Pset 10

Casey Alvarado

1.

$$\begin{tikzcd}
\underset{u(t)}{\text{step response:}} \arrow[r] \arrow[d, "\mathcal{L}"] & \underset{\text{impulse response: } h(t)=e^{-t}u(t)}{\boxed{\dot{y}+y=x}}^{\,h(t):} \arrow[r] & (1-e^{-t})u(t) \\
\frac{1}{s} \arrow[r] & \underset{H(s):}{\boxed{\frac{1}{s+1}}} \arrow[r] & \frac{1}{s\cdot(s+1)} \arrow[u, "\mathcal{L}^{-1}"]
\end{tikzcd}$$

If the impulse response is $h(t)=e^{-t}u(t)$, then w/ the Laplacian property we can get $H(s)=\frac{1}{s+a}$ where $a=1$

$$H(s)=\frac{1}{s+1}$$

$$\dot{y}+y=x \xrightarrow{\mathcal{L}} sH(s)+H(s)=X(s)$$

$$Y(s)=H(s)\cdot X(s)$$

$$Y(s)=\left(\frac{1}{s+1}\right)\cdot\left(\frac{1}{s}\right)=\frac{1}{s(s+1)}=\frac{A}{s}+\frac{B}{s+1}$$

$$\frac{s(s+1)}{s(s+1)}=A(s+1)+Bs=1=As+A+Bs=1$$

$$A+B=0 \qquad A=1$$

$$(1)+B=0$$

$$B=-1$$

$$Y(s)=\frac{A}{s}+\frac{B}{s+1}=\frac{1}{s}-\frac{1}{s+1}\rightarrow \mathcal{L}^{-1}\{Y(s)\}=u(t)-e^{-t}u(t)=y(t)$$

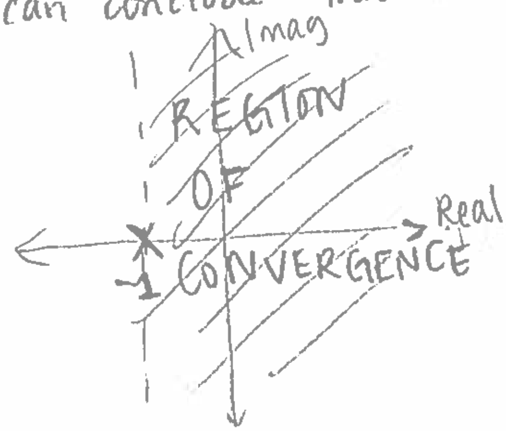
$$y(t)=u(t)(1-e^{-t})$$

$\boxed{\text{Finding the Regions of Convergence}}$

$$H(s)=\frac{1}{s+1}\rightarrow s+1=0\rightarrow s=-1$$

we can conclude that the Region of Convergence exists when $s>-1$.

$Re\{s\}>-1$



2A. $\dfrac{Y(s)}{Y_{sp}(s)} = \dfrac{K(s)H(s)}{1+K(s)H(s)}$, where $K(s) = \dfrac{K_I}{s}$ for any $H(s)$

$$\frac{Y(s)}{Y_{sp}(s)} = \frac{\frac{K_I}{s}H(s)}{1+\frac{K_I}{s}H(s)}$$

DC Gain $= \lim\limits_{s \to 0} \dfrac{Y(s)}{Y_{sp}(s)} = \lim\limits_{s \to 0} \dfrac{\frac{K_I}{s}H(s)}{1+\frac{K_I}{s}H(s)} = \dfrac{\infty}{\infty} = 1$

Since the DC Gain is 1, it does not depend on the value of K.

2B. $\dfrac{Y(s)}{Y_{sp}(s)} = \dfrac{K(s)H(s)}{1+K(s)H(s)}$ where $K(s) \overset{?}{=} \dfrac{K_I}{s}$ and $H(s) = \dfrac{1/J}{s+1/J}$

$$\frac{Y(s)}{Y_{sp}(s)} = \frac{\left(\frac{K_I}{s}\right)\left(\frac{1/J}{s+1/J}\right)}{1+\left(\frac{K_I}{s}\right)\left(\frac{1/J}{s+1/J}\right)} = \frac{\frac{\frac{K_I}{J}}{s(s+\frac{1}{J})}}{1+\frac{\frac{K_I}{J}}{s(s+\frac{1}{J})}} = \frac{\frac{\frac{K_I}{J}}{\cancel{s(s+\frac{1}{J})}}}{\frac{s(s+\frac{1}{J})+\frac{K_I}{J}}{\cancel{s(s+\frac{1}{J})}}} \times \frac{\cancel{s(s+\frac{1}{J})}}{s(s+\frac{1}{J})+\frac{K_I}{J}} \times \frac{\cancel{s(s+\frac{1}{J})}}{s(s+\frac{1}{J})+\frac{K_I}{J}}$$

$$\frac{Y(s)}{Y_{sp}(s)} = \frac{\frac{K_I}{J}}{s(s+\frac{1}{J})+\frac{K_I}{J}}$$

$$\frac{Y(s)}{Y_{sp}(s)} = \frac{\frac{K_I}{J}}{s^2+\frac{s}{J}+\frac{K_I}{J}}$$

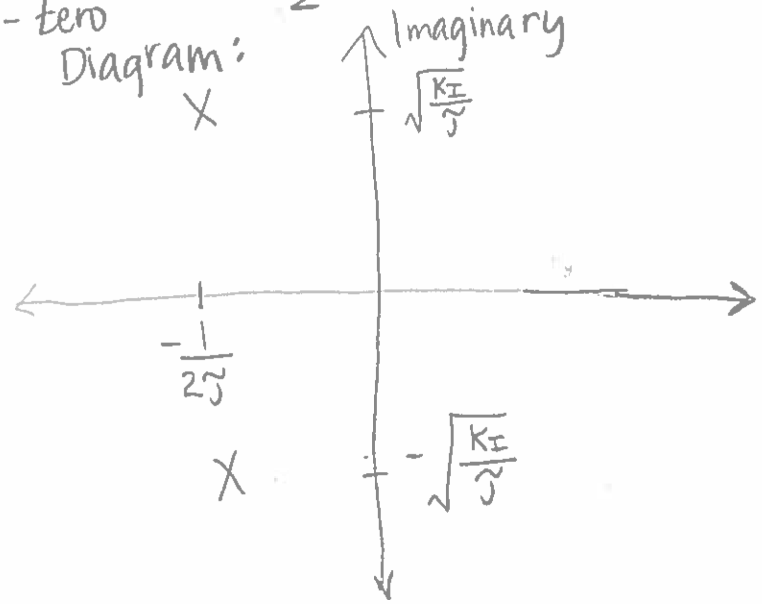
Finding the poles

$$s^2+\frac{s}{J}+\frac{K_I}{J} = 0$$

$$s = \frac{-\left(\frac{1}{J}\right) \pm \sqrt{\left(\frac{1}{J}\right)^2 - \frac{4K_I}{J}}}{2}$$

if $K \gg \frac{1}{J} \rightarrow s = \dfrac{-\frac{1}{J} \pm \sqrt{-4\frac{K_I}{J}}}{2} = \dfrac{-1}{2J} \pm j\sqrt{\dfrac{K_I}{J}}$

Pole-Zero Diagram:



This is a stable system because it includes the imaginary plane.

