

# ***Robotic Positioning System for Breast Radiation Treatment***

**Final Technical Report**

**May 4, 2009**

**EECS 398/399**

**By:** David Ramsay, Michael Spence, JaeJun Lee, Sangbum Kim (MECH Dept.)

**Advisors:** Professor Vera Chankong, with Dr. Jason Sohn and Samuel Byeongjun Park

## **Executive Summary:**

Currently, breast cancer patients must be positioned manually in the radiation room in order to properly align them for treatment. These patients lie on their back, and are correctly positioned by adjusting their seat in two possible ways: by changing the angle of their back support, or by moving the bench toward or away from the radiation machine. The actual radiation beam can be shifted along one axis, rotating around the patient in order to obliquely target the tumor with minimal radiation exposure to the rest of the patient.

The image analysis software used to determine beam placement and radiation intensity for treatment also provides the user with an appropriate pitch, roll, and yaw of the patient relative to the center of rotation of the radiation beam. These extra data, which provide more accurate targeting of the tumor while further reducing patient exposure, are currently ignored in medical facilities on campus because the patient couches do not have the capability to move along these axes.

This final result of this project was a prototype couch that attaches to the existing patient bench and is capable of pitch and roll motion (yaw to be included in future prototypes). The pitch and roll of this seat is controlled by linear actuators, which are manipulated remotely through a computer interface. A program and graphical user interface has been developed so that the couch will automatically position the patient using the pitch, roll, and yaw data, and then store that seat position in a patient database for reuse during future radiation treatments. Due to the mechanical design of this prototype, exact translation from calculated thrust cylinder movement to final couch movement has been hampered. The next prototype needs to address mechanical issues (outside the scope of the electrical engineers on this project); however, the control system and GUI are completed and prepared for future mechanical revisions.

# INDEX

Page	Topic
------	-------

2	1. Introduction
---	-----------------

3	2. Technical Specifications/Requirements
---	--

5	3. Methodology
---	----------------

10	4. Results and Verification
----	-----------------------------

14	5. Administrative/Project Management Plan/Gantt Chart
----	---

18	6. Design Implications
----	------------------------

18	7. Conclusions
----	----------------

## Appendices:

19	i. Mechanical Drawings
----	------------------------

21	ii. Pin Details
----	-----------------

24	iii. Full Description for GUI design
----	--------------------------------------

27	iv. Graphical User Interface Narrative and Screenshots
----	--

34	v. Calculation of Coordinate Transform Including Programming Code
----	---

## 1. Introduction

The goal of this project was to create a robotic patient-positioning seat (couch) to be used with breast radiation therapy. The existing patient couch can only move in two ways—the seat back can change its pitch, and the base of the table can move translationally toward or away from the machine. Modern imaging analysis software provides a desired pitch, roll, and yaw for the patient couch in order to target the tumor more effectively and to reduce radiation exposure to the rest of the patient.

In order to create a patient couch that can achieve this extended motion, the project team used Solidworks to design a prototype. This first prototype incorporates the desired roll and pitch motion and is able to lock onto the existing patient table base. Instead of the manual adjustment relied upon by the current system, this couch uses two linear actuators in order to achieve its motion. In the early prototype phase, the chair is accurately scaled, but the materials are cheaper substitutes for the typical materials used in the radiation treatment facility.

Once a life-size chair was developed, the motors were interfaced with a computer via a Galil motor controller, and software was developed using C++ to control them. As part of another future senior project, this software will be refined to use the data from the image analysis software as inputs, and to move the patient couch accordingly. Since the software provides roll, yaw, and pitch values relative to the iso-center of the radiation chamber (i.e. the center of rotation for the radiation beam), it will be necessary to translate this input to a different center of rotation before sending any control signals to the motors.

After the team verified that the software control system was working correctly, the next step was to set up a database to store patient couch settings in between radiation sessions. Finally, a user friendly graphical user interface was constructed and tested to make the entire process easy and intuitive for the radiation technician.

The existing patient chair system is designed by Q-fit ([www.q-fit.com](http://www.q-fit.com)). The table base, shown below, is outfitted with their kVue IGRT Couchtop. This couch top includes a standardized indexing system used by other Q-fit products, which include the current breast radiation board. These indexing systems are installed on all the radiation machines at the Cleveland Clinic, in order to consistently align patients in different radiation rooms.



*Figure 1. kVue IGRT Couchtop Indexing System*

Since the patient couch must be customized to conform to this particular indexing system, and was not designed as a standalone unit, but as an addition that can be cheaply and easily attached to an existing patient table base, the only preexisting unit of its type is known as the Accufix Quest Breast Board (also made by Q-fit). This board, shown below, is made of carbon fiber, and includes head and arm support. As mentioned earlier, this system is only capable of manual pitch adjustment, at predefined angles of 1.25, 5, 7.5, 10, 15, 20, and 25 degrees. The new system will be fully automated, feature an extra axis of rotation (roll), and include continuous angle selection (pitch).



*Figure 2. Accufix Quest Breast Board*

The material selection and configuration of the couch were crucial to successful implementation of this project. Since this chair is meant to be easily transferred on and off the couch top in order to replace it with different chairs for other types of radiation treatment, maintaining a light, mobile design is of utmost importance. Also, metal cannot be placed near the radiation beam, in order to avoid altering beam path and/or intensity. This, combined with the desired functionality, is the most important known difficulty to address. The prototype design/material choices reflect these considerations, and ultimately the final material selection process will ensure a light, sturdy chair (using materials like carbon fiber and Teflon).

## **2. Technical Specifications**

The electrical engineers enrolled in EECS 398/399 worked in conjunction with a mechanical engineer in order to produce this interdisciplinary project. Since this couch is the first of many prototypes, this section includes a breakdown of several specification subsections: the physical prototyping goals entrusted to the mechanical engineer, the project goals of the 398/399 students for this semester, and long term project goals and specifications for the completed bench.

### *Mechanical Specifications- Patient Couch Prototype*

Semester Goal: Working couch prototype using common materials with an aluminum mobile mounting system for the motor control hardware. **Completed for purposes of this first prototype. Design**

**improvements needed on couch for structural integrity (load bearing) and accurate thrust cylinder to couch position calculations. Mobile mounting system completed successfully.**

Final Project Goal: Working couch based on prototype design, but built with carbon fiber and other light, durable materials in place of the materials used for the prototype

#### Couch Specifications – **Completed except load bearing**

- patient back support 76 x 45.5 cm
- patient leg support- 40 x 45.5 cm
- must secure onto Accufix index patient system for table
  - 2 pegs, holes in base of couch for pegs
    - (1) circular, 1.3 cm diameter, centered 8.65 cm from edge
    - (1) elliptical, 2.6 cm long, 1.3 cm wide, centered 23.5 cm from edge
- load bearing for individual's weight (at least 600 lb)
- motor-controlled movement in two planes (pitch and roll)
- motors cannot be near plane of radiation beam

#### Mount Specifications - **Completed**

- 38 x 35 x 27 cm
- light and sturdy
- easy mounting and access to all relevant control electronics
- handle heat generated by control units
- integrated power supply

#### *Electrical Specifications- Patient Couch Prototype*

Semester Goal: Construct a working control interface that allows the user to enter data straight from the imaging software and measure accurate motor response

Final Project Goal: Integrate the working computer interface into a easy-to-use GUI design with a full patient database, complete with patient treatment history and patient case detail

#### Interface Specifications - **Completed**

- Galil DMC-4040 4 motor motion controller
- Stepper Motors/Power Supplies (E-AC drives) with Encoders
- 3 Hall Effect Sensors per motor to check motor position
- Highly shielded wires for performing in high radiation environment

-Galil Visual C++ provided libraries as starting libraries for motor control

#### Software Specifications – **Completed**

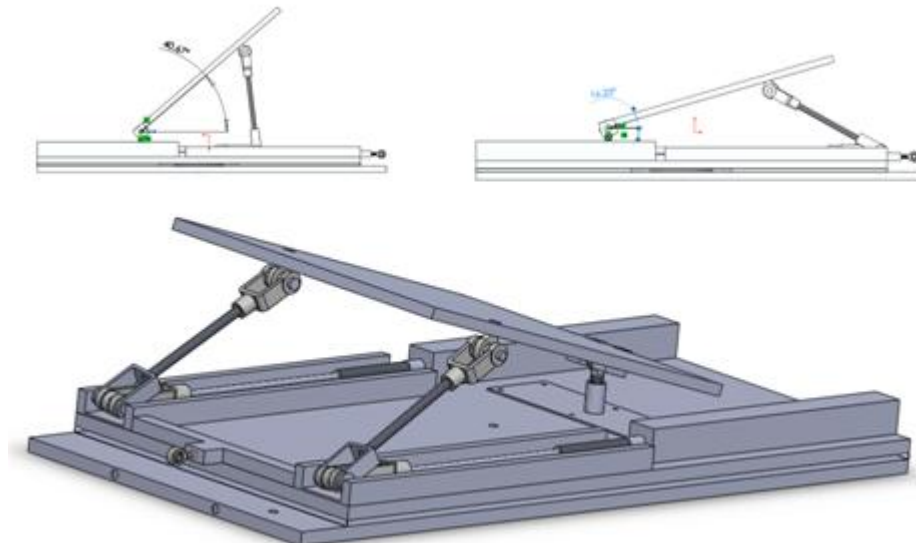
- easy to use graphical interface
- accurate calibration for motor control
- secure patient database with patient history and notes
- highly customizable default chair preset values for each patient
- advanced motor control menu, database maintenance features

### 3. Methodology

#### *Introduction*

This project consists of two main parts: a mechanical chair and a motor control system. The mechanical chair, which must rotate along two axes, was designed in Solidworks based on measurements taken from the existing system. The control system integrates the mechanical chair function so that it's controlled by an easy to use GUI on the technician's computer. Since the data used to position the chair is based on a coordinate system other than the center of rotation of the chair, there was significant engineering calculation involved in constructing a basic positioning application.

