NTNU – Trondheim
Norwegian University of
Science and Technology

Automatic Extraction of Doppler Parameters for the Assessment of Fetal and Maternal Health

Agnes Margrethe Heyer

Electronics System Design and Innovation
Submission date: June 2014
Supervisor: Ilangko Balasingham, IET
Co-supervisor: Hans Torp, ISB
Gabriel Kiss, ISB

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

Automatic extraction of Doppler parameters for the assessment of fetal and maternal health

Reducing the mortality rate among unborns, young children and pregnant women is one of the Millennium Developments Goals (MDG 4 and 5) of the United Nations (UN). Diagnostic ultrasound is the only imaging method to be used in pregnancy and widely offered to the general population in developed countries. The Umoja project, ultrasound for midwives in rural areas, aims to develop an extremely low cost, robust and portable ultrasound imaging system (the Umoja ultrasound system) for obstetric imaging specifically designed for operation in challenging rural areas of developing countries. The project is a joint effort between three main partners: the National Center for Fetal Medicine (NCFM) at St. Olavs Hospital and NTNU, the Department of Circulation and Medical Imaging (ISB) at NTNU, and GE Vingmed Ultrasound AS.

As the Umoja system is intended for non-expert users, automatic parameter extraction is of importance in order to reduce intra- and inter- observer measurement variability. The main objective of this master thesis is to investigate the feasibility of automatically extracting clinically relevant parameters from the Doppler spectrum data. Typically used clinical parameters are listed below:

- fetal heart rate, the Doppler signal from a moving heart has a characteristic periodic signature which can be used to measure the heart rate. This is of special interest for fetal heart, where ECG is difficult to obtain.
- the Doppler velocity waveform
- peak systolic, end-diastolic and mean velocities
- pulsatility index, a known predictor of pre-eclampsia

The algorithms that are deemed feasible will be integrated on the tablet and tested at NCFM, both by experienced midwifes and non-experienced users with limited or no prior knowledge of ultrasound.

Summary

In developing countries, the availability of personnel with training in the use of ultrasound devices and the availability of conventional ultrasound equipment may be very limited. It is therefore beneficial to automate the diagnostic features of an ultrasound device.

The Umoja project is an initiative by The National Center of Fetal Medicine (NCFM) at St. Olavs Hospital and the Norwegian University of Science and Technology (NTNU). The technical part of this project involves the development of the Umoja ultrasound system, a tablet-based ultrasound machine with an intuitive user interface.

An automatic heart rate detection algorithm was developed in order to be implemented on the Umoja system in the future. The algorithm was developed using Doppler IQ data from two patients. Methods using only high frequencies as well as only low (tissue) frequencies were evaluated. Manual heart rate measurements were made in order to verify the accuracy of the algorithm.

The automatic calculations differed from the manual measurements on average up to one BPM, with a negative bias. When using only low frequent tissue data, the results were improved for heart data which had not been filtered by the scanner.

Most of the rejected calculations were found in segments with low power.

Based on the Matlab timing results and the accuracy of the heart rate calculations, the algorithm appears to represent a viable method for heart rate detection on a low-cost ultrasound device.

Sammendrag

I utviklingsland kan det være begrenset tilgang på personale med opplæring i å bruke ultralydutstyr, og tilgangen på konvensjonelt ultralydutstyr kan være svært begrenset. Det er derfor gunstig å automatisere diagnoseringsfunksjonalitet i en ultralydmaskin.

Umojaprosjektet er et initiativ fra Nasjonalt Senter for Fostermedisin (NSFM) ved St. Olavs Hospital og Norgest Naturvitenskapelige Universitet (NTNU). Den tekniske delen av dette projektet omfattet utviklingen av Umoja-ultralydmaskinen, en nettbrett-basert ultralydmaskin med et intuitivt brukergrensesnitt.

En automatisk algoritme for hjerteratedeteksjon ble utviklet for å implementeres på Umoja-systemet i fremtiden. Algoritmen ble utviklet ved hjelp av Doppler IQ-data fra to pasienter. Metoder som innebar bruk av utelukkende høyfrekvent innhold, samt utelukkende lavfrevent (vev) innhold, ble evaluert. Manuelle hjerteratemålinger ble gjort for å kunne verifisere nøyaktigheten til algoritmen.

De automatiske utregningene skilte seg fra de manuelle målingene i snitt opptil ett slag per minutt, med et negativt bias. Ved bruk av bare lavfrekvent vevsdata bedret resultatene seg for hjertedata som ikke hadde blitt forhåndsfiltrert av scanneren.

De fleste av de forkastede utregningene befant seg i segmenter med lav signaleffekt.

Basert på tidsanalyse i Matlab og nøyaktigheten til hjerterate-utregningene ser algoritmen ut til å utgjøre en brukbar metode for hjerteratedeteksjon for en lavkost ultralydmaskin.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fetal and maternal health assessment represent a considerable challenge in developing countries. In these countries, the availability of personnel with training in the use of ultrasound devices may be very limited.

With the introduction of affordable ultrasound hardware, medical ultrasound machines represent a portable and cost-effective means of reducing fetal and maternal mortality. It is therefore beneficial to automate the diagnostic features of an ultrasound device as much as possible so that the user may focus on diagnostic use of the equipment rather than the specifics of the particular ultrasound device.

1.1 The Umoja Project

The Umoja project is an initiative by The National Center of Fetal Medicine (NCFM) at St. Olavs Hospital and Department of Circulation and Medical Imaging (ISB) at The Norwegian University of Science and Technology (NTNU), in cooperation with GE Vingmed Ultrasound.

1.1.1 Background

NCFM has been involved in education and training of midwives in developing countries for the World Health Organization since 1997. The Umoja project is part of this activity, focused on teaching and training of obstetric ultrasound in the KwaZulu-Natal province of South-Africa.

1.1.2 The Umoja Ultrasound System

The development goal of the Umoja ultrasound system is to design a tablet-based ultrasound machine with an intuitive user interface. The Umoja system consists of scanner hardware combined with an Android tablet. The interface is touch-based, which is beneficial for creating an intuitive user experience.

1.2 Project Focus

Doppler ultrasound provides several parameters which can be extracted to assess fetal and maternal health. Parameters for diagnosing maternal health problems, such as the pulsatility index, can be of vital importance in countries where health care provision is limited. However, such a method would not involve a realtime implemntation. This project focuses exclusively on fetal health parameters, specifically the fetal heart rate, which can be extracted in realtime.

This project aims to design an algorithm for automatically detecting fetal heart rate from Doppler ultrasound, for later to be incorporated in the Umoja project. By automating the process of measuring the duration of the heart cycle, the operator can obtain this information in realtime. The accuracy of the resulting heart rate calculations will be considered.

The algorithm is intended to later be implemented on the Umoja ultrasound scanner. As the processing capabilities of an Android tablet is not specifically tailored to medical applications, unlike conventional ultrasound hardware, processing speed should be a consideration. Evaluating the processing speed of the algorithm and possible optimizations is relevant for this reason.

1.2.1 Previous Work

The algorithm described in this thesis is a continuation of previous work on a similar algorithm for the final-year specialization project at NTNU. The specialization project was smaller in scope, and mainly focused on the implementation described by Jezewski [6].

1.2.2 Review of Existing Methods

There are multiple methods for non-invasive fetal heart rate detection.

Implementations using autocorrelation of Doppler signals were described in detail by Hua and Jezewski [3, 6]. Jezewski modified the autocorrelation function by applying a triangular window and used a segment size with length adapted according to previous measurements.

In some cases, secondary analysis of the heart rate calculations are useful. Signorini and Kimura explores spectral analysis of the fetal heart rate variability, where Kimura uses wavelet transform for this purpose [4, 7].

There are also other uses for HR analysis, such as cardiac gating. Brekke describes a method for using tissue Doppler in order to synchronize 3D Doppler recordings to the same point in the heart cycle [8].

Some requirements for the resolution of heart rate measurements were suggested by Wickham and Voicu [9, 5].

Direct ECG was used in order to evaluate heart rate results of a few of the described algorithms [11, 10, 6]. A non-invasive method using ECG was also described by Karin [10].

1.3 Thesis Layout

Chapter 2 of the thesis details the most important theoretical knowledge used as a basis for the development of the detection algorithm.

In chapter 3, the different components of the algorithm are explained, along with the decisions involved in choosing the algorithm parameters and how they can be adjusted for optimal results. Some example plots from various stages of the algorithm are also shown for illustrative purposes.

Chapter 4 describes the acquired sets of Doppler data used in testing. Manual measurements were made in order to evaluate the automatic calculations. The step-by-step approach used in acquiring these measurements is described. The chosen set of parameters in the evaluation of the algorithm are also described.

Chapter 5 focuses mainly on the end results, which consist of segments of the recorded Doppler spectra, the corresponding mean frequency estimates and the resulting heart rate calculations.

In chapter 6, the results are discussed and evaluated. Suggestions for further development are made.

Chapter 7 sums up the findings and presents the advantages and disadvantages of the approach.

Chapter 2

Theory

2.1 Doppler Ultrasound

Doppler ultrasound utilizes the Doppler effect in order to measure the velocity of blood and tissue movement. The ultrasound probe emits an ultrasound beam with a frequency response determined by the probe design. When the beam is reflected by objects, the objects moving towards the ultrasound probe result in a positive frequency shift of the probe center frequency, whereas objects moving away from the probe result in a negative frequency shift. This shift modulates the envelope of the emitted pulse.

2.1.1 IQ Demodulation

In order to extract this Doppler shift, the negative frequencies of the real-valued Doppler signal are filtered out via the Hilbert transform. When the Hilbert transform of a signal is added to the signal (eq. 2.1), the real-valued signal becomes the complex pre-envelope of the signal. This makes it possible to represent each sample as a complex number with an in-phase (I) and quadrature (Q) component.

$$g_+(t) = g(t) + i\hat{g}(t) \tag{2.1}$$

The remaining signal is demodulated so that the center frequency is shifted down to zero. After demodulation, the spectrum consists of negative frequencies (the negative Doppler shift) and the positive frequencies (the positive Doppler shift). This makes it possible to distinguish between the negative and positive Doppler shift, as opposed to the real-valued signal received by the probe.

After demodulation, the IQ data is passed on to be further processed by the scanner software.

