definición del Sistema (Posición Motor CD)

[18]:
```
t.tic()
GE = 15.91545709
GCE = 0.636618283
GIE = 7.234298678
GU = 0.094248
DT = 0.001
```

```
J = 3.2284E-6;
b = 3.5077E-6;
K = 0.0274;
R = 4;
L = 2.75E-6;
te = 1.0
ns = 500
T=np.linspace(0,te,ns)
Input = [(np.radians(45)*min((t-0.25)/0.005,1)) if t> 0.25 else 0 for t in T]

s = control.TransferFunction.s
TF = K/(s*((J*s+b)*(L*s+R)+K**2))
t.toc()
```

```
Elapsed time is 0.007362 seconds.
```

1.3.2 Mamdani Controller One Input

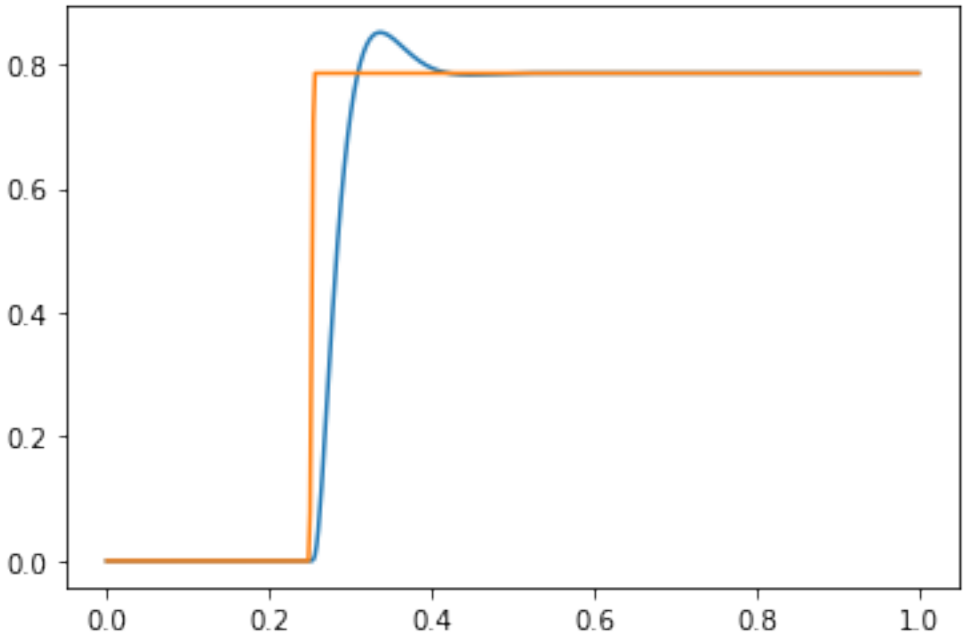
[19]:
```
t.tic()
PidFuzzController = fuzzy_controller(Mamdani1,typec='Fuzzy1',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.006109 seconds.
```

[20]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 2.959247 seconds.
```

[21]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.151567 seconds.
```

1.3.3 Mamdani Controller Two Inputs

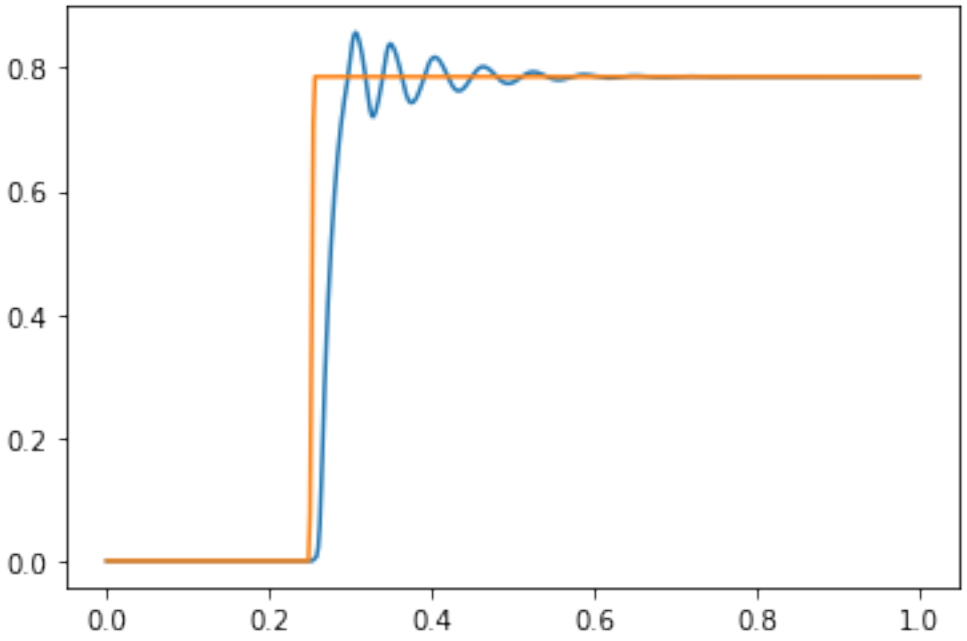
[22]:
```
t.tic()
PidFuzzController = fuzzy_controller(Mamdani2,typec='Fuzzy2',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.006568 seconds.
```

[23]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 10.240298 seconds.
```

[24]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.273470 seconds.
```

