Python Control Systems

There is a fantastic Python control systems toolbox that was developed to have many of the same capabilities as the MATLAB toolbox (https://python-control.readthedocs.io/en/0.9.0/ (https://python-control.readthedocs.io/en/0.9.0/)). The code shown here is largely an extension of that code written to make the original toolbox a little easier to use. If you would like you can use that toolbox directly.

The first step in using the code will be to "import" it into your current working file. The file containing the class code should be in the same directory you are currently working in (e.g. both file smust be in C:\Users\Andrew\Desktop\Process Control if that is where you are located).

In [1]:
```
from ProcessModelingAndControlClass import * # this will allow you to use the
```

Example 1: $G(s) = \frac{1}{s+1}$ with step function inputs

First you should define a transfer function using the command "TransferFunction". This has four main arguments 1) the numerator, 2) the denominator, 3) time delay value and 4) a custom name for your system. The numerator and denominator values are the coefficients of the polynomial starting with the largest order (e.g. $s^2 + 2s + 1$ would be written as [1, 2, 1])

Defining the transfer function immediately computes the poles and zeros and displays them.

In [2]:
```
sys1 = TransferFunction(Numerator = 1, Denominator = [1,1], Systemlabel = 'Sy
```

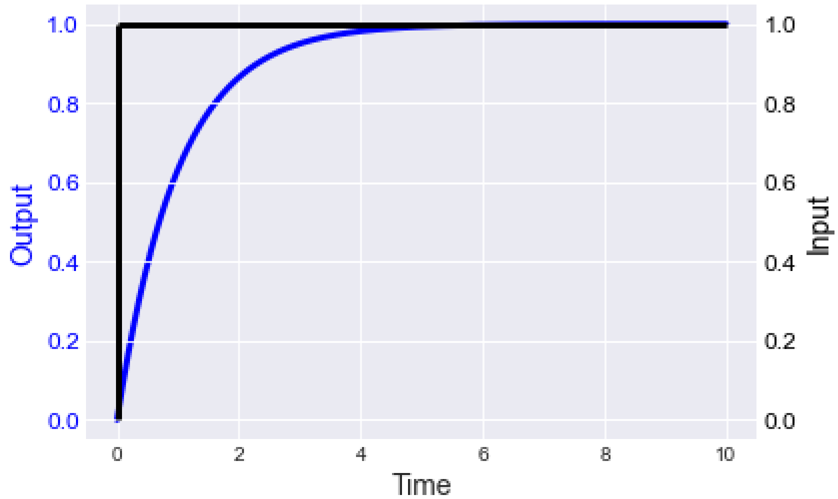
```
######################################################
##### System 1 Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################
```

Next you can define an input. You must choose the magnitude of the input, the type of the input, and the input end time if a square input function is chosen

In [3]:
```
sys1.InputFunction(Magnitude = 1, Type = 'Step')
```

Once you have defined a transfer function and an input you can plot the output of the system. The output will be displayed on the left vertical axis in color and the input will be displayed on the right vertical axis in black.

In [4]:
```
PlotResponse(sys1)
```

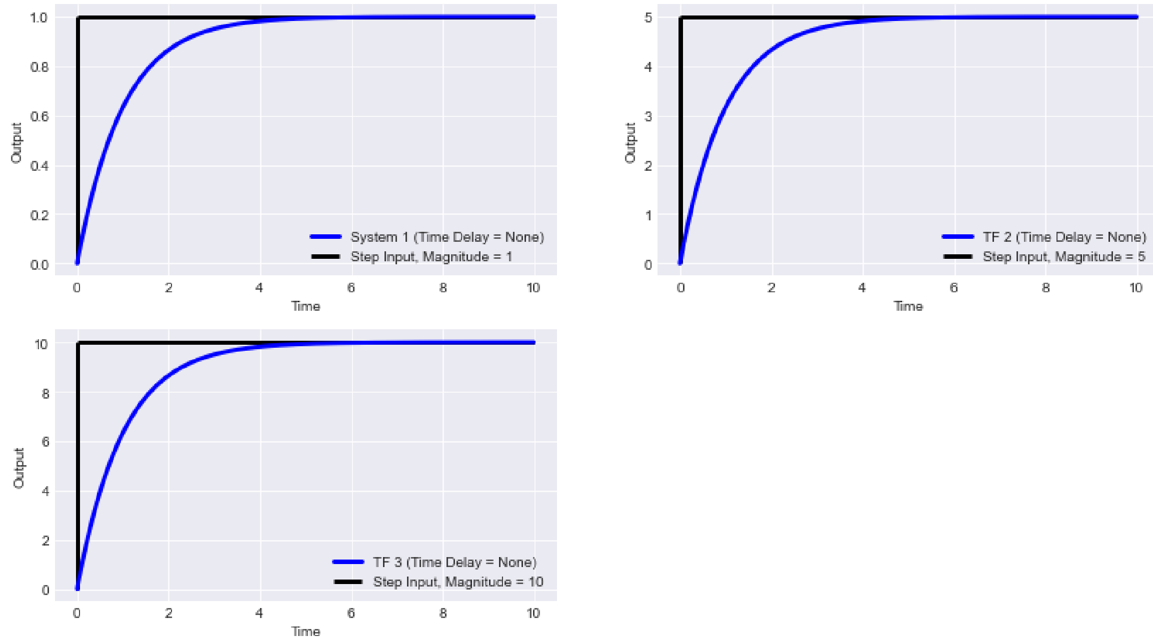


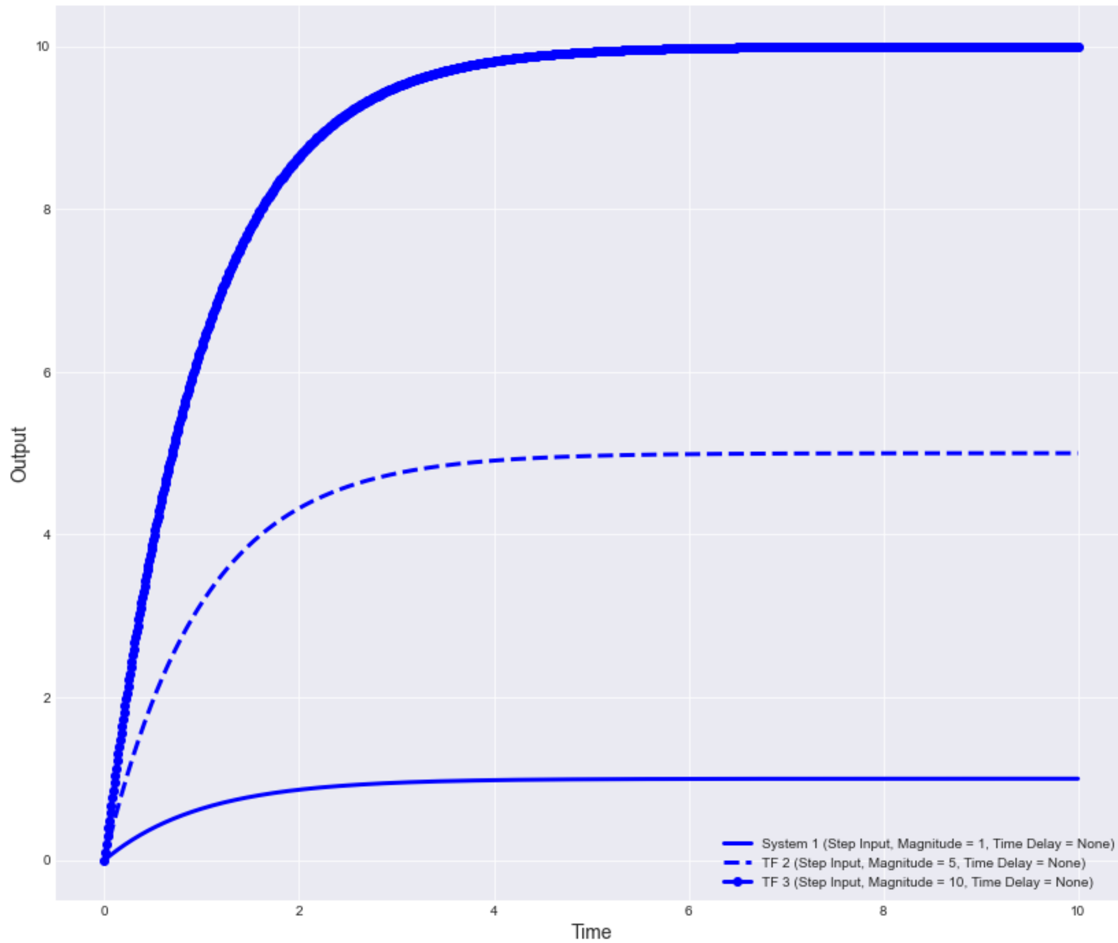
Multiple transfer functions may be defined and compared in one figure. You also can define the numerator and denominator directly without using the commands "numerator" and "denominator" as long as they are in the correct order.

