

# **SIGNALS AND SYSTEMS LABORATORY 1:**

## Introduction to Matlab

### **INTRODUCTION**

In this lab, you will learn the basic and intermediate functions of MATLAB and apply them to the theory of Convolution and Fourier Transforms. MATLAB is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages. Our objectives in this lab are to

#### 1. Understand MATLAB

We will be using the tutorial provided by Mathworks. The following website will provide the tutorial material. [http://www.mathworks.com/academia/student\\_center/tutorials](http://www.mathworks.com/academia/student_center/tutorials)

- a. Creating Variable in Matlab
- b. Performing Calculation
- c. Visualizing Data
- d. Creating Scripts with Matlab

#### 2. Gain Experience with MATLAB tools

- a. Convolution
  - i. Write a convolution program
  - ii. Use CONV
- b. Low-pass filter
  - i. Voice signal
  - ii. ECG signal

## SIGNALS AND SYSTEMS LABORATORY 1:

Introduction to Matlab  
Gain Experience with MATLAB tools

**Assignments will be collected in Lab #2. Work in group is allowed.**

### **Experiment #1:**      **Linear Convolution**

Given a pair of sequences, use discrete convolution to find the response to the input  $x[n]$  of the linear time-invariant system with impulse response  $h[n]$ .

$x[n]$ = square wave with amplitude 1

$y[n]$ = triangle wave with amplitude 2

- i. Use PLOT command to plot the input and impulse response
- ii. Compute the convolution by hand, use MATLAB to plot the results
- iii. Write a MATLAB function to compute the convolution of the two finite-length sequences and plot the results.  
For length  $N$  input vector  $x$ , the DFT is a length  $N$  vector  $X$ , with elements

$$X(k) = \sum_{n=1}^N x(n) \cdot \exp(-j \cdot 2 \cdot \pi \cdot (k-1) \cdot (n-1) / N), \quad 1 \leq k \leq N.$$

- iv. Use CONV command to verify the results from b).

### **Experiment #2:**      **Low-pass Filters**

Part 1:

In this section of the tutorial, you will create a signal with added high frequency noise. Later in the tutorial, you will use a low-pass filter to eliminate high frequency noise.

1. Type  
    load mtlb
2. You can hear a voice say "MATLAB." This is the signal to which you will add noise  
    soundsc(mtlb,Fs)
3. Create a noise signal

- ```
noise = cos(2*pi*3*Fs/8*(0:length(mtlb)-1)/Fs);
(You can hear the noise signal by typing soundsc(noise,Fs))
(You can also use random function to introduce noise)
```
4. Add the noise to the original signal
 

```
u = mtlb + noise;
```
  5. Scale the signal with noise
 

```
u = u/max(abs(u));
```

 (You scale the signal to try to avoid overflows later on. You can hear the scaled signal with noise by typing soundsc(u,Fs))
  6. Display the frequency spectrum using FFT (experiment 2)
  7. View the scaled signal with noise
 

```
specgram(u,256,Fs);colorbar
```

 (In the spectrogram, you can see the noise signal as a horizontal line at about 2800 Hz, which is equal to  $3*Fs/8$ )
  8. Use low-pass filters to eliminate high frequency
 

```
b = ones(1,10)/10;      % 10 point averaging filter
fy = filtfilt(b,1,x);   % Noncausal filtering
fyy = filter(b,1,x);    % Normal filtering
```
  9. Use FFT to plot the power spectrum of the filter signal and compare it to both, the original and the corrupted signals.

## Part 2: Real World Practical Application

In this section of the tutorial, you will be provided an ECG (Electrocardiogram) signal with noise. Your task is to create a matlab program to determine the frequency of the noise and eliminate the noise signal.