1.3.4 F.L. Smidth Controller One Input

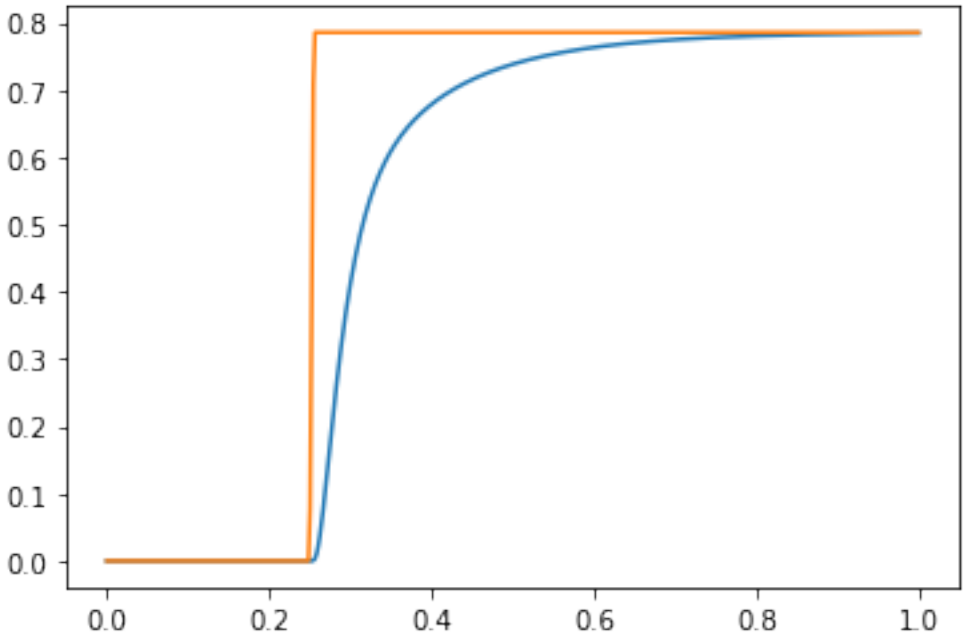
[25]:
```
t.tic()
PidFuzzController = fuzzy_controller(FLS1,typec='Fuzzy1',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.005494 seconds.
```

[26]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 3.359658 seconds.
```

[27]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.244372 seconds.
```

1.3.5 F.L. Smidth Controller Two Inputs

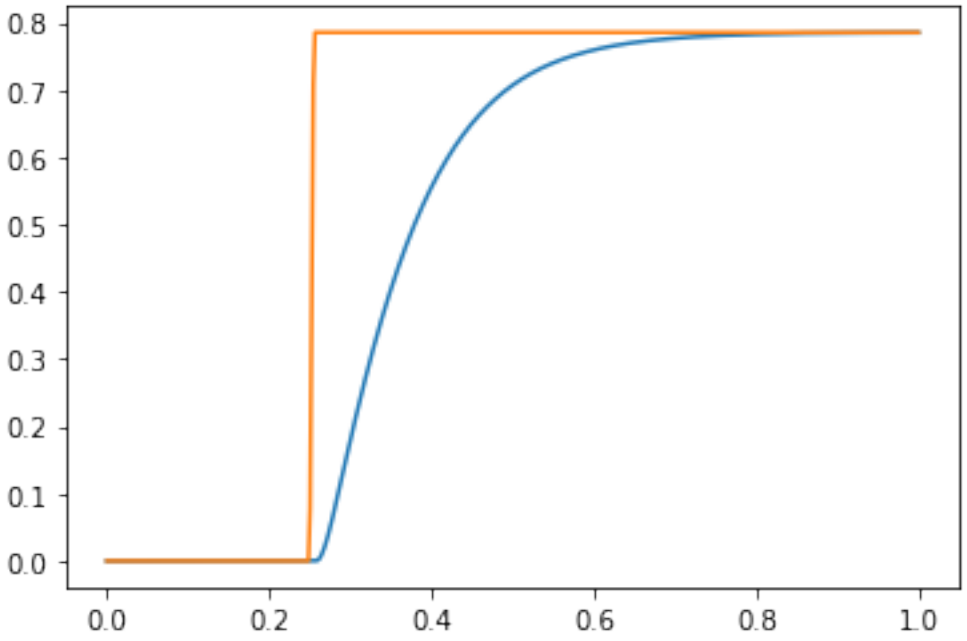
[28]:
```
t.tic()
PidFuzzController = fuzzy_controller(FLS2,typec='Fuzzy2',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.006110 seconds.
```

[29]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 7.303331 seconds.
```

[30]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.183430 seconds.
```

