Northeastern University
Department of Electrical and Computer Engineering

MRI/MEG Compatible Grip Force Dynamometer
for Stroke Patients

Capstone Design Final Report
April 17, 2007

Capstone Group Members:
Eric Krejci (leader)
Ben Pinkus
Jason Trinque
Oshin Karamian
John Michaud
Arian Kamali
Paul Bohn

Advisor: Prof. Bahram Shafai
Table of Contents

Abstract 3
Introduction 4
Problem Formulation 4
Design Goals 5
Problem Analysis 6
Design Approaches (hardware) 7
Design Approaches (software) 9
Design Implementation (hardware) 13
Design Implementation (software) 15
Design Prototype Analysis 18
Division of Tasks 20
Timeline 21
Cost Analysis 22
Conclusion 24
Reference List 25
Appendix 26
Abstract

There is currently a need for a device to allow researchers at the Martinos Center for Biomedical Imaging to better understand the brain activity related to improving motor functions in stroke patients. In previous experiments they have been comparing brain functions of healthy subjects to those of stroke patients during MRI and MEG scans while the subjects have been performing a series of hand motor functions. These experiments have been helping them to understand how the kinematics of motor task performance affects brain activity. Our capstone design project involved the design of an MRI and MEG compatible hand tracking device that will be used by both stroke patient groups and healthy patient groups. It will fulfill a current need for a study involving stroke patients during which directed hand grip strength will be observed during both MRI and MEG brain scans. Our design includes pressure sensors enclosed within a PVC cylinder that will be held by a patient’s hand and will track directed hand activities. It also features control software for the hand tracking device which collects data from the device while also time synchronizing our device’s data with data from the MRI and MEG devices. In addition, it includes a software interface that will direct the patient. An important part of our software design focused on is its ability to compare both the measurements of healthy patients to those of stroke patients and also its ability to compare previous measurements of the same stroke patient to future measurements. This will allow researchers to gather a large amount of data not only on how stroke patient’s motor functions compare to healthy patients, but also on how the brain of a particular stroke patient is changing and improving after their stroke.
Introduction

Medical Imaging is an important part of diagnosing and understanding illnesses today. It allows doctors and researchers to gain a visual understanding of the many functions and systems of the human body. Magnetic Resonance Imaging (MRI) and Magnetoencephalography (MEG) devices are important medical imaging tools that help doctors and researchers better understand the human body. These devices use magnetic fields to gain an understanding of key body functions. One very important part of the body that can be observed with both MRI and MEG devices is the human brain. When performing a brain scan with an MRI device the device induces a strong magnetic field onto the brain then tracks the changes of the brain. During an MEG brain scan the device measures the tiny magnetic fields produced by electrical activity in the brain using highly sensitive devices. Because the brain is the control center for the entire body, a great deal of information about the human body can be understood through analyzing results from MRI and MEG brain scan data. In addition, creating experiments using these devices allows researchers and doctors to draw important conclusions about the results of certain activities on individual parts of the body.

Problem Formulation

There are a variety of experiments that have been conducted on subjects while MRI and MEG brain scans were performed. These experiments help doctors and researchers to better understand the variety of functions that the brain performs. In addition, these experiments are also used to understand changes in brain behavior and performance both during and after an illness or injury. Currently, researchers at the Martinos Center for Biomedical Imaging are using MRI and MEG brain scan experiments to better understand the brain functions of stroke patients. In previous experiments they have been comparing brain functions of healthy subjects to those of stroke patients during MRI and MEG scans while the subjects have been performing a series of hand motor functions. These experiments have been helping them to understand how the kinematics of motor task performance affects brain activity. One very recent study involved a device that measured angular velocity of finger motion during an fMRI (see Fig. A3). This experiment helped researchers to better understand the motor functions of stroke patients during various stages of their rehabilitation.

