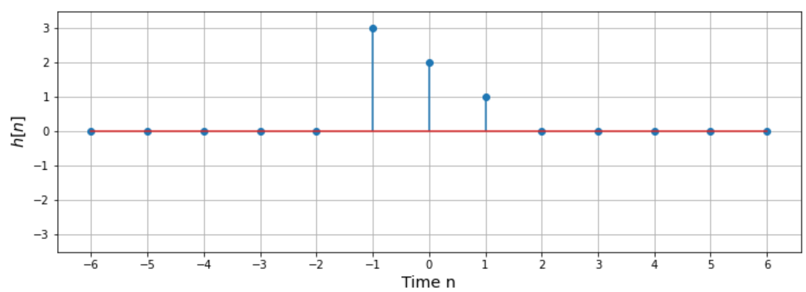
Signals & Systems
Homework #3 Solutions

ECE 315 – Fall 2022

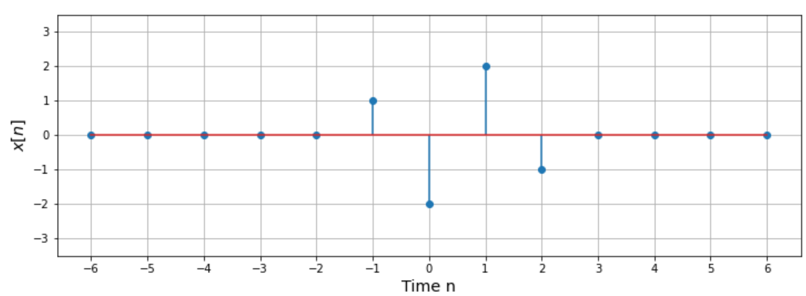
153 points total

Due Wednesday, November 9, 2022

1. Suppose a discrete-time LTI system has the impulse response $h[n]$

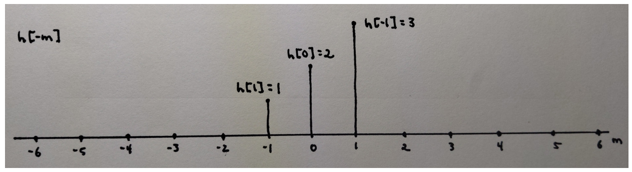


and that the system is stimulated with the signal $x[n]$

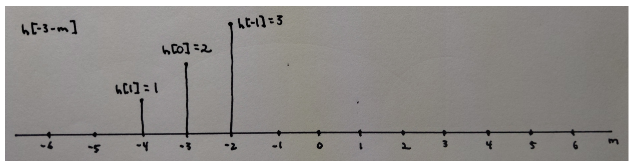


where $h[n]$ and $x[n]$ are 0 at all values of n not shown.

(a) Find and plot the signal $h[-m]$ as a function of the integer m by hand. (5 points)



(b) Find and plot the signal $h[-3-m] = h[-(m+3)]$ as a function of the integer m by hand. (5 points)



(c) Calculate all nonzero values of the output $y[n]$ of the system in response to the input signal $x[n]$ using convolution by hand. (15 points)

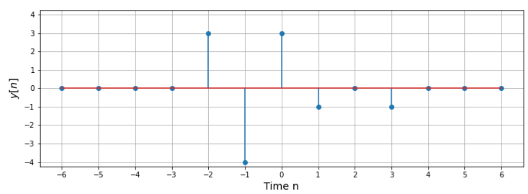
For $n < -2$, the nonzero values of $h[n-m]$ align with zero values of $x[m]$ and $y[n] = 0$. For $-2 \le n \le 3$,

$$
\begin{aligned}
y[-2] &= \sum_{m=-\infty}^{\infty} x[m]h[-2-m] = x[-1]h[-1] = 1 \cdot 3 = 3, \\
y[-1] &= \sum_{m=-\infty}^{\infty} x[m]h[-1-m] = x[-1]h[0] + x[0]h[-1] \\
&= 1 \cdot 2 + (-2) \cdot 3 = 2 - 6 = -4, \\
y[0] &= \sum_{m=-\infty}^{\infty} x[m]h[-m] = x[-1]h[1] + x[0]h[0] + x[1]h[-1] \\
&= 1 \cdot 1 + (-2) \cdot 2 + 2 \cdot 3 = 1 - 4 + 6 = 3, \\
y[1] &= \sum_{m=-\infty}^{\infty} x[m]h[1-m] = x[0]h[1] + x[1]h[0] + x[2]h[-1] \\
&= (-2) \cdot 1 + 2 \cdot 2 + (-1) \cdot 3 = -2 + 4 - 3 = -1, \\
y[2] &= \sum_{m=-\infty}^{\infty} x[m]h[2-m] = x[1]h[1] + x[2]h[0] = 2 \cdot 1 + (-1) \cdot 2 = 2 - 2 = 0, \\
y[3] &= \sum_{m=-\infty}^{\infty} x[m]h[3-m] = x[2]h[1] = (-1) \cdot 1 = -1.
\end{aligned}
$$

Finally, when $n > 3$, the nonzero values of $h[n-m]$ align with zero values of $x[m]$ and $y[n] = 0$. So, altogether,

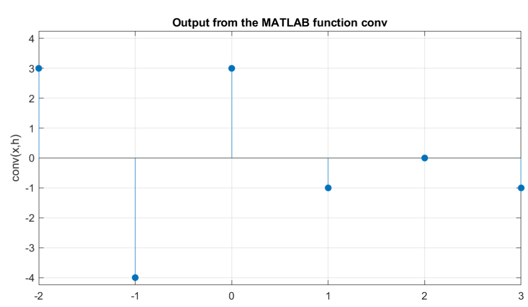
$$y[n] = \begin{cases} 3, & n = -2 \\ -4, & n = -1 \\ 3, & n = 0 \\ -1, & n = 1 \\ 0, & n = 2 \\ -1, & n = 3 \\ 0, & \text{otherwise.} \end{cases}$$

This signal is plotted below.

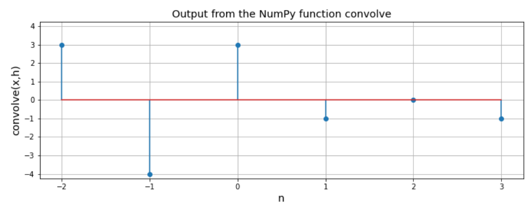


(d) Verify your solution using the MATLAB function `conv` or the NumPy function `convolve`. Include your code and its output in your answer. (7 points)

MATLAB code used to evaluate the convolution of $x[n]$ and $h[n]$ is in Appendix A. The output from this code is plotted below. The MATLAB function `conv` returned a vector that contained the nonzero values of the convolution. Note that the nonzero values match but the software didn't provide the matching values of n. I had to manually add them after determining them above.



Python code used to evaluate the convolution of $x[n]$ and $h[n]$ is in Appendix B. The output from this code is plotted below. The NumPy function `convolve` returned a vector that contained the nonzero values of the convolution. Note that the nonzero values match but the function `convolve` didn't provide the matching values of n. I had to manually add them after determining them above.



