EEE 321

Report: Signals and Systems Lab 4

Yigit Turkmen – 22102518

April 7, 2024

Introduction:

This report examines the lab 4 on Fourier series representation of continuous-time signals. Utilizing MATLAB, the assignment explores the computation of Fourier series coefficients, the effects of time-domain operations on these coefficients, and the analysis of a second-order linear time-invariant system.

Part 1

Part 1.1 Implementing Fourier Series Analysis

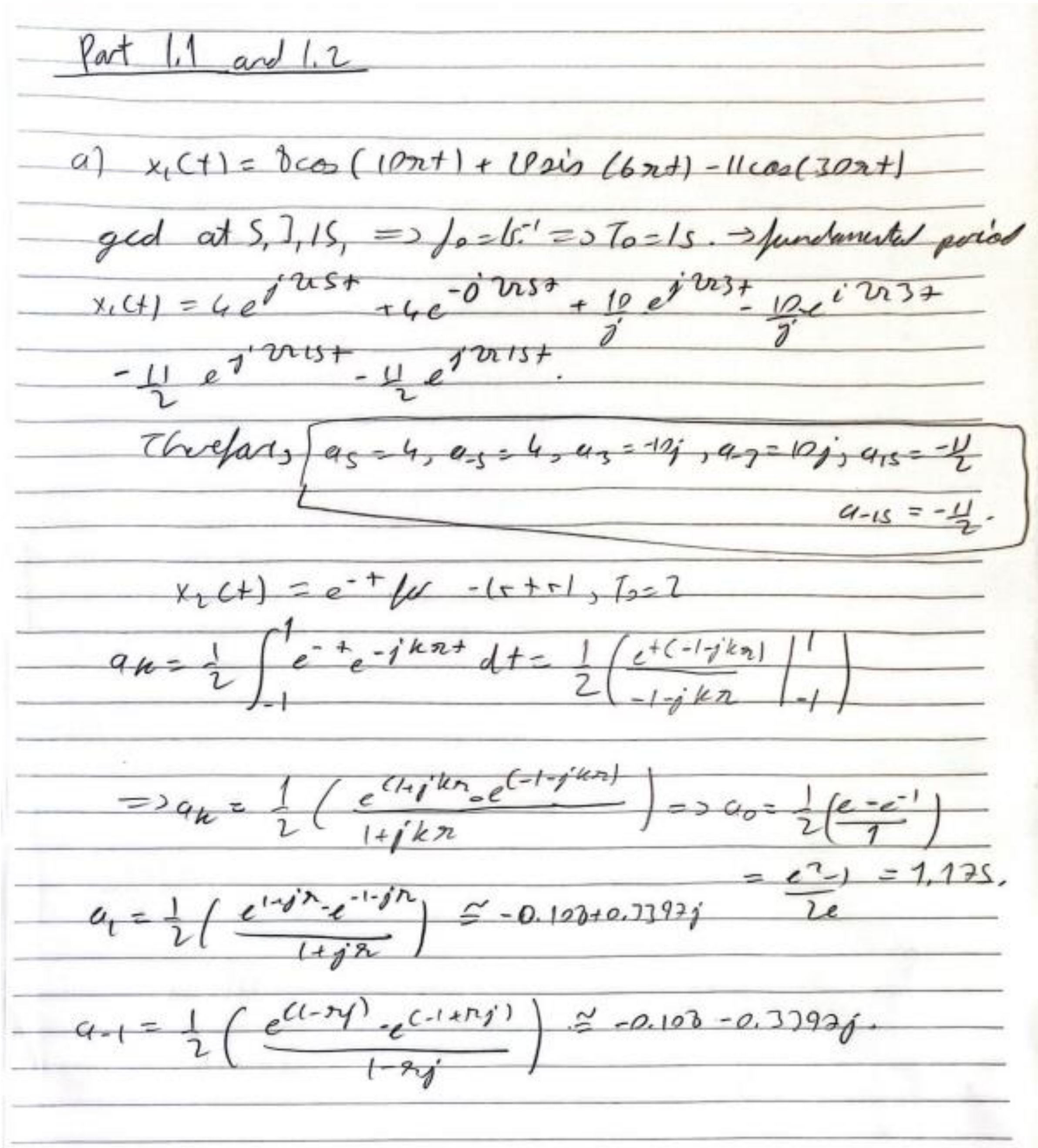
Related code for the function:

```
function [fsCoeffs] = FSAnalysis(x, k)
fsCoeffs = zeros(1,2*k+1);
stepSize = 0.001;
sum = 0;
T = (length(x) - 1) * stepSize;
k_arr = -k : 1 : k;
t = 0:stepSize:T;

for n = 1:1:length(fsCoeffs)
    k_used =  k_arr(n);
    for currentStep = 0 : stepSize : T
    x_index = int64(currentStep * (length(x)-1)/T + 1);
    sum = sum + ( x(x_index) * exp(-1i*k_used*2*pi*currentStep/T) * stepSize);
    end
    fsCoeffs(n) = sum / T;
    sum = 0;
end
```

Part 1.2 Testing the Function

See the image and figures below for this part. Then see the answers and comments. Note that k index means the k + 31. So if you see 1 for k index you should perceive it as the k = -30 for that.



