Report of Signals and Systems Lab Assignment 4

Written by HUANG Guanchao, SID 11912309 and GONG Xinrui, SID 11911233.

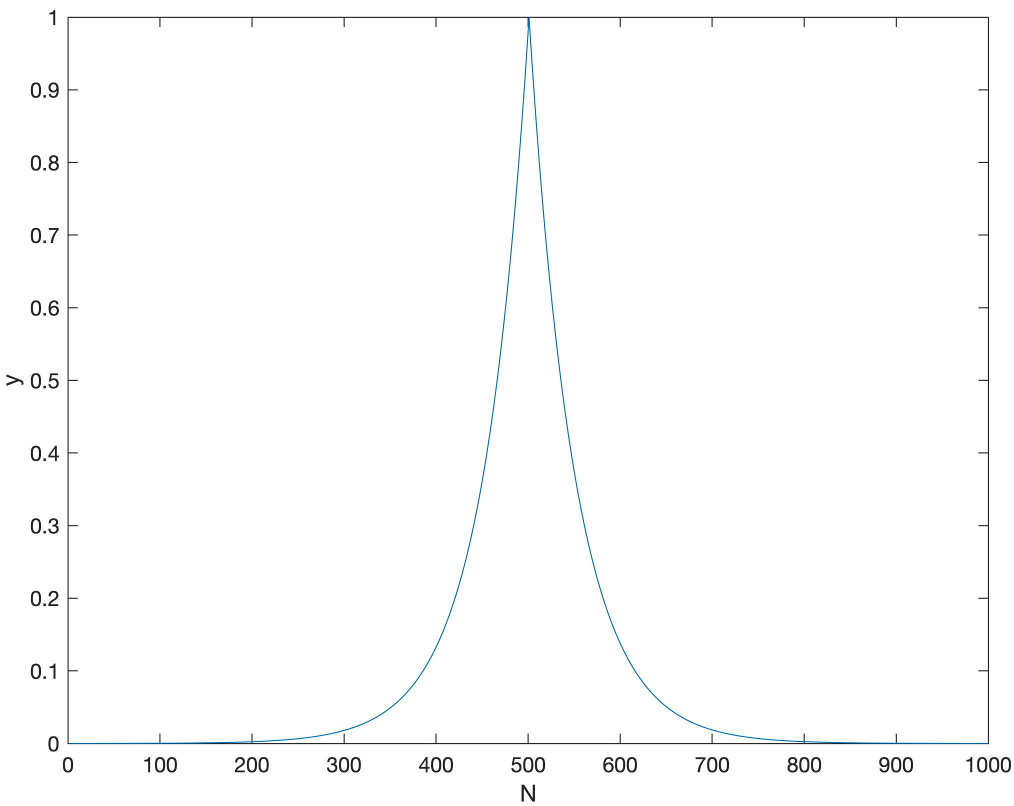
4.2

Basic Problem a

From the given expression of $x(t)$, we can rewrite $x(t)$ into $x(t) = e^{-2t}u(t) + e^{2t}u(-t)$. So $X(j\omega) = \dfrac{1}{2+j\omega} + \dfrac{1}{2-j\omega}$, then we standardize the expression into $X(j\omega) = \dfrac{4}{4+\omega^2}$.

Basic Problem b

The plot of vector y is shown below.

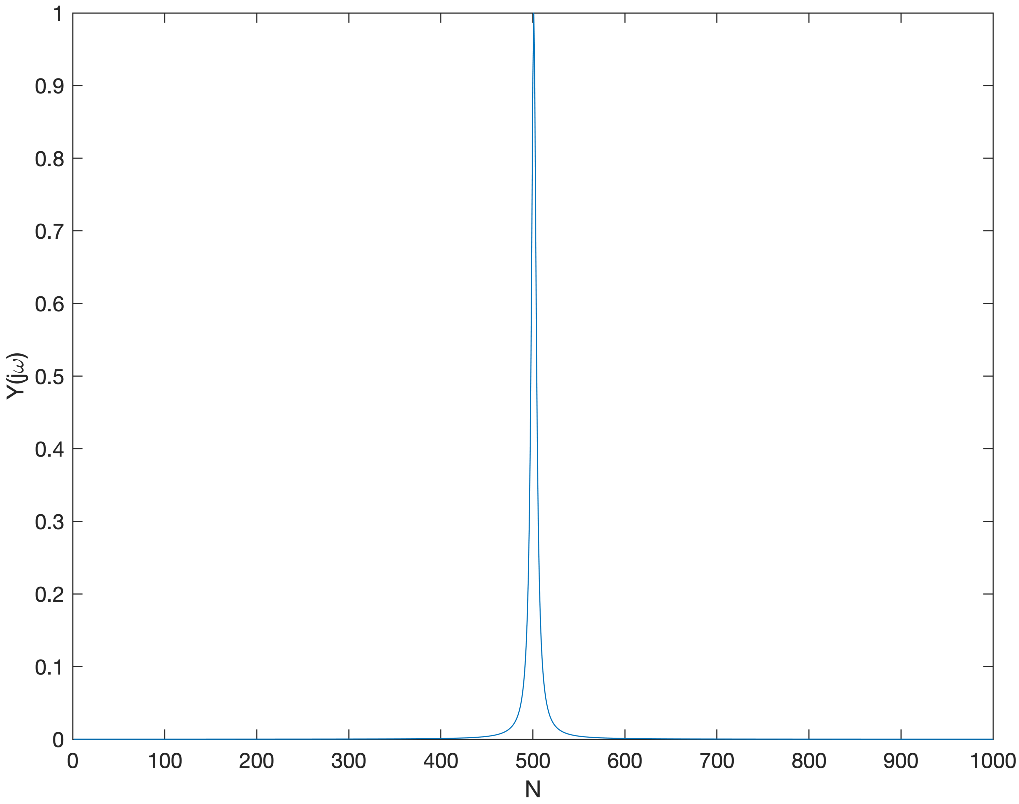


The MATLAB script is shown below.

```
tau = 0.01;
T = 10;
N = T / tau;
t = (0:tau:T - 0.01);
x = exp(-2 .* abs(t));
y = exp(-2 .* abs(t - 5));

plot(y)
xlabel('N')
ylabel('y')
```

Basic Problem c

The plot of magnitude of $Y(j\omega)$ is shown below.

