EXPERIMENTAL EVALUATION OF VARIOUS ADAPTIVE SIGNAL PROCESSING TECHNIQUES FOR EXTRACTING FETAL ELECTROCARDIOGRAMS FROM NONINVASIVE MEASUREMENTS

A Thesis in
Electrical Engineering
by
Yuqing Dong

© 2016 Yuqing Dong

Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science

May 2016
The thesis of Yuqing Dong was reviewed and approved* by the following:

Kenneth W. Jenkins
Professor of Electrical Engineering
Thesis Adviser

Shizhuo Yin
Professor of Electrical Engineering

Kultegin Aydin
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School.
ABSTRACT

Electrocardiograms (ECGs) are used in the medical field for observing the cardio health of a patient. The fetal ECGs (FECGs) work similarly but are applied for monitoring fetal health during pregnancy, which is more complicated than normal ECGs. The state-of-the-art FECG techniques are not well enough developed to be used as a practical tool in clinical practice. The goal of this thesis topic is to develop a method to improve the accuracy of the fetal ECGs by applying digital filtering techniques to weaken and even remove interference signals including the maternal signal and echo noise. The experiments confirmed that the frequency domain non-power normalized sequential processing removes most maternal interference and applying the adaptive comb filter as the very last step further refines the result. Adaptive filtering techniques are applied to self-adjust the filter parameters to provide the optimal performance of the filtering process. A sequential filtering structure with linear prediction (LPC), an adaptive noise canceller (ANC), and an adaptive comb filter (ACF) was determined to by Matlab simulation. Critical values for several important adaptive filter parameters such as optimal descending step size, iteration numbers and order, and filter coefficients were also determined during the experiments. This thesis presents a theory of how LPC-ANC-ACF sequential processing is capable to extract fetal signals from noninvasive abdominal measurements. Results are presented to demonstrate that 1) frequency domain processing is more effective than time domain processing in signal pre-process stage, 2) multi-channel adaptive processing is more accurate than signal channel processing, and 3) the adaptive comb filter processing has a better performance in extracting fetus signal than non-adaptive does. In addition, results illustrate that sequential LPC-ANC processing
followed by comb filtering is more effective than direct looping these three stages while the sequential in-loop LPC-ANC-ACF processing is more effective than the single ACF processing.

Keywords: Adaptive signal processing; Adaptive comb filtering; Fetal electrocardiogram; Fetal health monitoring; Sequential processing.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>List of Charts</td>
<td>ix</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 2: BACKGROUND AND LITERATURE REVIEW</td>
<td>3</td>
</tr>
<tr>
<td>CHAPTER 3: METHODOLOGY</td>
<td>7</td>
</tr>
<tr>
<td>Signal Pre-Process with Linear Prediction and Adaptive Noise Canceller</td>
<td>7</td>
</tr>
<tr>
<td>Experimental comparisons of Frequency Domain and Time Domain Adaptive Filtering</td>
<td>8</td>
</tr>
<tr>
<td>Window Size Determination in Partial Power Normalization</td>
<td>9</td>
</tr>
<tr>
<td>Comb Filtering Application</td>
<td>10</td>
</tr>
<tr>
<td>Comb Filtering in Sequential Processing</td>
<td>10</td>
</tr>
<tr>
<td>Filter Type Selection for Comb Filtering</td>
<td>11</td>
</tr>
<tr>
<td>Adaptive Comb Filtering Application</td>
<td>11</td>
</tr>
<tr>
<td>Adaptive Comb Filtering in Sequential Processing</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER 4: RESULTS</td>
<td>13</td>
</tr>
<tr>
<td>The Steepest Descent Optimization in LMS Algorithm</td>
<td>13</td>
</tr>
<tr>
<td>Frequency Domain and Time Domain Adaptive Filtering Normalization</td>
<td>13</td>
</tr>
<tr>
<td>Window Size Determination in Partial Power Normalization</td>
<td>16</td>
</tr>
<tr>
<td>Comb Filtering Application</td>
<td>16</td>
</tr>
<tr>
<td>Comb Filtering in Sequential Processing</td>
<td>16</td>
</tr>
<tr>
<td>Filter Type Selection for Comb Filtering</td>
<td>19</td>
</tr>
<tr>
<td>Adaptive Comb Filtering Application</td>
<td>19</td>
</tr>
<tr>
<td>Adaptive Comb Filtering in Sequential Processing</td>
<td>20</td>
</tr>
<tr>
<td>CHAPTER 5: CONCLUSION</td>
<td>26</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>27</td>
</tr>
<tr>
<td>Appendix A</td>
<td>28</td>
</tr>
<tr>
<td>Appendix B</td>
<td>33</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure Page
Figure 1: Block diagram of time domain LMS ................................................................. 3
Figure 2: Block diagram of frequency domain LMS .......................................................... 4
Figure 3: Stage 1, Linear Prediction .................................................................................. 7
Figure 4: Stage 2, Adaptive Noise Canceller ....................................................................... 8
Figure 5: Diagonal Entries in Frequency Power Normalization ............................................ 9
Figure 6: LPC-ACF block diagram ...................................................................................... 10
Figure 7: ANC-ACF block diagram .................................................................................... 10
Figure 8: Power Normalization and Non-power Normalization Comparison in Iteration 1 ......................................................................................................................... 14
Figure 9: Power Normalization and Non-power Normalization Comparison in Iteration 2 ......................................................................................................................... 14
Figure 10: Power Normalization and Non-power Normalization Comparison in Iteration 3 ..................................................................................................................... 15
Figure 11: Power Normalization and Non-power Normalization Comparison in Iteration 4 ..................................................................................................................... 15
Figure 12: Applying Comb Filter at the 1st Iteration ............................................................. 17
Figure 13: Applying Comb Filter at the 2nd Iteration ......................................................... 17
Figure 14: Applying Comb Filter at the 3rd Iteration ........................................................... 18
Figure 15: Applying Comb Filter at the 4th Iteration ........................................................... 18
Figure 16: Comparison of 4 non-adaptive comb filtering techniques .................................... 19
Figure 17: Comparison of multi-alpha and single alpha updating ACF ................................. 20
Figure 18: Comparison of multi-alpha and single alpha updating ACF (ACF applied at 7th iteration) .............................................................................................................. 21
Figure 19: Comparison of multi-alpha and single alpha updating ACF (ACF applied at 5th iteration) .............................................................................................................. 22
Figure 20: Comparison of multi-alpha and single alpha updating ACF (ACF applied at 3rd iteration) ............................................................... 22

