BILKENT UNIVERSITY
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

EEE 321
Signals and Systems
Lab Assignment 4

due: 05.04.2024, Friday at 23:55 on Moodle

This lab assignment is on the Fourier series representation of continuous-time signals. You will be implementing a function for the analysis equation and observe the effects of operations in time on the Fourier series coefficients. Then, you will be working with a linear time-invariant system to understand the effects of this system on the Fourier series coefficients of the input.

Please work on this assignment before coming to the laboratory. At the end of the laboratory session, you must show your completed work for **ALL** parts. After the lab session, you will have more time to format your completed work as a report and submit it on Moodle. Some parts will be performed by hand, and others will be done using MATLAB. Please address all the questions asked in the assignment and include all your codes as text and derivations for all parts. Before the submission deadline, make sure to upload your work as a readable, well-formatted single PDF file. Note that the MATLAB codes will be tested on MATLAB R2023a. Ensure your code does not raise an error or a warning in earlier versions. Do NOT forget to add proper captions, axis labels, and titles for any plot you provide.

Part 1

In the lectures, you have learned that a continuous-time periodic signal can be expressed as a summation of harmonically related frequency components as follows:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk(2\pi/T_o)t} \qquad \text{(synthesis formula)} \tag{1}$$

where a_ks are the weight of the each kth harmonic component, $e^{jk(2\pi/T_o)t}$, contained in the periodic signal $x(t)$, or the Fourier series coefficients. Here, $\frac{2\pi}{T_o}$ is equal to the periodic signal's fundamental frequency ω_o. Corollary, you may express the Fourier series coefficients as follows:

$$a_k = \frac{1}{T_o}\int_{T_o} x(t)e^{-jk(2\pi/T_o)t}dt \qquad \text{(analysis formula)} \tag{2}$$

1.1 Implementing Fourier Series Analysis

In this part, you are going to implement Eqn.(2) on MATLAB. In other words, you will write a function that will provide you with the Fourier series coefficients of a periodic signal. To implement Eqn.(2) on a discrete platform such as MATLAB, we need to make some changes. First, notice that this equation includes an integral that can be discretized as a summation of the samples of your continuous-time periodic signal. Then, the boundaries of your summation should be adjusted to include one complete period of your sampled signal, such as from $n = 0$ to $n = N - 1$, where N is the total number of samples in a single period of your signal. Given that you have $1/T_s$ samples in one second, you can calculate the number of samples (N) in one period of your signal (T_o) using direct proportionality. Once you determine the boundaries of your summation, you may simply replace T_o with N, and continuous time variable t with sample index n in your calculations. In the end, your function should look like:

function [fsCoeffs] = FSAnalysis(x, k)

- **fsCoeffs:** An array that contains the Fourier series coefficients of your signal from **-k** to **k**
- **x:** One complete period of the sampled continuous-time signal
- **k:** The number of two-sided Fourier series coefficients that you will estimate (i.e., you will find the coefficients $a_{-k}, a_{-k+1}, \ldots, a_0, \ldots, a_{k-1}, a_k$)

1.2 Testing the Function

In this part, you will be testing whether your function works correctly or not. For this, you will first derive the Fourier series coefficients of the signals provided below. Then, you will estimate the Fourier series coefficients of these signals using the function you wrote in Part 1.1, **FSAnalysis**, and compare your results.

a) First, determine the fundamental period of $x_1(t)$ and then find the Fourier series coefficients of both signals. Make sure you calculate the exact values for at least three non-zero coefficients for each signal, as you will be comparing them with the estimated ones on MATLAB.

$$\begin{aligned} i)\ x_1(t) &= 8\cos(10\pi t) + 20\sin(6\pi t) - 11\cos(30\pi t) \\ ii)\ x_2(t) &= e^{-t} \quad \text{for } -1 < t < 1, \ \text{periodic with } T = 2\text{ s} \end{aligned}$$

b) Then, using your function **FSAnalysis**, estimate the Fourier series coefficients of the signals $x_1(t)$ and $x_2(t)$ for $T_s = 0.001$ s and $k = 30$. For each signal, plot the real and imaginary part of the estimated Fourier series coefficients in a subplot using the **stem** command. Then, compare your calculations in part a) with the values in your plots. Comment on your results.

c) Show that the Parseval's Relation holds computationally, using the output of your function **FSAnalysis** and the signal $x_1(t)$ on MATLAB. (**Note:** As your signal is the sampled version of your continuous-time signal, do not forget to multiply the summation you found for the time domain with the sampling period T_s.)