Part 1.1 and 1.2

a) $x_1(t) = 8\cos(10\pi t) + 10\sin(6\pi t) - 11\cos(30\pi t)$

gcd at 5, 3, 15, $\Rightarrow f_0 = 1s^{-1} \Rightarrow T_0 = 1s$. → fundamental period

$x_1(t) = 4e^{j2\pi 5t} + 4e^{-j2\pi 5t} + \frac{10}{j}e^{j2\pi 3t} - \frac{10}{j}e^{j2\pi 3t}$

$-\frac{11}{2}e^{j2\pi 15t} - \frac{11}{2}e^{j2\pi 15t}$.

Therefore, $\boxed{a_5 = 4,\ a_{-5} = 4,\ a_3 = -10j,\ a_{-3} = 10j,\ a_{15} = -\frac{11}{2},\ a_{-15} = -\frac{11}{2}}$.

$x_2(t) = e^{-t}$ for $-1 \le t \le 1$, $T_0 = 2$

$$a_k = \frac{1}{2}\int_{-1}^{1} e^{-t}e^{-jk\pi t}\,dt = \frac{1}{2}\left(\frac{e^{t(-1-jk\pi)}}{-1-jk\pi}\Big|_{-1}^{1}\right)$$

$$\Rightarrow a_k = \frac{1}{2}\left(\frac{e^{(1+jk\pi)} - e^{(-1-jk\pi)}}{1+jk\pi}\right) \Rightarrow a_0 = \frac{1}{2}\left(\frac{e - e^{-1}}{1}\right) = \frac{e^2-1}{2e} = 1.175.$$

$$a_1 = \frac{1}{2}\left(\frac{e^{1+j\pi} - e^{-1-j\pi}}{1+j\pi}\right) \cong -0.108 + 0.3397j$$

$$a_{-1} = \frac{1}{2}\left(\frac{e^{(1-\pi j)} - e^{(-1+\pi j)}}{1-\pi j}\right) \cong -0.108 - 0.3397j.$$

Picture 1: Part 1.2

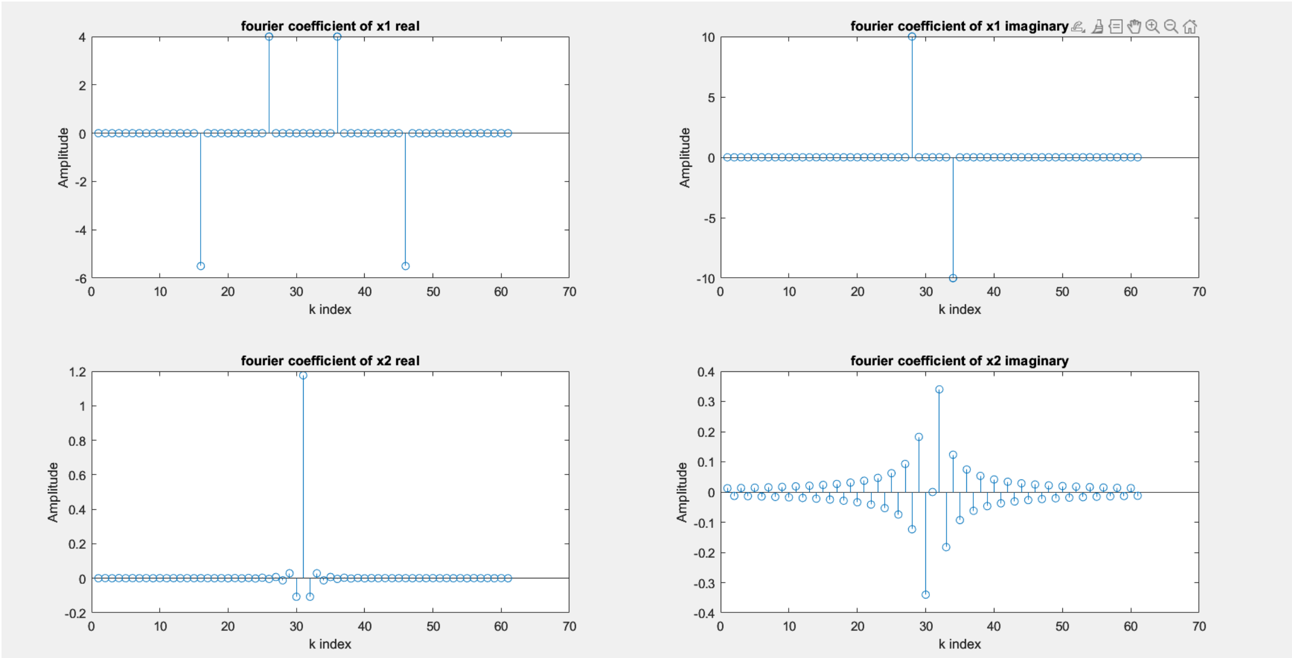


Figure 1: Plots of fourier coefficients of x1 and x2 with real and imaginary parts respectively.

