Accomodarsi Solutions

School of Engineering Science
Simon Fraser University
Burnaby, BC  V5A 1S6
accomodarsi@gmail.com

October 28, 2006

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: Post-Mortem Report for an Auto-Conforming Ergonomic Chair

Dear Dr. Rawicz:

The attached document, *Post-Mortem Report for the Auto-Conforming Ergonomic Chair*, outlines the process our team went through when designing and implementing our project for ENSC 340. Our objective was to build a prototype task chair that would sense the dimensions of each user and automatically adjust to provide optimal support and comfort.

This document details the current state of the ACE Chair, describes deviations from the original design, and suggests future improvements. We also analyze the budgetary and time constraints encountered and reflect individually on our interpersonal experiences and the technical knowledge we have gained.

Accomodarsi Solutions is comprised of highly motivated, innovative, and enthusiastic students who study Engineering Science at SFU: Jennard Dy, Stephanie Fung, Eric Lee and Eric Leung. If you have any questions or concerns, we would be glad to address them. The team can be reached by email at accomodarsi@gmail.com.

Sincerely,

Stephanie Fung

Stephanie Fung
Chief Executive Officer
Accomodarsi Solutions

Enclosure: Post-Mortem Report for the Auto-Conforming Ergonomic Chair

Accomodarsi Solutions

# Post-Mortem Report for the Auto-Conforming Ergonomic Chair

**Project Team:** Jennard Dy
Stephanie Fung
Eric Lee
Eric Leung

**Contact Person:** Jennard Dy
accomodarsi@gmail.com

**Submitted to:** Dr. Andrew Rawicz – ENSC 340
Mike Sjoerdsma – ENSC 305
Brad Oldham

School of Engineering Science
Simon Fraser University

**Date:** October 28, 2006

**Revision:** 1.0

# Table of Contents

## List of Figures and Tables

## Glossary

| | |
|---|---|
| **ADC** | Analog-to-Digital Converter. An electronic component that converts an analog signal to a digital signal for input to a digital system. |
| **ATX power supply** | Advanced Technology Extended power supply. A standardized computer power supply commonly used in today's personal computers. |
| **EEPROM** | Electrically Erasable Programmable Read Only Memory. A type of non-volatile memory that can typically last 100,000 erase-write cycles. Data is retained even when powered off. It is commonly found on many microcontrollers. |
| **Ergonomic** | Exhibiting good design so as to maximize productivity by reducing fatigue and discomfort [1]. |
| **GPIO** | General Purpose Input/Output. Set of input/output ports on an embedded system (i.e. microcontroller) that can be either set to being an input or output. |
| **H-bridge** | An electronic circuit that allows a DC motor to run in forward and reverse direction without swapping wires. |
| **JTAG ICE** | Joint Test Action Group In-Circuit Emulator. JTAG: standard for testing sub-blocks of an integrated circuit. The combination of JTAG and ICE can be used as a debugging tool for embedded systems. |
| **LCD** | Liquid Crystal Display. An electronic component that is commonly used to display text and numbers in embedded systems. |
| **LED** | Light-emitting diode. An electronic component which emits light when a voltage is applied. Often used as a status indicator on electronic equipment. |
| **MOSFET** | Metal Oxide Semiconductor Field Effect Transistor. An electronic component that acts as a voltage-controlled switch. Desirable characteristics include the capability to handle high current and fast switching times. |
| **PCB** | Printed circuit board. A thin, layered board on which electronic components are mounted and interconnected by copper traces on the various layers. |

| | |
|---|---|
| **PWM** | Pulse-Width Modulation (Modulated). A modulation technique utilizing changes in a signal's duty cycle to convey information over a communications channel. |
| **SolidWorks** | Software used to create and model 3-D representations of objects. |
| **SPI** | Serial peripheral interface. A synchronous serial data link that commonly connects a microcontroller to external peripherals, such as an LCD. |
| **TQFP** | Thin Quad Flat Pack. A type of compact surface mount packaging for integrated circuits with a high pin-count. |
| **USART** | Universal synchronous asynchronous receiver transmitter. A piece of electronic hardware that translates between parallel and serial transmission of data bits. |

# 1 Introduction

For the past thirteen months, Accomodarsi Solutions has been actively designing and implementing a proof-of-concept prototype of the proposed Auto-Conforming Ergonomic (ACE) Chair. Intended to decrease user error in adjustment of current ergonomic chairs and to improve the fit of current ergonomic chairs, the ACE Chair is designed to enhance user comfort and reduce the risk of musculoskeletal disorders.

The main feature of the ACE Chair is the hassle-free, one-touch adjustment. The ACE Chair provides an Auto Mode where the ACE Chair automatically adjusts the various body supports to fit the user. The user can also fine-tune the adjustments with the simple and intuitive User Interface (UI) module.

In this document, we discuss the current state of the prototype, provide future plans and recommendations, and assess the development of the prototype from a project management standpoint. Furthermore, each team member from Accomodarsi Solutions offers insights and reflections on the inter-personal and technical experiences gained from collaborating on this project for the past thirteen months.

# 2 Current State of the Prototype

## *2.1 ACE Chair Features*

The proof-of-concept prototype ACE Chair is an office task chair that has been augmented with adjustable armrests and lumbar support. From the user's point of view, the prototype is a solid piece of furniture on which to sit. Once seated, the user can adjust the following dimensions: armrest height, lumbar support height, and lumbar support size. Via the User Interface (UI) module, the user is able to manually raise and lower the armrests, raise and lower the lumbar support, and increase or decrease the size of the lumbar support. The user receives status messages and feedback through the LCD and LEDs on the UI. Additionally, the user can initiate the automatic adjustment sequence by pressing the "mode button" on the UI.

The ACE Chair's primary innovation is the Auto Mode automatic adjustment sequence. Based on the output of various sensors, the ACE Chair automatically adjusts the lumbar and arm supports to best suit the user. In the prototype, the user simply sits down and moves the power switch to the on position. The user then sits up and rests their forearms at a comfortable height above the armrests. Once seated in the desired position, the user presses the mode button and the automatic adjustment sequence commences. The armrests adjust by first moving to their lowest position, and then moving up until contact is made with the user's arms. Next, the lumbar support adjusts by moving to the lowest position, and then moving up to the highest position while measuring the user's back. Based on the measurements, the lumbar support moves to the best position. Finally, the lumbar support inflates or deflates according to the desired pressure set by the user on the UI module.

## 2.2 How the ACE Chair Works

### 2.2.1 Mechanical

The foundation of the ACE Chair is its mechanical assemblies. The prototype is constructed on an existing office task chair platform with a mechanically adjustable seat height, footrest height, and backrest tilt. The original office task chair did not have armrests. An automatically adjusting seat height is not a requirement, thus the original pneumatic seat height and footrest height mechanisms are used. However, because of space limitations, the ACE Chair seat height has a higher minimum height than the original chair. The original backrest was replaced with a larger backrest with a movable lumbar support. A new mounting mechanism was required for the new backrest, resulting in a fixed-angle backrest.

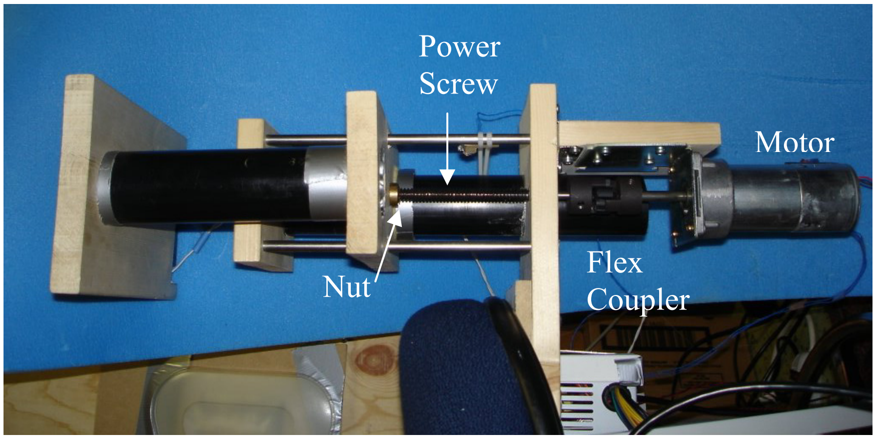


**Figure 1: Armrest Height Mechanism**

**(Oriented horizontally with top of the armrest on the left).**

The armrest height and lumbar height adjustments both require precise controlled linear motion over a fixed range of movement. To achieve this articulation, each adjustment uses a DC motor coupled to a power screw, which drives a nut attached to an appropriate solid object. The same principle is used for both the armrest and lumbar adjustments, but the adjustments were implemented differently.

The lumbar support is able to increase and decrease size through the use of pressurized air. The lumbar support consists of an ergonomically shaped inflatable back support. Two solenoid valves seal off the lumbar support to keep the amount of air within the lumbar support constant, to connect it to the air source for inflation, or to let air escape to the atmosphere for deflation. An air pressure regulator and flow regulator control the input from the pressurized air source. A safety pressure relief valve ensures that the lumbar support does not burst from an over-pressure situation.

### 2.2.2 Electronics

Figure 2 is a high-level depiction of the electronic system on the ACE Chair.

