Design and Implementation of a Multi-Purpose Data Acquisition and Experiment Control System for MRI and Neuroscientific Lab Environments

Mirza Abdel Jabar Baig

Degree of
Department of Biomedical Engineering

McGill University
Montreal, Quebec, Canada
April 2011

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering

Acknowledgements

I would like to thank my supervisor Dr. Amir Shmuel for providing me great insight and guiding me throughout my project implementation and thesis preparation.

I would also like to thank all the other members of the lab for their contribution. They helped me whenever I had any questions.

This project was supported by Montreal Neurological Institute (MNI), Canada Industry Center of Excellence in Commercialization and Research (CECR) grant awarded to Amir Shmuel.

Abstract

Currently existing patient monitoring and data acquisition systems are proprietary, and they do not support the use of custom devices that might be required for some experiments. The design and implementation of a new patient monitoring and data-acquisition system is presented here, which supports the use of any new device following a simple and short increment of programming by the user. We demonstrate the usefulness of the system by describing the design and implantation of several devices already integrated into the system. In addition, we describe hardware and software design of a body movement detection system, based on our developed data acquisition system. This system can be used to train human subjects or alert animals to stay still, before they undergo an MRI scan. The design and implementation of the movement detection system is based on accelerometers that are integrated into our developed data-acquisition and monitoring system. Finally, we discuss the flexibility of the system and devices that can be integrated to it in the future.

Résumé

Actuellement existants de surveillance des patients et des systèmes d'acquisition de données sont la propriété, et ils ne supportent pas l'utilisation de dispositifs sur mesure qui pourrait être nécessaire pour certaines expériences. La conception et la mise en œuvre d'un nouveau patient et la surveillance du système d'acquisition de données est présentée ici, qui prend en charge l'utilisation de tout nouvel appareil suite à une augmentation de courte et simple de la programmation par l'utilisateur. Nous démontrons l'utilité du système en décrivant la conception et l'implantation de plusieurs dispositifs déjà intégré dans le système. En outre, nous décrivons la conception matérielle et logicielle d'un système de détection de mouvements du corps, sur la base de notre système d'acquisition de données mis au point. Ce système peut être utilisé pour former des sujets humains ou des animaux d'alerte de rester encore, avant de subir un examen pare IRM. La conception et la mise en œuvre du système de détection de mouvement est basée sur les accéléromètres qui sont intégrés dans notre point d'acquisition de données et système de surveillance. Enfin, nous discutons de la flexibilité du système et les périphériques qui peuvent être intégrées à l'avenir.

Table of Contents

List of Figures

Chapter 1 Introduction

1.1 Motivation and Objective

1.1.1 Motivation

This thesis describes a system to monitor and record an animal subject's vital signs, non-physiological data. In addition, the system controls experiments by triggering other systems including stimulation systems.

Many research labs that work with animals use some sort of patient monitor to record and/or preview physiological data in real time during experiments. This allows the user to monitor the subject's vital signs and take action when required. Veterinarian patient monitors have been used in surgery, anaesthesia, and other research settings to monitor physiological parameters [1-4]. They display in real-time and record trends in heart rate, SpO2, CO_2 levels, and respiration. The number of parameters that they monitor varies depending on how advanced and pricy the system is. These systems are geared towards clinical rather than scientific use. Many of these systems do not record waveforms such as ECG and respiration. In our lab, ECG and respiration waveforms must be recorded at high sampling rate and for long durations to perform artefact removal on the imaging signals. Patient monitors also do not support other important features needed in our lab such as experiment control through stimulation, triggering of external systems, non-physiological data acquisition, data-acquisition at high-sampling rate, and user-defined custom analysis in real time.

In addition to monitoring vital signs of the subjects, work in research labs also requires the system to output commands and data to external devices as part of the experiment's protocol. Communication with external devices for stimulation and other purposes, in addition to monitoring physiological parameters, is supported by data acquisition (DAQ) systems geared towards research and education, such as PowerLab (ADinstruments), Spike2 (CED), and MP series systems (BIOPAC Systems). Spike2 is similar to PowerLab when it comes to data acquisition and analysis. It does not support interface to its DAQ hardware (Power 1401) without using CED's proprietary software [5]. It is also more expensive than PowerLab so it will not be discussed in detail. Each of the mentioned systems comes with its

own software (LabChart from ADinstruments and AcqKnowledge from BIOPAC) for central data acquisition and experiment control. Both PowerLab [6-9] and MP series systems [10-14] have been used extensively in the research field. The drawback of these systems is that their software cannot be modified to fit the needs of a specific lab. In the case of our lab, for example, we needed to stimulate subjects using a custom-built airpuff stimulator at specific times during an experiment. This cannot be accomplished with the existing DAQ systems or by patient monitoring systems because they do not support communication with this custom made device. Another type of stimulus that is needed in our lab is automated rotatory somatosensory stimulus using a soft brush. The device that provides this stimulation needs to receive commands through a serial port of a personal computer (PC). Existing systems cannot support this specific stimulation device unless a request is made to the respective application engineers to change their software program to include code to communicate with this device.
Although both mentioned existing systems support the use of external proprietary modules for recording data such as CO_2, pulse oximetry, blood pressure, respiration, etc, only BIOPAC has a wide selection of sensors available to be used in the Magnetic Resonance Imaging (MRI) scanner environment [15].
Due to the lack of support for third party devices by both DAQ systems, a new system is needed that can be easily expandable for use with other third party sensors and devices. This can also lower the cost of the overall system, because there are more choices of sensors and no funds are spent on expensive BIOPAC data acquisition system ($5000 for MP150 according to a quotation). If the sensors used for vital signs monitoring are completely MRI compatible, then the new system can also be used in the MRI environment as long as the data acquisition hardware is outside the magnet room.
Another major functionality which is not included in existing systems is the support for automated anaesthesia control. This is to make sure the subject stays in a desired physiological condition during an experiment. Although the proposed system currently does not include this functionality, it can be easily added by implementing an additional module.

1.1.2 Objective

The main goal of my Master's project was to design and implement a multi-purpose DAQ system (LabState) that can be used as both veterinarian medical monitor and control system in a research lab environment. It should be able to record physiological data for monitoring subject's vital signs, such as ECG, EEG, respiration, heart rate, SpO2, etc, control other systems in the lab by sending out triggers, and stimulate the subject using different stimulation devices. It should support the addition of custom modules and functions that could be programmed by anyone who is familiar with LabVIEW. It could also be used in the MRI environment and/or with humans, granted that the sensors used are compatible for their respective use in terms of safety and reliability. In either case, any non-sensor component of the system would have to stay outside the MRI room.
The objectives included programming a number of modules and functions, and design and implementation of the hardware for the different modules. One of the devices I created was a body movement detection system that was converted into a stand-alone system which did not require the use of LabState, but has its own software application called StillState. We needed a body movement detection system because of motion artifacts arising in MRI scans due to the subject's body (including head) motion [16]. This system can be used during alert animal training to detect when the animal moves beyond tolerable range such that the trial can be stopped during the training. It can also be used to train children to not move during MRI scans (however this requires further safety testing and official approval). There is already a commercially available system that performs this kind of motion detection, but it's expensive and has many features that are not needed, such as recording both linear and rotational displacements in all three axes in real-time [17].

1.2 Thesis Outline

Chapter 2 discusses existing veterinary patient monitors and DAQ systems.

Chapter 3 contains detailed requirements and design criteria that both the DAQ system (LabState) and movement detection system (StillState) need to meet. It also discusses constraints on the systems.
Chapter 4 describes the hardware used for LabState and StillState.
Chapter 5 describes the design and architecture of LabState and some of its central parts.
Chapter 6 describes the design and function of some of the major modules that are used in LabState. For some modules, it also explains how the hardware was designed (in cases it was custom built).
Chapter 7 describes the design and implementation of the functions that are used in LabState, especially Get Rate.
Chapter 8 describes how the software and hardware for the movement detector system were selected, designed, and implemented.
Chapter 9 describes how LabState and StillState were tested to make sure their functionality was correct.
Chapter 10 provides a summary of the thesis and discusses additional work that can be done in the future to enhance the DAQ system and body movement detection system.

Chapter 2 Existing Systems

2.1 Veterinary Patient Monitors

Patient monitors (also called medical monitors) measure a patient's physiological signs and display them to the user or examiner. Normally, an animal's physiological signs are monitored using a veterinary patient monitor. Separate monitors exist for humans. In our lab, the focus is on small and large anaesthetized animals. Therefore a veterinary patient monitor is required for recording and displaying the subject's vital signs.

Currently, there are many different types of veterinary patient monitors available in the market. Most of them are handheld and only acquire a few parameters, such as Novametrix Tidal Wave 710Sp Monitor [18]. This device can monitor CO_2, respiration rate, SpO_2, and heart rate; and it runs on batteries. However it does not have any ability to connect to a computer and save the displayed data. Other systems are more advanced, can acquire data from more sensors, and display more waveforms on the screen [19-21]. They also allow printing of some of the data directly onto a printing paper. However they still do not provide any software on a computer which lets the user connect the system to the computer and store data at high sampling rate in order to allow offline processing. For example, Advisor® Vital Sign Monitor by SurgiVet only lets the user record trends of data such as heart rate, temperature, and SpO2 at 30 second interval onto a flash card. ECG waveform can be recorded for only up to 30 seconds [21]. The recorded data length is not long enough for our experiments.

A different patient monitoring system called PC VetGuard+ allows vital signs monitoring directly on a computer. The computer software communicates with the sensors wirelessly using Bluetooth. It also allows the user to save the data for later analysis with a proprietary software [22]. It does not allow conversion of data into MATLAB for additional analysis. The sampling rate of the recorded data is also not documented, but it is of no use since the data cannot be loaded into MATLAB. Another drawback of this system is that it does not support input/output of additional analog and digital channels for synchronization and

experiment control. Also, it does not support communication with third party or custom built devices such as the airpuff stimulator used in the lab.

2.2 Research Oriented DAQ Systems

2.2.1 PowerLab system

PowerLab data acquisition systems comprise of hardware and software. They are research oriented data acquisition systems. They can be used to record physiological signals from humans and animals. The hardware consists of a PowerLab recording unit and various other devices that can be connected to it. The main software is called LabChart which enables the user to record, view, and analyze data in real-time on a computer [23].

The PowerLab recording unit is available in different models, and the number of inputs and outputs varies with the model. The most advanced model is ML880/P which consists of 16 analog inputs and 2 analog/stimulation outputs. There are also 4 “pod” connectors to which specialized instruments and transducers can be connected. Each analog input comes with its own built-in amplifier and filtering unit so signals of low voltage (μv) can be recorded using this device. Maximum sampling rate is 20 kS/s per channel if 9-16 inputs are used. There is also a serial port connector available for future use to connect devices that communicate using RS-485 serial communication protocol [24].

There is no known way of recording from a PowerLab unit using a well documented application program interface (API). Therefore the only software that PowerLab can be used with is LabChart or any other proprietary software that ADInstruments sells. In addition, there is currently no known method to import PowerLab data in real-time for online analysis by an external user-written software program [25].

2.2.2 BIOPAC MP series

BIOPAC produces two major data acquisition units: MP100 and MP150. MP100 supports aggregate sampling rate of 75 kS/s while MP150 supports an aggregate sampling rate of 400 kS/s. MP150 also supports 16 analog inputs and 2 analog

outputs for stimulation [26]. AcqKnowledge is a software program that can not only acquire, display, and store data from MP100 or MP150, but can also analyze it in real-time. It can perform online filtering of data, calculate heart rate from ECG, output stimulation pulses, etc [27]. There is no programming knowledge required to use this program. Users cannot implement their own code in it to communicate with external devices. There is a limited scripting language option available (purchased separately) but it is not sufficient to enable connectivity with external devices in synchrony with the main data acquisition from the MP device [28].

Unlike PowerLab recording units, MP150 data acquisition hardware can be used with custom user-written applications. This is possible by using BIOPAC Hardware API (BHAPI) sold separately by BIOPAC. BHAPI provides a set of functions through a 32-bit Windows DLL that can be used in any major programming language such as C/C++, Visual Basic, and LabVIEW to acquire data from a MP150 device [29]. This essentially gives a developer the ability to write a single program that can not only acquire data from BIOPAC data acquisition hardware but also other devices that are not supported by BIOPAC in synchrony.

Most BIOPAC sensors can also be used without the need for a MP device. This requires that the user purchase a special power supply (IPS100C) which connects to multiple amplifiers and outputs their analog signals through the phone jacks on its front panel. Signals from these phone jacks can be acquired using a third party analog data acquisition device. **Figure 2-1** shows how IPS100C can be connected to different amplifiers to get data from the sensors. If none of the sensors that are in contact with the subject are electric (e.g. air pressure based) then there is no need for electric isolation from the DAQ board. Hence there is no need for OUTISO and HLT100C modules.

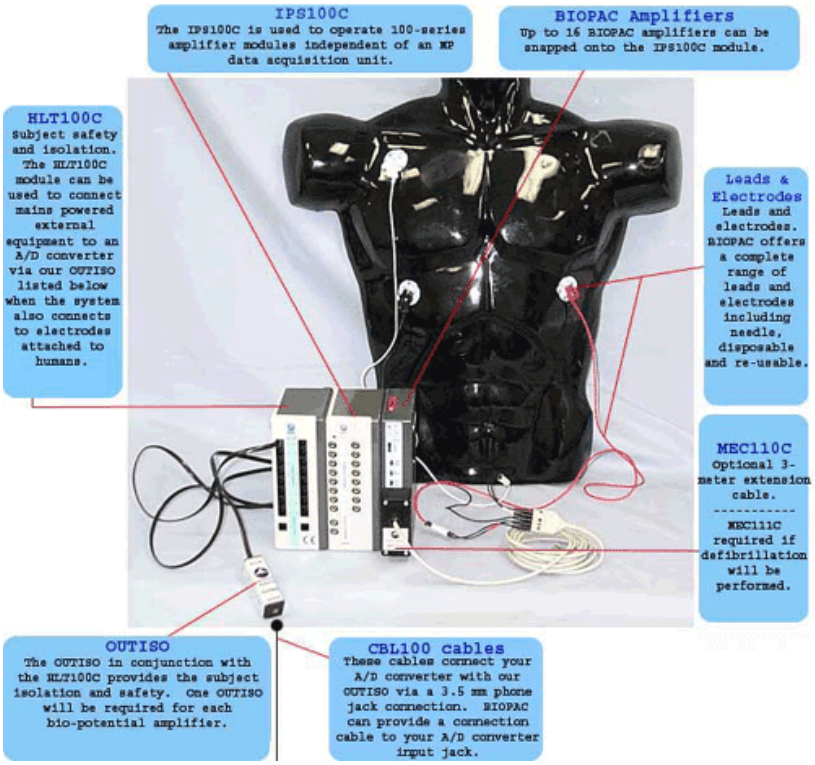

Figure 2-1: Setup showing usage of a BIOPAC sensor without the need for MP series DAQ device [30].

Since most BIOPAC sensors can be used with third party DAQ cards, and BIOPAC software does not cover the needs of our application, there is no need to use expensive MP series data acquisition system from BIOPAC. A cheaper general purpose DAQ card can be purchased from other manufacturers such as National Instruments (NI) or Measurement Computing (MC). If the portability of USB or ethernet cable connection to the computer is not needed then a PCI DAQ card can be used to reduce the price even further. This is the option we decided to pursue for the lab setups because portability was not a priority.

Chapter 3 System Requirements and Constraints

3.1 Requirements of LabState

LabState is a software application that has to be capable to acquiring, displaying, and recording data from a data acquisition card along with external modules. A list of the main requirements for the system is given below:

- Operating system: Windows XP or higher
- Analog inputs: 16 channels or more
- Digital inputs: 8 channels or more
- Sampling rate/channel: 25 kS/s for 4 channels. 5 kS/s or higher for other channels.
    - Maximum 50 μs jitter for digital edge detection
- Correlated digital inputs sampling
- Variable sampling rate
- Ability to scale analog input data linearly
- Add/edit variables and change their values during an experiment
- Support addition of time-stamped comments
- Multiple alarms and voice alerts
- Communication with user-written modules and functions
- Be able to display and record any data acquired from the modules
- The following modules must be created: Nonin II Pulse oximeter, Capnostat CO_2, Airpuff Stimulator, Digital (electric) and TMS Stimulator, Rotatory stimulation
- Function to calculate the fundamental frequency in a signal should be implemented (low priority)

Some of the requirements are explained in details below. The requirement of 16 analog and 8 digital channels was set after going through items whose signals could potentially be recorded in our lab. Requirements of digital inputs were decided upon similarly. These inputs are required to record TTL signals. Although TTL signals can be recorded using analog inputs, this is not optimal, since they can cause a great deal of crosstalk with other signals being recorded. When running an experiment, the user needs to note down comments about the current state of the experiment. Usually the users have a notebook with them to write those comments along with the time. One of the requirements for LabState is to allow the user to write text comments directly to the data file and associate them with a timed-event, rather than in a notepad or somewhere else. In addition

the user should be able to write variables to the data file and update their values throughout the course of the experiment.

3.1.1 Sampling rate

The maximum data acquisition sampling rate required for the setups in the lab is 25 kS/s. One of reasons for the high sampling rate is that we wanted to be able to record MRI gradient signals. These signals will be recorded in the MRI setup. Gradient signals are used to denoise neurophysiological signals recorded in the MRI. Therefore they must be acquired at the same sampling rate routinely used when recording neurophysiological signals, including action potentials (20-25 kS/s). Second reason for supporting recording at 25 kS/s is that both setups require the recording of stimulation pulses that can have very short pulse duration (0.1 ms). All TTL signals are acquired through the digital inputs of the DAQ card rather than analog inputs. Since digital inputs are used, Nyquist theorem is not considered when calculating minimum sampling rate. The minimum sampling rate should enable the detection of all the pulses in the signal and it should also enable detecting rising/falling edges with expected accuracy. For a 0.1 ms long pulse duration, the minimum sampling rate for detecting all the pulses is 10kS/s=1/0.1ms. Higher sampling rate will increase the accuracy of rising/falling edge detection. At 10 kS/s sampling rate, the jitter in edge detection is 100 us. If the TTL is sampled at 25 kS/s then the jitter is 40 us which is accurate enough for our experiments. Another reason for recording high sampling rate is that optical imaging camera frame toggles are acquired from two different optical imaging systems to synchronize their imaging data offline. One of the cameras produces a short (100 us) pulse for each frame acquired. According to the reasoning mentioned for sampling stimulation pulses, this signal must also be sampled at 10 kS/s or higher to make sure all the pulses are detected. 25 kS/s is used to reduce edge detection jitter even further. Therefore LabState needs to be designed in such a way that it supports up to 25 kS/s of sampling rate for at least four channels without making the system unstable.

