CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

ECG NOISE CANCELLATION
USING KERNEL ADAPTIVE FILTERING

A graduate project submitted in partial fulfillment of the requirements

For the degree of Masters of Science in

Electrical Engineering

The graduate project of is approved:

_____	_____
Deborah van Alphen, Ph.D.	Date
_____	_____
Prof. Benjamin F. Mallard	Date
_____	_____
Xiyi Hang, Ph.D. Chair.	Date

California State University, Northridge

Table of Contents

List of Figures

Abstract

ECG NOISE CANCELLATION

USING KERNEL ADAPTIVE FILTERING

Master of Science in Electrical Engineering

In this paper, several adaptive filter algorithms are proposed for noise cancellation of ECG signals and determining the accuracy of ECG signal features. Adaptive filters function based on minimizing the error between input signal which is a noisy ECG signal and its reference input, which is one of many correlated artifacts present in the ECG signal. These artifacts are 60Hz-power line interference, baseline wander, motion artifact and muscle artifact (EMG). Different adaptive filter algorithms used in this project are Least Mean Squares (LMS), Normalized Least Mean Squares (NLMS), Kernel Least Mean Squares (KLMS) and Normalized Kernel Least Mean Squares (NKLMS). The main focus of this paper is to compare the performance of the new kernel adaptive filter to already existing least mean squares filter. The adaptive filter is based on kernel tricks presented in mathematics. For results, the clean ECG signal was compared to the filtered ECG signal using one of above algorithms and the accuracy of detecting peak locations of P, Q, R, S and T waves were evaluated.

Section 1: Introduction

Electrocardiography (ECG) is a reading of electrical activity of the heart. It is detected by several electrodes attached to the body surface and recorded using an external device. It is used as a test to gather information about heart diseases, surgical procedures, etc. In simpler terms, an ECG is to a human body what a computer troubleshoot message is to a personal computer; in the same way that today's personal computers can now detect hardware malfunction from a system, an ECG can also be used to detect heart diseases. An ECG signal on external device displays as shown in Figure 1.1 below.

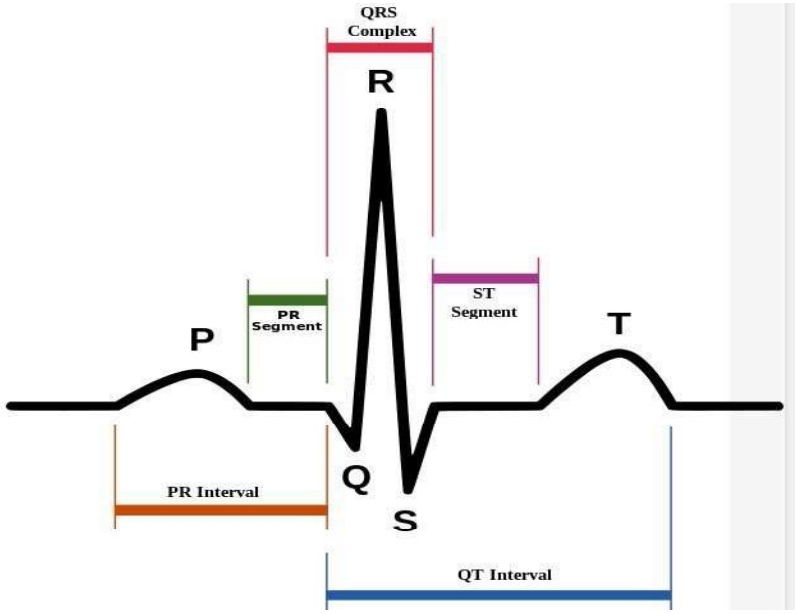

Figure 1.1 ECG signal and its Features [3]

A typical ECG consists of a P-wave, a QRS complex and a T-wave. On the average, P-wave has a duration of 80 milliseconds (ms), and it occurs during normal atrial depolarization. The QRS complex occurs after a P-wave and it replicates the rapid depolarization of the right and left ventricles. Due to the fact that ventricles have larger

muscle mass than atria, QRS complex amplitudes are generally higher than P-wave amplitudes and have average duration of 80ms to 120ms. T-wave represents repolarization of ventricles of the heart and has average duration of 160ms.

The human heart consists of four chambers: the right and left atriums and the right and left ventricles. Atrium is a receiving chamber that pumps blood into the heart while ventricle is a discharging chamber that pumps blood out of the heart. Figure 1.2 below shows the human heart and its atrium and ventricle locations.

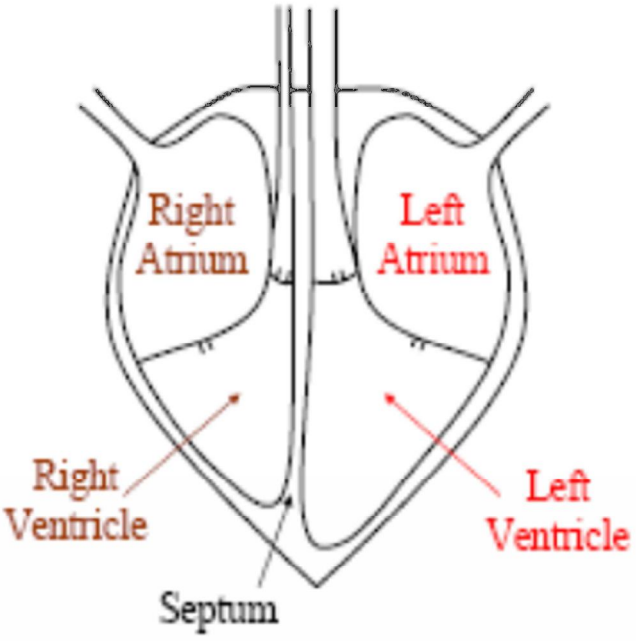

Figure 1.2 The Human Heart and its Atrium and Ventricle Locations

A recorded ECG signal is not noise-free. Noise existing in ECG signals can be removed using analog devices such as electronic circuits or using digital filters. As computational microprocessor power increases, the need to remove noise has been shifted to the use of digital filters. One of the very effective digital filters to remove ECG noise has been

adaptive filtering techniques. It is very simple to implement and works rather well with the ECG signal due to their nonlinear nature.

This paper demonstrates different adaptive filtering structures. This paper will also attempt to explain different noise sources in ECG signals, how they are caused and how they can be canceled. This paper also demonstrates usage of different adaptive filtering techniques to cancel noise sources from an ECG signal and to accurately detect its characteristics.

Section 2: Artifacts

ECG noise is contributed by environmental and biological sources. Examples of environmental noise are 60Hz power interference noise and instrumentation noise generated using hardware. Examples of biological noise are baseline wander, electromyogram (EMG) and motion artifact. In this report, the causes of each noise source will be explained as well as methods to eliminate the noise.

2.1 Instrumentation Noise

This is one of the environmental noise sources. This noise is generated by hardware that is recording the ECG signal. Since this noise can be removed using Low Noise Amplifiers (LNAs), no digital filtering is needed.

2.2 60-Hz Power Line Interference

Alternating current (AC) is the type of electricity that we get from the wall. It changes direction 60 times per second, hence 60Hz. It is an environmental noise in the ECG that results from poorly grounded ECG recording machine. Because of the alternating current feature, this noise appears at 60Hz and its harmonics; meaning if the sampling rate is 360Hz, this noise appears at 60Hz, 120Hz, 180Hz, etc. For simplicity in this project, it was assumed that power-line interference only appears at 60Hz. For removal, Type I adaptive filter is used to subtract 60Hz sinusoidal from ECG signal. Figure 2.1 below demonstrates this interference.