2.1.2 PW Doppler

Pulsed Wave (PW) Doppler is a method which uses a sampled ultrasound pulse in order to acquire a Doppler signal from a specific depth range. With a continuous wave (CW) Doppler signal, there is no way to know precisely in which region the scatterers are located. PW Doppler solves this problem, but also introduces aliasing. This means that the maximum measurable velocity is lower than for CW Doppler

Pulse Repetition Frequency

The PW Doppler pulses are emitted at intervals decided by the Pulse Repetition Frequency (PRF). The signal is also sampled at this rate, which means the PRF is equivalent to the sampling frequency f_s. Due to aliasing, the highest measurable Doppler frequency will be $^{\text{PRF}}/_{2}$. The Nyquist velocity is expressed in equation 2.2 [2], where f_t is the center frequency of the probe, c is the propagation velocity of sound in water, and θ is the angle of the probe relative to the movement of the scatterer.

$$v_{max} = \frac{\text{PRF}_{max} \cdot c}{4f_t \cdot cos(\theta)} \tag{2.2}$$

2.2 Spectral Doppler

In order to analyze the Doppler signal, frequency analysis is performed so that the signal can be displayed visually. In addition, the Doppler shift lies within audible frequencies and can be listened to by the user. The Doppler spectrum consists of low frequency components from tissue, high frequency components from blood, and a thermal noise floor.

2.2.1 Clutter

Clutter in a Doppler spectrum consists of low frequency components in the Doppler data, such as movement of tissue (e.g. blood vessel walls).

When the Doppler spectrum is generated, strong clutter frequencies will interfere with the higher frequency content of the spectrum when using short-time Fourier transform. The spectrum is generated by using time-limited windows of the signal, which results in sidelobes after transformation. A Hamming window has less pronounced sidelobes, but it does not solve the clutter problem.

Clutter Filtering

In order to ensure that the blood frequencies are visible in the spectrum, a clutter filter can remove the strong reflections from tissue. The clutter filter should have a short transition region to prevent blood frequencies from being filtered out.

2.2.2 Duplex Mode

In addition to the Doppler spectrum display, a 2D color Doppler image can be displayed alongside the spectrum so that the areas with the strongest blood flow can easily be found. A grayscale image is displayed underneath the color Doppler image. This enables the user to keep track of the location of the beam while acquiring data for the Doppler spectrum. However, by using different imaging modes simultaneously requires the probe to alternate between different types of pulses for each type of imaging mode. This leads to gaps in the Doppler spectrum. Therefore, the color Doppler image must be frozen prior to spectral Doppler acquisition, unless the gaps in the signal can be sufficiently estimated and substituted.

2.3 Mean Frequency Estimation

The Doppler spectrum discussed previously forms a two-dimensional image. In order to detect the heart rate from a one-dimensional signal, the mean frequency must be estimated.

2.3.1 Autocorrelation Method

The mean frequency can be estimated by using the autocorrelation estimator in equation 2.3 [1], where N is the number of samples to average and * represents complex conjugation. By increasing N, the estimator bias can be reduced.

$$R_N(1) = \frac{1}{N}\sum_{k=1}^{N} z(k+1)z(k)^* \tag{2.3}$$

The mean angular frequency is estimated as shown below [8].

$$\omega_N = \arg(R_N(1)) \tag{2.4}$$

2.3.2 Signal Envelope

Some of the existing algorithms which were explored used the envelope of the Doppler signal [6, 5, 7] instead of using frequency estimation. In one implementation, a real-valued Doppler signal, $A(n)$, is derived by using the Hilbert transform of the signal, $\hat{s}(n)$ (equation 2.5) and filtered with a lowpass filter with a cutoff frequency around 50 Hz [6].

$$A(n) = \sqrt{s^2(n) + \hat{s}^2(n)} \tag{2.5}$$

The Matlab `hilbert(xr, n)` [1] function uses an n-point FFT to compute the Hilbert transform.

2.4 Fetal Heart Rate Detection

Fetal Heart Rate detection algorithms have been available in obstetric ultrasound for decades.

The manual detection features in modern ultrasound machines commonly involve the manual selection of a recorded segment of the fetal Doppler spectrum. The heart rate is calculated based on the selection length averaged over the required number of heart cycles.

Automatic detection algorithms involve autocorrelation as a means of detecting the heart period. Another method, which was used on less capable hardware, is the average magnitude difference function (AMDF) [3].

[1] www.mathworks.se/help/signal/ref/hilbert.html

2.4.1 Clinical Significance

The heart rate can indicate a wide variety of fetal health problems, such as hypoxia or respiratory stress [4].

Fetal Arrythmia

Healthy fetal heart rate lies in the rage of 110-160 BPM [5]. Fetal arrythmia, such as tachycardia (above 160 BPM) or bradycardia (below 110 BPM), is suited for detection with an automatic algorithm.

Fetal Heart Rate Variability

Another parameter which may indicate fetal health issues is the fetal heart rate variability (FHRV). The FHRV is measured on a beat-to-beat basis, and a high FHRV measurement is considered a sign of good fetal health [9].

In order to sufficiently measure the FHRV, a heart rate detection algorithm should be able to achieve a resolution of 0.25 BPM, according to Voicu, whereas Wickham accepted a resolution of 1 ms [5, 9].

2.4.2 FHR Measurement Methods

In this project, the goal is to measure the fetal heart rate non-invasively. There are multiple approaches to achieving fetal heart rate measurements in a non-invasive manner. A completely manual approach involves the use of a fetoscope where the midwife can listen to the sound of the heart beat by placing the instrument on the abdomen of the mother. However, automated and more accurate methods are preferred.

Fetal Ultrasound

Ultrasound cardiotography measures fetal heart rate and uterine contractions. Initially, the fetal heart rate could be measured automatically by integrating the Doppler signal and using a level detector [11]. The second generation of devices began using autocorrelation methods as a means of detecting the heart period.

In addition to heart rate detection, spectral analysis of the FHRV may be performed using wavelet transform [7].

Instead of measuring the heart rate itself, heart rate detection may also be used to detect the same event in each heart cycle (cardiac gating), for instance so that several 3D ultrasound recordings of the heart may be synchronized in order to improve framerate. This can be achieved with tissue Doppler [8].

Fetal ECG

Indirect ECG uses the maternal ECG signal (abdominal electrocardiography, AECG) which is processed to separate the fetal ECG from the maternal ECG [10]. This is a non-invasive method.

Verification of Accuracy

Direct fetal ECG can be used as a way to compare the results from ultrasound detection to the heart rate [6, 11]. Direct FECG is acquired by placing scalp electrodes on the fetus, which is an invasive procedure and limited to the time of labor. Another invasive method is measurements of umbilical blood oxygen levels to compare fetal heart rate variability results [7].

Chapter 3

Fetal Heart Rate Detection Algorithm

3.1 Overview

The fetal heart rate detection algorithm was developed in Matlab. It operates partially on the full datasets and on segments. In a realtime implementation, the algorithm would be fully based on segments.

The main steps of the heart rate dection algorithm are shown in figure 3.1.

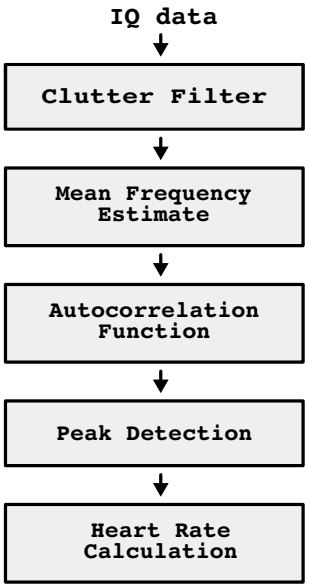

Figure 3.1: The main components of the algorithm

The first step involves the selection of the parts of the spectrum to be considered by filtering the IQ data. The resulting spectrum should be periodic in time.

The next step is to estimate the mean frequency. This estimated frequency will be used in the autocorrelation step in order to detect the heart period.

The autocorrelation function calculated from the frequency estimate segment is used in a peak detection step. The location of the detected peak yields the period in samples.

Finally, the heart rate is calculated using the sampling frequency and the peak location in samples. This heart rate is displayed and color coded based on the corresponding cost function.

The first steps are going to have the most influence on the end result.

3.2 Mean Frequency Estimation

3.2.1 Clutter Filtering

The IQ-signal is highpass filtered so that low-frequency components from tissue movement (clutter) are removed.

A Chebychev filter was designed using the `filterbuilder` tool in Matlab and used as clutter filter. The clutter filter requires a narrow transition region so that the clutter may be removed without also removing blood frequencies in the process.

The filter was assigned a filter order of 10 in order to have a stopband attenuation of 80 dB.

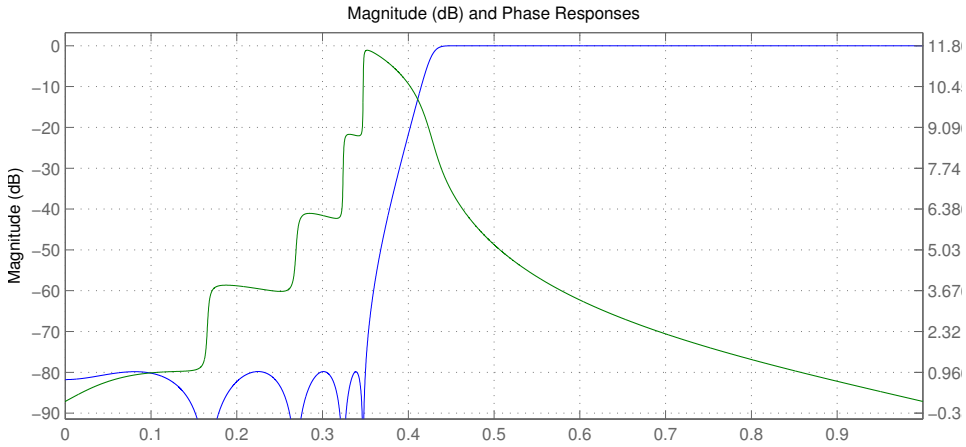

Figure 3.2: Chebychev clutter filter, magnitude (blue) and phase response (green)

3.2.2 Using Clutter

When using a lowpass filter to only consider the low frequency content of the heart signal, the viability of this approach will be limited to heart signals.

After clutter filtering, the remaining heart signals contain several peaks per cycle. In theory, up to six different peaks may appear due to different types of heart contractions and valve movement, but in a recorded Doppler signal there will be fewer peaks [5]. In the remaining Doppler spectrum shown in figure 3.3 contain two positive peaks and two negative peaks.

As the peaks in the Doppler spectrum appear in a periodic fashion, the correlation will also peak at periodic intervals where two beats are correlated with each other. In addition, the peaks will correlate to a lesser degree with each other, resulting in additional, smaller correlation peaks.

