Lab1,EE4C5,DSP

Prof. Naomi Harte, Dr Jean-Baptiste Monteil

September 18, 2023

1 Introduction

In this lab, we provide MATLAB code snippets for each question. These are NOT the answers to the question, but you can get inspiration from the provided syntax to derive the necessary code. A basic level of expertise in MATLAB is assumed. You should take a refresher course online in MATLAB if you feel you need it. Visit https://matlabacademy.mathworks.com/details/matlab-onramp/gettingstarted if you need a course.

As you work, you should keep track of your MATLAB code and outputs. Note all instances where you are asked to make observations or answer questions. You need to submit a write up, though it does not need to be constructed as a highly formal report. This will have been discussed fully in lectures. The student must address all the questions in a PDF document, include all figures that demonstrate they did the work, and provide their MATLAB scripts created during the lab and its preparation separately (i.e. do NOT just paste code into your document). Observations can be made in the form of bulleted points or similar. Just make it clear what element of the lab any diagrams/points are referring to, e.g. by using headings. All code should be appropriately commented. Students should refer to Blackboard for the associated deadline for their assigned Lab group.

You will need headphones to listen to the signals when necessary. Before listening to any signals, please check your volume to avoid damaging your hearing.

Remember you must submit your rough work on Blackboard in ADVANCE of attending the clinic. There is a separate assignment in Blackboard for the pre- and post-clinic submissions

2 Operations on simple sequences

The goal for these initial exercises is to manipulate simple sequences in MATLAB. Specifically, we consider plotting, delaying, adding/substracting finite sequences.

1. Go to the workspace, create a new script named lab1.m

2. Plot the signal $x = \delta[n] + 2\delta[n-1] - \delta[n-3] + \delta[n-4] + 2\delta[n-5]$. Please annotate the figure, with suitable labels and a title. Ensure you have plotted the sequence as a discrete-time sequence. Be careful - MATLAB presumes the first element in an array is the $1st$ element. Your sequence begins with $n = 0$. Have you done this correctly? Include the plot in your report.

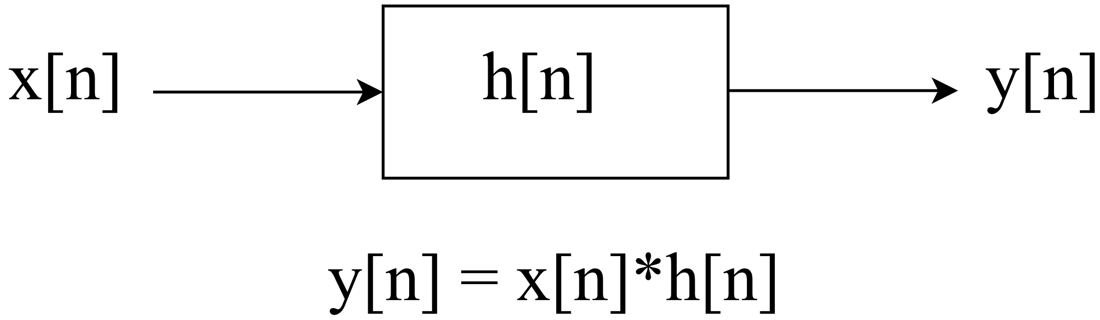
```
figure % new figure
stem(t,x) % x as a function of t
title('my title is...') % Figure title
xlabel('the label for x axis is...')
ylabel('the label for y axis is...')
title('this is a title')
xlim([0 1]) % Restrict x values between 0 and 1
ylim([-1 1]) % Restrict y values between -1 and 1
```

3. Delay the signal x by 2 samples. Plot the original signal and the delayed signal on the same figure using the $subplot$ facility in MATLAB. Advance the original signal x by 3 samples. Include all three versions of the signal in a single figure, again using $subplot$. Clearly label the three signals and note whether the delay/advance is correctly implemented. Include the figure in your report.

```
t = -2:step:10 %%create a vector starting at -2 and finishing at 10
    with an increment of step
horzcat(x,y) %%concatenate horizontally two arrays
zeros(1,3) %% create an array of 3 zeros
```

4. **Addition** of two signals. Consider the sequence $x1 = -1\delta[n-1] + 3\delta[n-2]$ and the sequence $x2 = 3\delta[n-1] + 3\delta[n-2] + \delta[n-3] - 2\delta[n-4]$. Add the two signals $x1$ and $x2$. Note they must be added in a time-aligned manner. Using $subplot$, make a new figure with the original sequences $x1$ and $x2$ and your output. Did you correctly align them? Explain briefly your method to time align the two signals.

5. Develop a function that takes as input two sequences, their respective time indices, and outputs the signal which is the sum of the two input sequences, and its time indices. It should also plot the



inputted sequences and output in a single figure. Remember to always label axes in any plot.

```
function[y,nOut] = AddSeq(x1,n1,x2,n2)
```

Demonstrate that your function works using a suitably challenging sequence. Include the figures generated by your function. Comment on how confident you are that the function will work for all inputs. **Warning:** to write the function, create another script named as the function, i.e. AddSeq.m, and test the function in the main script lab1.m

6. **Convolution** is a mathematical way of combining two signals to form a third signal. In linear systems, it relates the input signal $x[n]$, the output signal $y[n]$, and the *impulse response* $h[n]$. If the system being considered is a filter, the impulse response is called the filter kernel, the convolution kernel, or simply, the kernel.

For discrete signals, the convolution is defined as:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{1}$$

Consider the signal $x[n] = \delta[n+3] - \delta[n+1] + \delta[n-1] - \delta[n-3]$, the impulse response

$$h_0[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Plot $x[n]$ and $y[n] = x[n] * h_0[n]$. Include the plots in your report. Compare $x[n]$ and $y[n]$. What do you notice? Please explain the effect of $h_a[n]$ on the input signal ($h_a[n] = 1$ for $n = a$, 0 otherwise).

7. **Optional Challenge** with a longer sequence. The .cvs file $IRL\_DLY\_RR\_2021\_grid.cvs$ contains rainfall data from Met Eireann.This particular file contains values on a 1x1km grid Irish Grid (TM65) covering the land of Ireland for the year 2021. If you open the file (in excel or similar) you can see

headers for the date and grid references. Interpolation models were used to calculate the rainfall and temperature grid point values with model inputs from observations recorded at Met Éireann stations. The daily rainfall total is presented without a decimal point. For example, a value of 123 is equivalent to a daily rainfall total of 12.3 mm.

You need to isolate the annual rainfall from a single grid point and use that as one sequence. Plot the sequence to check you have isolated it correctly (sanity check - how many days are in a year?) Then generate another sequence $h[n]$ which when convolved with the rainfall data sequence, will behave as a moving average filter. Convolve the two sequences and plot the output. Compare it to the original rainfall sequence. What can you observe? Experiment with longer and shorter lengths for the $h[n]$. What can you conclude about the weights of $h[n]$ in order to truly calculate a meaningful average value? In your write up, show a few suitably annotated plots with the original sequence of rainfall, the h[n] used, and the convolution output.

```
T = readtable('myfilename.csv'); %built in matlab function to
    read teh cvs file including headers
Tarray = table2array(T); % strip headers
s = Tarray(:,227); % row or column?
s = Tarray(227,:); % row or column?
s = Tarray(117,3:end); % skip first two elements
y = conv(x,h) % convolve two sequences
```

3 Sine wave model

We consider the following continuous signal:

$$x(t) = \sin(2\pi f_0 t) \tag{3}$$

It is a pure sinusoid with frequency f_0, amplitude 1, phase 0. The signal corresponds to a music note. The higher the frequency is, the more high-pitched the sound is. For instance, a sinusoid with frequency $f_0 = 440$ Hz corresponds to A/*la*. This analogic signal cannot be studied in MATLAB (it contains an infinite number of samples and values are not quantized), hence we can study the sampled and quantized version of this signal. We define the signal $x[n]$ sampled with sampling frequency

f_s, which contains N samples, as:

$$x[n] = \sin(2\pi f_0 t[n]) \tag{4}$$

The times $t[n]$ are defined as:

$$t[n] = \frac{n}{f_s} \quad for \quad 0 \leq n \leq N-1 \tag{5}$$

By default, MATLAB quantizes all values with 64 bits. In this lab, the considered signal has the following features: sampling frequency $f_s = 8000$ Hz, fundamental frequency $f_0 = 800$ Hz, duration $d = 2$ secs.

4 Operations on sine waves

The goal is to simulate your first numeric signal with MATLAB.

1. Go to the workspace and create the script lab1_part2.m

2. Determine the relationship between N, d and f_s. Enter the 3 variables in the MATLAB code to determine their numerical values.

How do you choose N when d is not an integer, so that you do not lose information? *Hint:* there is no problem to append zeros at the end of the signal

```
a = round(b); %round b to the closest integer
a = floor(b); %round b to the inferior integer
a = ceil(b); %round b to the superior integer
```

3. Construct in MATLAB the time vector t of size N that contains the different values $t[n]$. **Warning: in this lab, we work with column vectors!**

```
%% Sample Matlab code
signal = first:step:last;
x1 = 0:0.1:1;  %Row vector 0, 0.1, 0.2, ...
x2 = 3:8; %Row vector 3, 4, 5, ...
x3 = (0:5)'; %Column vector 0, 1, 2, ...
```

4. Construct the signal x containing the values $x[n]$ (the signal which is the sampled version of $x(t)$).

```
z = y+2; % we add an offset 2 to all items of y
z = 5*y; % we multiply all values of y by 5
z = y.^3; % all values of y raised to the power 3
z = log(y); % logarithm of all values
z = abs(y); % absolute values
z = cos(y/7);
```

5. Listen to x with headphones. Is it deep? High-pitched? Try with different values of f_0 (between 300 and 1200 Hz). What can you infer about the physical meaning of frequency?

```
sound(x,Fs)
soundsc(x,Fs)
```

6. What is the fundamental period T_0 of the signal $x(t)$ before sampling? Plot the sampled signal $x[n]$ in the interval $[0, 4T_0]$ and annotate the figure. Try for different values of f_0 between 300 and 1200 Hz. Please include one figure, complete with annotation in your report.

```
figure % new figure
stem(t,x) % x as a function of t
title('my title is...') % Figure title
xlabel('the label for x axis is...')
ylabel('the label for y axis is...')
xlim([0 1]) % Restrict x values between 0 and 1
ylim([-1 1]) % Restrict y values between -1 and 1
```

7. Delay the signal x ($x[n]$) by 2 seconds. We recall that the unit of f_s is samples/sec. Hint: you need to convert delay in seconds to delay in samples.

```
delayed_signal = delayseq(signal,delay); %delays or advances the
    signal in data by the number of samples specified in delay.
    Positive values of delay delay the signal, while negative values
    advance the signal.
delayed_signal = sin(2*pi*(n-delay)*fo/fs); %delays or advances the
    signal in data by the number of samples specified in delay.
    Positive values of delay delay the signal, while negative values
    advance the signal.
```

8. Add to the signal x:

- another signal $y1$ with fundamental frequency $f_0 = 400$ Hz, and sampling frequency $f_s = 8000$ Hz, and duration $d = 2$ seconds;
- another signal $y2$ with fundamental frequency $f_0 = 400$ Hz, and sampling frequency $f_s = 8000$ Hz, and duration $d = 4$ seconds (x and $y2$ start simultaneously);
- another signal $y3$ with fundamental frequency $f_0 = 400$ Hz, and sampling frequency $f_s = 8000$ Hz, and duration $d = 2$ seconds (x starts at $t = 0$ seconds, $y3$ starts at $t = 1$ seconds).

Warning: sequences to be added or multiplied must be of equal length, but also properly time-aligned.

In your submission, include a plot with all the above sinusiods with correct time alignment with the x axis labelled in units of **seconds**.

```
z=x+y; %add two vectors element wise
vertcat(x,y); %concatenate vertically two column vectors
```

9. Same questions as the bullets in 4.8. where you **multiply** x and y. Propose a Matlab function that takes as input two signals and their respective time length, and outputs the sum/multiplication of the signals. Note that the two signals can have different duration but have the same sampling frequency.

```
z=x*y; %multiply two vectors element wise
```

10. Same questions as the bullets in 4.8 and 4.9. where you **convolve** x and y. Propose a Matlab function that takes as input two signals and their respective time length, and outputs the convolution of the signals. Note that the two signals can have different duration but have the same sampling frequency.

```
z=conv(x,y); %convolution of vectors x and y
```

5 Write-up for (post-clinic) submission

The purpose of this lab is that you engage with the assigned tasks, get a better appreciation of the real-world analysis of signals, and make connections between what you are learning in lectures and practical use of those concepts. Therefore, you should not be spending a huge amount of time on the write-up. The purpose of the submission is to show you have done all the assigned tasks, that you have thought about the questions asked, and that you are achieving the required level of proficiency in MATLAB.

With this in mind, you should submit:

- A single pdf with all required figures and commentary on the tasks in the lab. Bulleted comments are allowed.
- The pdf above must use a sans serif font, and adequate font size and spacing.
- You must include the declaration on plagiarism as per your course handbook.
- You can use snippets of code in the pdf to illustrate a point you wish to make, but all code must be submitted in .m files.
- Ultimately, we should be able to run your code and recreate your plots etc from what you submit.
- All code should be commented.