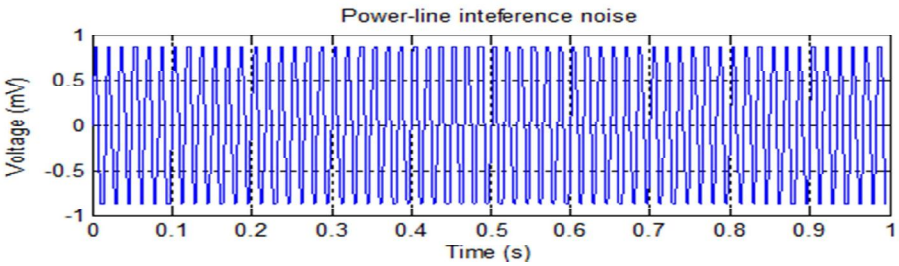

Figure 2.1 PLI noise. Number of Samples = 360.

2.3 Baseline Wander

Baseline drift is a biological noise that appears in ECG signal. In this noise, isoelectric line changes position which makes the ECG looks like its "wandering". It is caused by moving cables during recording, patient movement, loose electrodes, etc. Van Alste and Scheilder [17] describe an efficient finite impulse response (FIR) notch filter that is effective in removing baseline wander and power interference noise from ECG. The adaptive filter to remove this noise is a special case of notch filtering, with notch at zero frequency to remove 0-0.5 Hz frequencies. For removal, Type I filter is used with reference noise being constant at value 1. The convergence parameter should be smaller than 0.006 to remove baseline drift. Demonstration of this noise is shown in the Figure 2.2 below.

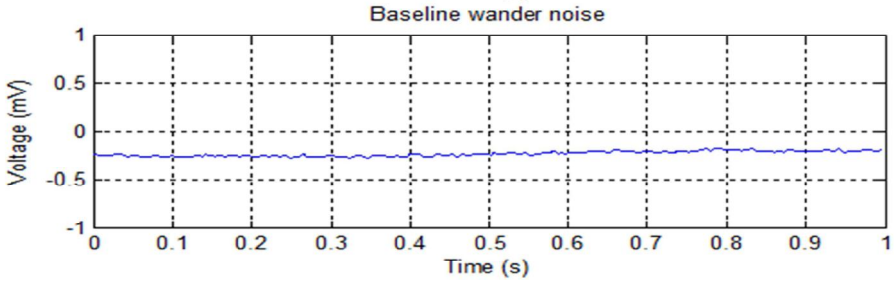

Figure 2.2 Baseline Wander Artifact. Number of Samples = 360.

2.4 Muscle Artifact

Muscle artifact or electromyography (EMG) is a biological noise that can appear in the ECG signal. EMG measures muscle response to a nerve's stimulation of the muscle; it is an electrical potential generated by muscle cells when the cells are neurologically or electrically activated. In the ECG, muscle artifact can be produced by unwanted reaction to electrodes. This explains why they ask patients to remain calm during the ECG recordings to avoid unwanted noise. To reduce this noise, typically more than one ECG lead is used while recording. Type I adaptive filter applied for motion artifact will also remove EMG from signal. Demonstration of this artifact is shown in Figure 2.3 below.

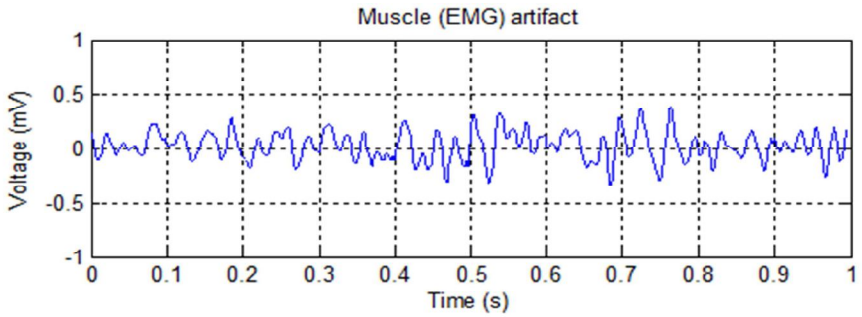

Figure 2.3 EMG Artifact. Number of Samples = 360.

2.5 Motion Artifact

Motion artifact is the most difficult biological noise to cancel, which can appear in the ECG signal. The spectrum of this noise is broad and it completely overlaps the ECG's spectrum. Most linear filtering methods are unable to remove this noise source. Adaptive filter can eliminate this noise source by having the adaptive filter reference input set to an impulse value of 1 that corresponds to the beginning of the P-wave. This way, adaptive filter only subtracts the P-QRS-T complex from the signal and the

remainder will be motion artifact. Adaptive filter Type I is used to subtract this noise from the ECG. Demonstration of motion artifact is shown in Figure 2.4 below.

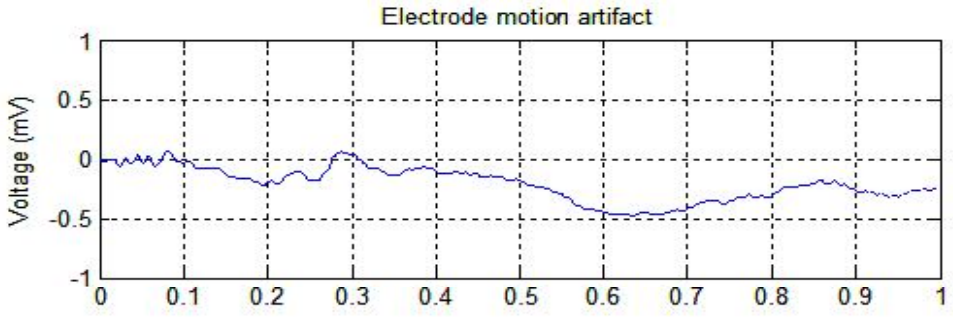

Figure 2.4 Motion Artifact. Number of Samples = 360.

2.6 Noisy ECG

A noisy ECG typically includes one or more artifacts described in previous subsections. The noisy ECG used in this project consists of PLI, Baseline Wander, Motion artifact and Muscle artifact added to a clean ECG signal. A one second segment of noisy ECG is shown below in Figure 2.5.

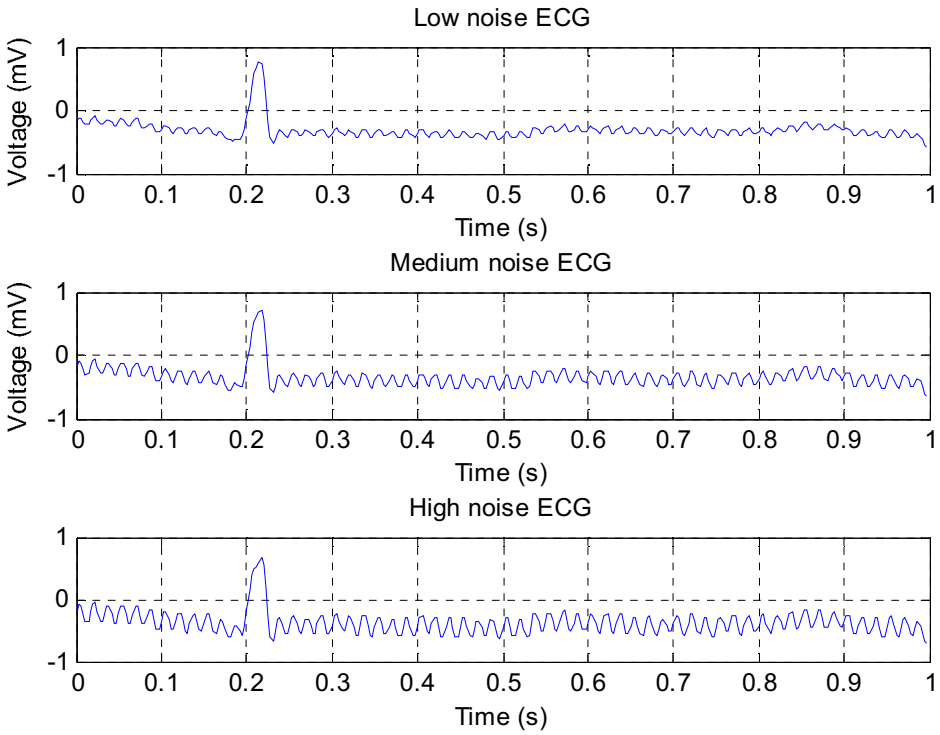

Figure 2.5 Different ECG Noise. Number of Samples = 360.

