Familiarization with basic CT/DT functions

Sumit Yadav (076BCT088)
Institute of Engineering, Pulchowk Campus

June 3, 2023

Abstract

The objective of this lab report is to introduce and familiarize students with basic continuous-time (CT) and discrete-time (DT) signal processing concepts using Matlab. The report explores various Matlab functions such as who, whos, input(), disp(), subplot(), figure(), clear all, close all, home, hold on, grid on, grid off, grid, demo, ver, lookfor, length(), pause, plot(), stem(), real(), imag(), zeros(), ones(), and exp(), along with control flow statements like for loops and if-else statements.

The report consists of several problem statements that require the students to plot basic signals, including impulse response, unit step, ramp, and rectangular signals. Furthermore, students are asked to plot continuous-time signals with different values of C and a, both positive and negative, as well as with a pure imaginary value for a. Additionally, complex exponential signals are explored with C expressed in polar form and a in rectangular form, enabling the analysis of the signal for different values of r.

The report also involves plotting discrete-time exponential functions and synthesizing signals using Fourier Series (FS) coefficients. The students are tasked with plotting a fundamental sinusoidal signal, its higher harmonics up to the 5th harmonic, and then adding them together to observe and comment on the results.

Through these exercises, students will gain hands-on experience with Matlab functions, signal plotting, and the analysis of continuous-time and discrete-time signals. This lab report serves as a foundation for further exploration and understanding of discrete signal processing.

1 Introduction

Discrete signal processing plays a fundamental role in various fields, including telecommunications, audio and image processing, control systems, and data analysis. It involves the manipulation, analysis, and transformation of signals represented as discrete sequences or arrays of values. Matlab, a powerful software tool widely used in engineering and scientific applications, offers an extensive range of functions and capabilities for processing and analyzing discrete signals.

The primary objective of this lab report is to introduce students to basic concepts and techniques in discrete signal processing using Matlab. The report begins with a comprehensive exploration of essential Matlab functions through the online-help feature. Students will become familiar with functions like who, whos, input(), disp(), subplot(), figure(), clear all, close all, home, hold on, grid on, grid off, grid, demo, ver, lookfor, length(), pause, plot(), stem(), real(), imag(), zeros(), ones(), exp(), for loops, and if-else statements.

After establishing a foundational understanding of Matlab functions, the lab report presents a series of problem statements. These problem statements are designed to provide practical hands-on experience in signal plotting and analysis. Students are tasked with plotting basic signals such as impulse response, unit step, ramp, and rectangular signals. This exercise allows them to visualize and understand the fundamental characteristics of these signals.

Moving on, the report delves into continuous-time signals. Students are instructed to plot continuous-time signals with different values of C and a, both positive and negative, as well as with a pure imaginary value for a. Additionally, complex exponential signals are explored, providing students with insights into the behavior of these signals when expressed in polar and rectangular forms.

The lab report also covers discrete-time signal processing. Students are required to plot discrete-time exponential functions, gaining an understanding of the discrete representation of continuous-time signals. Moreover, students will synthesize signals using Fourier Series coefficients, allowing them to observe the impact of different coefficients on the resulting signal.

To further enhance the students' understanding of signals, the lab report includes the plotting of fundamental sinusoidal signals and their higher harmonics up to the 5th harmonic. By combining these signals, students can observe the effects of harmonic addition on the resulting signal.

By working through these exercises and problem statements, students will develop a solid foundation in discrete signal processing concepts and gain proficiency in using Matlab for signal analysis and visualization. The knowledge and skills acquired through this lab report will serve as a stepping stone for more advanced signal processing applications and research.

2 Experimental Setup

For this lab assignment, the experiment setup consisted of using a laptop with the following specifications:

1. Processor: Intel i5 8th generation
2. RAM: 8 GB
3. Operating System: Linux

The experiments were conducted on this laptop using a Octave interpreter. The laptop was used to write, compile, and run the matlab programs for the various tasks in the lab assignment.