In [5]:
```
sys2 = TransferFunction(1, [1,1])
sys3 = TransferFunction(1, [1,1])

sys2.InputFunction(Magnitude = 5, Type = 'Step')
sys3.InputFunction(Magnitude = 10, Type = 'Step')
```

```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################

######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################
```

In [6]:
```
CompareResults(sys1, sys2, sys3)
```

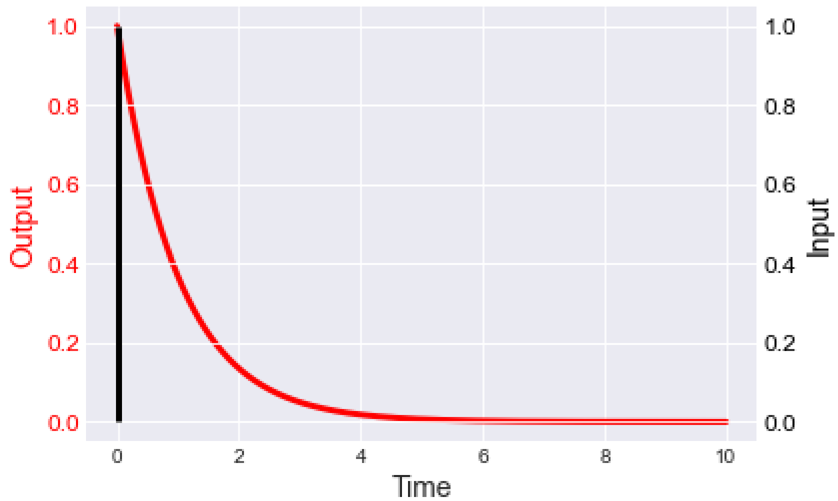




Example 2: $G(s) = \frac{1}{s+1}$ with impulse function inputs

In [7]:
```
sys1 = TransferFunction(1, [1,1])
sys1.InputFunction(Magnitude = 1, Type = 'Impulse')

PlotResponse(sys1)
```

```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################
```

