In [2]:

```
pip install control
```

```
Collecting control
  Downloading control-0.8.3.tar.gz (249 kB)
     |████████████████████████████████| 249 kB 3.5 MB/s eta 0:00:01
Requirement already satisfied: numpy in /srv/conda/envs/notebook/lib/pytho
n3.6/site-packages (from control) (1.19.1)
Requirement already satisfied: scipy in /srv/conda/envs/notebook/lib/pytho
n3.6/site-packages (from control) (1.5.2)
Requirement already satisfied: matplotlib in /srv/conda/envs/notebook/lib/
python3.6/site-packages (from control) (3.3.2)
Requirement already satisfied: certifi>=2020.06.20 in /srv/conda/envs/note
book/lib/python3.6/site-packages (from matplotlib->control) (2020.6.20)
Requirement already satisfied: pillow>=6.2.0 in /srv/conda/envs/notebook/l
ib/python3.6/site-packages (from matplotlib->control) (7.2.0)
Requirement already satisfied: python-dateutil>=2.1 in /srv/conda/envs/not
ebook/lib/python3.6/site-packages (from matplotlib->control) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/srv/conda/envs/notebook/lib/python3.6/site-packages (from matplotlib->con
trol) (2.4.7)
Requirement already satisfied: cycler>=0.10 in /srv/conda/envs/notebook/li
b/python3.6/site-packages/cycler-0.10.0-py3.6.egg (from matplotlib->contro
l) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /srv/conda/envs/notebo
ok/lib/python3.6/site-packages (from matplotlib->control) (1.2.0)
Requirement already satisfied: six>=1.5 in /srv/conda/envs/notebook/lib/py
thon3.6/site-packages (from python-dateutil>=2.1->matplotlib->control) (1.
15.0)
Building wheels for collected packages: control
  Building wheel for control (setup.py) ... done
  Created wheel for control: filename=control-0.8.3-py2.py3-none-any.whl s
ize=260982 sha256=cfa4ef004892dea27d11141296e8fda99417967d9e029843510004d9
ed258d25
  Stored in directory: /home/jovyan/.cache/pip/wheels/38/cd/b8/9f67ad46525
980cf57dc48bb98876aeab3b5771c342a213d4a
Successfully built control
Installing collected packages: control
Successfully installed control-0.8.3
Note: you may need to restart the kernel to use updated packages.
```

Importing necessary libraries

In [3]:

```
import control as co
import matplotlib.pyplot as plt
import numpy as np
```

In [4]:

```
def get_numerator():
    num = []
    while True:
        n = input("Enter the coefficients for the numerator polynomial and 'done' to ex
it : ")

        try:
            n = float(n)
            num.append(n)
        except:
            if n == "done":
                break
    return num

def get_denominator():
    den = []
    while True:
        n = input("Enter the coefficients for the denominator polynomial and 'done' to
 exit : ")

        try:
            n = float(n)
            den.append(n)
        except:
            if n == "done":
                break
    return den
```

In [5]:

```
numerator  = get_numerator()
print(numerator)
```

```
Enter the coefficients for the numerator polynomial and 'done' to exit : 1
Enter the coefficients for the numerator polynomial and 'done' to exit : 5
Enter the coefficients for the numerator polynomial and 'done' to exit : 5
Enter the coefficients for the numerator polynomial and 'done' to exit : d
one
[1.0, 5.0, 5.0]
```

In [6]:

```
denominator = get_denominator()
print(denominator)
```

```
Enter the coefficients for the denominator polynomial and 'done' to exit :
1
Enter the coefficients for the denominator polynomial and 'done' to exit :
1.65
Enter the coefficients for the denominator polynomial and 'done' to exit :
5
Enter the coefficients for the denominator polynomial and 'done' to exit :
6.5
Enter the coefficients for the denominator polynomial and 'done' to exit :
2
Enter the coefficients for the denominator polynomial and 'done' to exit :
done
[1.0, 1.65, 5.0, 6.5, 2.0]
```

In [7]:

```
system = co.tf(numerator,denominator)
print("The transfer function is : \n",system)
```

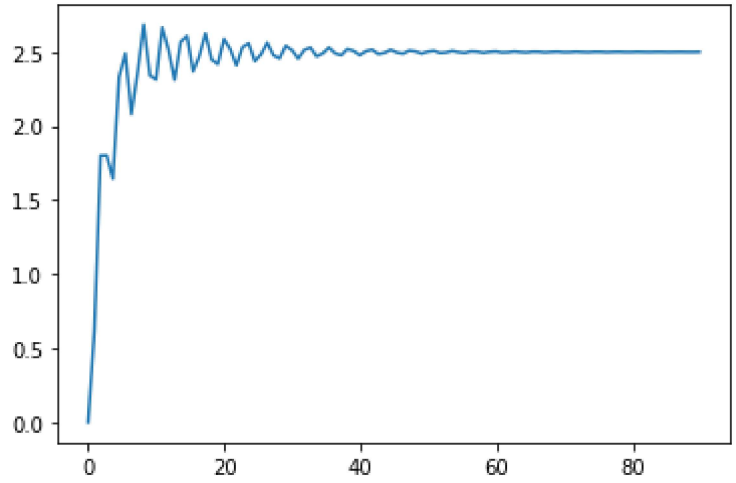
```
The transfer function is :

          s^2 + 5 s + 5
---------------------------------
s^4 + 1.65 s^3 + 5 s^2 + 6.5 s + 2
```

