AICTE-SDP on Open Source Software, PSG Tech, June 22nd,2007 – July 6th,2007

Scilab

for

Digital Signal Processing

S Subathra

Project Engineer, NRCFOSS

AU-KBC Research Centre

MIT Campus of Anna University

Chennai – 600 044

subathra@au-kbc.org

National Resource Centre for Free & Open Source Software

Plan for Presentation

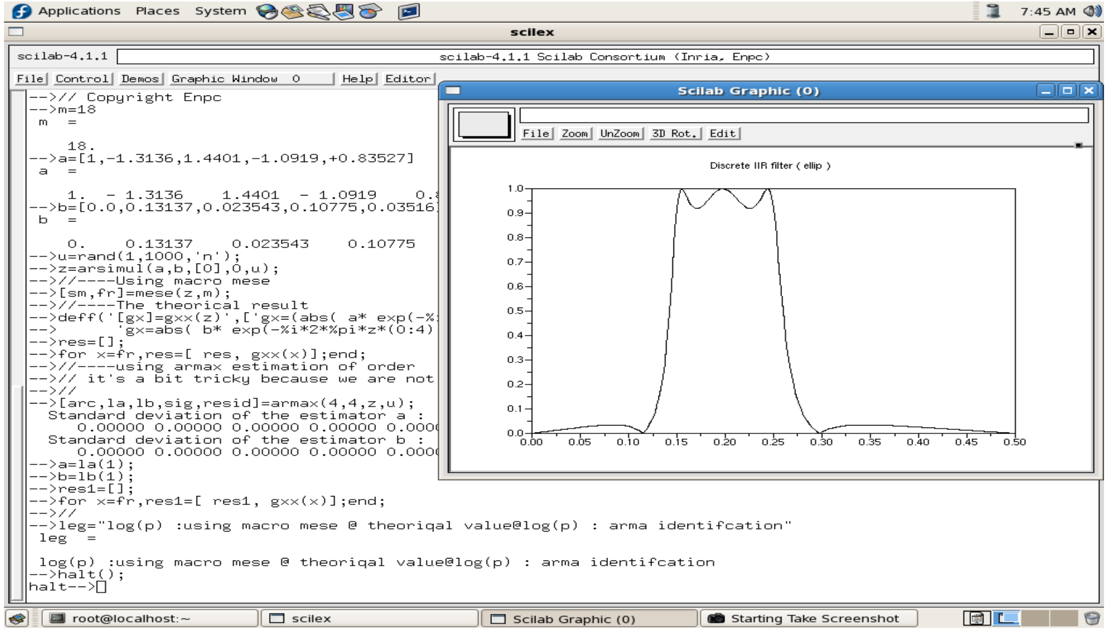
Tutorial Session

- Signal Processing with Scilab   45mins (Tutor)

Lab Session

- Installation Demo   10mins (Tutor)
- Installation of Scilab   15mins (Participants)
- Scilab Build-in Demo   50mins (Participants)
- Hands-on Practise   60mins (Participants)

Tutorial Session



Signal Processing with Scilab

Outline

- Scilab
- Signal Processing
- Handling vectors & matrices
- Using plot functions
- Signal generation
- Sampling
- Convolution
- Correlation
- Discrete Fourier Transform
- Digital Filter Design

Scilab

- Free software
- Numerical programming
- Rapid Prototyping
- Extensive libraries and toolboxes
- Easy for matrix manipulation

Scilab

- Free software
- Nume
- Rapid
- Extensive libraries and toolboxes
- Easy for matrix manipulation

Ideal for Signal Processing

Signal Processing

- A signal carries information, and the objective of signal processing is to extract useful information carried by the signal.
- The method of information extraction depends on the type of signal and the nature of the information being carried by the signal.
- Signal Processing is concerned with the mathematical representation of the signal and the algorithmic operation carried out on it to extract the information present.

Signal Processing

- A 1-D signal is a function of a single independent variable.
  - Eg: Speech Signal => time
- A 2-D signal is a function of a two independent variable.
  - Eg: Image(like photograph) => two spatial variables
- A M-D signal is a function of more than one variable.
  - Eg: Each frame of a black-and-white video signal is a 2-D image signal that is a function of two discrete spatial variables, with each frame occurring sequentially at discrete instants of time. => two spatial variables & time

Signal Processing

- The representation of the signal can be in terms of basis functions in the domain of the original independent variable(s), or it can be in terms of basis functions in a transformed domain.
- A signal can be generated by a single source or by multiple sources.
  - Single Source => Scalar
  - Multiple Source => Vector

Signal Processing

- The representation of the signal can be in terms of basis functio… variable … in a transfor…

How to process vectors with Scilab

- A signa… multiple…
  - Single Source => Scalar
  - Multiple Source => Vector

Handling vectors

- To create a zero vector with 20 dimensions

  >a=zeros(1,20);

- To create a vector and store it in a variable

  >b=[1 2 3 5 1];

- To extract the element 1 to 3 of the above vector

  >b(1:3)

  [1 2 3]

Handling matrices

- To initialize matrix

  >m=zeros(3,3);

  >k=rand(3,3);

  >h=[1 4 6 7; 4 3 2 1; 1 4 3 8];

- To extract the elements of the matrix

  >h(2:3,2:4)

  |1 7 8|

  |2 1 5|

- To get the size of matrix

  >size(h)

  >[3 4]

- Transpose of a matrix

  >h=[1 4 6 7; 4 3 2 1; 1 4 3 8];

