LAB MANUAL

Local Computer Networks Module

Signal Processing Laboratories

Lab 1: Generation and representation of signals

Objectives of the lab

- Generate various deterministic or random signals.
- Visualize the signals on graphs using Matlab.
- Generate and visualize different signals using Simulink blocks.

1. Generation and representation of signals in Matlab

1.1 Signal sinusoidal

- Ecrire un code matlab qui génère une sinusoïde de fréquence f=320 Hz pour une durée de 20 ms. La fréquence d'échantillonnage est de 10 kHz. Représenter le signal généré.

- Répéter l'opération pour f=600, 1000, 4400, 5000, 5600 Hz. Que remarquez vous ?

1.2 Uniform Random Signal - Gaussian Random Signal

- Generate 1000 points of a random signal with a uniform distribution (Using the command rand), and represent the created signal.

- Repeat the same operation for a random signal with a Gaussian distribution (normal distribution). (Using the command randn).

1.3 Compound Signal

- Generate and represent a signal y that contains 2 sine waves, one of frequency 60 Hz and amplitude 1, and the other of frequency 120 Hz and amplitude 2. Take t=0:0.001:0.5;

- Generate and represent Gaussian white noise of the same size as the t vector using the command randn.

- Now, considering the signal y disturbed by Gaussian white noise multiplied by 0.5, visualize the new signal called yn.

- Represent the three signals y, the noise, and yn in three sections of the same figure. What do you notice?

1.4 Sinc signal (cardinal sine)

$x(t)=\sin(2\pi ft)/ 2\pi ft$

Plot the signal x(t) for f=20 Hz and t belonging to [-0.1, 0.1], step=0.001. Repeat the operation for f=40 and 60 Hz. Comment on the results.

2. Visualization of signals using Simulink

In the example below, there are blocks representing individual functions. For instance, the 'sine wave' block is a sinusoidal generator that can be visualized using the 'scope' block, simulating an oscilloscope.

a. Using Simulink blocks, visualize the following analog signals:

1. Sinusoidal signal with an amplitude of 3 and a period of T=15s and then 5s.

2. Cosinusoidal signal with an amplitude of 3 and a period of T=10% and then 5s.

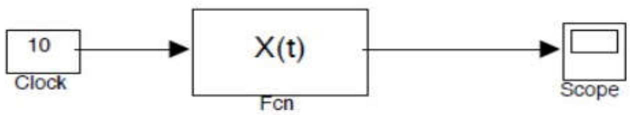
3. Gaussian white noise with a power of 0.2 (Use the Band-Limited White Noise block).

4. Uniform noise with an amplitude of 3 (Use the Uniform Random Number block).

To create these signals, utilize the blocks found in Simulink/sources for the different signals and use the scope found in Simulink/sinks for signal visualization. Double-click on any block to modify its parameters.

b. Create, using Simulink, the signal $x(t) = 2\sin(2\pi f_1 t) + \sin(2\pi f_2 t)$, with f_1=0.1Hz and f_2=1Hz over the time interval t belonging to [0,20]. Visualize the generated signal.

To generate this signal, use the Fcn block located in Simulink/User-Defined Functions.



The clock block represents time, which is the input to the system. The two blocks, clock, and Fcn, together form the generator of the signal x(t).

3. Exercise (Representation of common signals)

Write a Matlab code that generates the following signals for a period $t \in [-10, 10]$ seconds with a step of 0.001;

- The unit impulse (Dirac impulse).
- The unit step of Heaviside.
- Rectangular signal (rectangular function).
- Sine and decreasing exponential: $\sin(0.35 \times t) * e^{(0.2 \times t)} \times u[t]$
  where u is the unit step.
- Ramp function.

Lab 2: Fourier Transform and Spectral Analysis

Objectives of the Lab:

- Calculate and represent the Fourier transform using Matlab.
- Analyze a vibratory signal through Fourier transform.

1. Temporal and Frequency Representation

Consider the function:
x(t)= t*exp(-a*t)*u(t), a>0 and u(t) : unit step function.

a. Plot the signal x(t) between -5 and 5 for a=2, with a sampling interval $\Delta t=0.01$ s (T_s).

b. Formally calculate the Fourier transform X(f) and plot it on another figure between -5 Hz and 5 Hz with a frequency step $\Delta f=0.01$ Hz (F_s).

c. Plot the magnitude and phase of the Fourier transform (using the abs and angle functions).

2. Calculation and representation of a Fourier transform using Matlab

a. To approximate the continuous Fourier transform of a signal x(t), represented with a sampling interval Δt, the command is used:

FT = fftshift(Ts * fft(x)) ;

- Plot the amplitude spectrum of the Fourier transform of x(t) between -5 Hz and 5 Hz. Compare with the result from part (1).

b. The inverse Fourier transform is obtained using the command: xt=abs(ifft(FT)/ Ts). Does one exactly retrieve the original signal?