ThinkDSP (/github/CaseyAlvarado/ThinkDSP/tree/master)
/ ps10.ipynb (/github/CaseyAlvarado/ThinkDSP/tree/master/ps10.ipynb)

In [6]:

```
%matplotlib inline
%run convenience.ipynb
from __future__ import print_function
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

np.set_printoptions(precision=2,suppress=True) # numpy output options

pi=np.pi
j=1j
```

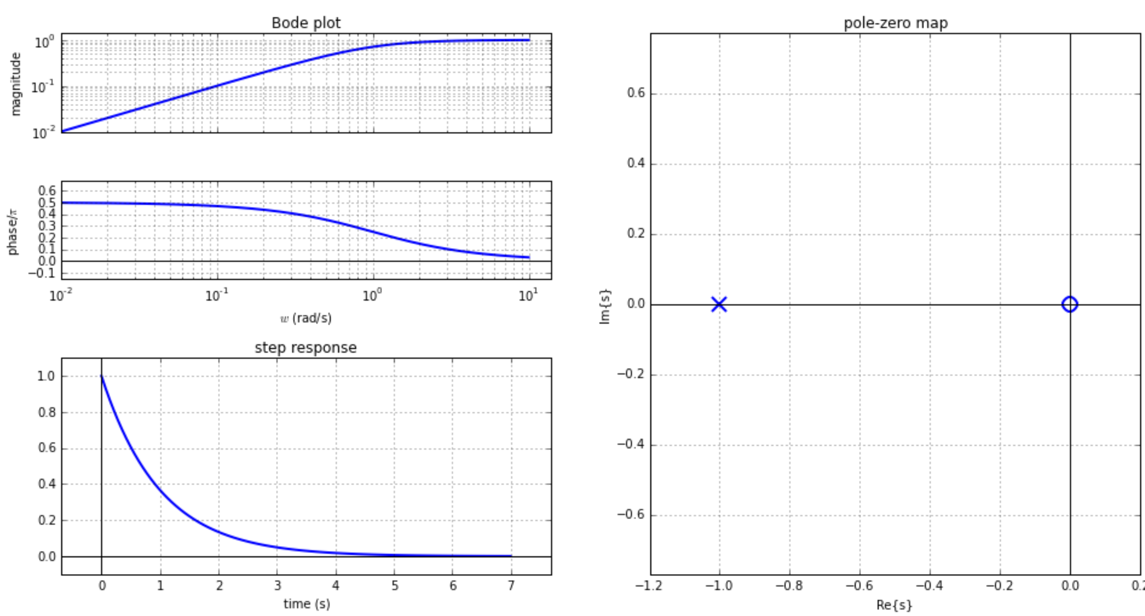
PROBLEM 3:

Problem 3a.

In [7]:

```
#H = s/(s+1)
combinedplot(signal.lti([1,0], [1,1]))
```

```
/home/casey/anaconda/lib/python2.7/site-packages/matplotlib/axes/_axes
.py:475: UserWarning: No labelled objects found. Use label='...' kwarg
 on individual plots.
  warnings.warn("No labelled objects found. "
```

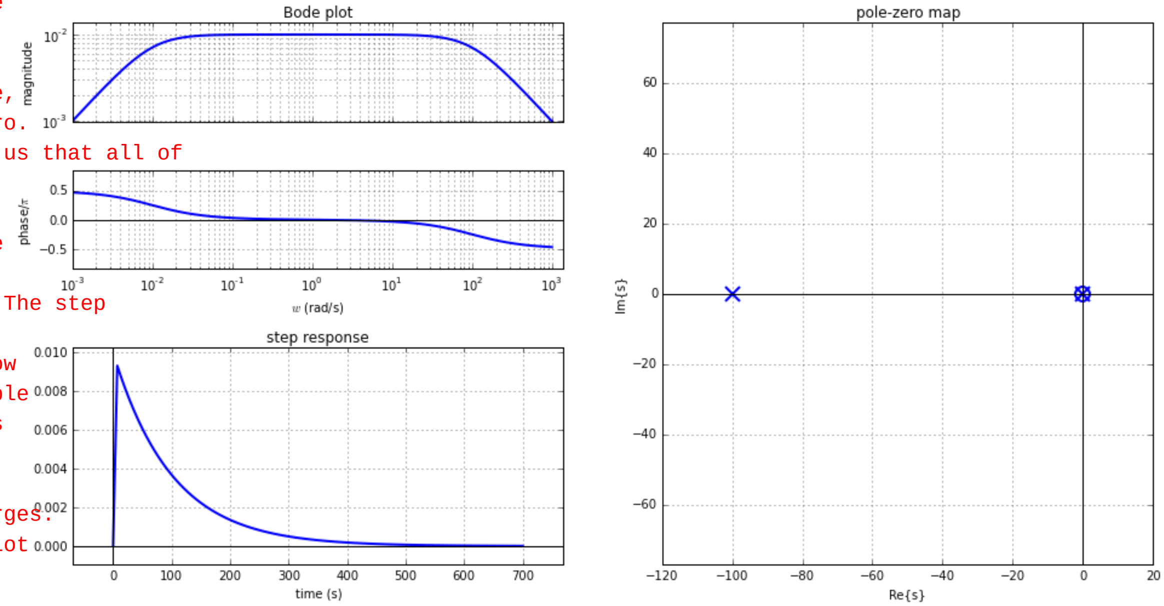


This is a first order system, therefore there will only be one pole and one zero. There is one real pole that includes the imaginary axis. Because it includes the imaginary axis, this problem could have been done with the Fourier transform and the system will converge. The step response proves that the step response converges to 0. This means this system is stable. The bode plot provides us with an idea of how the system will act across all frequencies. It seems to squish all frequencies in the high negative frequency range.

In [8]:

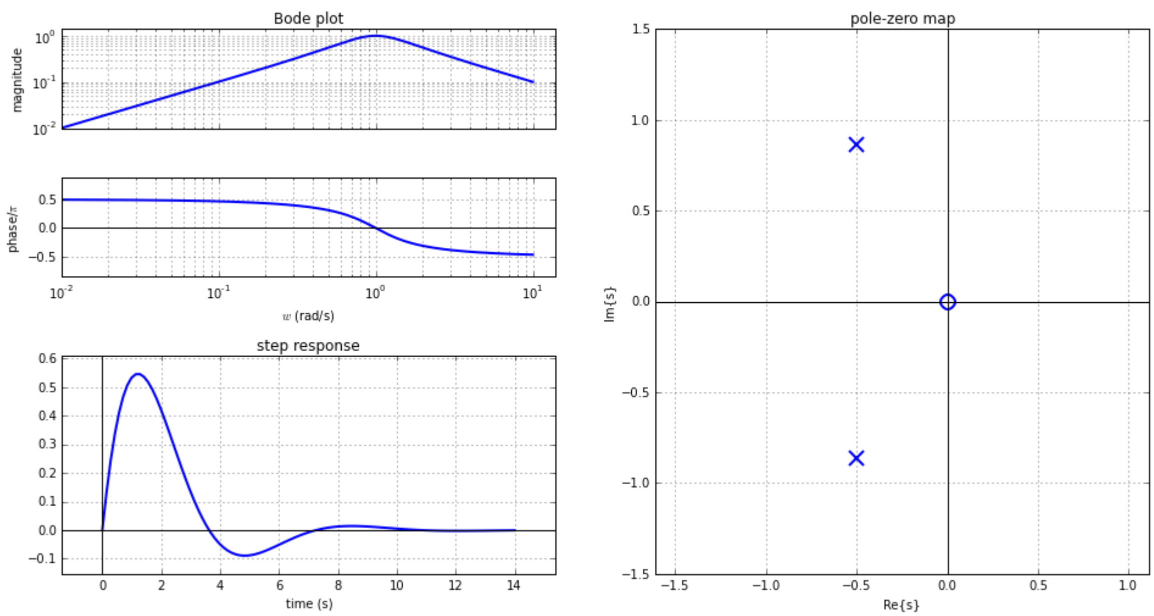
```
# H = s/(s^2+100s+1)
combinedplot(signal.lti([1, 0], [1,100,1]))
```

The denominator is a quadratic, therefore, we should expect to see 2 poles. The numerator is a first order, therefore, we expect to see one zero. The Pole-Zero map shows us that all of the poles are on the left side of the imaginary axis. They are all real poles without an imaginary component. The step response converges to a very small value. We know this system will be stable because all of the poles are on the left side of the imaginary axis and the step response converges. Furthermore, the bode plot shows up that this is a low pass filter.



