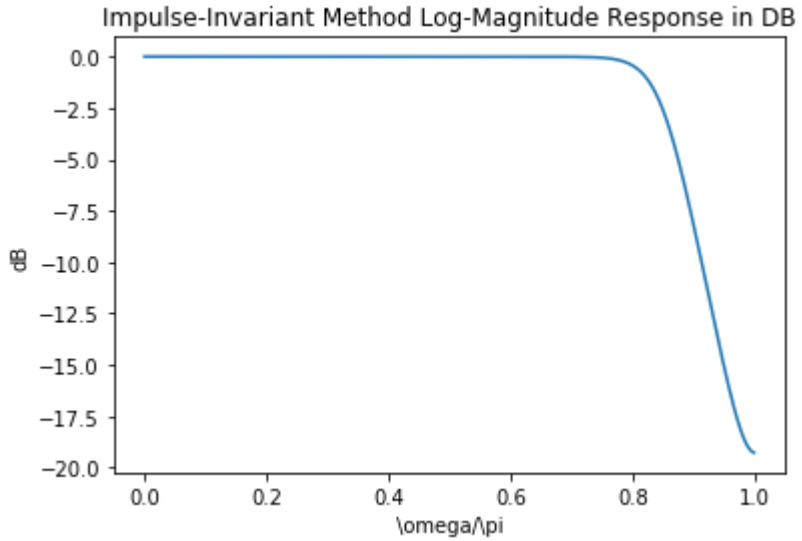
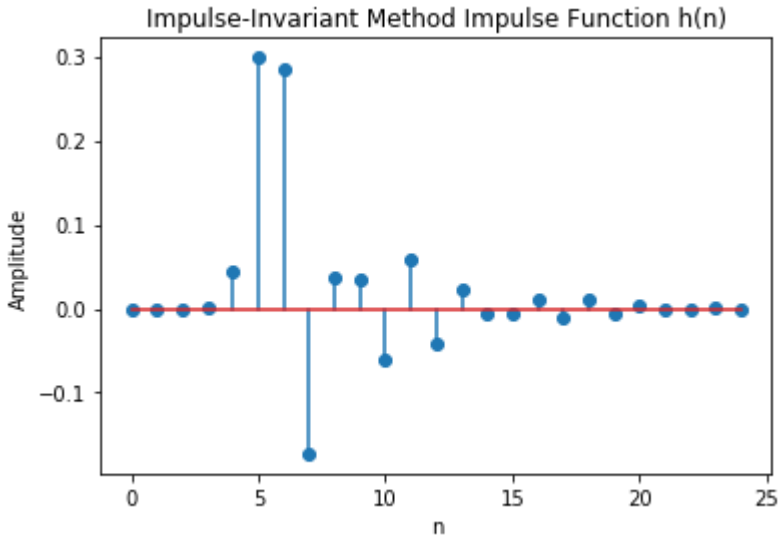
In [20]:
```
import numpy as np
import scipy
import math
import scipy.signal
from scipy import signal
import matplotlib.pyplot as plt
```