As can be seen in Figure 2.5, 3 different levels of noisy ECG were used to be filtered.

Section 3: Adaptive Filters

Adaptive filters have 2 basic filter structures. One of the adaptive filter structures is shown in Figure 3.1 below,

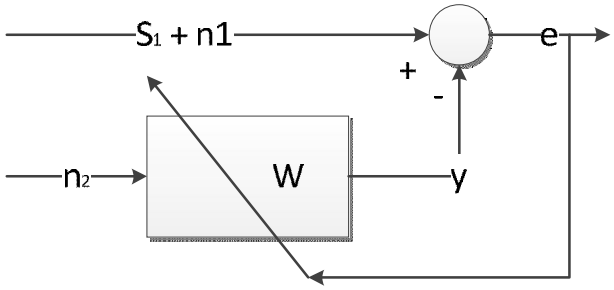

Figure 3.1 Adaptive Filter Structure 1 (Type I)

In this filter structure, primary input is signal s_1 with additive noise n_1 and reference input is noise n_2, which can be any baseline wander, power-interference noise, etc. and it is correlated with n_1. Both noise sources n_1 and n_2 should not be correlated with the signal s_1. Filter output is y and filter error is $e = (s_1 + n_1) - y$. W is the weight function that is updated every iteration and the basic structure for it is,

(1)

The other basic adaptive filter stricture is shown in Figure 3.2 below.

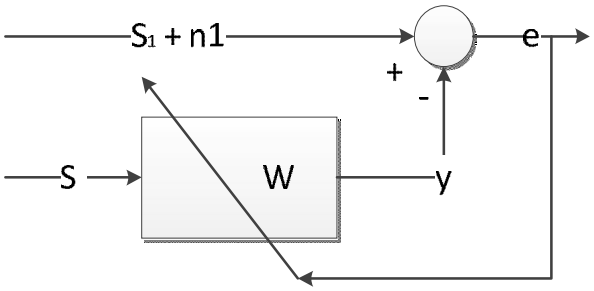

Figure 3.2 Adaptive Filter Structure 2 (Type II)

Unlike filter structure 1, this structure has s_2 as its reference signal. In this filter structure, primary input is signal s_1 with additive noise n_1 and reference input is noise s_2, which can be the clean ECG signal and it is correlated with s_1. Because of non-linear nature of the ECG signal, this adaptive filter structure cannot be used to remove noise sources from the ECG signal.

3.1 Least Mean Squares (LMS) Algorithm

LMS algorithm is a linear adaptive filter algorithm invented by B. Widrow and T. Hoff. This filter works based on finding filter coefficients that leads to minimizing the mean square of the error which is the difference between the desired signal and error signal. Its stochastic gradient descent method adapts to instantaneous error.

In order for LMS filter to approach the optimum filter weights, the algorithm starts by assuming small weights (zero) and at each step, finds the gradient of the mean square error and then updates the weights. If mean square error is positive, error is increasing positively, and if same error is used, filter weights need to be reduced accordingly. And vice versa, if gradient is negative, filter weights need to be increased. Weight update equation is shown below,

$$w(i) = w(i-1) + \mu\, e(i)\, x(i)\,, \tag{2}$$

where $e(i), x(i), \mu$ *and* $w(i)$ are the error, the input, step-size parameter, and weight function respectively.

Under the assumption that the step-size parameter μ is significantly small, LMS algorithm converges in the mean square provided that u satisfies condition below,

$$0 < u < \frac{1}{\lambda_{max}}\ [2], \tag{3}$$

where λ is largest eigenvalue of the correlation matrix.

LMS algorithm can be summarized as shown in Figure 3.3 below.

Inputs:
- *x = input signal*
- *dn = desired signal*
- *M = filter length*
- *μ = step-size factor*

Outputs
- *y = output of filter*
- *e = error signal*

Pseudo-code
- *Initialize:*
  - *filter coefficient w(1) = 0*
- *Loop1:*
  - *For every i, do*
    - *Calculate e(i) = d(i) - w(i)^T x(i)*
    - *Update w(i) = w(i) + μ e(i) x(i)*
  - *End*

Figure 3.3 LMS algorithm

3.2 Normalized Least Mean Squares (NLMS) Algorithm

NLMS algorithm was developed due to sensitivity of the LMS algorithm to scaling of its input $X(i)$, making it very difficult to choose the correct step-size to ensure stability of the algorithm. This algorithm works based on normalizing the power of the input and therefore, stabilizing the filter.

The modified weight update equation is shown below,

$$w(i) = w(i) + \frac{\mu\, e(i)\, x(i)}{\epsilon + \|x(i)\|^2}\ [2], \tag{4}$$

where is epsilon. The epsilon is a very small non-zero number to avoid division by 0 in denominator.

NLMS algorithm can be summarized as shown in Figure 3.4 below.

Inputs:
- *x = input signal*
- *dn = desired signal*
- *M = filter length*
- *μ = step-size factor*
- *E = epsilon*

Outputs
- *y = output of filter*
- *e = error signal*

Pseudo-code
- *Initialize:*
  - *filter coefficient w(1) = 0*
- *Loop1:*
  - *For every i, do*
    - *Calculate e(i) = d(i) - w(i)^T x(i)*
    - *Update w(i) = w(i) + μ e(i) x(i)/(E + ‖ x(i) ‖^2)*
  - *End*

Figure 3.4 NLMS algorithm

3.3 Gaussian Reproducing Kernel Hilbert Space (RKHS)

Gaussian RKHS is a mercer kernel. A kernel $k(i,j)$, is a continuous function that takes two inputs i and j and maps them to a real value independent of the order of the inputs, i.e., $k(i,j) = k(j,i) \in R$. The Mercer's theorem [18] "is a representation of a symmetric positive-definite function on a square as a sum of a convergent sequence of product functions" [15]. Since our noise sources are Gaussian, Gaussian RKHS is used instead of Polynomial RKHS.

Algorithm for Gaussian RKHS is shown in Figure 3.5 below.

Inputs:
x (i) = Input 1
x (j) = Input 2
α = Kernel Parameter
Outputs:
k = Gaussian Kernel
Pseudo-code:
$Calculate: KernelG(x(i),x(j)) = exp(-\alpha\|x(i) - x(j)\|)$

Figure 3.5 Gaussian RKHS

3.4 Kernel Least Mean Squares (KLMS) Algorithm

The basic LMS algorithm works best on the linear data, where mapping between the input signal and the desired signal is linear. For nonlinear data, that mapping between the desired signal and the input signal is nonlinear; the LMS algorithm will show poor performance. To improve performance, the LMS algorithm must be reformulated to be capable of learning nonlinear mapping. To create nonlinear mapping, KLMS algorithm uses RKHS to transform the input into higher dimension feature space. Since input

signal is in higher dimension, it will provide richer representation of it and therefore, its stochastic gradient descent will be more effective for nonlinear mapping of the LMS algorithm for nonlinear problems.