Figure 21: Comparison of multi-alpha and single alpha updating ACF (ACF applied at 2nd iteration) ............................................................... 22

Figure 22: Comparison of multi-alpha and single alpha updating ACF (ACF applied at 1st iteration) ............................................................... 23

Figure 23: Comparison of with and without Stage 2 (LPC-ACF) ............................................................... 24

Figure 24: Comparison of with and without Stage 2 (ANC-ACF) ............................................................... 24

Figure 25: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration) ............................................................... 33

Figure 26: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration) ............................................................... 33

Figure 27: Sample 244, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration) ............................................................... 34

Figure 28: Sample 244, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration) ............................................................... 34

Figure 29: Sample 244, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration) ............................................................... 35

Figure 30: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration) ............................................................... 35

Figure 31: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration) ............................................................... 36

Figure 32: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration) ............................................................... 36

Figure 33: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration) ............................................................... 37

Figure 34: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration) ............................................................... 37

Figure 35: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration) ............................................................... 38

Figure 36: Sample 746, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration) ............................................................... 38
Figure 37: Sample 746, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration) .................................................................................................................. 39

Figure 38: Sample 746, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration) .................................................................................................................. 39

Figure 39: Sample 840, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration) .................................................................................................................. 40

Figure 40: Sample 840, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration) .................................................................................................................. 40

Figure 41: Sample 840, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration) .................................................................................................................. 41
LIST OF CHARTS

<table>
<thead>
<tr>
<th>Chart</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart 1: Sample Test for μ Optimization in Frequency Domain</td>
<td>28</td>
</tr>
<tr>
<td>Chart 2: Sample Test for μ Optimization in Time Domain</td>
<td>29</td>
</tr>
<tr>
<td>Chart 3: Frequency Domain and Time Domain Comparison</td>
<td>30</td>
</tr>
<tr>
<td>Chart 4: Window Size Comparison in Frequency Domain Partial Power</td>
<td>31</td>
</tr>
<tr>
<td>Normalization</td>
<td></td>
</tr>
<tr>
<td>Chart 5: Periodicity Test in Comb Filter</td>
<td>32</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Professor Kenneth William Jenkins for the helpful comments, guidance and engagement through the learning process of this master thesis. Furthermore I would like to thank Ziyan Yao for helping me doing extra comparison experiments with different parameters. Also, I like to thank the PhysioNet website who offered me free web access to large collections of recorded physiologic signals that I used in simulation experiments. I would like to thank my loved families and friends, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together.
CHAPTER 1: Introduction

Fetal ECGs contain important information that indicates potential cardio defects of the fetus [1]. Two main methods are currently applied on FECGs. One is called the invasive method, which places a scalp electrode directly on the fetus after the water sac has broken. Another is called a non-invasive method for which several electrodes are connected externally to the abdominal area of the pregnant mother and positioned as close to the fetus as possible[2,3]. The invasive method provides highly accurate results, but it can only be performed during labor. The non-invasive method takes advantage by providing a long-term monitoring of the fetal cardiac development that is valuable for future medical adjustments since it can be performed during the whole period of pregnancy and delivery.