1.3.6 Takagi-Sugeno Controller One input

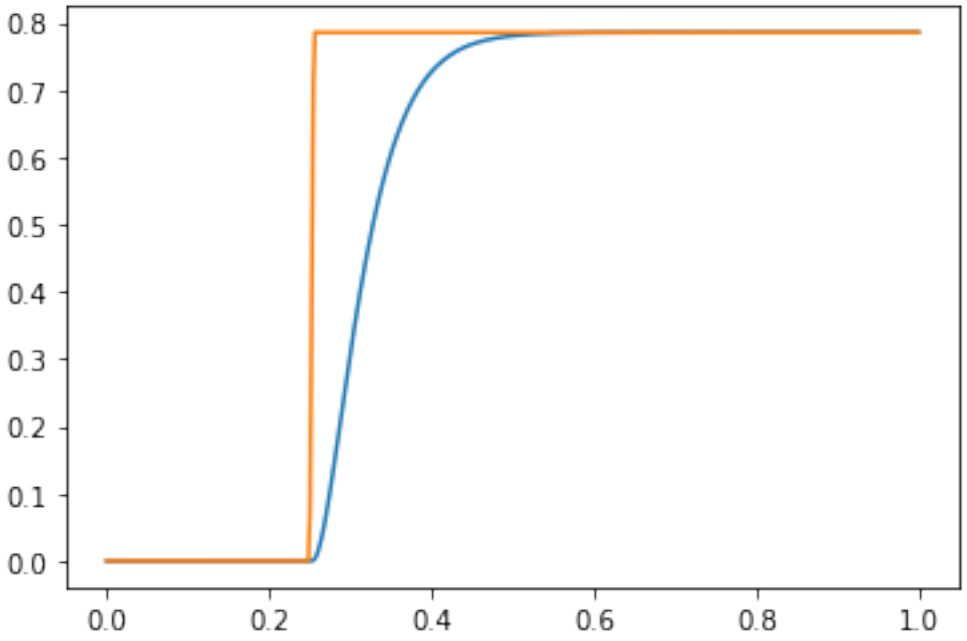
[31]:
```
t.tic()
PidFuzzController = fuzzy_controller(TSG1,typec='Fuzzy1',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.005211 seconds.
```

[32]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 0.951516 seconds.
```

[33]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.125511 seconds.
```

1.3.7 Takagi-Sugeno Controller Two Inputs

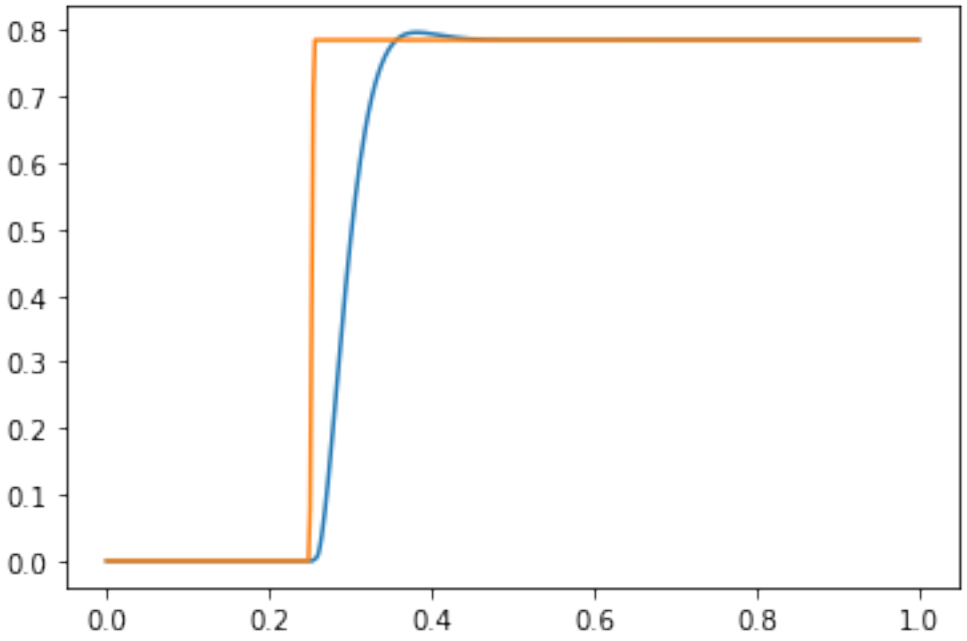
[34]:
```
t.tic()
PidFuzzController = fuzzy_controller(TSG2,typec='Fuzzy2',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.005840 seconds.
```

[35]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 7.664945 seconds.
```

[36]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.132986 seconds.
```

1.3.8 Linear Proportional Fuzzy Controller

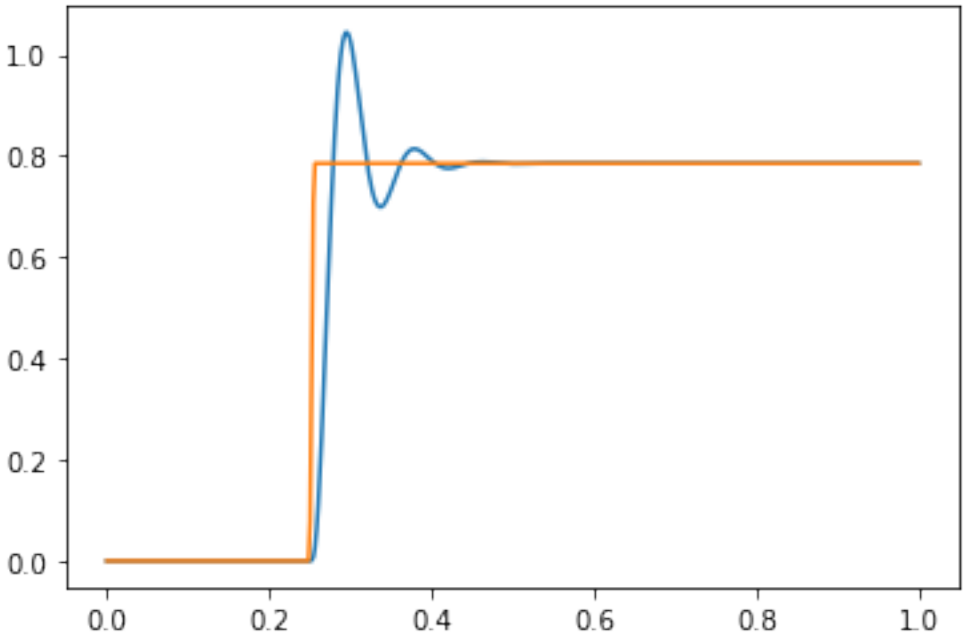
[37]:
```
t.tic()
PidFuzzController = fuzzy_controller(LinearP,typec='P',tf=TF,DT = T[1], GE=15.
 ↪91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.006447 seconds.
```