Variable sampling rates must be supported for different channels because if a device is outputting data at a slow rate then sampling at high rate is a waste of hard disk space and processing power. Having support for variable samplings rates lets the user record channels from devices with very high and very low throughput simultaneously. For example, both electrocardiogram (ECG) and electroencephalogram (EEG) signals are recorded by using electrodes and then filtered and amplified by an AC amplifier. It has been recommended that minimum sampling rate for ECG should be 1 kS/s [31]. EEG signal also contains frequencies of up to several hundred hertz [32]. Therefore it is a good practice to sample EEG at a minimum of 1 kS/s sampling rate. However, we sample both ECG and EEG at a higher rate (5 kS/s) to reduce aliasing further since there is no programmable anti-aliasing filter on the DAQ card (as explained in section 4.1). This is quite lower than minimum of 25 kS/s used for some stimulation pulses, so recording EEG and ECG at 25 kS/s is a waste of computing power and hard disk space. This is why variable sampling rates are needed so some signals as be acquired at lower sampling rates.

Finally, the acquisition of digital signals must be correlated with the analog signals so they all should be synchronized. If digital acquisition starts slightly after or before analog acquisition then the recorded analog data has to be aligned with the digital data offline.

3.1.2 Scaling

The scaling feature is required to record signals properly using any of the DAQ cards. Occasionally, an external device has to send high numbers using analog output but it cannot generate voltage levels that high. It sends a scaled down number that needs to be scaled up to enable the recording of the real value. For example, the temperature value recorded through an analog channel needs to be presented in degrees with two digits (eg: 24.5). The temperature sensor cannot produce this kind of voltage. Therefore, the temperature sensor sends a number which is 10 times smaller than the actual number (0.245 in this case). Scaling this number can bring it back to 24.5 and record and display it on the screen.

3.1.3 Alarms and voice alerts

One important feature of any patient monitoring system is to sound an alarm whenever one the physiological conditions of the subject is out of normal range. For example if the heart rate is too high or too low, or the respiration is too slow or too fast then an alarm goes off. This warns the user (or experimenter) to pay attention to the subject in case his/her eyes are not focused on the monitor. This purposed system should support the addition of ranges for different channels, as entered by the user, outside of which an alarm is sounded. The system should also support automated alerts of selected channel values using computerized voice. For example, the system can say exactly what the heart rate is every minute to keep the user updated on the current heart rate when he/she is busy with the surgery and not paying continuous attention to the monitor.

3.1.4 Modules

The system should support the addition and use of user-written modules to communicate with any additional external devices. This can include devices that output data to the system or receive data from the system. The communication with the devices can take place using any available method such as TCP/IP, USB, GPIB, Serial (RS232), etc. Data received from the devices should be sampled at up to 1 kHz. Many medical instruments output data at a much lower rate so 1 kHz is a safe choice. Anything higher would require more processing power and can slow down the overall system.

3.1.5 Functions

Addition and use of user-written functions should also be supported by the system. Data from one or more of the analog and digital channels is passed to each function which may return a new set of data to be recorded and/or displayed.

3.2 LabState Design Criteria

A decision tree showing different system and DAQ hardware usage options that were thought of is shown in **Figure 3-1**. As described in chapter 2, the veterinary

patient monitors were not used because most of them do not even allow connection to a PC for data acquisition and have low sampling rate. Those that do allow connection to PC only acquire data from a select few sensors and there is no way to synchronize their acquisition with other device. Also, there is no way to stimulate subject or control external device using the software that comes with those patient monitors. The scientific DAQ systems such as PowerLab and MP Series systems were again not used because their software does not allow communication with external devices. We decided to pursue the option of creating a new system in LabVIEW. C++ was not used because it required much longer development time. MATLAB requires less development time compared to C++ because it contains instrumentation toolbox to perform data acquisition more easily and it also contains build-in functions for manipulation and analysis of data. However, this development time is longer than that required for LabVIEW, because parallel tasks cannot be executed as easily. The graphical user interface (GUI) in MATLAB is also limited while LabVIEW has the ability to let the developer create complex GUI interfaces without devoting to it much development effort. There were two main manufacturers of DAQ cards: MC and NI. The third DAQ card was an MP series card by BIOPAC. It could be programmed in LabVIEW just like other DAQ cards because BIOPAC provided API for it. However it was not used because it was relatively expensive. MCC's API interface in LabVIEW was not as good as NI's for analog and digital acquisition. Much more coding was needed to acquire the data. For example MCC's API only supports acquisition of data from a range of channels so if only a select few channels are needed then the resulting output has to be processed to remove the unwanted channels. Also, MCC did not provide any BNC panels which was inconvenient and would have required us to create them ourselves. So for analog and digital acquisition no card from MCC was used. However, as mentioned below, a cheap counter card by MCC was used because the counter API interface in LabVIEW was simple and it provided rising-edge counting which no NI card in the similar price range provided. NI provided PCI and USB

type of cards. Both of them were used for main analog and digital acquisition depending on the portability requirement of the setup.

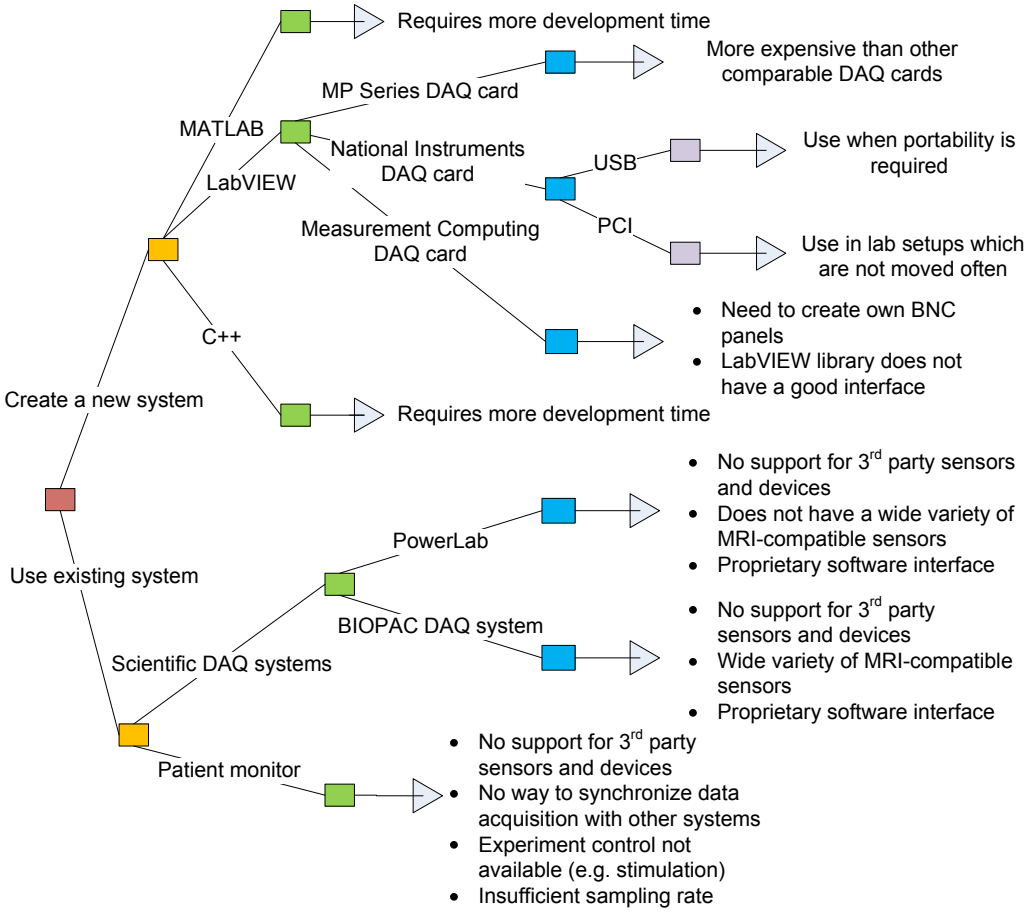

Figure 3-1: Decision tree for choosing the DAQ card.

Once the DAQ hardware is chosen, each sensor is selected such that it can be controlled by LabState through its own module. The communication protocol used with most of the sensors is RS232 while other sensors can be connected directly to one or more analog inputs.

3.3 LabState Constraints

The budget for the hardware in each setup was set at $7000. This was lower than existing patient monitors or DAQ systems mentioned earlier.

Each manufacturer of DAQ cards implements the software application program interface (API) differently. LabState is written to acquire analog and digital data

from any NI DAQ card which can be accessed using DAQmx API. Therefore, cards from any other manufactures cannot be used. The software is written and compiled on a Windows machine in LabVIEW so it can only run on a Windows based PC. Although it can ported to MAC OS (since LabVIEW for MAC is available) after some modifications, it is currently not supported.
Since one of the requirements of the design is correlated digital input, only the DAQ cards that supported correlated digital input from NI can be used. Otherwise no data can be acquired from the digital inputs.
The maximum number of plots and display labels supported for real-time display of data is 16 each. This is because the plots and display labels cannot be dynamically created so each one is manually added during the user interface design. However if more plots or display labels need to be added, the program can be modified easily and recompiled. The more plots and display labels there are (whether they are used or not), the more memory the program requires.

3.4 StillState Requirements, Design Criteria, and Constrains

StillState is a motion detection system that consists of a software application along with a movement detector device to acquire data related to subject's motion. Two main parameters of the selected movement detector device are the number of axes and sensitivity. It should provide acceleration data from two or more axes (three would be ideal), such that almost any kind of movement by the subject could be detected. It should also have very high sensitivity so a slight movement can trigger the alarm.
The system should display and record data from all the axes supported by the device. It should be able to record this data at 2 kS/s or higher sampling rate because millisecond resolution is needed to detect fast motion. It should let the user calibrate the device when there is no motion, and support the use of alarms to warn of motion in any of the axes. Finally, it should also support output of a trigger which is sent to one or more external devices whenever an excessive motion occurs. The trigger is a digital TTL signal.
Constraints: The budget for the entire setup was set to $250.

The stand-alone software program can acquire, display, and record data from 2 axes of the accelerometer. Even if the DAQ card used supports additional analog inputs, the program does not need to have the functionality to allow the user to use those additional inputs for recording extra data.

Chapter 4 Hardware Usage

4.1 LabState Hardware

As a core application, LabState acquires analog and/or digital data from a DAQ card. In our lab, we have two DAQ systems. A third one is located at the MRI control room. Each setup uses a different DAQ card. They use PCI 6259, PCI 6221, and USB 6221, respectively. A list of main specifications for all the cards is shown in **Table 4-1**. This data is taken from the documentation of each card. The links to the documentations is provided in **Table 4-2**. The only major differences between PCI 6259 and 6221 is that the former supports twice the amount of analog inputs (32 total) and has a aggregated sampling rate of 1 MS/s as opposed to 250 kS/s for the latter. So far in each setup only up to 8 differential analog channels with a sampling rate of no more than 25 kS/s were needed, so PCI 6221 works fine along with PCI 6259. In case additional channels are needed, the setup with PCI 6259 can be used or some of the analog channels can be sampled in single-ended mode if possible.

Table 4-1: Specification of DAQ cards

Parameter	**PCI/USB 6221**	**PCI 6259**
Analog Inputs	8 differential	16 differential
Analog Sampling Rate	250 kS/s	1 MS/s
Sampling Modes	differential, single-ended	differential, single-ended
Analog Input Resolution	16 bits	16 bits
AC Input Range	±10 volts	±10 volts
Digital I/O	24	48
Low-pass Filter Cutoff	700 kHz	1.7 MHz

Another important specification to notice is that none of the cards have any adjustable anti-aliasing filter. There is a fixed low-pass filter with cut-off at 700 kHz for PCI/USB 6221 and 1.7 MHz for PCI 6259. No matter what the acquisition sampling rate is, there will always be some aliasing. However, if the

input signal has already been filtered by either the sensor's hardware or an external amplifier then this aliasing is automatically reduced as long as the sampling rate is above the Nyquist rate [33]. For recording ECG, EEG, and respiration pad, a band-pass filter with high frequency of 300 Hz is used in the lab. Therefore, sampling at 600 Hz will ideally eliminate aliasing. However, analog filters do not have very sharp transition band so sampling at approximately 6 times the highest frequency is generally recommended [34]. In our lab, we sample most of the signals at 5 kS/s which is greater than 6 fold the highest frequency for any analog input so aliasing is significantly reduced.

Duplicate hardware used in setup I and II is shown in **Table 4-2**. The hardware used in setup III in the MRI environment is listed in **Table 4-3**. The first two setups are almost identical except for the type of DAQ card they use and that one of setups uses a Capnostat 5 CO_2 sensor for monitoring large animals. The other setup applies a pressure transducer with respiration pad from BIOPAC to monitor respiration of small animals.

Table 4-2: List of hardware and sensors used for the two lab setups

Hardware	**Price ($)**	**Documentation**
NI PCI-6259/21	1497/740	http://www.ni.com/pdf/manuals/371022k.pdf
NI BNC-2090A	485	http://www.ni.com/pdf/manuals/372101a.pdf
USB-1024LS	100	http://www.mccdaq.com/PDFs/Manuals/USB-1024LS.pdf
USB-8COM-120V	168	http://www.byterunner.com/byterunner/product_name=USB-8COM-120V+%28120+volt,+60+hz+version%29
Nonin OEM III Vet Pulse Oximeter	300	http://www.nonin.com/documents/OEM%20III%20Module%20Specifications.pdf
Respironics Capnostat 5 CO_2 Monitor (setup I only)	1500	http://oem.respironics.com/Downloads/01-1742-4101194-Capnostat.pdf
TSD110-MRI pressure transducer (setup II only)	2075	https://www.biopac.com/respiration-pad-transducer

Model 1700 Differential AC Amplifier	1950	http://www.a-msystems.com/s-129-model-1700-differential-ac-amplifier.aspx

Table 4-3: List of hardware and sensors used for setup III (MRI based)

Hardware	Price ($)	Documentation
NI USB-6221 BNC	1670	http://www.ni.com/pdf/manuals/371022k.pdf
Nonin 8600FO Pulse Oximeter	3000	http://www.nonin.cz/ceniky/cenik-7-1187777759.pdf
TSD110-MRI pressure transducer	2075	https://www.biopac.com/respiration-pad-transducer

Each of these cards goes into a PCI slot of a desktop PC. Once they are placed inside the computer, a BNC panel is connected to a connector on each card. BNC 2090A panel is used, as shown in **Figure 4-1**. This provides the capability to connect BNC cables to analog inputs/outputs and counters because most of the devices in the lab have BNC connectors.

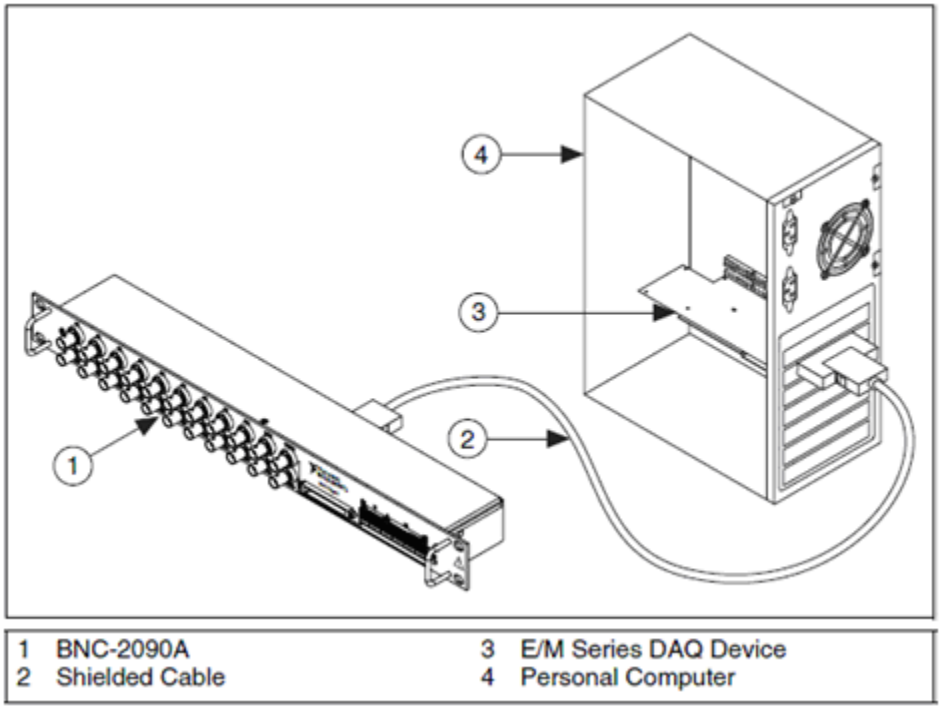

Figure 4-1: Diagram of connections of BNC-2090A to a PCI DAQ card [35].

USB 6221 is used in the setup III because it needs to be portable. The card and BNC panel are both in one box which connects the computer using a USB cable. There is no need for a PCI slot or another BNC panel. If one of the above PCI cards is used instead, then a desktop computer is required. USB 6221 has the same functionality as PCI 6221 except that it's more portable and can be used with a notebook by simply connecting it using a USB cable, as **Figure 4-2** shows.

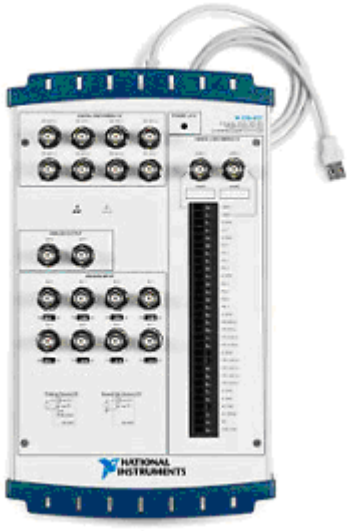

Figure 4-2: An image of BNC 6221 [36].

All three mentioned cards have two counters/timers each. A counter/timer is a device that can perform two different functions: the *counter* part refers to the device's ability to count the number of times an event or process has occurred [37]. This is used in cases such as when a trigger signal needs to be counted until it has reached a certain number at which time, for example, a stimulus is sent. The *timer* part refers to the device ability to generate very precise pulses of certain frequency and duty cycle [38]. This can be used to generate a digital stimulus which is used, for example, to blink an LED or as an input to an electric stimulator.

Model 1700 differential AC amplifier (A-M Systems) was used to amplify ECG signal from a pair of differential electrodes.

4.1.1 USB-1024LS usage

In our lab, two counters were sufficient for most experiments. One of them was used to count triggers and the other one was used to output stimulus based on the counted triggers. However, on certain occasions an additional digital pulse to create Transcranial Magnetic Stimulation (TMS) along with electric stimulation was needed to be outputted. Therefore, an additional data acquisition card was used to add one more counter to the system. This card was USB-1024LS by MC. It has one 32-bit counter which only counts pulses from the input signal, but not generate any pulses. So it couldn't be used to generate stimulation or TMS pulses. It was used to count triggers and the two counters on the NI DAQ cards were used to send electric stimulation and TMS pulses. Although it has been mentioned before that library interface for MC cards in LabVIEW is not well written, this only applies to the libraries that are used to acquire analog and digital data. To just count pulses from the input, the interface is very simple and easy to use. Similarly priced DAQ cards (e.g. USB6008) from NI only supported pulse counting on the falling edge but we needed to count on the rising edge of the triggers for synchronizing with other systems in the lab. For example, the optical imaging system produced a short pulse every time the camera took a frame. These pulses (frame toggles) were used as triggers. Stimulation of the subject was needed to be synchronized with one of these pulses which would not have been possible if the card only counted on the falling edge of the signal.