  >h'

  |1 2 9|

  |4 1 2|

  |6 7 1|

  |7 8 5|

Operator Notation

- To multiply two matrices ‘a’ and ‘b’

  >a*b

- For multiplying transpose of matrix ‘a’ with itself

  >a’*a

- For elementwise multiplication and addition

  >c=a.*b

  >c=a.+b

Using Plot Function

- Plot function is used to produce number of two dimensional plots

- To generates a random vector of length 10 and plots the points with straight line joining each points

```
>y=rand(1,10);
>plot(y);
```

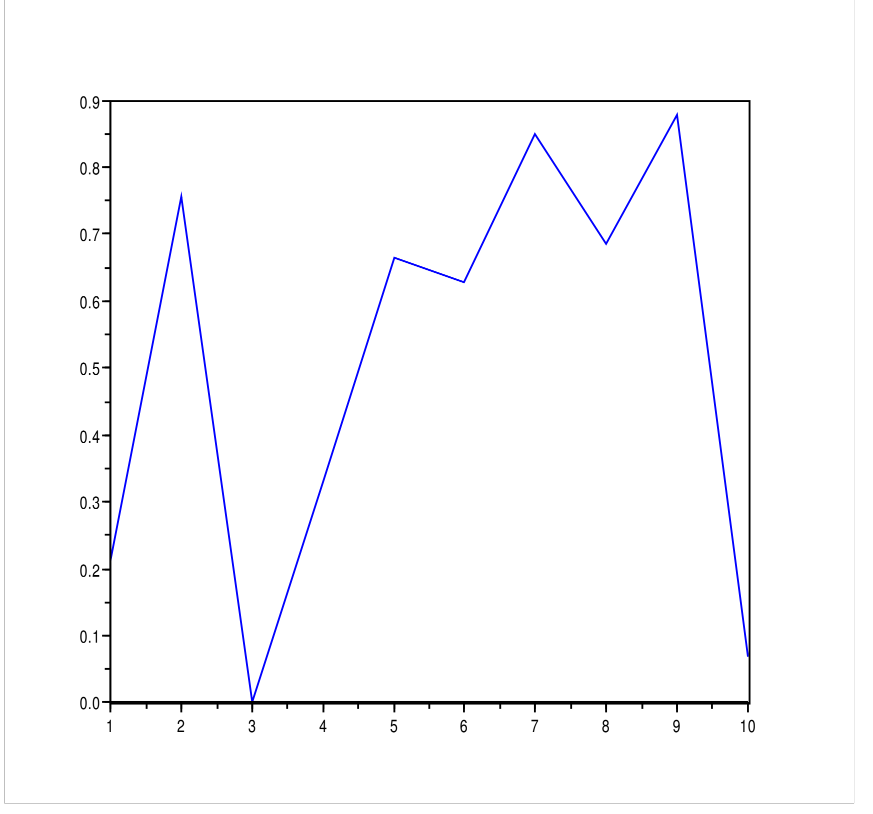


Fig. 1 Plot Function - Sample 1

Using Plot Function – color

- Color of the line joining the points changed using a parameter
  - ‘y’ – yellow
  - ‘c’ – cyan
  - ‘g’ – green
  - ‘w’ – white
  - ‘m’ – magenta
  - ‘r’ – red
  - ‘b’ – blue
  - ‘k’ – black

```
>y=rand(1,10);
>plot(y,’r’);
```

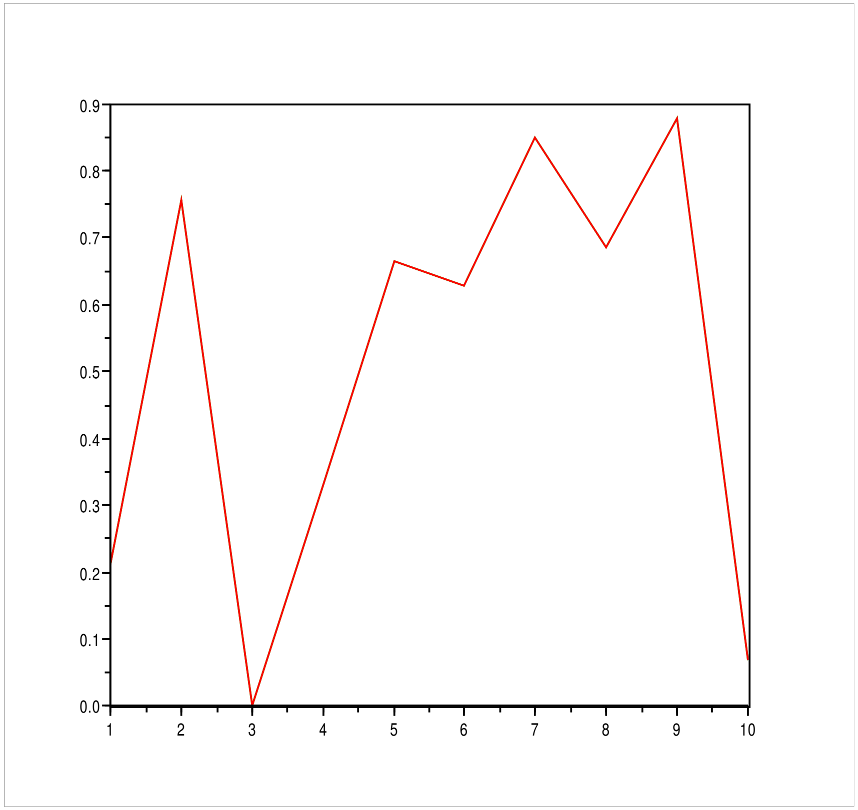


Fig. 2 Plot Function - Sample 2

Using Plot Function

- To produce a discontinious plot
  - ‘.’ – uses dot at each point
  - ‘*’ – uses asterisks at each point
  - ‘d’ – uses blank diamond at each point
  - ‘^’ – uses upward triangles at each point
  - ‘o’ – uses circle at each point
  - ‘+’ – uses cross at each point
  - ‘v’ – uses inverted triangle at each point

```
>y=rand(1,10);
>plot(y,'r');
```

