0

0

5

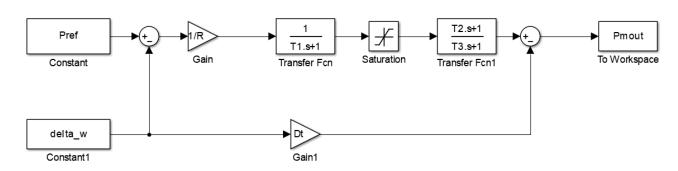


Figure 1: Tgov1 Simulink Model MATLAB vs Python Tgov1 Simulation with Zero Initial Conditions Simulink SciPy 0.5 Pref Input 0 10 5 15 20 25 30 **Absolute Difference** 0.01 0.005 0 5 10 20 25 15 30 **Relative Difference** Percent N

Figure 2: Simulation Comparison

15

20

25

30

10

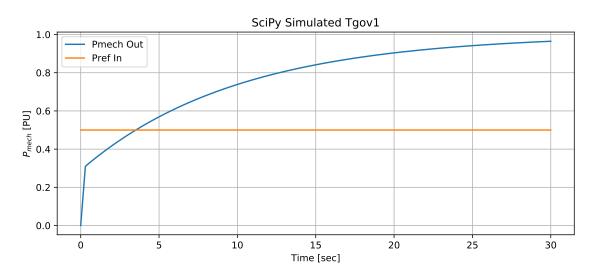


Figure 3: matplotlib Output

Thad Haines Page 2 of 2

## Python Code:

```
import numpy as np
   import scipy.signal as sig
   import matplotlib.pyplot as plt
4
   import scipy.io as sio
5
6
   # Inputs
   Pref = .5 # will be a PU of Pref from Generator
8
   delta_w = 0.0
9
   # Simulation Parameters
   t = np.linspace(0,30,101)
11
12
   R = 0.05
13
   Vmax = 1.0
   Vmin = 0.0
14
15
   T1 = 0.5
16 \mid T2 = 3.0
17 \mid T3 = 10.0
18 \mid Dt = 0.0
19
20
   # System Creations
21
   sys1 = sig.TransferFunction(1,[T1, 1])
22
   sys2 = sig.TransferFunction([T2, 1],[T3, 1])
23
24
   # Input to system
25
   u = (Pref-delta_w)/R
26
27
   # First Block
   _{-}, y1, x1 = sig.lsim2(sys1, [u]*t.size, t)
28
29
30
   # limit Valve position
31
   for x in range(t.size):
32
       if y1[x]>Vmax:
33
            y1[x] = Vmax
34
       elif y1[x] < Vmin:</pre>
            y1[x] = Vmin
36
   # Second block
37
   _, y2, x2 = sig.lsim2(sys2, y1, t)
38
   # Addition of damping
40
41
   y2 = y2 + [delta_w*Dt]*y2.size
42
43
   # Plot output
   plt.plot(t,y2, label="Pmech Out")
44
   plt.plot(t,[u*R]*t.size, label="Pref In")
45
46
   plt.title('SciPy Simulated Tgov1')
47
   plt.ylabel(r'$P_{mech}$ [PU]')
   plt.xlabel('Time [sec]')
48
   plt.grid()
49
50
   plt.legend()
51
   plt.show()
52
53
   # Output data dictionary as .mat
   pyTgov = {'t_py': t,}
54
              'y_py': y2,}
   sio.savemat('tgovTest', pyTgov)
56
```