In [9]:

```
# H = s/(s^2 + s + 1)
combinedplot(signal.lti([1, 0], [1,1,1]))
```

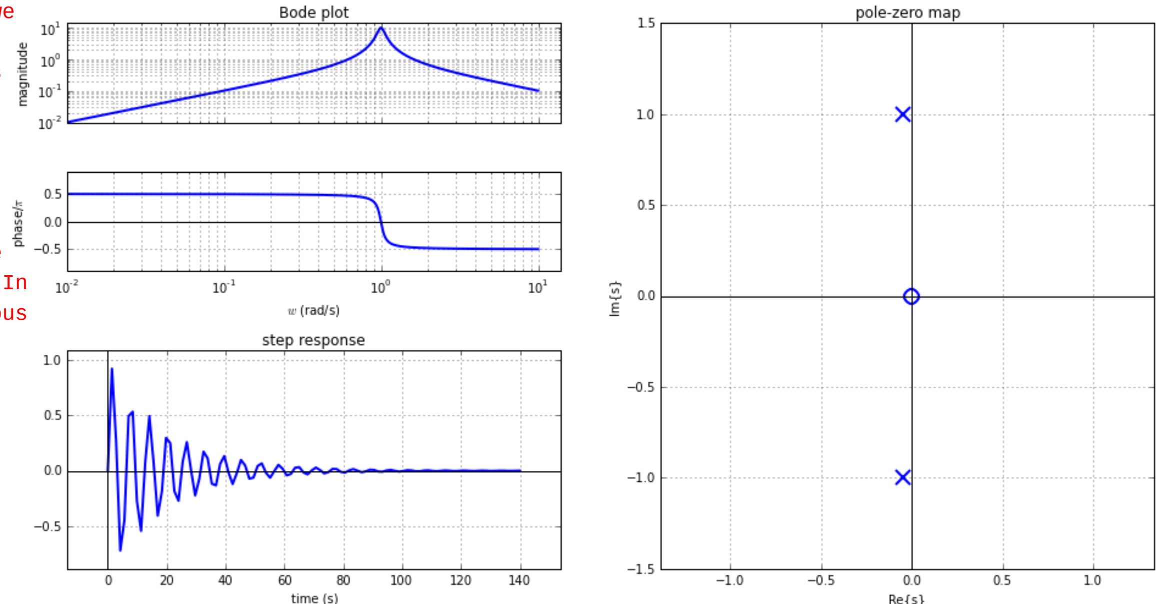


The numerator is a first order, therefore, we expect to see one zero at 0. The denominator is a quadratic, therefore, we expect to see two poles. These two poles are both on the left side of the imaginary axis and they have an imaginary component, making it an imaginary pole. The step response has some oscillation. The oscillations start off high in amplitude and then through some intense dampening, converge to 0. If we squint really hard, the bode plot looks similar to a band pass filter with a 3 order of magnitude band width.

In [10]:

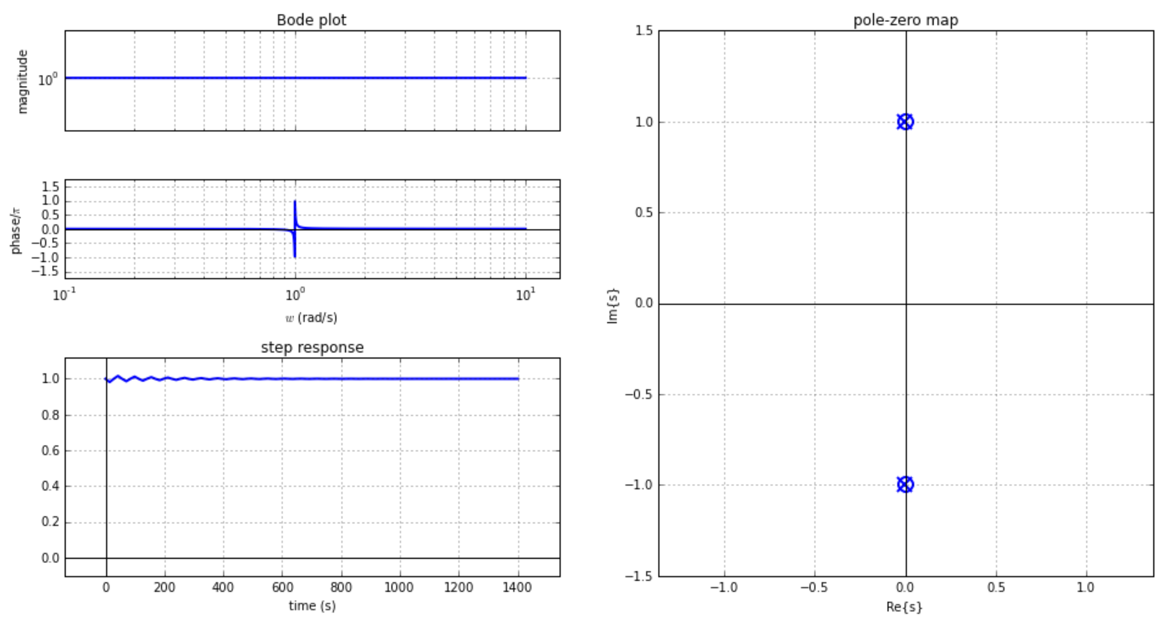
```
# H = s/(s^2 + 0.1s + 1)
combinedplot(signal.lti([1, 0], [1,0.1,1]))
```

Again, the numerator is a first order without a constant, therefore, we expect the zero to be at 0. The denominator is a quadratic, so the pole-zero maps shows two poles. These poles are both barely on the left side of the imaginary axis. They are mainly imaginary poles. In comparison to the previous problem, notice that as our poles approach the imaginary axis,the step response results in more oscillations before converging to 0. This tells us that the system will eventually reach stability but there will be some oscillation first. The Bode plot looks like a sharper version of the previous problem's bode plot.



In [11]:

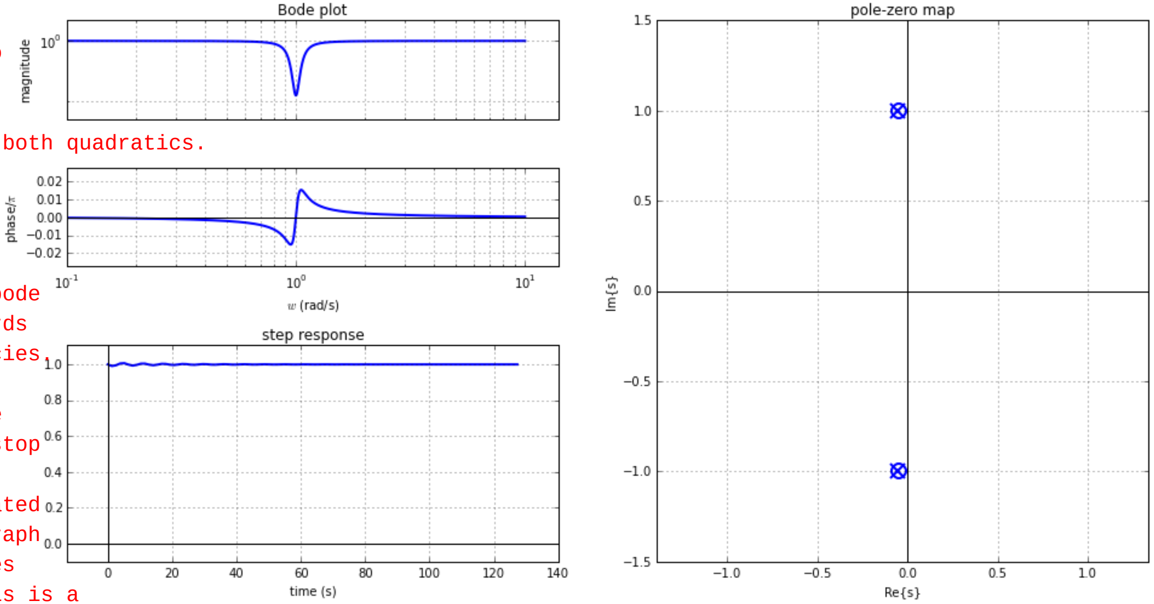
```
# H = (s^2-0.01s+1)/(s^2 + 0.01s + 1)
combinedplot(signal.lti([1,-0.01,1], [1,0.01,1]))
```