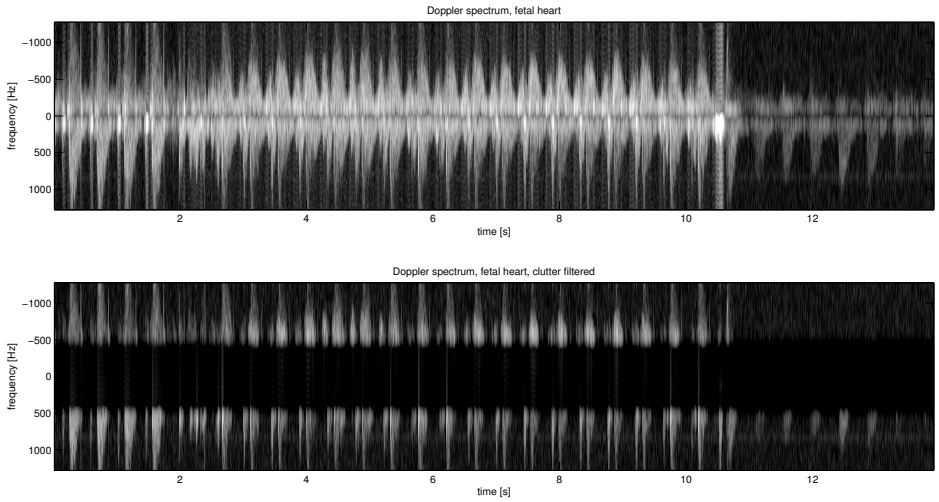

Figure 3.3: Original spectrum (top) and clutter filtered spectrum (bottom), fetal heart, patient A

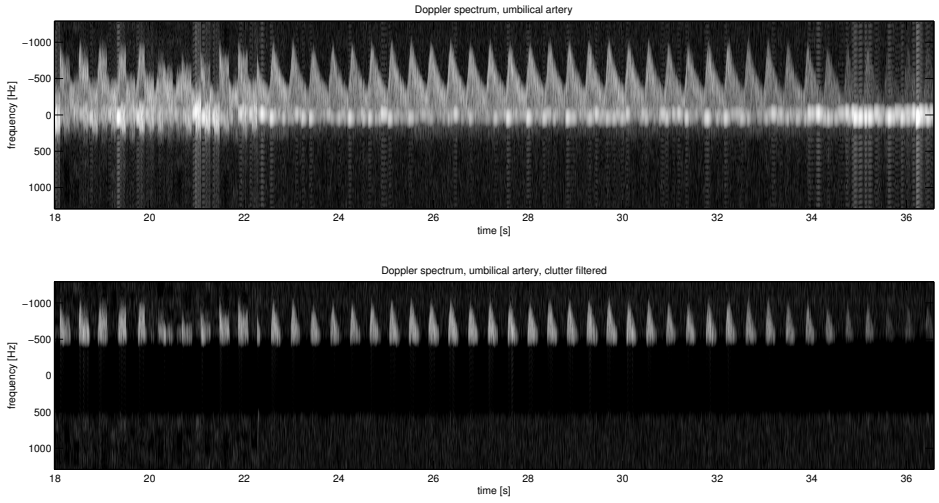

Figure 3.4: Original spectrum (top) and clutter filtered spectrum (bottom), umbilical artery, patient B

The recorded data from the umbilical artery results in a much simpler signal after clutter filtering (see figure 3.4), compared to the heart spectrum (figure 3.3). Each cycle of the remaining signal consists of a single, clearly defined peak.

3.2.3 Reducing Estimator Bias

The mean angular frequency estimate is the average of N samples, as described in section 2.3.1. This can be implemented as a moving average FIR filter, which can be optimized by using FFT.

Realtime

For the segment approach, which will be more practical in a realtime implementation, using a ring buffer is suitable. The ring buffer will allow for the same result as when using a continous dataset.

With a ring buffer, the summation of the samples does not need to be fully repeated for each calculation. The sum can be recalculated for each step by subtracting the oldest sample and adding the newest sample within the averaging length.

3.2.4 Power Thresholding

When the IQ-signal is high pass filtered, blood velocities which lie below the cutoff frequency will also be filtered out. In segments where the bloodflow frequencies lie in the stopband, only thermal noise will remain after filtering. This will result in noisy angular frequency estimates in these segments. In order to prevent this, the signal is thresholded to remove problematic segments.

Power Estimate

A logarithmic power estimate is calculated as follows:

$$P_N = \frac{20}{N} \sum_{k=1}^{N} log_{10}(z(k)) \tag{3.1}$$

This method produces a smooth estimate which is suitable for thresholding.

Samples in the filtered IQ-signal are zeroed if they correspond to the samples in the power estimate where the power lies below a set threshold.

The power estimate is averaged in the same way as the mean frequency estimator. By keeping the noise in the power estimate low, the thresholding will also be more stable, instead of alternating rapidly between rejection (zeroing) and the accepted frequency estimate values.

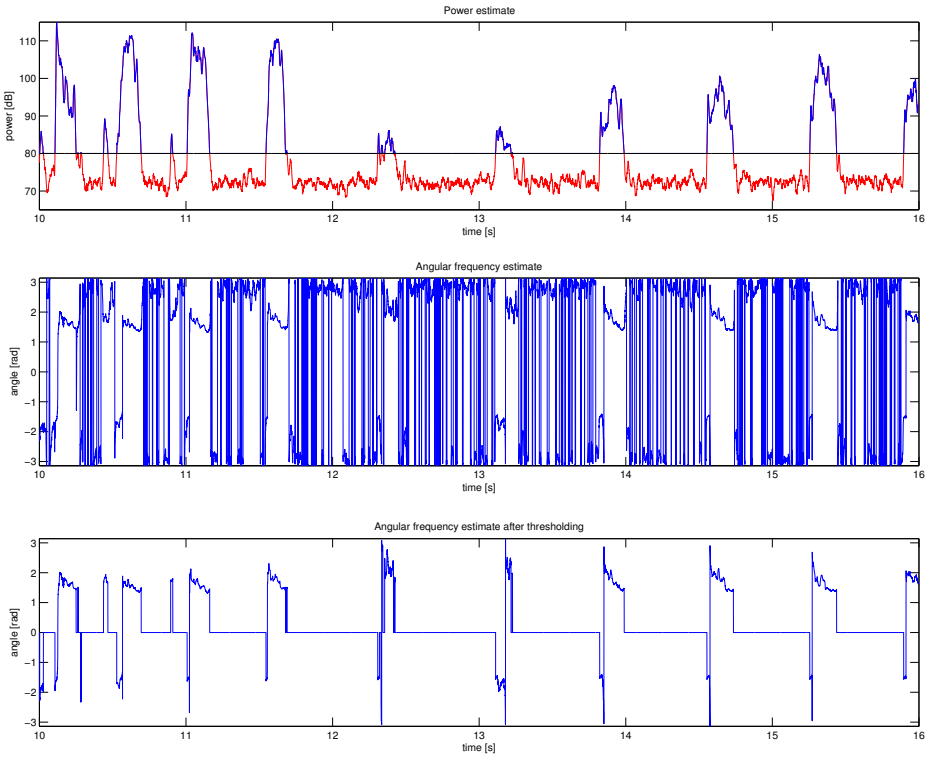

Figure 3.5: Power estimate with 80 dB threshold (top) and mean frequency estimates before thresholding (middle) and after thresholding (bottom)

Figure 3.5 shows the frequency estimate of the signal before and after thresholding based on the power estimate. The estimate contains a lot of noise in the segments below the threshold, shown in red. The noisy segments where the spectrum contains mainly thermal noise result in a noisy frequency estimate. After thresholding, the periodicity in the estimate is more easily identified.

Signal Power vs. Threshold

If the threshold is set too high, the thresholding step is more likely to remove useful parts of the estimate. The acquired datasets in this case had power estimate troughs of 70 dB and peaks around 100-110 dB. Dataset A7 is shown in figure 3.5 and was the dataset containing some of the weakest peaks (85 dB peak at the lowest). A segment of B6 also contained some peaks around 85 dB. This needs to be considered when choosing the threshold level. In this case, choosing a threshold of 85 dB will remove the weakest peaks altogether, whereas a threshold level below 70 dB is ineffective.

Realtime

The power estimate is averaged the same way as the frequency estimate. A ring buffer solution should work equally well in this case. The power threshold is a constant value which does not change over time.

3.3 Autocorrelation Function

3.3.1 Decimation

The signal is decimated in order to reduce the number of samples used for each autocorrelation step. As the decimation factor is increased, the accuracy of the results will be reduced. As there will be a trade-off between accuracy and processing speed, the choice of decimation factor will depend on the implementation and the capabilities of the hardware.

Lowpass Filter

In order to prevent aliasing in the decimation step, the signal can first be lowpass filtered. The cutoff frequency of the filter is f_{Nyquist}/M, where M is the decimation factor. However, this filter is not needed if there is sufficient averaging of the estimate, as averaging will also act as a lowpass filter.

If the lowpass filtering is done after the thresholding, there will be more ripples in the estimate due to filtering of the transition between the original signal and the thresholded parts. This will lead to changes in the autocorrelation function and lead to less accurate results. Therefore, anti-aliasing should take place prior to thresholding, if needed.

Decimation Factor

The conversion from heart period in samples to heart rate in frequency is non-linear, and the heart rate resolution is lower for higher heart rates. The decimation will further affect the resolution of heart rates in the higher end of the range more than the lower end. If fetal heart rate variability is a desirable feature, decimation may lead to inaccurate results.

The optimal choice is to avoid decimation altogether. However, this will result in a large segment size. For instance, if the PRF is 2500 Hz and the segment length is two cycles at 70 BPM, the segment length will be 4286 samples, as calculated from equation 3.2.

$$\frac{samples}{segment} = \frac{\frac{cycles}{segment} \cdot 60 \cdot \frac{samples}{second}}{BPM} \tag{3.2}$$

3.3.2 Cycles Per Segment

The segment used for each autocorrelation step needs to contain at least two heart cycles in order to correctly detect the distance between two heart beats. If the algorithm should be able to correctly measure all heart rates in the chosen range of heart rates, the segment length must be able to contain at least two cycles of the lowest detectable heart rate.

If the maternal heart rates should be detectable, the segment length should accommodate this. For instance, if the segment is too short, the maternal segment may come close to only containing a single heart beat. At that point, thresholding becomes important in order to avoid noise which may result in correlation peaks larger than the heart period peak.

