2012

QUANTIFICATION OF PRETERM INFANT FEEDING COORDINATION: AN ALGORITHMIC APPROACH

Pallavi Ramnarain
Virginia Commonwealth University

QUANTIFICATION OF PRETERM INFANT FEEDING COORDINATION: AN ALGORITHMIC APPROACH

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by,

PALLAVI RAMNARAIN
B.S. VIRGINIA COMMONWEALTH UNIVERSITY, 2006
M.S. VIRGINIA COMMONWEALTH UNIVERSITY, 2010

Director: DR. PAUL A. WETZEL
ASSOCIATE PROFESSOR
DEPARTMENT OF BIOMEDICAL ENGINEERING

Virginia Commonwealth University
Richmond, Virginia
May, 2012

Acknowledgments

The author wishes to thank several people. I would like to thank my advisor, Dr. Wetzel, for his unending patience and guidance throughout this process. I would also like to thank Dr. Pickler for encouraging me to expand my knowledge base beyond engineering, and Marty Lewis, without whom I would have no data to work with. Over the years several undergraduate students have helped with this project, but the one who worked the longest with me and contributed the most was Robert Thompson. Thank you for reminding me that sometimes all I need is a good cup of coffee. I would also like to thank my lab mates, Isti Arief and Thea Pepperl, and fellow graduate students Colleen McLoughlin and David Burch, for making the many long hours spent in the lab far more enjoyable than I ever imagined. Last but not least, I would like to thank my family for their unending support over the last five or so years it has taken me to graduate. Without them I would have never made it this far.

Table of Contents

List of Figures

List of Tables

Abstract

QUANTIFICATION OF PRETERM INFANT FEEDING COORDINATION:
AN ALGORITHMIC APPROACH

BY PALLAVI RAMNARAIN, PhD.

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

VIRGINIA COMMONWEALTH UNIVERSITY, 2012.

Major Director: DR. PAUL A. WETZEL
ASSOCIATE PROFESSOR
DEPARTMENT OF BIOMEDICAL ENGINEERING

Oral feeding competency is a primary requirement for preterm infant hospital release. Currently there is no widely accepted method to objectively measure oral feeding. Feeding consists primarily of the integration of three individual feeding events: sucking, breathing, and swallowing, and the objective of feeding coordination is to minimize aspiration. The purpose of this work was to quantify the infant feeding process from signals obtained during bottle feeding and ultimately develop a measure of feeding coordination. Sucking was measured using a pressure transducer embedded within a modified silicone bottle block. Breathing was measured using a thermistor embedded within nasal cannula, and swallowing was measured through the use of several different piezoelectric sensors. In addition to feeding signals, electrocardiogram (ECG) signals were obtained as an indicator of overall infant behavioral state during feeding. Event detection algorithms for the individual feeding signals were developed and validated, then used for the development of a measurement of feeding coordination. The final suck event

detection algorithm was the result of an iterative process that depended on the validity of the signal model. As the model adapted to better represent the data, the accuracy and specificity of the algorithm improved. For the breath signal, however, the primary barrier to effective event detection was significant baseline drift. The frequency components of the baseline drift overlapped significantly with the breath event frequency components, so a time domain solution was developed. Several methods were tested, and it was found that the acceleration vector of the signal provided the most robust representation of the underlying breath signal while minimizing baseline drift. Swallow signal event detection was not possible due to a lack of available data resulting from problems with the consistency of the obtained signal. A robust method was developed for the batch processing of heart rate variability analysis. Finally a method of coordination analysis was developed based on the event detection algorithm outputs. Coordination was measured by determining the percentage of feeding time that consisted of overlapping suck and breath activity.

Introduction

In 2006 there were 4.6 million recorded infant hospital stays in the United States, of which only 8% consisted of preterm infants [1]. However, during the same year, 47% of the total cost of infant hospitalizations was spent on preterm infants [1]. This is due in large part to the long hospital stay often associated with preterm infancy. The mean length of stay for preterm infants is 12.9 days, while term infants have an average length of stay of 1.9 days [1]. The American Academy of Pediatrics includes oral feeding competency as a primary criterion for hospital discharge for preterm infants [2]. Concurrently, ineffective oral feeding is one of the primary reasons preterm infants experience delayed discharges [3]. Many times, complications resulting from infant prematurity negatively affect the neurological maturation required for feeding control [4]. Keeping preterm infants beyond the time when normal discharge requirements are met has been shown to neither reduce post discharge costs nor improve post discharge outcomes [5]. With this in mind, it is of the utmost priority to find an appropriate quantitative measurement for feeding competency that is reliable.

A hallmark of competent oral feeding is the coordination of sucking, swallowing, and breathing. Safe feeding for preterm infants involves the transference of formula from the oral cavity to the gastrointestinal tract with minimal aspiration [6]. This process is complex. It involves the use of 26 pairs of muscles, 5 cranial nerve systems, and multiple cervical and thoracic segments [7]. All of the involved anatomic regions exist in close proximity and must be functionally integrated for coordinated feeding to occur. Stimulation of specific areas can elicit muscular activity along reflex pathways [8]. However, the use of “natural” forms of stimulation is preferred because it preserves the physiologic nature of recruitment patterns [9]. These patterns are an integral part of the motor learning process. Sometimes the life-saving interventions that preterm infants need can compromise the development of effective oromotor control [9]. This is because the presence of multiple items in contact with the face and external feeding areas of preterm infants can stimulate them in a way that hinders the development of proper oromotor patterns, especially for sucking [7].

The purpose of this project is to create computational algorithms to assess the level of coordinated behavior expressed by preterm infants during bottle feeding. However, in order for feeding coordination to be properly determined each type of feeding event (suck, swallow and breath events) must first be located, making event detection the mandatory first step of coordination analyses. After a brief explanation of how the data were acquired, each chapter provides an in-depth analysis of the individual feeding signals and heart rate variability. The final chapter integrates the output of the event detection algorithms to provide a measurement of coordination.

This project was completed as part of an NIH sponsored biobehavioral interventional study called Feeding Readiness in Preterm Infants, referred to as the PRO Study (NIH R01NR005182, Pickler, PI). The PRO Study enrolled 109 infants over a period of five years. The scope of this project was limited to writing the programs to analyze the collected data. This included event detection and classification. The signals of concern included the suck signal, breath signal, swallow signal, and electrocardiogram (ECG) signal.

Specific Aims

1. *Breath Event Detection*

Sub Aim A was to identify individual breath events. Breath modulation is an important indicator of feeding competency. An algorithm was developed to interpret the thermistor output as individual breath events.

2. *Suck Event Detection*

Sub Aim B was to identify individual suck events. Infant sucking capability is indicative of oral neuromuscular development and is a primary component of feeding coordination. An algorithm was developed to parse the output of the Medoff-Cooper Nutritive Sucking Apparatus into individual suck events.

3. *Swallow Event Detection*

Sub Aim C was to identify individual swallow events. Swallow observation provides verification of transference of food boluses to the stomach. Currently there is no accepted approach to measurement of swallow in preterm infants.

4. *Heart Rate Variability Analysis*

Heart rate variability (HRV) is well documented as a state indicator as well as an indicator of the maturation of the autonomic nervous system. The aim was to develop an automated approach to HRV analysis that enables the batch processing of files.

5. *Event Coordination*

The primary aim of this work was to provide a quantitative measure for the purpose of feeding coordination assessment. For the purpose of this work, feeding coordination assessment was based upon patterns in breathing activity and sucking activity, as well as the interactions of these functions.

Data Acquisition

Hardware

All signals were acquired using the BIOPAC MP150, a 16-bit analog to digital converter that has 16 analog input channels. All signals were sampled at 1000 Hz. To acquire the breath signal, a thermistor (US Sensor Model H1744) embedded within a nasal cannula was used. The sucking signal was obtained through the use of a proprietary modified bottle with an embedded pressure transducer [10]. Swallow signals were obtained using various piezoelectric sensors, all developed by Dymedix. Finally, ECG and SaO2 measurements were obtained using standard NICU sensors connected to the Criticare monitoring unit (Scholar III, Criticare Systems, Waukesha, WI) (see Figure 1: Data Acquisition Flow Diagram).

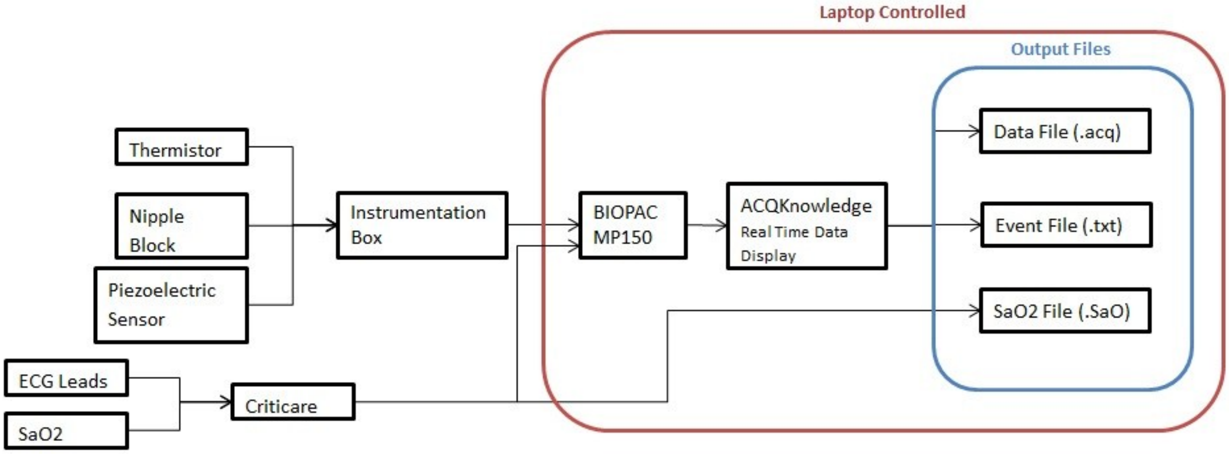

Figure 1: Data Acquisition Flow Diagram

Software

All post processing was done using a combination of BIOPAC's AcqKnowledge software (v. 3.9.1), MATLAB, and proprietary programs written for use in the lab. During data acquisition, the BIOPAC ACQKnowlege software provided real-time data displays, while also accepting event marker inputs from the data collectors. Event markers consisted of coded function keys on the laptop used to mark important points in time. The primary event markers were used to mark the beginning and end of feeding intervals. In addition, there were event marker keys coded to allow data collectors to mark visually detected feeding events, such as sucks or swallows, to aid in later data analysis.

All data acquired with the BIOPAC are stored in .acq file formats with accompanying event files. First, all event files were cleaned to remove any extra event markers. Then the feeding periods marked with "begin" and "end" event markers were extracted from the ACQ file using a program written by Dr. Wetzel. Finally individual feeding intervals were combined into one, continuous file that represents the full feeding time of each feeding observation. This was done in DOS (See Figure 2: Software Processing Flow Diagram).

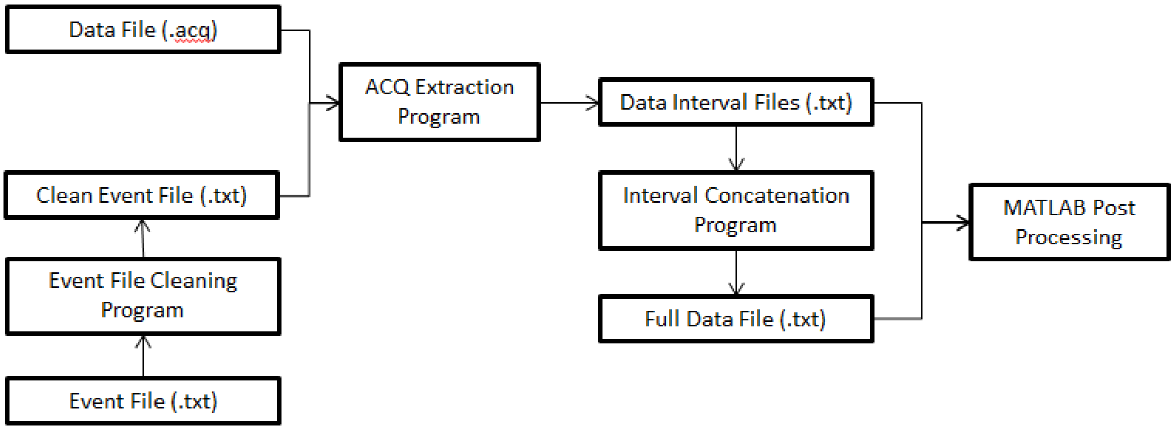

Figure 2: Software Processing Flow Diagram

Suck Event Detection

Background

Sucking is the primary form of nutrition acquisition for infants. Infants exhibit two types of sucking behavior: nutritive and nonnutritive suck. There is a common assumption that nonnutritive sucking assists with infant state modulation [11] and while nonnutritive sucking movements form the cornerstone of nutritive sucking movements, nutritive sucking exhibits a slower suck cycle and less bursts than nonnutritive sucking [12]. The most significant difference between nutritive sucking and nonnutritive sucking is that nutritive suck requires coordination with swallow and breath.

Mature nutritive sucking is composed of two phases: suction and expression [7]. The maturation of suction musculature is characterized in the literature through the use of the following measures: suction rate, amplitude, the first derivative of the suction signal, and the length of suck runs. Suction rate and the first derivative of the suction signal are considered to be indicative of the synchrony of the underlying musculature involved in suction. Suction amplitude is an indicator of the maximum suction force generated. Suck runs, or bursts, are generally defined as being 3 or more events in succession where interevent intervals are less than 2 seconds [13]. Stability and length of suck runs are indicative of how long an infant can maintain the synchronization for effective sucking.

It is widely believed that sucking behavior changes with maturation, and that this change can be used as an indication of maturation [4] [3] [7] [14] [6] [15] [10]. It has been shown that feeding outcomes are strongly related to the number of sucks in the first suck burst [16]. Vice &

Gewolb found that suck maturation can be divided into three major stages [13]. In the first stage, the infant exhibits rapid, patterned, low-amplitude nascent sucking activity. In the second stage the shape of suck pressure waveform progresses to irregular deflections at a rate of 2 to 3 events per second. In this stage sucking behavior is not necessarily coordinated to or linked with swallowing. In the final stage, the suck rate slows to 1 suck per second, the shape of the waveform stabilizes, and there is a more regular pairing of suck and swallow events. Barlow, however, described suck maturation as consisting of five phases [9]. Phase one is characterized by arrhythmic expression with no suction activity present. Phase two consists of the transition to rhythmic expression and the appearance of arrhythmic suction activity. In phase three rhythmic suction starts to be evident. Phase four is the progression to an alternating pattern of suction and expression, and phase five is marked by concomitant increases in suction amplitude and the duration of suck bursts. Combined these two different models of suck maturation provide useful hallmarks of the progression of suck maturation.

The development of a reliable measurement methodology to measure sucking would be a useful clinical tool to assess the normal development of suck patterns [17]. There have been several approaches to measuring sucking in preterm infants. Pickler *et al.* used a mercury strain gage [4]. Miller and Kang used B-mode ultrasound to assess the lingual movements associated with suck [11]. Amaizu measured suck through the use of two intranipple catheters connected to pressure transducers, which was very similar to the setup used by Vice & Gewolb [6] [13]. All of these assessment methods are either invasive or cumbersome.

Signal Acquisition

The sucking signal was obtained through the use of a modified nipple block with an embedded pressure transducer (see Figure 3: Pressure Transducer Embedded Within Nipple

Block). The nipple block was modified to induce laminar flow through the pressure transducer for accurate pressure assessments. In addition to containing the pressure transducer, the nipple block provided attachment points for standard infant formula reservoirs and for a modified silicone nipple. The nipple was modified to contain a center capillary that served as a continuation of the nipple block channel to ensure laminar flow. The pressure transducer's output signal was minimally filtered and amplified before acquisition. The BIOPAC MP150 was used for data acquisition in conjunction with the associated AcqKnowledge program, which provided a real time display of the output signal. The sucking signal was sampled at a rate of 1000 Hz.

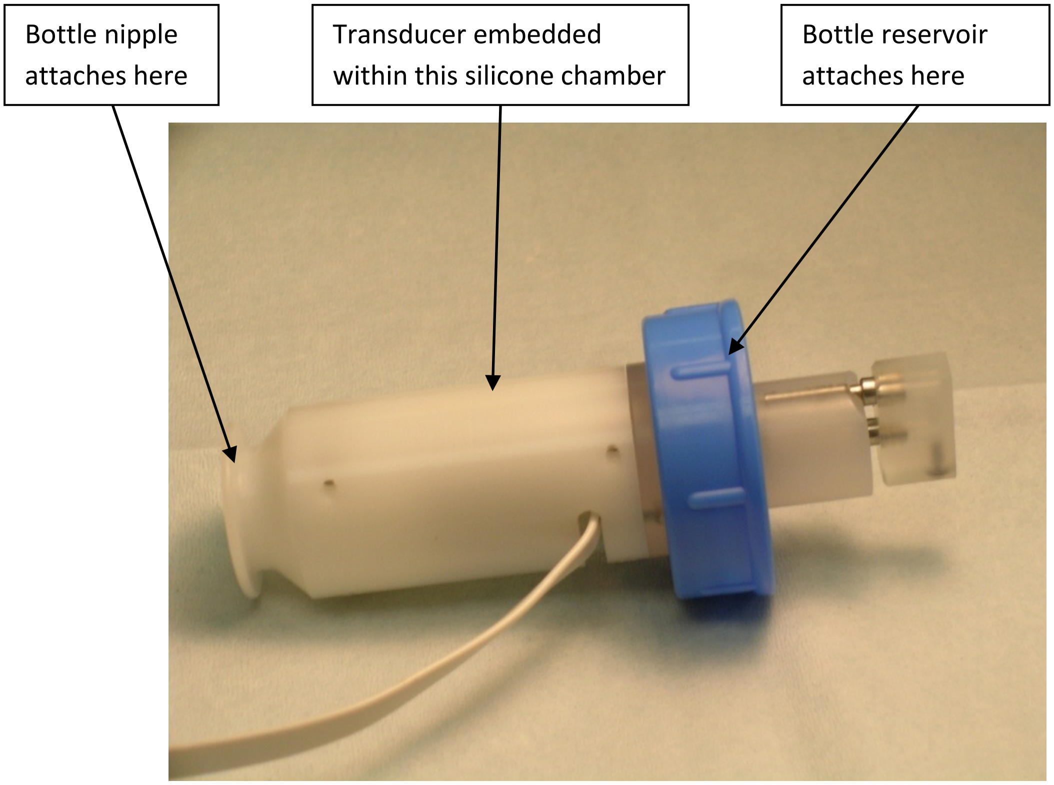
Bottle nipple attaches here
Transducer embedded within this silicone chamber
Bottle reservoir attaches here

Figure 3: Pressure Transducer Embedded Within Nipple Block

Suck Event Characterization Study

A study was completed to assess the physical characteristics of individual suck events. This was done to provide basic quantifiable characteristics for use as event detection parameters.

A total of 6695 individual measurements were made by hand from 36 different feedings. Each event was measured for type and classified into categorical groups by shape (see Figure 4). Type A consisted of events comprised of an increase in pressure followed by a decrease in pressure, with no change of direction between the initial increase and final decrease. Type B events were those where at the peak pressure of the event there existed a change of direction in the signal, and Type C events were a specialized subgroup of Type B where the change of direction included a return to baseline. From these measurements it was observed that the most commonly occurring type of event was Type B (see Figure 5: Representation of Distribution of Measured Suck Event Types and Table 1: Distribution of Suck Event Types). The ANOVA showed that there was a significant difference between the mean event durations for the three types of events (F = 10.97, DF = 2, $p < 0.0001$). Tukey's HSD showed that there was a significant difference between the mean event durations of Type A and Type B events. Type B events were found to be longer in duration by 48 ms (SE = 10.8, $p < 0.0001$) than Type A events. However, Type C event durations were not found to be statistically significantly different from Type A or Type B (see Table 2: Means and Confidence Intervals for Event Durations by Suck Type).

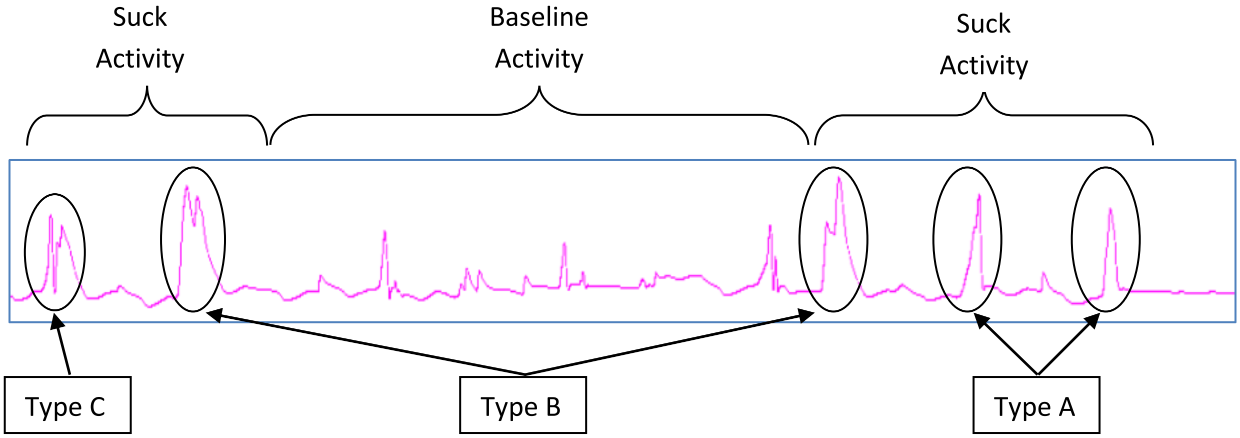

Figure 4: Examples of Suck Event Types

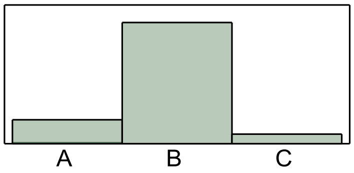

Figure 5: Representation of Distribution of Measured Suck Event Types

Table 1: Distribution of Suck Event Types

Level	Count	Probability
A	1009	0.15
B	5353	0.80
C	333	0.05
Total	6695	1.00

Table 2: Means and Confidence Intervals for Event Durations by Suck Type

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
A	1009	506.893	9.908	487.47	526.32
B	5353	554.844	4.302	546.41	563.28
C	333	520.750	17.247	486.94	554.56

Evolution of the Suck Algorithm

The suck signal is an approximation of intraoral pressure from the direct result of the effects of the pressure transducer characteristics and acquisition hardware. While it is usually a stable signal, with little drift, the inherent noise characteristics are inseparable from the underlying signal. The signal is comprised of baseline activity and true events (see Figure 4), both of which have overlapping frequency components. Baseline activity includes all detected pressure activity that cannot be classified as a true suck event. To be classified as a true suck event, the event must meet minimum amplitude and duration requirements.

