

Examples for using the UPAFuzzySystems library

Examples for using the UPAFuzzySystems library

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

Required libraries

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

Inference Systems

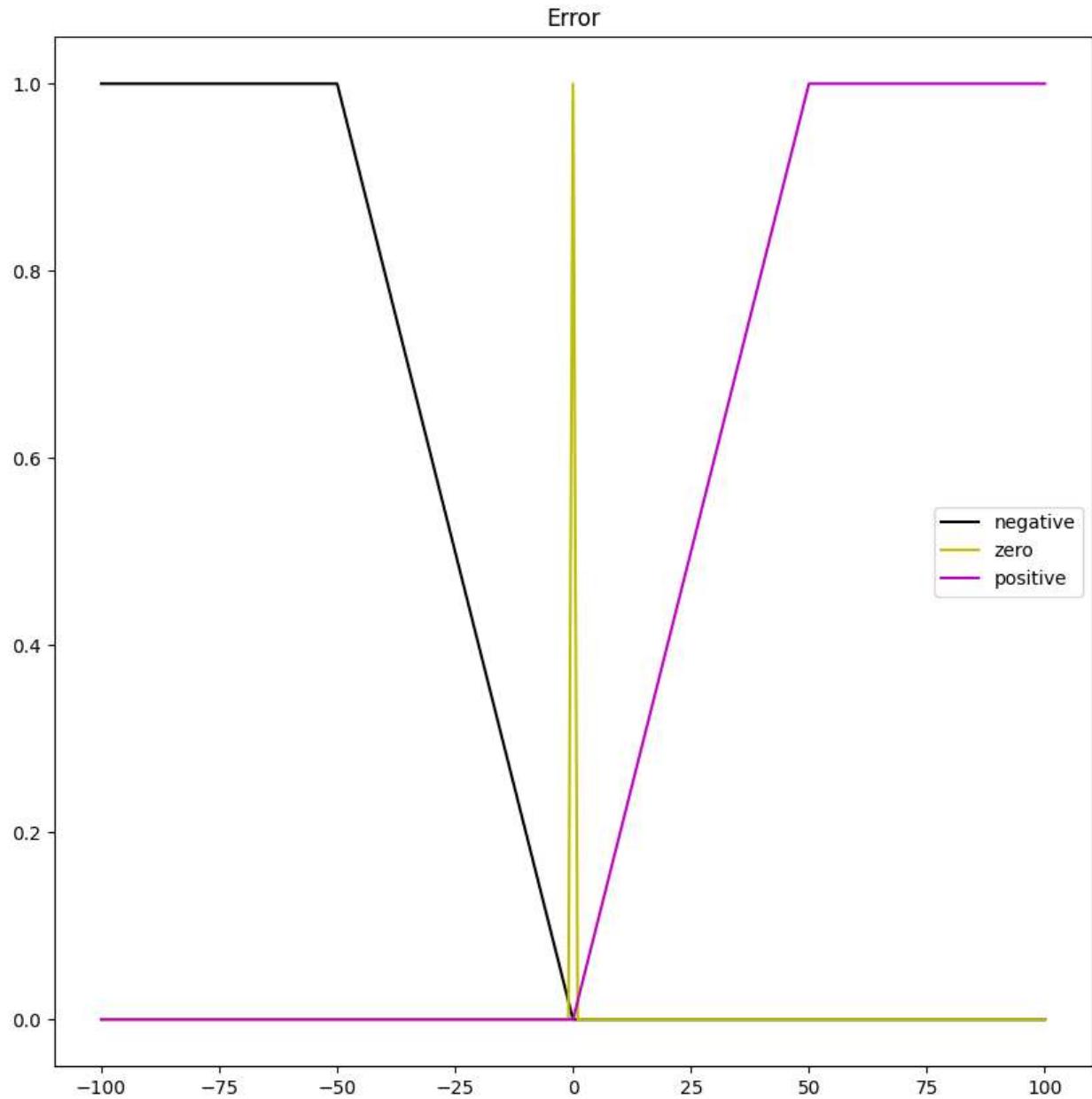
Mamdani Inference System One Input

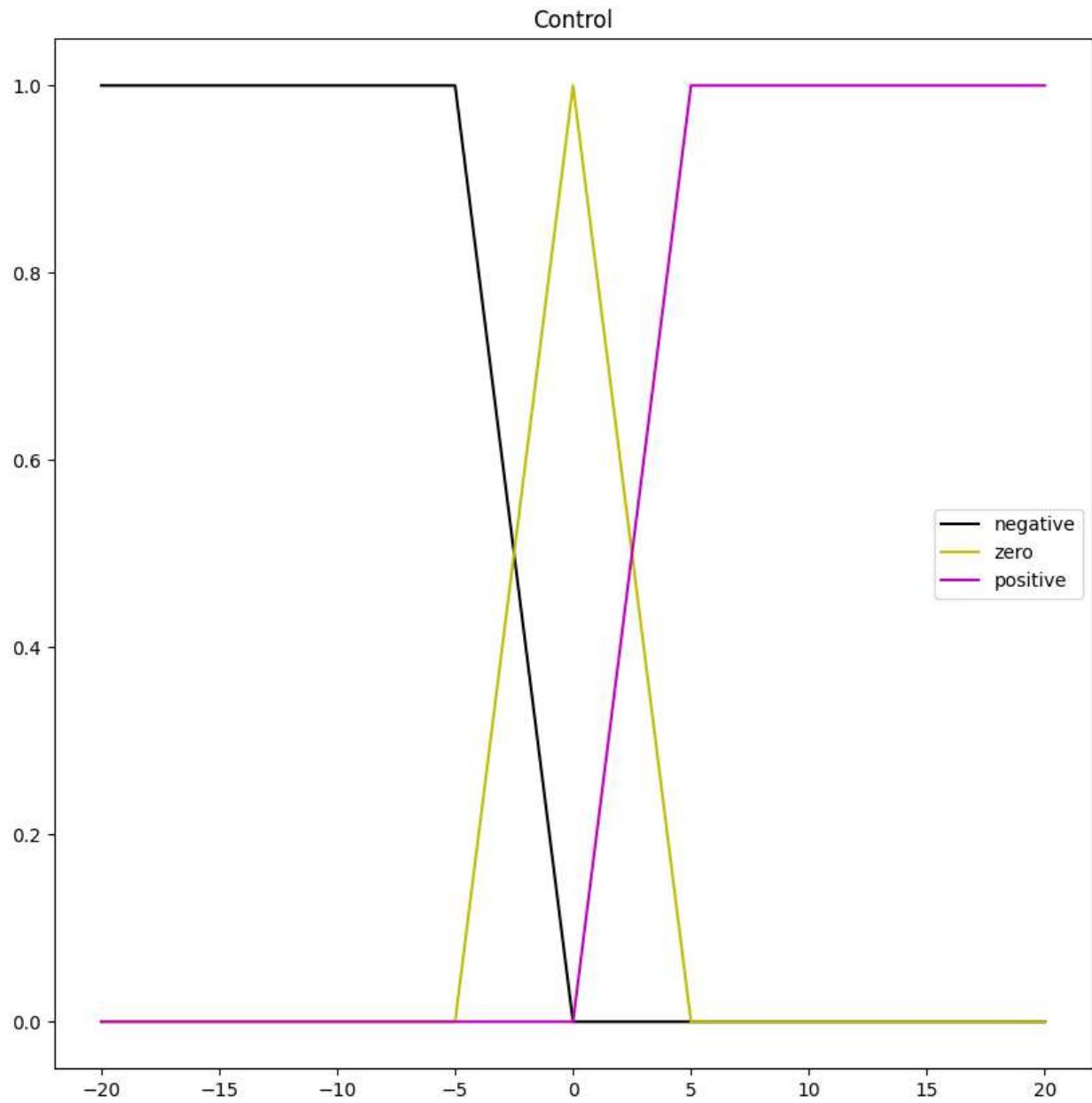
```
In [ ]: tt.tic()
t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trapmf',[-100,-100,-50,0])
Error_universe.add_fuzzyset('zero','trimf',[-1,0,1])
Error_universe.add_fuzzyset('positive','trapmf',[0,50,100,100])
Error_universe.view_fuzzy()

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
```

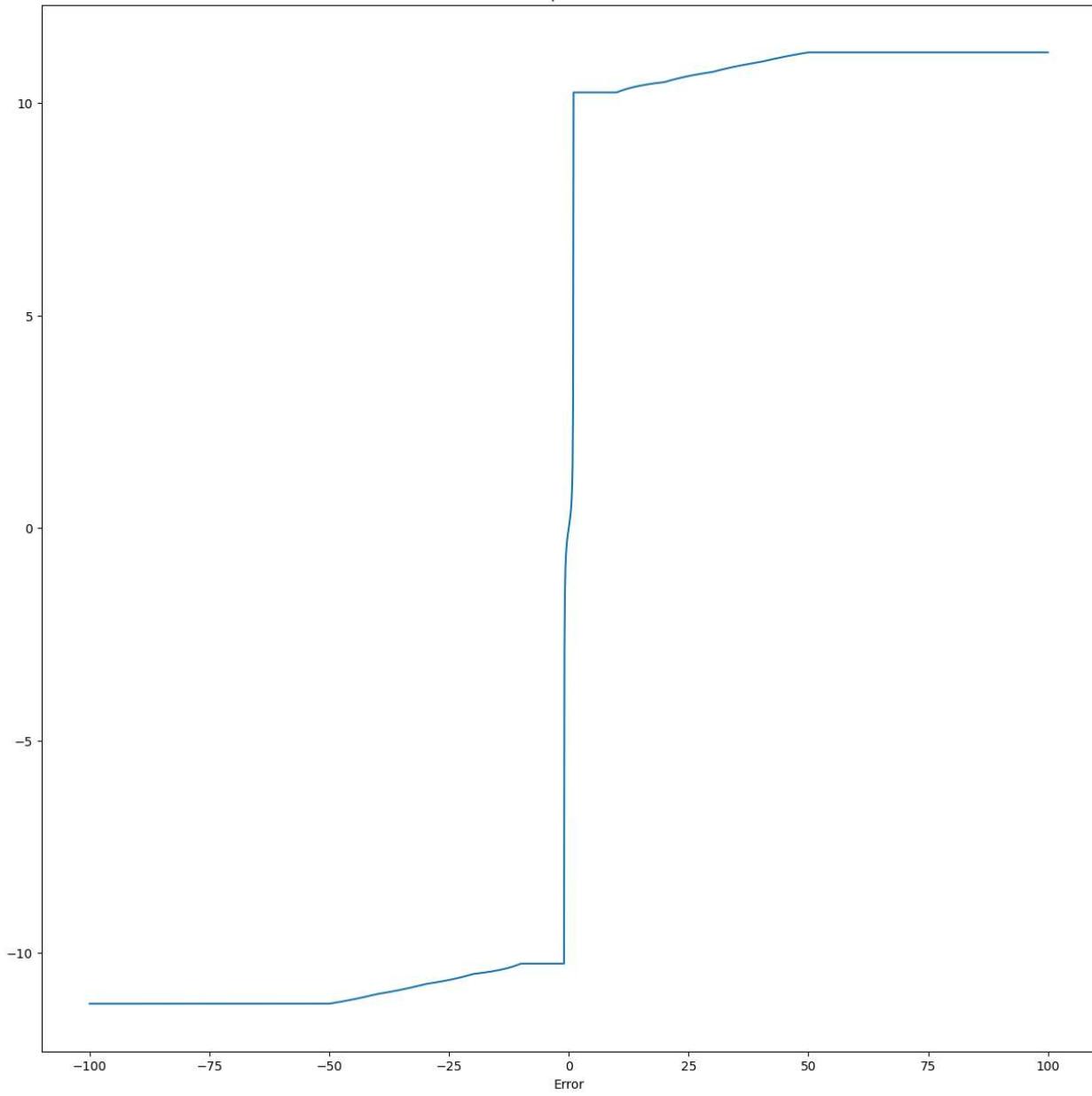




Elapsed time is 1.142388 seconds.

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

Surface Response: Mamdani



Elapsed time is 0.942981 seconds.

Mamdani Inference System Two Inputs

```
In [ ]: 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()

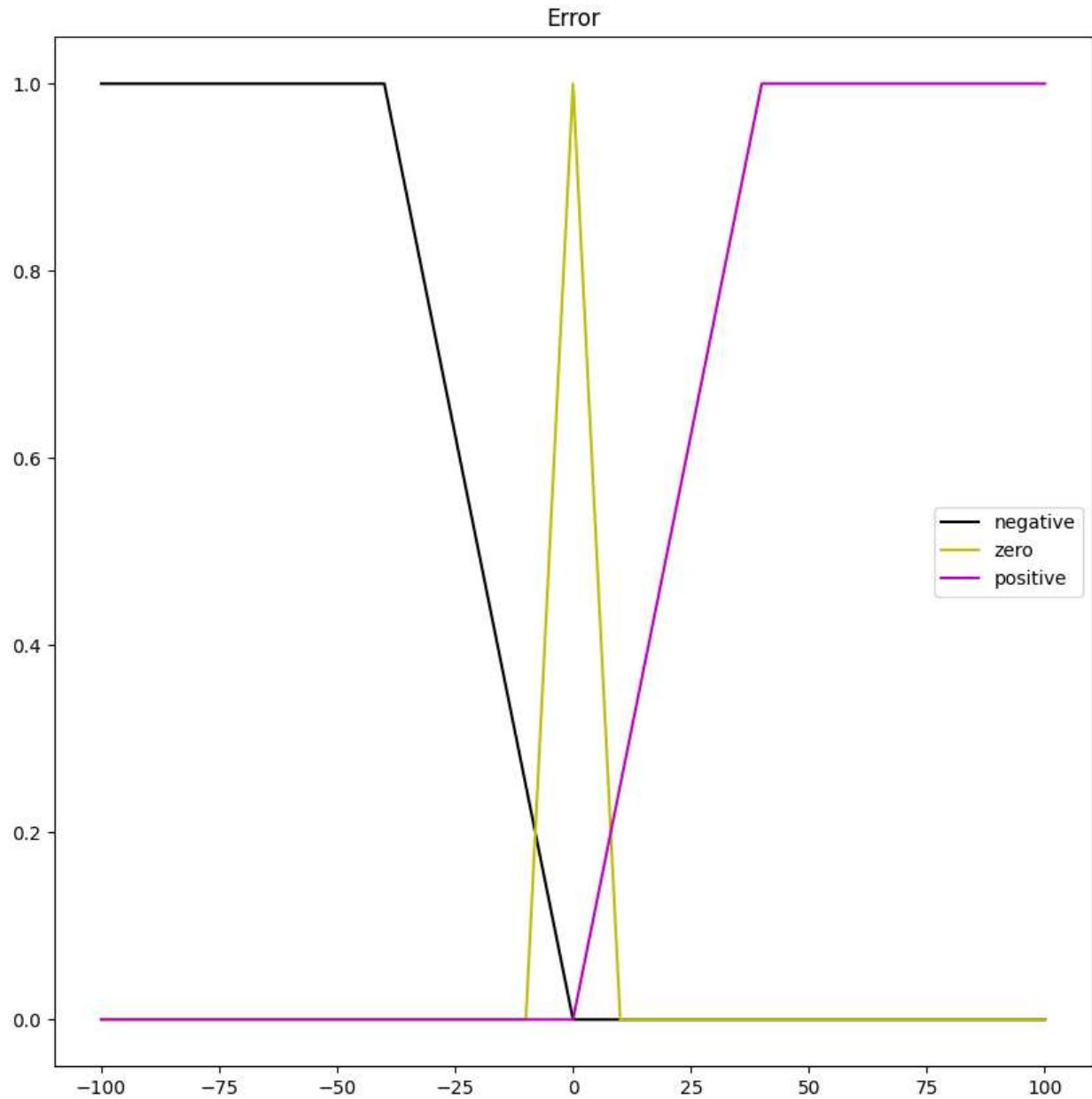
Control_universe = fuzzy_universe('Control', np.arange(-20,21,1), 'continuous')
Control_universe.add_fuzzyset('negative','trapmf',[-20,-20,-0.5,0])
Control_universe.add_fuzzyset('zero','trimf',[-0.01,0,0.01])
Control_universe.add_fuzzyset('positive','trapmf',[0,0.5,20,20])
Control_universe.view_fuzzy()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

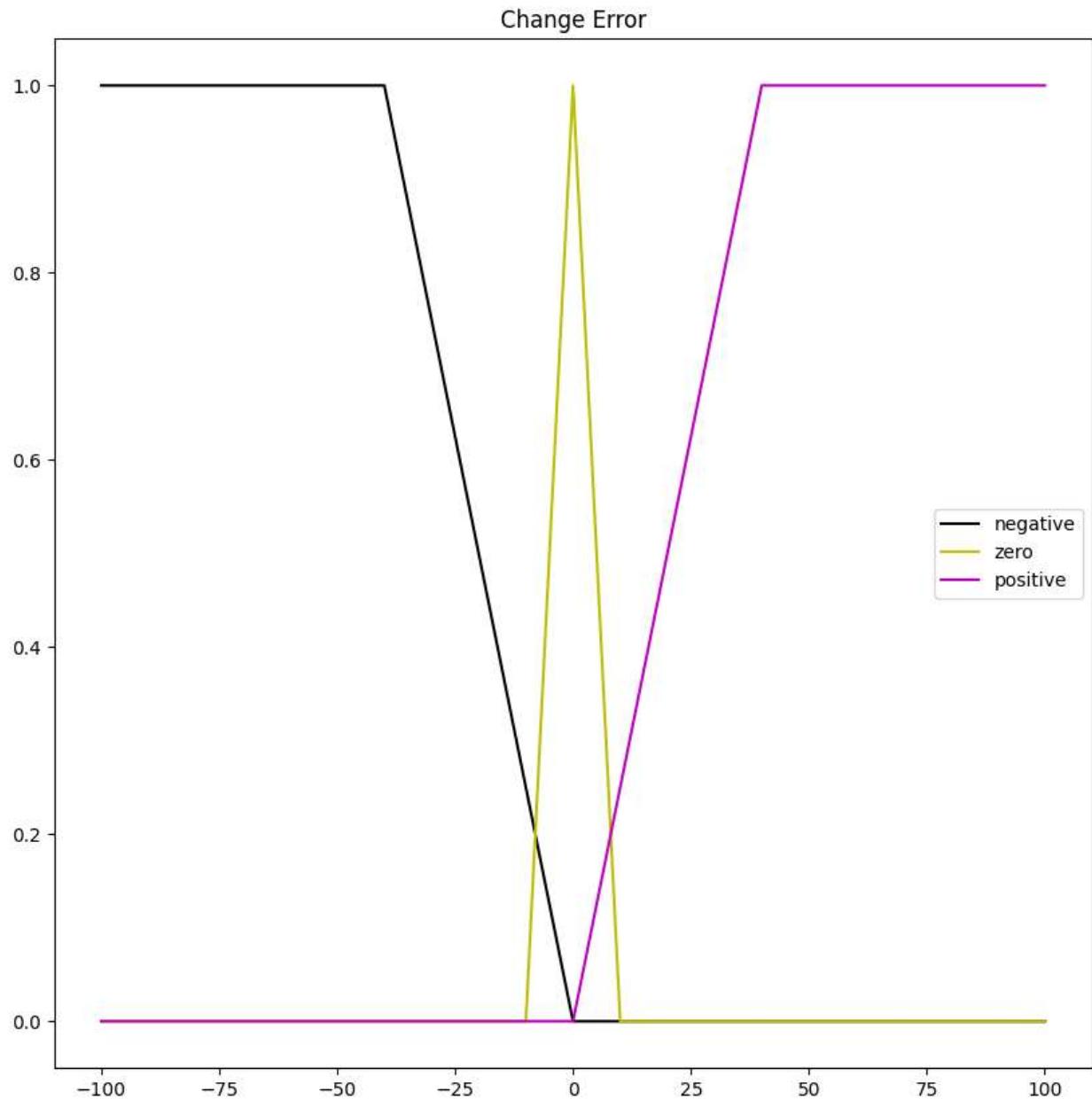
```
Mamdani2 = inference_system('Mamdani')
Mamdani2.add_premise(Error_universe)
Mamdani2.add_premise(ChError_universe)
Mamdani2.add_consequence(Control_universe)

