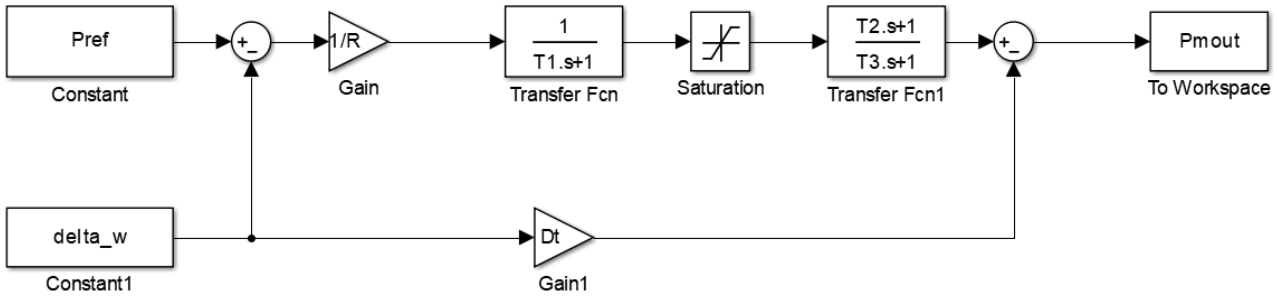

Figure 1: Tgov1 Simulink Model

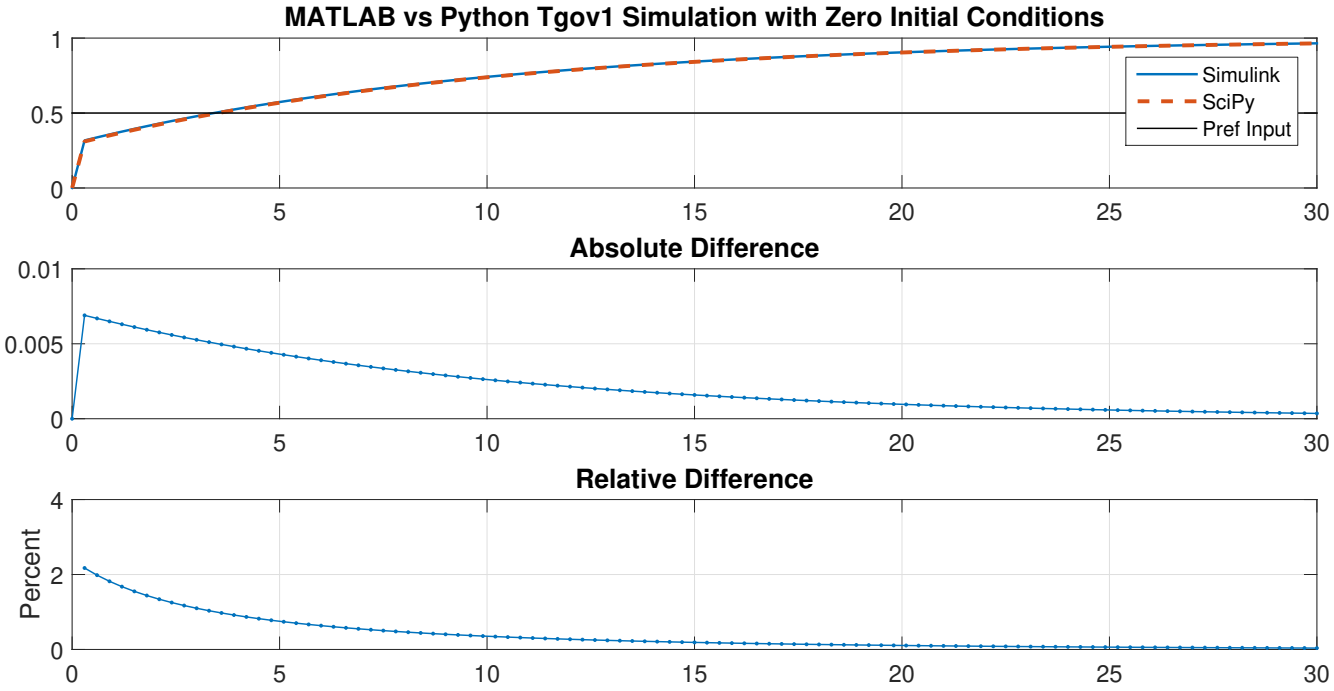

Figure 2: Simulation Comparison

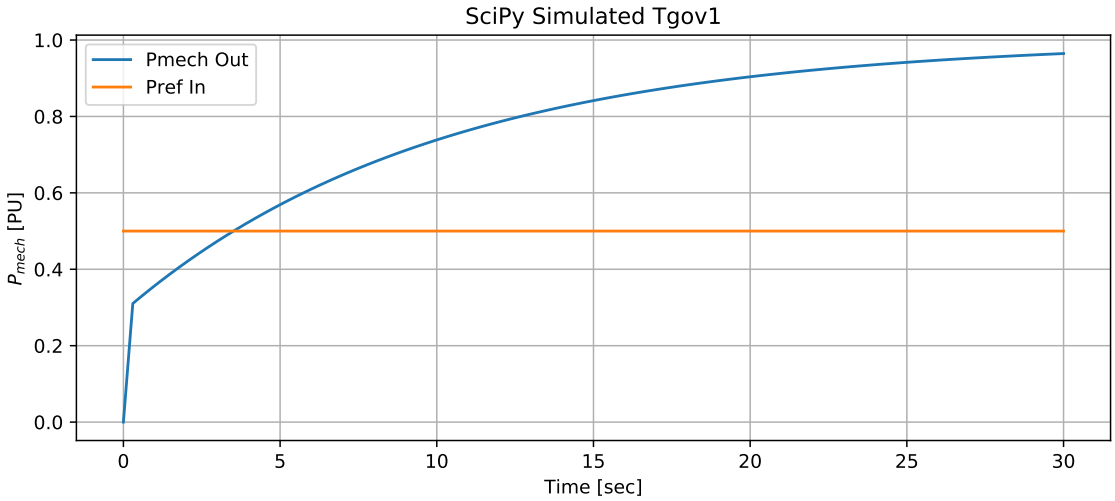

Figure 3: matplotlib Output

Python Code:

```python
import numpy as np
import scipy.signal as sig
import matplotlib.pyplot as plt
import scipy.io as sio

# Inputs
Pref = .5 # will be a PU of Pref from Generator
delta_w = 0.0

# Simulation Parameters
t = np.linspace(0,30,101)
R = 0.05
Vmax = 1.0
Vmin = 0.0
T1 = 0.5
T2 = 3.0
T3 = 10.0
Dt = 0.0

# System Creations
sys1 = sig.TransferFunction(1,[T1, 1])
sys2 = sig.TransferFunction([T2, 1],[T3, 1])

# Input to system
u = (Pref-delta_w)/R

# First Block
_, y1, x1 = sig.lsim2(sys1, [u]*t.size, t)

# limit Valve position
for x in range(t.size):
    if y1[x]>Vmax:
        y1[x] = Vmax
    elif y1[x]<Vmin:
        y1[x] = Vmin

# Second block
_, y2, x2 = sig.lsim2(sys2, y1, t)

# Addition of damping
y2 = y2 + [delta_w*Dt]*y2.size

# Plot output
plt.plot(t,y2, label="Pmech Out")
plt.plot(t,[u*R]*t.size, label="Pref In")
plt.title('SciPy Simulated Tgov1')
plt.ylabel(r'$P_{mech}$ [PU]')
plt.xlabel('Time [sec]')
plt.grid()
plt.legend()
plt.show()

# Output data dictionary as .mat
pyTgov = {'t_py': t,
          'y_py': y2,}
sio.savemat('tgovTest', pyTgov)
```