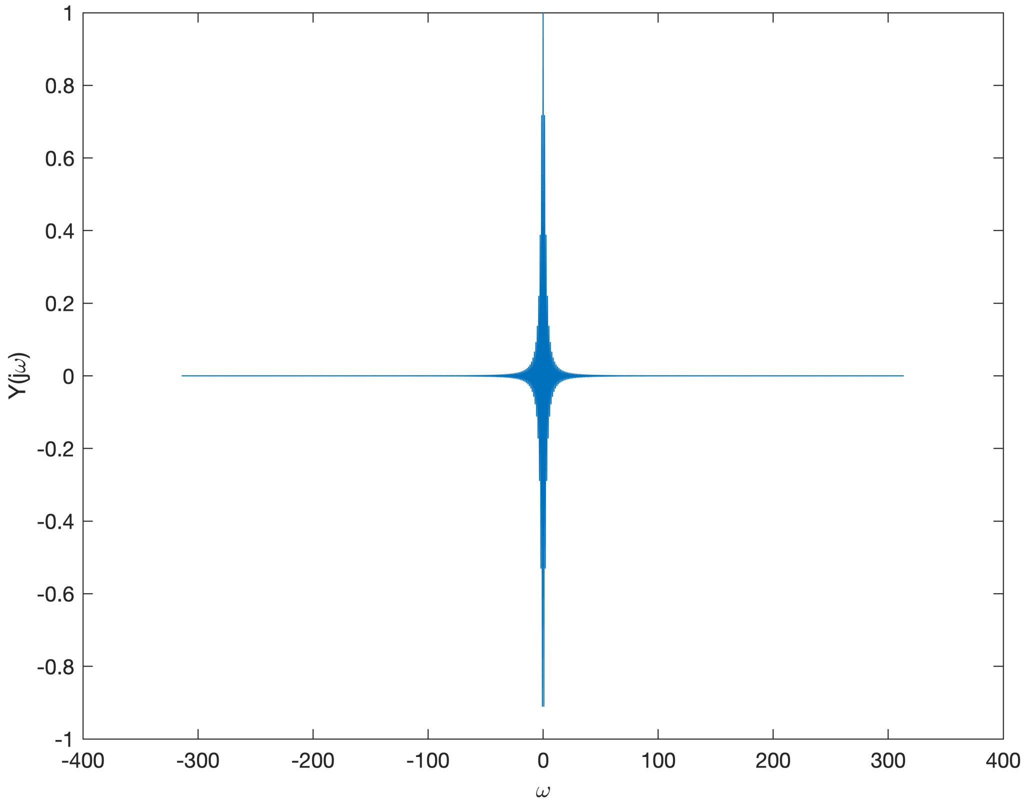


The MATLAB script is shown below.

```
tau = 0.01;
T = 10;
N = T / tau;
t = (0:tau:T - 0.01);
x = exp(-2 .* abs(t));
y = exp(-2 .* abs(t - 5));

plot(y)
xlabel('N')
ylabel('y')
```

Basic Problem d

The plot for $Y(j\omega)$ with `w` as x-axis is shown below.



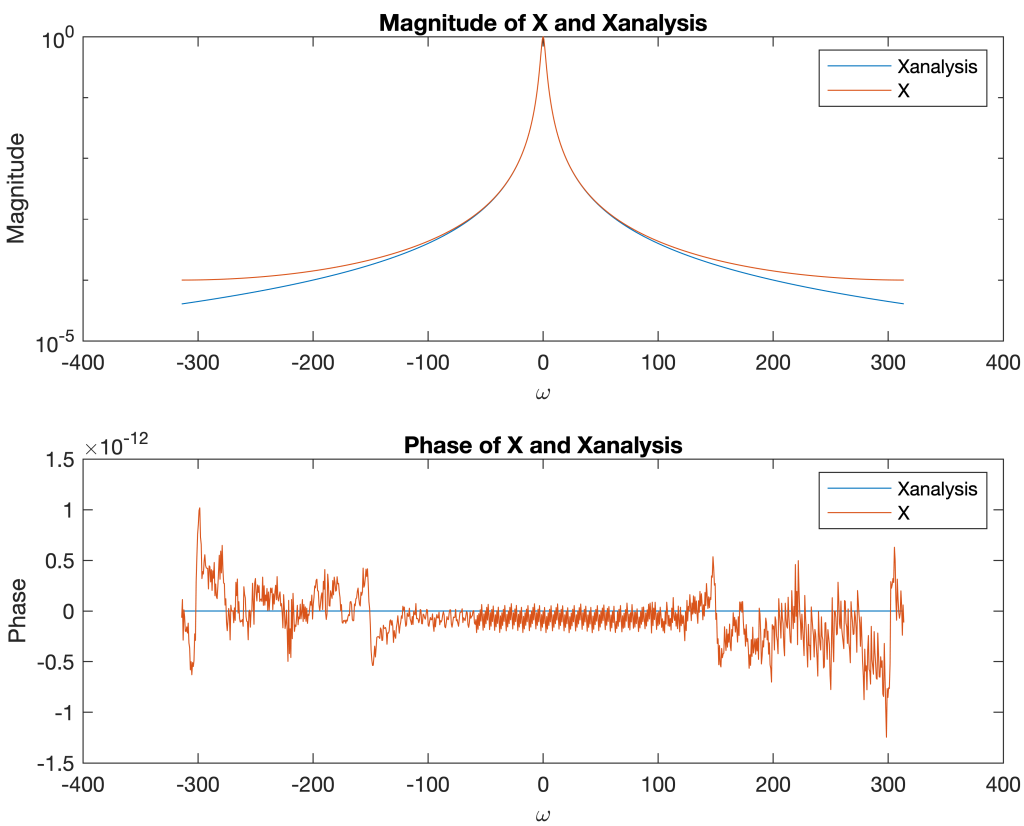
The MATLAB script is shown below.

```
tau = 0.01;
T = 10;
N = T / tau;
t = (0:tau:T - 0.01);
x = exp(-2 .* abs(t));
y = exp(-2 .* abs(t - 5));
Y = fftshift(tau * fft(y));
w = -(pi / tau) + (0:N - 1) * (2 * pi / (N * tau));

plot(w, real(Y))
xlabel('\omega')
ylabel('Y(j\omega)')
```

--

Basic Problem e,f

The plots of $X(j\omega)$ produced by `X = Y .* exp(1j * 5 * w)` and analytic expression of $X(j\omega)$ are shown below.