#### *Chosen Design*



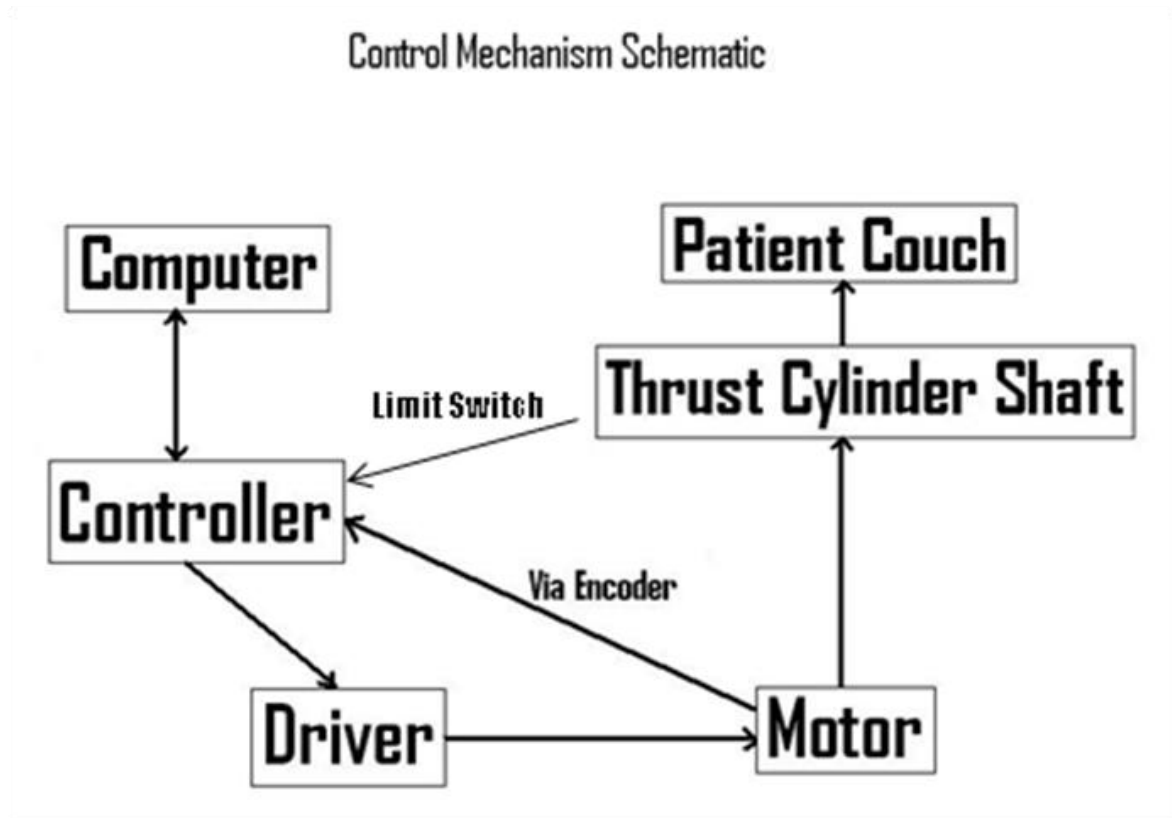
*Figure 3. Patient Chair Design*

For the physical couch, a final design was completed in Solidworks. After examining several options for the motor placement, the lowest motor stresses and furthest distance from the treatment area dictated the decision to keep all motors and thrust cylinders aligned along the base. As shown in the technical drawings above and in the appendix, the couch design is consistent with our technical specifications. It allows for a  $16.23^{\circ}$  -  $40.67^{\circ}$  angle variation in seat back pitch. In order to achieve rotation in all two axes, the motors were mounted in a specific configuration. Two motors control the seat back supports— by moving them in unison it is possible to control pitch, and by moving them asynchronously it can control the roll. The whole top section of the chair is mounted on a large bearing, which will allow it to rotate when it is pushed or pulled along this axis. Please see the technical drawings in the appendix for more detail.

The mount for the control interface was also discussed. This final control box provided space for and easy access to all the control hardware, as well as the capability of withstanding and dissipating any excess heat. Handles facilitate system mobility.

Due to hardship with connecting cables to motor drivers, three motor connectors and three driver connectors were constructed and mounted on the connecting panel made of 6 mm Acryl board. Also three power cables were made and combined to connect into one power socket used with a standard 10A AC power cable.

The final control system design (Figure 4 below) is made up of several different interacting components. All cables and other hardware were constructed to be prepared for the addition of a third motor in future prototypes. Computer control of the motor controller is accomplished using software libraries provided by the manufacturer (and used in the GUI in C++). The computer program downloads commands to the controller which then sends commands to the two motor drivers. These process and amplify the signal before sending the final output to the motors. The two motors drive the thrust cylinders on the patient couch and provide encoder feedback to the controller. The thrust cylinders that are used in this project do not come mounted with a homing switch or position sensor. Because of this, the team installed Hall Effect sensor limit switches at three positions on each shaft.



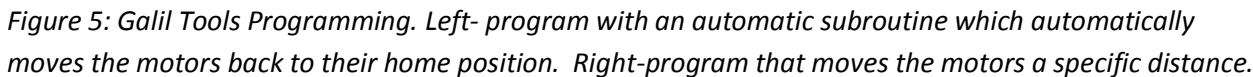
*Figure 4. Overview of System Setup*

The project team has tested and verified the connection between the computer and the motor controller via Ethernet cable. The other connections between components require shielded cables to maintain signal integrity in a high radiation environment. This has resulted in custom soldered cables, because the manufacturer does not make a cable that fits these specifications.

Six custom D-Subminiature cables were built: two 15-pin connector-side encoder/controller cables, three 15-pin driver side/44 pin controller cables, and the I/O 44 pin connector for limit switch inclusion. This involved synthesizing information from multiple manuals, obtaining and customizing materials and a steady soldering hand. Pin details including function and color scheme can be found in the appendices.

The computer control of this hardware interface was initially started with manufacturer provided software, called GalilTools. The team developed two main programs for use with the controller, one to simultaneously move each axis a specific distance, and one which automatically returned each of the two motors to a home position, moving them back toward the fully contracted position until a Hall Effect Limit Switch was triggered to indicate full retraction. Please see below for GalilTools code:

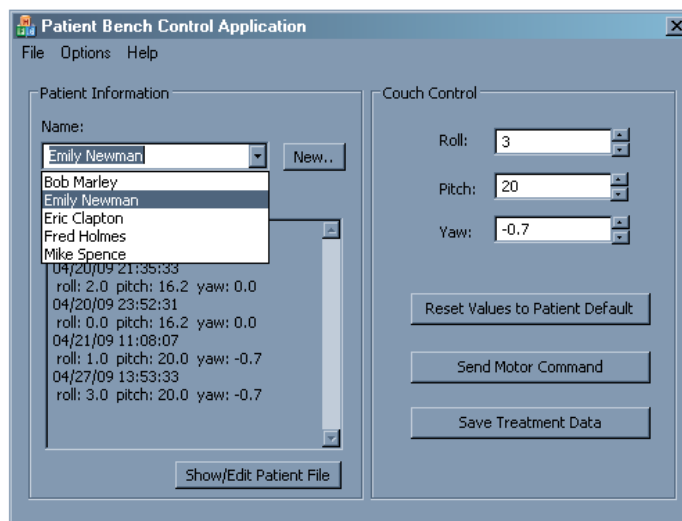




Page 8

prompts them to see if they want to save the latest data. Each patient also has a patient file, which can be viewed and updated by clicking the “Show/Edit Patient File” button. Clicking the “Reset Patient Defaults” button will reset the pitch, roll, and yaw values, and clicking the “Send Motor Command” button will move the chair to the values calculated using the roll, pitch, and yaw fields.

The menu and other functions of this control application can be found in detail in the appendix. In short, they include (1) an extensive menu for selecting patient chair defaults, (2) a way to manage the patient database, (3) patient file/treatment file editing screens, and (4) the add a new patient dialog.