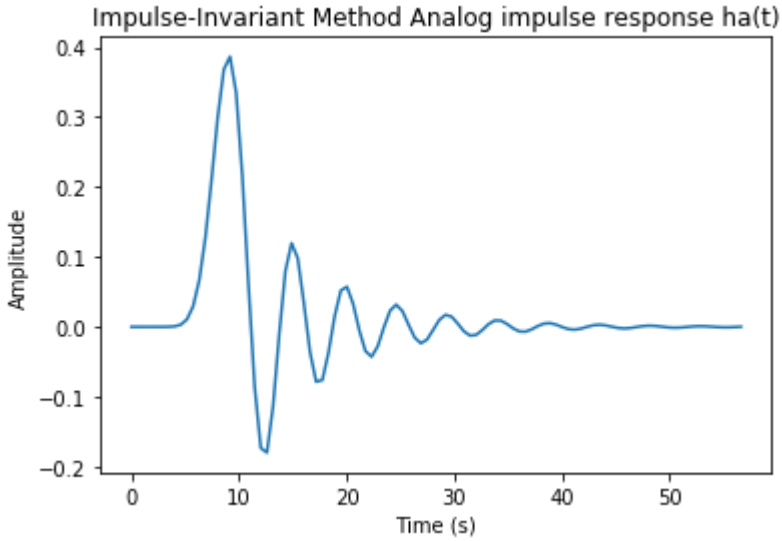
Problem 2a

In [41]:
```
omegap = 0.4 * np.pi
omegas = 0.6 * np.pi
Rp = 0.5
As = 50

numerator = np.log10((10**(Rp/10) - 1) / (10**(As/10) - 1))
denominator = 2 * np.log10(omegap / omegas)

N = math.ceil((numerator / denominator))
k = np.arange(N)

omegac1 = omegap / ((10**(Rp/10) - 1)**(1 / (2 * N)))
omegac2 = omegas / ((10**(As/10) - 1)**(1 / (2 * N)))

omegac = (omegac1 + omegac2) / 2

b, a = signal.butter(N, omegac, 'lowpass', analog = 'true')

coefficients, poles, k = signal.residue([0, b], a)

T = 2

poles = np.exp(poles * T)

#convert to z domain from s domain
bz, az = signal.invres(coefficients, poles, k)

w, h = signal.freqz(bz, az)

mag = np.abs(h)

dB = 20 * np.log10((mag) / np.max(mag))

plt.plot(w / np.pi, dB)
plt.title("Impulse-Invariant Method Log-Magnitude Response in DB")
plt.ylabel('dB')
plt.xlabel('\omega/\pi')
plt.show()


butterworth = signal.dlti(bz, az)
n, hn = signal.dimpulse(butterworth, n = 25)
plt.stem(n, np.squeeze(hn))
plt.title('Impulse-Invariant Method Impulse Function h(n)')
plt.ylabel('Amplitude')
plt.xlabel('n')
plt.show()

t, yt = signal.impulse2((b, a))
plt.plot(t, yt)
plt.title('Impulse-Invariant Method Analog impulse response ha(t)')
plt.ylabel('Amplitude')
plt.xlabel('Time (s)')
plt.show()
```



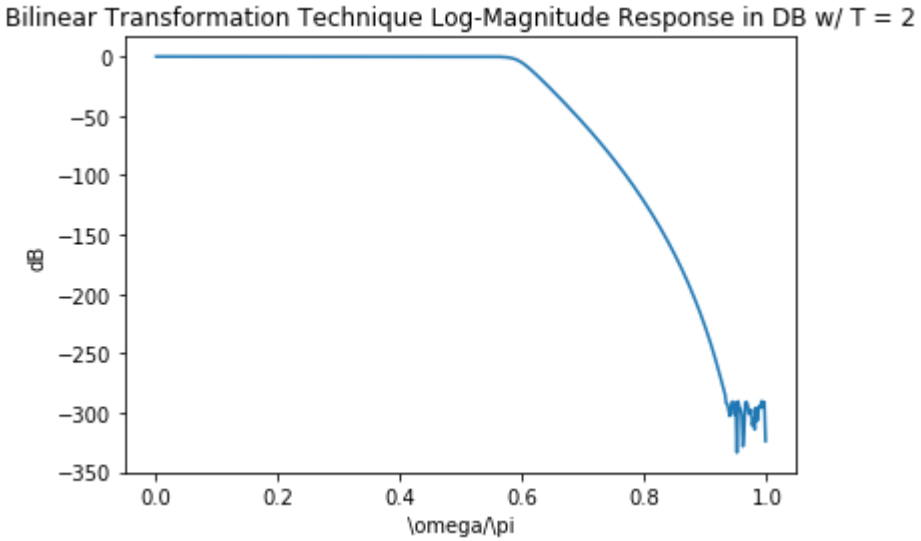

```
C:\Users\Tyan\Anaconda3\lib\site-packages\scipy\signal\ltisys.py:3368: Comple
xWarning: Casting complex values to real discards the imaginary part
  np.dot(system.B, u_dt[i, :]))
C:\Users\Tyan\Anaconda3\lib\site-packages\scipy\signal\ltisys.py:3370: Comple
xWarning: Casting complex values to real discards the imaginary part
  np.dot(system.D, u_dt[i, :]))
C:\Users\Tyan\Anaconda3\lib\site-packages\scipy\signal\ltisys.py:3374: Comple
xWarning: Casting complex values to real discards the imaginary part
  np.dot(system.D, u_dt[out_samples-1, :]))
C:\Users\Tyan\Anaconda3\lib\site-packages\ipykernel_launcher.py:43: UserWarni
ng: In Matplotlib 3.3 individual lines on a stem plot will be added as a Line
Collection instead of individual lines. This significantly improves the perfo
rmance of a stem plot. To remove this warning and switch to the new behaviou
r, set the "use_line_collection" keyword argument to True.
```