We can easily see that the magnitude are almost the same, but the phase have a great differences.

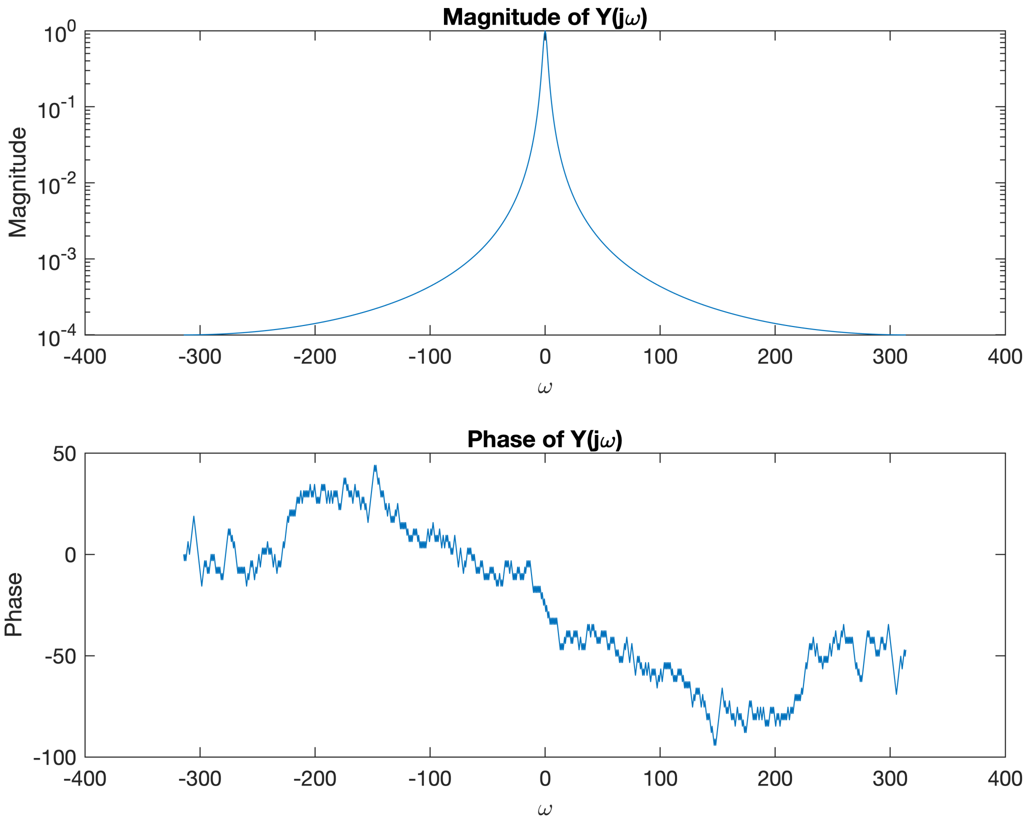
The MATLAB script is shown below.

```
tau = 0.01;
T = 10;
N = T / tau;
t = (0:tau:T - 0.01);
x = exp(-2 .*abs(t));
y = exp(-2 .* abs(t - 5));
Y = fftshift(tau *fft(y));
w = -(pi / tau) + (0:N - 1)* (2 *pi / (N* tau));

phase = exp(1j *5* w);
X = Y .* phase;

Xanalysis = 1 ./ (2 + 1j *w) + 1 ./ (2 - 1j* w);

subplot(2, 1, 1)
semilogy(w, abs(Xanalysis))
hold on
semilogy(w, abs(X))
title('Magnitude of X and Xanalysis')
xlabel('\omega')
ylabel('Magnitude')
legend('Xanalysis', 'X')
subplot(2, 1, 2)
plot(w, angle(Xanalysis))
hold on
plot(w, angle(X))
title('Phase of X and Xanalysis')
```

```
xlabel('\omega')
ylabel('Phase')
legend('Xanalysis', 'X')
```

Basic Problem g

The plot of $Y(j\omega)$ is shown below.



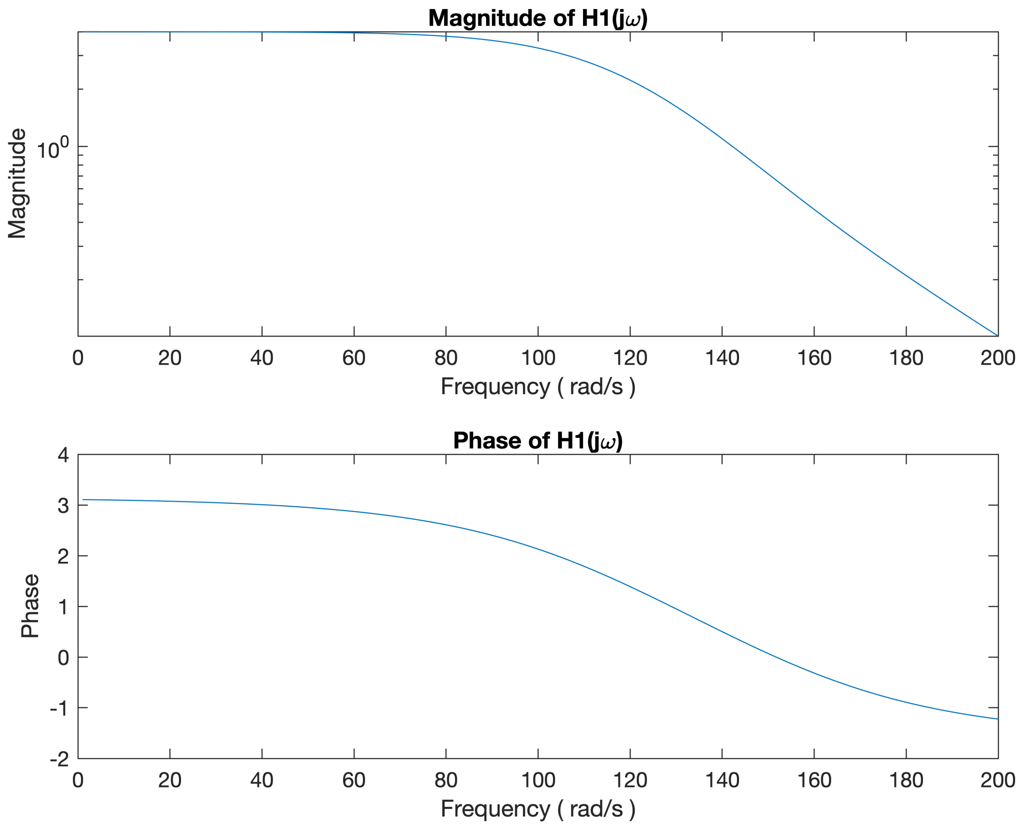
Compare with the plot of $X(j\omega)$, we can easily see that the magnitude are the same since the changes in phase will not affect mod. However, the phase are completely different.