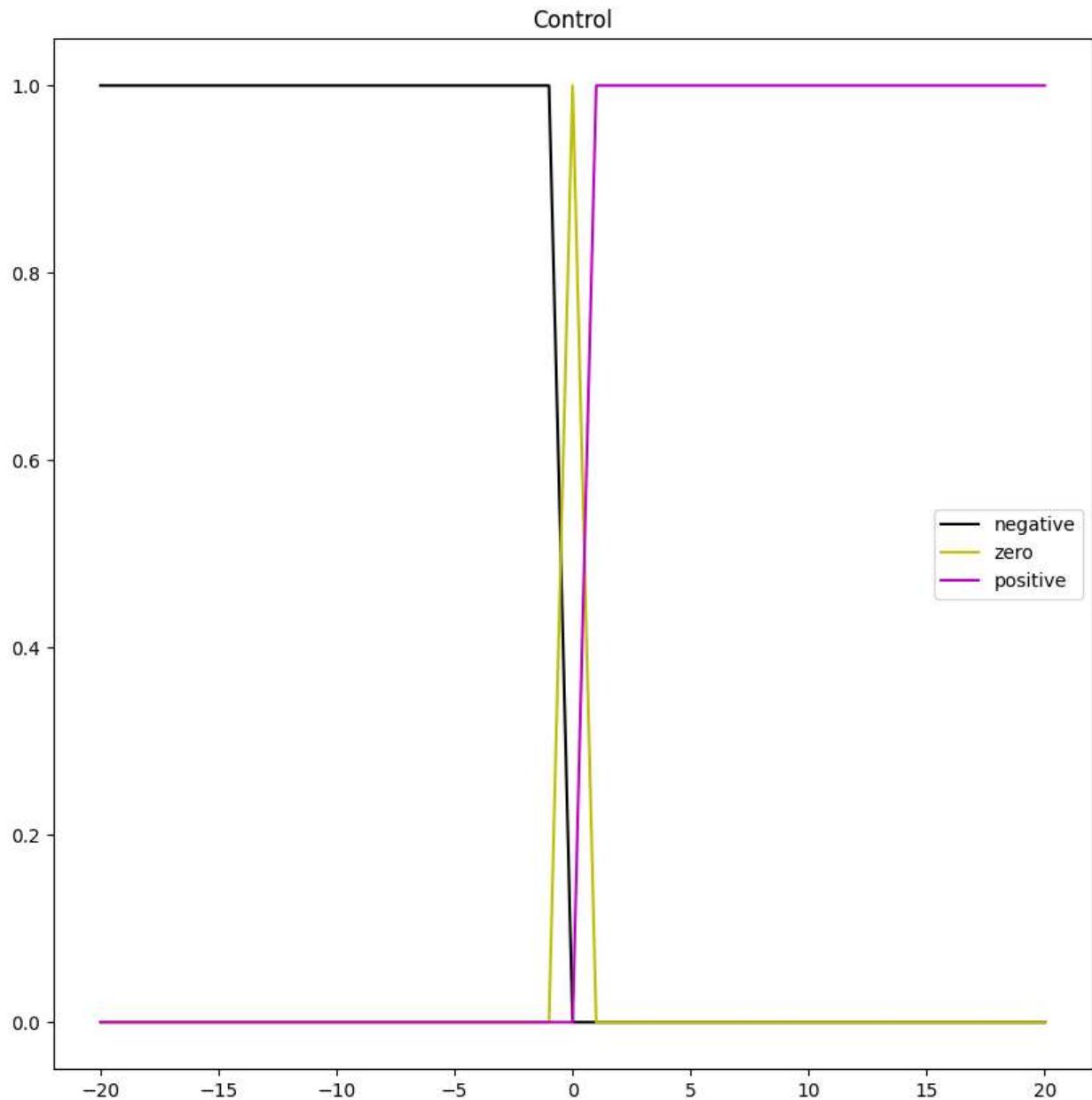
Mamdani2.add_rule([['Error','negative'], ['Change_Error','negative']], ['and'], [['Control','negative']])
Mamdani2.add_rule([['Error','negative'], ['Change_Error','zero']], ['and'], [['Control','negative']])
Mamdani2.add_rule([['Error','zero'], ['Change_Error','negative']], ['and'], [['Control','zero']])
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
```







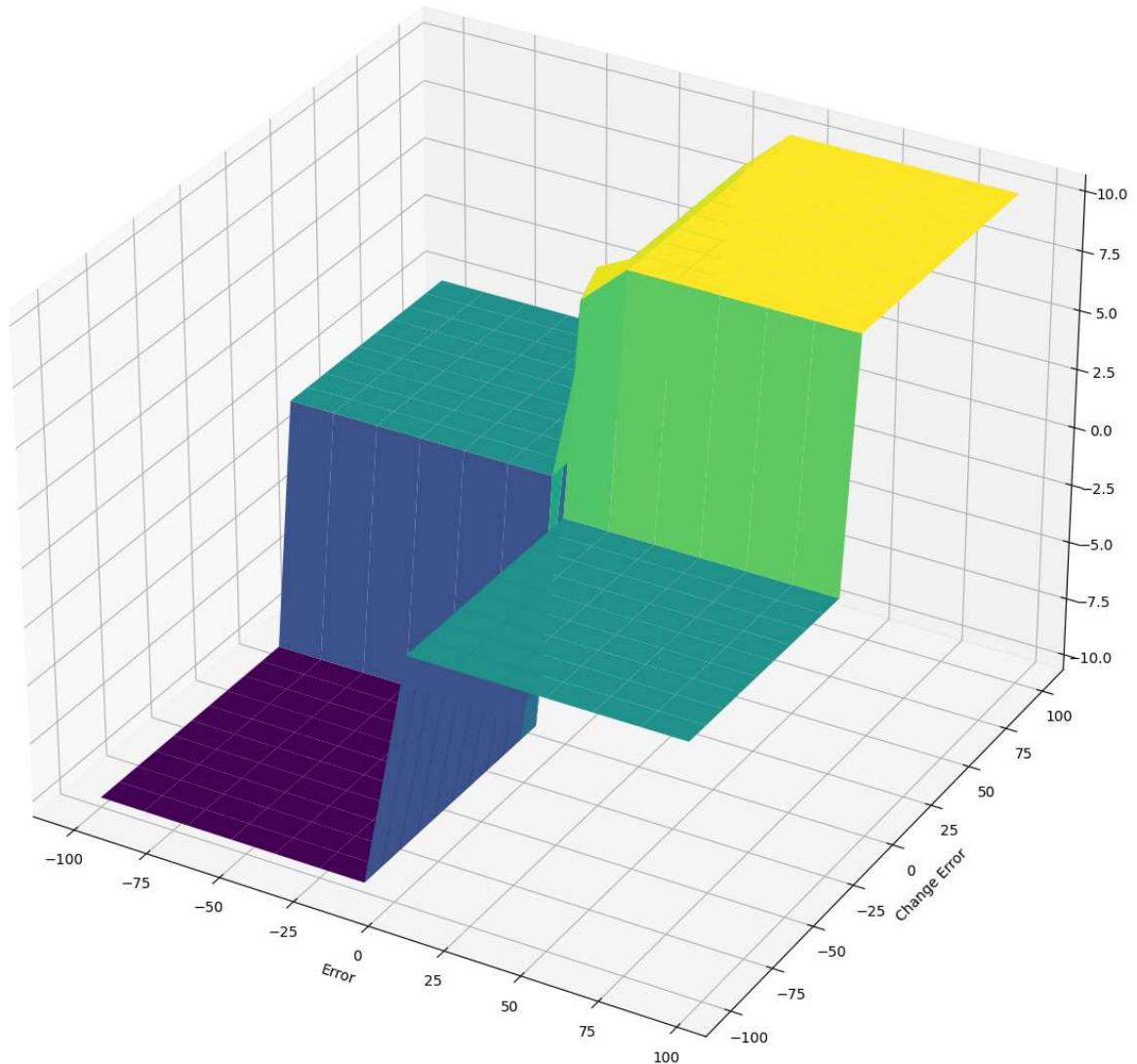
Elapsed time is 1.735792 seconds.

```
In [ ]: t.tic()
error_values = np.arange(-100,110,15)
change_error = np.arange(-100,110,10)

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

(21, 14)
(21, 14)
(14, 21)
```

Surface Response: Mamdani



Elapsed time is 1.024985 seconds.

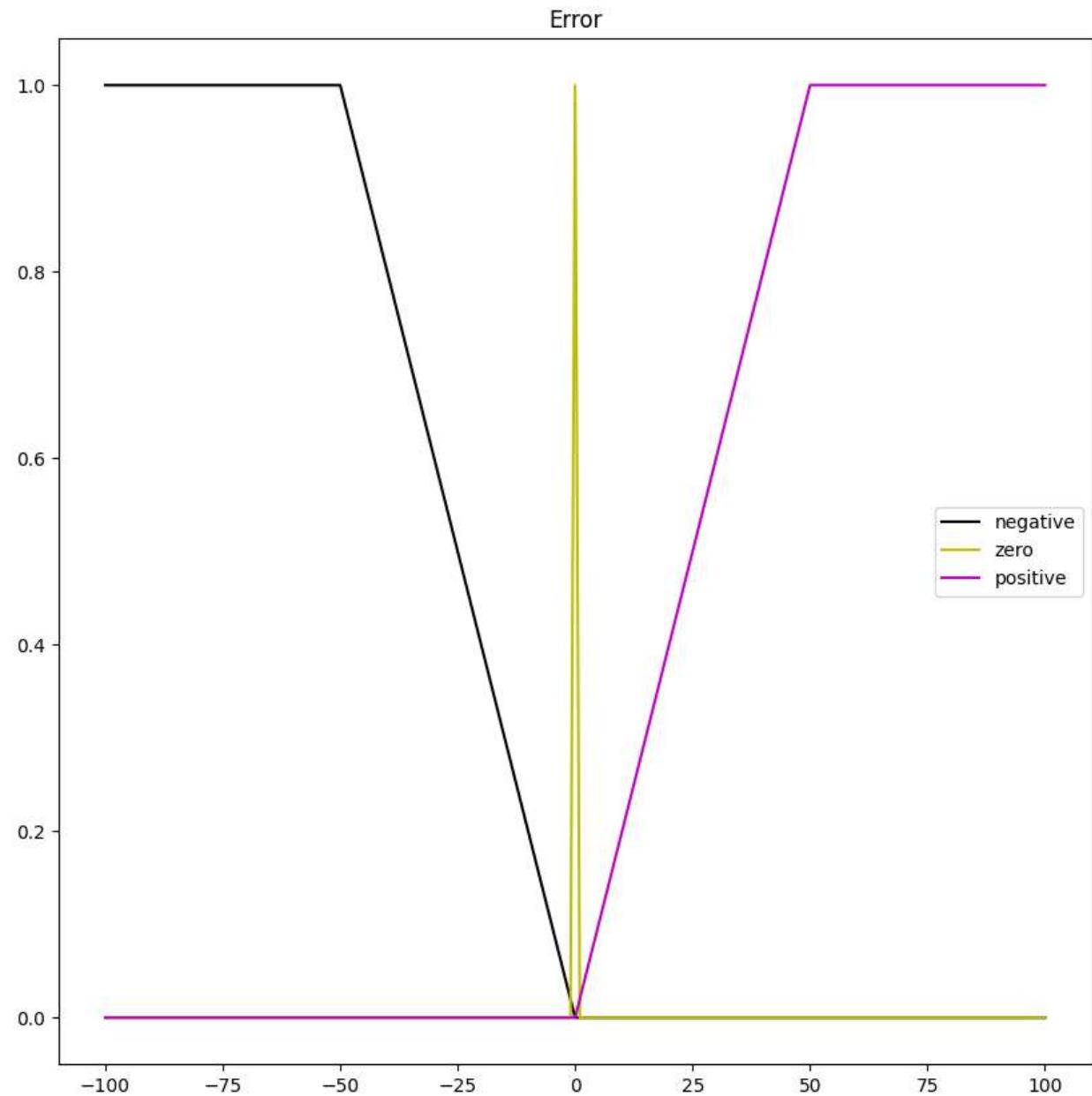
F.L. Smidth Inference System One Input

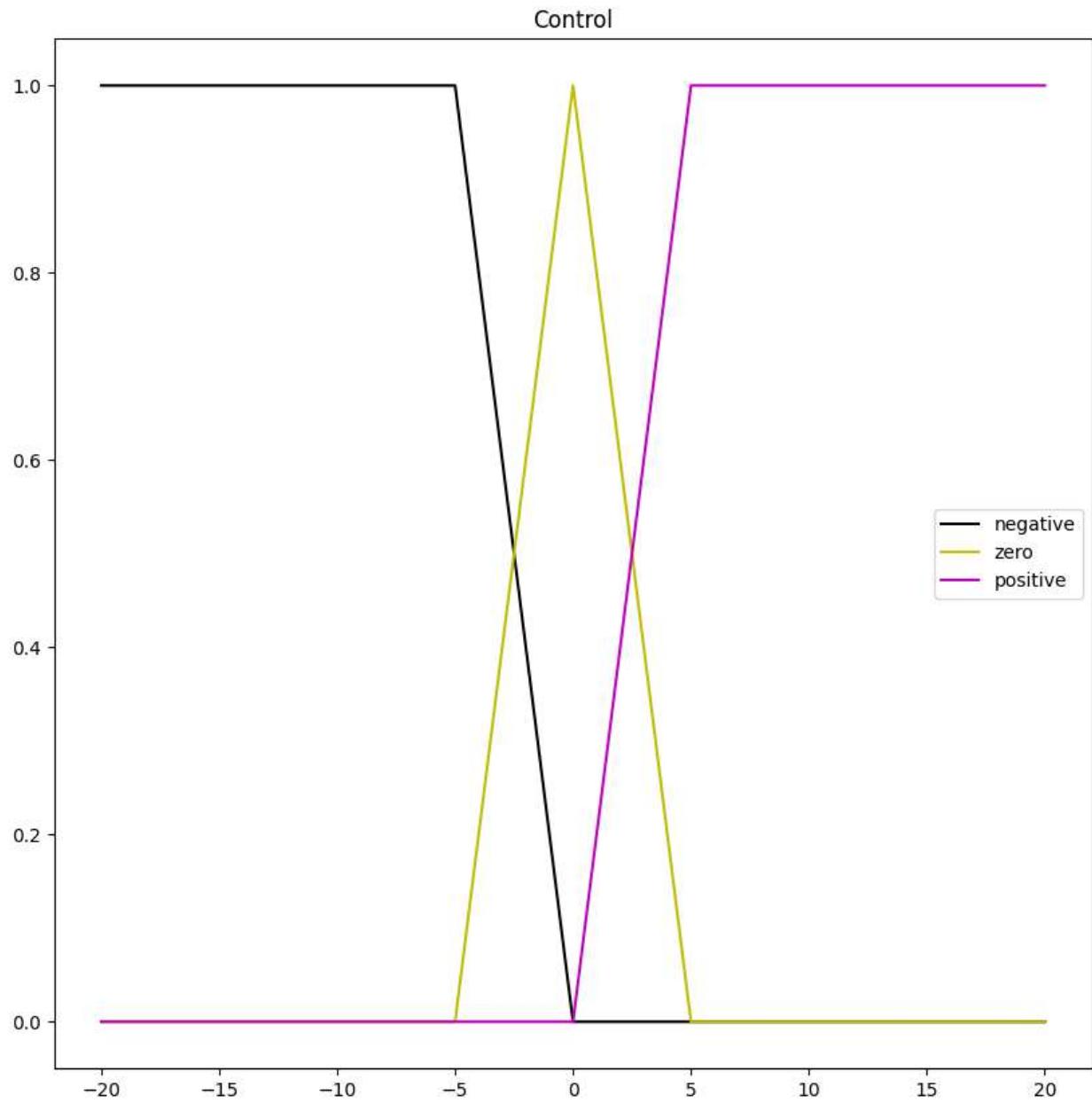
```
In [ ]: t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trapmf',[ -100,-100,-50,0])
Error_universe.add_fuzzyset('zero','trimf',[ -1,0,1])
Error_universe.add_fuzzyset('positive','trapmf',[0,50,100,100])
Error_universe.view_fuzzy()