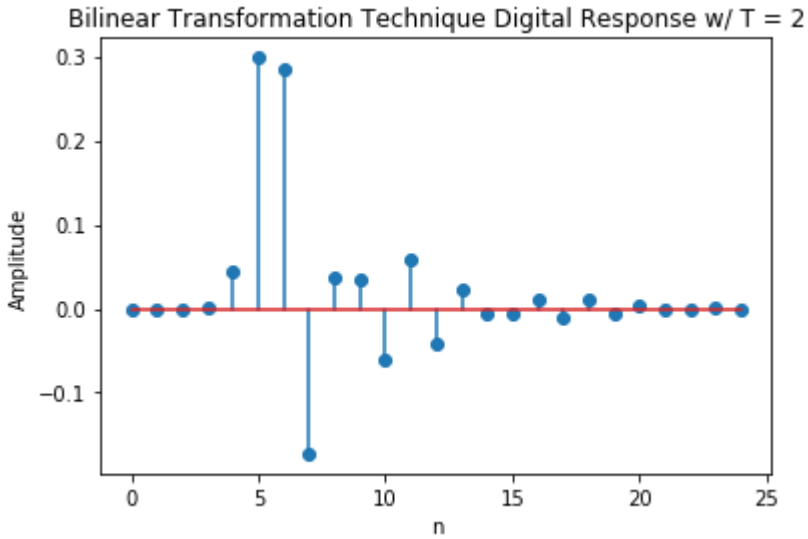

Impulse-Invariant Method Analog impulse response ha(t)
Amplitude
Time (s)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0
10
20
30
40
50

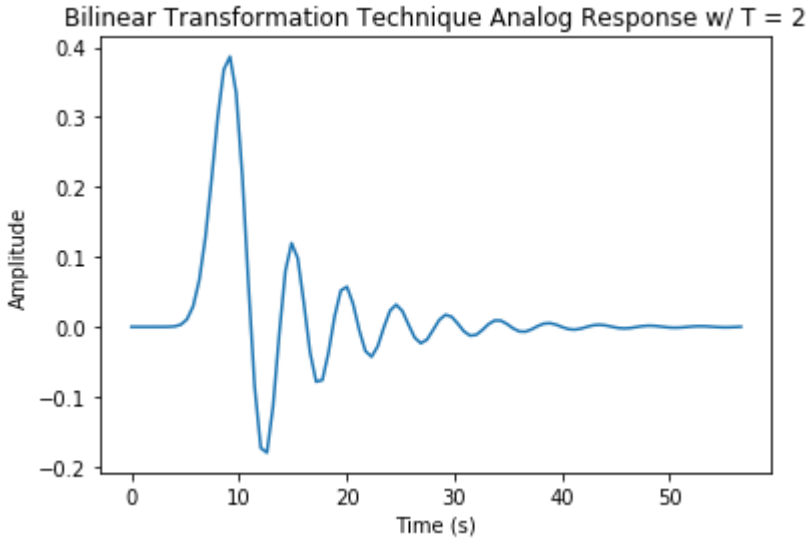
Problem 2b

In [42]:
```
#T = 2
omegap = 0.4 * np.pi
omegas = 0.6 * np.pi
Rp = 0.5
As = 50
T = 2

numerator = np.log10((10**(Rp/10) - 1) / (10**(As/10) - 1))
denominator = 2 * np.log10(omegap / omegas)

N = math.ceil((numerator / denominator))
k = np.arange(N)

omegac1 = omegap / ((10**(Rp/10) - 1)**(1 / (2 * N)))
omegac2 = omegas / ((10**(As/10) - 1)**(1 / (2 * N)))

omegac = (omegac1 + omegac2) / 2

b, a = signal.butter(N, omegac, 'lowpass', analog = 'true')

top, bottom = signal.bilinear(b, a, 1 / T)
w, h = signal.freqz(top, bottom)

mag = np.abs(h)
dB = 20 * np.log10((mag) / np.max(mag))

plt.plot(w / np.pi, dB)
plt.title("Bilinear Transformation Technique Log-Magnitude Response in DB w/ T
= 2")
plt.ylabel('dB')
plt.xlabel('\omega/\pi')
plt.show()


butterworth = signal.dlti(bz, az)
n, hn = signal.dimpulse(butterworth, n = 25)
plt.stem(n, np.squeeze(hn))
plt.title('Bilinear Transformation Technique Digital Response w/ T = 2')
plt.ylabel('Amplitude')
plt.xlabel('n')
plt.show()

t, yt = signal.impulse2((b, a))
plt.plot(t, yt)
plt.title('Bilinear Transformation Technique Analog Response w/ T = 2')
plt.ylabel('Amplitude')
plt.xlabel('Time (s)')
plt.show()
```