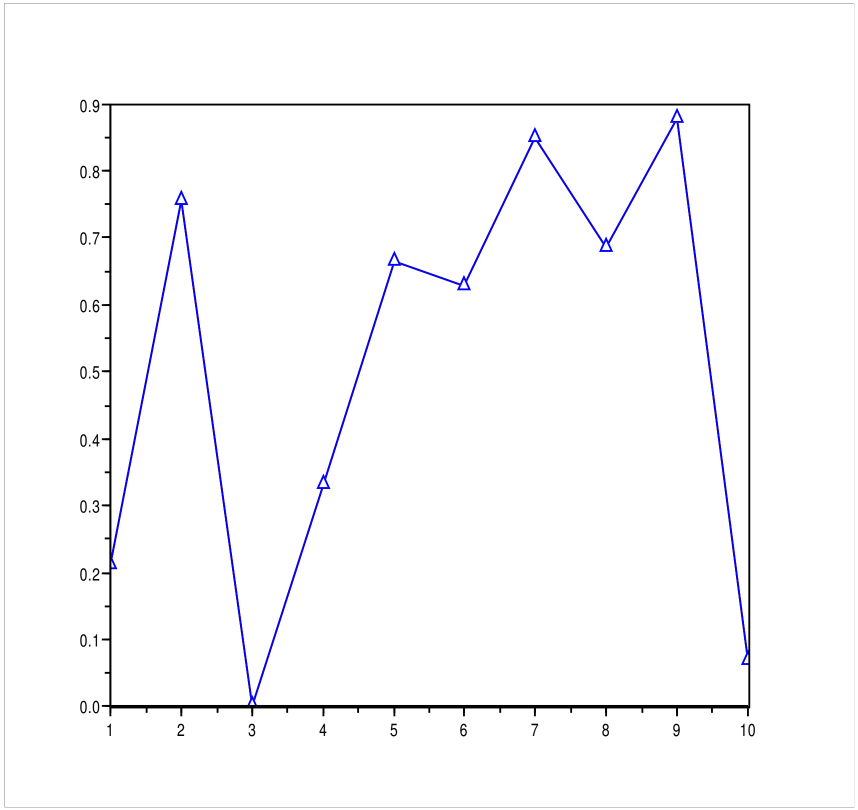


Fig. 3 Plot Function - Sample 3

Using Plot Function – subplot

- Several plots can be accommodated in a single figure window using 'sub-plot' command.

- This command takes three parameters as input, first two parameters specifies the grid size, i.e. how many rows and columns and the third parameter specifies the position of the plot in the grid.

- For e.g.,

  subplot(3,2,2)

  tells scilab that the figure window is divided as three rows, two columns and the plot has to be placed in the second column of the first row.

Using Plot Function – subplot

Example:

```
x1=rand(1,10);
subplot(2,2,1);
plot(x1);
x2=rand(1,20);
subplot(2,2,2);
plot(x2);
x3=rand(1,15);
subplot(2,2,3);
plot(x3);
x4=rand(1,25);
subplot(2,2,4);
plot(x4);
```

