# Lab Verification Test

**SCIPER number: __________**

This computer-based test requires the following equipment:

- Arduino IDE
- Matlab

The test has the duration of three hours. It consists of three parts: Signal Processing, worth 60 points, Systems Theory and Arduino, both consisting of 30 points. The maximum score you can get is 120 points; you will be awarded the maximum grade if you obtain 100 or more points; your potential bonus of points above 100 will be integrated in the overall weighted sum for the course grade.

The test is open book, i.e. all printed/written material is allowed (e.g., books, lecture notes, exercises, solutions, personal notes). In addition, you will use a computer from the computer room running Ubuntu, as you did in the labs. You are not allowed to use your personal computers. You can use internet to look for the necessary information, but **never to communicate with other students or anyone outside the computer room. Note that we will log all the communications and connections that your computer will create. Please, before starting, log out and close all mail and messaging services (Gmail, Yahoo, EPFL mail, Facebook, etc.).**

This intermediate test is available in English only and must be answered in English. Note that we do not grade on grammar as long as the meaning of the sentence is clear.

## Getting Started

To start with this test, you will need to **download** the material available on Moodle. Download `material.zip` and extract it in your home directory. The uncompressed folder `material` has the following contents:

- *part1*
  - *fft*
    - *part1_fourier.m:* Implement your code here (Questions 2 & 3)
    - *signal_y.mat*
  - *convolution*
    - *part1_convolution.m:* Implement your code here (Questions 6 & 7)
  - *filtering*
    - *analyze_signal.m*
    - *filter_data.m*
    - *signal.mat*
- *part2*
  - *part2.m:* Implement your code here (Questions 12 & 13)
- *part3*
  - *lvt_arduino*
    - *lvt_arduino.ino:* Implement your code here (Questions 16)
  - *part3.m: Implement your code here.* (Questions 20 & 21)
  - *acc_data.csv*

You will also need to **download** the empty answer sheet *SIS_22-23_test_answer_sheet.odt* and fill your answers into this file.

### Submitting your answers

Answers must be written on the provided answer sheets file (*SIS_22-23_test_answer_sheet.odt*), and the code implemented in the files of `material.zip` as required. Submit these two files and rename them as:

- *SIS_22-23_test_answer_sheet_yourfirstname_yourlastname.odt*
- *material_yourfirstname_yourlastname.zip*

**Both the answer sheet and zip file must be submitted via Moodle** by the end of the test. Please **do not change** the folder structure and **do not copy your code into the answer sheet if not explicitly asked for.**

Make sure that all the code that you submit **compiles/runs (warnings are acceptable)**. Otherwise, there will be a **significant penalty**.

The questions have the following notation, similar to your lab exercises:

- **S**: The question can be solved using only additional simulation or simple operation with the computer.
- **Q**: The question can be answered theoretically, without any simulation or computer use.
- **I**: The problem has to be solved by implementing and/or compiling some code and running it for validation.

**Please do not go beyond this page before the test start time.**

**This page is intentionally left blank.**

**Please do not go beyond this page before the test start time.**

# Part I: Signal Processing (60 points)

## Fourier transform and signal reconstruction (21pts)

Before answering the following questions, you will need to go to the folder *part1/fft*. There you have a collection of signals that are needed to complete the next exercises. Solve all implementation questions in *part1_fourier.m*.

Consider the following signal expressed in continuous time domain as:

$$y(t) = \cos(2\pi * 1 * t) + \cos(2\pi * 3 * t + 0.5) + \cos(2\pi * 5 * t)$$

1. **(Q)** (4pts): From the expression for `y(t)`, how many main frequencies does the signal have? Write down these main frequencies `[f1, f2, …, fn]`.

2. **(I)** (4pts): Load *signal_y.mat* from the folder *part1/fft* into Matlab. This file contains a digital signal `y[n]`, corresponding to the signal `y(t)` sampled using a sampling frequency of F_s = 1000Hz. By looking at `y[n]`, how many samples does it have? For how many seconds was y(t) sampled?

3. **(I)** (6pts): Compute the FFT of `y[n]` using the `fft()` function. Plot the amplitude of the computed FFT (single-sided amplitude spectrum) <u>without using interpolation</u> and zoom (reasonably) into the frequencies of interest. Insert your plot in the answer sheet.

4. **(Q)** (3pts): For each main frequency `[f1, f2, …, fn]` that you identified in Question **1**, what is the corresponding phase of the Fourier transform `[Φ1, Φ2, …, Φn]`. **Justify your answer.**

5. **(Q)** (4pts): What would be the minimum sampling frequency that we could use to sample this signal, in order to be able to correctly reconstruct the signal afterwards? What happens if the sampling frequency is lower than this minimum? **Justify your answer.** (*Note: if you didn't find the signal's frequency, just refer to it as f in your answer*).

## Convolution (15pts)

Before answering the following questions, you will need to go to the folder *part1/convolution*.

Consider the following two signals:

$$x[n] = 3 * cos(0.5n) \qquad\qquad h[n] = \begin{cases} 1, & n = 0 \\ -1, & n = 1 \\ 0, & otherwise \end{cases}$$

6. **(I)** (8pts): Plot both signals in Matlab between $n = [-50{:}50]$, <u>without interpolation</u>. Insert your plots in the answer sheets. (Note: $n$ is an integer, use Matlab command `zeros(1, length(x))` to generate a vector of the same length).
7. **(I)** (7pts): Using Matlab, compute the discrete convolution of $h[n]$ and $x[n]$, i.e., $y[n] = (h * x)[n]$. Report your results ($y[n]$) by plotting the result for $n = [-50; 50]$ and insert the plot in the answer sheet.

## Filtering (24pts)

In Lab 4, you learned about filters and how to design them using the `filterDesigner`. In the following questions, you will have to load and analyze a signal, and filter it using `filterDesigner`. In folder the *part1/filtering* you will find the following files:

- *signal.mat* – Matlab file containing a variable *signal* corresponding to the signal, which is sampled at 8 kHz.
- *analyze_signal.m* – Matlab function similar to *analyzeSoundSignal.m* (Lab 4), that plots the signal and its Fourier transform. *Note: here, the frequency domain plots only show half of the frequency spectrum (first quadrant), and the frequencies are properly scaled.*
- *filter_data.m* – Matlab function to filter the data using the filter you will design.

**8.** **(Q)** (6pts): You just recorded a piece of sound, but unfortunately, during your recording the fire alarm went off and ruined your recording. Luckily, the fire alarm is only a tone with two frequencies and can be filtered out. Go to folder *part1/filtering,* load *signal.mat* and the variable `signal` will contain the ruined audio signal and analyze the signal using the function *analyze_signal.m*. You should clearly see the two frequencies of the fire alarm tone. What type of filter would you implement (low pass, high pass, band pass or band stop) to remove the fire alarm tone? **Justify your answer.**

**9.** **(I)** (12pts): Use `filterDesigner` to implement the filter of Question **8**. Set the following parameters:

- *Response Type:* according to your answer in Question **8**
- *Design Method:* FIR – Equiripple.
- *Filter Order:* Minimum order
- *Options:* Density Factor: 20
- *Magnitude Specifications:* leave it as default.

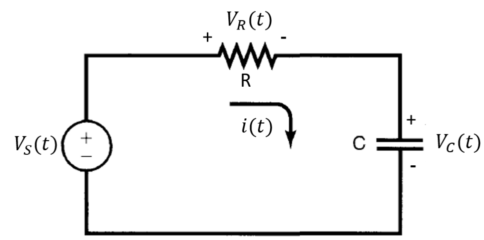
The part left for your design is then defining the *Frequency Specifications:* `Fs`, `Fpass1`, `Fstop1` `Fstop2`, and `Fpass2`. Please report the values you choose in the answer sheet. **Justify your answer.**

**10.** **(I)** (6pts): Now you can filter the signal using the filter coefficients obtained in Question **9** in `filterDesigner` you can export the coefficients of the filter which you just designed by clicking on `File → Export`. When you click the `Export` button, the coefficients will be stored in your workspace in the variable `Num`. Use *filter_data.m* to filter the signal and generate the time domain and frequency domain plots of the filtered signal. Paste the plot in your answer sheet.
Note: You don't need to submit code for this question.

# Part II: System Theory (30 points)

Before answering the following questions, you will need to go to the folder *part2*. Solve all implementation questions in *part2.m*.