Currently, there is a need at the Martinos Center for a device that can track directed hand grip strength during both an fMRI and MEG brain scan. There are many considerations in the design of this device as it must be compatible with both the MRI and MEG environments. In an MRI environment it is possible for sensor signals to become very contaminated by the noise induced by the MRI device. In addition, the highly magnetic field created by the MRI device can be very dangerous if ferrous materials are used within the field. As a result, it was important to shield our equipment from the magnetic field in order to minimize signal contamination and to use non-magnetic materials in our design. In an MEG environment the MEG device could be easily affected by noise from the sensors (see Fig. A1), or other electronics that are too close to the bore of the device. Therefore, it was vital for our design to minimize RF and other electronic noise near the
MEG bore. An additional hardware design constraint was that our device must accurately measure force while keeping displacement at a minimum in order to achieve unbiased results. An important software design goal was to accurately trigger the MRI and MEG devices so that they are properly time synchronized with our device. Additionally, we had to make certain that the patient interface was simple and helped to minimize patient head movement during the MRI and MEG scans to avoid unwanted measurement artifacts. Our main objective was to design our devise so that it maximizes comfort and usability for both researchers and the patient.

**Design Goals**

1. Static Force Testing Required
   - Little to no hand displacement during force measurement test is required.

2. Patient grip-force measurements must be repeatable
   - The way in which the device is operated can affect repeatability
   - Reliability accurate measurement are required
   - Limit mobility of the users hand to fix hand orientation relative to the grip-force device.
   - Build the device to operate independent of grip orientation.

3. Grip-force device must operate in MRI/MEG environments
   - This induces extra material design constraints
   - These special environments introduce extra noise that must not interfere with the sensitivity requirements of the design.
   - All Active components must be the correct distance away from the MEG device.

4. Sensitivity Requirements
   - The minimum/maximum grip-forces that will be applied to device
   - The accuracy and precision of our design
   - The grip-force device will be calibrated by current standards

5. Ergonomic Constraints
   - Anthropometry
   - Human Capabilities and limitations (hand mobility and dexterity problems)
   - Design for natural hand positions

6. Software Requirements
   - Collect Sensor Data
   - Trigger MRI/MEG
   - Calibrate Sensors
   - Easy for both Operator and Subject to Use
Problem Analysis

MR/MEG Compatible Grip Force Dynamometer for Stroke Patients

- MR/MEG Trigger
- MR/MEG Device
- Pressure Sensors
- Sensor Circuits/Wiring
- Grip Force Dynamometer Control SW
- Dynamometer
- User Interface SW
  - Proctor Interface
  - Patient Interface
Design Approaches

Hardware

Problem Definition

Our objective was to create a hand dynamometer that is compatible with both the low magnetic field of an MEG machine as well as the high magnetic field of an MRI machine. The dynamometer will be used by recent stroke patients to measure their static grip force while being scanned by each machine separately. The four things that held weight in the design approach of our dynamometer are as follows: ergonomics, repeatability, shielding from MRI fields, and also having a low effect on the MEG machine.

Design Approach

Our plan included a design that would be small and comfortable for recent stroke patients to use and will allow for consistent and repeatable results. In our thought process we designed several dynamometers that would encompass the hand or restrict movement in order to make the results have a standard reference. Some of these ideas included sensors mounted to the palm or digits, or other movement restrictive sensors. We soon realized that that was not the best approach and we revised our plan to instead of restricting movement to have a standard orientation to instead make orientation not matter during testing.

In order to achieve more accurate results, we planned on using a non-displacement type design. One idea we had was to use some type of stress ball that could be squeezed to sense the pressure. However, as the ball compressed, the direction of motion is changed as the fingers curl and will vary from person to person.

Another idea that we had considered was a device that would use hydraulics to measure force. However, this idea was not pursued for the final design. See appendix for a proposal for this design.

Our final idea consisted of a cylinder that can be gripped from any orientation with different shells to be used with different sized hands (see Fig A7/A8). This would help to solve both the repeatability and ergonomic constraints. Our design included perpendicular force sensors encompassed in the device and combines them up in a way to calculate the total grip force exerted by the patient.
Materials

Cylinder

The material choice for the cylinder was very important because the device must pass the MRI/MEG safety regulations including the deflection angle test. It also must be sturdy enough to hold its shape while being squeezed. This design also includes the use of undyed plastic because it is non-magnetic, strong, and easily obtainable.