However, current non-invasive FECG techniques are not able to effectively remove maternal interference. A variety of noise sources corrupts the fetal measurement including, mainly, maternal signal, electrode noise, and noise generated by physical movements by both fetus and mothers. The maternal interference is not only the mother's heart beat, which is a relatively stronger periodic signal than the fetal heartbeat, but there is also non-periodic noise including the maternal heart beat distorted by body tissue, noise from muscle movement, and echoes from both the fetal and maternal heart beats.

To minimize the influence of maternal interference, the noise should be eliminated and the maternal cardio signal should be suppressed. Since single stage of filtering process is not able to process the signal with multiple type of noise, a sequential process with multiple filters that are designed to handle specific types of noise becomes a better option. Multi-stage sequential adaptive filtering processing techniques consisting of three stages
were developed. The three stages consisted of adaptive linear prediction (LPC), adaptive noise cancelling (ANC), and adaptive comb filtering (ACF) [4, 5, 6]. The linear prediction in stage one adjusts the power of both the maternal and fetal components to the similar levels so that the adaptive noise canceller in stage two can then very effectively reduce the noise interference. The adaptive comb filter in stage three extracts the fetal signal and suppresses the maternal signal further to the noise floor and, at the same time, adjusts its filter coefficients for the optimized performance during each iteration. All the data used in the following experiments included maternal thorax and abdomen signals, and fetal signals that are provided by the PhysioNet website. The samples labeled "ecgca886" and "r01" are the primary signals using during most experiments in this work.
CHAPTER 2: Background and Literature Review

The adaptive filtering algorithm used for all of the adaptive filters in this work is the Least-Mean-Square (LMS) algorithm [2]. The LMS time domain adaptive filter (TDAF) is characterized by the following equations:

\[ y[n] = w^T[n] x[n] \]  \hspace{1cm} (1)

\[ e[n] = d[n] - y[n] \]  \hspace{1cm} (2)

\[ w'[n+1] = w'[n] + 2\mu e[n]x[n] \]  \hspace{1cm} (3)

where \( x[n] = [x[n], x[n-1], \ldots, x[n-M+1]]^T \) is a vector of current and past input samples, \( d[n] \) is the desired signal, \( y[n] \) is the adaptive filter output, and \( w'[n] \) is the vector of the adaptive filter coefficients (tap weights) at iteration \( n \). The LMS algorithm is an adaptive algorithm that approximates the steepest descent optimization algorithm. This approximation provides an accurate adaptive algorithm that is simple to implement and not computationally intensive. The approximation can be done both in time domain and frequency domain. Figure 1 and figure 2 show the block diagram of these two approximation methods. The structures are mostly the same except the linear transformation block in figure 2, frequency domain LMS.

![Figure 1 Block diagram of time domain LMS](image-url)
Figure 2 Block diagram of frequency domain LMS

There are several important factors to be accounted for when using the LMS algorithm. The first is the algorithm step size parameter $\mu$. Different values of $\mu$ will provide different convergence rates for the system, but may also introduce gradient noise. Care must also be taken not to have a value of $\mu$ too large or too small; large values of $\mu$ will cause the system to become unstable, while small values of $\mu$ will prevent the system from converging. The next parameter is the length $M$ of the filter (for FIR implementations), and the last parameter is the amount of time delay, if any, in the input signal. The LMS algorithm minimizes the average error between the desired signal $d[n]$ and filter output $y[n]$. A large amplitude signal converges fast, however, a small amplitude signal does the opposite, i.e. it converges more slowly. The maternal signal with higher amplitude than fetal signal starts to converge early at a significantly fast rate. One of the important properties of $\mu$, the step size of steepest descent, is that it controls the system convergence rate.

A variation of the LMS algorithm called the Power Normalized LMS algorithm was also studied in this work. This form of the LMS algorithm allows large variations in the input signal without causing the system to become unstable. Equations (4) and (5) show the power normalized filter update equations. The term $\alpha$ is a constant. For all cases in this paper, both in the time domain and frequency domain, a value of $\alpha = 0.001$ was used. As
seen in Equation (5), \( \sigma^2[n] \) is a parameter based on past values of the input. This memory prevents the filter update from drastically changing between values.

\[
\mathbf{w}^T[n + 1] = \mathbf{w}^T[n] + \frac{\mu \mathbf{e}[n]}{\alpha + \sigma^2[n]} \mathbf{x}[n] \tag{4}
\]

\[
\sigma^2[n] = \frac{1}{M} \sum_{i=1}^{M} x^2[n-i] \tag{5}
\]

In this experiment, the Matlab for loop code is applied to approximate the optimized value of \( \mu \). By increasing the \( \mu \) factor in each loop, maternal signal converges faster into the noise level while the fetal convergence rate has no significant change. At the same time, larger \( \mu \) value allows larger noise damping range. The \( \mu \) optimization test is run separately for frequency domain and time domain since the optimized values are different in each case. The optimized \( \mu \) is determined when the fetal signal is exposed from the maternal interference while the noise is not excessively introduced into the system.