4.5

Basic Problem a

We can easily get `a1 = [1 1.5 0.5]`, `b1 = [1 -2]`, so `H1 = freqs(b1, a1)`.

The plot is shown below.



The MATLAB script is shown below.

```
b1 = [1 -2];
a1 = [1 3/2 1/2];
H1 = freqs(b1, a1);

subplot(2, 1, 1)
semilogy(abs(H1))
title('Magnitude of H1(j\omega)')
xlabel('Frequency ( rad/s )')
ylabel('Magnitude')
subplot(2, 1, 2)
plot(unwrap(angle(H1)))
title('Phase of H1(j\omega)')
xlabel('Frequency ( rad/s )')
ylabel('Phase')
```

Basic Problem b

Since we have already know that `b1 = [2 -1]` and `a1 = [1 1.5 0.5]`, we then use `[r1, p1] = residue(a1, b1)` to compute `r1` and `p1`. The result is `r1 = [6 -5]` and `p1 = [-1 -0.5]`.

So we get $H_1(j\omega) = \dfrac{6}{j\omega + 1} + \dfrac{-5}{j\omega + 0.5}$.

The MATLAB script is shown below.

```
b1 = [1 -2];
a1 = [1 3/2 1/2];
[r1, p1] = residue(b1, a1);
```

Basic Problem c

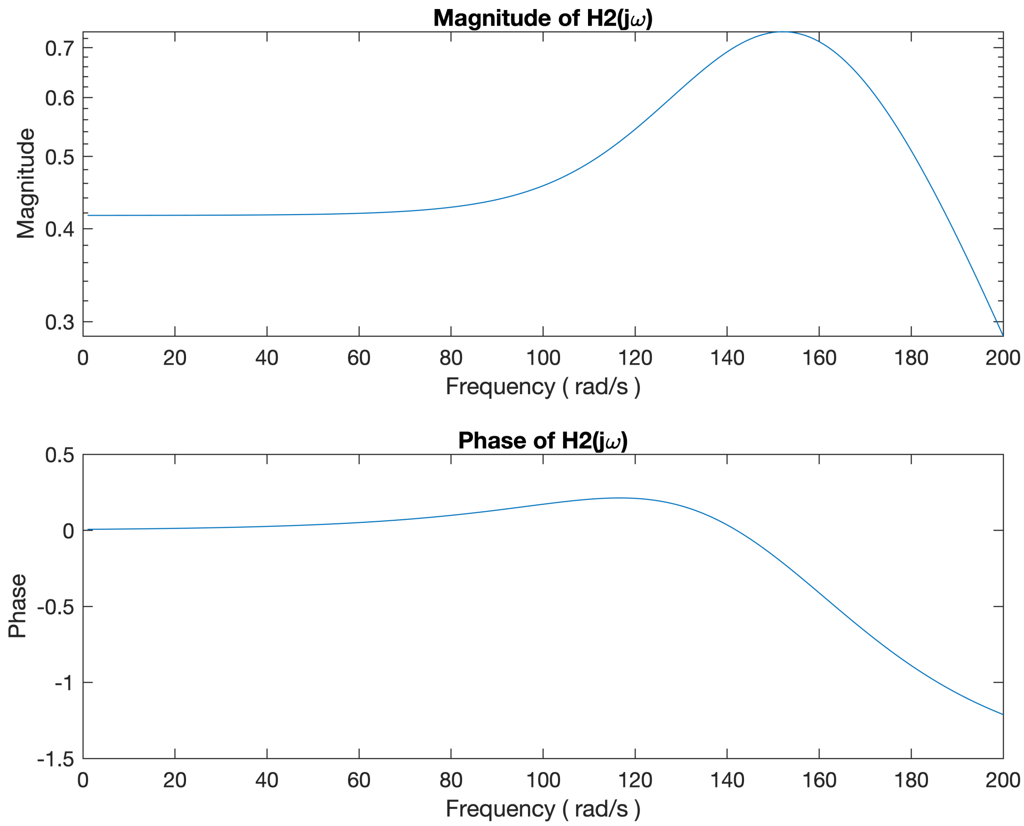
From problem b, where we have already get $H_1(j\omega) = \dfrac{6}{j\omega + 1} + \dfrac{-5}{j\omega + 0.5}$, we can easily get $h_1(t) = (6e^{-t} - 5e^{-0.5t})u(t)$.

And it is easy to see that $H_1(j\omega)$ is integrable.

Intermediate Problem d

We can easily get `a2 = [1 7 16 12]`, `b2 = [3 10 5]`, so `H2 = freqs(b2, a2)`.

The plot is shown below.



The MATLAB script is shown below.

```
b2 = [3 10 5];
a2 = [1 7 16 12];

H2 = freqs(b2, a2);
subplot(2, 1, 1)
semilogy(abs(H2))
title('Magnitude of H2(j\omega)')
xlabel('Frequency ( rad/s )')
ylabel('Magnitude')
subplot(2, 1, 2)
plot(unwrap(angle(H2)))
title('Phase of H2(j\omega)')
xlabel('Frequency ( rad/s )')
ylabel('Phase')
```

Intermediate Problem e

Using the same method as in problem a, `[r2,p2] = residue(b2,a2)`, and `a2 = [1 7 16 12]`, `b2 = [3 10 5]`. We can get `r2 = [2 1 -3]` and `p2 = [-3 -2 -2]`.