Part 2

In this part, you will be observing the effects of various operations in the time domain on the Fourier series coefficients. For this purpose, let us define the signal $x_3(t)$ as follows:

$$x_3(t) = r(t) - r(t-3) - 3u(t-3) \quad \text{periodic with} \quad T = 4\text{ s}$$

where $u(t)$ is the unit-step function and $r(t) = tu(t)$ is the ramp function. First, find the Fourier series coefficients of $x_3(t)$ using the function **FSAnalysis** and plot the real and imaginary parts of the Fourier series coefficients on a subplot using the **stem** command. You may use the same T_s and k values as provided in Part 1.2.

Then, express the Fourier series coefficients for the following signals defined in terms of $x_3(t)$. Also, obtain the Fourier series coefficients of these signals using the **FSAnalysis** function and plot the real and imaginary parts of the coefficients on the same figure as subplots using the **stem** command. Compare your plots for the coefficients of $x_3(t)$ and $z_i(t)$ signals, where $i = 1,2,3,4,5$. What changes do you observe in the Fourier series coefficients for each time-domain operation? Comment on your results for each case. (**Note 1:** As you are working with sampled continuous-time signals, if you would like to implement an impulse response, do not forget to divide its amplitude by T_s.) (**Note 2:** For part d), set the limits of the y axis as **ylim=([-1, 1.5])** in both the real and imaginary part plots to observe the effects more clearly.)

a) $z_1(t) = x_3(-t)$

b) $z_2(t) = \frac{dx_3(t)}{dt}$

c) $z_3(t) = x_3(t+2)$

d) $z_4(t) = \mathcal{E}v\{x_3(t)\}$

e) $z_5(t) = x_3^2(t)$

Part 3

3.1 A Second-Order System

The input-output relationship of the system depicted in Figure 1 can be described by the following second-order differential equation:

$$M\frac{d^2y(t)}{dt^2} + c\frac{dy(t)}{dt} + \kappa y(t) = f(t) \tag{3}$$

where $y(t)$ is the displacement with respect to the initial position of the center of mass, $\frac{dy(t)}{dt}$ is the velocity, $\frac{d^2y(t)}{dt^2}$ is the acceleration of the mass in this system and $f(t)$ is the function that defines the exerted force on the mass with respect to time. The coefficients M, c, and κ are the effective mass, damping coefficient, and stiffness of the spring, respectively.

By using the properties of continuous-time Fourier series coefficients, express both sides of Eqn.(3) in terms of their Fourier series coefficients, and then relate the Fourier series coefficients of $y(t)$, denoted by b_k, and those of $f(t)$, named a_k. Also, state this relation, in terms of the system's frequency response, $H(j\omega)$. Then, express $H(jk\omega_o)$ in terms of M, c, and κ.

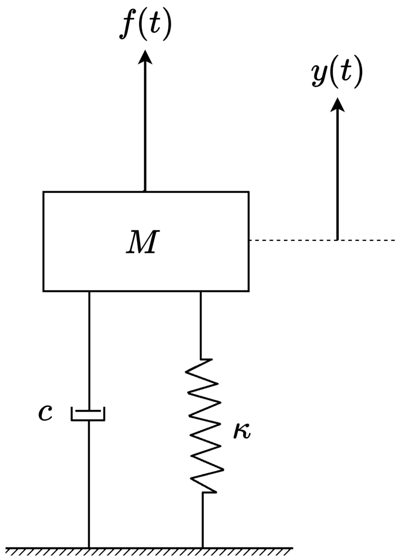

Figure 1: A depiction of the second-order physical system.

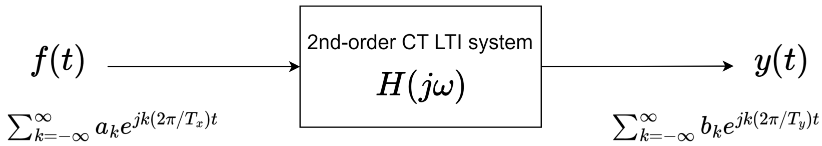

Figure 2: Abstraction of the physical system as a block diagram.

3.2 Implementation of the Second-Order System

In this part, you will be implementing the system in Part 3.1 on MATLAB. To do this, set $M = 100$, $c = 0.1$, and $\kappa = 0.1$. Also, assume that the force exerted on this system is $x_3(t)$, defined in Part 2. You may use the same T_s and k values as in Part 1.2. Assume that the system is initially at rest. Similar to the discretization you have done for the continuous-time second-order differentiation using backward approximation in Lab 3, express Eqn.(3) as a difference equation in discrete time, in terms of $y[n]$ and $x_3[n]$, using backward approximation for the first- and second-order derivatives. Then, express the output $y[n]$ as a combination of $x_3[n]$ and the previous values of itself. After expressing $y[n]$ this way, you may implement this discretized second-order physical system on MATLAB. You should obtain $y[n]$ as a result, which is the discretized version of the displacement function $y(t)$. Plot $x_3[n]$ and $y[n]$ on the same figure using **subplot**. To visualize better, you may use the **plot** function instead of the **stem** function for plotting $x_3[n]$ and $y[n]$, as they are the sampled versions of the continuous-time signals $x_3(t)$ and $y(t)$. Note that, although the implementation of such a continuous-time system on MATLAB required discretization, the relation you observe using the discretized signals and their Fourier series coefficients still holds for the continuous-time signals $x_3(t)$ and $y(t)$.

Finally, using your derivations from Part 3.1 and the **FSAnalysis** function, obtain the Fourier series coefficients of the input $x_3(t)$ and the output $y(t)$. Plot the real and imaginary parts of the Fourier series coefficients of both signals using the **stem** command on a 2×2 **subplot**. Comment on the effects of this system on the Fourier series coefficients of the input.

4 Final Remarks

Throughout this assignment, you are **NOT** allowed to use symbolic operations in MATLAB. Submit the results of your own work in the form of a well-documented lab report on Moodle. Borrowing full or partial code from your peers or elsewhere is **NOT** allowed and will be punished. The axes of all plots should be scaled and labeled. To modify the styles of the plots, add labels, and scale the plots, use only MATLAB commands; do **NOT** use the GUI of the figure windows. When your program is executed, the figures must appear exactly the same as you provide in your solution. You need to write your MATLAB codes not only correctly but efficiently as well. Please include all evidence (plots, screen dumps, MATLAB codes, MATLAB command window print-outs, etc.) as needed in your report. Append your MATLAB code at the end of your assignment as text, not as an image, and do **NOT** upload it separately. You can use the "Publish" menu of MATLAB to generate a PDF file from your codes and their outputs and append it to the end of your report. If you do this, please also indicate the part that the code corresponds to with a label. Typing your report instead of handwriting some parts will be better. If you decide to write some parts by hand, please use plain white paper. Please do not upload any photos/images of your report. Your complete report should be uploaded on Moodle as a single good-quality PDF file by the given deadline. Please try to upload several hours before the deadline to avoid last-minute problems that may cause you to miss the deadline. Please DO **NOT** submit files by e-mail or as hard copies.

Please name the PDF file you submit on Moodle as follows using only lower-case English characters for your first name, middle name (if any), and last name. Please use your full name as it appears on the Bilkent system.

LAB4_firstname_middle name_lastname.pdf
filename example for Ayşenur Çiğdem Sürücü:
LAB4_aysenur_cigdem_surucu.pdf