The frequency domain adaptive filter (FDAF) is another variation of the adaptive algorithm used to apply the LMS algorithm to separate frequency channels of the input signal, rather than the entire input [2]. Although several types of mappings can be used to transform the input signal, the one used in this paper is the Discrete Fourier Transform (DFT) that is implemented with the Fast Fourier Transform (FFT) algorithm. The input reference signal is windowed according to the length of the filter and then subsequently mapped to the discrete frequency (DFT) domain. Each frequency channel is subsequently adaptively filtered and summed, and the summation of the filtered channels provides the output of the system, \( y[n] \). This output is then subtracted from the noisy abdominal signal. Once again, the filter was configured in either a linear predictive or noise cancelling configuration. The following equations characterize the frequency domain adaptive filter (FADF):

\[
\mathbf{w}[n + 1] = \mathbf{w}[n] + 2\mu \mathbf{e}[n] \mathbf{x}[n] \tag{6}
\]

\[
y[n] = \mathbf{x}[n] \cdot \mathbf{w}^H[n] \tag{7}
\]

where \( \mathbf{w}[n] \) and \( \mathbf{x}[n] \) are the filter coefficient and input signal vectors that have been transformed into the frequency domain.
A power normalized variation of the transform domain filter was also applied to the signals. Equation 8 shows that the frequency domain power normalization is similar to the time domain power normalization in Equation 4. However, rather than a single $\sigma[n]$ value, multiple $\sigma_k[n]$ values corresponding to the different frequency channels of the transformed signals are used. $\bar{\sigma}[n]$ is a matrix containing the corresponding $\sigma_k[n]$ values on the matrix diagonal. $\bar{I}$ is an identity matrix, so that $\alpha$ may be applied to control possible numerical ill-conditioning that may occur each channel of the filter.

$$W[n+1] = W[n] + 2\mu_k[n](\alpha\bar{\sigma}^2[n])^{-1}X[n]$$

$$\sigma_k[n] = X_k[n] \cdot X_k[n]$$

The transfer function of a FIR comb filter is shown in Equation 10. The parameter $R$ is the delay of echoes and the parameter $\alpha$ is the signal loss during the reflection and propagation of the signal. Adaptive comb filter is adjusting the $\alpha$ values in order to gain the optimized filter performance.

$$H(z) = 1 + \alpha z^{-R} + \alpha^2 z^{-2R} + \cdots + \alpha^{N-1} z^{-(N-1)R}$$
CHAPTER 3: METHODOLOGY

Signal Pre-Process with Linear Prediction and Adaptive Noise Canceller

Both stage 1, Linear Prediction (LPC), and stage 2, Adaptive Noise Canceller (ANC), are using LMS algorithm while the configuration of two stages are different and hence the have specialized functionalities.

Figure 3 is the block diagram of stage 1, LPC. It brings both maternal signal and fetal signal components to approximately the same amplitude level. Abdominal signal is the input signal and is duplicated as two identical signals. One copy is set as reference signal (signal \(d(n)\)) and the other is pushed to LMS adaptive filter with added delay. An error (signal \(e(n)\)) is generated by taking the difference of these two signals and to optimize the adaptive filter response. The output (signal \(y(n)\)) is then passed to stage 2 for noise removal processing.

Figure 4 is the block diagram of stage 2, ANC. It reduces noise interference in order to further expose the periodical signals, especially the fetal signal. The processed abdominal signal, which is also the output of previous stage, is the primary input signal and the thorax signal is the reference signal. The difference of these two signals produces the error signal which is to optimize the adaptive filter in this stage. The output is looped back in the next iteration or becoming the input of the third stage which is covered in the next few sections.
Experimental comparisons of Frequency Domain and Time Domain Adaptive Filtering

Frequency domain and time domain filter are tested in the experiment to determine a better filter for further processing and future study. Referring to the previous study [9], the sequential processing offers the better result after executing the whole iteration loop 4 times. The experiment compares the 4 iteration results for both frequency domain and time domain filters and plots the results onto the same chart to facilitate the comparison. The criteria in this comparison are: 1) whether the fetal signal is differentiable from the maternal interference. 2) whether the noise is kept in a low level. 3) which system convergence rate is faster.

Power Normalization and Non-power Normalization

The power normalization described above is the algorithm that brings different input signals to the same power level. This technique is commonly applied to unifying input voice level in audio communication. Variations from personal speaking habits, people have loud or soft voice, power normalized microphone brings all the speaker voice to an acceptable volume by suppressing the loud voice or extracting the soft voice. In fetal ECG measurement, maternal interference is considerably larger than fetal signal. Power normalization is applied to control the strength of typical channels in $X[n]$ by updating the coefficients in $\sigma[n]$. In time domain the $\sigma[n]$ is a single value while in frequency domain it is a 64 by 64 matrix. A large $X[n]$ value is scaled down by a large $\sigma^2$ factor and in otherwise for a small $X[n]$. The non-zero constant $\alpha$ prevents the system...
becoming unstable when $\sigma$ approaches zero so that the denominator becomes zero. The parameter $\alpha$ is set as 0.01 during the experiments. However, power normalization loads up the system calculation time. The experiments below compare power normalization and non-power normalization in both frequency domain and time domain. The comparison results are also plotted on the same chart. Slight improvement with power normalization is not considered as finding a better processing method since extra calculation time added by power normalization reduces the system processing efficiency.