The numerator and the denominator are similar quadratics with a sign change in front of the b term. We expect to see two poles and two zeros that overlap. The poles and zeros are all imaginary. This provides the step response with a tiny oscillations, but then it does converge to 1, making the DC Gain 1. The magnitude Bode Plot is completely flat and the phase bode plot is is also flat until is peaks around 0. This system will likely be stable, but it is not the best system.

In [12]:

```
# H = (s^2+0.1s+1)/(s^2 + 0.11s + 1)
combinedplot(signal.lti([1,0.1,1], [1,0.11,1]))
```



The poles are close to the imaginary axis with a small real component. There are two poles and two zeros because the numerator and the denominator are both quadratics. The step response has some tiny oscillations up to 10 seconds, but it does remain constant around 1. The bode plot has a sharp downwards peak around low frequencies. If you squint, the magnitude portion of the plot looks like a band stop that kills all low frequencies in a designated band width. The phase graph of the bode plot switches angle at the origin. This is a somewhat stable system because both of the poles are on the left side of the imaginary axis and the step response does converge to a value. However, it is not comfortably stable.

4A. $H(s) = \dfrac{1}{s^2 - 0.01s + 1}$

Step Response:

$$Y(s) = H(s) \cdot X(s) = \left[\frac{1}{s^2 - 0.01s + 1}\right]\left[\frac{1}{s}\right] = \frac{1}{s(s^2 - 0.01s + 1)}$$

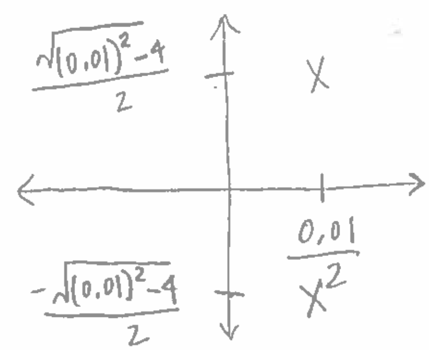
Step Response Plot included on next page

Pole Response:

$$s^2 - 0.01s + 1 = 0$$

$$s = \frac{-(-0.01)}{2} \pm \frac{\sqrt{(-0.01)^2 - 4(1)(1)}}{2} = \frac{0.01}{2} \pm \frac{\sqrt{(0.01)^2 - 4}}{2} j$$

Pole-Zero Map



(Also included graphically on next page).

4B. Proportional Control means $K(s) = K$

$$H'(s) = \frac{K(s)H(s)}{1 + K(s)H(s)} = \frac{K\left(\frac{1}{s^2 - 0.01s + 1}\right)}{1 + K\left(\frac{1}{s^2 - 0.01s + 1}\right)}$$

$$H'(s) = \frac{\frac{K}{\cancel{s^2 - 0.01s + 1}}}{\frac{s^2 - 0.01s + 1 + K}{\cancel{s^2 - 0.01s + 1}}} \times \frac{s^2 - 0.01s + 1}{s^2 - 0.01s + 1 + K} \quad \frac{K}{s^2 - 0.01s + 1 + K}$$

Step Response:

$$Y(s) = H'(s) \cdot X(s) = \left[\frac{K}{s^2 - 0.01s + 1 + K}\right]\left[\frac{1}{s}\right] = \frac{K}{s^3 - 0.01s^2 + (1+K)s}$$

Plot done on graph attached on next page

Pole Zero:

$$s^2 - 0.01s + 1 + K = 0 \rightarrow s = \frac{-(-0.01)}{2} \pm \frac{\sqrt{(-0.01)^2 - 4(1)(1+K)}}{2}$$

Pole-Zero Map plotted with Python. Done on seperate page attached. Done for multiple k, feedback gain, values. The system cannot be stabilized because if we evaluate the pole zero map plot on the next page, none of the poles are ever on the left side of the imaginary axis. This is important because if they are on the left side, then the imaginary axis is included and

the function converges. I ~~attempte~~ used multiple values of K and none of these gains provide a function with poles on the left side of the imaginary axis.
Also, if we evaluate the step response plots for all of these K gain values, we notice that none of these step responses converge. This means that the system is never stable, no matter the K gain value.

4C. Integral Control means $K(s) = \frac{K}{s}$

$$H''(s) = \frac{K(s)H(s)}{1 + K(s)H(s)} = \frac{\left(\frac{K}{s}\right)\left(\frac{1}{s^2-0.01s+1}\right)}{1 + \left(\frac{K}{s}\right)\left(\frac{1}{s^2-0.01s+1}\right)} =$$

$$H''(s) = \frac{\frac{K}{\cancel{s(s^2-0.01s+1)}}}{\frac{s(s^2-0.01s+1)+K}{s(s^2-0.01s+1)}} \times \frac{\cancel{s(s^2-0.01s+1)}}{s(s^2-0.01s+1)+K} \times \frac{s(s^2-0.01s+1)}{s(s^2-0.01s+1)+K} = \frac{K}{s^3-0.01s^2+s+K}$$

~~Pole~~ I do not really want to find the poles of a cubic function, so I just plugged it into Python and plotted the pole-zero map that way.
As for the step response, $Y(s) = H(s)X(s)$

$$Y(s) = \left(\frac{K}{s^3-0.01s^2+s+K}\right)\left(\frac{1}{s}\right) = \frac{K}{s^4-0.01s^3+s^2+Ks}$$

I also plugged this into python to produce a step response graph. We can see from evaluating the pole-zero map and the step response, that all of the poles are never all on the left side of the imaginary axis and none of the step responses converge for any values of K, the feedback gain.

4D. Diffential Control means $K(s) = sK$

$$H'''(s) = \frac{K(s)\,H(s)}{1 + K(s)H(s)} = \frac{(sK)\left(\frac{1}{s^2-0.01s+1}\right)}{1 + (sK)\left(\frac{1}{s^2-0.01s+1}\right)} = \frac{\frac{sK}{\cancel{s^2-0.01s+1}} \times \frac{\cancel{s^2-0.01s+1}}{s^2-0.01s+1+k}}{\frac{s^2-0.01s+1+k}{s^2-0.01s+1} \times \frac{s^2-0.01s+1}{s^2-0.01s+1+sK}}$$

$$H'''(s) = \frac{sK}{s^2-0.01s+1+K}$$

I put this into Python and plotted the pole-zero map and the step response. We can see from evaluating the pole-zero map and the step response, that this system converges. If $K>0$, all of the poles are on the left side of the imaginary axis. This system definitely converges to 0 when $0<K<3$. Beyond ~~this~~ these values of k, the step response has a negative slope on the line, so this step response might converge at higher frequencies than the ones displayed on my x-axis. This behavior can be expected for values of k: $3<k<100$. Beyond this region ($k>100$), it looks like the system will never converge and therefore, never reach stability.