[38]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 0.975129 seconds.
```

[39]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.171029 seconds.
```

1.3.9 Derivative Linear Proportional Proportional Fuzzy Controller

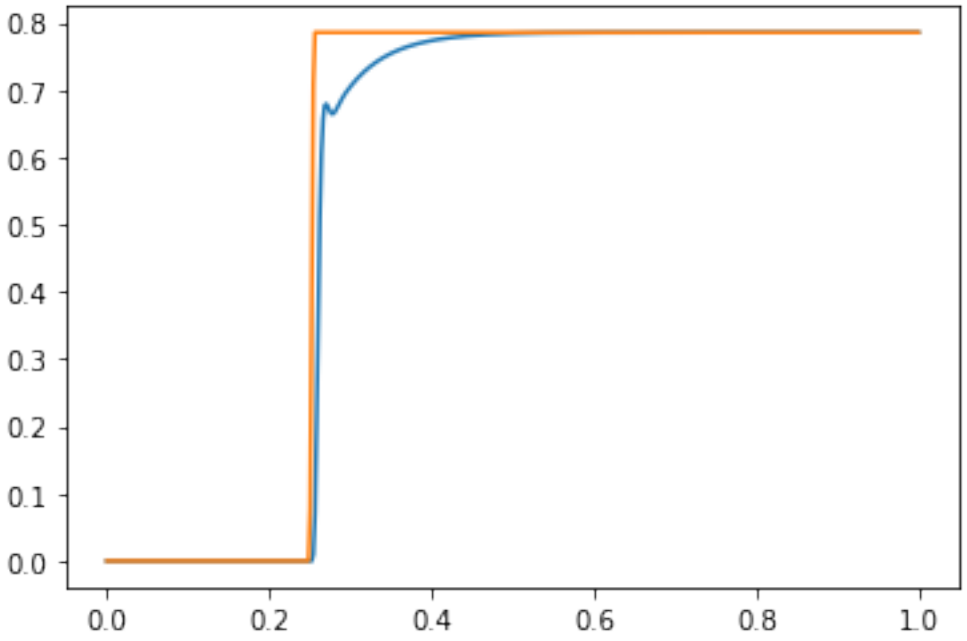
[40]:
```
t.tic()
PidFuzzController = fuzzy_controller(Linear,typec='PD',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248, GCE=0.636618283)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.008434 seconds.
```

[41]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 1.711767 seconds.
```

[42]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.138711 seconds.
```

1.3.10 Linear Proportional Derivative-Integral Fuzzy Proportional Controller

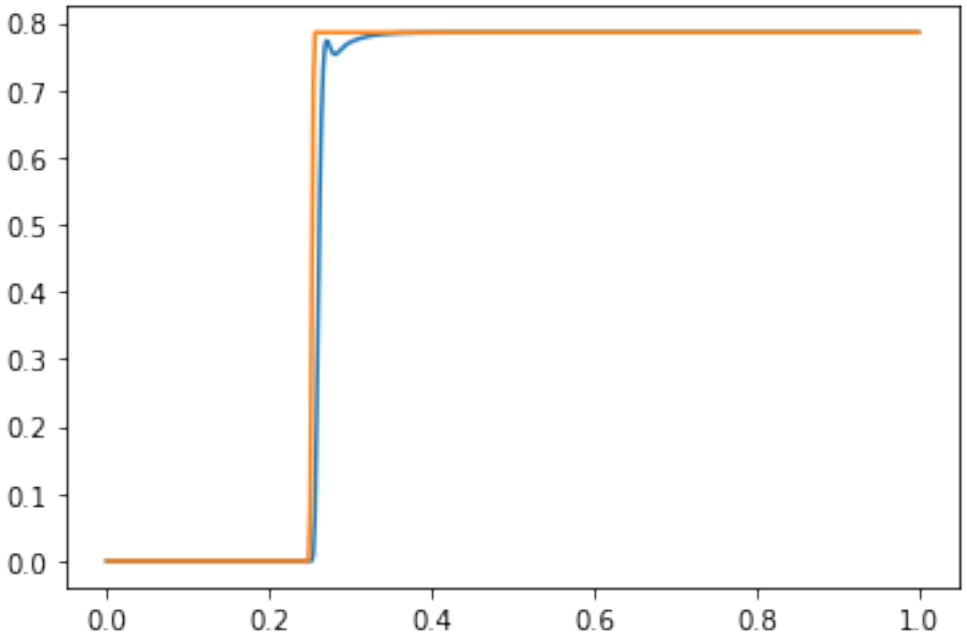
[43]:
```
t.tic()
PidFuzzController = fuzzy_controller(Linear,typec='PD-I',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248, GCE=0.636618283, GIE=7.234298678)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.008617 seconds.
```

[44]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 2.607848 seconds.
```

[45]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
tt.toc()
```




```
Elapsed time is 0.167314 seconds.
Elapsed time is 58.751305 seconds.
```