By choosing longer segment lengths, the maternal heart rate can be correctly detected. However, this increases the number of samples for autocorrelation. Another, but minor, drawback is that it will take longer for the algorithm to adjust to changes in heart rate.

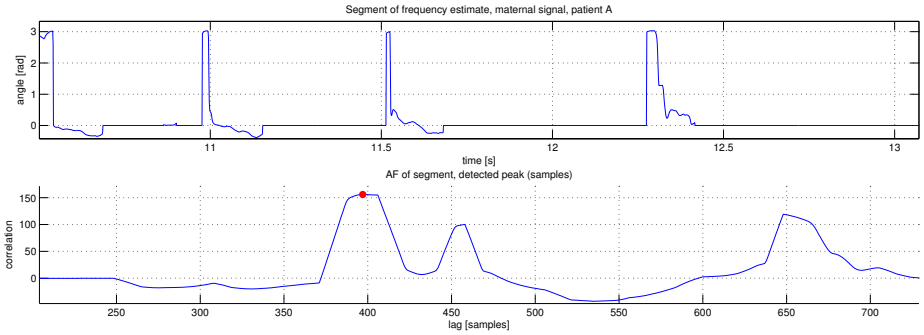

Figure 3.6: Frequency estimate and autocorrelation function, transition from fetal heart cycles to maternal heart cycles, detected peak marked with red.

In figure 3.6, the frequency estimate initially contains fetal heart cycles, but switches to maternal heart cycles at the 11-second mark. The autocorrelation function will contain three peaks, one for the fetal heart cycle, one for the transition which contains a partial maternal heart cycle, and the maternal heart cycle. The shortest period is more likely to be detected, as this will be more likely to appear multiple times in the segment. With a shorter window, the algorithm will quickly detect the switch, but may not correctly detect the maternal heart rates.

Static Segment Size

A static segment size is used, as it is more suited for realtime application. The amount of samples used for each autocorrelation step is constant, which means the processing time will also be predictable.

Step Size

The step size is determined by the amount of desired calculations per cycle. The location of the next segment will be advanced by the following number of samples:

$$\text{step} = \frac{f_s}{\text{calculations per second}} \tag{3.3}$$

3.3.3 Autocorrelation

A segment is selected for autocorrelation calculation. The autocorrelation needs only be calculated for lag values $l \in [0, \text{maxcorr}]$, where *maxcorr* is the maximum detectable heart period in samples. The correlation at lag zero is the energy of the signal. The segment used for peak detection lies within the range $l \in [\text{mincorr}, \text{maxcorr}]$, where *mincorr* is the minimum detectable heart period in samples.

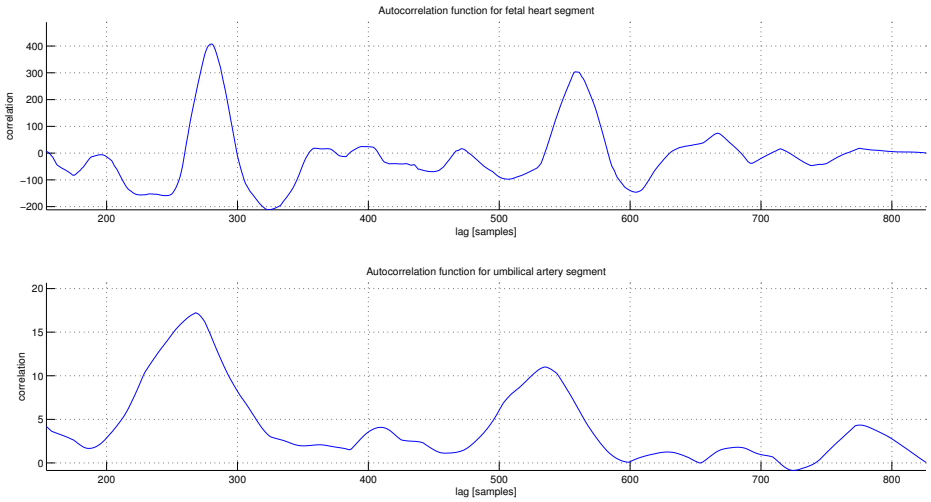

Figure 3.7: AF for heart segment (top) and umbilical segment (bottom), patient B

Figure 3.7 shows autocorrelation functions for the two types of acquisition targets. In both cases, there is one dominating peak for each heart cycle, and peak detection resulted in accurate heart periods.

Heart Rate Range

The *mincorr* and *maxcorr* values are calculated based on the range of heart rates the algorithm should be able to detect, as shown in equations 3.4 and 3.5.

$$\text{mincorr} = \frac{60 \cdot f_s}{\text{BPM}_{max}} \tag{3.4}$$

$$\text{maxcorr} = \frac{60 \cdot f_s}{\text{BPM}_{min}} \tag{3.5}$$

By limiting the detectable heart rate, the number of calculations in the autocorrelation step will be reduced. However, if the true heart rate lies outside of this range, it will not be correctly detected. Extremely low or extremely high heart rates are not physically possible and should always be eliminated, as peaks in these ranges may still occur and take priority over the correct peak.

Autocorrelation Bias

In a biased autocorrelation function, the correlation values decrease with lag. By using an unbiased autocorrelation function, the peaks in the function will have similar maximum values, rather than peak values tapering off as the lag increases. The biased autocorrelation used in this algorithm is shown in 3.6.

The signal used in the autocorrelation step is a real signal, which eliminates the need for complex conjugation.

$$R_{xx} = \frac{1}{N} \sum_{k=1}^{N} x(k)x(k-1) \tag{3.6}$$

When detecting a single peak, using the unbiased autocorrelation proved counterproductive. The algorithm is more likely to favor detection of peaks for lower lag values when using the biased autocorrelation. If multiple peaks are to be detected, the removal of bias will be beneficial, as the peaks will be more similar throughout the AF.

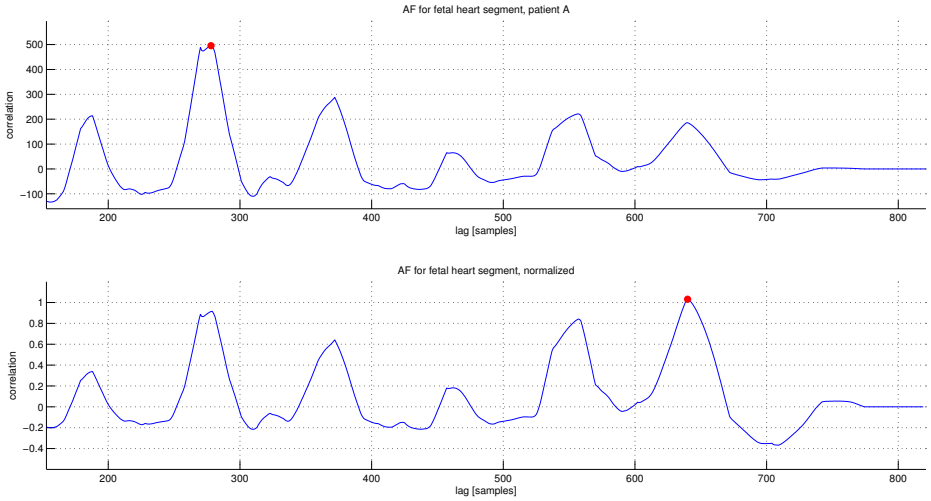

Figure 3.8: Autocorrelation function unscaled (top) and unbiased (bottom) with detected peak shown in red

Windowing of AF

The autocorrelation function will show the highest correlation values for low lag values. Previously, the peak detection was a simple detection of the maximum value. In that case, it was necessary to dampen the autocorrelation for low lag values, as it could affect the detected maximum. A triangular window was applied to the AF to dampen less likely heart rates. This method has the benefit of dampening additional peaks in each heart cycle, specifically in the heart signals.

A simpler approach is to use a rectangular window, although this also requires peak detection which can rule out any maxima detected at the very start of the segment, midslope. This will prevent detection of positions too early in the autocorrelation function. The rectangular window discards lag values outside of the chosen window, which means that the chosen range will have an impact on the possible BPM values which can be calculated.

3.3.4 Cost Function

The cost function should indicate the accuracy of the calculations so that it can be used as a visible indicator for the user to see if the signal acquired is suitable for heart rate calculation.

The cost function is implemented as the ratio between the maximum value of the autocorrelation function divided by the signal energy. This is the same method used in the previous implementation, as shown below.

$$\text{AFR} = \frac{R_N(0)}{R_N(l)_{max}} \tag{3.7}$$

3.3.5 Peak Detection

Peak detection is performed on the AF in order to find the duration of the heart period in samples. The autocorrelation function will ideally peak at the location where the lag matches the heart period.

The segments used to calculate the AF are assumed to contain three heart cycles. This means that the autocorrelation function will contain several large peaks. Without normalization, these peaks will taper in value with increasing lag.

In this algorithm, the peak with the maximum correlation value is chosen.

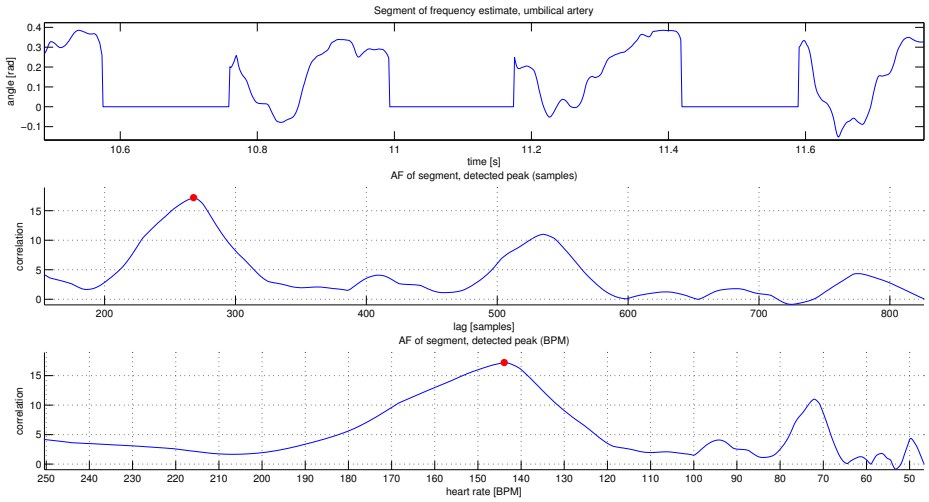

Figure 3.9: Frequency estimate segment (top), AF with peak detection (middle) and FHR conversion (bottom)

In figure 3.8 in section 3.3.3, there are two visible peaks per cycle. In figure 3.9, which is from the umbilical artery, there is only one clear peak per cycle. This makes it easier to detect the peak correctly.