The step response by hand:

$$Y(s) = X(s)\cdot H(s) = \frac{1}{\cancel{s}} \cdot \left(\frac{\cancel{s}K}{s^2-0.01s+1+k}\right) = \frac{K}{s^2-0.01s+1+K}$$

Step Response plotted in Python, explanation above.

Pole-zeros by hand:

$$s^2-0.01s+1+K=0$$

$$s = \frac{-(-0.01)}{2} \pm \frac{\sqrt{(-0.01)^2-4(1)(1+k)}}{2}$$

Pole-zero map for multiple values of k plotted in Python. Explanation above.

Problem 4a.

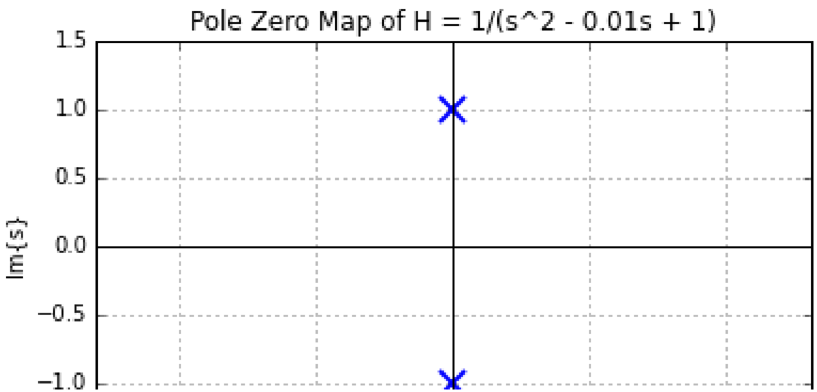
In [13]:

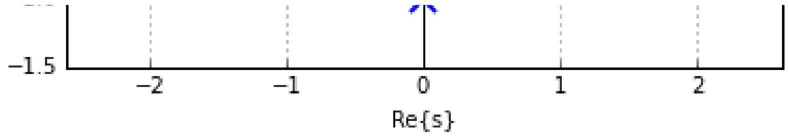
```
sys = signal.lti([0, 1], [1,-.01,1])
plt.subplot(2,1,1)
plt.subplots_adjust(hspace=1, bottom = 3, top = 5)
pzmap(sys)
plt.title('Pole Zero Map of H = 1/(s^2 - 0.01s + 1)')

plt.subplot(2,1,2)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys,T=t)
timeplot(t,s)
plt.title('Step Response of H = 1/(s^2 - 0.01s + 1)')
```

```
/home/casey/anaconda/lib/python2.7/site-packages/scipy/signal/filter_d
esign.py:397: BadCoefficients: Badly conditioned filter coefficients (
numerator): the results may be meaningless
  "results may be meaningless", BadCoefficients)
```

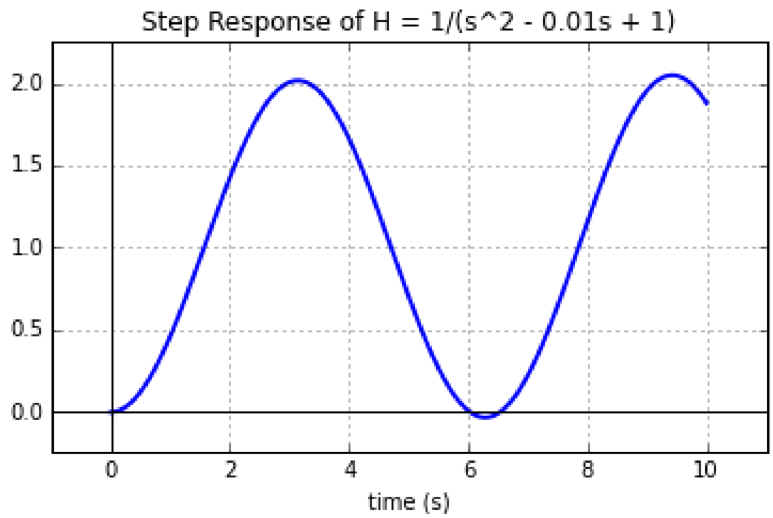
Out[13]:

```
<matplotlib.text.Text at 0x7f1b33eb1dd0>
```





This is not a stable system. Both of the poles are on the right side of the imaginary axis and the step response does not converge.



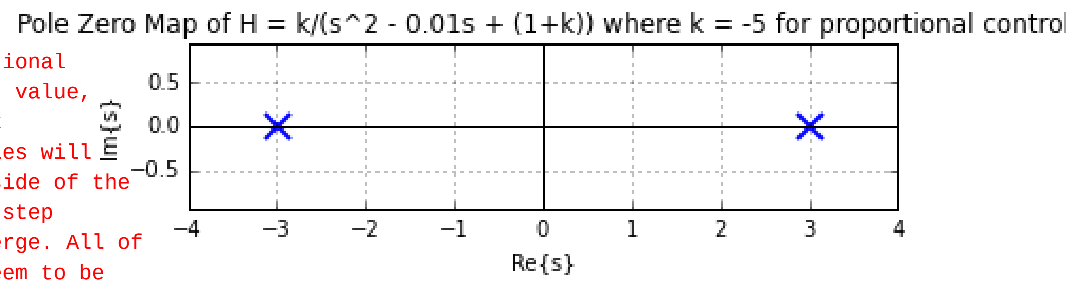
In [18]:

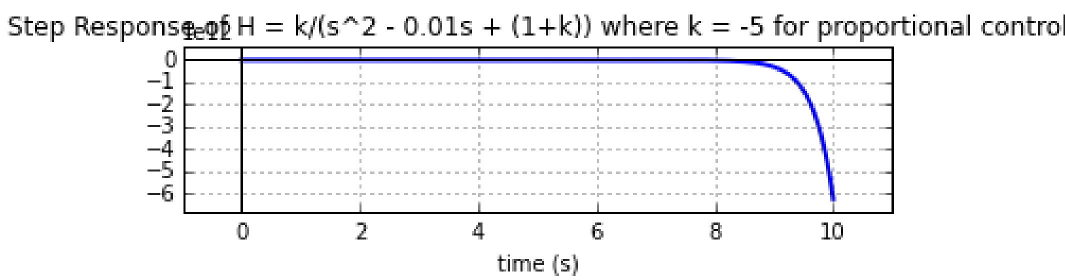
```
#proportional control
plt.subplots_adjust(hspace=1, bottom = 3, top = 6)
k = -10
sys_p1 = signal.lti([0,k], [1,-.01,1+k])
plt.subplot(6,1,1)
pzmap(sys_p1)
plt.title('Pole Zero Map of H = k/(s^2 - 0.01s + (1+k)) where k = -5 f
or proportional control')

plt.subplot(6,1,2)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_p1,T=t)
timeplot(t,s)
plt.title('Step Response of H = k/(s^2 - 0.01s + (1+k)) where k = -5 f
or proportional control')

k = -0.99999
sys_p2 = signal.lti([0,k], [1,-.01,1+k])
plt.subplot(6,1,3)
pzmap(sys_p2)
plt.title('Pole Zero Map of H = k/(s^2 - 0.01s + (1+k)) where k = -0.9
999 for proportional control')

plt.subplot(6,1,4)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_p2,T=t)
```