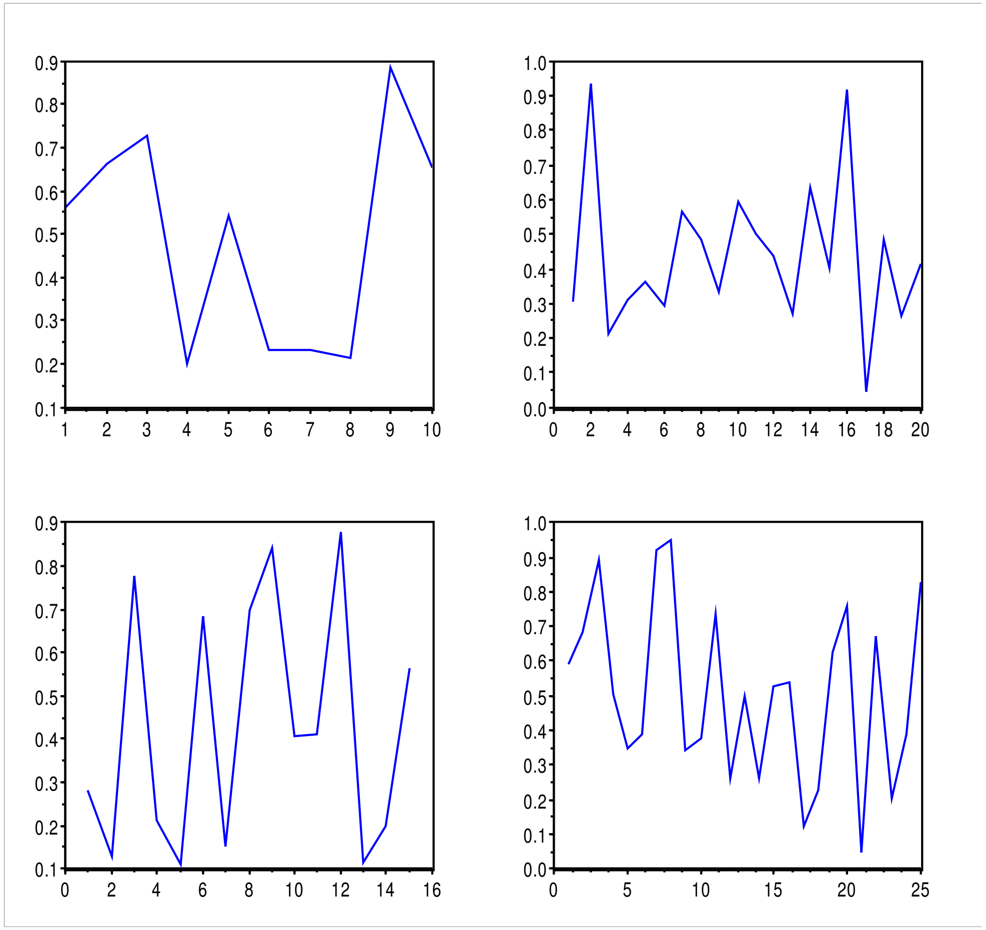


Fig. 4 Plot Function - Sample 4

Using Plot Function - label

- To mark label, the function ‘gca’ returns the handle for the current axes

```
t=1:1:50
y=rand(1,50);
plot(t,y);
a=gca();
a.x_label.text=“time”;
a.y_label.text=“amplitude”;
```

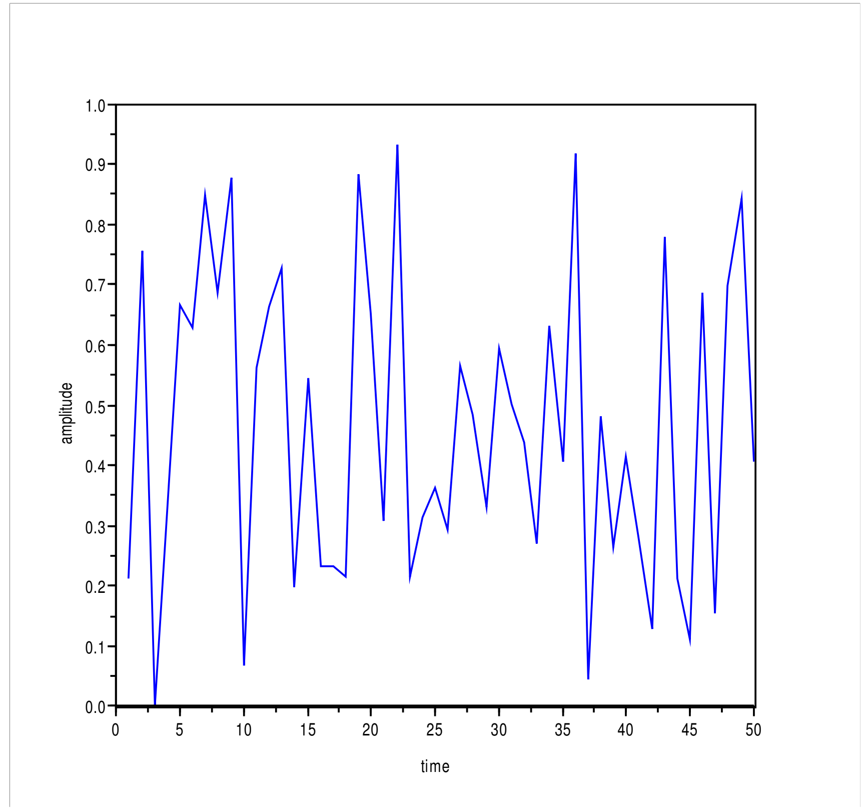


Fig. 5 Plot Function - Sample 5

Using Plot Function - Mesh plot

- Mesh plot can be generated using plot3d command.

```
m=rand(5,7);
i=1:1:5;
j=1:1:7;
plot3d(i,j,m);
```