Visually, experts described the amplitude requirements as simply “significantly greater than baseline”. Duration requirements were implied from the anatomical restrictions and were derived from descriptive statements. One such example is the statement “A sucking rate can never be as high as 5 sucks per second,” which leads to the constraint that to be a true event, a suck must be longer than 200 ms in duration.

The various attempts at event detection evolved as a function of the understanding of the true signal. The first signal model employed envisaged the signal as being composed of a stable baseline with distinct, singular events protruding from it. The failure of the first attempts at event detection led to an investigation of the true characteristics of the signal, the results of which showed that the signal was composed different types of suck events as well as baseline activity that resembles suck events on a smaller scale. A deeper understanding of this model led to the final event detection algorithm.

As described previously, there are three main types of suck events. These different event types usually exist simultaneously within a single feeding. Their varying characteristics lead to a wide range of amplitudes and durations for true suck events, and the overlapping spectrums of the baseline activity and sucking activity make the implementation of traditional filtering techniques impossible. Also, filtering can introduce amplitude and phase changes that would negatively impact the necessary physiologic parameters and coordination analyses.

The first version of the suck algorithm attempted to determine event onsets and ends from the first derivative, or velocity vector, of the signal. Unfortunately the velocity vector proved too susceptible to noisy baseline activity and provided a signal in which it was not possible to distinguish true events from noise. Since smoothing and filtering were not options, subsequent attempts at event detection focused on methods that could be employed on the raw signal.

The next attempt at a suck event detection algorithm used a thresholding technique on the raw signal. The first thresholding technique involved the use of a moving average window. Averages within shorter windows were artificially inflated by the amplitude of suck events, while larger windows resulted in too much data loss. Roughly 10 seconds of data were needed in order to approximate baseline, and that comprised of a minimum of 1/30th of the data segments.

The next thresholding attempt involved the use of the overall mean of the signal, or the DC value. This is depicted in Figure 6 where the threshold (shown in red) is overlaid on a portion of suck data. Green circles mark event onsets while blue circles mark event ends. While this thresholding technique resulted in a closer approximation of baseline than the moving average threshold, the static threshold value was a poor approximation of baseline when there were discrete changes in the signal offset.

The final approach combined the previous thresholding approaches. Baseline was first modeled as the mode of the data in 10 second windows. An event amplitude threshold was then set as a combination of the statistical properties of the signal and the calculated baseline. Everything below the threshold was zeroed, introducing the assumption that events had to have amplitudes above the value of the threshold. Approximate onsets and ends were determined from the zeroed signal. Those values were then used to determine true event onsets and ends from the original signal. This approach is visually described in Figure 7.

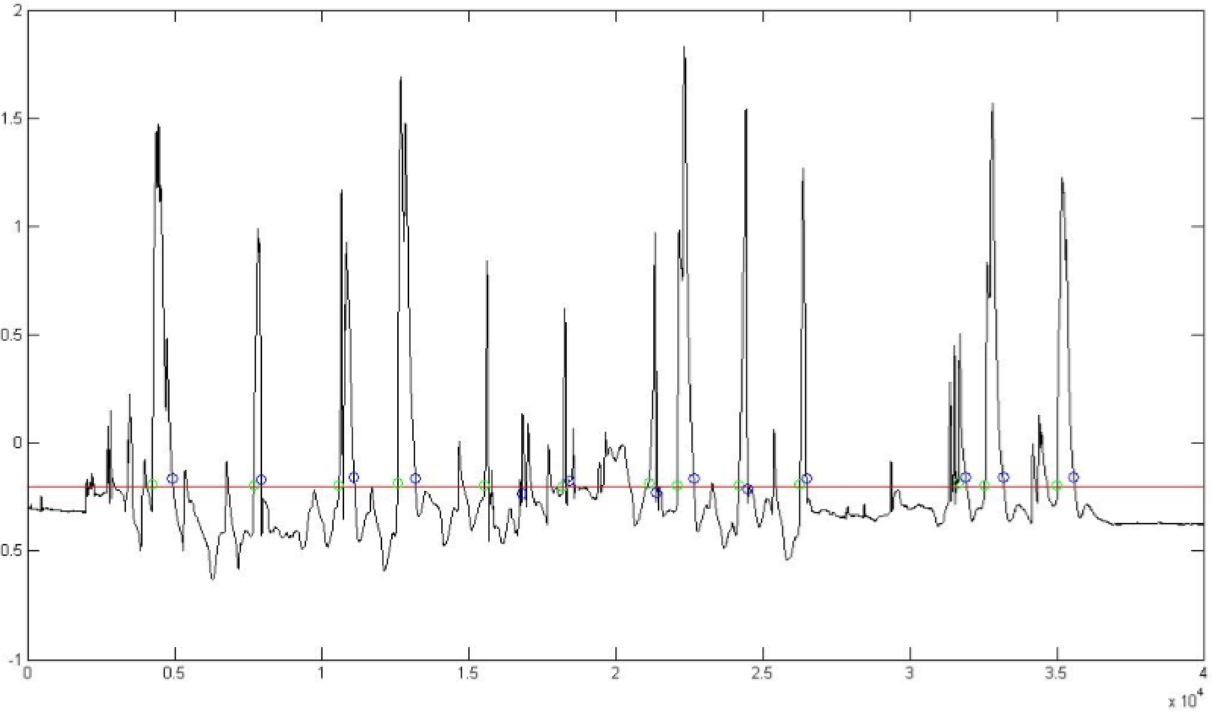

Figure 6: Example of Suck Event Detection Using DC Threshold

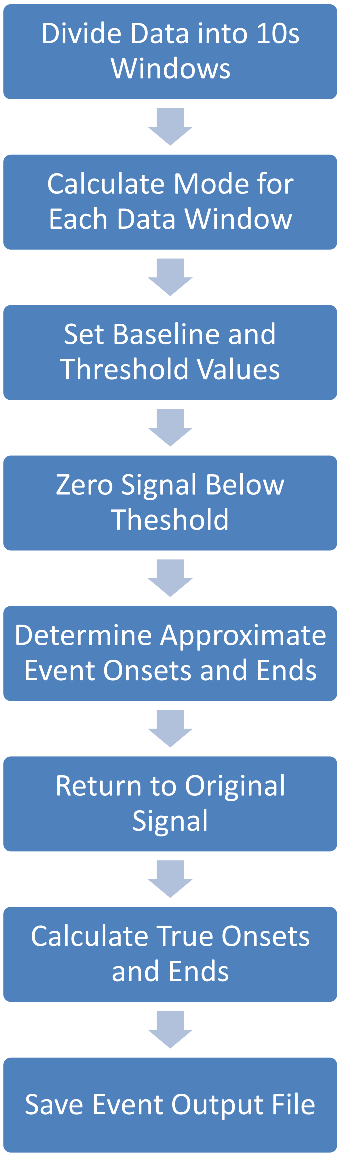
Divide Data into 10s Windows
Calculate Mode for Each Data Window
Set Baseline and Threshold Values
Zero Signal Below Theshold
Determine Approximate Event Onsets and Ends
Return to Original Signal
Calculate True Onsets and Ends
Save Event Output File

Figure 7: Algorithmic Flow Diagram for Suck Algorithm

Event Detection Validation

Preliminary analysis of the suck data showed a very low Type I error rate (mean = 3%), resulting in an average event detection rate of 97% (see Table 3: Suck Validation Results). The high Type II error rate was found to be the direct result of the combination of event definitions (with respect to characteristics like event duration and amplitude) and assumptions incorporated in the algorithm and expert inconsistency. In order to explore the cause for the difference between the number of points the expert detected versus the number of points algorithmically detected was found to be an application of inconsistent detection rules by the expert versus the absolute application of consistent detection rules in the automated detection. A possible cause of the inconsistencies in applied detection criteria may be the validation process. For the validation process the expert manually marked and examined sucking signals within the ACQKnowledge program. Differences in applied scaling during scrolling can lead to variations in detection criteria as events are determined through an inherent visual comparison with the rest of the signal. Figure 8 shows an example of expert inconsistency. The highlighted event was identified as an event by both the expert and the algorithm. The expert did not identify the following event (marked in red) but did identify the third event (marked in green), while the algorithm identified both events. The algorithm, however, used absolute values for baseline and threshold measurements, which enabled uniform application of detection criteria throughout the signal.

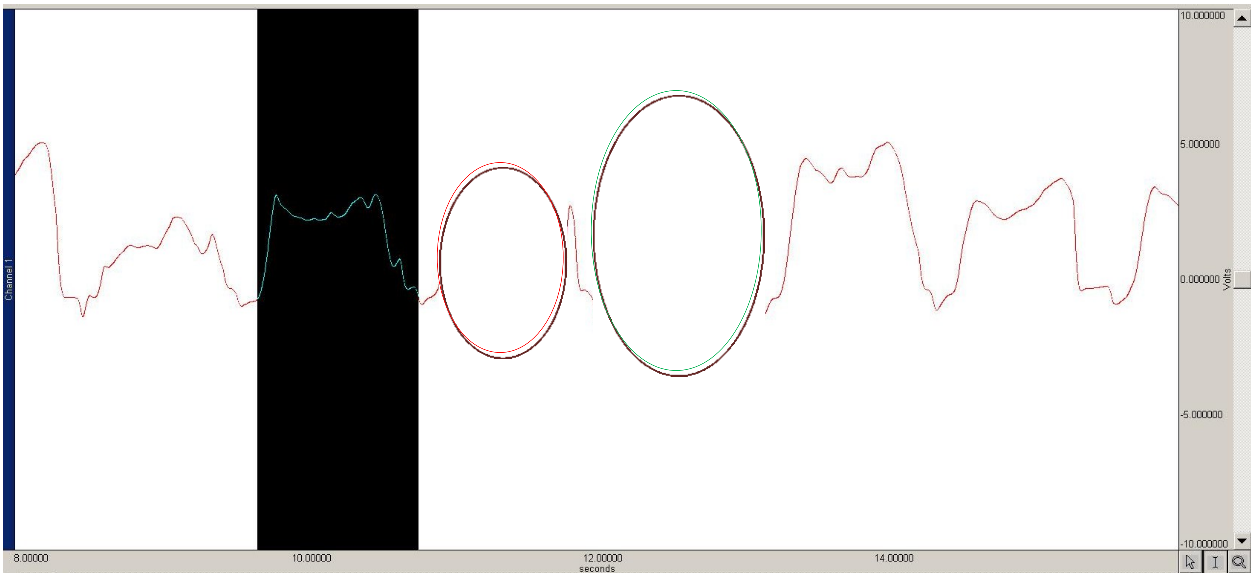

Figure 8: Example of Expert Inconsistency

Table 3: Suck Validation Results

File	# Expert Events	# Algorithm Events	Matched Points	False Positives	% Type I Error	% Type II Error
03398	42	69	42	27	0	39
03699	157	195	157	38	0	19
03799	203	217	203	14	0	6
03898	288	295	288	7	0	2
04098	180	210	180	30	0	14
04398	186	259	186	73	0	28
04599	192	225	192	33	0	15
04699	160	153	153	0	4	0
05098	121	165	121	44	0	27
05298	153	110	110	0	28	0
Total	1682	1898	1632	266	3	14

Breathe Event Detection

Background

Breathing is a vital function intrinsic to the survival of any human being. With preterm infants it is an important indicator of maturation and feeding competency, which is a hallmark for hospital release. Respiratory patterns in infants vary greatly from adults.

The increased metabolic rate of newborn mammals, including human infants, necessitates an increase in ventilation [18]. This increase in ventilation can result from an increase in tidal volume, an increase in breathing frequency, or a combination of the two. All of the options have mechanical constraints. An increase in tidal volume results in an increase in the elastic component of breathing work and an increase in the distortion of the compliant chest wall. An increase in breathing frequency results in an increase in the frictional work of breathing, and it requires strenuous efforts during inspiration to adequately ventilate the lungs in a shorter period of time. The time constant of respiratory systems in newborn infants is approximately 220 ms [18]. The breathing pattern in newborn infants is highly variable. It can consist of occasional deep or very shallow breaths, slow breathing periods, rapid bursts, short apnea and interruptions of expiratory flow [18].

Preterm infants switch suddenly between breathing patterns as a function of state of consciousness [13]. There are two primary respiratory patterns in preterm infants: regular and periodic. Periodic refers to periods of ventilation that are interrupted by brief apnea [13]. Cohen *et al.* established measurements of periodicity as important features to be included in classification of the maturation of preterm infants.

Respiratory patterns in preterm infants during feeding change as a function of maturation. Swallow influences respiration through the shared musculature and anatomical spaces involved. At the onset of feedings Vice & Gewolb noted a reduction in breathing rate and tidal volume, and as the feeding progressed the respiratory airflow pattern increased in irregularity [13].

The inherent variations in respiratory patterns necessitate the use of a measuring device capable of accurately measuring these dynamic changes. Thermistors provide a semi-quantitative, indirect way to measure respiration [19]. A thermistor is a device where electric resistance varies as a function of temperature [20]. As the speed of airflow passing the thermistor increases, the change in the thermistor's temperature increases and approaches that of the passing air [8]. Also, thermistors are recommended by the American Thoracic Society Guidelines for the measurement of airflow in pediatric applications [21].

Thermistors are classified as semi-quantitative because the flow output signal they provide is not a direct measurement of actual nasal airflow. Instead it is a combination of nasal airflow and the time constant of the sensor. In 1998 Farre *et al.* found that thermistors could not accurately measure actual flow, and that the relationship between the peak to peak amplitude of thermistor and respiration signals was nonlinear [22]. They also found that the response of a thermistor depends heavily on the airflow pattern, distance from nose and section of nostrils. The thermistor output signal is a direct measurement of the change in temperature of the actual thermistor, and, with correct placement, an indirect measure of nasal airflow resulting from the change in temperature between inspiration and expiration. The amplitude of the output signal is not an accurate reflection of actual airflow magnitude because of the effects of the associated time constant [21]. The temperature change sensed by a thermistor results from convective heat transfer, and Farre *et al.* found that the true breathing airflow signal and the sensor temperature signal were related by a nonlinear differential equation [22]. They also found that the most influential factor affecting the accuracy of the thermistor output was air convection around the device. Respiratory efforts can be inferred from thermistor measurements, but they are really a measurement of nasal airflow. It has been found that thermistor measurements significantly underestimate apneic events [19]. Also, the presence of a nasal cannula can significantly increase the nasal airway resistance in subjects with narrow nares or deviated nasal septum [19].

The physical geometry of the thermistor sensor can also play a role in its response time and accuracy. Primiano *et al.* found that when the surface temperature of the thermistor was less than the dew point of the gas it was measuring, a small layer of condensation coated the sensor, delaying its response time [22]. As the temperature reaches the dew point, the sensor output stabilizes until the condensation evaporates, and the resulting dry thermistor tracks temperature properly. This is less of concern with the thermistors used in this study as the surface area of the sensing portion was so small that any condensation effects were negligible.

In 2009 Series *et al.* found that current thermistor technology is accurate enough to detect apnea [23]. They also noted, though, that a decrease in thermistor output could be the result of either an apnea or mouth breathing. Their definition of an apnea was a 50% decrease in thermistor signal for 10 seconds or more and/or an accompanying 2% decrease in SaO2.

The temperature difference within a thermistor is so small that it is negligible, and as such a lumped heat capacity model can be used when modeling its response [24]. Storck found that the thermistor temperature does not directly describe the respiratory phase in a meaningful way, but its time derivative illustrates respiration well [24]. This is because the derivative describes heat flux, which changes more rapidly than temperature.

One of the biggest challenges in processing nasal airflow signals obtained from thermistors is accurately eliminating artifact. Artifact is evident primarily as signal offset and baseline drift, although at times 60 Hz power line interference can also be observed. The closest physiologic signal comparison is the electrocardiogram (ECG). ECGs often exhibit similar forms of signal noise as respiration. Noise sources that result in baseline wandering in ECG signals include power line interference, electrode contact noise, and EMG. Respiration can also contribute to ECG baseline drift. Like ECG signals, baseline drift in respiration signals is in-band noise, meaning the frequency of the signal drift falls within the frequency range of the actual signal itself.

To investigate these issues, Afsar *et al.* compared 7 techniques for baseline removal in ECG signals: cubic spline curve fitting, linear spline curve fitting, median filtering, finite impulse response high pass filtering, adaptive filtering, wavelet adaptive filtering, and empirical mode decomposition [25]. For their application wavelet adaptive filtering worked best because their goal was to remove baseline drift while preserving the morphology of the ST wave. These seven approaches constitute the usual approaches to baseline removal. High pass filtering is not a viable solution for this application because the frequency range of the baseline drift overlaps with the frequency content of the signal. Wavelets are inappropriate for this application for the same reason.

In choosing the final method for baseline drift removal, a comparison was done between linear approximation, cubic spline approximation, an adaptive filter, a first derivative based approach, and a fifth approach described as a second derivative signal modeling approach.

Signal Acquisition

Data was collected from 9 preterm infants during bottle feedings. This data was a subset of a larger study of preterm infant feeding (NIH R01NR05182, RH Pickler, PI). All respiratory data was collected using thermistors made by U.S. Sensor (model H1744). Figure 9 shows the thermistor and Figure 10 is a schematic of the thermistor. The thermistors were embedded in modified pediatric nasal cannula. The signal passed through a bridge circuit (see a simplified representation in Figure 11: Example Thermistor Bridge Circuit) was then differentially amplified through an instrumentation amplifier (Analog Devices, AD-524) then filtered through an active 2nd order low pass filter (f_c = 10 Hz) before being digitized. The BIOPAC MP150 was used to sample the thermistor signals at a rate of 1000 samples/sec. Post processing was done using MATLAB (© Mathworks, Ltm). The four methods were compared to a data expert's findings. For this analysis one instance of data was used, file 01901.

Figure 9: Photograph of Thermistor **[26]**

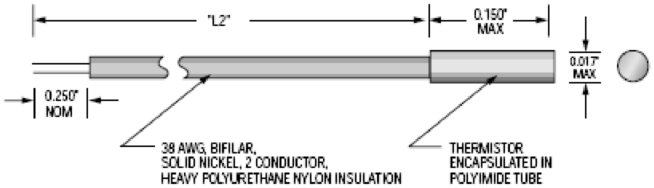
"L2"
0.150" MAX
0.017" MAX
0.250" NOM
38 AWG, BIFILAR, SOLID NICKEL, 2 CONDUCTOR, HEAVY POLYURETHANE NYLON INSULATION
THERMISTOR ENCAPSULATED IN POLYIMIDE TUBE

Figure 10: Schematic of Thermistor **[26]**

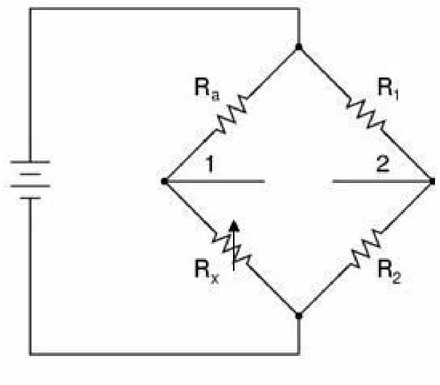

Figure 11: Example Thermistor Bridge Circuit

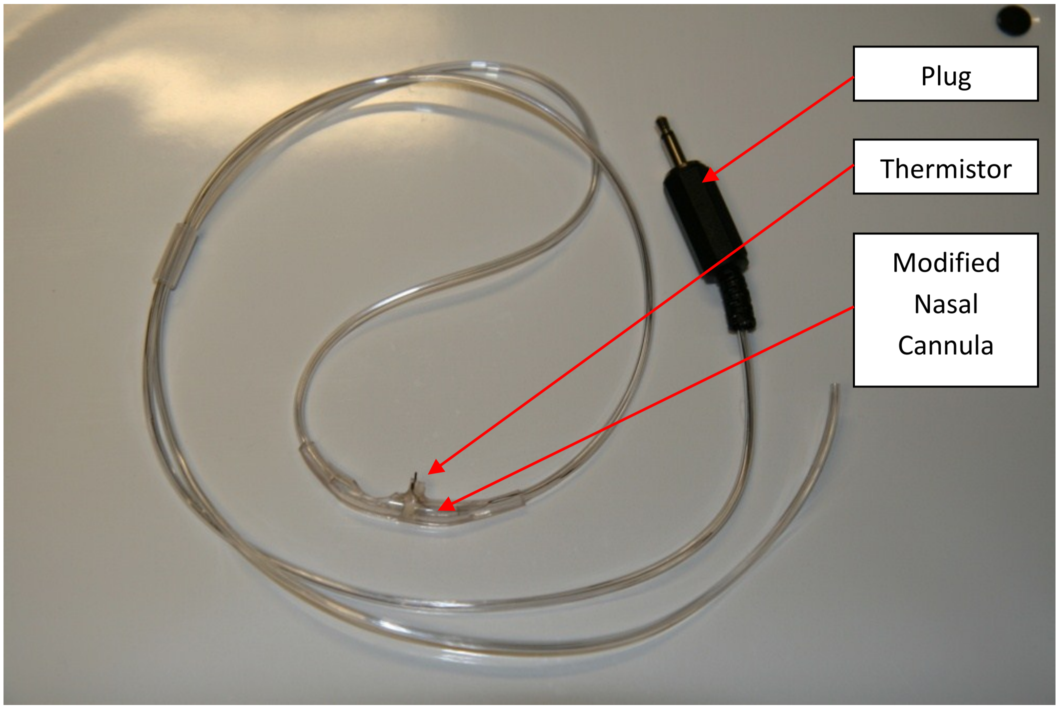

Figure 12: Depiction of Sensor

Event Detection Algorithm

The same event detection algorithm was used to compare all the methods. The algorithm sets a threshold based on the statistical properties of the signal, and then zeros the signal below that level. All points where the signal crosses the threshold are treated as potential onset of breath locations. Maximums for each region where the signal crosses the threshold are calculated. These calculated points serve as the event markers, except for instances where the points are closer than 200 ms.