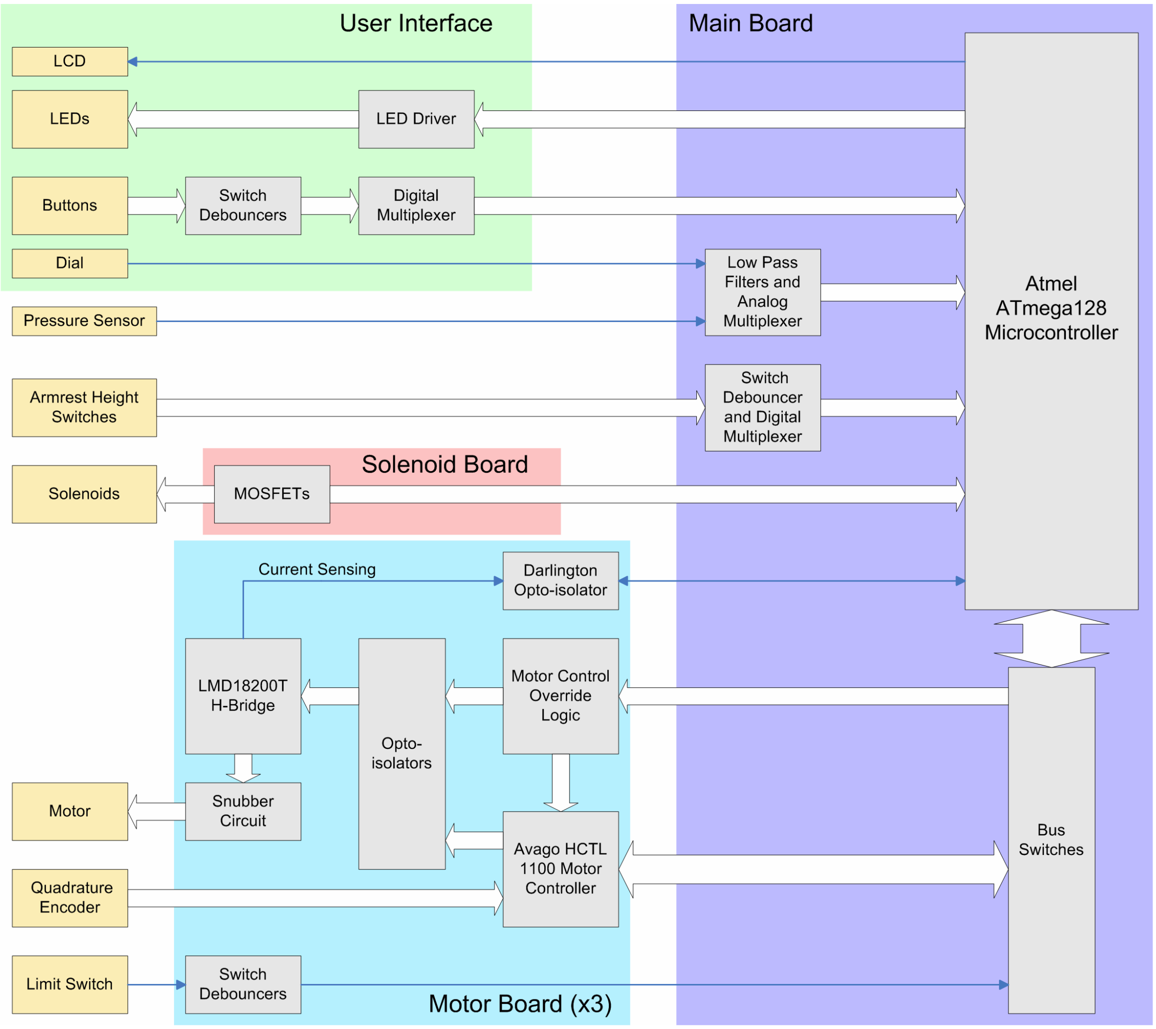


**Figure 2: Electronic System Block Diagram.**

The ACE Chair electronics link the software with the mechanical elements, providing the chair with power, control signals, and sensor data. The prototype's electronics are all constructed on through-hole vector boards. However, use of surface mount components was unavoidable and single-sided PCBs were constructed to adapt the surface mount components to the through-hole boards. The electronics on the prototype are centralized in a wooden enclosure placed underneath the seat. The enclosure contains six main vector boards: a main processing board, three motor boards, a solenoid driver board, and a UI board. In addition, there is a small vector board located near the top of the backrest that contains the pressure sensor, as well as a small vector board that distributes power to all the other boards. A "polyswitch," or self-resetting fuse, protects each board. Additionally, each motor board has a 2.5 amp circuit breaker on the 12-volt line powering the motor. The prototype's computing circuitry operates from a 5-volt linear regulated power supply located underneath the front of the seat. The motors and solenoids are powered by the 12-volt output of a standard ATX computer power supply located underneath the rear of the seat. The two power supplies are electrically isolated from each other and do not share a common ground. Both power supplies require a 115-volt 60 Hz AC power supply typical of North American wall outlets.

The main processing board contains a 64-pin 8-bit Atmel AVR microcontroller that directs operations on the ACE Chair. On the analog side, connected to the microcontroller are three current-sensing outputs from the motor boards and an analog multiplexer connected to the UI dial and the pressure sensor. On the digital side, the microcontroller is connected to every other board. The UI makes use of the USART and SPI busses to communicate with the LCD and LED driver, respectively. General-purpose input/output (GPIO) pins are used to communicate with the solenoid and motor boards, as well as to read the input from the UI buttons and the armrest height switches. Two multiplexing/de-multiplexing bus switches are used to connect the microcontroller to the three motor boards.

Each motor-powered mechanical adjustment (two armrest heights and one lumbar height) is controlled by a motor board. The motor board contains a 40-pin HCTL-1100 motor controller that performs low-level real-time feedback control of the motor. The motor controller outputs pulse-width modulated (PWM) and direction signals that drive an H-bridge connected to the motor terminals. Each motor contains a two-channel quadrature encoder that is connected to the motor controller to provide position feedback. A limit switch is present on each adjustment to provide a known reference home position. The H-bridge provides a current sensing output proportional to the current being driven through the motor. This current sensing output is fed back into the AVR microcontroller for safety purposes. The motor controller outputs and the H-bridge inputs are connected by optoisolators. The current sensing output is transmitted by a Darlington optoisolator to the AVR microcontroller. An 1800uF capacitor helps power the H-bridge and a resistor-capacitor snubber circuit is connected across the motor terminals for noise suppression.

The solenoid board contains driver and noise suppression circuitry to power the two solenoid valves that control the lumbar support size. The two control signals from the main processing board are the inputs to an optoisolator, with the outputs of the optoisolator connected to the gates of MOSFET drivers. The UI board contains an LCD module with a dedicated PIC microcontroller. The LCD module accepts high-level commands from the AVR microcontroller and displays the appropriate text on the LCD.

The UI board also contains an LED driver chip to control the output of 13 debugging LEDs and 3 status LEDs. On the input side, the UI board contains a switch debouncer and digital multiplexer for reading the user input buttons. A potentiometer and voltage divider form the dial electronics.

### 2.2.3 Software

The ACE Chair has four software modes: main, hardware self-test, motor remote control, and error log view. The main mode contains the software that normally interacts with the user. The hardware self-test mode was used to verify initial connectivity and functionality of the hardware during early development. While developing the prototype, it was extremely useful to be able to manually move the motors to any position; hence, the motor remote control mode was created. Lastly, the normal software mode contains an error logging facility to aid in debugging. The error log view mode allows offline browsing of the error log for error analysis. The software is written primarily in C with small amounts of inline assembly.

The main software mode uses the FreeRTOS kernel to provide a multi-tasking, pre-emptive context-switching, real-time operating system environment. Eight tasks execute concurrently to monitor the hardware for user-input and safety thresholds. Each automatic adjustment (armrest height, lumbar height, lumbar size) is performed by a task, resulting in three transient tasks. These tasks are activated when the user presses the mode button to enter automatic adjustment; the tasks are de-activated when the adjustment process is complete. In addition to the FreeRTOS tasks, the software contains interrupt service routines (ISRs) to provide interrupt-driven input/output operations for the hardware. Interrupts are used for the USART, SPI, EEPROM, and ADC.

The software contains three internal safety mechanisms: a watchdog timer, error logging, and a software interrupt. The watchdog timer is a hardware feature of the microcontroller that performs a hardware reset if the software enters an unknown state. We implemented an error logging facility to write messages to the EEPROM in case of error conditions. The most common error condition we encountered was overflow of internal queues. A software interrupt is provided to allow the software to quickly halt all operations and bring the hardware into a safe state. The software interrupt is activated if the pressure in the lumbar support is too high, or if the current in the motors is too large. Once activated, the software interrupt ISR deflates the lumbar support and stops all motors.

The AVR microcontroller does not directly control the motors. Instead, the software sends high-level commands to the HCTL-1100 motor controller on each motor board. The HCTL-1100 translates the high-level commands into actual PWM and direction signals to drive the motors. The high-level commands used in the prototype are: move to a position, move at a velocity, and move to a position using a velocity profile. In addition, the AVR microcontroller periodically issues a read position command. Each mechanical adjustment has one limit switch to mark the home reference position. Once the position is known, a software limit switch is implemented at the other extreme of the range of adjustment. By comparing the actual position with a hard-coded maximum position, the software ensures that the motors do not move out of range. If a motor position is not known, at the first attempt to move the motor, the motor will home to the limit switch, then return to its original position. Once homing is complete, the initial movement command is executed. To avoid unnecessary homing movements, the motor positions are stored in EEPROM when the on/off switch is turned off. The next time the ACE Chair is turned on, the motor positions are read back from the EEPROM and homing is not required.

### 2.2.4 System

Combining the mechanical, electronics, and software subsystems, the prototype ACE Chair is able to safely provide manual adjustment of each dimension, as well as an automatic adjustment sequence. A software power on/off switch is located on the UI. Pressing this switch turns the chair on and off, but does not remove power from the system. The ACE Chair is intended to be operated with power applied all the time. After the user presses the on/off switch, the ACE Chair displays a welcome message and is ready to accept user input.

It is an ergonomic requirement that the armrests are at the same height whenever possible. However, the armrests are individually driven by separate motors. Thus, to keep the armrests at the same height, the software performs an equalizing process. The armrest positions are read, and if they differ by too much, the higher armrest moves to the same height as the lower armrest.

If the user presses the up/down button, the ACE Chair performs a manual adjustment of the lumbar height or armrest height, depending on the position of the armrest/lumbar slider switch. The mechanism continues moving until the user releases the up/down button, the hardware limit switch is reached, or the software limit switch is reached. If the user adjusts the dial, the solenoids will open or close to inflate or deflate the lumbar support. The pressure sensor reads the pressure in the lumbar support to close the feedback loop. Once the actual pressure is within a threshold of the desired pressure, the solenoids will close to seal the lumbar support.

When the user presses the mode button to activate automatic adjustment, the software activates a task to adjust the armrest height. The armrests move down to the lowest position, hitting the limit switches. Then the armrests move up until a contact sensor located on the top of each armrest is pressed. Once the user's arm contacts the sensor, the armrests stop moving. If the armrests are at different heights, they equalize to the same height. After de-activating the armrest height task, the software activates the lumbar height task. The software moves the lumbar to the lowest position, increases the sampling rate of the pressure sensor, and closes the solenoids to seal the lumbar support. The lumbar then moves at a constant velocity toward the top. As the lumbar is moving, the pressure sensor is sampled and the smallest value is saved along with the position of the motor corresponding to the smallest sensor reading. When the lumbar support reaches the top, the motor is commanded to move to the saved position corresponding to the smallest sensor reading. It is assumed that the smallest sensor reading is taken when the lumbar support is at the point on the user's back with the deepest curvature. Lastly, the lumbar size task is activated to control the solenoids to inflate or deflate the lumbar support until the desired pressure is reached. Figure 3 shows a flowchart illustrating the automatic operation of the chair, while Figure 4 shows the manual operation of the chair.