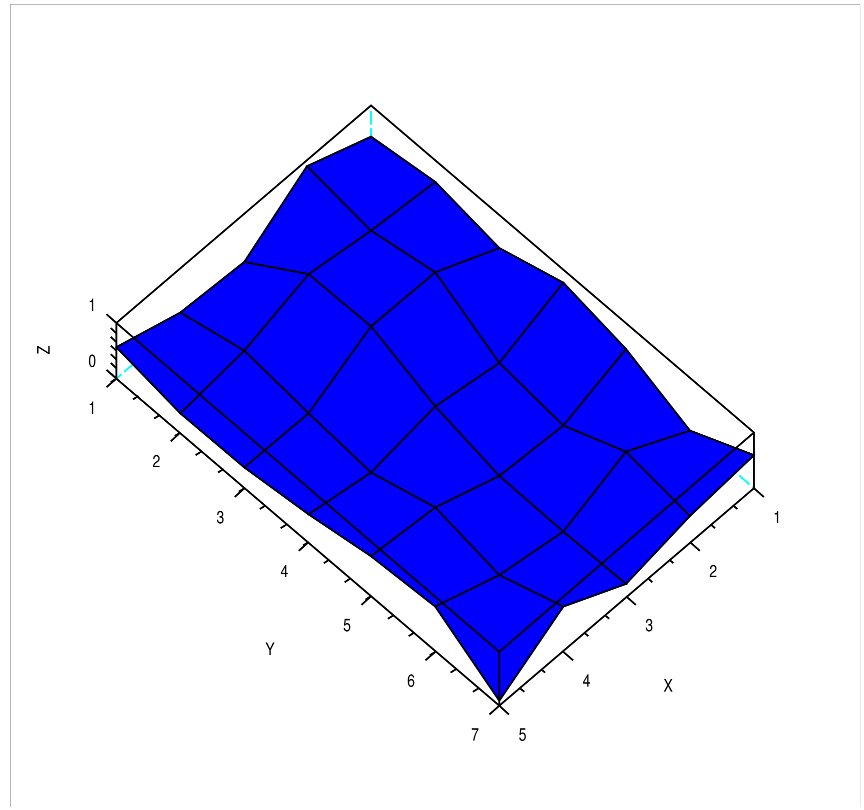


Fig. 6 Plot Function - Sample 6

Signal generation

```
//sine wave
t=0:0.01:3.14;
y=sin(2*3.14*t);
subplot(2,1,1);
a=gca();
a.x_label.text="Time";
a.y_label.text="Amplitude";
plot2d(t,y);
xtitle("Sine Wave Generation");

//cos wave
t=0:0.01:3.14;
y=cos(2*3.14*t);
subplot(2,1,2);
a=gca();
a.x_label.text="Time";
a.y_label.text="Amplitude";
plot2d(t,y);
xtitle("Cosine Wave Generation");
```

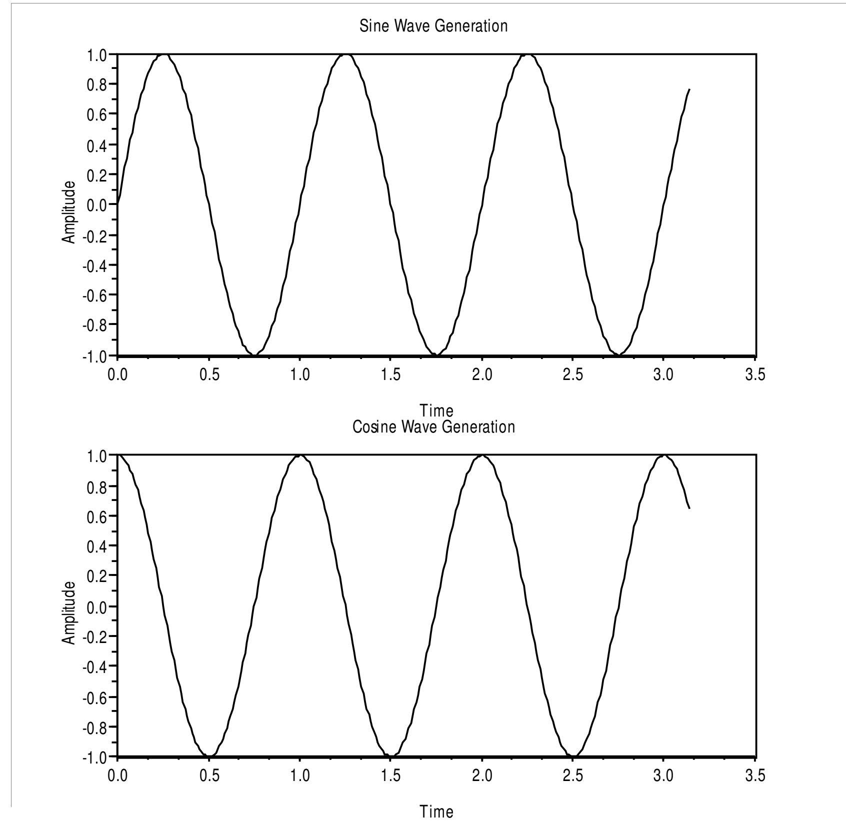


Fig. 7 Signal generation

Sampling

- The process in which the analog continuous signal are measured at equal interval of time and finally a discretized set of digital numbers are created.

- For proper sampling of analog signal Nyquist-Shannon sampling theorem should be satisfied, i.e. sampling frequency should be a greater than twice the maximum frequency of the input signal.

Sampling - Example

```
fo=input('Frequency of since wave in hz=');
ft=input('Sampling frequency in hz=');
t=0:0.01:1;
T=1.0/ft;
x=sin(2*3.14*fo*t);
subplot(2,1,1);
a=gca();
a.x_label.text="Time";
a.y_label.text="Amplitude";
plot(t,x);
xtitle("Continuous signal");

n=0:ft;
y=sin(2*3.14*fo*n*T);
subplot(2,1,2);
a=gca();
a.x_label.text="Time";
a.y_label.text="Amplitude";
plot2d3(n,y);
xtitle("Sampled Signal");
```