Baseline Drift Attenuation

Due to the characteristics of this signal, before accurate event detection could be achieved, a reliable baseline drift removal technique had to be developed. Four different methods of baseline drift attenuation were compared. The first three methods selected all attempted to model the baseline drift in order to subtract it. These included a linear approximation, a cubic spline interpolated approximation, and a recurrent neural network approach mimicking an adaptive filter. The final method for comparison involved calculating the first and second derivatives of the signal in order to attenuate the baseline drift.

Linear Approximation

Linear approximation of baseline drift was chosen as a method for comparison due to its ease of implementation. The raw thermistor output was interpolated using a linear approximation that included downsampling to 2 Hz. Downsampling enabled a smoothing of the waveform that better approximated the baseline drift. Frequencies above 2 Hz started to mimic the true nasal airflow signal, which would result in dramatic signal loss. Once the linear approximation of baseline drift was calculated, it was subtracted from the original thermistor output signal, and the final result was visually examined to gage the efficacy of the methodology. The event detection algorithm was run using the subtracted signal as an input and points were matched to an expert's opinion.

Figure 13 shows the linear approximation of the thermistor output signal. Unfortunately the linear interpolation did not closely approximate the signal drift in all instances (see Figure

14: Discongruencies in the Baseline Approximation of the Linearly Interpolated Signal). The linearly interpolated signal was subtracted from the original thermistor output signal, resulting in a clear removal of the thermistor signal's DC component (see Figure 15: Comparison of Thermistor Output to Subtracted Linear Interpolation). However this approach was not able to eliminate the underlying signal drift (see Figure 16: Evident Residual Baseline Drift after Subtraction of Linear Approximation).

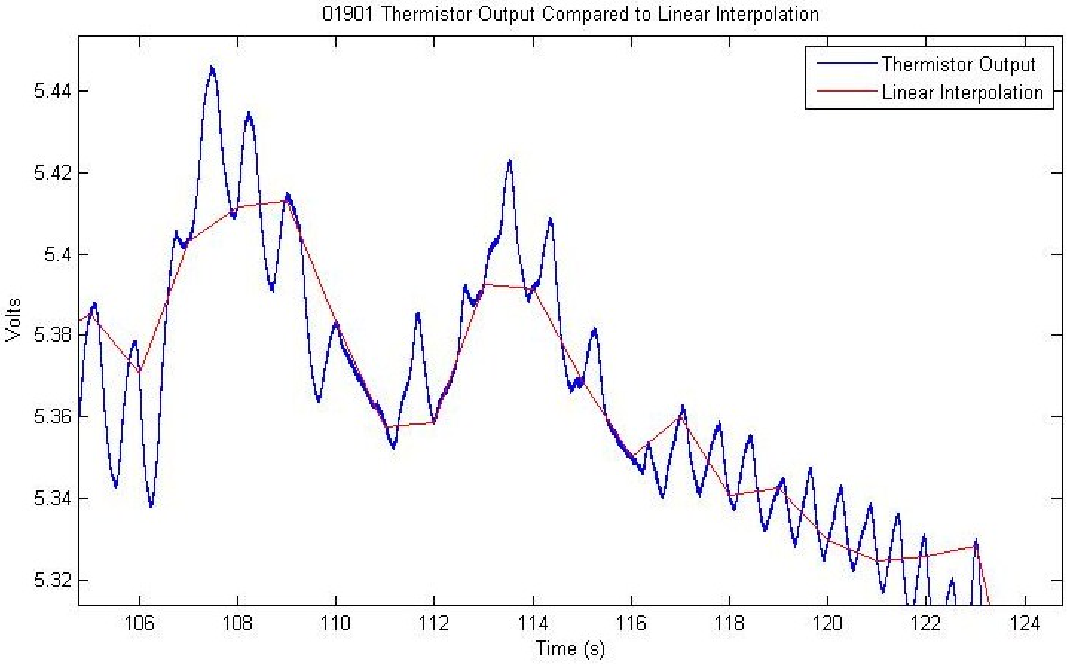

Figure 13: Linear Approximation of Baseline Drift

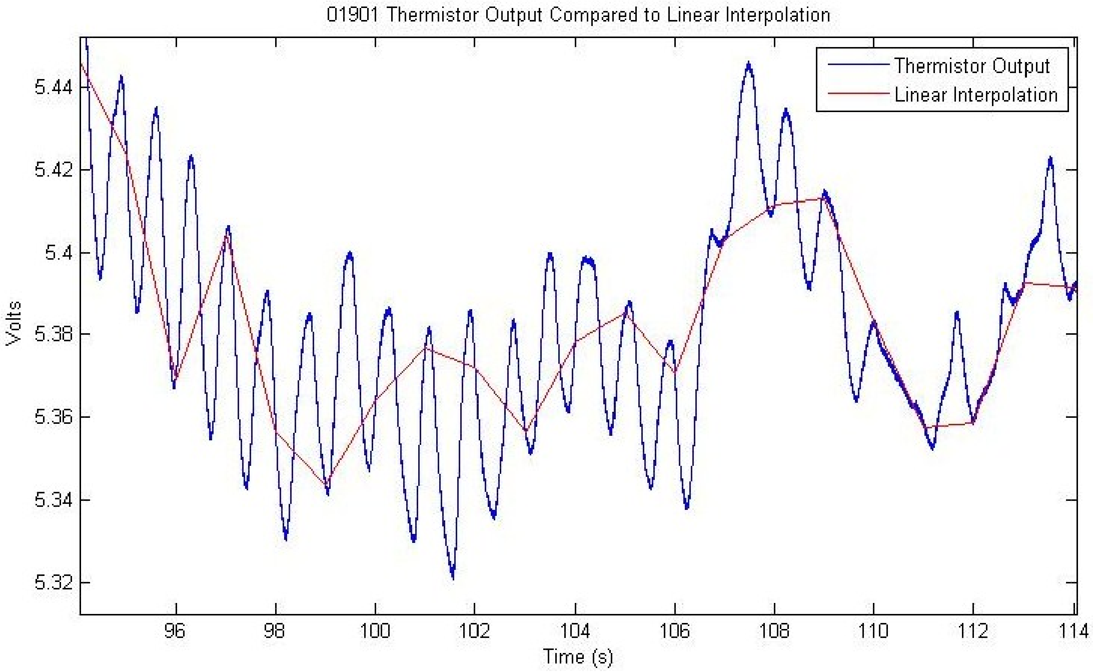

Figure 14: Discongruencies in the Baseline Approximation of the Linearly Interpolated Signal

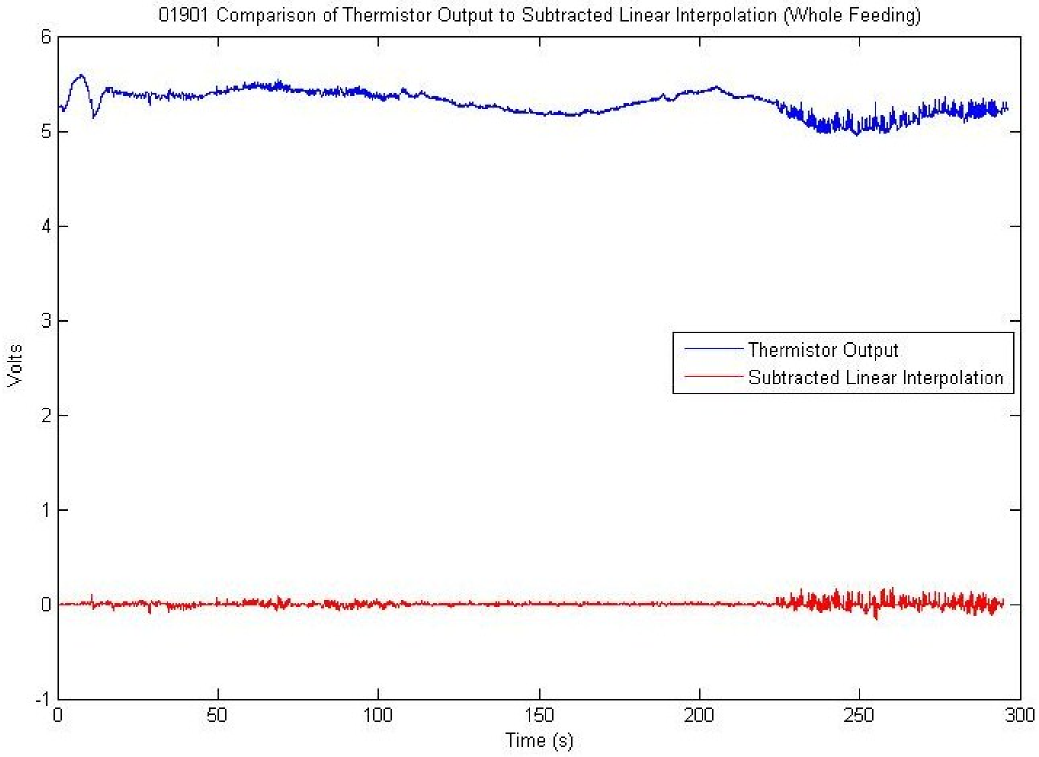

Figure 15: Comparison of Thermistor Output to Subtracted Linear Interpolation

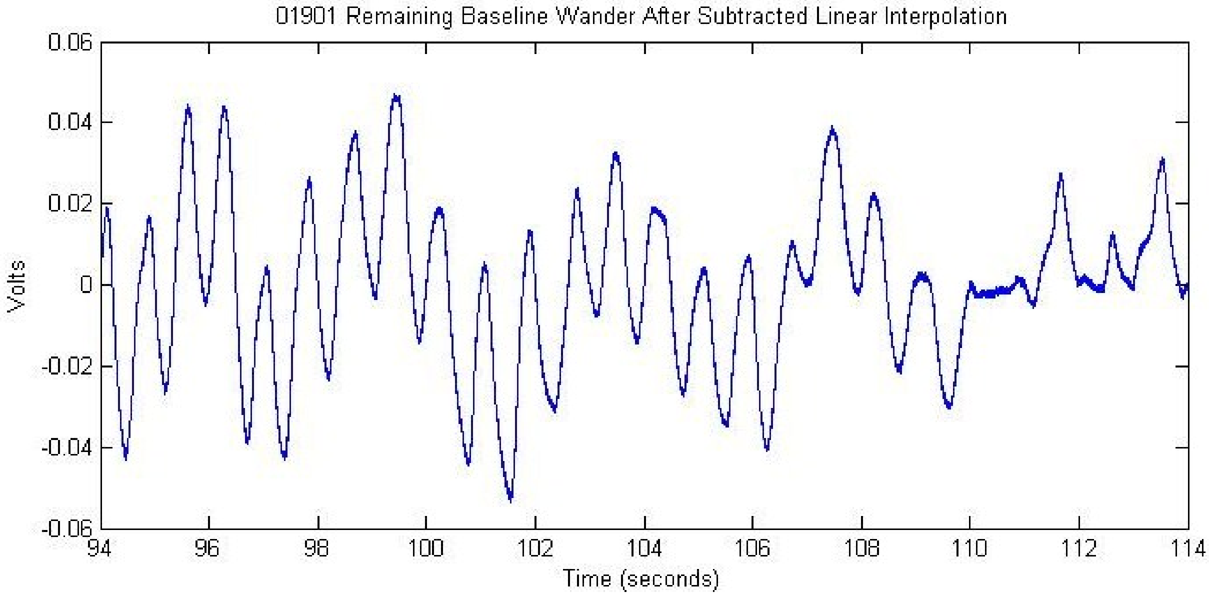

Figure 16: Evident Residual Baseline Drift after Subtraction of Linear Approximation

Cubic Spline Approximation

The cubic spline approximation of baseline drift was chosen as a method for comparison because its approximation is more accurate than the linear method due to its incorporation of every point of the signal. A cubic spline interpolation of the raw thermistor output signal was calculated. The resulting signal was then down sampled to 1 Hz. The sampling rate of the cubic spline signal was chosen to be 1 Hz in an attempt to only approximate the baseline wandering and exclude the actual respiratory signal. After the cubic spline was calculated, it was subtracted from the thermistor output signal, and the resulting waveform was visually examined for validity. The subtracted signal was then used as the input for the event detection algorithm, and the final output points were matched to those of an expert's opinion.

Figure 17 shows the cubic spline approximation of the thermistor output signal. From this figure it is clear that this polynomial-based approach was a more accurate approximation method than the linear spline. Figure 18 shows how the subtraction of the cubic spline approximation from the original thermistor output signal eliminated the bulk of the thermistor signal's DC component. A closer inspection, though, shows that the baseline drift is still present in the signal, although to a much lesser extent than with the linear approximation method (see Figure 19: Residual Baseline Drift after Cubic Spline Subtraction).

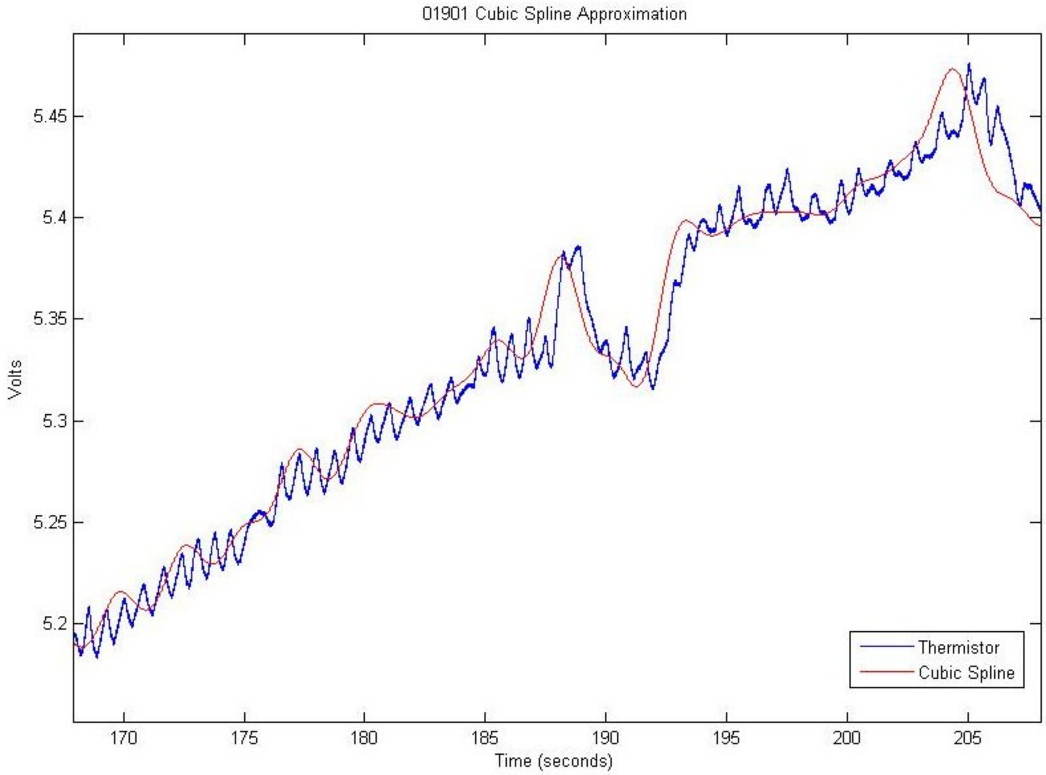

Figure 17: Cubic Spline Approximation of Baseline Drift

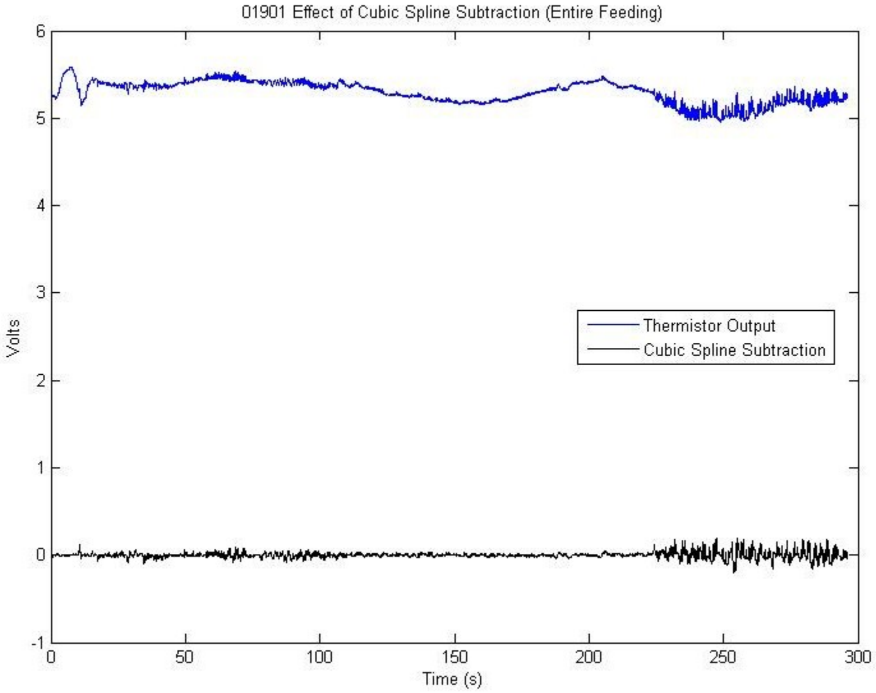

Figure 18: Effect of Cubic Spline Subtraction on Thermistor Signal

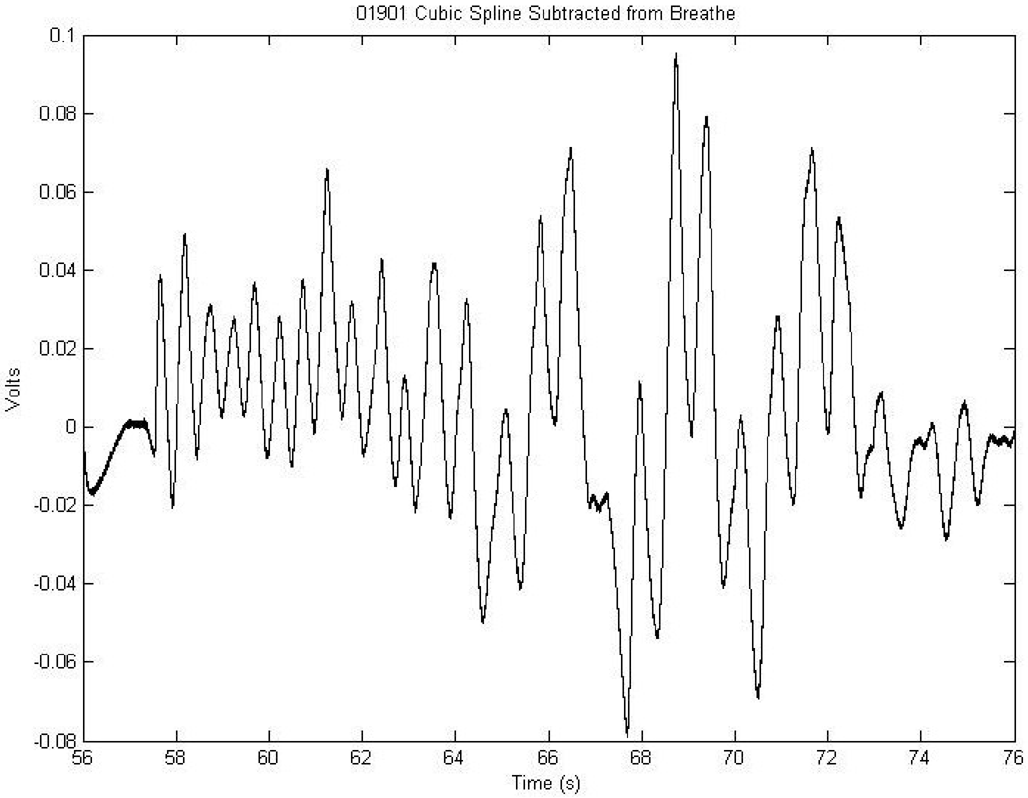

Figure 19: Residual Baseline Drift after Cubic Spline Subtraction

Adaptive Filter (Elman Network) Approximation

An adaptive filter is a filter that changes its coefficients as a function of the input signal. It is self-adjusting and can use many different types of training algorithms. Figure 20 shows a standard arrangement for an adaptive filter. The input (x[n]) goes to both an unknown system (H) and to the finite impulse response (FIR) filter (W). The filter's output (y[n]) is compared to the unknown system's output (d[n]), which is the desired signal. An error is calculated (e[n]) and the coefficients of the FIR filter are adjusted accordingly.

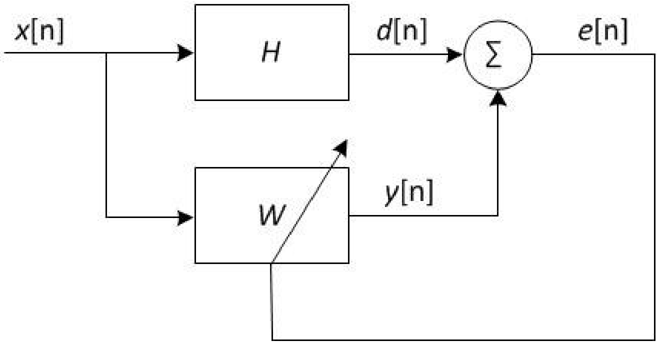

Figure 20: Diagram of an Adaptive Filter [27]

The most commonly used form of an adaptive filter uses the least mean-square algorithm. That approach was inappropriate for this set of data as it assumes the underlying process is stationary and requires that the solution space to be linearly separable. The adaptive filtering approach chosen to approximate the underlying baseline wander embedded in the thermistor output signal used the Elman neural network architecture. This was chosen because it is a recurrent neural network, which makes it suited to handle time series data, and because its architecture was developed for the purpose of amplitude detection. Recurrent neural networks store information in their hidden nodes that impact subsequent training epochs.