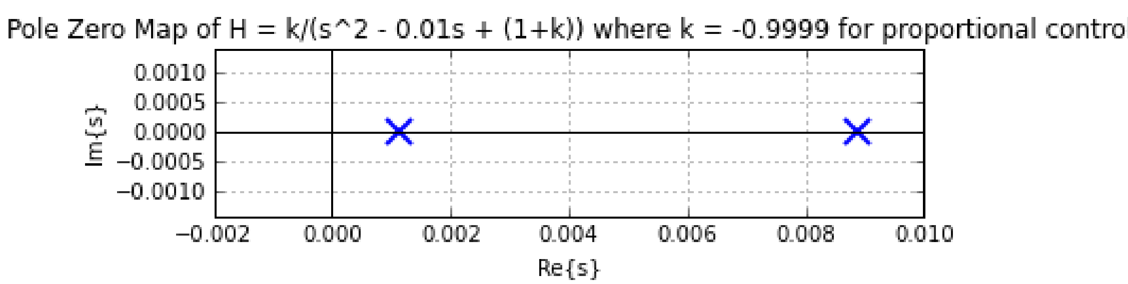
```
timeplot(t,s)
plt.title('Step Response of H = k/(s^2 - 0.01s + (1+k)) where k = -0.9
999 for proportional control')

k = 2000
sys_p3 = signal.lti([0,k], [1,-.01,1+k])
plt.subplot(6,1,5)
pzmap(sys_p3)
plt.title('Pole Zero Map of H = k/(s^2 - 0.01s + (1+k)) where k = 2 fo
r proportional control')


plt.subplot(6,1,6)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_p3,T=t)
timeplot(t,s)
plt.title('Step Response of H = k/(s^2 - 0.01s + (1+k)) where k = 2 fo
r proportional control')
```

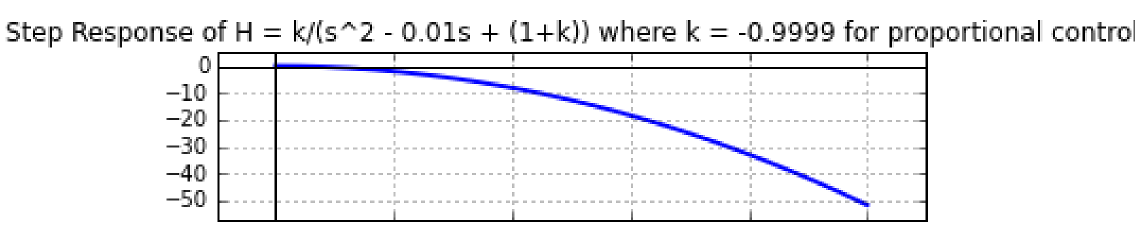
Out[18]: <matplotlib.text.Text at 0x7f1b3111fc10>

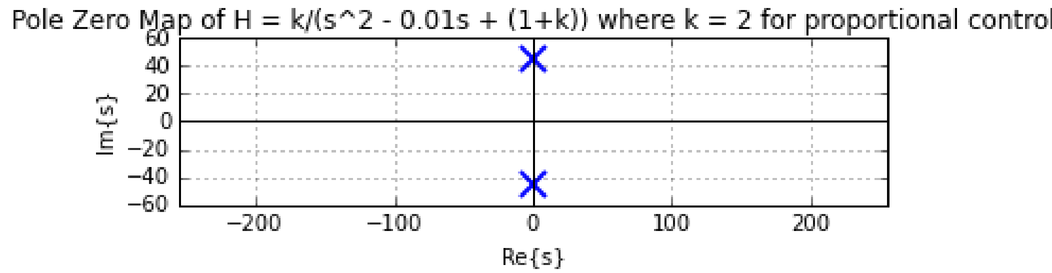
I cannot stabilize the system with proportional control. No matter what value, I left for the feedback gain, k, all of the poles will not exist on the left side of the imaginary axis and the step response will not converge. All of these step responses seem to be going off to infinity or in the last case oscillating wildly, never converging.

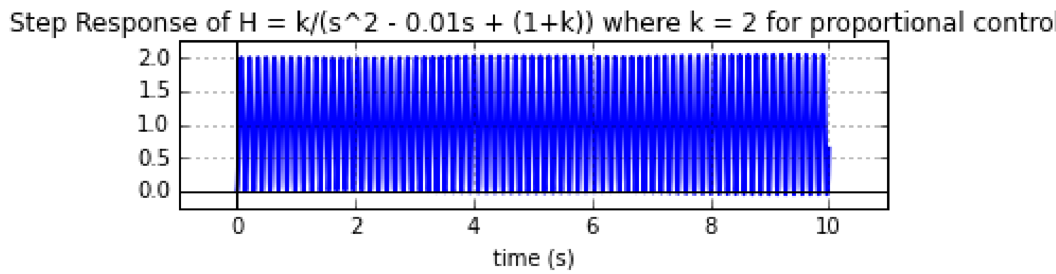












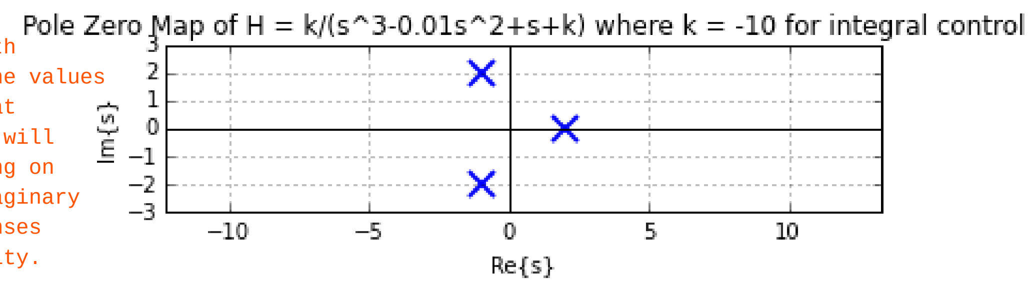
In [19]:

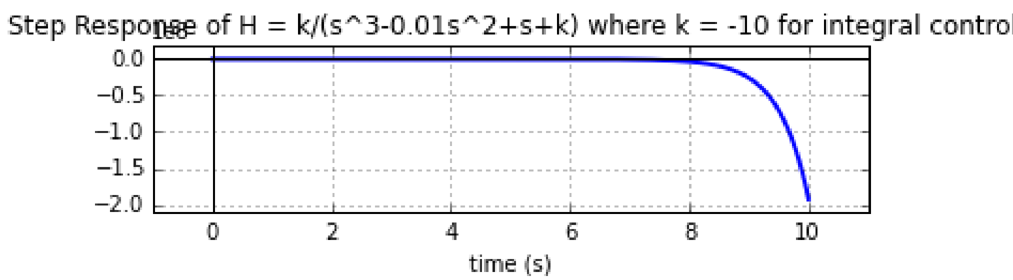
```
#now integral control
plt.subplots_adjust(hspace=1, bottom = 3, top = 6)
k = -10
sys_i1 = signal.lti([0,k], [1,-.01,1, k])
plt.subplot(6,1,1)
pzmap(sys_i1)
plt.title('Pole Zero Map of H = k/(s^3-0.01s^2+s+k) where k = -10 for
integral control')

plt.subplot(6,1,2)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_i1,T=t)
timeplot(t,s)
plt.title('Step Response of H = k/(s^3-0.01s^2+s+k) where k = -10 for
integral control')

k = 2
sys_i3 = signal.lti([0,k], [1,-.01,1, k])
plt.subplot(6,1,3)
pzmap(sys_i3)
plt.title('Pole Zero Map of H = k/(s^3 - 0.01s^2 + s + k) where k = 2
for integral control')

plt.subplot(6,1,4)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_i3,T=t)
timeplot(t,s)
plt.title('Step Response of H = k/(s^3-0.01s^2+s+k) where k = 2 for in
tegral control')

k = 2000
sys_i4 = signal.lti([0,k], [1,-.01,1, k])
```