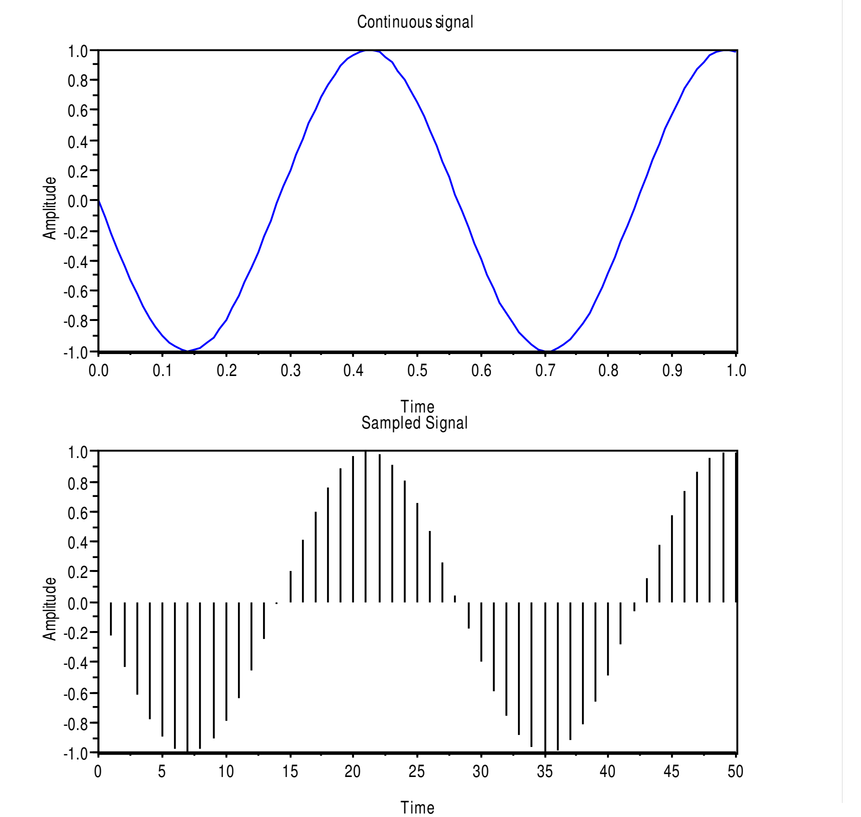


Fig. 8 Sampling

Convolution

- Convolution is the process in which the response of a LTI system is computed for an input signal.

- The LTI system can be defined using the impulse response of the system.

- Convolution of the input signal and impulse response of the system gives the output signal. This process is also called digital filtering.

- Mathematically,

$$g(n) = \sum_{t=-\infty}^{\infty} h(n-t) * x(t)$$

Correlation

- Correlation takes two signals as input and produces third signal as output.
- If both the input signals are one and the same, then it is called as auto correlation
- If both the input signals are different, then it is called as cross correlation
- Correlation is used to measure the similarity between the two input signals at that particular time.
- Mathematically,

$$g(\tau) = \sum_{n=-\infty}^{\infty} x_1(n) x_2(n-\tau)$$

Discrete Fourier Transform

- Used for analysing the frequency components of a sampled signal
- DFT decomposes the input signal into set of sinusoids
- In DFT, we take a sequence of real numbers(sampled signal) as input and it is transformed into a sequence of complex numbers.
- Mathematicall

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N} \qquad k = 0, 1, \ldots\ldots N-1$$

Digital Filter Design – Analog & Digital

- Filter - Used to remove the unwanted signal from the useful signal.
  - For eg, removing noise from audio signal

- Two types – Analog & Digital filters
  - Analog filters uses inductors, capacitors to produce the required effects
  - Digital filter uses modern processors for doing the same

Digital Filter Design – FIR & IIR

- Digital filters are classified as non recursive filters(FIR) and recursive filters(IIR)
  - FIR Filter – the output will be dependent on the current & previous input values
  - IIR Filter – dependent on previous output in addition to the input values

FIR – LPF

```
// Low pass filter
fp=input('Enter the cutoff frequency in
    Hz -- fp=');
n=input('Enter the order of the filter --
    n=');
F=input('Enter sampling frequency in Hz
    --F=');
wc=fp/F;
[coeffval,famp,ffreq]=wfir('lp',n,[wc 0],
    'hm', [0 0]);
//frequency response of the filter
plot2d(ffreq,famp);
a=gca();
a.x_label.text="Frequency";
a.y_label.text="Magnitude";
xtitle("FIR low pass filter");
```