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']]]])
```

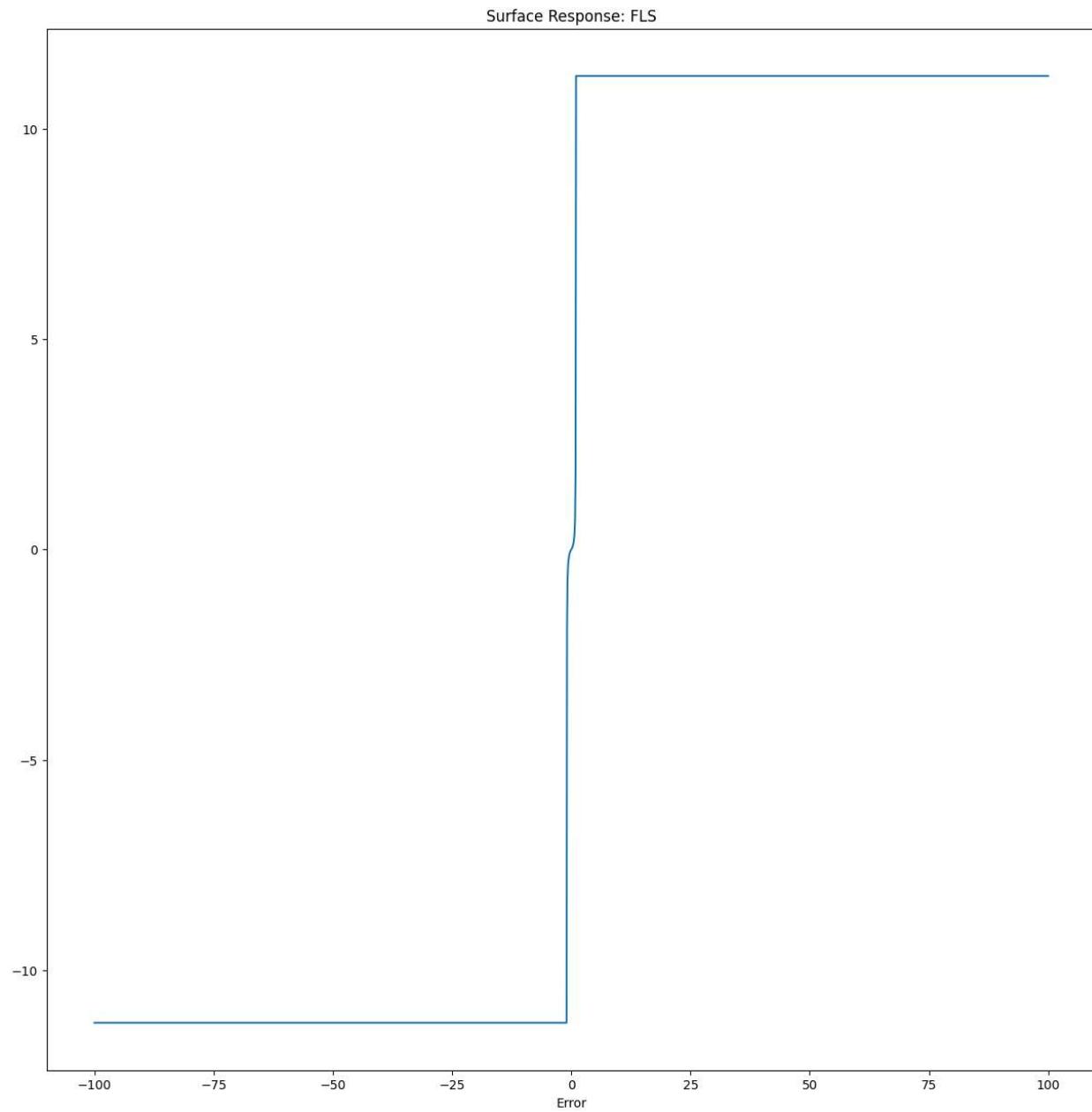
```
FLS1.add_rule([[['Error', 'positive']],[],[['Control', 'positive']]])  
FLS1.configure('FLSmidth')  
FLS1.build()  
t.toc()  
  
def Error_universe  
def Control_universe
```





Elapsed time is 0.960627 seconds.

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



Elapsed time is 0.575644 seconds.

F.L. Smidth Inference System Two Inputs

```
In [ ]: 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()
```

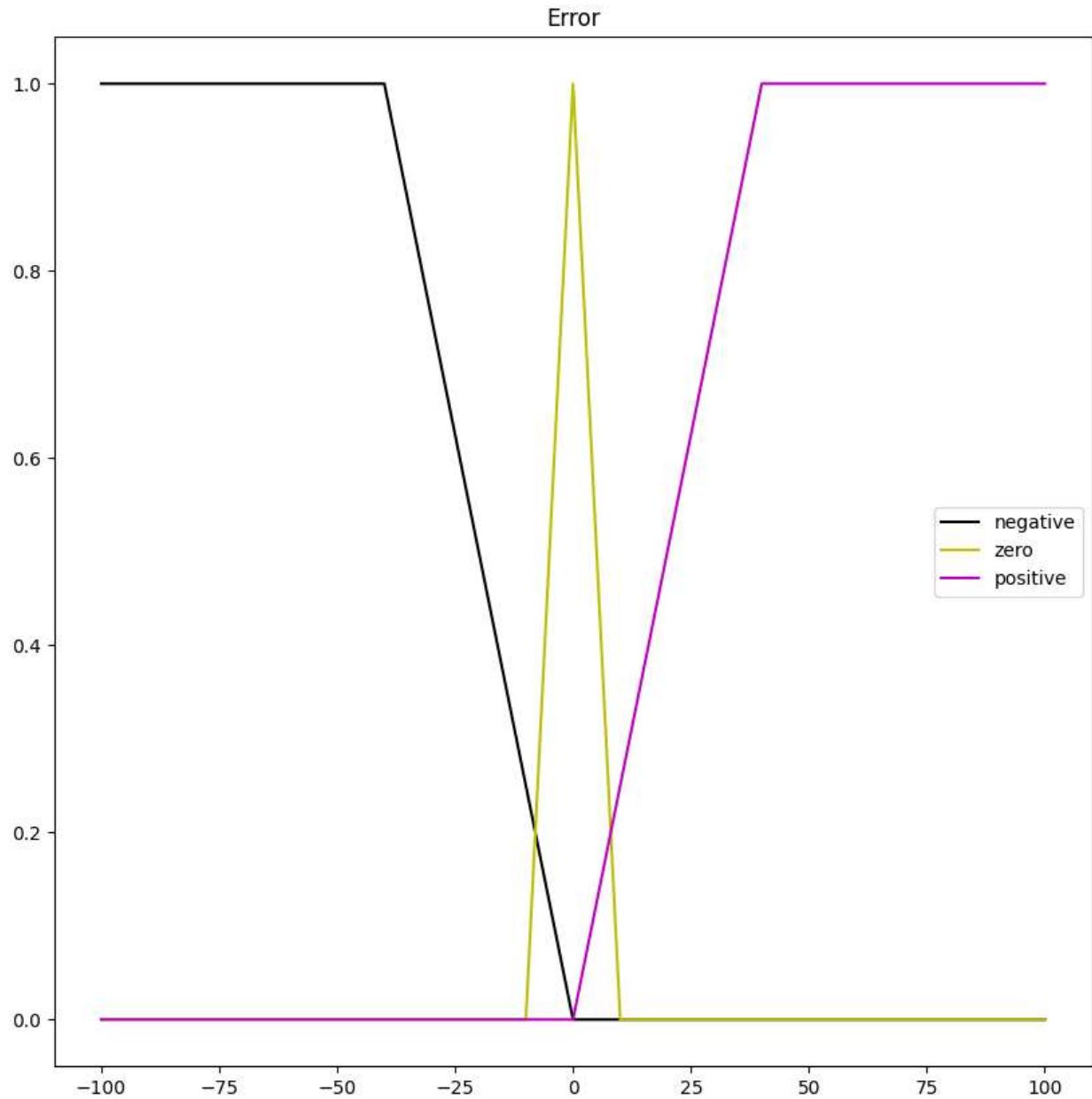
```
FLS2 = inference_system('FLSmidth')
FLS2.add_premise(Error_universe)
FLS2.add_premise(ChError_universe)
FLS2.add_consequence(Control_universe)

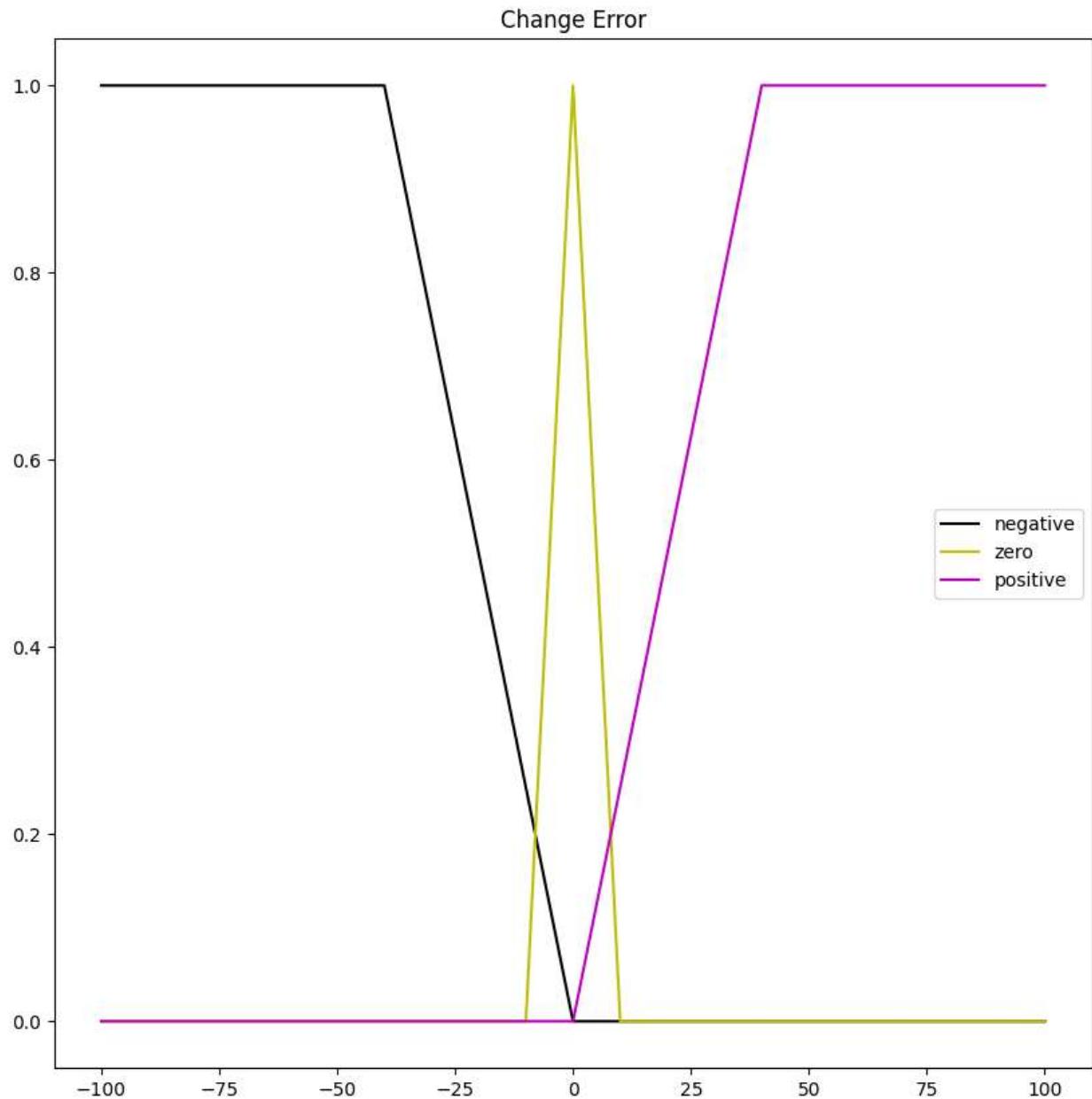
FLS2.add_rule([['Error', 'negative'], ['Change Error', 'negative']], ['and'], [['Control', 'negative']])
FLS2.add_rule([['Error', 'negative'], ['Change Error', 'zero']], ['and'], [['Control', 'negative']])
FLS2.add_rule([['Error', 'zero'], ['Change Error', 'negative']], ['and'], [['Control', 'zero']])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

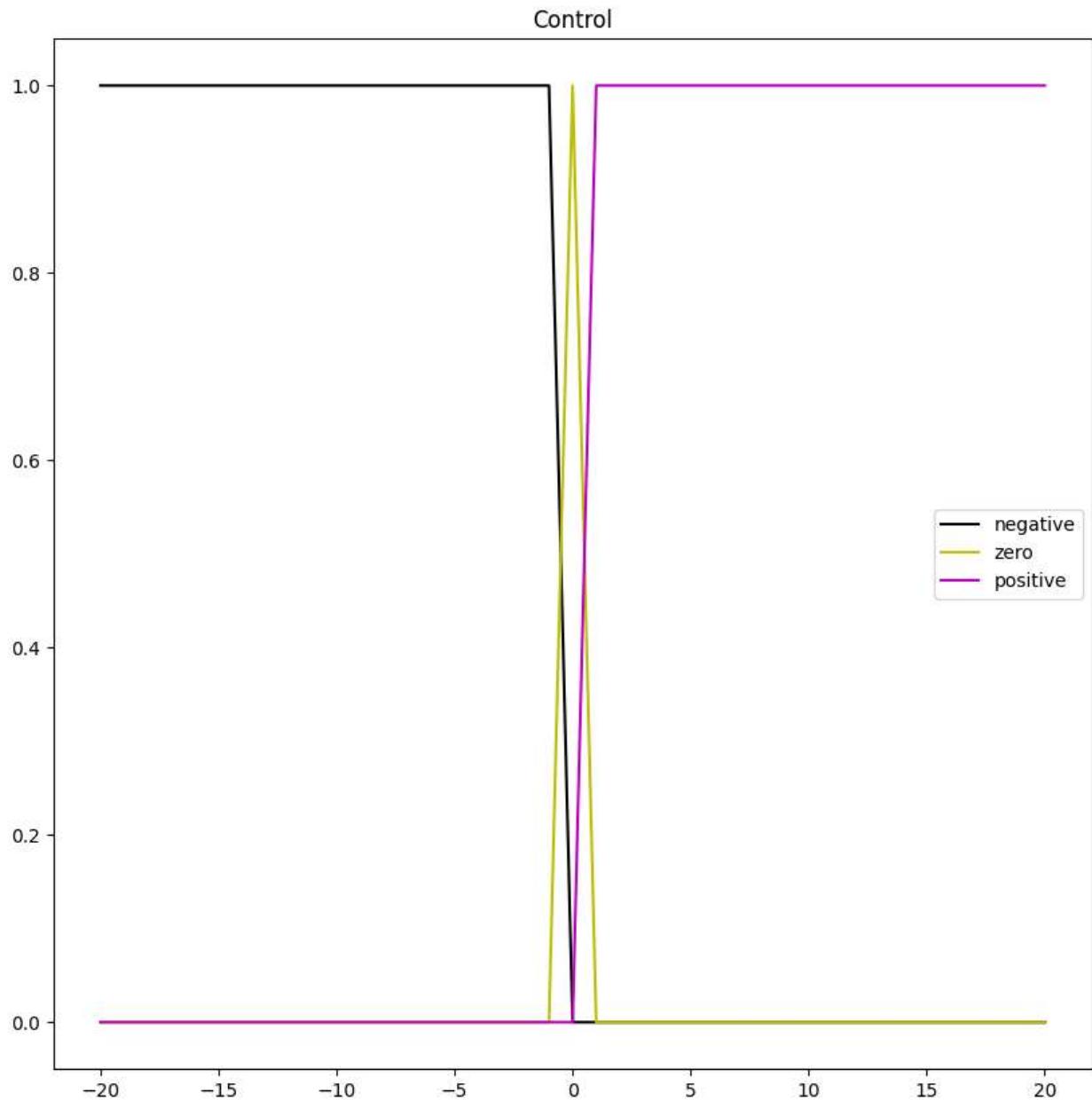
FLS2.configure('FLSmidth')

FLS2.build()
t.toc()

del Error_universe
del ChError_universe
del Control_universe
```







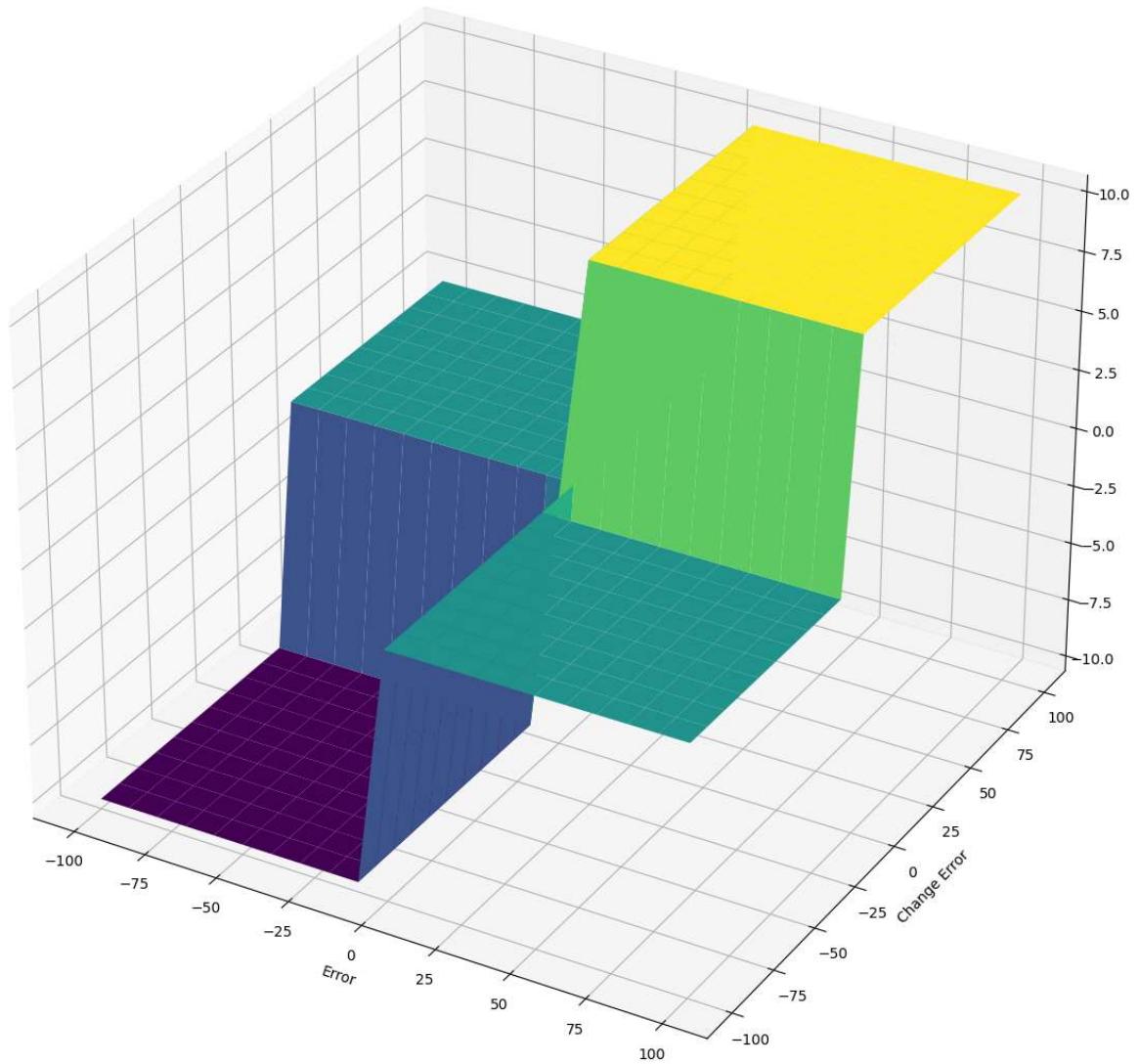
Elapsed time is 1.153397 seconds.

