SP Lab-1 Report

Name: Vedant & Sarvesh
Table number: 8
Roll number: 2023112012, 2023102039

1.1 : Symbolic variables and Integration in MATLAB

(i) Defining Symbolic Variable

A symbolic variable is used to hold an expression instead of just a numeric value. The symbolic toolbox can reason about the expressions, such as finding the solution to equations, or such as doing calculus.

Why use Symbolic Variables?

Symbolic variables are not restricted to integers or floating point numbers, so the toolbox can reason about irrational values with full theoretic precision. Symbolic numbers can have a higher precision than hardware numbers so you can compute to more decimal places.

$$syms\ t;$$

Above command is used to define the symbolic variable 't'.

(ii) Built-in function for Integration

$$expr\ =\ t * sin(t);$$
$$F\ =\ int\,(expr, t,\ a,\ b);$$

Here, we define the expression which we want to integrate wrt t as expr. $int()$ is the built-in command for numerical integration. It has following parameters:

expr ⟶ function to be integrated wrt t
T ⟶ symbolic variable for the symbolic expression
a ⟶ lower limit for the definite integral
b ⟶ upper limit for the definite integral

```
syms t;
% syms is used to define the integral variable.
expr= t*sin(t);
% expr is the expression which is integrated wrt t in the below line.
F= int(expr,t,[0 1]);
% int is the function to integrate expr wrt t from 0 to 1
F
% displays F in the Command Window
```

F = $\sin(1) - \cos(1)$

1.2 : Finding Fourier series coefficients

For any periodic signal $x(t)$, Fourier series coefficients can be calculated by using:

$$a_k = \frac{1}{T}\int_{\langle T\rangle} x(t)e^{-jk\omega_0 t}\, dt, \qquad k \in \mathbb{Z}$$

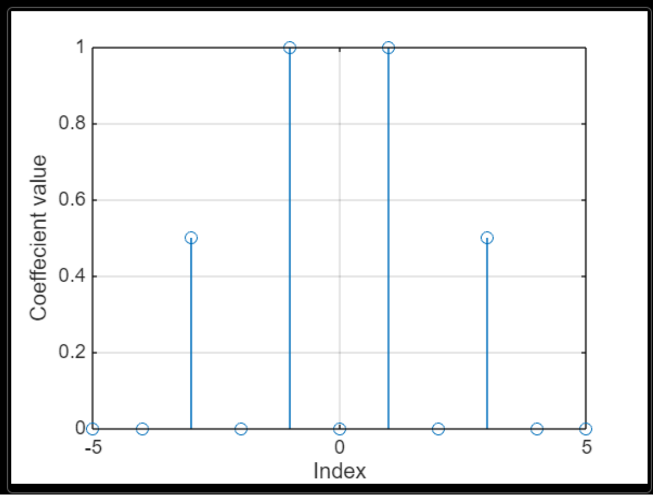
(a) Matlab Function “$fourierCoeff(t, xt, T, t1, t2, N)$”

```matlab
function F = fourierCoeff(t, xt, T, t1, t2, N)

    % Declaring a vector array of all zeros
    % As we will have 2*N+1 coefficients (-N to N and one 0)
    % so the size of the vector should be that
    F = zeros(1, 2*N + 1);

    % Calculate the coefficients for each of the 2*N+1 terms
    %we use a for loop which goes from the -N to N
    x = 1/T;

    for k = -N:N
        % Define the integranl
        integral_find = xt * exp(-2*pi*k*(1j)*t*x);
        % Calculate the coefficient using symbolic integration
        coeff = x * int(integral_find, t, t1, t2);
        % Store the coefficient in the vector
        F(k + N + 1) = coeff;
    end
end
```

Explanation of the code is given with the code using comments.

(b) Matlab Script

```matlab
syms t;
% Defining t as the symbolic variable
xt = 2*cos(2*pi*t) + cos(6*pi*t);
% Declaring signal xt using t

T = 1;          % Time period of xt
N = 5;          % Number of FS coefficients
t1 = -T/2;      % Integration lower limit
t2 = T/2;       % Integration upper limit

vector = fourierCoeff(t, xt,T, t1, t2, N);

% Plotting coeffecients in graph
x_axis = -N:N;
figure;
stem(x_axis,vector);
grid on;
xlabel('Index');
ylabel('Coeffecient value');
```



Time period of xt can be calculated by taking the LCM of the time periods of the individual terms. Time period of $Cos(2\pi t) = 1$ and of $Cos(6\pi t) = \frac{1}{3}$, therefore Time period of their sum is equal to LCM of $1 \ \& \ \frac{1}{3}$ which is 1.

given;

$x(t) = 2\cos(2\pi t) + \cos(6\pi t)$.

$$a_k = \frac{1}{T}\int_{t_1}^{t_2} x(t)\, e^{-jk\omega_0 t}\, dt \qquad \left\{\begin{matrix} T=1 \\ t_1=0 \\ t_2=1 \end{matrix}\right\}$$

$$\left\{\omega_0 = \frac{2\pi}{T} = 2\pi\right\}$$

$$\Rightarrow x(t) = 2\left\{\frac{e^{2\pi jt} + e^{-2\pi jt}}{2}\right\} + \left\{\frac{e^{6\pi jt} + e^{-6\pi jt}}{2}\right\}$$

$$\Rightarrow x(t) = e^{2\pi jt} + e^{-2\pi jt} + \frac{e^{6\pi jt} + e^{-6\pi jt}}{2}$$

↳ put it in a_k $\qquad (\omega_0 = 2\pi)$

$$\Rightarrow a_k = \int_0^1 \left\{e^{2\pi jt} + e^{-2\pi jt} + \frac{e^{6\pi jt} + e^{-6\pi jt}}{2}\right\} e^{-jk2\pi t}\, dt$$

$$\Rightarrow a_k = \int_0^1 \left(e^{2\pi jt(1-k)}\right) dt + \int_0^1 e^{-2\pi jt(1+k)}\, dt + \int_0^1 \frac{e^{2\pi jt(3-k)}}{2}\, dt + \int_0^1 \frac{e^{-2\pi jt(3+k)}}{2}\, dt$$

$$\Rightarrow a_k = \left.\frac{e^{2\pi jt(1-k)}}{2\pi j(1-k)}\right|_0^1 + \left.\frac{e^{-2\pi jt(1+k)}}{-2\pi j(1+k)}\right|_0^1 + \left.\frac{e^{2\pi jt(3-k)}}{4\pi j(3-k)}\right|_0^1 + \left.\frac{e^{-2\pi jt(3+k)}}{-4\pi j(3+k)}\right|_0^1$$

after simplification:-

$$\Rightarrow \boxed{a_k = \frac{k\sin(\pi k)}{\pi}\left(\frac{11-3k^2}{(1-k^2)(9-k^2)}\right)}$$

Now, this a_k is defined for $k = \pm 1 \,\&\, \pm 3$
as it forms $\frac{0}{0}$ form

$$\lim_{k \to 1} \frac{k \sin(\pi k)}{\pi} \left(\frac{11 - 3k^2}{(1-k^2)(9-k^2)} \right) = 1/2$$

$$\lim_{k \to -1} \frac{k \sin(\pi k)}{\pi} \left(\frac{11 - 3k^2}{(1-k^2)(9-k^2)} \right) = 1/2$$

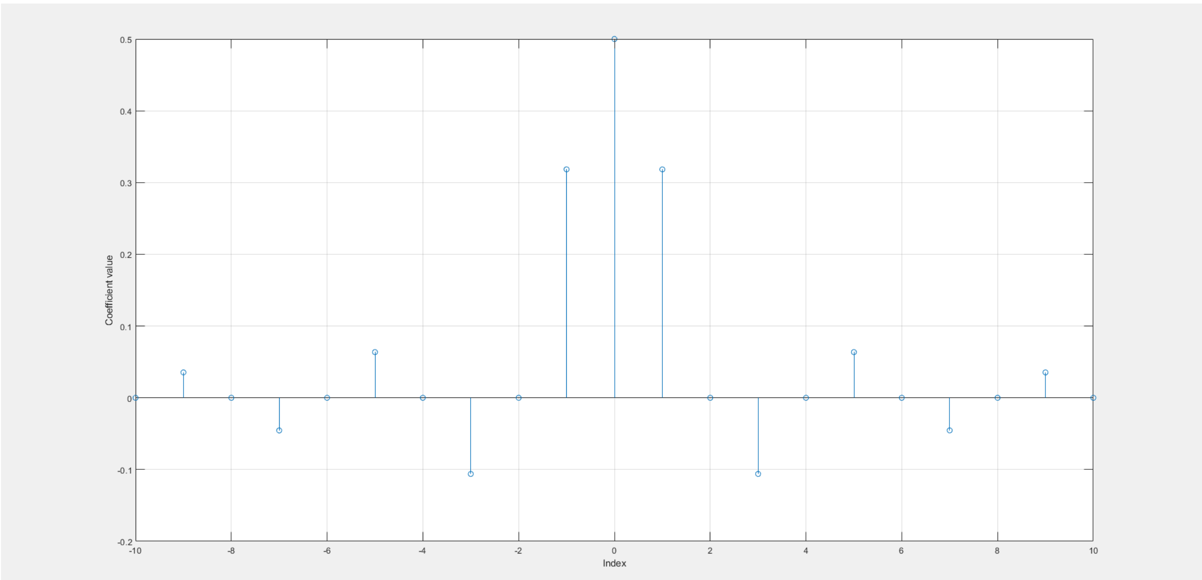
$$\lim_{k \to 3} \frac{k \sin(\pi k)}{\pi} \left(\frac{11 - 3k^2}{(1-k^2)(9-k^2)} \right) = 1$$

$$\lim_{k \to -3} \frac{k \sin(\pi k)}{\pi} \left(\frac{11 - 3k^2}{(1-k^2)(9-k^2)} \right) = 1$$

apply L'Hopital

(c) Fourier Coefficients for Periodic Square Wave

$$x(t) = \begin{cases} 1, & -T_1 \le t \le T_1 \\ 0, & T_1 < |t| < T/2 \end{cases}$$



1.3 : FS reconstruction and finite FS approximation error

In this experiment, we basically have a signal. We find the fourier coefficients of the signals and then reconstruct our signal by the given formula to see if perfect reconstruction is possible or not for the given signal.

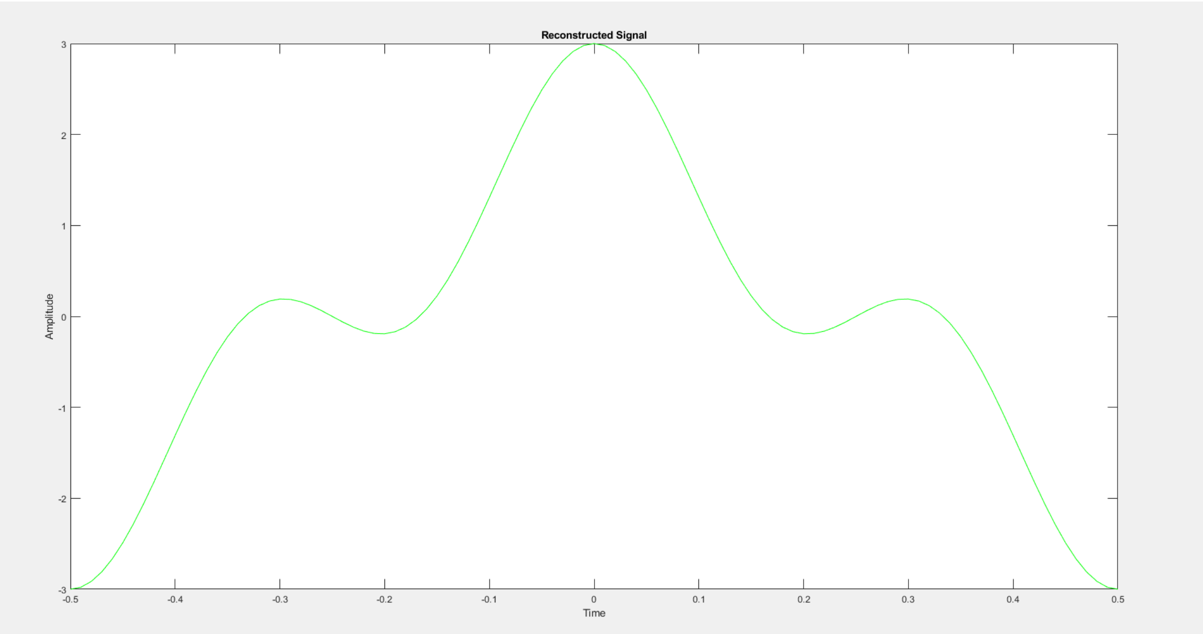
$$\hat{x}(t) = \sum_{k=-N}^{N} a_k \, e^{jk\omega_0 t}$$

Following is the code for $partialfouriersum()$ function :-

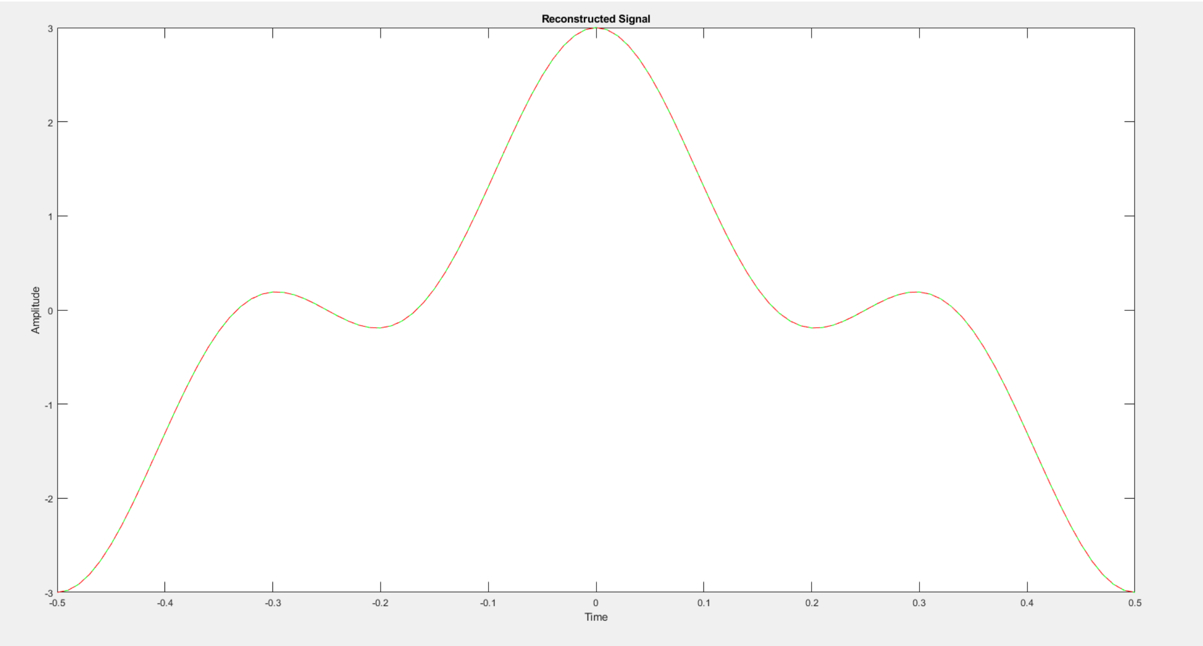
```
% Function to calculate the reconstructed signal
function sum_signal = partialfouriersum(vector, time_limit, T)
    % Initialize the summation
    N = (length(vector)-1)/2;
    sum_signal = zeros(size(time_limit));
    dummy = 1/T;

    for k = -N:N
        a_k = vector(k+N+1);
        x = a_k * exp((1j)*k*time_limit*2*pi*dummy);
        sum_signal = sum_signal + x;
    end
end
```

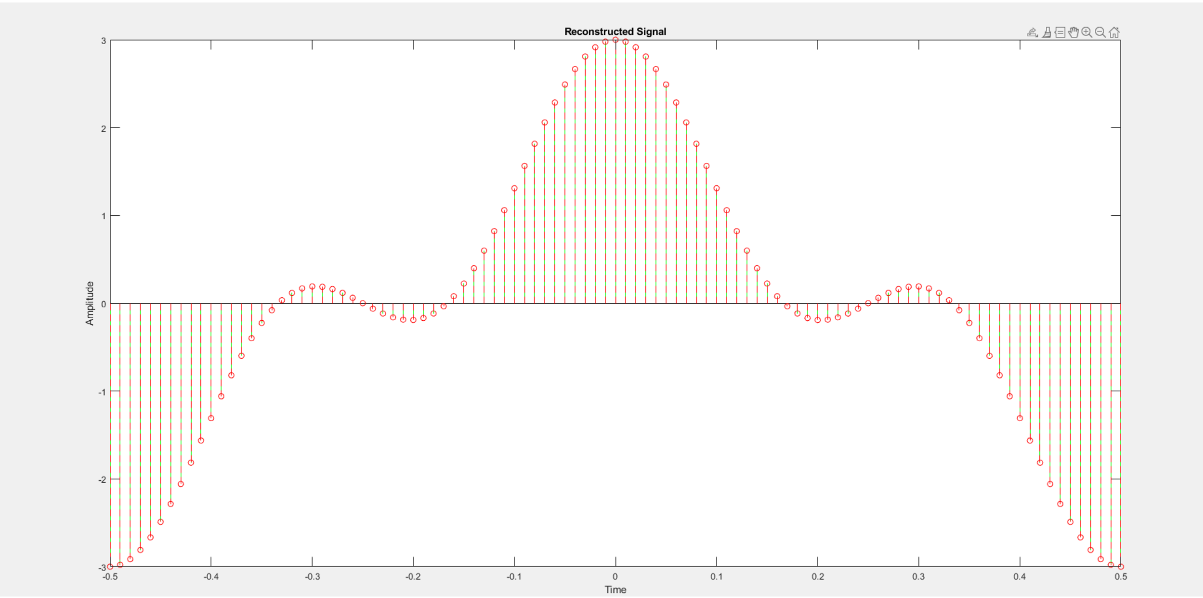
Following is the main code :-

```
% Main script
syms t;
x_t = 2*cos(2*pi*t) + cos(6*pi*t);
T = 1;          % Time period
N = 5;         % Number of harmonics
t1 = -T/2;
t2 = T/2;       % Integration limits
time_limit = -0.5 : 0.01 : 0.5;

% Calculate Fourier coefficients
vector = fourierCoeff(t, x_t, t1, t2, N, T);

% Reconstruct the signal using the Fourier series
reconstructed_signal = partialfouriersum(vector, time_limit, T);


% Plot the reconstructed signal
% plot(vector);
% figure;
plot(time_limit, real(reconstructed_signal),'g');
hold on;
x_t = 2*cos(2*pi*time_limit) + cos(6*pi*time_limit);
plot(time_limit, x_t,'--r');
xlabel('Time');
ylabel('Amplitude');
title('Reconstructed Signal');
hold off;
```



In the following code we plot the x_t and reconstruction signal together so that we can see the error .
We plot the dotted graph of x_t as we observe perfect reconstruction .





We can see that there is perfect reconstruction !

For calculating the error following is the code for it :-

```
%now calculating the error
maximum_absolute_error = abs(reconstructed_signal-x_t);
find_error = max(maximum_absolute_error);
find_error
```

And the following output :-

```
error =

   1.3334e-15
```

So the error in our reconstruction is negligible.

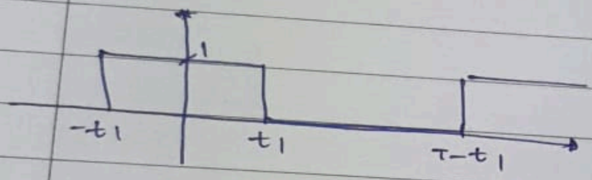
1.4 : Gibbs Phenomenon

In this experiment, we will study the effect of time period and samples on the Square wave by generating its fourier coefficients and reconstructing it.

Following is the mathematical analysis for coefficients of x(t) given.

•> given,

$$x(t) = \begin{cases} 1, & -t_1 < t < t_1 \\ 0, & t_1 < t < T - t_1 \end{cases}$$



We have; $a_k = \frac{1}{T} \int_{\langle T \rangle} x(t) \, e^{-jk\omega_0 t} \, dt$

$$\Rightarrow a_k = \frac{1}{T} \int_{-t_1}^{t_1} e^{-jk\omega_0 t} \, dt$$

$$\Rightarrow a_k = \frac{1}{T} \left. \frac{e^{-jk\omega_0 t}}{(-jk\omega_0)} \right|_{-t_1}^{t_1} = \frac{1}{T} \left\{ \frac{e^{jk\omega_0 t_1} - e^{-jk\omega_0 t_1}}{jk\omega_0} \right.$$

We have, $\omega_0 = \frac{2\pi}{T}$

$$\Rightarrow a_k = \frac{1}{T} \, \frac{e^{jk\omega_0 t_1} - e^{-jk\omega_0 t_1}}{jk\frac{2\pi}{T}}$$

$$\Rightarrow a_k = \frac{1}{k\pi} \left(\frac{e^{jk\omega_0 t_1} - e^{-jk\omega_0 t_1}}{2j} \right) \rightarrow (\sin\theta)$$

$$\Rightarrow a_k = \frac{1}{\pi k} \sin(k \omega_0 t_1)$$

$$\Rightarrow a_k = \frac{1}{\pi k} \sin\left(\frac{k 2\pi t_1}{T}\right) \times \frac{2t_1}{T} \times \frac{1}{\frac{2t_1}{T}}$$

$$\Rightarrow \boxed{a_k = \frac{2t_1}{T} \operatorname{sinc}\left(\frac{2k t_1}{T}\right)}$$

= ans.

In this we can see that the coefficient depends on the value of T.

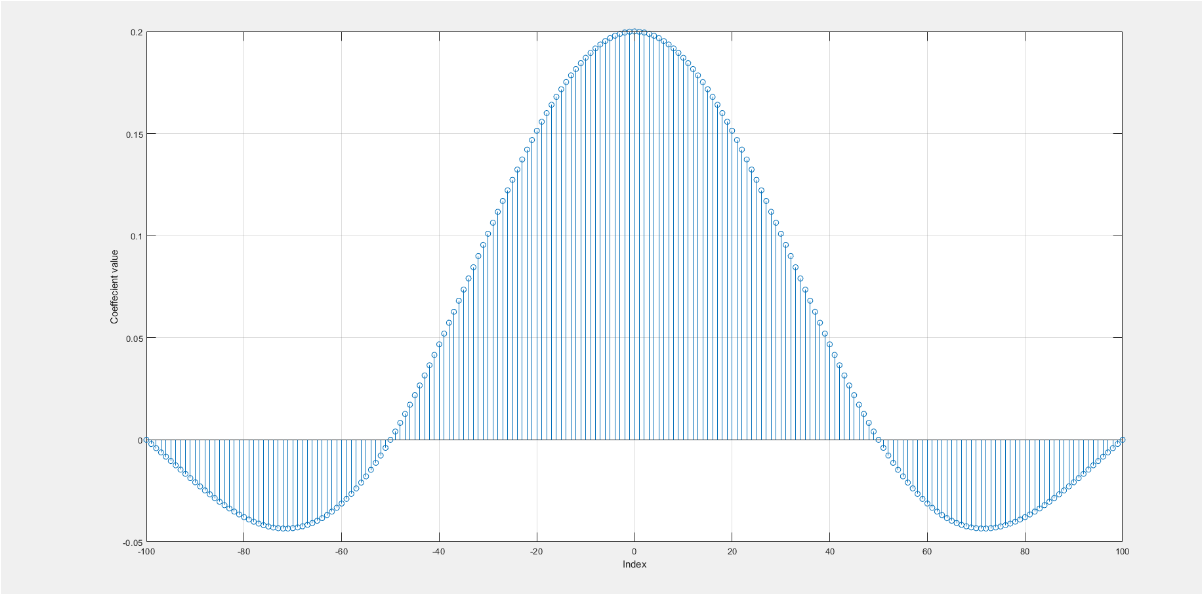
Code for calculating the coefficients :-

For fourier coefficients we have already given the code above.

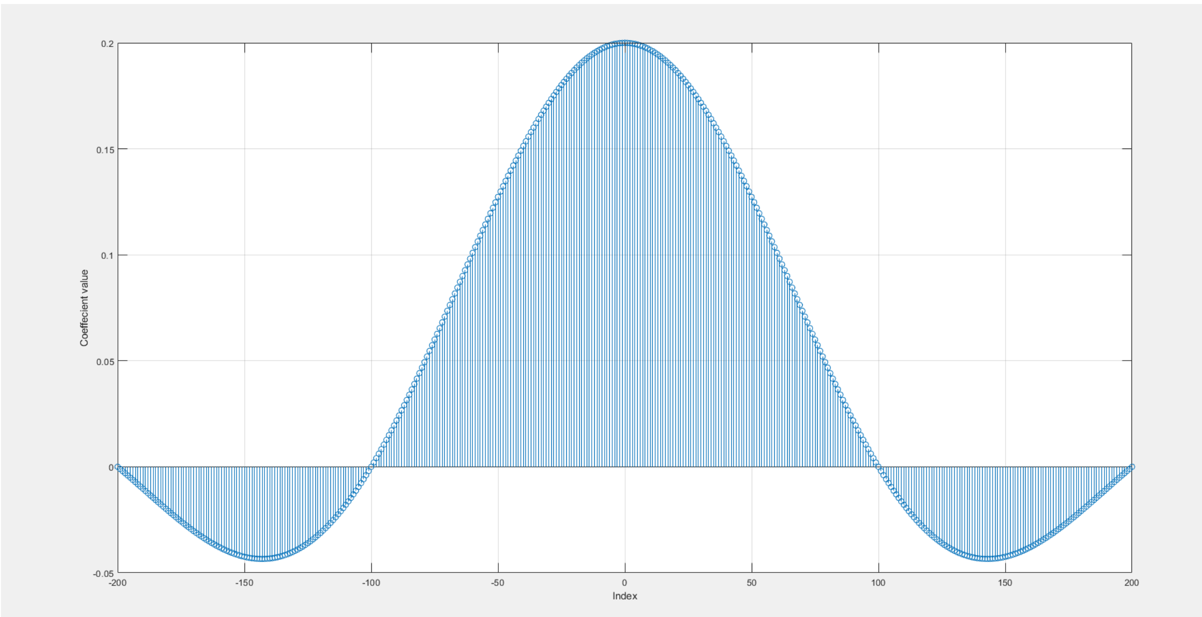
The main code:-

```
syms t;
t1 = 0.1;
T = 1;
N = 10;
x_t = piecewise(-t1<t<t1,1,t1<t<T-t1,0);

vector = fourierCoeff(t,x_t,-t1,t1,N,T);

%now plotting this graph coeffe
x_axis = -N:N;
figure; stem(x_axis,vector);grid on;
xlabel('Index');
ylabel('Coeffecient value');
```

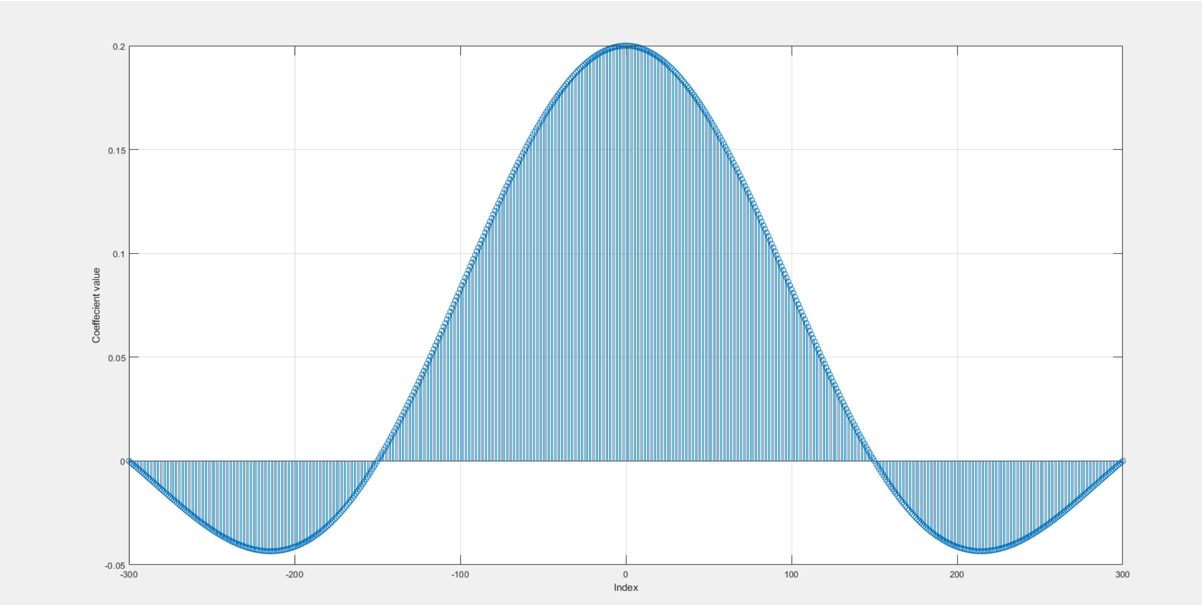
Plot for T = 10 :-



Plot for T = 20 :-



Plot for T = 30 :-



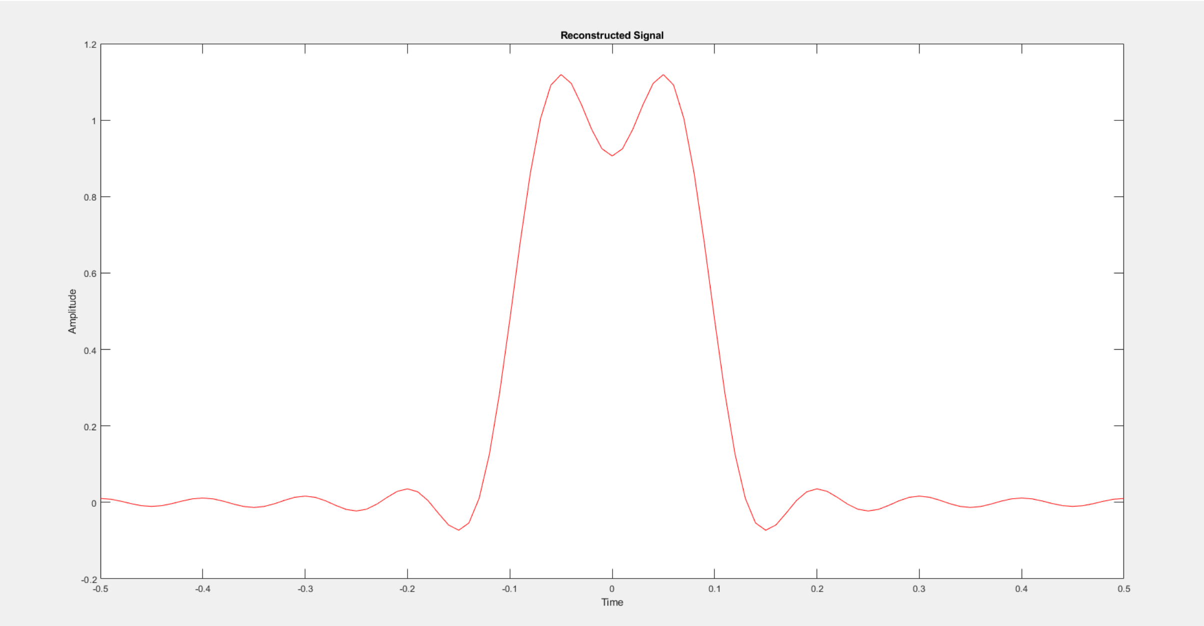
On increasing T we can see that the maximum magnitude at k = 0 (i.e 0.2) always remain constant but the number of coefficients increase as N increases so the x - axis graph expands.

Now we have found out the fourier coefficients , now we can reconstruct the signal back using these coefficients :-

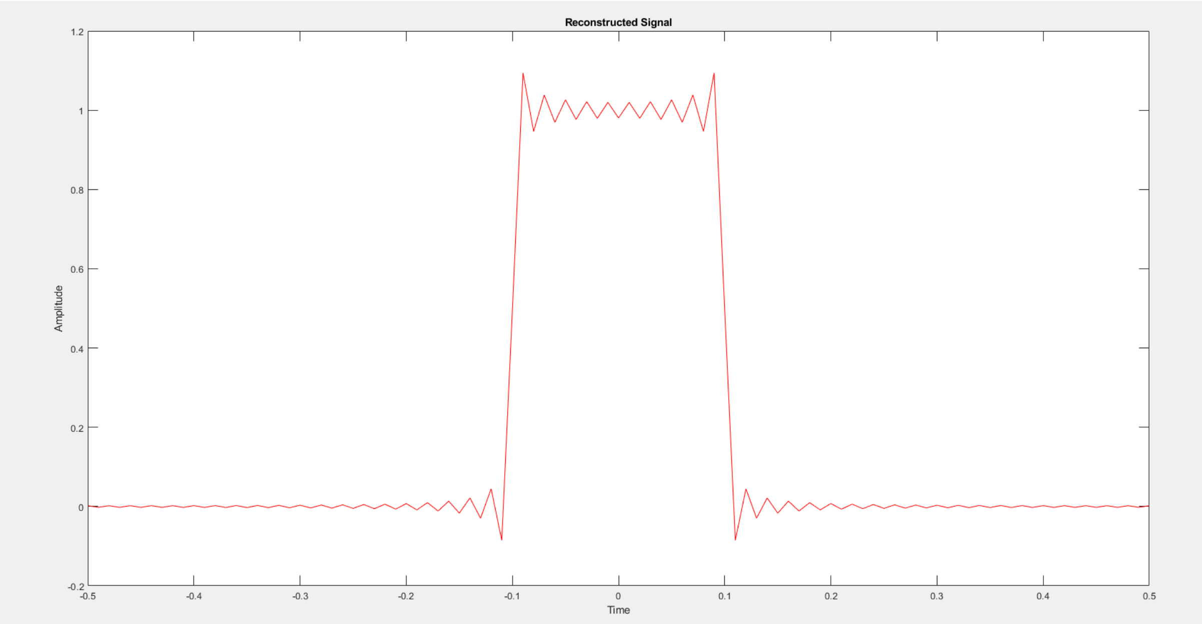
As the partialfouriersum code is already given above.
This is the main code:-

```
syms t;
t1 = 0.1;
T = 1;
N = 10;
x_t = piecewise(-t1<t<t1,1,t1<t<T-t1,0);

vector = fourierCoeff(t,x_t,-t1,t1,N,T);

%now finding the reconstructed signal
time_limit =    -0.5:0.01:0.5;
reconstruction = partialfouriersum(vector,time_limit,T);

plot(time_limit,real(reconstruction),'r');
xlabel('Time');
ylabel('Amplitude');
title('Reconstructed Signal');
```

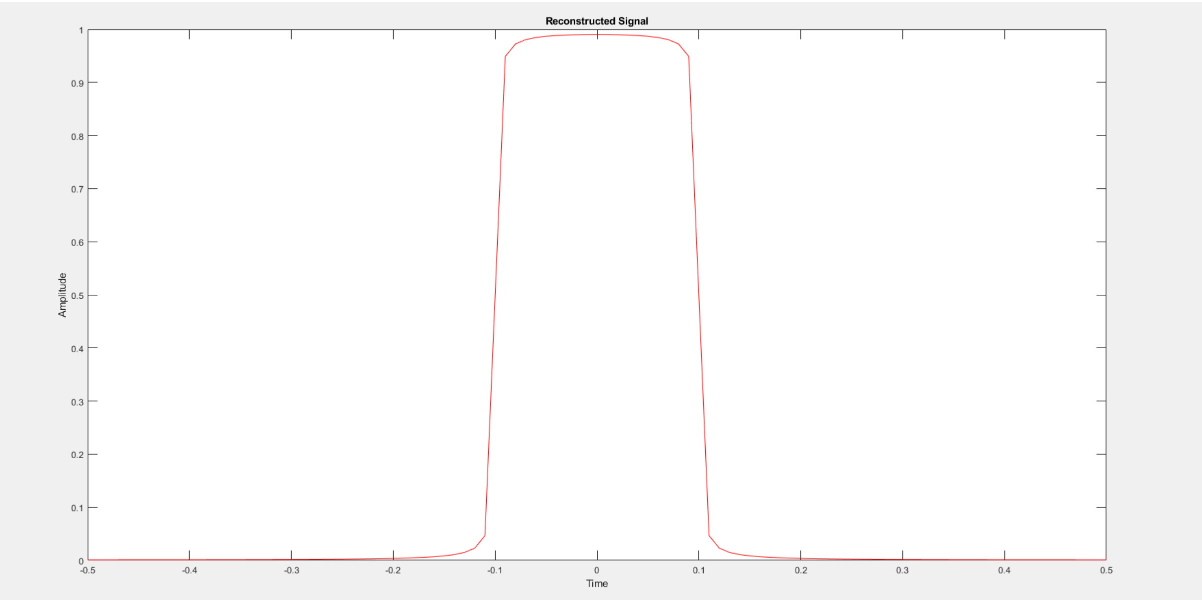
Plot for N = 10 :-



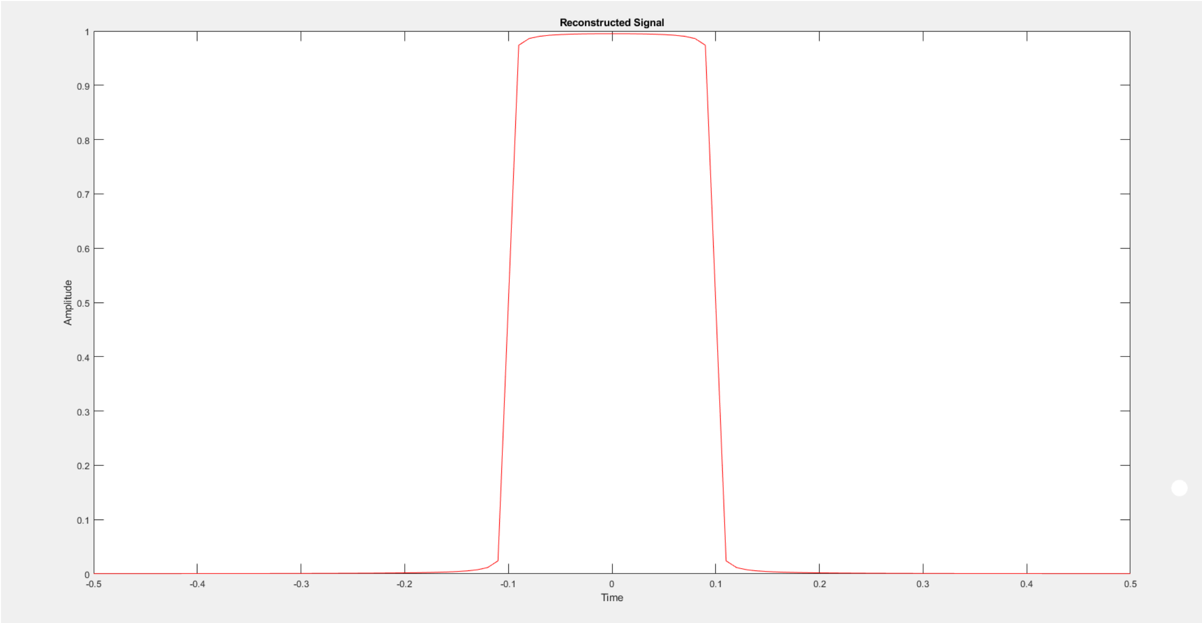
Plot for N = 50 :-



Plot for N = 100 :-



Plot for N = 200 :-



We observe that when the value of N increases the graph reconstructs more perfectly and the error nearly tends to 0 !
As we find more coefficients of the signal and then again sum them so reconstruction is perfect as N increases .