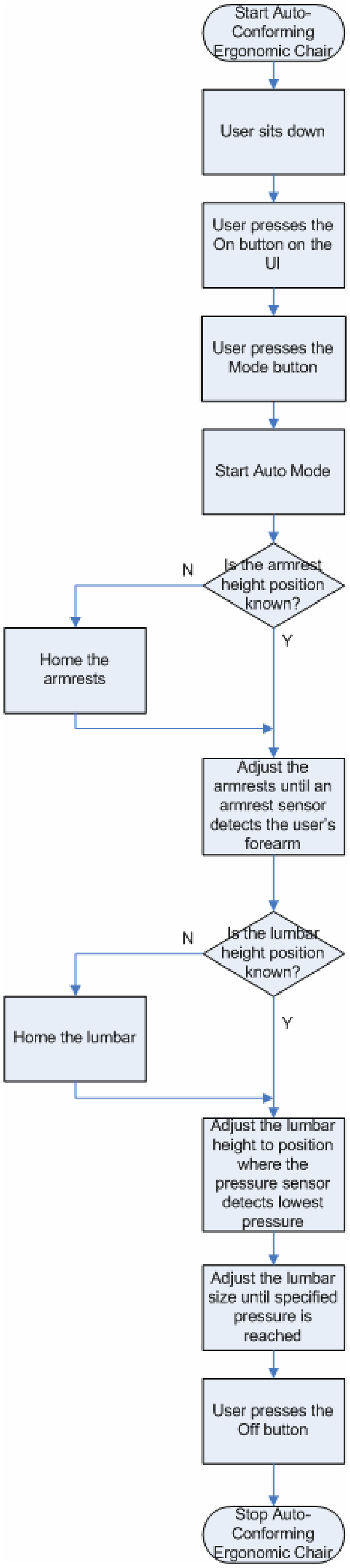


**Figure 3: System Operation Flowchart of the Auto Mode.**

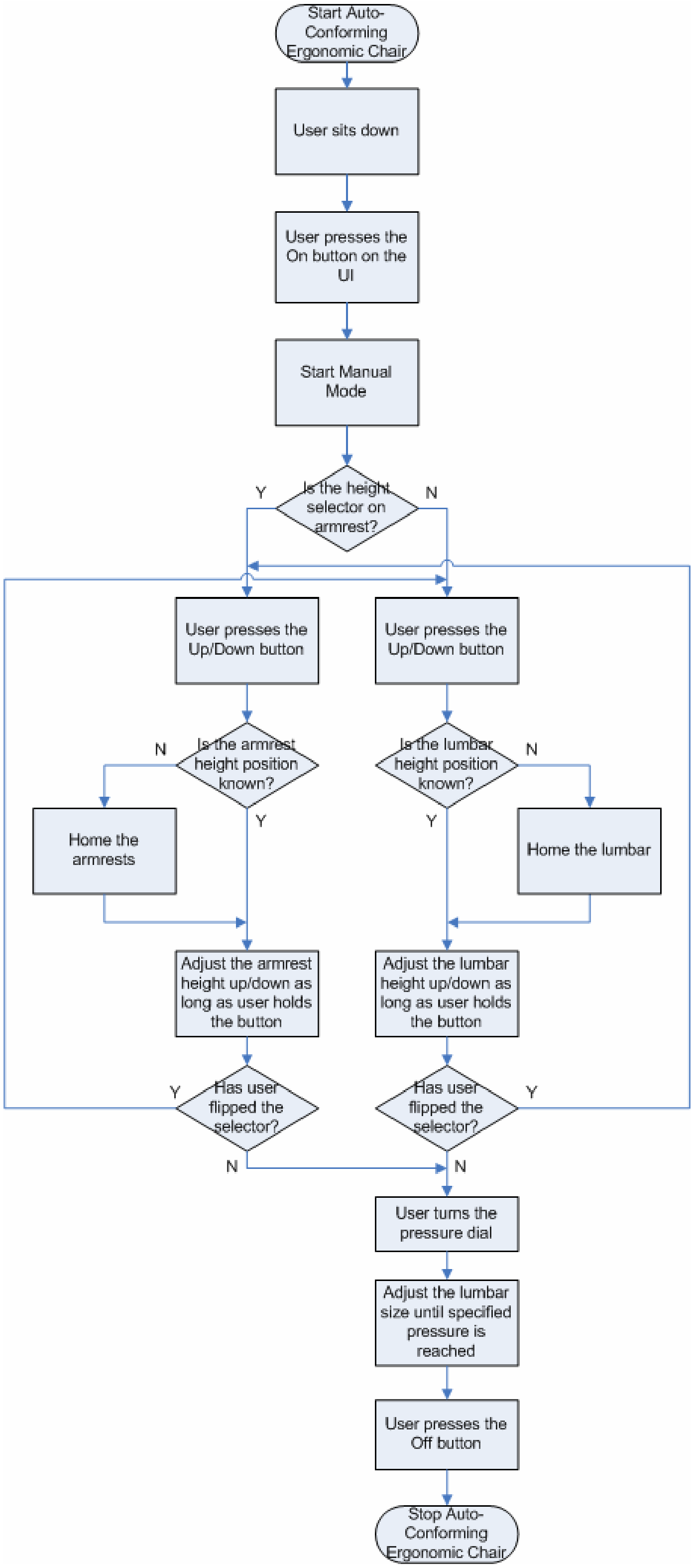


**Figure 4: System Operation Flowchart of the Manual Mode.**

## 2.3 Deviation From Original Design Specification

This section describes the differences between the prototype ACE Chair and the design specification [3].

### 2.3.1 Mechanical

The only difference between the prototype ACE Chair and the overall system specified are the armrest width and seat depth adjustments. Because of complexity, time, cost, and difficulty, it was decided early in the implementation stage to remove these adjustments from the prototype. From an ergonomic perspective, armrest width and seat depth are less critical adjustments.

From a high-level perspective, the ideas behind the specified mechanical design were carried through and implemented in the prototype. Due to the mechanical simplifications, the prototype has simplified sensing, removing the load sensors, distance sensors, presence sensors and flow sensor. After removing the armrest width and seat depth adjustments, the implemented actuators closely match the specified design.

#### 2.3.1.1 Armrests

As mentioned earlier, the armrest width adjustment was removed from the prototype. This simplification, combined with the mechanical challenges described later in Section 2.4.1, necessitated a different design for the armrest height adjustment than the one in the design specification. Although the exact implementation is different, the prototype still uses a power screw and nut to provide mechanical articulation. A fixed armrest width removed the requirement for the distance sensors and one of the DC motors. The armrest height switches are similar to the original design, but have a smaller activation threshold.

#### 2.3.1.2 Seat Depth

The seat depth adjustment was removed from the prototype, eliminating the need for one of the DC motors and the associated mechanical parts. Additionally, as it was unnecessary to sense the user's contact with the backrest, the strain gages were removed.

#### 2.3.1.3 Lumbar Size

The lumbar support size adjustment mechanism in the prototype is nearly identical to the original design. The bleeder was removed because it was unnecessary (opening the lumbar support to the atmosphere was sufficient). The flow sensor was removed because it was deemed too inaccurate to provide a useful measure of the volume inside the lumbar support. The flow sensor would have difficulty distinguishing direction of flow, and would also have difficulty obtaining an initial known volume. As well, the processing required to continuously sample the flow sensor to perform the integration would provide too much burden for the microcontroller. Finally, we were unable to source a reasonably priced, yet physically robust flow sensor that did not require additional hardware such as air filters. The prototype has no way of knowing the volume in the lumbar support. However, against a fixed surface such as the user's back, an increasing volume will result in an increasing pressure. This characteristic enabled us to use the pressure sensor as a safety monitor. The original design called for an inflatable latex lumbar support. Although our initial experiments with latex air bladders were promising, we had great difficulty trying to construct an appropriately shaped latex lumbar support. Instead, we found a plastic inflatable lumbar support and simply refitted it to our air system.

### 2.3.2 Electronics

Various aspects of the electronics were also changed during implementation. These changes pertain to the power distribution system, the signal conditioning, and the computation unit.

#### 2.3.2.1 Power Distribution System

The original power supply was intended to be a single ATX PC power supply with an additional low dropout linear voltage regulator on the 5-volt line. This was replaced with an ATX PC power supply for the 12-volt line and a separate 5-volt power supply. Because we did not use PCBs, the prototype does not have ground planes. However, attempts have been made to separate the 5-volt digital ground wires from the 5-volt analog ground wires. Heatsinks were not necessary for any of the electronics.

#### 2.3.2.2 Input Signal Conditioning

For the input signal conditioning, after eliminating many of the analog sensors, the only remaining analog sensors were the pressure sensors and the current sensing output of the H-bridges. To avoid using an amplifier, we chose pressure sensors with integrated filtering and amplification circuitry. The current sensing output is amplified by the Darlington optoisolator. The optoisolators for the other signals do not amplify the input. We used switch debouncing ICs on the armrest height switches, removing the need for digital filtering. However, digital filtering was required for all of the analog inputs. With our reduced sensing needs, we also had reduced needs for ADC channels. However, the prototype still requires more ADC channels than provided by the microcontroller. Rather than using an external ADC chip, we connected some of the analog inputs via an analog multiplexer.

#### 2.3.2.3 Output Signal Conditioning

For the output signal conditioning, the main difference is in the pulse-width modulation (PWM) outputs. The original design intended the microcontroller to generate the PWM signals that would control the H-bridges. However, with the advent of the dedicated motor controllers in the prototype, the main microcontroller does not generate the PWM signals.

#### 2.3.2.4 Computational Unit

The computation unit is an area of significant deviation between the prototype and the original design specification. The original intention was to have a single monolithic processor controlling all of the hardware. After an extensive survey, we found that suitable processors would have at least 100 pins in a small surface mount package. The difficulty of wiring such a chip, combined with the software difficulty of writing and coordinating multiple real-time feedback control loops, led us away from the monolithic processor. Since motor control represented the majority of the processing complexity, we decided to use dedicated motor controller chips. Fortunately, the motor controller we used communicates via a parallel bus, increasing the speed of communication. The parallel bus also removed any problems related to bus arbitration because the main processor simply has to set the bus switch select lines to establish a direct connection from the main processor to the motor controller. The bus switch hardware ensures that no bus conflicts occur.

With a reduced number of adjustments, reduced sensors, and reduced computational requirements, the number of inputs and outputs decreased significantly in the prototype. For the analog inputs, the pressure sensor and lumbar size dial remained connected to a multiplexed input, but the number of motors was reduced to three and the remaining sensors were all removed. For the digital inputs, the encoders were no longer connected to the main processor, and only three limit switches were required. The seat presence sensor was removed and the number of user buttons reduced to six. For the digital outputs, the motor direction and power control became the responsibility of the motor controllers, and we used an LED driver IC to drive the sixteen status LEDs, rather than directly connecting the LEDs to the main processor. Taking into account all of the simplifications and reductions, we still ended up following the design specification of using a TQFP chip with 64 pins. The current prototype uses all but one of the 64 pins, so future expansion would likely involve either moving to a larger chip, or using more hardware peripherals, multiplexers, or I/O port expanders.

### 2.3.3 Software

For the control software, the original intentions were generally followed, although the implementation details differed. The task scheduler became the FreeRTOS kernel. The motor feedback control and encoders tasks became the responsibility of the motor controllers. A single task on the main processor combined the limit switch task with arbitration of access to the motor controllers. The lumbar support volume task was eliminated because the prototype does not have a flow sensor. Similarly, the load sensing task was removed because the prototype does not have force sensors. The user control task was modified to poll the state of the UI buttons. The main process was divided into a main task and three automatic adjustment tasks. The armrest width determination task was removed since the armrest width was not adjustable. An updated table of the control software is shown in Table 1.