The goal of the network was to track the gross signal amplitude changes to approximate the baseline drift for subsequent removal. Two inputs went to a hidden layer of 10 nodes. The inputs were a 10 second sample of the thermistor output signal and an amplitude approximation signal that served as the network target. The 10 second data segment was chosen because MATLAB was unable to process longer segments of data. The network was trained with the data over 1000 epochs, and the final resulting weights were used to approximate the data.

The neural network did not reach convergence. Convergence was defined as occurring at the epoch after which the error rate stayed below 0.01. Figure 21 shows the mean squared error of the Elman network plotted by epoch number. The final error rate did not reach the target error rate of 0.01. Figure 22 shows the original data, the amplitude approximation signal that served as the network target, and the actual network output. From this figure it is clear that the neural network approach was not practical for the removal of signal drift. The output of the neural network could not be analyzed using the event detection algorithm because a final approximation of the baseline drift was never achieved, so it was not possible to subtract the network approximation from the original breath signal.

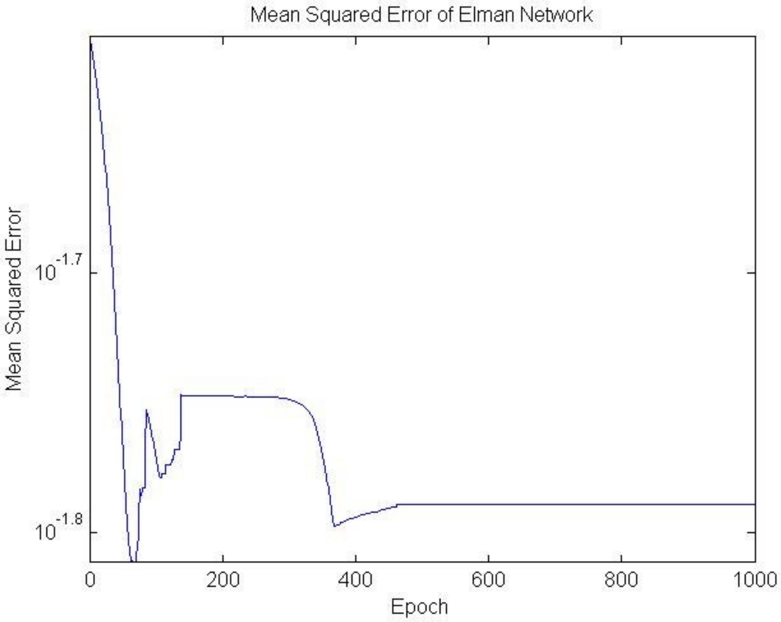

Figure 21: Mean Squared Error of the Elman Network Training

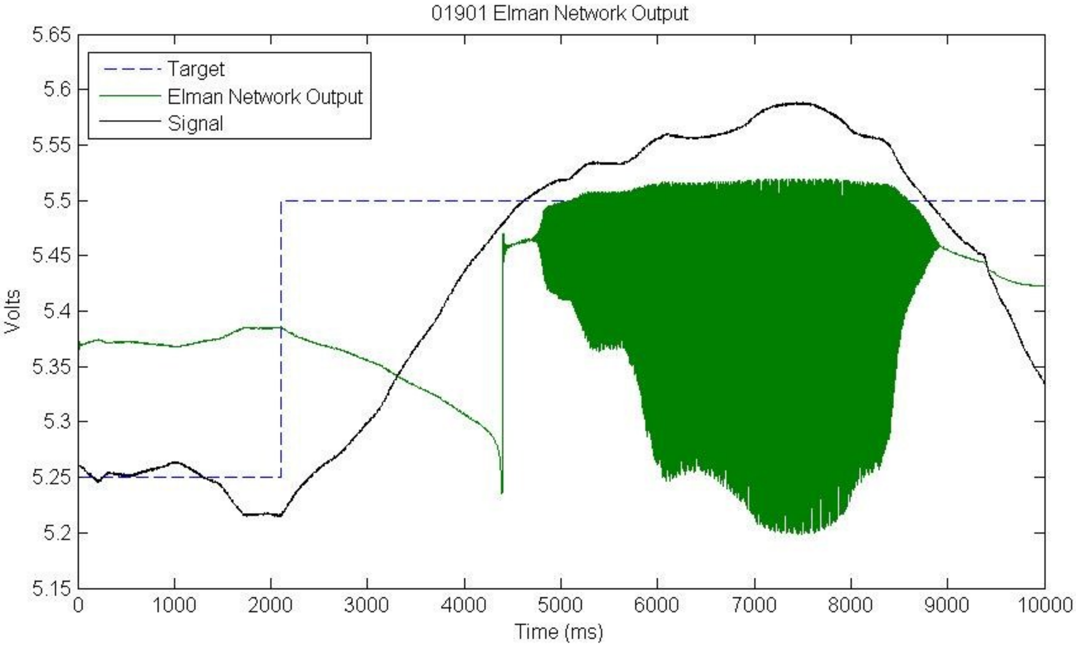

Figure 22: Comparison of the Thermistor Signal, Amplitude Approximation, and Neural Network Output

Second Derivative Based Signal Modeling Approach

Breathing is a nonstationary process [28]. As such, breathing signals can be modeled as the sum of two sinusoids and a constant.

Equation 1: Model of Breathing Signal

$$A = A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t) + C$$

The constant is the DC offset. One sinusoid is the actual breathing waveform, with a frequency of approximately 1 Hz, and it is superimposed over the second sinusoid, of a much lower frequency, which is the baseline drift. When the first derivative of the signal is calculated it removes the DC offset.

Equation 2: First Derivative of Breathing Signal

$$A' = A_1\omega_1 \cos(\omega_1 t) + A_2\omega_2 \cos(\omega_2 t)$$

The second derivative does not completely remove the baseline drift, but it attenuates the amplitude to near zero.

Equation 3: Second Derivative of Breathing Signal

$$A'' = -A_1\omega_1^2 \sin(\omega_1 t) - A_2\omega_2^2 \sin(\omega_2 t)$$

Griffiths *et al.* first explored this model of nasal airflow in 2005 [21]. Figure 23 shows example outputs for this approach. From this figure it is clear that the DC component of the original nasal airflow signal is removed in the velocity calculation. It is also clear that the baseline drift attenuates to near zero with the calculation of the acceleration signal. For these reasons, the second derivative modeling approach was selected for the removal of baseline drift in the thermistor nasal airflow signals.

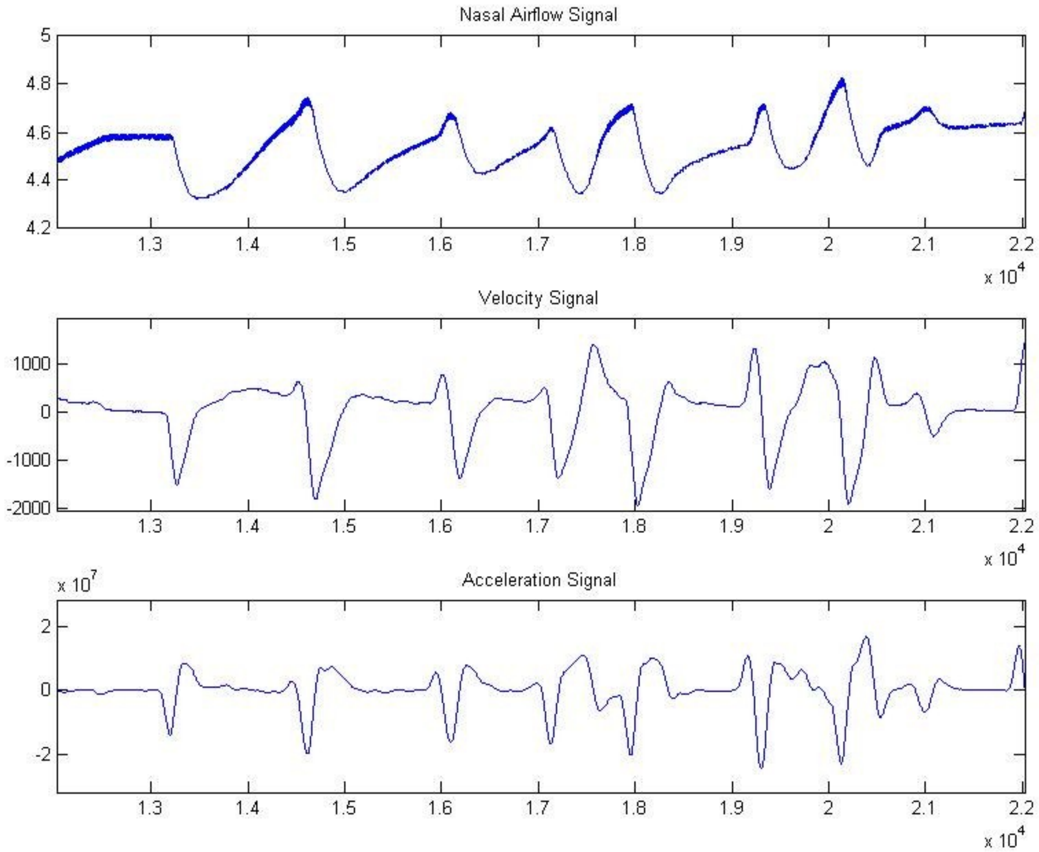

Figure 23: Comparison of Each Stage of the Algorithmic Flow

Analysis of Accuracy of Second Derivative Signal Modeling Approach

Figure 24 shows the algorithmic flow used. First the velocity signal was calculated from the original digitized thermistor output signal using a 20 point central difference method. The velocity signal was then smoothed using a 100 point rectangular window moving average technique. The same steps were repeated for the calculation and smoothing of the acceleration signal. The number of points chosen in both the derivative and smoothing techniques were empirically chosen. The resulting acceleration signal was filtered to remove any remnants of high frequency noise with a 10th order Butterworth low pass filter. This filter and order were chosen out of the need prevent amplitude distortion while minimizing the amount of introduced phase distortion. The filter cutoff frequency was 10 Hz, which was selected to prevent the distortion of event transitions, while leaving the bulk of the spectral content of the signal (which is primarily from 0 to 2 Hz) unaffected. The filtered acceleration signal was then used as the input to the event detection algorithm previously described. Positive peaks were located as they mark the onset of inhalation while negative peaks mark the onset of exhalation. The outputs were then compared to an expert's opinion. A match was described as being within 400 ms of the expert detected point. The data files used to test the algorithm were chosen such that each was free of any noise source not directly related to the signal acquisition process (free of data collection anomalies).

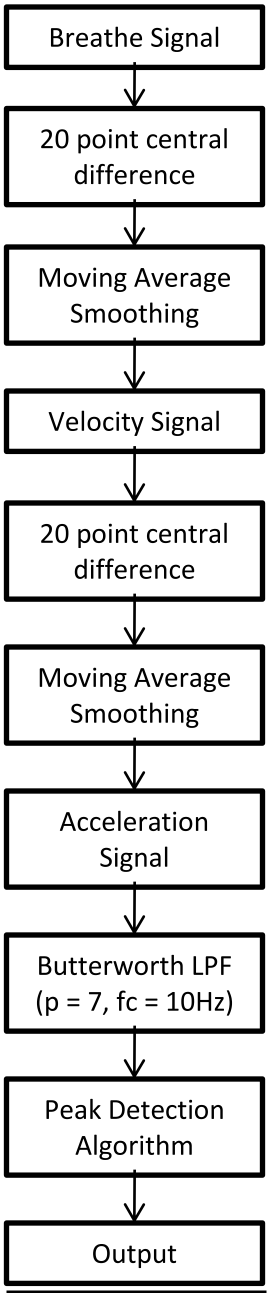

Figure 24: Algorithmic Flow

Baseline Drift Attenuation Method Comparison Results

Figure 25 shows the signal outputs at each stage of analysis. It is clear that the velocity signal does not contain the signal offset that is contained in the original thermistor output signal. Furthermore, the acceleration signal has less baseline drift and resembles the original signal's morphology more accurately, with only a slight phase shift. Table 4 shows a comparison of three of the methods described. While the cubic spline approximation approach detected more points than the linear approximation method, it had a higher Type I error rate. Type I errors consisted of points detected by the expert that were not detected by the algorithm. The most accurate method is the second derivative method (Type I Error = 18%).

Table 5 shows the results for the analysis of accuracy of the second derivative method. The differentiated signal modeling approach combined with the described event detection algorithm yielded an average accuracy rate of 78%.

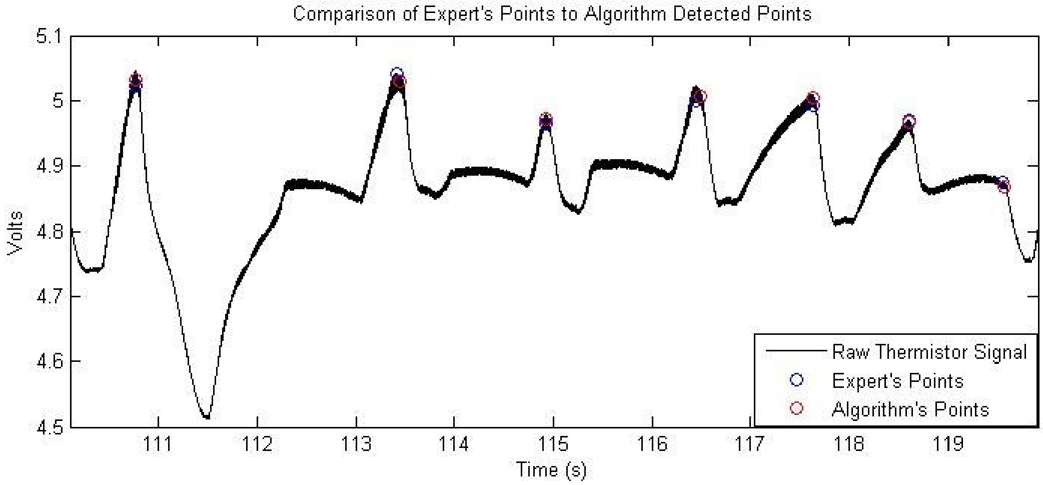

Figure 25: Comparison of Algorithmic Event Detection Using Second Derivative Method and Expert-Detected Points

Table 4: Comparison of Each Method's Output to Expert's Findings for File 01901

Method	Expert's # Points	Detected # Points	% Type I Error
Linear Interpolation	304	251	54%
Cubic Spline	304	277	59%
Second Derivative	304	308	18%

Table 5: Analysis of Second Derivative Method

Subject	Expert's # Points	Algorithm's # Points	% Type I Error
1205	233	277	4%
1599	179	187	16%
01901_alt	304	308	18%
2098	324	321	17%
2398	353	260	37%
2498	240	288	36%
2599	217	197	21%
02698_alt	247	240	39%
2798	320	258	27%
7112	299	370	8%
AVG	271.6	270.6	22%

Swallow Event Detection

Pharyngeal Structure and Function

Effective swallow is crucial to feeding. Swallowing is the transfer of a bolus from the oral cavity to the gastrointestinal tract. The act of swallowing is essentially a pumping motion in which the tongue can be considered an incompressible muscular hydrostat [29]. There are two functional components to a swallow: an oral phase and a pharyngeal phase, also referred to as oral transfer and esophageal transfer [29]. The oral phase is considered to be largely voluntary and can vary greatly based on factors like taste, environment, hunger, and motivation. In the oral phase the bolus is formed in the central groove of the tongue and then propelled backwards. The pharyngeal phase is the actual swallow response. Swallowing is a complex motor event. The pharyngeal phase is initiated by the elevation and retraction of the soft palate, resulting in the closure of the nasopharynx. This is followed by laryngeal displacement and close at the epiglottis level. Then the upper esophageal sphincter (UES) must relax in order to open. Effective oral feeding requires that enough intrabolus pressure must be generated to open the UES [30]. Only after all of these conditions are met can bolus propulsion occur. The pharyngeal phase ends when the bolus has cleared the pharynx in its entirety.

In order for swift and safe swallow to occur, the swallowing mechanism must be an adaptive process to accommodate variables like bolus size and respiratory requirements [3]. One of the primary requirements for adaptation is that is that the UES opening must accommodate bolus size with regard to both opening dimension and duration. The pharynx is mechanically optimized for larger bolus volumes. When the tongue is stressed there is an increase in intrabolus pressure and increase in the vigor of bolus expulsion. One of the primary ways bolus size is accommodated is through a gradient of tongue actions. However, as bolus volume changes the significance of the roles played by the various anatomical components changes as well. With larger volumes the tongue plays a more important role in bolus propulsion, while with smaller volumes the pharyngeal constrictors are more important.

There are 5 main functional elements of swallow: laryngeal closure, nasopharyngeal closure, UES opening, bolus propulsion, and pharyngeal clearance. Malfunction of any of these functional elements leads to different feeding obstructions. Improper laryngeal closure leads to aspiration. Incomplete nasopharyngeal closure leads to nasopharyngeal regurgitation. If the UES opening does not occur completely or adequately, it can lead to dysphagia, post swallow aspiration, or diverticula formation. Improper bolus propulsion leads to a sluggish, misdirected bolus, and incomplete pharyngeal clearance leads to post swallow residue and aspiration.

The inherent variability in the swallow response makes it difficult to establish normative values for the duration or the timing of the events that comprise swallow activity. Of all of these inconsistencies, the most consistent is the duration and propagation of the contraction by the pharyngeal constrictors associated with pharyngeal clearance. For this reason Kahrilas suggests that any timeline of the subevents of swallow activity must start with the end of the swallow event instead of the beginning [31].

Earlier literature assumed that swallow induced peristalsis was complete [32]. Since it has been found that while persistalsis can be complete in preterm infants, the process remains immature at term [32]. Swallowing changes as infants grow older. Rommel *et al.* (2011) made two significant biomechanical observations with respect to how swallowing changes with infant age [33]. First there is a reduced pharyngeal peak pressure about 1 cm above the upper esophageal sphincter. Second, they observed changes in the amount of time it takes the UES to relax. This indicates the development of consistency in this motor mechanism. An indicator of swallow maturation is swallowing rate [6].

Goldfield *et al.* found that swallow coordination is organized around patterns of relative phase [29]. They also found that the increased and decreased lag between different anatomical structures may be a means for changing the efficiency of conversion of muscular effort to mechanical action. The tongue and soft palate movements are in antiphase, creating space for the bolus while blocking the nasal passages. This activity is believed to enhance the piston like movement that initiates swallow [29].

Deglutition Apnea

A primary requirement for safe and effective swallow is the cessation of breath activity. This is referred to as the obligatory deglutition apnea [7]. Swallows generally vary in duration from around 350 ms to 700 ms [7], so the deglutition apnea must respond and adapt to the factors that affect swallow duration. In order for the deglutition apnea to occur, the respiratory generator in the brainstem must be temporarily suppressed [7](Barlow). Without proper coordination, the deglutition apnea can be risky for infants [6]. To decrease risk, swallow must occur during a safe phase of respiration [6]. Unlike term infants, preterm infants favor swallowing during the inspiratory phase of respiration, and it is believed that air suction during inspiration helps propel the bolus down the pharynx [3]. The general observed pattern for swallow-respiration coordination is: inhale, swallow apnea, and exhale (Barlow). This pattern prevents aspiration of residual food in the pharynx [7].

Swallowing is controlled through a network of cortical areas that share loci with other ororhythmic movements, like speech [7]. The stimulation of oral and pharyngeal sensory afferents during suck can stimulate swallow, however in the absence of peripheral sensory input, interneurons in the brainstem can generate a basic swallow pattern [7]. In neonates, the control mechanisms for the striated and smooth muscle in the esophagus are not fully developed [32].

Measurement

Due to anatomical constraints and underdeveloped musculature, measuring swallow in infants has been very difficult to achieve. Measurement attempts have included a pressure drum placed over the hyoid [6] [3], invasive micomanometry [30], pharyngeal pressure measurements made with a pressure transducer embedded within the nasogastric feeding tube [13], multiple intraluminal impedance measures (MII) [34], and video fluoroscopy [29]. In addition to these methods, Barlow mentions the use of microphones, accelerometers, and motor evoked potentials. Most of the measurement methods are invasive and require the placement of a measuring device inside of the infant. The non-invasive methods, such as a pressure drum placed over the hyoid, have shown inconsistent results at best.

Currently there is no widely accepted method to accurately measure swallow events with repeatability. This is due largely to constraints imposed by the anatomy of the infant. The underdeveloped musculature of the preterm infant generates a weak swallow signal. Measurement of this weak signal is further complicated by sensor placement limitations. The area of interest is on the scale of only a few centimeters. Finally, the skin on the neck of infants is so loose that even good placement doesn't ensure proper contact. Often you can see a slippage of the sensor in relation to the muscle group generating the signal of interest.

The first method applied was the use of electromyography (EMG) with electrodes placed over the hyoid. While this seemed promising in theory, and in adult models, in application there was too much noise and too high impedance. Also, optimal sensor placement was difficult due to the previously stated anatomical realities.

The next attempted method involved the use of a piezoelectric sensor referred to as "the yellow pad" (Dymedix). The yellow pad was able to pick up on what appeared to be a swallow signal in low noise environments when coupled with correct sensor placement. The negative aspects of the yellow pad were that it was very sensitive to electrical noise and that it was both too stiff and too large. The stiffness and size of the yellow pad contributed to a "cuffing" effect, where the sensor would form into a cuff-like shape causing decreased contact and inhibiting the infant's range of movement.