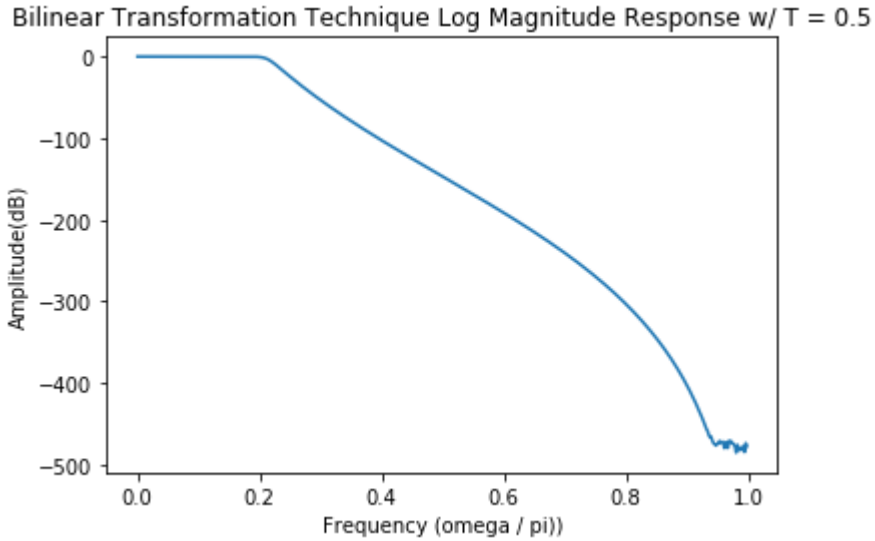



```
C:\Users\Tyan\Anaconda3\lib\site-packages\scipy\signal\ltisys.py:3368: Comple
xWarning: Casting complex values to real discards the imaginary part
  np.dot(system.B, u_dt[i, :]))
C:\Users\Tyan\Anaconda3\lib\site-packages\scipy\signal\ltisys.py:3370: Comple
xWarning: Casting complex values to real discards the imaginary part
  np.dot(system.D, u_dt[i, :]))
C:\Users\Tyan\Anaconda3\lib\site-packages\scipy\signal\ltisys.py:3374: Comple
xWarning: Casting complex values to real discards the imaginary part
  np.dot(system.D, u_dt[out_samples-1, :]))
C:\Users\Tyan\Anaconda3\lib\site-packages\ipykernel_launcher.py:36: UserWarni
ng: In Matplotlib 3.3 individual lines on a stem plot will be added as a Line
Collection instead of individual lines. This significantly improves the perfo
rmance of a stem plot. To remove this warning and switch to the new behaviou
r, set the "use_line_collection" keyword argument to True.
```

