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/). 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 ControlCode import * # this will allow you to use the code in your current file
```

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 = 'System 1')
```

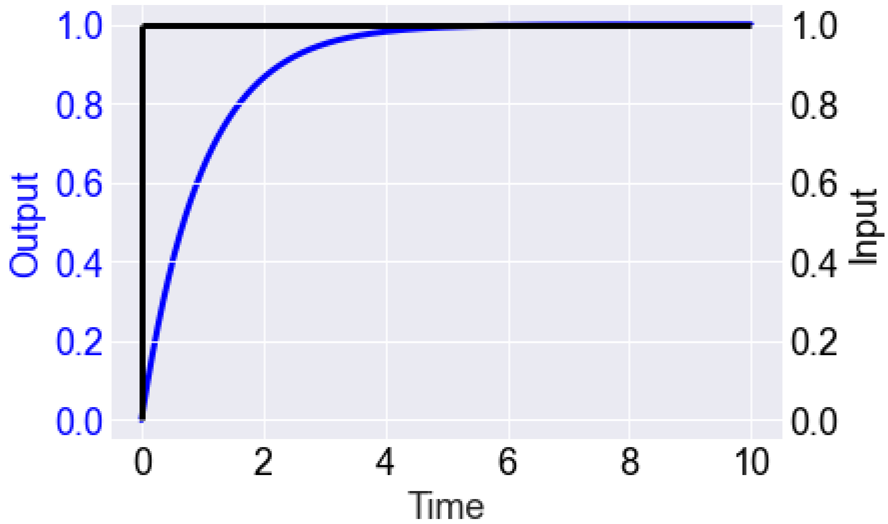
```
########################################################
##### System 1 Characteristics
########################################################
## Zeros: None
## Poles:                [-1.0]
########################################################
```

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')
```

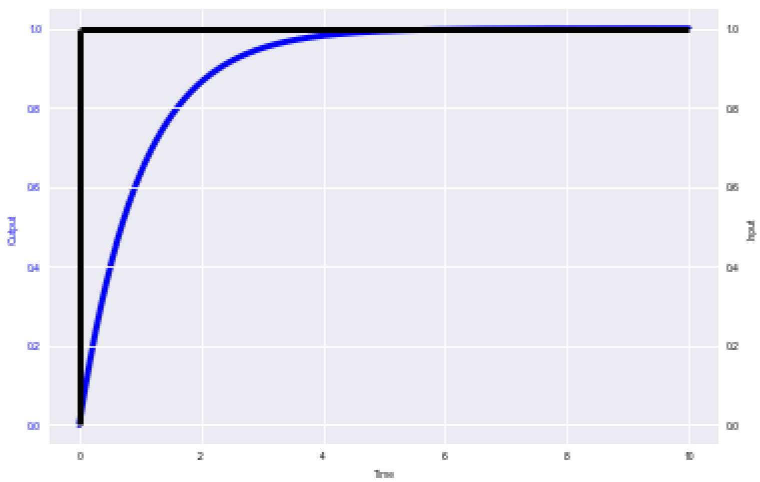
Once you have defined a transfer function and an input you can plot the output of the system. The output will be displayed on the left vertical axis in color and the input will be displayed on the right vertical axis in black.

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

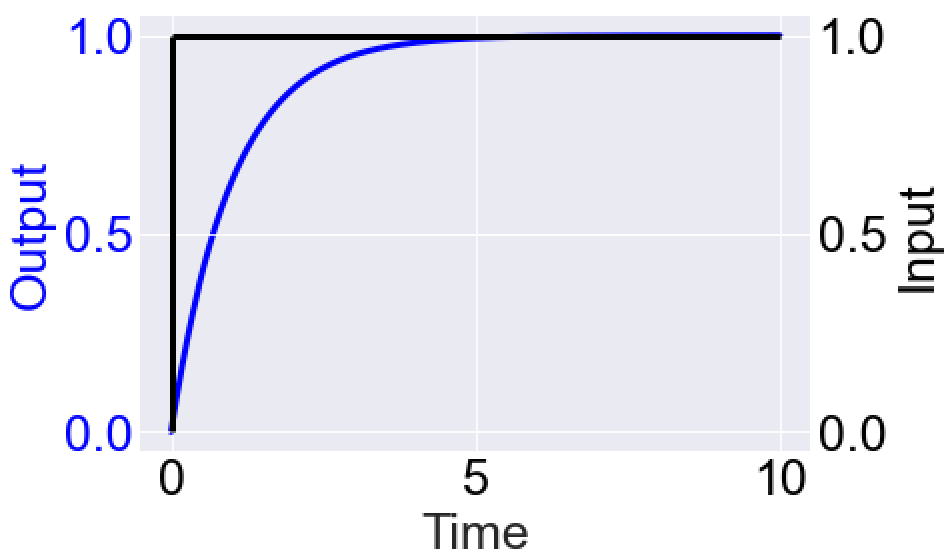


You can change the size of the axis labels and tick marks too.

In [5]:
```
PlotResponse(sys1, FontSize = 5)
```



In [6]:
```
PlotResponse(sys1, FontSize = 25)
```

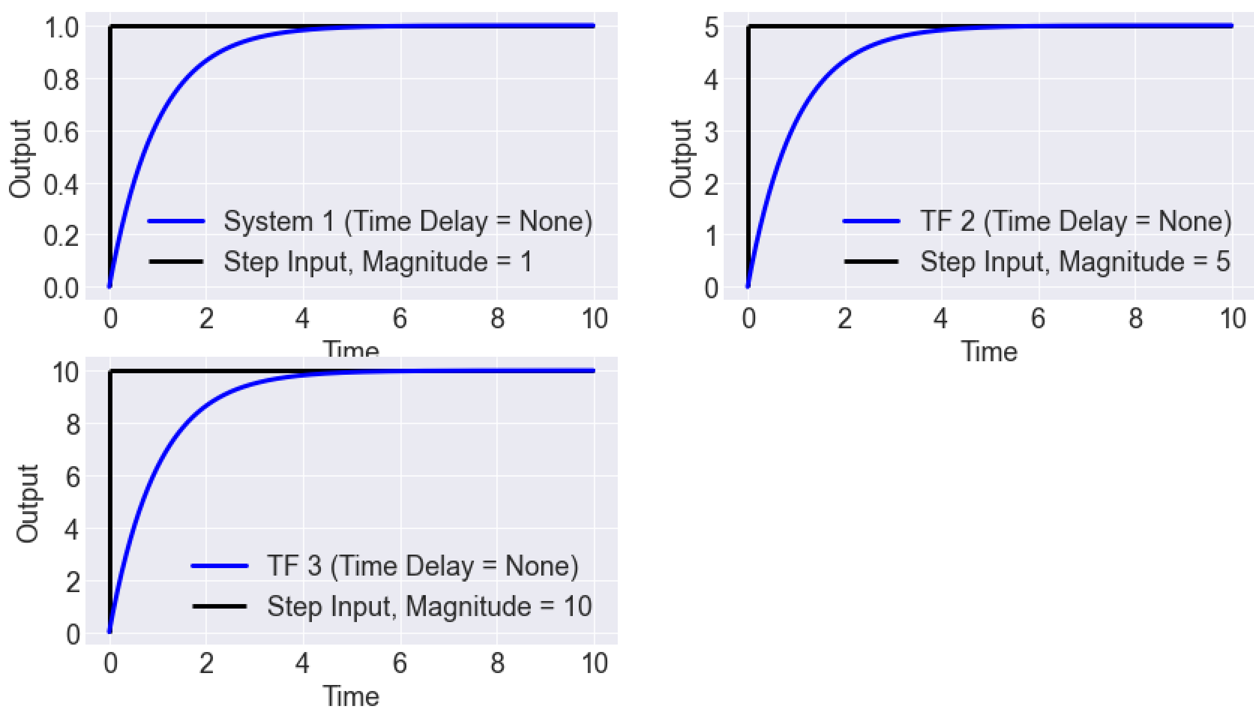


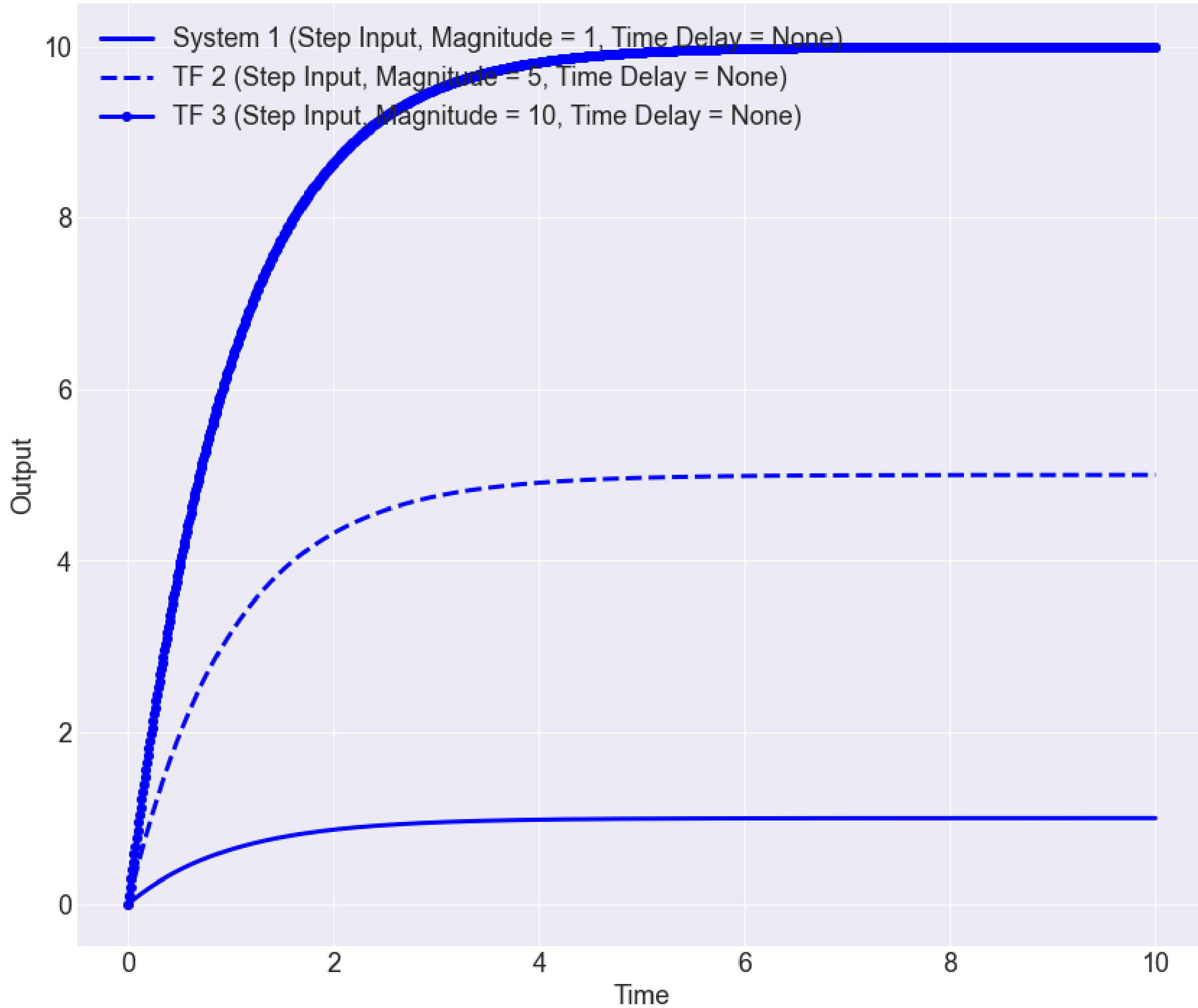
Multiple transfer functions may be defined and compared in one figure. You also can define the numerator and denominator directly without using the commands "numerator" and "denominator" as long as they are in the correct order.

In [7]:
```
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]
########################################################
```

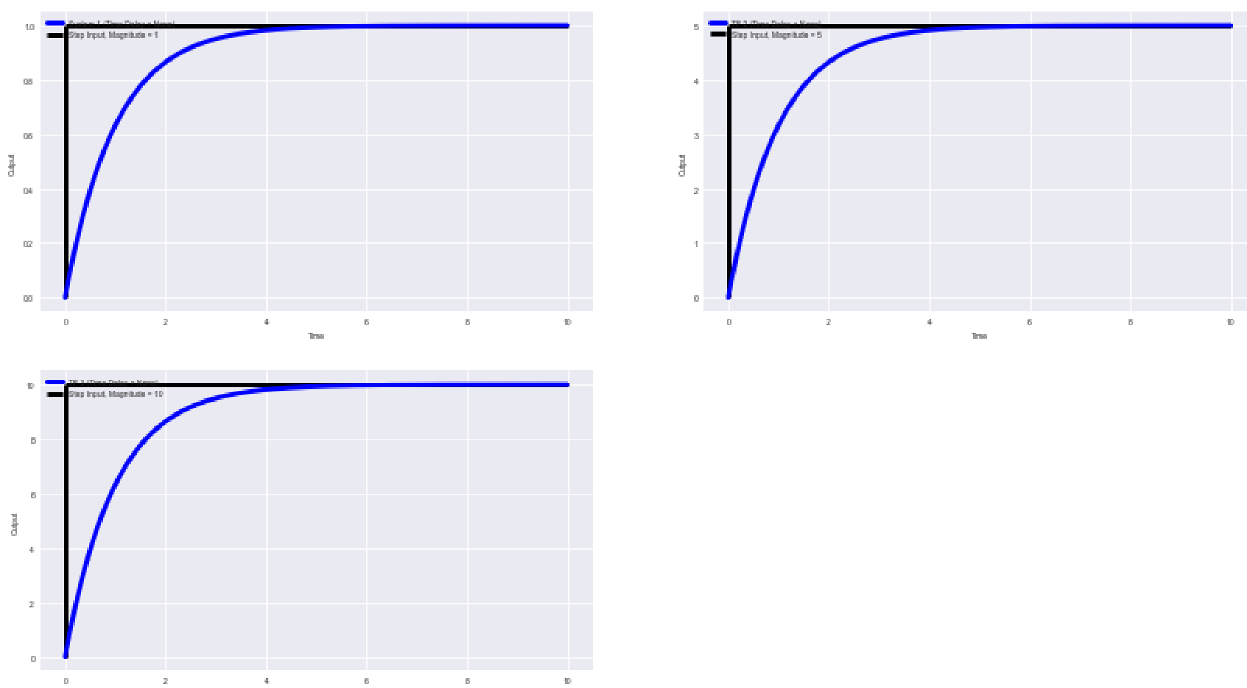
In [8]:
```
CompareResults(sys1, sys2, sys3)
```

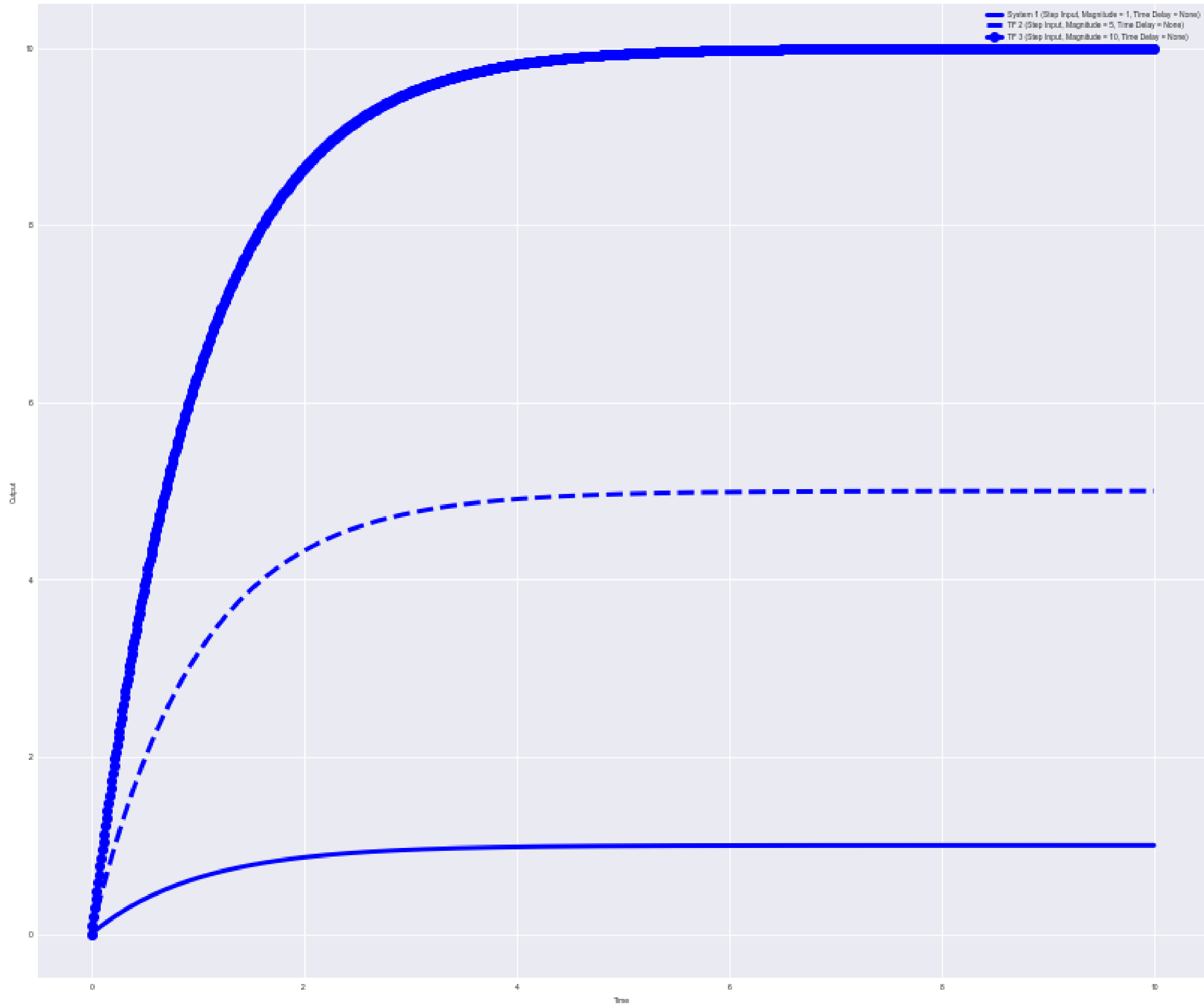




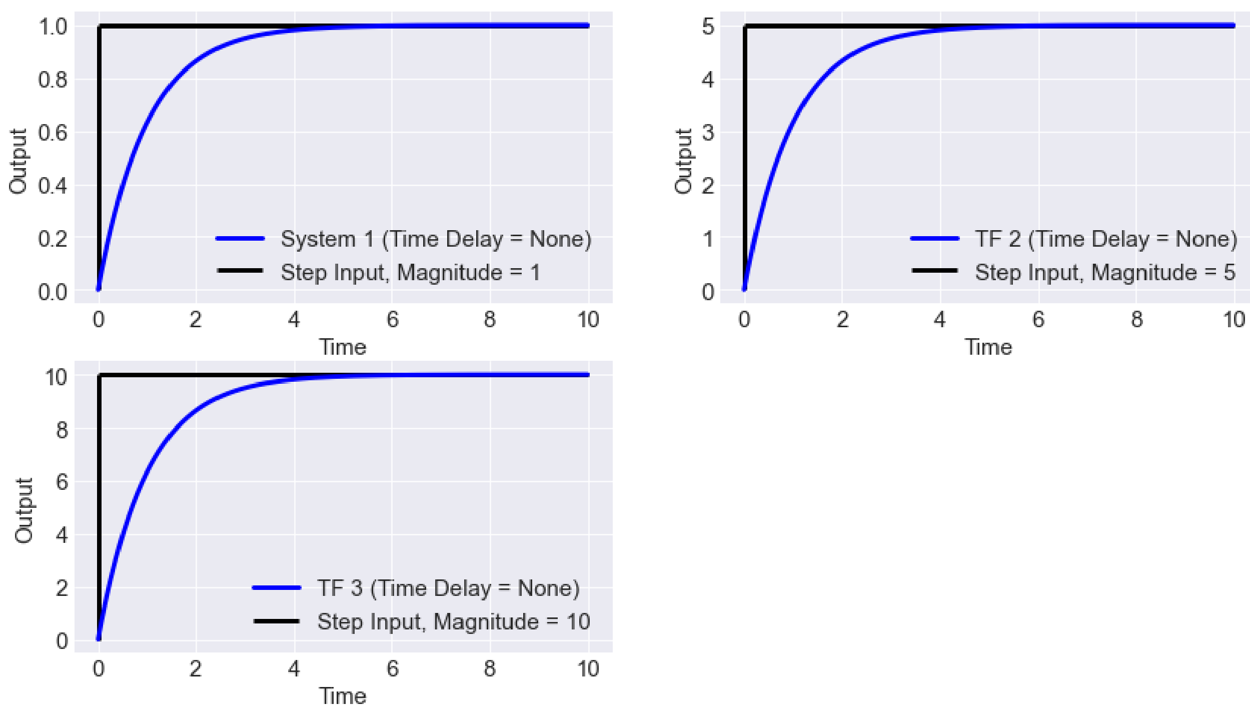
You can change the size of the axis labels and tick marks too.

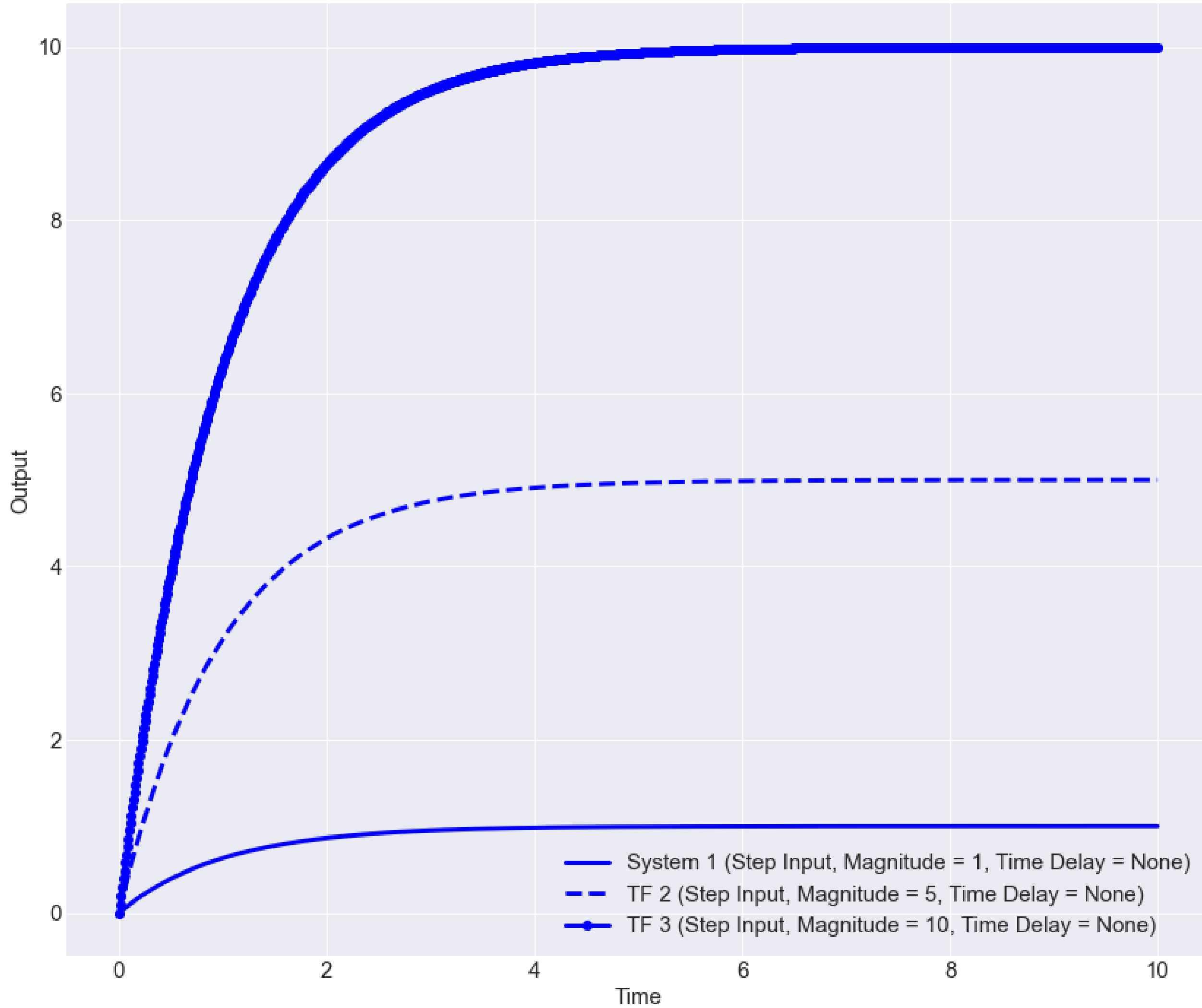
In [9]:
```
CompareResults(sys1, sys2, sys3, FontSize = 5)
```





In [10]:
```
CompareResults(sys1, sys2, sys3, FontSize = 15)
```



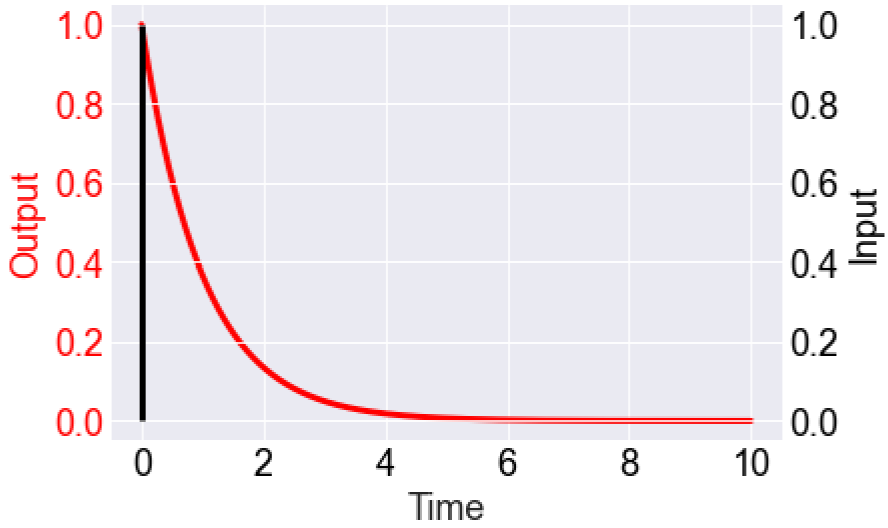


Example 2: $G(s) = \frac{1}{s+1}$ with impulse function inputs

In [11]:

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

PlotResponse(sys1)
```

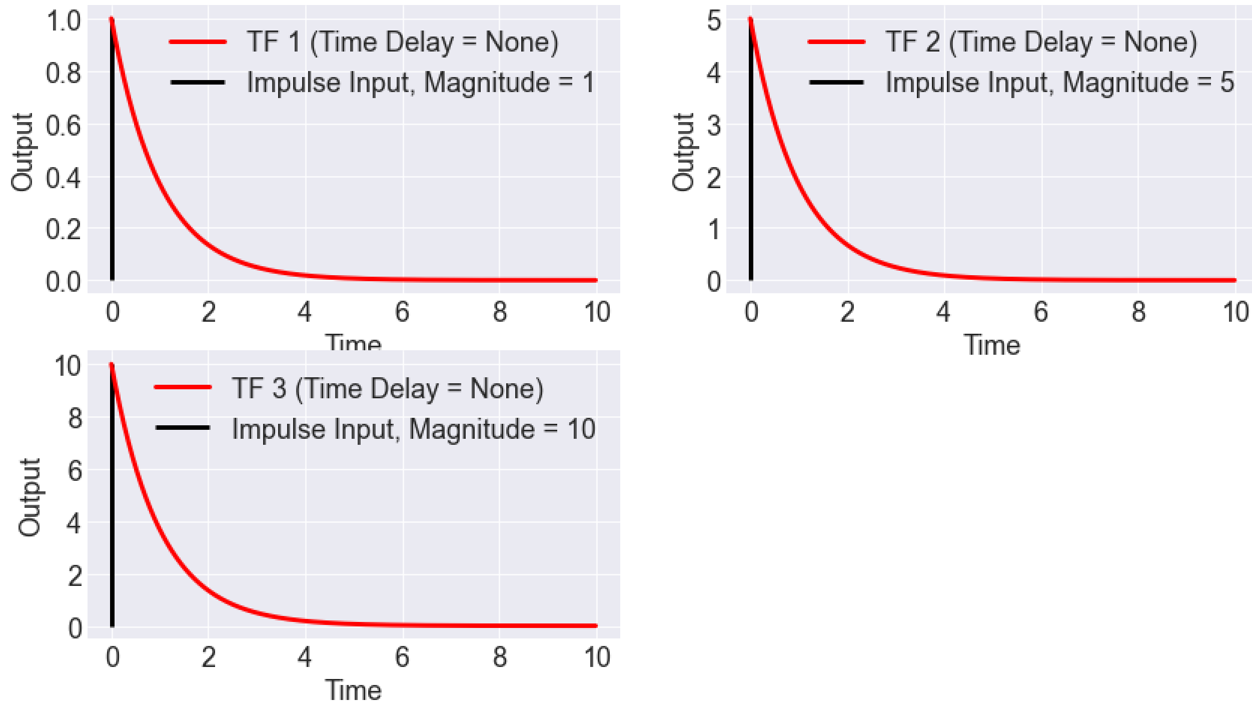
```
#######################################################
##### Transfer Function Characteristics
#######################################################
## Zeros: None
## Poles:               [-1.0]
#######################################################
```

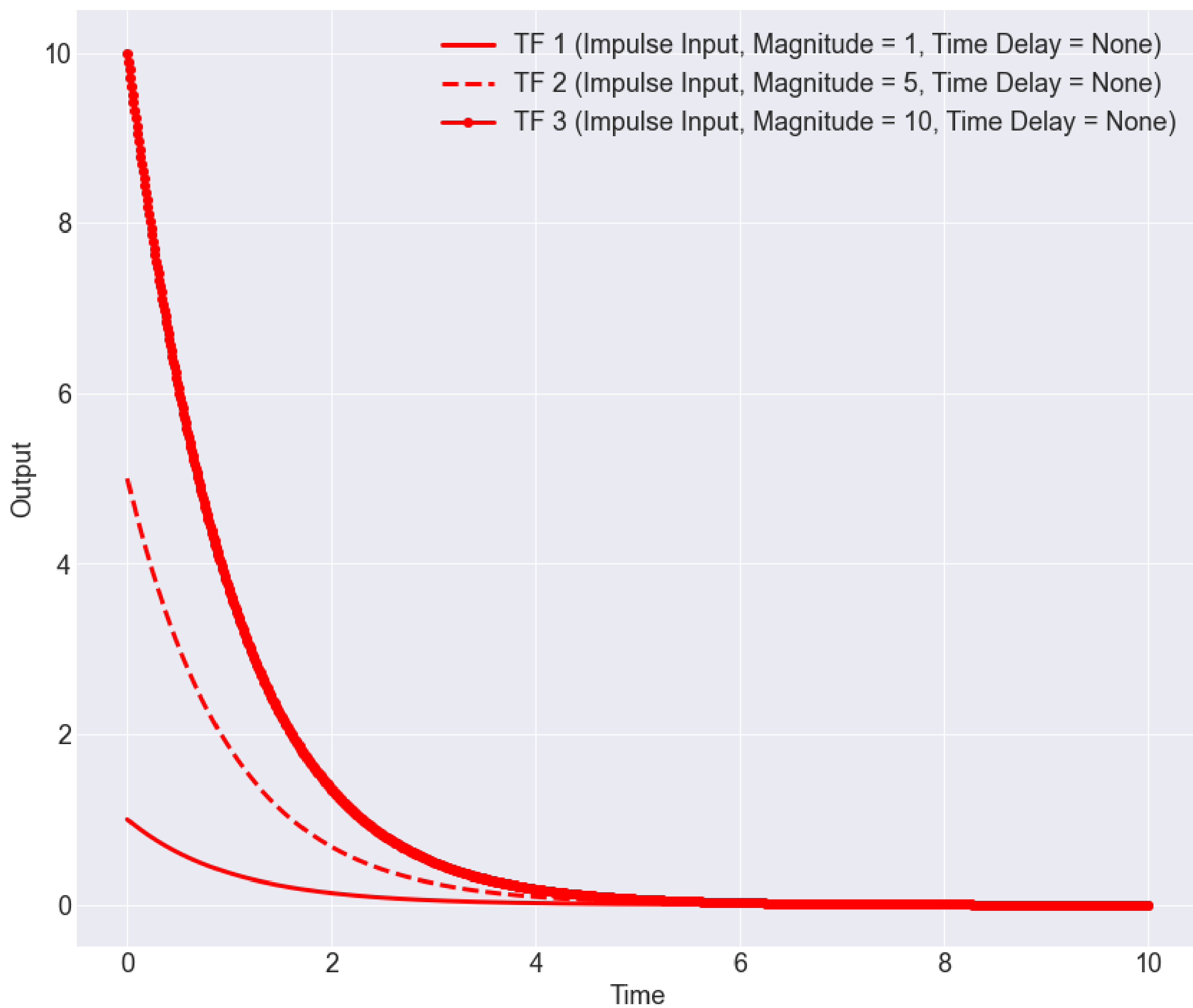


In [12]:
```
sys2 = TransferFunction(1, [1,1])
sys2.InputFunction(Magnitude = 5, Type = 'Impulse')

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 [13]:
```
CompareResults(sys1, sys2, sys3)
```



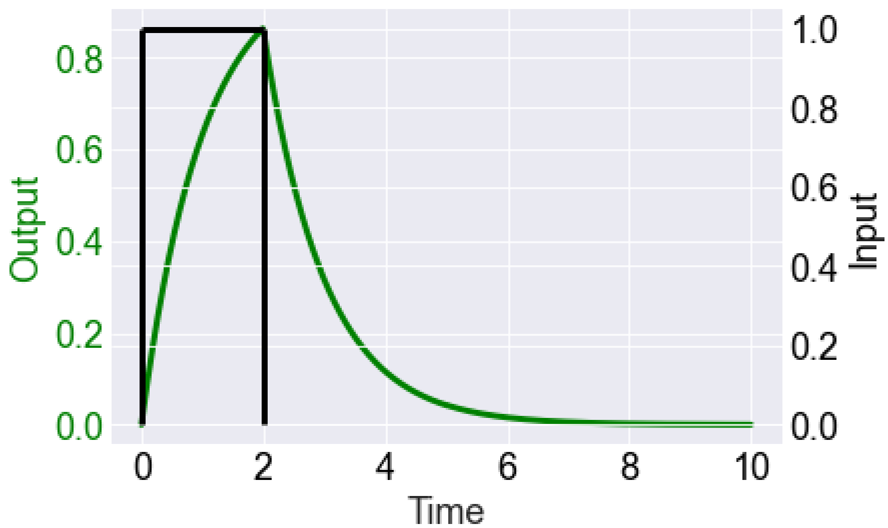


Example 3: $G(s) = \frac{1}{s+1}$ with square function inputs

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

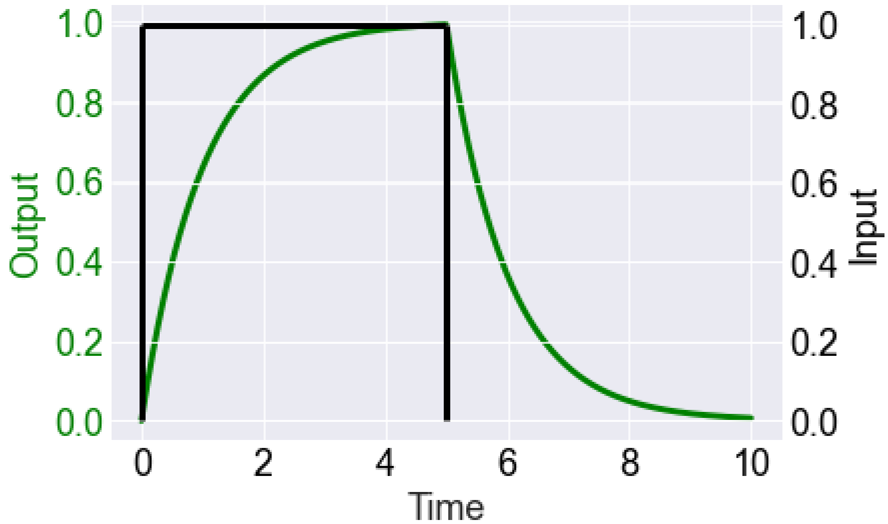
In [14]:
```
sys1 = TransferFunction(1, [1,1])
sys1.InputFunction(Magnitude = 1, Type = 'Square', InputEndTime = 2)

PlotResponse(sys1)
```

```
########################################################
##### Transfer Function Characteristics
########################################################
## Zeros: None
## Poles:               [-1.0]
########################################################
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWarning: return
_x specified for a transfer function system. Internal conversion to state space used; re
sults may meaningless.
  warnings.warn(
```



In [15]:
```
sys1.InputFunction(Magnitude = 1, Type = 'Square', InputEndTime = 5)

PlotResponse(sys1)
```

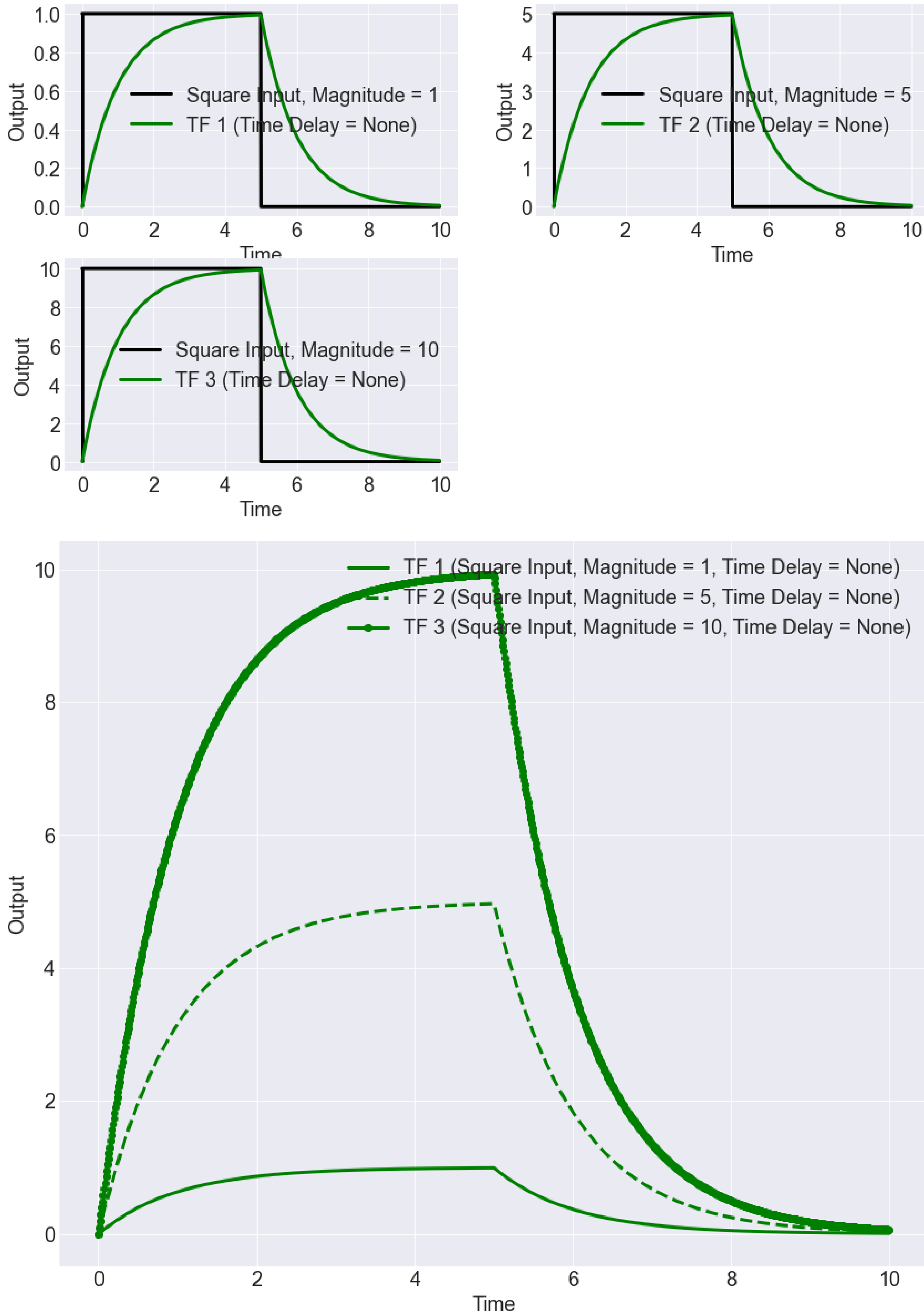
```
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWarning: return
_x specified for a transfer function system. Internal conversion to state space used; re
sults may meaningless.
  warnings.warn(
```



In [16]:
```
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: UserWarning: return
_x specified for a transfer function system. Internal conversion to state space used; re
sults may meaningless.
  warnings.warn(
```

In [17]:
```
CompareResults(sys1, sys2, sys3)
```

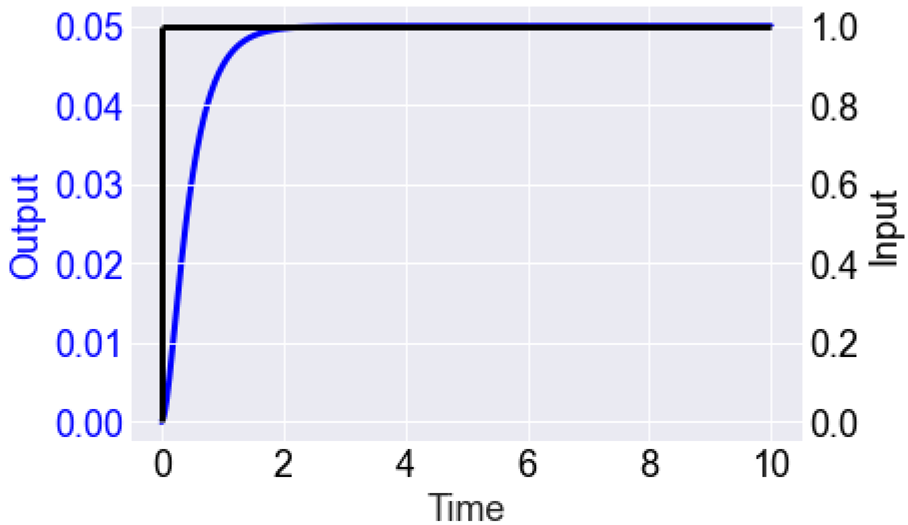


Example 2: $G(s) = \frac{1}{s^2+10s+20}$

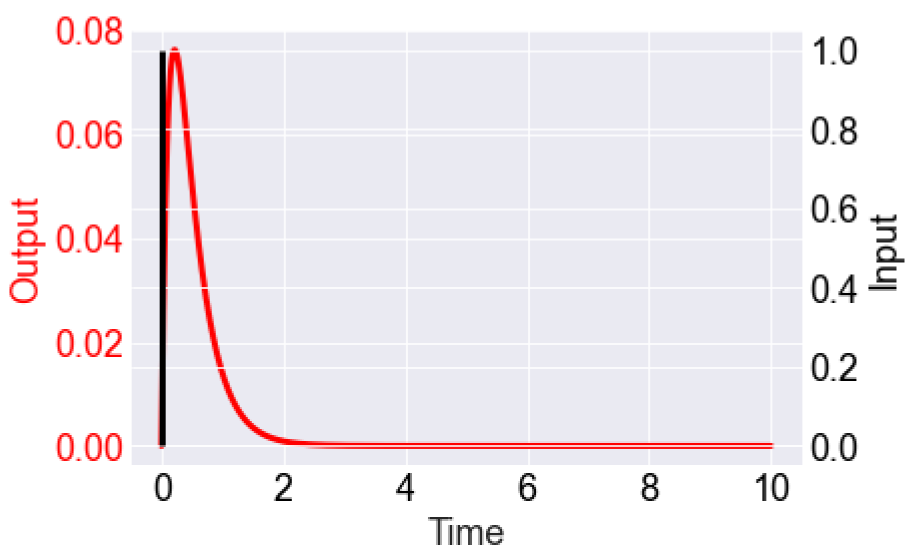
In [18]:
```
sys1 = TransferFunction(1, [1,10,20], Systemlabel=' Sys 1')
```

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

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

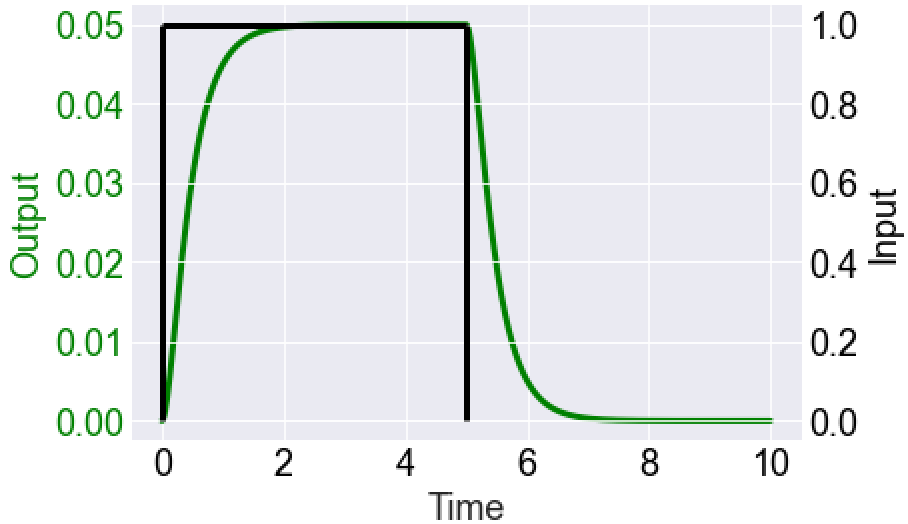


In [20]:
```
sys1.InputFunction(1, 'Impulse')
PlotResponse(sys1)
```



In [21]:
```
sys1.InputFunction(1, 'Square', InputEndTime = 5)
PlotResponse(sys1)
```

```
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWarning: return
_x specified for a transfer function system. Internal conversion to state space used; re
sults may meaningless.
  warnings.warn(
```

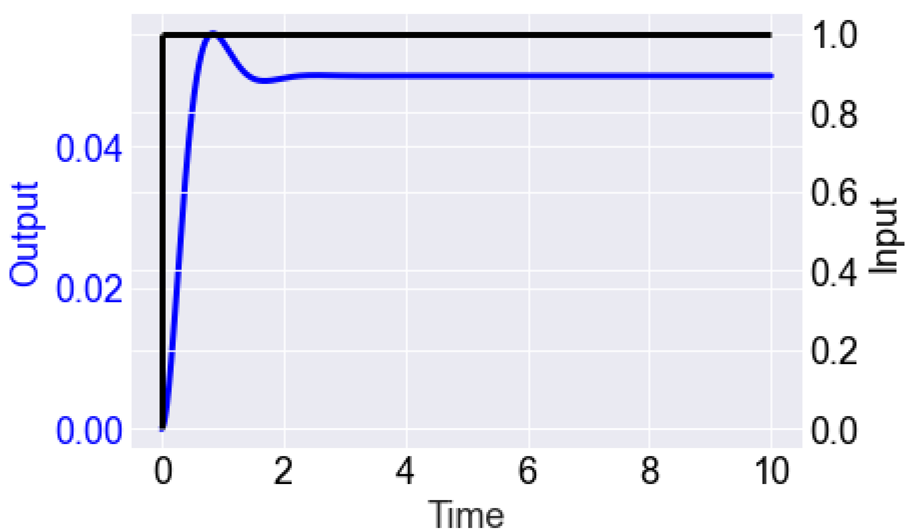


Example 3: $G(s) = \frac{1}{s^2+5s+20}$

In [22]:
```
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)]
##########################################################
```

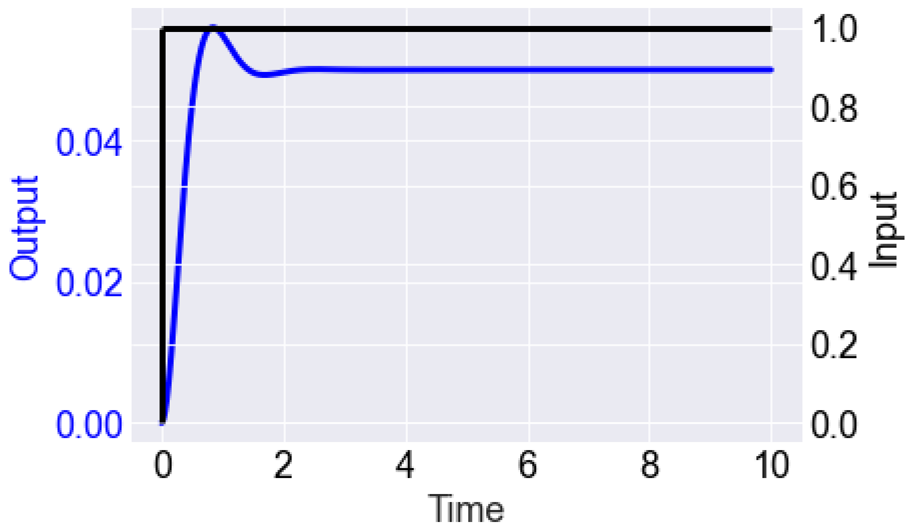
In [23]:
```
sys1.InputFunction(1, 'Step')
PlotResponse(sys1)
```



In [24]:
```
sys2 = TransferFunction(1, [1,5,20], Systemlabel=' Sys 2')

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

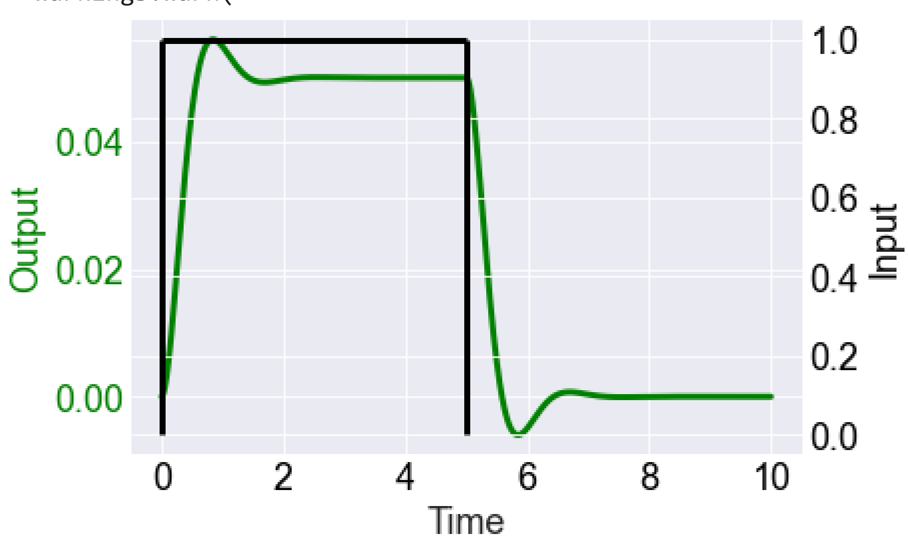
```
##########################################################
#####  Sys 2 Characteristics
##########################################################
```

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

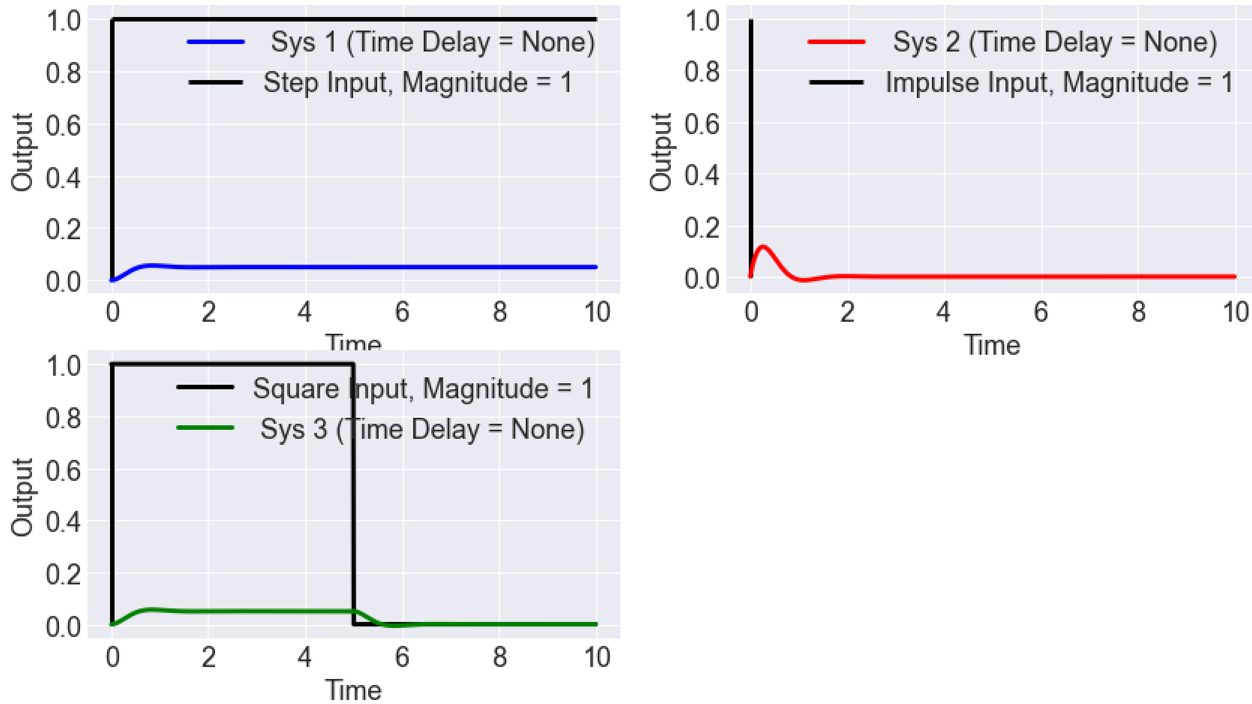


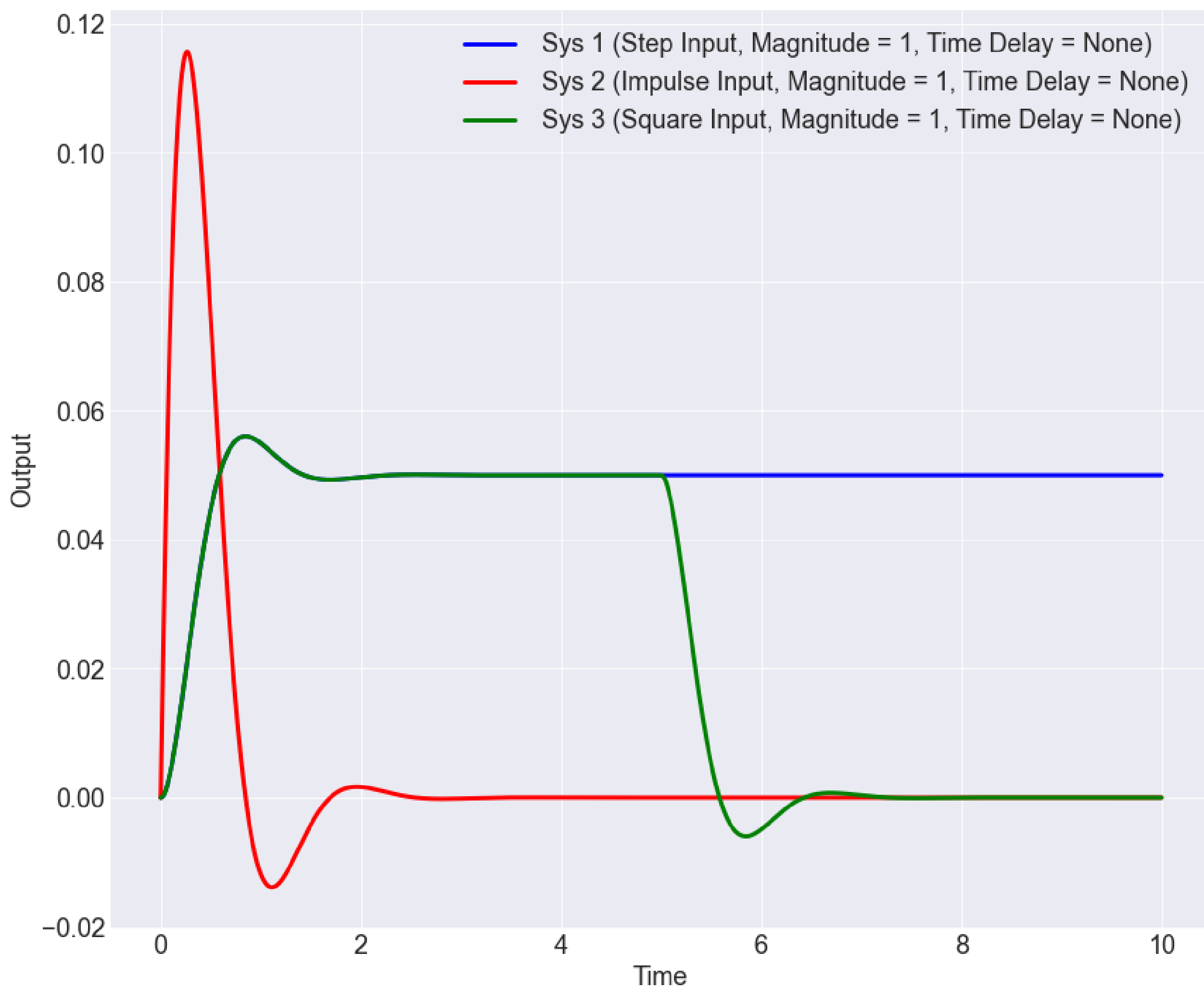
In [25]:
```
sys3 = TransferFunction(1, [1,5,20], Systemlabel=' Sys 3')

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: UserWarning: return
_x specified for a transfer function system. Internal conversion to state space used; re
sults may meaningless.
  warnings.warn(
```



In [26]:
```
CompareResults(sys1, sys2, sys3)
```





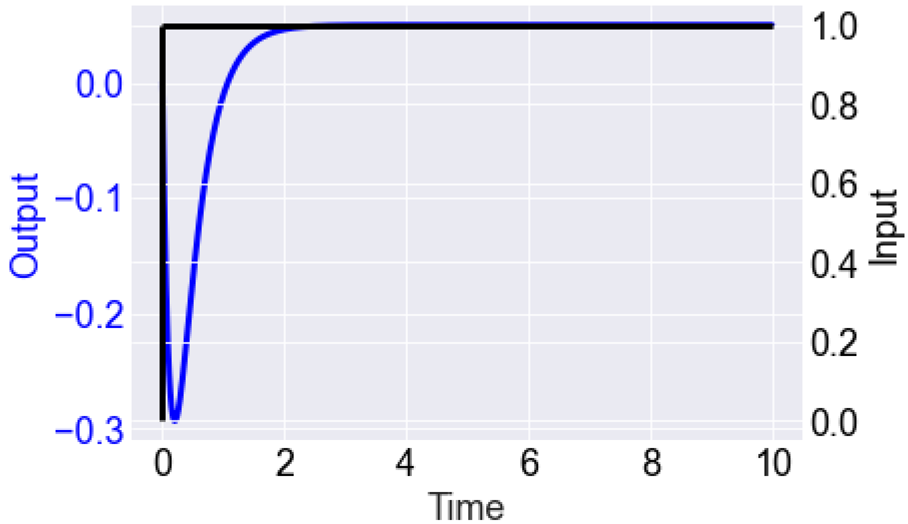
Example 4: $G(s) = \frac{1-4s}{s^2+10s+20}$

In [27]:
```
sys1 = TransferFunction([-4, 1], [1,10,20], Systemlabel=' Sys 1')
```

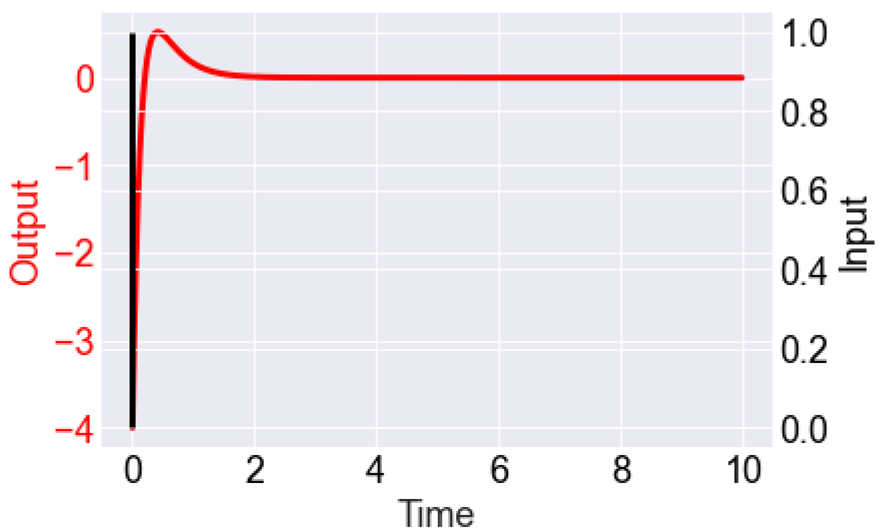
```
######################################################
#####  Sys 1 Characteristics
######################################################
## Zeros: [0.25]
## Damping Coefficient: 1.0 (Critically Damped)
## Poles:               [-7.236, -2.764]
######################################################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

In [28]:
```
sys1.InputFunction(1, 'Step')
```

In [29]:
```
PlotResponse(sys1)
```

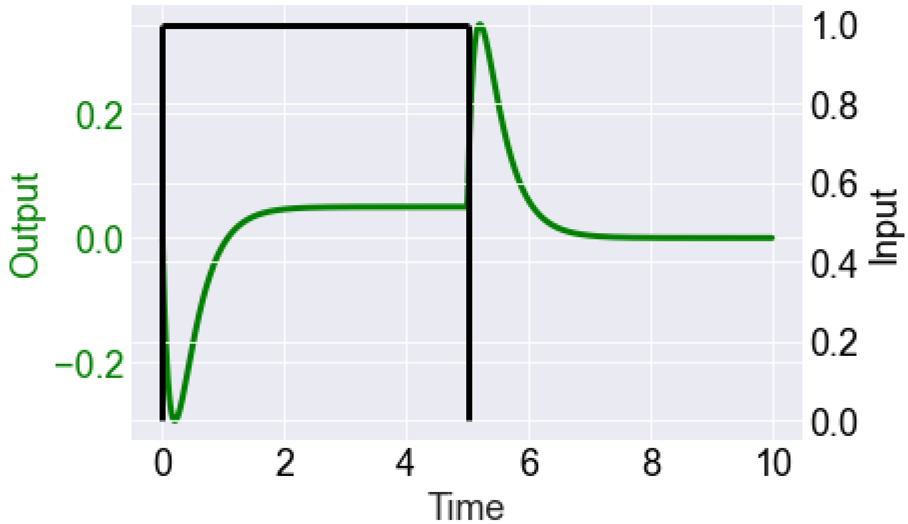


In [30]:
```
sys1.InputFunction(1, 'Impulse')
PlotResponse(sys1)
```



In [31]:
```
sys1.InputFunction(1, 'Square', InputEndTime = 5)
PlotResponse(sys1)
```

```
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWarning: return
_x specified for a transfer function system. Internal conversion to state space used; re
sults may meaningless.
  warnings.warn(
```



In [32]:

```
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')
```

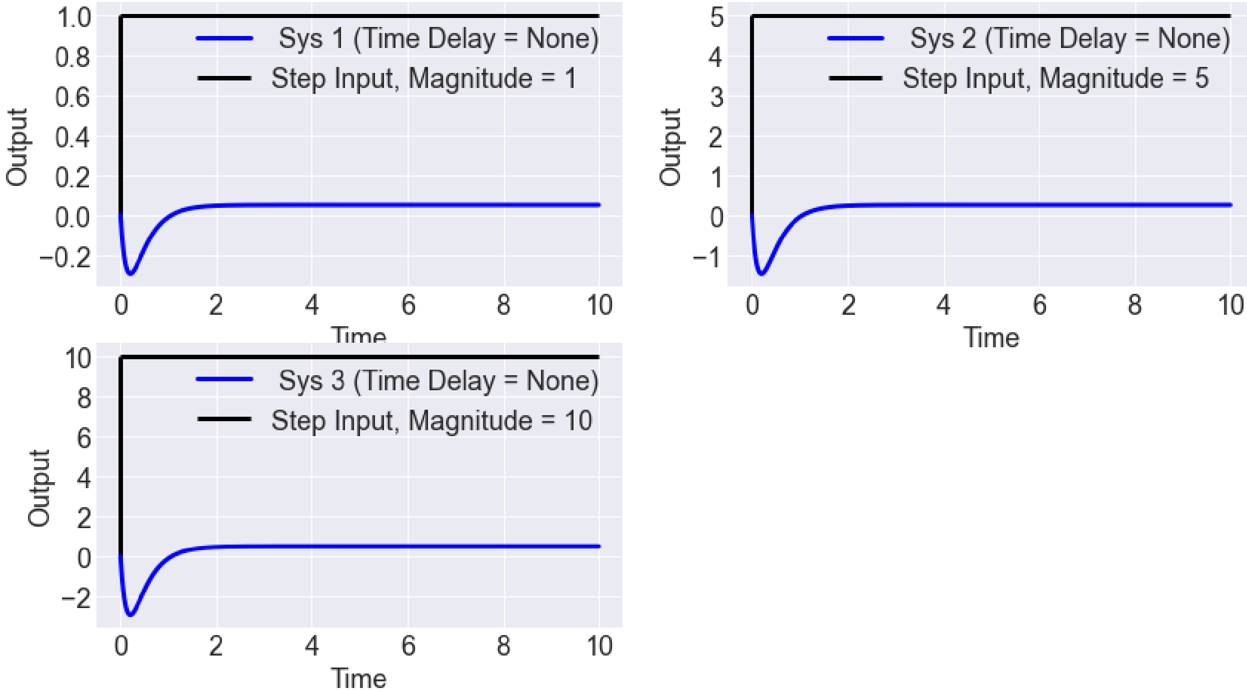
```
######################################################
#####  Sys 1 Characteristics
######################################################
## Zeros: [0.25]
## Damping Coefficient: 1.0 (Critically Damped)
## Poles:                [-7.236, -2.764]
######################################################

######################################################
#####  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]
######################################################
```

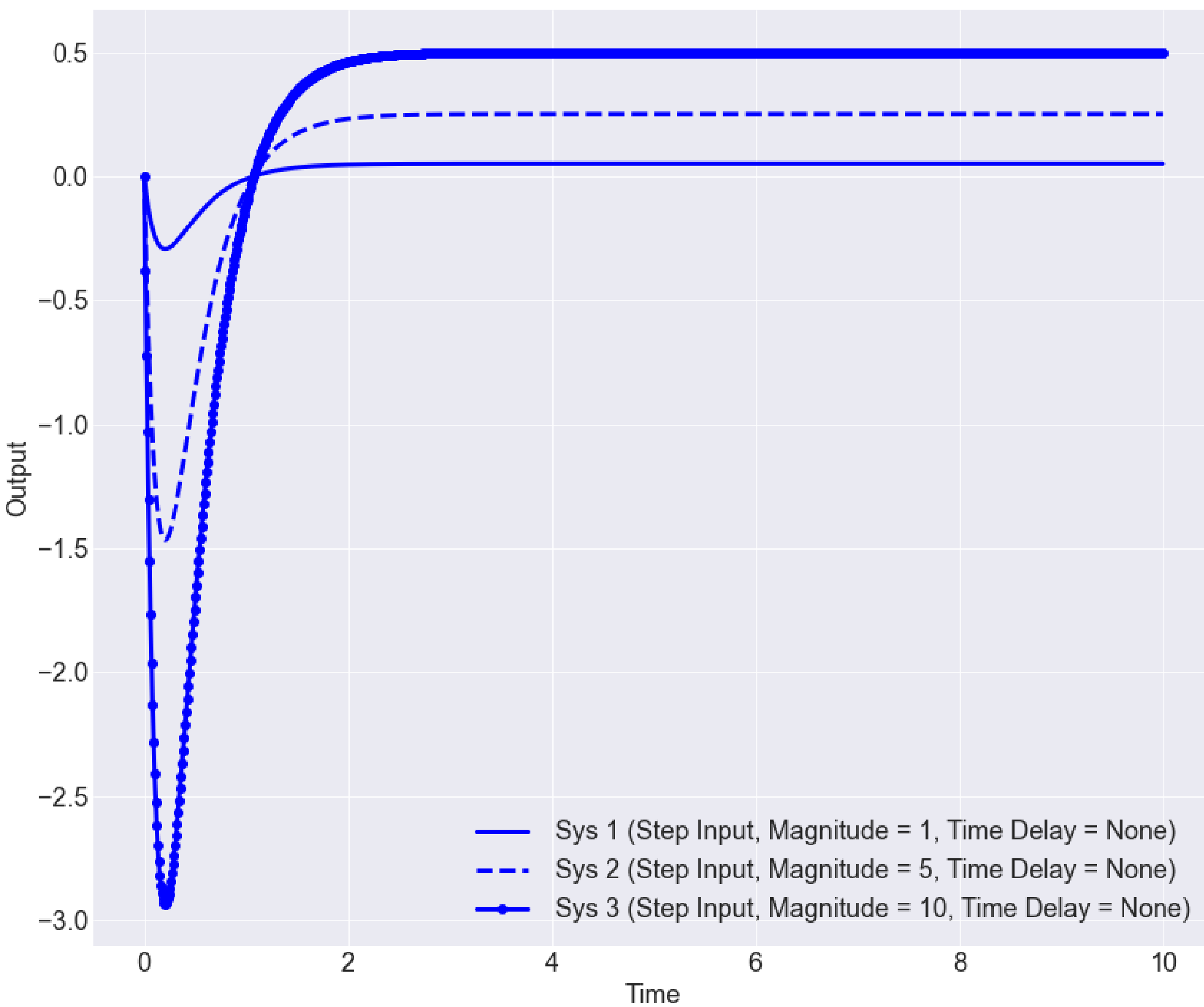
In [33]:

```
sys1.InputFunction(1, 'Step')
sys2.InputFunction(5, 'Step')
sys3.InputFunction(10, 'Step')
```

In [34]:

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





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

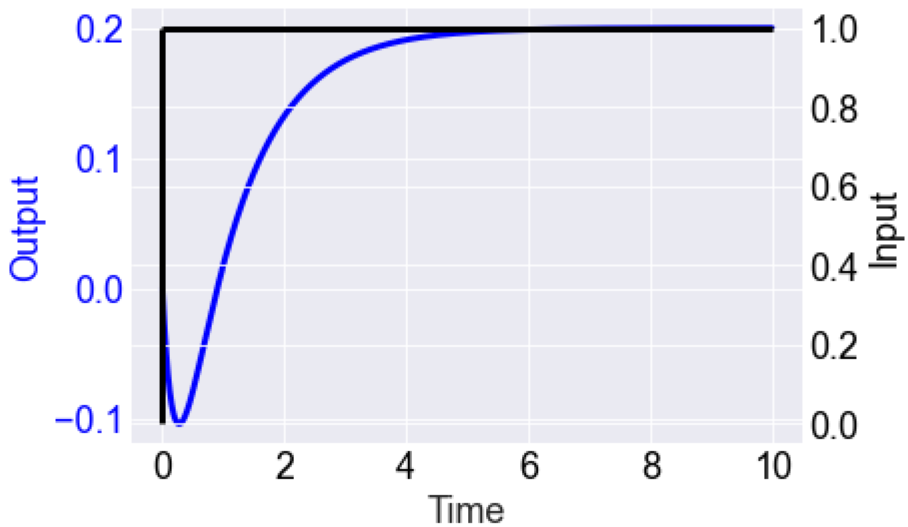
Time delays can be added with the argument "TimeDelay." Time delays are computed using the Pade approximation which will be covered later in the course.

In [35]:
```
sys1 = TransferFunction(1, [1,5], TimeDelay = 2)
```

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

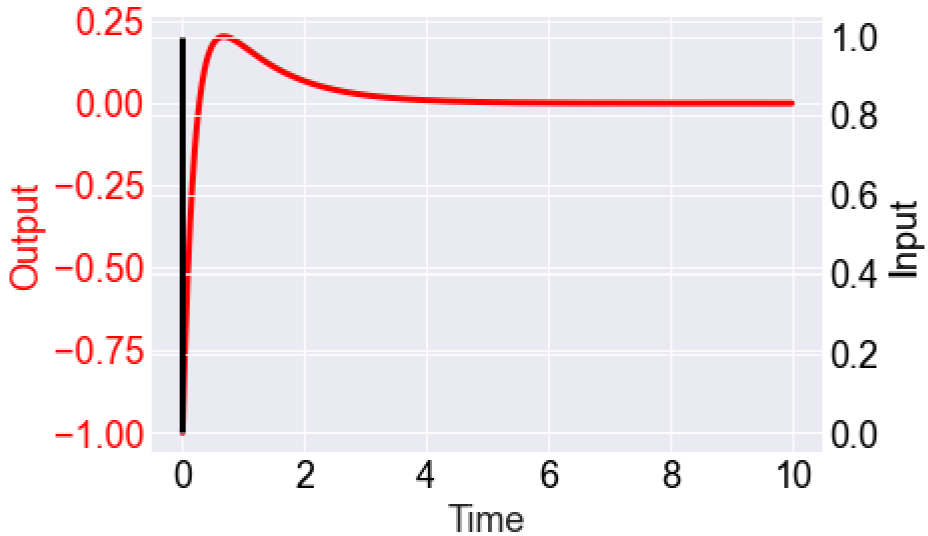
In [36]:
```
sys1.InputFunction(1, 'Step')
```

In [37]:
```
PlotResponse(sys1)
```



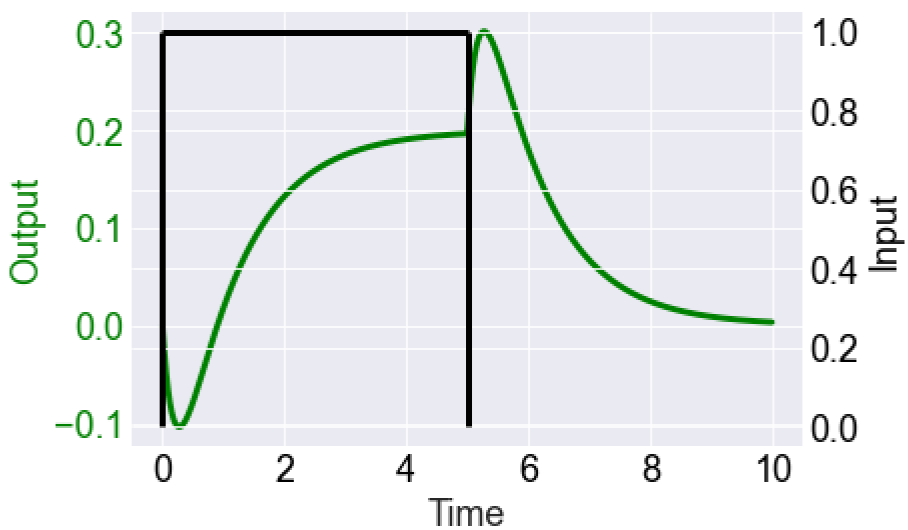
In [38]:
```
sys2 = TransferFunction(1, [1,5], TimeDelay = 2)
sys2.InputFunction(1, 'Impulse')
PlotResponse(sys2)
```

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

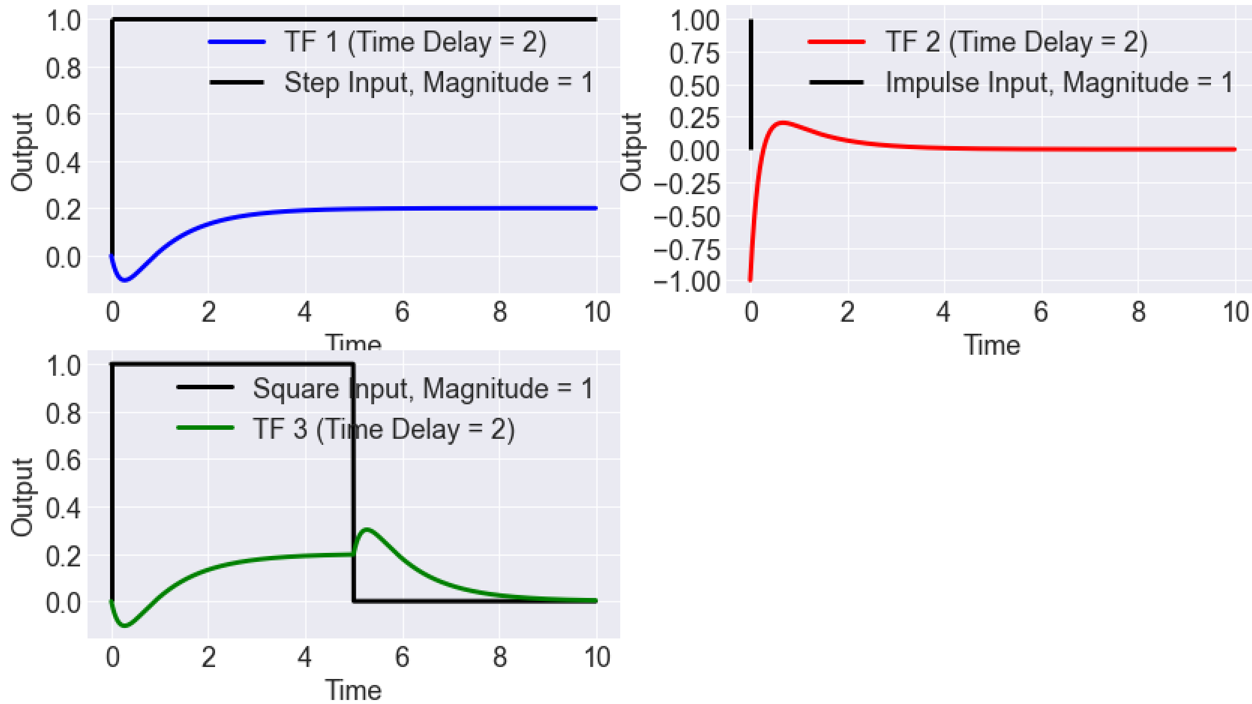


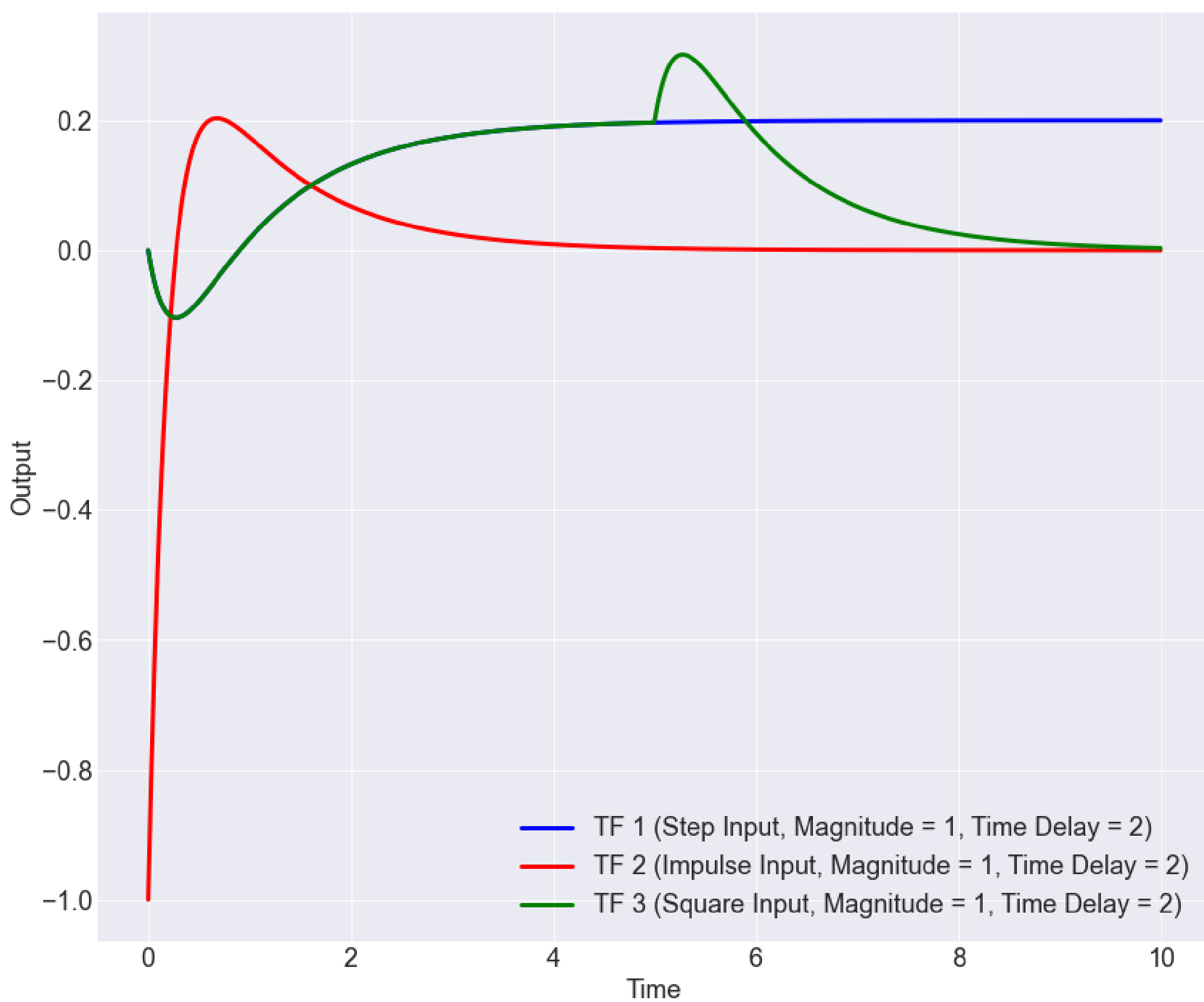
In [39]:
```
sys3 = TransferFunction(1, [1,5], TimeDelay = 2)
sys3.InputFunction(1, 'Square', InputEndTime = 5)
PlotResponse(sys3)
```

```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: [1.0]
## Poles:               [-5.0, -1.0]
######################################################
C:\Users\Andrew\anaconda3\lib\site-packages\control\timeresp.py:293: UserWarning: return
_x specified for a transfer function system. Internal conversion to state space used; re
sults may meaningless.
  warnings.warn(
```



In [40]:
```
CompareResults(sys1, sys2, sys3)
```





Example 6: How to work with attributes of a transfer function

When you define a system using TransferFunction (e.g. sys1 = TransferFunction(Numerator = 1, Denominator = [1,1]), this makes sys1 a CLASS. The class has attributes such as time, poles, zeros, outputs, etc. You can access these with the format "sys1.__" where the attribute you want to get replaces the dashed line.

Here you'll see how to get the time and response of a system and plot it manually. You'll also see how to manually adjust the end time of the simulation and the number of points.

Step 1: Define the system and simulate a step response.

In [41]:
```
from ControlCode import *

sys = TransferFunction(Numerator=1, Denominator=[1, 1])

sys.InputFunction(Magnitude = 1, Type = 'Step')
```

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

The simulation time and output can be retrieved with sys.Time and sys.Output. You can also check the shape to see if the dimensions match what you would expect with sys.Time.shape and sys.Output.shape

In [42]:
```
sys.Time
```

Out[42]:
```
array([ 0.        ,  0.01001001,  0.02002002,  0.03003003,  0.04004004,
        0.05005005,  0.06006006,  0.07007007,  0.08008008,  0.09009009,
        0.1001001 ,  0.11011011,  0.12012012,  0.13013013,  0.14014014,
        0.15015015,  0.16016016,  0.17017017,  0.18018018,  0.19019019,
        0.2002002 ,  0.21021021,  0.22022022,  0.23023023,  0.24024024,
        0.25025025,  0.26026026,  0.27027027,  0.28028028,  0.29029029,
        0.3003003 ,  0.31031031,  0.32032032,  0.33033033,  0.34034034,
        0.35035035,  0.36036036,  0.37037037,  0.38038038,  0.39039039,
        0.4004004 ,  0.41041041,  0.42042042,  0.43043043,  0.44044044,
        0.45045045,  0.46046046,  0.47047047,  0.48048048,  0.49049049,
        0.5005005 ,  0.51051051,  0.52052052,  0.53053053,  0.54054054,
        0.55055055,  0.56056056,  0.57057057,  0.58058058,  0.59059059,
        0.6006006 ,  0.61061061,  0.62062062,  0.63063063,  0.64064064,
        0.65065065,  0.66066066,  0.67067067,  0.68068068,  0.69069069,
        0.7007007 ,  0.71071071,  0.72072072,  0.73073073,  0.74074074,
        0.75075075,  0.76076076,  0.77077077,  0.78078078,  0.79079079,
        0.8008008 ,  0.81081081,  0.82082082,  0.83083083,  0.84084084,
        0.85085085,  0.86086086,  0.87087087,  0.88088088,  0.89089089,
        0.9009009 ,  0.91091091,  0.92092092,  0.93093093,  0.94094094,
        0.95095095,  0.96096096,  0.97097097,  0.98098098,  0.99099099,
        1.001001  ,  1.01101101,  1.02102102,  1.03103103,  1.04104104,
        1.05105105,  1.06106106,  1.07107107,  1.08108108,  1.09109109,
        1.1011011 ,  1.11111111,  1.12112112,  1.13113113,  1.14114114,
        1.15115115,  1.16116116,  1.17117117,  1.18118118,  1.19119119,
        1.2012012 ,  1.21121121,  1.22122122,  1.23123123,  1.24124124,
        1.25125125,  1.26126126,  1.27127127,  1.28128128,  1.29129129,
        1.3013013 ,  1.31131131,  1.32132132,  1.33133133,  1.34134134,
        1.35135135,  1.36136136,  1.37137137,  1.38138138,  1.39139139,
        1.4014014 ,  1.41141141,  1.42142142,  1.43143143,  1.44144144,
        1.45145145,  1.46146146,  1.47147147,  1.48148148,  1.49149149,
        1.5015015 ,  1.51151151,  1.52152152,  1.53153153,  1.54154154,
        1.55155155,  1.56156156,  1.57157157,  1.58158158,  1.59159159,
```

```
       1.6016016 ,  1.61161161,  1.62162162,  1.63163163,  1.64164164,
       1.65165165,  1.66166166,  1.67167167,  1.68168168,  1.69169169,
       1.7017017 ,  1.71171171,  1.72172172,  1.73173173,  1.74174174,
       1.75175175,  1.76176176,  1.77177177,  1.78178178,  1.79179179,
       1.8018018 ,  1.81181181,  1.82182182,  1.83183183,  1.84184184,
       1.85185185,  1.86186186,  1.87187187,  1.88188188,  1.89189189,
       1.9019019 ,  1.91191191,  1.92192192,  1.93193193,  1.94194194,
       1.95195195,  1.96196196,  1.97197197,  1.98198198,  1.99199199,
       2.002002  ,  2.01201201,  2.02202202,  2.03203203,  2.04204204,
       2.05205205,  2.06206206,  2.07207207,  2.08208208,  2.09209209,
       2.1021021 ,  2.11211211,  2.12212212,  2.13213213,  2.14214214,
       2.15215215,  2.16216216,  2.17217217,  2.18218218,  2.19219219,
       2.2022022 ,  2.21221221,  2.22222222,  2.23223223,  2.24224224,
       2.25225225,  2.26226226,  2.27227227,  2.28228228,  2.29229229,
       2.3023023 ,  2.31231231,  2.32232232,  2.33233233,  2.34234234,
       2.35235235,  2.36236236,  2.37237237,  2.38238238,  2.39239239,
       2.4024024 ,  2.41241241,  2.42242242,  2.43243243,  2.44244244,
       2.45245245,  2.46246246,  2.47247247,  2.48248248,  2.49249249,
       2.5025025 ,  2.51251251,  2.52252252,  2.53253253,  2.54254254,
       2.55255255,  2.56256256,  2.57257257,  2.58258258,  2.59259259,
       2.6026026 ,  2.61261261,  2.62262262,  2.63263263,  2.64264264,
       2.65265265,  2.66266266,  2.67267267,  2.68268268,  2.69269269,
       2.7027027 ,  2.71271271,  2.72272272,  2.73273273,  2.74274274,
       2.75275275,  2.76276276,  2.77277277,  2.78278278,  2.79279279,
       2.8028028 ,  2.81281281,  2.82282282,  2.83283283,  2.84284284,
       2.85285285,  2.86286286,  2.87287287,  2.88288288,  2.89289289,
       2.9029029 ,  2.91291291,  2.92292292,  2.93293293,  2.94294294,
       2.95295295,  2.96296296,  2.97297297,  2.98298298,  2.99299299,
       3.003003  ,  3.01301301,  3.02302302,  3.03303303,  3.04304304,
       3.05305305,  3.06306306,  3.07307307,  3.08308308,  3.09309309,
       3.1031031 ,  3.11311311,  3.12312312,  3.13313313,  3.14314314,
       3.15315315,  3.16316316,  3.17317317,  3.18318318,  3.19319319,
       3.2032032 ,  3.21321321,  3.22322322,  3.23323323,  3.24324324,
       3.25325325,  3.26326326,  3.27327327,  3.28328328,  3.29329329,
       3.3033033 ,  3.31331331,  3.32332332,  3.33333333,  3.34334334,
       3.35335335,  3.36336336,  3.37337337,  3.38338338,  3.39339339,
       3.4034034 ,  3.41341341,  3.42342342,  3.43343343,  3.44344344,
       3.45345345,  3.46346346,  3.47347347,  3.48348348,  3.49349349,
       3.5035035 ,  3.51351351,  3.52352352,  3.53353353,  3.54354354,
       3.55355355,  3.56356356,  3.57357357,  3.58358358,  3.59359359,
       3.6036036 ,  3.61361361,  3.62362362,  3.63363363,  3.64364364,
       3.65365365,  3.66366366,  3.67367367,  3.68368368,  3.69369369,
       3.7037037 ,  3.71371371,  3.72372372,  3.73373373,  3.74374374,
       3.75375375,  3.76376376,  3.77377377,  3.78378378,  3.79379379,
       3.8038038 ,  3.81381381,  3.82382382,  3.83383383,  3.84384384,
       3.85385385,  3.86386386,  3.87387387,  3.88388388,  3.89389389,
       3.9039039 ,  3.91391391,  3.92392392,  3.93393393,  3.94394394,
       3.95395395,  3.96396396,  3.97397397,  3.98398398,  3.99399399,
       4.004004  ,  4.01401401,  4.02402402,  4.03403403,  4.04404404,
       4.05405405,  4.06406406,  4.07407407,  4.08408408,  4.09409409,
       4.1041041 ,  4.11411411,  4.12412412,  4.13413413,  4.14414414,
       4.15415415,  4.16416416,  4.17417417,  4.18418418,  4.19419419,
       4.2042042 ,  4.21421421,  4.22422422,  4.23423423,  4.24424424,
       4.25425425,  4.26426426,  4.27427427,  4.28428428,  4.29429429,
       4.3043043 ,  4.31431431,  4.32432432,  4.33433433,  4.34434434,
       4.35435435,  4.36436436,  4.37437437,  4.38438438,  4.39439439,
       4.4044044 ,  4.41441441,  4.42442442,  4.43443443,  4.44444444,
       4.45445445,  4.46446446,  4.47447447,  4.48448448,  4.49449449,
       4.5045045 ,  4.51451451,  4.52452452,  4.53453453,  4.54454454,
       4.55455455,  4.56456456,  4.57457457,  4.58458458,  4.59459459,
       4.6046046 ,  4.61461461,  4.62462462,  4.63463463,  4.64464464,
       4.65465465,  4.66466466,  4.67467467,  4.68468468,  4.69469469,
       4.7047047 ,  4.71471471,  4.72472472,  4.73473473,  4.74474474,
       4.75475475,  4.76476476,  4.77477477,  4.78478478,  4.79479479,
       4.8048048 ,  4.81481481,  4.82482482,  4.83483483,  4.84484484,
```

```
4.85485485, 4.86486486, 4.87487487, 4.88488488, 4.89489489,
4.9049049 , 4.91491491, 4.92492492, 4.93493493, 4.94494494,
4.95495495, 4.96496496, 4.97497497, 4.98498498, 4.99499499,
5.00500501, 5.01501502, 5.02502503, 5.03503504, 5.04504505,
5.05505506, 5.06506507, 5.07507508, 5.08508509, 5.0950951 ,
5.10510511, 5.11511512, 5.12512513, 5.13513514, 5.14514515,
5.15515516, 5.16516517, 5.17517518, 5.18518519, 5.1951952 ,
5.20520521, 5.21521522, 5.22522523, 5.23523524, 5.24524525,
5.25525526, 5.26526527, 5.27527528, 5.28528529, 5.2952953 ,
5.30530531, 5.31531532, 5.32532533, 5.33533534, 5.34534535,
5.35535536, 5.36536537, 5.37537538, 5.38538539, 5.3953954 ,
5.40540541, 5.41541542, 5.42542543, 5.43543544, 5.44544545,
5.45545546, 5.46546547, 5.47547548, 5.48548549, 5.4954955 ,
5.50550551, 5.51551552, 5.52552553, 5.53553554, 5.54554555,
5.55555556, 5.56556557, 5.57557558, 5.58558559, 5.5955956 ,
5.60560561, 5.61561562, 5.62562563, 5.63563564, 5.64564565,
5.65565566, 5.66566567, 5.67567568, 5.68568569, 5.6956957 ,
5.70570571, 5.71571572, 5.72572573, 5.73573574, 5.74574575,
5.75575576, 5.76576577, 5.77577578, 5.78578579, 5.7957958 ,
5.80580581, 5.81581582, 5.82582583, 5.83583584, 5.84584585,
5.85585586, 5.86586587, 5.87587588, 5.88588589, 5.8958959 ,
5.90590591, 5.91591592, 5.92592593, 5.93593594, 5.94594595,
5.95595596, 5.96596597, 5.97597598, 5.98598599, 5.995996  ,
6.00600601, 6.01601602, 6.02602603, 6.03603604, 6.04604605,
6.05605606, 6.06606607, 6.07607608, 6.08608609, 6.0960961 ,
6.10610611, 6.11611612, 6.12612613, 6.13613614, 6.14614615,
6.15615616, 6.16616617, 6.17617618, 6.18618619, 6.1961962 ,
6.20620621, 6.21621622, 6.22622623, 6.23623624, 6.24624625,
6.25625626, 6.26626627, 6.27627628, 6.28628629, 6.2962963 ,
6.30630631, 6.31631632, 6.32632633, 6.33633634, 6.34634635,
6.35635636, 6.36636637, 6.37637638, 6.38638639, 6.3963964 ,
6.40640641, 6.41641642, 6.42642643, 6.43643644, 6.44644645,
6.45645646, 6.46646647, 6.47647648, 6.48648649, 6.4964965 ,
6.50650651, 6.51651652, 6.52652653, 6.53653654, 6.54654655,
6.55655656, 6.56656657, 6.57657658, 6.58658659, 6.5965966 ,
6.60660661, 6.61661662, 6.62662663, 6.63663664, 6.64664665,
6.65665666, 6.66666667, 6.67667668, 6.68668669, 6.6966967 ,
6.70670671, 6.71671672, 6.72672673, 6.73673674, 6.74674675,
6.75675676, 6.76676677, 6.77677678, 6.78678679, 6.7967968 ,
6.80680681, 6.81681682, 6.82682683, 6.83683684, 6.84684685,
6.85685686, 6.86686687, 6.87687688, 6.88688689, 6.8968969 ,
6.90690691, 6.91691692, 6.92692693, 6.93693694, 6.94694695,
6.95695696, 6.96696697, 6.97697698, 6.98698699, 6.996997  ,
7.00700701, 7.01701702, 7.02702703, 7.03703704, 7.04704705,
7.05705706, 7.06706707, 7.07707708, 7.08708709, 7.0970971 ,
7.10710711, 7.11711712, 7.12712713, 7.13713714, 7.14714715,
7.15715716, 7.16716717, 7.17717718, 7.18718719, 7.1971972 ,
7.20720721, 7.21721722, 7.22722723, 7.23723724, 7.24724725,
7.25725726, 7.26726727, 7.27727728, 7.28728729, 7.2972973 ,
7.30730731, 7.31731732, 7.32732733, 7.33733734, 7.34734735,
7.35735736, 7.36736737, 7.37737738, 7.38738739, 7.3973974 ,
7.40740741, 7.41741742, 7.42742743, 7.43743744, 7.44744745,
7.45745746, 7.46746747, 7.47747748, 7.48748749, 7.4974975 ,
7.50750751, 7.51751752, 7.52752753, 7.53753754, 7.54754755,
7.55755756, 7.56756757, 7.57757758, 7.58758759, 7.5975976 ,
7.60760761, 7.61761762, 7.62762763, 7.63763764, 7.64764765,
7.65765766, 7.66766767, 7.67767768, 7.68768769, 7.6976977 ,
7.70770771, 7.71771772, 7.72772773, 7.73773774, 7.74774775,
7.75775776, 7.76776777, 7.77777778, 7.78778779, 7.7977978 ,
7.80780781, 7.81781782, 7.82782783, 7.83783784, 7.84784785,
7.85785786, 7.86786787, 7.87787788, 7.88788789, 7.8978979 ,
7.90790791, 7.91791792, 7.92792793, 7.93793794, 7.94794795,
7.95795796, 7.96796797, 7.97797798, 7.98798799, 7.997998  ,
8.00800801, 8.01801802, 8.02802803, 8.03803804, 8.04804805,
8.05805806, 8.06806807, 8.07807808, 8.08808809, 8.0980981 ,
```

```
8.10810811, 8.11811812, 8.12812813, 8.13813814, 8.14814815,
8.15815816, 8.16816817, 8.17817818, 8.18818819, 8.1981982 ,
8.20820821, 8.21821822, 8.22822823, 8.23823824, 8.24824825,
8.25825826, 8.26826827, 8.27827828, 8.28828829, 8.2982983 ,
8.30830831, 8.31831832, 8.32832833, 8.33833834, 8.34834835,
8.35835836, 8.36836837, 8.37837838, 8.38838839, 8.3983984 ,
8.40840841, 8.41841842, 8.42842843, 8.43843844, 8.44844845,
8.45845846, 8.46846847, 8.47847848, 8.48848849, 8.4984985 ,
8.50850851, 8.51851852, 8.52852853, 8.53853854, 8.54854855,
8.55855856, 8.56856857, 8.57857858, 8.58858859, 8.5985986 ,
8.60860861, 8.61861862, 8.62862863, 8.63863864, 8.64864865,
8.65865866, 8.66866867, 8.67867868, 8.68868869, 8.6986987 ,
8.70870871, 8.71871872, 8.72872873, 8.73873874, 8.74874875,
8.75875876, 8.76876877, 8.77877878, 8.78878879, 8.7987988 ,
8.80880881, 8.81881882, 8.82882883, 8.83883884, 8.84884885,
8.85885886, 8.86886887, 8.87887888, 8.88888889, 8.8988989 ,
8.90890891, 8.91891892, 8.92892893, 8.93893894, 8.94894895,
8.95895896, 8.96896897, 8.97897898, 8.98898899, 8.998999  ,
9.00900901, 9.01901902, 9.02902903, 9.03903904, 9.04904905,
9.05905906, 9.06906907, 9.07907908, 9.08908909, 9.0990991 ,
9.10910911, 9.11911912, 9.12912913, 9.13913914, 9.14914915,
9.15915916, 9.16916917, 9.17917918, 9.18918919, 9.1991992 ,
9.20920921, 9.21921922, 9.22922923, 9.23923924, 9.24924925,
9.25925926, 9.26926927, 9.27927928, 9.28928929, 9.2992993 ,
9.30930931, 9.31931932, 9.32932933, 9.33933934, 9.34934935,
9.35935936, 9.36936937, 9.37937938, 9.38938939, 9.3993994 ,
9.40940941, 9.41941942, 9.42942943, 9.43943944, 9.44944945,
9.45945946, 9.46946947, 9.47947948, 9.48948949, 9.4994995 ,
9.50950951, 9.51951952, 9.52952953, 9.53953954, 9.54954955,
9.55955956, 9.56956957, 9.57957958, 9.58958959, 9.5995996 ,
9.60960961, 9.61961962, 9.62962963, 9.63963964, 9.64964965,
9.65965966, 9.66966967, 9.67967968, 9.68968969, 9.6996997 ,
9.70970971, 9.71971972, 9.72972973, 9.73973974, 9.74974975,
9.75975976, 9.76976977, 9.77977978, 9.78978979, 9.7997998 ,
9.80980981, 9.81981982, 9.82982983, 9.83983984, 9.84984985,
9.85985986, 9.86986987, 9.87987988, 9.88988989, 9.8998999 ,
9.90990991, 9.91991992, 9.92992993, 9.93993994, 9.94994995,
9.95995996, 9.96996997, 9.97997998, 9.98998999, 10.        ])
```

In [43]:

```python
sys.Time.shape
```

Out[43]:

```
(1000,)
```

In [44]:

```python
sys.Output
```

Out[44]:

```
array([0.        , 0.00996008, 0.01982095, 0.02958361, 0.03924903,
       0.04881818, 0.05829203, 0.06767151, 0.07695757, 0.08615115,
       0.09525315, 0.1042645 , 0.11318609, 0.12201883, 0.13076359,
       0.13942125, 0.14799268, 0.15647874, 0.16488027, 0.17319813,
       0.18143314, 0.18958613, 0.19765791, 0.2056493 , 0.2135611 ,
       0.22139409, 0.22914906, 0.2368268 , 0.24442806, 0.25195361,
       0.25940421, 0.2667806 , 0.27408353, 0.28131371, 0.28847188,
       0.29555875, 0.30257504, 0.30952145, 0.31639867, 0.32320739,
       0.3299483 , 0.33662206, 0.34322936, 0.34977084, 0.35624718,
       0.362659  , 0.36900697, 0.37529171, 0.38151385, 0.38767402,
       0.39377283, 0.3998109 , 0.40578883, 0.41170722, 0.41756666,
       0.42336774, 0.42911104, 0.43479714, 0.44042661, 0.446     ,
       0.45151788, 0.45698081, 0.46238932, 0.46774396, 0.47304527,
       0.47829378, 0.48349002, 0.4886345 , 0.49372773, 0.49877025,
       0.50376253, 0.5087051 , 0.51359843, 0.51844303, 0.52323937,
       0.52798794, 0.53268922, 0.53734367, 0.54195176, 0.54651396,
       0.55103071, 0.55550248, 0.55992971, 0.56431285, 0.56865232,
       0.57294858, 0.57720204, 0.58141314, 0.5855823 , 0.58970993,
       0.59379645, 0.59784227, 0.60184779, 0.60581342, 0.60973955,
```

```
       0.61362657, 0.61747488, 0.62128486, 0.62505689, 0.62879136,
       0.63248862, 0.63614906, 0.63977305, 0.64336093, 0.64691309,
       0.65042986, 0.65391161, 0.65735867, 0.66077141, 0.66415015,
       0.66749524, 0.67080701, 0.6740858 , 0.67733193, 0.68054573,
       0.68372752, 0.68687762, 0.68999634, 0.693084  , 0.69614091,
       0.69916737, 0.70216368, 0.70513015, 0.70806708, 0.71097475,
       0.71385347, 0.71670351, 0.71952516, 0.72231872, 0.72508444,
       0.72782262, 0.73053353, 0.73321744, 0.73587461, 0.73850532,
       0.74110983, 0.74368839, 0.74624128, 0.74876873, 0.75127102,
       0.75374838, 0.75620106, 0.75862932, 0.76103339, 0.76341351,
       0.76576993, 0.76810288, 0.77041259, 0.7726993 , 0.77496324,
       0.77720462, 0.77942368, 0.78162063, 0.78379571, 0.78594912,
       0.78808108, 0.79019181, 0.79228152, 0.79435041, 0.7963987 ,
       0.79842658, 0.80043427, 0.80242196, 0.80438985, 0.80633814,
       0.80826703, 0.8101767 , 0.81206736, 0.81393918, 0.81579236,
       0.81762708, 0.81944353, 0.82124189, 0.82302233, 0.82478504,
       0.8265302 , 0.82825797, 0.82996854, 0.83166206, 0.83333872,
       0.83499868, 0.83664211, 0.83826916, 0.83988001, 0.84147482,
       0.84305374, 0.84461694, 0.84616457, 0.84769678, 0.84921373,
       0.85071558, 0.85220246, 0.85367453, 0.85513195, 0.85657484,
       0.85800337, 0.85941767, 0.86081788, 0.86220414, 0.8635766 ,
       0.86493539, 0.86628064, 0.8676125 , 0.86893109, 0.87023654,
       0.871529  , 0.87280858, 0.87407541, 0.87532963, 0.87657136,
       0.87780072, 0.87901783, 0.88022282, 0.88141581, 0.88259692,
       0.88376626, 0.88492396, 0.88607013, 0.88720488, 0.88832833,
       0.88944058, 0.89054176, 0.89163198, 0.89271133, 0.89377993,
       0.89483789, 0.89588532, 0.89692231, 0.89794897, 0.8989654 ,
       0.89997172, 0.90096801, 0.90195437, 0.90293091, 0.90389773,
       0.90485492, 0.90580257, 0.90674078, 0.90766965, 0.90858927,
       0.90949973, 0.91040112, 0.91129353, 0.91217705, 0.91305177,
       0.91391779, 0.91477517, 0.91562402, 0.91646441, 0.91729643,
       0.91812016, 0.91893569, 0.9197431 , 0.92054246, 0.92133387,
       0.92211739, 0.9228931 , 0.92366109, 0.92442144, 0.9251742 ,
       0.92591948, 0.92665732, 0.92738782, 0.92811104, 0.92882706,
       0.92953595, 0.93023778, 0.93093262, 0.93162053, 0.9323016 ,
       0.93297588, 0.93364344, 0.93430436, 0.93495869, 0.93560651,
       0.93624787, 0.93688285, 0.9375115 , 0.93813389, 0.93875008,
       0.93936014, 0.93996412, 0.94056208, 0.94115408, 0.94174019,
       0.94232047, 0.94289496, 0.94346373, 0.94402683, 0.94458433,
       0.94513628, 0.94568272, 0.94622373, 0.94675934, 0.94728962,
       0.94781462, 0.94833439, 0.94884899, 0.94935845, 0.94986285,
       0.95036222, 0.95085661, 0.95134609, 0.95183068, 0.95231045,
       0.95278544, 0.95325571, 0.95372128, 0.95418222, 0.95463857,
       0.95509037, 0.95553768, 0.95598052, 0.95641896, 0.95685303,
       0.95728278, 0.95770825, 0.95812948, 0.95854651, 0.95895939,
       0.95936816, 0.95977285, 0.96017352, 0.96057019, 0.96096292,
       0.96135173, 0.96173667, 0.96211777, 0.96249508, 0.96286864,
       0.96323847, 0.96360462, 0.96396712, 0.96432601, 0.96468132,
       0.9650331 , 0.96538137, 0.96572618, 0.96606755, 0.96640552,
       0.96674012, 0.96707139, 0.96739936, 0.96772407, 0.96804554,
       0.96836381, 0.96867891, 0.96899087, 0.96929972, 0.9696055 ,
       0.96990823, 0.97020794, 0.97050468, 0.97079845, 0.9710893 ,
       0.97137725, 0.97166234, 0.97194458, 0.97222402, 0.97250067,
       0.97277456, 0.97304573, 0.9733142 , 0.97357999, 0.97384314,
       0.97410366, 0.97436159, 0.97461695, 0.97486977, 0.97512007,
       0.97536787, 0.97561321, 0.9758561 , 0.97609658, 0.97633466,
       0.97657037, 0.97680373, 0.97703477, 0.9772635 , 0.97748996,
       0.97771416, 0.97793613, 0.97815589, 0.97837346, 0.97858886,
       0.97880211, 0.97901325, 0.97922228, 0.97942922, 0.97963411,
       0.97983696, 0.98003778, 0.98023661, 0.98043345, 0.98062834,
       0.98082128, 0.9810123 , 0.98120142, 0.98138866, 0.98157403,
       0.98175755, 0.98193925, 0.98211913, 0.98229723, 0.98247355,
       0.98264811, 0.98282094, 0.98299204, 0.98316145, 0.98332916,
       0.9834952 , 0.98365959, 0.98382234, 0.98398347, 0.984143  ,
       0.98430094, 0.9844573 , 0.98461211, 0.98476537, 0.98491711,
```

```
0.98506734, 0.98521607, 0.98536331, 0.9855091 , 0.98565343,
0.98579632, 0.98593779, 0.98607785, 0.98621652, 0.9863538 ,
0.98648972, 0.98662428, 0.98675751, 0.9868894 , 0.98701998,
0.98714927, 0.98727726, 0.98740398, 0.98752944, 0.98765364,
0.98777662, 0.98789836, 0.98801889, 0.98813823, 0.98825637,
0.98837334, 0.98848914, 0.98860379, 0.9887173 , 0.98882967,
0.98894093, 0.98905108, 0.98916013, 0.9892681 , 0.98937499,
0.98948082, 0.98958559, 0.98968932, 0.98979201, 0.98989368,
0.98999434, 0.990094  , 0.99019266, 0.99029035, 0.99038705,
0.9904828 , 0.99057759, 0.99067144, 0.99076435, 0.99085634,
0.99094741, 0.99103758, 0.99112684, 0.99121522, 0.99130272,
0.99138934, 0.99147511, 0.99156002, 0.99164408, 0.9917273 ,
0.9918097 , 0.99189128, 0.99197204, 0.992052  , 0.99213116,
0.99220954, 0.99228713, 0.99236395, 0.99244001, 0.9925153 ,
0.99258985, 0.99266366, 0.99273673, 0.99280907, 0.99288069,
0.9929516 , 0.9930218 , 0.99309131, 0.99316012, 0.99322825,
0.99329569, 0.99336247, 0.99342858, 0.99349403, 0.99355883,
0.99362298, 0.9936865 , 0.99374938, 0.99381164, 0.99387328,
0.9939343 , 0.99399471, 0.99405453, 0.99411374, 0.99417237,
0.99423042, 0.99428788, 0.99434477, 0.9944011 , 0.99445687,
0.99451208, 0.99456674, 0.99462085, 0.99467443, 0.99472747,
0.99477999, 0.99483198, 0.99488345, 0.99493441, 0.99498487,
0.99503482, 0.99508427, 0.99513323, 0.99518171, 0.9952297 ,
0.99527721, 0.99532425, 0.99537082, 0.99541693, 0.99546257,
0.99550777, 0.99555251, 0.99559681, 0.99564066, 0.99568408,
0.99572707, 0.99576963, 0.99581176, 0.99585348, 0.99589478,
0.99593567, 0.99597615, 0.99601623, 0.9960559 , 0.99609519,
0.99613408, 0.99617258, 0.99621071, 0.99624845, 0.99628581,
0.99632281, 0.99635943, 0.99639569, 0.99643159, 0.99646713,
0.99650232, 0.99653716, 0.99657165, 0.99660579, 0.9966396 ,
0.99667307, 0.99670621, 0.99673901, 0.99677149, 0.99680365,
0.99683549, 0.996867  , 0.99689821, 0.9969291 , 0.99695969,
0.99698997, 0.99701995, 0.99704963, 0.99707902, 0.99710811,
0.99713692, 0.99716543, 0.99719366, 0.99722162, 0.99724929,
0.99727669, 0.99730381, 0.99733067, 0.99735725, 0.99738357,
0.99740963, 0.99743543, 0.99746098, 0.99748627, 0.9975113 ,
0.99753609, 0.99756063, 0.99758493, 0.99760898, 0.9976328 ,
0.99765637, 0.99767972, 0.99770283, 0.99772571, 0.99774836,
0.99777079, 0.99779299, 0.99781497, 0.99783673, 0.99785828,
0.99787961, 0.99790073, 0.99792164, 0.99794234, 0.99796284,
0.99798313, 0.99800321, 0.9980231 , 0.99804279, 0.99806229,
0.99808159, 0.99810069, 0.99811961, 0.99813834, 0.99815688,
0.99817524, 0.99819341, 0.99821141, 0.99822922, 0.99824686,
0.99826432, 0.99828161, 0.99829872, 0.99831567, 0.99833244,
0.99834905, 0.9983655 , 0.99838178, 0.99839789, 0.99841385,
0.99842965, 0.99844529, 0.99846078, 0.99847611, 0.99849128,
0.99850631, 0.99852119, 0.99853592, 0.9985505 , 0.99856494,
0.99857923, 0.99859338, 0.99860739, 0.99862126, 0.99863499,
0.99864859, 0.99866205, 0.99867538, 0.99868857, 0.99870163,
0.99871456, 0.99872737, 0.99874004, 0.99875259, 0.99876502,
0.99877732, 0.99878949, 0.99880155, 0.99881349, 0.99882531,
0.99883701, 0.99884859, 0.99886006, 0.99887141, 0.99888265,
0.99889378, 0.9989048 , 0.99891571, 0.99892651, 0.9989372 ,
0.99894778, 0.99895826, 0.99896864, 0.99897891, 0.99898908,
0.99899915, 0.99900912, 0.99901899, 0.99902876, 0.99903843,
0.99904801, 0.99905749, 0.99906688, 0.99907617, 0.99908538,
0.99909449, 0.9991035 , 0.99911243, 0.99912127, 0.99913003,
0.99913869, 0.99914727, 0.99915576, 0.99916417, 0.9991725 ,
0.99918074, 0.9991889 , 0.99919698, 0.99920498, 0.99921289,
0.99922073, 0.99922849, 0.99923618, 0.99924379, 0.99925132,
0.99925878, 0.99926616, 0.99927347, 0.9992807 , 0.99928787,
0.99929496, 0.99930198, 0.99930894, 0.99931582, 0.99932263,
0.99932938, 0.99933606, 0.99934267, 0.99934922, 0.9993557 ,
0.99936212, 0.99936847, 0.99937476, 0.99938099, 0.99938715,
0.99939326, 0.9993993 , 0.99940528, 0.99941121, 0.99941707,
```

```
0.99942288, 0.99942863, 0.99943432, 0.99943995, 0.99944553,
0.99945105, 0.99945652, 0.99946193, 0.99946729, 0.9994726 ,
0.99947785, 0.99948305, 0.9994882 , 0.9994933 , 0.99949834,
0.99950334, 0.99950829, 0.99951319, 0.99951803, 0.99952283,
0.99952759, 0.99953229, 0.99953695, 0.99954156, 0.99954613,
0.99955065, 0.99955513, 0.99955956, 0.99956394, 0.99956829,
0.99957259, 0.99957684, 0.99958106, 0.99958523, 0.99958936,
0.99959345, 0.9995975 , 0.99960151, 0.99960548, 0.99960941,
0.9996133 , 0.99961715, 0.99962096, 0.99962474, 0.99962848,
0.99963218, 0.99963584, 0.99963947, 0.99964306, 0.99964661,
0.99965013, 0.99965362, 0.99965707, 0.99966048, 0.99966387,
0.99966721, 0.99967053, 0.99967381, 0.99967706, 0.99968027,
0.99968346, 0.99968661, 0.99968973, 0.99969282, 0.99969588,
0.99969891, 0.99970191, 0.99970488, 0.99970782, 0.99971073,
0.99971361, 0.99971646, 0.99971929, 0.99972208, 0.99972485,
0.99972759, 0.9997303 , 0.99973299, 0.99973565, 0.99973828,
0.99974089, 0.99974347, 0.99974603, 0.99974856, 0.99975106,
0.99975354, 0.99975599, 0.99975842, 0.99976083, 0.99976321,
0.99976557, 0.99976791, 0.99977022, 0.99977251, 0.99977477,
0.99977702, 0.99977924, 0.99978144, 0.99978361, 0.99978577,
0.9997879 , 0.99979001, 0.99979211, 0.99979418, 0.99979623,
0.99979826, 0.99980026, 0.99980225, 0.99980422, 0.99980617,
0.9998081 , 0.99981002, 0.99981191, 0.99981378, 0.99981564,
0.99981747, 0.99981929, 0.99982109, 0.99982287, 0.99982464,
0.99982638, 0.99982811, 0.99982982, 0.99983152, 0.9998332 ,
0.99983486, 0.9998365 , 0.99983813, 0.99983974, 0.99984134,
0.99984292, 0.99984449, 0.99984603, 0.99984757, 0.99984909,
0.99985059, 0.99985208, 0.99985355, 0.99985501, 0.99985645,
0.99985788, 0.9998593 , 0.9998607 , 0.99986209, 0.99986346,
0.99986482, 0.99986617, 0.9998675 , 0.99986882, 0.99987013,
0.99987142, 0.9998727 , 0.99987397, 0.99987522, 0.99987647,
0.9998777 , 0.99987892, 0.99988012, 0.99988132, 0.9998825 ,
0.99988367, 0.99988483, 0.99988597, 0.99988711, 0.99988823,
0.99988935, 0.99989045, 0.99989154, 0.99989262, 0.99989369,
0.99989475, 0.9998958 , 0.99989683, 0.99989786, 0.99989888,
0.99989989, 0.99990088, 0.99990187, 0.99990285, 0.99990382,
0.99990477, 0.99990572, 0.99990666, 0.99990759, 0.99990851,
0.99990942, 0.99991033, 0.99991122, 0.9999121 , 0.99991298,
0.99991384, 0.9999147 , 0.99991555, 0.99991639, 0.99991723,
0.99991805, 0.99991887, 0.99991967, 0.99992048, 0.99992127,
0.99992205, 0.99992283, 0.9999236 , 0.99992436, 0.99992511,
0.99992586, 0.9999266 , 0.99992733, 0.99992805, 0.99992877,
0.99992948, 0.99993018, 0.99993087, 0.99993156, 0.99993224,
0.99993292, 0.99993359, 0.99993425, 0.9999349 , 0.99993555,
0.99993619, 0.99993683, 0.99993746, 0.99993808, 0.9999387 ,
0.99993931, 0.99993991, 0.99994051, 0.9999411 , 0.99994169,
0.99994227, 0.99994285, 0.99994342, 0.99994398, 0.99994454,
0.99994509, 0.99994564, 0.99994618, 0.99994671, 0.99994724,
0.99994777, 0.99994829, 0.99994881, 0.99994932, 0.99994982,
0.99995032, 0.99995081, 0.9999513 , 0.99995179, 0.99995227,
0.99995275, 0.99995322, 0.99995368, 0.99995414, 0.9999546 ])
```

In [45]:

```
sys.Output.shape, sys.Time[-1] #indexing with -1 gives the last value
```

Out[45]:

```
((1000,), 10.0)
```

So both the simulation time and the simulation output are 1x1000 arrays by default. The simulation time is 10 so there are enough points to make a smooth curve.

Step 2: Manually change the final simulation time and number of points

This can be done when you initially define the transfer function by changing TFinal and NumPoints.

In [46]:
```
sys = TransferFunction(Numerator=1, Denominator=[1, 1],\
                       TFinal=10, NumPoints=50)

sys.InputFunction(Magnitude = 1, Type = 'Step')
```

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

In [47]:
```
sys.Time.shape, sys.Output.shape
```

Out[47]:
```
((50,), (50,))
```

So the total simulation time is 10 with 50 points which means the time and output will be 1x50 arrays. You can choose different numbers. The more points you have compared to the final simulation time, the smoother your output curve will be.

In [48]:
```
sys_more_points = TransferFunction(Numerator=1, Denominator=[1, 1],\
                       TFinal=10, NumPoints=1000)

sys_more_points.InputFunction(Magnitude = 1, Type = 'Step')
```

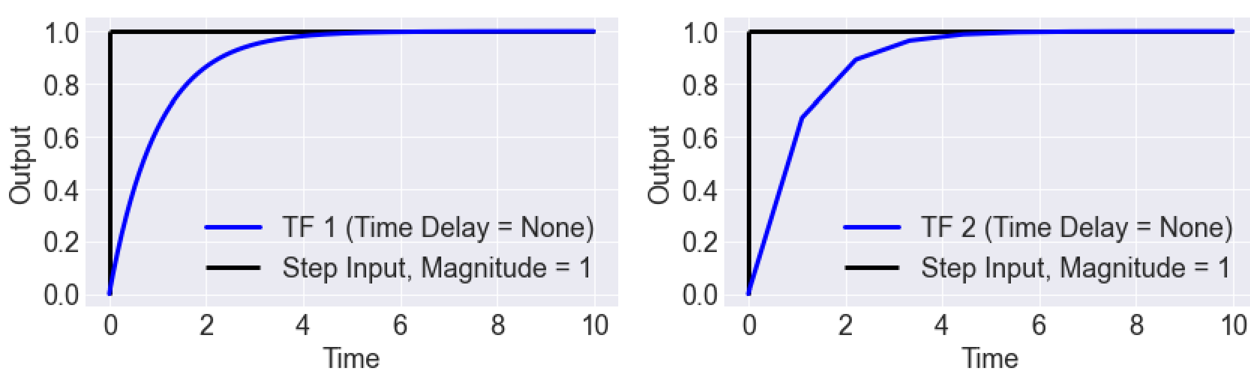
```
######################################################
##### Transfer Function Characteristics
######################################################
## Zeros: None
## Poles:                [-1.0]
######################################################
```

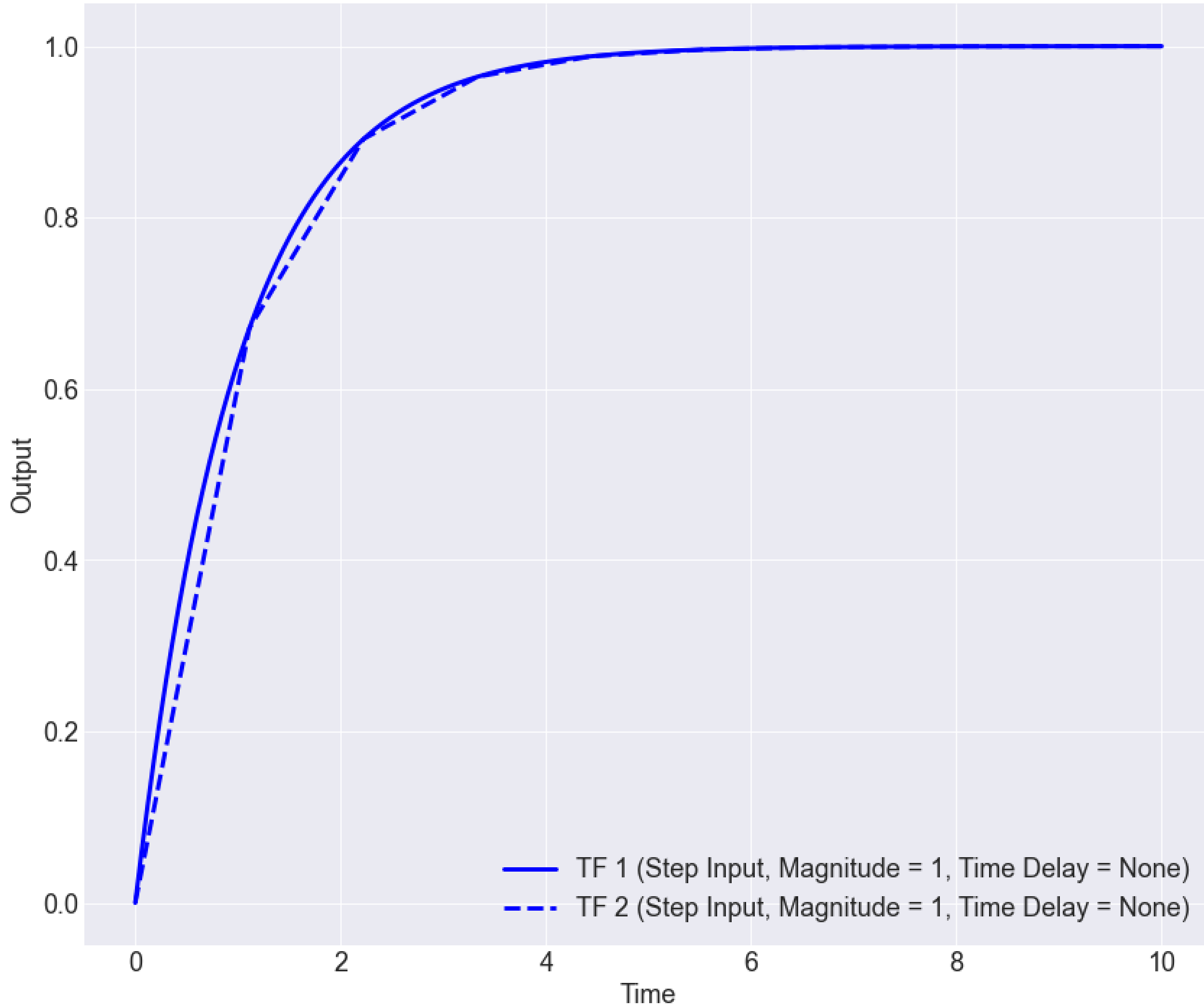
In [49]:
```
sys_less_points = TransferFunction(Numerator=1, Denominator=[1, 1],\
                       TFinal=10, NumPoints=10)

sys_less_points.InputFunction(Magnitude = 1, Type = 'Step')
```

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

These systems will have the same results, just with a different number of points

In [50]:
```
CompareResults(sys_more_points, sys_less_points)
```





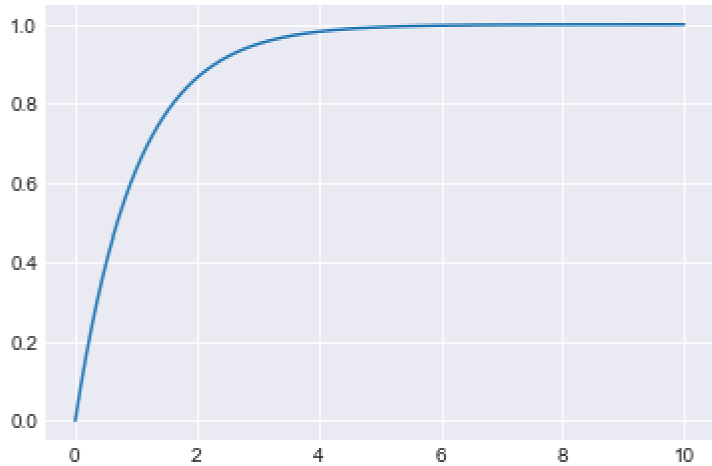
Step 3: Manually plot response

In [51]:
```
sys = TransferFunction(Numerator=1, Denominator=[1, 1])

sys.InputFunction(Magnitude = 1, Type = 'Step')
```

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

In [52]:
```
sim_time = sys.Time

sim_response = sys.Output
```

In [53]:
```
import matplotlib.pyplot as plt

plt.figure()

plt.plot(sim_time, sim_response)

plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10