4.1.2 Sensors

Both OEM III veterinary pulse oximeters and capnostat 5 CO_2 sensor were OEM devices purchased directly from the manufacturers. Each pulse oximeter was enclosed in a plastic box for protection. The capnostat already came enclosed in a proper box. Both sensors were connected to the PC using a RS232 serial cable. Because modern computers do not have RS232 ports, a USB to serial converter (USB-8COM-120V) was used to create RS232 ports on the PC.
In setup I, capnostat 5 is used to get both the end-tidal CO_2 and respiration waveform. In setup II where small animals are used, a respiration pad connected to a pressure transducer is used to get the respiration waveform. The pressure

transducer TSD110-MRI is connected to the general purpose transducer amplifier DA100C which is then connected to the isolated power supply module IPS100C for giving power to the transducer and letting the user connect its output to one of the analog inputs on the DAQ card.

In setup III, Nonin 8600FO pulse oximeter was used to monitor the subject's SpO_2, heart rate, and plethysmogram. All the outputs were analog so they were directly connected to the DAQ card using BNC cables. For hygienic purposes, two different sensor cables for this pulse oximeter were used. One was used with human subjects and the other one with animals only.

For respiration monitoring of animal subjects, a tube-based respiration pad was used. Data was recorded from it through the amplifier provided by BIOPAC which was kept outside the magnet room. Analog output of the amplifier was connected to the DAQ card. For human subjects, same pressure transducer was used except the respiration pad was replaced with a respiration belt (still tube based).

4.2 StillState Hardware

StillState records acceleration in x and y axis and uses this data to decide whether motion has occurred. There are many different accelerometers available on the market. Most of them are digital and a few are analog. Since movement detection in milliseconds timeframe was needed, an analog accelerometer was used. Data from digital accelerometers is not outputted at a high enough rate for our application. Keeping in the mind the requirement of acceleration detection in 2 or more axes and high sensitivity, the best one found was ADXL203 (Analog Devices, $35, http://www.analog.com/static/imported-files/data_sheets/ADXL103_203.pdf). It detects acceleration along two axes (x and y) and it has a sensitivity of 1000 mV/g. Any accelerometer found with three axes had a much lower sensitivity. A list of all the components used is given in table 3. The design of the system using this accelerometer is described below. Because the accelerometer outputs analog data, it needs to be recorded by a DAQ card. USB 6008 (NI, $180, http://www.ni.com/pdf/manuals/371728b.pdf) was

used because it is small, portable, cheap, and has to ability to record at least two analog channels at a sufficient sampling rate (up to 5 kS/s). It is also compatible with LabState, which is useful in case additional analog data needs to be recorded using it.

4.3 Subject Safety

4.3.1 LabState hardware

During any experiment, the safety of the subject is crucial. Any damage to the subject is possible only through the sensors and other device that are directly attached to it. In our lab setup, all the sensors that are used (pulse oximeter, CO_2 sensor, pressure sensor) are approved for use with animal subjects and they have been tested thoroughly by the manufactures. LabState merely acquires data from the control units of these sensors and it has no direct control of any sensor. Therefore it is safe to use LabState to monitor an animal subject's vital signs as long as it is made sure that all the sensors being used are approved for that purpose and they are safely used by the experimenter.

For connecting the individual sensors, the experimenter has to follow the guidelines included in the sensor's operating manual. Some of the safety precautions are mentioned for the used devices below (as taken from their respective manuals).

4.3.1.1 Nonin pulse oximeter

OEM III Vet Pulse Oximeter control unit (not the sensor itself) is not to be used in the MRI environment. It also does not meet defibrillation-proof requirements so it cannot be used while defibrillating the subject. During experiments in the lab, both of these safety guidelines are met. Additionally, the experimenter makes sure that the blood flow is not hindered at the sight of sensor attachment so the reading is not inaccurate.

Nonin 8600FO control unit can be placed inside the MRI room but it should be as far away as possible from the bore. The sensor however can be placed inside the bore to acquire the pulse oximetry data.

4.3.1.2 Capnostat 5 CO_2 monitor

This monitoring device consists of a control unit and a tube-based sensor which connects to the subject side and can be used to monitor levels of carbon dioxide. Due the fact that it has no electrical connection with acquisition device, it is electrically isolated from it. However safety precautions must be taken to make sure the data read from the sensor is correct. For example, the tube of the sensor must not be entangled so it provides an open airway for the flow for CO_2. The sensor must also be removed from the circuit whenever aerosolized medication is delivered. Increased viscosity of the fluid may contaminate the sensor window and cause the sensor to fail. All these guidelines are followed while performing experiments in the lab.

4.3.1.3 Pressure transducer

Similar to the capnostat 5 monitor, TSD110-MRI pressure transducer also does not require any electrical connections to the subject because it consists of a tube-based sensor. Therefore, it does not cause electric safety hazards to the subject due to the acquisition device.

4.3.1.4 AC amplifier

To record EEG and ECG, subdermal needle electrodes (745 12- Series, King Medical, [39]) are directly connected to the subject and their signals are acquired through a Model 1700 differential amplifier. This amplifier is not electrically isolated from the inputs (electrodes) so if there is a malfunction then it is possible that a high voltage is produced at the inputs which can cause large amount of current be sent to the electrodes. The probability of this malfunction, however, is very small. A capacitative-coupler or any other isolation device can be used to isolate the subject from the amplifier if additional safety is desired. This was not considered as part of the requirements for the thesis.

4.3.1.5 Stimulation

For electric and TMS stimulation, LabState sends TTL signals to the respective stimulation devices for turning the stimulation on or off. A365R Isolator (WPI, [40]) is used for electric stimulation and Rapid (Magstim, [41]) is used for TMS stimulation. The devices used for stimulation are approved by the manufactures to be used in animal research and hence are safe to use as long as their usage

protocol is followed in the stimulation paradigm (for example, the current sent using the stimulation device is within the safe limits). For TMS stimulation, a strict protocol must also be followed to make sure the subject is not harmed. This involves control of rate and power of the stimulation. These parameters are controlled by the experimenter, according to the published guidelines of safe usage of TMS.

4.3.2 StillState hardware

For performing movement detection, the circuit board (including the accelerometer) on the StillState sensor is electrically powered with 5 volts. There is no electric isolation of the subject if the sensor is directly attached the subject. Therefore, for animal training, the sensor is placed under the seat of the animal to monitor for any movement or vibration of the seat due to the animal moving. This is a safe placement that does not harm the subject. For human testing where the sensor has to be placed in a headband and wrapped around the subject's head, this device is not approved yet. Although it was tested and provided good data as far as movement sensing goes, it is not approved yet for general use in humans.

Chapter 5 LabState: Core Design

The software application was written in LabVIEW which is an ideal development environment to use, because it is commonly used for data acquisition and instrument control. It has built-in support for parallel execution which can easily support the execution of the main application and its modules simultaneously. The general execution of LabState is shown in **Figure 5-1**. When LabState starts, it is in standby mode. The user opens up the correct parameter file or changes current settings manually before starting an experiment. At the start of the experiment, all the used modules and functions are executed along with the DAQ unit. At the same time, the display unit starts to show all the acquired data, and the alarms & alert unit starts to analyze acquired data for out of range values. All the acquisition and display processes run parallel to each other and they shared data using queues.

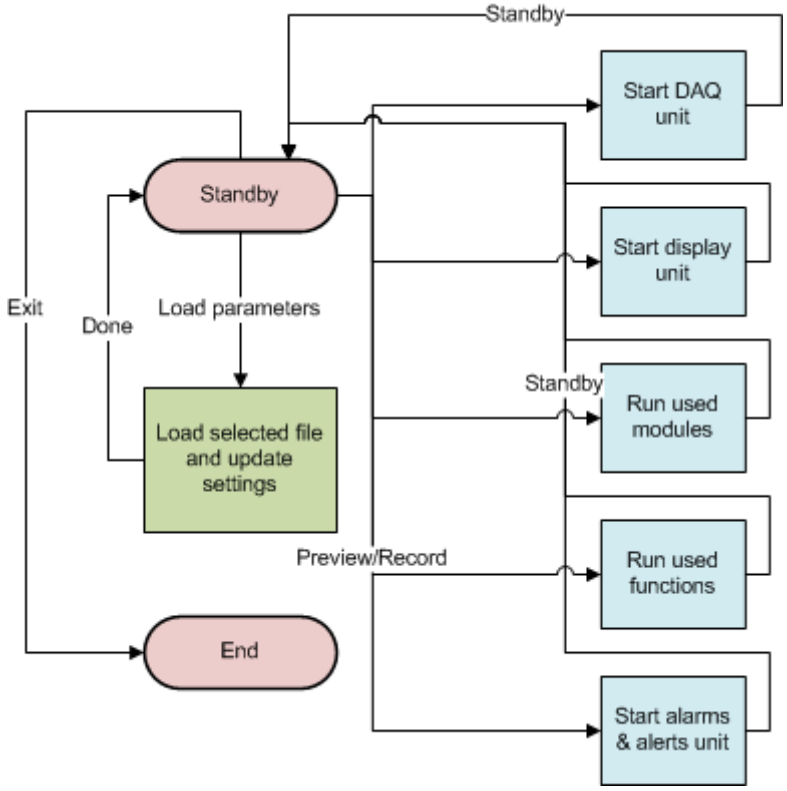

Figure 5-1: Flowchart diagram of general LabState software execution.

5.1 DAQ Unit

Analog and digital data sharing is established through a *Producer/Consumer* design pattern. This design pattern is usually used when multiple processes need to be handled simultaneously, while they run at different rates [42]. Two processes run in parallel. One is called *producer* and the other is called *consumer*. The producer reads data from the DAQ card and puts it in a queue. The queue is a data buffer. Theoretically, it can hold as much data as the system memory allows. The consumer simultaneously removes data from the queue whenever it becomes available. The consumer process always runs faster than the producer process. If it does not, the producer is putting data in the queue faster than the consumer can remove it. This eventually leads to full memory, so no more data can be added to the queue. Both processes keep running simultaneously until the experiment or data acquisition is stopped.

In the DAQ unit, there are 3 total producer/consumer processes, as shown in **Figure 5-2**. First one is a producer (P1) which reads data from the DAQ card. Second one is both a consumer and producer (CP2). The consumer part (C2) removes data from queue #1 which was written to by producer P1. This data can then be recorded and/or passed on to any functions that require the data. Producer P2 puts some or all the data in queue #2. The data is a two-dimensional array where each row corresponds to data from one analog or digital channel. If the channel is only being recorded and not displayed on the screen, it is not put into queue #2. This is because the consumer of queue #2 (C3) is part of the display unit which removes data from the queue and displays it on the screen. The reason why there are two chained producer/consumer processes is to enable the display unit to display data from the recording unit. This speeds up both the display and recording of data because both are running simultaneously. If there are jitters or pauses in the display unit, it does not affect the recording unit which needs to execute fast all the time so it passes data to the functions properly and the buffer

on the DAQ card does not get filled up.

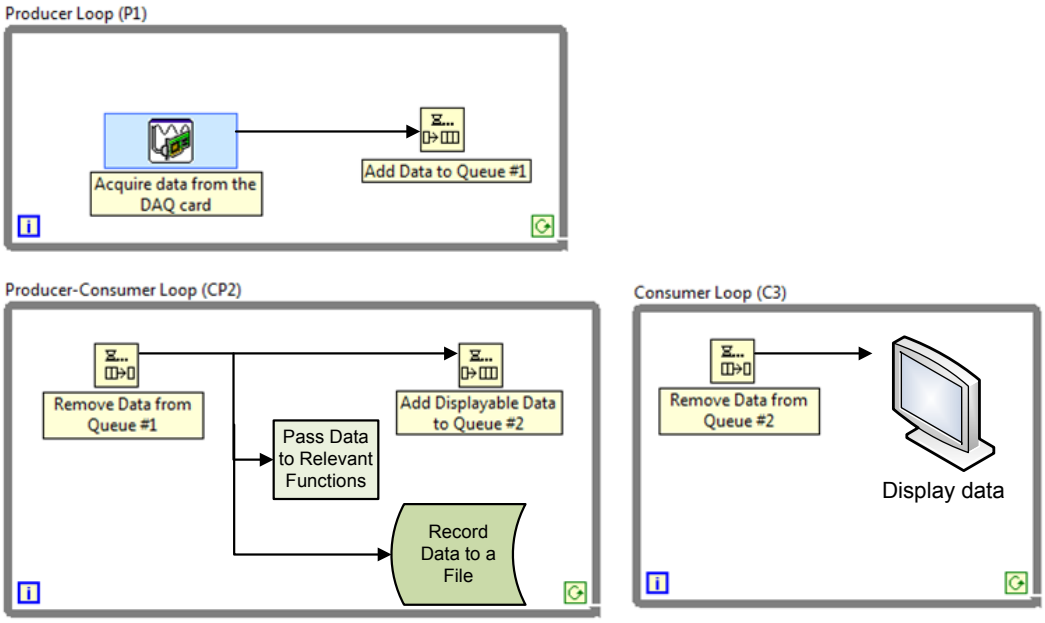

Figure 5-2: Data input and output scheme of three producer-consumer processes.

The DAQ unit is responsible for acquiring analog/digital data from the card, saving it to a data file if in record mode, calling any used functions and passing them the required data, and putting data in a queue so the display unit can access it.

5.2 Correlated Digital Input

Usually most DAQ cards that allow the acquisition of digital data do not have an internal clock to time the acquisition. The data is acquired as fast as possible. This way the user cannot acquire the data at a specific sampling rate. The DAQ cards that are used in the lab allow correlated digital I/O which means that, even though there is no internal clock for digital I/O, an external clock can be used to drive the sampling of digital data. The timing diagram for analog and digital sampling is shown in **Figure 5-3**. Digital data acquisition task is started before the analog one so it waits for the analog task to start. As soon as analog task starts, it starts to sample and acquire data. Whenever analog data is sampled, clock pulses are produced in the DAQ card's clock named *AI SampleClock*. Every time a sample is

about to be acquired, a brief pulse is produced in *AI Sample Clock*. Essentially this clock has pulses of frequency which are identical to the analog sampling rate. Therefore, this clock is used as an external clock to drive the correlated digital acquisition. This way digital data sampling starts at the same time as analog sampling, and it has the same sampling rate.

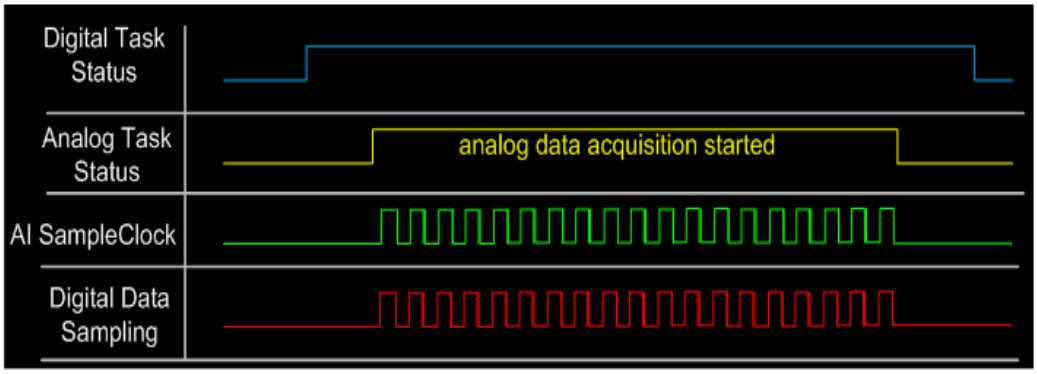

Figure 5-3: Timing diagram of analog and digital data sampling.

Digital task has to be started slightly before the analog task so that it does not miss any initial clock pulses that might be generated if the analog task starts before the digital one. If an analog task is started before the digital task then some samples of analog data are already acquired before the digital data acquisition starts. This will result in different data size for analog compared to digital data, and they will not be aligned in the time domain.

5.3 Multiple Sampling Rates

As already discussed before, sampling rate of digital data is identical to that of the analog data because it is correlated with the analog sampling clock. Additionally, none of the DAQ cards used support acquisition of data from different channels at different sampling rates. This is a hardware limitation, because there is only one analog sample clock available at any given time. So simultaneous acquisition of data from multiple channels can only take place at the same rate, as defined by the analog sample clock. In our lab, we needed to acquire data from some channels at much lower rate than other channels. For example temperature from an external device is acquired through an analog channel. It is not necessary to acquire it at 5 kS/s as used in many other channels because the temperature does not change that

fast. It is reasonable to acquire it at 1 S/s. In order to be able to acquire data at different sampling rates for each channel, the data is first acquired at the maximum sampling rate and then decimated. Decimation of analog signals involves filtering of data using a digital filter with the high-pass frequency set to half the new sampling rate and then resampling it. No filter is used while decimating digital signals. There is however a limitation with this method: the maximum sampling rate must be divisible by each channel's sampling rate. Maximum sampling rate is the highest sampling rate of all the channels defined. These channels include both analog and digital ones. The decimation factor for each channel is given by:

$$Channel\_DF = \frac{Max\_Sampling\_Rate}{Channel\_Sampling\_Rate}$$

Modified data for each channel is acquired after decimation:

$$New\_Channel\_Data = decimate(Channel\_Data, Channel\_DF)$$

This decimation is only performed for data meant to be recorded to a file. For display purposes, all the data is decimated by a single factor – one for analog and one for digital. Modified data for each set of channels is given by:

$$New\_Ana\log\_Data = decimate(Ana\log\_Data, Ana\log\_DF)$$

$$New\_Digital\_Data = decimate(Digital\_Data, Digital\_DF)$$

Table 5-1 shows an example set of channels with different sampling rates and decimation factors.

Table 5-1: Multiple channels with different desired sampling rates, analog decimation factor of 10 and digital decimation factor of 5.

	Recorded (Desired) Sampling Rate (Hz)	**Acquired Sampling Rate (Hz)**	**Display Sampling Rate (Hz)**
Analog Channel 1	1000	10000	1000
Analog Channel 2	10000	10000	1000
Analog Channel	2000	10000	1000

3			
Digital Channel 1	5000	10000	2000
Digital Channel 2	1000	10000	2000
Digital Channel 3	2000	10000	2000

5.4 Alarms and Voice Alerts

One of the main features of LabState is to sound alarms whenever certain data values are out of range, and output alerts of one or more data values after fixed intervals. The alarms can be configured for any number of channels, but all of those channels need to be configured for display and not just recording because the data queues used for displaying data are used by the alarms unit. For each channel, a range for the data is defined outside of which an alarm is sounded. If multiple channels have values outside of their respective ranges then still only one alarm is sounded until all the channels' values come within their respective ranges.

Voice alerts are computerized voices that artificially speak values from one or more channels. The interval of alerts is one minute. For example users can enable alerts for heartbeat so that a computerized voice (using Microsoft Agent technology) will repeat every one minute: "The heart rate is 200 beats per minute". This is useful in cases where the user or experimenter is focusing on performing surgery or is busy with something else, and is not watching at the monitor. He/she can still hear important values such as heart rate and respiration rate so if they increase or decrease the experimenter can perform the proper procedure to keep them steady.

