

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

DEDAN KIMATHI UNIVERSITY OF TECHNOLOGY

Digital Signal Processing Laboratory Manual



Ciira MAINA, PhD

May 2015

Contents

1 Introduction 2

1.1 Safety Information 2

2 System Setup 2

2.1 Software Installation 2

3 Laboratory 2: Speech in the time and frequency domain 3

3.1 Objectives 3

3.2 Apparatus 3

3.3 Background 3

3.3.1 The discrete Fourier transform 4

3.4 Procedure 5

3.4.1 Record Some Speech 5

3.4.2 Compute a spectrogram 5

A Laboratory Report Format 7

1 Introduction

This laboratory manual is designed to accompany the Digital Signal Processing (DSP) course at Dedan Kimathi University of Technology. It describes laboratory exercises aimed at integrating the Raspberry Pi into the curriculum. The development of these laboratory exercises was made possible by a generous grant from The Kenya Education Network (KENET).

The Raspberry Pi which retails at approximately \$30 was developed with the aim of improving computing education by providing access to an affordable computer with most of the capabilities of a modern computer. It is about the size of a credit card (see Figure 1) and as we will see in the labs described here, it can perform a number of important DSP tasks.



Figure 1: The Raspberry Pi. A Kenyan bank card is shown for comparison.

1.1 Safety Information

A number of safety precautions must be observed when handling and operating the Raspberry Pi. You will have received a copy before starting the labs.

2 System Setup

The computing environment used in this lab is the raspbian Debian Wheezy operating system. This operating system is quite similar to Ubuntu. The labs will be based on Octave, which is very similar to MATLAB, and will make use of other open source tools such as SoX, the Swiss Army knife of sound processing programs. The exercises have been tested on Octave version 3.6.2.

2.1 Software Installation

To run the labs, Octave and SoX must be installed.

- To install Octave, ensure you are connected to the internet and type `sudo apt-get install octave`
- We will also require the signals package which can be installed by typing `sudo apt-get install octave-signal`

- To install SoX type `sudo apt-get install sox`

These software has already been installed on the devices that will be issued to you. To verify that the pieces of software are installed you should type `octave --version` and `sox --version` on the command line.

3 Laboratory 2: Speech in the time and frequency domain

3.1 Objectives

The objectives of this laboratory are:

1. To introduce the speech processing in the time and frequency domains
2. To introduce the idea of spectrograms for speech analysis.

3.2 Equipment

For this lab you will need the following (If an item is not provided the student is expected to come with their own to the lab):

1. Raspberry Pi Model B+
2. USB Microphone
3. Mini USB cable (not provided)
4. Ear phones or speakers (not provided)

3.3 Background

Several applications require the analysis of frequency content of signals. In continuous time signals this is achieved using the Fourier series (for periodic signals) or the Fourier transform (for aperiodic signals). In real world applications continuous time signals are first discretized before they are analysed. In this case we use the Discrete Time Fourier Transform (DTFT) to analyze the signals. The DTFT is a sampled version of the frequency spectrum of the sampled signal and it is efficiently implemented using the Fast Fourier Transform (FFT).

To compute the frequency content of the speech signals we divide the signal into overlapping segments. We then compute the DTFT of the segments and obtain a sampled version of the spectrum.¹ This produces a 2-dimesional image known as a *spectrogram*. Figure 3 shows a spectrogram obtained from a recording of the author saying ‘moja’. It is obtained from a 2s recording sampled at 16kHz. It appears as a heatmap with high values represented by deep red and low values deep blue. Intermediate values appear yellow. From this spectrogram we see that the word is spoken in the interval between 0.8s and 1.4s. If we extract the segment at 1s corresponding to 32ms (512 samples at 16kHz) and compute it’s DTFT we obtain the results shown in Figure 2. We see that most of the spectral content is below 1 kHz.

¹In general this is a complex function and we must plot both the magnitude and phase for a complete representation. In speech applications we often ignore phase and work with the magnitude spectrum.

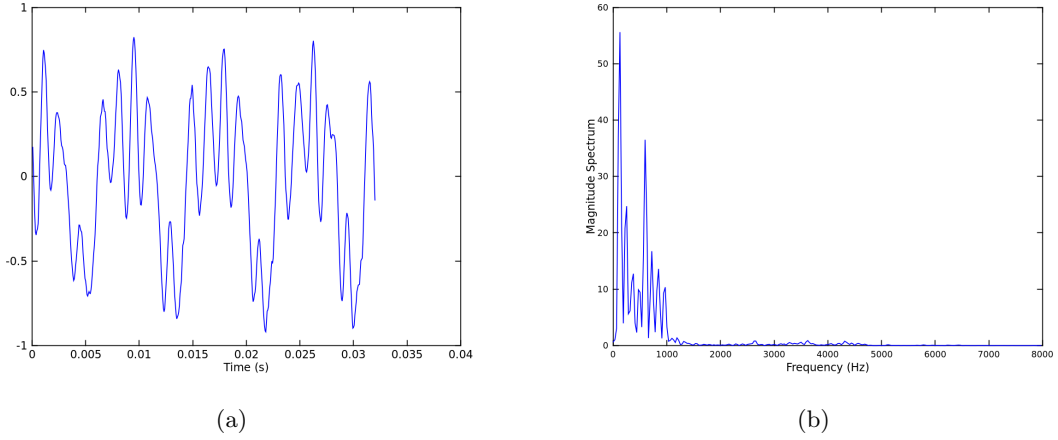


Figure 2: Speech waveform of a voiced speech segment (a) and the corresponding magnitude spectrum (b).