```
In [ ]: 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)
```

Surface Response: FLSmidt



Elapsed time is 0.602983 seconds.

Takagi-Sugeno Inference System One Entry

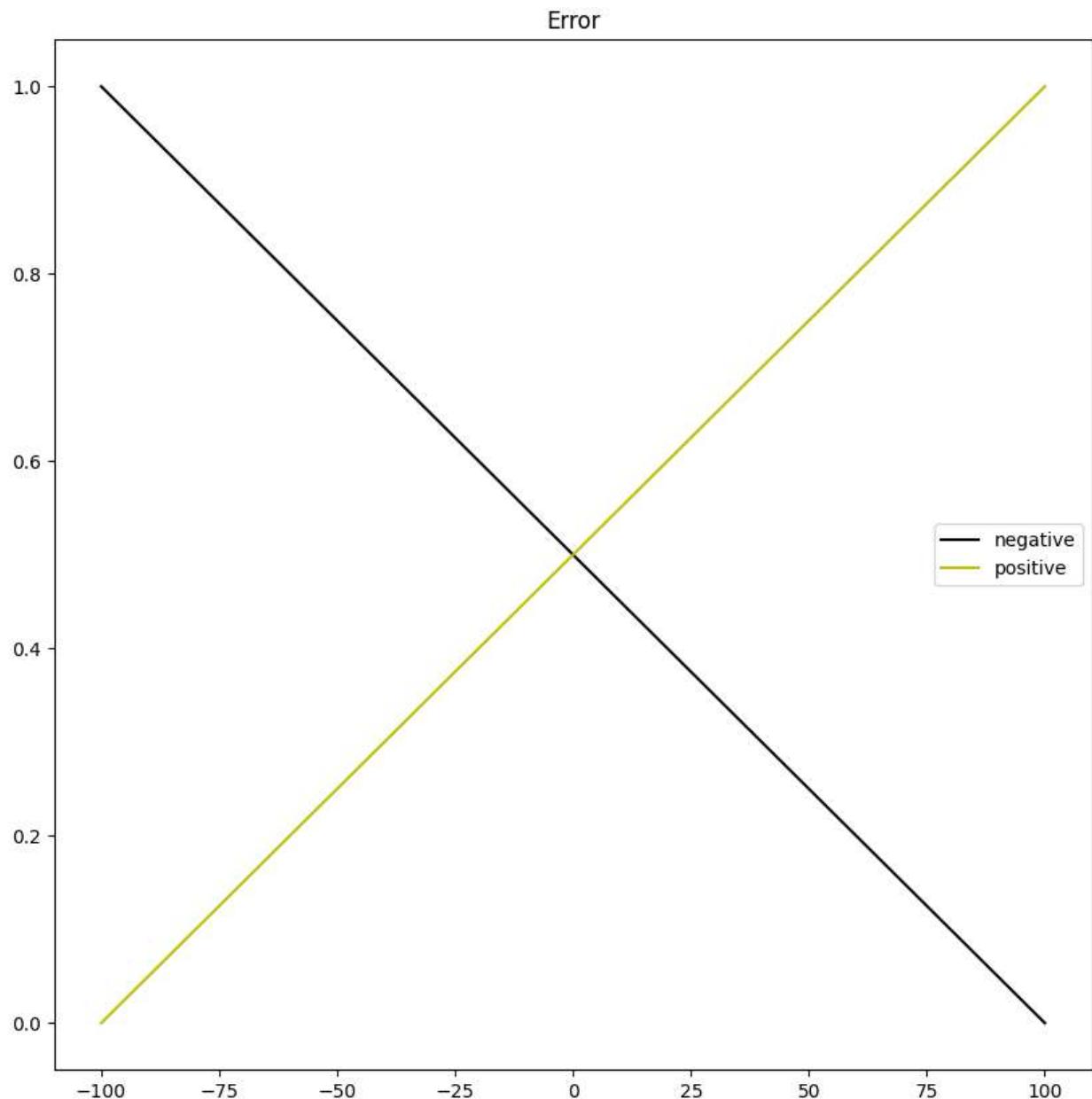
```
In [ ]: t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trimf',[ -200,-100,100])
Error_universe.add_fuzzyset('positive','trimf',[ -100,100,200])
Error_universe.view_fuzzy()

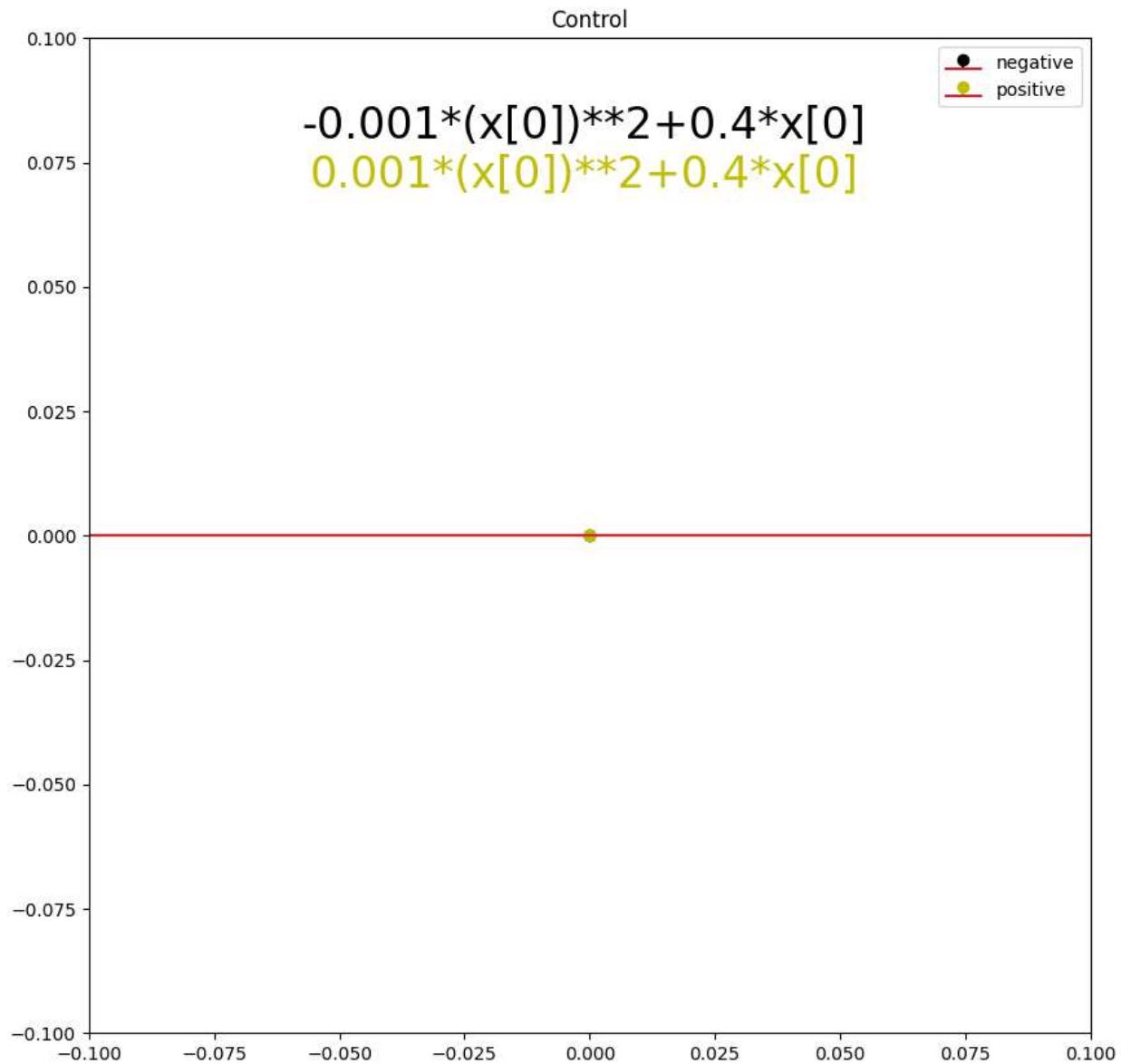
Control_universe = fuzzy_universe('Control', np.arange(-20,22,2), 'continuous')
Control_universe.add_fuzzyset('negative','eq','-0.001*(x[0])**2+0.4*x[0]')
Control_universe.add_fuzzyset('positive','eq','0.001*(x[0])**2+0.4*x[0]')
Control_universe.view_fuzzy()

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()  
t Error_universe  
t Control_universe
```

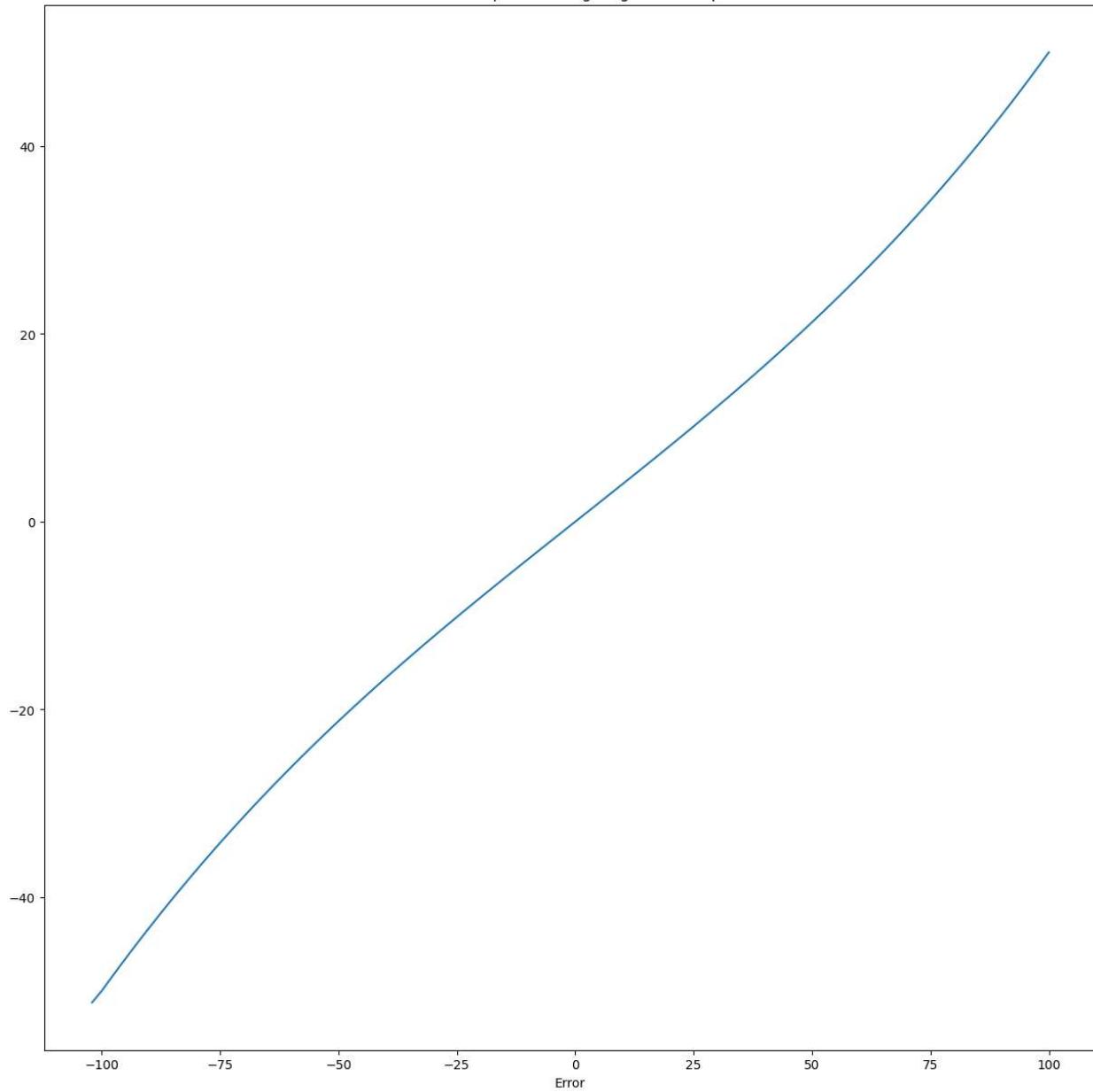




Elapsed time is 0.792942 seconds.

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

Surface Response: Takagi-Sugeno One Input



Elapsed time is 0.421366 seconds.

Takagi-Sugeno Inference System Two Inputs

```
In [ ]: 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()

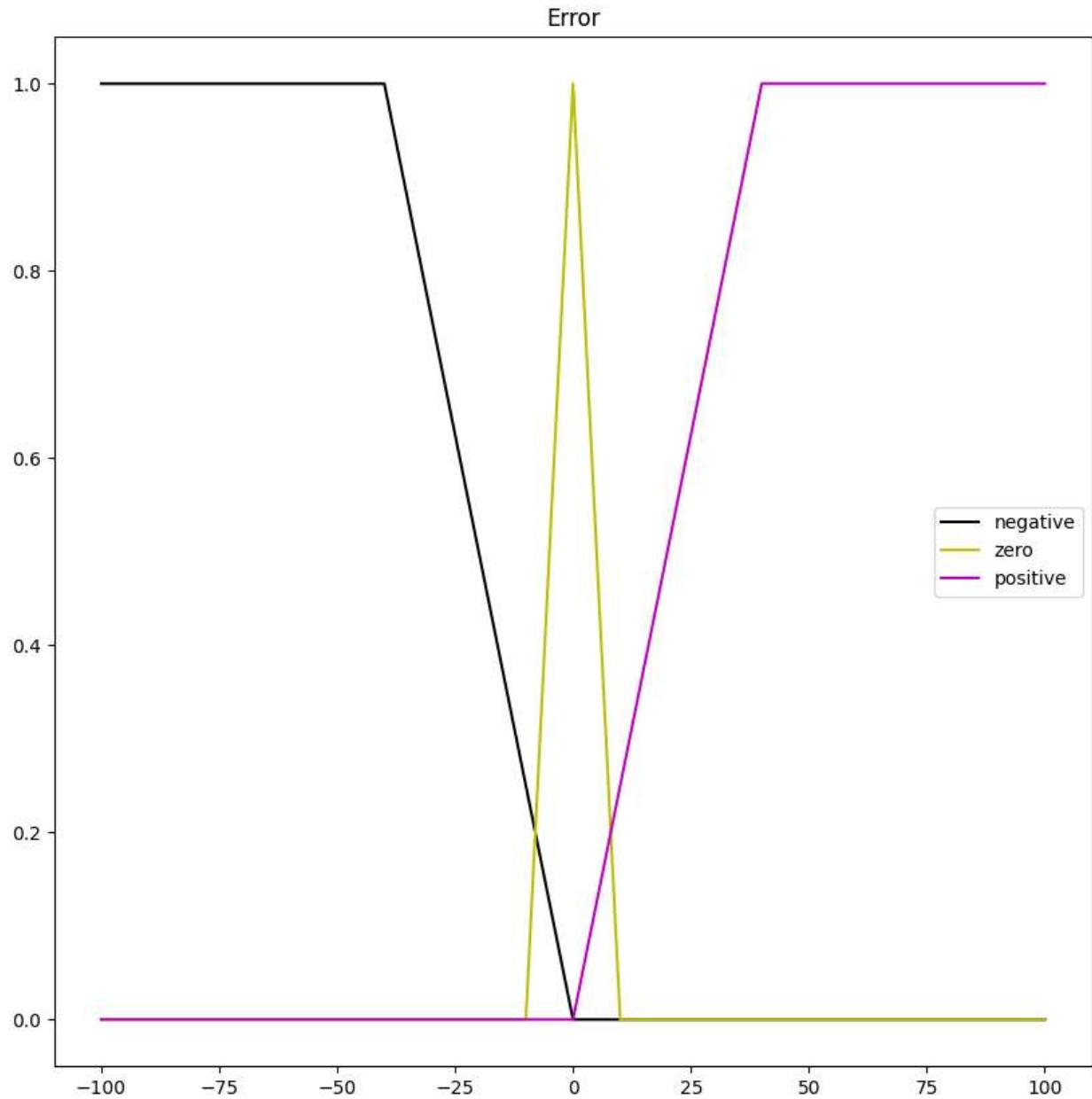
Control_universe = fuzzy_universe('Control', np.arange(-20,22,2), 'continuous')
Control_universe.add_fuzzyset('negative','eq','0.8*x[0]+0.1*x[1]')
Control_universe.add_fuzzyset('zero','eq','0.8*x[0]+0.005*x[1]')
Control_universe.add_fuzzyset('positive','eq','0.8*x[0]+0.1*x[1]')
Control_universe.view_fuzzy()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