1.3.11 Mamdani Proportional Fuzzy Proportional Controller

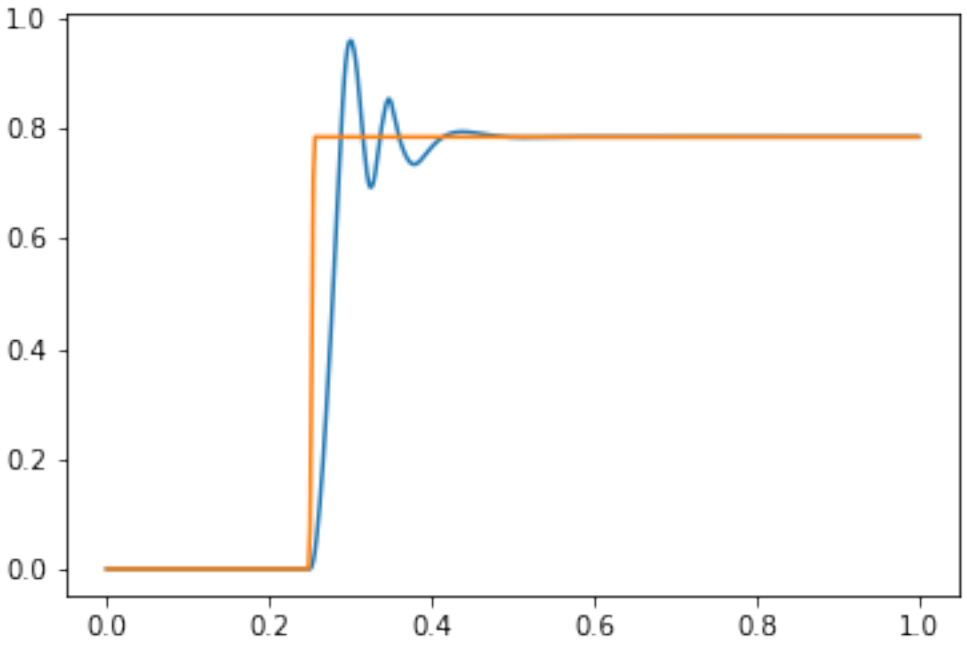
[46]:
```
t.tic()
PidFuzzController = fuzzy_controller(Mamdani1,typec='P',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.005912 seconds.
```

[47]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 3.471624 seconds.
```

[48]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.110735 seconds.
```

1.3.12 Mamdani Derivative Proportional Fuzzy Proportional Fuzzy Controller

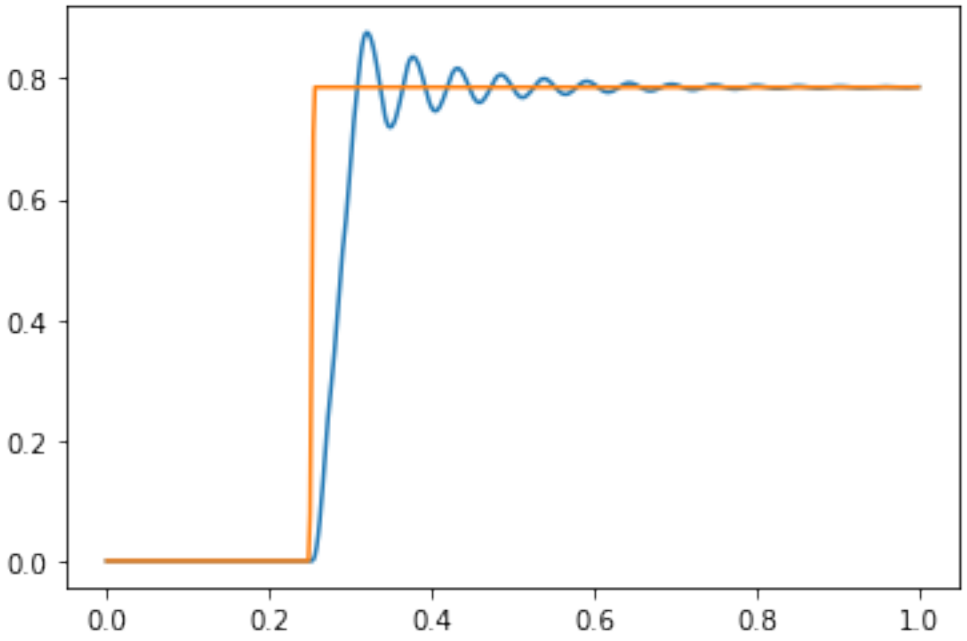
[49]:
```
t.tic()
PidFuzzController = fuzzy_controller(Mamdani2,typec='PD',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248, GCE=0.636618283)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.006226 seconds.
```

[50]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 10.988428 seconds.
```

[51]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.145536 seconds.
```