3. Analysis of a Vibratory Signal. (Open a new Matlab file)

- Save the file "vibratory_signal.mat" in the Matlab directory. Load the file using the command: ***load('vibratory_signal.mat')***.

- Calculate the duration of the vibratory signal.

- Plot the signal as a function of time on one figure.

- Calculate its Fourier transform and plot it on another figure using ***fft*** and ***fftshift*** over 20480 frequency points.

- Identify the frequency of the defect and provide its frequency in Hertz.

Lab 3: Sampling

Objective:

- This lab introduces the fundamental concepts associated with transforming continuous-time signals into discrete-time signals through a procedure known as sampling.
- Study of the sampling effect

Consider a sinusoidal signal defined by:

$s(t) = \sin(2\pi f_0 t)$, where $f_0 = 1\text{Hz}$.

1. Choose to sample this signal at the frequency fs = 4 Hz.

Verify that this sampling frequency is correct from a theoretical point of view (justifying the response). Write a program to generate and display several periods of the signal s(t).

2. Display its spectrum using the fft function (the FFT algorithm, for Fast Fourier Transform, is a fast algorithm for computing the DFT, Discrete Fourier Transform), for example, as follows:

TFD = fft(s, N) / N; % DFT of the signal s, over N samples

Note: TFD is a variable; you can name it as you like.
Observe the effect of undersampling by varying the frequency of the signal at the following values:

f_0 = fs/10, f0 = fs/4, f0 = fs/2, f0 = fs*3/4, f0 = fs*10/9

Interpret the results by reasoning about the spectrum.

3. Exercise: Representation of common digital signals

We will explore the functions that generate common digital signals using the ***stem*** function.

Write Matlab code that generates the following discrete signals for a period of [-10, 10] seconds with an equal step of 1:

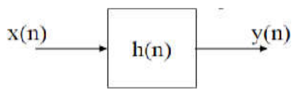
- Unit impulse (Dirac impulse).
- Unit step of Heaviside.
- Rectangular signal (rectangular pulse).

Lab 4: Filtering

Objective: Study of an Impulse Response and Discovery of Sptool.

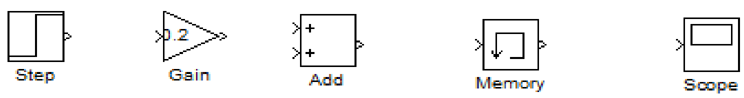
1. Simulink

h(n) is the impulse response of a digital system. The recursive equation for the output is given by:



$$y(n) = 0.1x(n) + 1.3y(n-1) - 0.4y(n-2).$$

a) Implement this filter in Simulink and analyze the step response. (X is a unit step).
The blocks to be used:



b) Write a program in MatLab that implements this system (using the signal x(n) as a unit step defined over the interval [1,20]). Verify the obtained values by comparing them with those from Simulink.

c) Calculate the FFT of the filter.

2. Discovery of Sptool Manipulations:

- In the Matlab command window, type sptool to launch Sptool.
- From the Sptool dialog box, you can import and/or export data to/from files or the Matlab command window.
- With Sptool, you can interactively create digital filters, visualize their different responses, and view signals and spectra. We will base this on the synthesis of a low-pass FIR filter with an attenuated band that has a sampling frequency of 48 kHz and the following characteristics:
  - ✓ Passband frequency (Fp): 9600 Hz.
  - ✓ Cutoff frequency (Fs): 12000 Hz.

Getting Started with SPTOOL

1. Filter Synthesis

- In the SPTool window, click on 'New Design.' A Filter Designer window will open.
- Enter the filter type as Butterworth, then the template (Uncheck the minimum Order box) (Fs, Rs, SamplingFrequency), and click on 'Design Filter.'
- Then click on 'File / Close.' Your synthesized filter appears with the name filt1; give it a name using 'Edit / Name.'
- Create in the same way two Chebyshev filters (with the same characteristics).

2. Visualization of Filter Characteristics

- In the SPTool window, choose a filter, then click on 'View.' A Filter Viewer window opens. Interpret the results (impulse response, step response, frequency response, poles/zeros, etc.).
- In the Filter Viewer window, select the responses you want to observe (zeros/poles, magnitude, phase, group delay, impulse, step response...) and the scales to use. Zoom in, explore.
- Switch back to the SPTool window and select the three created filters (hold down the ctrl key).
- Compare the three filters.

3. Visualization of Signals

- In the Matlab command window: Create a vector t from 0 to 4π with 1024 points. Create a vector :

  y = 2sin(2πf1t) + sin(2πf2*t) with f1 = 1 Hz and f2 = 10 Hz.

- Load the file 'essai.wav' located on the desktop using the wavread function. (CAUTION: do not enter any more instructions in the Command Window afterward).
- In the SPTool window, import these two signals with 'File / Import.' ('ans' corresponds to the last response given by Matlab, therefore to the 'son.wav' signal in vector form, and 'y' to the created signal).
- Choose one or more, click on 'View,' and explore the new Signal Browser window.