The KLMS algorithm weight update algorithm is similar to the LMS weight update algorithm and it is shown below,

$$w(i) = w(i) + \mu\, e(i)\, \varphi(i) \text{ [2]}, \tag{5}$$

where $\varphi(i) = \varphi\big(x(i)\big)$ is the higher dimension of input signal. As size of the input signal increases, size of the φ will increase as well; in case of the Gaussian kernel, it will increase to infinity. This may cause problem and the need for alternative way of computation. Using the kernel trick, we can reduce the computation, so that filter output in the input space can effectively be computed by the kernel evaluation. A kernel trick is a way of mapping inputs into an inner product space without the explicit need to compute the mapping [19]. By using the kernel trick, updated weight equation will transform to,

$$w(i)^T \varphi(\ ) = \mu \sum_{j=1}^{i} e(j) k(x(j), x') \text{ [2]}, \tag{6}$$

where is the inner products between transformed inputs.

Comparing the above weight update algorithm of KLMS algorithm to weight update of the LMS algorithm, we notice that the KLMS algorithm is calculated weight-less. Instead, it uses sum of all the past errors multiplied by the kernel evaluations on the previous data. This will enable computation of the output with a single inner produce by having direct access to weights.

We denote $y(i)$ as the output of the filter at time i , we will have the following new algorithm,

$$f(i-1) = \mu \sum^{i-1} e(j)\, k\big(x(j), x(i)\big) \text{ [2]} \tag{7}$$

This new algorithm is KLMS algorithm, which simply is the LMS in RKHS, and filtering is done by the kernel evaluation. Figure 3.4 below summarizes Equation 7.

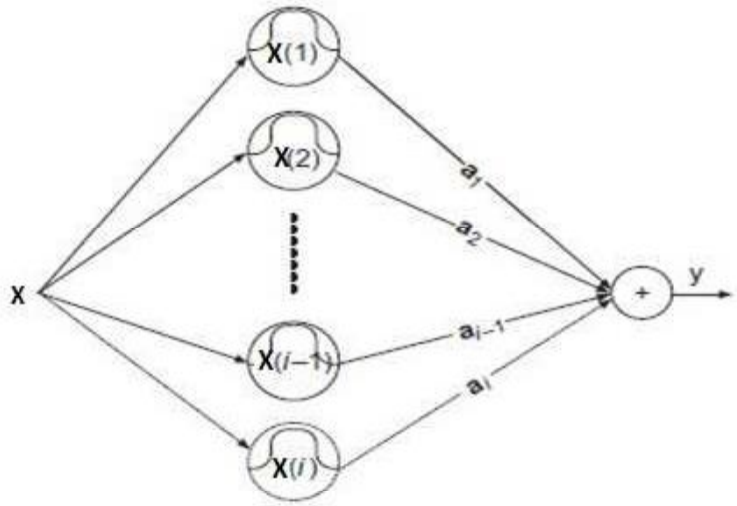

Figure 3.4 KLMS topology [2]

Unlike the LMS and the NLMS algorithms, the KLMS algorithm has both linear and non-linear step-size parameters. This allows improved results for the KLMS algorithm on non-linear signal filtering. The KLMS algorithm can be summarized as shown in Figure 3.5 below.

```
Inputs:
    x = input signal
    dn = desired signal
    M = filter length
    μ = step-size factor
    α = Kernel Parameter
    E = epsilon
Outputs
    y = output of filter
    e = error signal
Pseudo-code
    Initialize:
      filter coefficient w(1) = 0
      expansion coefficient a(1) = μ dn(1)
    Loop1:
      For every i, do
        Calculate for every j = 1:i-1
            y(i)= ∑aj(i) KernelG(x(i), x(j) )
        Calculate e(i) = d(i) - w(i)^T x(i)
        Update w(i) = w(i) + μ e(i) x(i)
        Update a(i) = μ e(i)
      End
```

Figure 3.5 KLMS algorithm

3.5 Normalized Kernel Least Mean Squares (NKLMS) Algorithm

Just like the NLMS algorithm that was derived from the LMS algorithm to improve performance of the LMS algorithm, the NKLMS algorithm is driven from the KLMS algorithm to improve KLMS algorithm performance. Updated weight function for NKLMS algorithm is shown below,

$$w(i) = w(i) + \frac{\mu\, e(i)\, x(i)}{\epsilon + \|\varphi(i)\|^2} \tag{8}$$

where $\|\varphi(i)\|^2$ is $\|\varphi(i)\|^2 = k(x(i), x(i))$.

Just like KLMS algorithm, NKLMS algorithm also has both linear and non-linear step-size parameters. KLMS algorithm can be summarized as shown in Figure 3.6 below.

Inputs:
- *x = input signal*
- *dn = desired signal*
- *M = filter length*
- *μ = step-size factor*
- *α = Kernel Parameter*

Outputs
- *y = output of filter*
- *e = error signal*

Pseudo-code
- *Initialize:*
  - *filter coefficient w(1) = 0*
  - *expansion coefficient a(1) = μ dn(1)*
- *Loop1:*
- *For every i, do*
  - *Calculate for every j = 1:i-1*
    - $Y(i) = \sum a_j(i)\ KernelG(x(i), x(j))$
  - *Calculate* $e(i) = d(i) - w(i)^T\ x(i)$
  - *Update* $w(i) = w(i) + \mu\ e(i)\ x(i) / (E + \| \phi(i) \|^2)$
  - *Update* $a(i) = \mu\ e(i)$
- *End*

Figure 3.6 NKLMS algorithm

3.6 Multistage Adaptive Filter

The multistage adaptive filter being used in this project is demonstrated in Figure 3.7 below.

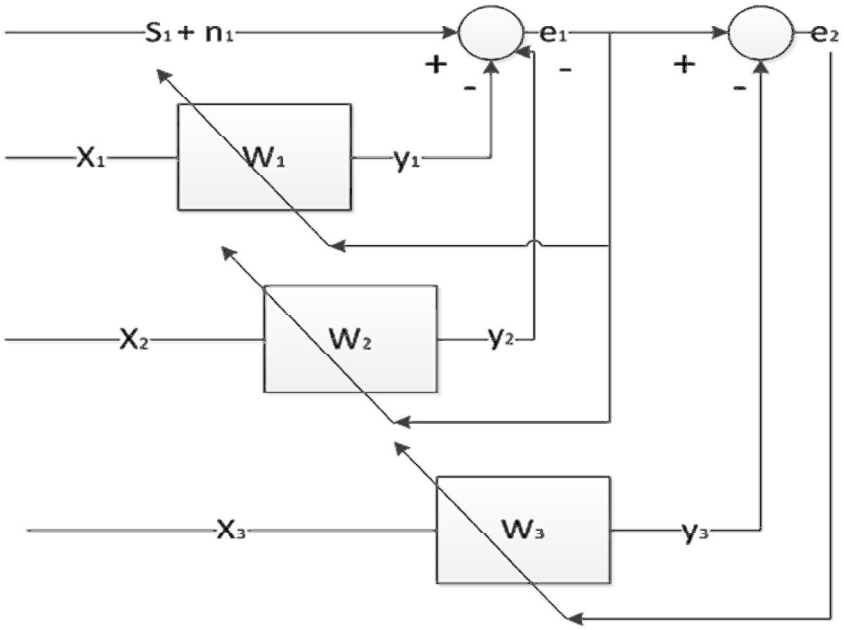

Figure 3.7 Multistage adaptive filter used in this project [8]

Our desired signal $s_1 + n_1$ is the noisy ECG signal, X_1 is 60Hz-PLI, and X_2 is baseline wander artifact. Output of this stage, appears at e_1. Signal e_1 will be used in QRS detection to generate X and used for filter final stage. Final clean ECG signal appears at y .

This filter is divided into 2 stages. 1st Stage removes 60Hz PLI and Baseline Wander. 2nd removes muscle artifacts and motion artifacts.