3.4 Heart Rate

The heart rate is calculated based on the detected heart cycle in samples, the lag value of the correlation peak l, as show in equation 3.8.

$$\text{BPM} = \frac{60 \cdot f_s}{l} \tag{3.8}$$

3.4.1 Cost Representation

In order to convey cost information to the user, a cost function threshold value is chosen so that calculations corresponding to cost values below the threshold can be marked as inaccurate. A color is assigned to the output values based on the cost function. If the cost is low, the assigned color is green. Otherwise, the color is red.

A third color, in this case yellow, is used as an indicator of stability. A green value is colored yellow if any of the n previous values were colored red. This tells the user that the

algorithm is waiting for a stable signal. False detections may occur, sometimes consisting of several consecutive values.

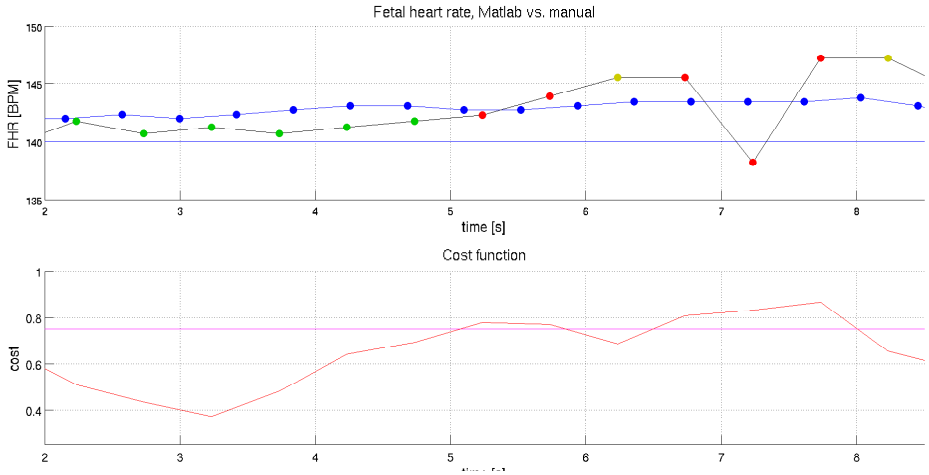

Figure 3.10: The bottom plot shows cost function (red) with threshold (magenta). Dots in top plot are green if cost lies below threshold, red if above.

An example of a cost function segment is shown in figure 3.10, along with the corresponding color coding for the heart rate calculations, where $n = 1$ and the threshold for red is 0.75.

By assigning the yellow color to the first n values following a rejected value, the green values will be limited to the more stable segments. The yellow dots in figure 3.10 would otherwise be green.

3.5 Acquisition Target Considerations

The frequency content of the Doppler spectrum varies depending on the acquisition target, which means that the parameters used in the algorithm may be adjusted to improve results for the different acquisition targets if only one of them is considered.

3.5.1 Filter Method

When deciding on a filter method prior to frequency estimation, the clutter content of the signal should first be considered. The fetal heart signals are suited both for detection after lowpass filtering and highpass filtering, as they contained strong clutter components and high frequent components (high blood velocity). The umbilical artery signal was unsuited as it contained a much weaker clutter signal.

3.5.2 Clutter Filter Cutoff

The fetal heart signal contains frequencies covering a larger spectrum band. This means that the heart signal can handle a higher cutoff frequency for the clutter filter. For instance, the fetal heart datasets can still yield accurate results with a higher cutoff frequency than the umbilical dataset. The umbilical dataset in this case contained peaks at around 1000

Hz (roughly 80% of the Nyquist frequency), whereas the heart data contained frequencies past the Nyquist frequency (aliasing). If the filter stopband ends at 1000 Hz, there will be no accurate umbilical heart rate calculations, whereas the heart data will still yield accurate results provided sufficient signal power.

Chapter 4

Validation

4.1 Data Acquisition

In order to test the algorithm, Doppler data was recorded on two separate occasions. The datasets varied in both in acquisition quality and content.

4.1.1 Instrumentation

All sets of Doppler IQ data used in this project were recorded using a GE Vingmed Vivid E9 ultrasound scanner. The data was exported as DICOM[1] files, from which the Doppler IQ data was extracted using a Matlab script.

4.1.2 Datasets

The dataset from patient A consisted exclusively of fetal heart data. The dataset from patient B consisted of data both from the fetal heart and the umbilical artery. Some of the datasets from patient B had to be discarded, as the color Doppler mode was still live during the acquisition of the Doppler spectrum during parts of the acquisition.

The viable datasets (table 4.1) consisted of five datasets from patient A, one of which contained maternal heart frequencies. In addition, there were two viable datasets acquired from patient B, namely one fetal heart dataset and one umbilical artery set.

Dataset	Target
A2	Fetal heart
A3	Fetal heart
A4	Fetal heart
A6	Fetal heart
A7	Maternal
B6	Fetal heart
B10	Umbilical artery

Table 4.1: Content of datasets

[1]Digital Imaging and Communication in Medicine standard format

Neither patient was reported to have any health issues relevant to the resulting acquisitions.

The data from patient A appear to have been clutter filtered when compared to the data from patient B where there is a substantially higher amount of clutter. As shown in the data segments in figure 4.1, the higher heart frequencies of the bottom spectrum are obscured.

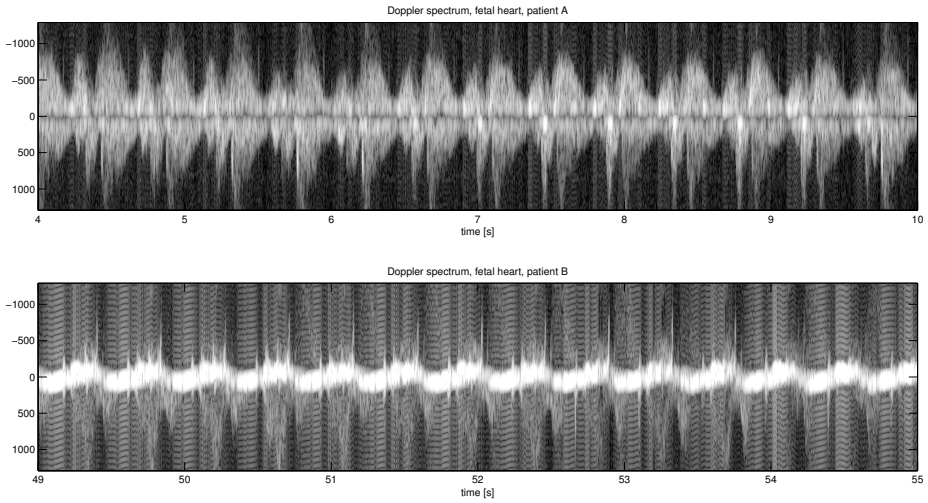

Figure 4.1: Comparison of fetal heart data from patient A (top) and patient B (bottom)

4.1.3 Doppler Parameters

The sampling rate is needed when calculating the heart rate from the number of samples. The PRF used by the scanner could be extracted from the exported DICOM files.

The PRF values are listed in table 4.2.

Dataset	f_s [Hz]
A2	2550
A3	2550
A4	2550
A6	2550
A7	2550
B6	2572
B10	2572

Table 4.2: Sampling frequencies/PRF

4.2 Manual Heart Rate Measurements

Manual heart rate measurements were made by manual inspection of the Doppler spectrum of each data set. Originally, the manual measurements were made entirely with Matlab by plotting the spectrum in a figure window and recording the mouse input coordinates within the image. However, as the horizontal resolution varied depending on the

length of the spectrum, a method which resulted in more consistent resolution was used, as described below.

- The IQ data was lightly clutter filtered in Matlab to reduce sidelobe noise while retaining as much of the frequency content as possible.
- A frequency spectrum was generated from the filtered IQ data using an FFT length of 256, Hamming window lengths of 32 and 75% overlap between windows.
- Each spectrum was then exported as a grayscale PNG image file.
- The images were examined in Gimp, an image editing software, where the x-coordinates (position in time) of the mouse cursor were manually noted once for each period of the heart cycle.
- The resulting coordinates were then input into a Matlab script in order to calculate the heart rate.

To evaluate the automatic algorithm, the manual measurements were averaged over three heart cycles. Each heart rate value was calculated using the time difference between an interval of three sample points, Δt. To achieve a central moving average, the sample is plotted at the time point in the center of the interval, t_s.

$$\Delta t = t(n+3) - t(n) \tag{4.1}$$

$$t_s = \frac{t(n) + t(n+3)}{2} \tag{4.2}$$

As the resulting calculated heart rate is non-linear, the detected heart periods were returned along with the heart rate calculations for later comparison.

4.2.1 Manual and Automatic Comparison

The manual measurements were linearly interpolated in order to compare the manual results with the results from the Matlab algorithm.

Mean Difference

The mean and standard deviation of the difference between the samples was calculated. The difference between each automatic heart rate measurement $m(k)$ and the interpolated manual measurement $a(k)$ was defined as $d(k) = m(k) - a(k)$.

$$\bar{d} = \frac{1}{N}\sum_{k=1}^{N}(m(k) - a(k)) \tag{4.3}$$

$$\sigma_d = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(d(k) - \bar{d})^2} \tag{4.4}$$

The mean difference is the bias of the algorithm output.

Viable Segments

Most of the evaluated dataset segments were less than five seconds in length. With the exception of the umbilical dataset, the majority of the remaining sets contained stable, high power signals for roughly one third of the recording.

Dataset	Total length [s]	Selected interval [s]	Interval length [s]
A2	15.7	5.5 – 9.5	4.0
A3	15.8	7.4 – 10.5	3.1
A4	15.0	1.7 – 6.0	4.3
A6	13.9	4.8 – 7.3	2.5
A7	16.0	12.5 – 15.0	2.5
B6	24.2	43.9 – 52.0	8.1
B10	18.6	23.2 – 33.8	10.6

Table 4.3: Time intervals for comparison of manual and automatic results

4.3 Timing and C++ Comparison

A stand-alone C++ implementation of the previous algorithm was developed. This implementation contained frequency estimation, decimation and calculation of the autocorrelation function with a triangular window applied.

All timing results were sourced from a laptop with an Intel Core i3-2310M CPU, 2 x 2.10 GHz, running 64-bit Linux Mint. The C++ implementation was timed using the Unix `time` command[2].