*Figure 6: GUI Main Application*

Patient information is stored in a database folder, with each patient having their own archive file. This database structure is created using the CArchive class of the MFC library.

Integration of the software control programs created in GalilTools into the GUI interface was the last stage of the software development. Galil provides an example program, called DMCPP, which includes the appropriate class libraries to interface with the motor controller. In order for the Visual C++ libraries to work, the controller must be installed into the ‘controller registry’ using DMC net, another part of Galil’s software bundle. Once the controller was responding to the DMCPP software, the team worked to integrate the class libraries into the existing patient database.

This integration, after numerous stages of development and testing, was completed. The software, upon opening, attempts to connect to the controller. If the controller is not connected, the user receives the following error message before the GUI is opened:



*Figure 7: Controller Connect Error Message*

Otherwise, a message stating 'Controller Found' appears before the user accesses the main GUI screen. The software automatically uploads the homing sequence to the motor controller's buffer, and executes the algorithm. This starts both of the thrust cylinders moving backward toward their home position until they each hit their respective homing Hall Effect limit switches. After this is completed, any user click of "Send Motor Command" will upload another program (discussed earlier, and written in GalilTools software), which has a position command that is based on the given roll, pitch, and yaw values input on the main screen. The calculation that was used to determine this translation can be found in the appendix.

#### **4. Results and Verification**

The prototype board has been constructed from plywood, metal joints, and two motor units with thrust cylinders. The chosen motors are Parker Compumotor HV three stack stepper motors with encoders, which are mounted to 32 mm high precision thrust cylinders with a 300 mm stroke. This allows the prototype couch a 16° - 40° angle variation in pitch and +/- 5° angle variation in roll, accomplishing the goals of the project. The team verified these angle measurements were accurate to the initial design by physical measurement in the lab. The completed prototype is shown in Figure 8 on the next page. The couch is mounted on a secondary board, so that yaw motion is possible— though the thrust cylinder and motor were not available to control this motion at the end of the project. Mechanical drawings can be found in the appendices.



*Figure 8: Completed Prototype*

The control unit is constructed using a 20 x 20 mm aluminum profile system because of its lightness, sturdiness, and heat sinking capabilities. A 6mm acrylic board and metal joints are also used. More information of the control unit can be found in the 'Technical Specifications' and 'Methodology' sections. All the connectors are tested and proved to work properly.



*Figure 9: Completed Control Unit*



*Figure 10: Completed Control Unit with Cables Attached*

The control interface is complete. The final parts decided on for this interface were the Galil DMC-4040 4 motor control unit, the Carlo Gavazzi SPD 24301 power supply for the controller, and three Parker E-AC motor drivers. As described above, each unit connects to other units through 15, 25, and 44 pin d subminiature connectors. Since these are high precision motors in a high radiation environment, it has been necessary for the project team to use highly shielded cable. This cable cannot be ordered with the proper pin interconnections for this particular control system, so significant time has been spent to solder and verify the d-sub connectors used to hook up the control system. This process is complete for all cables. The cables have been verified against the interface documentation and have been electrically tested for shorts.

The Galil DMC-4040 motor controller is connected via cross-over Ethernet cable to a computer running both Galil's test software and the team's GUI, and all are functioning properly. The software interface has been implemented in Microsoft Visual C++ 2008. The Microsoft Foundation Class library was chosen for the basic GUI application design, and provided libraries from Galil are integrated to achieve motor control. Verification of this hardware interface took place in several stages. The motor driver includes a testing movement feature, so the team began by verifying that the driver to motor connection was working properly. The team then verified the computer to motor controller connection with a simple command to upload messages to the LCD display. Once this was functioning properly, the team attached the thrust cylinders, and tested their GalilTools software for position movement on the thrust cylinder. The conversion based on the documentation (using a resolution of 10000 ticks/motor revolution, 8 revolutions per linear inch) proved to be true, as the team measured thrust cylinder movement and compared it to the expected motion. Finally, Hall Effect sensor limit switches were attached, and verified using the GalilTools 'watch' window to make sure their input to the software was updating properly. The final Hall Effect sensor test included running the homing algorithm written by the team, which automatically moved the motors back toward the fully retracted positioning, stopping

each when their respective limit switches were triggered. This process works consistently and finalizes the verification process for the hardware portion of the bench.

Each step of the software functionality was verified as it was coded, to make sure that it worked appropriately. After everything was completed, a simulated patient database was input, holding twenty sample patients with various defaults and treatment histories. The team went through and verified that functionality was maintained with the large database, trying multiple scenarios that could result in errors. The control application functioned flawlessly throughout the testing. The patient database was maintained even when the software process was killed or the software was exited incorrectly. The database system is robust, and since patient data is stored in individual patient files, it is very difficult to corrupt anything more than a single patient file.

The integration of GalilTools with MFC C++ applications was verified in small steps throughout a long development process. First, simple commands were tested using the provided DMCP application provided by Galil, the motion controller manufacturer. Once this was responding appropriately and consistently, the integration into the GUI and patient database was started.

First, a simple message box was used to indicate a connection and communication between the software and the controller. Second, simple one line commands, using the provided function `DMCWin.Command(<commandstring>)` were attempted from within the software and accurate responses were measured. Finally, entire programs were downloaded into the buffer and tested repeatedly. Software commands to return the program from the buffer also ensured that the right algorithms were being uploaded through the GUI environment. Finally, the team tested several programs called from the same GUI instance, to make sure that the software would be able to run the homing algorithm, and overwrite that with the movement algorithm during the same program instance. This also was measured and worked as specified. The final version of the software includes a verified automatic connection to the motor controller, an automatic homing subroutine that will utilize the limit switches to move the motors back to their fully retracted position, and has been verified to issue different movement commands depending on the user input.

The coordinate system calculation was verified multiple times by hand, and transforms the data from a pitch and roll to a corresponding thrust cylinder length correctly. Checking through the code also reveals that it was coded properly, and responds correctly to user changes in pitch and yaw. Because this math provides only a basic approximation of the patient board, and variables in the mechanical design—like tilted support bars or offsets in the actual seat back from chair pivot points—cannot be accounted for accurately, the response of this coordinate transform is not as accurate as direct linear commands and corresponding measurements taken from the thrust cylinder. However, the coordinate transform is correct, and has been verified, and responds correctly to different user inputs. The resulting commands behave exactly as expected, issuing the correct linear commands given any pitch and roll inputs based on the measurements from the couch design. The problem is that a correct linear motion in the thrust cylinder results in a variation in pitch and roll from the expected pitch and roll values due to external variables in the mechanical design.

Fortunately, the team met all of its goals for this prototype stage, including a working and verified coordinate conversion. Variability and further accuracy of the coordinate system and mechanical design will be addressed in the next stages of this project when new a mechanical design is created and implemented.



## Semester End Verification Goals (Completed)

- Demonstrate motor controller interfaces with all motors
- Test Hall Effect Sensors limiting by ramping motors to their limits
- Show basic software control of motors
- Test motor repeatability by sending repeated command string and measuring cylinder error
- Include a basic GUI application that converts coordinate systems correctly for chair position
- Verify this coordinate conversion mathematically by measuring changes in cylinder length
- Construct a mechanical chair prototype that is capable of rotation in three axes
- Construct a control mounting unit for easy access and maneuverability of control hardware

### 5. Administrative

The project specifications and requirements evolved and new elements surfaced since first starting the project and writing the proposal. This was addressed by altering the goals and specifications for the semester as outlined in the interim technical report.

Two major endeavors that were not considered lengthy tasks have become very time consuming. These include design and construction of a mounting system for the control components and research and construction of interconnecting cables for the control system components. This has set the team back on the progress of other tasks at hand as seen in the Gantt Chart and Task List on the following pages.

The first unexpectedly long endeavor was constructing custom cables for the system control components. This involved obtaining wires and careful skillful soldering to 15, 25, and 44 pin connectors, as well as wiring in expansion capabilities for the use of Hall effect sensors for limit switch control. Each connector went to a different control system component with its own corresponding user manual. These components have many uses and thus it proved to be difficult to discern the regions of interest within the manuals. This cable construction task took several weeks at the beginning of the semester.

The second task and additional requirement of designing and constructing a mounting system was undertaken by JaeJun. This took most of JaeJun's time resources and prevented him from working on other tasks as originally set in the management plan of the proposal and thus the plan was modified.

Other unexpected situations resulted in problematic delays, including problems with the initial motor control and the necessity of including Hall Effect sensor limit switches. In addition, when testing the Hall Effect sensors, the lab manager Samuel accidentally shorted out the motor controller. This set the team back by a few days at a crucial point in the last two weeks of the project. The team borrowed a controller and worked extra time to account for this.