Section 4: Numerical Experiment

Selecting the optimum filter parameters for this noise cancellation is important. Generally for ECG filtering, filter parameters were calculated through approximation and then manually adjusted for best performance through trial and error. For PLI and Baseline Wander removal, the smaller the step-size, the better the overall filter will perform. The filter will converge slower but fewer artifacts will go through. For Filter 1 and 2 (Figure 3.7), linear and non-linear step-size of 0.006 and filter length of 20 was selected. For the 3rd filter, the correct filter length must be calculated first. Since we know that P-QRS-T segment consists of PR interval and QT interval, and knowing PR interval has duration of 50 to 120ms and QT interval has duration of 420ms, we can now estimate that P-QRS-T segment has duration of 470 to 540ms. Knowing that the sampling rate is 360Hz, which would correspond to 169.2 to 194.4 samples; that would mean filter length for P-QRS-T segment has to be at least 195 to ensure that segment will not be cut-off early. Therefore, for the 3rd filter, linear and non-linear step-size was selected to be 0.4 and filter length was calculated to be 200. Filter parameters were summarized in Table 4.1 as illustrated below.

	Linear Step-size	**Non-Linear Step-size**	**Filter Length**
Filter 1	0.006	0.006	20
Filter 2	0.006	0.006	20
Filter 3	0.4	0.4	200

Table 4.1 Filter Parameters.

Noisy ECG signals were created by adding artifacts to clean the ECG signals. The clean ECG signal was 1 minute duration of the signal record 100 from the MIT-BIH database [12]. All artifacts were also obtained from the MIT-BIT database [13]. The program to generate the noisy ECG signals is illustrated in Appendix A.

Once the noisy ECG signals were created, they were fed into main MATLAB program [Appendix B]. In the main program, performance of LMS, NLMS, KLMS and NKLMS algorithms on the noisy ECG signals were tested.

For the ECG filtering, first PLI and Baseline Wander were filtered out, and then QRS analysis was applied to determine the beginning of P-wave value for final filter stage of muscle artifact and motion artifact removal. This process was done for each LMS, NLMS, KLMS and NKLMS algorithm 3 times for each of 3 noisy ECG signals. For QRS detection, a 3rd party program was used which detects QRS features using Wavelet Method [7].

Generally when it comes down to analysis of the signals, the most common method of performance analysis is signal to noise ratio (SNR). However, SNR analysis for an ECG is meaningless because scientists and Emergency Medical Technicians (EMTs) that work with ECG signals are not interested in its SNR, but rather in its accuracy of detecting its wave parameters.

Instead of SNR, true positives (TPs), false negatives (FNs) and false positives (FPs) of wave parameters were calculated. TPs are the peak locations that were detected in filtered ECG and matched to peak locations to clean ECG. FNs are peak locations that were detected in clean ECG but were absent in filtered ECG. FP's are peak locations that

were present in filtered ECG but never existed in clean ECG. To analyze accuracy further, intervals at which peak locations were detected were also considered. 0.00s interval means that there was a match in the ECG feature between the ECG clean signal and the filtered ECG signal by sample number. Interval of ±0.01s means that peak locations are up to ±3 samples off based on the fact that the sampling rate is 360 samples per second and ±1% of 360 would be ±3.6. Interval of ±0.02s means that peak locations are up to ±7 samples off.

The results for LMS, NLMS, KLMS, and NKLMS algorithms are shown in the four tables below.

LMS	NoisyECG1			NoisyECG2			NoisyECG3		
P wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	5.3%	94.7%	93.5%	5.3%	94.7%	94.2%	4%	96%	95.5%
±0.01 s	22.7%	77.3%	72.6%	21.3%	78.7%	76.8%	20%	80%	77.6%
±0.02 s	28%	72%5	66.1%	30.7%	69.3%	66.7%	29.3%	70.7%	67.2%
Q wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	4%	96%	95.2%	2.7%	97.3%	97.1%	2.7%	97.3%	97%
±0.01 s	30.7%	69.3%	62.9%	16%	84%	82.6%	14.7%	85.3%	83.6%
±0.02 s	54.7%	45.3%	33.9%	40%	60%	46	28%	72%	68.7%
R wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	4%	96%	95.2%	4%	96%	95.7%	5.3%	94.7%	94%
±0.01 s	28%	72%	66.1%	28%	58	69.6%	24%	76%	73.1%
±0.02 s	65.3%	34.7%	21%	62.7%	37.3%	31.9%	56%	44%	37.3%
S wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	1.3%	98.7%	98.4%	0%	100%	100%	1.3%	98.7%	98.5%
±0.01 s	29.3%	70.7%	64.5%	32%	68%	65.2%	17.3%	82.7%	80.6%
±0.02 s	69.3%	30.7%	16.1%	69.3%	30.7%	24.6%	61.3%	38.7%	31.3%
T wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	0%	100%	100%	0%	100%	100%	1.3%	98.7%	98.5%
±0.01 s	9.3%	90.7%	88.7%	14.7%	63	84.1%	17.3%	82.7%	80.6%
±0.02 s	29.3%	70.7%	64.5%	28%	72%	69.6%	32%	68%	64.2%

Table 4.2 LMS Results. Number of Samples = 21600.

NLMS	NoisyECG1			NoisyECG2			NoisyECG3		
P wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	33.3%	66.7%	64.3%	28%	72%	63.2%	16%	84%	78.9%
±0.01 s	40%	60%	57.1%	40%	60%	47.4%	30.7%	69.3%	59.6%
±0.02 s	40%	60%	57.1%	42.7%	57.3%	43.9%	34.7%	65.3%	54.4%
Q wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	1.3%	98.7%	98.6%	1.3%	98.7%	98.2%	0%	100%	100%
±0.01 s	22.7%	77.3%	75.7%	13.3%	86.7%	82.5%	9.3%	90.7%	87.7%
±0.02 s	33.3%	66.7%	64.3%	34.7%	65.3%	54.4%	21.3%	78.7%	71.9%
R wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	5.3%	94.7%	94.3%	2.7%	97.3%	96.5%	2.7%	97.3%	96.5%
±0.01 s	38.7%	61.3%	58.6%	21.3%	44	71.9%	20%	80%	73.7%
±0.02 s	53.3%	46.7%	42.9%	42.7%	57.3%	43.9%	45.3%	54.7%	40.4%
S wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	5.3%	94.7%	94.3%	5.3%	94.7%	93%	5.3%	94.7%	93%
±0.01 s	36%	64%	61.4%	24%	76%	68.4%	18.7%	81.3%	75.4%
±0.02 s	56%	44%	40%	41.3%	58.7%	45.6%	38.7%	61.3%	49.1%
T wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	1.3%	98.7%	98.6%	0%	100%	100%	1.3%	98.7%	98.2%
±0.01 s	16%	84%	82.9%	5.3%	94.7%	93%	9.3%	90.7%	87.7%
±0.02 s	33.3%	66.7%	64.3%	17.3%	82.7%	77.2%	18.7%	81.3%	75.4%

Table 4.3 NLMS Results. Number of Samples = 21600.

