Computer Assignment 1

EEE391- Basics of Signals and Systems

Efe Acer
21602217

Bilkent University, CS

Contents

Question 1

Part a

The sinusoidal signals corresponding to the last three digits of my Bilkent ID are $x_2(t)$, $x_1(t)$, and $x_7(t)$. The second, first and seventh rows of the two-dimensional array `X` corresponds to $N + 1 = 1001$ uniformly sampled values from the sinusoidal signals, where the sampling rate is $f_s = 1000\text{Hz}$. These samples are accessed by row-wise indexing: `signal2 = X(2, :)`, `signal1 = X(1, :)` and `signal7 = X(7, :)`.

After the samples were acquired, `plot` method of `Matlab` is used to visualize the signals. `plot` method requires the input values to a function together with the corresponding function outputs, as its arguments. In our case, the function outputs are the uniformly sampled values contained in the second, first and seventh rows of `X`. The function inputs are basically the matching time points.

A digital signal that is uniformly sampled can be modeled as:

$$x[n] = x(nT_s), n = 0, 1, 2, ..., N \tag{1}$$

The model implies that the time points corresponding to $x[0]$, $x[1]$, $x[2]$, to $x[N]$ are 0, T_s, $2T_s$, to NT_s. Here T_s is the uniform sampling rate, which is indeed the inverse of the sampling frequency. We can compute T_s as follows:

$$T_s = \frac{1}{f_s} \xrightarrow{f_s = 1000\text{Hz}} T_s = \frac{1}{1000}\text{s} = 0.001\text{s} = 1\text{ms} \tag{2}$$

Knowing T_s, we can form a `Matlab` array `t` that corresponds to the time points; 0, 0.001, 0.002, ..., 0.999, 1; which are given in terms of seconds.

Having the time points, stored in `t`, and the signal samples, stored in the specified rows of X, we plot the three signals by giving the pairs (`t`, `signal2`), (`t`, `signal1`) and (`t`, `signal7`) as inputs to the `plot` method. We add the necessary plot titles and axis labels using the relevant `Matlab` methods. The complete `Matlab` code for this part is provided in the Appendix of the report. The resulting plots are presented starting from the next page.