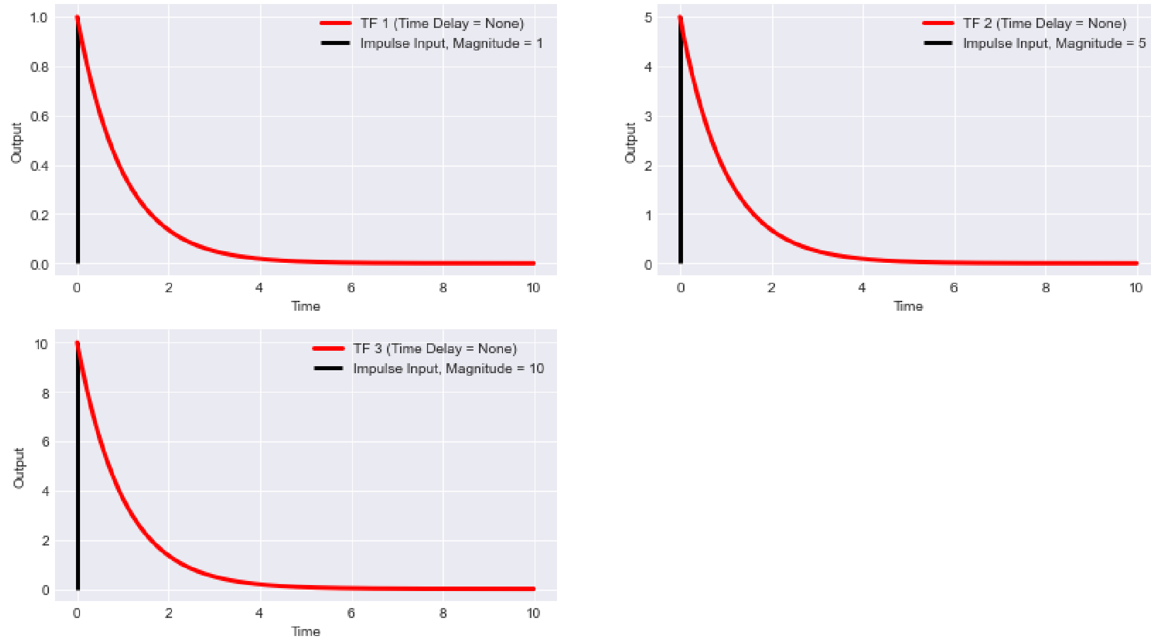


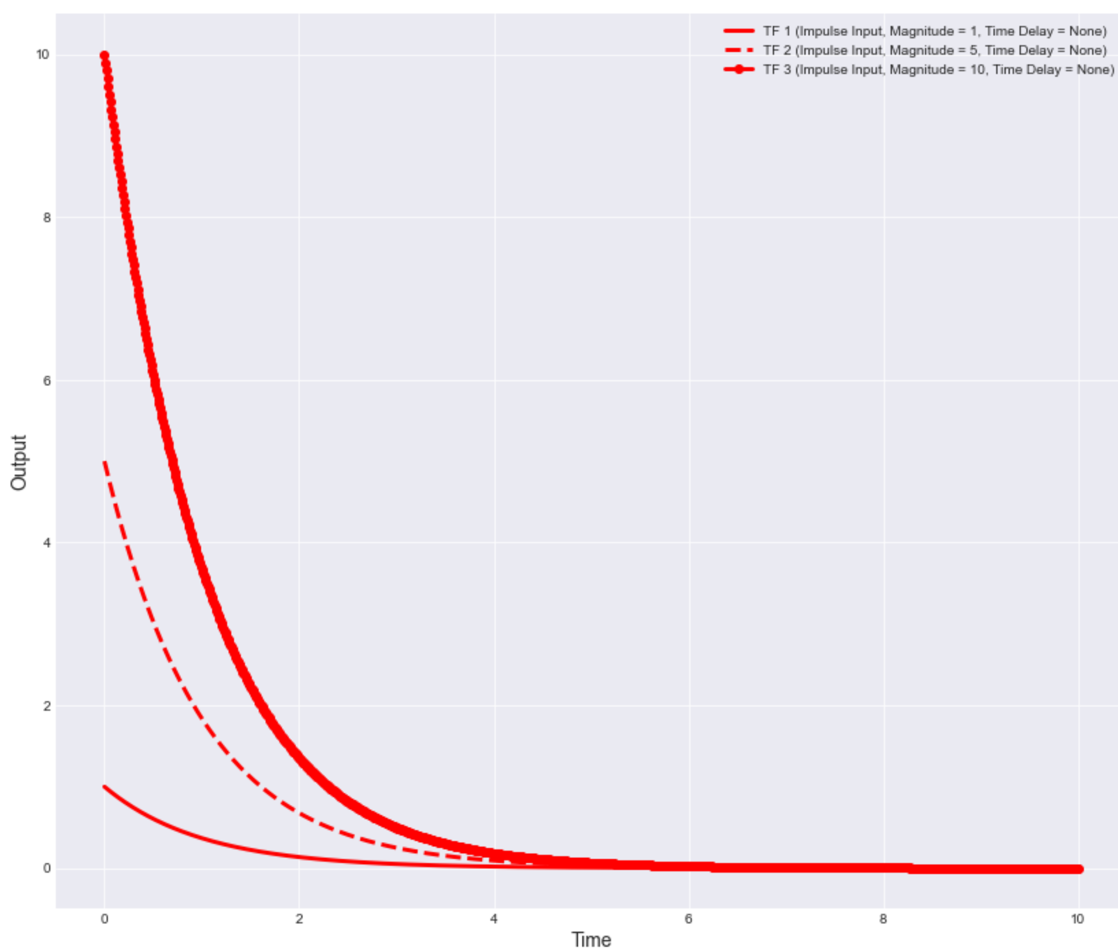
In [8]:
```
sys2 = TransferFunction(1, [1,1])
sys2.InputFunction(Magnitude = 5, Type = 'Impulse')

sys3 = TransferFunction(1, [1,1])
sys3.InputFunction(Magnitude = 10, Type = 'Impulse')
```

```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################

######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################
```

In [9]:

```
CompareResults(sys1, sys2, sys3)
```





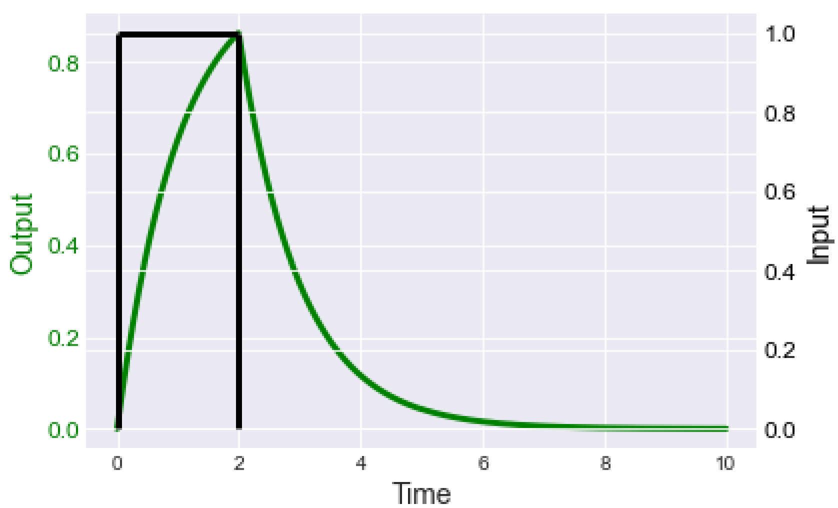
Example 3: $G(s) = \frac{1}{s+1}$ with square function inputs

Square inputs work the same way except you must also define the time the input ends with the argument "InputEndTime."

In [10]:

```
sys1 = TransferFunction(1, [1,1])
sys1.InputFunction(Magnitude = 1, Type = 'Square', InputEndTime = 2)

PlotResponse(sys1)
```

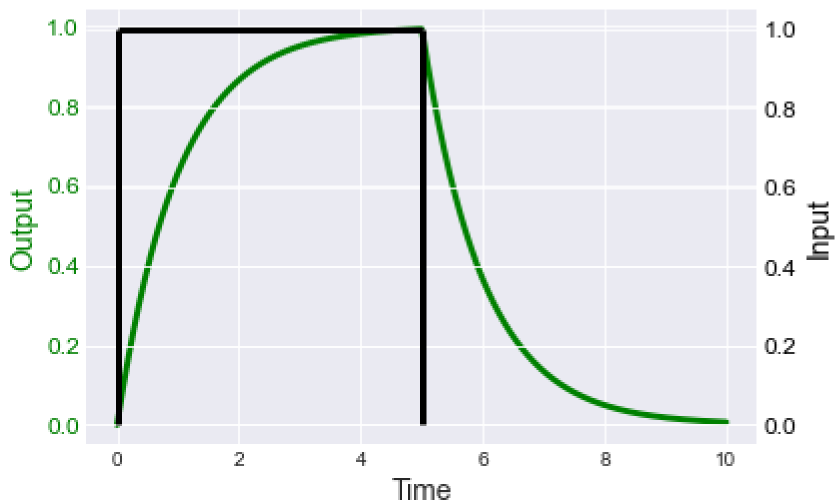
```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################

C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWa
rning: return_x specified for a transfer function system. Internal conversi
on to state space used; results may meaningless.
  warnings.warn(
```



In [11]:

```
sys1.InputFunction(Magnitude = 1, Type = 'Square', InputEndTime = 5)

PlotResponse(sys1)
```

```
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWa
rning: return_x specified for a transfer function system. Internal conversi
on to state space used; results may meaningless.
  warnings.warn(
```



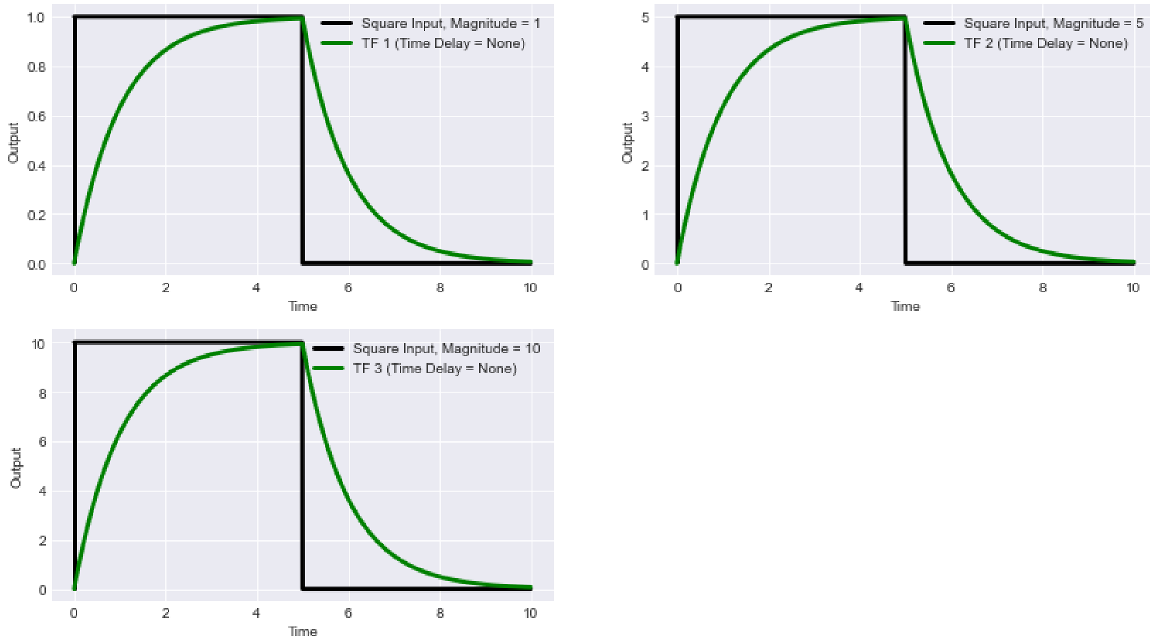
In [12]:

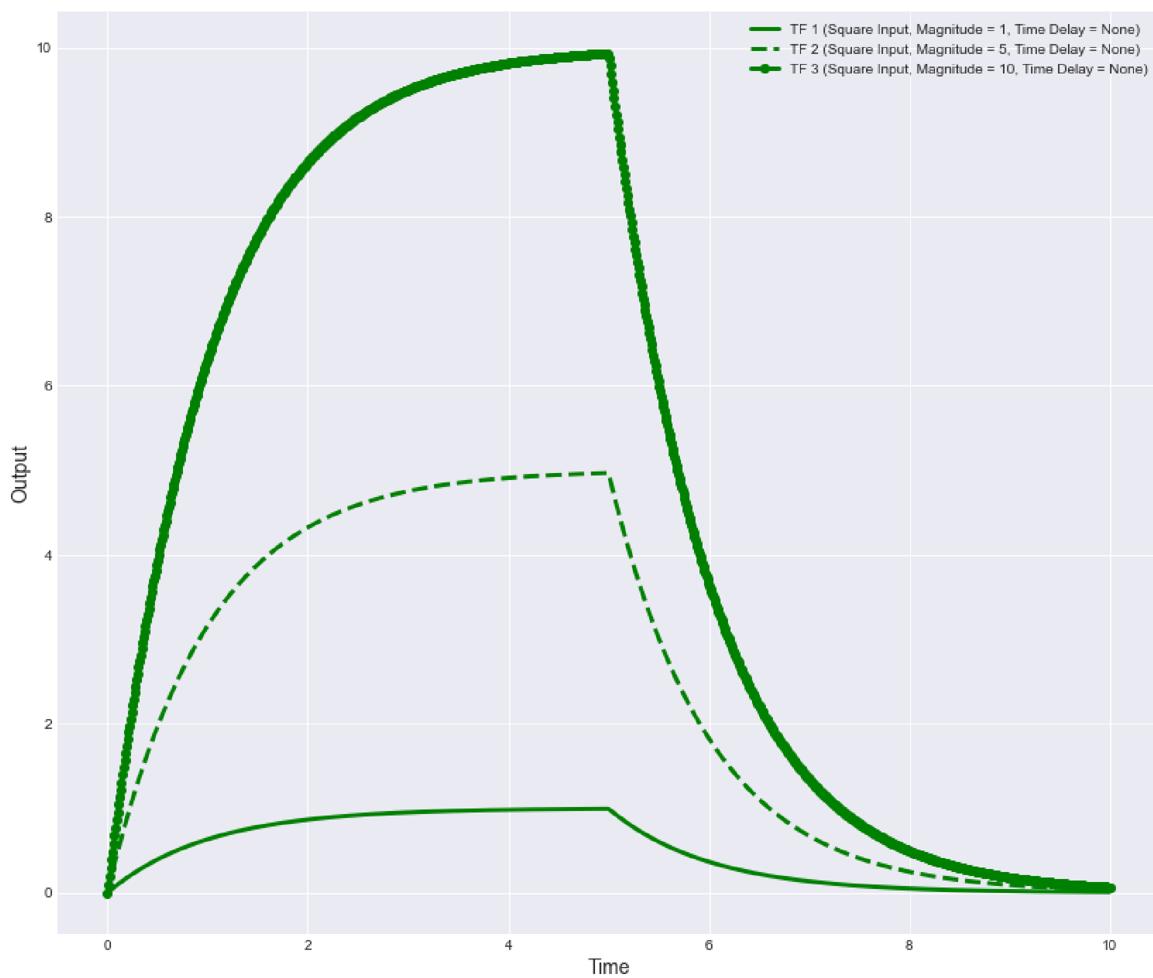
```
sys2 = TransferFunction(1, [1,1])
sys2.InputFunction(Magnitude = 5, Type = 'Square', InputEndTime = 5)

sys3 = TransferFunction(1, [1,1])
sys3.InputFunction(Magnitude = 10, Type = 'Square', InputEndTime = 5)
```

```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################

######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:               [-1.0]
######################################################

C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWa
rning: return_x specified for a transfer function system. Internal conversi
on to state space used; results may meaningless.
  warnings.warn(
```

In [13]:

```
CompareResults(sys1, sys2, sys3)
```



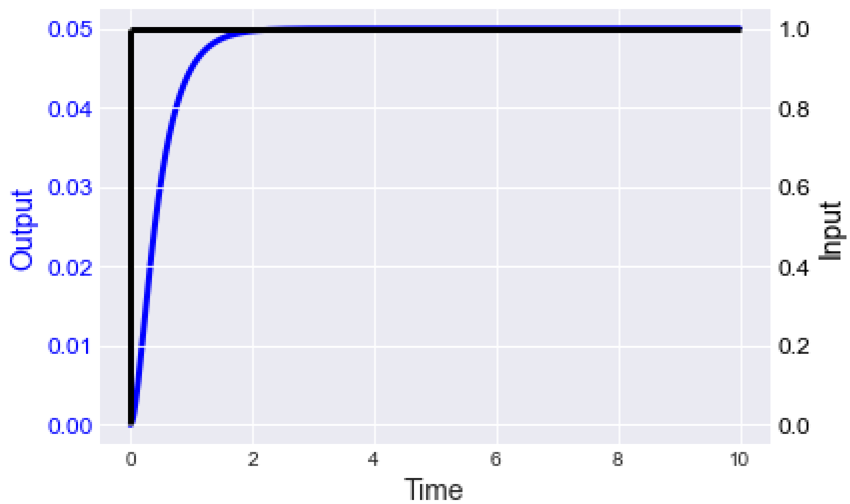


Example 2: $G(s) = \frac{1}{s^2+10s+20}$

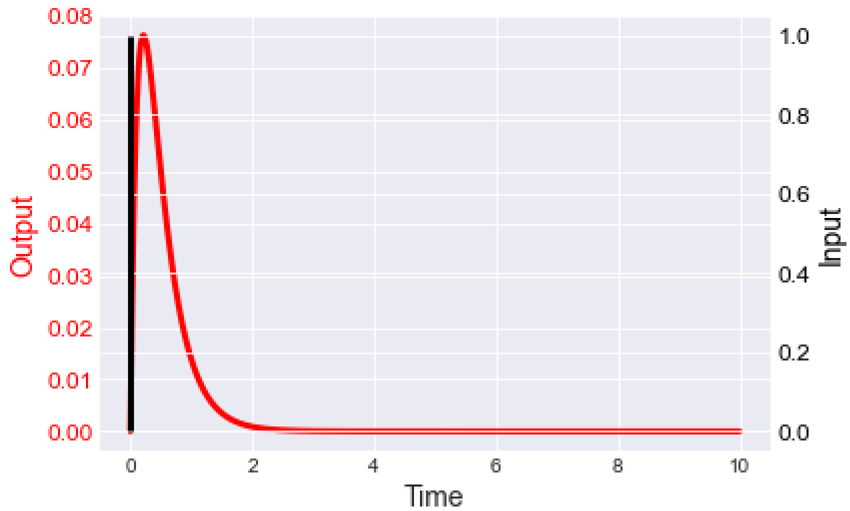
In [14]:
```
sys1 = TransferFunction(1, [1,10,20], Systemlabel=' Sys 1')
```

```
######################################################
#####  Sys 1 Characteristics
######################################################
## Zeros: None
## Damping Coefficient: 1.0 (Critically Damped)
## Poles:               [-7.236, -2.764]
######################################################
```