1.3.13 Derivative Proportional Fuzzy Proportional Controller - Integral Mamdani

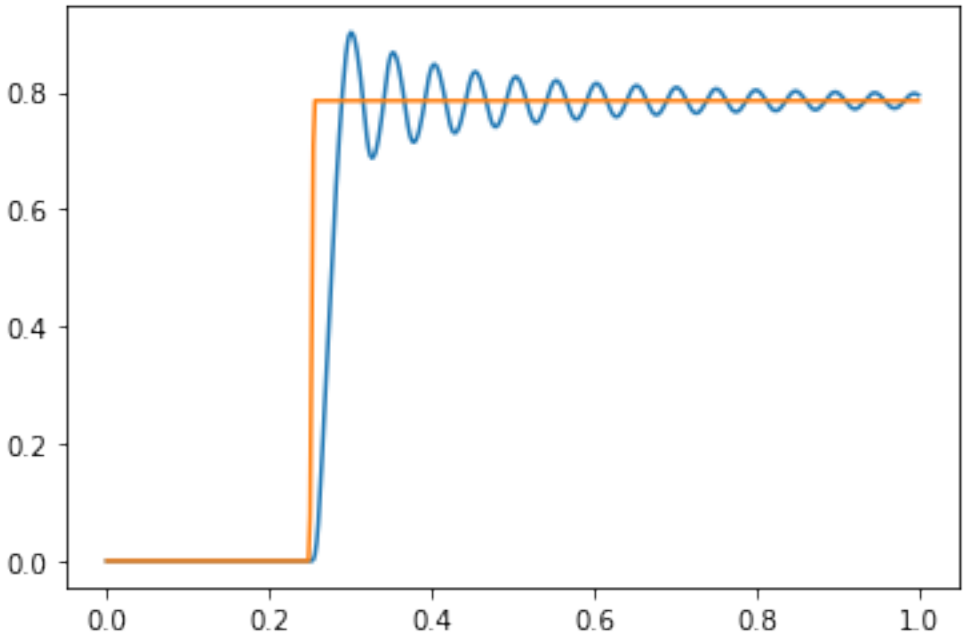
[52]:
```
t.tic()
PidFuzzController = fuzzy_controller(Mamdani2,typec='PD-I',tf=TF,DT = T[1],␣
↪GE=15.91545709, GU=0.094248, GCE=0.636618283, GIE=7.234298678)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.005980 seconds.
```

[53]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 9.330375 seconds.
```

[54]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
tt.toc()
```




```
Elapsed time is 0.139629 seconds.
Elapsed time is 83.602089 seconds.
```

1.3.14 F. L. Smidth Fuzzy Proportional Fuzzy Controller

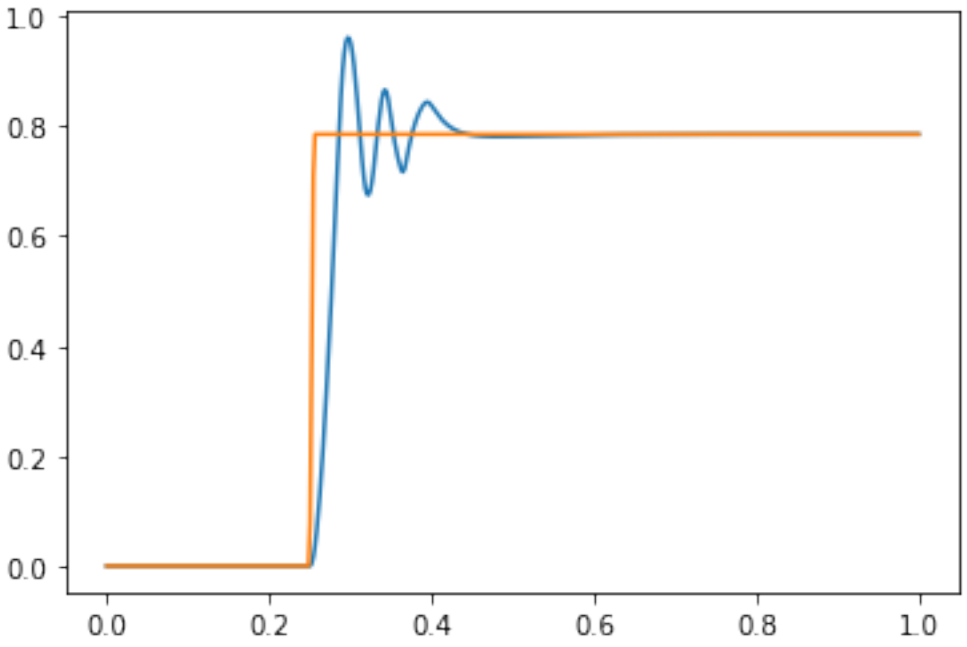
[55]:
```
t.tic()
PidFuzzController = fuzzy_controller(FLS1,typec='P',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.006337 seconds.
```

[56]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 3.270884 seconds.
```

[57]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.162472 seconds.
```

1.3.15 F. L. Smidth Derivative Proportional Fuzzy Proportional Controller

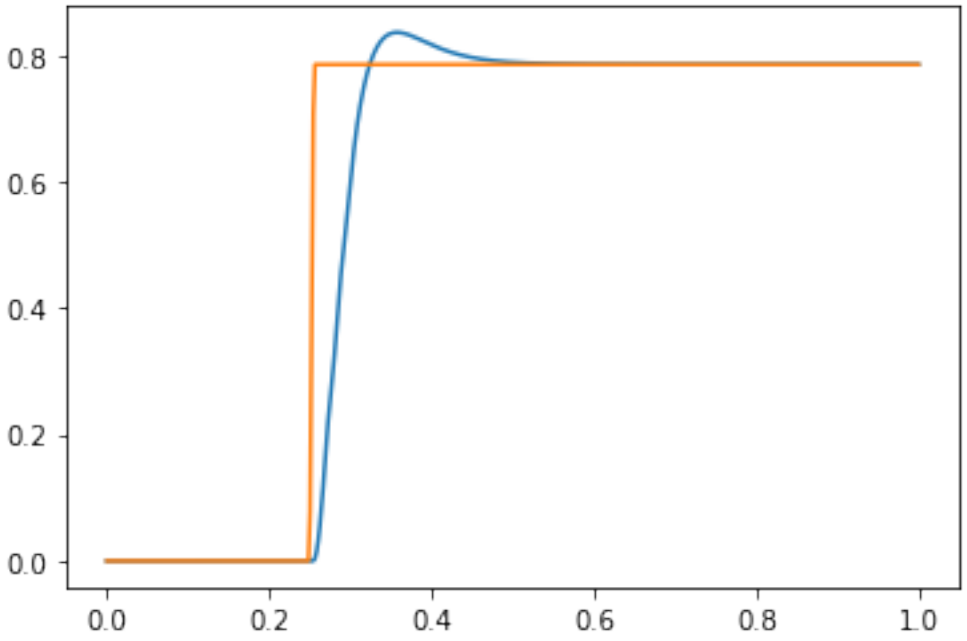
[58]:
```
t.tic()
PidFuzzController = fuzzy_controller(FLS2,typec='PD',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248, GCE=0.636618283)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.005264 seconds.
```