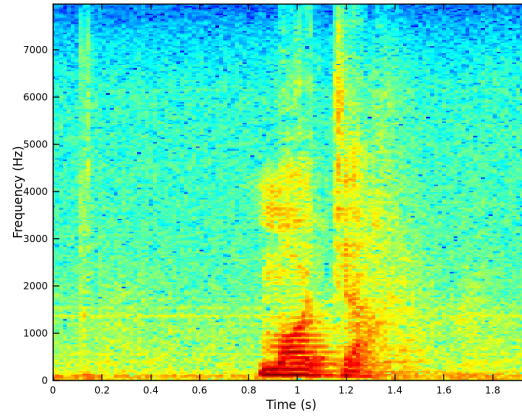


Figure 3: Spectrogram of the word ‘moja’.

3.3.1 The discrete Fourier transform

In engineering applications we are often interested in determining the frequency spectrum of a signal. We can approximate the spectrum of a continuous time signal $x(t)$ by using a transform called the Discrete Fourier transform (DFT). This DFT consists of samples of the spectrum. First we sample $x(t)$ to form the discrete time signal $x[n]$ of length N samples. We compute the DFT $X[k]$ of $x[n]$ by implementing the DFT equation

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn} \quad k = 0, 1, \dots, \frac{N}{2}$$

The DFT can be used to approximate the spectrum of $x(t)$. The DFT is a sampled version of the spectrum with the k th sample $X[k]$ corresponding to the frequency $f_k = \frac{k}{NT_s} Hz$.

3.4 Procedure

3.4.1 Record Some Speech

1. Using the `rec` function of the SoX package create a two second recording of yourself saying one of the 10 digits in Kiswahili.
 - (a) Start the recording for two seconds

```
rec -c 1 -r 8000 digit.wav trim 0 2
```

This will record a single channel audio signal for 2 seconds sampled at 8 kHz and store it in `digit.wav`
 - (b) Say a digit
 - (c) Play the file to ensure you actually recorded yourself. If you are not satisfied with the recording, try again

```
play digit.wav
```

3.4.2 Compute a spectrogram

We will now compute a spectrogram of the speech signal recorded in the previous section.

1. Launch Octave.
2. Load the speech signal recorded in the previous section and produce a plot of the speech signal with the x-axis representing time in seconds.

```
[x, Fs, bps] = wavread ('digit.wav' );
```
3. We will compute the spectrogram by dividing the speech signal into segments 32ms long (256 samples at 8kHz) with 50% overlap between neighboring segments and taking the FFT of each segment. We plot the magnitude of this FFT for all the segments as a 2-dimensional image using the function `specgram`.

```
alpha=0.5; %Overlap  
N=256;% 32ms window size  
figure(2)  
clf()  
specgram(x,N,Fs,hanning(N),alpha*N);  
xlabel('Time (s)', "fontsize", 18)  
ylabel('Frequency (Hz)', "fontsize", 18)
```

 - (a) Can you tell from the spectrogram the points in the recording containing the spoken digit?
 - (b) Change N so that the segments are 64 ms long and plot a new spectrogram. What is the effect on the spectrogram?
 - (c) In what frequency range is most of the energy of your signal?
 - (d) Can you identify clear peaks in the magnitude spectrum?
4. From the spectrogram and time domain plots, you can determine the regions containing speech and those without.

- (a) Obtain a segment 32ms long from a region with speech and another segment of the same duration from a segment without speech and plot these time domain signals side by side with time in seconds as the x-axis. Comment on the difference between the signals. Compute and compare the energy of both signals.
- (b) Using the `fft` function, compute the DFT of both signals in (a) above and compare the frequency domain content of the signals. Plot the magnitude of the FFT versus frequency in Hz.

A Laboratory Report Format

For each of the laboratory exercises in this manual, you will be required to had in a report. Each report should have the following sections:

1. Title page:

The title page should include

- Course code and title
- Title of the lab
- Name and registration number of the student
- Date

2. Introduction:

In the introduction you should indicate the objectives of the laboratory and give any necessary background information such as relevant theory.

3. Procedure:

Here you should describe in detail the steps carried out in the lab. The details should allow someone else to reproduce your work. You can include short code segments here but code listings should be placed in the appendix.

4. Results and Discussion:

Here you should describe all the results obtained and compare them with what was theoretically expected. Data should be presented appropriately using tables and figure which are clearly labelled

5. Conclusion:

Summarize the laboratory report and indicate whether the objectives of the lab were achieved.

6. References:

Include any references using the IEEE format <http://www.ieee.org/documents/ieeecitationref.pdf>.

7. Appendix:

Include any additional information such as detailed schematics and code listings here.