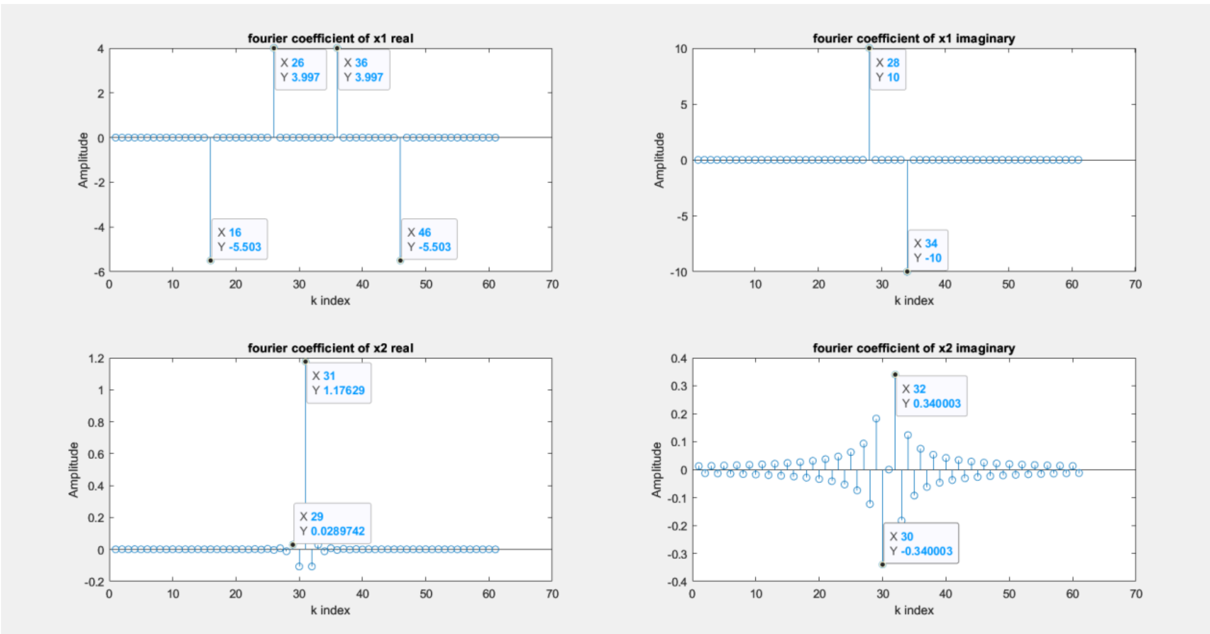


Figure 2: Plots of fourier coefficients of x1 and x2 with real and imaginary parts respectively with the some values highlited.

Table for matlab values vs calculated values:

	K = 3	K = -3	K = 5	K = -5	K = 15	K = -15
x1 calculated	**-10j**	**10j**	**4**	**4**	**-5.5**	**-5.5**
x1 matlab	**-10j**	**10j**	**3.997**	**3.997**	**-5.503**	**-5.503**

	k = 0	K = 1	K = -1
x2 calculated	**1.175**	**-0.108 + 0.3397j**	**-0.108 - 0.3397j**
x2 matlab	**1.17629**	**-0.107138 + 0.34j**	**-0.107138 – 0.34j**

Comment: So from these tables we can easily observe that calculated and matlab values are really close to each other which yields almost no error.

Parseval part:

X1_integral is the integral result.

And parseval is the summation result.

So wee see 292.5090 – 292.5185 = 0.0095 error in value. You can check these values simply using command window after running my code.

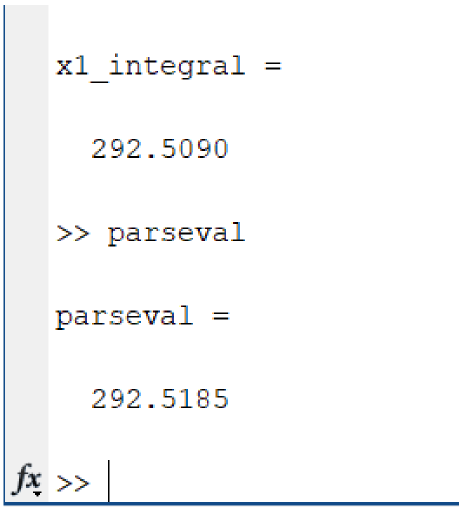


Figure 3: Parseval part

Part 2

See the figures below for this part. Then see the answers and comments. Note that k index means the k + 31. So if you see 1 for k index you should perceive it as the k = -30 for that index.

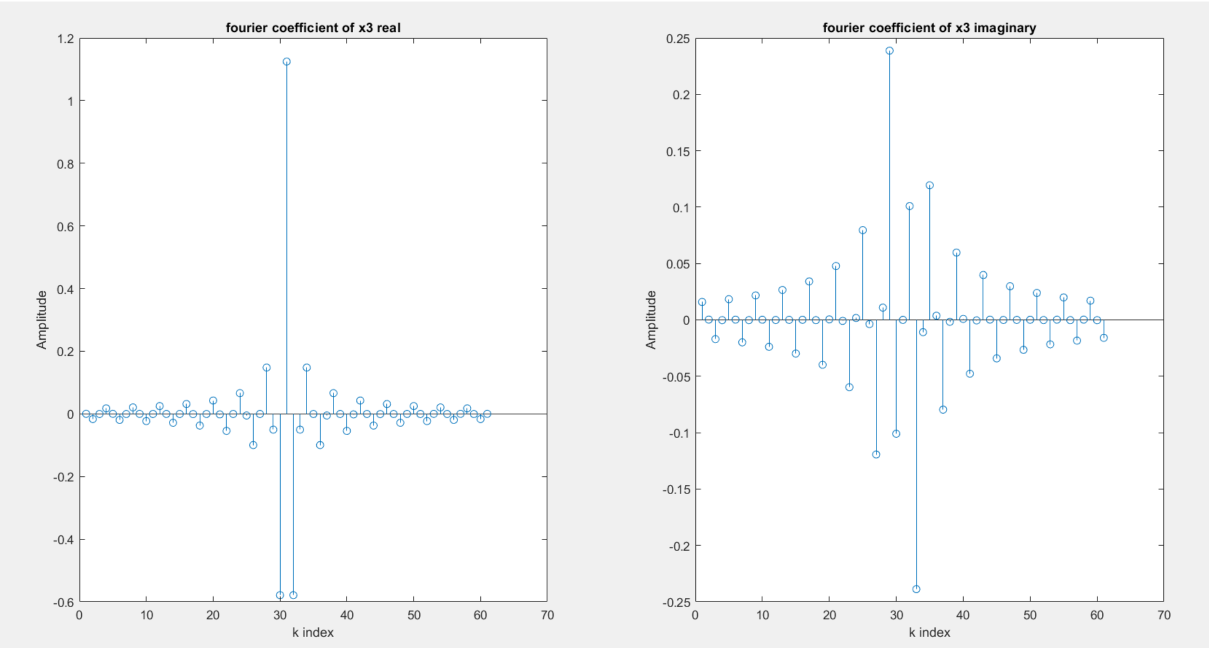


Figure 4: Plots of fourier coefficients of x3, for real and imaginary parts respectively

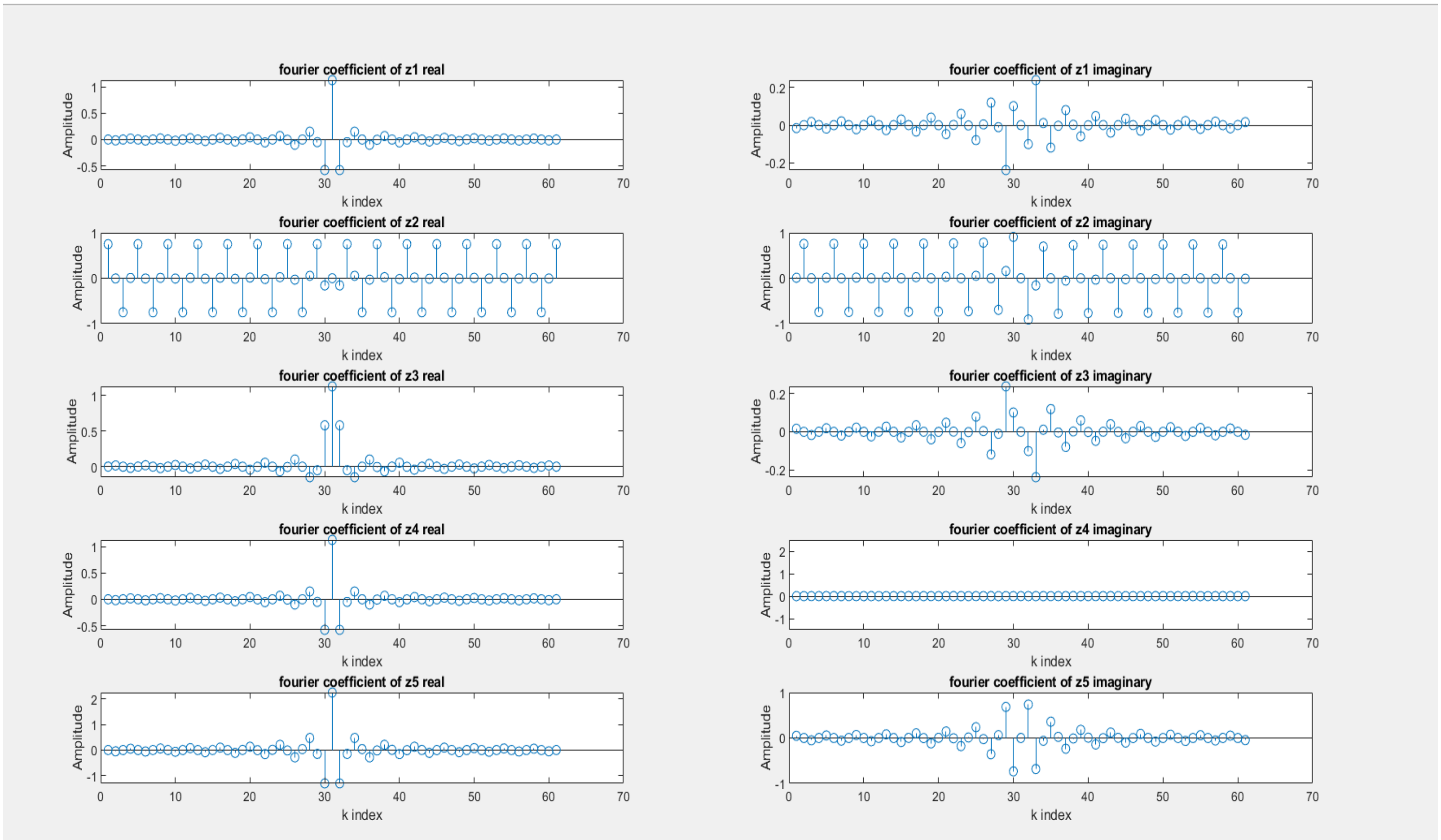


Figure 5: Plots of fourier coefficients z1 to the z5 for real and imaginary parts

Comments on x1 vs z1:

So we see that the real part stays same but imaginary part is taken symmetry according to the x axis. It is because when t is taken as -t the exponential part becomes (-) version of the normal one which yields the cosine part of the term stays same since cos(negative) = cos(positive), however imaginary part becomes the negative of its normal version since i*sin(negative) = -i*sin(positive). Therefore, we can observe these changes when we compare the plots of fourier coefficients of x1 and z1.

Comments on x1 vs z2:

When derivative is taken $jk\omega_0 a_k = jk\frac{2\pi}{T}a_k$ this applies to the a_k, so from here we see amplication of some of the signals since we multiply each term with “j” and “k” and “w0”. Also if we note that since the k is 0 for a_0 we should observe new fourier coefficients should be zero which is the case in here. Therefore, we can observe these changes when we compare the plots of x1 and z2.

Comments on x1 vs z3:

Now here we have $a_k e^{-jk\omega_0 t_0} = a_k e^{-jk(2\pi/T)t_0}$ where, w0 is 2pi / 4, and t0 = -2 => exponential term is $e^{jk\pi}$ so we multiply each old a_k with this exponential term to obtain new fourier coefficients. So when k is odd, this term becomes “-1”. When k is even this term becomes “1”. So we should observe these changes in z3 when we compare it with x1. And we see these results when we examine the each plots. To see it easily, just look at the center of the plots where k = 0 which means k_index = 31 then examine it by looking right or left where symmetry is taken according to the x axis.

Comments on x1 vs z4:

In this part we have $x_e(t) = \mathcal{E}v\{x(t)\} \quad [x(t) \text{ real}] \qquad \mathcal{R}e\{a_k\}$

As a result we see that real part of the z4 is same with the x3 signal. And imaginary part of z4 is very close to zero which is the expected result.

Comments on x1 vs z5:

When we multiply x3 with itself, we see convolution for a_k with itself for new a_k terms. So, when a function is squared, its Fourier coefficients do not simply double or square themselves. Instead, they interact with each other to create a new set of coefficients through the process of convolution, which typically results in a more complex frequency spectrum with additional frequency components not present in the original function. Therefore, we see this amplication adjustments when comparing the plots of x1 and z5. Real parts of fourier coefficients of z5 seems higher amplitude of the normal one . However, imaginary parts of are the higher in amplitude wise and also the negative of the normal ones which are the expected results.

Part 3

3.1 A Second-Order System

For this part see the image below.

Part 3

Part 3.1

$$M\frac{d^2y(t)}{dt^2} + c\frac{dy(t)}{dt} + ky(t) = f(t)$$

$y(t) \to b_k$, $f(t) \to a_k$

$$\frac{dy(t)}{dt^2} \to jk\omega_0 b_k$$

$$\frac{d^2y(t)}{dt^2} \to j^2k^2\omega_0^2 b_k = -k^2\omega_0^2 b_k$$

$$\Rightarrow M(-k^2)\omega_0^2 b_k + cjk\omega_0 b_k + kb_k = a_k.$$

Frequency response part:

$$M(j\omega)^2 Y(j\omega) + c(j\omega)Y(j\omega) + kY(j\omega) = F(j\omega)$$

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)}, \quad Y(j\omega)(-M\omega^2 + cj\omega + k) = F(j\omega)$$

$$\Rightarrow H(j\omega) = 1/(-M\omega^2 + cj\omega + k).$$

small k

$$\Rightarrow H(j\omega_0 k) = \frac{b_k}{a_k} = \frac{1}{(-Mk^2\omega_0^2) + (cjk\omega_0) + k}$$

Picture 2: Part 3.1

3.2 Implementation of the Second-Order System

For this part see the image below and the figures related to the plots. Then see the answers and/or comments for this part. Note that k index means the k + 31. So if you see 1 for k index you should perceive it as the k = -30 for that index.

Part 3.2

$$y[n] \xrightarrow{\text{derivative}} x[n] - x[n-1]$$

$$y[n] \xrightarrow{\text{second derivative}} x[n] - x[n-1] - x[n-1] + x[n-2]$$

$$\rightarrow x[n] - 2x[n-1] + x[n-2] \ \text{\} second derivative.}$$

$$\Rightarrow y[n] = \frac{\left(x_3[n] + (2m+c)\,y[n-1] + (-m)\,y[n-2]\right)}{m+c+k}$$

Picture 3: Part 3.2

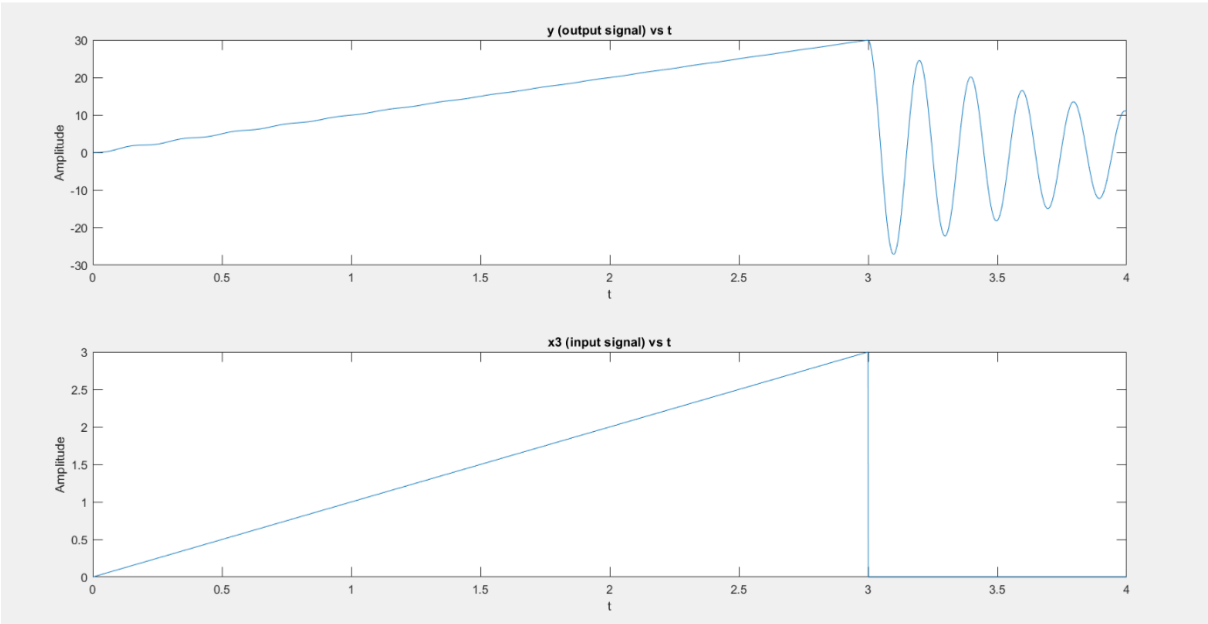


Figure 6:Plots of input signal (y) vs t and output signal (x3) vs t

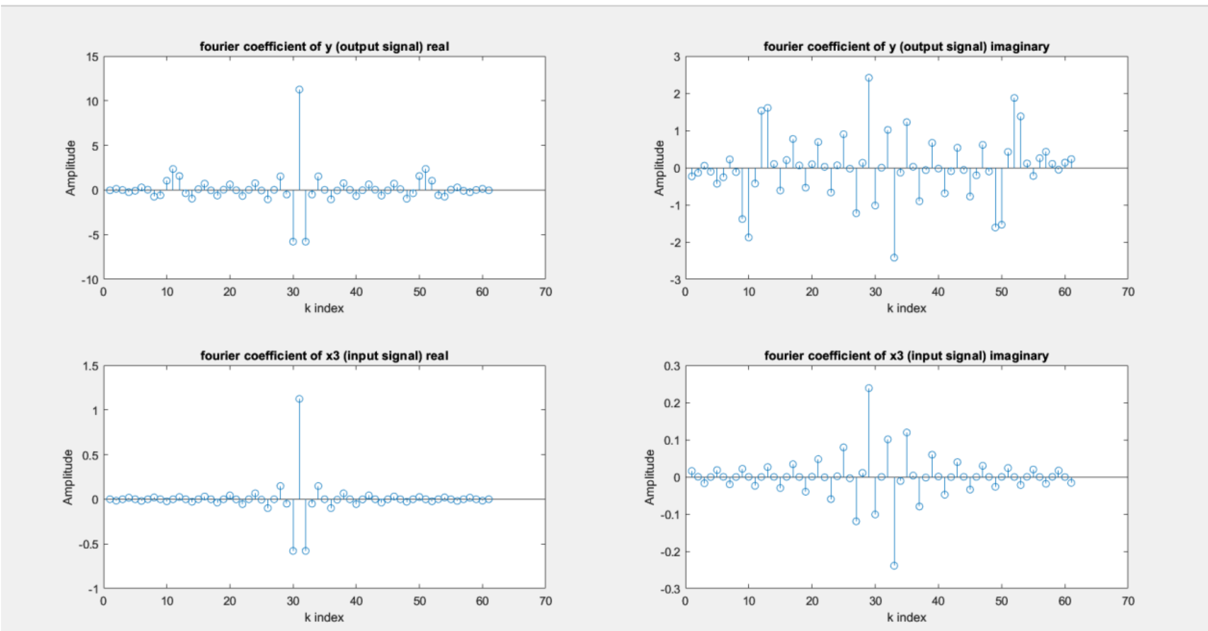


Figure 7: Plots of the fourier coefficients of input and output signals with real and imaginary parts

Comments on the effects of this system on the Fourier series coefficients of the input:

In analyzing the effects of the given second-order system on the Fourier series coefficients of the input signal $x3(t)$, the system's characteristics—defined by mass M, damping coefficient c, and stiffness k—modulate the amplitude and phase of each frequency component of the input signal. Specifically:

- Amplitude Modulation: The system acts as a frequency-dependent filter, amplifying or attenuating the amplitude of the Fourier coefficients based on its frequency response. Lower frequencies might pass through with less attenuation, while higher frequencies could be significantly reduced due to damping.
- Phase Shift: The phase of the Fourier coefficients is shifted in a frequency-dependent manner, affecting the alignment and shape of the output waveform in time.
- Selective Frequency Response: Resonance at the system's natural frequency can cause substantial amplification of the corresponding Fourier component, with the extent of amplification controlled by the damping factor.

Thus, the system modifies the input signal's spectral content by changing the magnitude and phase of its Fourier series coefficients, reflecting the system's physical properties and its response to different frequencies.

Whole codes for the Assignment:

```
function [fsCoeffs] = FSAnalysis(x, k)
fsCoeffs = zeros(1,2*k+1);
stepSize = 0.001;
sum = 0;
T = (length(x) - 1) * stepSize;
k_arr = -k : 1 : k;
t = 0:stepSize:T;

for n = 1:1:length(fsCoeffs)
    k_used =  k_arr(n);
    for currentStep = 0 : stepSize : T
    x_index = int64(currentStep * (length(x)-1)/T + 1);
    sum = sum + ( x(x_index) * exp(-1i*k_used*2*pi*currentStep/T) * stepSize);
    end
    fsCoeffs(n) = sum / T;
    sum = 0;
end
```

“FSA.m”

```
T1 = 1;
Ts = 0.001;
t1 = 0:Ts:T1;
k = 30;
x1 = 8*cos(10*pi*t1) + 20*sin(6*pi*t1) - 11*cos(30*pi*t1);
x1_fourier = FSA(x1,k);
T2 = 2;
t2 = -T2/2:Ts:T2/2-Ts;

x2 = exp(-t2);
N2 = length(x2);
x2 = [x2((N2)*0.5:N2),x2(1:(N2)*0.5)];
x2_fourier = FSA(x2,k);


subplot(2,2,1)
stem(real(x1_fourier))
title("fourier coefficient of x1 real")
ylabel("Amplitude")
xlabel("k index")

subplot(2,2,2)
stem(imag(x1_fourier))
title("fourier coefficient of x1 imaginary")
ylabel("Amplitude")
xlabel("k index")

subplot(2,2,3)
stem(real(x2_fourier))
title("fourier coefficient of x2 real")
ylabel("Amplitude")
xlabel("k index")

subplot(2,2,4)
stem(imag(x2_fourier))
title("fourier coefficient of x2 imaginary")
ylabel("Amplitude")
xlabel("k index")

sum_a = 0;
new = abs(x1_fourier).^2;
for n = 1:length(new)
    sum_a = sum_a + new(n);
end


x1_integral = 0;

for currentStep = 0 : Ts : T1
        x1_index = int64(currentStep * (length(x1)-1)/1 + 1);
        x1_integral = x1_integral +  (abs(x1(x1_index))^2) * Ts;
end

parseval = sum(abs(x1_fourier).^2);
```