Window Size Determination in Partial Power Normalization

Power normalization slows down the system processing performance. However, partial power normalization can be performed in frequency domain filter which saves time for specific channels that have excessive large amplitude. The basic calculation algorithms in frequency domain and time domain filter are different which is one reason that partial power normalization can only be performed on frequency domain filter. The output $X[n]$ is a one channel signal in both filter but Fourier Transformation refines the $X[n]$ in frequency domain filter so that the refined $X[n]$ becomes a 64 channel signal instead of a 1 channel signal in time domain filter. The coefficient factor $\sigma^2$ becomes a 64 by 64 diagonal matrix accordingly. The 64 diagonal entries the scalars the $X[n]$. According to Figure 5, the value of 64 scalars are non-uniform but have a symmetrical and periodical trend. The first channel (up to $1.05 \times 10^5$) and two other peaks (up to $1 \times 10^4$) are extremely large while the most channels are around 50.

![Figure 5: Diagonal Entries in Frequency Power Normalization](image)

For partial power normalization, the window size determines how many channels are power normalized. In this experiment, window sizes ranging from 1 to 64 are tested and
the results are compared with non-power normalized plot. If the result does no further improvement or is just slightly better, the non-power normalization will still overweigh the power normalization method.

**Comb Filtering Application**

A comb filter is a digital filter that specifically removes a periodic signal. It can be an infinite impulse response (IIR) filter or a finite impulse response (FIR) filter. In the respect of calculation efficiency, the comb filter in this experiment is an IIR filter that is described as follows:

\[ H(z) = \frac{1}{1 - az^{-N_f}} \]  

(11)

Its frequency response spectrum has a series of harmonically related notches which allow signals that have frequency components at its peaks pass through the filter and leave other frequency signals retained. In a fetal ECG, both fetal and maternal signals are periodic and are differentiable in frequency response spectrum [5,6]. The maternal frequency generally falls in a range of 70-75 beats per minute, while the considerably smaller fetal component has a frequency in the range of 120-125 beats per minute [7].

Comb filtering is sensitive to the periodicity value, \( N_f \). A non-optimized periodicity value causes the comb filter to distort fetal signal and leave a portion of the maternal interference in the channel. This experiment tests the importance of the periodicity value in the filter setting by substituting various values of \( N_f \), including the proper value \( N_{f_p} = 468 \) and other values which are relatively close to it.

**Comb Filtering in Sequential Processing**

The adaptive filter sequential processing was proved to be effective by the previous work. The comb filter in the sequential processing could further refine the output from linear prediction and adaptive noise canceller. This experiment tests the performance of two sequential structures: 1) in-loop structure and 2) out-of-loop structure. The 1) in-loop structure sets comb filter as stage three processing and applies the filter in each iteration while the 2) out-of-loop only applies filter once as the final stage after
stage one and stage two finished all the process iterations. In the proper sequential structure, the basal drift should be removed and no additional noise should be introduced into the result.

Filter Type Selection for Comb Filtering
The result of synthetic signal as input indicates applying comb filter as the final step of the process produces a relatively clear fetal signal while there is still a small amount of periodical noise interfering the result with real signal input. This noise could be generated by phase shifting of the fetus signal during the filtering process since the periodicity of both signals are similar. Zero-phase comb filter is designed to minimize the phase shifting since it comb filters the signal twice with different operation direction. Experiment is conducted by comparing the performance of regular comb filter and zero-phase comb filter. In the cases of zero-phase comb filter, three different methods are also compared: 1) flipping the signal before comb filtering it, 2) comb filtering the signal before flipping it, and 3) using the `filtfilt` command in Matlab which function as zero-phase comb filter.

Adaptive Comb Filtering Application
An adaptive comb filter (ACF) is the comb filter that be able to adjust its filter coefficients. While regular comb filter has fixed filter coefficients and may not have the optimized performance for all signals. The ACF evolves its filter response to fit different signals so that, theoretically, gives a better performance compared to the non-adaptive comb filter.

ACF has two forms of adaptive structures: LPC and ANC. Block diagrams are shown in figure 6 and 7. In LPC structure, the input signal splits into two identical signals as reference signal (delay added) and desired signal while in ANC structure, the output from previous stage is the desired signal and the thorax signal is the reference signal.
To verify assumption that ACF works better, a FIR ACF is applied in the stage three and its transfer function is in the form of equation 10. By self adjusting the $\alpha$ values in each iteration while handling the same signal, the transfer function eventually transforms to the optimized form that specifically works for that signal.