General Roles:

**David:** GUI Design and implementation, Galil Tools programming, wire/hardware construction

**Mike:** Wire/hardware construction, motor control, project management and documentation, media management

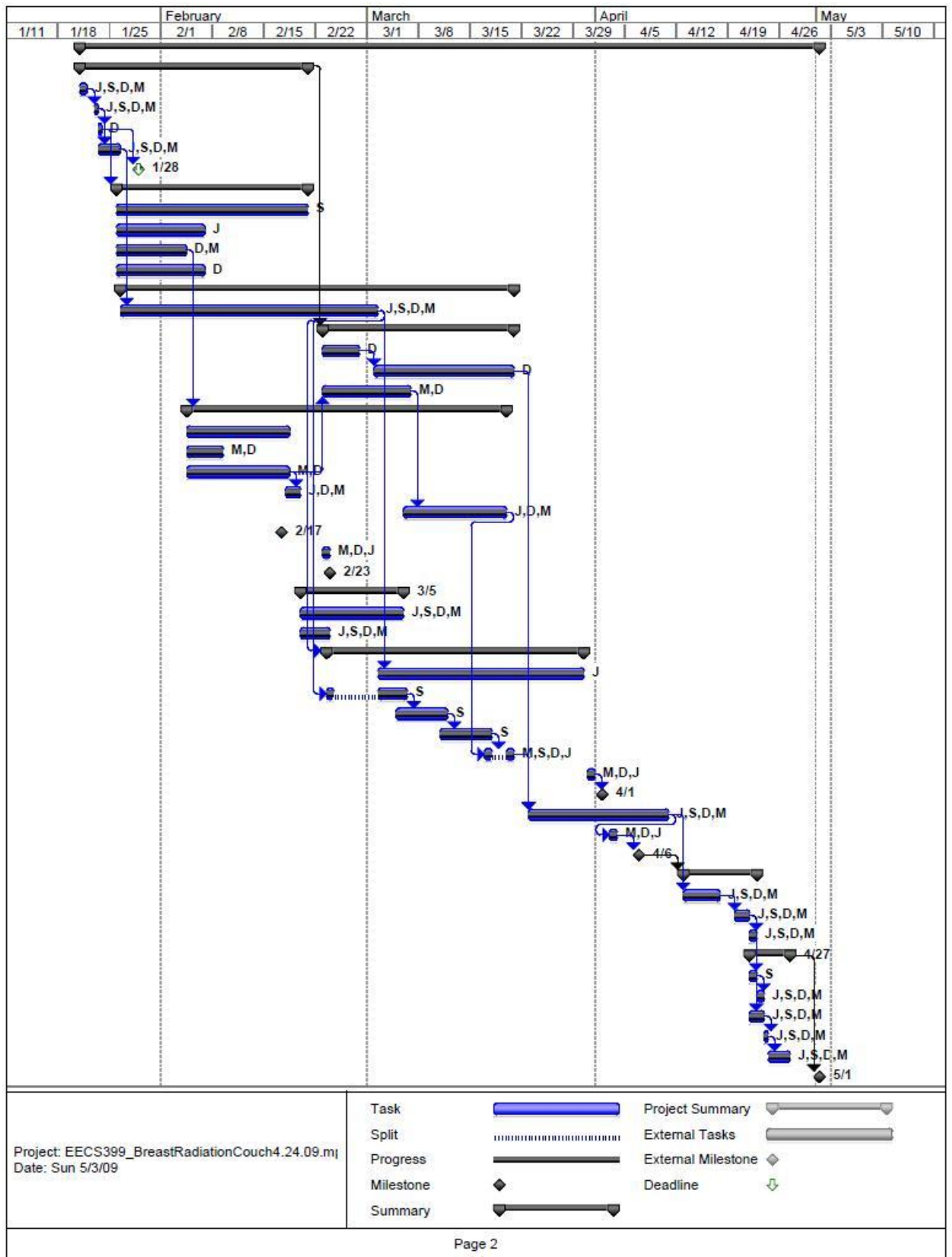
**JaeJun:** Mount construction and design, including wire construction for mount interface, couch construction with Sangbum

Sangbum: Mechanical engineer on the project, designed and constructed mechanical couch

The Gantt chart and Task List for the project can be seen on the following pages.



# Gantt Chart



## Task List

ID	Task Name	Duration	Start	Finish	% Complete	Predicted	Resource Names
1	Start	72.5 days	Wed 1/21/09	Fri 5/1/09	100%		
2	Preliminary Design	23 days	Wed 1/21/09	Fri 2/20/09	100%		M,D,S,J
3	Research Clinical Couch	1 day	Wed 1/21/09	Wed 1/21/09	100%		J,S,D,M
4	Meet with Technical Advisor	0.5 days	Fri 1/23/09	Fri 1/23/09	100%	3	J,S,D,M
5	Project Concept Document	0.5 days	Fri 1/23/09	Fri 1/23/09	100%	4	D
6	Evaluate existing technology	1 day	Fri 1/23/09	Mon 1/26/09	100%	4	J,S,D,M
7	Write Project Proposal Document	3 days	Mon 1/26/09	Wed 1/28/09	100%	5	D,M
8	Prepare Proposal Presentation	20 days	Mon 1/26/09	Fri 2/20/09	100%	5	M,D,J
9	Soldworks Drawing of Couch	4 wks	Mon 1/26/09	Fri 2/20/09	100%		S
10	Design of Mounting System	2 wks	Mon 1/26/09	Fri 2/6/09	100%		J
11	Rough Design Control System - Hardware	1.5 wks	Mon 1/26/09	Wed 2/4/09	100%		D,M
12	Research Software Alternatives for Interface	2 wks	Mon 1/26/09	Fri 2/6/09	100%		D
13	Design Execution and Assembly	39.5 days	Mon 1/26/09	Fri 3/20/09	100%		
14	Research/Order/Wait for Parts and Materials	5 wks	Mon 1/26/09	Mon 3/2/09	100%	6	J,S,D,M
15	Design Software	20 days	Mon 2/23/09	Fri 3/20/09	100%	2	
16	Background Programming	1 wk	Mon 2/23/09	Fri 2/27/09	100%		D
17	User Interface	3 wks	Mon 3/2/09	Fri 3/20/09	100%	16	D
18	Control System Commands and Execution	2 wks	Mon 2/23/09	Fri 3/6/09	100%	22	M,D
19	Assembly of Control System	31.5 days	Wed 2/4/09	Thu 3/19/09	100%	11	
20	Control Mount Design	2 wks	Wed 2/4/09	Wed 2/18/09	100%		
21	Interface With Computer	3 days	Wed 2/4/09	Mon 2/9/09	100%		M,D
22	Construct Cables for Control System Connection	2 wks	Wed 2/4/09	Wed 2/18/09	100%		M,D
23	Connect Control System Components	2 days	Wed 2/18/09	Thu 2/19/09	100%	22	J,D,M
24	Motor Control Testing	2 wks	Fri 3/6/09	Thu 3/19/09	100%	18	J,D,M
25	Oral Status Report 1	1.33 days	Mon 2/16/09	Tue 2/17/09	100%		M,D,J
26	Update Gantt Chart and Prepare for Meeting	1 day	Mon 2/23/09	Mon 2/23/09	100%		M,D,J
27	Design Review Meeting	1 day	Mon 2/23/09	Mon 2/23/09	100%		J,D,M
28	Interim Technical Report	10 days	Fri 2/20/09	Thu 3/5/09	100%		J,S,D,M
29	Prepare Interim Technical Report - Write Content	2 wks	Fri 2/20/09	Thu 3/5/09	100%		J,S,D,M
30	Prepare Interim Technical Report - Figures	2 days	Fri 2/20/09	Mon 2/23/09	100%		J,S,D,M
31	Construction of Prototype 1	25 days	Mon 2/23/09	Mon 3/30/09	100%	14	
32	Control Mount Construction	4 wks	Mon 3/2/09	Mon 3/30/09	100%	14	J
33	Machine Plywood to Specifications	1 wk	Mon 2/23/09	Fri 3/6/09	100%	14	S
34	Connect Motors to Couch Base	1 wk	Thu 3/5/09	Wed 3/11/09	100%	33	S
35	Connect Bearings	1 wk	Wed 3/11/09	Tue 3/17/09	100%	34	S
36	Connect Control System to Couch	2 days	Tue 3/17/09	Fri 3/20/09	100%	35,24	M,S,D,J
37	Prepare for Oral Status Presentation	1 day	Tue 3/31/09	Tue 3/31/09	100%		M,D,J
38	Oral Status Report 2	1 day	Wed 4/1/09	Wed 4/1/09	100%	37	M,D,J
39	Advanced Programming/GUI	3 wks	Mon 3/23/09	Fri 4/10/09	100%	36,17	J,S,D,M
40	Update Gantt Chart and Prepare for Meeting	1 day	Fri 4/3/09	Fri 4/3/09	100%	39	M,D,J
41	Design Review Meeting	1 day	Mon 4/6/09	Mon 4/6/09	100%	40	J,D,M
42	Analysis of Prototype 1	8 days	Mon 4/13/09	Wed 4/22/09	100%	41	
43	Record and Analyze Data	1 wk	Mon 4/13/09	Fri 4/17/09	100%	39	J,S,D,M
44	Verify Effectiveness/Accuracy of Prototype 1	2 days	Mon 4/20/09	Tue 4/21/09	100%	43	J,S,D,M
45	Verify Performance to Specifications	1 day	Wed 4/22/09	Wed 4/22/09	100%	44	J,S,D,M
46	Final Oral Project Presentation	3.5 days	Wed 4/22/09	Mon 4/27/09	100%		J,S,D,M
47	Pictures/Video of Working Model	1 day	Wed 4/22/09	Wed 4/22/09	100%	44	S
48	Review All Data and Draw Conclusions	1 day	Thu 4/23/09	Thu 4/23/09	100%	47	J,S,D,M
49	Prepare Powerpoint Presentation	2 days	Wed 4/22/09	Thu 4/23/09	100%	44	J,S,D,M
50	Assign Parts for Discussion	0.5 days	Fri 4/24/09	Fri 4/24/09	100%	49	J,S,D,M
51	Practice Presentation	1 day	Fri 4/24/09	Mon 4/27/09	100%	50	J,S,D,M
52	Assemble and Compose Final Technical Report	4 days	Mon 4/27/09	Fri 5/1/09	100%	46	J,S,D,M