In [15]:
```
sys1.InputFunction(1, 'Step')
PlotResponse(sys1)
```

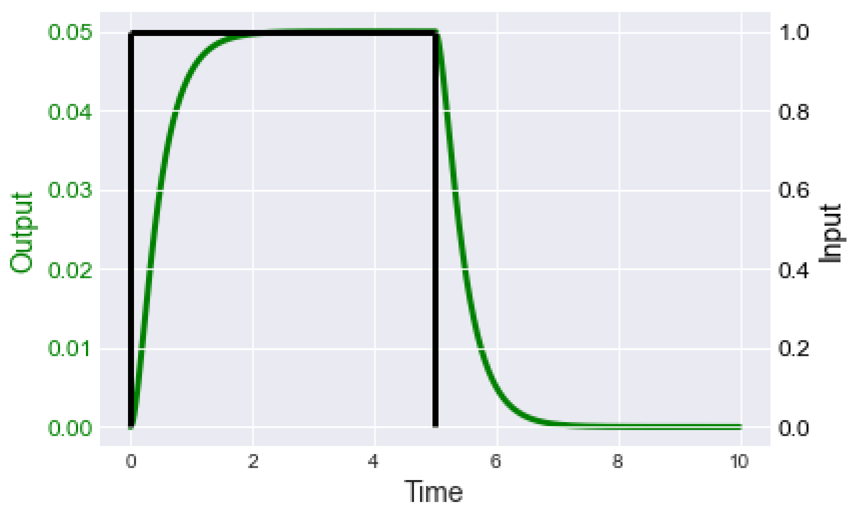


In [16]:
```
sys1.InputFunction(1, 'Impulse')
PlotResponse(sys1)
```



In [17]:
```
sys1.InputFunction(1, 'Square', InputEndTime = 5)
PlotResponse(sys1)
```

```
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWa
rning: return_x specified for a transfer function system. Internal conversi
on to state space used; results may meaningless.
  warnings.warn(
```

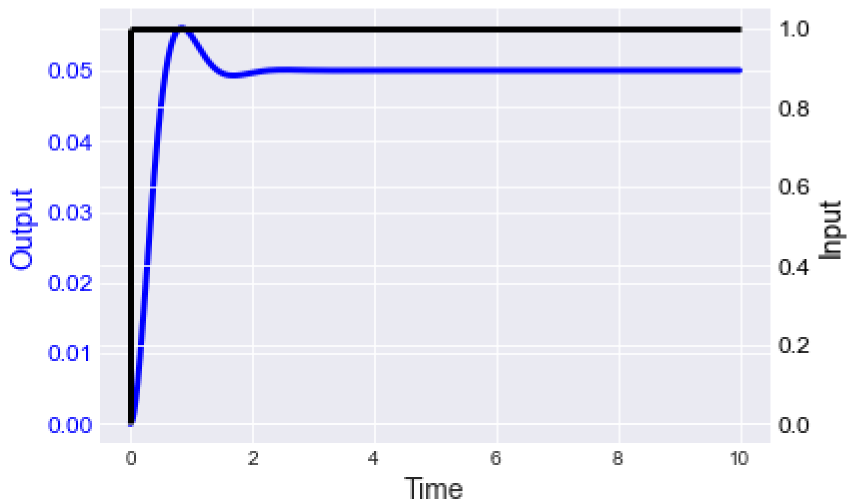


Example 3: $G(s) = \frac{1}{s^2+5s+20}$

In [18]:
```
sys1 = TransferFunction(1, [1,5,20], Systemlabel=' Sys 1')
```

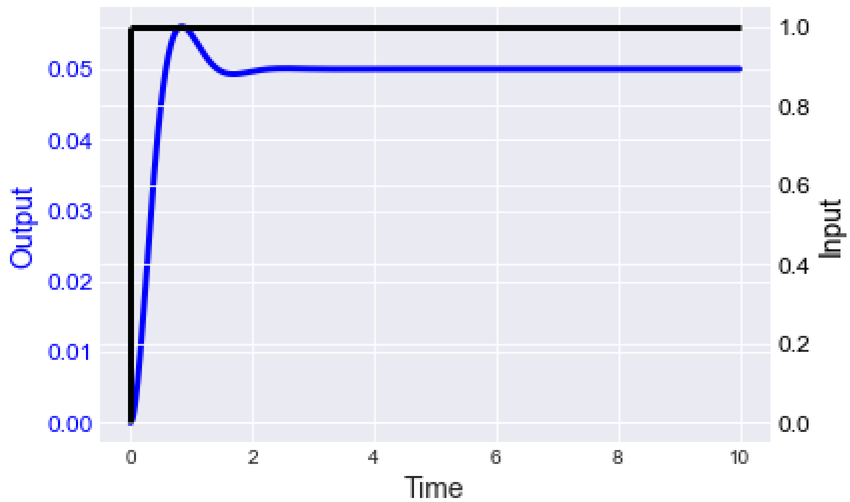
```
######################################################
#####  Sys 1 Characteristics
######################################################
## Zeros: None
## Damping Coefficient: 0.559 (Underdamped)
## Poles:               [(-2.5+3.708j), (-2.5-3.708j)]
######################################################
```

In [19]:
```
sys1.InputFunction(1, 'Step')
PlotResponse(sys1)
```



In [20]:
```
sys2 = TransferFunction(1, [1,5,20], Systemlabel=' Sys 2')

sys2.InputFunction(1, 'Impulse')
PlotResponse(sys1)
```

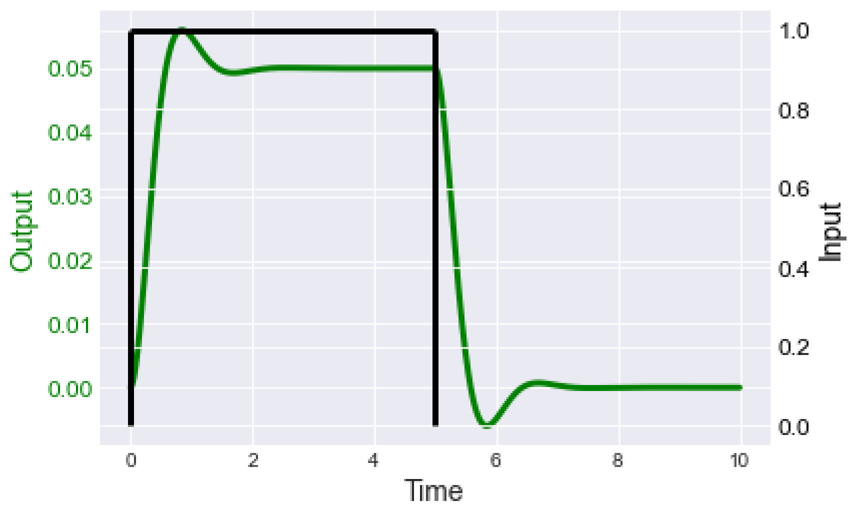
```
######################################################
#####  Sys 2 Characteristics
######################################################
## Zeros: None
## Damping Coefficient: 0.559 (Underdamped)
## Poles:               [(-2.5+3.708j), (-2.5-3.708j)]
######################################################
```