2. Consider the discrete-time LTI system

$$-8y[n] + 2y[n-1] + y[n-2] = -2x[n] + 3x[n-1].$$

(a) Find the general solution $y_h[n]$ to the homogeneous equation

$$-8y_h[n] + 2y_h[n-1] + y_h[n-2] = 0. \text{ (5 points)}$$

For the linear constant-coefficient difference equation,

$$a_0 = -8, a_1 = 2, a_2 = 1, b_0 = -2, \text{ and } b_1 = 3.$$

The bases r of the exponential functions that are solutions to the homogeneous equation are the solutions of the characteristic polynomial equation

$$0 = a_0 r^2 + a_1 r + a_0 = -8r^2 + 2r + 1.$$

The values of r are given by the quadratic formula

$$r = \frac{-2 \pm \sqrt{4+32}}{-16} = \frac{1}{8} \pm \frac{6}{16} = \frac{1}{8} \pm \frac{3}{8} = \frac{1}{2}, -\frac{1}{4}.$$

Let

$$r_1 = \frac{1}{2} \quad \text{and} \quad r_2 = -\frac{1}{4}.$$

Then the general solution to the homogeneous problem is

$$y_h[n] = c_1 r_1^n + c_2 r_2^n = c_1 \left(\frac{1}{2}\right)^n + c_2 \left(-\frac{1}{4}\right)^n,$$

where c_1 and c_2 are constants.

(b) Find initial values $\hat{h}[0]$ and $\hat{h}[1]$ for the impulse response $\hat{h}[n]$, where

$$-8\hat{h}[n] + 2\hat{h}[n-1] + \hat{h}[n-2] = \delta[n]$$

and $\hat{h}[n] = 0$ for $n < 0$. (5 points)

The recursive relationship

$$\hat{h}[n] = \frac{1}{a_0}\left(\delta[n] - \sum_{k=1}^{N} a_k \hat{h}[n-k]\right) = -\frac{1}{8}\left(\delta[n] - 2\hat{h}[n-1] - \hat{h}[n-2]\right)$$

allows the calculation of values of $\hat{h}[n]$ using two previous values. In this case,

$$\begin{aligned}
\hat{h}[0] &= -\frac{1}{8}\left(\delta[0] - 2\hat{h}[-1] - \hat{h}[-2]\right) = -\frac{1}{8} \\
\hat{h}[1] &= -\frac{1}{8}\left(\delta[1] - 2\hat{h}[0] - \hat{h}[-1]\right) = -\frac{1}{8}\left(-2 \cdot \left(-\frac{1}{8}\right)\right) = -\frac{1}{32}.
\end{aligned}$$

(c) Use the initial values calculated in the previous part to determine the values of the undetermined coefficients in the solution $\hat{h}[n] = y_h[n]u[n]$, which is the impulse response for the problem

$$-8y[n] + 2y[n-1] + y[n-2] = x[n]. \text{ (10 points)}$$

The impulse response $\hat{h}[n]$ has the form

$$\hat{h}[n] = \left(c_1 \left(\frac{1}{2}\right)^n + c_2 \left(-\frac{1}{4}\right)^n\right) u[n],$$

where c_1 and c_2 are constants. The constants c_1 and c_2 are determined by the equations

$$\begin{aligned}
c_1 + c_2 &= \hat{h}[0] = -\frac{1}{8} \\
\frac{1}{2}c_1 - \frac{1}{4}c_2 &= \hat{h}[1] = -\frac{1}{32}.
\end{aligned}$$

These equations can be solved using the augmented matrix with the specified row operations

$$\left(\begin{array}{cc|c} 1 & 1 & -\frac{1}{8} \\ \frac{1}{2} & -\frac{1}{4} & -\frac{1}{32} \end{array}\right) \quad R_2 - \frac{1}{2}R_1 \longrightarrow R_2$$

$$\left(\begin{array}{cc|c} 1 & 1 & -\frac{1}{8} \\ 0 & -\frac{3}{4} & \frac{1}{32} \end{array}\right) \quad -\frac{4}{3}R_2 \longrightarrow R_2$$

$$\left(\begin{array}{cc|c} 1 & 1 & -\frac{1}{8} \\ 0 & 1 & -\frac{1}{24} \end{array}\right) \quad R_1 - R_2 \longrightarrow R_1$$

$$\left(\begin{array}{cc|c} 1 & 0 & -\frac{1}{12} \\ 0 & 1 & -\frac{1}{24} \end{array}\right)$$

So,

$$c_1 = -\frac{1}{12}, c_2 = -\frac{1}{24},$$

and

$$\hat{h}[n] = \left(-\frac{1}{12}\left(\frac{1}{2}\right)^n - \frac{1}{24}\left(-\frac{1}{4}\right)^n\right)u[n].$$

(d) Use the solution calculated in the previous part to construct the impulse response for the original difference equation

$$-8y[n] + 2y[n-1] + y[n-2] = -2x[n] + 3x[n-1]. \text{ (5 points)}$$

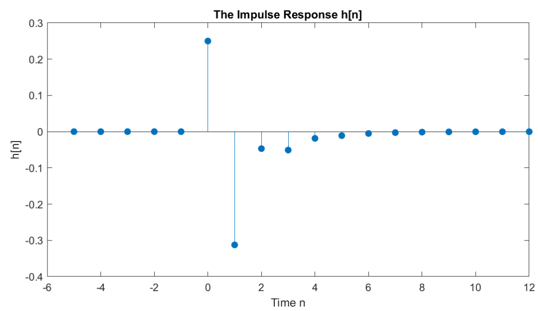
The impulse response for the original equation is

$$\begin{aligned} h[n] &= b_0\hat{h}[n] + b_1\hat{h}[n-1] \\ &= -2\hat{h}[n] + 3\hat{h}[n-1] \\ &= -2\left(-\frac{1}{12}\left(\frac{1}{2}\right)^n - \frac{1}{24}\left(-\frac{1}{4}\right)^n\right)u[n] \\ &\quad + 3\left(-\frac{1}{12}\left(\frac{1}{2}\right)^{n-1} - \frac{1}{24}\left(-\frac{1}{4}\right)^{n-1}\right)u[n-1]. \end{aligned}$$

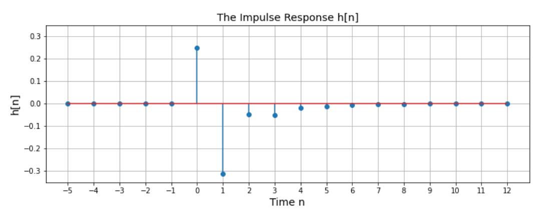
(e) Use MATLAB or python to plot the impulse response of the original system. If the impulse response is complex, plot its magnitude and phase and plot it as a stem plot in the 3 dimensions of its real and imaginary parts and time n. (10 points)

Notice that the impulse response $h[n]$ is real-valued.

MATLAB code for plotting the impulse response $h[n]$ is in Appendix C. The output from this code is below.



Python code for plotting the impulse response $h[n]$ is in Appendix D. The output from this code is below.



