Homework 1

This homework requires the following equipment:

- Matlab with
  - Control systems toolbox
  - Signal processing toolbox
- DISAL Arduino kit with
  - Arduino IDE

Information

In the following, text you will find several exercises and questions.

- The notation **S** means that the question can be solved using only additional simulation.
- The notation **Q** means that the question can be answered theoretically, without any simulation.
- The notation **I** means that the problem has to be solved by implementing a piece of code and performing a simulation.
- The notation **B** means that the question is optional and should be answered if you have enough time at your disposal.

Outline

This homework includes questions about signal processing topics and embedded systems that have been covered in the course up to date. In Part 1, you are asked a question about discrete convolution of two signals. In Part 2, you are given the acceleration data collected by an Arduino board and you will analyze the signal by Fourier transform, sample it without aliasing, filter out the noise and reconstruct the signal. In Part 3, you will analyze a thermal system, write its governing differential equations, find its impulse and step responses. You will also learn to find the response to an arbitrary input. Finally, in Part 4, you will design your own earthquake monitoring system using the DISAL Arduino kit you got familiar with in the Lab 5 and 6.

Collaboration policy

This homework aims also to foster the collaboration capabilities of students to work as a team. Teams have been defined by taking into account student preferences and eventually approved centrally by the Head TA of the course. Therefore, by default, the submitted deliverables will be considered a product of the team and students belonging to the same team will receive the same grade.

While intra-team collaboration is promoted, **any inter-team collaboration is not allowed** and, in case of evidence of copies between teams, there will be severe penalties for all parties involved.

In case of issues within a group in terms of contribution fairness or unenrollment of a group member from the course, please contact the Head TA as soon as possible in order to find a solution (e.g., splitting the team or grading separately the individual contributions).

In order to cope with odd student numbers and the potential unpredictable collaboration issues mentioned above, we are forced to consider team sizes different from two while maintaining the same assignment for all teams. In case of teams consisting of three students, we will apply a malus on the graded team performance, while for single students we will apply a bonus on the graded team performance. For these exceptional cases, the bonus and malus values will be communicated directly by the Head TA in timely fashion.

Getting Started

The questions in this homework can be solved more easily if you review the corresponding labs. Before starting please take a glance at the assignments, solutions and provided code for those labs.

To start with this homework, you will need to download the material available on Moodle. Download material_hw1.tar.gz or material_hw1.zip and extract it in your home directory (you can type tar xvfz material_hw1.tar.gz). Now start Matlab and change your "Current directory" to be material_hw1/part01/. For each part, there will be templates for you to write your code.

Please install Control System Toolbox and Signal Processing Toolbox if you have not installed it already. You can do so by pressing Add-Ons button in Matlab main window.

Submitting your answers

You need to submit a report and all your implemented code for this homework via Moodle. The report including results, explanations and plots should be in .pdf format and named as *SIS_21-22_Homework-1_ Group-GroupNo.pdf*. The code submission should include the folders for part01, part02, part03 and part04 and they should be zipped into the file named *SIS_21-22_Homework-1_Group-GroupNo.zip*. Please use the templates scripts and functions given inside material_hw1.zip to implement your code, do not use any separate .m file or function. Include all codes you have written (i.e. plotting, configuring etc.). Due date for the homework will be announced via Moodle.

Part I: Convolution (10 points)

1. **(I, 10pts)** Consider the following two signals:

$$s[n] = \begin{cases} \cos(0.5 * n), & -5 \le n \le 5 \\ 0, & otherwise \end{cases} \qquad h[n] = \begin{cases} 1, & n = -25,0,25 \\ 0, & otherwise \end{cases}$$

   a. Plot both signals in Matlab between $n = [-50:50]$, <u>without interpolation</u>.
   b. Complete the function take_conv.m provided to you in the folder part01 to compute the discrete convolution of $h[n]$ and $s[n]$, i.e., $y[n] = (h * s)[n]$. Find $y[n]$ and plot the result for $n = [-50; 50]$.

Part II: Fourier Transform, Signal Reconstruction and Filtering (40 points)

Before answering the following questions, you will need to go to the folder part02. Suppose that we want to monitor the working condition of an automatic waste classification system.

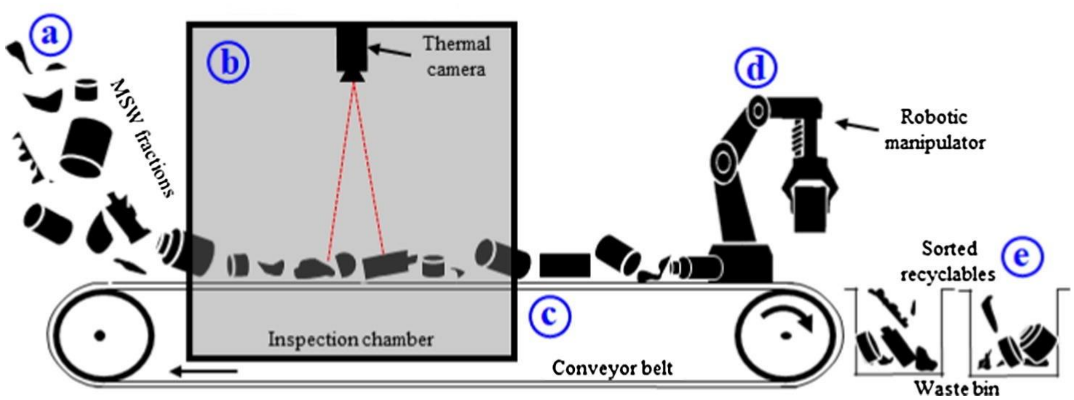


Figure 1. The illustration of an automatic waste classification system (picture from: https://www.sciencedirect.com/science/article/pii/S0956053X17306876)

The conveyer belt needs to move and stop at a fixed frequency to allow the manipulator to catch each waste. In order to study the movement of the conveyer belt, an accelerometer embedded in an Arduino board will be used. The acceleration data of the belt was sampled at 5 kHz along with timestamps (variable t) and sent back to the host machine through Wi-Fi. These data were recorded in the file ArduinoData.mat. Load the data into Matlab by using the command load('ArduinoData.mat') and observe the signal.

2. **(I, 3pts)** Plot the signal both in time domain and frequency domain (by using the Matlab's fft()), and insert it on the report. Take units of frequency as Hz and put the zero-frequency component in the center. Describe the signal by analyzing its frequency components.

3. **(Q, 8pts)** According to the instruction of the machine, the acceleration of the conveyer belt is a sine function with the period around 0.5s, 1s and 1.5s depending on different operating levels (fast, medium and slow).

   a. What is the highest frequency of the signal you obtained in the previous question?
   b. Could you specify which level was set when the data was recorded? **Justify your answer**
   c. What are the amplitudes of the original signal?
   Hint: List here the amplitudes of the first three main components.
   d. Assuming that all signal components beyond 50 Hz are actually the measurement noise, what type of filter would you implement (low pass, high pass, band pass or band stop) to remove the noisy signals? **Justify your answer.**

4. **(I, 4pts)** Recall that the *FIR* filter form is defined as:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

   Where we get our actual filtered value y[n] by summing up M previous unfiltered values x[n-k] multiplied with the corresponding coefficient b_k. When we design a filter, the final result is this set of coefficients b_k. The number of coefficients M is corresponding to the filter order. In the FilterDesigner tool you can export the coefficients of the filter which you just designed by clicking on File → Export. When you then click the Export button, the coefficients will be stored in your workspace in the variable Num.
   Answer the following questions for the filter you will design.

   a. Design a finite impulse response (*FIR*) filter to eliminate the noise on the downsampled signal by leveraging Matlab's FilterDesigner introduced in Lab 4 with reasonable phase delay (with filter order of 150). Report all filter parameters (Fpass, Fstop or Fpass1, Fstop1 Fstop2, and Fpass2 etc.) **Justify your answer.**
   b. Plot the magnitude and phase response of this filter.
   Hint: You can switch the plot to show magnitude and/or phase response with the buttons in the toolbar above of the FilterDesigner.

5. **(I, 4pts)** Complete the filter_data() function to filter the noisy signal by using a FIR filter with the coefficients obtained in Question 4. Plot the filtered signal in both time and frequency domain. Insert the plots in your report.
6. **(I, 6pts)** Answer the following questions for the signal:

   a. What would be the minimum sampling frequency? **Justify your answer**
   b. What happens if the sampling frequency is lower than this minimum and why? **Justify your answer.**
   c. In practice, we usually use a sampling frequency Fs as 10 times greater than the maximal frequency. Complete the function sample_data() to downsample the signal at Fs.

7. **(I, 15pts)** Reconstruct the noise-free signal by using zero-order-hold interpolation, first-order-hold interpolation and Whittaker-Shannon interpolation at 10 kHz.

a. Could you explain the differences between these three interpolation methods?

b. Complete the reconstruct_signal()function to reconstruct the signal with different interpolation methods and plot the reconstructed signals in both time and frequency domain. Insert the plots to the report.

c. Are there any differences between these plots? **Justify your answer.**

Part III: System Theory (40 points)

Now, set part03 as your "Current Directory". Consider the insulated thermal chamber with the thermal capacitance of *C* and the temperature of *T(t)* given in Fig. 2. The chamber contains a heater where it radiates heat with the rate of *q(t)*. The room has a thin wall on its one edge whose thermal resistance is *R* and the ambient temperature on the outer wall is $T_o(t) \ll T(t)$. All quantities are in SI units.

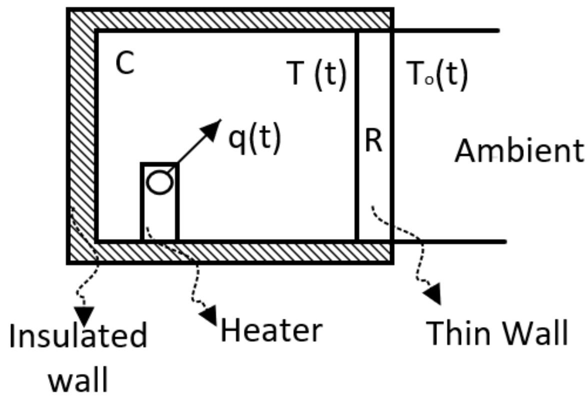
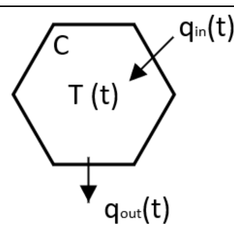
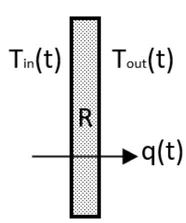


Figure 2. Insulated thermal chamber

Two separate information about the thermal dynamics of the system are given as follows:

Table 1. Thermal system components

Explanation	*Representation*	*Differential Equation*
A room with the temperature *T(t)* and the thermal capacitance *C*. The total heat rates entering into the room and exiting from the room are $q_{in}(t)$ and $q_{out}(t)$, respectively.		$C\frac{dT(t)}{dt} = \sum q(t) = \sum q_{in}(t) - \sum q_{out}(t)$
A thin wall with the thermal resistance *R*. The heat rate crossing the wall is *q(t)*. Inside and outside temperatures are $T_{in}(t)$ and $T_{out}(t)$, respectively.		$T_{in}(t) - T_{out}(t) = Rq(t)$

8. **(Q, 4pts)** By using the thermal capacitance and thermal resistance differential equations, write the differential equation governing the evolution of the temperature *T(t)* in the insulated chamber given in Fig. 1. Define the heater's heat generation rate *q(t)* as the first input, ambient temperature $T_o(t)$ as the second input and the chamber temperature *T(t)* as the output.

9. **(Q, 4pts)** By taking the Laplace transform of both sides of the differential equation you found in Question 8, find the transfer functions $H_1(s) = \frac{X(s)}{U_1(s)}$ and $H_2(s) = \frac{X(s)}{U_2(s)}$ where $X(s)$ is the Laplace transform of the output, $U_1(s)$ is the Laplace transform of first input and $U_2(s)$ is the Laplace transform of the second input you defined. Explicitly write each step.

Hint 1: Use Laplace tables given in Lab 3 to find the Laplace transform of the derivative of functions.

Hint 2: For multiple-input single-output systems, to find transfer function that maps specific input to the output, you can set the other input to zero.

10. **(Q, 4pts)** Consider that $H_1(s) = \frac{R}{CRs+1}$. Find the impulse response $h_1(t)$ by taking the inverse Laplace transform of $H_1(s)$ analytically.
    Hint: Use Laplace transform tables given in Lab 3.

11. **(I, 4pts)** Consider that $H_2(s) = \frac{1}{CRs+1}$. Find the impulse response $h_2(t)$ by taking inverse Laplace transform of $H_2(s)$ on Matlab.
    Hint: Define the transfer function symbolically and use ilaplace().

12. **(I, 4pts)** Take *R* = 1 and *C* = 1. Plot the step response and find the time constant of system $H_1(s)$ by

    a. By defining the transfer function with tf()and using this transfer function with step().
    b. By taking the integral of impulse response you found in Question 10 on Matlab using int(). Comment on the comparison of the results you found in part a and b of this question. *Note: Take the integration constant as zero.*

13. **(S, 4pts)** Plot the step response of $H_1(s)$ for *R*=1, *C*=5 and *R*=5, *C*=1. How does the increasing thermal capacitance of the chamber and the thermal resistance of the wall (separately) affect the temperature evolution inside the room? How do you explain this physically? Compare the time constants of both systems.