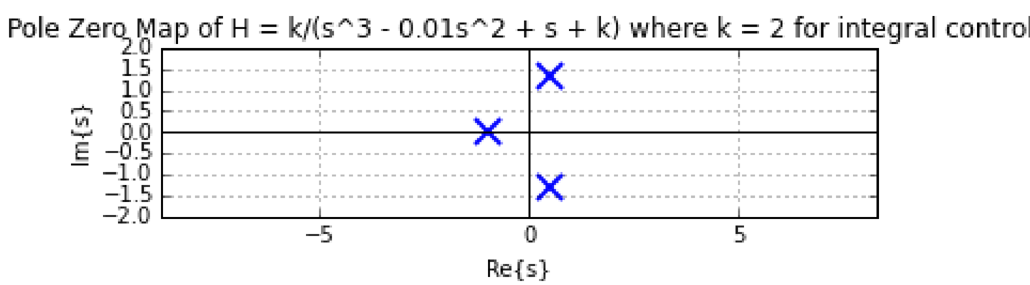
```
plt.subplot(6,1,5)
pzmap(sys_i4)
plt.title('Pole Zero Map of H = k/(s^3 - 0.01s^2 + s + k) where k = 20
00 for integral control')

plt.subplot(6,1,6)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_i4,T=t)
timeplot(t,s)
plt.title('Step Response of H = k/(s^3-0.01s^2+s+k) where k = 2000 for
 integral control')
```

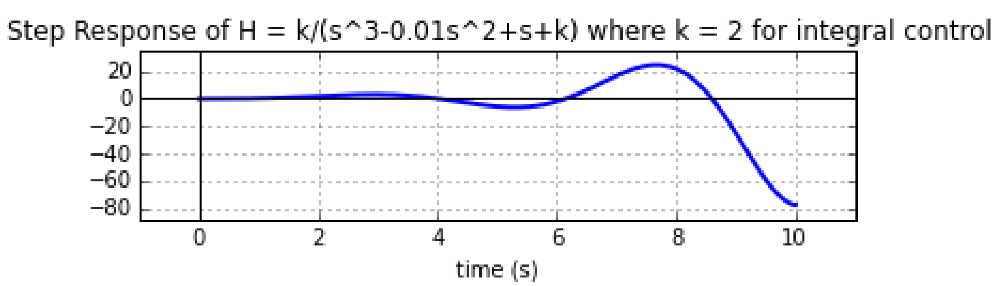
Out[19]: `<matplotlib.text.Text at 0x7f1b30ca8250>`

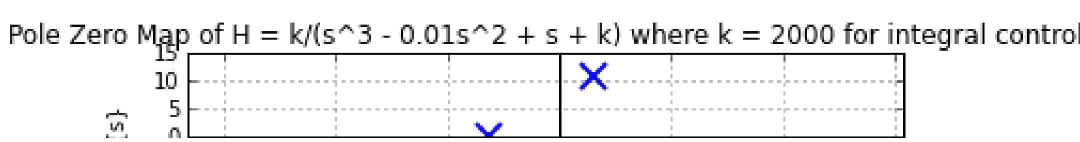
It seems I also cannot stabilize the system with integral control. For the values of feedback gain, k, that I tested, none of these will result in the poles being on the left side of the imaginary axis and the step responses all seem to go to infinity.

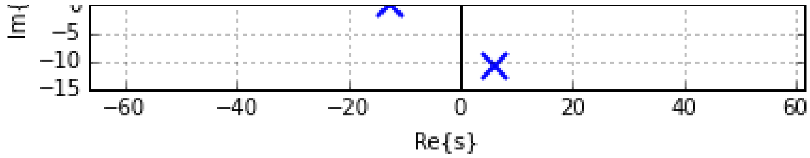


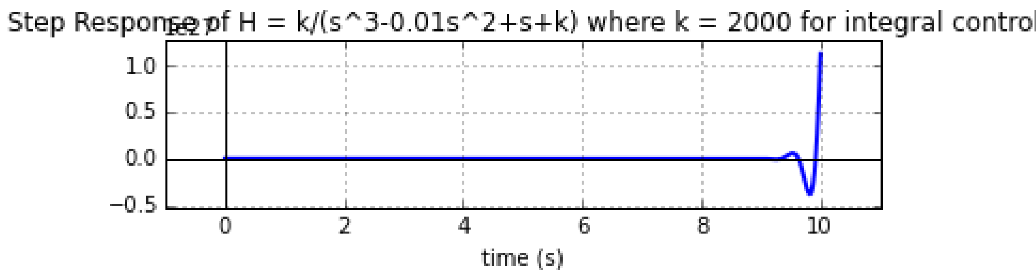












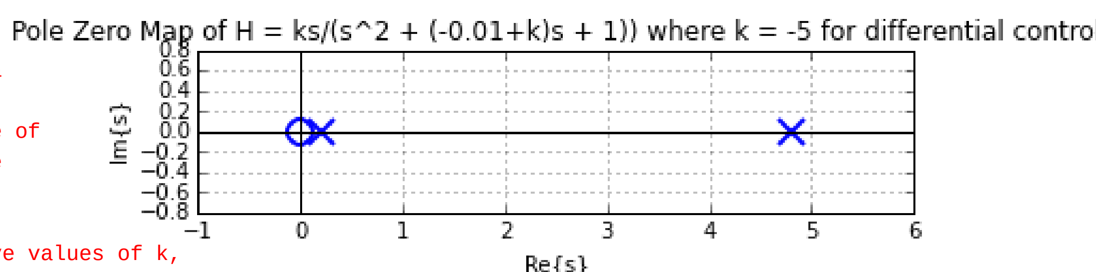
In [22]:

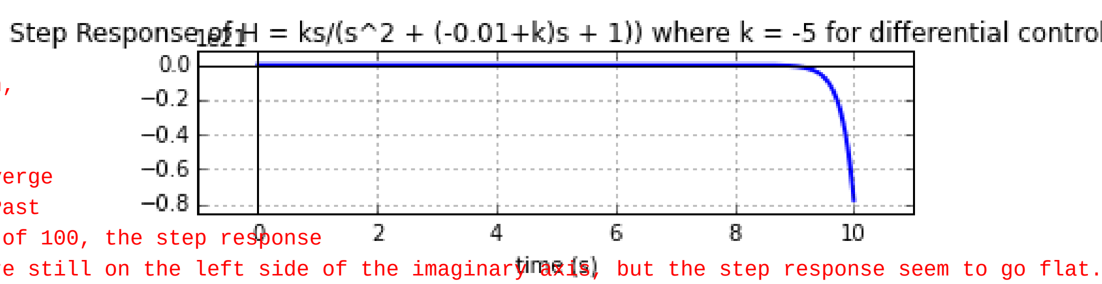
```
#now differential control
plt.subplots_adjust(hspace=1, bottom = 3, top = 8)
k = -5
sys_d1 = signal.lti([k, 0], [1,(-.01+k), 1])
plt.subplot(10,1,1)
pzmap(sys_d1)
plt.title('Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = -
5 for differential control')

plt.subplot(10,1,2)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_d1,T=t)
timeplot(t,s)
plt.title('Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = -
5 for differential control')

k = 0
sys_d2 = signal.lti([k, 0], [1,(-.01+k), 1])
plt.subplot(10,1,3)
pzmap(sys_d2)
plt.title('Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 0
 for differential control')

plt.subplot(10,1,4)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_d2,T=t)
timeplot(t,s)
plt.title('Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 0
 for differential control')

k = 1
sys_d3 = signal.lti([k, 0], [1,(-.01+k), 1])
plt.subplot(10,1,5)
pzmap(sys_d3)
plt.title('Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1
 for differential control')

plt.subplot(10,1,6)
```