After seeing the capabilities and limitations of the yellow pad, other commercially available piezoelectric sensors from the same company were explored. The first in this series was the Pediatric Snore Sensor, a self-adhesive piezoelectric sensor that was used with a lateral orientation directly over the hyoid. In comparison to the yellow pad, the Pediatric Snore Sensor was significantly more compliant. The sensor's compliancy coupled with its self-adhesive nature enabled better adherence to the region of interest. Unfortunately the Pediatric Snore Sensor proved to be a little too small to ensure consistent placement. Also, with the increased compliance of the sensor's backing material led to an increase in sensitivity to common noise sources. These include motion artifact and electrical interference.

Ultimately the Adult Snore Sensor was found to be the most viable commercially available sensor. Like the Pediatric Snore Sensor, the Adult Snore Sensor was self-adhesive and far more compliant than the yellow pad, but unlike the pediatric sensor, the adult sensor was larger, enabling more consistent placement across subjects. Furthermore, an exploration into the nature of the piezoelectric material led to the realization that optimal sensor orientation was longitudinal (in reference to the main axis of the infant's body) and slightly offset to the hyoid. This sensor and methodology combination yielded the highest quality and most consistent swallow signals. Unfortunately, the signal acquired with the Adult Snore Sensor is still largely undecipherable.

There are four main contributing factors to the quality of the swallow signal. These are motion artifact, electrical noise, placement, and orientation. Motion artifact is the result of any movement unrelated to the movement associated with the signal of interest. Examples include neck flexion, extension, or rotation during feeding. Figure 1 shows an example of swallow data that likely consists of the underlying true event signal overlaid with various noise sources, including motion artifact. Electrical noise comes from various sources. Power line interference is readily identifiable and easily addressed. More complex is the fact that the NICU is full of wirelessly transmitted electrical signals. The full extent to which they affect the signal is largely indeterminable due to the lack of information regarding the specific nature of these signals. Orientation and placement of the sensor both impact the time domain shape of the acquired signal and subsequently the frequency components. While orientation and placement were optimized after the implementation of the Adult Snore Sensor, the various iterations of experimental determination for optimum application greatly limit the amount of consistent data available for analysis.

In addition, each iteration of measurement attempts that involved a change in the placement of the piezoelectric sensor or its orientation impacted the shape of the signal. Data experts were unable to determine unique signal characteristics that could be used for event detection parameters. The potential for event determination based on activity in other feeding signals, primarily suck and breath signals, was explored. It was found that there were no readily identifiable time domain or frequency domain event characteristics. Due to the lack of event characteristics, swallow data was not analyzed.

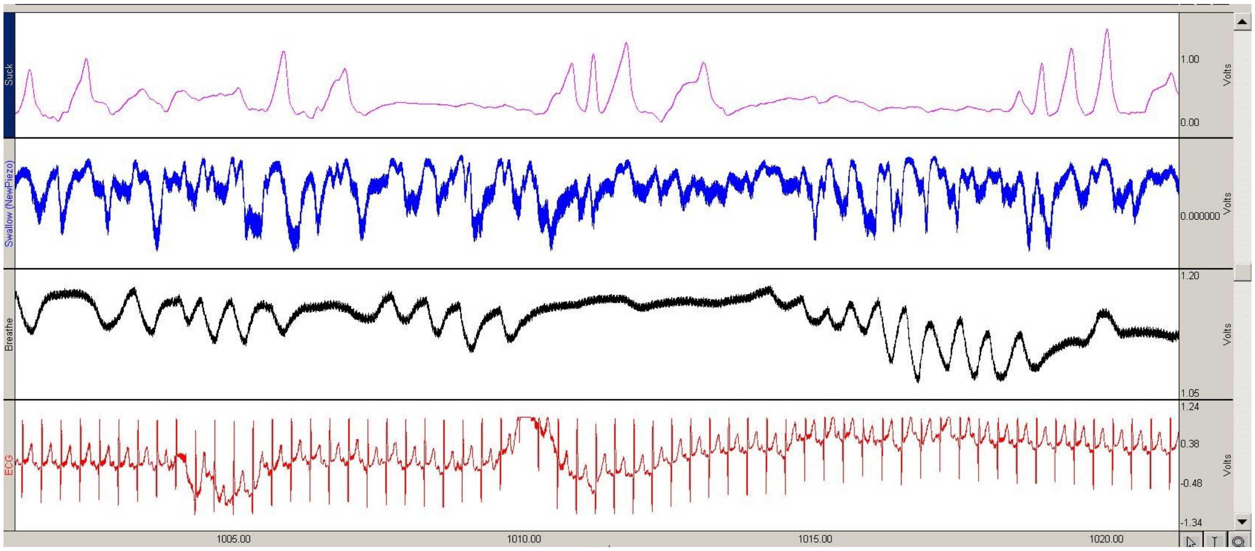

Figure 26: Example of swallow data. The swallow signal (seen in the second row) often exhibits unidentifiable noise that is indistinguishable from the true signal.

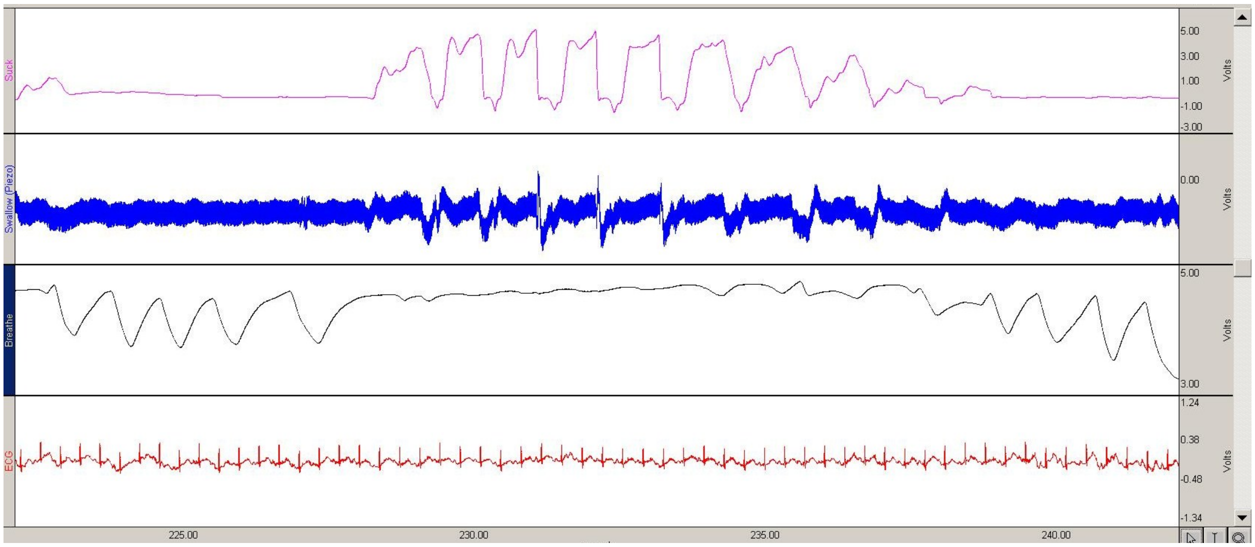

Figure 27: Example of swallow data. This is a common example of a noisy acquired swallow signal.

Heart Rate Variability (HRV) Analysis

Background

Heart rate variability (HRV) is a noninvasive measurement of cardiac and autonomic nervous system output and functionality. It has been widely used as a measure of behavioral state in both clinical and research settings. HRV is not measured directly. Instead it is calculated from an electrocardiogram (ECG) signal. The general process involves the repeated detection of a specific point of the cardiac wave, and then calculating the distance between each successive occurrence. The most commonly used characteristic is the R peak, and the distances between each successive occurrence is referred to as R-R intervals. A signal is generated from the R-R intervals, and the variation of that signal is what is referred to as heart rate variability.

Until the 1970s, heart rate variability was considered noise by most scientists [35]. It wasn't until after that time that HRV was shown to correlate to the neurological control system. Heart rate variability is controlled by several mechanisms, all of which are under the control of the autonomic nervous system. These include the respiratory sinus arrhythmia and blood pressure Mayer waves as well as overall autonomic responsiveness, or how quickly the autonomic nervous system is able to adapt to changes [36]. Another possible influence on HRV is the presence of electromagnetic fields produced by infant incubator motors [37]. HRV exhibits both short and long term complexities. Nonlinear methods can be used to extract information regarding complexity [38]. The average R-R interval is around 350ms in preterm infants, which corresponds to a heart rate of about 170 bpm [36]. It has been shown that decelerations in heart rate precede acute neonatal illness [36]. Flower *et al.* found a previously uncharacterized oscillatory heart rate deceleration pattern in preterm infants [36]. The ratio of low frequency to high frequency power of the R-R interval signal is often reported as a measure of autonomic stability as it represents the balance between the parasympathetic and sympathetic components of the autonomic nervous system. Kreuger *et al.* found that from 28 to 34 weeks the relative influence of the parasympathetic branch of the autonomic nervous system on cardiac activity increases linearly [39].

Methodology

The algorithmic flow diagram for heart rate variability (HRV) processing is depicted in Figure 28. After acquisition, the data was imported into MATLAB where a bandpass filter was applied. Filter specifications: FIR, 5-15 Hz, Tukey Window. The filtered files were saved as text files.

Originally R peak extraction was performed using a program called GetHRV, written by Dr. Wetzel. This program generated an output file containing all of the R-R intervals as well as locations of R Peaks. Each output file had an associated log file that had to be checked for data irregularities. When the program reached a portion of data that it could not process, it simply stopped running, and the only way to learn of that occurring was to manually examine the log files. In instances where the program could not process the entire file, the filtered version of that particular data file had to be opened in ACQKnowledge, adjusted manually (if possible), and resaved. Then the program had to be rerun for each such file. Finally, all the R-R intervals files had to be loaded into MATLAB for frequency analysis.

Fairly early on it was evident that this process was too time consuming. Instead a streamlined approach was developed using a modification of an open source R-peak extraction program. The modified R-peak extraction program ran within MATLAB, was able to process files in batch, and generated “.mat” files that were faster to save and load within MATLAB. The R-peak detection portion of the program used a template based matching approach, where portions of high correlations corresponded to a template match, and therefore an R-peak detection. The generated output file consisted of R-R intervals and R-peak locations. This was used to create an unevenly sampled secondary signal.

A cubic spline was applied to the signal to even the sampling distance. During this process the signal was upsampled to 200 Hz. The signal was then downsampled to 2 Hz to provide the frequency resolution needed to accurately see the shape of power spectral density (PSD) curve. Binned frequency analysis was done using the Yule-Walker autoregressive method. This method was chosen because the ECG signal is nonperiodic and because it is the most prevalent parametric PSD estimation method in literature. The power contained within the low frequency and high frequency bins as well as the ratio of low frequency to high frequency were computed, and all output measures were written to a single output summary file in text format.

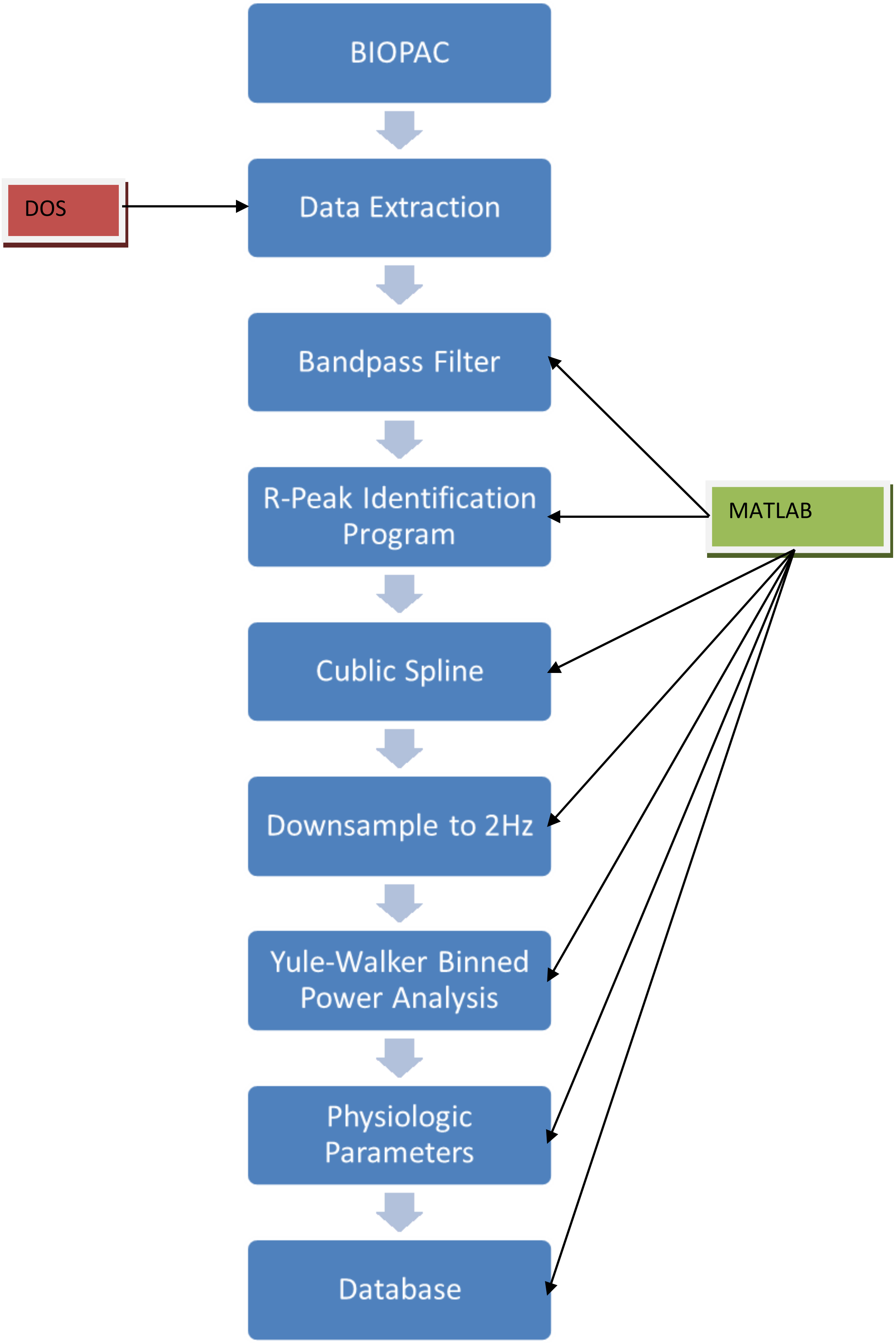

Figure 28: Algorithmic Flow Diagram for HRV Analysis

Interactions Respiration and HRV

Respiration is a main contributor to the high frequency peak in HRV power spectra. Respiratory induced heart rate variability is referred to as respiratory sinus arrhythmia (RSA). Many studies have explored the interaction between respiratory rate and heart rate variability [40] [41] [42] [43] [44] [45] [46]. Heart rate generally increases during inspiration and decreases during expiration, due to the negative relationship between respiratory rate and vagal power [47]. The sino-atrial node (SA node) can be considered a self-sustaining oscillator which resets upon receiving a volley of vagal stimuli, prolonging the time between heartbeats [47]. The vagus activity is inhibited during inspiration and disinhibited during expiration [47].

In preterm infants cardiac output is largely dependent on heart rate, and spontaneous respiratory efforts are irregular [48]. Rassi and Indic have both illustrated the presence of synchronous oscillations of heart rate and respiration in preterm infants, but both have shown that the interaction is inconsistent at best [48] [49]. Effective cardiac-respiratory system integration is a hallmark of development in preterm infants and is often used as a contributing criterion for release from the neonatal intensive care unit [16] [13]. Indic found that the respiratory patterns of preterm infants were irregular with a wide range of periodicities interspersed with frequent breath apnea [49]. Most studies have focused on time domain interactions and only used spectral estimations as secondary measurements of temporal systemic relationships [48] [49] [16] [13]. Any relationship between these signals in the time domain should also be evident in the frequency domain. As such, the interaction of respiration and HRV in the frequency domain was examined for a subset of the subject population.

Data from the first sample were from five minute observations of 10 preterm infants (9 African American, 1 Caucasian) obtained during bottle feedings once clinical feeding competency had been achieved. All subjects were in sinus rhythm. Half of the subjects were male and half were female. Average weight at time of observation was 2.3 kg. After receiving written informed parental consent, data were acquired using methods approved by the VCU IRB. The subjects used for this study were a subset of a larger study of preterm infant feeding (NIH R01NR005182, Pickler, PI).

Infant respiration was measured indirectly from nasal airflow measurements obtained from a thermistor implanted in nasal cannula. ECG signals were obtained with three electrodes in a standard Type I arrangement. The data were acquired using the BIOPAC^{TM} MP150 data acquisition system. All data were sampled at a rate of 1000 Hz. The ECG signal was bandpass filtered from 5 to 15 Hz to remove signal drift and artifact using the BIOPAC^{TM} AcqKnowledge software. R-peaks were identified using a standard moving average threshold approach, developed by the authors, and R-R intervals were calculated. The corresponding respiration signal was processed in MATLAB. A second derivative modeling approach was used as the basis for breath event detection. The second derivative was calculated to eliminate signal offset while attenuating the drift to near zero. The resulting waveform was a model of the true nasal airflow signal. Breath events were defined as the onset of inhalation. After detection, the time between breath events was calculated to generate a signal similar to the R-R interval signal used in HRV analysis, called the respiratory tachogram.

The tachograms are plots of interval duration (seconds) by time of event occurrence (seconds). After DC removal, both the heart and respiratory rate tachograms were interpolated to 200 Hz using a cubic spline approach (MATLAB function SPLINE) to compensate for the uneven spacing of the samples. The signals were then downsampled to 2 Hz. Power spectral densities (PSD) were estimated using the Yule-Walker AR parametric estimation method

(MATLAB function PYULEAR) with an order of 20 and NFFT = 256. The model order 20 was chosen as recommended by ESC/NASPE Task Force [50]. The correlation between the PSD of the respiratory rate tachogram and the PSD of the heart rate tachogram was subsequently determined for each subject using the MATLAB function CORRCOEF.

Table 6 shows the means and variances of the infant heart and respiratory rate intervals. The average number of heart beats detected during five minutes was N = 825.6 beats, the mean R-R interval duration was 0.35 s with an average variance of 56.1 ms^2. The average number of breaths detected was N = 295.9 breaths, the mean inter-breath interval duration was 1.165 s, with an average variance of 449.0 ms^2.

Table 6: Means and Variances of Infant Heart Rate and Respiratory Rate Intervals

	HR Interval			RR Interval		
Subject	N [beats]	Mean [sec]	Variance [ms^2]	N [breaths]	Mean [sec]	Variance [ms^2]
01599	855	0.352	1.1	576	0.522	1187
02098	912	0.328	2.2	321	0.922	31.8
02398	747	0.402	3.9	368	0.815	23.4
02698	264	0.374	23.4	187	1.569	440.3
02798	805	0.383	36.6	232	1.323	199.8
04598	881	0.348	8.8	426	0.719	36.5
04898	796	0.386	222.4	154	1.762	1079.5
05098	1213	0.247	126.5	220	1.342	390.1
05298	967	0.309	54.8	180	1.650	1017.
06499	816	0.373	1.7	295	1.030	84.2
Average:	825.6	0.35	56.1	295.9	1.165	449.0

Table 7 shows the correlation between the heart and respiratory rate PSDs . The overall results obtained showed statistically significant low correlations for infants (R = 0.44, SD = 0.19, p = 0.002).

Table 7: Heart Rate PSD and Respiratory Rate PSD Correlations by Subject

Subject	R	p
01599	0.53	<0.0001
02098	0.73	<0.0001
02398	0.34	0.0001
02698	0.51	<0.0001
02798	0.22	0.0001
04598	0.26	0.025
04898	0.78	<0.0001
05098	0.33	<0.0001
05298	0.28	0.002
06499	0.39	<0.0001
Mean	0.44	0.002
S.D.	0.19	

The physiological relationships between the cardiac and respiratory systems were not evident in the frequency domain for this sample of subjects. This section explores why the results do not show the relationship between the cardiac and respiratory systems.

The preterm infants used in this study were performing a physically demanding task, bottle feeding, at the time of data collection. The difficulty of the task is manifested in the data in the forms of periods of apnea, irregular breathing bursts, and periods of increased heart rate. Figures 29 and 30 show the heart and respiratory rate raw tachograms for two subjects and their corresponding PSDs. Subject 02098 and subject 02798 respectively represent examples of high and low correlations detected (see Table 7: Heart Rate PSD and Respiratory Rate PSD Correlations by Subject). From Figures 29 and 30 it is evident that irregular breathing patterns shifted the frequency content of the respiratory tachogram, resulting in the low correlations observed.

It should be noted that the measurement technique employed for respiration observation, mainly the use of a thermistor to measure nasal airflow, has limitations that could affect this data. Detected apneic periods may actually be periods of mouth breathing and not true apnea. This is evident in the subjects with a low number of breaths detected (see Table 6: Means and Variances of Infant Heart Rate and Respiratory Rate Intervals). Also the R-R peak detection algorithm was limited by the irregularity of the ECG signal, which may cause extremely high or low R-R intervals (see Table 6: Means and Variances of Infant Heart Rate and Respiratory Rate Intervals). ECG signal irregularities stem from artifact during signal acquisition.

Although the correlations observed were statistically significant, they were modest in strength. These results were consistent with the limitations of the populations being examined. This suggests that the lack of correlation is more likely a result from deviations from normal physiologic functioning than from an underlying inherent disintegration of the cardiac and respiratory systems.

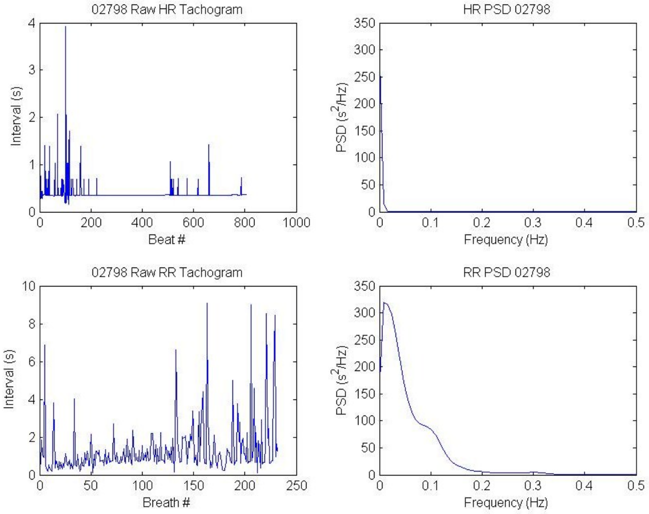

Figure 29: Raw tachograms and PSD comparison for Subject 02798
Irregular heart rate and respiratory patters led to the low PSD correlation exhibited by this subject.