Project: EECS399\_BreastRadiationCouch4.24.09.mj  
Date: Sun 5/3/09

Task

Split

Progress

Milestone

Summary

Project Summary

External Tasks

External Milestone

Deadline

Page 1

## **6. Design Implications**

The robotic couch for breast radiation treatment will prove to be a useful advance in treatment delivery. Current manual adjustment systems are slow or unable to make small changes in pitch elevation, and no current technologies include the use of roll and yaw to move patients into optimum position. This prototype may prove to be the first step towards a large impact on how this treatment, both from a performance and operations optimization standpoint and through improved medical results of radiation therapy.

Manufacturing costs would depend on the number of units demanded by the market. The materials cost of constructing prototype 1 of the couch is near \$8,000. Construction of the cables is a very time consuming task, but with large class production and automation this could be significantly reduced. Component costs would decrease with a larger order size.

Because of the medical environment, for the final product must be made of appropriate materials for cleanliness and sterility. Wires must be magnetically shielded to avoid interactions with other devices in the treatment rooms. This does not significantly increase cost, but failure to pay attention to this could cause interference and thus possible inaccurate patient treatment.

## **7. Conclusions**

The mechanical and electrical design of the couch is complete for two axis rotation (roll and pitch). The electrical hardware as well as the software is prepared for the inclusion of a third axis of rotation in future mechanical prototypes.

Unclear specifications caused some initial reevaluations of the project plan and semester goals. The interim technical report defined specifications that would not be changed further; after this point, the work distribution was more easily managed.

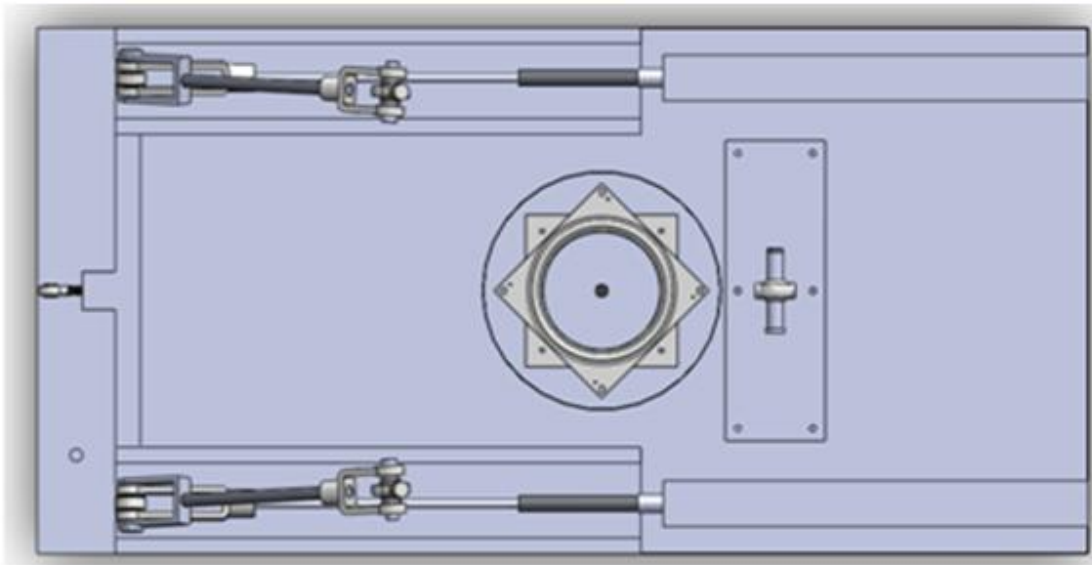
The specifications and goals of the project were met. The coordinate transform for roll and pitch movement is not as accurate as would be preferred, but a correct coordinate transform with corresponding correct linear motion has been implemented. Changes in the mechanical design of future prototypes will require alterations of these transform calculations, but if done intelligently it will prove to facilitate completely accurate coordinate transform.

This project has implications far outside of Senior Project class. The final product could improve results of radiation therapy for cancer patients. The current prototype needs further revisions and refinement, but it is an excellent starting place for a robotic position system for breast cancer radiation treatment.

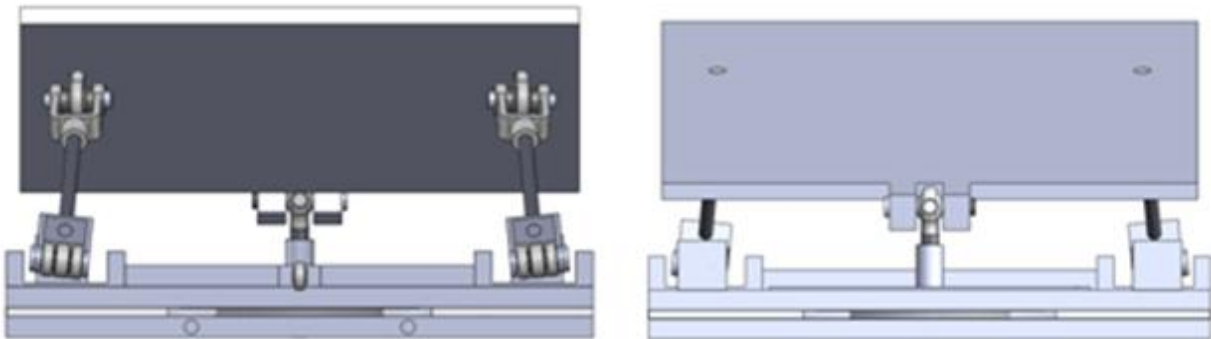
## Appendices

### i. Chair Model Technical Details

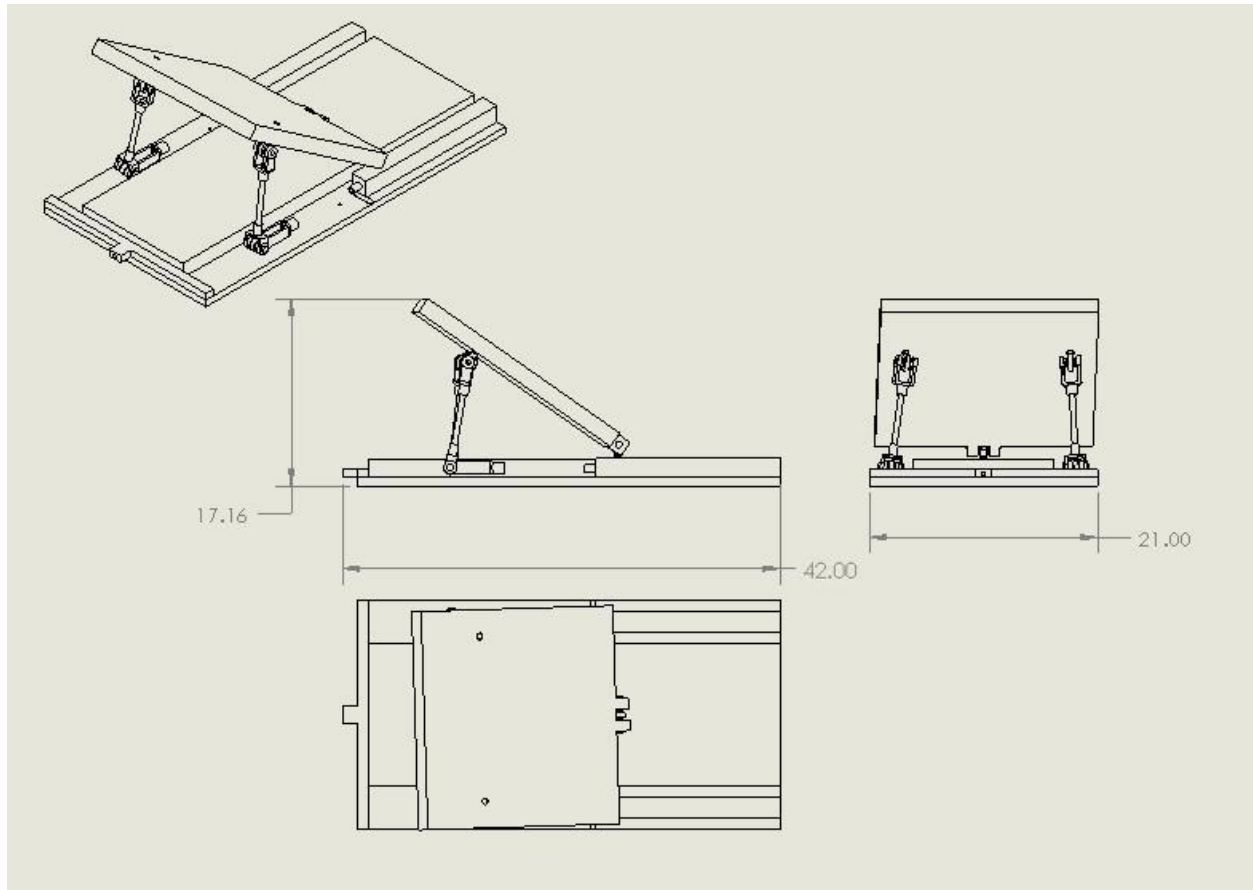
*Top View with Upper Layer Seat Removed*