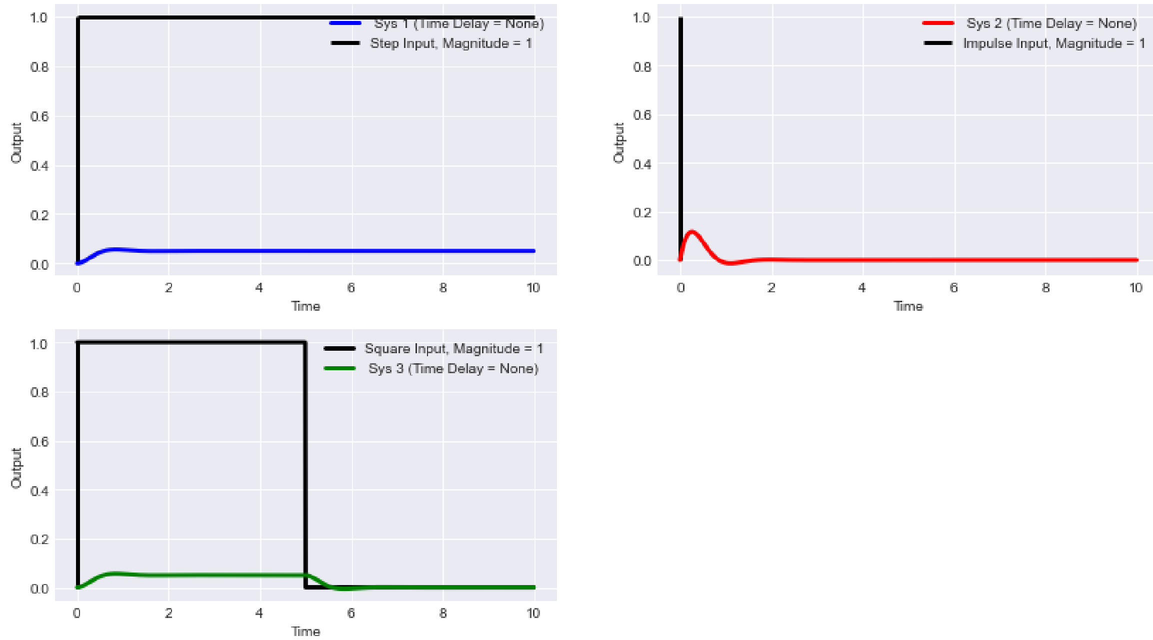
In [21]:

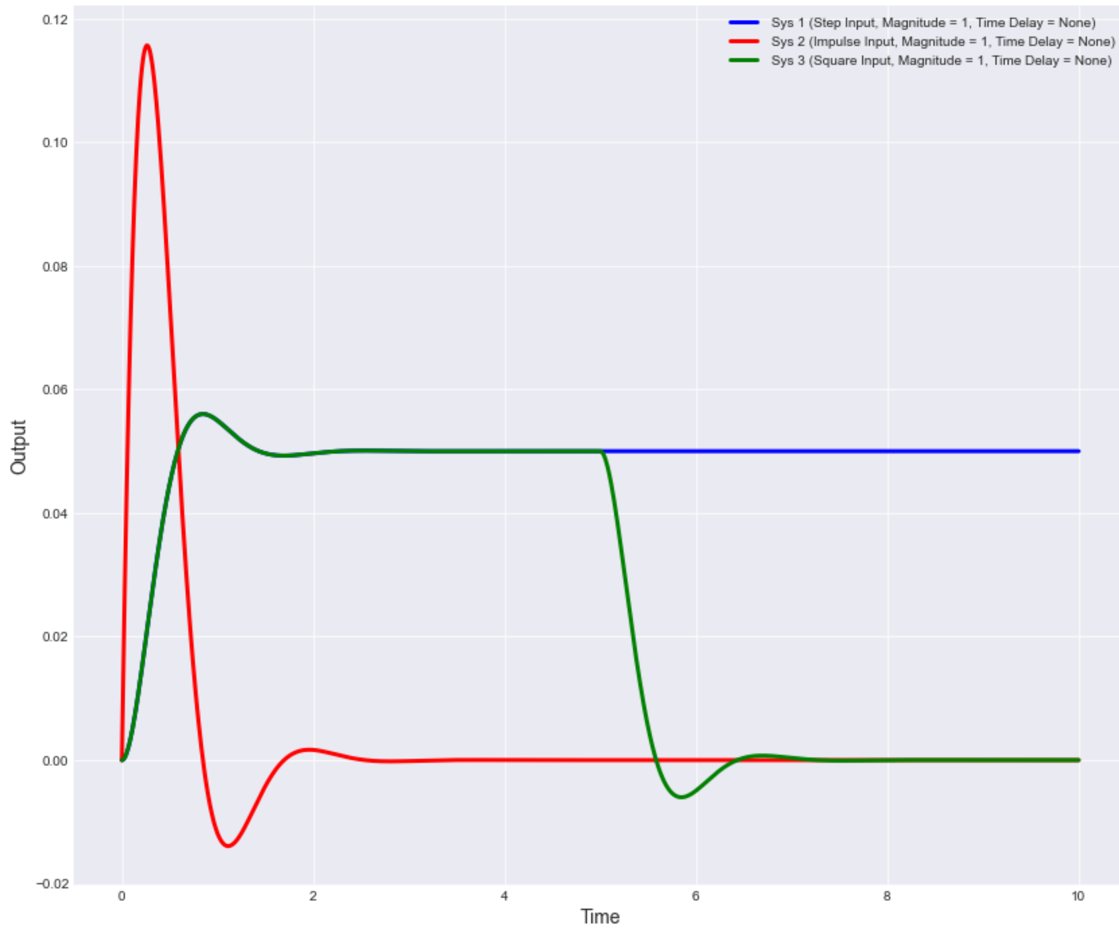
```
sys3 = TransferFunction(1, [1,5,20], Systemlabel=' Sys 3')

sys3.InputFunction(1, 'Square', InputEndTime = 5)
PlotResponse(sys3)
```

```
######################################################
#####  Sys 3 Characteristics
######################################################
## Zeros: None
## Damping Coefficient: 0.559 (Underdamped)
## Poles:                [(-2.5+3.708j), (-2.5-3.708j)]
######################################################

C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWa
rning: return_x specified for a transfer function system. Internal conversi
on to state space used; results may meaningless.
  warnings.warn(
```