3. Consider the discrete-time LTI system

$$-8y[n] + 2y[n-1] + y[n-2] = -2x[n] + 3x[n-1].$$

(a) Find the transfer function for the system analytically. (5 points)

In this case,

$$a_0 = -8, a_1 = 2, a_2 = 1, b_0 = -2, \text{ and } b_1 = 3.$$

The transfer function is

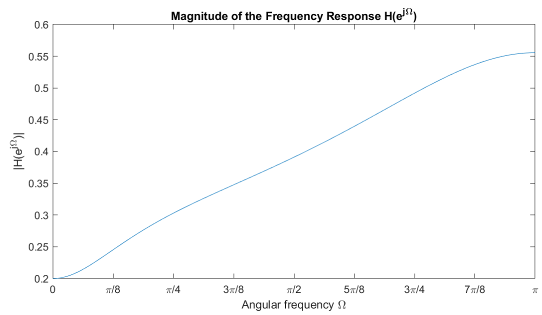
$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} = \frac{-2 + 3z^{-1}}{-8 + 2z^{-1} + z^{-2}} = \frac{z(3 - 2z)}{1 + 2z - 8z^2}.$$

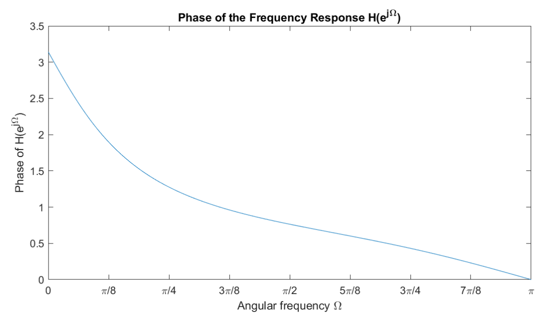
(b) Find the frequency response for the system analytically and use MATLAB or python to plot the magnitude and phase of your result as functions of angular frequency Ω for $0 \leq \Omega \leq \pi$. (10 points)

The frequency response is

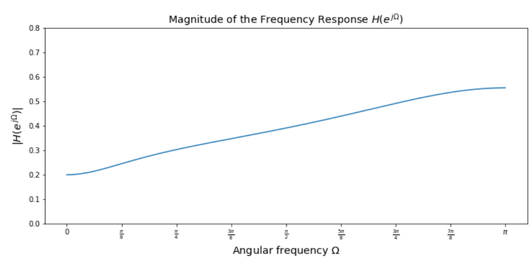
$$H\left(e^{j\Omega}\right) = \frac{-2 + 3e^{-j\Omega}}{-8 + 2e^{-j\Omega} + e^{-j2\Omega}} = \frac{e^{j\Omega}(3 - 2e^{j\Omega})}{1 + 2e^{j\Omega} - 8e^{j2\Omega}}$$

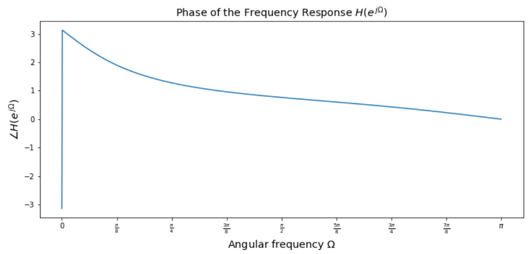
MATLAB code for plotting the magnitude and phase of $H\left(e^{j\Omega}\right)$ is in Appendix E. The output of this code is below.





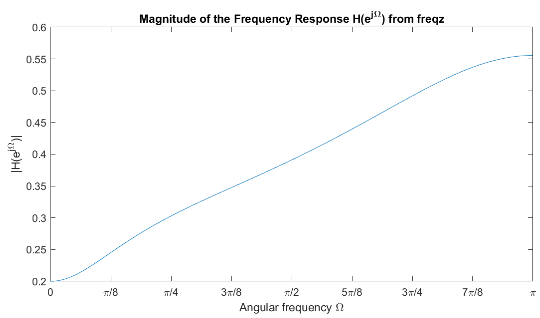
Python code for plotting the magnitude and phase of $H\left(e^{j\Omega}\right)$ is in Appendix F. The output of this code is below.

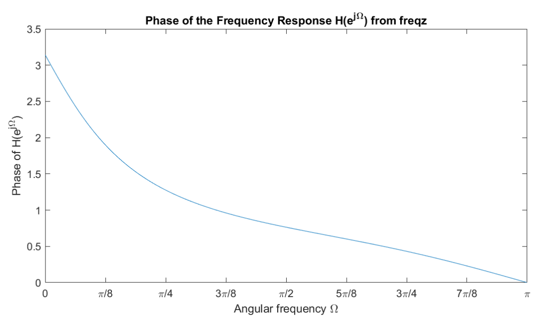




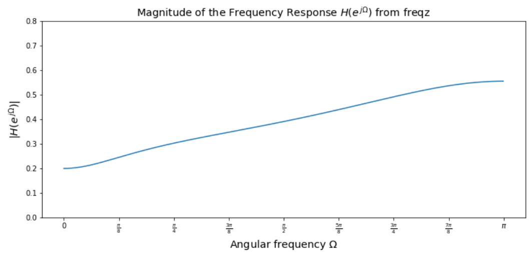
(c) Use the MATLAB function `freqz` or the SciPy function `freqz` to determine and plot the magnitude and phase of the frequency response for $0 \leq \Omega \leq \pi$. (7 points)

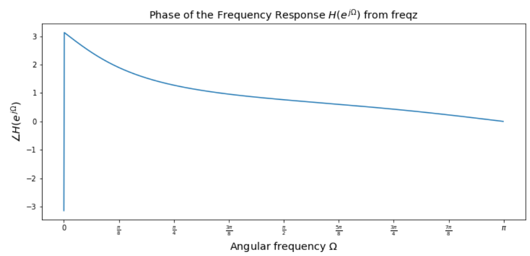
MATLAB code for determining and plotting the frequency response using the function `freqz` is in Appendix G. The output from this code is below. Notice that these plots match those generated in Part (b).





Python code for determining and plotting the frequency response using the SciPy function `freqz` is in Appendix H. The output from this code is below. Notice that these plots match those generated in Part (b).





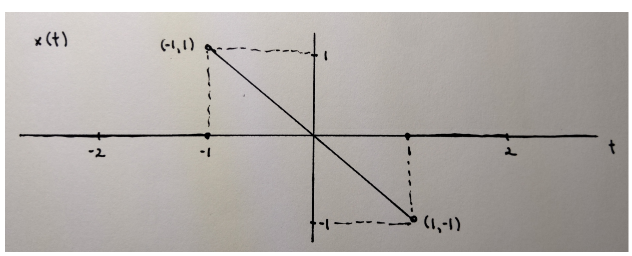
4. Suppose that

$$h(t) = \begin{cases} 1 - t^2, & 0 < t < 1 \\ 0, & \text{otherwise} \end{cases}$$