KLMS	NoisyECG1			NoisyECG2			NoisyECG3		
P wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	6.7%	93.3%	92.8%	5.3%	94.7%	94.4%	5.3%	94.7%	94.4%
±0.01 s	26.7%	73.3%	71%	25.3%	74.7%	73.2%	24%	76%	75%
±0.02 s	36%	64%	60.9%	32%	68%	66.2%	36%	64%	62.5%
Q wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	9.3%	90.7%	89.9%	6.7%	93.3%	93%	5.3%	94.7%	94.4%
±0.01 s	56%	44%	39.1%	49.3%	50.7%	47.9%	46.7%	53.3%	51.4%
±0.02 s	58.7%	41.3%	36.2%	50.7%	49.3%	46.5%	50.7%	49.3%	47.2%
R wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	18.7%	81.3%	79.7%	21.3%	78.7%	77.5%	14.7%	85.3%	84.7%
±0.01 s	93.3%	6.7%	0%	96%	4%	0%	97.3%	2.7%	0%
±0.02 s	93.3%	6.7%	0%	96%	4%	0%	97.3%	2.7%	0%
S wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	5.3%	94.7%	94.2%	4%	96%	95.8%	4%	96%	95.8%
±0.01 s	60%	40%	34.8%	64%	36%	32.4%	65.3%	34.7%	31.9%
±0.02 s	88%	12%	4.3%	93.3%	6.7%	1.4%	92%	8%	4.2%
T wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	4%	96%	95.7%	1.3%	98.7%	98.6%	1.3%	98.7%	98.6%
±0.01 s	20%	80%	78.3%	18.7%	81.3%	80.3%	26.7%	73.3%	72.2%
±0.02 s	44%	56%	52.2%	37.3%	62.7%	60.6%	38.7%	61.3%	59.7%

Table 4.4 KLMS Results. Number of Samples = 21600.

NKLMS	NoisyECG1			NoisyECG2			NoisyECG3		
P wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	6.7%	93.3%	92.8%	5.3%	94.7%	94.4%	5.3%	94.7%	94.4%
±0.01 s	26.7%	73.3%	71%	25.3%	74.7%	73.2%	24%	76%	75%
±0.02 s	36%	64%	60.9%	32%	68%	66.2%	36%	64%	62.5%
Q wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	9.3%	90.7%	89.9%	6.7%	93.3%	93%	5.3%	94.7%	94.4%
±0.01 s	56%	44%	39.1%	49.3%	50.7%	47.9%	46.7%	53.3%	51.4%
±0.02 s	58.7%	41.3%	36.2%	50.7%	49.3%	46.5%	50.7%	49.3%	47.2%
R wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	18.7%	81.3%	79.7%	21.3%	78.7%	77.5%	14.7%	85.3%	84.7%
±0.01 s	93.3%	6.7%	0%	96%	4%	0%	97.3%	2.7%	0%
±0.02 s	93.3%	6.7%	0%	96%	4%	0%	97.3%	2.7%	0%
S wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	5.3%	94.7%	94.2%	4%	96%	95.8%	4%	96%	95.8%
±0.01 s	60%	40%	34.8%	64%	36%	32.4%	65.3%	34.7%	31.9%
±0.02 s	88%	12%	4.3%	93.3%	6.7%	1.4%	92%	8%	4.2%
T wave Peaks	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**
0.00 s	4%	96%	95.7%	1.3%	98.7%	98.6%	1.3%	98.7%	98.6%
±0.01 s	20%	80%	78.3%	18.7%	81.3%	80.3%	26.7%	73.3%	72.2%
±0.02 s	44%	56%	52.2%	37.3%	62.7%	60.6%	38.7%	61.3%	59.7%

Table 4.5 NKLMS Analysis results. Number of Samples = 21600.

The time it takes for computer to execute each section of program on system with Intel Quad core Xeon 1240 CPU with 8GB of ram is provided in figure below.

	LMS	**NLMS**	**KLMS**	**NKLMS**
Time to Compute (Seconds)	0.220	0.307	589.061	589.491

Table 4.6 Computation time on Intel Xeon 1240 CPU. Number of Samples = 21600.

Graphical results are provided in Figures 4.1, 4.2, 4.3, and 4.4 below.

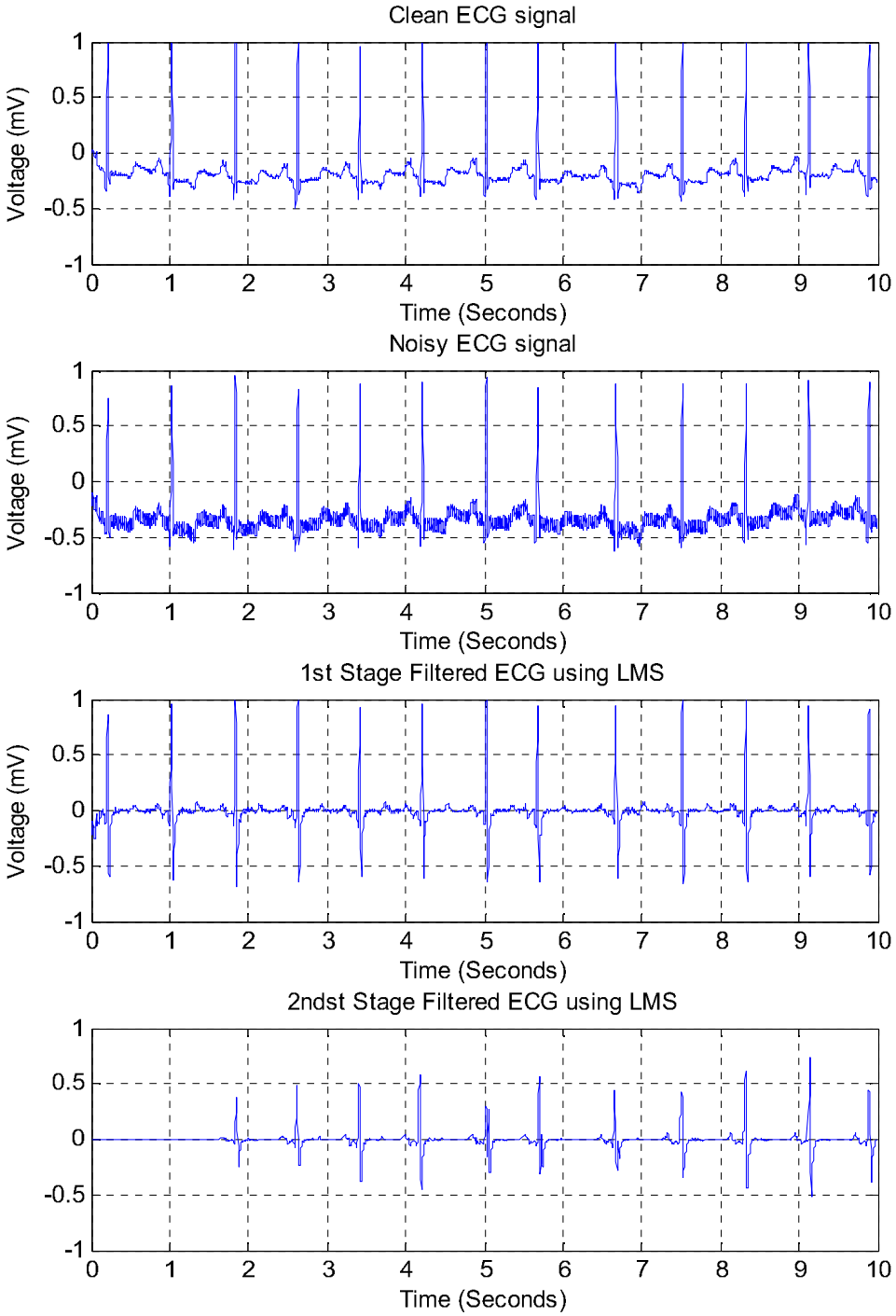

Figure 4.1 Results using LMS algorithm. Number of Samples = 3600.