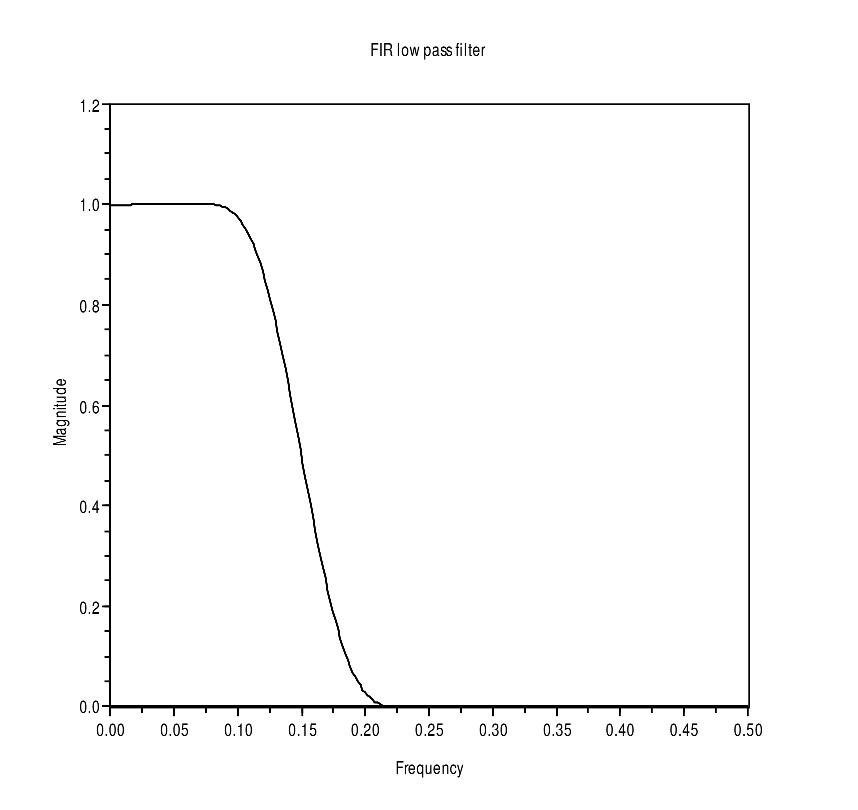


Fig. 9 FIR - LPF

FIR – HPF

```
// High pass filter
fp=input('Enter the cutoff frequency in
    Hz -- fp=');
n=input('Enter the order of the filter --
    n=');
F=input('Enter sampling frequency in Hz
    --F=');
wc=fp/F;
[coeffval,famp,ffreq]=wfir('hp',n,[wc 0],
    'hm', [0 0]);
//frequency response of the filter
plot2d(ffreq,famp);
a=gca();
a.x_label.text="Frequency";
a.y_label.text="Magnitude";
xtitle("FIR high pass filter");
```

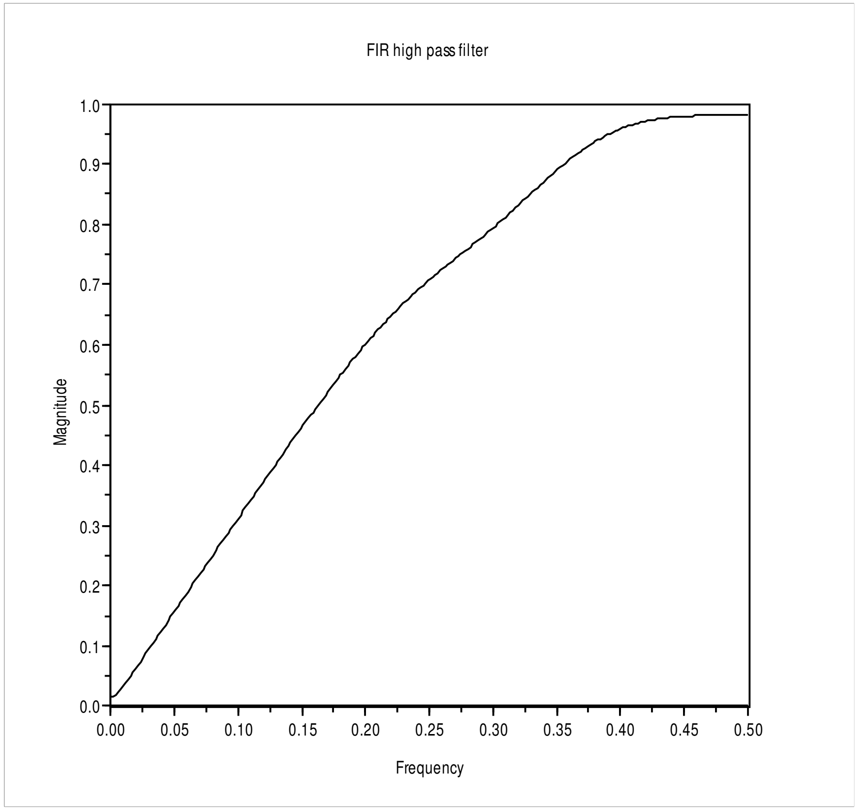


Fig. 10 FIR - HPF

IIR – LPF

```
// Low pass filter
fp=input('Enter the cutoff frequency in Hz --
    fp=');
n=input('Enter the order of the filter --n=');
F=input('Enter sampling frequency in Hz --
    F=');
wc=fp/F;
hz=iir(3,'lp','butt',[wc 0],[0 0]);
[hzm,fr]=frmag(hz,256);
plot2d(fr',hzm');
a=gca();
a.x_label.text="Frequency";
a.y_label.text="Magnitude";
xtitle('Discrete IIR filter low pass');
q=poly(0,'q'); //to express the result in terms of
    the ...
hzd=horner(hz,1/q) // delay operator q=z^-1
```