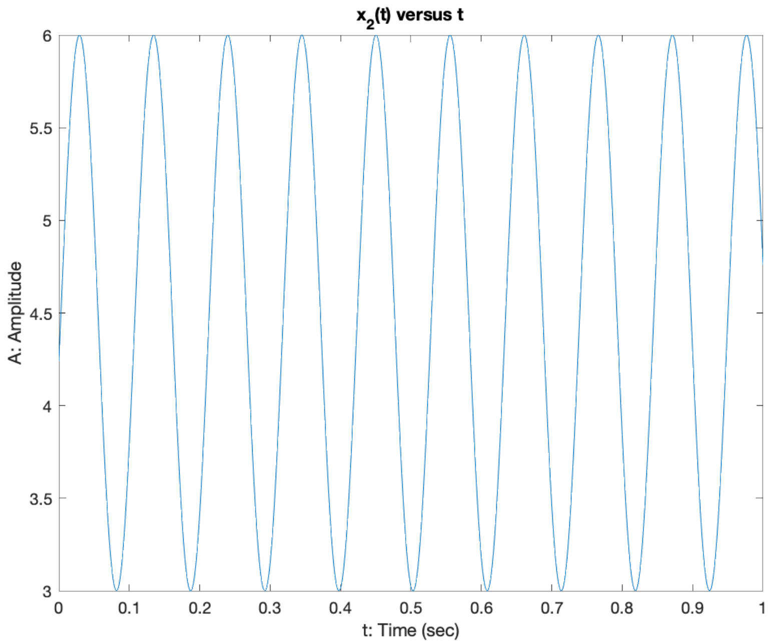

Figure 1: Second Signal versus Time

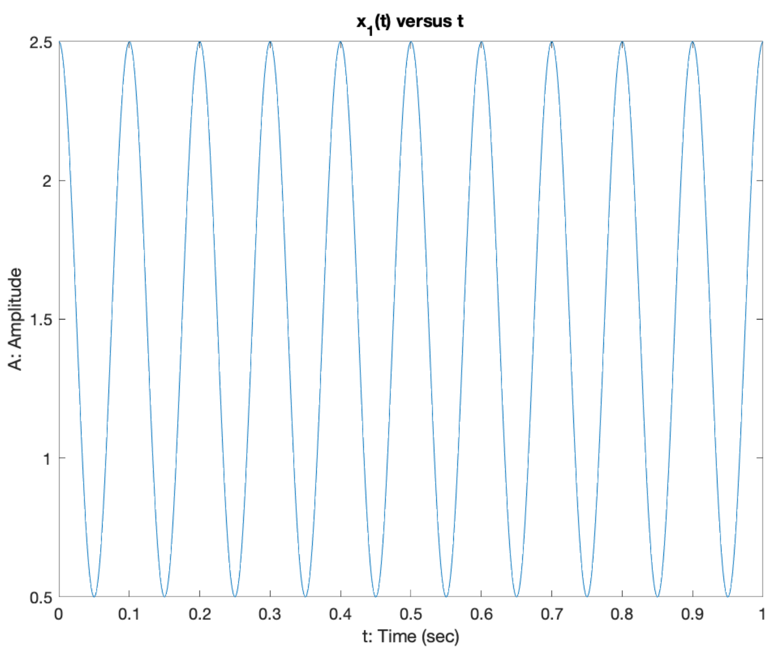

Figure 2: First Signal versus Time

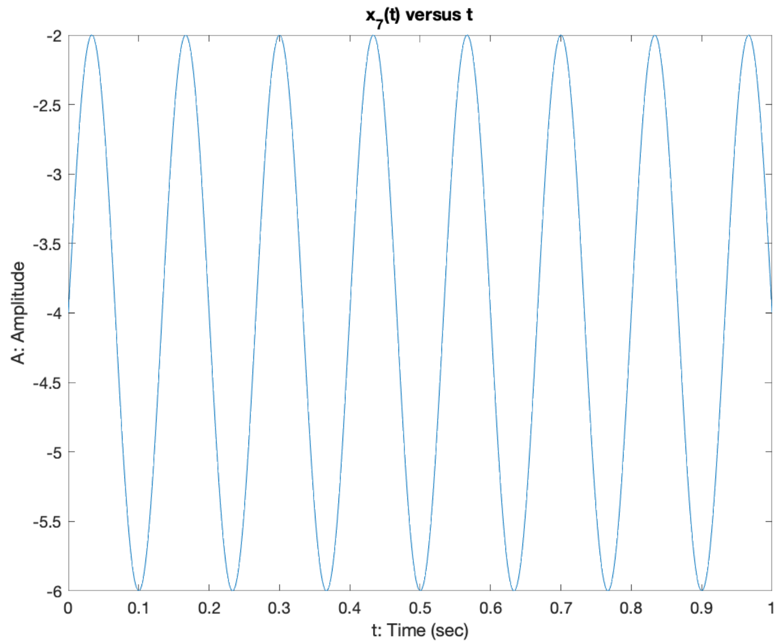

Figure 3: Seventh Signal versus Time

Part b

A sinusoidal signal $x_i(t)$ is conventionally modelled as:

$$x_i(t) = C_i + A_i cos(2\pi f_i t + \phi_i). \tag{3}$$

In the above equation, the paramater definitions are the following:

$$C_i = \text{The DC component of } x_i(t)$$
$$A_i = \text{The amplitude of } x_i(t)$$
$$f_i = \text{The frequency (in Hz) of } x_i(t)$$
$$\phi_i = \text{The phase (in degrees) of } x_i(t)$$

By inspection on the graphs presented in figures 1, 2 and 3; we can estimate the signal parameters. The rationale behind the estimation by inspection is illustrated in the figure given at the beginning of the next page.

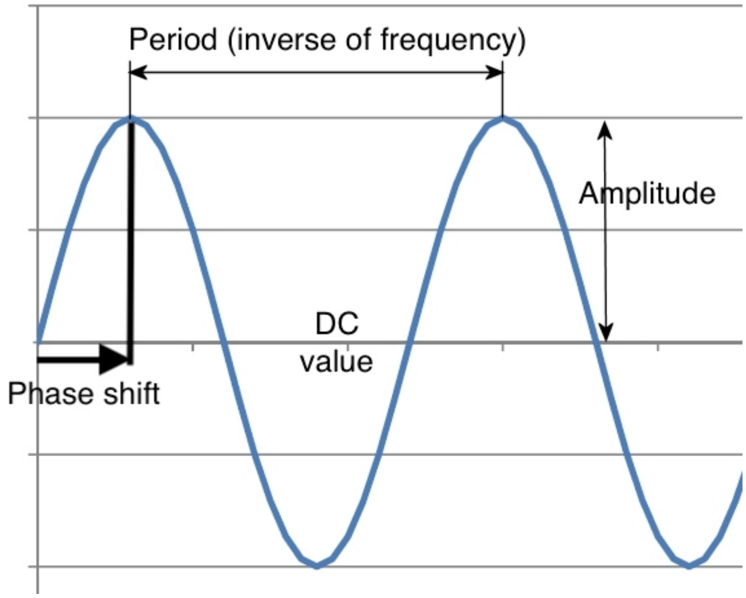

Figure 4: Estimation by Inspection

The logic illustrated in figure 4 is a direct implication of the parameter definitions. Applying this logic to the three signals, we estimate the parameters as:

$C_2 \approx 4.5$
$A_2 \approx 1.5$ $\longrightarrow$ $x_2(t) = 4.5 + 1.5cos(62.83t + 0.03)$ (radians)
$f_2 \approx 1/T_2 = (1/0.1)\text{Hz} = 10\text{Hz}$ $\quad x_2(t) = 4.5 + 1.5cos(6.28t + 0.03)$ (radians)
$\phi_2 \approx 0.03\text{radian} \approx 1.72°$ $\quad x_2(t) = 4.5 + 1.5cos(359.89°t + 1.72°)$ (degrees)

$C_1 \approx 1.5$
$A_1 \approx 1$ $\longrightarrow$ $x_2(t) = 1.5 + cos(62.83t)$ (radians)
$f_1 \approx 1/T_1 = (1/0.1)\text{Hz} = 10\text{Hz}$ $\quad x_2(t) = 1.5 + cos(6.28t)$ (radians)
$\phi_1 \approx 0\text{radian} = 0°$ $\quad x_2(t) = 1.5 + cos(359.89°t)$ (degrees)

$C_7 \approx -4$
$A_7 \approx 2$ $\longrightarrow$ $x_2(t) = -4 + 2cos(44.86t + 0.03)$ (radians)
$f_7 \approx 1/T_1 = (1/0.14)\text{Hz} = 7.14\text{Hz}$ $\quad x_2(t) = -4 + 2cos(0.88t + 0.03)$ (radians)
$\phi_7 \approx 0.03\text{radian} \approx 1.72°$ $\quad x_2(t) = -4 + 2cos(50.29°t + 1.72°)$ (degrees)

Part c

In this part of the question, we develop a `Matlab` function called `parameters_of_sin` that can estimate the parameters we found by inspection in the previous part. The function receives the sampling rate, `fs`, and the uniform samples of the sinusoidal signal, `x`, as its input. There exists a simple programmatic approach to find these parameters, this approach is as follows:

1. First, we find the maximum and the minimum sample values. The average of the maximum and minimum sample values gives the DC component C_i, since the values in the positive part of the y-axis neutralizes those of the negative part.

2. Second, we subtract the just-computed DC component from the maximum sample value to find the amplitude, A_i.

3. Third, we find the sample indices corresponding to the first positive peak value and the first negative peak value. We define the first positive peak value as the first sample value that exceeds $C_i + A_i - A_i/1000$; and similarly the first negative peak value as the first sample value that is less than $C_i - A_i + A_i/1000$. This is because the sample peaks do not exactly correspond to the real signal peaks. The absolute difference between these indices gives us the number of samples in half a period, we compute this value. Then, we multiply this value by two and find the number of samples in one period, and divide the sampling rate by the number of samples in a period to compute the actual frequency of the sinusoidal signal, f_i. This is due to the fact that $f_i = f_s/(\#\text{ of samples in a period})$.

4. Last, we multiply the first positive peak index with the sampling period (inverse of the sampling rate) to find the phase shift in radians. Then, we multiply this value by $180/\pi$ and convert the phase shift to degrees, the result is returned as ϕ_i.

The algorithm described above shares the same logic with the inspection method used in part b. Another `Matlab` program that calls the `parameters_of_sin` function and reports the parameter estimation results for the signals $x_2(t)$, $x_1(t)$ and $x_7(t)$ is written. The output of the program in the `Matlab` command window is given below:

```
>> EEE391_CA1
Signal 2: 4.50 + 1.50 cos(59.28 t + 0.03)
Signal 1: 1.50 + 1.00 cos(62.83 t + 0.00)
Signal 7: -4.00 + 2.00 cos(46.89 t + 0.03)
```

Note that the arguments of the cosines are given in radians, since the assignment sheet asks so. Also note that the radial frequencies are leaved as their original, in other words, no modular arithmetic is performed. The functions together with the program is available in the Appendix.

Part d

The parameters estimated by inspection in part b and the parameters estimated programmatically in part c are not the same. This is due to the sensitivity difference between the human eye and a computer. Inspecting a graph can tell you far less than a computer program that can scan the whole array of samples. Therefore, the program is expected to provide more reliable results. However, the difference is not gigantic.

Also the graph produced by the `plot` command acts as a continuous time signal, since the human eye sees the signal as a whole rather than an array of samples. Although, the computer program sees the signal as a discrete-time signal represented by a finite array of sample values. This results in different parameter estimations, consequently different numeric results.

Appendix

Complete Matlab Code

```
%% QUESTION 1
% Author: EFE ACER

load('data.mat');

%% PART A

[NUM_SIGNALS, NUM_SAMPLES] = size(X);

SAMPLING_RATE = 1000; % Hz
SAMPLING_INTERVAL = 1 / SAMPLING_RATE; % sec

% Get the signals corresponding to the last three digits of my ID
signal2 = X(2, :);
signal1 = X(1, :);
signal7 = X(7, :);

% Populate the time values
t = (0: NUM_SAMPLES - 1) * SAMPLING_INTERVAL;

% Plot the signals
figure(1);
plot(t, signal2);
title('x_2(t) versus t');
xlabel('t: Time (sec)');
ylabel('A: Amplitude');

figure(2);
plot(t, signal1);
title('x_1(t) versus t');
xlabel('t: Time (sec)');
ylabel('A: Amplitude');

figure(3);
plot(t, signal7);
title('x_7(t) versus t');
xlabel('t: Time (sec)');
ylabel('A: Amplitude');

%% PART C
```

```
fprintf('Signal 2: ');
print_signal(SAMPLING_RATE, signal2);
fprintf('Signal 1: ');
print_signal(SAMPLING_RATE, signal1);
fprintf('Signal 7: ');
print_signal(SAMPLING_RATE, signal7);

function [C, A, f, phi] = parameters_of_sin(fs, x)
% PARAMETERS_OF_SIN estimates the DC component, amplitude, frequency
% (in Hz) and phase shift (in degrees) of a sinusoidal signal, given the
% sampling rate, fs, and the uniform sample values, x.
    min_val = min(x);
    max_val = max(x);
    C = (min_val + max_val) / 2;
    A = max_val - C;
    first_pos_peak_idx = find(x > C + A - A / 1000, 1);
    first_neg_peak_idx = find(x < C - A + A / 1000, 1);
    samples_in_a_period = 2 * abs(first_pos_peak_idx - first_neg_peak_idx);
    f = fs / samples_in_a_period;
    phi = mod(first_pos_peak_idx / fs * 180 / pi, 360);
end

function print_signal(fs, x)
% PRINT_SIGNAL estimates and prints the analytical form of the
% continuos-time sinusoidal signal (with radian argument) corresponding to
% the given uniform samples, x, sampled with rate, fs.
    [C, A, f, phi] = parameters_of_sin(fs, x);
    w_radians = 2 * pi * f;
    phi_radians = phi * pi / 180;
    fprintf('%.2f + %.2f cos(%.2f t + %.2f)\n', C, A, w_radians, phi_radians);
end
```