*Back/Front View of Completed Chair*



### *Couch Rendering with Dimensions*



ii. Pin Layout Details

Driver Side of Driver/Controller Connection - 25 Pin

<u>Pin #</u>	<u>Function</u>	<u>Color</u>
1	Step+	Green
2	Direction+	Blue
3	NC	
4	NC	
5	NC	
6	NC	
7	NC	
8	NC	
9	Fault+	Red
10	NC	
11	Reset+	Orange
12	NC	
13	NC	
14	Step-	Black GND
15	Direction-	Black GND
16	Shutdown+	Brown
17	Shutdown-	White (GND)
18	NC	
19	NC	
20	NC	
21	Fault-	White (GND)
22	NC	
23	Reset-	White (GND)
24	NC	
25	B+	



## Controller Side of Driver/Controller Connection - 44 Pin

### ICM-42000 External Driver (A-D) 44 pin HD D-Sub Connector (Male)

Controller Side of Driver/Controller Connection - 44 Pin

Pin#	Label	Description	Pin#	Label	Description	Pin#	Label	Description
1	RES	Reserved	16	STPA	PWM / Step A	31	STPB	PWM / Step B
2	STPC	PWM / Step C	17	RES	Reserved	32	RES	Reserved
3	RES	Reserved	18	STPD	PWM / Step D	33	GND	Digital Ground
4	RES	Reserved	19	DIRA	Sign / Direction A	34	DIRB	Sign / Direction B
5	DIRC	Sign / Direction C	20	RES	Reserved	35	RES	Reserved
6	RES	Reserved	21	DIRD	Sign / Direction D	36	GND	Digital Ground
7	AENA	Amplifier Enable A	22	AEC1	Amp Enable Common 1	37	AENB	Amplifier Enable B
8	AEND	Amplifier Enable D	23	AENC	Amplifier Enable C	38	AEC2	Amp Enable Common 2
9	N/C	No Connect	24	N/C	No Connect	39	GND	Digital Ground
10	-12V	-12V from Controller	25	+12V	+12V from Controller	40	MCMA	Motor Command A
11	MCMB	Motor Command B	26	RES	Reserved / MCMDA_N <sup>1</sup>	41	RES	Reserved / MCMDB_N <sup>1</sup>
12	RES	Reserved / MCMD_C_N <sup>1</sup>	27	MCMC	Motor Command C	42	MCMD	Motor Command D
13	N/C	No Connect	28	RES	Reserved / MCMDD_N <sup>1</sup>	43	GND	Digital Ground
14	N/C	No Connect	29	N/C	No Connect	44	N/C	No Connect
15	+5V	+5V from Controller	30	N/C	No Connect			

**Encoder Side of Encoder/Controller  
Connection - 10 Pin**

<u>Pin #</u>	<u>Function</u>	<u>Color</u>
1	NC	
2	Ground	Green w/ White
3	I- (Z-)	White w/ Orange
4	I+ (Z+)	Orange w/ White
5	A-	White w/ Blue
6	A+	Blue w/ White
7	+5VDC	White w/ Green
8	NC	
9	B-	White w/ Brown
10	B+	Brown w/ White

**Controller Side of Encoder/Controller Connection - 15 Pin**

**ICM-42000 Encoder 15 pin HD D-Sub Connector (Female)**

Controller Side of  
Encoder/Controller  
Connection

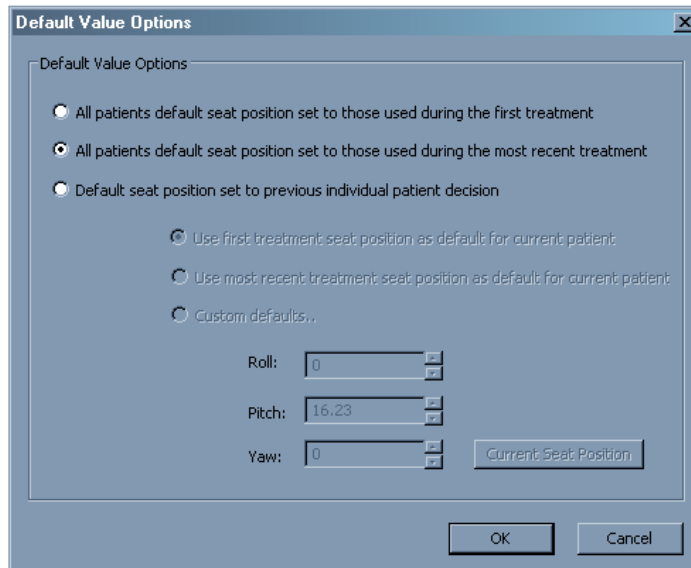
Pin #	Label	Description	Color Code
1	MI+	Index Pulse	Orange w/ White
2	MB+	B+ Main Encoder Input	Brown w/ White
3	MA+	A+ Main Encoder Input	Blue w/ White
4	AB+	B+ Aux Encoder Input	Brown w/ White
5	GND	Digital Ground	Green w/White
6	MI-	/Index Pulse	White w/ Orange
7	MB-	B- Main Encoder Input	White / Brown
8	MA-	A- Main Encoder Input	White w/ Blue
9	AA-	A- Aux Encoder Input	White w/ Blue
10	HALA	A Channel Hall Sensor	RED
11	AA+	A+ Aux Encoder Input	Blue w/ White
12	AB-	B- Aux Encoder Input	White w/ Brown
13	HALB	B Channel Hall Sensor	ORANGE
14	HALC	C Channel Hall Sensor	BLACK
15	+5V	+5V from Controller	White w/ Green



### iii. Full Description for GUI Design

The main menu has two main fields: File and Options. The File selection reveals an Exit option, which will close the program. The Options field reveals the 'Manage Patient Database' tab as well as the 'Default Value Options' tab. Each will bring up their respective screens:

#### Default Value Options:



Default Value Options

Default Value Options

- ☐ All patients default seat position set to those used during the first treatment
- ☒ All patients default seat position set to those used during the most recent treatment
- ☐ Default seat position set to previous individual patient decision

- ☒ Use first treatment seat position as default for current patient
- ☐ Use most recent treatment seat position as default for current patient
- ☐ Custom defaults..

Roll:

Pitch:

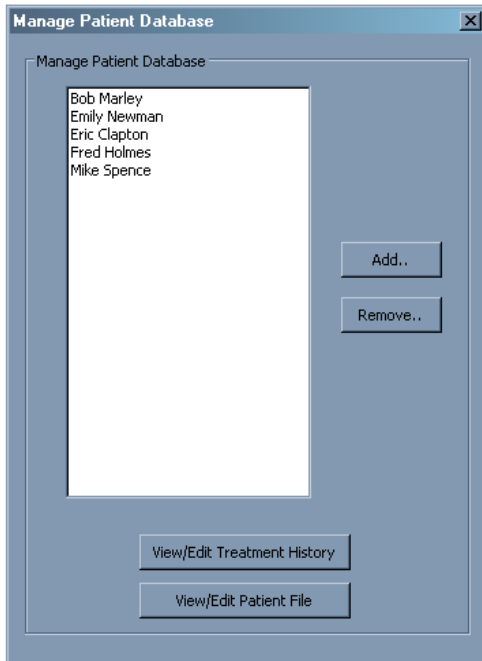
Yaw:

Current Seat Position

OK Cancel

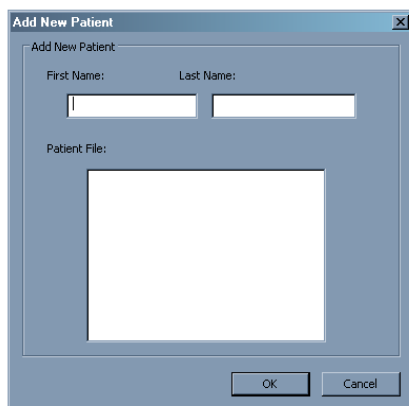
This option allows the user to select which values will be the patient chair default values. The user can choose for all patients to use their first treatment or their most recent treatment as their default, or can choose to make the defaults on a patient by patient basis. Individual patients may use their first treatment or their most recent treatment as defaults, or can have a custom default position entered by the technician on this screen.