5.5 Comments and Variables

LabState has a built-in feature which not only lets the user add text comments with automated timestamps, but also modify variable values with automated

timestamps. **Figure 5-4** shows the interface for adding comments and modifying variables. All the variables are defined at the beginning of the experiment. Then during the experiment the user can click a button and update the value of one or more existing variables. For example, for most experiments in our lab, one of the variables defined is called "gas anaesthesia level" which is initially set to 5%. During the experiment the user usually needs to keep changing the relative volume of the gas anaesthesia to keep the animal from waking up (increase) or be too anaesthetized (decrease). For future records, and automated anaesthesia control, it is useful to know when anaesthetic level was changed and by how much during the experiment. This variable is then compared to other variables and physiological state of the subject for statistical analysis. Instead of making all these changes in the notebook, the user can simply update the variable "gas anaesthesia level" to a new value whenever needed. All the changes are properly stored in the data file (with timestamps of milliseconds precision) for later use.

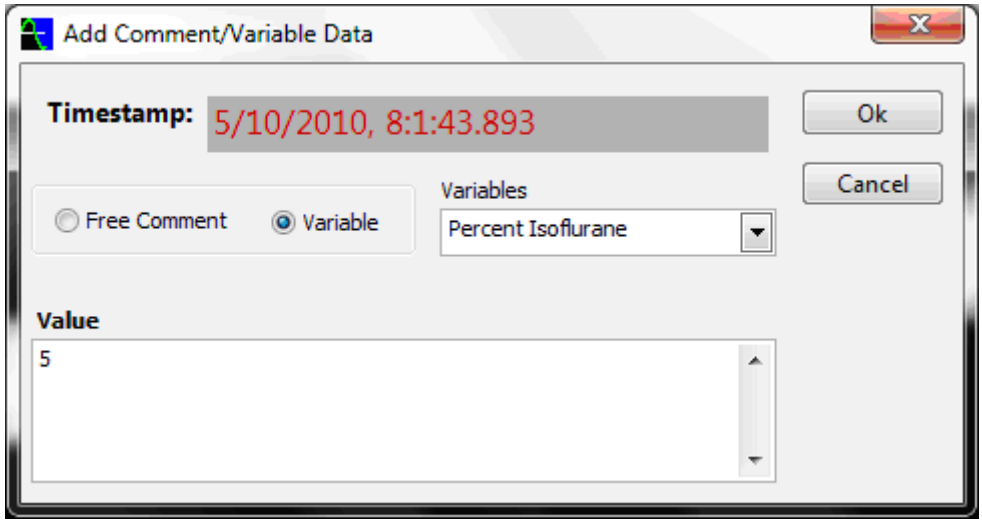

Figure 5-4: Image of interface for adding comments and modifying variable values.

5.6 Event Log

The timing of software programs is not perfect, even with the parallel multithreading capabilities of LabVIEW. When a process is started or stopped, there may be nanoseconds to milliseconds delay between when the process actually executes and when the command is sent to execute it. The design of

LabState is such that the core analog/digital DAQ unit runs separately from each module and function. However the DAQ unit and most modules acquire data that needs to be synchronized. For example, SpO_2 is acquired from the pulse oximeter using a module. This acquisition starts as soon as DAQ unit starts acquiring analog/digital data because the DAQ unit sends a signal to the module which is waiting for it to start. There is milliseconds of delay between the start of analog/digital acquisition and SpO_2 acquisition. The delay is also not constant from experiment to experiment. In order to align the analog/digital data with the SpO_2 offline, the timestamps with the millisecond precision are stored in data file for the start of analog/digital acquisition and SpO_2 acquisition. In the case of SpO_2, it is not crucial that it is perfectly aligned with other data. A few milliseconds of delay is tolerable because SpO_2 does not change that often. However for other signals such as plethysmogram, it needs to be synchronized as perfectly as possible. Plethysmogram waveform's shape is linked to the heart beat. If an optical imaging system is used to take brain images during an experiment, there is noise in the images due to the heart beating and the resulting blood flow changes in the brain. This noise needs to be removed or at least minimized for proper analysis of the images. One way to remove this noise is the to record the plethysmogram, align it with the camera frames, and use an algorithm to remove heart beat noise based on the amplitude of the plethysmogram signal at each frame.

Whenever the DAQ unit or a module starts acquiring data, an event is created with a precise timestamp and the name of the module or DAQ unit that started acquiring the data. All the events are recorded in the experiment's data file for easy access offline. Modules can produce additional events as well. For example, a stimulation module produces an event every time a new stimulation is sent, along with some information about the pattern of stimulation. This information can be used offline to figure out exactly when the stimulation is sent relative to other signals (e.g. triggers).

Chapter 6 LabState Modules Design and Implementation

The core programming of LabState only supports simple recording and/or display of data from any supportive card by NI. However, for most experiments communication with other device is necessary. This is where the modules come into play. Each module can be created separately (as long as a few rules are followed) and then LabState can easily communicate with the module. The users can change the settings of each module through LabState and run it automatically whenever an experiment starts. I wrote several modules for the lab, including OEM III Vet pulse oximeter by Nonin, custom built airpuff stimulator, Capnostat 5 CO_2 monitor by Respironics Novametrix, MRI-1 ventilator by CWE, and EG00700 temperature sensor by MedLab. For each sensor, the module simply communicated with the device using RS232 communication protocol as described in the device's operating manual and acquired the data for display and recording. Most of the time, I simply programmed the device that the module communicated with but in some case I designed and built the device itself. A few important modules are discussed below.

6.1 Electrical Stimulation

This module allows the user to send digital pulses using the existing DAQ card. These pulses are then sent to a linear stimulator isolator unit which produces a constant electric current for the duration of the pulse. Once the module is started along with the experiment, there are three main states that it goes through: baseline, stimulus, and inter-stimulus interval (ISI). **Figure 6-1** shows general execution of this module. The same paradigm of stimulation is used for airpuff and rotatory stimulation. During baseline and ISI, there is no stimulation so the output pulse is low (at a voltage of approximately 0). A stimulation state is defined as the time period when a periodic signal of a certain frequency and pulse duration is generated. As mentioned earlier, a counter/timer is used to generate these precise pulses. When the module goes from stimulus to ISI state, it is defined as one cycle. It will go back from ISI to stimulus state until total number

of cycles meets or exceeds the user-defined limit. The exact time period for each state is defined by the user, and it is defined in units of *volumes* instead of fixed time units. *Volumes* refers to the total number of triggers counted in that state. Triggers are either external or internal pulses that are recorded by LabState and counted by the module continuously. For example, in one of the setups in the lab, triggers are generated by an optical imaging system, and each pulse in the triggers corresponds to the camera taking one picture frame. It is good to start/stop stimulation at the exact time when the frame is taken so that later on during analysis, we do not have to resynchronize the images with the stimulation pulses. In another setup, the triggers are generated by one of the modules of LabState because an optical imaging system is not used. The frequency of the triggers is defined by the user.

Once the count for one state is met, it moves on to the next state. If the triggers stop, then the module remains at the current state and only stops when the experiment stops. If the defined number of cycles have been executed then the module stops until it is restarted by the next experiment.

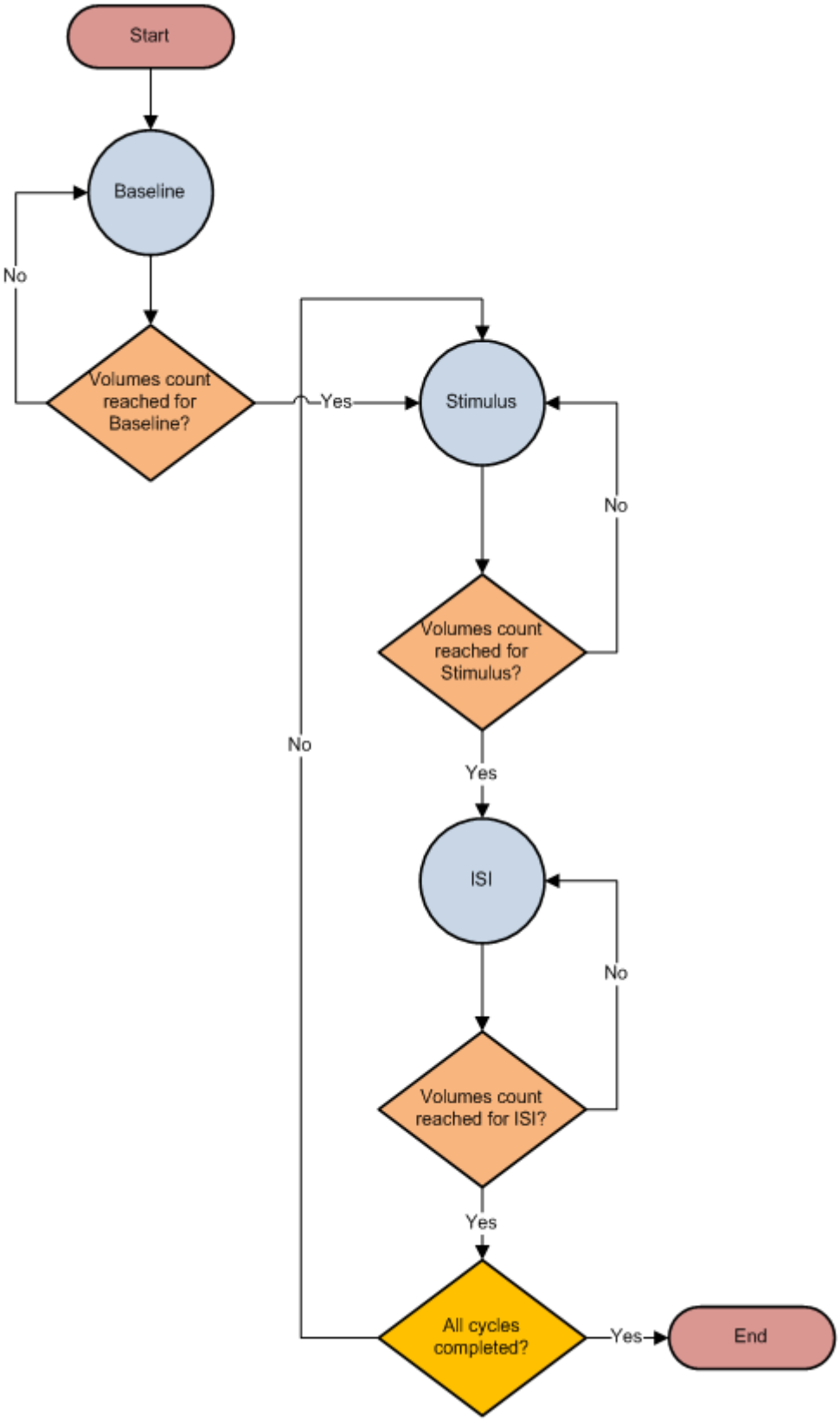

Figure 6-1: State diagram of the electric stimulation module.

Although the module is named after electric stimulation, it just sends out a TTL of a certain frequency and pulse duration. It can be used to control any device that depends on TTL input. It has been used in the MRI to perform stimulation in humans using a custom-built vibro-tactile device [43].

6.1.1 TMS stimulation

A sub-module of this module is a program that outputs digital pulses on a separate channel which goes to the TMS system. This sub-module has two states: baseline and stimulus, as shown in **Figure 6-2**. As presented previously, there are no pulses in the baseline state. The duration of the baseline state is defined by the number of triggers. For example, if the triggers are introduced at 2Hz and the number of triggers for baseline is defined as 50, then this sub-module will be in the baseline state for 25 seconds.

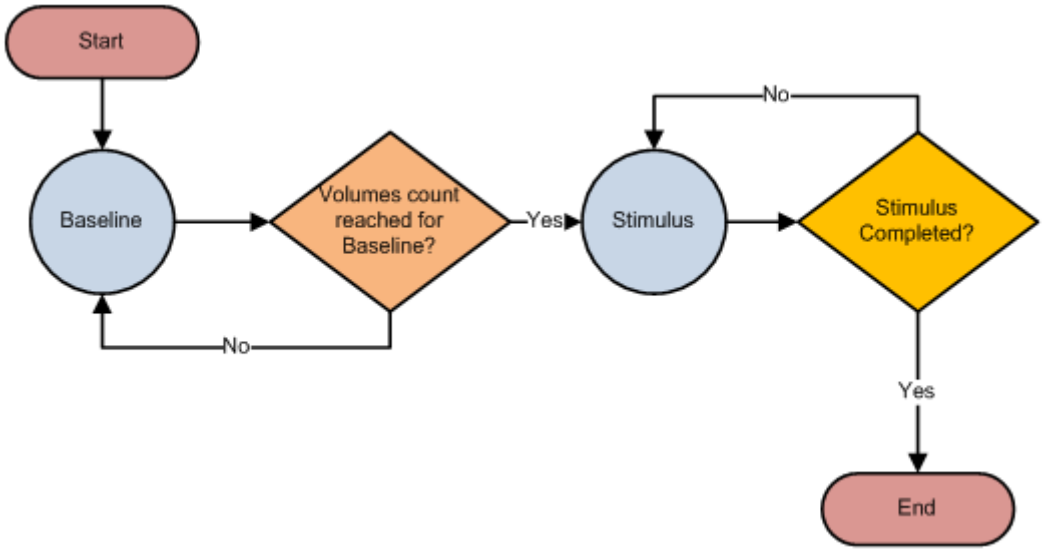

Figure 6-2: State diagram of TMS stimulation module.

The pattern of pulses during the stimulus state is a bit different from normal stimulation. **Figure 6-3** shows how the stimulus pattern is created according to the parameters. The pattern is defined by the following parameters: primary cycles, primary delay, secondary cycles, secondary delay, frequency, pulse Duration, number of pulses. The sub-module executes until all the primary cycles have been executed. All the secondary cycles are run in each primary cycle. At the beginning of a primary and secondary cycle, this is a period of time when the

output is low (no pulses are produced). This is defined by primary and secondary delay (in milliseconds). These delays can provide baseline periods in the data, because there is no stimulation. They also serve as inter-stimulus intervals when looking at the electrophysiological responses of the subject. During each secondary cycle, there is a certain number of pulses sent. Each pulse has a defined frequency and pulse duration.

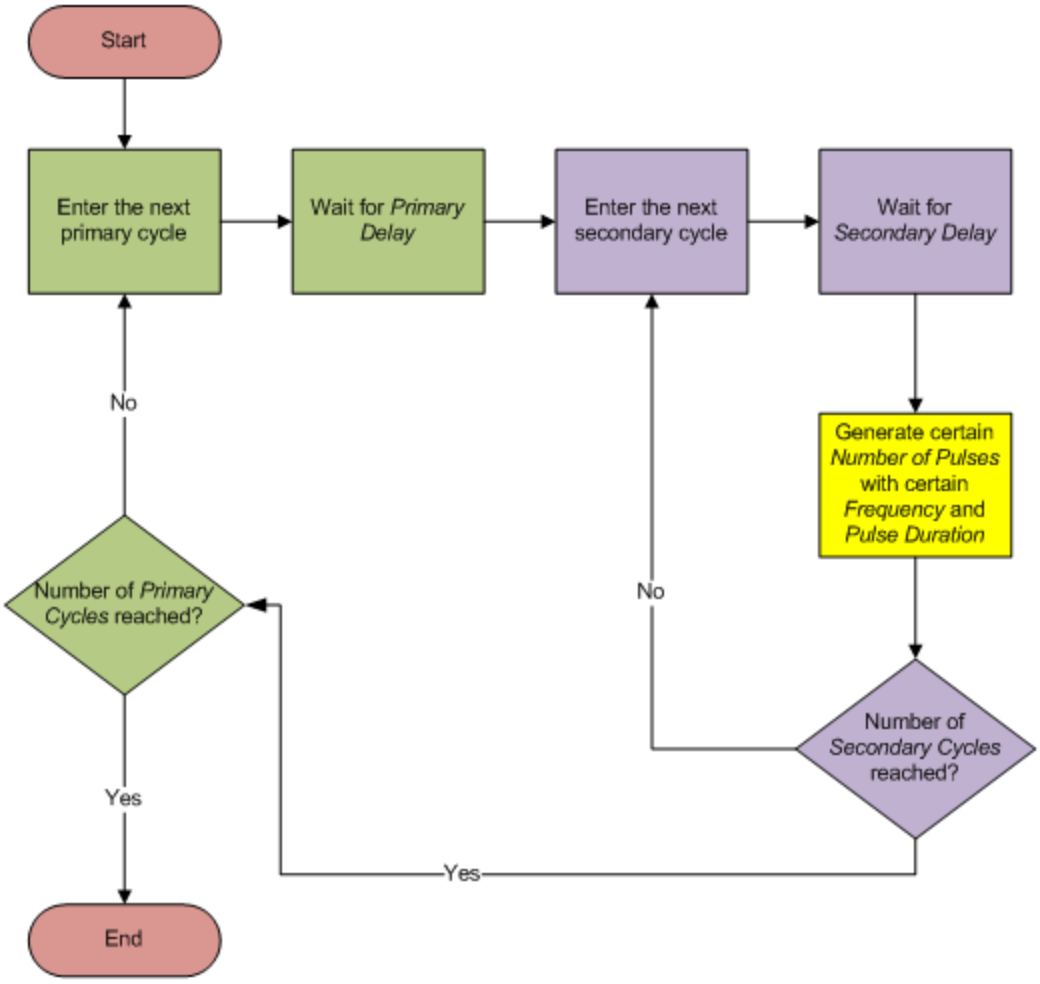

Figure 6-3: A flowchart diagram showing the execution of the TMS module during stimulation phase.

The stimulation pulse pattern that can be created using this paradigm is used to provide TMS theta bursts. An example pattern is shown in **Figure 6-4**. There are fast pulses applied (at 50 Hz) for a short period of time rather than slow pulses (eg: 1 Hz) for a longer duration. The paradigm is known to inhibit brain activity [44].

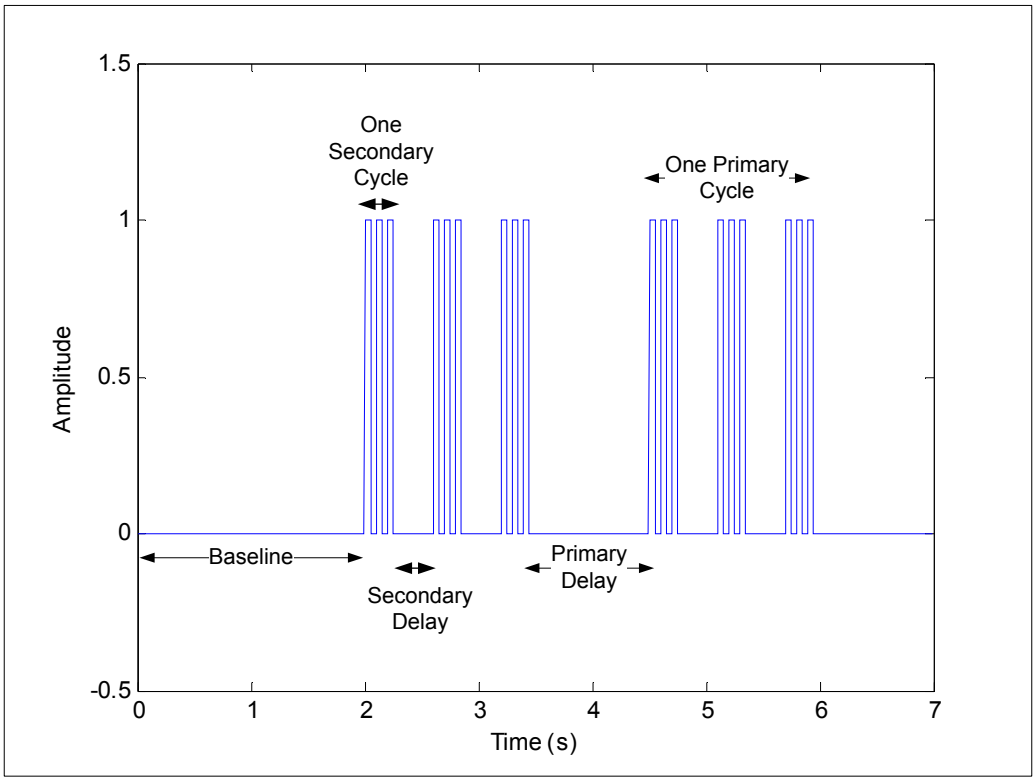

Figure 6-4: Graph of an example stimulus produced by the TMS stimulation module.