**Table 1: Software Tasks.**

| Process | Priority | Description |
|---|---|---|
| Motor command issuing +<br>Limit switches | 1 | Issues commands to move the motors. Also ensures safety by examining the output of limit switches, encoders, and motor controllers. |
| Analog Mux Access | 2 | Controls read access to analog sensors. |
| Current Sensing | 3 | Implements stall protection for the motors |
| Pressure and dial | 4 | Monitors the pressure level for safety purposes. Also monitors the pressure dial on the user interface. |
| Button Checker | 5 | Checks user interface buttons and notifies other tasks of changes. |
| Main Process + Manual Mode | 6 | Implements the logic for Auto and Manual mode. Makes decisions based on user input and sensor readings. |
| Armrest Height Determination | 7 | Determines a suitable height for the armrests. Also equalizes the armrests if they are at different heights. |
| Lumbar Height | 7 | Determines a suitable height for the lumbar support. |
| Lumbar Support Size | 7 | Inflates or deflates the lumbar support to the specified pressure. |
| User Display | 8 | Controls the status LEDs and displays messages to the LCD. |
| Idle task | 9 | Runs when no other tasks are running. Does nothing. |

### 2.3.4 User Interface

The prototype User Interface matches the original design, minus the buttons for armrest width and seat depth adjustment. With just the armrest height and lumbar height adjustments present, it was logical to combine the functions on to a common up/down button. Because of hardware limitations on GPIO pins, the prototype has two debugging buttons rather than four. The debugging buttons are located on the enclosure, rather than on the UI. During normal operation, the user should not have to press the debugging buttons, so we moved them to simplify the interface and avoid confusion. The UI software differs in implementation: an RTOS task polls the buttons rather than an interrupt handler, and debouncing is performed by hardware.

## *2.4 Challenges in Design and Implementation*

### 2.4.1 Mechanical

Numerous challenges presented themselves during this phase of the project. First of all, none of us specialize in mechanical design nor did we have significant skills in designing or fabricating mechanical systems. Starting off with the mechanical design of the various components on the chair, we first had to learn about different methods of actuation for each of the adjustments. In the end we decided that it would be simplest and most economical to use power (or ACME) screws to transform a motor's rotational motion into linear motion. Designing the actual mechanism housing the power screw to actuate the motion was a challenge. Another major challenge was to find (i.e. learn about) various mechanical components that would allow the system to be more tolerant of inaccuracies in manufacturing. The spider flex couplers that connect the motors to the power screw are an example of such a component.

Although we decided to use an air-filled lumbar support, we had no idea how air systems work. This led us to research air system designs before we could begin designing our own air system. As mentioned in Section 2.3.1.3, we experimented with several low-cost options for the lumbar support before deciding on the current solution. Our solenoids are truly miniature, with tiny fittings. This oversight imposed a restriction on the airflow in our system, and limited the inflation and deflation rate of the lumbar support. The connectors in the air system are of low-quality and could not be tightened too much for fear of breakage. As a result, small leaks appeared throughout the air system. Finally, our original intention was to deflate the lumbar support by simply exposing it to the atmosphere. However, as it turns out, relying on atmospheric pressure equalization is not sufficient to completely deflate the lumbar support. The lumbar support requires an external pressure to generate sufficient vacuum before it will completely deflate. Normally, the weight of the seated user pressing on the lumbar support generates sufficient pressure. Without a user pressing on the lumbar support, however, it is unable to completely deflate. As well, the low flow rate hinders the deflation process.

After completing the designs, we then had to learn SolidWorks in order for us to share our visualizations of the components and how they would fit on the chair. Before the designs could be finalized, we needed to specify all the parts we needed for the chair. Estimating the force and torque requirements for the adjustments was a challenge. Due to budget limitations, we often utilized parts from surplus retailers, requiring us to revise our designs to fit these one-of-a-kind parts. Also, because parts of certain sizes were expensive, we used improperly-sized parts whenever they were more economical. We then devised methods to couple these parts to the rest of the system.

After completing the mechanical design, we quickly realized that fabricating our SolidWorks design would not be possible. The first reason is that we had designed the components in metal. We did not have the necessary tools to support construction entirely in metal. Also, we discovered imprecision in the drill press and lathe. The lathe is misaligned and the drill presses have an inherent tilt on the stand that lead to holes that are tilted. Secondly, the team lacked fabrication skills. Since we did not have the skill to use a milling machine, we were unable to fabricate components with more complex shapes. Furthermore, we could not effectively construct most of our original designs to meet the required tolerances. As a result, we revised our designs to be simpler and more robust against manufacturing tolerances. The primary construction materials became wood and plastic, and many more screws and fasteners were used in the new designs.

Since we needed to redesign the affected components quickly, the new designs were only drawn on paper and not in SolidWorks. We could not completely visualize the end result of all the components placed on the chair. As a result, we had difficulty when it came to fitting all the components onto the chair.

Furthermore, accidents, bad design, unforeseen properties of the material, or poor construction often led to broken parts. Sometimes the broken part could be repaired, but many times a re-make or even a redesign of the part was required.

### 2.4.2 Electronics

In general, the electronics were fairly well-behaved with no major problems. However, a few minor problems required solutions or workarounds. The electronics in the prototype are extensive, spanning six boards and requiring more than 70 metres of wire. When possible, we tried to use through-hole components to make soldering easier. Unfortunately, many of our parts simply did not have through-hole packages and we were forced to make single-sided PCBs to adapt the surface-mount parts to the through-hole vector boards. Soldering surface-mount parts with many pins was difficult enough. The sheer amount of soldering and fabrication required of us was a greater challenge. Making six boards took some external assistance, but outsourcing was not a panacea. We had to thoroughly check and re-solder some of the connections performed by third-party fabricators.

After assembling the main processing board, we found that the ADC on the microcontroller did not work. After verifying electrically connectivity and checking the software, the ADC still did not work. We consulted Atmel technical support as well as a forum for AVR users and both sources said that it should work. However, the ADC still did not work. At our wits end, we arranged for assistance from a local engineering company. The night before we were to visit them, the ADC magically started working perfectly. Since then, we have not had problems with the ADC. We still do not know why the ADC was not working, and what happened to get it working.

From a design perspective, the electronics had two minor problems. First, we had originally required a presence sensor – something to detect a user's physical presence on the chair as a sanity check. We had intended to use a flex sensor connected to a Schmitt trigger. Unfortunately, upon trying this configuration, it was found that sitting on a flex sensor did not produce a large enough flex. We were unable to find a suitable alternative, so the prototype lacks a presence sensor. The second design flaw was with our choice of power supply. To simplify design, we used a standard ATX PC power supply to provide 12 volts to the motors and solenoids. To save costs, we scrounged surplus power supplies. These power supplies were able to move one motor under light loads, but proved to be incapable of moving all three motors simultaneously. Therefore, we were forced to purchase a new power supply. The new supply was rated for 400W with at least 10 amps on the 12-volt line. Unfortunately, the new supply did not live up to its rating and would occasionally shut down due to overload. The new supply was capable of moving all three motors simultaneously, but not under a heavy load. We have yet to find an affordable 12-volt power supply capable of fulfilling our needs.

### 2.4.3 Software

The software design process was trouble-free; however, the software implementation process had some problems. Aside from programmer-induced syntax errors and logic errors, most of the software challenges were due to memory (RAM) limitations. Our microcontroller is a simple 8-bit processor lacking a memory management unit. Furthermore, FreeRTOS is a simple multi-tasking kernel, implementing only the core prioritization and context switching features of an RTOS. Unlike PCs running Windows, Linux or any other modern operating system, our prototype does not have the conveniences of memory isolation and protection. With only 4KB of RAM, memory constraints affected the software design process from an early stage. FreeRTOS requires each task to allocate the memory requirements beforehand, meaning a task's stack and heap cannot grow as needed. Because memory was tight, we sparingly allocated stack space to each task. Initially, we had no problems, but as tasks became more complex, memory usage often overflowed the allocated space. When this happened, a segmentation fault or kernel panic did not occur. Instead, tasks would overwrite each other's stack spaces and proceed to restore a context switch, sending execution off to never-never land. Thankfully, the watchdog timer would quickly reset the chip, but it became quite the challenge to find what section of the software was responsible for the latest watchdog reset.

Our primary debugging tool was a JTAG In-Circuit Emulator (ICE) compatible with the AVR development software. This tool allowed us to set breakpoints and step through our software. The JTAG ICE was invaluable and we would not have been able to complete the software without it. However, because of the complexity of our software, the JTAG's limitations were pronounced. In particular, the JTAG could only support three simultaneous breakpoints. This was a problem considering we had at least eight tasks running concurrently, in addition to four interrupt handlers. We also found single stepping through the software to sometimes behave in unexpected ways. With multiple interrupts active, stepping to the next line of a task may actually end up in an interrupt handler or inside the internal RTOS software.

### 2.4.4 Integration

Integrating the mechanical and electronic parts and the software proved to be quite the challenge. Building the mechanical parts of the chair took some time. Mounting the boards and wiring them such that the electronics do not interfere with the user needed some further investigation. The space requirements for the electronics were restrictive so we had to come up with a way to make them fit. We also had to ensure that adding the electronics did not remove functionality from the chair. For example, the seat height must still be adjustable. We came up with a solution but only after taking many factors into consideration.

Another integration challenge that we had was tying the hardware and the software together. All the adjustable parts of the chair have to move properly as controlled by the software, such that they do not interfere with the other parts' movement. One issue we had was making sure that the automatic adjustment was done in order. After the armrests have adjusted, the lumbar height should follow suit. We had some issues making sure the sequence is performed properly. We had a problem with the state machine such that only the armrests would adjust. Another issue we had was the current sensing always triggering such that after the armrests reach the upper limit, the system would just die. It was tricky business trying to make everything work together without exhausting the memory and making sure that we are operating within the limits of our electronics.

# 3 Future Plans and Recommendations

## *3.1 Additional Features*

Additional features include the addition of more automatic adjustments, such as the seat tilt and armrest width. Adding the capability to detect user presence will help with power management and safety. To remember a user's settings, a memory feature can be added.

### 3.1.1 More Automatic Adjustments

The following adjustable parts can be made to adjust automatically in a similar fashion to the current automatically adjustable parts:

- Backrest tilt
- Headrest adjustment
- Armrest width
- Seat depth
- Seat height

The similarities only apply to how the adjustments are controlled. Mechanically, these features are more complex than our current features and will be discussed in more detail in Section 3.2.

### 3.1.2 User Presence

This feature is useful in the adjustment process of the chair. The current chair is unable to detect whether a user is present. If the user leaves the chair during the adjustment process, the user presence sensor will enable the chair to stop adjusting. It is strongly recommended for this feature to be implemented in the final ACE Chair

### 3.1.3 Settings Memory

This feature pertains to remembering a user's preference for the positions of the various motors and the pressure in the lumbar size. Equipped with a settings memory, the chair only needs to adjust the first time the user sits on the ACE Chair. The chair will adjust to the remembered setting the next time the user returns. This feature has a limitation. Throughout the day, a user's position changes so the chair settings need to change as well. One saved setting is not enough. The workaround for this is to save the settings once in the morning and once in the afternoon. The user then has to switch between the two settings. This feature is only recommended if it is definite that the chair will adjust automatically to an average person the same way every day, such that the auto-adjustment itself is redundant. The user interface also needs to be made as error-free as possible such that a user will choose settings for him and not for another user.

## *3.2 Mechanical*

Future plans for the ACE chair include having the adjustable parts mentioned in Section 3.1.

### 3.2.1 Backrest Tilt

Automatically adjusting the backrest tilt is useful because it provides the user the most comfort and support at the correct angle. This feature is a bit complicated to implement because the motion is nonlinear. A system involving gears to implement angular motion is needed. Moreover, space issues need to be considered as this gear system will likely take up a significant amount of space. More planning is required before this feature can be implemented. However, this feature is nonetheless useful so this should be added to the chair once a proper implementation is thought of.