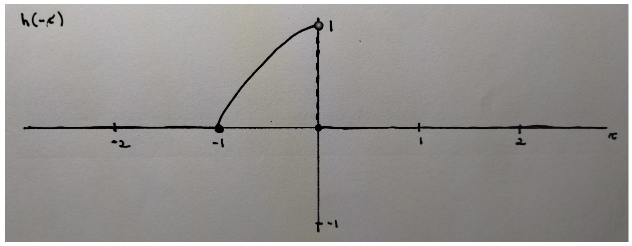
is the impulse response for a continuous-time LTI system and that the system is stimulated with the signal

$$x(t) = \begin{cases} -t, & -1 < t < 1 \\ 0, & \text{otherwise.} \end{cases}$$

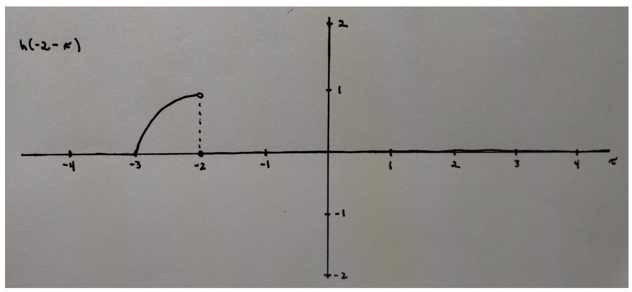
(a) Plot the signal $x(t)$ by hand. (5 points)



(b) Find and plot the signal $h(-\tau)$ as a function of τ by hand. (5 points)



(c) Find and plot the signal $h(-2-\tau) = h(-(\tau+2))$ as a function of τ by hand. (5 points)



(d) Calculate the output $y(t)$ of the system in response to the input signal $x(t)$ using convolution by hand. (20 points)

The output signal is

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\,d\tau.$$

Notice that the largest value of τ for which $h(t-\tau) \neq 0$ is $\tau = t$. So, when $t \leq -1$, the nonzero portions of $x(\tau)$ and $h(t-\tau)$ do not overlap and $y(t) = 0$.

When $-1 \leq t \leq 0$, the nonzero portions of $x(\tau)$ and $h(t-\tau)$ overlap but only part of the nonzero portion of $h(t-\tau)$ overlaps the nonzero portion of $x(\tau)$. This means that the lower limit is $\tau = -1$ and the upper limit is the largest value of τ where $h(t-\tau) \neq 0$, i.e. $\tau = t$. So, for $-1 \leq t \leq 0$,

$$y(t) = \int_{-1}^{t} x(\tau)h(t-\tau)\,d\tau = \int_{-1}^{t} (-\tau)(1-(t-\tau)^2)\,d\tau.$$

Here,

$$1-(t-\tau)^2 = 1-(t^2-2t\tau+\tau^2) = 1-t^2+2t\tau-\tau^2$$

so that the integrand is

$$(-\tau)(1-(t-\tau)^2) = -\tau\left(1 - t^2 + 2t\tau - \tau^2\right) = \tau^3 - 2t\tau^2 + (t^2-1)\tau.$$

The integrand has the antiderivative

$$\int \left(\tau^3 - 2t\tau^2 + (t^2-1)\tau\right)\, d\tau = \frac{1}{4}\tau^4 - \frac{2t}{3}\tau^3 + \frac{1}{2}(t^2-1)\tau^2 + c,$$

where c is an arbitrary constant. Therefore,

$$\begin{aligned}
y(t) &= \left(\frac{1}{4}\tau^4 - \frac{2t}{3}\tau^3 + \frac{1}{2}(t^2-1)\tau^2\right)\Bigg|_{-1}^{t} \\
&= \left(\frac{1}{4}t^4 - \frac{2}{3}t^4 + \frac{1}{2}(t^2-1)t^2\right) - \left(\frac{1}{4} + \frac{2t}{3} + \frac{1}{2}(t^2-1)\right) \\
&= \left(\left(\frac{3}{4} - \frac{2}{3}\right)t^4 - \frac{1}{2}t^2\right) - \left(\frac{1}{2}t^2 + \frac{2}{3}t - \frac{1}{4}\right) \\
&= \left(\frac{1}{12}t^4 - \frac{1}{2}t^2\right) - \left(\frac{1}{2}t^2 + \frac{2}{3}t - \frac{1}{4}\right) \\
&= \frac{1}{12}t^4 - t^2 - \frac{2}{3}t + \frac{1}{4}.
\end{aligned}$$

When $0 < t \leq 1$, the nonzero portion of $x(\tau)$ entirely overlaps the nonzero portion of $h(t-\tau)$, which is nonzero for $t-1 < \tau < t$. So, for $0 < t \leq 1$,

$$y(t) = \int_{t-1}^{t} x(\tau)h(t-\tau)\, d\tau = \int_{t-1}^{t} (-\tau)(1-(t-\tau)^2)\, d\tau.$$

The integrand is the same as before and has the same antiderivative. So,

$$\begin{aligned}
y(t) &= \left(\frac{1}{4}\tau^4 - \frac{2t}{3}\tau^3 + \frac{1}{2}(t^2-1)\tau^2\right)\Bigg|_{t-1}^{t} \\
&= \left(\frac{1}{12}t^4 - \frac{1}{2}t^2\right) - \left(\frac{1}{4}(t-1)^4 - \frac{2t}{3}(t-1)^3 + \frac{1}{2}(t^2-1)(t-1)^2\right),
\end{aligned}$$

where I used the expression determined above for the antiderivative evaluated at the upper limit. In this case,

$$\begin{aligned}
(t-1)^2 &= t^2 - 2t + 1 \\
(t-1)^3 &= (t-1)(t^2 - 2t + 1) = t^3 - 3t^2 + 3t - 1 \\
(t-1)^4 &= (t-1)(t^3 - 3t^2 + 3t - 1) = t^4 - 4t^3 + 6t^2 - 4t + 1
\end{aligned}$$

and the antiderivative evaluated at the lower limit is

$$\begin{aligned}
\frac{1}{4}(t-1)^4 - \frac{2t}{3}(t-1)^3 + \frac{1}{2}(t^2-1)(t-1)^2 &= \frac{1}{4}(t^4 - 4t^3 + 6t^2 - 4t + 1) \\
&\quad - \frac{2}{3}t(t^3 - 3t^2 + 3t - 1) \\
&\quad + \frac{1}{2}(t^2-1)(t^2 - 2t + 1) \\
&= \frac{1}{4}(t^4 - 4t^3 + 6t^2 - 4t + 1) \\
&\quad - \frac{2}{3}(t^4 - 3t^3 + 3t^2 - t) \\
&\quad + \frac{1}{2}(t^4 - 2t^3 + 2t - 1) \\
&= \left(\frac{3}{4} - \frac{2}{3}\right) t^4 + (-1 + 2 - 1)\, t^3 \\
&\quad + \left(\frac{3}{2} - 2\right) t^2 + \left(-1 + \frac{2}{3} + 1\right) t \\
&\quad + \frac{1}{4} - \frac{1}{2} \\
&= \frac{1}{12}t^4 - \frac{1}{2}t^2 + \frac{2}{3}t - \frac{1}{4}.
\end{aligned}$$

Therefore, for $0 < t \leq 1$,

$$\begin{aligned}
y(t) &= \left(\frac{1}{12}t^4 - \frac{1}{2}t^2\right) - \left(\frac{1}{12}t^4 - \frac{1}{2}t^2 + \frac{2}{3}t - \frac{1}{4}\right) \\
&= \frac{1}{4} - \frac{2}{3}t.
\end{aligned}$$