6.2 Airpuff and Rotatory Stimulation

Two additional stimulation methods are used in our lab: airpuff and rotatory. The airpuff method uses a device that sends out controlled puffs of air. This is used for whisker stimulation in rats. This type of stimulation differs from electric stimulation because it activates different areas of the brain and it is also more natural than electric pulse stimulation.

The module communicates with the airpuff device using a serial RS-232 protocol. The timing of the stimulation is defined just like in the electric stimulation module. It has three states and each state is defined by volumes. During stimulation, the module sends the device a command through the serial port which contains the frequency at which to produce the puffs. Whenever the stimulation state ends, the module sends another command which tells the device to stop the

air puffs. This keeps repeating until the defined number of cycles have been executed.

The rotatory stimulation module works in a similar fashion. The module sends the device a start command with the frequency whenever it comes to the stimulation states, and sends a stop command whenever the stimulation state ends.

The stimulation is usually performed on the subject's hands and the device consists of an electric motor with a soft drill brush attached to it. The brush rotates at a certain speed and it produces stimulation as long as it is in contact with the subject.

In order to be able to control the motor using a computer, a serial controlled motor driver was used [45]. The driver supported control of up to 2 motors using RS232 protocol. However, it could not directly be connected to the computer's serial port because of the voltage differences. The driver only worked at the voltage levels of up to 5 volts, but the computer outputs voltages between -12 and 12 volts. Direct connection can damage the circuit board. Therefore a RS-232 TTL shifter was used to convert high voltage levels into low levels for the driver to work. The circuit diagram is shown in **Figure 6-5**, and the complete product is shown in **Figure 6-6**.

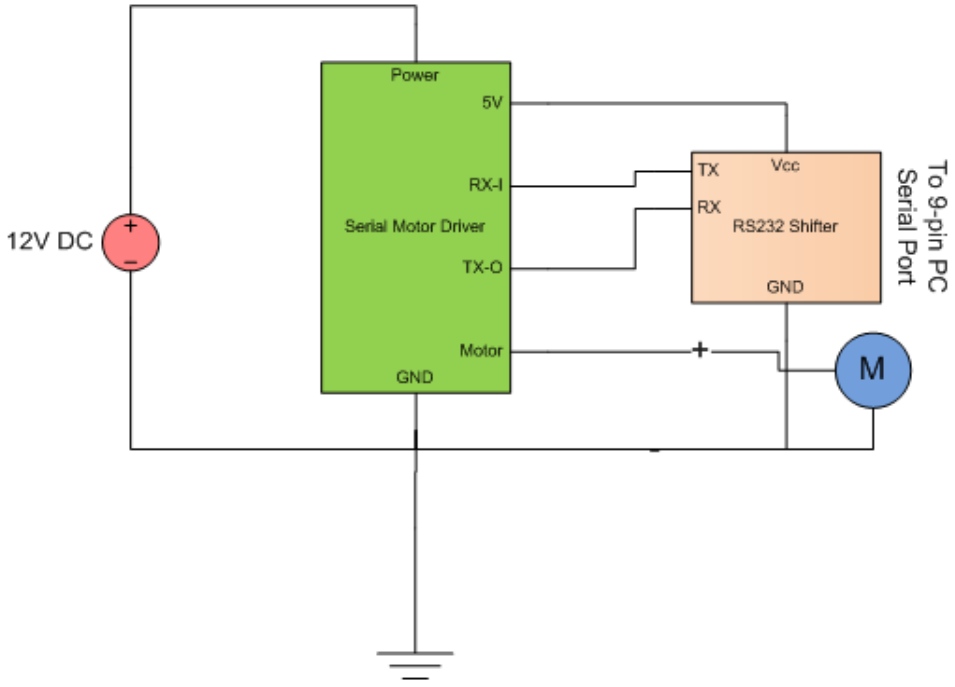

Figure 6-5: Circuit diagram for the motor controller.

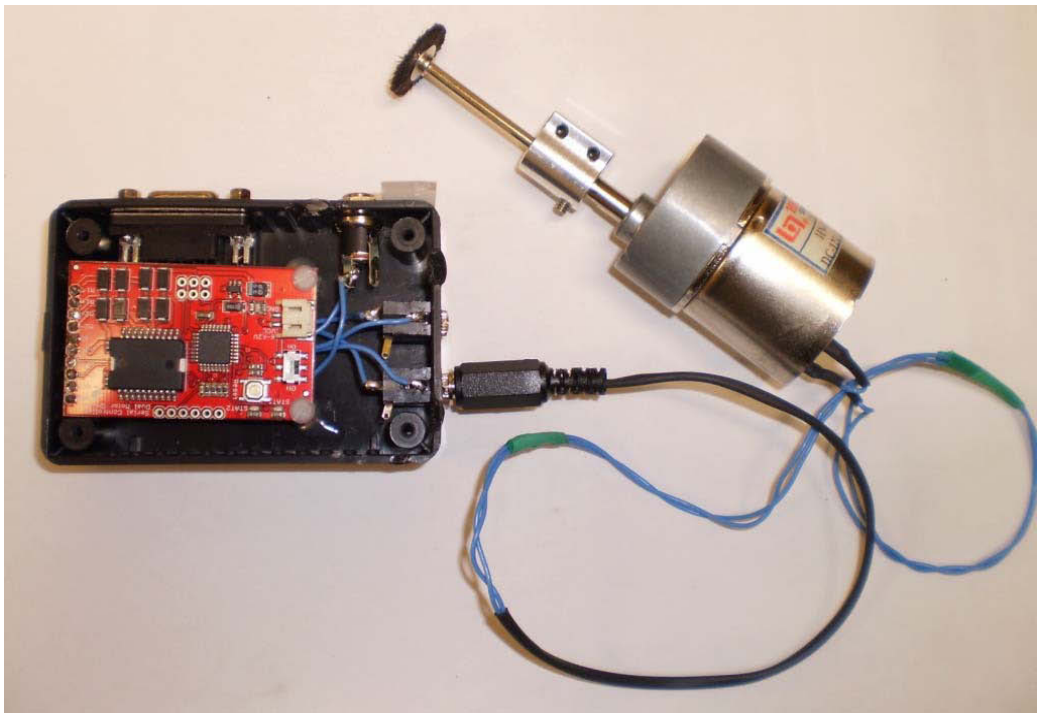

Figure 6-6: Assembled motor controller with motor and brush for somatosensory stimulation.

6.3 Juice Reward Controller

In the alert animal experiments, usually there is a reward system involved. In our lab, if the subject does everything as expected during a trial it gets a juice reward. If it makes a mistake then it gets no juice and that's how it realizes that it made a mistake [46]. This behaviour of an animal has been well documented. In order to give the juice reward, a gravity-based reward system (Crist Instruments) was purchased. It has a simple solenoid that closes or opens which in turn closes or opens the tube from which the liquid flows. The solenoid can be controlled electronically by a reward system controller which is purchased separately. The controller has many features including a timed on/off duration, delay in reward presentation, etc. None of those features were needed for the lab's applications. The reward system (the solenoid) is only needed to be turned on or off based on a TTL signal. If the TTL is high then the juice reward is on and if the TTL is low then the juice reward is off. LabState or another computer-controller system provides the TTL. A hardware unit that controlled the reward system based on an external TTL signal was produced. It turned out to be a much cheaper option that the reward system controller offered by Crist Instruments. The circuit diagram of the system is shown in **Figure 6-7**, and the complete system is shown in **Figure 6-8**. A simple transistor is needed to control the solenoid based on the TTL. An integrated circuits chip, ULN2803A (STMicroelectronics), was used because it contains 8 transistors, meets the current and voltage requirements for the solenoid, and circuitry for additional protection. Each output (OUT *) on the chip supports a maximum current of 500 mA. The solenoid used by the juice control has a current rating of 500 mA. Although the output can pass enough current for the solenoid, a second output (OUT 2) is connect in parallel with the first output (OUT 1) to double the maximum current output to 1 A. Because the outputs are connected together, the inputs are also connected together so they both receive the same TTL signal as the exact same time.

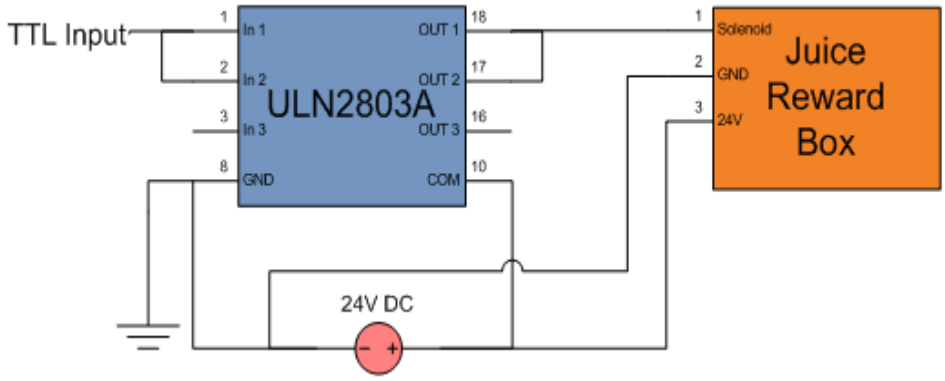

Figure 6-7: Circuit diagram of the juice reward controller.

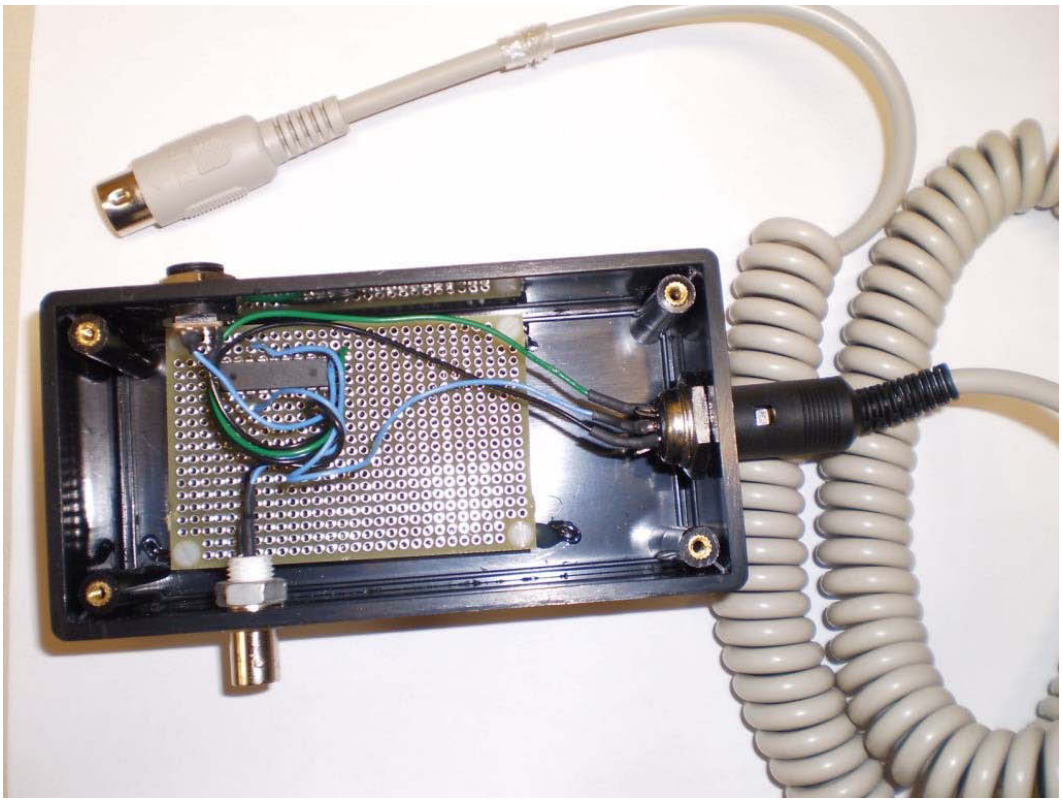

Figure 6-8: All the juice reward controller components assembled together.

The internal circuitry of one input-output of the chip is shown in **Figure 6-9**. There is a diode connected between one of the transistors' collector pin and COM. This protects the chip from reverse voltage which is produces whenever a solenoid is turned off. The maximum reverse voltage that it can handle is 50 volts. This is more than enough, because the highest voltage present in the system is only 24 volts.

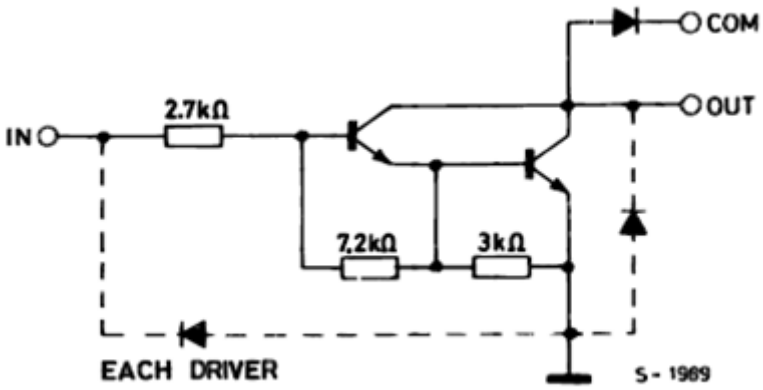

Figure 6-9: Circuitry between one input and output of ULN2803A [47].

Chapter 7 LabState Functions Design and Implementation

7.1 Get Rate

During any anaesthetized experiments, the physiology of the subject is continuously monitored, to make sure the subject is in a stable condition. The monitored parameters can include heart beat, respiration, heart rate, and SpO_2 level. In our lab, OEM III veterinary pulse oximeter (Nonin Medical) is used to monitor both the heart rate and SpO2. However, heart rate from the ECG is calculated to have another source of measurement besides the pulse oximeter, in case the device stops functioning during an experiment for reasons of low-quality oxygenation signal. There have been instances where the pulse oximeter produced a very weak plethysmogram and it couldn't get a proper heart rate because of it. This weak signal was usually caused by the fact that the subject did not have a sufficient blood flow at the site of the sensor placement. However, changing the location of sensor placement usually fixed this issue. So the calculated heart rate from the ECG was only a secondary method for getting the subject's heart rate. Even though it produces good results in the lab setting, it is not used as a reliable method for continuously monitoring the subject's heart rate. Pulse oximeter is still the method of choice.

7.1.1 Calculation methods

In the scope of the current thesis, it was required to apply a simple design for detecting heart rate from ECG. It was required to function in the lab setting, but was not meant to be robust to different electrode placements. There are many methods to calculate the fundamental frequency of a periodic signal, depending on the shape of the signal. If a signal has periodic spikes then a spike or peak detection algorithm can be used. Otherwise a method that involves Fast Fourier Transform (FFT) can be used. FFT takes an input signal and produces an output signal which shows the strength of the signal at a set of frequencies. For ECG specifically, there are countless methods published for detecting the QRS complexes in the ECG signal and calculating the heart rate based on that [48-52]. Different methods have shown variable success rate as shown in the tests

performed by the authors. Rather than performing a thorough review of all these methods, select the best one, and then implement and test it, only simple peak detection or FFT-based method were used. This is because we wanted to use a simple method that did not take long to implement and worked in our lab setting. A good ECG consists of periodic spikes, as shown in **Figure 7-1**. Notice how the peaks actually occur in the negative axis because the electrodes placement was reversed during that specific experiment. Hence it makes sense to use a spike detection algorithm. There are various algorithms already published, and many of them use wavelet transform [53-55]. However as a quick implementation, those complex algorithms were not used. A simple peak detection function available in LabVIEW was used instead. It has two main parameters: minimum inter-spike interval, and threshold. Minimum spike interval is the time, in seconds, during which it is assumed that no spike occurs. A very small number (1 ms) is used for this parameter so the spikes can be detected in ECG even if the frequency of spikes is very high (up to 10 Hz). The threshold is the amplitude above which the signal is considered to be a spike. This value can be set by the user under the module settings because it can change depending on the strength of the ECG. However, this method has problems. When the user places electrodes on the subject to record ECG, they are not always at the exact same spot each time. The strength of the signal can change from subject to subject, and finally the signal may be inversed because the two differential electrodes have switched placed. Due to any of these problems, the threshold value might need to be changed and the signal might need to be inversed before spike detection. This has to be done before each experiment which is not convenient. The Get Rate function should have the parameters set only once, assuming that the input is an ECG signal and the output is the heart rate.

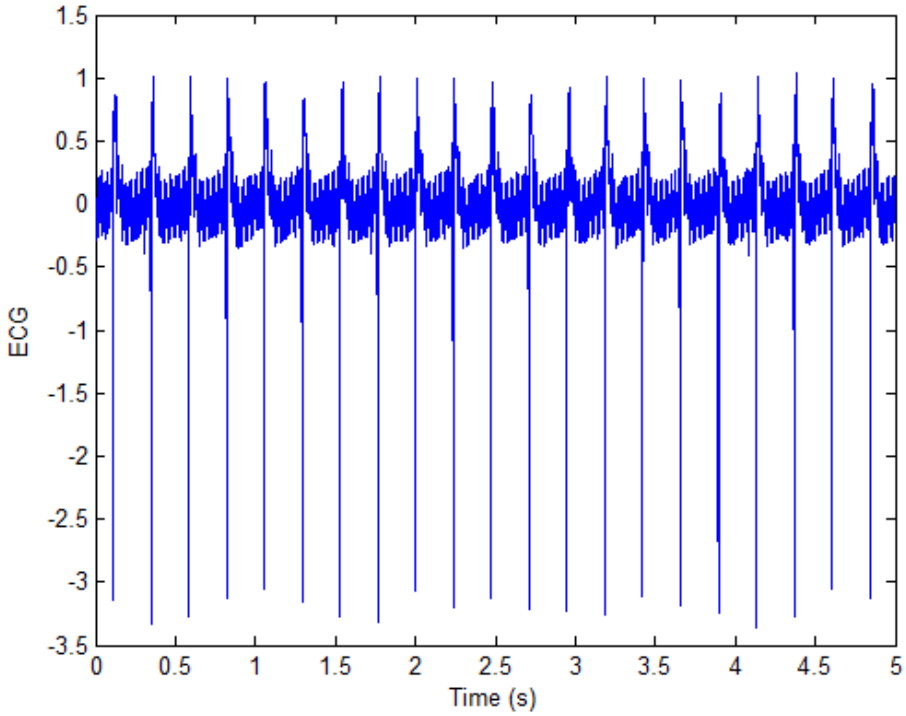

Figure 7-1: ECG signal recorded from a rat.