### 3.2.2 Headrest Adjustment

This feature allows the ACE Chair to support the user from the lower back up to the neck and head. Implementation is similar to the lumbar height adjustment, with a small motor driving the up and down movement of the headrest. Adding this feature is strongly recommended because the design of the lumbar height can be adapted and reused for the headrest adjustment.

### 3.2.3 Armrest Width

This feature will move the armrests toward and away from the user's sides. Having this adjustment will allow persons of varying sizes to sit in our chair while having the armrests in an ideal position from their bodies. In our design specifications, we mentioned that we can use "one motor for the armrest width, coupling the output such that it turns two screws simultaneously" [3]. This adjustment can definitely be automated but, like the backrest tilt, the design must be finalized as simultaneous screw turning can be difficult. An alternative design would use two motors. The same approach can be used as in our current armrest height design where two motors move the left and right armrests up and down, controlled by software to ensure they are of matching height. Replace up and down with outward and inward, and matching height with matching distance from the centre, and you have the makings of the automatic armrest width adjustment.

### 3.2.4 Seat Depth

The seat depth is a good adjustment to have because it allows the user's back to be supported by the backrest while the knees are in the right position away from the seat. If the seat is too deep then the seat will apply pressure to the back of the user's knees. If the seat is too shallow then the thighs will not be supported enough. This feature is tricky to implement because adjusting the seat depth will involve either moving the seat forward and backward, or moving the backrest forward and backward. Moving the seat would involve overcoming the mass of the user. Moving the backrest would probably be easier as the backrest can stop moving as soon as it touches the user's back. Therefore the recommendation would be to move the backrest for implementing the auto-adjustment of the seat depth in the future.

### 3.2.5 Seat Height

Moving the seat height involves overcoming the user's mass. This would require some kind of pneumatic system to properly adjust the seat height automatically. Trying to use a motor would be hard as the motor has to work hard to even move the seat a little. A pneumatic system would be slow though so any auto-adjustment would require some time. Also, most users have an idea of how to adjust their seat height, so an automatically adjustable seat height is of lower value. The recommendation for this feature would be to only implement this once the other four features are implemented and only if the adjustment time can be made short.

## *3.3 Electronics*

As the prototype is expanded mechanically with additional adjustments, the electronics would have to be expanded as well. The paradigm of having a dedicated motor controller for each motor, and a less powerful main processor, has been demonstrated to work well. Because of limitations on the number of GPIO pins, we chose to connect the main processor to the motor controllers through bus switches. This means that only one motor controller can be accessed at a time, but this limitation has not been a problem. Logically, it would seem that from an electronics perspective, adding additional adjustments is simply a matter of connecting more motor boards. Unfortunately, the current electronics are near the limits of what is possible with the current hardware choices. To support more motors and more adjustment dimensions, a radically new design is required. The bus switches we used were the only parts we could find that would provide a data path with the width and number of connection points required. Chips that connect four or more destinations on a 20-pin data bus do not exist. Thus, it is recommended to look at programmable logic solutions if more motor boards would be connected. The logic functions themselves are not complicated (it's only multiplexing and de-multiplexing); it is the width of the data bus and the number of end points that is challenging.

Aside from additional motor boards, presence sensors and additional pressure sensors are easy candidates for electronics upgrades. The current main processing board connects the pressure sensors to the ADC via an analog multiplexer. The analog multiplexer has six open channels. As mentioned earlier, the presence sensor proved to be too much of a challenge to implement in the current prototype. However, presence sensors provide great value to the robustness of the ACE Chair, and as such, should be a priority. Possible solutions might be capacitive sensors.

Our current electronics are massive, requiring a large wooden enclosure to contain everything. An obvious area for future development would be to use surface mount components wherever possible, and to place everything on PCBs. Initially, we were unsure of how to mount the various boards. Thus, the electronics were spread over multiple boards and connected with ribbon cables. With part miniaturization, it is likely that multiple boards could be combined onto a single PCB, eliminating the need for the bulky ribbon cables and connectors. After the electronics were designed and assembled, we became aware of various minor fixes for the electronics. The fixes mostly involve adding passive components, such as additional pull-up resistors, filter capacitors, and line-termination resistors. If PCBs are made, the new boards would implement these small fixes.

## 3.4 Software

The current software has realized most of the potential of the current mechanical and electronics hardware. Without additional hardware, future software development would mostly involve optimization, polishing, and robustness improvements. The software is not stable under all conditions and some minor bugs could still be fixed. The automatic adjustment sequence uses transient tasks that are activated and de-activated as necessary. However, if the automatic adjustment sequence is interrupted, the transient tasks must be reset. This custom resetting of tasks is not implemented. The prototype has not undergone extensive user testing, and the lumbar height adjustment algorithm could still be improved. On the other hand, if additional hardware were added to the prototype, additional software development would certainly be required to utilize the new hardware.

# 4 Budgetary and Time Constraints

In the following sub-sections, we will compare the proposed budget and schedules to our actual spending and project timeline.

## 4.1 Budget

Table 2 compares the proposed budget in our project proposal [2] to our actual spending on the prototype.

**Table 2: Comparison of Estimated and Actual Cost.**

| | Estimated Cost | Actual Cost |
|---|---:|---:|
| Electronics + Processors | $200 | $1,065 |
| Mechanical Parts | $100 | $728 |
| Miscellaneous | $250 | $314 |
| Actuators | $500 | $268 |
| PCB Manufacturing | $200 | $116 |
| Chairs | $100 | $25 |
| Power Supply | $100 | $21 |
| Sensors | $350 | $0 |
| **Total** | **$1,800** | **$2,538** |

The estimated and actual costs for electronics and mechanical parts differ by $738. This discrepancy is due to the design complexity that was unforeseen at the outset of this project. Our design choices, which emphasized safety and durability, required a larger number of parts. Duplicate electronic parts were purchased to hedge against accidental failure and human error.

The category of miscellaneous spending includes shipping and customs fees as well as material to improve the aesthetics of the ACE Chair. The major area of spending in this category, however, was to purchase tools that we were unable to borrow.

We made the initial budget estimate for sensors different than those actually used in the prototype. As a result of a thorough design and parts-sourcing phase, we were able to eliminate redundant sensors and chose sensors that were available free-of-charge as manufacturers' samples.

As a result of our decision to construct the prototype from vector boards instead of PCBs, we managed to save some money in the category of PCB Manufacturing. However, we still needed to make our own PCBs to accommodate some of the electronic components on the vector boards. The amount saved in PCB manufacturing was spent on acquiring materials to construct vector boards. The purchase of these materials increased our spending in the category of Electronics and Processors.

After reducing the proposed set of functionality, we eliminated the need to purchase several powerful (and more costly) motors. The lower final cost of actuators reflects this decision.

Figure 5 shows the distribution of spending by category, as a percentage of the total cost.

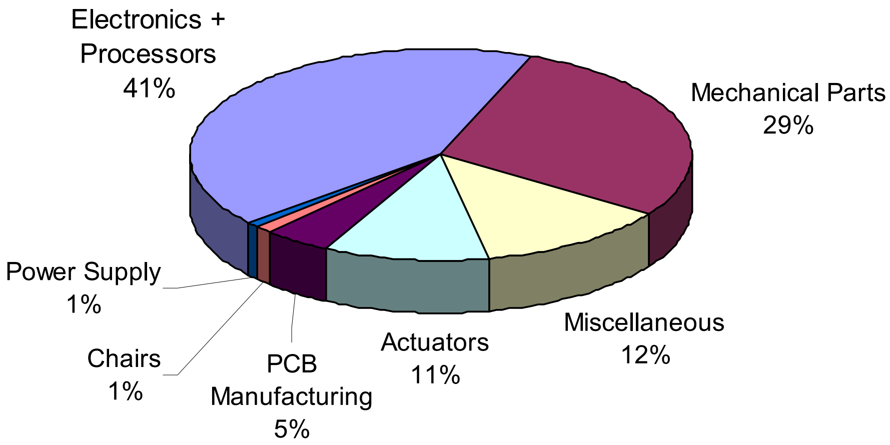
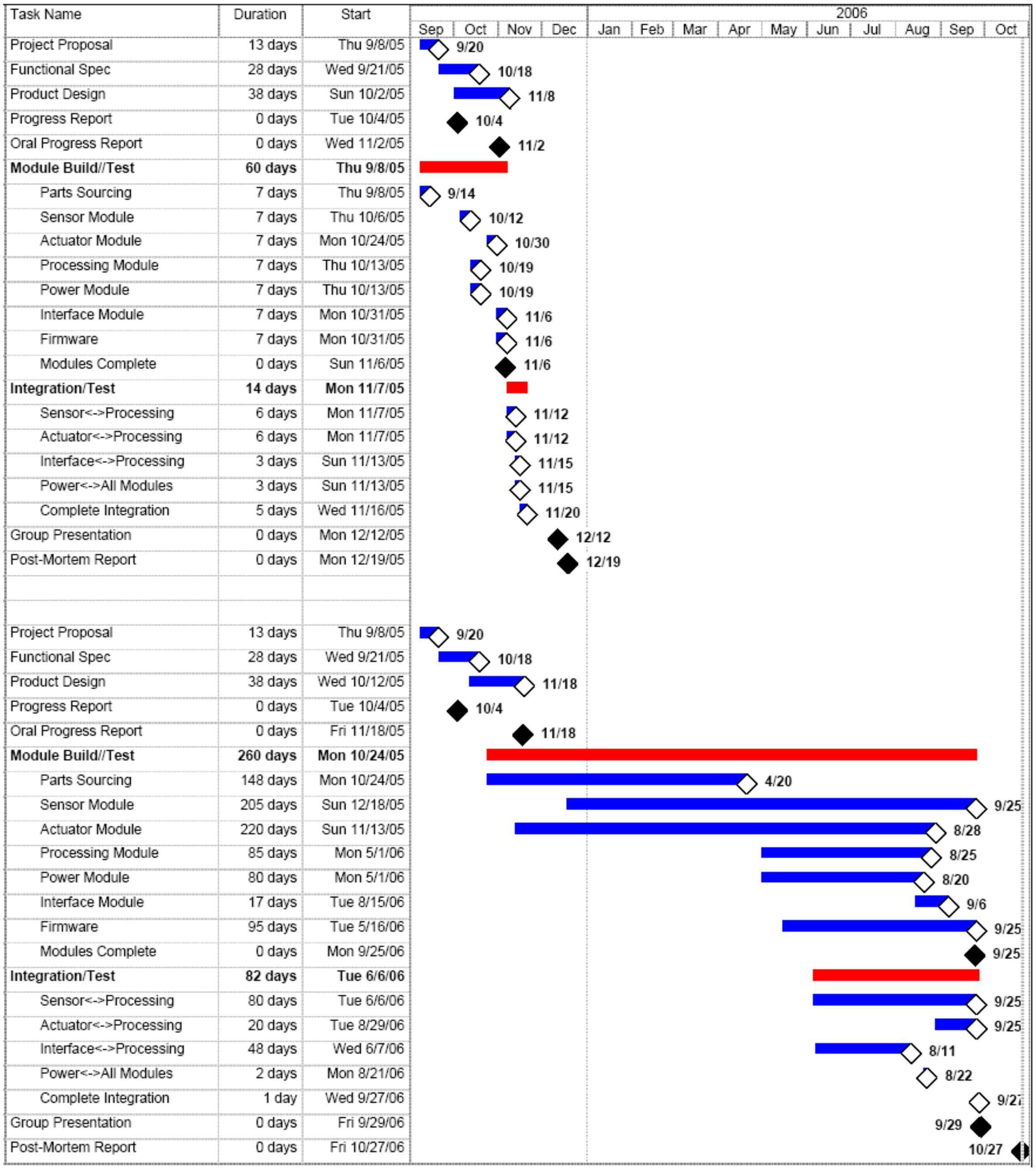