“lab4part1.m”

```
T = 4;
Ts = 0.001;
k = 30;
t = 0:Ts:T;

u_t_3 = (t-3 >= 0);
r_t = (t >= 0) .* t;
r_t_3 = (t-3 >= 0) .* (t-3);

x3 = r_t - r_t_3 - 3*u_t_3;
x3_fourier = FSA(x3,k);

figure(1)
subplot(1,2,1)
stem(real(x3_fourier))
title("fourier coefficient of x3 real")
ylabel("Amplitude")
xlabel("k index")

subplot(1,2,2)
stem(imag(x3_fourier))
ylabel("Amplitude")
title("fourier coefficient of x3 imaginary")
xlabel("k index")

z1 = fliplr(x3);
z2 = (t >= 0) - (t-3 >= 0) -3*(t-3==0)/Ts;
z3 = (t-1 >= 0) - (t+2>= 0).* (t+2) - 3*(t+2 >= 0) .* (t+2);
z4 = (x3 + z1) / 2;
z5 = x3.^2;

N3= length(z3);
z3 = [x3(N3*0.5:N3),x3(1:N3*0.5)];

z1_fourier = FSA(z1,k);
z2_fourier = FSA(z2,k);
z3_fourier = FSA(z3,k);
z4_fourier = FSA(z4,k);
z5_fourier = FSA(z5,k);

figure(2)
subplot(5,2,1)
stem(real(z1_fourier))
title("fourier coefficient of z1 real")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,2)
stem(imag(z1_fourier))
title("fourier coefficient of z1 imaginary")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,3)
```

```
stem(real(z2_fourier))
title("fourier coefficient of z2 real")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,4)
stem(imag(z2_fourier))
title("fourier coefficient of z2 imaginary")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,5)
stem(real(z3_fourier))
title("fourier coefficient of z3 real")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,6)
stem(imag(z3_fourier))
title("fourier coefficient of z3 imaginary")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,7)
stem(real(z4_fourier))
title("fourier coefficient of z4 real")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,8)
stem(imag(z4_fourier))
title("fourier coefficient of z4 imaginary")
ylabel("Amplitude")
xlabel("k index")

ylim([-1.5, 2.5])

subplot(5,2,9)
stem(real(z5_fourier))
title("fourier coefficient of z5 real")
ylabel("Amplitude")
xlabel("k index")


subplot(5,2,10)
stem(imag(z5_fourier))
title("fourier coefficient of z5 imaginary")
ylabel("Amplitude")
xlabel("k index")
```

“lab4part2.m”

```
T = 4;
Ts = 0.001;
k = 30;
t = 0:Ts:T;
m = 100;
c = 0.1;
small_k = 0.1;
u_t_3 = (t-3 >= 0);
r_t = (t >= 0) .* t;
r_t_3 = (t-3 >= 0) .* (t-3);

x3 = r_t - r_t_3 - 3*u_t_3;

y = zeros(1,length(x3));
y(1) = 0;
y(2) = x3(2)/100.2;
for n = 3:length(x3)
    y(n) = (x3(n) + (2*m+c)*y(n-1) + (-m)*y(n-2)) / (m+c+small_k) ;

end

y_fo = FSA(y,k);

dy_fo = zeros(1,length(y_fo));
dy2_fo = zeros(1,length(y_fo));
for n = 1:length(y_fo)
    k = ((n-(length(y_fo)+1)/2));
    dy_fo(n) = y_fo(n) * (1i*k*2*pi)/T;
end

for n = 1:length(y_fo)
    k = ((n-(length(y_fo)+1)/2));
    dy2_fo(n) = y_fo(n) * (-1)*(k^2)*4*(pi^2)/(T^2);
end
x3_fo = m * dy2_fo + c*dy_fo + small_k * y_fo;
x3_foo = FSA(x3,k);

figure(1)
subplot(2,1,1)
plot(t,y);
xlabel("t")
ylabel("Amplitude")
title("y (output signal) vs t")

subplot(2,1,2)
plot(t,x3)
xlabel("t")
ylabel("Amplitude")
title("x3 (input signal) vs t")

figure(2)
subplot(2,2,1)
stem(real(y_fo))
ylabel("Amplitude")
title("fourier coefficient of y (output signal) real")
```

```
xlabel("k index")

subplot(2,2,2)
stem(imag(y_fo))
ylabel("Amplitude")
title("fourier coefficient of y (output signal) imaginary")
xlabel("k index")

subplot(2,2,3)
stem(real(x3_foo))
ylabel("Amplitude")
title("fourier coefficient of x3 (input signal) real")
xlabel("k index")

subplot(2,2,4)
stem(imag(x3_foo))
ylabel("Amplitude")
title("fourier coefficient of x3 (input signal) imaginary")
xlabel("k index")
```

“lab4part3_2.m