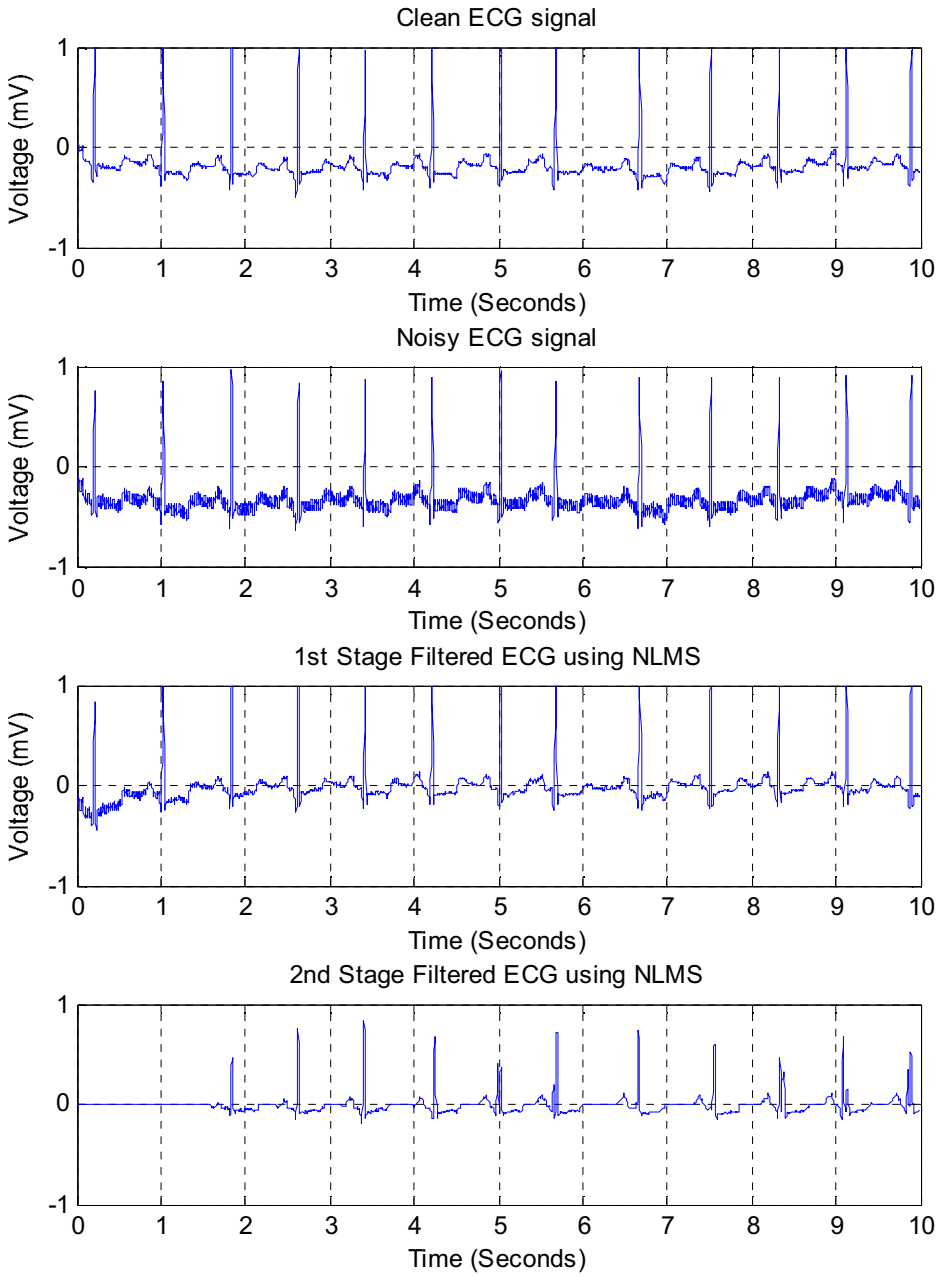

Figure 4.2 Results using NLMS algorithm. Number of Samples = 3600.

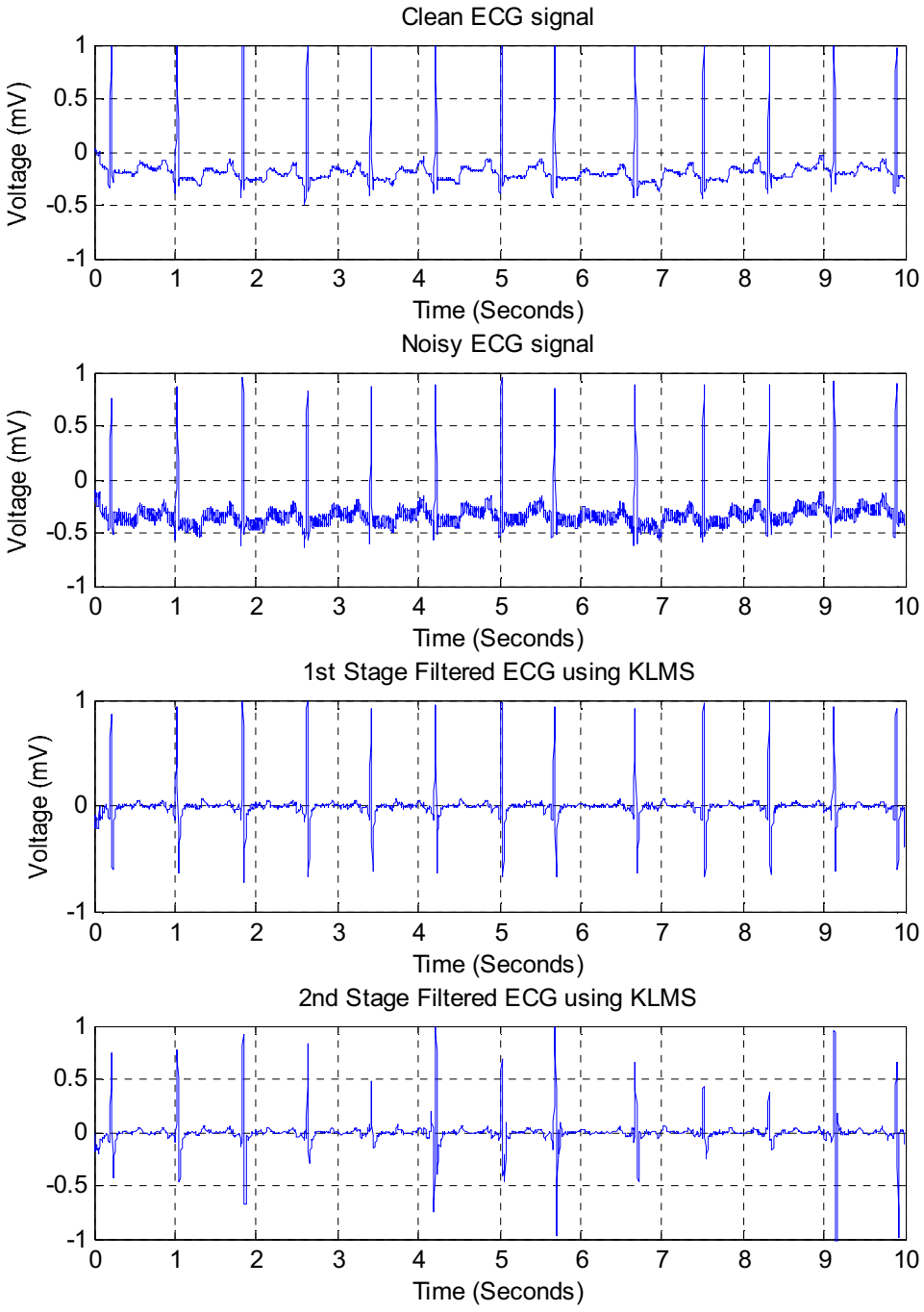

Figure 4.3 Results using KLMS algorithm. Number of Samples = 3600.