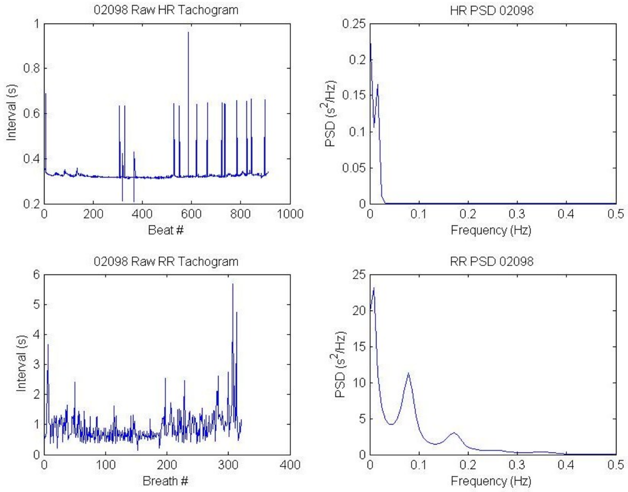

Figure 30: Raw tachograms and PSD comparison for Subject 02098
A far more regular heart rate and respiration rate led to the high correlation exhibited by this subject.

Coordination Chapter

Background

Feeding coordination can be thought of as existing in two levels. The first level is comprised of only the temporal relationships between feeding events, resulting in overall feeding ratios that have been shown to be indicative of infant maturation. The second level of coordination integrates the temporal relationships between events with an overall state indicator, in this case heart-rate variability (HRV), providing a measure of the overall level of effectiveness for the safe transference of food to the digestive tract.

Safe oral feeding is defined as coordination of feeding events such that the aspiration of food is minimized [6] [7]. Roughly 40% of the patients in feeding clinics are preterm infants [7]. The primary factors that affect preterm feeding are: gestational age, muscle tone, heart rate, respiratory state, behavioral state, energy level, and sucking behaviors [4]. It has been found that the quantitative assessment of feeding patterns may enable the prediction of general neurological impairment as well as resulting feeding dysfunction [13]. As such, feeding patterns have been speculated to be a predictive measure of short and long-term feeding and developmental difficulties [51] [7]. Better feeding outcomes are most strongly predicted by feeding experiences [4]. This is because preterm infants do learn from environmental experiences, so the quality and quantity of bottle feeding experience can impact their transition from nasogastric feeding to oral feeding [4]. Also, the timing of the introduction of oral feeding has been shown to contribute to the organization of early sucking patterns [15].

Effective feeding coordination requires the maturation and synchronization of muscles within each feeding function as well as the safe coordination between each feeding function. In this context, synchronization is limited to include only those interactions of muscles within the same group of musculature, as defined by the event (suck, swallow, breathe) [6]. Coordination refers to the interactions of muscles between the different musculatures associated with each event [6]. Effective feeding coordination can be considered as achieved when the infant can take oral feedings with no sign of aspiration, oxygen desaturation, apnea or bradycardia [7]. Typically accepted feeding event coordination ratios are (suck:swallow:breath) 1:1:1 or 2:2:1 [7]. Successful coordination of suck/swallow/breathe may be an indicator of neurological maturation in preterm infants [4]. The three main feeding events are tightly coupled motor behaviors. As such, coordination is essential to effective feeding. Suck, swallow, and respiration and their coordination mature at different rates [6].

Traditionally, ex utero maturation has been treated as equivalent to in utero maturation despite environmental differences [6]. This trend is slowly reversing through the discovery of new research regarding the effect of environmental stimuli on the maturation of feeding skills. Feeding maturation in preterm infants has been shown to be enhanced by early introduction to oral feeding [15]. The overall maturation process would suggest the presence of a dynamic neural control mechanism to regulate the closely occurring feeding events [7].

Suck-swallow maturation can be used as an index of overall feeding readiness. It consists of the ratio of sucks to swallows, and the intervals between peak suction and the onset of swallow and between peak expression and the onset of swallow [6]. The presence of an immature swallow reflects immature sucking ability and an immature state of the coordination of suck/swallow/breathe events [7]. As suck-swallow maturation progresses, there is an increase in the stability of suck rhythms and an increase in the aggregation of suck and swallow events into "runs" [51]. This pattern supports the idea that rhythmic stability is a sign of maturation. Suck or swallow runs are defined as 3 or more events in succession with interevent intervals being less than 2 seconds [51]. The ideal rate of suck and swallow runs is one event per second with an event occurrence ratio of 1:1 [6].

Swallow-breathe maturation is mainly determined by the respiratory phase during which swallows occur, with the primary sign of maturation being the occurrence of swallow events during safer phases of respiration [6]. The idea behind this is that as preterm infants mature, they should have less feeding apnea [7]. Before coordination can be assessed, though, proper event detection for the individual feeding events must occur.

Coordination is defined as the emergence of order from interactions among component parts, and it occurs at multiple levels of the motor system [29]. Coordination patterns can be based on muscle activation patterns or on kinematic variables. Rhythmically moving components tend to influence each other. In accordance with this, Goldfield's description of coordination as being when "components enter stable patterns that persist temporarily under specific conditions, resist perturbation under certain kinds of change, and then rapidly reorganize when conditions exceed task-specific boundaries" seems to be the most fitting description of infant feeding coordination. During infant feeding, the various individual components must occur within rapid succession of each other while accommodating changes in respiration, bolus size, and overall physiologic state.

Methodology

Coordination was defined as overlapping temporal patterns of suck and breath events. Three different methods of generating a measure of coordination were investigated before one was chosen. The first method was to examine coordination within instances of burst activity. Burst activity was defined as periods of consecutive events in which events are less than 2 seconds apart. As part of the parameter outputs for both the suck and breath signals bursts were determined. This method of coordination assessment was not chosen because it was found that suck bursts do not necessarily coincide with breath bursts, making this a measure that could not be uniformly defined or applied.

The second method investigated was to look at the temporal patterns between event occurrences across signals. The goal would be to determine some sort of pattern of breath and suck event occurrences. This method was not selected because the discrete data generated was not of clinical significance.

The final method, which was ultimately chosen, determined the amount of time spent in overlapping events. Within this approach there were two different ways to assess total time spent feeding. The first method, referred to as total feeding time, consisted of the sum of the individual feeding intervals, as extracted and marked by the event files. The second method, referred to as total time spent in feeding activity, consists of a subset of total feeding time defined as the total amount of time during which sucking or breathing activity occurred.

This coordination assessment was made using the validated outputs of the breath and suck event detection algorithms, and as such was not independently validated. Binary vectors were composed for each feeding behavior. Each vector started as a 1 by n array of zeros (n= number of sample points in the original full feeding file). In the suck vector, 1's replaced the zeros for all sample points during which sucking activity occurred as determined by the previously obtained event onset and end times. In the breath vector, 2's replaced zeros for all sample points between an onset of inhalation and an onset of exhalation. A coordination vector was made by adding across the elements of the individual event vectors. In the coordination vector zeros correspond to instances where neither sucking nor breath activity is occurring, ones correspond to the presence of sucking activity alone, twos correspond to the presence of only breath activity, and threes mark the points where suck and breath both occur. From the coordination vector a series of event parameters were calculated and saved in an external file. All three vectors were combined to make a coordination matrix, which was saved for future analysis. The individual event parameter files were combined into a summary text file. This process is summarized in Figure 31.

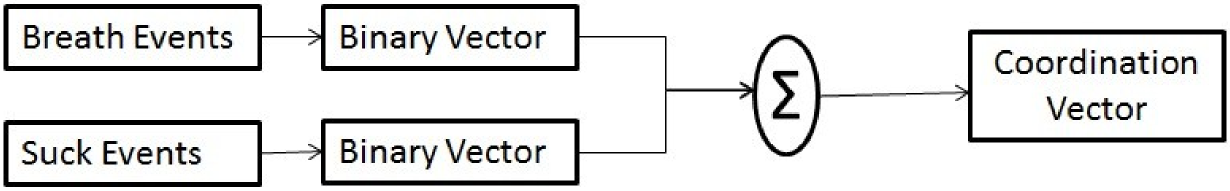

Figure 31: Coordination Flow Diagram

Results

Overall, 1101 full feeding files were processed using the coordination algorithm. Files were chosen by the availability of both suck and breath event files. This list was further limited by the exclusion of all event files that contained less than 3 events. Files were not excluded for any other reasons. Of the 1101 files processed, only 85 correspond to discharge feedings, where infants were assumed to be competent feeders. Anecdotally, it was observed that there was actually a higher percent overlap of feeding activity out of full feeding time in the discharge feedings, although assessment of the significance of this difference was biased by the disproportionately small sample size of discharge feedings. The mean proportion of time spent in feeding activity that was spent in overlapping activity was found to be 0.58 (SD = 0.186), and the mean percent of total feeding time spent in overlapping events was found to be 12.6% (SE =

0.37). This means that on average over half of the time spent in feeding activity consisted of overlapping events while that same amount of time made up less than 13% of the total feeding time.

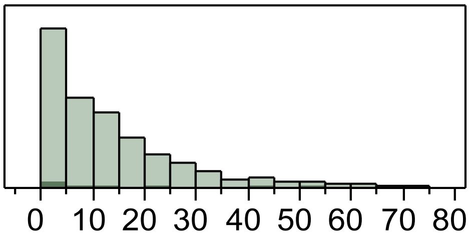

Figure 32: Distribution of Percent Overlap of Full Feeding Time (mean = 12.6, SD = 12.27)

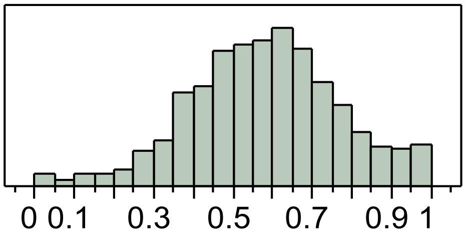

Figure 33: Distribution of Proportion of Time Spent in Overlapping Feeding Activities out of Total Time Spent in Feeding Activity (mean = 0.58, SD = 0.186)

Discussion

The primary purpose of this project was to use the event detection outputs to generate a measure of coordination, and as such the basis for coordination measurement was the location and duration of individual feeding events. In order to assess coordination, event detection algorithms were developed. Both the final algorithms for breath and suck event detection were the result of an evolutionary process in which signal models were iteratively updated.

The suck algorithm evolved in relation to the understanding, and therefore modeling, of the suck signal. As the nuances and varied characteristics of the signal were established, they provided valuable parameters for event detection. Unlike the measurement of the other feeding signals, measurement of sucking behavior was largely uniform in implementation. This was achieved through the standardized application of a singular measurement device. Despite this high level of measurement consistency, there were still data inconsistencies. These were largely due to differences in infant sucking patterns and capabilities. In some instances, signal saturation occurred when the detected oral cavity pressure exceeded the pressure transducer's maximum detected pressure. In other cases infant sucking activity was so weak that many events were not greater than the detection threshold and therefore went undetected.

While the characterization of suck event types provided insight into favored behaviors, there were suck events that were not explored. Of particular interest would be the small events with maximum amplitudes below the detection threshold. These events were referred to as "nibbles" by the primary data collector. The activity seemed to indicate some transference of fluid but without the generation of the pressure necessary for a true suck. This may be a form of expression without suction, or a learned behavior in which an infant uses their tongue to extract milk from a bottle nipple without exerting the muscular effort to generate suction pressure.

There were several factors to consider when examining preterm infant respiration signals obtained with thermistors during bottle feedings. One such factor was the heating of sensor surroundings from the temperature of the formula used during feeding. This could be a contributing factor to signal drift seen by the thermistor. Another factor to consider was the placement of the sensor on the infant. If the thermistor was not placed in close proximity to the infant's nostril, the acquired signal would be of very low amplitude and have a low signal to noise ratio. Sometimes infant movement during feeding caused the sensor to slip in relation to the nostril. The final major factor that affected signal quality before acquisition was the placement of the thermistor within the nasal cannula. If the sensing portion of the thermistor was too far inside the cannula, the sensor would have a low signal to noise ratio and low signal amplitude resulting from a decrease in the variations of the temperature. A portion of the sensing part of the thermistor could also end up contained within the hot glue, which would have a detrimental effect on the overall function of the sensor by insulating the sensor from variations in temperature, increasing thermal mass. If the thermistor was too far outside of the cannula, a lot of artifact could be introduced from excessive movement of the sensor or from the sensor not being within the flow of airway currents. All of these factors contributed to the quality and consistency of the signal that was obtained during data acquisition.

For the breath event detection algorithm, four methods of baseline removal were compared. The results show that the second derivative modeling approach was far more accurate than the linear approximation or cubic spline approximation, and the Elman network was never able to generate an approximation of baseline drift. While the double differentiation technique was the most effective method at removal of baseline drift, there were several weaknesses to this approach. One such weakness was when the original signal had a portion of low amplitude event occurrences and a portion of high amplitude even occurrences. Here the algorithm failed to detect events because they fell below the detection threshold. To compensate for this weakness, the number of events detected could be compared to an average preterm infant respiratory rate. If the value is significantly under a reasonable level, then it can be identified as potentially erroneous. The second derivative signal can be plotted, and a more appropriate threshold can be visually determined and implemented.

Another weakness of this approach was that there was no way to distinguish true apnea from instances of mouth breathing. Also, when the algorithm indicated the presence of an apneic episode, there is a chance that it could be that the amplitude of the signal was so low it passed below the algorithmic threshold, which can indicate shallow breathing and does not necessarily indicate a full cessation of breath. Both of these are due to the placement of the thermistors outside of the nasal cavity. While there are other ways to measure breath and respiration, this thermistor and nasal cannula technique is the least invasive and least cumbersome. As such it is best to address these weaknesses by simply being mindful of them during data analysis and interpretation.

Anecdotally, it was observed that at times the double differentiation of the original signal made it possible to detect breath events that were hidden in the noise of the original signal, meaning at times this algorithmic approach was able to “salvage” otherwise unusable data. Further evaluation of this approach would benefit from a quantification of this phenomenon. While this double differentiation modeling and algorithmic event detection approach has its downsides, it provided the most effective method of baseline drift removal.

Repeatable, reliable measurement of swallow has yet to be achieved using non-invasive techniques. While swallow parameters exist in the literature, they are primarily the result of invasive measurements, and therefore do not necessarily translate well to external observations. In the Data Acquisition section event markers were explained. It is important to note that there were several feeding observations where data collectors marked swallows as they were observed visually. To mark an event, the event had to be visually confirmed, the appropriate function key on the laptop had to be found, and finally pressed. All of these processes occurred after the event had occurred. The reaction time from visual confirmation to the motor execution of pressing the button was greater than the event duration. This led to an unpredictable degree of time lag between the true time of event occurrence and the time marked by the event marker, which made the event markers less than useful in post collection data analysis. It is worth noting, however, that because visual confirmation of swallow by trained nurses is possible, there remains hope for the development of a consistent, non-invasive measurement technique, provided the visual cues for swallow detection can be translated into a measurement apparatus.

Heart rate variability processing proved to be surprisingly difficult. The most glaring issue was ECG signal noise. Most commonly this appeared in the form of movement artifact that rendered portions of the ECG signal useless through muscle activity overlap and signal saturation. Another frequent data issue that occurred was an inversion of the signal, which consistently caused failure across all the attempted R-peak detection methods. This happened when the order of the two chest leads was reversed. The final source most common source of signal noise was baseline drift, but this was easy to attenuate through traditional filtering techniques because the frequency components of the signal components of interest (the R-peaks) were clearly different than those of the baseline drift. Any time shift that may have been introduced in the application of digital filters during data processing posed no concern to the final HRV result. This is because the final measures of HRV were calculated as a descriptive summary of the signal in its entirety, as opposed to the individual events. In addition, the timing of the individual events did not have to be rematched to feeding events for any type of analysis, so minute time shifts did not have the ability to exert larger effects on further steps of data processing. A potential way to avoid a lot of the HRV signal processing difficulties would be to use a cardiac tachometer. This would be less susceptible to noise while providing a pulse signal from which HRV can be determined. Also, the full characteristics of an ECG signal are not needed for HRV analysis. HRV only needs one consistently detectable feature of the ECG waveform, in this case the R-peaks, to generate a tachogram. Since the tachogram is the signal that is ultimately processed for HRV analysis, a cardiac tachometer would provide enough information while bypassing the ECG signal processing phase of HRV analysis.

The interactions between HRV and respiratory rate variability (RRV) were examined as another assessment of neurological maturation. In a fully mature infant, one would expect to see the cardiac and respiratory systems changing in relation to each other to compensate shifts in activity. In this case, however, it was observed that even at discharge feedings in which feeding competency was assumed to exist, the correlations between HRV and RRV were relatively low. This seemed to indicate that neurologic integration had yet to be fully achieved in these subjects.

The purpose of this project was to develop a quantifiable measurement of coordination. To that end the coordination algorithm proposed achieved this goal by quantifying the degree of overlap of feeding events, namely sucks and breaths. There are limitations to this approach, however. The most significant limitation stems from the way in which breath events were defined. A breath event was defined as starting at the onset of inhalation and ending at the onset of exhalation. This time span can include breath holds or deglutition apneas. It would be more accurate to detect two separate breath events, inhalation and exhalation, and count the breath activity as occurring only during those particular phases of respiration. Unfortunately, given the measurement device employed, this is not possible. The rapid response time of the thermistor employed yields a very accurate representation of inhalation and exhalation, which can contain rapid changes in signal amplitude. However, the rapid response time of the thermistor prevents it from depicting the steady state no flow output during breath holds and deglutition apneas. During those instances the response of the thermistor decays and/or rises to ambient temperature. Due to the properties of the thermistor employed, the acquired signal most accurately represents respiration during periods of rapid or constant breathing and yields less accurate representations of breath during periods when there is a high likelihood of cessation of breath for any reason.

There were several limitations to the event detection algorithms. The suck event detection algorithm was limited by its inability to adapt to rapid changes in baseline. It was also limited by the static threshold values. Future work related to event detection should include an adaptive baseline as well as an adaptive threshold. In addition, an exploration of the different types of suck events would be interesting. One study claimed that the change in direction at the peak of the Type B and Type C events is the result of pressure changes from swallow activity. Future work may be able to identify such events based on duration and then use them as indirect markers of swallow activity.

Another possible future exploration could look at the time decay properties of the thermistor used, with a focus on detection of breath holding behavior. Since the thermistor measures the change in temperature between inhalation and exhalation, it would be interesting to see how rapidly the sensor changes its output in the absence of airflow.

Of utmost importance, regarding the event detection attempts, is the development and testing of a piezoelectric swallow measuring device. A custom sensor that optimizes stiffness, uniform placement, and orientation (to maximize the advantage of the sensing material's mechanical properties), should be developed. In addition, testing must be completed to determine time domain characteristics of the swallow signal, like shape and duration. Only after the signal can be modeled can an event detection algorithm be developed.

Immediate future work should look at the interactions of maturity and the coordination measure as well as the interactions of maturity and HRV. Those interactions are part of the primary goals of the PRO Study. In addition, once significant interactions between maturity and other feeding and state variables have been determined, it could be possible to develop some sort of scale or indicator of coordination.

Long term contributions of this work to the literature include providing support to the dynamic systems approach (DSA) to the development of cognition and action in early childhood. Current theories of development assume an end state before development has even occurred. For example, in the case of this work the assumed end state would be a particular pattern of coordination. From that it is assumed that all individual subjects are developing towards a common goal. This developmental rigidity is also rooted in the use of central pattern generators as a means to explain developmental patterns. DSA varies from this approach in that it builds upon the principals of chaos theory and nonlinear system dynamics to model and assess the inter-subject variance in developmental patterns. So while traditional theories of development are based on a model in which the infant is a system and developmental variance is noise, DSA proposes that this developmental variance is what forms the basis of cognition and as such plays a fundamental role in overall development. While the literature regarding preterm infant maturation is still fundamentally rooted in the traditional theoretical framework, more and more studies seem to be exploring this variance in development. DSA cannot be considered a developmental theory as of yet due to a lack of empirical evidence. This work, however, may provide another description of developmental variance with which may support the dynamic systems approach to development.

Works Cited

[1] R. Russell, N. Green, C. Steiner, S. Meikle, J. Howse, K. Poschman, T. Dias, L. Potetz, M. Davidoff, K. Damus and J. Petrini, "Cost of hospitalization for preterm and low birth weight infants in the United States," *Pediatrics,* vol. 120, no. 1, pp. e1-e9, 2007.

[2] American Academy of Pediatrics Committee on Fetus and Newborn, "Hospital Discharge of the High Risk Neonate," *Pediatrics,* vol. 122, no. 5, pp. 1119-1126, 2008.

[3] C. Lau, E. Smith and R. Schanler, "Coordination of suck-swallow and swallow respiration in preterm infants.," *Acta Paediatrica,* pp. 721-727, 2003.

[4] R. Pickler, A. Best, B. Reyna, G. Gutcher and P. Wetzel, "Predictors of nutritive sucking in preterm infants.," *Journal of Perinatology,* vol. 26, no. 11, pp. 693-9, November 2006.

[5] J. Silber, S. Lorch, P. Rosenbaum, B. Medoff-Cooper, S. Bakewell-Sachs, A. Millman, L. Mi, O. Even-Shoshan and E. GJ, "Time to send the preemie home? Additional maturity at discharge and subsequent health care costs and outcomes.," *Health Research and Educational Trust,* pp. 444-463, 2009.

[6] N. Amaizu, R. Schulman, R. Schanler and C. Lau, "Maturation of oral feeding skills in preterm infants.," *Acta Paediatrica,* pp. 61-67, 2008.