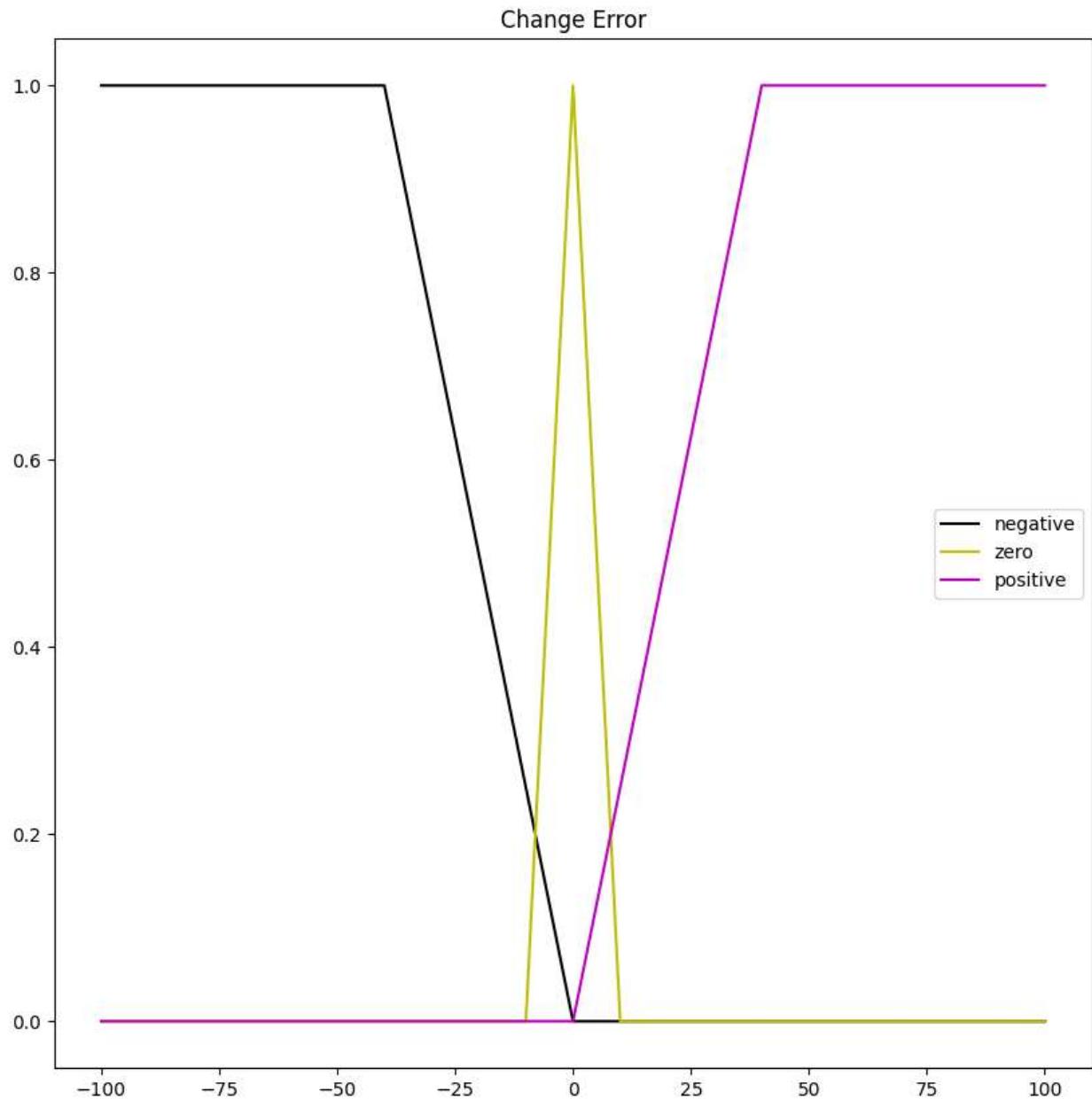
```
TSG2 = inference_system('Takagi-Sugeno Two Inputs')
TSG2.add_premise(Error_universe)
TSG2.add_premise(ChError_universe)
TSG2.add_consequence(Control_universe)

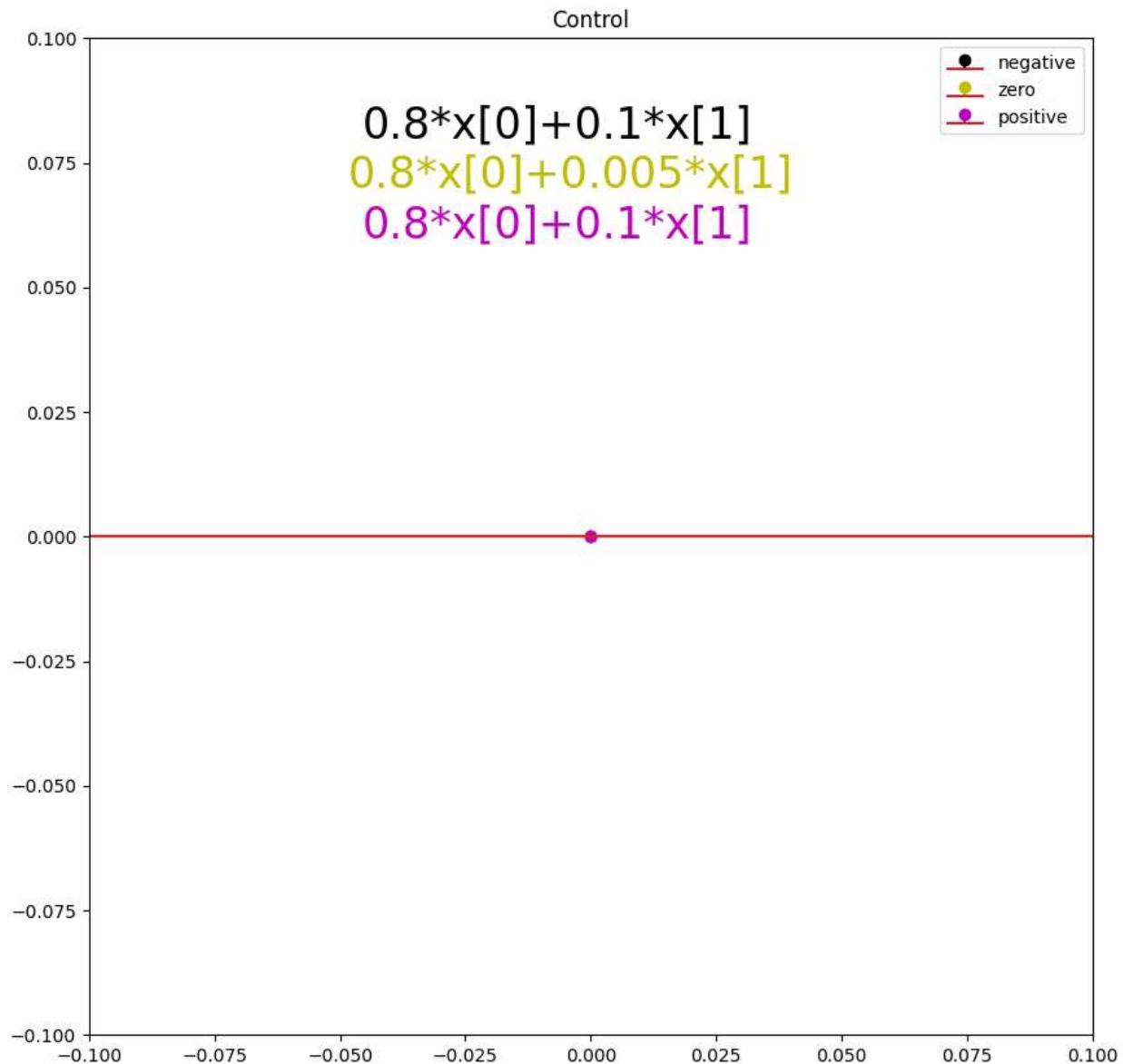
TSG2.add_rule([['Error', 'negative'], ['Change Error', 'negative']], ['and'], [['Control', 'negative']])
TSG2.add_rule([['Error', 'negative'], ['Change Error', 'zero']], ['and'], [['Control', 'negative']])
TSG2.add_rule([['Error', 'zero'], ['Change Error', 'negative']], ['and'], [['Control', 'zero']])
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
```







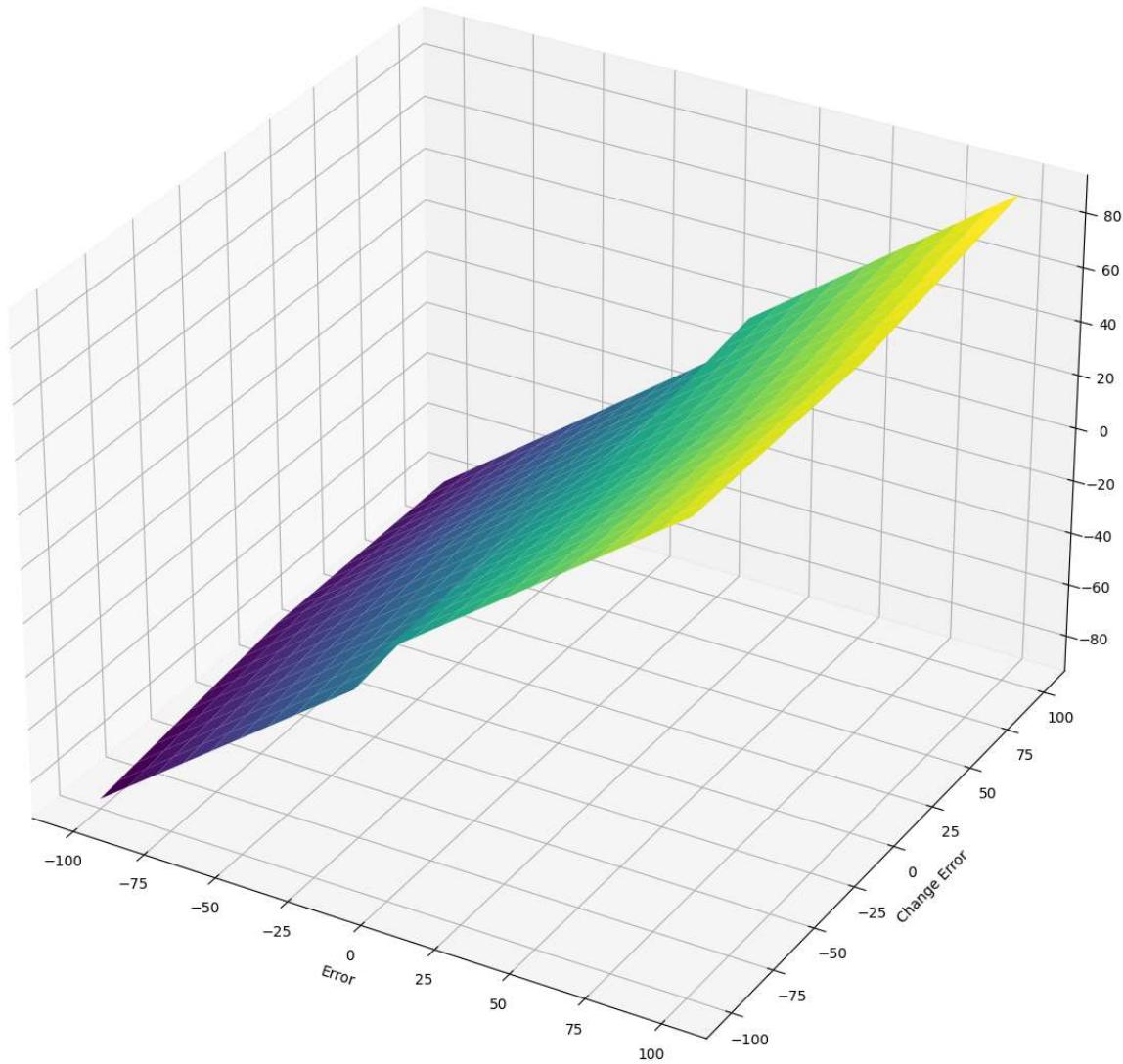
Elapsed time is 1.364364 seconds.

```
In [ ]: 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)
```

Surface Response: Takagi-Sugeno Two Inputs



Elapsed time is 0.576296 seconds.

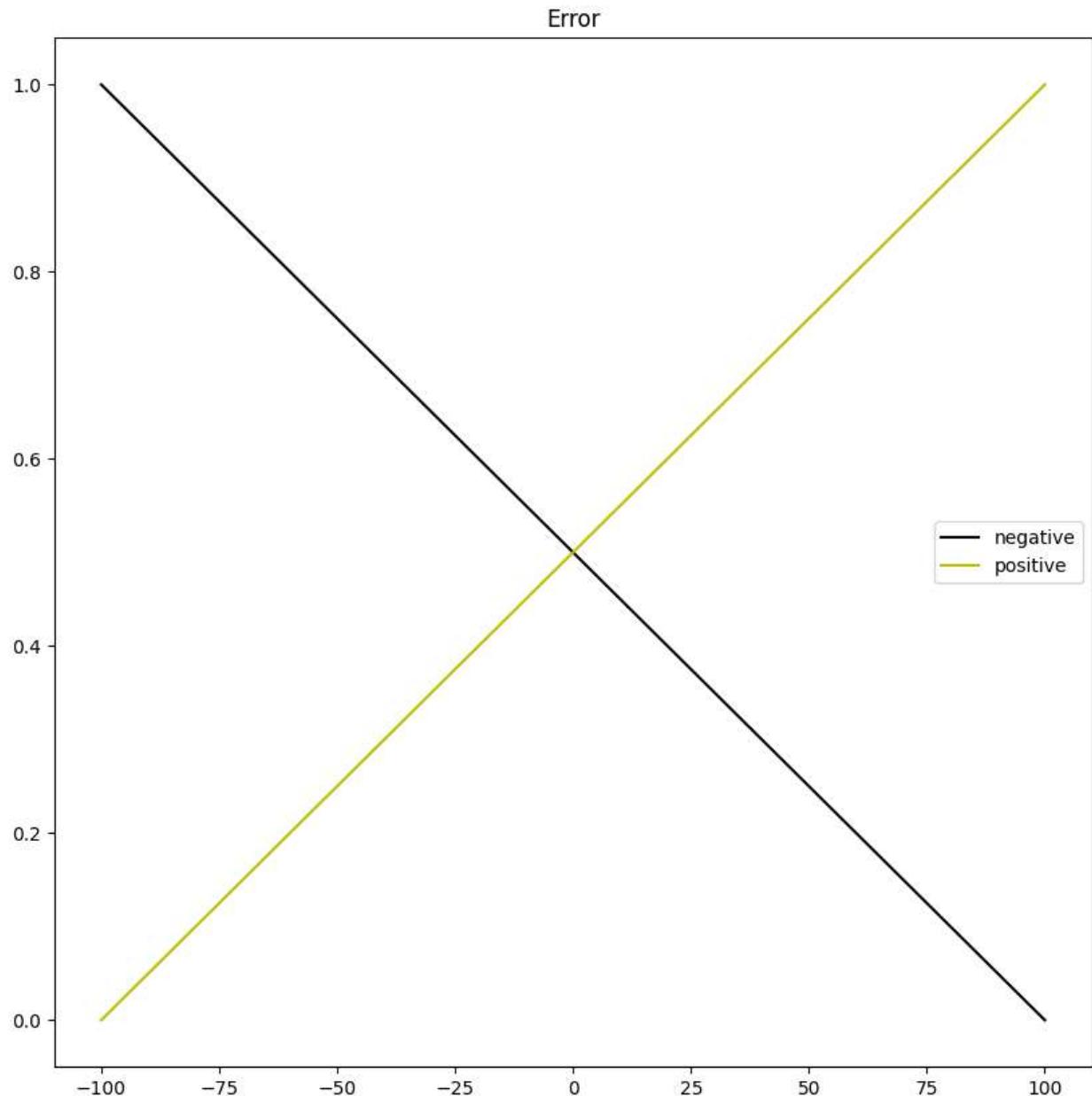
Sistema Difuso Lineal Una Entrada

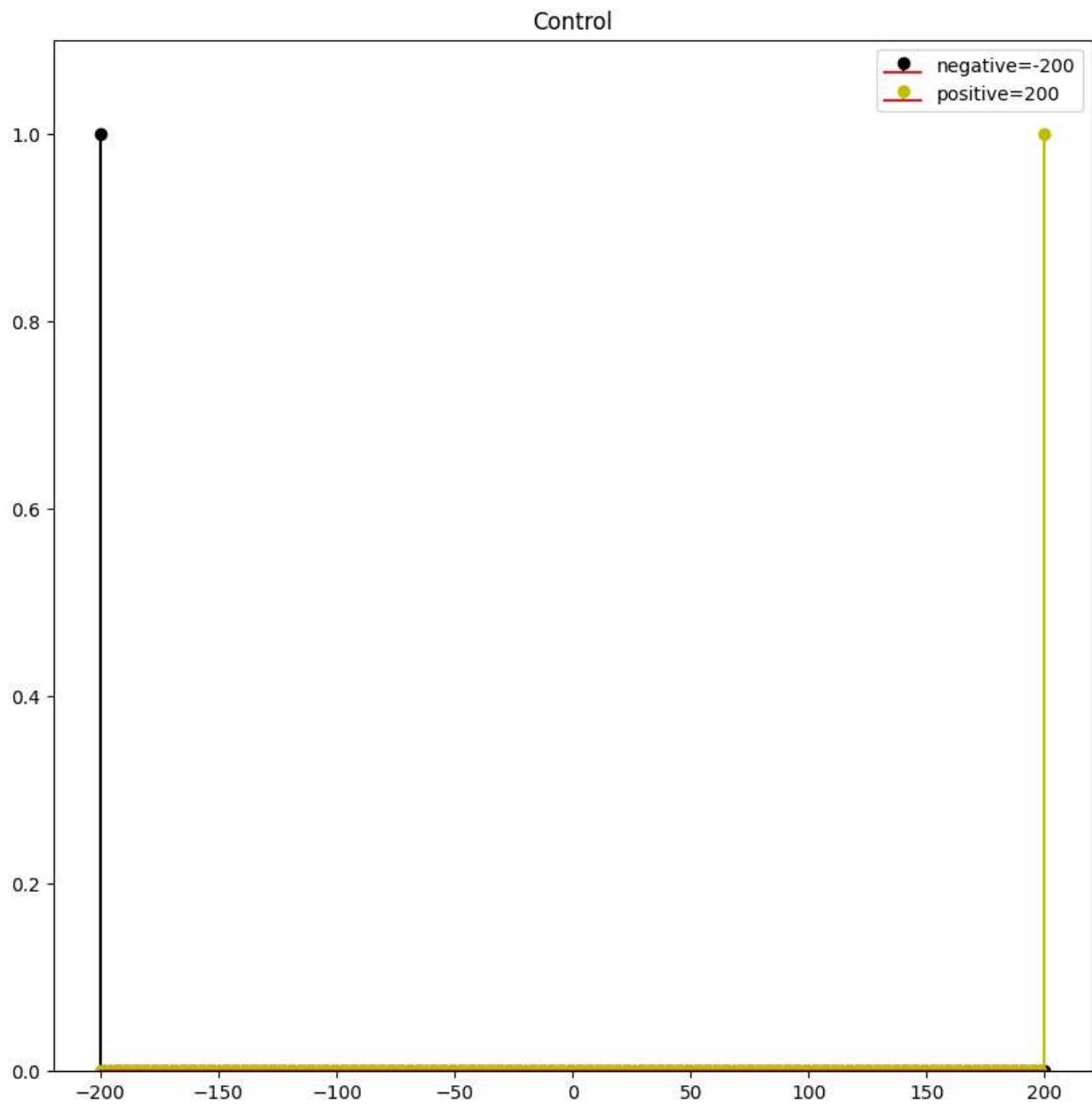
```
In [ ]: t.tic()
Error_universe = fuzzy_universe('Error', np.arange(-100,101,1), 'continuous')
Error_universe.add_fuzzyset('negative','trimf',[ -200,-100,100])
Error_universe.add_fuzzyset('positive','trimf',[ -100,100,200])
Error_universe.view_fuzzy()

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')
```