14. **(I, 4pts)** We would like to represent the system $H_1(s)$ in discrete domain, i.e $H_1(z)$. Use c2d()function on Matlab to discretize the system $H_1(s)$ with the zero-order-hold method and with the sampling time of 0.1 seconds. Take R=1 and C=1. Plot the step response and compare it with the continuous counterpart.

15. **(I, 4pts)** Consider that $H_1(z) = \frac{0.09516}{z-0.9048}$. Find and plot the step response of the discrete system by using $H_1(z)$ and the *z*-domain representation of step function in Matlab.
    Hint: Use iztrans() to find the step response, i.e. $x_s[n] = Z^{-1}(S(z) \cdot H_1(z))$, *where S(z) is the z-transform of step function.*

16. **(I, 4pts)** Define the time series for *n* using the code below:

    ```
    n = 0:1:100;
    ```

    Represent the step response of the discrete system by using this time series n, i.e. $x_s[n]$. Plot the Fast Fourier Transform (FFT) of the step response by using the function fft()in Matlab and observe and report the frequency components. While calculating the FFT, take the units of frequency as Hz and put the zero-frequency component in the center.

17. **(I, 4pts)** Now we would like to see the response of the discrete system $H_1(z)$ to an arbitrary input. Load the HeaterData.mat. Plot these data with the accompanying time-stamps and observe the time evolution of the heater's input. Now, apply this input to the discrete system and plot the response (the temperature of the chamber).
    Hint: Use lsim() function of Matlab

Part IV: Embedded Programming (30 points)

In this part of the homework, you will design your own earthquake monitoring system using the DISAL Arduino kit. You will write a program that detects a small earthquake using the accelerometer on the DISAL Shield and you will simulate the earthquake using your phone's vibration feature. Go to the folder part04.

18. **(Q, 4pts)** Design a way to use the accelerometer to detect the vibration of your phone. More specifically, combine the acceleration data on X, Y and Z into one value (i.e. *earthquake detection value*) to find the total acceleration of the system. This value should be low when the phone is not vibrating and high when the phone is vibrating. Keep in mind that, since the accelerometer is a very noisy sensor, an initial calibration is necessary to obtain the baseline value of the sensors when there is no vibration. The calibration values should be subtracted to the accelerometer values during your experiments to eliminate the sensor offset. **Describe your solution** (use the help of formulas, diagrams, pseudocode where appropriate).

19. **(I, 4pts)** From the Arduino IDE, open the file earthquake_detector.ino inside the folder earthquacke_detector. **Implement an initial calibration** of the accelerometer sensor in the function *calibrate()* by computing the average of 100 values for each axis of the accelerometer and save them in the variables xavg, yavg and zavg. These values will be used as baseline during earthquake detection. Do not forget to initialize all the components that you need in the setup() function.

20. **(I, 8pts)** In the earthquake_detector.ino file, implement the solution you designed in Question 18. Test your solution by printing the earthquake detection value on the DISAL Shield's screen and check that this value increases and decreases according to the pulse of your phone vibration. The frequency is already set to 10 Hz in the *loop()* function.

21. **(I, 3pts)** In the earthquake_detector.ino file, add code to send your earthquake detection value to a base station. In the receiver.ino file, write code that gets the earthquake detection value and send it to the computer. Test your solution.

22. **(I, 2pts)** Complete the script python/saveSerialData.py to save the earthquake detection data on your computer.

23. **(Q, 2pts)** Setup your experiment. You should have an earthquake detector Arduino capable of detecting vibrations using its accelerometer. Place it on top of your phone. A receiver Arduino should be connected to your computer and should receive the earthquake detection data with a frequency of 10 Hz. Make your phone vibrate and record the data on your computer to the file *earthquake_data.txt* using the saveSerialData.py script. *Hint: make sure to reset the earthquake detector Arduino after placing it on your phone to get the baseline calibration data for the accelerometer.*

24. **(I, 3pts)** Write a Python script to clean your dataset in python/cleanData.py. Save your data to a file called earthquake_data_clean.txt.

25. **(I, 4pts)** Now you have collected the data of the earthquake using the Arduino board, you cleaned it and saved it into the file earthquake_data_clean.txt. Open the folder matlab. Complete the code of plotter.m and plot the signal both in time domain and frequency domain (by using the Matlab's fft()). Insert the plots into the report. Describe the signal by analyzing its frequency components.