### Manage Patient Database:



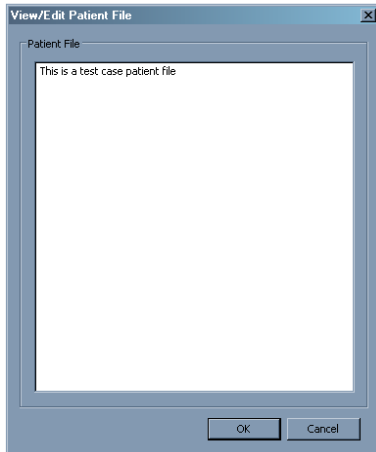
This screen allows the user to add or remove patients from the database, and individually edit treatment histories or patient files for every patient. Changes in the database are immediately reflected in the main application.

### Add New Patient:



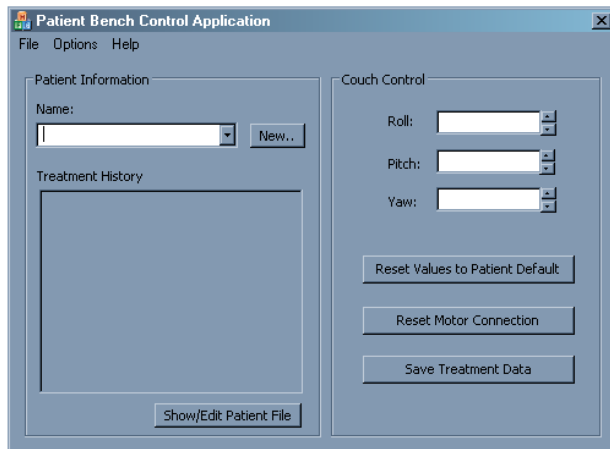
This is the add new patient screen, which initializes patients based on their first and last name inputs, as well as a patient file.

View/Edit Patient File, Edit Treatment History:



This is the View/Edit Patient File screen, which is the same as the Edit Treatment History screen except for the captions. This allows the user to view and edit the patient files and treatment files for patients in the database.

#### iv. Full Narrative with Screenshots for GUI Design



From Front Panel:

1. User Click "New"
2. Load Add New Patient Dialog

1. User Click "Show/Edit Patient File"

2. Load View/Edit Patient File/History Dialog using name in combo box

1. User Click "Reset Values to Patient Default"

2. Update Roll, Pitch, and Yaw Edit Boxes to default values using name in combo box, default indication variable, and corresponding database

1. User Click "Reset Motor Connection"

2. Update motor control commands by sending a shutdown override, and then re-homing the motors

1. User Click "Save Treatment Data"

2. Update patient treatment history file using couch values for name in combo box

1. User Change Patient Name from Name Combo Box

2. Check if Roll, Pitch, and Yaw edit box values are stored for previous name at the current date in the patient treatment file. If not, prompt to save couch preset values for previous name. If so, continue.

3. Load and display patient treatment history file for new name in combo box

4. Stop current motor motion

5. Update Roll, Pitch, and Yaw Edit Boxes to default values using default indication variable and corresponding database

1. User Click Arrow by Roll, Pitch, or Yaw

2. Update Value in Edit Box accordingly

1. Change in Values of Roll, Pitch, or Yaw Edit Boxes

2. Update motor control commands accordingly

1. User Click "X"

2. Check if Roll, Pitch, and Yaw edit box values are stored for current name at the current date in the patient treatment file. If not, prompt to save couch preset values for previous name. If so, continue.
3. End motor connection

From Main Menu:

1. User Click File->Exit
2. Check if Roll, Pitch, and Yaw edit box values are stored for current name at the current date in the patient treatment file. If not, prompt to save couch preset values for previous name. If so, continue.
3. End motor connection

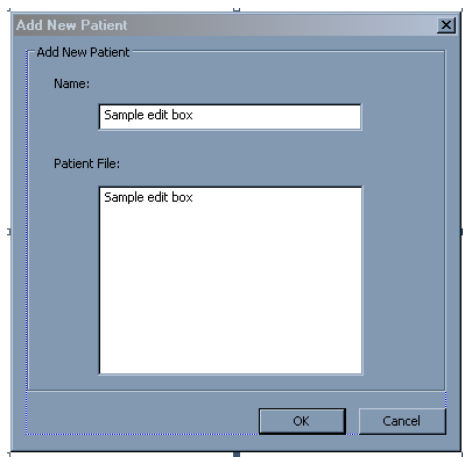
1. User Click Options->Manage Patient Database
2. Load Manage Patient Database Dialog
1. User Click Options->Advanced Motor Control
2. Load Advanced Motor Control Dialog

1. User Click Options->Default Value Options
2. Load Default Value Options Dialog

1. User Click Help->Help Contents
2. Open text file with help notes

1. User Click Help->Release Notes
2. Open text file with notes about specific version of software

1. User Click Help->About Control Application
2. Open About Dialog Box



From Add New Patient Dialog:

Load default text into edit boxes

1. User Click 'OK'

2. Check if name is different from all other names in database. If not, prompt if user wants to create a new patient file. If so, save name with '2' amended. If not, exit to main dialog. Otherwise, save name in patient database.

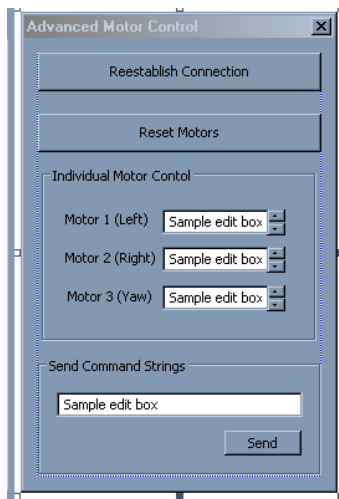
3. Save text from patient file in database associated with new patient name.

4. Save treatment history file for patient, with date of patient file being created.

5. Return to main dialog or patient database, depending on which called it.

1. User Click 'Cancel'

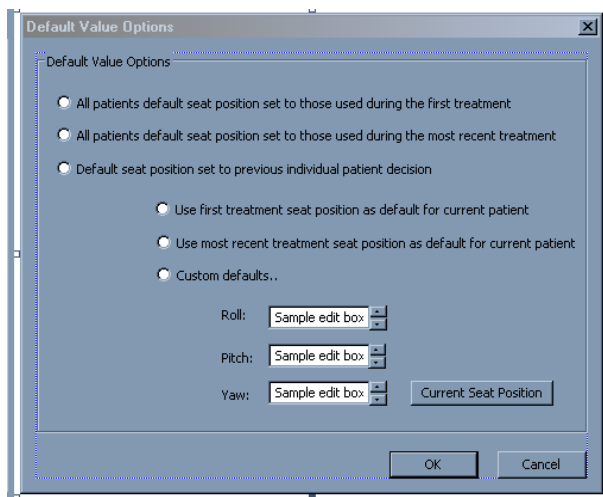
2. Return to main dialog or patient database, depending on which called it.



From Advanced Motor Control Dialog:

Use ActiveX controls for motor control

1. User Click 'X'
2. Return to main dialog



From Default Value Options Dialog:

Load main default variable; load individual patient default and custom values corresponding to name in combo box in main dialog

Only allow second set of radios and 'current seat position' button to be clickable if first set is selected to individual patient decision

1. User Click 'Current Seat Position'

2. Set Roll, Pitch, and Yaw edit boxes equal to corresponding values on the main dialog

1. User Click Arrow by Roll, Pitch, or Yaw

2. Update Value in Edit Box accordingly

1. User Click 'OK'

2. Store main default variable, individual patient default, and custom roll, pitch, yaw values.

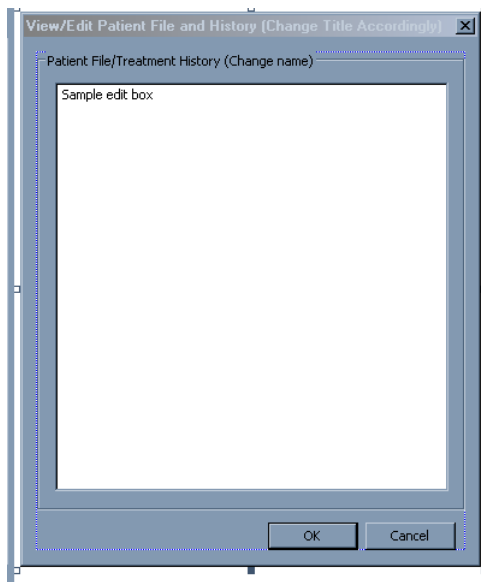
1. User Click 'Cancel'