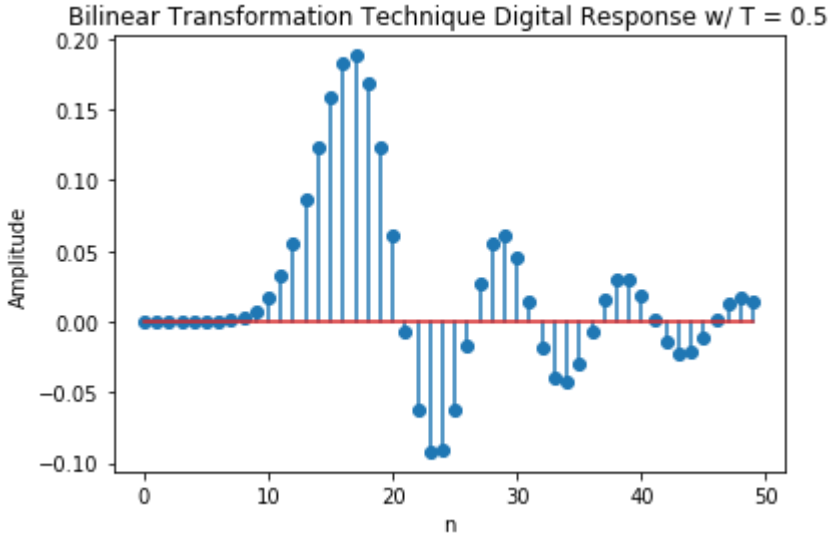


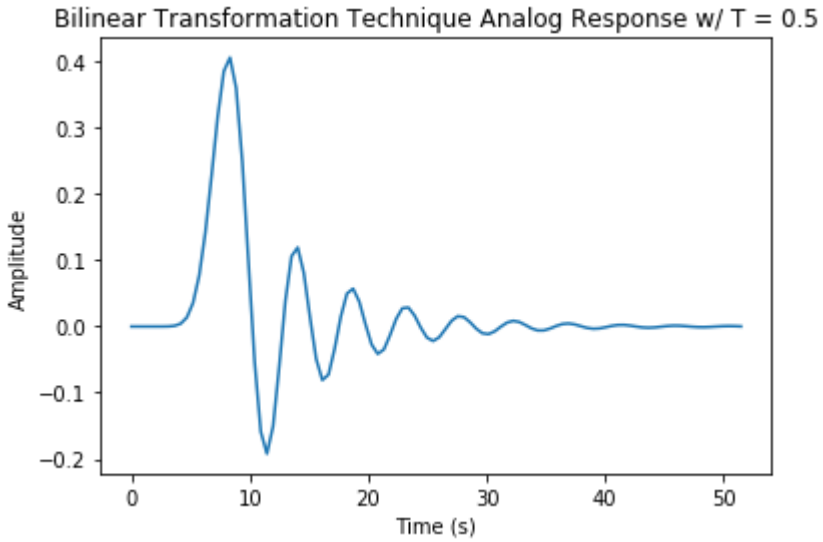

Bilinear Transformation Technique Analog Response w/ T = 2
Amplitude
Time (s)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0
10
20
30
40
50

In [43]:
```
#T = 0.5

omegap = 0.4 * np.pi
omegas = 0.6 * np.pi
Rp = 0.5
As = 50

T = 0.5
omegapstar = (2 / T) * np.tan((omegap * T) / 2)
omegasstar = (2 / T) * np.tan((omegas * T) / 2)

b, a = signal.iirdesign(omegapstar, omegasstar, gpass = Rp, gstop = As, ftype
= 'butter', analog = True)

numerator, denominator = signal.bilinear(b, a, 1 / T)
omegaz, hz = signal.freqz(numerator, denominator)

magnitude = np.abs(hz)

db = 20 * np.log10((magnitude) / np.max(magnitude))

plt.title('Bilinear Transformation Technique Log Magnitude Response w/ T = 0.
5')
plt.xlabel('Frequency (omega / pi))')
plt.ylabel('Amplitude(dB)')
plt.plot( omegaz / np.pi, db)
plt.show()

butterworth = signal.dlti(z, p)
n, hn = signal.dimpulse(butterworth, n = 50)
plt.stem(n, np.squeeze(hn))
plt.title('Bilinear Transformation Technique Digital Response w/ T = 0.5')
plt.ylabel('Amplitude')
plt.xlabel('n')
plt.show()

t, yt = signal.impulse2((b, a))
plt.plot(t, yt)
plt.title('Bilinear Transformation Technique Analog Response w/ T = 0.5')
plt.ylabel('Amplitude')
plt.xlabel('Time (s)')
plt.show()
```




```
C:\Users\Tyan\Anaconda3\lib\site-packages\ipykernel_launcher.py:29: UserWarni
ng: In Matplotlib 3.3 individual lines on a stem plot will be added as a Line
Collection instead of individual lines. This significantly improves the perfo
rmance of a stem plot. To remove this warning and switch to the new behaviou
r, set the "use_line_collection" keyword argument to True.
```





In [ ]: