ECE 418 STATISTICAL DIGITAL SIGNAL PROCESSING

FINAL PROJECT REPORT

TITLE: ECG SIGNAL DENOISING USING NORMALIZED-LEAST-MEAN-SQUARE ALGORITHM BASED ADAPTIVE FILTER

LAKSHMI SRIDEVI

UIN:668383906

Table of Contents

1.0 ABSTRACT

Electrocardiogram(ECG) is a method of measuring the electrical activities of heart. Every portion of ECG is very essential for the diagnosis of different cardiac problems. But the amplitude and duration of ECG Signal is usually corrupted by different noises. In this project, we will deal with denoising the corrupted ECG signal at hand. The Normalized-Least-Mean-Squares algorithm based adaptive filter is used for this purpose. In many applications involving noise cancellation, the changes in signal characteristics are quite fast. This requires the utilization of adaptive algorithms which converge rapidly. Normalized Least Mean Squares adaptive filters have been used in a wide range of signal processing application because of its simplicity in computation and implementation. The NLMS algorithm has an advantage of low computational complexity. There is always a trade-off between computational complexity and fast convergence when it comes to adaptive filtering algorithms. We approach the NLMS algorithm by updating its filter co-efficient at each time instant, to fulfill a suitable trade-off between convergence rate and computational complexity. The performance of the proposed NLMS algorithm is fully studied through the analysis used in adaptive filters and general expressions.

2.0 INTRODUCTION

ECG is generated by the heart muscle and measured on the skin surface of the body. When the electrical abnormalities of the heart occur, the heart cannot pump and supply enough blood to the body and brain. As ECG is a graphical recording of electrical impulses generated by heart, it is needed to be done when chest pain occurred such as heart attack, shortness of breath, faster heartbeats, high blood pressure, high cholesterol and to check the heart's electrical activity. An ECG is very sensitive, different types of noise and interference can corrupt the ECG signal as the real amplitude and duration of the signal can be changed. ECG signals are mostly affected by white noise, colored noise, electrode movement noise, muscle artifact noise, baseline wander, composite noise and power line interference. These noise and interference makes the incorrect diagnosis of the ECG signal. So, the removal of these noise and interference from the ECG signal has become very crucial. Different types of digital filters (FIR and IIR) have been used to solve the problem. However, it is difficult to apply these filters with fixed coefficients to reduce different types of noises, because the ECG signal is known as a non-stationary signal. Recently, adaptive filtering has become effective and popular methods for processing and analysis of ECG Signal. It is well known that adaptive filters with Least Mean Square(LMS) Algorithm show good performance for processing and analysis of signal which are non-stationary. In this study, we have used the Normalized Least Mean Square filter to denoise the ECG Signal and the performance is evaluated. The Normalized Least Mean square filter is shown to remove all the specified noise more significantly.

3.0 ADAPTIVE FILTERS

An adaptive filter is a filter that self-adjusts its transfer function according to an optimizing algorithm. It adapts the performance based on the input signal. Such filters incorporate algorithms that allow the filter coefficients to adapt to the signal statics. There are different approaches used in adaptive filtering, which are as follows:

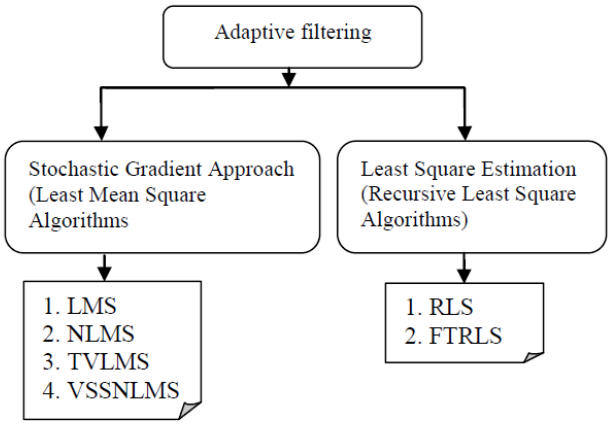

Adaptive filtering
Stochastic Gradient Approach (Least Mean Square Algorithms
Least Square Estimation (Recursive Least Square Algorithms)
1. LMS
2. NLMS
3. TVLMS
4. VSSNLMS
1. RLS
2. FTRLS

Figure1. Hierarchy of adaptive filters

Adaptive techniques use algorithms, which enable the adaptive filter to adjust its parameters to produce an output that matches the output of an unknown system. This algorithm employs an individual convergence factor that is updated for each adaptive filter coefficient at each iteration.

Principle of Adaptive Filter

The block diagram(Figure2.) indicates that, if the value of N(n) is known, then after subtracting this from the mixed signal d(n), the original signal X(n) is obtained. But it is difficult due to the harmonics of noise signal. For this reason, an estimated noise signal N´(n) is calculated through some filters and measurable noise source S(n). If N´(n) is closer to N(n), then the estimated desired signal is X´(n) closer to the original signal X(n).

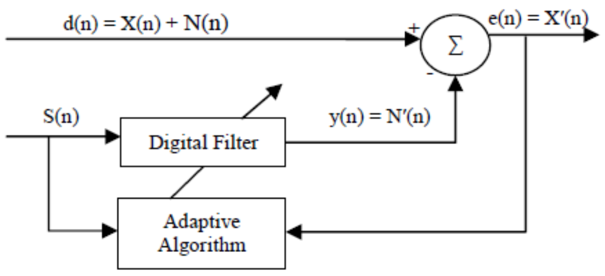


Figure2. Principle of Adaptive Filter.

We will now discuss in detail about the Normalized Least Squares algorithm based adaptive filter in the following sections.

3.1 LEAST MEAN SQUARES ALGORITHM

Least mean squares (LMS) algorithms are class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. It is well known and widely used due to its computational simplicity. It is this simplicity that has made it the benchmark against which all other adaptive filtering algorithms are judged.

The basic idea behind LMS filter is to approach the optimum filter weights (R^{-1} P), by updating the filter weights in a manner to converge to the optimum filter weight. The algorithm starts by assuming a small weight (zero in most cases), and at each step, by finding the gradient of the mean square error, the weights are updated. That is, if the MSE-gradient is positive, it implies, the error would keep increasing positively, if the same weight is used for further iterations, which means we need to reduce the weights. In the same way, if the gradient is negative, we need to increase the weights. So, the basic weight update equation is:

$$\boldsymbol{w(n+1) = w(n) + 2\mu e(n)x(n)} \qquad (1)$$

Here x(n) is the input vector of time delayed input values, x(n) = [x(n) x(n-1) x(n-2) ... x(n-N+1)]$^{T.}$ The vector w(n) = [w0(n) w1(n) w2(n) .. wN-1(n)]^T represents the coefficients of the adaptive FIR filter tap weight vector at time n. The parameter μ is known as the step size parameter and is a small positive constant. This step size parameter controls the influence of the updating factor. Selection of a suitable value for μ is imperative to the performance of the LMS algorithm, if the value is too small the time the adaptive filter takes to converge on the optimal solution will be too long; if μ is too large the adaptive filter becomes unstable and its output diverges.

Implementation of the LMS Algorithm

Each iteration of the LMS algorithm requires 3 distinct steps in this order:

1. The output of the FIR filter, y(n) is calculated using equation 2.
   $$\boldsymbol{y(n) = \sum_{i=0}^{N-1} w(n)x(n-i) = w(n)^T x(n)} \qquad (2)$$
2. The value of the error estimation is calculated using equation 3.
   $$\boldsymbol{e(n) = d(n) - y(n)} \qquad (3)$$
3. The tap weights of the FIR vector are updated in preparation for the next iteration, by equation 4.
   $$\boldsymbol{w(n+1) = w(n) + 2\mu e(n)x(n)} \qquad (4)$$

The main reason for the LMS algorithms popularity in adaptive filtering is its computational simplicity, making it easier to implement than all other commonly used adaptive algorithms. For each iteration the LMS algorithm requires 2N additions and 2N+1 multiplications (N for calculating the output, y(n), one for 2μe(n) and an additional N for the scalar by vector multiplication.

3.2 NORMALIZED LEAST SQUARES ALGORITHM

One of the primary disadvantages of the LMS algorithm is having a fixed step size parameter for every iteration. This requires an understanding of the statistics of the input signal prior to commencing the adaptive filtering operation. In practice this is rarely achievable. The Normalized Least Mean Square algorithm (NLMS) is an extension of the LMS algorithm which bypasses this issue by calculating maximum step size value. Step size value is calculated by using the following formula.

$$\boldsymbol{Step\ size} = \frac{\mathbf{1}}{\boldsymbol{dot\ product}(\boldsymbol{input\ vector}, \boldsymbol{input\ vector})}$$

This step size is proportional to the inverse of the total expected energy of the instantaneous values of the coefficients of the input vector x(n). This sum of the expected energies of the input samples is also equivalent to the dot product of the input vector with itself, and the trace of input vectors auto-correlation matrix, R.

$$\boldsymbol{tr}[\boldsymbol{R}] = \boldsymbol{E}\left[\sum_{i=0}^{N-1} \boldsymbol{x}^2(\boldsymbol{n}-\boldsymbol{i})\right] \qquad (5)$$

The recursion formula for the NLMS algorithm is stated in equation 6.

$$\boldsymbol{w}(\boldsymbol{n}+\mathbf{1}) = \boldsymbol{w}(\boldsymbol{n}) + \frac{\mathbf{1}}{\boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n})}\boldsymbol{e}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n}) \qquad (6)$$

Implementation of the NLMS Algorithm

The NLMS algorithm has been implemented in MATLAB. As the step size parameter is chosen based on the current input values, the NLMS algorithm shows far greater stability with unknown signals. This combined with good convergence speed and relative computational simplicity make the NLMS algorithm ideal for the real time adaptive noise cancellation system. As the NLMS is an extension of the standard LMS algorithm, the NLMS algorithms practical implementation is very similar to that of the LMS algorithm. Each iteration of the NLMS algorithm requires these steps in the following order.

1. The output of the adaptive filter is calculated.
   $$\boldsymbol{y}(\boldsymbol{n}) = \sum_{i=0}^{N-1} \boldsymbol{w}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n}-\boldsymbol{i}) = \boldsymbol{w}^{\boldsymbol{T}}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n}) \qquad (7)$$
2. An error signal is calculated as the difference between the desired signal and the filter output.
   $$\boldsymbol{e}(\boldsymbol{n}) = \boldsymbol{d}(\boldsymbol{n}) - \boldsymbol{y}(\boldsymbol{n}) \qquad (8)$$
3. The step size value for the input vector is calculated.
   $$\boldsymbol{\mu}(\boldsymbol{n}) = \frac{\mathbf{1}}{\boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n})} \qquad (9)$$
4. The filter tap weights are updated in preparation for the next iteration.
   $$\boldsymbol{w}(\boldsymbol{n}+\mathbf{1}) = \boldsymbol{w}(\boldsymbol{n}) + \frac{\mathbf{1}}{\boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n})}\boldsymbol{e}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n}) \qquad (10)$$

Each iteration of the NLMS algorithm requires 3N+1 multiplications, this is only N more than the standard LMS algorithm. This is an acceptable increase considering the gains in stability and noise attenuation achieved.

4.0 DATASET & APPROACH

1. An ECG dataset with 159 samples were obtained from the internet.

2. The clean ECG signal data is stored in a variable called ‘ecg’ which is of length 159*1, taken as a column vector.

3. The reference noise signal is named ‘no’ and the corrupted signal is called ‘ecgn’ both of which are column vectors of dimensions 159*1.

4. The NLMS algorithm is implemented using MATLAB. The ECG data which contains the original signal, noise signal and the corrupted ECG signal is imported to the workspace.

5. The initial values of weights(W_ini) and the reference input(X_ini) are taken as a random set of values using the randn() and rand() functions respectively for the given order of the filter(N) which in this case is fetched from the user.

6. The convergence of the algorithm is noted through an infinite loop which instantaneously updates the filter coefficients based on the formula mentioned in the previous section.

7. The code is iterated throughout the length of the reference input signal. The graphs of the clean ECG signal, corrupted ECG signal and the Denoised signal is plotted with respect to the number of iterations or the sample index.

8. The convergence parameter or the step size is ideally chosen to be a value between 0 and 1 ($0<\boldsymbol{\mu}<1$). The four cases corresponding to the convergence parameter as taken in the implementation are as follows:
   i. $0 < \boldsymbol{\mu} < \mathbf{1/\lambda m}$ where λm is the largest diagonal value of eigen value matrix of autocorrelation matrix of the reference input, x.
   ii. $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Sxm})$ where Sxm is the maximum of Power spectral density of autocorrelation matrix of the reference input, x. A small value less than 0.1 can be added to the denominator to avoid division by zero, when a zero-signal power occurs.
   iii. $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Px})$ where Px is the signal power of the reference input, x. A small value less than 0.1 can be added to the denominator to avoid division by zero, when a zero-signal power occurs.
   iv. $0 < \boldsymbol{\mu} < \mathbf{1}$

   It is seen that lower the step size (convergence parameter), better is the noise removal.

5.0 MATLAB CODE

Code for Implementing the NLMS Algorithm

```
function nlms2(org,x,d,N,mu,alpha,W_ini,X_ini)
 %org-original signal
 %x-reference input, here the reference to noise
 %d-desired or primary input, here the signal plus noise
 %N-no. of taps i.e., filter length or order

%mu-step-size or convergence parameter usually
%(i) 0<mu<1/lambdam where lambdam is the largest diagonal value of eignvalue matrix of
autocorrelation matrix of x or
%(ii) 0<mu<(1/N*Sxm) where Sxm-maximum of PSD of x or
%(iii) 0<mu<(1/N*Px),where Px-signal power of x or approximately
%(iv) 0<mu<1;
%lower the mu value, better the noise removal but slower the speed of
%convergence and VICE VERSA.
%Add a small positive value < 0.1 to
%denominator of (ii) and (iii) in order to avoid division-by-zero in case of zero signal power

%alpha-small positive real value approximately 0<alpha<1, closer to unity
 %W_ini-initial weight vector
 %X_ini-initial state vector i.e., initial values of reference input
 %W-final weight vector
 %e-error signal e=d-W*x, this is the signal recovered

Lx = length(x);
[m,n] = size(x);
if (n>m)
  x = x.';
end

if (~exist('Wini','var')||isempty(Wini))
  W = zeros(N,1);
else
  if (length(Wini)~=N)
    error('Weight initialization does not match filter length');
  end
  W = Wini;
end

if (~exist('Xini','var')||isempty(Xini))
  x = [zeros(N-1,1); x];
else
  if (length(Xini)~=(N-1))
    error('State initialization does not match filter length minus one');
```

```
   end
   x = [Xini; x];
end

n=1:Lx;
disp('Ctrl+C to terminate')
while (1)
    for k = 1:Lx
       X = x(k+N-1:-1:k);
       y = W'*X;
       e(k,1) = d(k,1) - y;
       p = alpha + X'*X;
       W = W + ((2*mu*e(k,1))/p)*X;
    end;

    plot(n,ol,'r',n,e,'g');
    title('Denoising ECG Signal using Adaptive NLMS Algorithm');
    xlabel('Number of Samples');
    ylabel('Error');
    legend('Original ECG Signal','Error Signal(Recovered)');
    pause(1);

end;
```

Main function

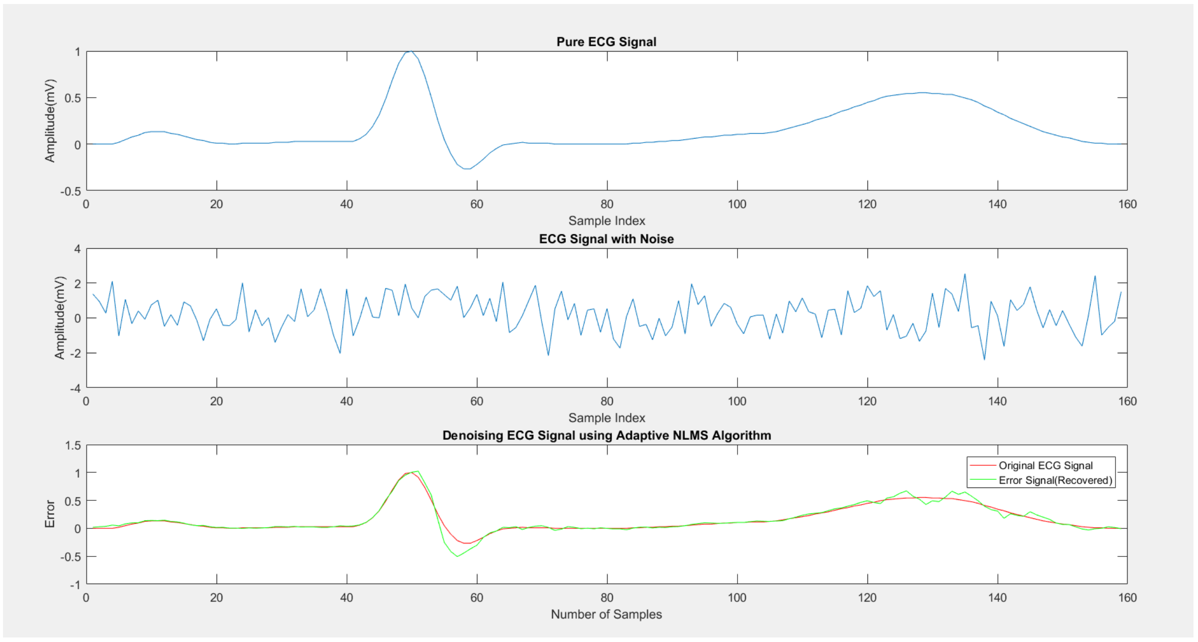
```
clear all;
clc;
%ecg: Clean ECG signal
%no: The reference noise
%ecgn: Noised ECG Signal
load('ecg.mat');

%Initializing variables for function call
org=ecg;%org-original signal
x=no;%x-reference input, here the reference to noise
d=ecgn;%d-desired or primary input, here the signal plus noise
N=input('Enter the filter order: ');%N-no. of taps i.e., filter length or order
figure; subplot(3,1,1); plot(ecg);
    title('Pure ECG Signal');
    xlabel('Sample Index');
    ylabel('Amplitude(mV)');

subplot(3,1,2); plot(ecgn);
    title('ECG Signal with Noise');
    xlabel('Sample Index');
    ylabel('Amplitude(mV)');
```

```
subplot(3,1,3);
%mu-step-size or convergence parameter usually
%(i) 0<mu<1/lambdam where lambdam is the largest diagonal value of eignvalue matrix of
autocorrelation matrix of x or
autocorr_x=corrmtx(x,N,'autocorrelation');
ac_x=toeplitz(autocorr_x);
eig_x=eig(ac_x);
lambda_m1=max(eig_x);
mu1=1/lambda_m1;

%(ii) 0<mu<(1/N*Sxm) where Sxm-maximum of PSD of x or
p_x=periodogram(x);
Sxm=max(p_x);
mu2=1/((N*(Sxm+0.1)));

%(iii) 0<mu<(1/N*Px),where Px-signal power of x or approximately
Px=norm(x)^2/length(x);
mu3=1/((N*(Px+0.1)));

%(iv) 0<mu<1;
mu4=0.01;

%lower the mu value, better the noise removal but slower the speed of
%convergence and VICE VERSA.
%Add a small positive value < 0.1 to
%denominator of (ii) and (iii) in order to avoid division-by-zero in case of zero signal power

alpha=0.9;%alpha-small positive real value approximately 0<alpha<1, closer to unity

W_ini=randn(N,1);%W_ini-initial weight vector
X_ini=rand(N-1,1);%X_ini-initial state vector i.e., initial values of reference input

mu=input('Enter a number, either 1,2,3 or 4: ');
switch mu
   case 1
      nlms2(org,x,d,N,mu1,alpha,W_ini,X_ini)
   case 2
      nlms2(org,x,d,N,mu2,alpha,W_ini,X_ini)
   case 3
      nlms2(org,x,d,N,mu3,alpha,W_ini,X_ini)
   case 4
      nlms2(org,x,d,N,mu4,alpha,W_ini,X_ini)
   otherwise
      disp('Invalid Input')
end
```

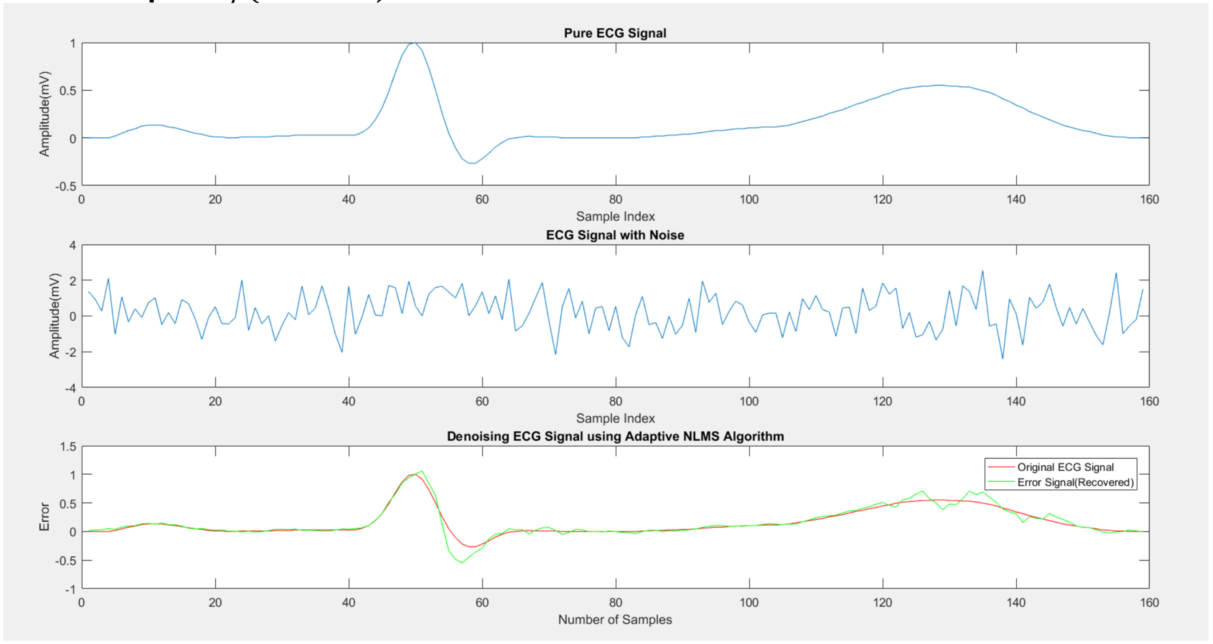
6.0 RESULT & DISCUSSION

The simulated results for the NLMS algorithm based adaptive filter are as shown below. Since the convergence is instantaneously run on an infinite loop, the final convergence of the denoised(recovered) signal with respect to the rec. The values of N chosen for this experiment are 4,8,12 and 32 for four different cases of $\boldsymbol{\mu}$.

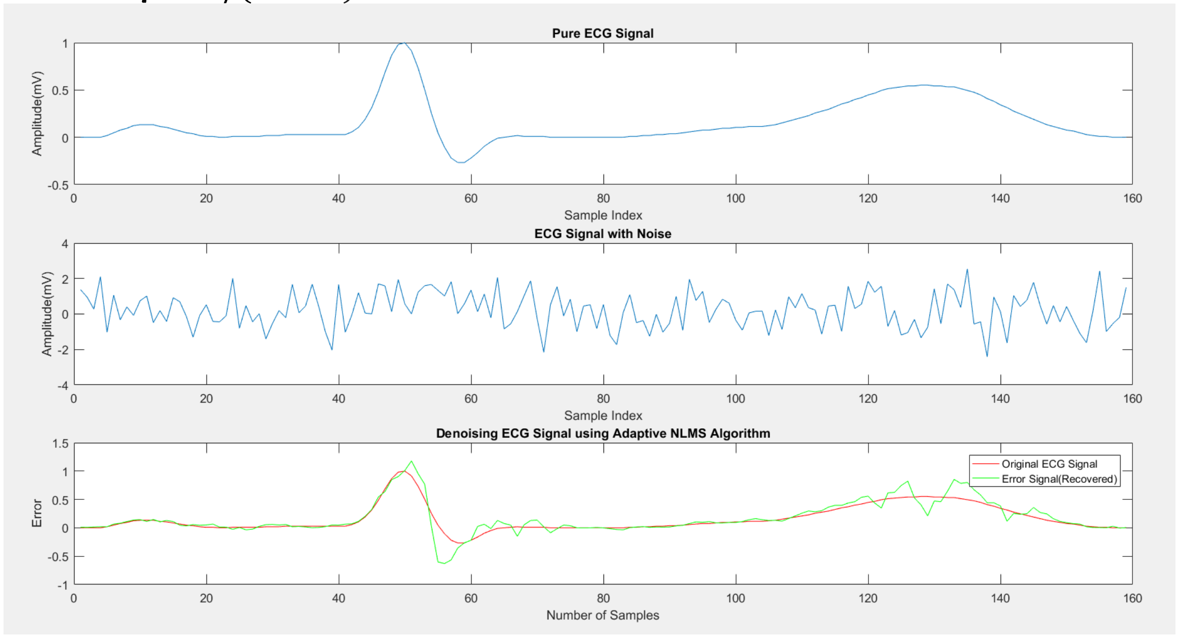
Filter order N=4
Case 1: $0 < \boldsymbol{\mu} < \mathbf{1}/\lambda\mathrm{m}$



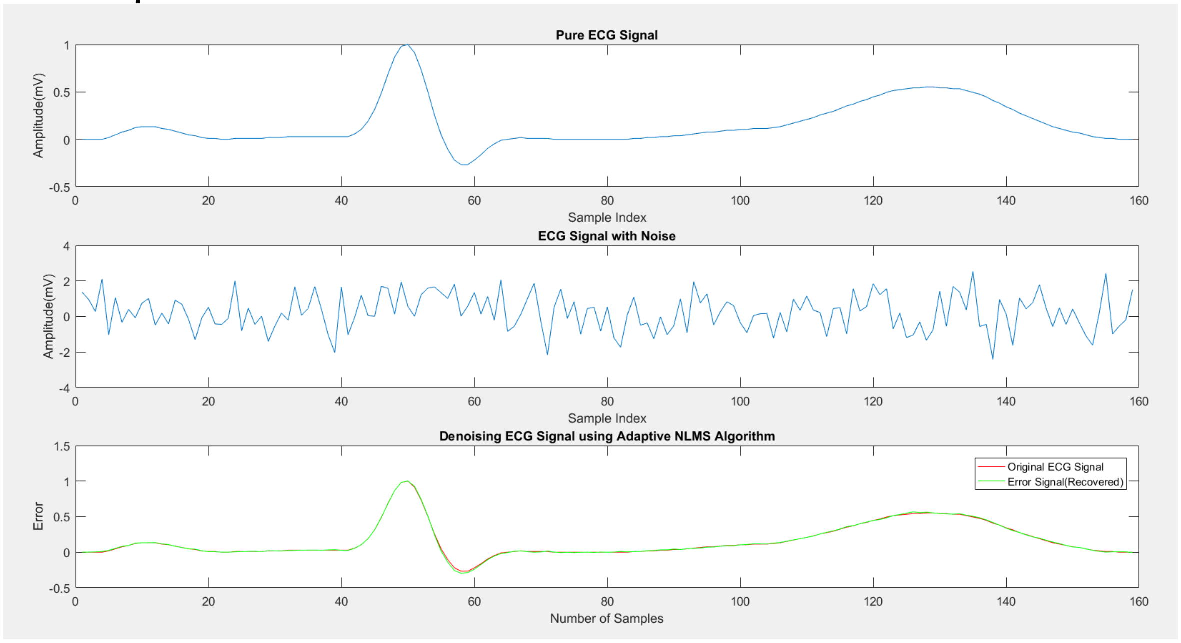
Case 2: $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Sxm})$



Case 3: $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Px})$

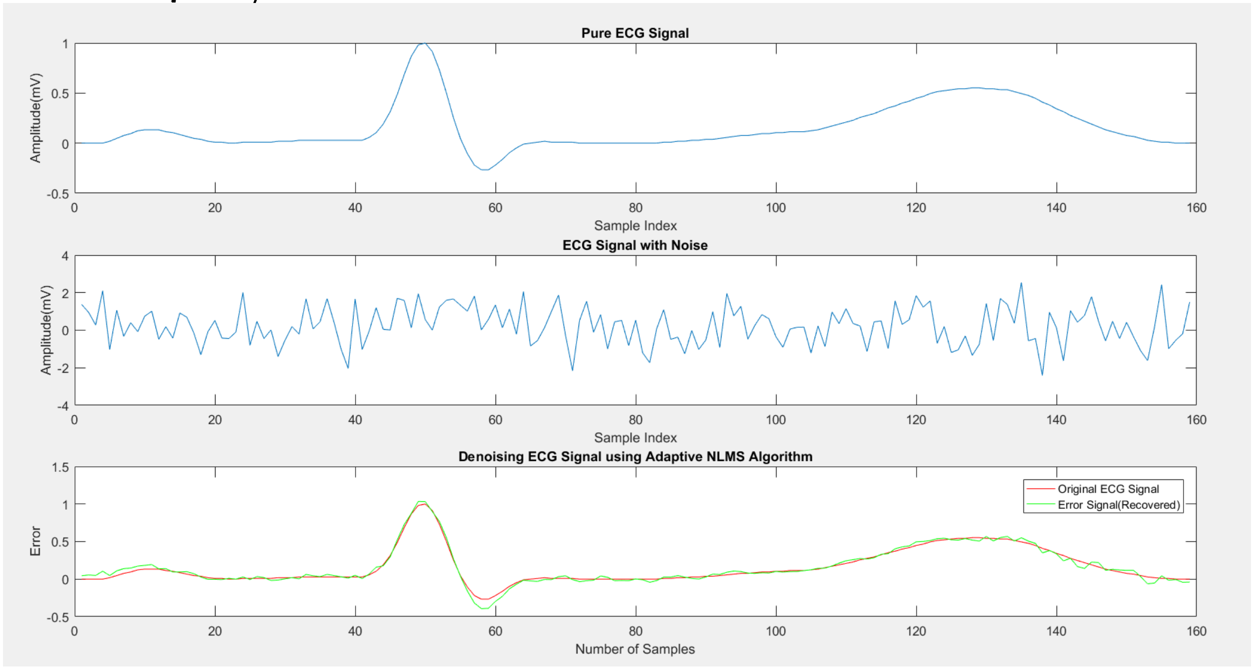


Case 4: $0 < \boldsymbol{\mu} < \mathbf{1}$

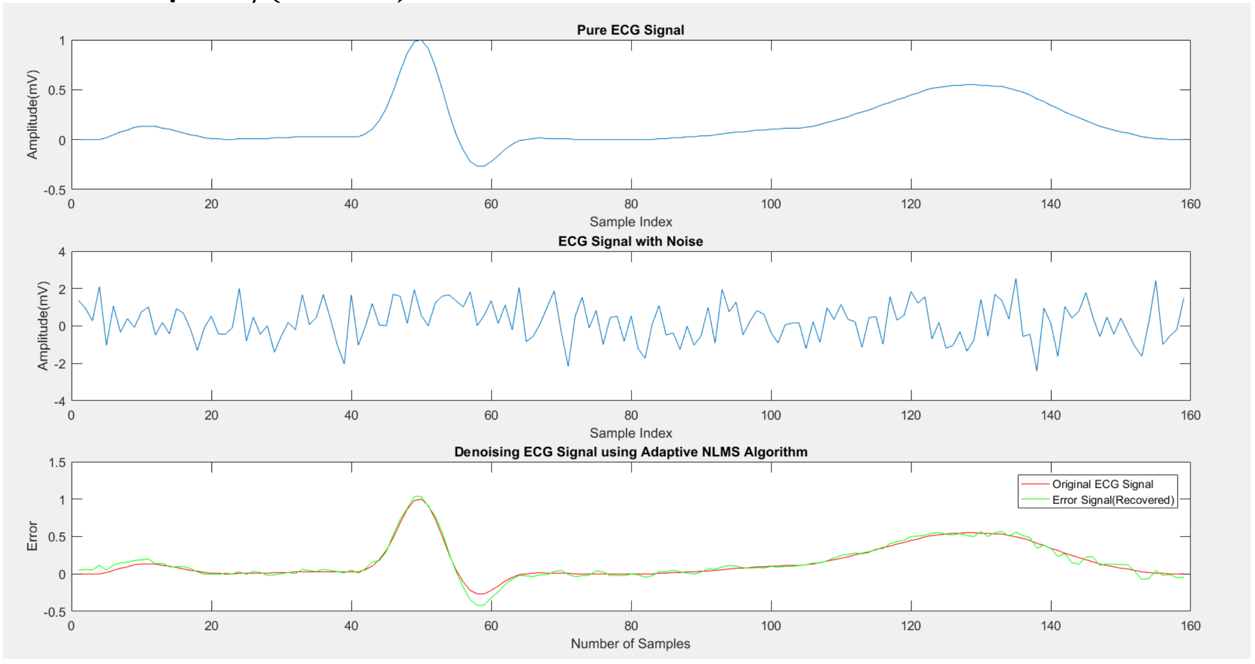


Filter Order N=8

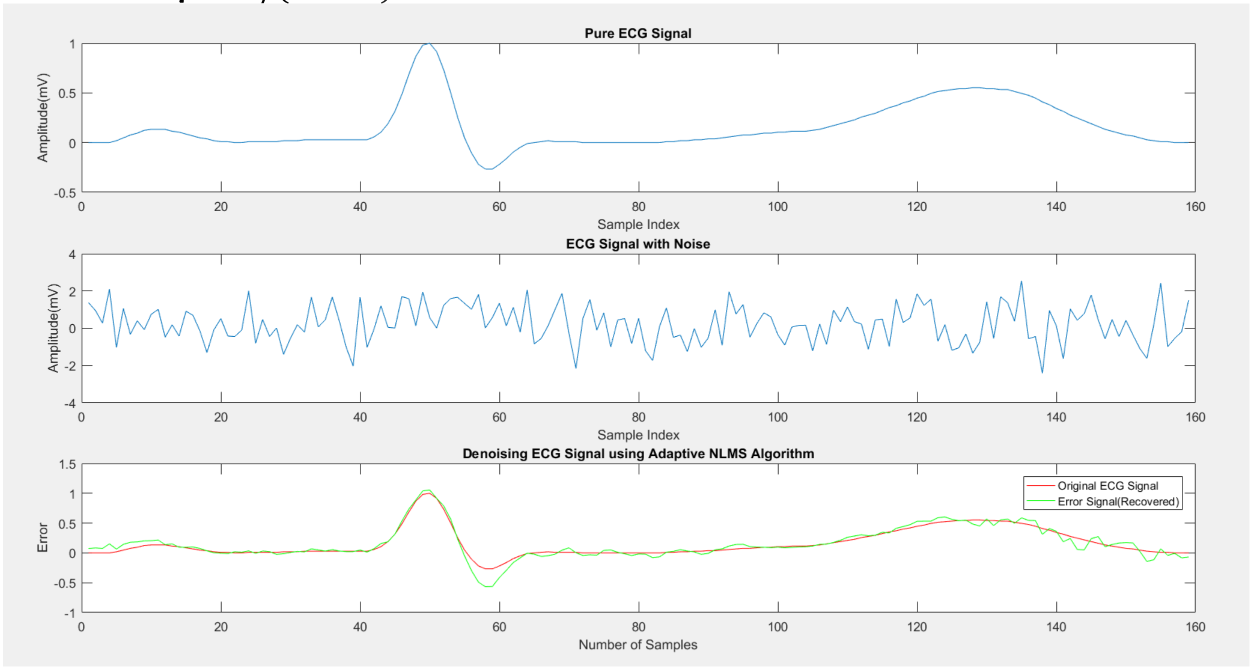
Case 1: $0 < \mu < 1/\lambda m$



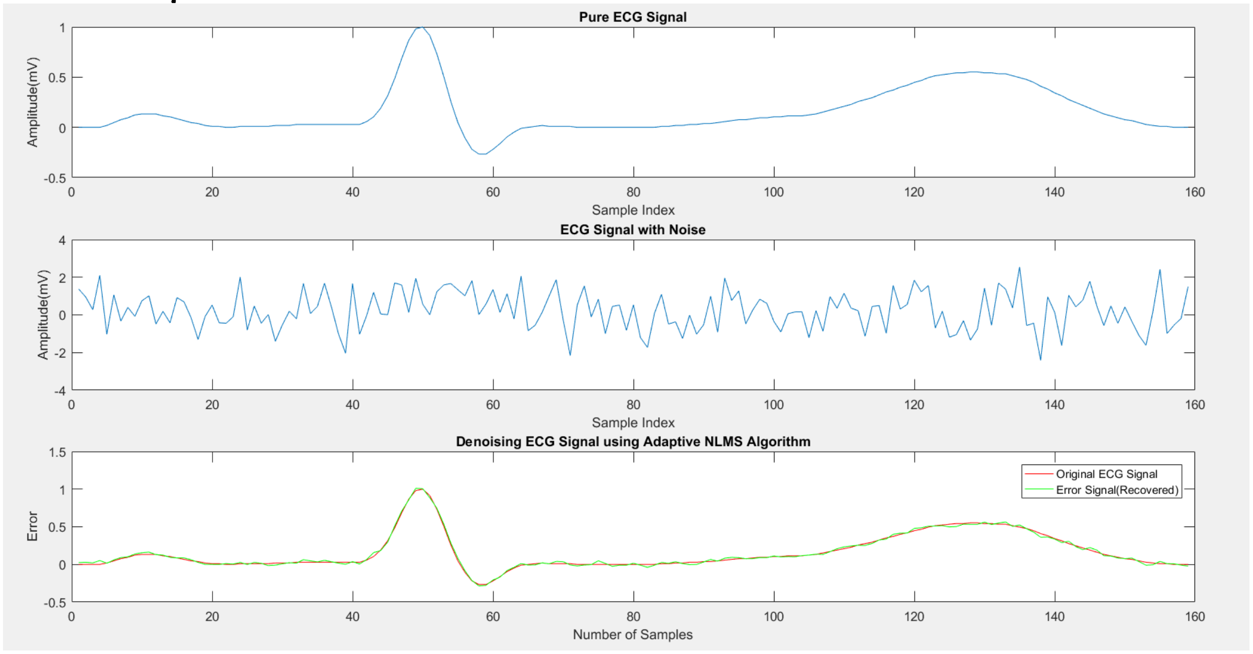
Case 2: $0 < \mu < 1/(N * Sxm)$



Case 3: $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Px})$

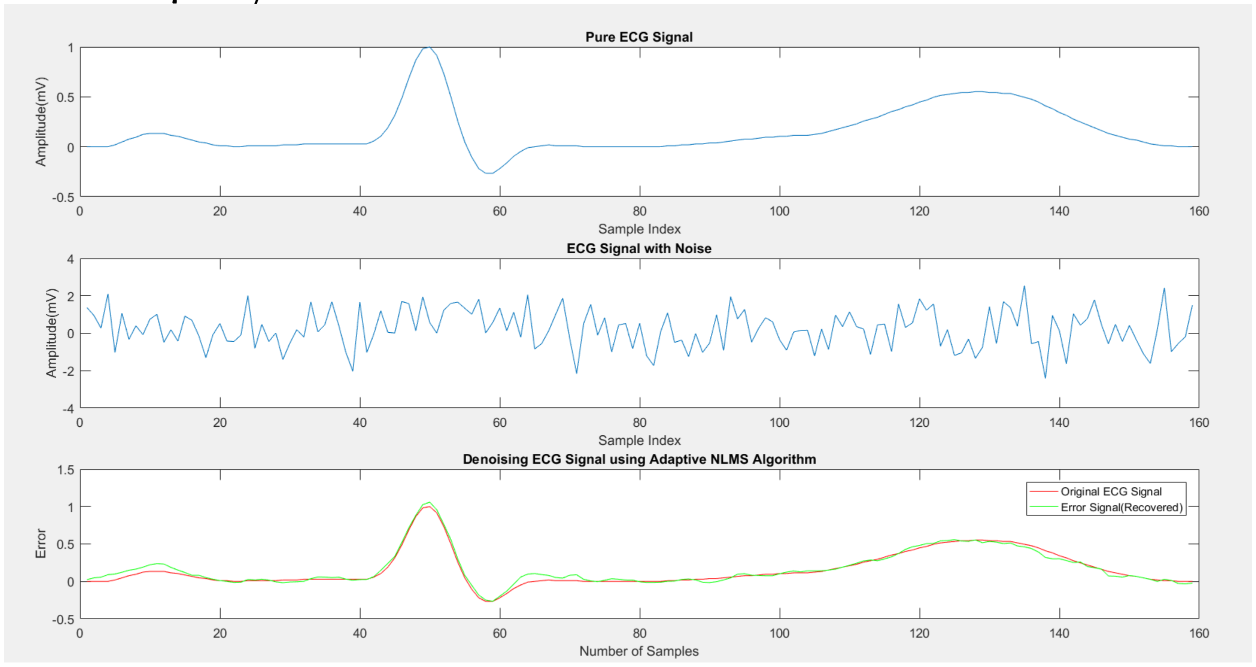


Case 4: $0 < \boldsymbol{\mu} < \mathbf{1}$

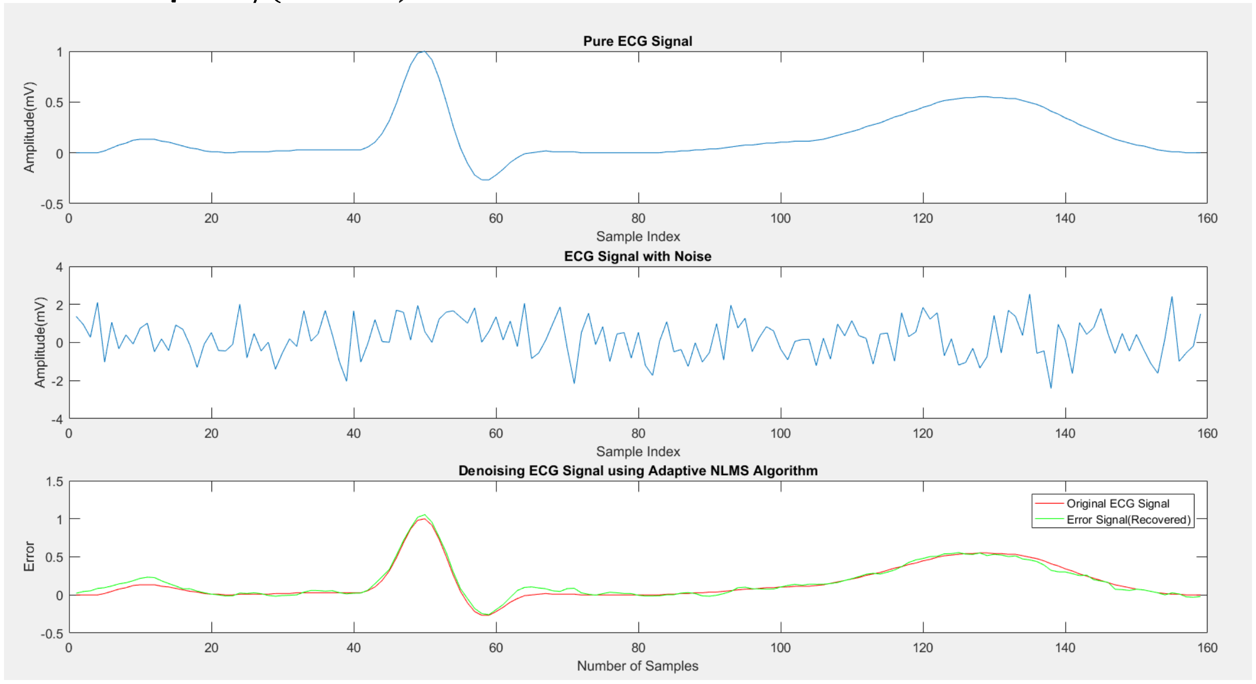


Filter Order N=12

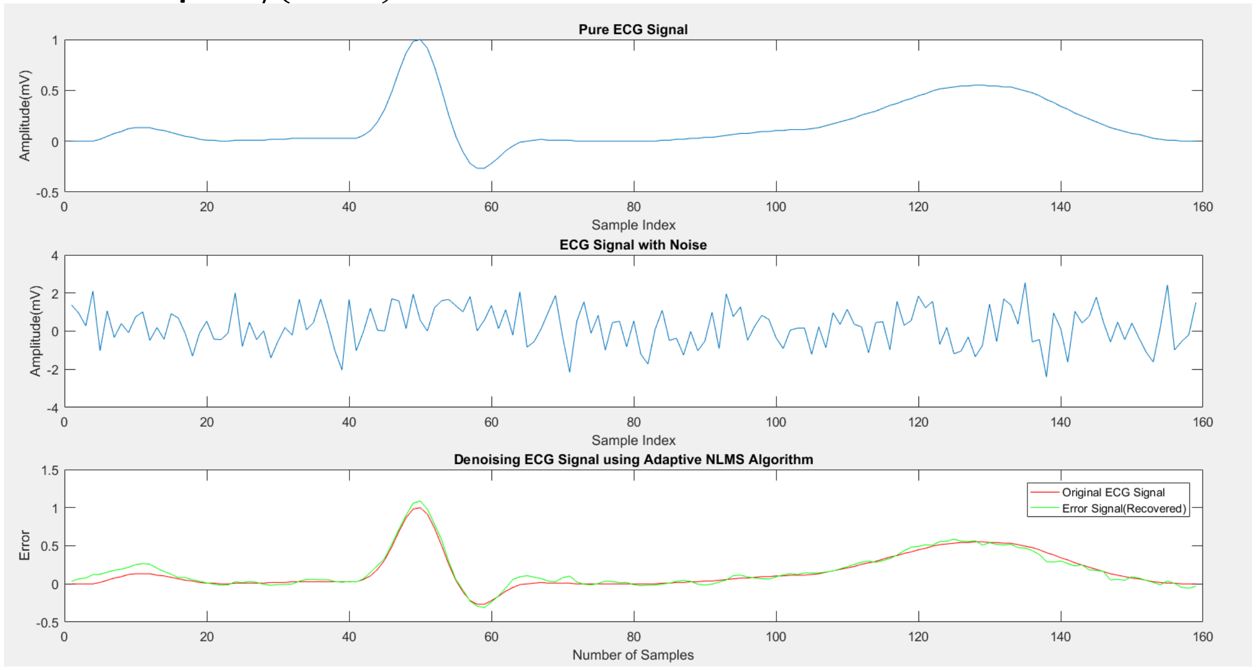
Case 1: $0 < \boldsymbol{\mu} < \mathbf{1}/\lambda\mathbf{m}$



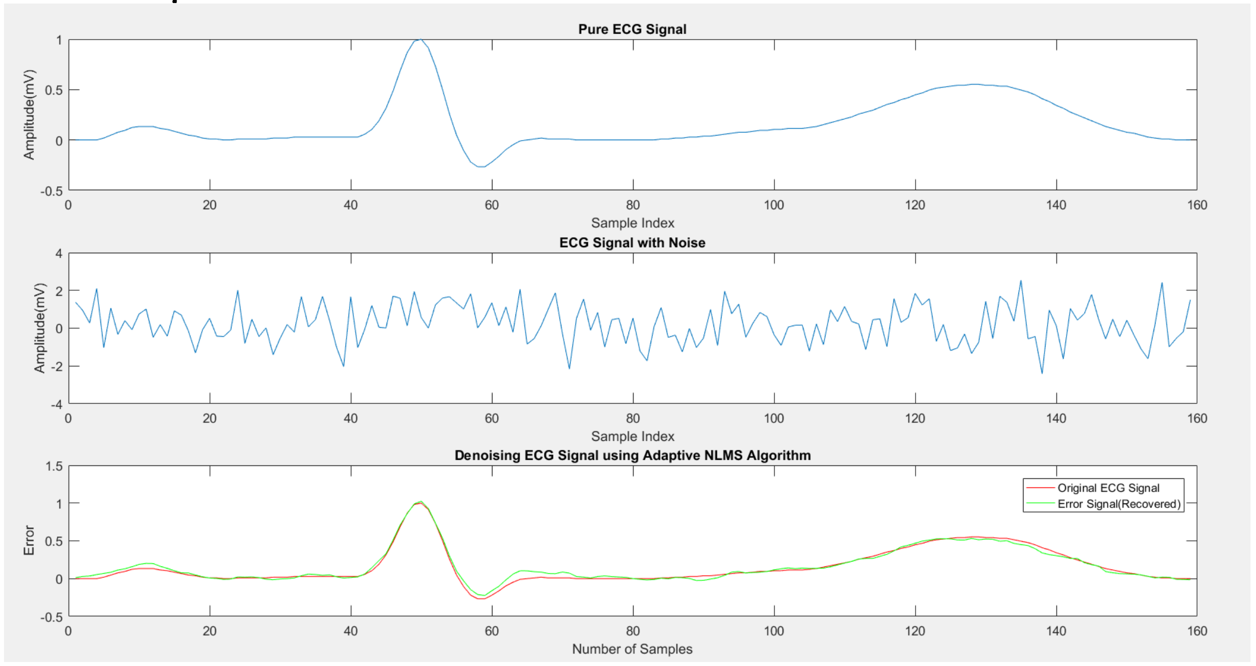
Case 2: $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Sxm})$



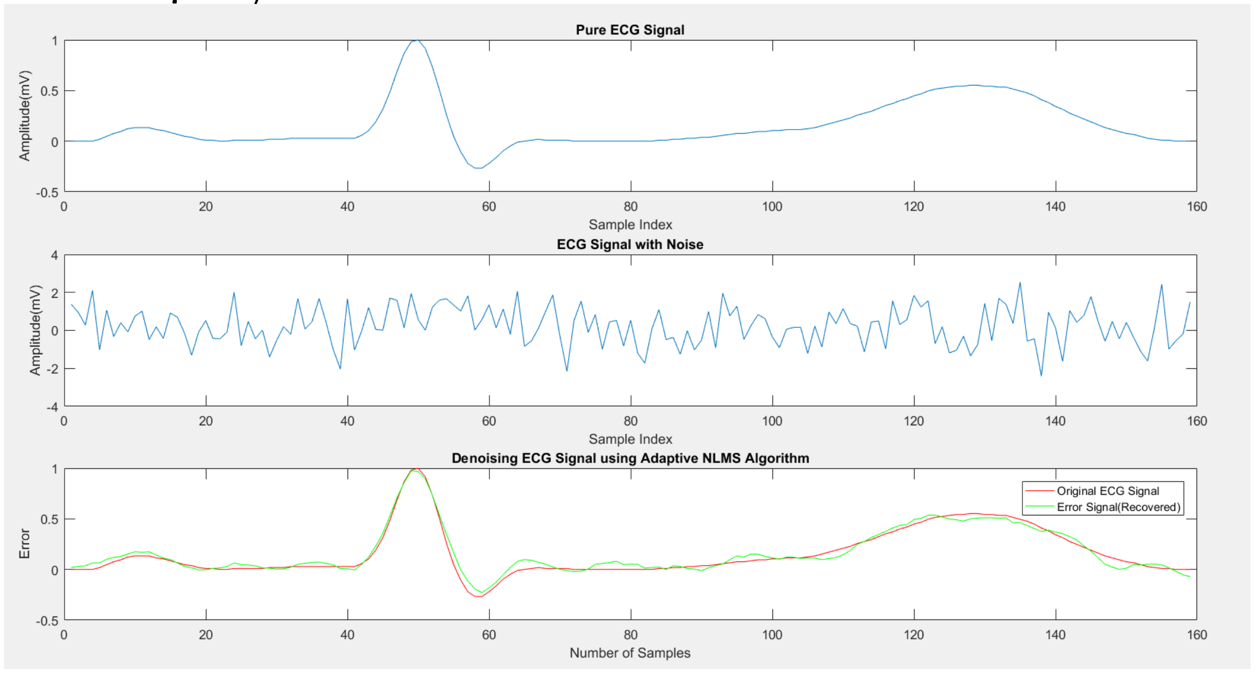
Case 3: $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Px})$



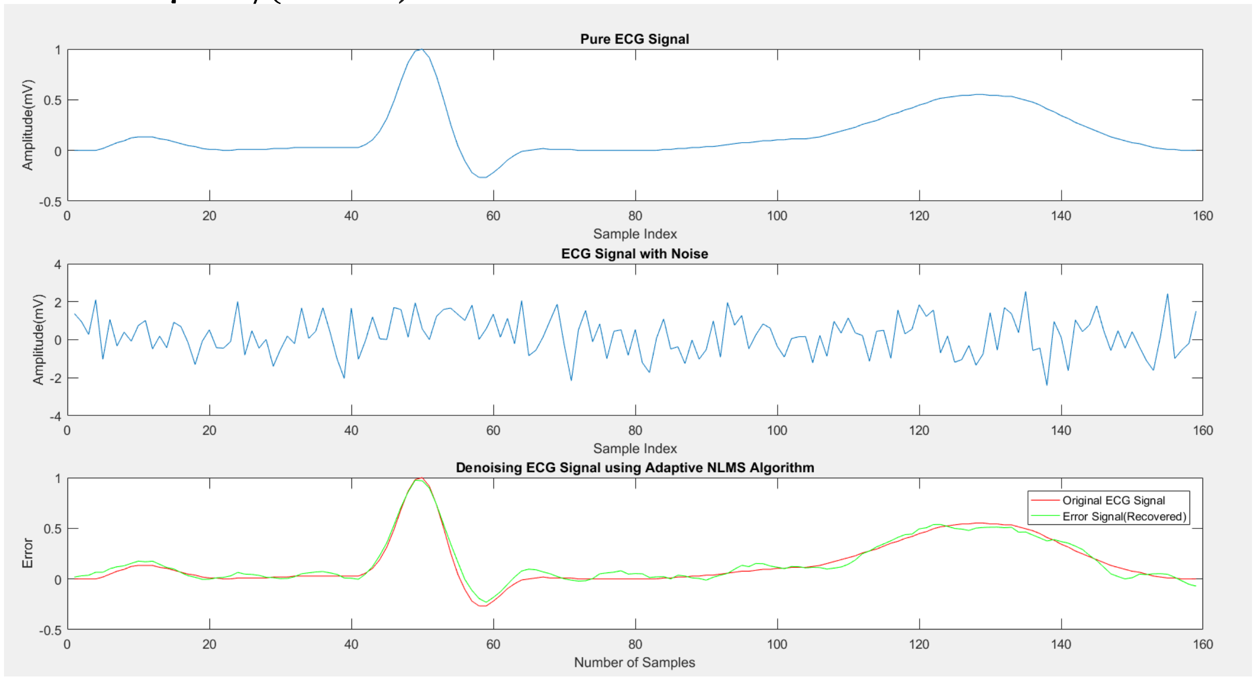
Case 4: $0 < \boldsymbol{\mu} < \mathbf{1}$



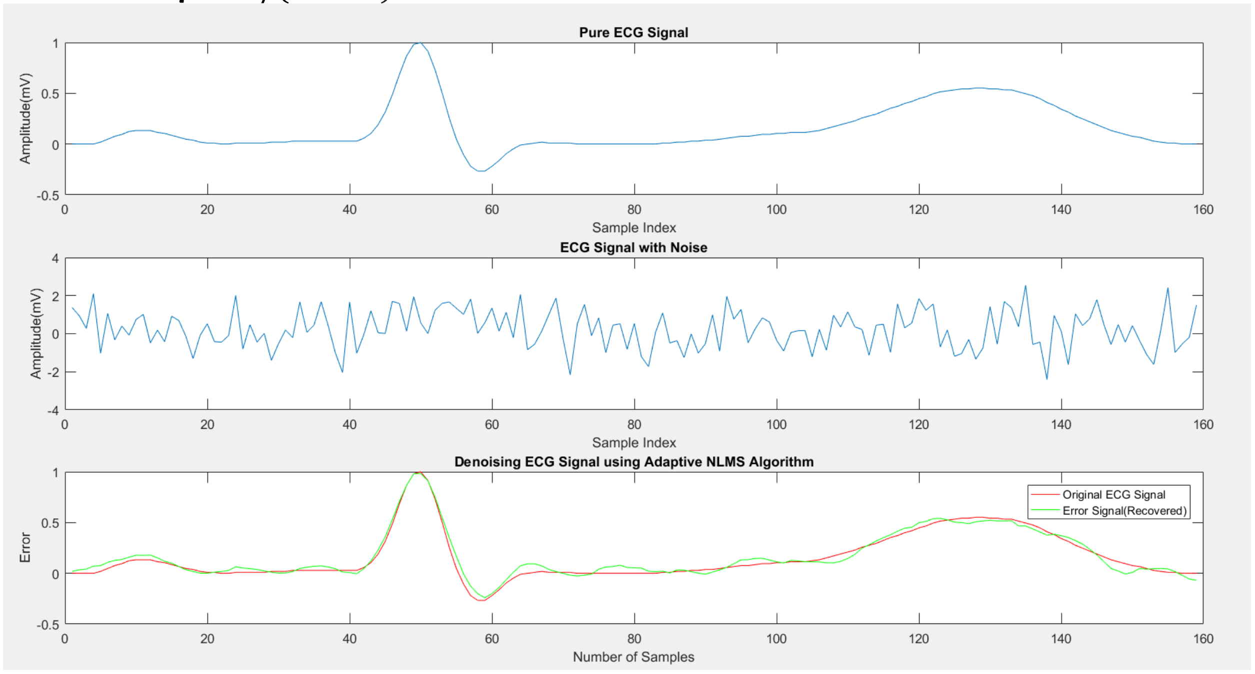
Filter order N=32
Case 1: $0 < \boldsymbol{\mu} < \mathbf{1}/\boldsymbol{\lambda}\mathbf{m}$