In [22]:
```
CompareResults(sys1, sys2, sys3)
```





Example 4: $G(s) = \frac{1-4s}{s^2+10s+20}$

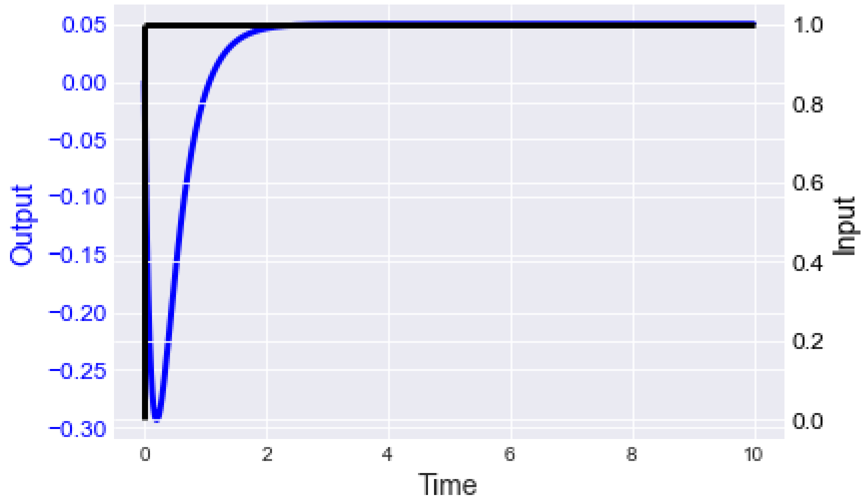
In [23]:
```
sys1 = TransferFunction([-4, 1], [1,10,20], Systemlabel=' Sys 1')
```

```
#######################################################
#####  Sys 1 Characteristics
#######################################################
## Zeros: [0.25]
## Damping Coefficient: 1.0 (Critically Damped)
## Poles:                  [-7.236, -2.764]
#######################################################
```

In [24]:
```
sys1.InputFunction(1, 'Step')
```

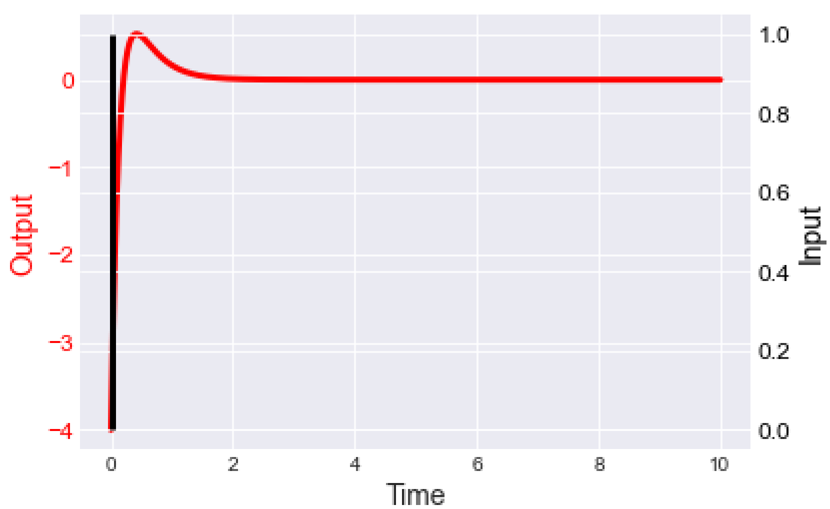
In [25]:

```
PlotResponse(sys1)
```



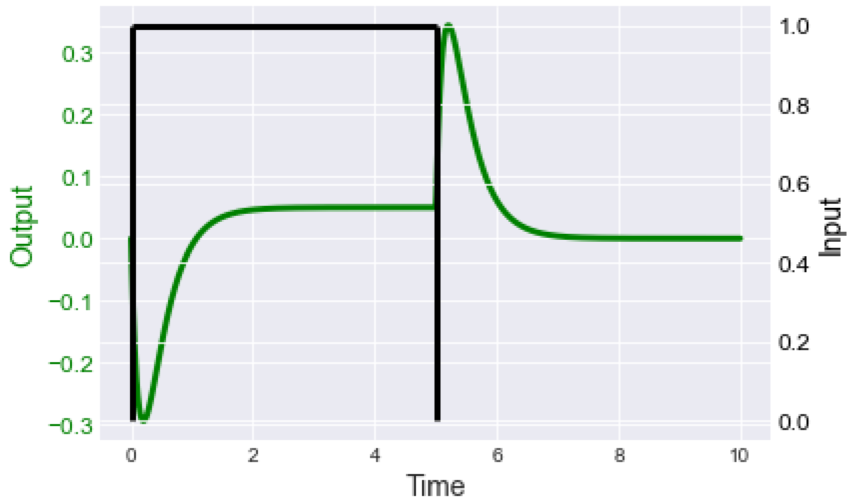
In [26]:

```
sys1.InputFunction(1, 'Impulse')
PlotResponse(sys1)
```



In [27]:
```
sys1.InputFunction(1, 'Square', InputEndTime = 5)
PlotResponse(sys1)
```

```
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWa
rning: return_x specified for a transfer function system. Internal conversi
on to state space used; results may meaningless.
  warnings.warn(
```