In [8]:

```
[t,y] = co.step_response(system)
plt.plot(t,y)
```

Out[8]:

```
[<matplotlib.lines.Line2D at 0x7f5a52e0c978>]
```



Peak Time

In [14]:

```
print('Maximum value of y : ', end = '')
maximum_value = max(abs(y))
print(round(maximum_value,2))

index_value = np.where(y == maximum_value)

T_p = t[index_value]

print("Peak time : ",round(T_p[0] , 2), "s" )
overshoot = abs(maximum_value - y[-1] / y[-1] * 100)
print("Overshoot ",round(overshoot,2),'%')
```

```
Maximum value of y : 2.68
Peak time :  8.16 s
Overshoot  97.32 %
```

Settling Time

In [15]:

```
for i in range(2, len(y)):
    #print(i)
    if abs(y[-i] / y[-1]) > 1.02:

        T_s = t[(len(y) -i)] - t[0]
        break


print("settling Time : {} s".format(round(T_s,2)))
```

```
settling Time : 26.29 s
```

Rise Time

In [22]:

```
for i in range(0, len(y)):
    if y[i]> y[-1] * 0.95:
        second = t[i] - t[0]
        break

for i in range(0, len(y)):
    if y[i]> y[-1] * 0.05:
        first = t[i] - t[0]
        break

T_r = second - first
print("Rise Time : ",round(T_r,2))
```

```
Rise Time :  4.53
```

By :

Ishaan Kharbanda

UE179039

Matlab Assignment 2

1. M- File :

Editor - C:\MATLAB\bin\Ishaan_Kharbanda_assig2.m

ISHAAN_KHARBANDA_UE179039_assignment2.m | Ishaan_Kharbanda_assig2.m

```
zeta = input(' Enter Damping Ratio : ');
natural_freq = input('Enter Natural Frequency : ');

N = [natural_freq^2];
D = [1 2*zeta*natural_freq natural_freq^2];

system = tf(N,D)

step(system)
stepinfo(system)
```

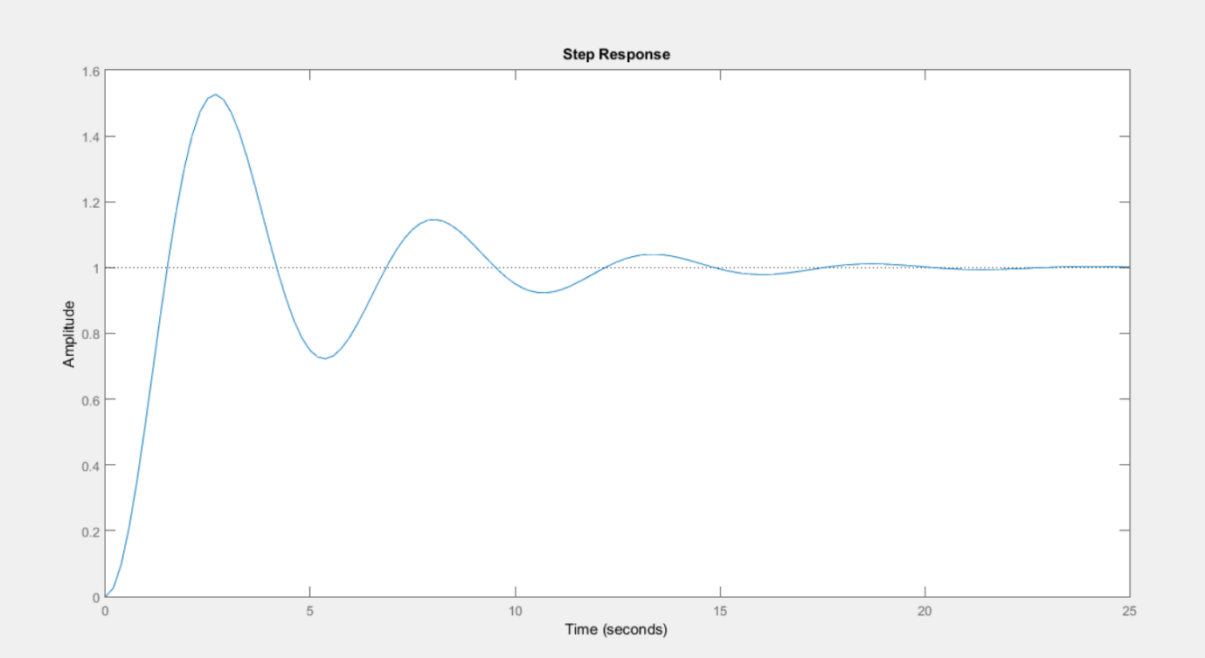
2. Output :
   a. MATLAB Window:

Command Window

New to MATLAB? See resources for Getting Started.

```
>> Ishaan_Kharbanda_assig2
 Enter Damping Ratio : 0.2
Enter Natural Frequency : 1.2

system =

         1.44
  -------------------
  s^2 + 0.48 s + 1.44

Continuous-time transfer function.


ans =

        RiseTime: 1.0047
    SettlingTime: 16.3301
     SettlingMin: 0.7228
     SettlingMax: 1.5265
       Overshoot: 52.6542
      Undershoot: 0
            Peak: 1.5265
        PeakTime: 2.6863
```

b. Graph:


Step Response
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0
5
10
15
20
25