[2]http://unixhelp.ed.ac.uk/CGI/man-cgi?time

Chapter 5

Results

The algorithm was applied to the acquired datasets and the difference between the manual and automatic results was calculated. A combination of parameters which yielded accurate results was found, and some of the parameters were changed to see the effect on the end result compared to the original configuration.

5.0.1 Power Threshold

The effects of the power thresholding varied considerably between the datasets. The maternal heart rate segment was most sensitive to changes in the power threshold level, while the two sets from patient B were least affected.

5.0.2 Decimation Factor

The difference between consecutive heart rate calculations was calculated in order to find the lowest time resolution, ΔT, shown in table 5.1. As the heart rate conversion is non-linear, the heart rate resolution, ΔBPM, was calculated for a specific heart rate. As the fetal heart rate variability is dependent on resolution, the chosen heart rate was normal fetal heart rate (140 BPM).

$$\Delta\text{BPM} = \frac{60}{T_{140} - \frac{\Delta T}{2}} - \frac{60}{T_{140} + \frac{\Delta T}{2}} \tag{5.1}$$

Decimation factor	Time resolution [ms]	FHR resolution [BPM]
-	0.392	0.13
2	0.784	0.26
3	1.176	0.38
4	1.569	0.51

Table 5.1: Time resolution when using a PRF of 2550 Hz

5.1 Calculated Fetal Heart Rate

Averaging of the automatic heart rate values was deactivated in order to more clearly see the variation in the calculations. This makes it easier to see the effect of low power signals.

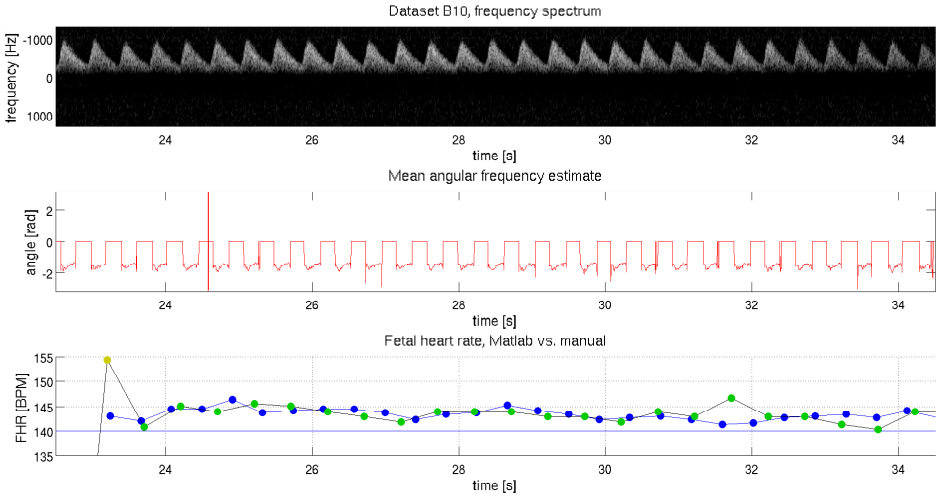

Figure 5.1: Heart rate calculation from umbilical data, patient B

The frequency estimate for the umbilical data, shown in figure 5.1, is the most robust of the data sets with regards to parameter changes, with the exception of methods using low frequency content.

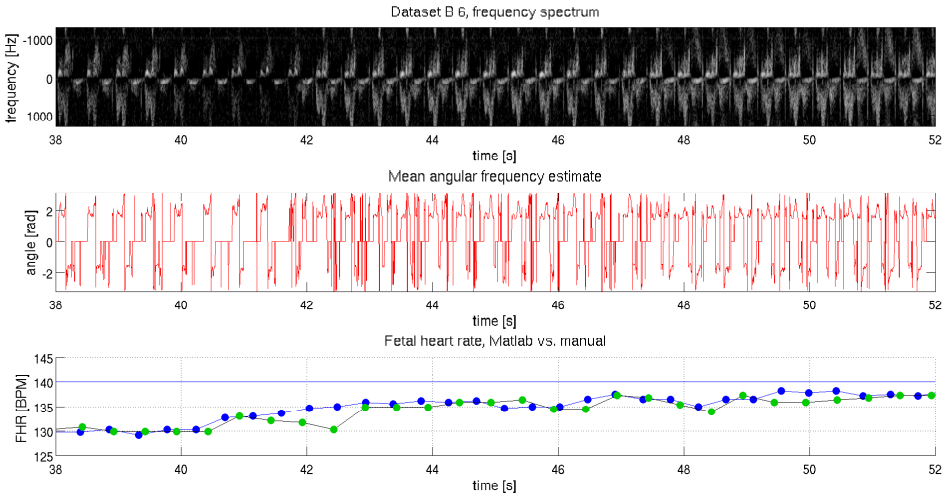

Figure 5.2: Heart rate calculation from fetal heart data, patient B

The fetal heart rate shown in figure 5.2 matches the manual measurements fairly well for most of the selected segment. However, in the interval from 40 to 43 seconds, the signal power is reduced and parts of the frequency estimate is zeroed by the thresholding. This leads to a larger variation in the heart rate values.

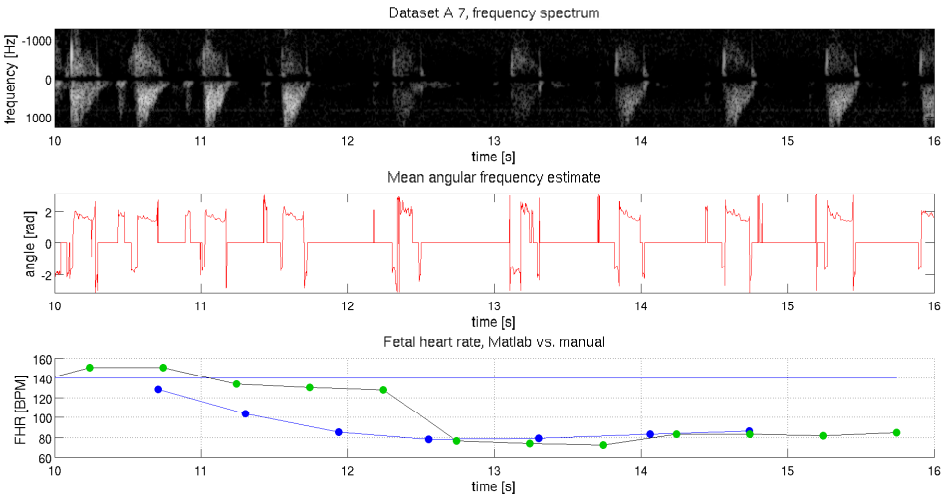

Figure 5.3: Heart rate calculation, maternal heart rate, patient A

In figure 5.3, the signal switches from a fetal heart signal (around 140 BPM) to a maternal heart signal (around 80 BPM). The manual measurements shown were averaged, which leads to a larger mismatch between the manual measurements and the automatic measurements during the transition.

5.1.1 Cycles Per Segment

The choice of segment length was important when detecting the maternal heart rate. As shown in figure 5.4, the selected segment is too short for detection of the correct heart cycle period, marked as a red dot where detection fails. When the segment length is increased to two cycles (for an expected heart rate of 70 BPM, equivalent to four cycles at 140 BPM), the detection succeeds.

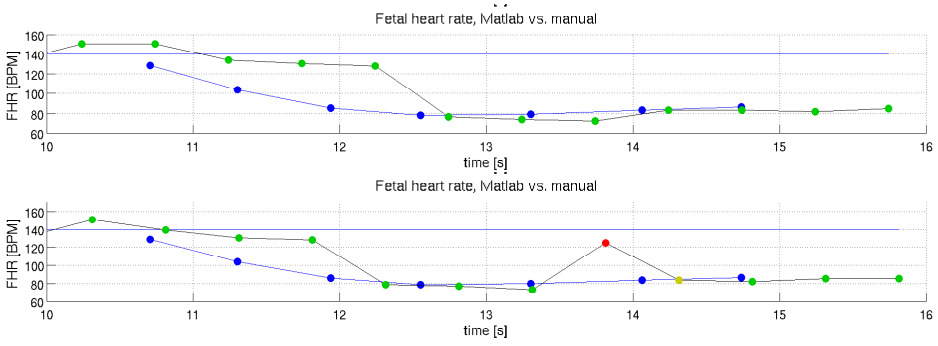

Figure 5.4: Maternal heart rate with a power threshold of 75 dB and 1.5 cycles at 70 BPM (bottom) and two cycles (top)

Another difference between figure 5.3 and figure 5.4 is that the algorithm detects the transition from fetal to maternal heart rate sooner when the segment length is shorter than when it is longer.

5.1.2 Accuracy

The difference between the manual and automatic measurements for the most stable segments are shown in table 5.2. The manual measurements were averaged to better match the manual measurement method which would have been used by the midwife to make manual measurements during an examination. In order to more easily identify problematic segments, there was no averaging of the automatic measurements.

Dataset	Mean difference [BPM]
A2	0.66 ± 2.15
A3	0.76 ± 3.40
A4	-1.14 ± 1.02
A6	-1.06 ± 1.49
A7	-2.04 ± 3.13
B6	-0.57 ± 1.05
B10	-0.10 ± 1.64

Table 5.2: Comparison of manual measurements and automatic measurements with 75 dB threshold, 32-point averaging, decimation factor 4, clutter filtered

Most of the sets have a slight negative bias where the automatic measurements underestimate the heart rate.

Dataset	Mean difference [BPM]
A2	1.46 ± 2.23
A3	2.84 ± 1.13
A4	-8.38 ± 23.53
A6	-0.04 ± 1.17
A7	-2.18 ± 2.91
B6	0.09 ± 0.81
B10	0.60 ± 1.13

Table 5.3: Comparison of manual measurements and automatic measurements with 75 dB threshold, 32-point averaging, no decimation, clutter filtered

Table 5.3 shows the difference between manual and automatic measurements when there is no decimation. The large variation in dataset A4 is a result of a noisy segment which has several peaks in its autocorrelation function (see figure 5.5) which almost have the same correlation. This segment varies between successful detection and failure depending on small variations in the maximum value of the three peaks.