**Figure 5: Actual Cost of Prototype by Category.**

## 4.2 Schedule

Figure 6 shows the Gantt charts showing the proposed and actual project timeline. The start date and duration of each stage is indicated in the columns to the left. The end of each stage is marked by a white diamond with a finish date. Milestones are indicated by black diamonds.



| Task Name | Duration | Start |
|---|---|---|
| Project Proposal | 13 days | Thu 9/8/05 |
| Functional Spec | 28 days | Wed 9/21/05 |
| Product Design | 38 days | Sun 10/2/05 |
| Progress Report | 0 days | Tue 10/4/05 |
| Oral Progress Report | 0 days | Wed 11/2/05 |
| **Module Build//Test** | **60 days** | **Thu 9/8/05** |
| Parts Sourcing | 7 days | Thu 9/8/05 |
| Sensor Module | 7 days | Thu 10/6/05 |
| Actuator Module | 7 days | Mon 10/24/05 |
| Processing Module | 7 days | Thu 10/13/05 |
| Power Module | 7 days | Thu 10/13/05 |
| Interface Module | 7 days | Mon 10/31/05 |
| Firmware | 7 days | Mon 10/31/05 |
| Modules Complete | 0 days | Sun 11/6/05 |
| **Integration/Test** | **14 days** | **Mon 11/7/05** |
| Sensor<->Processing | 6 days | Mon 11/7/05 |
| Actuator<->Processing | 6 days | Mon 11/7/05 |
| Interface<->Processing | 3 days | Sun 11/13/05 |
| Power<->All Modules | 3 days | Sun 11/13/05 |
| Complete Integration | 5 days | Wed 11/16/05 |
| Group Presentation | 0 days | Mon 12/12/05 |
| Post-Mortem Report | 0 days | Mon 12/19/05 |
| | | |
| Project Proposal | 13 days | Thu 9/8/05 |
| Functional Spec | 28 days | Wed 9/21/05 |
| Product Design | 38 days | Wed 10/12/05 |
| Progress Report | 0 days | Tue 10/4/05 |
| Oral Progress Report | 0 days | Fri 11/18/05 |
| **Module Build//Test** | **260 days** | **Mon 10/24/05** |
| Parts Sourcing | 148 days | Mon 10/24/05 |
| Sensor Module | 205 days | Sun 12/18/05 |
| Actuator Module | 220 days | Sun 11/13/05 |
| Processing Module | 85 days | Mon 5/1/06 |
| Power Module | 80 days | Mon 5/1/06 |
| Interface Module | 17 days | Tue 8/15/06 |
| Firmware | 95 days | Tue 5/16/06 |
| Modules Complete | 0 days | Mon 9/25/06 |
| **Integration/Test** | **82 days** | **Tue 6/6/06** |
| Sensor<->Processing | 80 days | Tue 6/6/06 |
| Actuator<->Processing | 20 days | Tue 8/29/06 |
| Interface<->Processing | 48 days | Wed 6/7/06 |
| Power<->All Modules | 2 days | Mon 8/21/06 |
| Complete Integration | 1 day | Wed 9/27/06 |
| Group Presentation | 0 days | Fri 9/29/06 |
| Post-Mortem Report | 0 days | Fri 10/27/06 |

**Figure 6: Gantt Charts (Top: Proposed, Bottom: Actual).**

The original schedule can also be found in the proposal [2]. As is evident from the Gantt charts, the project was completed approximately nine months later than originally proposed. The main cause of this slippage is an over-ambitious feature set coupled with an overly-optimistic schedule. The feature set was not narrowed down until after December 2005. Submission of the design specification document, which marks the end of the design cycle, occurred in November 2005. However, the actual product design phase extended past this date since we needed to modify the design after reducing the feature set.

The parts sourcing stage consumed a large portion of the development cycle. In this stage, extra time was spent on research in order to reduce costs as much as possible. The lengthy duration of the sensor module phase does not reflect the short implementation bursts that occurred throughout the build phase. In contrast, the actuator, processing, and firmware modules required constant effort throughout the duration of the periods specified on the Gantt chart.

It is interesting to note that the original proposed schedule shows development stages that occur with minimal overlap. We found that in reality this is not the case. By performing the stages iteratively and in parallel, we achieved the large overlap seen in the lower Gantt chart.

Instead of reducing the feature set drastically and rushing through the design and implementation, the Accomodarsi team has chosen to focus on quality, safety, and thoroughness in all stages of product development. With the extra time spent, the team has been able to deliver a functioning ACE Chair that is safe to use, durable, comfortable, and aesthetically pleasing.

Although the Accomodarsi team realized at the outset that the proposed feature set was an unrealistic undertaking for a four-month completion, we all felt that it was important to deliver no less than the current set of features in order to successfully demonstrate the concept of the ACE Chair. We feel that we have achieved this goal with our product presentation and demonstration on September 29, 2006.

# 5 Interpersonal and Technical Experiences

The following section contains individually written descriptions of each member's contribution to the project, what was learned, and what each team member would do differently if they were to undertake a similar project again. This section also includes a description of group dynamics within our team.

## *5.1 Eric Leung*

Initially we started off with Stephanie and two Erics in August 2005. We brainstormed numerous ideas and talked to a lot of people about our ideas. As Eric and I had worked together on a fairly substantial extra-curricular project before, we knew four months is not much time to complete any big projects. However, we really want to make something that is truly not done/made/developed before by anyone. With this in mind, just a few days before Jennard joined us in September, the three of us determined that the ACE Chair would be a suitable candid for our ENSC340 project.

Boy, were we ever wrong! The four-month project turned out to be thirteen months!

In the beginning I assisted Jennard in designing the mechanical assemblies. Once the mechanical design was available, I worked with Jennard to construct the chair. I taught Jennard on using the lathe and we lathed the power screws for the lumbar support and the armrest movements. I had previous experience lathing aluminum stock, but lathing the stainless steel power screws was a different story and in the end, my skills using the lathe increased substantially. We had to clear several other major obstacles along the way, from trying to fabricate certain parts within the required tolerance to trying to fit the all parts/assemblies on the chair.

On the electronics side, I learned a lot while sourcing parts and verifying the schematics. (By the way, did I mention we have 6 circuit boards and many, many pages of schematics?) I gained more experience with etching PCBs when we made our own surface mount to through-hole converters on a PCB. I also did a lot of soldering. I managed to screw up a couple sample chips along the way, but I gained a lot of experience with soldering and de-soldering various through-hole and surface mount (TSSOP, SOT, and even 64-pin TQFP packages!) components. Although it came a bit late, I realized that a fume-extraction system is a must.

For software, I learned that software is one of my weakest points and I was not able to provide much assistance to the group other than work on the high-level design of the software and the state-machine for the operation of the chair.

One thing I must mention is that we spent a lot of time on designing the chair. A lot of time and effort (I must emphasize "a lot") were spent to make sure we had good designs for the mechanical parts, electronics, and software. This approach definitely worked and it paved a fairly smooth road for implementing the various systems. Looking back, we did not have to spend too much time debugging the various systems.

One major factor in the delay in the completion of the project was the amount of mechanical design required. We needed to make armrests and a lumbar support with the desired movement range that is small enough to fit on the chair and strong enough to handle most of a person's weight. It was very rewarding when Mike sat on the chair, and the chair could still operate. Nothing broke except for a ball bearing that popped out. (The ball bearing was easily put back into place.)

In retrospect, a lot was learned. Firstly, we learned to acquire cheap things to stay within budget. We frequented surplus stores, scrap metal yards and various sales. We tried to save money by making different items ourselves. Buying $500 dollar linear actuators? No. We made mechanical assemblies that perform the same task for much less cost. However, making such assemblies consumed a lot of time and a couple of the hand-made parts lacked strength and reliability and had to be re-designed and re-made.

The past 13 months had been a big challenge. Juggling between project, courses and co-op proved to be very hard. It took a lot of dedication, persistence and time. On the positive side, no group dynamics problems developed and we worked pretty well with each other. I don't think any of us hate each other and we are still friends with each other. On the negative side, we had time-management issues, specifically on estimating the time required to finish a certain task. Many small oversights over the course of the project accumulated to big oversights. An example oversight was that our functional and design spec were over ambitious. But let's not dwell on those.

Lastly, I cannot decide if the project overall was a success or a failure. In a project management viewpoint, we were pretty much a failure. However, given all pre-cautions we have taken, the meticulous designs of the various systems on the chair, and the amount of thought, effort, and time that was put into the project, we ended up with a working prototype. In that sense, I believe we have succeeded.

## 5.2 Jennard Dy

I was supposed to do the project with two of my friends. Unfortunately, both of them decided to withdraw from ENSC305 and ENSC340 and encouraged me to drop them, and just take them next semester. I was hesitant to do this since I planned to go on co-op the following spring. In the end, I continued with the course and I just needed to find a group to join. There were two groups that still needed one more member: a group of two and a group of three. I heard stories about the two Erics (from the group of three) from my two friends and they told me that if I joined their group, we would have a good project. Plus, since there were already three of them, if I join them, there would be less work for me, right? Little did I know what was in store for me…

A year has passed and the chair has finally been presented. Looking back, it was a good project and I did learn many things. First, I learned to write with a team various project documents like a project proposal, functional specifications, and design specifications. Our group initially had many features of the chair that we wanted to implement and we later learned that our goals were unrealistic. Everyone contributed to defining what the functions of the chair was going to be. With regards to design, I was mostly involved in the mechanical and software aspects. I was the main designer for the armrest height and lumbar height mechanisms. I did the initial sketches and refined the design with the help of the others. We went through many iterations and we finally had a design we could be happy with. We used SolidWorks for visualizing how the parts would move, and to draw the assemblies for use in the design specification.

After the design, Eric Leung and I worked on building the chair. We spent countless hours on the machine shop sawing wood with various saws, drilling various holes in wood and metal, and lathing the screws. Regarding the electronics, I was mostly involved in sourcing the parts with the rest of the group as we needed numerous parts. I pitched in making package layouts for the ICs in Eagle. For electronics design, I designed a circuit that will use the flex sensors that we sampled. Unfortunately, I found through testing that the flex needed was great. I tried to minimize the flex by altering resistor values and the sensor configuration but I could not reduce it anymore.

Later, we all worked on the high-level design of the software to control the various adjustable parts that our chair had. As for low-level design and implementation, I did the queue mechanism that allows data to be passed between the various tasks in the RTOS. I also implemented the button checking task, and then the LCD and LED display task with Steph using methods that Eric Lee implemented to interface with the hardware. Afterwards, Eric Leung designed the state machine and I implemented it along with the main control logic for the auto mode of lumbar height and lumbar size. In addition, I also implemented the logic for manual mode for the lumbar height and size, and armrest height. I also worked on the equalization feature to ensure that the armrests are the same height.