Lastly, when $1 < t \leq 2$, the right-hand end of the nonzero portion of $h(t-\tau)$, which is at $\tau = t$, lies beyond the right-hand end of the nonzero portion of $x(\tau)$ and

$$y(t) = \int_{t-1}^{1} x(\tau)h(t-\tau)\, d\tau = \int_{t-1}^{1} (-\tau)(1-(t-\tau)^2)\, d\tau.$$

The integrand is the same as before and has the same antiderivative. So,

$$
\begin{aligned}
y(t) &= \left(\frac{1}{4}\tau^4 - \frac{2t}{3}\tau^3 + \frac{1}{2}(t^2-1)\tau^2\right)\Bigg|_{t-1}^{1} \\
&= \left(\frac{1}{4} - \frac{2}{3}t + \frac{1}{2}t^2 - \frac{1}{2}\right) - \left(\frac{1}{4}(t-1)^4 - \frac{2t}{3}(t-1)^3 + \frac{1}{2}(t^2-1)(t-1)^2\right) \\
&= \frac{1}{2}t^2 - \frac{2}{3}t - \frac{1}{4} - \left(\frac{1}{4}(t-1)^4 - \frac{2t}{3}(t-1)^3 + \frac{1}{2}(t^2-1)(t-1)^2\right).
\end{aligned}
$$

The antiderivative evaluated at the lower limit was previously determined to be

$$
\frac{1}{4}(t-1)^4 - \frac{2t}{3}(t-1)^3 + \frac{1}{2}(t^2-1)(t-1)^2 = \frac{1}{12}t^4 - \frac{1}{2}t^2 + \frac{2}{3}t - \frac{1}{4}.
$$

Therefore, when $1 < t \leq 2$,

$$
\begin{aligned}
y(t) &= \frac{1}{2}t^2 - \frac{2}{3}t - \frac{1}{4} - \left(\frac{1}{12}t^4 - \frac{1}{2}t^2 + \frac{2}{3}t - \frac{1}{4}\right) \\
&= -\frac{1}{12}t^4 + t^2 - \frac{4}{3}t.
\end{aligned}
$$

When $t > 2$, the nonzero portions of $x(\tau)$ and $h(t-\tau)$ do not overlap and $y(t) = 0$.
Putting this altogether,

$$
y(t) = \begin{cases} \frac{1}{12}t^4 - t^2 - \frac{2}{3}t + \frac{1}{4}, & -1 \leq t \leq 0 \\ \frac{1}{4} - \frac{2}{3}t, & 0 < t \leq 1 \\ -\frac{1}{12}t^4 + t^2 - \frac{4}{3}t, & 1 < t \leq 2 \\ 0, & \text{otherwise.} \end{cases}
$$

It's possible to use the fact that $y(t)$ should be continuous as a sanity check. This means

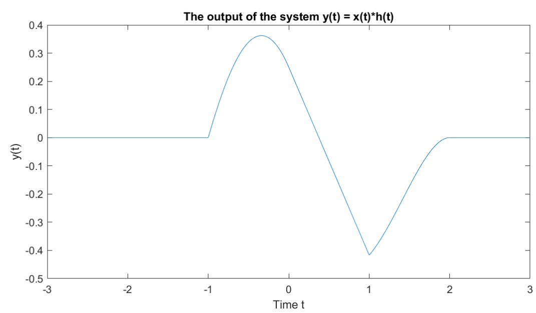
$$
\begin{aligned}
y(-1) &= 0 \\
y(0^+) &= y(0^-) \\
y(1^+) &= y(1^-) \\
y(2) &= 0.
\end{aligned}
$$

Using the above expressions,

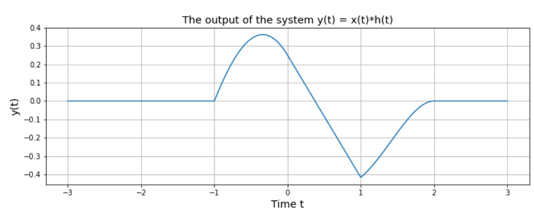
$$\begin{aligned} y(-1) &= \frac{1}{12} - 1 + \frac{2}{3} + \frac{1}{4} = 0 \\ y(0^+) &= \frac{1}{4} = y(0^-) \\ y(1^+) &= -\frac{5}{12} = y(1^-) \\ y(2) &= 0. \end{aligned}$$

(e) Use MATLAB or python to plot the convolution $y(t)$. (7 points)

MATLAB code for plotting $y(t)$ is in Appendix I. The output from this code is below.



Python code for plotting $y(t)$ is in Appendix J. The output from this code is below.



5. Consider the continuous-time LTI system

$$2\,y''(t) - 5\,y'(t) - 7\,y(t) = -3x''(t) + 11\,x(t)$$

(a) Find the transfer function for the system analytically. (5 points)

In this case,

$$a_2 = 2, a_1 = -5, a_0 = -7, b_2 = -3, b_1 = 0, \text{ and } b_0 = 11$$

and the transfer function is

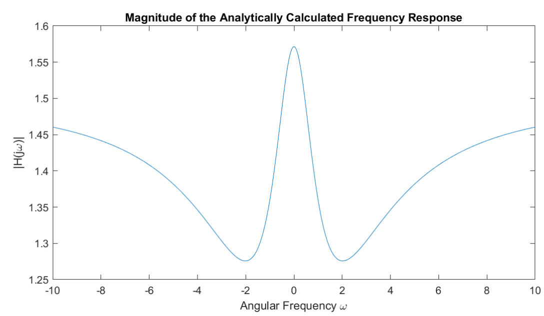
$$H(s) = \frac{\sum_{k=0}^{M} b_k s^k}{\sum_{k=0}^{N} a_k s^k} = \frac{-3s^2 + 11}{2s^2 - 5s - 7}.$$

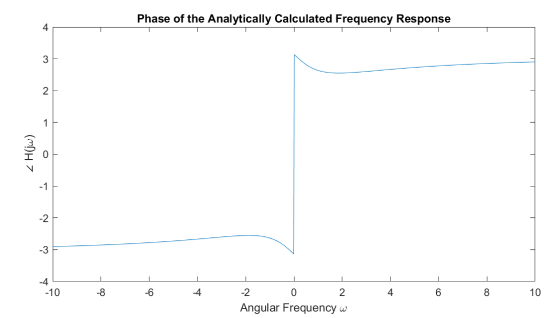
(b) Find the frequency response for the system analytically and use MATLAB or python to plot the magnitude and phase of your result as functions of angular frequency ω for $-10 \leq \omega \leq 10$. (10 points)

The frequency response is

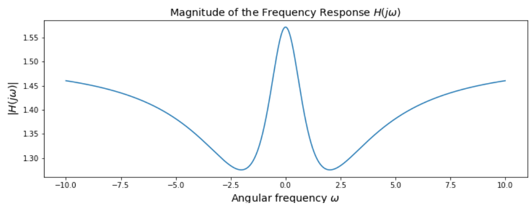
$$H(j\omega) = \frac{-3(j\omega)^2 + 11}{2(j\omega)^2 - 5(j\omega) - 7} = \frac{3\omega^2 + 11}{-2\omega^2 - 7 - j5\omega}$$

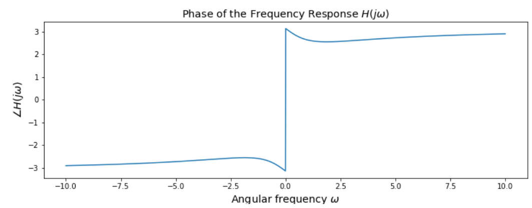
MATLAB code for plotting $H(j\omega)$ is in Appendix K. The output from this code is below.





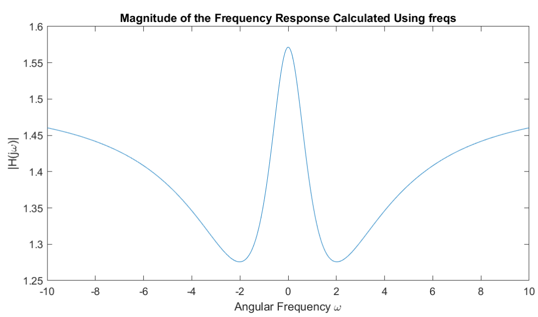
Python code for plotting $H(j\omega)$ is in Appendix L. The output from this code is below.

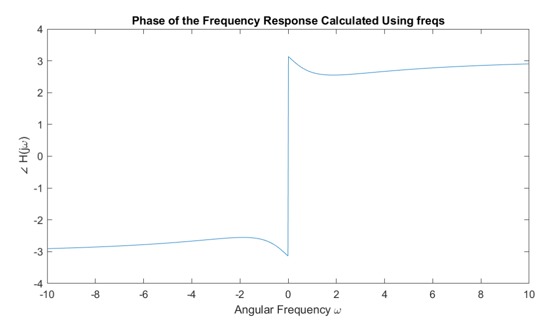




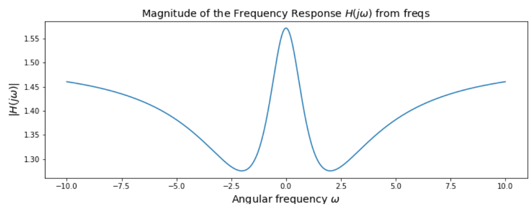
(c) Use the MATLAB function **`freqs`** or the SciPy function **`freqs`** to determine and plot the magnitude and phase of the frequency response for $-10 \leq \omega \leq 10$. (7 points)

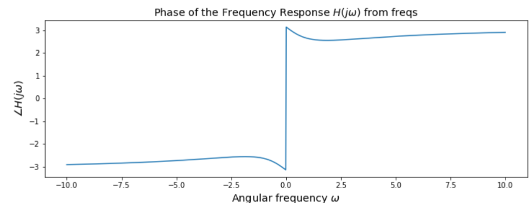
MATLAB code for plotting the frequency response using the function **`freqs`** is in Appendix M. The output from this code is below. Notice that this matches the output from Part (b).





Python code for plotting the frequency response using the SciPy function `freqs` is in Appendix N. The output from this code is below. Notice that this matches the output from Part (b).





A MATLAB Code for Problem 1, Part (d)

```
% Script to calculate the convolution of x[n] and h[n] in Problem 1d

% Nonzero portions of h and x
h = [3,  2, 1];
x = [1, -2, 2, -1];

% Times when x*h is possibly nonzero
% These are known from the earlier parts of Problem 1.
n = -2:3;

y = conv(x,h);

figure('Position', [0 0 800 400])
stem(n, y, 'filled')
xticks(n)
ylim([-4.25, 4.25])
title('Output from the MATLAB function conv')
xlabel('n')
ylabel('conv(x,h)')
```

```
grid on
saveas(gcf, 'prob1d_MATLAB.png')
```

B Python Code for Problem 1, Part (d)

```
#--------------------------------------------------------------#
# The convolution of the signal x with the impulse response h #
# calculated using the NumPy function convolve                 #
#--------------------------------------------------------------#

import numpy as np
import matplotlib.pyplot as plt

# Nonzero portions of h and x
h = [3,  2, 1];
x = [1, -2, 2, -1];

# Times when x*h is possibly nonzero
# These are known from the earlier parts of Problem 1.
n = np.arange(-2, 4);

y = np.convolve(x,h);

fig, ax = plt.subplots(1, figsize=(12,4))
ax.stem(n, y)
ax.set_ylim([-4.25, 4.25])
ax.set_title('Output from the NumPy function convolve', fontsize="x-large")
ax.set_xlabel('n', fontsize="x-large")
ax.set_ylabel('convolve(x,h)', fontsize="x-large")
ax.grid(True)
plt.savefig('prob1d_python.png')
plt.show()
```

C MATLAB Code for Problem 2, Part (e)

```
% Script for plotting the impulse response determined in Problem 2
```

```
r1 =  0.5;
r2 = -0.25;
c1 = -1.0/12.0;
c2 = -1.0/24.0;

nmin = -5;
nmax =  12;
n = nmin:nmax;

h = zeros(size(n));

zeroIdx = -nmin + 1;
for m = 0:nmax
    if m < 1
        h(zeroIdx + m) = -2.0*(c1*r1^m + c2*r2^m);
    else
        h(zeroIdx + m) = -2.0*(c1*r1^m + c2*r2^m)...
                         + 3.0*(c1*r1^(m-1) + c2*r2^(m-1));
    end
end

figure('Position', [200, 200, 800, 400])
stem(n, h, 'filled')
title('The Impulse Response h[n]')
ylabel('h[n]')
xlabel('Time n')
saveas(gcf, 'prob2e_MATLAB.png')
```

D Python Code for Problem 2, Part (e)

```
#-------------------------------------------------#
# Plot the impulse response derived in Problem 2. #
#-------------------------------------------------#

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

```
r1 =  0.5
r2 = -0.25
c1 = -1.0/12.0
c2 = -1.0/24.0

nmin =  -5;
nmax =  12;
n = np.arange(nmin, nmax + 1)

h = np.zeros(np.size(n))

zeroIdx = -nmin
for m in np.arange(0, nmax):
    if m < 1:
        h[zeroIdx + m] = -2.0*(c1*r1**m + c2*r2**m)
    else:
        h[zeroIdx + m] = -2.0*(c1*r1**m + c2*r2**m) + 3.0*(c1*r1**(m-1) + c2*r2**(

fig, ax = plt.subplots(1, figsize=(12,4))
ax.stem(n, h)
ax.set_title('The Impulse Response h[n]', fontsize="x-large")
ax.set_xlabel('Time n', fontsize="x-large")
ax.set_ylabel('h[n]', fontsize="x-large")
ax.set_xticks(n)
ax.set_ylim([-0.35, 0.35])
ax.grid(True)
plt.savefig('prob2e_python.png')
plt.show()
```

E MATLAB Code for Problem 3, Part (b)

```
% Script for plotting the frequency response determined analytically in
% Problem 3, Part b

Omega = linspace(0, pi, 1024);
z = exp(1j*Omega);
```

```
zsqrd = z.*z;

H = (3.0*z - 2.0*zsqrd)./(1.0 + 2.0*z - 8.0*zsqrd);

magH   = abs(H);
phaseH = angle(H);

ticks = (pi/8.0)*(0:8);
labels = {'0', '\pi/8', '\pi/4', '3\pi/8', '\pi/2',...
      '5\pi/8', '3\pi/4', '7\pi/8', '\pi'};

figure('Position', [400, 400, 800, 400])
plot(Omega, magH)
title('Magnitude of the Frequency Response H(e^{j\Omega})')
ylabel('|H(e^{j\Omega})|')
xlabel('Angular frequency \Omega')
xlim([0, pi])
xticks(ticks)
xticklabels(labels)
saveas(gcf, 'prob3b_mag_MATLAB.png')

figure('Position', [600, 600, 800, 400])
plot(Omega, phaseH)
title('Phase of the Frequency Response H(e^{j\Omega})')
ylabel('Phase of H(e^{j\Omega})')
xlabel('Angular frequency \Omega')
xlim([0, pi])
xticks(ticks)
xticklabels(labels)
saveas(gcf, 'prob3b_phase_MATLAB.png')
```

F Python Code for Problem 3, Part (b)

```
#-----------------------------------------------------------------------#
# Plot the magnitude and phase of the frequency response of the system #
#                                                                      #
#        -8y[n] + 2y[n-1] + y[n-2] = -2x[n] + 3x[n-1]                  #
```

```
#                                                                       #
# which was determined analytically in Problem 3, Part b                #
#-----------------------------------------------------------------------#

import numpy as np
import matplotlib.pyplot as plt

Omega = np.linspace(0, np.pi, 1024)
z = np.exp(1j*Omega)
zsqrd = np.multiply(z, z)

H = np.divide(3.0*z - 2.0*zsqrd, 1.0 + 2.0*z - 8.0*zsqrd)

magH   = np.abs(H);
phaseH = np.angle(H);

fig, ax = plt.subplots(1, figsize=(12,5))
ax.plot(Omega, magH)
ax.set_title('Magnitude of the Frequency Response $H(e^{\,j\Omega})$', fontsize="x
ax.set_ylabel('$|H(e^{\,j\Omega})|$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\Omega$', fontsize="x-large")
freqStep = np.pi/8.0
ax.set_xticks(freqStep*np.arange(0, 9))
ax.set_xticklabels(["0", "$\\frac{\pi}{8}$", "$\\frac{\pi}{4}$", "$\\frac{3\pi}{8}
                    "$\\frac{\pi}{2}$", "$\\frac{5\pi}{8}$", "$\\frac{3\pi}{4}$",\
                    "$\\frac{7\pi}{8}$","$\\pi$"])
ax.set_ylim([0.0, 0.8])
plt.savefig('prob3b_mag_python.png')
plt.show()

fig, ax = plt.subplots(1, figsize=(12,5))
ax.plot(Omega, phaseH)
ax.set_title('Phase of the Frequency Response $H(e^{\,j\Omega})$', fontsize="x-lar
ax.set_ylabel('$\\angle H(e^{\,j\Omega})$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\Omega$', fontsize="x-large")
ax.set_xticks(freqStep*np.arange(0, 9))
ax.set_xticklabels(["0", "$\\frac{\pi}{8}$", "$\\frac{\pi}{4}$", "$\\frac{3\pi}{8}
                    "$\\frac{\pi}{2}$", "$\\frac{5\pi}{8}$", "$\\frac{3\pi}{4}$",\
```

```
                    "$\\frac{7\pi}{8}$","$\\pi$"])
plt.savefig('prob3b_phase_python.png')
plt.show()
```

G MATLAB Code for Problem 3, Part (c)

```
% Script for using freqz to determine the frequency response for the system
% in Problem 3, Part c.

a = [-8.0,  2.0,  1.0];
b = [-2.0,  3.0,  0.0];

[h, w] = freqz(b, a, 1024);

mag_h = abs(h);
phase_h = angle(h);

ticks = (pi/8.0)*(0:8);
labels = {'0', '\pi/8', '\pi/4', '3\pi/8', '\pi/2', '5\pi/8', '3\pi/4', '7\pi/8',

figure('Position', [500, 500, 800, 400])
plot(w, mag_h)
title('Magnitude of the Frequency Response H(e^{j\Omega}) from freqz')
ylabel('|H(e^{j\Omega})|')
xlabel('Angular frequency \Omega')
xlim([0, pi])
xticks(ticks)
xticklabels(labels)
saveas(gcf, 'prob3c_mag_MATLAB.png')

figure('Position', [600, 600, 800, 400])
plot(w, phase_h)
title('Phase of the Frequency Response H(e^{j\Omega}) from freqz')
ylabel('Phase of H(e^{j\Omega})')
xlabel('Angular frequency \Omega')
xlim([0, pi])
xticks(ticks)
```

```
xticklabels(labels)
saveas(gcf, 'prob3c_phase_MATLAB.png')
```

H Python Code for Problem 3, Part (c)

```
#---------------------------------------------------------------------------#
# Plot the magnitude and phase of the frequency response of the system #
# in Problem 3, Part c using the SciPy function freqz.                 #
#---------------------------------------------------------------------------#

import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

a = [-8.0, 2.0, 1.0]
b = [-2.0, 3.0, 0.0]

w, H = signal.freqz(b, a, 1024)

magH   = np.abs(H);
phaseH = np.angle(H);

fig, ax = plt.subplots(1, figsize=(12,5))
ax.plot(w, magH)
ax.set_title('Magnitude of the Frequency Response $H(e^{\,j\Omega})$ from freqz',\
fontsize="x-large")
ax.set_ylabel('$|H(e^{\,j\Omega})|$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\Omega$', fontsize="x-large")
freqStep = np.pi/8.0
ax.set_xticks(freqStep*np.arange(0, 9))
ax.set_xticklabels(["0", "$\\frac{\pi}{8}$", "$\\frac{\pi}{4}$", "$\\frac{3\pi}{8}
                    "$\\frac{\pi}{2}$", "$\\frac{5\pi}{8}$", "$\\frac{3\pi}{4}$",\
                    "$\\frac{7\pi}{8}$","$\\pi$"])
ax.set_ylim([0.0, 0.8])
plt.savefig('prob3c_mag_python.png')
plt.show()
```

```
fig, ax = plt.subplots(1, figsize=(12,5))
ax.plot(w, phaseH)
ax.set_title('Phase of the Frequency Response $H(e^{\,j\Omega})$ from freqz',\
fontsize="x-large")
ax.set_ylabel('$\\angle H(e^{\,j\Omega})$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\Omega$', fontsize="x-large")
ax.set_xticks(freqStep*np.arange(0, 9))
ax.set_xticklabels(["0", "$\\frac{\pi}{8}$", "$\\frac{\pi}{4}$", "$\\frac{3\pi}{8}
                    "$\\frac{\pi}{2}$", "$\\frac{5\pi}{8}$", "$\\frac{3\pi}{4}$",\
                    "$\\frac{7\pi}{8}$","$\\pi$"])
plt.savefig('prob3c_phase_python.png')
plt.show()
```

I MATLAB Code for Problem 4, Part (e)

