Experiments in Matlab and Scilab : A Comparative Study

Imteyaz Ahmad

Associate Professor, Department of ECE, BIT Sindri, Dhanbad, India

ABSTRACT: Eight experiments were performed in Matlab. and Scilab. These experiments were from ploting sinewave ,generating basic signals(unit sample,unit step ,ramp),adding noise to sinewave and then removing it using suitable filters was done. A sinewave having sampling frequency of 256 Hz was choosen to perform experiment in Matlab. and Scilab. The message signal is of 1Hz.A random noise of same sampling frequency was added. A low pass FIR filter of order 51 was applied to remove high frequency noise.In Matlab. in built function to calculate SNR and MSE are available this is not the case with Scilab. Performance estimation in Matlab. was done, the input signal added with random noise is having SNR 4.07 dB and filtered output signal SNR is 2.33 dB due to delay of 97ms because of order of 51 in FIR LPF. Similarly PLI noise of 50 Hz was added and analysis was done, It has been observed that Matlab. is more user suitable to estimate performance of filter used than Scilab.

KEYWORDS: Matlab., Scilab.,FIR filter, SNR and MSE

I. INTRODUCTION

MATLAB is a high-level language and interactive environment that enables one to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran." The web page of the MathWorks, Inc. at www.mathworks.com states that Matlab was originally created by Cleve Moler, a Numerical Analyst in the Computer Science Department at the University of New Mexico. The first intended usage of Matlab, also known as Matrix Laboratory, was to make LINPACK and EISPACK available to students without facing the difficulty of learning to use Fortran. Steve Bangaret and Jack Little, along with Cleve Moler, recognized the potential and future of this software, which led to establishment of MathWorks in 1983. As the web page states, the main features of Matlab include high-level language; 2-D/3-D graphics; 7 mathematical functions for various fields; interactive tools for iterative exploration, design, and problem solving; as well as functions for integrating MATLAB-based algorithms with external applications and languages[1,2,3]. In addition, Matlab performs the numerical linear algebra computations using Basic Linear Algebra Subroutines (BLAS) and Linear Algebra Package (LAPACK)[4].

Scilab. is an open source, cross-platform numerical computational package as well as a high-level, numerically oriented programming language." Scilab was written by INRIA, the French National Research Institution, in 1990. The web page for Scilab at www.scilab.org states the syntax is largely based on Matlab language. This software is also intended to allow Matlab users to smoothly utilize the package. In order to help in this process, there exists a built in code translator which assists the user in converting their existing Matlab codes into a Scilab code. According to Scilab's web page, the main features of Scilab include hundreds of mathematical functions; high-level programming language; 2-D/3-D visualization; numerical computation; data analysis; and interface with Fortran, C, C++, and Java. Just like Octave, Scilab is also a free software distributed under CeCILL licenses. Scilab is fully compatible with Linux, Mac OS, and Windows platforms. Like Octave, the source code is available for usage as well as for editing. For

downloading instructions, refer to Appendix A.3. Scilab uses the two numerical libraries, LAPACK and BLAS[5,6,7]. Unlike Octave, the syntax and built-in Scilab functions may not entirely agree with Matlab[8]

II. METHODOLOGY

The experiment were performed to add random noise to message signal(frequency 1 Hz) in both Matlab. and Scilab[9,10,11]. and Low pass FIR filter of order 51 was used. In Matlab. in built function to calculate SNR and MSE are available this is not the case with Scilab. Analysis through experiment was done.

To add noise to a signal in Scilab

```
n=257; fs=256;
 t=0:1/fs:1;
 x=sin(2*%pi*t);
 no=rand(1,n);
y=x+no;
clf;
subplot(3,1,1);
plot(x);
 subplot(3,1,2);
plot(no);
subplot(3,1,3);
 plot(y);
```
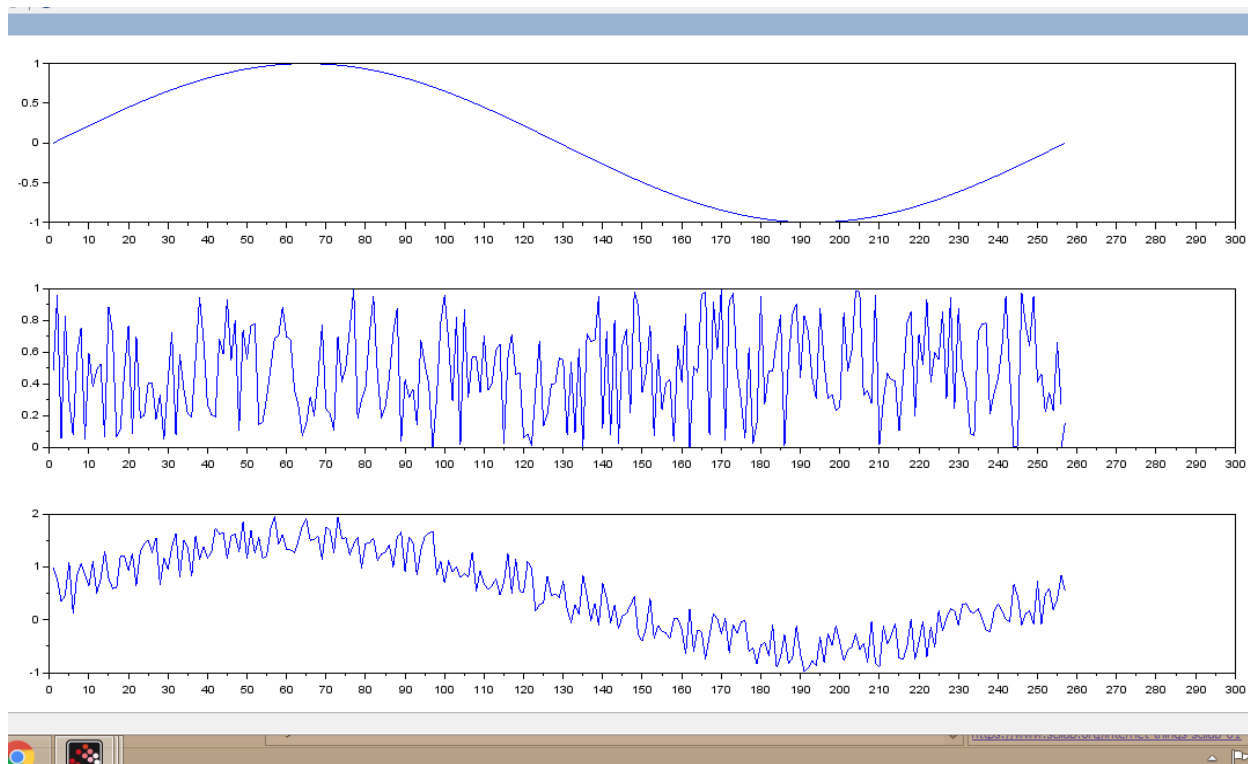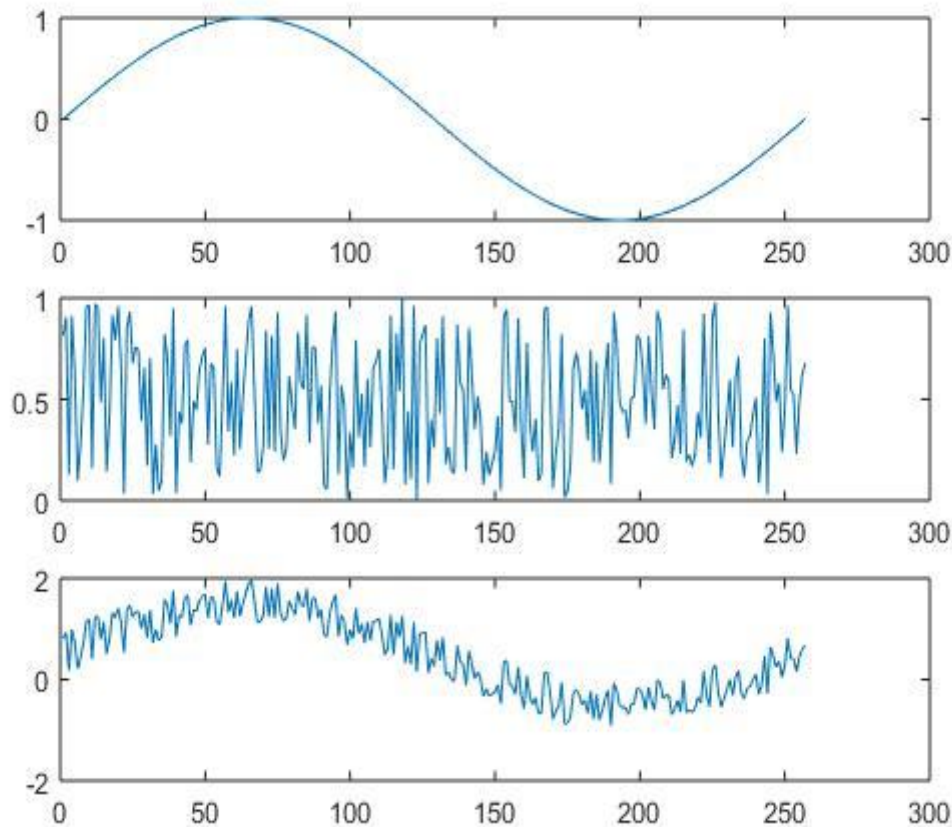
Result:

To add noise to a signal in Matlab

```
n=257;
fs=256;
 t=0:1/fs:1;
 x=sin(2*pi*t);
 no=rand(1,n);
y=x+no;
clf;
subplot(3,1,1);
plot(x);
 subplot(3,1,2);
plot(no);
subplot(3,1,3);
 plot(y);
```
Result:

to remove high frequency noise using FIR LPF in Scilab.

```
n=257;
Fs=256;
 t=0:1/Fs:1;
 x=sin(2*%pi*t);
 no=rand(1,n);
y=x+no;
clf;
Subplot(4,1,1);
plot(x);
 subplot(4,1,2);
plot(no);
subplot(4,1,3);
 plot(y);
 Fc=2;
N=51;
Fcn=2*Fc/Fs;
 h=wfir("lp",N-1,[Fcn],"re",[Fcn]);
 r=filter(h,1,y);
 subplot(4,1,4);
plot(r);
```

Result:
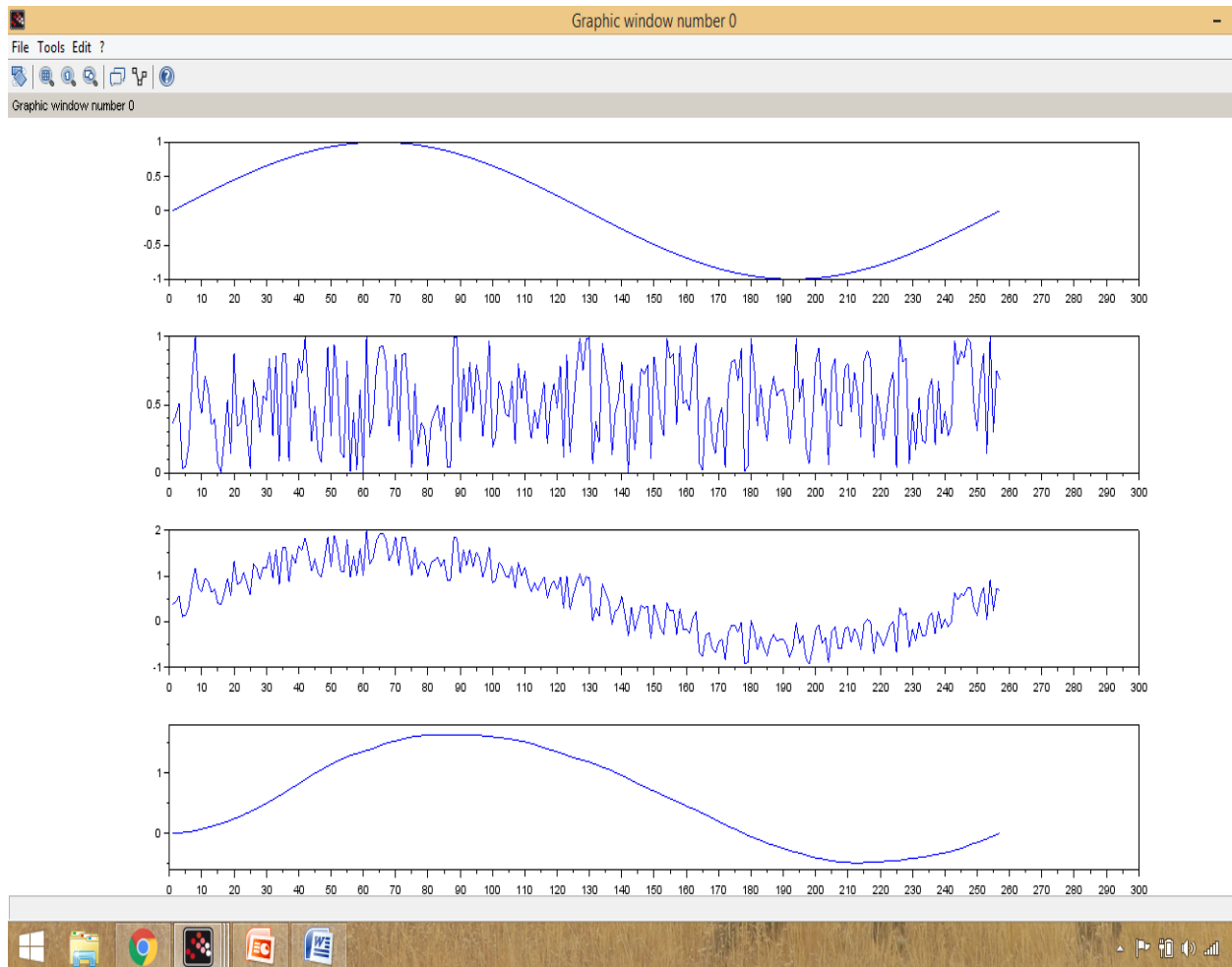
to remove high frequency noise using FIR LPF in Matlab.

```
n=257;
Fs=256;
 t=0:1/Fs:1;
 x=sin(2*pi*t);
 no=rand(1,n);
y=x+no;
clf;
subplot(4,1,1);
plot(x);
 subplot(4,1,2);
plot(no);
subplot(4,1,3);
 plot(y);
 Fc=2;
N=51;
Fcn=2*Fc/Fs;
```
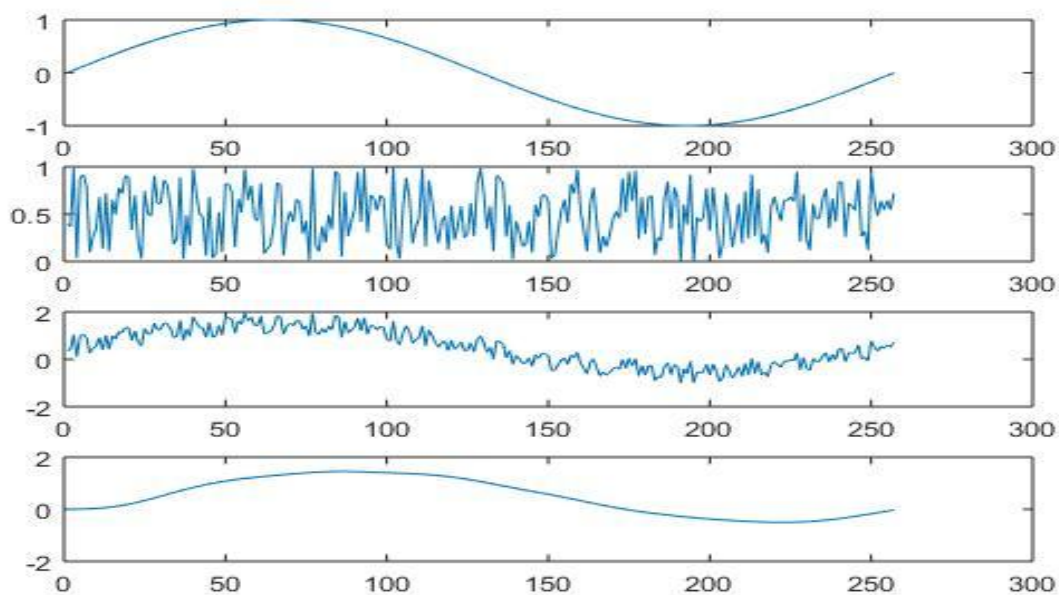
```
h=fir1(50,Fcn,'low');
r=filter(h,1,y);
subplot(4,1,4);
plot(r);
```
Result:

: to remove low frequency noise using FIR high pass filter in Scilab.
```
clf;
n=257;
Fs=256;
t=0:1/Fs:1;
x=sin(2*%pi*10*t);
xn=2*sin(%pi*t/2);
y=x+xn;
subplot(4,1,1);
plot(x);
subplot(4,1,2);
plot(xn);
subplot(4,1,3);
plot(y);
h=wfir("hp",31,[0.046875],"hm",[0.046875]);
r=filter(h,1,y);
subplot(4,1,4);
plot(r);
```
Result: : to remove low frequency noise using FIR high pass filter in Scilab.
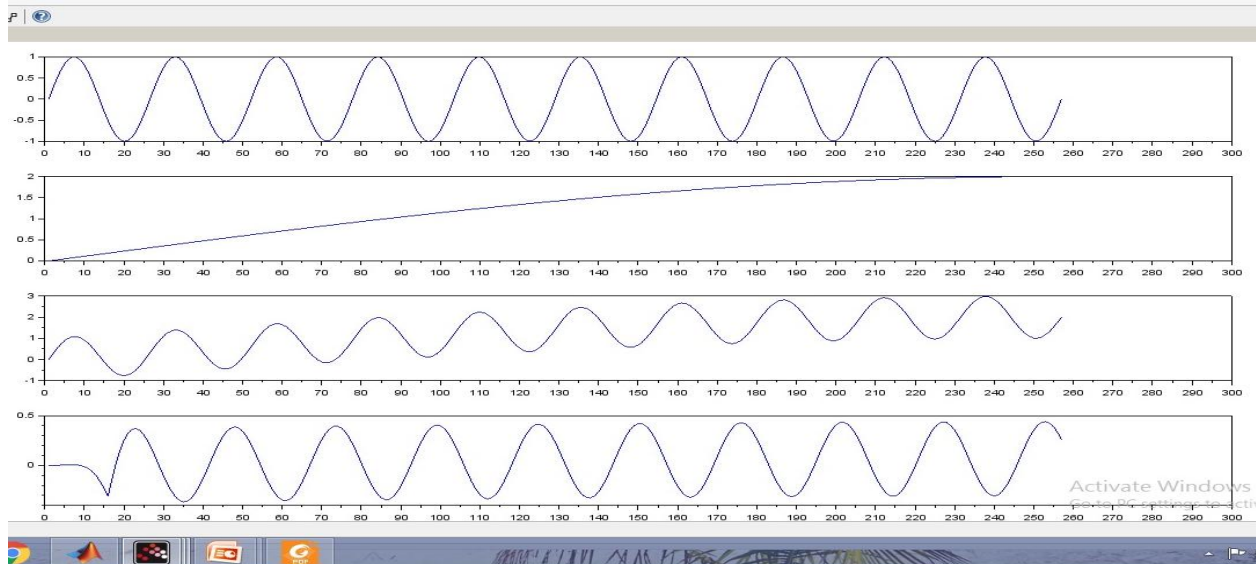
```
clf;
n=257;
Fs=256;
t=0:1/Fs:1;
x=sin(2*%pi*10*t);
xn=2*sin(%pi*t/2);
y=x+xn;
subplot(4,1,1);
plot(x);
subplot(4,1,2);
plot(xn);
subplot(4,1,3);
plot(y);
h=wfir("hp",31,[0.046875],"hm",[0.046875]);
r=filter(h,1,y);
subplot(4,1,4);
plot(r);
```


: to remove low frequency noise using IIR high pass filter in Matlab.

```
Fs=256;
t=0:1/Fs:1-1/Fs;
x=sin(2*pi*10*t);
xn=sin(pi*t/2);
y=x+xn;
subplot(4,1,1);
plot(x);
subplot(4,1,2);
plot(xn);
```

```
 subplot(4,1,3);
 plot(y);
Fc=6;
Fcn= 2*Fc/Fs;
[b,a]=butter(4,Fcn,'high');
 r=filter(b,a,y);
 subplot(4,1,4);
 plot(r);
```

Result:
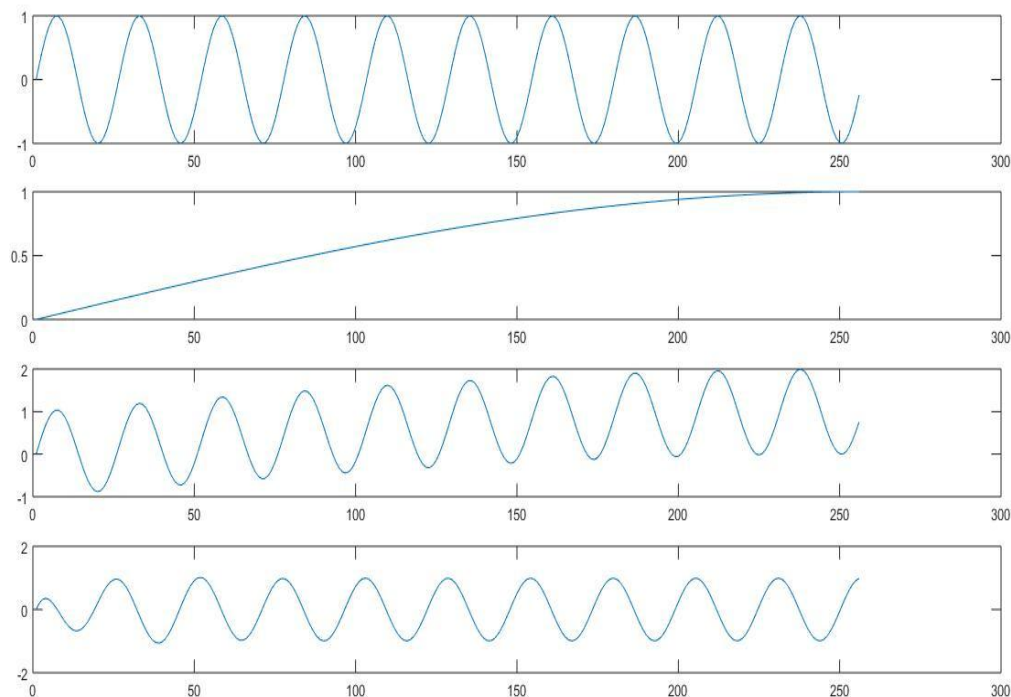
III. OBSERVATIONS

In Scilab, the Nyquist frequency (Fs/2=128 Hz) is normalized as 1. lp stands for LPF ,N-1 is the order of the filter, re stands for rectangular window and Fcn is the normalized cutoff frequency.FIR filter is having only zeros that is numerator function and denominator is 1.$H(z)=\sum_{n=0}^{n=N-1}h(n)z^{-n}$

In Matlab, the Nyquist frequency (Fs/2=128 Hz) is normalized as 1,N-1 is the order of the filter, Fcn is the normalized cutoff frequency.FIR filter is having only zeros that is numerator function and denominator is 1.
$H(z)=\sum_{n=0}^{n=N-1}h(n)z^{-n}$
 Performance estimation in Matlab. was done, the input signal added with random noise is having SNR 4.07 dB and filtered output signal SNR is 2.33 dB due to delay of 97ms because of order of 51 in FIR LPF.This analysis in Scilab can be done by writing codes which is not user friendly.
Performance estimation in Matlab. was done, for low frequency noise removal, the inpufrequency t signal added with base line wander noise is having SNR 2.87 dB and filtered output signal SNR is -3.58 dB.

Performance estimation in Matlab. was done, for PLI noise removal, the input signal added with PLI noise is having SNR 3.01 dB and filtered output signal SNR is 6.2004 dB.

IV. RESULT

In Matlab. analysis for low frequency ,high frequency and PLI noise were added to message signal(1Hz) was done using inbuilt command SNR, Which is not available in Scilab. Performance estimation in Matlab. was done, for PLI noise removal, the input signal added with PLI noise is having SNR 3.01 dB and filtered output signal SNR is 6.2004 dB.

V. CONCLUSIONS

Eight experiments were performed in Matlab. and Scilab. These experiments were from ploting sinewave ,generating basic signals(unit sample,unit step ,ramp),adding noise to sinewave and then removing it using suitable filters was done. A sinewave having sampling frequency of 256 Hz was choosen to perform experiment in Matlab. and Scilab. The message signal is of 1Hz.A random noise of same sampling frequency was added. In Matlab. analysis for low frequency ,high frequency and PLI noise were added to message signal(1Hz) was done using inbuilt command SNR, Which is not available in Scilab. Performance estimation in Matlab. was done, for PLI noise removal, the input signal added with PLI noise is having SNR 3.01 dB and filtered output signal SNR is 6.2004 dB.

REFERENCES

[1] Kevin P. Allen. Efficient parallel computing for solving linear systems of equations. UMBC Review: Journal of Undergraduate Research and Creative Works, vol. 5, pp. 8–17, 2004.
[2] Dietrich Braess. Finite Elements. Cambridge University Press, third edition, 2007.
[3] James W. Demmel. Applied Numerical Linear Algebra. SIAM, 1997.
 [4] Arieh Iserles. A First Course in the Numerical Analysis of Differential Equations. Cambridge Texts in Applied Mathematics. Cambridge University Press, second edition, 2009.
[5] Jeremy Kepner. Parallel MATLAB for Multicore and Multinode Computers. SIAM, 2009.
 [6] Neeraj Sharma. A comparative study of several numerical computational packages. M.S. thesis, University of Maryland, Baltimore County, 2010.
[7] Neeraj Sharma and Matthias K. Gobbert. Performance studies for multithreading in Matlab with usage instructions on hpc. Technical Report HPCF–2009–1, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2009.
[8] David S. Watkins. Fundamentals of Matrix Computations. Wiley, second edition, 2002.
[9] D V Rama Koti Reddy, Mohammad Zia Ur Rahman, Y Saneetha, N SruthiSudha, "Base line wander and power line interference elimination from cardiac signals using a Novel LMS algorithm based on differential inputs and error," Proceeding of the International Journal of Advanced Engineering & Application. Jan 2011
[10] Dr. K.L. Yadav and S. Singh "Performance evaluation of different adaptive filters for ECG signal processing", International Journal On Computer Science and Engineering, vol. 40, no. 5, pp. 1880–1883, 2010.
[11] Ying He, Hong He, Li Li, Yi Wu "The Applications and Simulation of Adaptive Filter in Noise Canceling". International Conference on Computer Science and Software Engineering 2008 IEEE