Regarding my experiences with the members of the group, it was actually pretty good. We helped each other when one of us has questions. They are all nice people and good to work with. We got to know each other better over the year and I did not find any of them annoying or irritating after that long time. That said, I noticed that everyone has his or her own unique quirk. Eric Lee would always start a conversation with “So…”. Eric Leung would always say “We have/We got/I see a problem” every ten minutes. Steph would always take a nap whenever we get together to work on the chair. We subsequently coined the phrase “pulling a Steph” to mean napping. As for me, I was the group clown. I am pretty sure that in every meeting or group session that we had, I make one joke or do something funny. I guess that was my way of coping with the stress of working on the chair. It was more for my sanity than actually making a conscious effort of making them laugh. In the end, I spent more time with them than any other people excluding (or maybe including) my family. In fact, I don't have the feeling of never wanting to see any of them again so that in itself is a testament on how well we worked with each other.

## 5.3 Stephanie Fung

Over the course of the year, I learned many things from both technical and interpersonal perspectives. On the technical side, I learned how to source electronic parts through various manufacturer websites and through hardcopy catalogs. I have learned more than I ever wanted to know about the different types of fuses, switches, LEDs, connectors, solenoids, and sensors available for purchase. After going through this phase of the project, I hope not to have to touch a DigiKey catalog again for a long time.

By designing and constructing the lumbar support subsystem, I learned about air systems and solenoids. It was a challenge to find affordable solenoids and air cushions that would keep us on budget. These challenges caused me to seek innovative designs and select cost-efficient materials.

I contributed to designing and building the user interface and solenoid boards. I learned about electronic circuits for debouncing input and for driving solenoids. From assembling the user interface and solenoid boards, I learned to point-solder and solder surface mount components. I also experienced the frustration of de-soldering and the effects of excessive fume inhalation. I also became adept at fabricating home-made PCBs.

In contributing to the software design, I learned about the complexities of designing a multithreaded system. While implementing the task that would display output to the user interface, I learned how to develop and debug software on the AVR.

With respect to project management, the only thing I learned about time estimation is that we still have a lot to learn. For me, much of the difficulty in time estimation came from not having previously completed a technical project of this level of complexity. Along with my team members, I learned (albeit too late) the importance of scoping out a project early on. If I were to undertake a similar project again, I would try to overestimate the time to complete each task. To better scope out the project, I would speak to experts to better understand the complexities of the project since they may not initially be apparent to me.

Working on this project for 13 months has thoroughly tested my dedication, patience, and perseverance. I have learned a lot about myself and my teammates with respect to these three qualities, and have formed a deep respect for their determination and persistence in completing this project. From my team members, I started to understand the value of meticulousness and thoroughness. Although I was often tempted to save time by choosing the first available design or component that would suit our needs, my team member would insist on further research. This was oftentimes frustrating but occasionally led to benefits such as a cost savings or an option that I had initially overlooked. Completing any given task in this manner took much longer than I would have liked, but the extra work put into thorough research, design, and planning would pay off in the end.

## 5.4 Eric Lee

A long time ago, four of us set out to propose, specify, design, and build a chair that adjusts to a seated person's body. In the genesis, three of us were forced to find a new idea because our previous idea, a license plate recognizing imaging device for parking enforcement, had already been done. From our pool of ideas, the chair was chosen as the simplest and most doable project. None of us would comprehend the enormous journey our project would become; at the outset, the chair seemed to be a reasonably tractable problem that we would solve in due course. What follows is a reflection on the biggest misjudgement of my university career.

My individual contributions to the measurable project deliverables were varied and many. I was the principle designer for the air system: getting air in and out of the lumbar support in a controlled manner. Together with Stephanie, I investigated the options for sensing and determining the optimal lumbar support height. I created the big picture of the overall electronics system. I also designed every schematic except for the solenoid board and the switch debouncer section of the UI board (the schematics totalled 19 pages). Eric was helpful in verifying and debugging my schematics. In terms of implementing those schematics, I soldered the first motor board, in addition to minor miscellaneous electronics such as switches, connectors, and the pressure sensor. On the software side of things, I was a major contributor to the high-level design of the overall software together with the rest of the team. I was then responsible for the bulk of the detailed low-level software design, also serving as the primary source of verification and feedback for others' low-level software designs. I implemented the majority of the low-level supporting software, starting with the initial software interaction with the electronics, then continuing on to the self-test mode, startup code, motor remote control mode, error logging, error log viewing mode, and HAL. For the main software implementation, I was involved to varying degrees with everything except for the LCD display and lumbar height tasks. I also performed most of the debugging of both mine and others' code. With regards to project documentation, I contributed my assigned sections and participated in the final assembly and proofreading of each document. I was also the main contributor to the infamous SolidWorks depiction of the fully assembled chair in our design specification.

Shifting to the immeasurable items, I helped source parts together with the entire team. As the major source of upfront money, I placed most of the online orders and kept track of the accounting. I also spent much time picking up deliveries, tracking shipments, and dealing with suppliers for mistakes and invoice documentation. I found many of our external assistance contacts, including some of the product samples, the electronics design help, and the third party soldering assistance of two motor boards. For the management aspects, I was responsible for much of the long-term planning of tasks and the overall direction of the project. I was also responsible for the short-term assigning of day-to-day and week-to-week tasks during the software development stage. For much of the project, I was forced to perform the numerous small tasks for general organization and to keep things running smoothly. This also often included motivation and progress tracking of individual group members and the team as a whole.

Carrying out these contributions has taught me many things ranging from positive skills to negative realities of life. Completing the tangible project deliverables has endowed me with many new hard skills. I had never designed an electronics project of this size, and to my great satisfaction, it worked without much revision. I gained experience using electronics parts including: polyswitches, optoisolators, AVR ATMega128 microcontroller, bus switches, and pressure sensors. I gained an understanding of basic air systems and their major components. I had never participated in a team software project of this size. This was my first experience with an RTOS on an 8-bit microcontroller, and also my first extensive use of C in an embedded application. I came to understand the team writing process and learned to use SolidWorks, especially the “SolidWorks magic glue.” I did not build any of the mechanical elements on the chair, but witnessing my team’s efforts has shown me the pros and cons of ad-hoc hodgepodge mechanical construction.

For soft skills, I learned firsthand the amount of work, patience, and knowledge that goes in to properly ordering, shipping, and receiving parts. Vendors sometimes play expensive games with your money and in the end, shipment by air is often cheaper (and obviously faster) than shipment by ground. Group organization and project management are ripe with opportunities for unplanned work. Although it may seem like there’s nothing to do, extra tasks can suddenly appear that will sap an entire day. Keeping everyone else busy with appropriate work, while doing your own work, is a challenge. Ordering tasks and assigning the best person to the job, while avoiding bottlenecks and dependencies, is extremely difficult. Being the leader is not always desirable.

This project has taught me scores of hard lessons. First and foremost: things take time, and what may seem trivially simple still takes a long time to do well. Second, the design process on paper and computer can infinitely perpetuate itself. One can never know if something works, or has an effect, until a real-world implementation is made. Eventually there is a point where one must commit to a design and hope it works. But that is not meant to discount the value of thorough designs; we had a relatively simple time debugging because our initial designs were sound. There were a few instances where concerns and problems in the design stage never materialized in the actual implementation. Unfortunately, there were problems and we did not have the luxury of multiple hardware revisions, except for the second and third motor boards (they had incremental improvements over the original). Thankfully, most of the problems could be solved with some hacking, kludging and taping (in true engineering fashion). Third, external contacts can often be flaky. Unless one is already all but agreeing to buy, receiving a future call or email with further information is not something on which to rely. Companies are generally not fond of entertaining queries for more information. It's hard to become a domain expert in 3 days of phoning, reading, and searching. On the flipside, the rare company will latch on and incessantly send advertisements at the slightest hint of an interest to spend money. Fourth, the 9-5 world of most companies does not fit well with class schedules. Phone communication is hard and days are wasted playing phone tag. A few times we easily burned 2-3 hours for 3 days in a row on the phone calling around, trying to learn about and source parts. Working with companies in eastern time zones just makes everything harder. Fifth, building a project on a constrained budget is awful. Not only because of lesson three, but also with no money, one often turns to scrounging, begging, and taking what is available. Proper design and implementation is much harder with parts that were not intended to be used together. Sixth, a lockable, secure space in which to store materials is at a premium. Finally, interdisciplinary teams add more variance to an already volatile mixture. It is not recommended to have an interdisciplinary team where group members are working toward different goals and on different timelines.

Having covered my contributions, knowledge gained, and lessons learned, all that remains are my reflections on the past thirteen months. Our group is the final group in the final offering of ENSC 340. This is somewhat fitting considering we have broken new ground in nearly all aspects of this project, good and bad.

The most glaring aspect of our project is the fact that we took thirteen months to complete it. But in hindsight, our hyperextended timeline was somewhat predictable. Looking at our project proposal and functional specification, it is clear that trouble was brewing. Given the timeframe, our target was overly ambitious to the point of lunacy. Compared to other projects, our budget was huge! However, we were wide-eyed, gung-ho, keener students eager to make something cool. Many of the domain experts we consulted in the early stages suggested to make the project simpler and to only focus on one or two adjustments. We ignored them, and this is reflected in the design specification. If we had listened, the project design would be markedly different. We began with eight adjustments, then moved to five and finally implemented three. However, our designs reflect the intention to have many adjustments. If we had designed for three adjustments from the beginning, our project would have been much simpler and completed in less time. Shooting for the stars and hitting the moon is a misguided approach. It is better to shoot for the moon and hit the moon.

Compounding our overly ambitious targets was a lack of funds. The time, money, quality tradeoff was apparent. We did not sacrifice quality, and we had little money, so our project took a long time. Our money situation was the worst by being unknown. We had a floating budget, so we were always trying to save money, never knowing when it was enough. This manifested itself in weeks upon weeks of sourcing surplus and inexpensive parts. It wasn't enough to figure out what parts we needed, we also had to find the cheapest versions of them. Many parts of the chair, especially the mechanical parts, were hand-made rather than purchased, in an attempt to save money. What we saved in money, we more than paid back in time. It would have been much better if we had a fixed budget. A fixed budget allows for proper planning and the best choice given the circumstances. If we had extra money, then we could save time by buying more pre-made things. On the other hand, if we were severely short funds, then we would reduce the feature set.

Our team consisted of four engineering students, with some help from a kinesiology student and an industrial design student. Before I discuss my fellow engineers, some comments on interdisciplinary teams are warranted. Working with Russell and Daphne had its good and bad sides. Overall, I think they got the short end of the stick as our actions threw a wrench into their course work. I believe their course involved the analysis and evaluation of a product from an ergonomics and usability point of view. The original intention was that we would have something ready for them to evaluate by the middle of November. Clearly, we missed that deadline. To complete their reports, they instead analyzed the concept of the ACE Chair. Part of their tasks included a user survey and they described how the chair would operate, rather than having the real chair. Not surprisingly, this didn't work too well. From our perspective, they provided useful information and suggestions on ergonomics, usability, design mechanisms, and product appearance. However, some of their ideas, while good in theory, were way above the level of our implementation capabilities. There was some conflict as they began to realize the limitations in our abilities. Overall, our courses should not have been so closely linked. It would have been better if we had just consulted them for advice, rather than have the chair as the topic of their work.