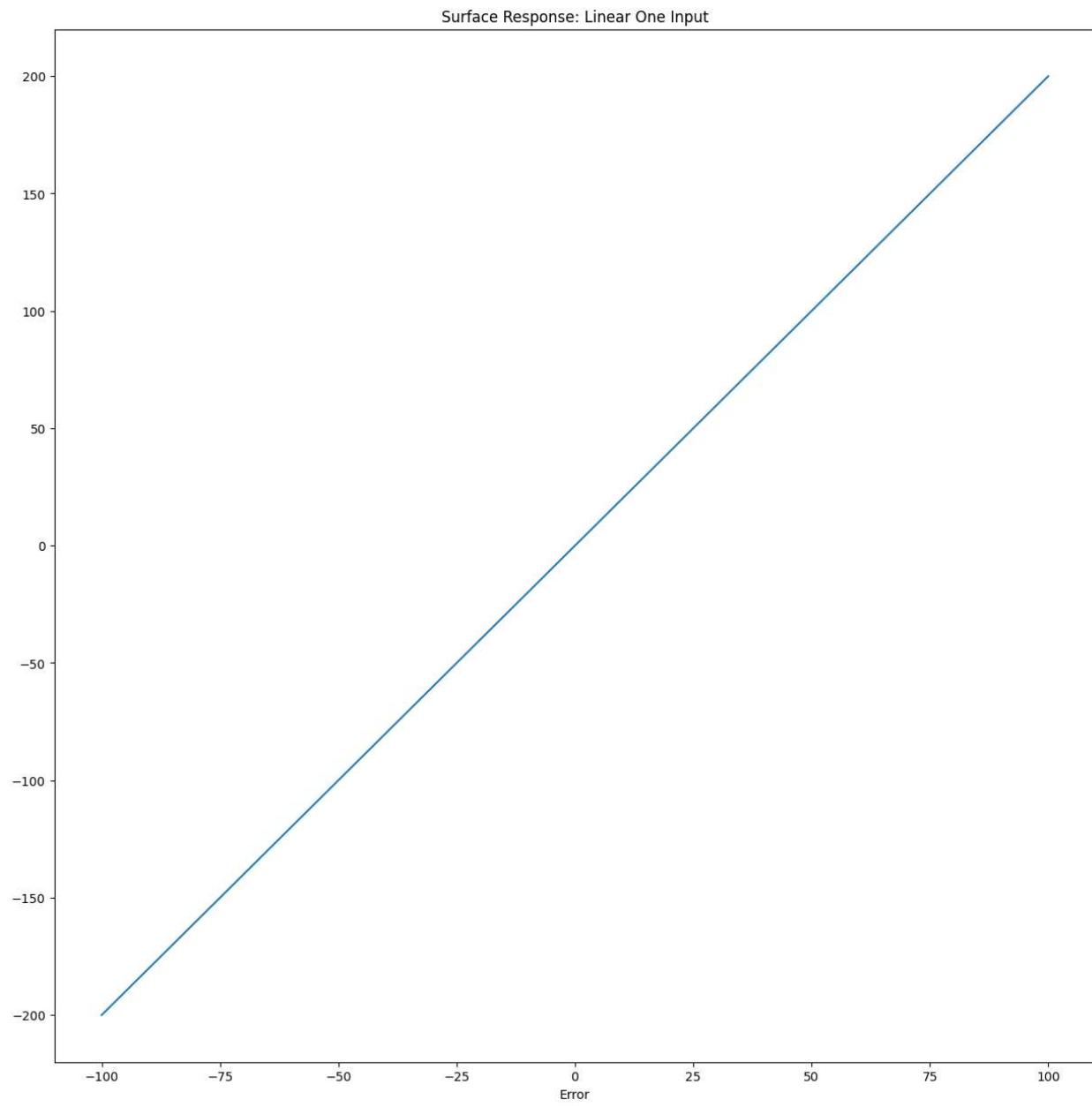
```
LinearP.build()  
t.toc()  
  
del Error_universe  
del Control_universe
```





Elapsed time is 0.828531 seconds.

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



Elapsed time is 0.663082 seconds.

Linear Fuzzy System Two Inputs

```
In [ ]: 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)

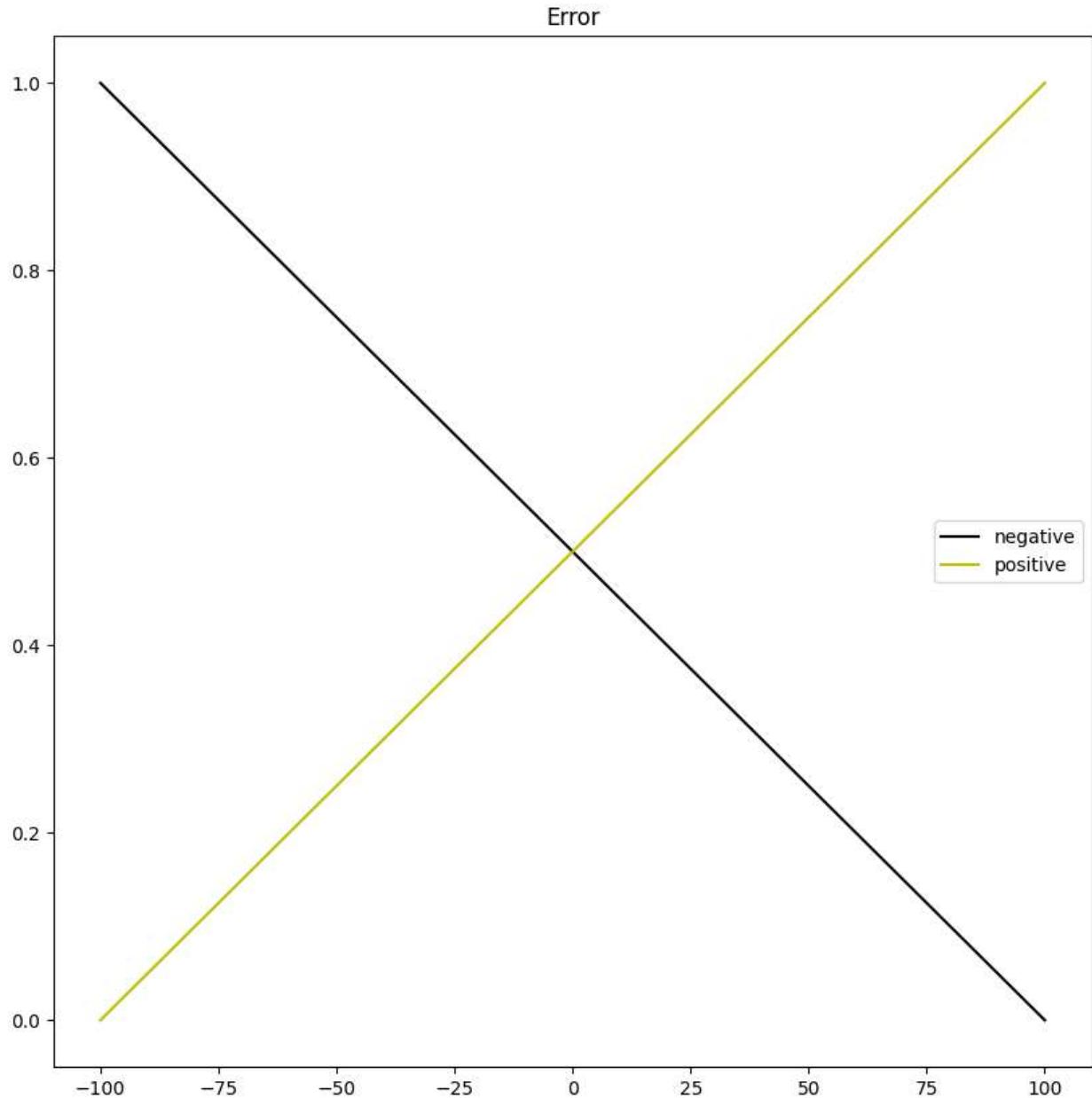
Linear.add_rule([[['Error', 'negative'], ['Change Error', 'negative']], ['and'], [['Control', 'negative']]])
Linear.add_rule([[['Error', 'negative'], ['Change Error', 'positive']], ['and'], [['Control', 'zero']]])
Linear.add_rule([[['Error', 'positive'], ['Change Error', 'negative']], ['and'], [['Control',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

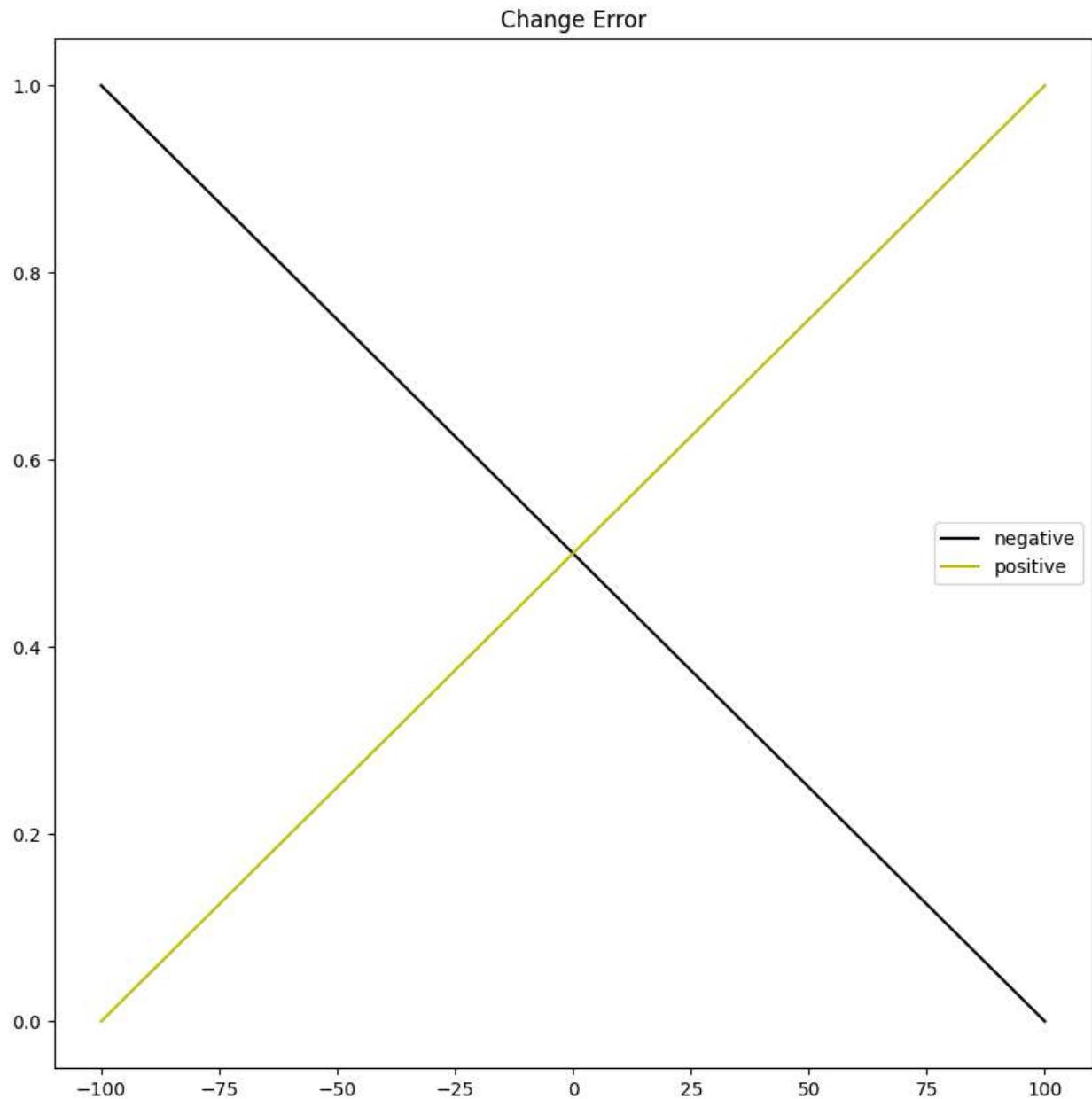
Linear.configure('Linear')

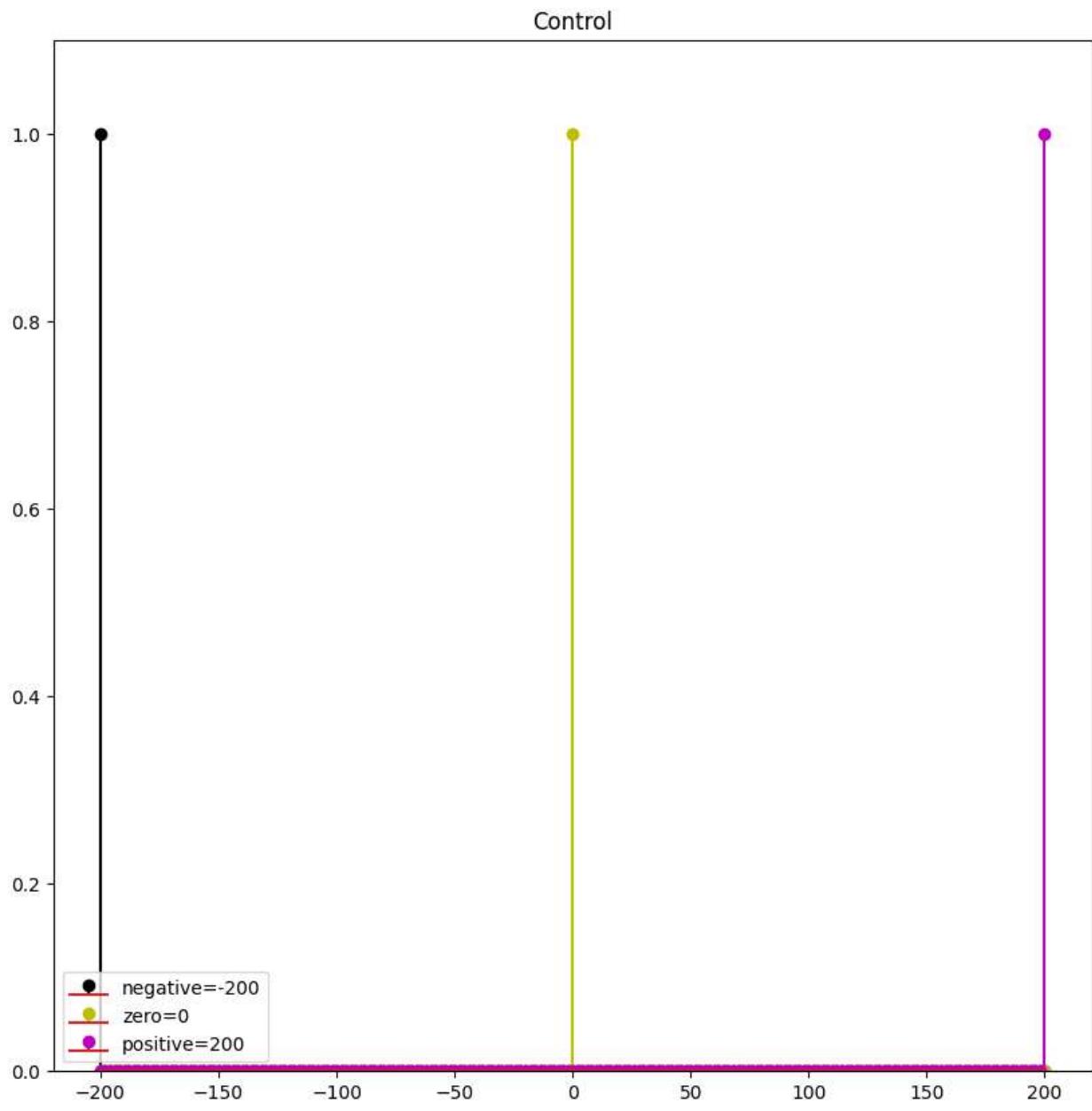
Linear.build()

t.toc()
del Error_universe
del ChError_universe
del Control_universe

```







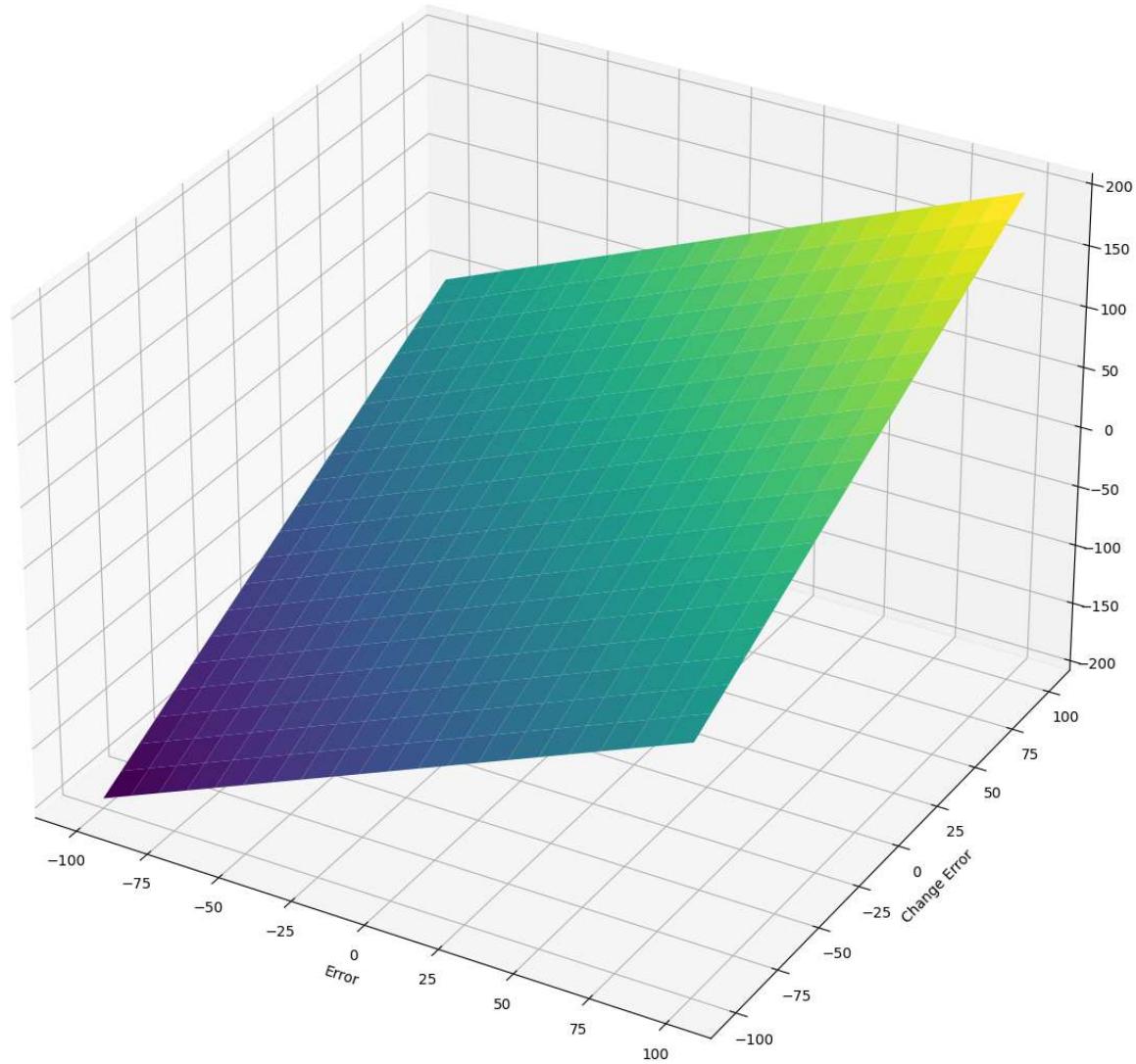
Elapsed time is 1.292930 seconds.

```
In [ ]: 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)
```

Surface Response: Linear



Elapsed time is 0.542930 seconds.