```
% Script for plotting the convolution y(t) determined in Problem 4.
% The polynomials are evaluated using Horner's rule, which is a more stable
% and accurate way of determining their values.

numpts = 2000;
t = linspace(-3.0, 3.0, numpts);
y = zeros(size(t));

oneTwelth  = 1.0/12.0;
twoThirds  = 2.0/3.0;
fourThirds = 4.0/3.0;

for m = 1:numpts
    if t(m) <= -1.0
        continue
    elseif t(m) <= 0.0
        y(m) = 0.25 + t(m)*(-twoThirds + t(m)*(-1.0 + oneTwelth*t(m)*t(m)));
    elseif t(m) <= 1.0
        y(m) = 0.25 - twoThirds*t(m);
    elseif t(m) <= 2.0
        y(m) = t(m)*(-fourThirds + t(m)*(1.0 - oneTwelth*t(m)*t(m)));
    else
```

```
        break
    end
end

figure('Position', [500, 500, 800, 400])
plot(t, y)
xlabel('Time t')
ylabel('y(t)')
title('The output of the system y(t) = x(t)*h(t)')
saveas(gcf, 'prob4e_MATLAB.png')
```

J Python Code for Problem 4, Part (e)

```
#-----------------------------------------------------#
# Plot the convolution y(t) of the input signal x(t) #
# and the impulse response h(t) from Problem 4.      #
#-----------------------------------------------------#

import numpy as np
import matplotlib.pyplot as plt

numpts = 2000
t = np.linspace(-3.0, 3.0, numpts)
y = np.zeros(np.size(t))

oneTwelth  = 1.0/12.0
twoThirds  = 2.0/3.0
fourThirds = 4.0/3.0

# The polynomials are evaluated using Horner's rule, which is a more stable
# and accurate way of determining their values.

for m in np.arange(0, numpts+1):
    if t[m] <= -1.0:
        continue
    elif t[m] <= 0.0:
        y[m] = 0.25 + t[m]*(-twoThirds + t[m]*(-1.0 + oneTwelth*t[m]*t[m]))
```

```
    elif t[m] <= 1.0:
        y[m] = 0.25 - twoThirds*t[m]
    elif t[m] <= 2.0:
        y[m] = t[m]*(-fourThirds + t[m]*(1.0 - oneTwelth*t[m]*t[m]))
    else:
        break

fig, ax = plt.subplots(1, figsize=(12,4))
ax.plot(t,y)
ax.set_xlabel('Time t', fontsize="x-large")
ax.set_ylabel('y(t)', fontsize="x-large")
ax.set_title('The output of the system y(t) = x(t)*h(t)', fontsize="x-large")
ax.grid(True)
plt.savefig('prob4e_python.png')
plt.show()
```

K MATLAB Code for Problem 5, Part (b)

```
% Script for plotting the analytically calculated frequency response in
% Problem 5, Part b

omega = linspace(-10, 10, 1024);
omega2 = omega.*omega;

H = (3.0*omega2 + 11.0)./(-2.0*omega2 - 7.0 - 5j*omega);

magH   = abs(H);
phaseH = angle(H);

figure('Position', [500, 500, 800, 400])
plot(omega, magH)
title('Magnitude of the Analytically Calculated Frequency Response')
ylabel('|H(j\omega)|')
xlabel('Angular Frequency \omega')
saveas(gcf, 'prob5b_mag_MATLAB.png')

figure('Position', [600, 600, 800, 400])
```

```
plot(omega, phaseH)
title('Phase of the Analytically Calculated Frequency Response')
ylabel('\angle H(j\omega)')
xlabel('Angular Frequency \omega')
saveas(gcf, 'prob5b_phase_MATLAB.png')
```

L Python Code for Problem 5, Part (b)

```
#--------------------------------------------------------------#
# Plot of the magnitude and phase of the frequency response of #
# the system determined analytically in Problem 5, Part b.     #
#--------------------------------------------------------------#

import numpy as np
import matplotlib.pyplot as plt

omega  = np.linspace(-10.0, 10.0, 2000)
omega2 = np.multiply(omega, omega)

H = np.divide(3.0*omega2 + 11, -2.0*omega2 - 7.0 - 5j*omega)

magH   = np.abs(H);
phaseH = np.angle(H);

fig, ax = plt.subplots(1, figsize=(12,4))
ax.plot(omega, magH)
ax.set_title('Magnitude of the Frequency Response $H(j\omega)$', fontsize="x-large
ax.set_ylabel('$|H(j\omega)|$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\omega$', fontsize="x-large")
plt.savefig('prob5b_mag_python.png')
plt.show()

fig, ax = plt.subplots(1, figsize=(12,4))
ax.plot(omega, phaseH)
ax.set_title('Phase of the Frequency Response $H(j\omega)$', fontsize="x-large")
ax.set_ylabel('Phase of $H(j\omega)$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\omega$', fontsize="x-large")
```

```
plt.savefig('prob5b_phase_python.png')
plt.show()
```

M MATLAB Code for Problem 5, Part (c)

```
% Script for plotting the frequency response in Problem 5 using the MATLAB
% function freqs

omega = linspace(-10.0, 10.0, 1024);

a = [ 2.0, -5.0, -7.0];
b = [-3.0,  0.0, 11.0];

H = freqs(b, a, omega);
magH   = abs(H);
phaseH = angle(H);

figure('Position', [500, 500, 800, 400])
plot(omega, magH)
title('Magnitude of the Frequency Response Calculated Using freqs')
ylabel('|H(j\omega)|')
xlabel('Angular Frequency \omega')
saveas(gcf, 'prob5c_mag_MATLAB.png')

figure('Position', [600, 600, 800, 400])
plot(omega, phaseH)
title('Phase of the Frequency Response Calculated Using freqs')
ylabel('\angle H(j\omega)')
xlabel('Angular Frequency \omega')
saveas(gcf, 'prob5c_phase_MATLAB.png')
```

N Python Code for Problem 5, Part (c)

```
#----------------------------------------------------------------------#
# Plot the magnitude and phase of the frequency response of the system #
# in Problem 5, Part c using the SciPy function freqs.                 #
```

```
#----------------------------------------------------------------------#

import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

a = [ 2.0, -5.0, -7.0]
b = [-3.0,  0.0, 11.0]

w, H = signal.freqs(b, a, worN=np.linspace(-10.0, 10.0, 1024))

magH   = np.abs(H);
phaseH = np.angle(H);

fig, ax = plt.subplots(1, figsize=(12,4))
ax.plot(w, magH)
ax.set_title('Magnitude of the Frequency Response $H(j\omega)$ from freqs',\
fontsize="x-large")
ax.set_ylabel('$|H(j\omega)|$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\omega$', fontsize="x-large")
plt.savefig('prob5c_mag_python.png')
plt.show()

fig, ax = plt.subplots(1, figsize=(12,4))
ax.plot(w, phaseH)
ax.set_title('Phase of the Frequency Response $H(j\omega)$ from freqs',\
fontsize="x-large")
ax.set_ylabel('$\\angle H(j\omega)$', fontsize="x-large")
ax.set_xlabel('Angular frequency $\omega$', fontsize="x-large")
plt.savefig('prob5c_phase_python.png')
plt.show()
```