There are two ways to adjust the transfer function: 1) adjusting one $\alpha$ value and updating the rest of the coefficients as powers of $\alpha$, 2) adjusting all $\alpha$s separately. The first method was expected to have a better performance since it preserves the comb filter structure.

*Adaptive Comb Filtering in Sequential Processing*

ACF sequential processing performance is evaluated by comparing the results of applying ACF in-loop and out-loop at different iteration. ACF processing is also compared with non-adaptive comb filter processing since the results of the latter filters contain echoes that cannot be removed by changing any parameters nor increasing iteration numbers. The ACF was expected to perform better and the way to apply it is determined in the experiment. Based on the result, further experiments were conducted and confirmed that stage 2 can be removed for computational simplicity.
CHAPTER 4: RESULTS

The Steepest Descent Optimization in LMS Algorithm

This experiment determines the optimized step size of steepest descent, $\mu$. Chart 1 in Appendix A, plots the frequency domain filter output signal from stage two with the $\mu$ varying from 1 to 50 while Chart 2 in Appendix A plots the time domain filter output with $\mu$ varying from 1 to 2000. While increasing the value of $\mu$, maternal signals are further suppressed but additional noise is also introduced into the system. Both filters eventually become unstable. Two filters have different optimized $\mu$ value: $\mu=5$ in frequency domain and $\mu=100$ in time domain.

Frequency Domain and Time Domain Adaptive Filtering

In this section, both frequency domain filter and time domain filters are tested and compared for 4 iterations. Chart 3 in Appendix A illustrates the comparison results which suggest frequency domain filters should used for future experiments. The results of frequency domain filter are listed in the left column and time domain filter's are listed in right column. The signal in frequency domain filter converges significantly faster and all the three fetal heart beat peaks are clear with ignorable maternal contamination. Fast convergence rate and clear fetal signal are not just indicators for a superior filter choice but also refine the input signal for the next iteration and next stage of process. The fact that the frequency domain filter works better than the time domain filter is the conclusion from this experiment.

Normalization

Power normalization tends to equalize all the peak amplitude which was expected to suppress maternal signal so that fetal signal would be exposed. The experiment runs both the frequency domain filter and time domain with power normalization and non-power normalization process. Figure 8, 9, 10, and 11 plots from the first iteration to the fourth iteration and the comparison is facilitated by plotting four outputs onto the same chart. The upper-left plots in each chart are the outputs of frequency domain non-normalized filters, the upper-right plots are for time non-normalized filters, the lower-left plots are
for frequency domain normalized filters, and lower-right plots are for time domain normalized filters.

Figure 8: Power Normalization and Non-power Normalization Comparison in Iteration 1
(Plot upper-left: Frequency Domain LMS, plot upper-right: Time Domain LMS, plot lower-left: Frequency Domain NLMS, and plot lower-right: Time Domain NLMS)

Figure 9: Power Normalization and Non-power Normalization Comparison in Iteration 2
(Plot upper-left: Frequency Domain LMS, plot upper-right: Time Domain LMS, plot lower-left: Frequency Domain NLMS, and plot lower-right: Time Domain NLMS)
Figure 10: Power Normalization and Non-power Normalization Comparison in Iteration 3
(Plot upper-left: Frequency Domain LMS, plot upper-right: Time Domain LMS, plot lower-left: Frequency Domain NLMS, and plot lower-right: Time Domain NLMS)

Figure 11: Power Normalization and Non-power Normalization Comparison in Iteration 4
(Plot upper-left: Frequency Domain LMS, plot upper-right: Time Domain LMS, plot lower-left: Frequency Domain NLMS, and plot lower-right: Time Domain NLMS)

All four figures also re-confirm the frequency domain filter has a better maternal interference elimination performance whenever in the case of power normalization or non-power normalization. The fetal peaks are not fully exposed from the maternal
interference by power normalization comparing to the otherwise case in both filters while the calculation time is significantly longer which leads to a conclusion that the power normalization is not worthwhile to perform in sequential processing since it adds computational cost while does not improve the system performance.

**Window Size Determination in Partial Power Normalization**

According to the explanation in Experiment Description section, power normalization technique still has its potential value in frequency domain filter. Revealed by figure 5, experiments could be conducted with different window size ranging from 64 tap weights down to 1 tap weight. Chart 4 in Appendix A lists 10 partial power normalization results but none of them proves the advantage of partial power normalization. The frequency domain non-power normalization is still the optimal method in sequential processing.