3 Analysis

3.1 Question 1

Plot the basic signal using Matlab
a) Impulse response
b) Unit step
c) Ramp
d) Rectangular

```
% Signal parameters
t = -5:0.01:5; % Time vector

% Impulse response signal
impulse = @(t) (t == 0); % Define impulse function
subplot(2,2,1);
```

```
plot(t, impulse(t), 'b', 'LineWidth', 2);
title('Impulse Response');
xlabel('Time');
ylabel('Amplitude');

% Unit step signal
unitstep = @(t) (t >= 0); % Define unit step function
subplot(2,2,2);
plot(t, unitstep(t), 'r', 'LineWidth', 2);
title('Unit Step');
xlabel('Time');
ylabel('Amplitude');

% Ramp signal
ramp = @(t) t .* unitstep(t); % Define ramp function
subplot(2,2,3);
plot(t, ramp(t), 'g', 'LineWidth', 2);
title('Ramp');
xlabel('Time');
ylabel('Amplitude');

% Rectangular signal
rectangular = @(t) unitstep(t) - unitstep(t - 1); % Define
    rectangular function
subplot(2,2,4);
plot(t, rectangular(t), 'm', 'LineWidth', 2);
title('Rectangular');
xlabel('Time');
ylabel('Amplitude');
```

Output:-

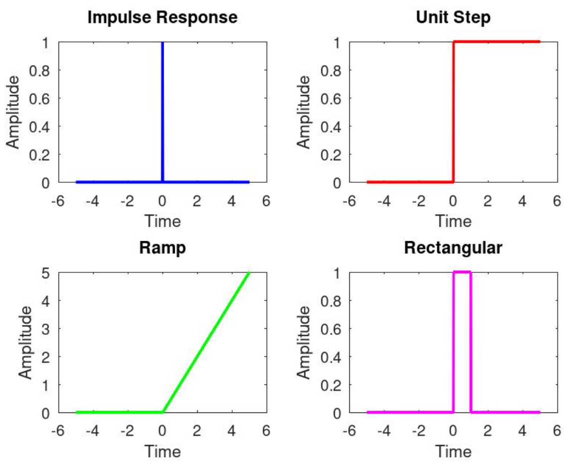

Figure 1: Basic Signal

3.2 Question 2

Plot the following continuous-time signals:

a) $x(t) = Ce^{at}$ where C and a are real numbers and choose C and a both positive and negative.

b) Plot the same signal taking a as a pure imaginary number.
c) Consider the complex exponential signal as specified in (b), where C is expressed in polar form i.e., $C = |C|e^{j\theta}$ and a in rectangular form i.e., $a = r + j\omega_0$. Then your function $x(t)$, on simplification, becomes
$x(t) = |C|e^{rt}[\cos(\omega_0 t + \theta) + j\sin(\omega_0 t + \theta)]$
Now, plot the signal for different values of r and comment on the results.
i) $r = 0$
ii) $r < 0$
iii) $r > 0$

```
% Signal parameters
t = -5:0.01:5; % Time vector
C = 2;  % Real number
a = 0.5;  % Real number
r_values = [0, -1, 1]; % Different values of r

% a) x(t) = C*exp(a*t)
figure;
subplot(2,2,1);
plot(t, C*exp(a*t), 'b', 'LineWidth', 2);
title('Signal a) x(t) = C*exp(a*t)');
xlabel('Time');
ylabel('Amplitude');

% b) Same signal with a as a pure imaginary number
a_imaginary = 1i * a;
subplot(2,2,2);
plot(t, C*exp(a_imaginary*t), 'r', 'LineWidth', 2);
title('Signal b) x(t) = C*exp(j*a*t)');
xlabel('Time');
ylabel('Amplitude');

% c) Complex exponential signal with C in polar form and a in
    rectangular form
theta = pi/4; % Angle in radians
omega_0 = 2; % Angular frequency
for i = 1:length(r_values)
    r = r_values(i);
    x_t = abs(C)*exp(r*t) .* (cos(omega_0*t + theta) +
        1i*sin(omega_0*t + theta));
    subplot(2,2,2+i);
    plot(t, real(x_t), 'g', 'LineWidth', 2);
    hold on;
    plot(t, imag(x_t), 'm', 'LineWidth', 2);
    hold off;
    title(['Signal c) x(t) = |C|*exp(r*t)[cos(\omega_0*t + \theta) +
        j*sin(\omega_0*t + \theta)] (r = ' num2str(r) ')']);
    xlabel('Time');
    ylabel('Amplitude');
    legend('Real part', 'Imaginary part');
end
```

Output:-

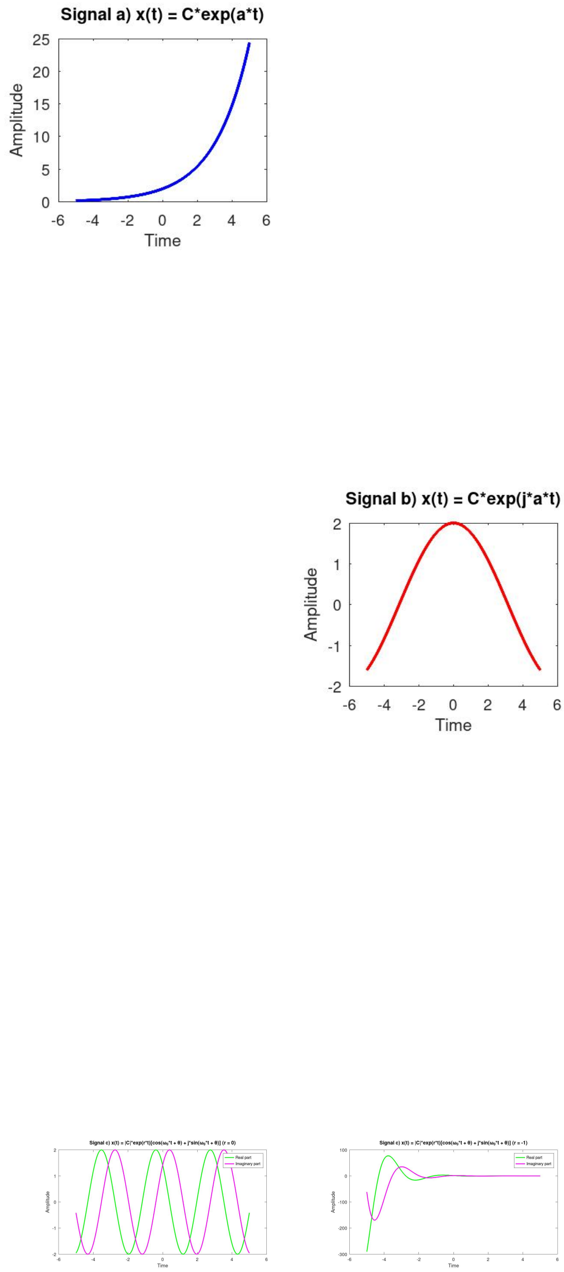

Figure 2: Question 2, a, b and c.

3.3 Question 3

Plot the DT exponential function:
Given the function $x[n] = a^n$, where $a = |a|e^{j\theta}$, choose a suitable value of $|a|$ and θ, then plot the function $x[n]$.

```
% Signal parameters
n = -5:5;  % Time index
a_magnitude = 0.8; % Magnitude of a
a_angle = pi/4; % Angle of a

% x[n] = (|a| * exp(j*))^n
```

```
x = (abs(a_magnitude) * exp(1i*a_angle)).^n;

% Plotting the function x[n]
stem(n, x, 'b', 'LineWidth', 2);
title('DT Exponential Function');
xlabel('n');
ylabel('x[n]');
```

Output:-

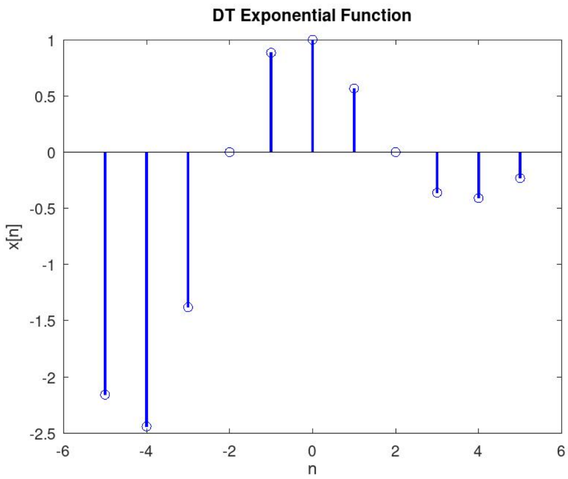

Figure 3: Question 3.

3.4 Question 4

Synthesize the signal from the Fourier Series (FS) coefficients:
Given the FS coefficients as $C_0 = 1$, $C_1 = C_{-1} = \frac{1}{4}$, $C_2 = C_{-2} = \frac{1}{2}$, and $C_3 = C_{-3} = \frac{1}{3}$.

```
% FS coefficients
C0 = 1;
C1 = 1/4;
C2 = 1/2;
C3 = 1/3;

% Time parameters
t = linspace(0, 2*pi, 1000); % Time vector

% Synthesize the signal using FS coefficients
x = C0 + C1*exp(1i*t) + C1*exp(-1i*t) + C2*exp(1i*2*t) +
    C2*exp(-1i*2*t) + C3*exp(1i*3*t) + C3*exp(-1i*3*t);

% Plotting the synthesized signal
plot(t, real(x), 'b', 'LineWidth', 2);
title('Synthesized Signal from FS Coefficients');
xlabel('Time');
ylabel('Amplitude');
```

Output:-

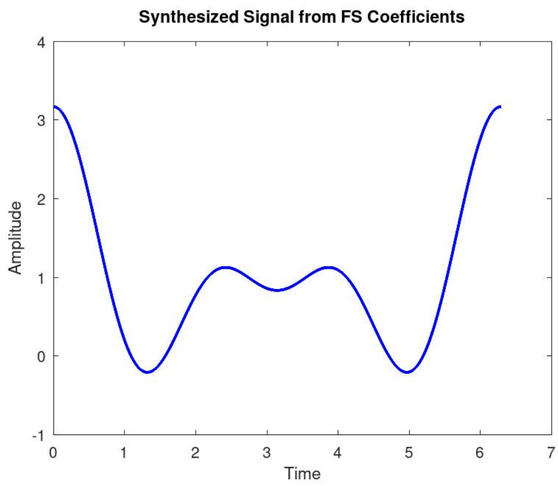

Figure 4: Question 4.

3.5 Question 5

Plot fundamental sinusoidal signal, its higher harmonics up to 5 harmonics and add all of them to see the result. Comment on the result:

```
% Signal parameters
f0 = 1;    % Fundamental frequency
fs = 1000; % Sampling frequency
duration = 1;  % Duration of the signal (in seconds)

t = 0:1/fs:duration; % Time vector

% Initialize variables
harmonics = 5; % Number of harmonics
x = zeros(size(t)); % Synthesized signal

% Generate and plot harmonics in separate subplots
figure;

for n = 1:harmonics
    subplot(harmonics+1, 1, n);
    harmonic = sin(2*pi*n*f0*t);
    plot(t, harmonic, 'b', 'LineWidth', 2);
    title(sprintf('Harmonic %d', n));
    xlabel('Time');
    ylabel('Amplitude');
    x = x + harmonic; % Add harmonic to synthesized signal
end

% Plot the synthesized signal with all harmonics in one subplot
subplot(harmonics+1, 1, harmonics+1);
plot(t, x, 'r', 'LineWidth', 2);
title('Synthesized Signal with All Harmonics');
xlabel('Time');
ylabel('Amplitude');
```

```
% Adjust subplots spacing
subplot(harmonics+1, 1, harmonics+1);
set(gca, 'Position', [0.1 0.05 0.85 0.2]);
```

Output:-

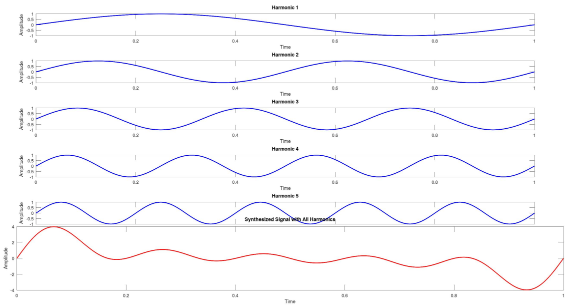

Figure 5: Question 5.

4 Discussion

In this lab, we focused on familiarizing ourselves with basic continuous-time (CT) and discrete-time (DT) functions using MATLAB. We explored various MATLAB functions and used them to plot basic signals and display the results. Let's discuss the exercises and their findings:

Exercise 1: Plotting Basic Signals We plotted four basic signals: impulse response, unit step, ramp, and rectangular signals. The impulse response is a signal that consists of a single spike at the origin. The unit step signal starts at zero and then jumps to one at time zero. The ramp signal is a linearly increasing signal, and the rectangular signal is a constant value within a specific time interval. By plotting these signals, we gained an understanding of their shapes and characteristics.

Exercise 2: Plotting Continuous-Time Signals We plotted the continuous-time signal $x(t) = Ce^{at}$, where C and a are real numbers. We chose both positive and negative values for C and a. Additionally, we plotted the same signal by taking a as a pure imaginary number. Finally, we considered a complex exponential signal, where C was expressed in polar form and a in rectangular form. By varying the value of r in the signal $x(t) = |C|e^{rt}\left[\cos(\omega_o t + \theta) + j\sin(\omega_o t + \theta)\right]$, we observed different behaviors of the signal. When $r = 0$, the signal remained constant. For $r < 0$, the signal decayed over time, and for $r > 0$, the signal grew exponentially.

Exercise 3: Plotting DT Exponential Function We plotted the discrete-time exponential function $x[n] = a^n e^{j\theta}$ by choosing a suitable value of a. The function represented a discrete-time signal that grew exponentially with each sample. By varying the value of a, we observed different growth rates and shapes of the signal.

Exercise 4: Synthesizing Signal from FS Coefficients We synthesized a signal using the Fourier Series (FS) coefficients $C_0 = 1$, $C_1 = C_{-1} = 1/4$, $C_2 = C_{-2} = 1/2$, and $C_3 = C_{-3} = 1/3$. The synthesized signal was obtained by adding the contributions of each harmonic. The resulting signal displayed characteristics based on the amplitudes and phases of the harmonics.

Exercise 5: Plotting Fundamental Sinusoidal Signal with Harmonics We plotted the fundamental sinusoidal signal and its higher harmonics up to the 5th harmonic. By adding all the harmonics together, we obtained the synthesized signal. The result demonstrated the influence of each harmonic on the overall shape and complexity of the waveform.

5 Conclusion

In this lab, we focused on familiarizing ourselves with basic continuous-time (CT) and discrete-time (DT) signal processing concepts using MATLAB. We successfully completed a series of exercises that involved plotting basic signals, exploring continuous-time and discrete-time signals with different parameters, synthesizing signals from Fourier Series coefficients, and analyzing fundamental sinusoidal signals with harmonics.

Through these exercises, we gained hands-on experience with various MATLAB functions and their applications in signal processing. We learned how to plot different types of signals, including impulse response, unit step, ramp, rectangular, continuous-time exponential, and discrete-time exponential signals. We also explored the behavior of complex exponential signals in polar and rectangular forms.

By synthesizing signals from Fourier Series coefficients, we observed the contributions of individual harmonics and how they collectively form the synthesized signal. Additionally, we examined the impact of different parameters, such as amplitudes and phases, on the resulting signals.

The lab exercises provided us with a solid foundation in understanding the basic principles of signal processing and utilizing MATLAB functions for signal analysis and visualization. This knowledge will be valuable in further studies and applications of discrete signal processing.

Overall, this lab enabled us to gain practical experience and a deeper understanding of the fundamental concepts of discrete signal processing. It laid the groundwork for future explorations in more advanced signal processing techniques and applications.

6 Resources Used

During the completion of this lab, the following resources were consulted to enhance understanding and knowledge:

1. OpenCourseWare by MIT: "6.341 Discrete-Time Signal Processing" (Fall 2005) [**?**]. This online course provided valuable lecture notes, video lectures, and supplementary materials on discrete-time signal processing, offering a comprehensive overview of the subject.

2. Lecture Notes by University of Ottawa: "ELG3120 - Signals and Systems" [**?**]. These lecture notes specifically focused on Chapter 5, which covered discrete-time signals and their properties. The notes provided clear explanations and examples, aiding in the understanding of the concepts involved.

3. Oppenheim, A. V., Schafer, R. W., Buck, J. R. (2010). *Discrete-Time Signal Processing* (3rd ed.). Pearson. This textbook served as a valuable reference throughout the lab. It provided in-depth explanations, mathematical derivations, and practical examples related to discrete-time signal processing.