[7] S. Barlow, "Central pattern generation involved in oral and respiratory control for feeding in the term infant.," *Current Opinion in Otolaryngology ad Head and Neck Surgery,* pp. 187-193, 2009.

[8] R. Braglien, W. Rokke and T. Markestad, "Stimulation of sucking and swallowing to promote oral feedign in premature infants.," *Acta Paediatrica,* pp. 1430-1432, 2007.

[9] S. Barlow and M. Estep, "Central pattern generation and the motor infrastructure for suck, respirtation, and speech.," *Journal of Communication Disorders,* pp. 366-380, 2006.

[10] B. Medoff-Cooper, J. Shults and J. Kaplan, "Sucking Behavior of Preterm Neonates As a Predictor of Developmental Outcomes," *Journal of Developmental & Behavioral Pediatrics,* vol. 30, no. 1, pp. 16-22, 2009.

[11] J. Miller and S. Kang, "Preliminary ultrasound observation of lingual movement patterns during nutritive versus non-nutritive sucking in a premature infant.," *Dysphagia,* vol. 22, no.

2, pp. 150-60, 2007.

[12] M. Poore, E. Zimmerman, S. Barlow, J. Wang and F. Gu, "Patterned orocutaneous therapy improves sucking and oral feeding in preterm infants.," *Acta Paediatrica,* vol. 97, no. 7, pp. 920-7, 2008.

[13] F. Vice and I. Gewolb, "Respiratory patterns and strategies during feeding in preterm infants," *Dev Med Child NEurol,* vol. 50, pp. 467-472, April 2008.

[14] I. Gewolb, F. Vice, E. Schweitzer-Kenney, V. Taciak and J. Bosma, "Developmental patterns of rhythmicsuck and swallow in preterm infants.," *Dev Med Child Neurol,* vol. 43, no. 1, pp. 22-27, 2001.

[15] R. Bromiker, I. Arad, B. Loughran, D. Netzer, M. Kaplan and B. Medoff-Cooper, "Comparison of sucking patterns at introduction of oral feeding and at term in Israeli and American preterm infants.," *Acta Paediatrica,* pp. 201-204, 2005.

[16] R. Pickler, A. Best and D. Crosson, "The efect of feeding experience on clinical outcomes in preterm infants," *Journal of Perinatology,* vol. 29, pp. 124-129, Feb. 2009.

[17] B. Medoff-Cooper, W. Bilker and J. Kaplan, "Sucking patterns and behavioral state in 1- and 2-day-old full-term infants.," *J Obstet Gynecol Neonatal Nurs,* vol. 39, no. 5, pp. 519-24, 2010.

[18] J. Mortula, "Breathing patterns in newborns," *J Appl Phsiol,* vol. 56, pp. 1533-40, June 1984.

[19] H. Akre and O. Skatvedt, "Advantages of measuring airflow in the pharynx with internal thermistors.," *Eur Arch Otorhinolaryngol,* vol. 257, pp. 251-255, 2000.

[20] A. Griffiths and A. W. S. S. J. Mau, "Improved detection of obstructive events in childhood speel apnoea with the use of the nasal cannula and the differentiated sum signal.," *J Sleep Res,* vol. 14, pp. 431-436, 2005.

[21] R. Farre, J. Monsterrat, M. Rotger, E. Ballester and D. Navajs, "Accuracy of thermistors and thermocouples as flow-measuring devices for detecting hypopnoeas.," *Eur Resp J,* vol. 11, pp. 179-182, 1998.

[22] F. Primiano, F. Montague and G. Saidel, "Measurement system for respiratory water vapor an dteperature dynamics.," *J Appl Physiol,* vol. 56, pp. 1679-1685, June 1984.

[23] F. Series and I. Marc, "Nasal pressure recording in the diagnosis of sleep apnoea hypopnoea

syndrome.," *Thorax,* vol. 54, pp. 506-510, 1999.

[24] K. Storck, M. Karlsson and P. Ask, "Heat transfer evaluation of the nasal thermistor technique.," *IEEE Trasactions in Biomedical Engineering,* vol. 43, pp. 1187-91, Dec 1996.

[25] F. Afsar, M. Arif and J. Yang, "Detection of ST segment deviation episodes in ECG using KLT with an ensemble neural classifier.," *Physiol Measures,* vol. 29, pp. 747-760, July 2008.

[26] US Sensor, "US Sensor Catalogue," 1 June 2010. [Online]. Available: http://www.ussensor.com/US_SENSOR_CATALOG.pdf.

[27] D. Jones, S. Appadwedula, M. Berry, M. Haun, D. Moussa and D. Sachs, "Adaptive Filtering: LMS Algorithm.," 1 June 2010. [Online]. Available: http://cnx.org/content/m10481/latest/.

[28] J. Fei and I. Pavlidis, "Virtual Thermistor," *IEEE EMBS Proceedings,* pp. 250-253, Aug. 2007.

[29] E. Goldfield, C. Buonomo, K. Fletcher, J. Perez, S. MArgetts, A. Hansen, V. Smith, S. Ringer, M. Richardson and P. Wolff, "Premature infant swallowing: patterns of tongue-soft palate coordination based upon videofluoroscopy," *Infant Behav Dev,* vol. 33, no. 2, pp. 209-18, 2010.

[30] T. Omari, "Apnea-associated reduction in lower esophageal scphincter tone in premature infants," *J Pediatr,* vol. 154, no. 3, pp. 374-8, 2009.

[31] P. Kahrilas, "Pharyngeal structure and function," *Dysphagia,* vol. 8, no. 4, pp. 303-7, 1993.

[32] A. Staiano, G. Boccia, G. Salvia, D. Zappulli and R. Clouse, "Development of esophageal peristalsis in preterm and term neonates," *Gastroenterology,* vol. 132, no. 5, pp. 1718-25, 2007.

[33] N. Rommel, "Phayngo-esophageal motility in neurologically impared children.," *J Pediatr Gastroenterol Nutr.,* vol. 53, pp. S21-22, 2011.

[34] C. Peter, C. Wiechers, B. Bohnhorst, J. Silny and C. Poets, "Detection of small bolus volumes using multiple intraluminal impedance in preterm infants.," *J Pediatr Gastroenterol Nutr,* vol. 36, no. 3, pp. 381-4, 2003.

[35] L. Rodriguez-Linares, A. Mendez, M. Lado, D. Olivieri, X. Vila and I. Gomez-Conde, "An open source tool for heart rate variability spectral analysis.," *Comput Methods Programs*

Biomed, vol. 103, no. 1, pp. 39-50, 2011.

[36] A. Flower, J. Moorman, D. Lake and J. Delos, "Periodic heart rate decelerations in premature infants," *Exp Biol Med,* vol. 235, no. 4, pp. 531-8, 2010.

[37] C. Bellieni, A. M, M. Maffei, S. Maffei, S. Perrone, I. Pinto, N. Stacchini and G. Buonocore, "Electromagnetic fields produced by incubators influence heart rate variability in newborns," *Arch Dis Child Fetal Neonatal Ed,* vol. 93, no. 4, pp. F298-301, 2008.

[38] R. Maestri, G. Pinna, A. Porta, R. Balocchi, R. Sassi, M. Signorini, M. Dudziak and G. Raczak, "Assessing nonlinear properties of heart rate variability from short-term recordings: are these measurements reliable?," *Physiol Meas,* vol. 28, no. 9, pp. 1067-77, 2007.

[39] C. Krueger, J. v. Oostrom and J. Shuster, "A longitudinal description of heart rate variability in 28-34-week-old preterm infants," *Biol Res Nurs,* vol. 11, no. 3, pp. 261-8, 2010.

[40] L. Bernardi, C. Porta, A. Gabutt, L. Spicuzza and P. Sleight, "Modulatory effects of respiration," *Auton Neurosci,* vol. 90, pp. 47-56, July 2001.

[41] T. Brown, L. Beightol, J. Koh and D. Eckberg, "Important influence of respiration on human R-R interval power spectra is largely ignored.," *J Appl Physiol,* vol. 75, pp. 2310-2317, Nov. 1993.

[42] M. Cohen and J. Taylor, "Short-term cardiovascular oscillations in man: measuring and modelling the physiologies," *J PHysiol,* vol. 542, pp. 669-683, Aug. 2002.

[43] D. Eckberg, "Human sinus arrhythmia as an index of vagal cardiac outflow," *J Appl Physiol,* vol. 54, pp. 961-966, Apr. 1983.

[44] M. Poyhonen, S. Syvaoja, J. Hartikainen, E. Ruokonen and J. Takala, "The effect of carbon dioxide, respiratory rate and tidal volume on human heart rate variability.," *Acta Anaesthesiol Scand,* vol. 48, pp. 93-101, Jan. 2004.

[45] T. Unok, M. Grap, C. Sessler, A. Best, P. Wetzel, A. Hamilton, K. Mellott and C. Munro, "Autonomic nervous system function and depth of sedationin adults receiving mechanical ventilation.," *Am J Crit Care,* vol. 18, pp. 42-50, Jan. 20009.

[46] M. Yildiz and Y. Ider, "Model based and experimental investigation of respiratory effect on the HRV power spectrum.," *Physiological Measurements,* vol. 27, pp. 973-988, Oct. 2006.

[47] P. Zhang, W. Tap, S. Reisman and B. Natelson, "Respiration response curve analysis of heart rate variability," *IEEE Transactions on Biomedical Engineering,* vol. 44, pp. 321-325,

Apr. 1997.

[48] D. Rassi, A. Mishin, Y. Zhuraavlev and J. Matthes, "Time domain correlation analysis of heart rate variability in presterm neonates.," *Early Human Development,* vol. 81, pp. 341-350, Apr. 2005.

[49] P. Indic, E. Salisbury, D. Paydarfar, E. Brown and R. Barbieri, "Interaction between heart rate variability and respiration in preterm infants," *Computers in Cardiology,* vol. 35, pp. 57-60, Sep. 2008.

[50] ESC/NASPE (European Society of Cardiology/North American Society of Pacing and Electrophysiology) Task Force, "Heart rate variability: standards of measurement, physiological interpretation and clinical use," *Circulation,* vol. 93, pp. 1043-1065, Sep. 1996.

[51] I. Gewolb, F. Vice, E. Schweitzer-Kenney, V. Taciak and J. Bosma, "Developmental patterns of rhythmic suck and swallow in preterm infants.," *Developmental Medicine and Child Neurology,* pp. 22-27, 2001.

Appendix 1: Protocol for Data Pre-Processing

1. Make a batch file (see ExtractAcq.bat) to run the program in each subdirectory from the root directory.
2. Place the batch file that calls the ExtractACQ program in each subdirectory (see D.bat).
3. Set the associated .inf file to calculate intervals based on the clean text event files.
4. Use ZTree to insert "-0000" between the subject i.d. and the "clean.txt" in each clean text file's filename, so that the event files all have the same root name as the associated ACQ files.
5. Open the DOS prompt and direct it to the root directory.
6. Call the first batch file (ExtractAcq.bat). This should result in the generation of all of the event files (in a text format) for each feeding file. Event files are appended with a capital E and the number of the event (for example "_E01.txt" would mark the first interval of a feeding file).
7. Return the clean text files to their original naming structure using ZTree.
8. Open each subdirectory and make a list of the directory contents from the dos prompt.
9. Use this list to make a concatenation file (see concat.bat) that generates the FULL files, consisting of all of the event files concatenated together.
10. Create a batch file for the root directory that will be used to open each subdirectory and run the concatenation file.

Appendix 2: Suck Event Detection and Parameter Generation

Filename: skBatch3.m

```
%Last Modified: 10/24/11
%Changes made: Run E01 and Full files separately (without matching)


directory = 'E:\SkFileCorrection';
cd(directory);
file1 = dir('*_E01.txt');
fileF = dir('*_FULL.txt');


  for n = 1:length(file1);
    InputFileName = file1(n).name;
    skMaster7(InputFileName)
  end
  for n = 1:length(fileF);
    FullFileName = fileF(n).name;
    skMaster8(FullFileName)
  end
```

Filename: skMaster7.m

```
function skMaster7(InputFileName)
%% Get event output for all E01 files
alldata = load(InputFileName);
suck = alldata(:,1);
```

```
events = skIdent3(suck);
%events = reshape(events,numel(events),1);
outname = [InputFileName(1:14) '_SuckEvents.mat'];
%fid = fopen(outname, 'w');
%fprintf(fid, '%u\n', events);
%fclose(fid);
save(outname, 'events');
```

Filename: skMaster8.m

```
function skMaster8(FullFileName)
%% Get event output for all full files
fid = fopen(FullFileName);
alldata = textscan(fid,'%f %f %f %f');
fclose(fid);
suck = alldata(:,1);
suck = cell2mat(suck);
events = skIdent3(suck);
%events = reshape(events,numel(events),1);
outname = [FullFileName(1:14) '_SuckEvents.mat'];
%fid = fopen(outname, 'w');
%fprintf(fid, '%u\n', events);
%fclose(fid);
save(outname,'events');
```

Filename: skIdent3.m

```
%Last edit: 10/11/11
```

```
%Chages between skIdent3 and skIdent2:
%added for loop (line 98) to cell "get rid of overlapping stops/starts"
%to accommodate instances where a middle event onset is lost during
%processing.
%things to add: what if we're missing more than one start value?
function [events] = skIdent3(suck)
%% Smoothing
%w = 201;                                              %length of window
%m = (1:(w-1)/2);                                        %symmetry of window
%suck_sm = zeros(length(suck),1);
%for n = ((((w-1)/2)+1):(length(suck)-((w-1)/2)));                %point of interest
%    M1 = suck(n-m);
%    M2 = suck(n+m);
%    suck_sm(n) = (sum(M1)+sum(M2)+suck(n))/(w*.001);
%end


%% Find Baseline
m = 10000;
n = round(numel(suck)/10000);
if round(numel(suck)/10000)*m < numel(suck)
   n = round(numel(suck)/10000);
elseif round(numel(suck)/10000)*m > numel(suck)
   n = round(numel(suck)/10000)-1;
end
sk = suck(1:n*m);
sk = reshape(sk,m,n);

a = mode(sk);


%check to make sure there isn't too much baseline variation

for n2 = 2:(length(a)-1);

   c = abs(a(n2) - a(n2-1));

   if c > 0.5;

       a(n2) = a(n2-1);

   end

end


a = repmat(a,m,1);

base = reshape(a,m*n,1);

b = suck((m*n+1):numel(suck));

b = ones(length(b),1);

b = b.*base(numel(base));

base = [base; b];


%% Zero Signal

thresh = base + std(suck);

suck_z = suck;

suck_z(suck_z < thresh) = 0;




%% Zero Begining & End of signal
```

```
suck_z(1) = 0;
suck_z(end)=0;

%% Detect Onsets & Ends (without for loops)

a = ~suck_z;
a = diff(a);
onsets = find(a == -1);
ends = find(a == 1);

%% Find true event starts & stops
start = zeros(length(suck),1);
for n = 1:length(onsets); %indexes onsets
  ref = onsets(n); %first reference point is the first value in onsets
  for m = 1:ref; %indexes points from beginning of signal to reference point
    if ref==1; % if the first point in onsets is the first sample point, the first start is the first sample point
      start(ref) = 1;
    elseif suck(ref) > suck(ref-1);  %if the signal value at the reference point is greater than the point before it
      ref = ref-1; %shift the reference point one position left
    elseif suck(ref)<=suck(ref-1); %if the signal value at the reference point is greater than or equal to the preceding point
      start(ref-1)=1; %mark a start at the preceding point
    end
  end
```

```
end


stop = zeros(length(suck),1);
for n = 1:length(ends);
    ref = ends(n);
    for m = 1:ref;
        if ref==length(suck);
            stop(ref) = 1;
        elseif suck(ref) > suck(ref+1);
            ref = ref+1;
        elseif suck(ref)<=suck(ref+1);
            stop(ref+1)=1;


        end
    end
end


%% get rid of overlapping stops/starts
%introduces the condition that one event must end before another begins
e=find(start);
f=find(stop);
if length(e) == length(f)
    events = [find(start) find(stop) find(stop)-find(start)];
elseif length(e) < length(f)
    for n3 = 1:length(e); %added on 10/11/11
```

```
        if e(n3) < f(n3);
            f(n3) = f(n3);
        elseif e(n3) >= f(n3); % what if we're missing more than one value?
            f(n3) = f(n3+1);
        end
    end
    f = f(1:length(e));
    events = [e f f-e];
elseif length(e) > length(f)
    e = e(1:length(f));
    events = [e f f-e];
end
a = events(2:end,1);
b = events(1:(end-1),2);
c=a-b;
c = c<0;
events(find(c)+1,1) = events(c,2);


%% Push Stops to next Start during burst sucking activity
%introduces the condition that events less than 1s apart are continuous
o = events(1:(length(events)-1),2);
p = events(2:length(events),1);
q = find(p-o<1000);
r = q+1;
events(q,2) = events(r,1);
```

```
%% Get rid of all events < 200 ms
%introduces assumption that event last a minimum of 200ms
events = events(events(:,3)>200,:);
```

Filename: skParamBatch.m

```
%This is the batch processing file for calculating all of the suck
%physiologic parameters from the suck event files. Calls functions
%"suckparam1" and "suckParam2".
%Input: suck event files
%Output: parameter files
%Last Edit: 4/27/11

directory = 'E:\SkFileCorrection';
cd(directory);
file1 = dir('*_E01_SuckEvents.mat');
fileF = dir('*_FUL_SuckEvents.mat');

for n = 1:length(file1);
   InputFileName = file1(n).name;
   suckparam3(InputFileName);
end

for n = 1:length(fileF);
   FullFileName = fileF(n).name;
   suckParam4(FullFileName);
```

```
end
```

Filename: suckParam4.m

```
%% suckParam4
%Suck Parameters - Full File
%calculates  parameters for:
%total number of sucks
%sucking duration (ms)
% # bursts
% #sk/burst
% mean burst duration

%Last Edit: 10/7/11
%changes: load function, input data type (.mat)

%% Function
function suckParam4(FullFileName)
%% get data
events = load(FullFileName);
events = events.events;

%% fix event durations
events(:,3) = events(:,2)- events(:,1);

%% total number of sucks
tot_sk = length(events(:,1));
```

```
%% sucking duration
suck_dur = sum(events(:,3));


%% find bursts
if tot_sk <= 2;
   num_bur = 0;
   sk_per_bur = NaN;
   mean_bur_dur = NaN;
elseif tot_sk > 2;
   o = events(1:(length(events)-1),2); %event ends
   p = events(2:length(events),1); %event starts following previous event ends
   %r = find(p-o < 2000); %finds events that are less than 2s apart
   %s = diff(r)>1; %marks burst borders
   s = find(p-o > 2000); %finds where the distance from one end to the next start is greater than
2s
   burStart = events((s+1),1);
   burStart = [events(1,1); burStart];
   burEnd = events(s,2);
   burEnd = [burEnd; events(length(events),2)];


   % number of bursts
   num_bur = length(burStart);


   % number of sucks/burst
```

```
    sk_per_bur = length(events)/num_bur;

    % mean burst duration
    mean_bur_dur = mean(burEnd-burStart);
end
%% write output files
skParam = [tot_sk suck_dur num_bur sk_per_bur mean_bur_dur];
outname = [FullFileName(1:14) '_SuckParam.txt'];
fid = fopen(outname, 'w');
fprintf(fid, '%f %f %f %f %f\n', skParam);
fclose(fid);
```