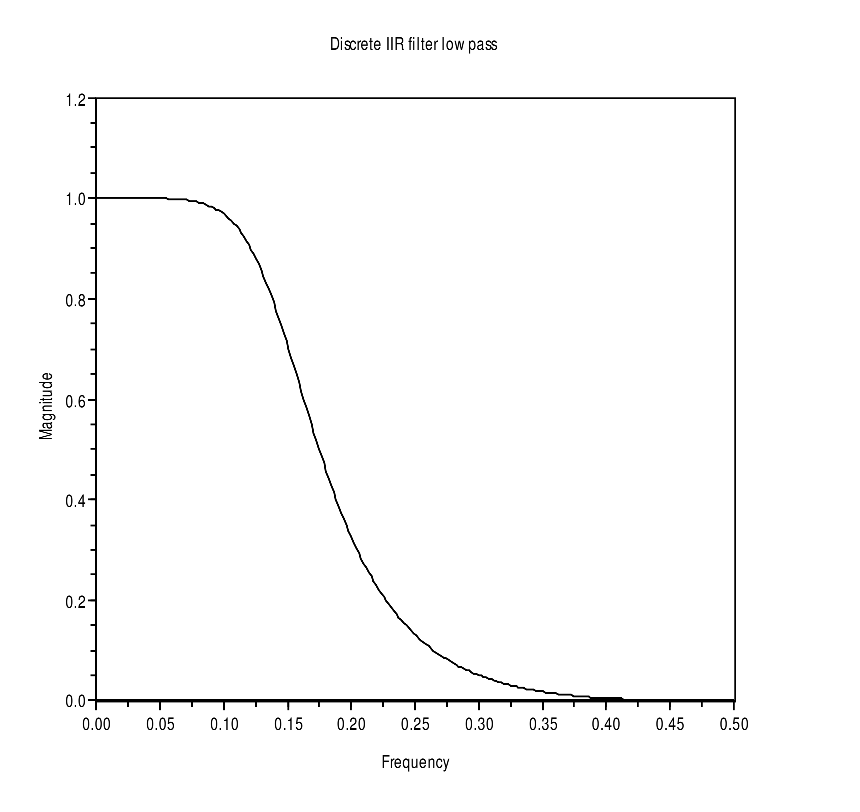


Fig. 11 IIR - LPF

IIR – HPF

```
// High pass filter
fp=input('Enter the cutoff frequency in Hz --
    fp=');
n=input('Enter the order of the filter --n=');
F=input('Enter sampling frequency in Hz --
    F=');
wc=fp/F
hz=iir(3,'hp','butt',[wc 0],[0 0]);
[hzm,fr]=frmag(hz,256);
plot2d(fr',hzm');
a=gca();
a.x_label.text="Frequency";
a.y_label.text="Magnitude";
xtitle('Discrete IIR filter high pass');
q=poly(0,'q'); //to express the result in terms of
    the ...
hzd=horner(hz,1/q) // delay operator q=z^-1
```

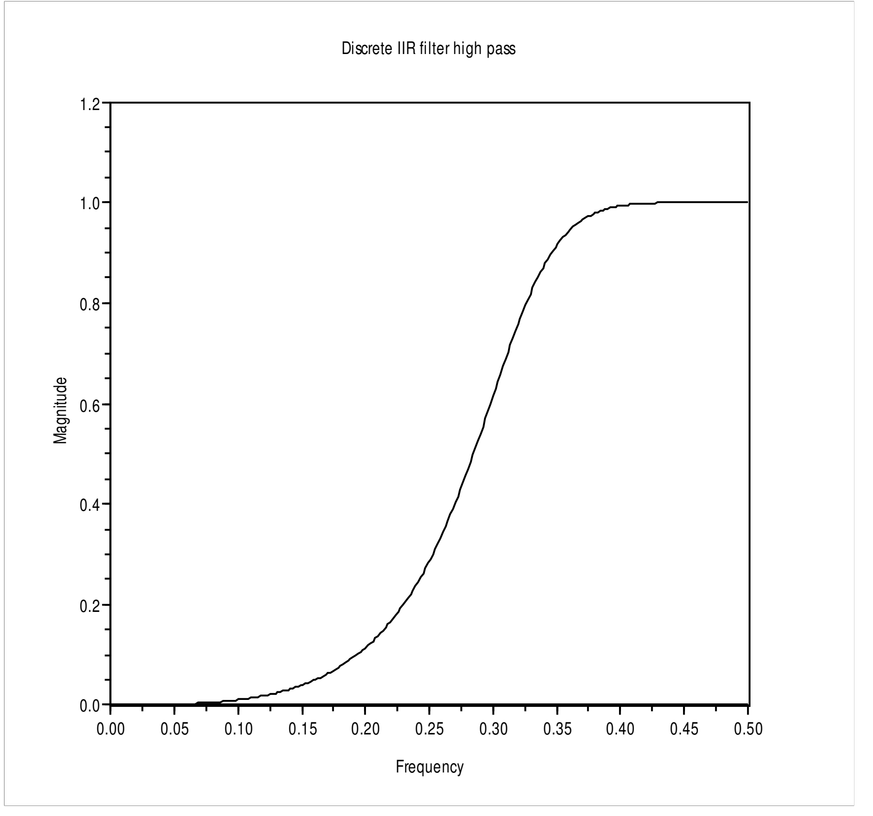


Fig. 12 IIR - HPF

Reference

[1] John G.Proakis and Dimitris G.Manolakis, "Digital Signal Processing principles, algorithms, and applications" – Third edition.

[2] Sanjit K. Mitra, "Digital Signal Processing a computer based approach" – Second edition.

[3] Scilab Documentation, Available at www.scilab.org

Lab Session

Installation

Scilab Installation – How To

- Go to www.scilab.org
- Check for the latest downloadable version of scilab (i.e. scilab-4.1.1.bin.linux-i686.tar.gz) and download it.
- Untar the scilab-4.1.1.bin.linux-i686.tar.gz using the command

  # tar –xzvf scilab-4.1.1.bin.linux-i686.tar.gz

  where

  x refers to extract,

  z refers to gzip,

  v refers to verbose,

  f refers to file
- Run Scilab by executing “scilab”(shell script in bin)

Scilab Build-in Demo

Scilab demonstration

http://www.scilab.org/doc/demos_html/index.html

Matlab and Scilab functions

http://www.scilab.org/product/dic-mat-sci/M2SCI_doc.htm

More about DSP

Signal Processing Information Base (SPIB)

http://spib.rice.edu/spib.html

DSP Online Classes

http://bores.com/index_dsp.htm

http://bores.com/index_online.htm

Thank you!