Fuzzy Controllers

Definición del Sistema (Posición Motor CD)

```
In [ ]: 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 0.007975 seconds.

Mamdani Controller One Input

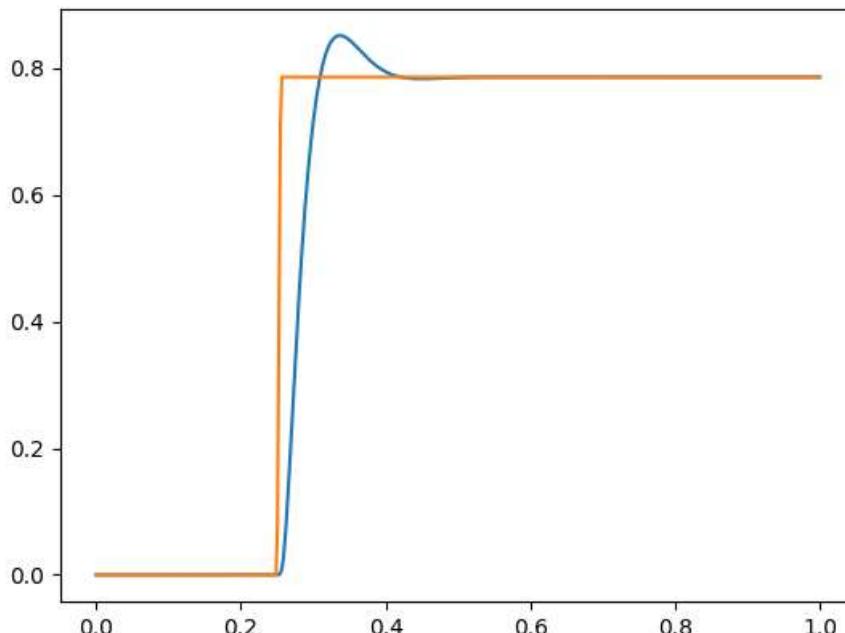
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(Mamdani1,typec='Fuzzy1',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.008941 seconds.

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

Elapsed time is 2.297757 seconds.

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



Elapsed time is 0.121022 seconds.

Mamdani Controller Two Inputs

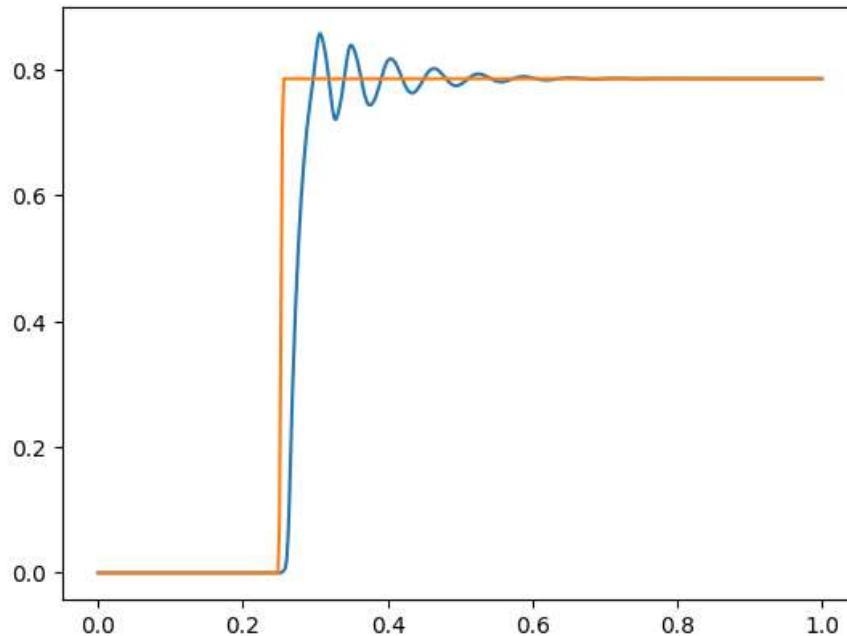
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(Mamdani2,typec='Fuzzy2',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.006433 seconds.

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