The second method that was tried involved filtering the signal and calculating its FFT [56-57]. Filtering is necessary because the signal contains high frequencies due to the surrounding noise. Keeping these frequencies can give a false positive output. A frequency range of 0.5 to 10 Hz is used for bandpass filtering so it encompasses the expected range of all the animals including rats, cats, and monkeys. Following the filtering, FFT is performed and then the frequency with the maximum amplitude is considered to be the fundamental frequency and hence the heart rate. The problem with this method is that sometimes the frequency that has the maximum amplitude does not correspond to the real heart rate. It is one of the harmonics of the fundamental frequency so the detected heart rate is usually double or triple of the real heart rate.

The third method that was used involved FFT and filtering too. First the FFT of the signal was obtained and the signal was filtered using the same frequency ranges as method #2. This way, all the frequencies outside of the expected range were completely eliminated. In addition, FFT data within the required frequency bands was extracted by indexing. The standard deviation of the resultant FFT data

was calculated, and the first frequency that had amplitude higher than a certain multiple of the standard deviation was considered to be the fundamental frequency and hence the heart rate. The multiplication constant, K, was set to 3 after testing different values with a large dataset. This value was optimal because it was large enough to not allow non-fundamental frequencies with high amplitudes to pass the threshold, yet it was small enough so that the fundamental frequency passed the threshold. This method works for any regular ECG signal (inversed or not) unless it's very noisy.

7.1.2 Visual comparison of methods

All three methods mentioned were used to compute the heart rate from an ECG signal and compared to the actual heart rate from the pulse oximeter. This is shown in Figure 7-2. Top graph shows a chunk of ECG upon which the algorithms were executed. Bottom graph shows the heart rates calculated from all three methods and pulse oximeter, over time. A buffer of 5 seconds is used for each method. This is why the heart rate is -1 for initial 5 seconds. For this period there is not enough data to perform any calculation. For spike detection, threshold of 0.5 was used because it is clearly visible from the top graph that spikes become distinct from baseline signal at or above this amplitude. The sampling rate was 10 kHz and a bandpass filter of 0.5 to 10 Hz was used for all the methods except spike detection for which no filter was used. The threshold multiplier value for FFT STD threshold method was set to 3.

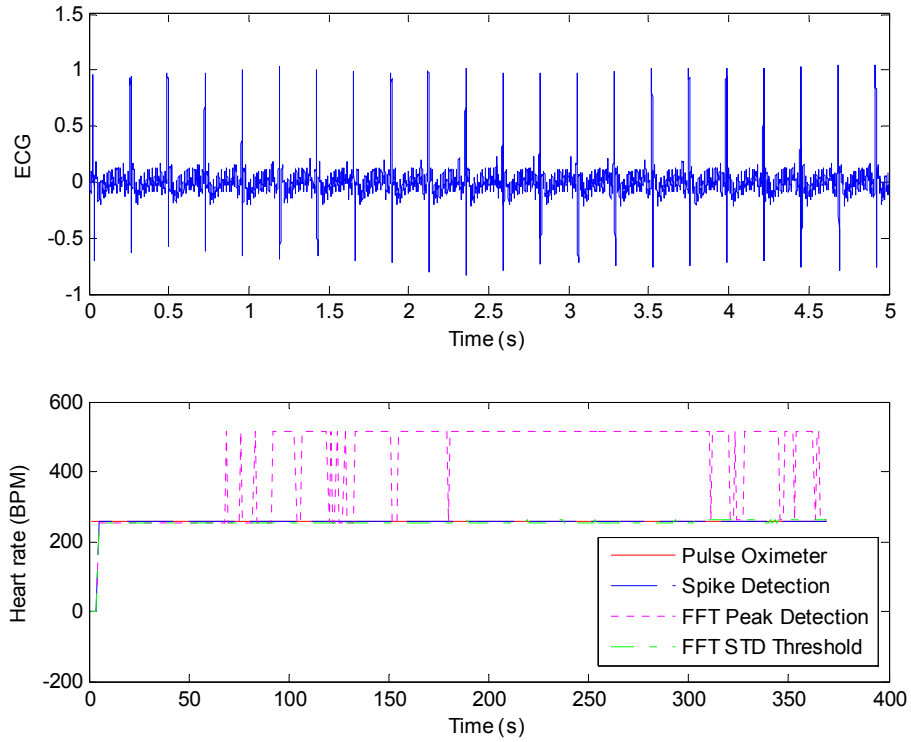

Figure 7-2: Sample ECG signal (top) and comparison of heart rate calculation methods (bottom).

Both the spike detection and FFT STD threshold method calculate heart rate which is very close to the one calculated by the pulse oximeter. However, FFT peak detection method outputs values that are almost double the actual heart rate occasionally. This shows that the best method to use is FFT STD thresholding among those tested within the scope of this thesis.

The same method is used to calculate the respiration rate from a respiration pad signal. The signal is similar to a sine wave, as shown in **Figure 7-3**. There are two reasons why spike detection is not the optimal method to use. First, there are no sudden spikes in the signal. The width of each local maxima is much wider than in the case of ECG. Secondly, the overall signal has some kind of trend (mostly linear) in it. The reason for that is unknown. It is most likely due to the internal mechanism of the respiration monitoring device. This trend keeps gradually changing the maximum amplitude of the signal. Because the spike detection algorithm uses a constant threshold value and assumes the maximum amplitude of

the signal stays almost the same during the experiment, it cannot be used on a signal which has a linear (or a higher degree) trend.

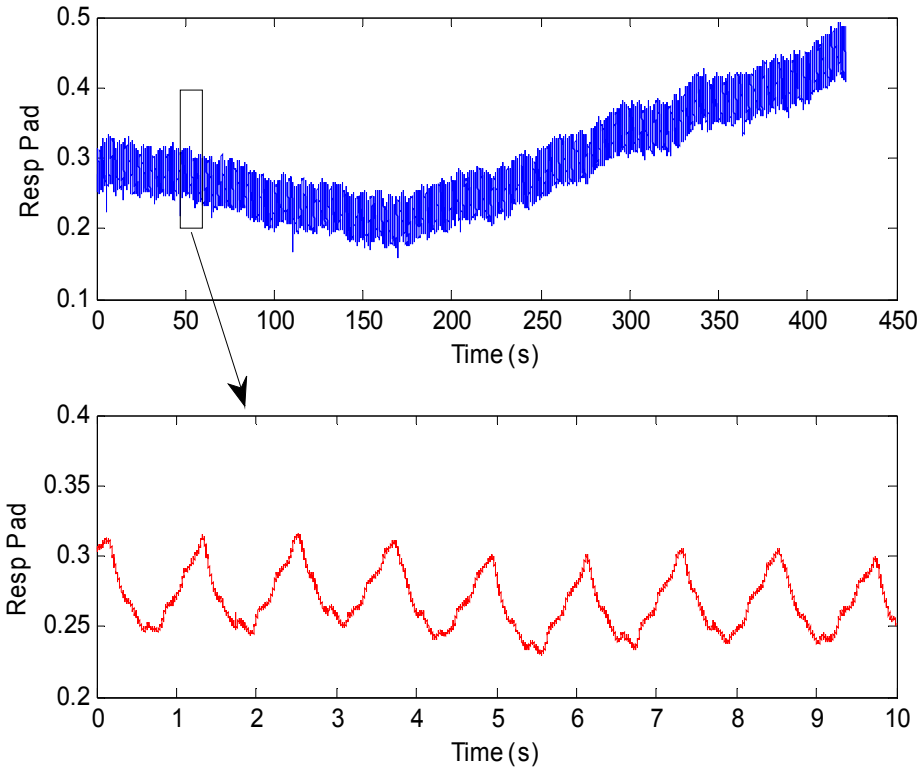

Figure 7-3: Graph of data from a respiration pad.

Using the second method presents the same problem as for the ECG. Occasionally, the amplitudes of 2nd or 3rd harmonics is higher than the amplitude of the fundamental frequency, which produces double or triple the actual respiration rate. Therefore, the third method is used again for calculating the respiration rate. The optimal standard deviation threshold value is found to be 3. The bandpass filter, however, is in the range of 0.1 to 2 Hz to take into account the expected respiration rates for rats, cats, monkeys, and humans. For humans, a respiration belt is tied around the subject's waists and it produces a signal similar to a respiration pad which is placed under the belly of the animal.

As mentioned earlier, the method implemented here to calculating the rate of the incoming waveform (whether it is ECG or respiration signal) is very basic and does not necessarily work for all the situations. Especially when the noise levels are different (due to change in the lab environment) or when different amplifier

settings are used for both signal, this may give inaccurate results. For heart rate calculation, this function is used only secondary to the pulse oximeter and it cannot be completely trusted to keep the subject safe. Precise, reliable heart rate calculation from ECG was beyond the scope of the thesis.

Chapter 8 StillState Design

MRI and functional MRI are currently widely used to take images of the brain for research purposes. During an MRI scan, the subject has to stay still, and not create any movement. Otherwise head motion can cause profound artifacts in the resulting images. Even a small change can corrupt the BOLD signal which can lead to false conclusions [16]. There are many artifact removal methods available that perform motion correction, but data obtained when no motion occurs during the MRI scan will always be better than data obtained by motion correction.
This created a need to develop a motion detection system that could be used to train the subject for the MRI scans, so that the motion during real experiments reduces to minimum to none. This motion detection system can either be used with human subjects (children) to train them to not move, or it can be used during alert animal experiments to train them to stay still.
For the human subjects, the system is very small. It is attached to a simple headband which can be tied around the forehead. Once the subject puts it on, it is calibrated and then the subject is trained. More details about this are presented below. For the alert animals, the system is attached to the bottom of the surface on which the subject is located. This way any kind of body movement will show occurrence of motion, so that proper steps can be taken afterwards.

8.1 Hardware Design

As described earlier, one of the main pieces of hardware used is an analog accelerometer with the ability to detect motion in two axes (x and y). A complete circuit diagram of the accelerometer function is shown in **Figure 8-1**. In order to record the analog accelerometer signals properly without any aliasing and high frequency noise, they have to be hardware filtered. The capacitors C_x and C_y are used to define the low-pass filtering band for x-axis and y-axis signal, respectively. The maximum frequency for each filter is given by the following equation [58]:

$$F_{-3\,dB} = \frac{5\ \mu F}{C_{(X,Y)}}$$

Usually the fastest movement expected for a subject is not going to exceed approximately 30 Hz, so low-pass filter from 0 to 50 Hz is chosen for both x and y axis. To achieve a -3dB cut-off frequency of 50 Hz, capacitor value of 0.1 µC is chosen. In order to decouple the power supply from the accelerometer chip, a capacitor C_{DC} is used in parallel with the power source. This reduces the noise coming from the power supply. The decoupling is necessary because the input power supply does not necessarily supply a constant regulated voltage all the time. Temporary changes in current requirement by the circuit create temporary changes in the supply voltage. The capacitor reduces these fluctuations in the circuit side. For additional noise removal, a 100 Ω resistors is used in parallel with C_{DC}.

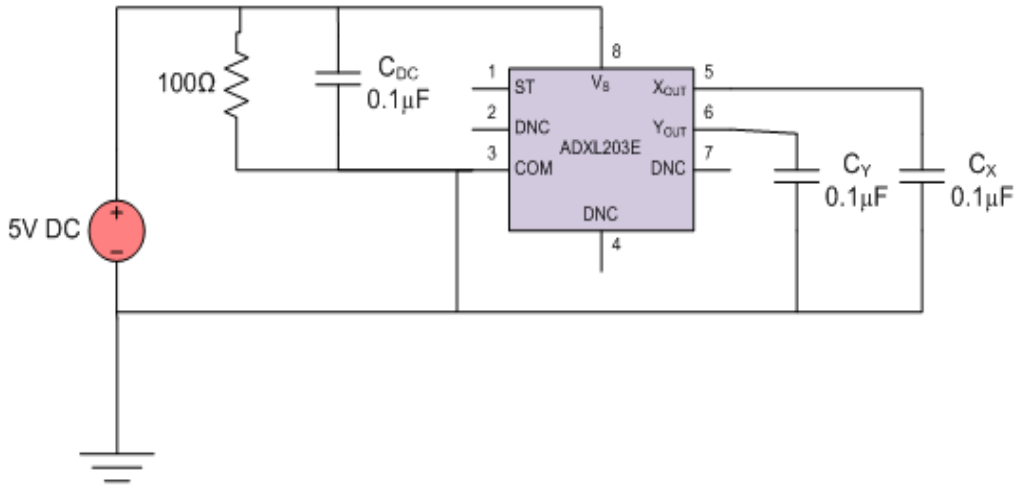

Figure 8-1: Circuit diagram of the body movement detection system.

Once the entire system is built from the circuit diagram, the power input and accelerometer outputs are connected to the USB 6008 DAQ card using a custom built cable. The cable contains shielding to reduce external environmental noise. The DAQ card automatically supplies the required power to the system upon execution of the software application. **Figure 8-2** shows the complete package.

Figure 8-2: Image of complete body movement detection hardware.

8.2 Software Design

The software application for StillState was also written in LabVIEW. It lets the user preview and record data from both axes, along with other features such as calibration, alarms, and trigger output. The state chart of the software is shown in **Figure 8-3**. Initially the program waits for the user input. Before starting motion detection, the system must be first calibrated. The user presses on *Calibrate* to start the calibration. During this procedure, the subject must stay completely still. The calibration can last as long as the user wants but usually 5 to 10 seconds is sufficient, as long as the subject was lying still during that time period. Once calibration has been performed, the user can start the motion detection by select *Preview* or *Record*. Preview method does not record any acquired data to a file on the hard disk. Once the data acquisition has started, the program will sound the alarm through PC speakers if the *Enable Alarms* option is selected. The criteria for an alarm is defined by the standard deviation (SD) factor value entered. After

the calibration is performed, the program calculates the mean and SD values for each axis from the data that was acquired during the calibration. These values are saved and then used along with SD factor to decide whether motion has occurred or not. A motion event is logged whenever the respective signal value changes beyond the lower and upper bounds, defined by the following equations:

$$Lower\ Bound_{(xyz)} = Mean_{(xyz)} - (SD\ Factor)SD_{(xyz)}$$

$$Upper\ Bound_{(xyz)} = Mean_{(xyz)} + (SD\ Factor)SD_{(xyz)}$$

If enabled by the user, the program also sends a TTL signal out of one of the digital ports of USB 6008 (or another NI card if the movement detection is a function of LabState). The TTL signal is low when there is no motion and it is high while there is motion. This signal can be recorded to later check offline when a motion had occurred. In one of the lab setups, it is sent to a control system that controls alert animals experiments and during an experiment if it receives a TTL pulse which is high, it recognizes that as an occurrence of motion and may stop the current trial.

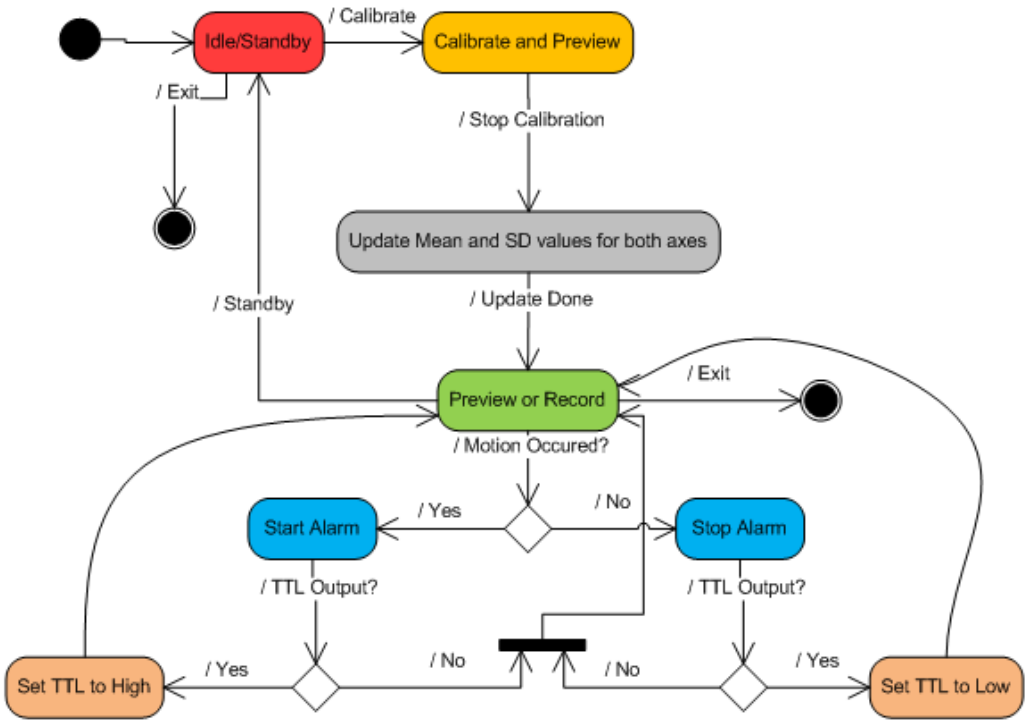

Figure 8-3: UML state chart diagram of StillState software execution.

8.3 Movement Detector Function

A LabState function was written to perform movement detection using the accelerometer-based hardware described above. This function has the same functionality as StillState, except that it does not communicate with the hardware directly to acquire or record data. The advantage of writing a function rather than using StillState in conjunction with LabState (when additional data needs to be acquired) is that it eliminates the need for a second DAQ card which StillState requires in stand-alone mode. It also provides the user with synchronized data in a single data file.

Just like any other function, Movement Detector has two modes: offline and execution. In the offline mode the user can change the settings of the function which are same as StillState settings. The system can also be calibrated right before an experiment starts. The execution mode is described in **Figure 8-4**. Unlike StillState, this function gets the acceleration data continuously from LabState. This is because LabState has control over the DAQ card so the function itself cannot access any input channels on it. The data is directly recorded by LabState into the experiment data file.

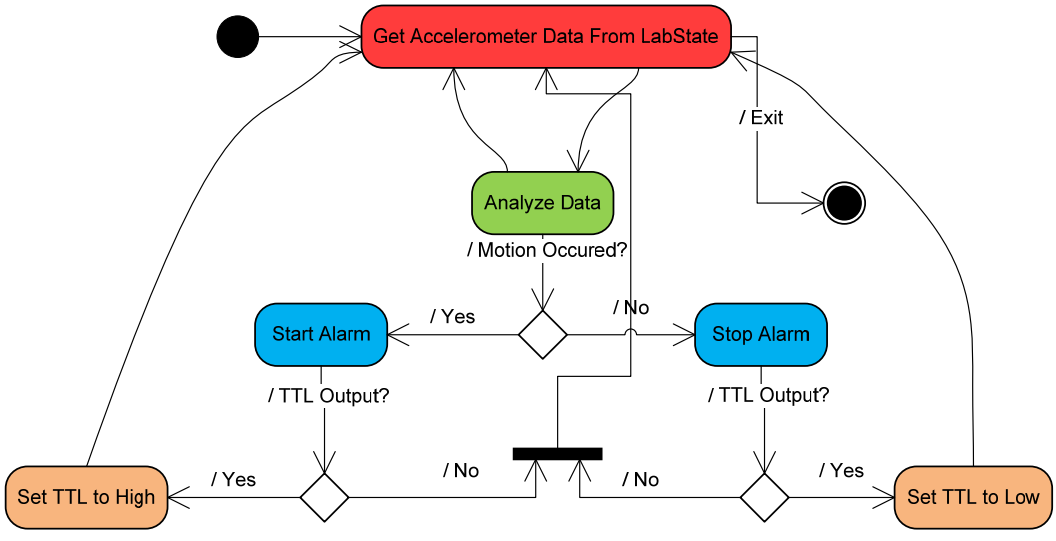

Figure 8-4: State chart of the Movement Detector function.