[59]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 8.252799 seconds.
```

[60]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.219881 seconds.
```

1.3.16 Derivative Proportional Fuzzy Proportional Controller - Integral F. L. Smidth

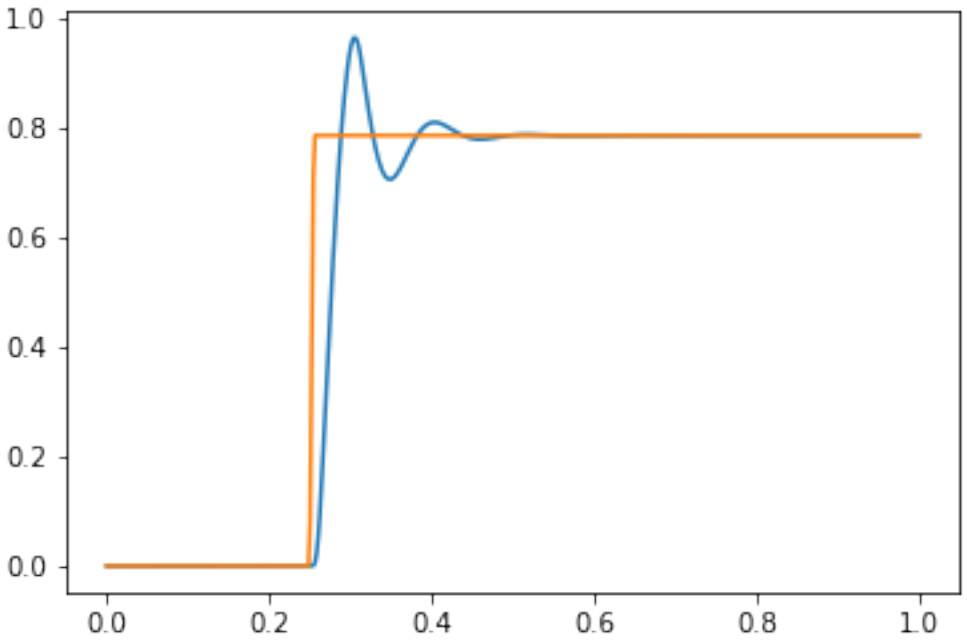
[61]:
```
t.tic()
PidFuzzController = fuzzy_controller(FLS2,typec='PD-I',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248, GCE=0.636618283, GIE=7.234298678)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.004874 seconds.
```

[62]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 11.898296 seconds.
```

[63]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
tt.toc()
```




```
Elapsed time is 0.135341 seconds.
Elapsed time is 108.199626 seconds.
```

1.3.17 Takagi-Sugeno Proportional Fuzzy Proportional Controller

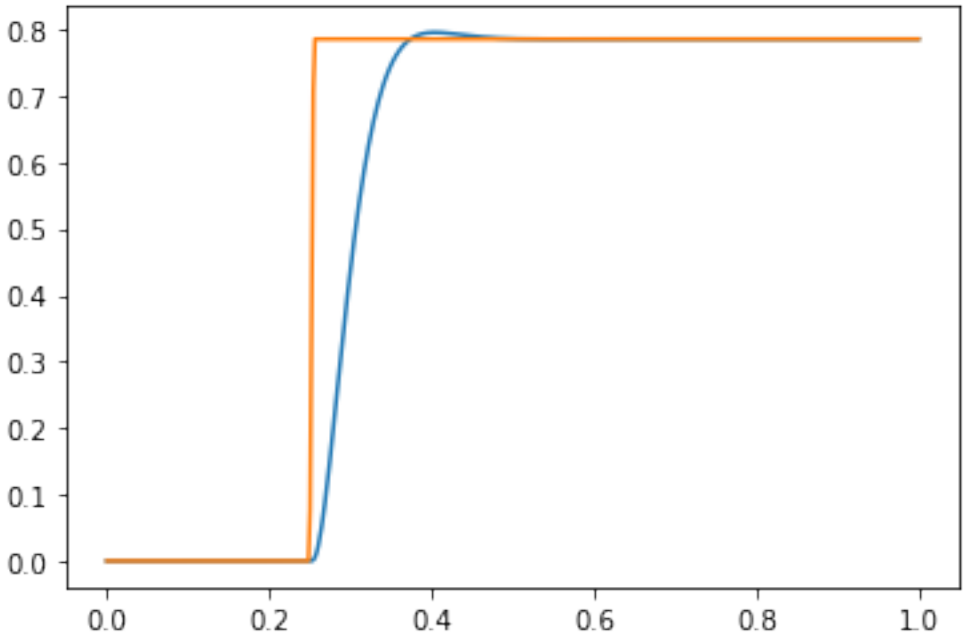
[64]:
```
t.tic()
PidFuzzController = fuzzy_controller(TSG1,typec='P',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.005243 seconds.
```

