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 1: Introduction to Octave **3**
3.1 Objectives 3
3.2 Background 3
3.2.1 Starting Octave 3
3.2.2 Simple calculations 3
3.2.3 Built in Functions 4
3.2.4 Vectors and Matrices 4
3.2.5 Plotting Graphs 4
3.2.6 Flow control 5
3.2.7 Scripts and Functions 5
3.3 Procedure 6

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 1: Introduction to Octave

3.1 Objectives

The objectives of this laboratory are:

1. To introduce the Octave programming language.
2. To demonstrate vector manipulation operations.
3. To demonstrate flow control in Octave.
4. To demonstrate definition of functions in Octave.

3.2 Background

Octave is a high level programming language that is very similar to Matlab. It is very useful for numerical computations and this makes it ideal for prototyping DSP applications. Also, by making use of a number of packages, you have access to a number of useful functions.

3.2.1 Starting Octave

1. To start Octave, type octave in the command line.
2. Octave will start and display the promt
   octave:1>
3. To determine the version of Octave installed, type version in the command line.

3.2.2 Simple calculations

You can use Octave as a simple calculator.

1. To add two numbers octave:1> 2+2
2. To subtract two numbers octave:1> 8-2
3. To multiply two numbers octave:1> 2*2
4. To raise one number to a power octave:1> 22 or octave:1> 2**2
5. We can define variables and perform mathematical operations on them for example
   octave:1> N=2
   octave:1> N*N

3.2.3 Built in Functions

There are a number of built in functions in Octave

1. For a list of built in functions visit `http://octave.sourceforge.net/octave/overview.html`
2. Determine the trigonometric functions present.
3. To compute the sine of an angle x in radians type

   sin(x)

   For example to compute $\sin(2\pi)$ type

   sin(2*pi)

   note that pi is a built in variable and we can determine its value by typing pi in the command line.
4. To find out what a function does and how it works, use help. For example help sin.

3.2.4 Vectors and Matrices

Vectors and matrices are very important in DSP applications. Many signals we will manipulate will be represented by vectors.

1. We can define a vector by placing its elements within square brackets.

   a=[1 2 3 4]
2. We can determine the first element of the vector by typing a(1)
3. We can determine the dimensions of the vector using the function size(a)
4. As defined, a is a row vector. We can form a column vector by typing a' which computes the transpose of a vector.
5. We can create M-by-N matrices using the functions ones and zeros. For example

   ones(2,2)
6. We can take the element-wise multiplication of two vectors using the .* operator, for example a.*a.
7. We can also take the dot product of two vectors by multiplying the vectors a*a'
8. Two compatible matrices can be multiplied using the * operator, for example ones(5,5)*ones(5,1)

3.2.5 Plotting Graphs

Octave can be used to produce good quality graphs for reports and publications.

1. Type help plot as read the help message.
2. To plot the simple function $y = x$ in the interval $0 \leq x \leq 1$,
   (a) Generate the vector x=linspace(0,1,100);

(b) Define y, y=x;

(c) To generate the figure type plot(x,y)

3. Explore the use of title, xlabel, ylabel, legend to label the axes and include titles for graphs.

4. The figure can be saved as a jpeg file by typing print('figname.jpg','-djpg')

3.2.6 Flow control

Here we discuss for loops, while loops and if else statements.

1. The syntax of a for loop is:

   ```
   for index = vector
   expression
   end
   ```

   For example to simply print the numbers one to ten we would write

   ```
   for i=1:10
   i
   end
   ```

2. The syntax of a while loop is:

   ```
   while (condition)
   expression
   end
   ```

   For example to print the numbers one to ten we would write i=1;

   ```
   while i <= 10
   i
   i++;
   end
   ```

3. The syntax for if else statements is

   ```
   if (condition)
   expression
   else
   expression
   end
   ```

3.2.7 Scripts and Functions

Sometimes it is more convinient to write a sequence of commands in a text file than to type them one after another in the command line. These files a saved with a .m extension and run from the octave command line by typing the filename without the .m extension.

Also when we want to repeat the same operation on different inputs, it is convinient to write a function. The code is written in a text file using the following format

```
function [output1,output2,...]=function_name(input1,input2,...)
... code goes here ...
```

The file is saved as function_name.m and this function can be called from other scripts.

3.3 Procedure

1. Sample the function $x(t) = \sin(2\pi t)$ at 100Hz and plot the function for $-2 \leq t \leq 2$. What is the period, T_0, of $x(t)$? What is the sampling period T_s?

2. Let $x[n]$ be the discrete time signal formed from sampling $x(t)$ at 100Hz. Is $x[n]$ periodic? What is the period N? If $x[n]$ is periodic, select an index n and print $x[n]$ and $x[n+N]$. Are the values equal? If not explain any differences observed.

3. In continuous time, the power of a periodic signal is given by

$$P_x = \frac{1}{T_0} \int_0^{T_0} |x(t)|^2 dt$$

   (a) What is the power of $x(t)$

   (b) Show that we can estimate P_x using the samples $x[n]$ as

$$P_x \approx \frac{1}{T_0} \sum_{n=1}^{N} x^2[n] T_s$$

   (c) Write a for loop to estimate P_x. How does your value compare with the value computed theoretically.

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.