2. Return to main dialog

1. User Click 'X'

2. Return to main dialog





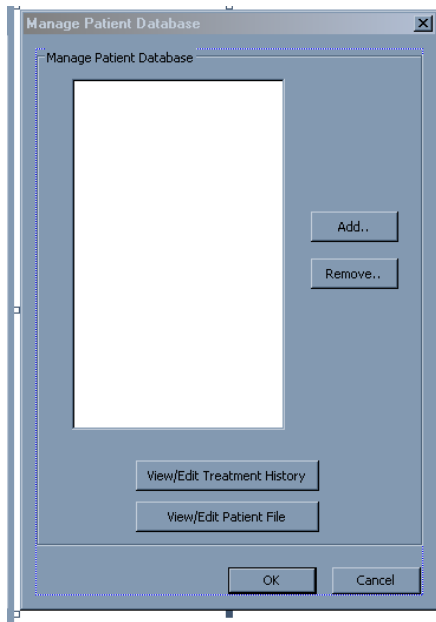
From View/Edit Patient File/History Dialog:

Load patient file/treatment history from file for current patient

1. User Click 'OK'
2. Save patient file/treatment history for current patient
3. Return to main dialog or patient database, depending on which called it

1. User Click 'Cancel'
2. Return to main dialog or patient database, depending on which called it

1. User Click 'X'
2. Return to main dialog or patient database, depending on which called it



From Manage Patient Database Dialog:

Load patient names from patient database

1. User Click 'Add'

2. Load Add New Patient Dialog

1. User Click 'Remove'

2. Prompt if user is sure. If not, return to this dialog. If so, continue.

3. Delete patient selected in list box from database.

1. User Click 'View/Edit Treatment History'

2. Load View/Edit Patient File/History Dialog

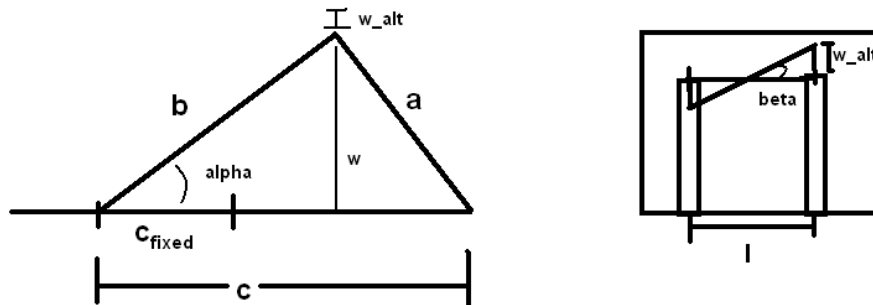
1. User Click 'View/Edit Patient File'

2. Load View/Edit Patient File/History Dialog

1. User Click 'X'

2. Return to main dialog

#### v. Calculation of Coordinate Transform Including Programming Code



The first picture represents the side view of the couch, with  $\alpha$  giving the pitch. The second view represents the back view of the couch, where  $l$  gives the distance between supports. Measurements from the solidworks model were used to find the appropriate values, so  $c_{fixed}$  was 10.25 inches, the seat back  $b$  was 15.7 inches long, the supports  $a$  were 11.9 inches tall, and  $l$  was 15.875 inches. From the thrust cylinder specifications, the portion of  $c$  beyond  $c_{fixed}$  could vary between 0 and 11.8 inches.

First, in order to take into account the roll, the team calculated an adjusted pitch value for the left and right thrust cylinders independently. This was done by calculating  $w$  using pitch input from the main window ( $=b*\sin(\alpha)$ ). Then, using  $l$  and the current roll input, an adjustment to the  $w$  height was calculated ( $w_{alt}=l/2*\tan(\beta)$ , where  $\beta=\text{roll}$ ). This value was then added to  $w$  for the right support and subtracted from  $w$  for the left, before being reconverted into an angle  $\alpha$  ( $\alpha_{left}=\sin^{-1}((w-w_{alt})/b)$ ,  $\alpha_{right}=\sin^{-1}((w+w_{alt})/b)$ ). Finally, using these new pitch values, a final value for  $c$  was solved for using the law of cosines:

$$0=c^2 + (-2b*\cos(\alpha_{corrected}))c + (b^2-a^2)$$

$$c= b*\cos(\alpha_{corrected}) \pm \text{sqrt}(b^2\cos^2(\alpha_{corrected})+a^2-b^2)$$

After solving for  $c$  and taking the appropriate root, a simple subtraction of  $c_{fixed}$  results in the distance that the thrust cylinder should extend. The actual portion of code (including the conversion to PWM commands, which gives a resolution of 10,000/revolution, 8 revolutions per inch) is shown below. It includes a safety feature of limiting commands sent to the motor to the intended range:

```
//resolution of motor set to 10000/rev, 8 revs per inch lateral motion
//thrust cylinder 300 mm stroke= 11.8 inches = 944500 full range of motion
double pitch= m_currpitch*(3.14159/180);
double roll= m_currroll*(3.14159/180);

double m_alength = 11.9;//chair support length
double m_blength = 15.7;//back length
```

```

double m_cfixed = 10.25;//cylinder fixed length fully contracted
double m_llength = 15.875;//length between chair supports

double m_clengthleft1 = 0;//cylinder side length
double m_clengthleft2 = 0;//cylinder side length
double m_clengthright1 = 0;//cylinder side length
double m_clengthright2 = 0;//cylinder side length

double m_cmovingleft = 0;//cylinder extension
double m_cmovingright = 0;//cylinder extension

double m_w = 0;//right angle vertical height, altered by roll
double m_walt = 0;//amount to alter w to account for roll
double m_newpitchleft = 0;//corrected for roll
double m_newpitchright = 0;//corrected for roll

m_w= m_blength*sin(pitch);

m_walt= m_llength*tan(roll)/2;

m_newpitchleft= asin((m_w+m_walt)/m_blength);
m_newpitchright= asin((m_w-m_walt)/m_blength);

m_clengthleft1=
m_blength*cos(m_newpitchleft)+sqrt(m_blength*m_blength*cos(m_newpitchleft)*co
s(m_newpitchleft)+m_alength*m_alength-m_blength*m_blength);
m_clengthleft2= m_blength*cos(m_newpitchleft)-
sqrt(m_blength*m_blength*cos(m_newpitchleft)*cos(m_newpitchleft)+m_alength*m_
alength-m_blength*m_blength);

m_clengthright1=
m_blength*cos(m_newpitchright)+sqrt(m_blength*m_blength*cos(m_newpitchright)*
cos(m_newpitchright)+m_alength*m_alength-m_blength*m_blength);
m_clengthright2= m_blength*cos(m_newpitchright)-
sqrt(m_blength*m_blength*cos(m_newpitchright)*cos(m_newpitchright)+m_alength*
m_alength-m_blength*m_blength);

m_clengthleft1== m_cfixed;
m_clengthleft2== m_cfixed;
m_clengthright1== m_cfixed;
m_clengthright2== m_cfixed;

if (m_clengthleft1>=0 && m_clengthleft1<=11.8)
    m_cmovingleft =m_clengthleft1;
else if (m_clengthleft2>=0 && m_clengthleft2<=11.8) m_cmovingleft
=m_clengthleft2;
else m_cmovingleft=0;

if (m_clengthright1>=0 && m_clengthright1<=11.8)
    m_cmovingright =m_clengthright1;
else if (m_clengthright2>=0 && m_clengthright2<=11.8) m_cmovingright
=m_clengthright2;
else m_cmovingright=0;

m_cmovingleft*=-80000;
m_cmovingright*=-80000;

```

```

if (m_cmovingleft<=0 && m_cmovingleft>=-944500 && m_cmovingright<=0 &&
m_cmovingright>=-944500)
{
    if (abs(m_cmovingleft-m_cmovingright)<210000)
    {
        m_cmovingleft-= m_motorleft;//m_cmovingleft now is the
appropriate move command
        m_cmovingright-= m_motorright;//m_cmovingright now is the
appropriate move command

        //send move command
        char szResponse[4096];
        m_DMCWin.Command("AB");
        char test[4096];
        sprintf(test,
                "#M\r\n"
                "YS=1\r\n"
                "MT 2.5, 2.5;\r\n"
                "DE 0, 0;\r\n"
                "DP 0, 0\r\n"
                "OP 0\r\n"
                "LU 0\r\n"
                "YC 1000, 1000\r\n"
                "YB 10000, 10000\r\n"
                "YA 1, 1\r\n"
                "KS 2, 2\r\n"
                "AC 2500, 2500\r\n"
                "DC 2500, 2500\r\n"
                "SP 100000, 100000\r\n"
                "PR %d, %d\r\n"
                "BG\r\n"
                "EN\r\n",int(m_cmovingleft),
int(m_cmovingright));

        m_DMCWin.DownloadFromBuffer(test);
        m_DMCWin.Command("XQ", szResponse,sizeof(szResponse));

        m_motorleft+=m_cmovingleft;
        m_motorright+=m_cmovingright;
    }
}

```