**Comb Filtering Application**

This section verifies that the comb filter is sensitive for the periodicity value which is not available on PhysioNet but can be easily calculated. Sample ecgca886, for example, is the testing data in this experiment has the periodicity value, denoted by $N_f$, 468, which is the time difference between 2 peaks of the fetal signal. By plugging the $N_f$ from 440 to 480 and comparing the result with the optimal value of $N_f = 468$, chart 5 in Appendix A indicates the significant performance difference between the optimal $N_f$ and other $N_f$ values. The periodicity value is a characteristic parameter for a cardio signal which means each data has its identity $N_f$ value. A fetal heat rate can be determined quite early in pregnancy by regular fetal ultrasound scans which means the value of $N_f$ is an accessible information for use in the sequential processing comb filter setting.

**Comb Filtering in Sequential Processing**

Stage three, comb filter, removes the basal drift and extracts the fetal signal out of the stage two output signal. However, it is important to determine the correct sequential position to apply it since pre-mature comb filtering cannot effectively extract fetal signal and may introduce noise into the system that worsens the next stage process. Since the optimized iteration number is 4, the following experiments test four cases that start
applying the comb filter from the first iteration to the fourth iteration. Figure 12, 13, and 14 are the plots for the in-loop structure and Figure 15 is for the out-of-loop structure.

**Figure 12:** Applying Comb Filter at the 1st Iteration  
(Plot upper-left: 1st Iteration with Comb Filter, plot upper-right: 2nd Iteration with Comb Filter, plot lower-left: 3rd Iteration with Comb Filter, and plot lower-right: 4th Iteration with Comb Filter)

**Figure 13:** Applying Comb Filter at the 2nd Iteration  
(Plot upper-left: 1st Iteration without Comb Filter, plot upper-right: 2nd Iteration with Comb Filter, plot lower-left: 3rd Iteration with Comb Filter, and plot lower-right: 4th Iteration with Comb Filter)
Both Figure 12 and 13 have noisy output in the last plot which means too much noise is introduced into the signal and fetal signals are hidden in the noise range. All figures
confirm that the comb filter effectively improves the signal right after it is applied into the system while only the out-of-loop structure provides the best waveform (figure 15) among the four. The basal drift is fully removed and noise level, though not fully removed, is controlled in an acceptable range.

Filter Type Selection for Comb Filtering

The previous experiment shows comb filtering process serves a good purpose of extracting fetal signal but there is still a certain amount of noise and inconsistent amplitude of fetal peaks. Since the noise is phase shifting of the fetus signal, zero-phase comb filter is a good option to recover the fetal signal.

Since a zero-phase comb filter is still theoretically a comb filter but processes the input signal twice with both direction. The order of flipping and comb filtering may bring about different results. By conducting the experiment described in the previous section, the result with signal ecgca886 in figure 16 indicates that flipping the signal before comb filtering it provides the best result over the other 3 cases. Results using more signals are shown in the Appendix B Section 1. With different samples and iteration counts, results show the potential of further improvement in stage 3 since the inconsistent amplitude issue is still not fully fixed and echo is introduced in some cases.

Figure 16. Comparison of 4 non-adaptive comb filtering techniques
Adaptive Comb Filtering Application

The non-adaptive comb filter just works fine for ecgca886 which happens to be a special case because in the case of other signals and other iteration numbers, echoes and noises are introduced. To push the performance limit of sequential processing mechanism, the third stage is loaded with ACF in order to automatically optimize the filter response. Two approaches are examined in the experiment: 1) adjusting the single filter coefficient, \( \alpha \), and update the rest of the coefficients with the power of \( \alpha \). 2) individually adjusting \( \alpha \)s. And both LPC and ANC structures are compared in order to determine an optimized configuration.

The result proves the previous assumption that method 1 would have a better performance inaccurate. Shown in the right plots in figure 17, the single alpha updating method fails to identify the correct signal since the aperiodic output means the system picks the wrong signal to process. In contrast, the left plots which is the multi-alpha updating provides 3 clear noise free peaks of fetal signal. The signal under processing is separated by 64 parts since there are 64 tap weights. Error is randomly embedded in the signal, in other words, may or may not in these 64 signal chunks.

Updating single alpha is using the information from one signal chunk only to update the rest 63 tap weights which is risky because the error, if exists, is amplified and the whole 64 tap weights are contaminated with error. While updating each of the alphas separately prevents single error been passed onto others. Each tap weights compensates each other and the error is over written. This makes the method 2, individually adjusting \( \alpha \)s, a better option.
The LPC structure, plotted as the lower plots in figure 17, provides narrower peaks which means the result is more accurate than the ANC structure (upper plots in figure 17). In addition, the noise is also controlled in a relatively low level in LPC structure.