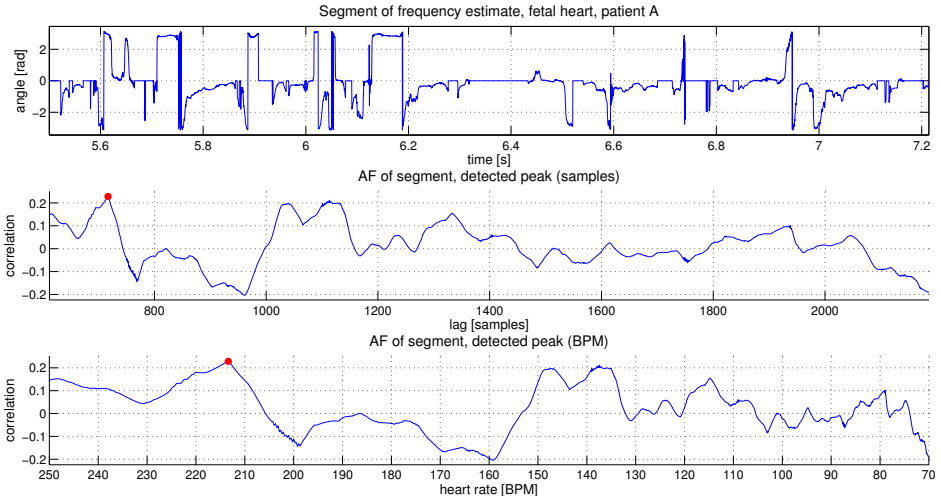

Figure 5.5: Noisy segment from dataset A4, no decimation

5.2 Tissue Movement

To test how the algorithm performed using tissue frequencies, the clutter filter was replaced with a 10th order Butterworth lowpass filter with a cutoff frequency of 0.04 times Nyquist. The thresholding used after the clutter filter was still useful following this filtering approach.

Dataset	Mean difference [BPM]
A2	0.92 $\pm$ 0.76
A3	5.68 $\pm$ 11.18
A4	-1.63 $\pm$ 2.42
A6	-6.02 $\pm$ 17.15
A7	17.19 $\pm$ 38.29
B6	-0.12 $\pm$ 1.05
B10	26.88 $\pm$ 32.86

Table 5.4: Comparison of manual and automatic measurements, tissue method, 75 dB threshold

The algorithm parameters are unchanged for the results shown in table 5.4. In table 5.5, the power threshold was lowered to 70 dB, which brought the results closer to the original method.

Dataset	Mean difference [BPM]
A2	1.22 ± 1.68
A3	4.64 ± 6.52
A4	-1.65 ± 4.42
A6	0.10 ± 5.24
A7	31.15 ± 41.74
B6	-0.12 ± 1.05
B10	26.88 ± 32.86

Table 5.5: Comparison of manual and automatic measurements, tissue method, 70 dB threshold

The results of the comparison between the manual measurements and the automatic measurements is shown in table 5.4. Notably, set B6 resulted in the most accurate results which were also superior to the clutter filter approach. Set A2 also yielded slightly better results with this method. The maternal heart segment and the umbilical artery segment, on the other hand, yielded highly inaccurate results.

The accuracy of the results from datasets B6 and B10 were unaffected by the threshold change.

5.2.1 Fetal Heart

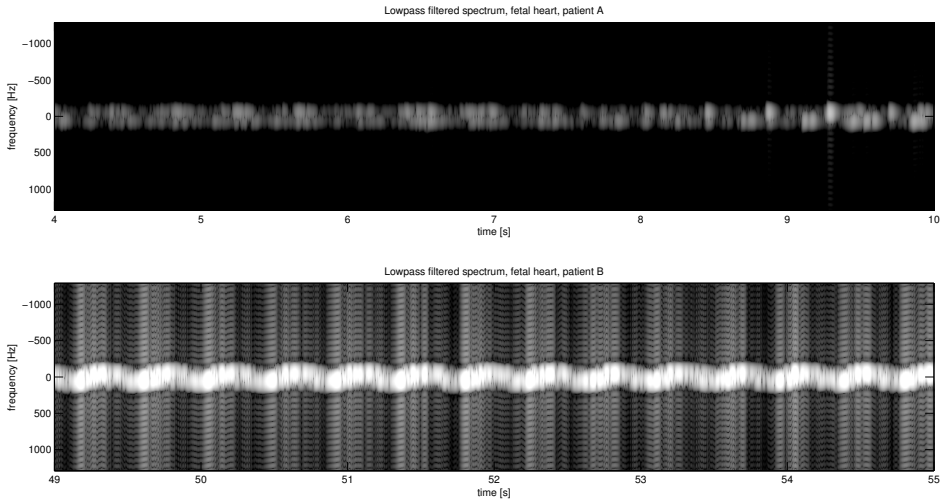

Figure 5.6: Lowpass filtered frequency spectrum, fetal heart, patient A (top) and patient B (bottom)

The clutter was much stronger in the data from patient B, as shown in figure 5.6, with calculations in figure 5.7. Figure 5.8 shows results from patient A, which had a weaker, prefiltered clutter signal.

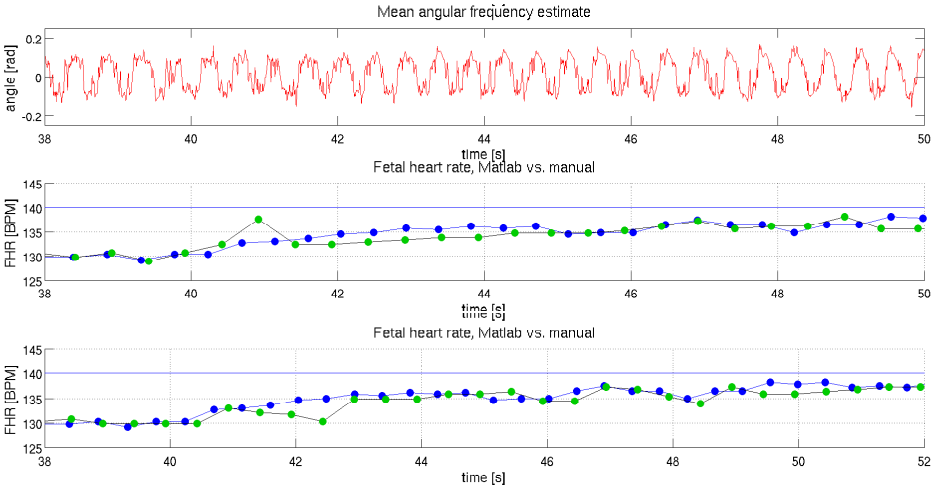

Figure 5.7: Heart rate calculation, low frequency, fetal heart, patient B. Clutter filter method (bottom) for comparison.

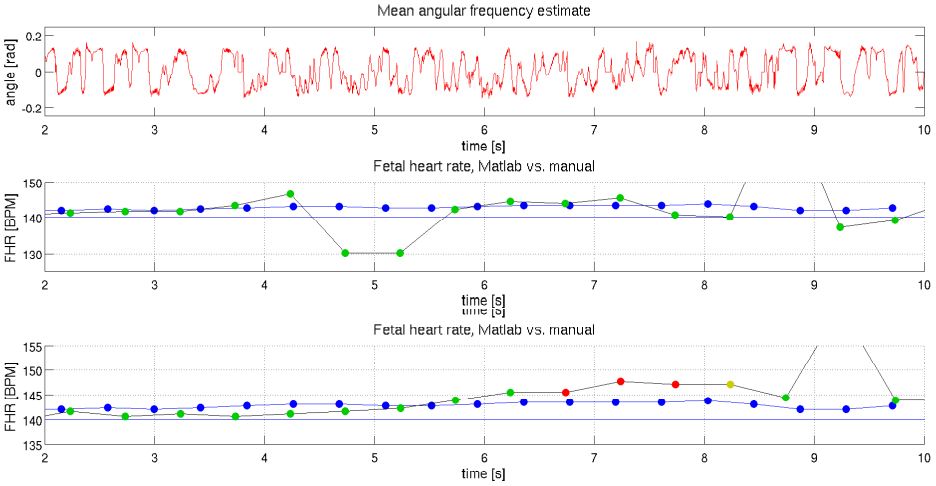

Figure 5.8: Heart rate calculations, low frequency, fetal heart, patient A. Clutter filter method (bottom) for comparison.

The results from set B6 seem to match the manual measurements more closely than in figure 5.2, while the results from A4 were less accurate. In the clutter method results for A4, the frequency estimate contains a noisy signal segment where the calculations are shown as red.

5.2.2 Umbilical Artery

The umbilical data was not viable for this approach, as this dataset lacks the presence of strong, periodic tissue movement as seen in the fetal heart data. The resulting calculations can be seen in figure 5.9.

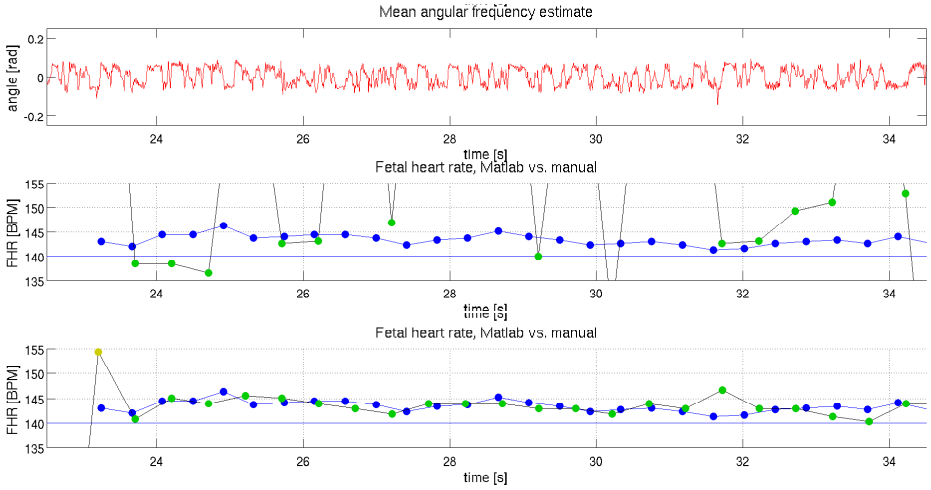

Figure 5.9: Heart rate calculations, low frequency, umbilical artery, 70 dB power threshold. Clutter filter method (bottom) for comparison.

5.3 Choice of parameters

The final choice of parameters is shown in table 5.6. These were the parameters found to yield the lowest difference between manual and automatic measurements.

Parameter description	Value
Clutter filter transition region	0.35 – 0.45 Nyquist
Clutter filter order	10
Averaging of frequency estimate	32 samples
Power threshold	75 dB
Decimation factor	4
Heart rate range	250 – 70 BPM
Segment length	2 cycles[1]
Calculations per second	2

Table 5.6: Chosen values for algorithm parameters

The decimation factor can be reduced in order to increase resolution of calculations if longer segments are acceptable. A decimation factor of four is not optimal with regards to accuracy, but serves as a compromise between accuracy and segment length.