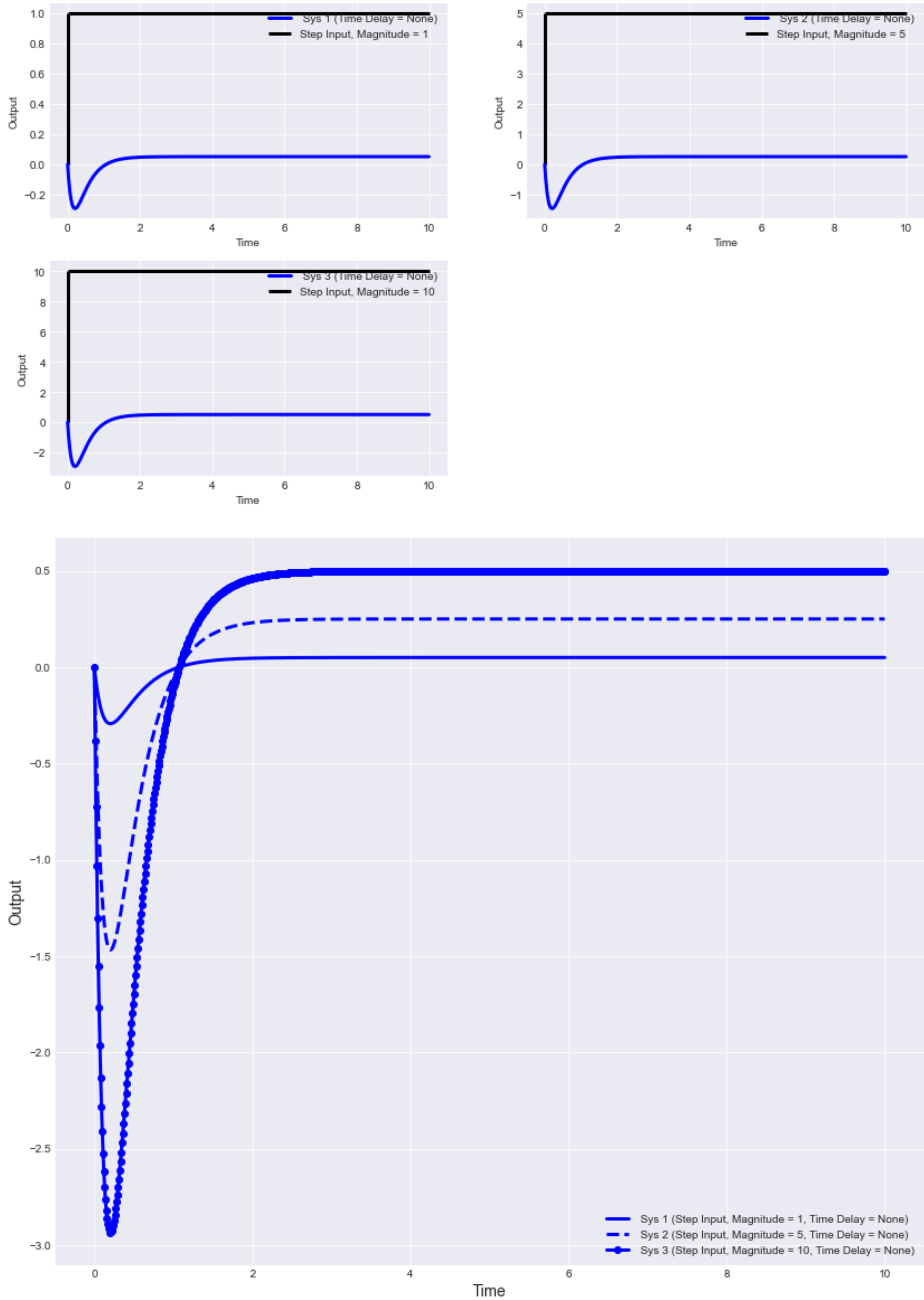


In [28]:
```
sys1 = TransferFunction([-4, 1], [1,10,20], Systemlabel=' Sys 1')
sys2 = TransferFunction([-4, 1], [1,10,20], Systemlabel=' Sys 2')
sys3 = TransferFunction([-4, 1], [1,10,20], Systemlabel=' Sys 3')
```

```
######################################################
#####  Sys 1 Characteristics
######################################################
## Zeros: [0.25]
## Damping Coefficient: 1.0 (Critically Damped)
## Poles:               [-7.236, -2.764]
######################################################

######################################################
#####  Sys 2 Characteristics
######################################################
## Zeros: [0.25]
## Damping Coefficient: 1.0 (Critically Damped)
## Poles:               [-7.236, -2.764]
######################################################

######################################################
#####  Sys 3 Characteristics
######################################################
## Zeros: [0.25]
## Damping Coefficient: 1.0 (Critically Damped)
## Poles:               [-7.236, -2.764]
######################################################
```

In [29]:
```
sys1.InputFunction(1, 'Step')
sys2.InputFunction(5, 'Step')
sys3.InputFunction(10, 'Step')
```

In [30]:
```
CompareResults(sys1, sys2, sys3)
```



Example 5: $G(s) = \frac{e^{-2s}}{s+5}$

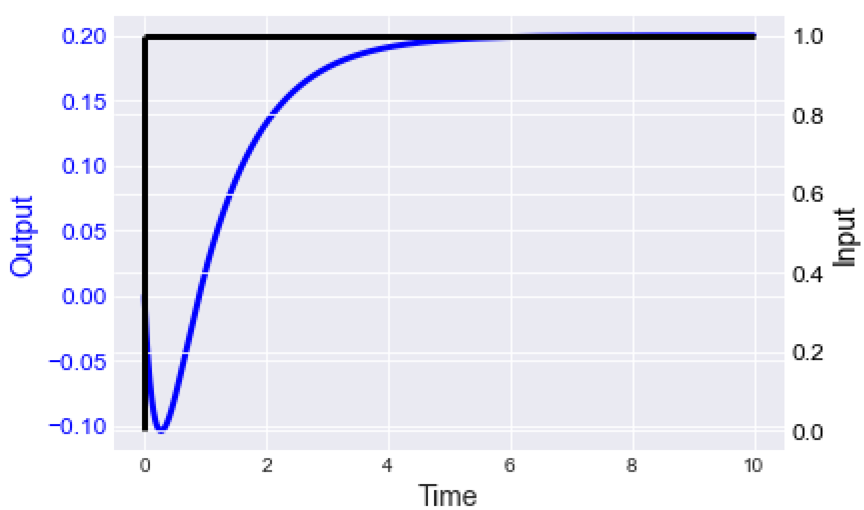
Time delays can be added with the argument "TimeDelay." Time delays are computed using the Pade approximation which will be covered later in the course.

In [31]:
```
sys1 = TransferFunction(1, [1,5], TimeDelay = 2)
```

```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: [1.]
## Poles:               [-5.0, -1.0]
######################################################
```

In [32]:
```
sys1.InputFunction(1, 'Step')
```

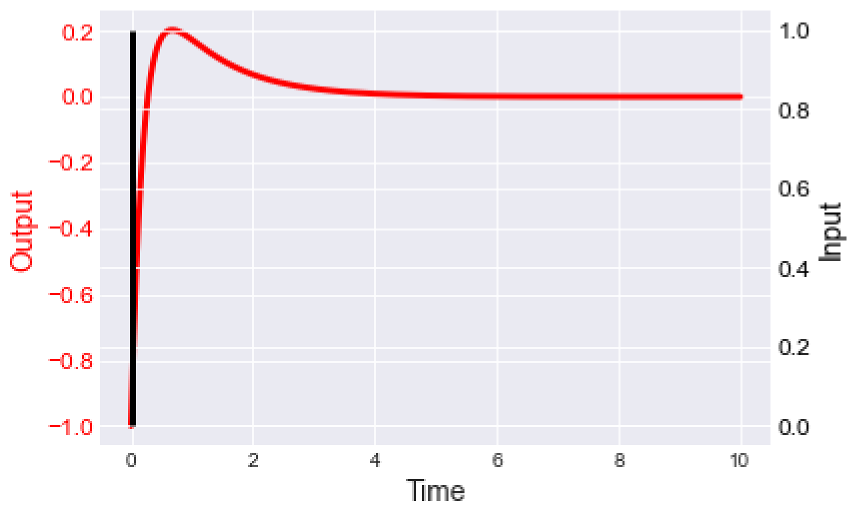
In [33]:
```
PlotResponse(sys1)
```



In [34]:

```
sys2 = TransferFunction(1, [1,5], TimeDelay = 2)
sys2.InputFunction(1, 'Impulse')
PlotResponse(sys2)
```

```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: [1.]
## Poles:                [-5.0, -1.0]
######################################################
```

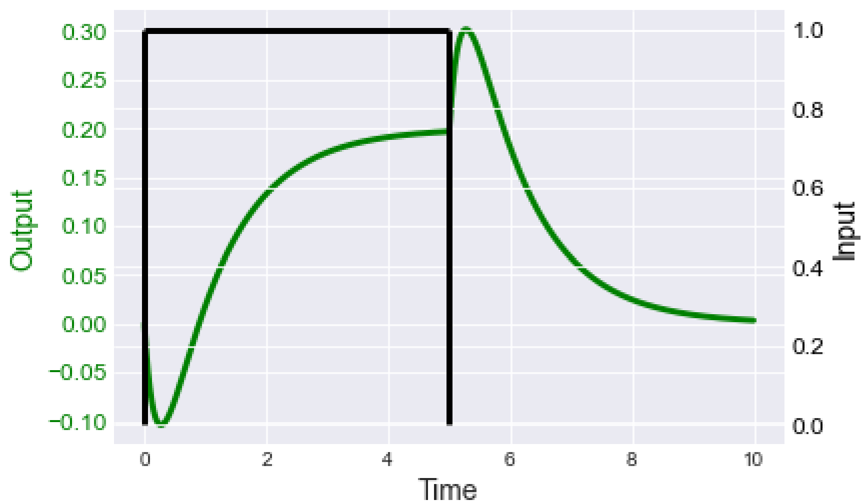


In [35]:

```
sys3 = TransferFunction(1, [1,5], TimeDelay = 2)
sys3.InputFunction(1, 'Square', InputEndTime = 5)
PlotResponse(sys3)
```

```
#####################################################
##### Transfer Function Characteristics
#####################################################
## Zeros: [1.]
## Poles:               [-5.0, -1.0]
#####################################################
```

```
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWa
rning: return_x specified for a transfer function system. Internal conversi
on to state space used; results may meaningless.
  warnings.warn(
```



In [36]:
```python
CompareResults(sys1, sys2, sys3)
```