*Adaptive Comb Filtering in Sequential Processing*

Same question for the ACF as for the non-adaptive comb filter, the position of the ACF in the sequential processing structure matters. The experiment runs both in-loop and out-loop structures and, not like non-adaptive comb filter, the results indicate ACF works better in the form of in-loop structure. Both adaptive and non-adaptive filter techniques including four filters (plot upper-left: regular comb filter, plot upper-right: zero-phase comb filter, plot middle-left: multi-alpha updating ACF in ANC structure, and plot middle-right: single alpha updating ACF in ANC structure, plot lower-left: multi-alpha updating ACF in LPC structure, and plot lower-right: single alpha updating ACF in LPC structure) are compared in Figure 18, 19, 20, 21, and 22. In order to let ACF reaches its optimized performance, experiment runs 7 iterations with non-adaptive comb filter applied at 7th iteration and ACF applied at 7th, 5th, 3rd, 2nd, and 1st iteration respectively.
Figure 18. Comparison of multi-alpha and single alpha updating ACF (ACF applied at 7th iteration)

Figure 19. Comparison of multi-alpha and single alpha updating ACF (ACF applied at 5th iteration)
Figure 20. Comparison of multi-alpha and single alpha updating ACF (ACF applied at 3rd iteration)

Figure 21. Comparison of multi-alpha and single alpha updating ACF (ACF applied at 2nd iteration)
From figure 18 and 19, zero-phase comb filter works better than others but starting from figure 20 to 22 shows multi-alpha updating ACF becomes the best option since the fetal peaks are more narrow and the echoes are mostly been removed.

In other words, when ACF is applied in-loop and less than 3 iterations, the non-adaptive zero-phase comb filter is the best option while ACF becomes better than it with more iteration of processes especially the ACF with LPC structure. More results with other test samples also confirm this conclusion. Plots are shown in Appendix B Section 2.

**Computational Simplification**

The results from previous section indicates the ANC is not as efficient as LPC. Stage 2 is also an ANC structure. Additional experiment is conducted with and without stage 2 to verify the necessity of noise canceller in sequential processing. The results are shown in figure 23 and 24, the upper 2 plots are with stage 2 process and the lower 2 plots are with stage 2 disabled.
Figure 23. Comparison of with and without Stage 2 (LPC-ACF)

Figure 24. Comparison of with and without Stage 2 (ANC-ACF)

For LPC structure ACF, the stage 2 does not improve the result too much but extra computational cost is added. In this case, stage 2 can be removed to maintain the high system performance in a more efficient way. Stage 2 does somehow improve the result for ANC structure ACF. However, LCP-ANC is proven to be better than ANC-ANC, which means stage 2 can be removed for computational simplification.
CHAPTER 5: CONCLUSION

The experimental results produced by this project confirm that adaptive filter sequential processing is able to successfully reduce maternal interference by sequentially applying adaptive linear prediction, adaptive noise cancellation, and comb filtering. It is not possible that all of the maternal signal and noise are always eliminated from the original signal, however a large portion of the interference is effectively suppressed. Current results indicate that the frequency domain adaptive filter (FDAF) has a better performance than the time domain adaptive filter (TDAF) in this application. The step size of steepest descent optimization is different for these two filters. The experimental results have also indicated that power normalization is not worthwhile to be performed in sequential processing due to the narrowband spectral content of both the maternal and the fetal signal components. The results also suggest that the “out-of-loop structure” for the comb filter provides better results in removing basal drift and introduces negligible noise into the system. The zero-phase comb filter is better than the regular comb filter in handling signal with phase shift while the multi-alpha updating ACF in LPC structure works even better after is applied in-loop for more than 3 iterations. Stage 2, ANC, can be removed for computational simplification purpose.
REFERENCES


Appendix A

Section 1: Filter Parameters Adjustment Tests

Chart 1: Sample Test for $\mu$ Optimization in Frequency Domain
Chart 2: Sample Test for $\mu$ Optimization in Time Domain
Chart 3: Frequency Domain and Time Domain Comparison
Chart 4: Window Size Comparison in Frequency Domain Partial Power Normalization
Chart 5: Periodicity Test in Comb Filter
Appendix B

Section 1: Comparison of Four Non-adaptive Comb Filters

(Plot upper-left: filter-flip zero-phase comb filter, plot upper-right: flip-filter zero-phase comb filter, plot lower-left: regular comb filter, and plot lower-right: zero-phase Matlab tool box comb filter)

Figure 25: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration)

Figure 26: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration)
Section 2: Comparison of Non-adaptive Comb Filters and ACF

(Plot upper-left: regular comb filter, plot upper-right: zero-phase comb filter, plot lower-left: multi-alpha updating ACF, and plot lower-right: single alpha updating ACF)

Figure 27: Sample 244, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration)

Figure 28: Sample 244, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration)
Figure 29: Sample 244, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration)

Figure 30: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration)
Figure 31: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration)

Figure 32: Sample 515, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration)
Figure 33: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration)

Figure 34: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration)
Figure 35: Sample 595, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration)

Figure 36: Sample 746, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration)
Figure 37: Sample 746, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration)

Figure 38: Sample 746, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration)
Figure 39: Sample 840, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 7th iteration)

Figure 40: Sample 840, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 4th iteration)
Figure 41: Sample 840, Comparison of Non-adaptive Comb Filters and ACF (ACF applied at 2nd iteration)