So $H_2(j\omega) = \dfrac{2}{j\omega+3} + \dfrac{1}{j\omega+2} + \dfrac{-3}{j\omega+2}$

The MATLAB script is shown below.

```
b2 = [3 10 5];
a2 = [1 7 16 12];
[r2,p2] = residue(b2,a2);
```

Intermediate Problem f

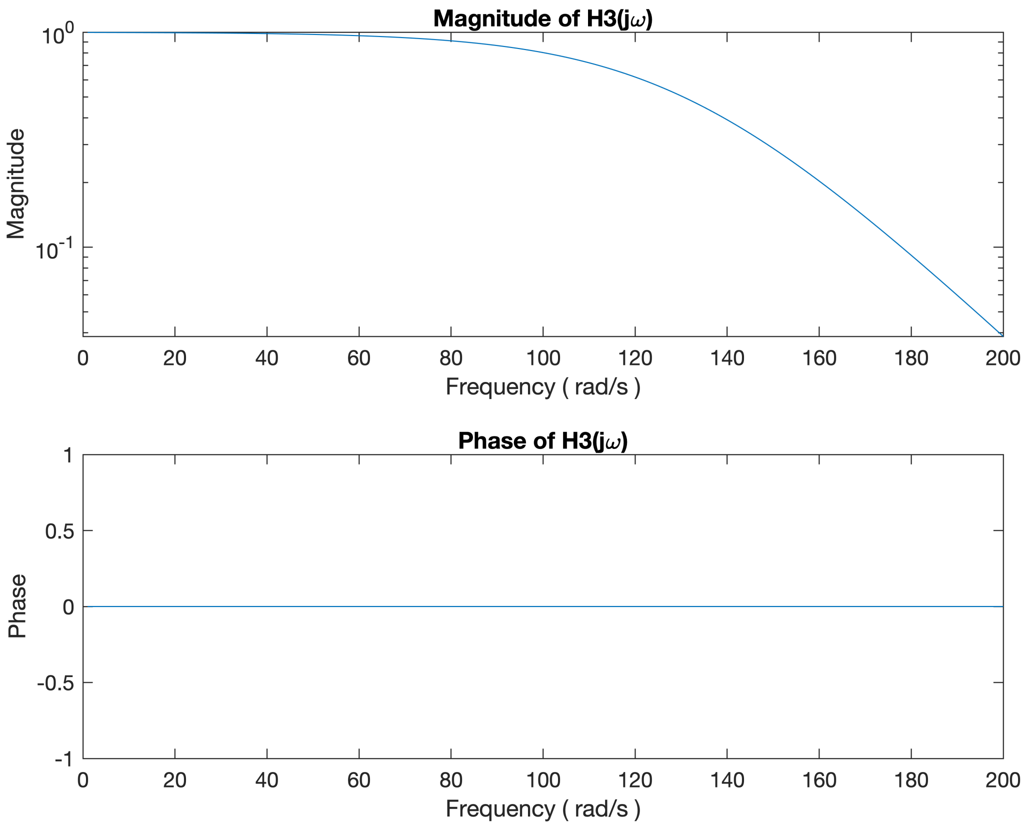
From problem e, where we have already get $H_2(j\omega) = \dfrac{2}{j\omega+3} + \dfrac{1}{j\omega+2} + \dfrac{-3}{j\omega+2}$, we can easily get $h_2(t) = \left(2e^{-3t} + e^{-2t} - 3e^{-2t}\right)u(t)$.

And it is easy to see that $H_2(j\omega)$ is integrable.

Advanced Problem g

We can easily get `a3 = [1 0 -4]`, `b3 = [0 0 -4]`, so `H3 = freqs(b3, a3)`.

The plot is shown below.



The MATLAB script is shown below.

```
b3 = [0 0 -4];
a3 = [1 0 -4];

H3 = freqs(b3, a3);
subplot(2, 1, 1)
semilogy(abs(H3))
title('Magnitude of H3(j\omega)')
xlabel('Frequency ( rad/s )')
ylabel('Magnitude')
subplot(2, 1, 2)
plot(unwrap(angle(H3)))
title('Phase of H3(j\omega)')
xlabel('Frequency ( rad/s )')
ylabel('Phase')
```

Advanced Problem h

Using the same method as in problem a, `[r3,p3] = residue(b3,a3)`, and `a3 = [1 0 -4]`, `b3 = [0 0 -4]`. We can get `r2 = [-1 1]` and `p2 = [2 -2]`.

So $H_3(j\omega) = \dfrac{-1}{j\omega - 2} + \dfrac{1}{j\omega + 2}$

The MATLAB script is shown below.

```
b3 = [0 0 -4];
a3 = [1 0 -4];
[r3, p3] = residue(b3, a3);
```

Advanced Problem i

From problem e, where we have already get $H_3(j\omega) = \dfrac{-1}{j\omega - 2} + \dfrac{1}{j\omega + 2}$, we can easily get $h_(3t) = \left(-e^{2t} + e^{-2t}\right) u(t)$, then we standardize the expression into $X(j\omega) = \dfrac{4}{4 + \omega^2}$

And it is easy to see that $H_2(j\omega)$ is integrable,and it is causal.

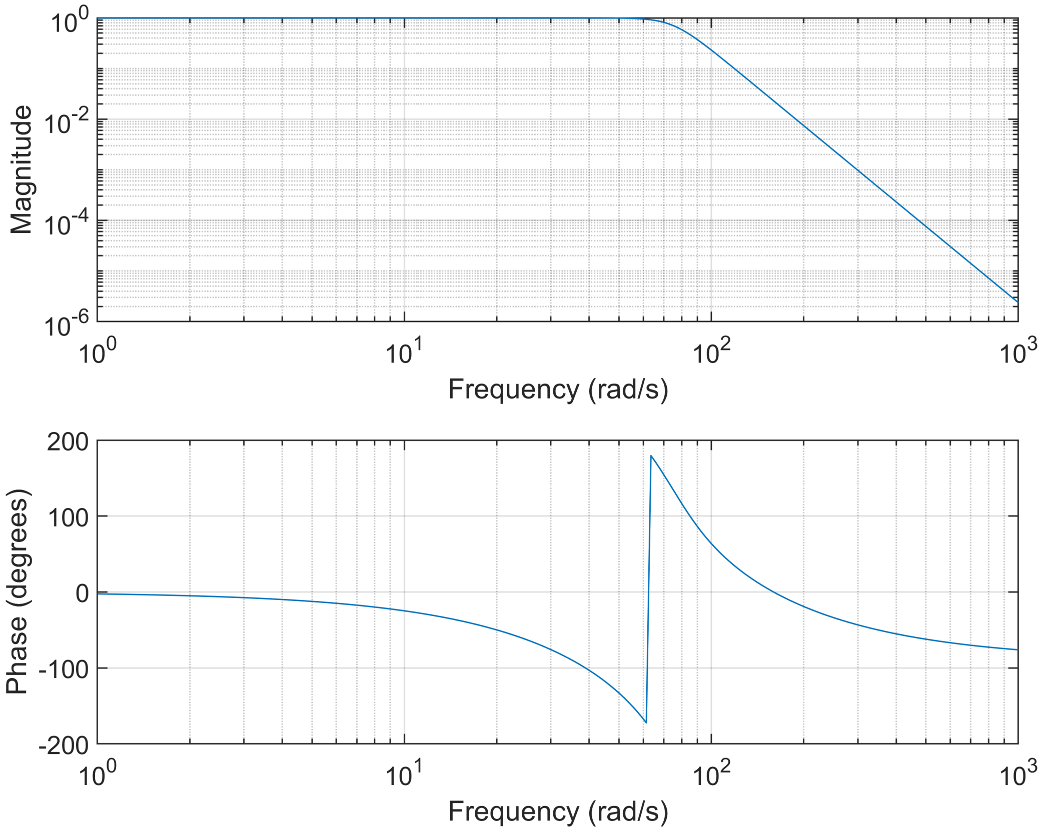
4.6

Basic Problem a

Since the corresponding Morse code of letter Z is dash dash dot dot, the signal is simply `z = [dash dash dot dot]`.

Basic Problem b

The plot of the frequency response of the filter is shown below.

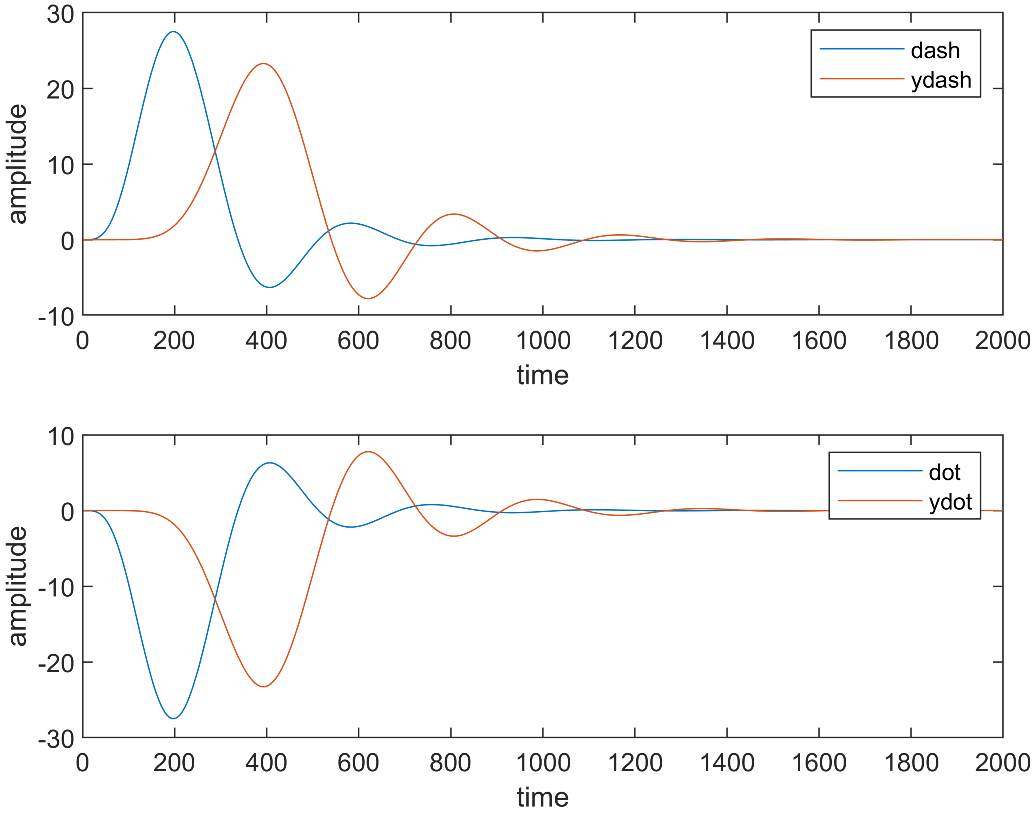


Basic Problem c

The response is obtained with the following lines.

```
ydash = lsim(bf, af, dash, t(1:length(dash)));
ydot = lsim(bf, af, dot, t(1:length(dot)));
```

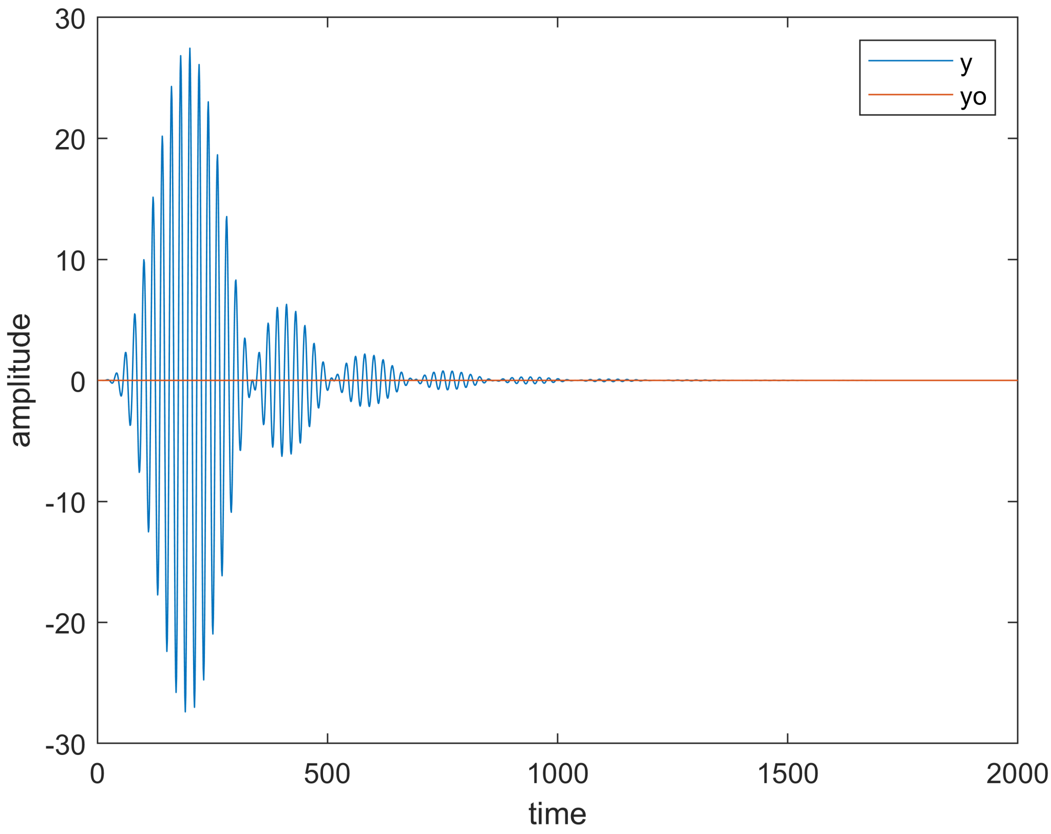
The plot of `dash`, `ydash`, `dot`, `ydot` is shown below.



According to the plot, after applying the filter, the `dot` and `dash` signals are phase shifted, but the amplitude is approximately the same. Hence we can state that their Fourier transforms lie roughly within the passband of the lowpass filter.

Basic Problem d

The plot of `y` and `yo` is shown below.



According to the plot, the response is 0, hence we can state that most of the energy in the Fourier transform of signal `y` will move outside the passband of the filter.

The MATLAB script is shown in the code block below.

```
load ctftmod.mat

y = dash .* cos(2 * pi * f1 .* t(1:length(dash)));
yo = lsim(bf, af, y, t(1:length(dash)));

plot(y)
hold on
plot(yo)
legend('y', 'yo')
xlabel('time')
ylabel('amplitude')
```

Intermediate Problem e

The Fourier transform for a cosine function in form of $\cos(2\pi ft)$ is

$$C(j\omega) = \frac{1}{2}\mathscr{F}\left\{e^{j2\pi ft} + e^{-j2\pi ft}\right\} = \pi[\delta(\omega + 2\pi ft) + \delta(\omega - 2\pi ft)].$$

Similarly, that of a sine function in form of $\sin(2\pi ft)$ is

$$S(j\omega) = \frac{1}{2j}\mathscr{F}\left\{e^{j2\pi ft} - e^{-j2\pi ft}\right\} = \pi j[\delta(\omega + 2\pi ft) - \delta(\omega - 2\pi ft)].$$

Therefore, according to the property of Fourier transform, we have

$$
\begin{aligned}
\mathscr{F}\{m(t)\cos(2\pi f_1 t)\cos(2\pi f_1 t)\} &= \frac{1}{2\pi}M(j\omega) * \frac{1}{2\pi}[C_1(j\omega) * C_1(j\omega)] \\
&= \frac{1}{4}\{M[j(\omega + 4\pi f_1 t)] + 2M(j\omega) + M[j(\omega - 4\pi f_1 t)]\} \\
\mathscr{F}\{m(t)\cos(2\pi f_1 t)\sin(2\pi f_1 t)\} &= \frac{1}{2\pi}M(j\omega) * \frac{1}{2\pi}[C_1(j\omega) * S_1(j\omega)] \\
&= \frac{j}{4}\{M[j(\omega + 4\pi f_1 t)] - M[j(\omega - 4\pi f_1 t)]\} \\
\mathscr{F}\{m(t)\cos(2\pi f_1 t)\cos(2\pi f_2 t)\} &= \frac{1}{2\pi}M(j\omega) * \frac{1}{2\pi}[C_1(j\omega) * C_2(j\omega)] \\
&= \frac{1}{4}\{M\{j[\omega + 2\pi(f_1 + f_2)t]\} + M\{j[\omega - 2\pi(f_1 + f_2)t]\} \\
&+ M\{j[\omega + 2\pi(f_1 - f_2)t]\} + M\{j[\omega - 2\pi(f_1 - f_2)t]\}\}
\end{aligned}
$$

Another Fourier transform is

$$
\begin{aligned}
\mathscr{F}\{m(t)\sin(2\pi f_1 t)\sin(2\pi f_1 t)\} &= \frac{1}{2\pi}M(j\omega) * \frac{1}{2\pi}[S_1(j\omega) * S_1(j\omega)] \\
&= -\frac{1}{4}\{M[j(\omega + 4\pi f_1 t)] - 2M(j\omega) + M[j(\omega - 4\pi f_1 t)]\}
\end{aligned}
$$

Intermediate Problem f

According to the experiment in d, and referring to the frequency response of the lowpass filter given, if the signal dash or dot is modulated into a trigonometric signal, say cosine or sine function, the result signal would not be able to pass the filter.