[65]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 1.219288 seconds.
```

[66]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.115056 seconds.
```

1.3.18 Takagi-Sugeno Derivative Proportional Diffuse Proportional Fuzzy Controller

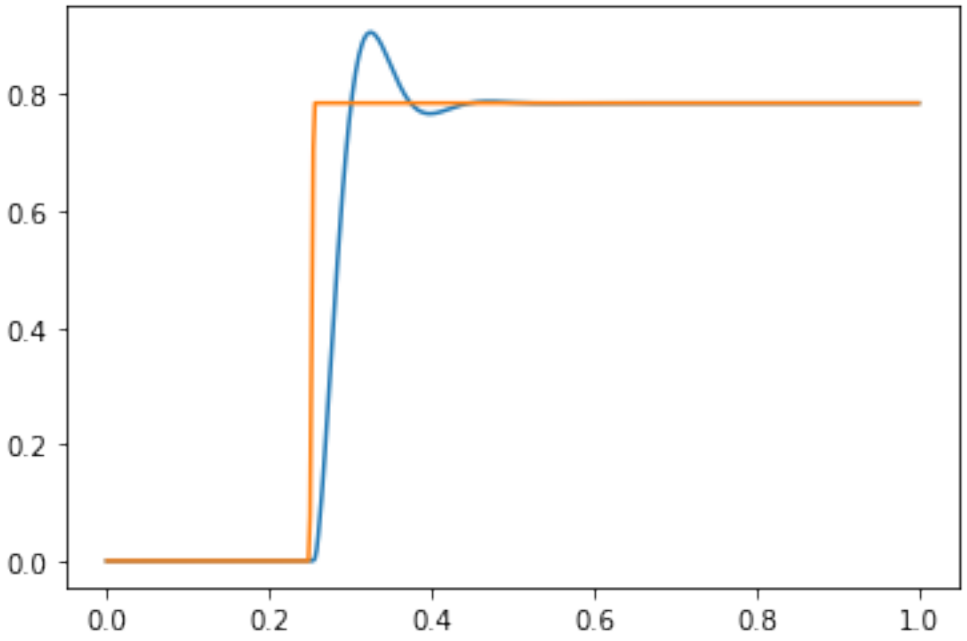
[67]:
```
t.tic()
PidFuzzController = fuzzy_controller(TSG2,typec='PD',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248, GCE=0.636618283)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.006295 seconds.
```

[68]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 13.630613 seconds.
```

[69]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
```




```
Elapsed time is 0.259567 seconds.
```

1.3.19 Derivative Proportional Fuzzy Proportional Controller - Integral Takagi-Sugeno

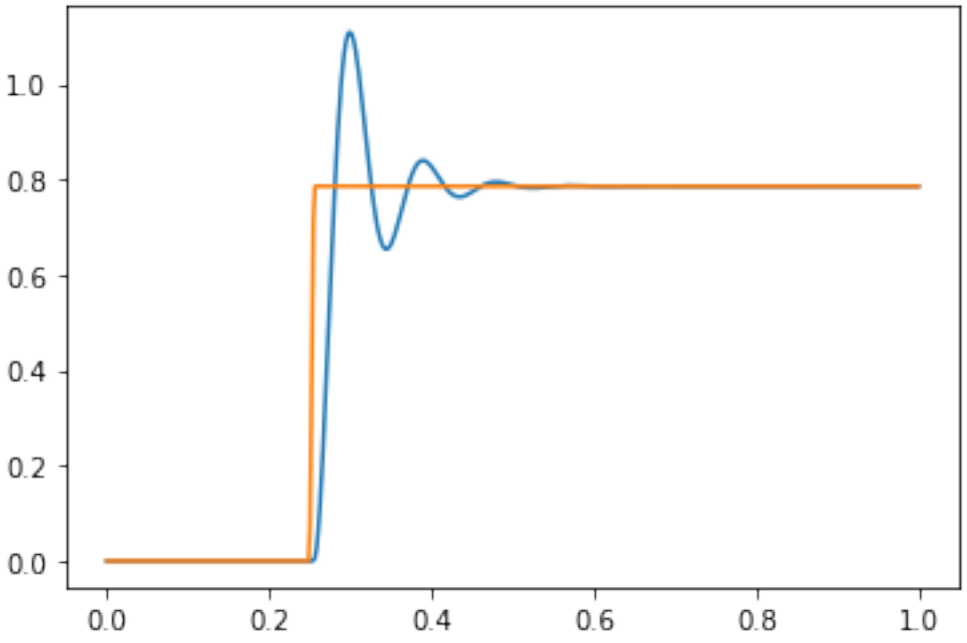
[70]:
```
t.tic()
PidFuzzController = fuzzy_controller(TSG2,typec='PD-I',tf=TF,DT = T[1], GE=15.
↪91545709, GU=0.094248, GCE=0.636618283, GIE=7.234298678)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

```
Elapsed time is 0.007682 seconds.
```

[71]:
```
t.tic()
T, Theta = control.input_output_response(PidFuzzSystemBlock,T,Input,0)
t.toc()
```

```
Elapsed time is 9.284757 seconds.
```

[72]:
```
t.tic()
plt.plot(T,Theta)
plt.plot(T,Input)
plt.show()
t.toc()
tt.toc()
```




```
Elapsed time is 0.240493 seconds.
Elapsed time is 134.310138 seconds.
```