Signals & Systems Lab Project.

General rules:

Submission deadline:

All MATLAB assignments should be submitted due to **Tuesday 20/12/2016 at 23:55**, any email after this date will be directly excluded without exceptions.

There is two bonus sources (added to total lab marks including lab experiments)

a. Early submission

- Early submissions are due **Sunday 18/12/2016 at 23:59**

b. Bonus section of each experiment

Submission mail:

signal.lab.2016@gmail.com

General rules:

(1) Students are to be divided into groups of 5-7 students each, with one team leader. The leader has the task of filling this form with the names and IDs of the team members.

This should be done no later than 07/12/2016

Form:

https://docs.google.com/forms/d/1xrWaHyxkM0uZE7EOr0JVARSed5Qhil4cIPBNAuVe0Uo/viewform?edit_requested=true

(2) Any copied codes will be awarded **"zero"** without notification.

(3) Projects sent by mail should have a subject of this format: team_number. A Team number will be assigned to each team

(4) The mail attachment should include:

1) One PDF report with all the figures, comments and other requirements arranged in the same order described in this file.
2) Separate M-files for each experiment. M-files should be well commented, indented, and properly organized. Variable naming should be self-explanatory and clearly indicating which variable/physical quantity it represents in order to ease the tracing of the code.

(5) The report must be in PDF format, any other format will not be graded, and the label of the report must be the same as the subject of the email.

(6) Overall organization of the report/M-files with logical development as well as following the rules described herein is highly appreciated.

Project aim

(1) To implement a general signal generator using MATLAB.
(2) Implementation of LTI systems using Impulse response and difference equations.
(3) Investigation of a system filtering (convolution) in time and frequency domain.
(4) Investigation of the effect of noise (random signal) on the transmitted signals
(5) Dealing with sound signals (.wav) files.

<u>Block diagram of the project</u>

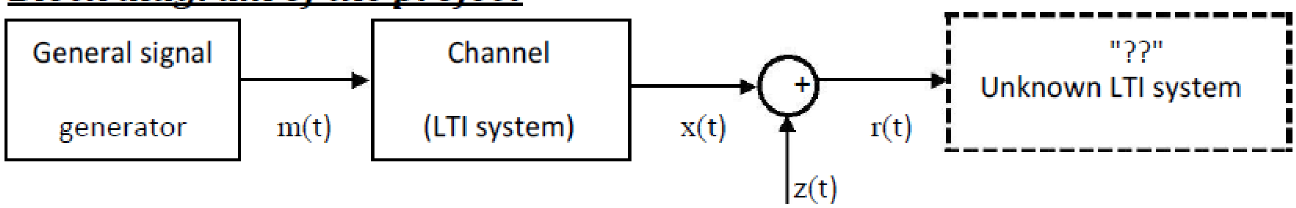

Figure 1

Experiment 1: it will be about the general signal generator

Experiment 2: it will be about modeling the LTI channel as an impulse response

Experiment 3: it will be about modeling the LTI channel as a difference equation

Experiment 4: it will be about sound processing

Experiment 1: General signal generator

It is required to implement a general signal generator that has the following specifications:

(1) When the program starts the program asks the user for the following parameters:

a. Sampling frequency of signal.
b. Start and end of time scale
c. Number of the break points and their positions (i.e. the points that the signal definition rule changes).

Example: The signal is defined from -2→0 as a DC signal and from 0→2 as ramp the user will enter that the number of break points =1 and the position at t=0.

(2) According to the number of break points the program asks the user at each region to enter the specifications of the signal at this region which are:

a. DC signal: Amplitude.
b. Ramp signal: slope – intercept.
c. Exponential signal: Amplitude – exponent.
d. Sinusoidal signal: Amplitude – frequency – phase.

(3) Then the program will form the previous signal using the indirect method (review section 2 for the indirect method)

Hint

To implement the different selection you may make use of the command "menu".

Requirements

1. Plot the output of the signal generator in both time and frequency domains.
2. Test your code by the generating the following message

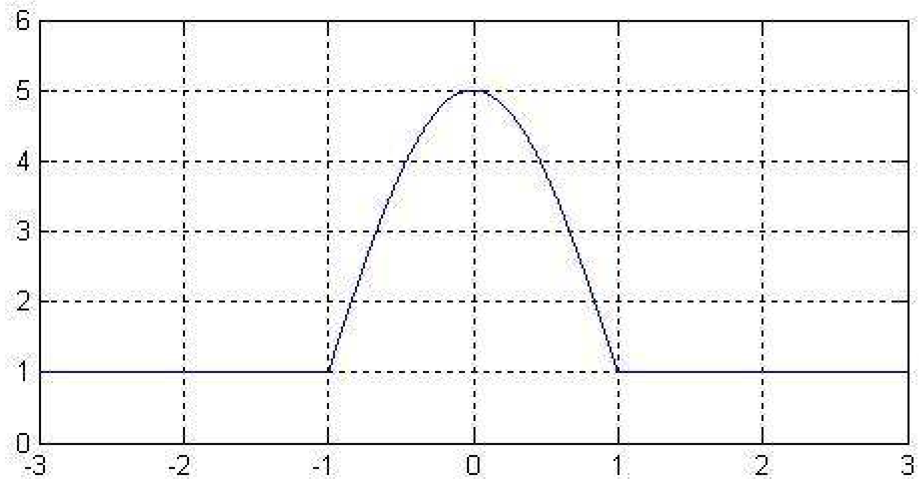

Figure 2

Experiment 2: LTI channel "impulse response"

After generating the message from the signal generator; we will pass it through LTI channel. We will model the channel using impulse response

Steps

1. You will use the message m(t) generated from experiment one
2. The program should again form the impulse response by the same way of generation of signal.
3. The program then calculates the output of the system **x(t)=y(t)⊛m(t) (using the command "conv")**
4. The program should have the ability to add noise z(t)(simply random signal) to the output signal **x(t)**
   The random signal generation is done as following

   Z(t) =sigma*randn(1,length(x))

   The output will be a Gaussian distributed noise with zero mean and standard deviation of sigma.
5. Your code should be able to return **m(t)** from **x(t) (using the command "deconv")**

Requirements

a. Plot the frequency response of the channel in both time and frequency domains.
b. Plot the signal output signal in both time and frequency domains.
c. You can test your code using the following channel impulse response

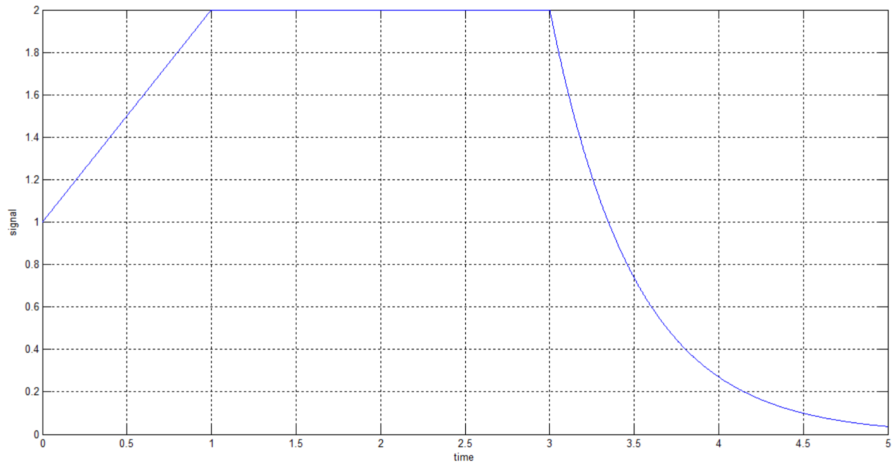

Figure 3

Experiment 3: LTI channel "Difference Equation"

After generating the message from the signal generator; we will pass it through LTI channel. We will model the channel using difference equation

Steps

1. You will use the message m(t) generated from experiment one
2. To generate the difference equation :
   - The user should enter the numerator and denominator of the transfer function of the system in figure 4
   - b. Then the program calculates the output of the system (using the command "filter")
3. Suggest a LTI system that will **reverse** the effect of the channel??

Requirements

a. Plot the frequency response of the channel in both time and frequency domains.

b. Plot the signal output signal in both time and frequency domains.

c. Plot the frequency response of the unknown system that will recover the transmitted signal m(t).

d. **Bonus**: can you check the stability of the original system & the recovery system using Matlab ??

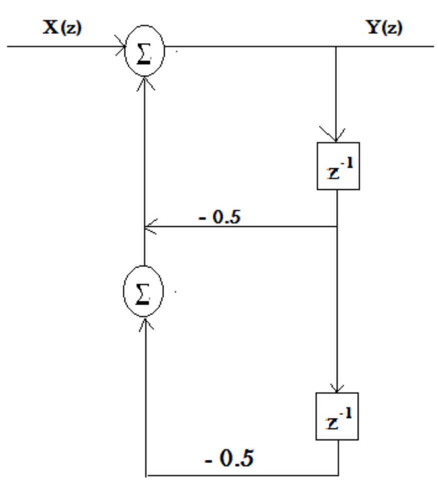

Figure 4

Experiment 4: sound processing

Steps

1. Input a sound file to Matlab using sampling frequency = 44100Hz
2. Plot the sound in time domain & frequency domain
3. Input the sound file to the system as shown in figure 5
4. Save the output sound from the system & play it
5. Comment on the output sound
6. **Bonus**: can you filter the input sound at 4 KHz??

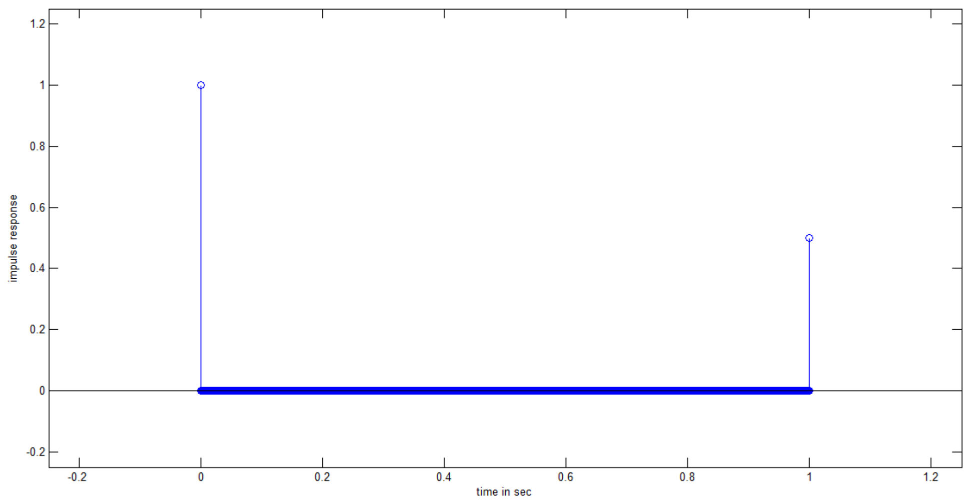

Figure 5