This project has brought me the pleasure of becoming better acquainted with three decent people. It's true that the project would not be possible without my teammates. Each person on my team was a source of joy, but also a source of disappointment. The chair has brought us together, but the chair has also driven us apart. As the reluctant leader of the group, I am obligated to review each person.

Eric and I had previously worked together on an extra-curricular project that was extended into a co-op term and three engineering competitions. Much of the technical know-how leveraged in this project was gained in that previous project. Thus, going into the course, I was confident in Eric's technical prowess. He delivered, with major contributions to the mechanical and electronics. His programming skills were weak, but I did not expect Eric to do much programming (he's an engineering physics student). I am disappointed in Eric because of his lacking participation in leadership and management. I had expected him to share the load, if not completely take over running the group, at least for part of the time. Numerous appeals were made, but he never stepped up. Thus, because he would not handle the organization, planning, and communications, I was saddled with the extra burden for most of the time. Eric receives a B.

Prior to this project, I did not personally know Jennard. However, I was in ENSC 460 with him during the summer semester. Whenever I saw Jennard, he was working on or finishing his assignments early. Jennard is also a senior computer engineering student nearing graduation with a few previous co-op terms. I expected Jennard to be a hard worker with large contributions to the software. He certainly demonstrated his dedication to the project, and things would sometimes get amusing during an overnight programming session. I could usually rely on Jennard to show up and give a good effort, although sometimes his presence required some prior verbal encouragement. I am disappointed in Jennard because of his demonstrated mediocre programming abilities. His designs were flawed and required many iterations of review and revision. When it came to implementation, he required clear instructions with lots of guidance. C is not his language, and some OS concepts would seem lost on him. Debugging and problem solving are not his strengths. In the late days of the project, he would generate code that would have bugs. Then he would give me a sad puppy look, commit his changes, and stumble off to co-op, knowing that I would make it work before his return the next evening. I had the extra burden of debugging and solving his problems, providing a safety net for him that he never returned for me. Jennard receives a C+.

Before September, Stephanie and I had known each other, but not on an intense working level. We had been in a group before, but that was three years ago. Similar to Jennard, Stephanie is also a senior computer engineering student nearing graduation with many previous co-op terms. I expected Stephanie to take on a major role in the software. She had not worked with the AVR before, but I intended to present her with a standard C environment (i.e. I would take care of the hardware details). Progress was good during the first and second semesters, and the beginning of the third semester. She and I developed most of the high-level software design, working out the general algorithms, flowcharts, and interactions. But then when it came to the low-level design and actual implementation, her work ethic vanished. Unfortunately, I am disappointed in Stephanie because of her poor work ethic. In the end, her contribution to the programming effort was minimal. For the last semester, she was not a team player, she was not dedicated, and she was not committed to the project. Completion of the project was not her main priority. She even took a tropical vacation. There was much effort required to bring her back into the fold, but her motivation was never truly restored. She never worked at the pace required to finish the chair at the end of September. I had the extra burden of completing her work, as she was not doing it. Stephanie receives a D.

The bulk of development, integration, progress and completed milestones occurred during the summer. People have a tendency to disappear, become busy, and slack off during the summer. I was guilty of this too, until I realized that the project would not get done at our lax pace. As the reluctant leader, group whip, and motivating force, when I didn't work, progress would slow down or stop completely. My teammates did not usually demonstrate initiative or self-motivation during the summer. Additionally, as I had a large technical role, my delays would eventually block others relying on my parts to complete their own tasks. Thus, I was forced to keep going. I have to mention the critical role of co-op flex hours. My company was extremely accommodating – basically I could set my own hours as long as I met my deadlines with a sufficient quality of work. To complete the chair, my life degenerated into a cycle of working overnight on the chair, getting to co-op in the early afternoon, leaving co-op in the evening, and arriving home for another night of work on the chair. Needless to say, this cycle was difficult, but it was required. It would be fair to expect the same of my group. Eric and Jennard sustained this pace for spurts, and Stephanie never reached this pace. Usually with a course, grades are the motivation for completion. However, with our sliding deadline and loosely defined target, the impetus was gone. At times, it was difficult to hold teammates accountable, causing great personal stress.

As people and friends, my fellow group members are excellent. But in terms of academics, I would not work with Eric again unless there is another person present to be the leader. I would not work with Jennard again until he gains substantial development experience and ability. I would not work with Stephanie again unless there is something to keep her committed.

Was it worth it? Would I do this again? In the near term, this project was a waste of everyone's time. It has delayed our graduation and at least for me, it has taken my life. This project's existence has thus far ruined my Christmas, New Years, spring semester break, summer semester, summer semester break, and co-op term. Currently, this project is a negative for me. Time will tell if that changes to a positive. For now, most of the good that has come from the chair could have been achieved with a much simpler project. But that's not the Eric way.

## 5.5 Description of Group Dynamics

Group dynamics within Accomodarsi Solutions followed the interactive organization structure. Each group member was officially an equal and we never clearly appointed a leader. As various tasks arose, the group would re-organize itself as necessary to place group members with the most appropriate tasks. Documentation was first performed individually to generate content, then as an entire group to integrate and proofread the separate parts. The design process was generally performed as an entire group and as small sub-groups. The large, overall design process was performed together as an entire group. Subsystems (air, mechanical, electronics, software, etc.) were primarily designed in pairs, with members assigned to tasks based on interest, ability and expertise. Implementation was performed primarily in pairs and as individuals working on separate modules. When working in groups, sometimes the group members would work collaboratively at the same time toward a common goal. Other times, group members would work individually, then verify and respond to each other's work. The group dynamics would best be described as “go with the flow,” or “flexibility when responding to rapidly changing circumstances.”

To facilitate group interaction and communication, Accomodarsi Solutions utilized various technologies. Having experienced the benefits from a previous class, the group has used a private SFU Caucus forum since the early stages of generating project ideas in the summer of 2005. When possible, the team held in-person group meetings. However, with each member attending a co-op work placement for some of the time, in-person meetings were not always possible. Instead, the team used instant messenger software for both individual and group conversations. As well, the team had a group e-mail address that would send a copy of the message to each member’s personal e-mail address. The software was developed entirely on a CVS server, facilitating distribution of updates, and identification of changes to the software. In the beginning, the group experimented with shared online calendars, collaboration and project management websites, but for various reasons, these technologies were not used.

In general, being a part of Accomodarsi Solutions was a positive experience. At the beginning, the group members did not all know each other well. Working together fostered social relations and the group members became friends. Most conflicts and disagreements were related to the design or implementation of the project, and not due to group dynamics. These philosophical differences were usually resolved through discussion and a comparison of the different approaches. However, not all was perfect with the team group dynamics. Because roles were not formally assigned, accountability was a problem and undesirable tasks were neglected or performed poorly. This often meant that meetings would go overtime and sometimes not everything that should have been completed would be completed. As well, because we had no official group hierarchy, the lack of accountability was compounded. The lack of accountability was a direct contributor to the additional time used to complete the project. Overall, group dynamics was not a problem for Accomodarsi Solutions.

## 5.6 Recommendations For Similar Projects

What would we do differently next time if we do a similar project? The answer to this is obvious: do not propose a project that will take a year to do. We were naïve to think that the project we proposed could be designed and implemented in four months.

First, we tried to do too much. In addition to the armrest height, lumbar height and lumbar size, we originally proposed to automatically adjust the armrest width and seat depth, as well as have lateral supports on the backrest. Since we made those adjustments part of our functional specification, we had to design them and include them in the design specification. Designing all those auto-adjustable parts took a while and we found that we didn’t have time at the end of the semester to implement them. We should have just concentrated on two or three features, which is what ended up happening when we began to implement. Proposing three features earlier in the semester would certainly have allowed our group to present in less than a year. For a different project, we should focus our goals more.

Second, during the spring semester, we should have forced each other to adhere to a strict schedule. All four of us were lax in terms of keeping each other up to task. Things did get done but some deadlines were missed, leading to another deferral for another semester to finally finish the chair. If we had followed the schedule that we originally set, we would have been done sooner. Since everyone had other things to do, we should have agreed on a certain number of hours each week that we will only do 340 tasks and nothing else. That way, we can concentrate on doing the project since we all agreed to spend a specific amount of time. The way we did it was agreeing on meeting a certain day or days each week on a week-by-week basis. This did not work too well since our schedules were different and we did not agree on what days we should keep free beforehand. If we do another project, everyone must commit a regular number of hours per week so that deadlines are followed. Penalties for missing commitments should be well-defined and agreed upon at the beginning of the project.

Third, our floating budget caused many delays in terms of sourcing and ordering parts. We should have had a fixed known budget from the beginning. A fixed budget would have reduced our timeline whether it was smaller or larger than our actual budget. A smaller budget would have helped limit the scope of what was attempted. On the other hand, a larger budget would have simplified the design and implementation process because we would have used more off the shelf parts.

Fourth, somewhat due to the design of the course, we did not have clear expectations for project scope and complexity. We should have insisted on clearer directions from the instructor regarding which features were expected to be completed. As well, we should have better understood the amount of work we were taking on.

Fifth, we should not have so closely-coupled our project with the coursework of the Kinesiology and Industrial Design students. Timelines between the three courses did not always match up nicely. In the end, our project delays were detrimental to the outcomes of their courses. However, the students provided valuable expertise on areas in which we had no previous knowledge. We should have consulted them, but we should not have become a part of their lives.

Finally, we spent an immense amount of time in the design stage for each subsystem in the project. We could have saved time by reducing the thoroughness of our analysis, proceeding to the implementation sooner. However, the time spent designing greatly reduced the number of serious problems in the implementation stage. At some point, designing must stop and implementation must begin, but it is worthwhile to properly perform the design stage.

# 6 Conclusion

Accomodarsi Solutions is pleased to present their implementation of the ACE Chair prototype at the end of a thirteen-month design and implementation cycle. The working ACE Chair enables a user to easily adjust the armrests and lumbar support to ensure maximum comfort. The prototype successfully demonstrates that automatic adjustment can reduce user error and improve ergonomic fit.

This document has addressed the current state of the ACE Chair and suggested feature enhancements. Deviations from the original design were explained, and challenges in design and implementation were discussed. We then compared our actual budget and schedule to the ones initially proposed. Each team member reflected on their experiences working on the project and with each other.

Working on this project has been a valuable experience for everyone. At the project’s end, each team member is glad and relieved to walk away with a wealth of knowledge about ergonomics, mechanics, electronics, and group dynamics. At the same time, we mourn the loss of a year that could have been spent on other learning opportunities, snowboarding, anime-watching, curling, and other pleasant activities.

# 7 References

[1] Dictionary.com, *Dictionary entry for "ergonomic"*, 2005. [Online]. Available: http://dictionary.reference.com/search?q=ergonomic. [Accessed: September 17, 2005].

[2] Accomodarsi Solutions, "Proposal for an Auto-Conforming Ergonomic Chair", Simon Fraser University, Burnaby, BC, Canada, September 2005.

[3] Accomodarsi Solutions, "Design Specification for the Auto-Conforming Ergonomic Chair", Simon Fraser University, Burnaby, BC, Canada, November 2005.