Appendix 3: Breath Event Detection and Parameter Generation

Filename: BreathProc2.m

```
%% Breath Event File Generator & parameter fix
%Last Updated: 2/6/12

%% Directory Setup
files = dir('*_FULL.txt');
load bwfilter.mat

%% Batch Processing
for n=1:numel(files);

      InputFileName=files(n).name;

      result = importdata(InputFileName);

  colnum = length(result(1,:)');
      if colnum == 4
            result = result(:,3);
      elseif colnum == 5
            result = result(:,4);
      end
```

```
        %% Breath Processing Algorithm (Double Differentiation)

n = (11:length(result)-10);
vel = (result(n+10)-result(n-10))/(20*.001);

        %vmovavg
w = 101;
m = (1:(w-1)/2);
vavg = zeros(length(vel),1);
for n = ((((w-1)/2)+1):(length(vel)-((w-1)/2)));
    M1 = vel(n-m);
    M2 = vel(n+m);
    vavg(n) = (sum(M1)+sum(M2)+vel(n))/(w*.001);
end

n = (2:length(vavg)-1);
acc = (vavg(n-1)-vavg(n+1))/(8*.001);

%movavg
w = 101;
m = (1:(w-1)/2);
avg = zeros(length(acc),1);
for n = ((((w-1)/2)+1):(length(acc)-((w-1)/2)));
    M1 = acc(n-m);
    M2 = acc(n+m);
    avg(n) = (sum(M1)+sum(M2)+acc(n))/(w*.001);
end

avg_filt = filter(bwfilter,avg);

        %% Event Detection: Onset of Inhalation
%combined breathmax5 and breaths7

maxis = ones(length(avg_filt),1);
maxs = zeros(length(avg_filt),1);
minis = ones(length(avg_filt),1);
mins = zeros(length(avg_filt),1);

center = mean(avg_filt);
upper_thresh = center + std(avg_filt);
lower_thresh = center - std(avg_filt);

for n = (1:length(avg_filt));
    if avg_filt(n) < upper_thresh %threshold
        maxis(n) = 0;
    end
end
            a = find(maxis); %indices of avg where there is probably a maximum
b = diff(a); %find breaks between max segments
```

```
m = (1:length(b));

d = find(b(m)~=1);

e = d-1;

e = e';

e = [e (length(a))];

e = a(e);%interval end

f = d + 1;

f = f';

f = [1 f];

f = a(f);%interval begin

        for m = (1:length(f));

                    x = avg_filt(f(m):e(m));

    y = max(x);

                    ind = find(x==y);

                    maxs(f(m)+ind) = 1;

        end


        breath = find(maxs);

breath = breath(diff(breath)>200);


        %sequential exhalation onset detection


        for o = 1:(length(breath));


                    if o < length(breath);
```

```
            sig = (avg_filt(breath(o):breath(o+1)));
            z = min(sig);
            ind2 = find(sig==z);
            mins(breath(o)+ind2) = 1;
        elseif o == (length(breath));
            sig = (avg_filt(breath(o):length(avg_filt)));
            z = min(sig);
            ind2 = find(sig==z);
            mins(breath(o)+ind2) = 1;
        end

    end

    breath2 = find(mins);
 events = [breath breath2];
 %output file
 name = InputFileName(1:15);
    name = [name '_BrEvent'];
 save(name,'events');

    %% Rob's Function
 %column names:
'#Breaths','#BreathBursts','AvgBreaths/Burst','MeanBurstDur','IBIRange','IBIMean','IBIStdDev','
1stOnset','1stBreaths/Burst','1stBurstDur','5minBreaths/Burst','5minMeanBurstDur'

    %Total Feeding Statistics
```

```
      BreathParam = zeros(1,12);

      %numBreaths = length(breath);

  interval = breath(2:end)-breath(1:end-1);

  bursts = find(interval>=2000);

  if bursts~=0

    totBursts = length(bursts(:))-1;

    numBreaths = length(breath((bursts(1)+1):bursts(end)));

    BrPerBurst = numBreaths/totBursts;

    avgBstDur = mean(breath(bursts(2:end))-breath(bursts(1:end-1)));

    IBIs = interval(bursts(:));

    IBIRange = max(IBIs)-min(IBIs);

    IBIMean = mean(IBIs);

    IBIStdDev = std(IBIs);

    %BreathParam(1:7) =
(numBreaths,totBursts,BrPerBurst,avgBstDur,IBIRange,IBIMean,IBIStdDev);

    %First Burst Statistics

    onset1 = breath(bursts(1)+1)-breath(1);

    BrPerBurst1 = bursts(2)-bursts(1);

    BurstDur1 = breath(bursts(2))-breath(bursts(1)+1);

    %BreathParam(8:10) = (onset1,BrPerBurst1,BurstDur1);

  end

      %First Five Minutes Statistics

      E01_name = InputFileName(1:10);

      E01_name = [E01_name '_E01.txt'];

      E01 = load(E01_name);
```

```
        E01_end = length(E01(:,1));


    fiveMin = 0;
    if (breath(end)-breath(1)) > E01_end
        for i = 2:length(breath)
            if fiveMin == 0
                dur = breath(i)-breath(1);
                if dur > E01_end
                    fiveMin = i;
                end
            end
        end
        bursts5 = find(interval(1:fiveMin)>2000);
        BrPerBurst5 = length(breath(bursts5(1)+1:bursts5(end)))/(length(bursts5(:))-1);
        avgBurstDur5 = mean(breath(bursts5(2:end))-breath(bursts5(1:end-1)));
        %BreathParam(11:12) = {BrPerBurst5,avgBurstDur5};
    else
        %BreathParam(11:12) = (NaN,NaN);
                    BrPerBurst5 = NaN;
                    avgBurstDur5 = NaN;
    end
        %end Rob's Function


        parameters = [numBreaths totBursts BrPerBurst avgBstDur IBIRange IBIMean
IBIStdDev onset1 BrPerBurst1 BurstDur1 BrPerBurst5 avgBurstDur5];
```

```
        namebeforedot = InputFileName(1:15);

        outname = [namebeforedot '_BreathParam.txt'];

    fid = fopen(outname,  'w');

    fprintf(fid, '%f %f %f %f %f %f %f %f %f %f %f %f\n', parameters);

    fclose(fid);


end
```

Filename: Breath.m

```
%% Directory Setup

%files = dir('*.txt') in command window

%load bwfilter.mat

FULL = files(142:end);

%assembles sampling pool


%% Batch Processing by 'files'

for n=1:numel(FULL);


    InputFileName=FULL(n).name;

    result = importdata(InputFileName);

    result = result(:,3);

    %specifiying the column for BreathFilter


    n = (11:length(result)-10);

    vel = (result(n+10)-result(n-10))/(20*.001);
```

```
%% Start of Breath Processing Algorithm
%vmovavg
   w = 101;
   m = (1:(w-1)/2);
   vavg = zeros(length(vel),1);
   for n = ((((w-1)/2)+1):(length(vel)-((w-1)/2)));
      M1 = vel(n-m);
      M2 = vel(n+m);
      vavg(n) = (sum(M1)+sum(M2)+vel(n))/(w*.001);
   end

   n = (2:length(vavg)-1);
   acc = (vavg(n-1)-vavg(n+1))/(8*.001);

   %movavg
   w = 101;
   m = (1:(w-1)/2);
   avg = zeros(length(acc),1);
   for n = ((((w-1)/2)+1):(length(acc)-((w-1)/2)));
      M1 = acc(n-m);
      M2 = acc(n+m);
      avg(n) = (sum(M1)+sum(M2)+acc(n))/(w*.001);
   end

   avg_filt = filter(bwfilter,avg);
```

```
%breathmax5
    maxis = ones(length(avg_filt),1);
    maxs = zeros(length(avg_filt),1);
    center = mean(avg_filt);
    thresh = center + std(avg_filt);
    for n = (1:length(avg_filt));
        if avg_filt(n) < thresh %threshold
            maxis(n) = 0;
        end
    end
    a = find(maxis); %indices of avg where there is probably a maximum
    b = diff(a); %find breaks between max segments
    m = (1:length(b));
    d = find(b(m)~=1);
    e = d-1;
    e = e';
    e = [e (length(a))]; %interval end terms
    e = a(e);
    f = d + 1;
    f = f';
    f = [1 f]; %interval begin terms
    f = a(f);
    for m = (1:length(f));
        x = avg(f(m):e(m));
        y = max(x);
        for l = (1:length(x));
            if x(l) == y
                                                for p = 0:200;
                    if (f(m)-1)+l < 200;
                        maxs((f(m)-1)+l) = 1;
                    elseif maxs(((f(m)-1)+l)-p) == 0;
                                                                        maxs((f(m)-1)+l) = 1;
                                                            end
                                                end
            end
        end
    end
    breath = find(maxs);


%Rob's Function
    BreathParam = cell(2,12);
    BreathParam(1,1:12) =
{'#Breaths','#BreathBursts','AvgBreaths/Burst','MeanBurstDur','IBIRange','IBIMean','IBIStdDev'
,'1stOnset','1stBreaths/Burst','1stBurstDur','5minBreaths/Burst','5minMeanBurstDur'};
%Total Feeding Statistics
%numBreaths = length(breath);
    interval = breath(2:end)-breath(1:end-1);
    bursts = find(interval>=2000);
    if bursts~=0
        totBursts = length(bursts(:))-1;
        numBreaths = length(breath((bursts(1)+1):bursts(end)));

        BrPerBurst = numBreaths/totBursts;

        avgBstDur = mean(breath(bursts(2:end))-breath(bursts(1:end-1)));

        IBIs = interval(bursts(:));

        IBIRange = max(IBIs)-min(IBIs);

        IBIMean = mean(IBIs);

        IBIStdDev = std(IBIs);

        BreathParam(2,1:7) =
{numBreaths,totBursts,BrPerBurst,avgBstDur,IBIRange,IBIMean,IBIStdDev};

        %First Burst Statistics

        onset1 = breath(bursts(1)+1)-breath(1);

        BrPerBurst1 = bursts(2)-bursts(1);

        BurstDur1 = breath(bursts(2))-breath(bursts(1)+1);

        BreathParam(2,8:10) = {onset1,BrPerBurst1,BurstDur1};

    end

%First Five Minutes Statistics

    fiveMin = 0;

    if (breath(end)-breath(1)) > 300000

        for i = 2:length(breath)

            if fiveMin == 0

              dur = breath(i)-breath(1);

              if dur > 300000

                    fiveMin = i;

              end

            end
        end
        bursts5 = find(interval(1:fiveMin)>2000);
        BrPerBurst5 = length(breath(bursts5(1)+1:bursts5(end)))/(length(bursts5(:))-1);
        avgBurstDur5 = mean(breath(bursts5(2:end))-breath(bursts5(1:end-1)));
        BreathParam(2,11:12) = {BrPerBurst5,avgBurstDur5};
    else
        BreathParam(2,11:12) = {'Not 5 Mins. Long!','Not 5 Mins. Long!'};
    end
%end Rob's Function


parameters = [numBreaths totBursts BrPerBurst avgBstDur IBIRange IBIMean IBIStdDev
onset1 BrPerBurst1 BurstDur1 BrPerBurst5 avgBurstDur5];
 format bank
 parameters;
namebeforedot = InputFileName(1:16);
    outname = [namebeforedot '_BreathParam.txt'];
    %save(outname, 'parameters', '-ascii'); TOOK THIS OUT
    fid = fopen(outname,  'w');
    fprintf(fid, '%f %f %f %f %f %f %f %f %f %f %f %f\n', parameters);
    fclose(fid);


end
```

Appendix 4: Heart Rate Variability Analysis & Parameter Generation

Filename: hrvProc.m

```
%Batch Processing File


%Calls subfunction "detectBeatsPR"


files = dir('*_E01.mat'); %Change input file type to process 1st interval & Full files.
def_dir=what; def_dir=def_dir.path;
cd(def_dir)
%% process files
  for n= 1:length(files);
    ecgin = load(files(n).name);
    ecgin = ecgin.data;



%% Detect RR-Intervals
  [iR,ibi]=detectBeatsPR(ecgin);



  %% write file
  outname = files(n).name;
  outname = outname(1:14);
  outname = [outname '_ibi'];
  save(outname,'ibi')

  end
```

Filename: detectBeatsPR.m

```
function [iR,ibi]=detectBeatsPR(ecgin)


            % Detect beats

    ut=0.4;

    lt=0.35;

    %warning off;

    template = load('sample_template.mat'); %load template

    template = template.template;

    signal = ecgin(:,2);

    %signal = decimate(signal,4); %decimate sampling rate to 256Hz to match template's
sampling rate.

  %% Match Template

  if size(signal,1)<size(signal,2) %convert to vector

    signal=signal';

  end

  if size(signal,2)>1 %discard time column if present

    signal=signal(:,2);

  end

  if nargin < 5

    optRefine=0;

    optSpeed=0;

  end
```

```
if nargin < 6
    optSpeed=0;
end
lenS = size(signal,1);
lenT = size(template,1);
hw=(lenT-1)/2; %half width of template
cw=hw+1; %points to center of template
rxy=zeros(lenS,1);

for i = 1:10:(lenS-lenT);
    %b=round(i);
    a=corrcoef(signal(i:(i+lenT-1)),template);
    rxy(i) = a(2);
end

%% Peak Detection
    indices=peakDetect(rxy,.4,.35,10);

%% check for peaks that are close together...choose one with hightest rxy
%this section entirely from the original matchTemplate.m
optClose=1;
if optClose

    bInd=true(length(indices),1); %logical array to hold which peaks to keep
    ibi=diff(indices);
    %bout = locateOutliers([],ibi,'sd',4);
    btmp = ibi<mean(ibi);
    bout =  btmp; %add '& bout' if reintroduce line 47

    d=diff(bout);

    b=find(d>0); b2=find(d<0);
    if length(b2)>length(b) %PR
        b2 = b2(1:length(b));
    elseif length(b2)<length(b)
        b = b(1:length(b2));
    end
    runLen=b2-b; %length of adjacent nan runs
    irun = find(runLen==2); %index of runs = 2
    if sum(irun)~=0 %if there are runs = 2
        bInd(b(irun)+2)=false; %remove peak
    end

    indices=indices(bInd);
    iR = indices;


end

if length(iR)>2 %if more than one beat was detected
```

```
    % Calc IBI
        ibi=zeros((length(iR)-1),2);
        ibi(:,1)=iR(1:end-1);
        rate = 1000;
        ibi(:,2)=diff(iR./rate); %calc ibi (seconds)
    else
        ibi=[];
    end

end
```

Filename: peakDetect.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Copyright (C) 2010, John T. Ramshur, jramshur@gmail.com
%
% This file is part of ECG Viewer
%
% ECG Viewer is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% ECG Viewer is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with ECG Viewer.  If not, see <http://www.gnu.org/licenses/>.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function peaks=peakDetect(s,thPrimary,thSecond,skipWin)
% peakDetect.m - detects peaks within the input signal s. A double
% threshold is applied to s : It switches the output to a
% high state when the input passes upward through the high threshold value
% (thPrimary). It then prevents switching back to low state until the input
% passes down through a lower threshold value (thSecond).
% <INPUTS>
%   s: input signal (vector)
%   thPrimary: primary threshold for detecting locations of matched templates (0-100%)
%   thSecond: secondary threshold for detecting locations of matched templates (0-100%)
%   skipWin: min number of samples to skip over once a peak is found
% <OUTPUTS>
%   indices: locations of peaks


  overPk = (s >= thPrimary); %logical array (1 or 0) of whether index is
                             %above or below thPrimary
  b=diff(overPk); % gives array of 0,1,or -1. passing above thr = 1, passing
```

```
            % below thr = -1
b(end)=0;       %make sure last sample is not a peak (cannot be a peak)


s1=find(b==1)+1; %find all samples/indexes passing above thresh


peaks=zeros(length(s1),1); %preallocate memory
x=1;
yy=zeros(length(s),1);
for i=1:length(s1)-1
   %Skip to next peak if necessary.
   %  This is useful if more than one SUCCESSIVE small peak
   %  exist that didn't drop below the second threshold.
   if x<=s1(i)
      x=s1(i);
      %loop until we drop below second threshold
      while (s(x+1)>=thSecond)
         x=x+1;
         yy(x)=1;
      end


      %find location of peak value
      tmp=find(s((s1(i):x))==max(s(s1(i):x))) + s1(i)-1;
      tmp=tmp(1); %make sure there is only one peak
      peaks(i)=tmp;
```

```
%               if i>5
%                   skipWin=floor(0.4*mean(diff(peaks(i-5:i))));
%                   x=x+skipWin;
%               end
        end
    end

    %remove any empty indices
    empty=(peaks==0); %find locations of empty peaks
    peaks(empty)=[];  %delete elements that are empty

end
```

Filename: hrvProcPart2.m

```
files1 = dir('*_ibi.mat');
            def_dir=what; def_dir=def_dir.path;
            cd(def_dir)

for m = 804:length(files1)
    input = files1(m).name;
    load(input);
    VLF = [0,0.01];
    LF = [0.04,0.15];
    HF = [0.15,0.4];
    window =128;
    noverlap = 64;

    nfft = 1024;

    fs=200;

    output = hrv(ibi,VLF,LF,HF,window,noverlap,nfft,fs);

    outname = [input(1:14) '_HRV.txt'];

    dlmwrite(outname, output, 'delimiter', '\t')

end
```

Filename: hrv.m

```
function output = hrv(ibi,VLF,LF,HF,window,noverlap,nfft,fs)


%Inputs          ibi = input file: two columns. col1 = time, col2=RR-interval duration

%                            VLF = two values specifying very low frequency range

%                            LF = two values specifying low frequency range

%                            HF = two values specifying high frequency range

%                            AR_order = model order --> not used bc no AR analysis

%                            window = #samples in window

%                            noverlap = #samples to overlap

%                            fs = cubic spline interpolation rate / resample rate

%                            nfft = # of points in the frequency axis

%                            methods = methods of calculating freqDomain. default is 3 methods.

%Outputs:   output is a structure containg all HRV. One field for each PSD method

%          Output units include:

%              peakHF,LF,VLF (Hz)
%             aHF,aLF,aVLF (ms^2)
%             pHF,pLF,pVLF (%)
%             nHF,nLF,nVLF (%)
%             PSD (ms^2/Hz)
%             F (Hz)


%check input
  if nargin<8
    error('Not enough input arguments!')
  end

  t=ibi(:,1)./1000; %time (s)
  y=ibi(:,2); %ibi (s)
  y=y.*1000; %convert ibi to ms
  %assumes ibi units are seconds
  maxF=fs/2;

  %prepare y
  y=detrend(y,'linear');
  y=y-mean(y);

  %Welch FFT
  [PSD,F]=calcWelch(t,y,window,noverlap,nfft,fs);
  output=calcAreas(F,PSD,VLF,LF,HF);
```

```
function [PSD,F]=calcWelch(t,y,window,noverlap,nfft,fs)
        %calFFT - Calculates the PSD using Welch method.
        %
        %Inputs:
        %       t = time vector associated with ibi, units = ms
        %       y = RR-intervals from ibi, detrended
        %       window = #samples in window
        %       noverlap = # samples to overlap
        %       nfft = # points on frequency axis
        %       fs = cubic spline sampling rate
        %Outputs:
        %       PSD = power spectral density estimation
        %       F = frequency axis

   %Prepare y
   fs = 200; %sampling rate of cubic spline
   t2 = t(1):1/fs:t(length(t));%time values for interp.
   t2 = t2';
   y=spline(t,y,t2); %cubic spline interpolation
   y=y-mean(y); %remove mean
   y = decimate(y,100); %reduces sampling rate to 2Hz for PSD estimation

   %Calculate PSD
```

```
    fs = 2; %specifies the new sampling rate
    [PSD,F] = pwelch(y,window,noverlap,(nfft*2)-1,fs,'onesided'); %uses a hamming window
end


function output=calcAreas(F,PSD,VLF,LF,HF,flagNorm)
%calcAreas - Calulates areas/energy under the PSD curve within the freq
%bands defined by VLF, LF, and HF. Returns areas/energies as ms^2,
%percentage, and normalized units. Also returns LF/HF ratio.
%
%Inputs:
%   PSD: PSD vector
%   F: Freq vector
%   VLF, LF, HF: array containing VLF, LF, and HF freq limits
%   flagNormalize: option to normalize PSD to max(PSD)
%Output:
%
%Usage:
%
%
%   Modified from Gary Clifford's ECG Toolbox: calc_lfhf.m


    if nargin<6
        flagNorm=false;
    end
    %normalize PSD if needed

    if flagNorm

        PSD=PSD/max(PSD);

    end


    % find the indexes corresponding to the VLF, LF, and HF bands

    iVLF= (F>=VLF(1)) & (F<=VLF(2));

    iLF = (F>=LF(1)) & (F<=LF(2));

    iHF = (F>=HF(1)) & (F<=HF(2));


    % calculate raw areas (power under curve), within the freq bands (ms^2)

    aVLF=trapz(F(iVLF),PSD(iVLF));

    aLF=trapz(F(iLF),PSD(iLF));

    aHF=trapz(F(iHF),PSD(iHF));

    aTotal=aVLF+aLF+aHF;


    %calculate LF/HF ratio

    lfhf =aLF/aHF;


    %create output structure

    output = [round(aLF) round(aHF) lfhf];


end

end
```

Appendix 5: Coordination Analysis & Parameter Generation

Filename: corlistGen.m

```
Dir1 = dir('*_BrEvent.mat');

Dir2 = dir('*_SuckEvents.mat');


corlist = zeros(length(Dir1),1);


for n = (1:length(Dir1));

   name1 = Dir1(n).name;

   name1 = name1(1:5);

   for m = 1:length(Dir2);

      name2 = Dir2(m).name(1:5);

      if name2 == name1;

         corlist(n) = 1;

      end

   end

end


a = find(corlist);

a = Dir1(a);


corList = a;

for o = 1:length(corList);
```

```
    sk = [corList(o).name(1:5) '-0000_FUL_SuckEvents.mat'];
    corList(o,2) = sk;
end
save('corList','corList')
```

Filename: corMatGen.m

```
%corMatGen

load corlist.mat;

for n = 983:length(corList);
    input = corList(n).name(1:10);
    skEv = load ([input '_FUL_SuckEvents.mat']);
    skEv = skEv.events;
    brEv = load (corList(n).name);
    brEv = brEv.events;
    full = [input '_FULL.txt'];
    fid = fopen(full);
    alldata = textscan(fid,'%f %f %f %f');
    fclose(fid);
    full = cell2mat(alldata);

    brCor = zeros(length(full),1);
    for o = 1:length(brEv);
        brCor(brEv(o,1):brEv(o,2)) = 1;
    end
    skCor = zeros(length(full),1);
    for m = 1:length(skEv);
        skCor(skEv(m,1):skEv(m,2)) = 2;
    end

    corVec = brCor + skCor;
    corMat = [skCor brCor corVec];

    outname = [input '_corMat.mat'];

    save(outname,'corMat')
end
```

Filename: corParamGen.m

```
%Coordination Parameters

dir1 = dir('*_corMat.mat');

for n=1:length(dir1)

    corMat = load(dir1(n).name);
    corMat = corMat.corMat;
    totTime = length(corMat);
    totSk = length(find(corMat(:,1)==2));
    totSkPer = totSk/totTime*100;
    totBr = length(find(corMat(:,2)==1));

    totBrPer = totBr/totTime*100;

    netSk = length(find(corMat(:,3)==2));

    netSkPer = netSk/totTime*100;

    netBr = length(find(corMat(:,3)==1));

    netBrPer = netBr/totTime*100;

    overlap = length(find(corMat(:,3)==3));

    overlapPer = overlap/totTime*100;

    feedTime = length(find(corMat(:,3)~=0));

    feedTimePer = feedTime/totTime;


    corParam = [totTime totSk totSkPer totBr totBrPer netSk netSkPer netBr netBrPer overlap
overlapPer feedTime feedTimePer];


    outname = [dir1(n).name(1:10) '_corParam.txt'];


    fid = fopen(outname,'w');

    fprintf(fid, '%f %f %f %f %f %f %f %f %f %f %f %f %f\n', corParam);

    fclose(fid);


end
```

Vita

Pallavi Ramnarain was born on January 22, 1984, in Quatrebornes, Mauritius, and is an American citizen. She graduated from the Maggie L. Walker Governor's School for Government and International Studies, Richmond, Virginia in 2002. It was there during her senior year when she was first introduced to biomedical engineering research through a mentorship program with Virginia Commonwealth University. She received her Bachelor of Science in Biomedical Engineering from Virginia Commonwealth University in 2006, followed by her Master of Science in Biomedical Engineering from Virginia Commonwealth University in 2010.