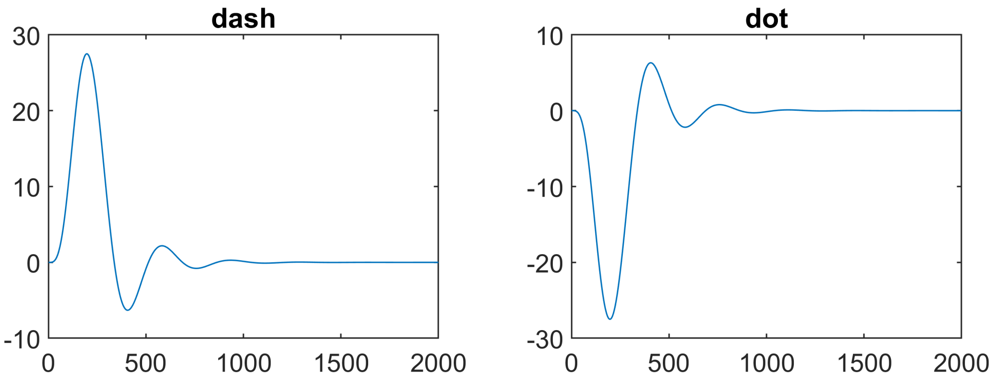
However, according to the derivation in e, we can find that if the message signal is multiplied by cosine or sine squared, there would be a low frequency term $M(j\omega)$ in the Fourier transform of the result signal, this term is able to pass the lowpass filter. On the contrary, if the message signal is multiplied by one cosine function and a sine function, there will not be such low frequency term.

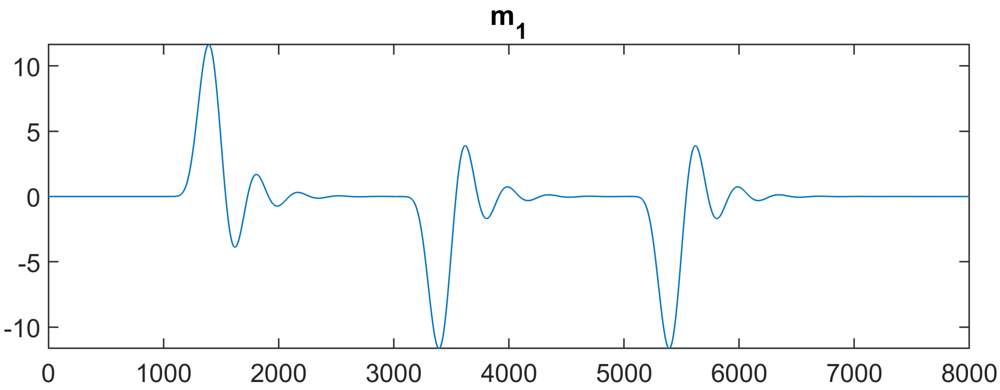
Therefore, we may multiply a cosine function with frequency f_1 to the signal $x(t)$, then apply the lowpass filter to $x(t)\cos(2\pi f_1 t)$, then we can find the message we want.

Such method can be implemented with the following codes.

```
load ctftmod.mat

y1 = x .* cos(2 * pi * f1 .* t(1:length(x)));
m1 = 2 * lsim(bf, af, y1, t(1:length(x)));

subplot(2, 2, 1)
plot(dash)
title('dash')
subplot(2, 2, 2)
plot(dot)
title('dot')
subplot(2, 2, [3, 4])
plot(m1)
title('m_1')
```

The plot is shown below.





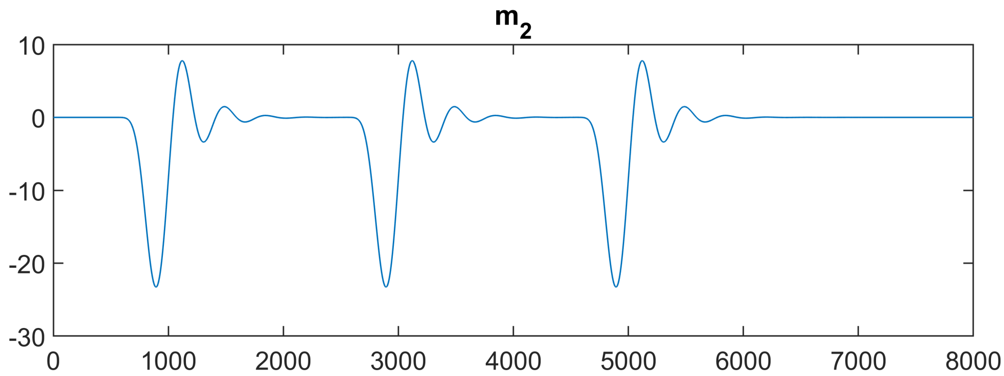
Compared to the wave form of `dash` and `dot`, m_1 is "dash dot dot", which represents letter D.

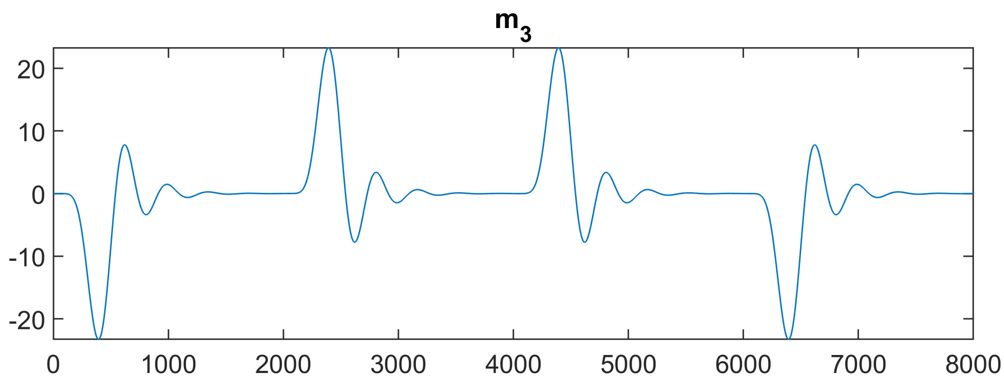
Intermediate Problem g

Similarly, we can obtain the message hided in $m_2(t)$ and $m_2(t)$. The MATLAB script is shown in the code block below.

```
load ctftmod.mat

y2 = x .* sin(2 * pi * f2 .* t(1:length(x)));
m2 = 2 * lsim(bf, af, y2, t(1:length(x)));
subplot(2, 1, 1)
plot(m2)
title('m_2')

y3 = x .* sin(2 * pi * f1 .* t(1:length(x)));
m3 = 2 * lsim(bf, af, y3, t(1:length(x)));
subplot(2, 1, 2)
plot(m3)
title('m_3')
```

The plot of signal $m_2(t)$ and $m_3(t)$ is shown below.





Therefore, the message hided is "DSP".

The future of technology lies in **D**igital **S**ignals **P**rocessing!