**11. (Q)** (10pts)



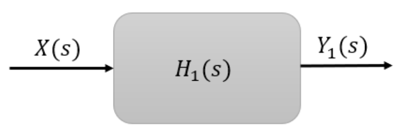

The RC circuit shown consists of an independent voltage source $V_S(t)$, a capacitor $C$ and a resistor $R$ whose voltage is shown by $V_R(t)$ (all in SI units).

a. Write the governing differential equation of the system by selecting $V_S(t)$ as input, and $V_R(t)$ as output.
(*Hint: The differential equation should consist of only the input, output, and parameters. You should use Kirchhoff's voltage rule to write the equation and proceed with taking the derivative of both sides of the equation*)
*(Hint: You can easily write math expressions with Insert → Object → Formula…)*

b. By taking the Laplace transform of both sides of the differential equation (analytically) you have found, calculate the transfer function $H(s) = \frac{Y(s)}{X(s)}$ , where $X(s)$ and $Y(s)$ are the Laplace transforms of the input and output, respectively.

c. By taking $R = 0.1$ Ohm and $C = 0.2$ F, compute the time-constant $\Gamma$ of the system. **Justify your answer.**
*(Hint: Symbols can be created with % in the Formula, e.g. %GAMMA gives Γ)*

**12. (I)** (9pts)



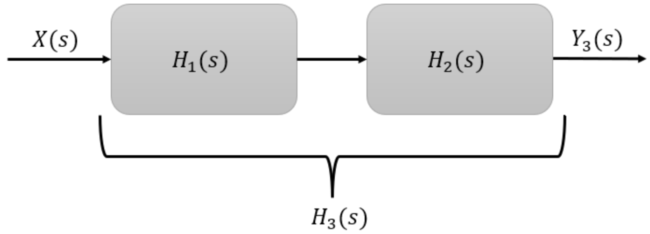

Given the continuous-time representation of the system with the transfer function $H_1(s) = \frac{2}{s+2}$ ,

a. Calculate the step-response $y_{u,1}(t)$ of the system by using Matlab's symbolic toolbox. Implement the code and report the formula of $y_{u,1}(t)$ in the answer sheet.

b. Plot the step response $y_{u,1}(t)$. What is the time-constant of the system? Find and indicate it on the plot by adding data points. Insert the plot in the answer sheet (with data points indicating time-constant). **Justify your answer.**

c. What is the magnitude of the response when the time goes to infinity? **Justify your answer.**

**13. (I)** (5pts)




Assume that the output of the first system in Question **12** is connected to a second system whose transfer function is $H_2(s) = 0.5/(s + 0.5)$, as shown in the figure above. Now, we would like to find the overall transfer function $H_3\,(s) = \frac{Y_3(s)}{X(s)}$.

a. Find $H_3(s)$ analytically, represent it with `tf()` function in Matlab and plot the step response $y_{u,3}(t)$ by using the `step()` function. Implement the code and insert both $H_3(s)$ and plot in the answer sheet.
b. What is the time constant of the system? Indicate it on the plot by adding data points. Report the plot in the answer sheet (with data points indicating time constant). Is the overall system $H_3\,(s)$ is faster than $H_1(s)$ ? **Justify your answer.**

**14. (Q)** (6pts) If you would like to represent the continuous system in Question **11** in a digital computer and find the digital transfer function $H(z)$, what are the steps to apply? Explain them with three sentences.
(*Hint: start with the differential equation you have obtained and suggest a solution based on the relevant lab material you have been familiarizing with*)

# Part III: Arduino (30 points)

Navigate to *part3*, where you will find all files required for this part. Solve all Matlab questions in *part3.m* and all other coding questions directly in *lvt_arduino.ino*.

**15.** **(Q)** (4pts) Have a look at the *lvt_arduino.ino* file. What are the main tasks this code does?

**16.** **(I)** (6pts) As we saw in Lab 6, the interrupt triggers for every state change of the UP button. Modify the ISR(PCINT0_vect) function to only trigger when pressing but not when releasing the button.

**17.** **(Q)** (4pts) Other than for reacting to button presses, interrupts can also be used to read sensor values. Explain an alternative to interrupts for reading sensor values and give one advantage and disadvantage.

Now open the file *acc_data.csv*. It contains data gathered from the accelerometer of a static Arduino, placed on a vibrating smartphone.

**18.** **(Q)** (2pts) What is the sampling frequency used here?

**19.** **(Q)** (2pts) Which column corresponds to the vertical axis? Why?

**20.** **(I)** (4pts) Use Matlab to plot the 3 signals on the same figure and insert the resulting figure (with axis labels!) in the answer sheet.
*(Hint:* `hold on` *allows you to place multiple plots on the same figure)*

**21.** **(I)** (8pts) Use a discrete Fourier transform to obtain the main vibration frequencies of the phone, other than the DC frequency. **Report the frequencies found in the answer sheet.**
*(Hint: use the norm of all three axis (centered at 0 respectively) to obtain a clearer signal prior to using Fourier; use Matlab* `fft` *function)*