There is no data preview and record option. The function simply analyzes the data according to the algorithms mentioned above for StillState, and decides whether a motion event has occurred or not.

For StillState, there was a special DAQ card (USB 6008) to which the movement detector was connected using special connectors. However now it is connected using BNC cables because that's what LabState uses for analog channels. The new cabling along with the movement detector headband is shown in **Figure 8-5**. When the function starts execution, it automatically powers the movement detector hardware using one of the analog outputs of the DAQ card. The function also continuously returns a TTL signal to LabState which is either low or high depending on whether motion occurred or not.

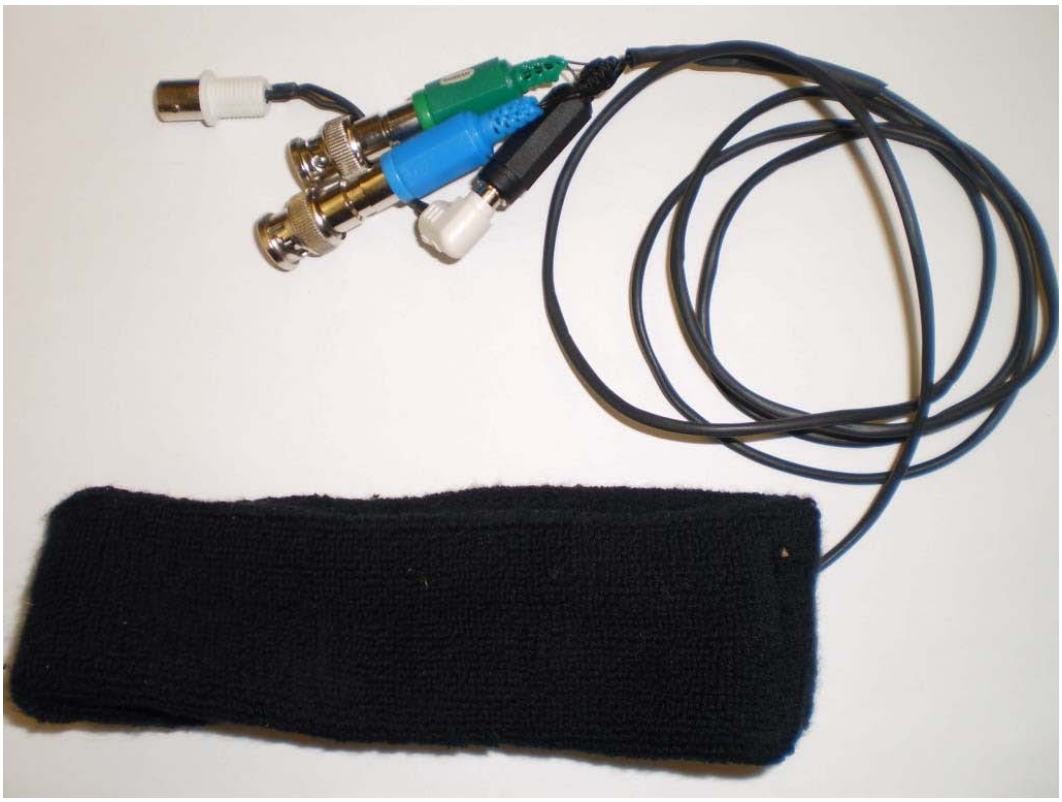

Figure 8-5: Image of BNC-based movement detector system.

Chapter 9 Testing and Results

Once LabState software was written and all the hardware and sensors were in order, it was thoroughly tested to make sure there were no errors that compromised the experiments and the safety of the subject. In this thesis, only parts of the results are shown because a complete display of results for all the tests will take up too much space due to the amount of images needed.

9.1 User Interface Test

The user interface was tested in details by testing each option in the program, changing settings systematically and making sure that the settings were reflected in the results. Each setting in the preferences including data storage path, display fonts, and acquisition time was modified. Channels were added, deleted, and modified to make sure the changes were reflected on the display and also in the recorded data. Multiple modules and functions were systematically used and unused and these changes were reflected in the displayed and recorded data.
The scaling of individual channels was tested by changing the scaling table of a channel with known input signal. Since the input of the channel was known, the acquired data was checked to make sure it was scaled according to the table.
Finally, the ability of the program to add comments was tested by randomly adding text comments during a recording session and checking the data file offline to see if those comments were added properly with valid timestamps. Similarly variable values were updated randomly and the results were seen in the recorded data file. **Figure 9-1** shows part of the resulting data file with comments and variables added to it. It also shows how LabSate updates the values of variable *Percent Isoflurane* whenever the user wanted to change it.

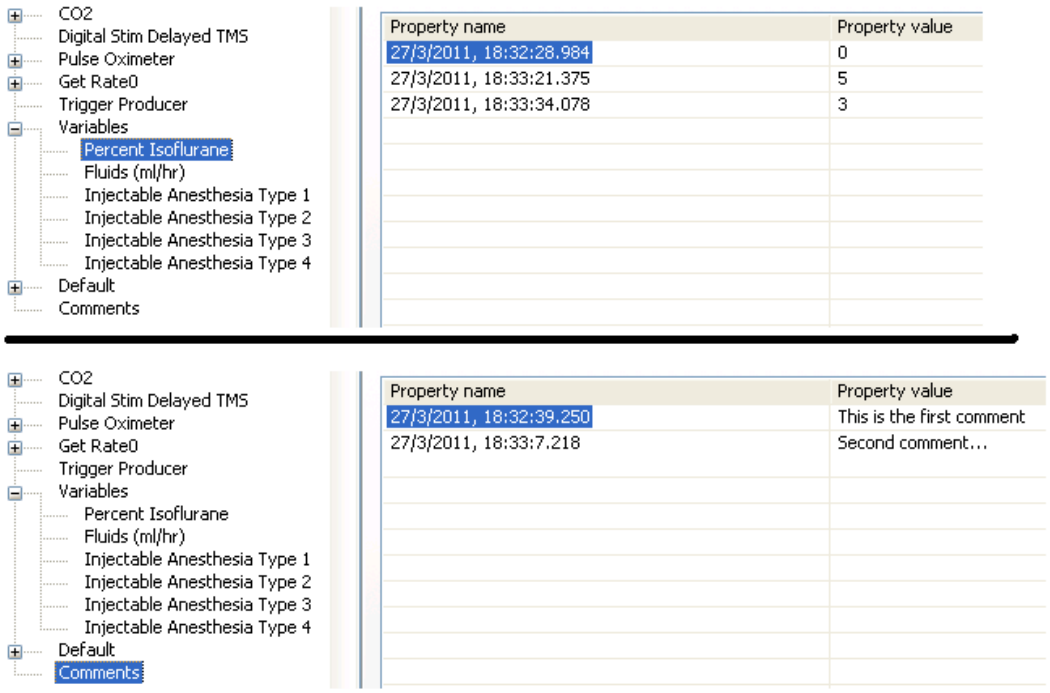

Figure 9-1: Comments and variables shown in the recorded data file.

9.2 Analog and Digital Inputs

Analog and digital signals from various sources were connected to the DAQ card for acquiring and recording in LabState while at the same time an oscilloscope was used to make sure the voltage levels and signal patterns matched. Analog signals included EEG, ECG, and respiration pulse from the pressure transducer. Digital signals included frame toggles from optical imaging system, custom generated triggers, electric and TMS stimulation waveforms. It was confirmed that LabState recorded all these signals properly. It also showed that correlated digital inputs worked because the digital signals were recorded at a certain sampling rates.

Maximum number of analog and digital signals were also recorded and displayed to the user for preview, as per LabState requirement. **Figure 9-2** shows an image of live display of data which includes 16 analog and 8 digital channels. Since only 16 plots are supported by LabState, some of the digital (D4-7) and analog channels (A12,15) are displayed as labels. Data of the labelled analog channels is also scaled to avoid displaying long decimal values that are in the original data.

Figure 9-3 shows content of the recorded data file. All 16 analog and 8 digital channels are properly recorded and the data values for some of them can be seen in the figure. All the analog and digital channels were recorded at 5 kS/s except A1 and D1-D3 which were recorded at 25 kS/s.

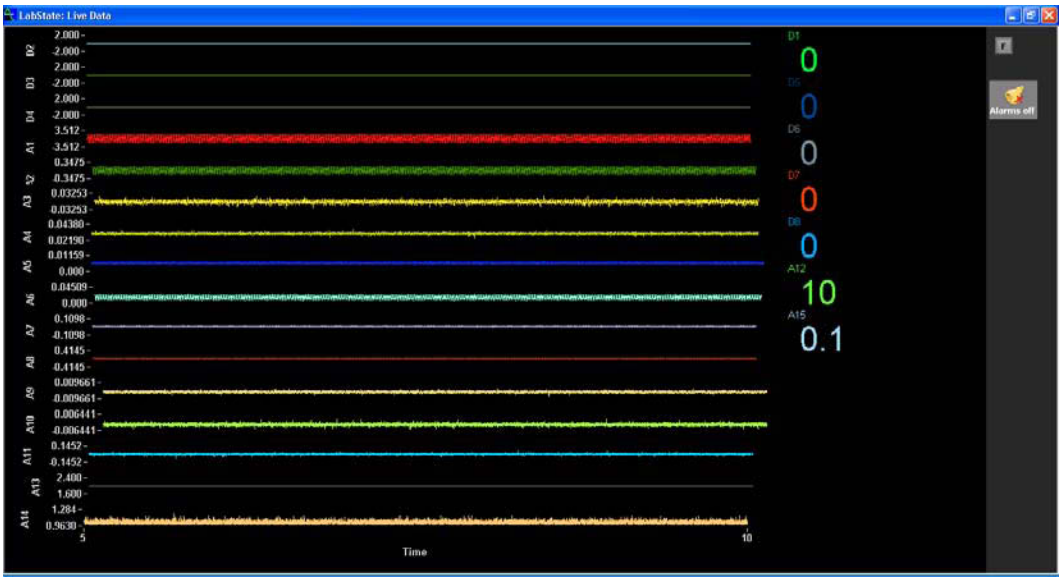

Figure 9-2: Data preview of 16 analog and 8 digital channels.

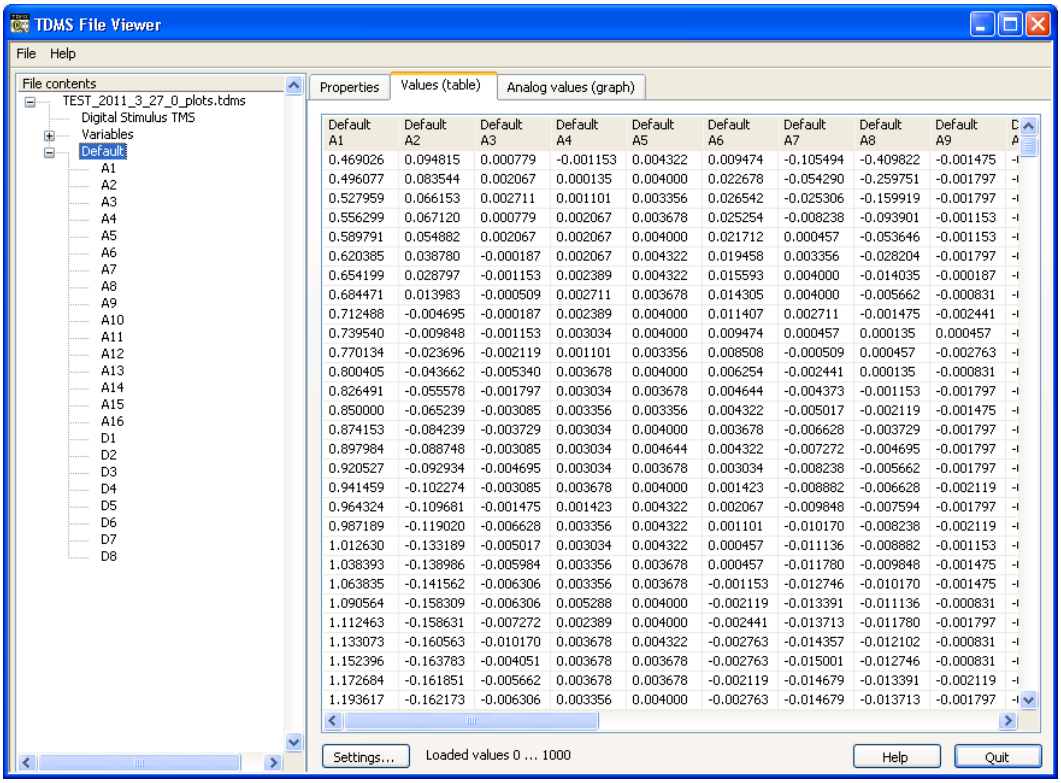

Figure 9-3: Recorded data file showing all the analog and digital channels.

9.3 Modules and Sensors

All the RS232-based sensors including pulse oximeter and CO_2 monitor did not have any analog outputs so an oscilloscope could not be used to verify their output. Each sensor came with its own software which was used to acquire data from the sensor. Immediately, LabState was used to acquire the data to make sure it matched the data obtained with the original software. For all sensors, the data was similar so it was confirmed that LabState acquired all the data properly.

The electric stimulation module was tested with multiple parameters to make sure the output matched the parameters. The outputted TTL pulses were also recorded in LabState and analyzed later in MATLAB to ensure the generated pattern matched the expected pattern. Similarly, the TMS stimulator's output waveform was analyzed the same way to ensure correct pulses were produced. **Figure 9-4** shows module settings dialog box used to set parameters for electric and TMS stimulation. The parameters shown were used to create stimulation and the results are shown in **Figure 9-5**. The triggers upon which the stimulation pulses are based are shown in green. TMS pulses are produced at 15Hz while the electric stimulation pulses are produced at 1Hz. Both pulses have 1 ms pulse duration.

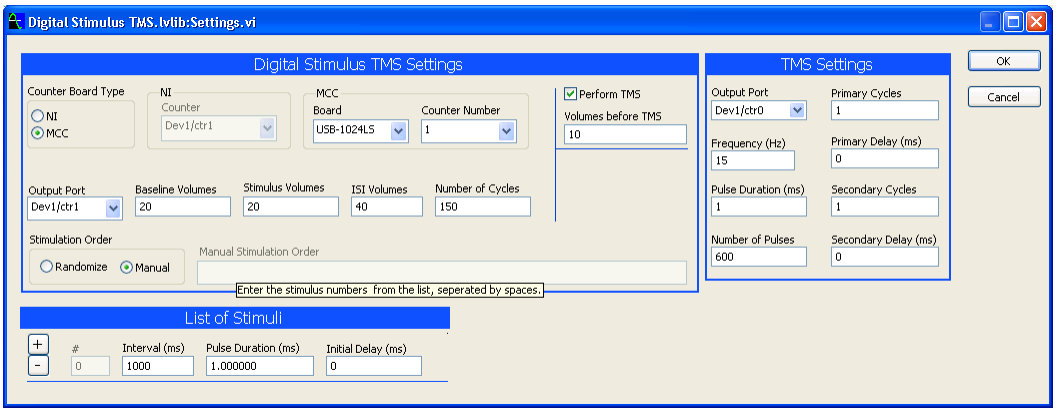

Figure 9-4: Module settings used for electric and TMS stimulation.

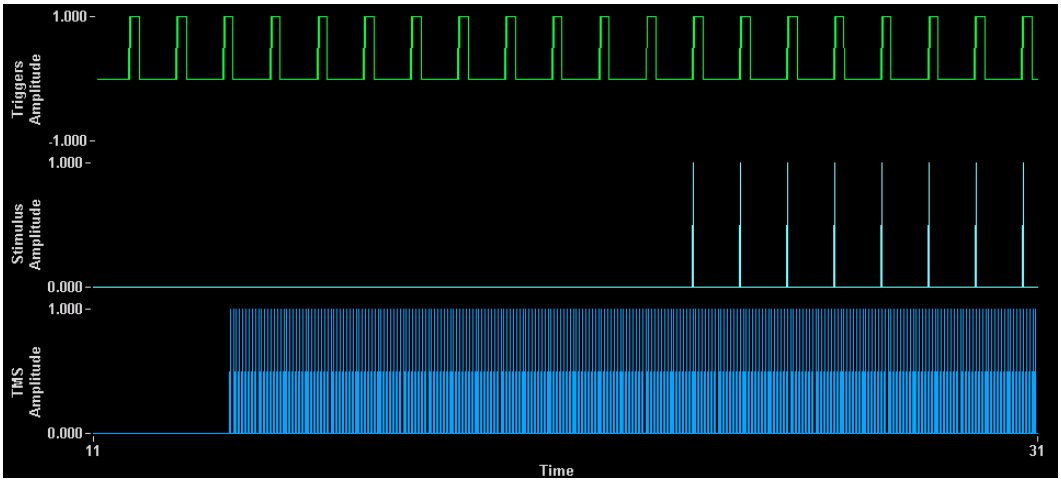

Figure 9-5: Electric stimulation and TMS pulses generated using above settings.

9.4 Alarms and Voice Alerts Testing

Alarms were thoroughly tested by using a custom generated waveform as input channel for each alarm. This way we could control the exact value of the channel. Different ranges from the alarm were set and it was found that the system only sounded the alarm when the input channel had data outside the entered range. Multiple alarms with multiple ranges were also set and it was observed that even if only one of the alarms detected data outside its respective range, the alarm sounded.

Voice alerts were also tested with custom generated waveform and the software successfully alerted the user of the correct data values.

Figure 9-6 shows a snapshot of the main LabState window while an experiment was being performed on a rat. The application was in record mode and it displayed several physiological signals as plots and labels (on the right side). The plot and data labels in yellow correspond to data from the pulse oximeter; plot and data labels in blue correspond to data from the CO_2 monitor; and the plot in white corresponds to data from the pressure transducer.