5.4 Matlab Timing Results

The Matlab profile tool was used to estimate whether the algorithm will be suited for a realtime implementation. The algorithm was evaluated four times per dataset and the results are shown in table 5.7.

[1]Cycle count refers to the minimum heart rate in the specified range.

Dataset	Size [samples]	Duration [s]	Matlab CPU time
A2	39754	15.69	11.21 ± 0.04
A3	39402	15.83	11.23 ± 0.10
A4	40766	15.98	11.50 ± 0.03
A6	34914	13.90	10.08 ± 0.03
A7	40766	15.98	11.49 ± 0.05
B6	61952	24.19	16.09 ± 0.04
B10	47542	18.59	13.03 ± 0.04

Table 5.7: Timing results, Matlab profile tool

The Matlab algorithm consistently processed the data faster than realtime. However, the timing results varied too much with each run to examine the effect of algorithm parameters in detail.

Calls to the Matlab `mean` function is the most time consuming step in the algorithm, with calls to this function accounting for nearly 50% of the total runtime. This function is called when the power and frequency estimates are calculated.

5.4.1 C++ Implementation

The C++ implementation of the previous algorithm implementation was able to process all the datasets in less than 0.5 seconds combined. Even though it is a simpler implementation which does not include a power estimate, the method for frequency estimation is the same as used in the new algorithm.

Chapter 6

Discussion

6.1 Power Threshold

The power threshold level had the most effect on low-power segments, as it was intended to correct for segments with mainly thermal noise content. It also compensated to some extent for insufficient segment length by removing noise which could result in additional peaks in the AF. However, in some cases the thresholding for a low power segment would be excessive and result in a less periodic signal. The threshold value thus needs to be tuned to fit the desired signal quality with regards to signal power.

6.2 Segment Selection

The results indicate that longer segments will yield more stable results, as the autocorrelation function will peak more strongly at the location of the most common heart period in the signal. However, as the fetal heart rate variability is an indicator of fetal health, the segment size should be limited in size so that this variability is detectable.

6.2.1 Decimation Factor

The decimation factor was examined to see the effects on the heart rate resolution. If a resolution of 0.25 BPM is desired at the normal fetal heart rate, the resolution is decreased too much even with a decimation factor of two. If the ultrasound system is not powerful enough to process the segments without decimation, calculations of fetal heart rate variability will not be sufficiently accurate judging by the heart rate resolution alone. However, as this in a non-linear property, the time resolution may be a more suitable measure.

6.3 Peak Detection

Some of the segments were not suitable for peak detection. There are some instances where no peaks can be detected, such as when segments yield a monotonically decreasing AF. In that case, the maximum value would be found at the lowest permitted lag value, and no peaks could be found using the `findpeaks` function in Matlab. A similar problem

occurs if the entire signal lies below the power threshold, such as after the 11.8 second mark in set A4. The result is an autocorrelation function which is a constant zero value. Also here, the maximum of this function will be the first value, as no subsequent values will be larger. However, this is a result of low-power signals, in which case the result should be discarded based on visual inspection of the Doppler spectrum.

6.4 Cost Function

The cost function gave a better impression of the accuracy of the calculations when the previous algorithm was used. In that case, the triangle window dampened peaks other than at the expected main peak location.

With the new algorithm, inaccurate results are more often caused by detection of a peak at a different location than the actual heart period. In this case, the maximum value may still be high relative to the signal energy, which will yield a low cost value. For most of the false detections, this problem can be masked by asking the user to wait until the signal has reported several successful detections in sequence.

6.5 Tissue Movement

The tissue method yielded accurate results for patient B, and for some of the recordings from patient A after parameter adjustment, even though the algorithm had not been fully optimized for this method. Despite the clutter signal being weaker in the sets from patient A, there are still segments in which the results resemble the manual measurements when using the tissue frequencies.

The low power sets yielded better results once the power threshold was lowered by 5 dB, but it had little effect on the datasets which had not been clutter filtered by the scanner. This indicates that the method may be usable in either case after some refinement, although unfiltered data is strongly preferable.

The unfiltered umbilical set contained a relatively weak clutter signal (seen in figure 3.4 in section 3.2.2) which did not appear to be strongly periodic. A possible explanation could be that the angle of insonation was not optimal for capturing the movement of the umbilical artery walls, but multiple acquisitions would have been needed in order to verify this.

6.6 Realtime Considerations

The comparison between the runtime of the Matlab algorithm and the C++ implementation of the previously developed algorithm indicates that the current algorithm should be suitable for a realtime implementation.

Even though the decimation reduces the amount of samples contained in each segment, the full segment must still be buffered before it can be decimated. The length of the segment in time must therefore also be considered.

Vector autocorrelation in Matlab is optimized by using FFT[1], which means the Matlab

[1]Matlab version 8.0.0.783, `xcorr.m` in the signal processing toolbox, line 105

implementation of the algorithm uses a more efficient method for autocorrelation than the autocorrelation in the C++ implementation. This is one example of an optimization step that can make the C++ implementation more efficient.

6.7 Comparison of Results

In several of the cited articles, the evaluation of the algorithm is either based on results which were compared against invasive methods, such as direct FECG measurements during labor [6] or umbilical blood oxygen saturation [7], or simulated signals [4] or only considering implementation-related aspects [3]. In this case, only non-invasive Doppler ultrasound recordings were performed, and the results were only compared to the corresponding manual measurements.

6.7.1 Accuracy

The results were evaluated visually by comparing the calculated Doppler spectrum and the corresponding heart rate calculations. In parts where signal power drops, the accuracy is also reduced.

After comparing the automatic algorithm with the manual measurements by calculating the difference, the algorithm was found to have a slight, mainly negative bias between 0.5 and 1 BPM for sets containing a single acquisition target. However, the manual measurements themselves were not evaluated for accuracy, for instance by having a medical professional perform manual measurements.

6.7.2 Resolution

The resolution of the measurements was evaluated and compared to requirements for heart rate and heart period resolution [4, 9]. As long as decimation was avoided, a resolution of 0.25 BPM could be achieved for the desired heart rate range. With a decimation factor of two, the same could be achieved only for heart rates lower than 140 BPM.

However, if the goal is $\pm$ 1 ms resolution in the heart rate measurements, a decimation factor of two could be used.

6.8 Future Work

The peak detection could be improved, such as by calculating an average of multiple peak distances. However, this may be undesirable if heart rate variability should be detectable.

If the heart rate calculations can be performed without decimation, adding functionality which supports more in-depth analysis of the fetal heart rate variability will be valuable for detecting fetal health issues [4].

Chapter 7

Conclusion

An algorithm for automatically detecting fetal heart rate was developed in Matlab and later tested on two patients. Doppler IQ data was exported from a GE Vingmed E9 ultrasound scanner during the two examinations.

The data from patient A consisted of heart data clutter filtered by the scanner. The data from patient B consisted of one heart set and one umbilical set, unfiltered.

The algorithm successfully detects both fetal and maternal heart rates and produces fairly accurate results (mean difference of 0-1 BPM on average, for an optimal configuration) given a stable signal. The high frequency method yields accurate results for both umbilical and heart data. The algorithm was biased towards lower heart rates in most datasets (five out of seven).

Umbilical data seems to be best suited for the current algorithm parameters, as the spectrum peaks only once per cycle. This means fewer ambiguous peaks in the autocorrelation function, as there should ideally be only one main peak for the detection to succeed.

Using a lowpass filter was a viable method as well, provided that the signal contained strong, periodic clutter components. The unfiltered heart data from patient B yielded better results, whereas this method could not be used for the umbilical data. The disadvantage of this approach is that two separate implementations may be needed.

Based on the Matlab profile data and the accuracy of the results, the algorithm appears to represent a viable method for heart rate detection on a low-cost ultrasound device.

Bibliography

[1] B. Angelsen, H. Torp, *Excerpts from Ultrasound Imaging - Waves, Signals and Signal processing inMedical Ultrasonics, Vol II*. NTNU, 2000.

[2] P. R. Hoskins, et. al, *Diagnostic Ultrasound: Physics and Equipment (2nd Edition)*. Cambridge University Press, 2010.

[3] X. Hua, et. al, *A new algorithm for detecting fetal heart rate using ultrasound Doppler signals*. Elsevier, 2004.

[4] M. G. Signorini, et. al, *Monitoring Fetal Heart Rate during Pregnancy: Contributions from Advanced Signal Processing and Wearable Technology*. Hindawi Publishing Corporation, 2014.

[5] I. Voicu, et. al, *New Estimators and Guidelines for Better Use of Fetal Heart Rate Estimators with Doppler Ultrasound Devices*. Hindawi Publishing Corporation, 2014.

[6] J. Jezewski, et. al, *A novel technique for fetal heart rate estimation from Doppler ultrasound signal*. BioMedical Engineering OnLine, 2011.

[7] Y. Kimura, et. al, *Time-frequency analysis of fetal heartbeat fluctuation using wavelet transform*. The American Physiological Society, 1998.

[8] S. Brekke, *Techniques for enhancement of temporal resolution in three- dimensional echocardiography*. NTNU, 2007.

[9] P. J. D. Wickham, et. al, *Development of Methods for Quantitative Analysis of the Fetal Heart Rate*. Journal of Biomedical Engineering, 1983.

[10] J. Karin, et. al, *Non Invasive Fetal ECG Monitoring*. IEEE – Computers in Cardiology, 1994.

[11] C. H. L. Peters, et. al, *Beat-to-beat detection of fetal heart rate: Doppler ultrasound cardiotocography compared to direct ECG cardiotocography in time and frequency domain*. Institude of Physics Publishing, 2004.

Appendix A

README.txt

```
-----------------------------------------------------------------
Automatic extraction of Doppler parameters for the
assessment of fetal and maternal health

Agnes Heyer
June 2014, NTNU
-----------------------------------------------------------------
Source code and data files

Matlab algorithm
=================================================================

plotall.m runs Umojatest for all datasets, plots results
Umojatest.m calculates estimate, steps through dataset
getxcorrmax.m calculates autocorrelation, finds maximum
cheby.m designs clutter filter

Manualtest.m calculates manual measurements

/data contains all datasets
/manual contains all manual measurements


C++ algorithm
=================================================================

reader binary file, compiled for 64-bit Linux
runall.sh bash script, runs algorithm for all datasets
reader.cpp source file
reader.hpp header file
Makefile compiles source files

/data datasets, exported from Matlab to plaintext
```