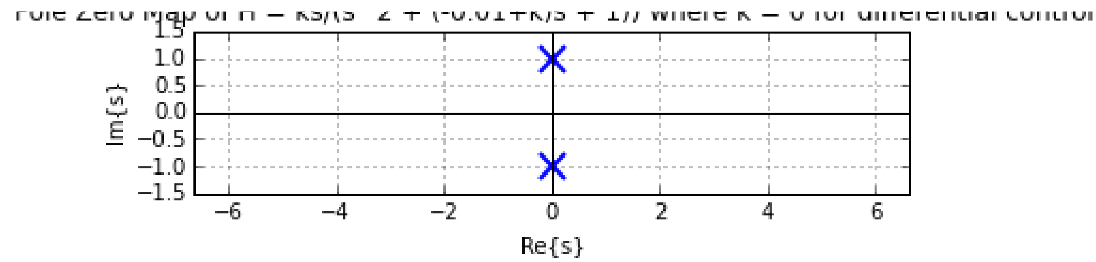
```
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_d3,T=t)
timeplot(t,s)
plt.title('Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1
 for differential control')

k = 15
sys_d4 = signal.lti([k, 0], [1,(-.01+k), 1])
plt.subplot(10,1,7)
pzmap(sys_d4)
plt.title('Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1
5 for differential control')

plt.subplot(10,1,8)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_d4,T=t)
timeplot(t,s)
plt.title('Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1
5 for differentialcontrol')

k = 100
sys_d5 = signal.lti([k, 0], [1,(-.01+k), 1])
plt.subplot(10,1,9)
pzmap(sys_d5)
plt.title('Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1
00 for differential control')

plt.subplot(10,1,10)
t=np.linspace(0,10,1000)
_t,s=signal.step(sys_d5,T=t)
timeplot(t,s)
plt.title('Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1
00 for differential control')
```

Out[22]: <matplotlib.text.Text at 0x7f1b2b68cd50>

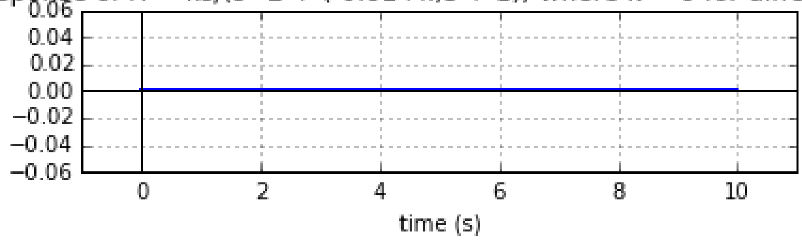
We can stabilize this system with a low feedback gain! If we try a negative gain value, the pole are on the right side of the imaginary axis and the step response goes to infinity. However, for small positive values of k, around 1-3, the poles exist on the left side of the imaginary axis and the step response converges to 0 after some oscillation. In we increase the feedback gain, k, up to 20 or higher but less than 100 the system promises that it will converge but after a longer time. Past a feedback gain value, k, of 100, the step response flattens out. The poles are still on the left side of the imaginary axis, but the step response seem to go flat. Perhaps the system will still stabilize, but it will likely take a really long time or it might not.



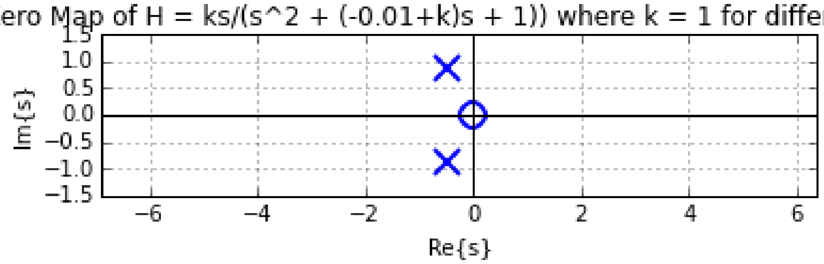




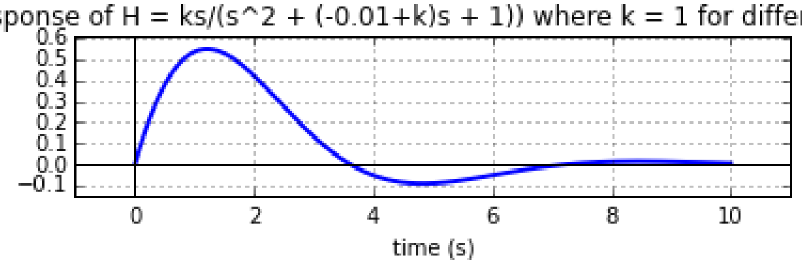
Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 0 for differential control



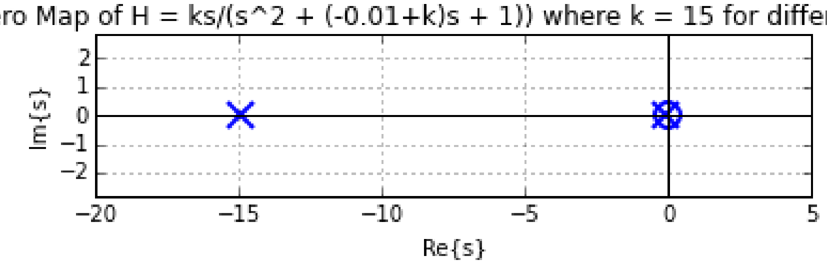
Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1 for differential control



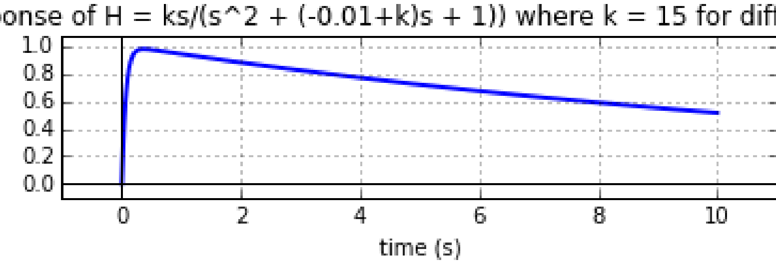
Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 1 for differential control



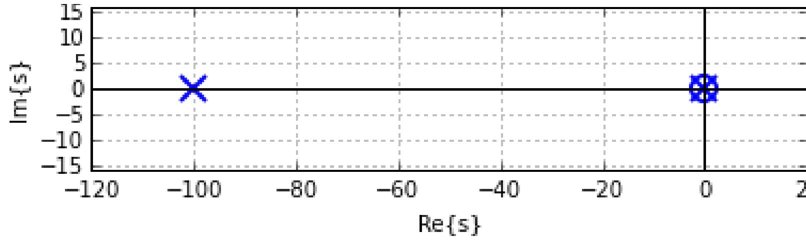
Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 15 for differential control



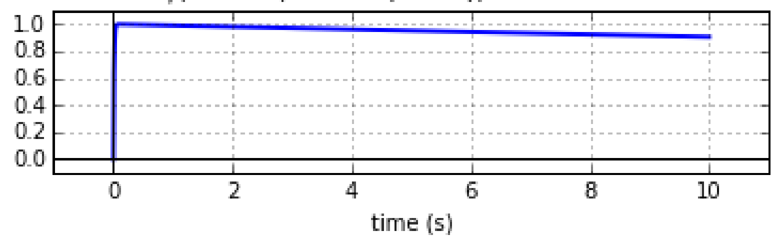
Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 15 for differentialcontrol



Pole Zero Map of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 100 for differential control



Step Response of H = ks/(s^2 + (-0.01+k)s + 1)) where k = 100 for differential control



In [17]: