[5ESE0] Lab 7: Signal processing system

1 Introduction

In this final Lab of the Signal Processing Basics (5ESE0) course, each participant has to make exercises similar to the assignments of previous Labs of this course. Each participant has to make exercises, sometimes by hand and other times in Matlab, to derive and evaluate (intermediate) results. All the findings must be reported in a, logically structured, technical report consisting of clear formal text, equations, tables and plots.

The main goal of this lab is to analyse (mainly by hand) and implement (mainly in Matlab) a basic signal processing system and report the main results and conclusions in a short technical report in English.

2 General information

The final grade of this report will count for 20% of 5ESE0. Furthermore this report also counts for the "Writing 1"-skill [PRV31], resulting in a PASS/FAIL (in Dutch "VO/ON"="VOldoende/ONvoldoende").

IMPORTANT: Exercises that require Matlab (Exercise 1 a,b,c,d; Exercise 2b; Exercise 3b; Exercise 4 a,b,c; Exercise 5a) must be described in the report as well as submitted in Matlab Grader. In case one of the two is missing, the Exercise will be assessed as incorrect. Make sure that the code that you submit in Matlab Grader produces exactly the same results that you use in the report. For example, if your axis have labels in the report, but the code in Matlab Grader does not assign these axis labels, the figure will be considered incorrect.

2.1 Grading criteria [5ESE0]

The technical report will be graded as follows: For each sub-exercise the maximum score is given. In total the maximum score equals **29 (+ 1 bonus) points**. The final grade of is calculated by dividing the resulting number of points by 2.9. The **maximum possible grade is 10**. Note here that the number of points that you can score for each sub-exercise is not necessarily related to the number of tests that is performed for that same sub-exercise in Matlab Grader. For example, in an Exercise for which you can score 3 points it might be that only 1 point is based on the Matlab submission and the other 2 are based on the rest of your answer. The report must have a logical structure containing the following elements:

Title:	*Must be to-the-point*
Abstract:	*Must summarize all essential parts of the report; Max 100 words.*
Introduction:	*Describes clearly background and purpose of report; Max. 10 sentences.*
Technical content:	*Discuss the different exercises of the lab short and clear* *Discuss results shortly, show the relation with theory* *Show correctness of Matlab code*
Text:	*Use formal text (do not use "I" or "we", do not address the reader)* *Use correct syntax*
Equations:	*Are part of the text* *Are numbered* *Are on a separate line*
Figures and Tables:	*Are not part of the text* *Are numbered* *Have a clear caption* *Have to be referred to in the text*
Conclusions:	*Should be sufficiently supported by evidence in the report* *No new information is given*

A more detailed description of each of these elements has been made available on Canvas during the course.

2.2 Grading criteria [PRV31]

The "Writing 1"-skill [PRV31] will be graded according to the following Rubrics table:

Criterion	Insufficient	Sufficient	Good
Quality of the message	Problem analysis is incorrect. The text is technically wrong and/or factually incorrect. None of the statements in the report are supported by evidence or references.	Problem analysis is unclear and/or incomplete. Most of the text is technically sound and factually correct. Most of the statements in the report are supported by evidence in the report itself or by references.	Problem analysis is clear and complete. The text is technically sound and factually correct. All statements in the report are supported by evidence in the report itself or by references.
Structure	There is no structure in the report. There is no distinction between the introduction, body, or conclusion. The structure does not help to understand the content of the message.	There is some structure in the report. There is some distinction between introduction, body, and conclusion.	The report contains an introduction, body, and conclusion.
Use of language/ formulation	Formulation is not according to scientific writing principles. There are many spelling errors and/or sentences are incomplete or illogical. The report is not easy to read.	Some formulations are according to scientific writing principles. Most of the language used is correct without flaws. Most of the report is easy to read.	Formulation is according to scientific writing principles. The language used is correct and without flaws. The report is easy to read.
Lay-out	There is no title page. The text is not divided into different paragraphs. No attention is paid to the lay-out of the report.	The title page contains most of the necessary information. Parts of the text are divided into different paragraphs. Some attention is paid to the lay-out of the report.	There is a title page which contains the necessary information (title, student's name and ID, course, date). The text is divided into different paragraphs. A formal font and font size is used. The report looks professional.
Tailored to target group	The report reads like a diary instead of a technical report. The tone of the text is too colloquial / informal. Often, 'I','we', etc is used.	Some of the text is colloquial / informal. Sometimes 'I','we', etc is used.	The scientific level of the report is tailored to fellow scientist. The report does not contain colloquial / informal language. No 'I','we', etc is used.

2.3 Some general notes

- This lab is an individual assignment, thus each individual participant has to make this lab and write a technical report.
- The report must have a clear frontpage which contains a title, name and ID.
- Excluding frontpage, the report may not exceed 6 pages (font ≥11pt).
- The report must be uploaded in Canvas before Thursday January 17, 2019 at 17:15.
- A printout of this uploaded version has to be delivered before Thursday January 17, 2019 at 17:15 during the contact hours in your own tutor room.
- In case you fail PRV31, you are only entitled to a re-exam for this Lab when
  - the sum of the grades for 5ESE1 and 5ESE2 is at least 8 (SUM(5ESE1,5ESE2)≥8) **AND**
  - the lowest grade for 5ESE1, 5ESE2, and 5ESE3 is at least 3 (MIN(5ESE1,5ESE2,5ESE3)≥3) Note: 5ESE1 is the written exam, 5ESE2 is Lab7, and 5ESE3 is the average grade for Lab2-6.
- The report will NOT be graded in case:
  - The frontpage does not contain all information as described above;
  - The report is uploaded after the deadline;
  - The printout is delivered after the deadline;
  - The printout version is not the same version as the uploaded version.
- Proven plagiarism will result in exclusion of the course. Note that your report can also be compared for plagiarism to reports of students from previous years.
- Each participant might be invited to explain and/or defend his/her report in case of unclarities.
- When you are asked to play a sound in Matlab, you must describe what you hear and, when appropriate, compare what you hear to what you heard in earlier exercises. Always make sure to include your Matlab code for playing the sound in the appendix.

3 Some useful Matlab hints

- **Playing sounds** In the exercises, you will have to play sound files. You can do this via `sound(x,fs)` or (better) `soundsc(x,fs)`, where `x` is the sound file and `fs` the frequency at which the sound file was sampled. When playing a sound you have to take care of the sampling frequency `fs` since the sound should be audibly (human audio spectrum 20 [Hz] −− 20 [kHz]). Obviously you should use a play-time which is not too short or not too long. It is strongly advised to read the `help sound` or `help soundsc` in Matlab. Furthermore you should try the following command: `load handel`, which loads a piece of a music signal in the Matlab workspace. Find out the number of samples and sample frequency of this sound file. From these numbers you can calculate the play-time. Use `sound(y,Fs)` or `soundsc(y,Fs)` to verify the calculated play-time by measuring the length when playing the sound.

  You should realize that Matlab can only work with numbers. So if you want to play a continuous-time sound file in Matlab you have to convert it first to the discrete-time domain. Usually this is done by using a sample rate f_s which is high enough ("$f_s \gg 2\times$ highest frequency present in the signal"). For example the following piece of Matlab code, plays a "pseudo" continuous-time sinusoidal signal of 500 [Hz] at a sample rate of 5000 [Hz]:

```
%Play a "pseudo" continuous-time signal
fs=5000;
Ts=1/fs;
t=0:Ts:2;
f0=500;
xt=sin(2*pi*f0*t);
sound(xt,fs)
```

- **Plots or stems** A plot of a "pseudo" continuous function is typically made by using the Matlab function `plot`, while discrete signals are typically plotted by using the Matlab function `stem`.

  The following piece of Matlab code defines a "pseudo" continuous-time 5 [Hz] sinusoidal signal $x(t) = sin(2\pi 5t)$ in the time-span $t = 0 - 0.5$ [sec]. Furthermore it defines a discrete-time version $x[n] = \sin(0.1\pi n)$ in the index range $n = 0 - 50$.

```
%"pseudo" continuous-time
fs=100;
Ts=1/fs;
t=0:Ts:0.5;
f0=5;
xt=sin(2*pi*f0*t);

%discrete indices
n=0:50;
xn=sin(0.1*pi*n);

figure;
plot(t,xt);
hold on; %make sure that stem does not overwrite the plot
stem(n*Ts,xn);
xlabel('n{\cdot}T_s or t');
ylabel('Amplitude');
```

  The Matlab code also describes how to plot both signals $x(t)$ (using `plot`) and $x[n]$ (`stem`), which results in the figure as depicted in Figure 1.

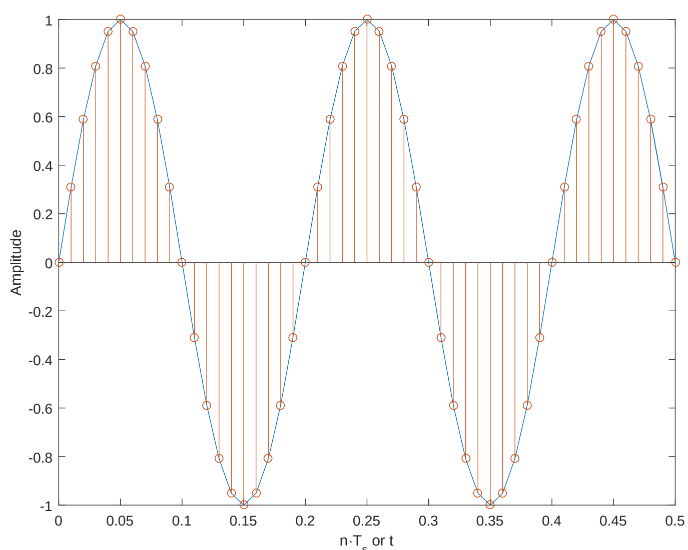

Figure 1: Example of plot and stem.

- **Axes of plots** In case you want to make a plot of e.g. a frequency response within the fundamental interval, you can use the following example code to have π on your horizontal axis.

```
ww=-pi:0.001:pi; %frequency axis
tt=sinc(ww); %arbitrary function for vertical axis
figure;plot(ww,abs(tt))
set(gca,'XLim',[-pi,pi],'xtick',[-pi:0.5*pi:pi],...
'XTickLabel',{'-\pi','-0.5\pi','0','0.5\pi','\pi'}) %set limits, tick positions,
and tick labels for plot
```

- **Using the find function** In Exercise 4, you will need to use the `find` function to locate specific points in a vector. Let's assume you have a frequency response/function that looks like the sinc from the previous Matlab example and you need to find the two smallest absolute values for x for which $sinc(x)$ is zero. In other words, find the first zero crossings on either side of the main lobe of the sinc function. You can do this via:

```
xx=-pi:0.001:pi; %frequency axis
tt=sinc(xx);
[~,posMaximum]=max(abs(tt));
zeroCrossing1=find(tt(posMaximum:end)<0,1,'first'); %look forwards
zeroCrossing2=find(tt(posMaximum:-1:1)<0,1,'first'); %look backwards

%correct for using only part of tt as argument in find function
zeroCrossing1=posMaximum-1+zeroCrossing1;
zeroCrossing2=posMaximum+1-zeroCrossing2;

%the variables "zeroCrossing" are now the index in the vector tt where the
%value of tt first becomes negative on either side of the main lobe.
%However, you were asked to give the value of x. So determine which value
%of x corresponds to these indices
xx(zeroCrossing1)
xx(zeroCrossing2)
```

In case the values of xx in the code above are not equal to -1 and +1, the reason for this is that the resolution of xx is not high enough. Note that the Matlab function `sinc` scales the input by a factor π, as compared to the conventional sinc function:

$$\text{MATLAB :} \qquad \text{sinc}(x) = \frac{\sin(x\pi)}{x\pi}, \tag{1}$$

$$\text{Normal :} \qquad \text{sinc}(x) = \frac{\sin(x)}{x}, \tag{2}$$

4 Signal processing system

Consider the signal processing system as illustrated in Fig. 2. The input continuous-time signal

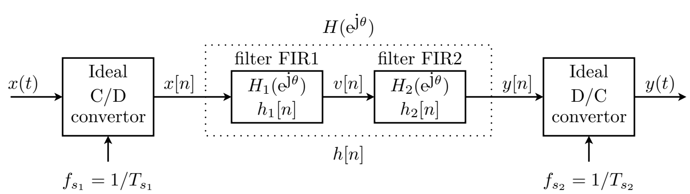

Figure 2: Signal processing system.

$x(t)$ consists of a sum of three sinusoids:

$$\begin{aligned} x(t) &= 2\cos\left(2\pi 200t + \frac{\pi}{3}\right) + 5\cos\left(2\pi 350t + \frac{\pi}{2}\right) + 5\cos\left(2\pi 600t + \frac{\pi}{4}\right) \quad (3) \\ &= x_1(t) + x_2(t) + x_3(t). \quad (4) \end{aligned}$$

The ideal C/D convertor runs at a sample rate of f_{s_1} [Hz] and produces the discrete-time samples $x[n]$. The samples $x[n]$ are first filtered by a Finite Impulse Response (FIR1) filter, with impulse response $h_1[n]$ and frequency response $H_1(e^{j\theta})$. This FIR1 filter is a so-called nulling filter and it produces the samples $v[n]$. These samples are filtered by a filter (FIR2), with impulse response $h_2[n]$ and frequency response $H_2(e^{j\theta})$. This FIR2 filter is a bandpass filter and it produces the samples $y[n]$. Finally these discrete-time output samples $y[n]$ are converted back to the continuous-time output signal $y(t)$ with an ideal D/C convertor, which runs at a sample rate of f_{s_2} [Hz].

The goal of the Lab is to calculate and/or evaluate all intermediate steps of this signal processing system and report your findings in a, logically structured, technical report consisting of clear formal text, equations, tables and plots.

5 C/D converter

Exercise 1

a, [1 pt]) Use Matlab to play 2 [sec] of the "pseudo" continuous-time input signals $x_1(t)$, $x_2(t)$, $x_3(t)$, and $x(t)$ at a sample frequency of 2500 [Hz].

b, [3 pt]) The C/D converter in Fig. 2 operates at a sampling frequency of $f_{s_1} = 1/T_{s_1} = 1000$ [Hz]. After sampling, each of the sinusoids in $x(t)$ can be written in the following general form:

$$x_i[n] = A_i \cos(\theta_i n + \phi_i) \qquad \text{for } i = 1, 2, 3 \quad (5)$$

in which the normalized frequency θ_i is in the Fundamental Interval (FI), $\theta_i \in (-\pi, \pi]$.

Calculate by hand for the sinusoids $x_i[n]$ the normalized frequency θ_i, the amplitide A_i and phase ϕ_i, all for $i = 1, 2, 3$. Also determine whether aliasing has occurred. Represent your findings in a table with accompanying explanation.

c, [2 pt]) Use Matlab to play the discrete-time signals $x_1[n]$, $x_2[n]$, $x_3[n]$ and $x[n]$ at frequency of $f_{s_1} = 1000$ [Hz] and compare what you hear with the sounds of the "pseudo"continuous-time signals $x_1(t)$, $x_2(t)$, $x_3(t)$, and $x(t)$ of the first exercise. Make sure that the sounds last exactly 2 [seconds].

d, [2 pt]) Use Matlab to calculate the samples of $x[n] = \sum_i x_i[n]$ and make a plot in which you show the samples $x[n]$ on the vertical axis and $n \cdot T_{s_1}$ on the horizontal axis. Show in the same plot (or subplot) the signal $x(t)$ as a function of t. Define the limits of your horizontal axes such that you show the same fragments of $x[n]$ and $x(t)$.

6 Nulling filter FIR1

Recall from the previous Labs that a length 3 FIR nulling filter can be described by the following Difference Equation (DE):

$$v[n] = x[n] - 2\cos(\theta_{nul})\, x[n-1] + x[n-2], \quad (6)$$

in which θ_{nul} is the nulling frequency. In this exercise we will use $\theta_{nul} = 0.4\pi$.

Exercise 2

a, [2 pt]) Calculate by hand an analytical expression for the impulse response $h_1[n]$ and the frequency response $H_1(e^{j\theta})$ of the nulling filter FIR1.

b, [3 pt]) In general we can write the frequency response as follows:

$$H(e^{j\theta}) = \left|H(e^{j\theta})\right| \cdot e^{j\phi_H(e^{j\theta})}, \tag{7}$$

in which $\left|H(e^{j\theta})\right|$ represents the magnitude- and $\phi_H(e^{j\theta})$ represents the phase-response.
Show analytically that the frequency response $H(e^{j\theta})$ can be written as:

$$H(e^{j\theta}) = e^{-j\theta}\left[2\cos(\theta) - 2\cos(\theta_{nul})\right]. \tag{8}$$

In order to make a "smooth" plot, use Matlab to evaluate $\left|H(e^{j\theta})\right|$ for 501 different, equally distributed, values of θ in the FI ($|\theta| \leq \pi$). Plot in 2 subplots (both in the FI $|\theta| \leq \pi$) the magnitude- and phase-response of the nulling filter FIR1 and show that indeed the filter FIR1 nullifies the desired frequency $\theta_{nul} = 0.4\pi$.
Note: In this exercise it is not allowed to use the `freqz` function

Exercise 3
a, [1 pt]) In the ideal case the nulling filter nullifies one of the frequencies of signal $x[n]$, while the other signals will not be affected. Give for the ideal case an analytical expression for the output samples of FIR1. These samples are denoted by $v_{th}[n]$. Use Matlab to play $v_{th}[n]$ at a sample frequency f_{s_1}.

b, [2 pt]) Use Matlab to generate the samples $v[n]$, which are the output samples of FIR1. Furthermore use Matlab to play $v[n]$ at a sample frequency f_{s_1} and verify if indeed one frequency has been nullified (e.g. compare with the sound of $v_{th}[n]$).
Explain possible differences between the theoretical samples $v_{th}[n]$ and actual output samples $v[n]$ of FIR1. Give more than one possible difference.

7 Bandpass filter FIR2

A BandPass Filter (BPF) passes frequencies in a region around a certain specified center frequency. The filtercoefficients for the BPF FIR2 which is used in this exercise, are defined as follows:

$$h_2[n] = \frac{2}{L}\cos(n\theta_c), \qquad \text{for } n = 0, 1, \cdots, L-1 \tag{9}$$

where θ_c is the center frequency of the passband and the integer number L is the filter length. Thus for $\theta_c = \frac{\pi}{4}$ and $L = 4$ the BPF FIR2 has the following impulse response:

$$h_2[n] = \frac{1}{2}\delta[n] + \frac{1}{4}\sqrt{2}\delta[n-1] - \frac{1}{4}\sqrt{2}\delta[n-3] \tag{10}$$

and a realization scheme is depicted in Fig. 3.

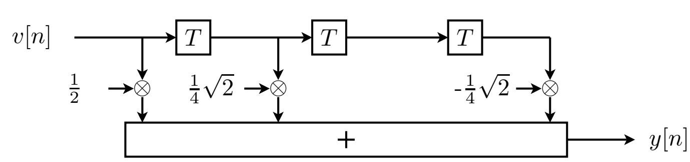

Figure 3: Realization scheme of an example BPF with $\theta_c = \frac{\pi}{4}$ and $L = 4$.

The *passband* of the BPF is defined by the region of the frequency response where $\left|H_2\left(e^{j\theta}\right)\right|$ is close to its maximum value. If we define the maximum to be $H_{2,max}$, then the passband width W_{pb}

is defined as the width of the frequency region where the ratio $\left|H_2\left(e^{j\theta}\right)\right|/H_{2,max}$ is larger than $1/\sqrt{2} \approx 0.707$ (see also Fig. 4 for an illustration of the passband for the BPF of Eq. (10)). You can obtain the passband width W_{pb} by using the Matlab function `find` to locate the frequencies for which the ratio exceeds $1/\sqrt{2}$.

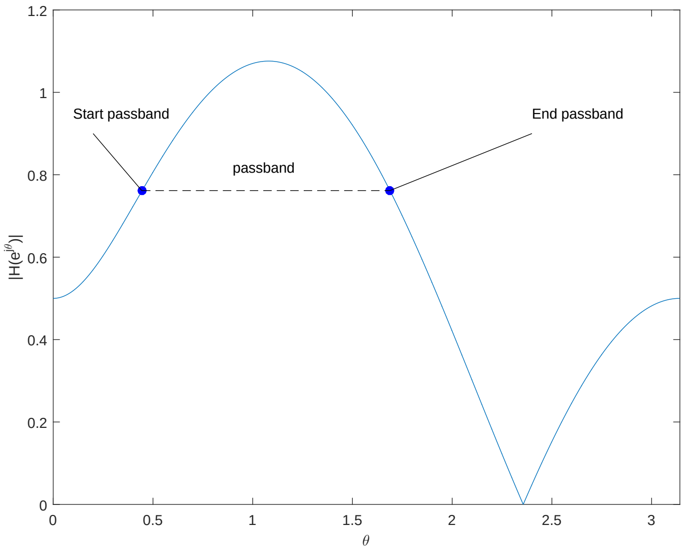

Figure 4: Passband of the BPF with $\theta_c = \frac{\pi}{4}$ and $L = 4$.

In the exercises below we use a BPF with $\theta_c = 0.7\pi$.

Exercise 4

a, [3 pt]) Use the Matlab function `freqz` to make a "smooth" plot of the magnitude response $|H_2(\mathrm{e}^{\mathrm{j}\theta})|$ of the BPF FIR2. Place in one plot (e.g. with different colors) the result for the values $L = 10$, $L = 20$ and $L = 40$. Determine for each of these 3 different BPF's the passband width W_{pb}.
Which conclusion can you draw from these results for the passband width W_{pb} when the filter length L is doubled or halved?

b, [2 pt]) Use Matlab to find the smallest value L_{min} of the BPF filter length that reduces the amplitude of a sinusoidal signal with frequency $\theta = 0.82\pi$ [a] by a factor of 10 or more. Use Matlab to make a "smooth" plot of the magnitude response of the resulting BPF and use this plot to explain how this BPF filter can pass one component and suppress or attenuate another component. Note: this new BPF still has $\theta_c = 0.7\pi$.

[a] *Note that the sinusoidal signal with frequency $\theta = 0.82\pi$ is merely "theoretic" and is not necessarily a component in $v[n]$.*

c, [2 pt]) Use Matlab to filter the output samples $v[n]$ of FIR1 with filter FIR2, which is the BPF of the previous sub-question. The obtained output samples are denoted by $y[n]$. Use Matlab to play $y[n]$ at a frequency of f_{s_1} [Hz] and explain what you hear.

8 D/C converter

The output samples $y[n]$ are converted to the continuous-time domain signal $y(t)$, as illustrated in Fig. 2. In the following exercise we assume $y[n] = x_2[n]$ (see Eq. (5)).

Exercise 5
a, [2 pt]) Evaluate by hand and analytical expression for the continuous-time signal $y(t)$ for the following three different cases: $f_{s_2} = 800$ [Hz] ($f_{s_2} < f_{s_1}$), $f_{s_2} = f_{s_1} = 1000$ [Hz] and $f_{s_2} = 1200$ [Hz] ($f_{s_2} > f_{s_1}$). Use Matlab to play these three different "pseudo" continuous-time signals at a sampling frequency of 2500 [Hz] and compare these results with the sound of the original "pseudo" continuous-time signal $x_2(t)$.

9 Overall system

Exercise 6
a, [2 pt]) Describe shortly (max 1/2 page) your findings of the overall signal processing system as illustrated in Fig. 2.

b, [2+1 bonus pt]) Formulate shortly (max 1/4 page) a possible suggestion for reducing the complexity of the signal processing system of Fig. 2.