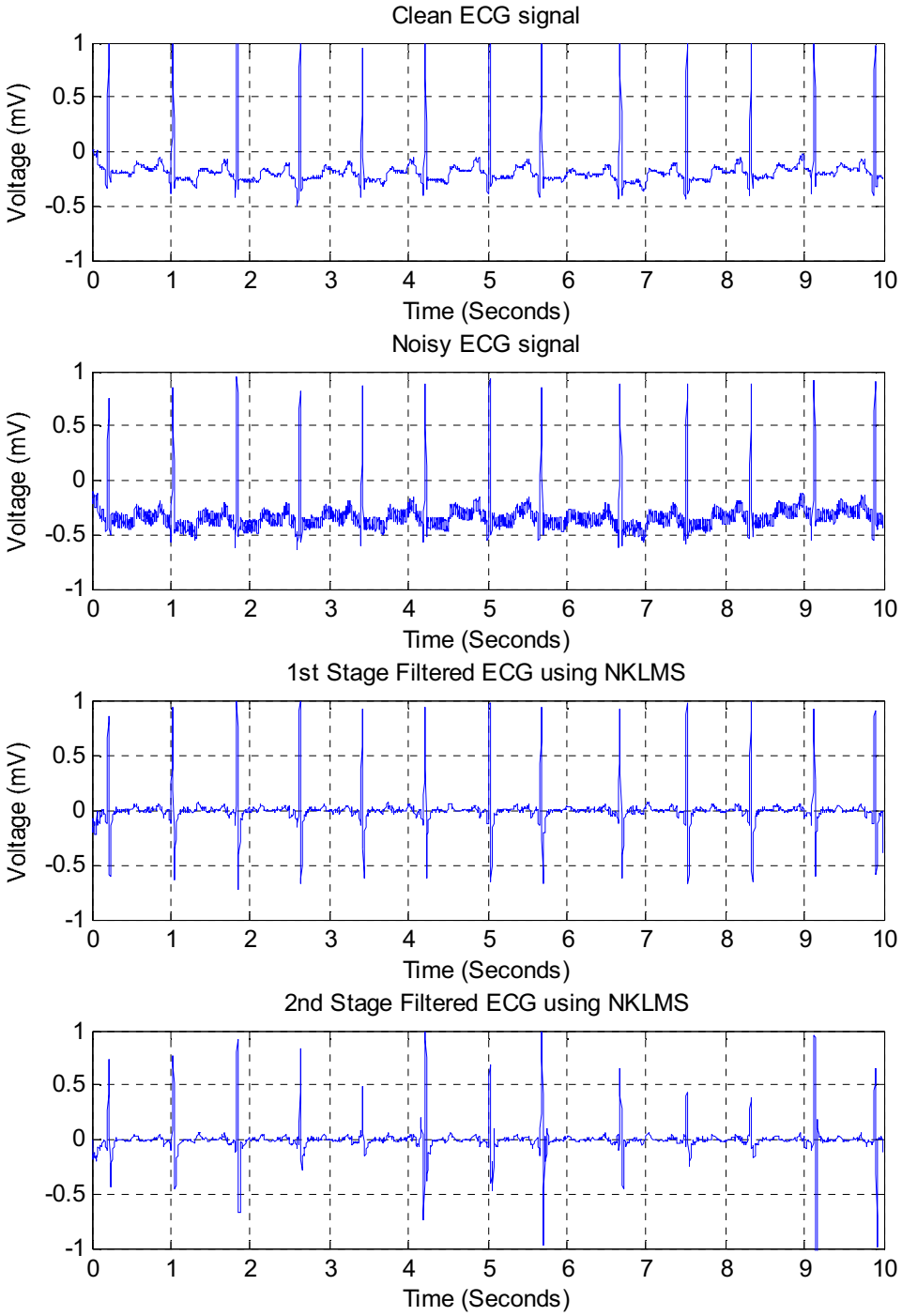

Figure 4.4 Results using NKLMS algorithm. Number of Samples = 3600.

Section 5: Conclusion

Based on results from graphs and tables, it can be seen that the KLMS and NKLMS algorithms have much better performance than the LMS and NLMS algorithms. However, even though this program is designed for online computation, it still takes a while to compute based on calculation time provided in Table 4.6.

As for improving the results, one can implement variable step-size algorithm which can help on faster convergence of algorithm at the start. Faster adaptation allows the signal to be analyzed by smaller segments, which will help on reducing the Gaussian RKHS. Reduction in the Gaussian RKHS means fewer algorithm computations.

Another thing to note is that the KLMS and NKLMS algorithms have linear and non-linear step-size parameters. For the purpose of this project both parameters were set to be equal. By trial and error, it is possible to adjust non-linear parameters to improve the performance of the KLMS and NKLMS algorithms.

	LMS	**NLMS**	**KLMS**	**NKLMS**
Data Modeling	Linear	Linear	Linear and Non-Linear	Linear and Non-Linear
Computation	Fast	Fast	Moderate	Moderate
Simplicity	Simple	Simple	Moderate	Moderate

Table 4.7 Summary of Algorithm Differences

References

[1] S. Haykin, Adaptive Filter Theory, Prentice Hall, 2002.

[2] W. Liu, J. Principe and S. Haykin, Kernel Adaptive Filtering: A Comprehensive Introduction, Wiley, 2010.

[3] http://en.wikipedia.org/wiki/Electrocardiography. (obtained 3/1/2013).

[4] http://en.wikipedia.org/wiki/Adaptive_filter. (obtained 3/1/2013).

[5] http://en.wikipedia.org/wiki/Kernel_adaptive_filter. (obtained 3/1/2013).

[6] http://www.cnel.ufl.edu/~weifeng/publication.htm. (obtained 3/1/2013).

[7] http://technical.grasshoppernetwork.com/?q=node/156. (obtained 2/3/2013).

[8] N. V. Thakor, J. G. Webster and W. J. Tompkins, "Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection," *Biomedical Engineering, IEEE Transaction,* vol. 38, no. 8, pp. 785-794, Aug. 1991.

[9] E. R. Ferrara and B. Widrow, "Fetal electrocardiogram enhancement by time-sequenced adaptive filtering," *Biomedical Engineering, IEEE Transactions on,* Vols. BME-29, no. 6, pp. 458-460, June 1982.

[10] B. Widrow, J. Glover and J. McCool, "Adaptive noise canceling: principles and applications," *Proceedings of the IEEE,* vol. 63, no. 12, pp. 1692-1716, 1975.

[11] M. Yelderman, B. Widrow, J. Ciotfi, E. Hesler and J. Leddy, "ECG enhancement by

adaptive cancellation of electrosurgical interference," *lEEE Trans. Biomcd. Eng,* Vols. BME-30, no. 7, pp. 392-398, July 1983.

[12] http://www.physionet.org/physiobank/database/mitdb/100.dat. (obtained 3/1/2013).

[13] http://www.physionet.org/physiobank/database/nstdb/.(obtained 3/1/2013).

[14] http://www.grasshoppernetwork.com/Technical/Share/?q=node/156.(obtained 3/1/2013).

[15] http://en.wikipedia.org/wiki/Mercer%27s_theorem. (obtained 3/1/2013).

[16] http://fourier.eng.hmc.edu/e161/lectures/gaussianprocess/node8.html. (obtained 3/1/2013).

[17] J . A. Van Alste and T. S. Schilder. “Removal of baseline wander and power-line interference from the ECG by an efficient FIR filter with a reduced number of taps,“ *IEEE Tram. Biomed. Eng.*. vol. BME-32. pp. 1052-1062, 1985.

[18] Mercer, J. (1909), "Functions of positive and negative type and their connection with the theory of integral equations", *Philosophical Transactions of the Royal Society A* 209 (441–458): 415–446.

[19] http://www.cs.berkeley.edu/~jordan/courses/281B-spring04/lectures/lec3.pdf. (obtained 3/1/2013).