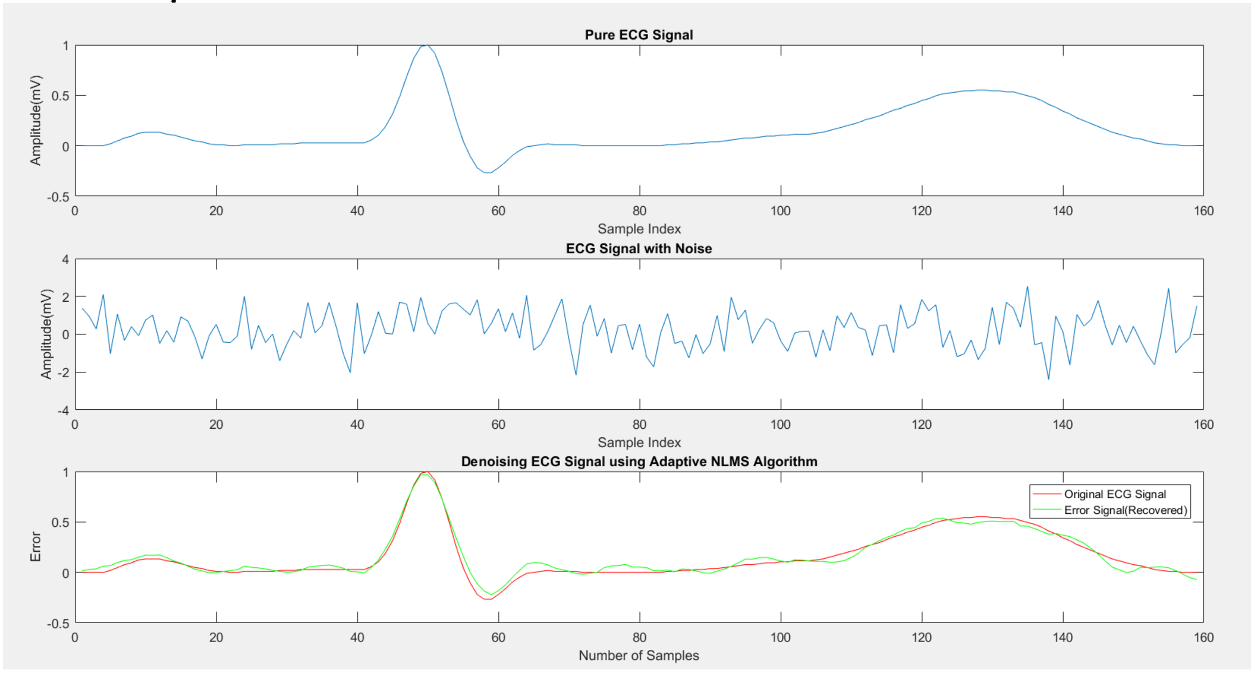
Case 2: $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Sxm})$



Case 3: $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Px})$



Case 4: $0 < \boldsymbol{\mu} < \mathbf{1}$



The simulated NLMS algorithm shows that irrespective of the filter order, the convergence parameter decides the rate of convergence of the recovered ECG signal with respect to the original signal. In the first three cases, the convergence was quick, however, the least value among all was chosen in case 4($\boldsymbol{\mu}$ =0.01) which showed the best convergence but at a decreased rate. The changes in the convergence pattern for each case can observed at higher filter orders, in this case, we have chosen N=32.

7.0 CONCLUSION

A 160 sample ECG signal with an amplitude of 1mV was used to implement the Normalized Least Mean Square algorithm based Adaptive filter. Another dataset of the same size assuming random values with zero mean and unit variance was taken to be the noise. The noise is then added to the clean ECG signal to get the desired mixed signal. Finally, the noise was removed using the Normalized Least Mean squares algorithm. In the above-mentioned formulation, we observed the performance of the adaptive NLMS filter with respect to the convergence parameter and the filter order. The simulated NLMS algorithm showed that irrespective of the filter order, the convergence parameter decides the rate of convergence of the recovered ECG signal with respect to the original signal. In the first three cases ($0 < \boldsymbol{\mu} < \mathbf{1}/\boldsymbol{\lambda}\mathbf{m}$, $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Sxm}),$ $0 < \boldsymbol{\mu} < \mathbf{1}/(\boldsymbol{N} * \mathbf{Px})$) **,**the convergence was quick, however, the least value among all was chosen in case 4($\boldsymbol{\mu}$ =0.01) which showed the best convergence but at a decreased rate. The changes in the convergence pattern for each case can observed at higher filter orders, in this case, we have chosen N=32. Therefore we can conclude that a smaller step size removes noise efficiently but reduces the speed of convergence and vice versa. The NLMS algorithm being a more robust version of the LMS algorithms exhibits a better balance between simplicity and performance in real time applications.

8.0 REFERENCES

1. Simon Haykin: Adaptive Filter Theory, Prentice Hall, 2002, ISBN 0-13-048434-2.
2. Monson H. Hayes: Statistical Digital Signal Processing and Modeling, Wiley, 1996, ISBN 0-471-59431-8.
3. ECG Signal Denoising by Using Least Mean Square and Normalized Least Mean Square Algorithm based Adaptive Filter-Uzzal Biswas, Anup Das, Saurov Debnath and Isabela Oishee-3rd INTERNATIONAL CONFERENCE ON INFORMATICS, ELECTRONICS & VISION 2014
4. C. Chandrakar and M.K. Kowar,"Denoising ECG signals using Adaptive Filter Algorithm," Int. J. of Soft Computing and Engineering (IJSCE), vol. 2, no. 1, pp. 120-123, March 2012.
5. A. B. Sankar, D. Kumar and K. Seethalakshmi, "Performance Study of Various Adaptive Filter Algorithms for Noise Cancellation in Respiratory Signals," An International Journal(SPIJ), vol. 4, no. 5, pp. 267-278, December 2010.
6. Homana, I.; Topa, M.D.; Kirei, B.S.; "Echo cancelling using adaptive algorithms", Design and Technology of Electronics Packages, (SIITME) 15th International Symposium., pp. 317-321, Sept.2009.
7. Paleologu, C.; Benesty, J.; Grant, S.L.; Osterwise, C.; "Variable step-size NLMS algorithms for echo cancellation" 2009 Conference Record of the forty-third Asilomar Conference on Signals, Systems and Computers., pp. 633-637, Nov 2009.
8. Soria, E.; Calpe, J.; Chambers, J.; Martinez, M.; Camps, G.; Guerrero, J.D.M.; "A novel approach to introducing adaptive filters based on the LMS algorithm and its variants", IEEE Transactions, vol. 47, pp. 127-133, Feb 2008.
9. Tandon, A.; Ahmad, M.O.; Swamy, M.N.S.; "An efficient, low-complexity, normalized LMS algorithm for echo cancellation", IEEE workshop on Circuits and Systems, 2004. NEWCAS 2004, pp. 161-164, June 2004.