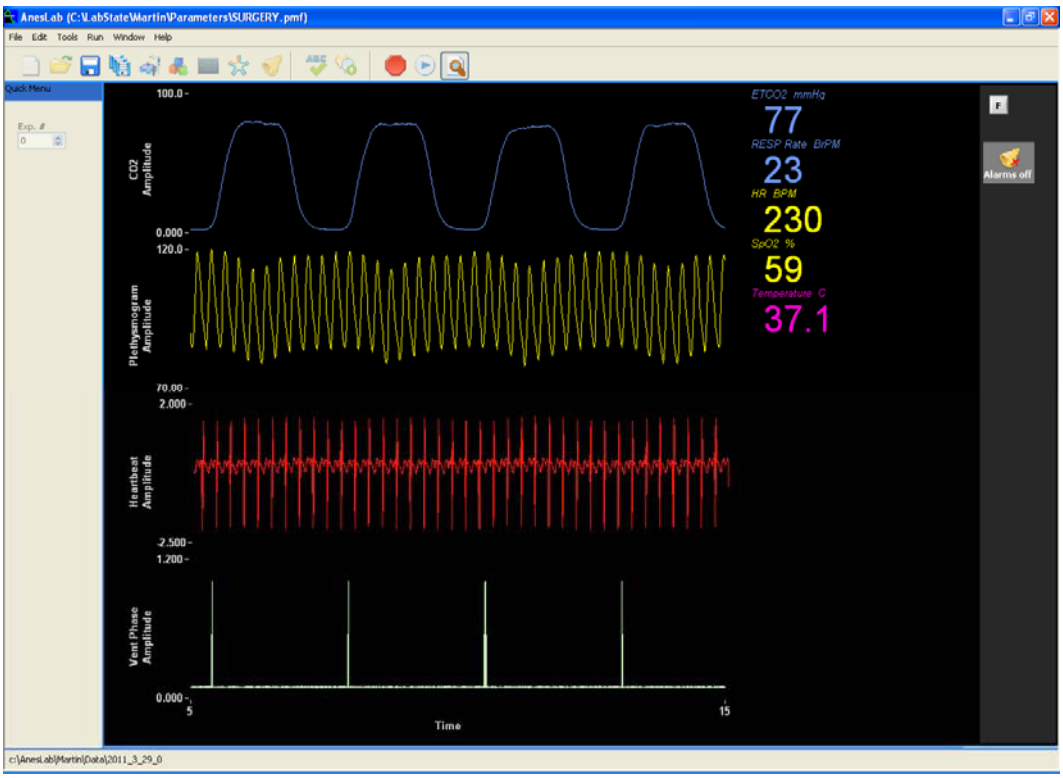

Figure 9-6: Snapshot of LabState window

9.5 StillState Testing

All the functionality of StillState was tested to make sure there were no bugs. The following testing procedure was performed:

- Start the program and click on *Preview* to view the accelerometer data live on the screen
- Set *Autoscale* to true and click on *Calibrate*
- After about 10 seconds, click on Calibrate again to stop calibration. Set *Autoscale* to off to change *SD Factor* to 50. This changes the scales of both plots to show white horizontal lines which represent the minimum and maximum threshold lines.
- Click on *Standby*. Add a valid path under *Data File Path* and a name under *Data File Name*. Set *Enable Alarm* and *Pulse on Movement* to true. Choose a valid output port from the list. Connect this port to an oscilloscope. Then click on *Record*. All the acquired data is recorded to a file now.

- Move the sensor and it will be reflected on one or both plots. If the acquired signal level goes beyond the threshold lines, an alarm is sounded until the signal level falls within the range.
- Movement also causes a TTL pulse to be produced on the output port which can be seen on the oscilloscope.
- When *Enable Alarm* is turned on or off then it turns the alarm sound on or off.
- When *SD Threshold* value is modified then the changed value is reflected in shifted threshold lines. A larger value means more movement needs to be performed to create the alarm.
- When *Autoscale* is turned on or off then it changes the scaling of both plots.
- When *Time Range (s)* is changed, the plots' time axis range changes with it.

A snapshot of the main StillState window is shown in **Figure 9-7**. It is recording from the accelerometer system and the data is also shown on the screen. The calibration has been performed, which provides the two white threshold lines on each plot. When the signal shows values beyond the lines, a motion event is assumed to occur and an alarm starts sounding (if enabled) and a high TTL pulse is sent (if enabled). As one can see, when the subject moves the data deviates from the accepted range (crosses the threshold line) and an alarm sounds until the movement stops.

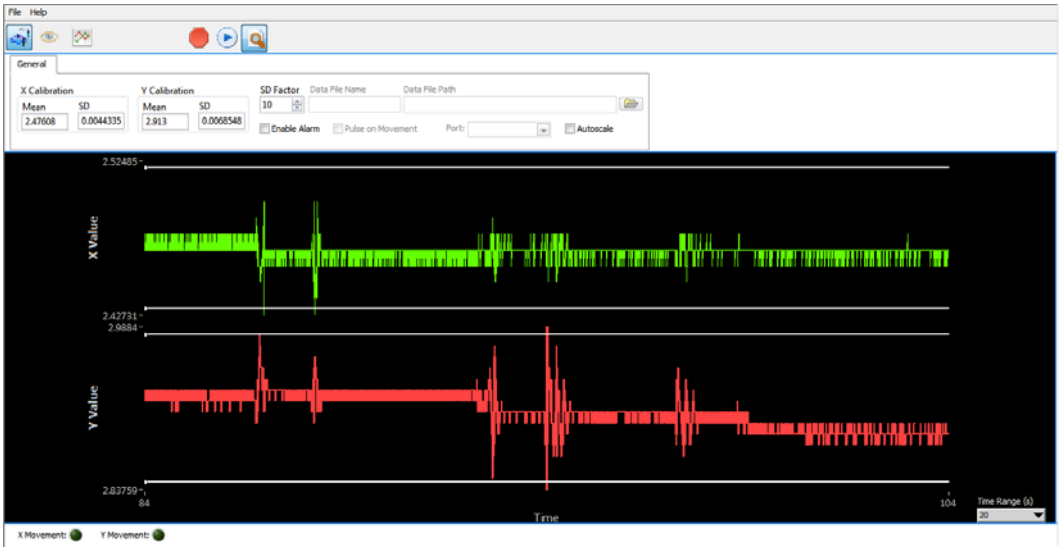

Figure 9-7: Snapshot of StillState window

Chapter 10 Summary and Future Work

10.1 Summary

This thesis describes the design and implementation of a combined patient monitoring and data-acquisition system, and a stand-alone body movement detection system. The patient monitoring and data-acquisition system is named LabState and it is being used in our lab during experiments to acquire physiological data and stimulate the subjects. There are several units of code in LabState and each unit runs in parallel with the others. This makes the overall execution faster. Data between units and sometimes within a unit is transmitted using queues. Queues are ideal in cases in which data loss cannot be tolerated but temporary data buffering is tolerable. LabState supports the addition of user-written modules for adding new and custom devices to the system. The devices can either acquire data, output data, or both. The synchronization of data acquired via the main DAQ card (analog and digital) with data from the modules is accomplished offline using timestamps from the events log created by LabState. The hardware design for some of the modules such as juice reward controller and rotatory stimulation device was also presented.

There is also support for user-written functions so data can be continuously analyzed in real-time and the output recorded or displayed during acquisition as well. One such function is called Get Rate which can calculate the fundamental frequency in a signal. It is used to calculate the heart rate from ECG and respiration rate from a respiration pad. Three different algorithms were written and tested, and the results were shown. Spike detection using spike thresholding was shown to be good for heart rate calculations but its parameters had to be changed for each experiment to make it work with the new ECG signal. This algorithm was not optimal for respiration rate calculations because the respiration signal had some kind of trend in it which gradually changed the maximum signal amplitude over the course of an experiment. This change required change in the threshold value of peak detection function. Out of the other methods: FFT peak

detection, and FFT STD thresholding, the FFT peak detection method was not accurate because it produced double or triple the real values.
The design of body movement detection hardware using accelerometer was shown along with the resulting final system. The design of the software to perform movement detection was presented along with the final application program. The system needs to be calibrated at the beginning of a session while the subject is not moving. It sounds an alarm if motion occurs and can also send a TTL pulse out to other systems.

10.2 Future Work

Both LabState and StillState are complete applications in the sense that they perform the initial requirements. However, improvements can be made to both systems in the future. The alarms and alerts functionality of LabState can be modified to provide volume control and multiple tones for multiple channels. Currently there is only one tone for all the channels. A module for estimating online anaesthesia regime can be implemented which might require additions in the core LabState software to make this combination more user-friendly. Currently, if the data is being acquired faster than it is being displayed or recorded, then a buffer overflow occurs which gives a warning to the user to close other applications that might be hogging the computer, reduce channels' sampling rate, or increase the "chunk size" for the data that the system acquires and records continuously. In the future, a process can be added to LabState which will pursue automated changes to the chunk size so the buffer never overflows.
StillState can also be enhanced by adding algorithms to calculate the movement distance in both axes based on the acceleration data. This can give an approximate distance the subject moves. Usually in the mock MRI scanner the subject is lying down so head movement will only cause noticeable movement in two of the axes so this feature can be useful without the use of a three axes accelerometer.
The entire StillState system needs to be tested for subject safety so it can be used with human subjects.

In addition to an accelerometer, a camera system can be used to monitor the subject's body movement. The use of both systems in combination will provide a better sensitivity.

References

1. Lisle, T.C., et al., *Inflammatory lung injury after cardiopulmonary bypass is attenuated by adenosine A(2A) receptor activation.* Journal of Thoracic and Cardiovascular Surgery, 2008. **136**(5): p. 1280-1288.
2. Sun, L., J. Schian, and N.B. Smith, *Novel adaptive control system for ultrasound hyperthermia treatment of prostate disease.* 2003 Ieee Ultrasonics Symposium Proceedings, Vols 1 and 2, 2003: p. 1274-1277.
3. Zhao, M., et al., *Spatiotemporal dynamics of perfusion and oximetry during ictal discharges in the rat neocortex.* J Neurosci, 2009. **29**(9): p. 2814-23.
4. Lubbers, W.C., et al., *Vacuum-assisted venous drainage during fetal cardiopulmonary bypass.* ASAIO J, 2005. **51**(5): p. 644-8.
5. *Spike2*. 2011 [cited 2011 February 1]; Available from: http://www.ced.co.uk/pru.shtml?pow1401u.shtml.
6. Sabharwal, R., et al., *Effect of hypothermia on baroreflex control of heart rate and renal sympathetic nerve activity in anaesthetized rats.* Journal of Physiology-London, 2004. **557**(1): p. 247-259.
7. Man, K., et al., *FK 409 ameliorates small-for-size liver graft injury by attenuation of portal hypertension and down-regulation of Egr-1 pathway.* Annals of Surgery, 2004. **240**(1): p. 159-168.
8. Man, K., et al., *Liver transplantation in rats using small-for-size grafts - A study of hemodynamic and morphological changes.* Archives of Surgery, 2001. **136**(3): p. 280-285.
9. Van Erp, C., N.G. Irwin, and A.J. Hoey, *Long-term administration of pirfenidone improves cardiac function in mdx mice.* Muscle & Nerve, 2006. **34**(3): p. 327-334.
10. Palmieri, R.M., et al., *Arthrogenic muscle response to a simulated ankle joint effusion.* Br J Sports Med, 2004. **38**(1): p. 26-30.
11. Lukasik, V.M., et al., *Cardiopulmonary effects of propofol anesthesia in chickens (Gallus gallus domesticus).* Journal of Avian Medicine and Surgery, 1997. **11**(2): p. 93-97.
12. Hopkins, J.T., et al., *Effect of knee joint effusion on quadriceps and soleus motoneuron pool excitability.* Med Sci Sports Exerc, 2001. **33**(1): p. 123-6.
13. Palmieri, R.M., M.A. Hoffman, and C.D. Ingersoll, *Intersession reliability for H-reflex measurements arising from the soleus, peroneal, and tibialis anterior musculature.* Int J Neurosci, 2002. **112**(7): p. 841-50.
14. Leker, R.R., et al., *NAP, a femtomolar-acting peptide, protects the brain against ischemic injury by reducing apoptotic death.* Stroke, 2002. **33**(4): p. 1085-92.
15. *MRI Solutions for Human and Animal Studies*. 2011 [cited 2011 February 1]; Available from: http://www.biopac.com/ProductImages/corporate%20images/biopac_mri_catalog_web.pdf.

16. Oakes, T.R., et al., *Comparison of fMRI motion correction software tools.* Neuroimage, 2005. **28**(3): p. 529-43.
17. *MoTrak™*. 2010 [cited 2010 October 2]; Available from: http://www.pstnet.com/software.cfm?ID=96.
18. *Respironics Novametrix Tidal Wave 710Sp Monitor*. 2011 [cited 2011 February 1]; Available from: http://www.dremed.com/catalog/product_info.php/products_id/406.
19. *DRE Waveline Pro Monitor*. 2011 [cited 2011 February 1]; Available from: http://www.dremed.com/catalog/product_info.php/products_id/1821.
20. *4Vet Portable Multiparameter Veterinary Monitor*. 2011 [cited 2011 February 1]; Available from: http://www.dreveterinary.com/product_info.php/cPath/457_250_402/products_id/1307#documents.
21. *Advisor® Vital Signs Monitor*. 2011 [cited 2011 February 1]; Available from: http://www.surgivet.com/upload/products/product_relateddocs/Advisor%20Op%20Man.pdf.
22. *PC-VetGuard+ Monitor*. 2011 [cited 2011 February 1]; Available from: http://www.vmedtech.com/PC-VetGard+.htm.
23. *Data Acquisition Systems - PowerLab*. 2011 [cited 2011 February 1]; Available from: http://www.adinstruments.com/products/data*acquisition/.
24. *PowerLab /30 Series: Owner's Guide*. 2011 [cited 2011 February 1]; Available from: http://www.adinstruments.com/products/manuals/PowerLab_30_Series_OG.pdf.
25. *LabChart 7.2*. 2011 [cited 2011 February 1]; Available from: http://www.adinstruments.com/products/software/research/LabChart-Software/.
26. *MP Systems Hardware Guide*. 2011 [cited 2011 February 1]; Available from: http://www.biopac.com/Manuals/mp_hardware_guide.pdf.
27. *AcqKnowledge Software Guide*. 2011 [cited 2011 February 1]; Available from: https://www.biopac.com/Manuals/acqknowledge_software_guide.pdf.
28. *BIOPAC BASIC SCRIPTING LICENSE FOR ACQKNOWLEDGE*. 2011 [cited 2011 February 1]; Available from: https://www.biopac.com/BIOPAC-Scripting-AcqKnowledge-Windows-knowledge-base#LowerTab.
29. *BHAPI BIOPAC Hardware API*. 2011 [cited 2011 February 1]; Available from: http://www.biopac.com/Manuals/app_pdf/app218.pdf.
30. *Isolated Power Supply Module -IPS100C*. 2011 [cited 2011 February 1]; Available from: http://www.biopac.com/isolated-power-supply-module.
31. Rijnbeek, P.R., J.A. Kors, and M. Witsenburg, *Minimum bandwidth requirements for recording of pediatric electrocardiograms.* Circulation, 2001. **104**(25): p. 3087-90.

32. Vanhatalo, S., J. Voipio, and K. Kaila, *Full-band EEG (FbEEG): an emerging standard in electroencephalography.* Clin Neurophysiol, 2005. **116**(1): p. 1-8.
33. Landau, H.J., *Sampling, data transmission, and the Nyquist rate.* Proceedings of the IEEE, 1967. **55**(10): p. 1701-1706.
34. Young, S., *Computerized Data Acquisition and Analysis for the Life Sciences: A Hands-on Guide*. 2001.
35. *BNC-2090A User Manual*. 2010 [cited 2010 October 6]; Available from: http://www.ni.com/pdf/manuals/372101a.pdf.
36. *NI USB-6221 BNC*. 2010 [cited 2010 October 6]; Available from: http://sine.ni.com/nips/cds/view/p/lang/en/nid/203867.
37. Graf, R.F., *Modern dictionary of electronics*. 7th ed. 1999, Boston: Newnes. 869 p.
38. *Counter/Timer*. 2010 [cited 2010 October 4]; Available from: http://directory.adeptscience.co.uk/products/search/1/1/1016/cat1016.html.
39. *Subdermal/Scalp EEG Needles, Corkscrew & Stimulating Probes*. 2011 [cited 2011 February 1]; Available from: http://kingmedical.com/Subdermal.htm.
40. *A365R High Voltage Isolator, Bipolar Rechargeable*. 2011 [cited 2011 February 1]; Available from: http://www.wpiinc.com/index.php/SYS-A365R.html.
41. (2011) *Magstim Rapid Operator's Manual*. **2011**.
42. *Application Design Patterns: Producer/Consumer*. 2006 [cited 2010 October 1]; Available from: http://zone.ni.com/devzone/cda/tut/p/id/3023.
43. Chakravarty, M.M., et al., *Design, construction, and validation of an MRI-compatible vibrotactile stimulator intended for clinical use.* J Neurosci Methods, 2009. **184**(1): p. 129-35.
44. Huang, Y.Z., et al., *Theta burst stimulation of the human motor cortex.* Neuron, 2005. **45**(2): p. 201-6.
45. *Serial Motor Driver User Guide*. 2009 [cited 2010 October 7]; Available from: http://www.sparkfun.com/datasheets/Robotics/SFE03-0012-UserGuide-ROB-09571-serialmotordriver.pdf.
46. Coleman, K., et al., *Training rhesus macaques for venipuncture using positive reinforcement techniques: A comparison with chimpanzees.* Journal of the American Association for Laboratory Animal Science, 2008. **47**(1): p. 37-41.
47. *EIGHT DARLINGTON ARRAYS*. 2003 [cited 2010 October 2]; Available from: http://www.st.com/stonline/products/literature/ds/1536/uln2803a.pdf.
48. Christov, II, *Real time electrocardiogram QRS detection using combined adaptive threshold.* Biomed Eng Online, 2004. **3**(1): p. 28.
49. Friesen, G.M., et al., *A comparison of the noise sensitivity of nine QRS detection algorithms.* IEEE Trans Biomed Eng, 1990. **37**(1): p. 85-98.
50. Kohler, B.U., C. Hennig, and R. Orglmeister, *The principles of software QRS detection.* IEEE Eng Med Biol Mag, 2002. **21**(1): p. 42-57.

51. Ligtenberg, A. and M. Kunt, *A robust-digital QRS-detection algorithm for arrhythmia monitoring.* Comput Biomed Res, 1983. **16**(3): p. 273-86.
52. Pan, J. and W.J. Tompkins, *A real-time QRS detection algorithm.* IEEE Trans Biomed Eng, 1985. **32**(3): p. 230-6.
53. Du, P., W.A. Kibbe, and S.M. Lin, *Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching.* Bioinformatics, 2006. **22**(17): p. 2059-2065.
54. Jarman, K.H., et al., *A new approach to automated peak detection.* Chemometrics and Intelligent Laboratory Systems, 2003. **69**(1-2): p. 61-76.
55. Nenadic, Z. and J.W. Burdick, *Spike detection using the continuous wavelet transform.* Ieee Transactions on Biomedical Engineering, 2005. **52**(1): p. 74-87.
56. Brigham, E.O. and R.E. Morrow, *Fast Fourier Transform - a Detailed Explanation Is Offered of an Algorithm That Reduces Computer Time and Allows Its User to Employ Powerful Frequency Techniques Once Condidered Inefficient.* Ieee Spectrum, 1967. **4**(12): p. 63-&.
57. Cochran, W.T., et al., *What Is Fast Fourier Transform.* Proceedings of the Institute of Electrical and Electronics Engineers, 1967. **55**(10): p. 1664-&.
58. *ADXL103/ADXL203 Datasheet Rev C.* 2010 [cited 2010 October 4]; Available from: http://www.analog.com/static/imported-files/data_sheets/ADXL103_203.pdf.