```
t.toc()
Elapsed time is 6.142238 seconds.
```

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



Elapsed time is 0.114060 seconds.

F.L. Smith Controller One Input

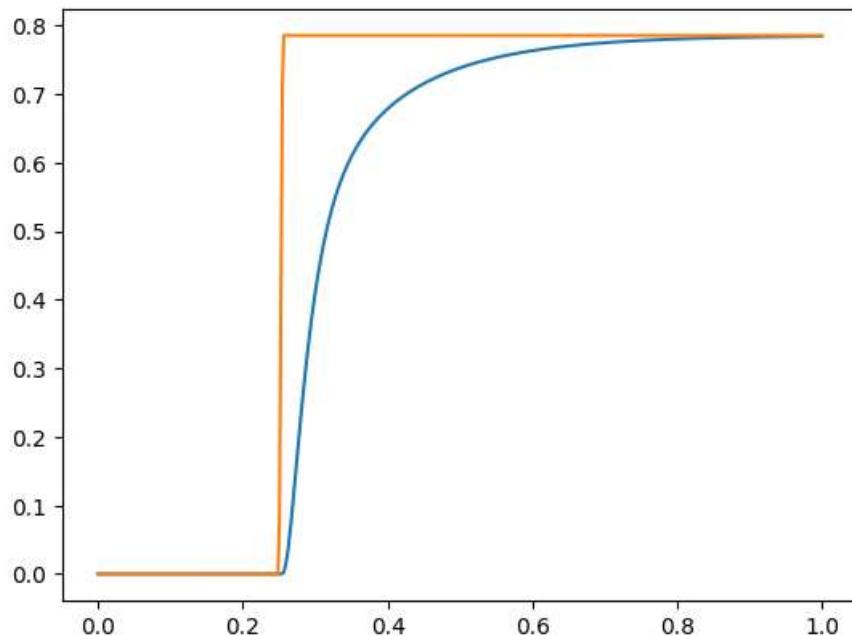
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(FLS1,typec='Fuzzy1',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.003768 seconds.

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

Elapsed time is 2.042734 seconds.

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



Elapsed time is 0.140849 seconds.

F.L. Smidth Controller Two Inputs

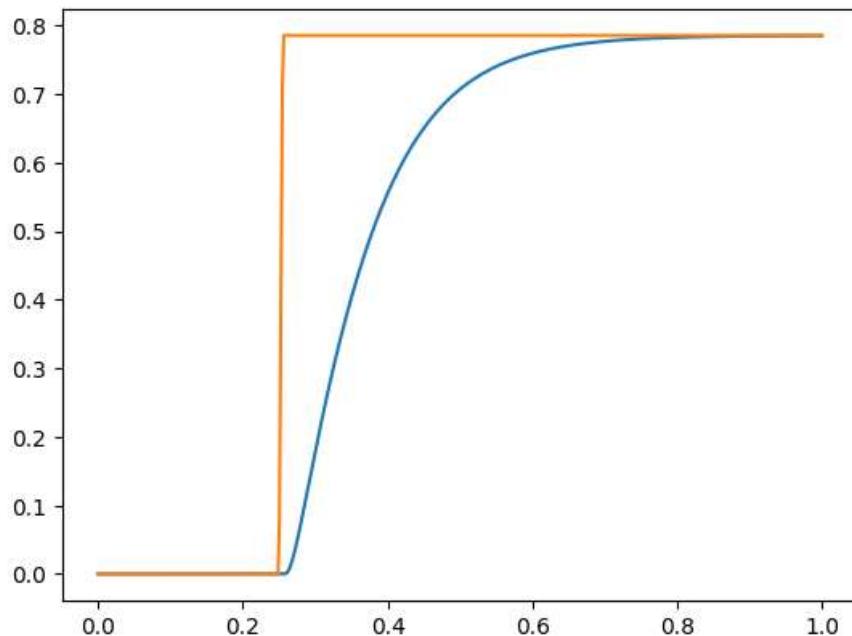
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(FLS2,typec='Fuzzy2',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.005357 seconds.

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

Elapsed time is 5.004740 seconds.

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



Elapsed time is 0.147617 seconds.

Takagi-Sugeno Controller One input

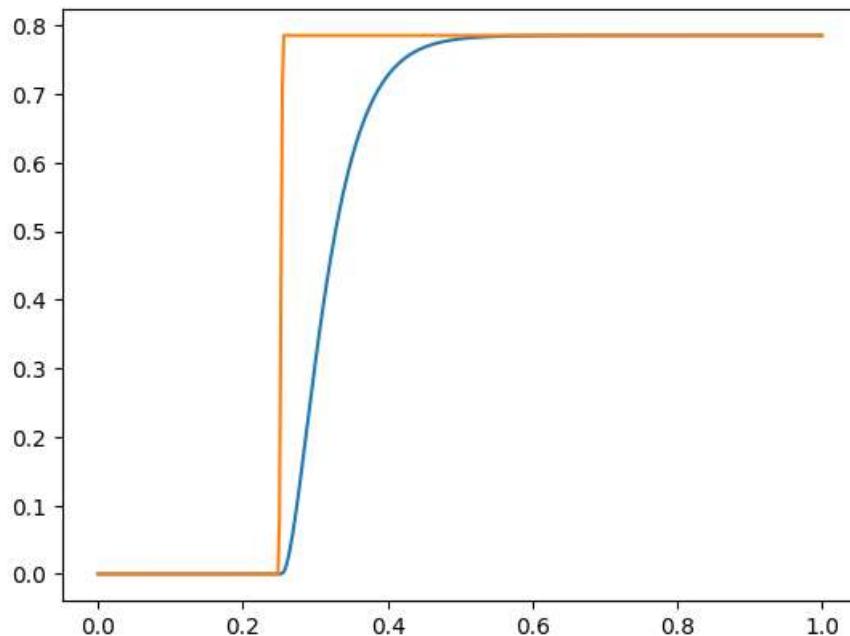
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(TSG1,typec='Fuzzy1',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.004462 seconds.

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

Elapsed time is 0.946106 seconds.

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



Elapsed time is 0.123291 seconds.

Takagi-Sugeno Controller Two Inputs

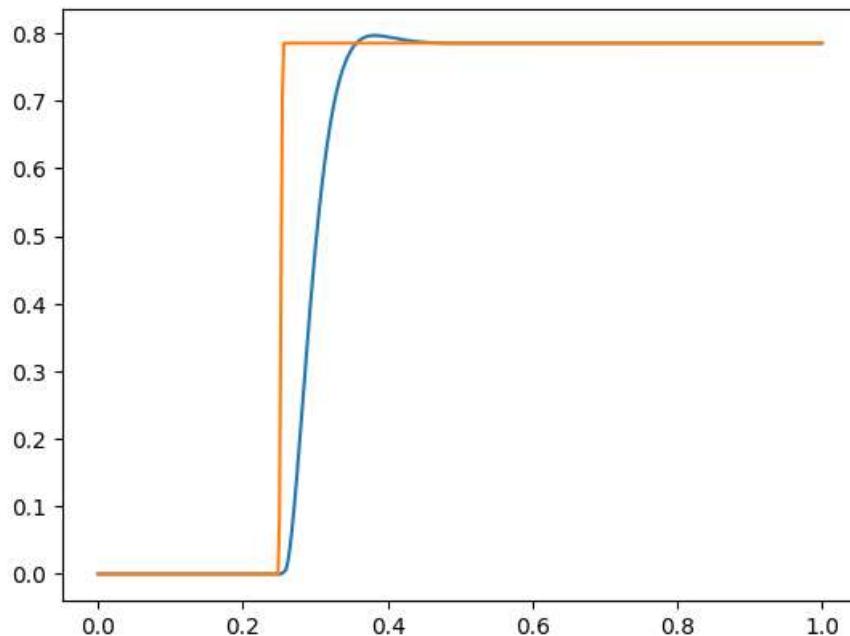
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(TSG2,typec='Fuzzy2',tf=TF,DT = T[1])
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.004541 seconds.

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

Elapsed time is 5.672765 seconds.

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



Elapsed time is 0.123270 seconds.

Linear Proportional Fuzzy Controller

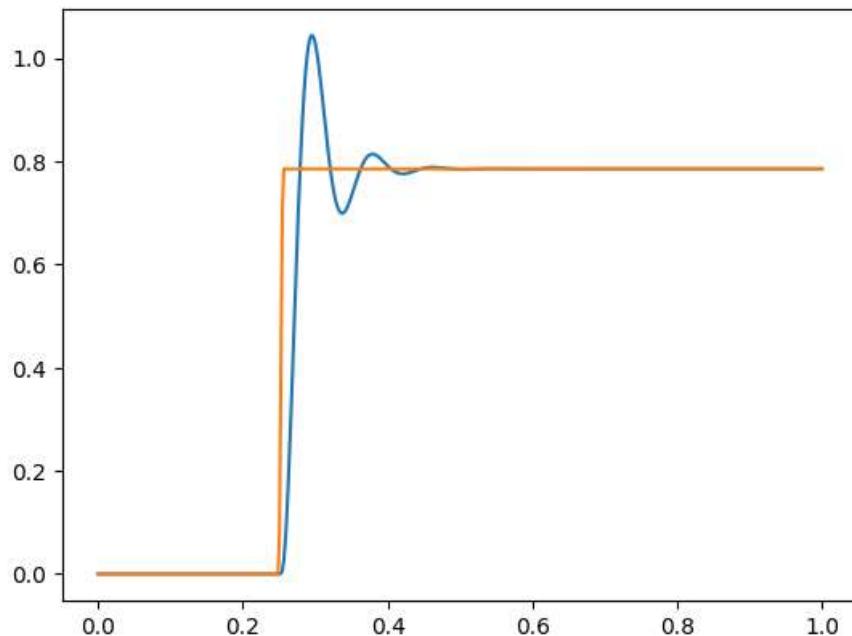
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(LinearP,typec='P',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.005195 seconds.

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

Elapsed time is 0.810658 seconds.

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



Elapsed time is 0.118650 seconds.

Derivative Linear Proportional Proportional Fuzzy Controller

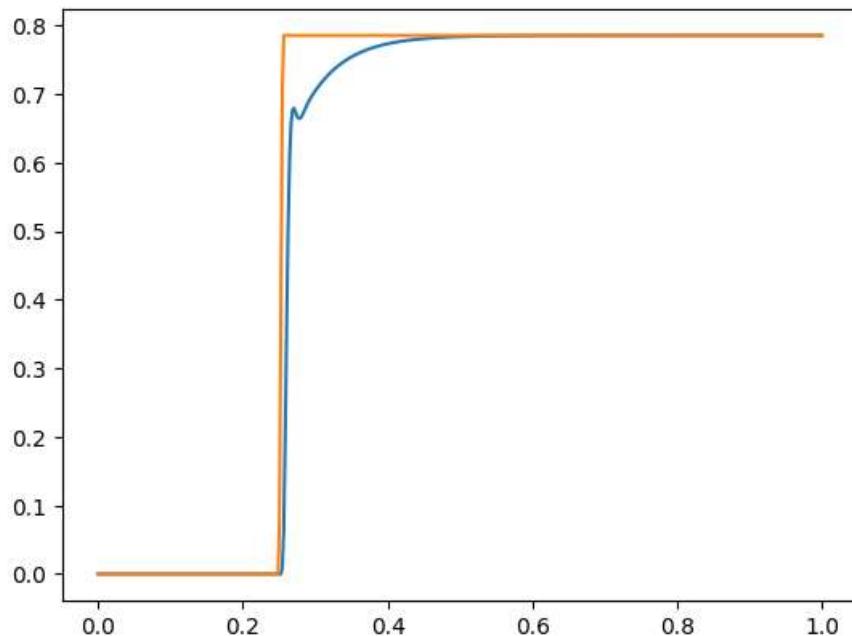
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(Linear,typec='PD',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE=0.6
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.006150 seconds.

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

Elapsed time is 1.502865 seconds.

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



Elapsed time is 0.132070 seconds.

Linear Proportional Derivative-Integral Fuzzy Proportional Controller

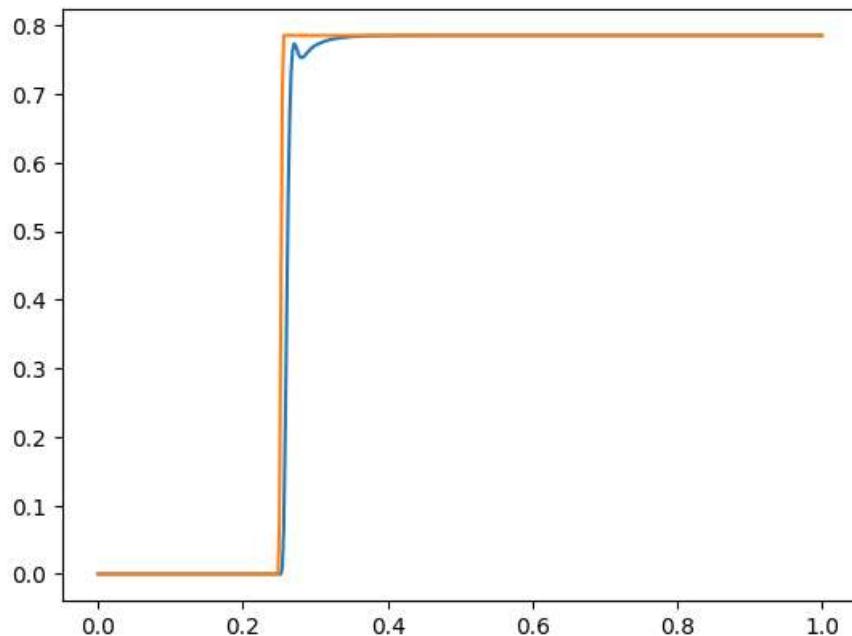
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(Linear,typec='PD-I',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE=0
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.003939 seconds.

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

Elapsed time is 1.581946 seconds.

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



Elapsed time is 0.127647 seconds.

Elapsed time is 44.450140 seconds.

Mamdani Proportional Fuzzy Proportional Controller

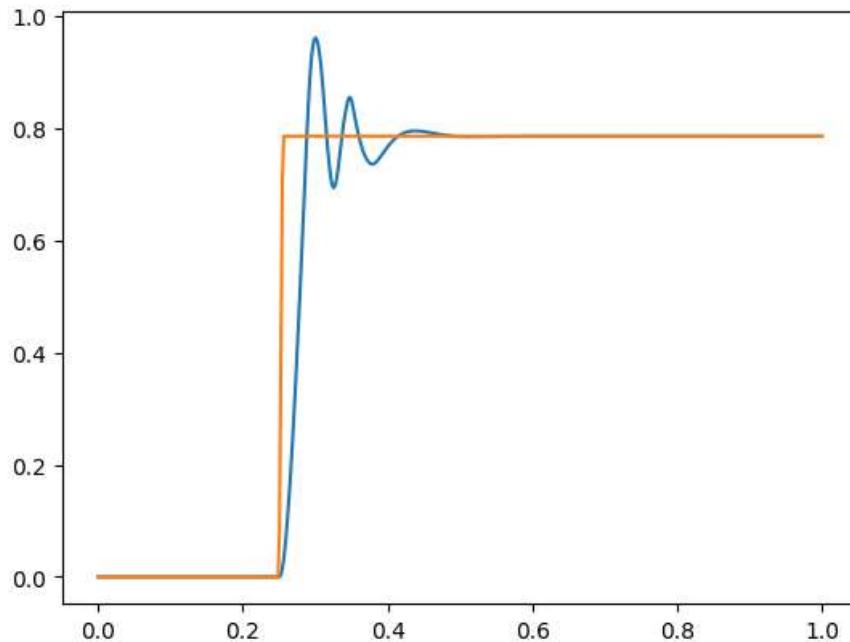
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(Mamdani1,typec='P',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.005755 seconds.

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

Elapsed time is 1.945566 seconds.

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



Elapsed time is 0.219610 seconds.

Mamdani Derivative Proportional Fuzzy Proportional Fuzzy Controller

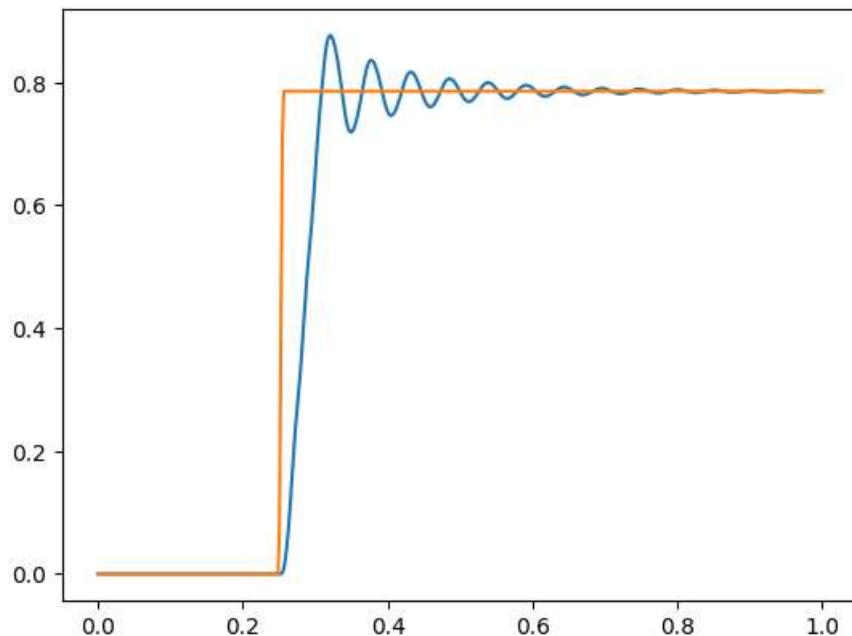
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(Mamdani2,typec='PD',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE=0
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.005032 seconds.

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

Elapsed time is 6.270452 seconds.

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



Elapsed time is 0.115674 seconds.

Derivative Proportional Fuzzy Proportional Controller - Integral Mamdani

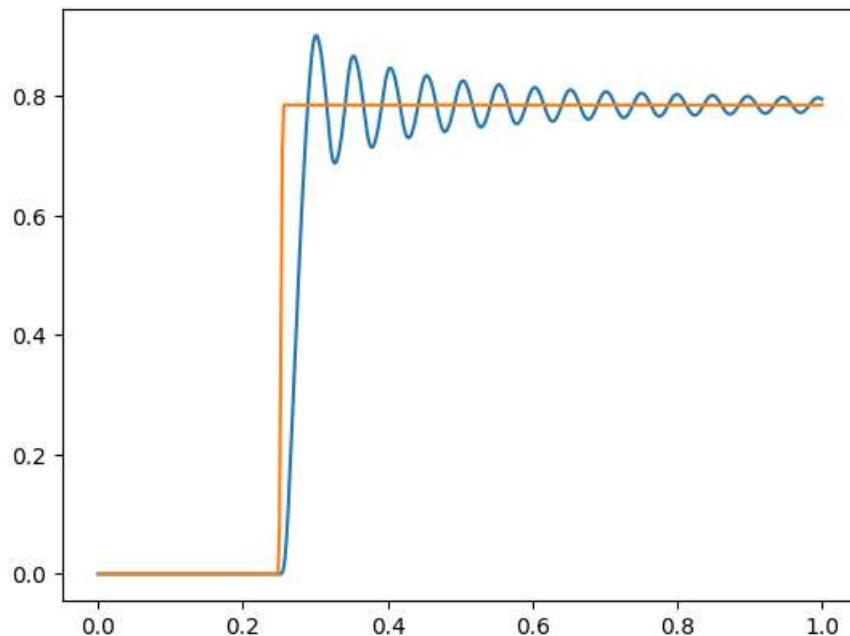
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(Mamdani2,typec='PD-I',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.003680 seconds.

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

Elapsed time is 6.422633 seconds.

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



Elapsed time is 0.117112 seconds.

Elapsed time is 59.989536 seconds.

F. L. Smith Fuzzy Proportional Fuzzy Controller

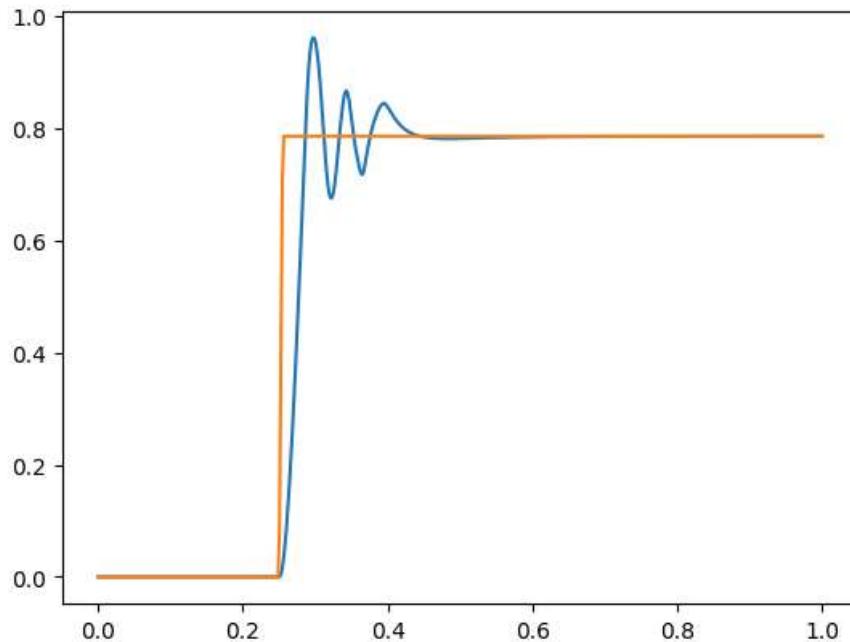
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(FLS1,typec='P',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.004368 seconds.

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

Elapsed time is 2.278017 seconds.

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



Elapsed time is 0.115263 seconds.

F. L. Smith Derivative Proportional Fuzzy Proportional Controller

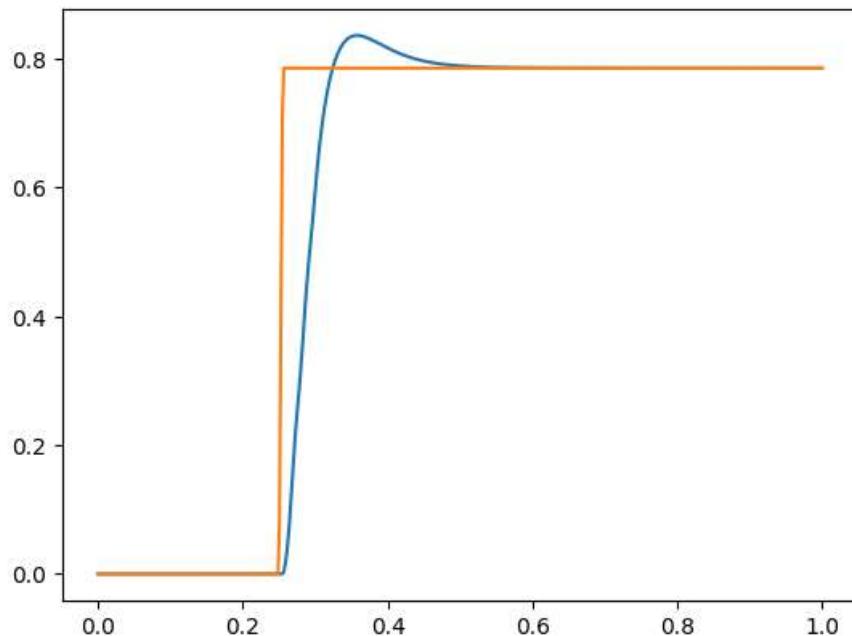
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(FLS2,typec='PD',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE=0.636
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.004187 seconds.

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

Elapsed time is 5.021214 seconds.

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



Elapsed time is 0.118756 seconds.

Derivative Proportional Fuzzy Proportional Controller - Integral F. L. Smidh

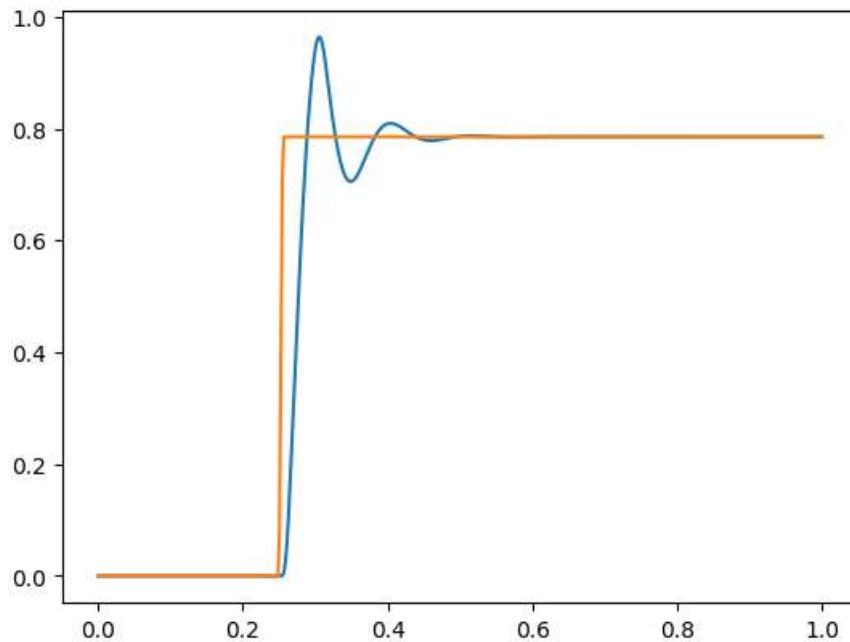
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(FLS2,typec='PD-I',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE=0.6
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.004832 seconds.

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

Elapsed time is 4.879782 seconds.

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



Elapsed time is 0.131521 seconds.

Elapsed time is 72.977505 seconds.

Takagi-Sugeno Proportional Fuzzy Proportional Controller

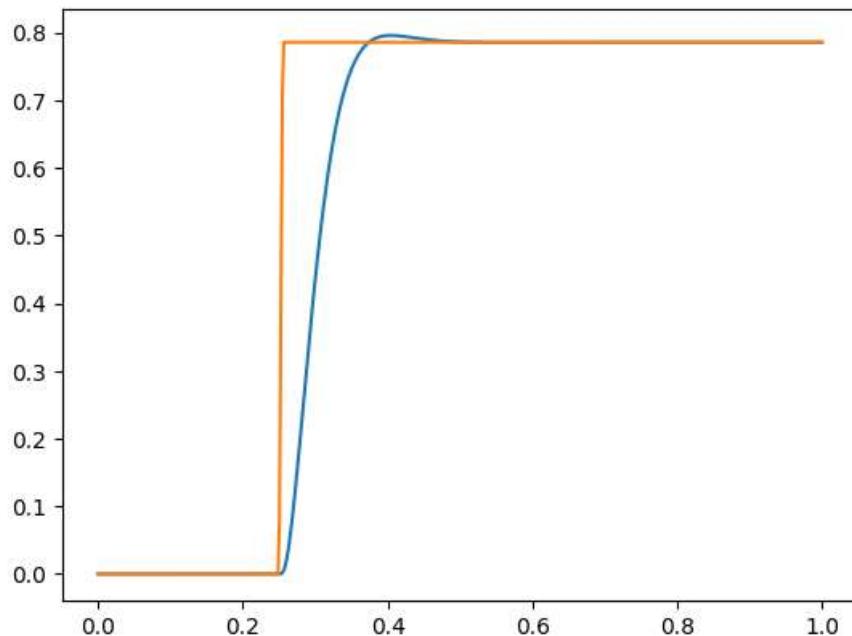
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(TSG1,typec='P',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248)
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.004373 seconds.

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

Elapsed time is 0.988998 seconds.

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



Elapsed time is 0.174716 seconds.

Takagi-Sugeno Derivative Proportional Diffuse Proportional Fuzzy Controller

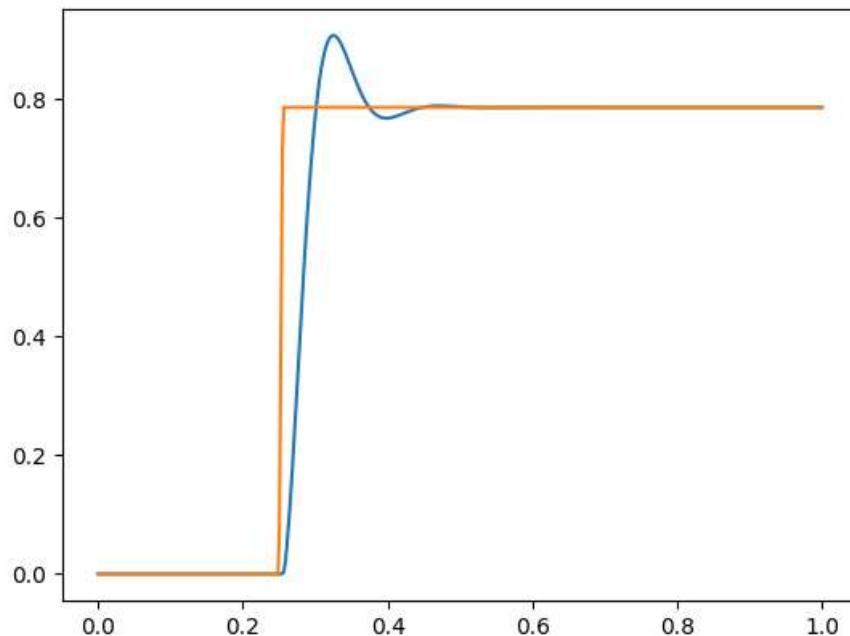
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(TSG2,typec='PD',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE=0.636
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.007282 seconds.

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

Elapsed time is 5.672214 seconds.

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



Elapsed time is 0.114377 seconds.

Derivative Proportional Fuzzy Proportional Controller - Integral Takagi-Sugeno

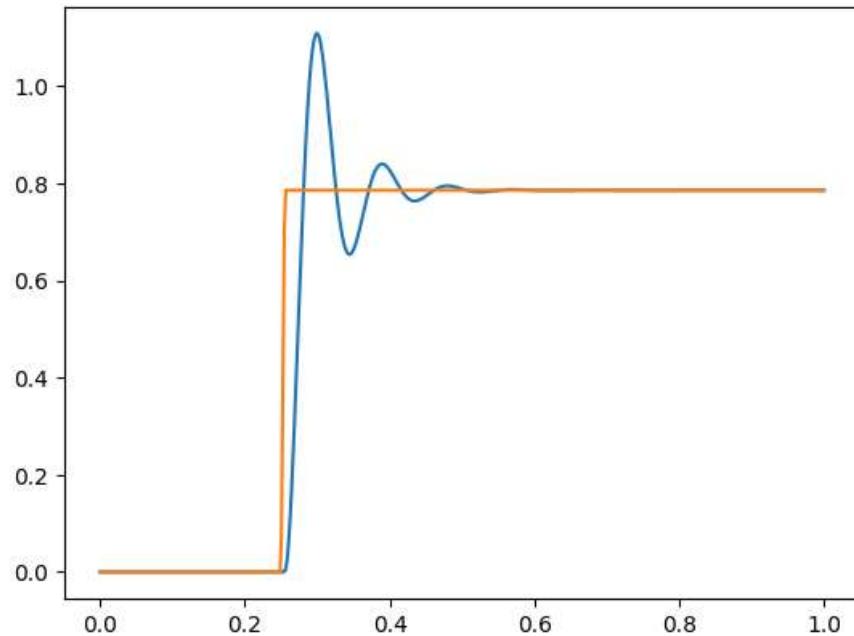
```
In [ ]: t.tic()
PidFuzzController = fuzzy_controller(TSG2,typec='PD-I',tf=TF,DT = T[1], GE=15.91545709, GU=0.094248, GCE=0.6
PidFuzzController.build()
PidFuzzControllerBlock = PidFuzzController.get_controller()
PidFuzzSystemBlock = PidFuzzController.get_system()
t.toc()
```

Elapsed time is 0.006175 seconds.

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

Elapsed time is 5.691887 seconds.

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



Elapsed time is 0.108522 seconds.

Elapsed time is 86.235713 seconds.