Sensors

These were picked specifically to work in both extreme environments and pass MRI and MEG specifications. We used a piezoresistive pressure sensor. This is a differential type sensor that will function well being placed in between the two walls of the cylinder. Most pressure sensors are of the diaphragm type using a silicon back plate.
We felt that this type of sensor would be a good choice due to its simple design and small size. It would easily integrated into the cylinder and will hopefully cause a minimal amount of interference with the MRI/MEG.

**Wiring**

While researching connectivity solutions, we came across two possible ideas. Our first inclination was to use fiber optics because of the low loss transmission characteristics, as well as their lack of ferromagnetic materials. However, to use this type of connection, we would have needed additional circuitry on the cylinder side of the design for communication. Additionally, fiber optics were not within our project budget. Moreover, as part of our design approach, we are trying to minimize the size and amount of circuitry on the cylinder end of the system to reduce the amount of MRI/MEG interference and to keep the cost down.

A solution that we used to maximize most of our design constraints was the use of twisted shielded pairs of wires (See Fig. A6). These are low cost, low emission, and were easy to integrate into our design with no extra circuitry.

**Connectors**

By using twisted shielded pairs of wires, we eliminated the need for a connector on the cylinder side of the system. The only connector needed was to connect the cylinder to the computer interface. This connector was not critical to the design, and simple serial port connector that can be connected directly to a computer (see Fig. A5) was used.

**Software (see Fig A2)**

**Problem Definition**

This project required software that allowed us to collect and coordinate the data coming in from the sensors. It also required us to have a user interface that will allow the patient to monitor their proximity to their strength goal and allow the doctor controlling the test to alter the test as they see fit.
Design Approach

In order to monitor/control the information coming in from the sensor-device we wanted to create functions that would serve to:

Collect Raw Data

This function would output the raw data coming from the sensor devices. If there are multiple sensors we would package the data as a packet, including a starting bit, the data from each sensor, a counter, and the ending bit. For example:

\[ 8 \ 011033001 \ 10 \]

Where the 8 is the starting bit, 011 and 033 are data from sensors 2 and 3, 001 is the counter bit, and 10 is the end bit. This would allow us to easily parse the data and monitor any missing or dropped packets.

Collect Sensor Data

This set of functions would take the output from our raw data function and output the data for that sensor. For example, function SensorData1 would output 011.

The sensor data would be preserved in a class. This class would include the actual sensor data at that time, the value of the sensor when the patient applies the minimum amount of strength and the value of the sensor data when the patient applies the maximum amount of the strength.

Convert Data

The data from the sensor would either come in a scale in a number range (for example, 000-255) or would have some kind of millivolt output which we would be able to read. We also would require a function that could take this data and convert it into an accurate and usable form (such as psi or in newtons).

Trigger MRI or MEG

For this application we needed to trigger when the MRI or MEG begins its data acquisition.

At the MEG, there are 8 lines (with BNC connectors) for stimulus triggers. These channels are sampled and saved in the same way as the actual MEG channels. The MEG analysis software is set up to look for TTL-like pulses in these channels (up or down transitions in the voltage); those are used to indicate the onset of a stimulus presentation, or a button press response from the subject. Typically, a trigger pulse is sent every time a stimulus is presented, and the trigger combination provides a binary code for the stimulus type.
We planned to use the Parallel Port to transmit trigger pulses from the PC to the MEG acquisition. We have an adapter for the D25 connector to a BNC cable.

The situation is somewhat different in MRI. Usually, only a single trigger pulse is sent to the MR scanner to start the acquisition. The timing of the stimuli and responses is saved only in the PC into a log file, to be used later in the MRI analysis.

**Time Stamp**

The software would need to time stamp our data in order to correlate the sensor data to the MEG/MRI data. We would create a function that will output the time.

**Average Sensor Data**

The design would require a function that is able to average the value between all of our sensors so that we could get a number for the total force/pressure exerted by the patient. This function would calculate the data from all the sensors and output the average value.

**Capture Data**

We would require some kind of routine in order to capture the data coming in, the routine should be able to:

1. Trigger the MRI or MEG
2. Output the timestamp
3. Capture the data from the sensors
4. Write this data to a file

**Calibrate Sensors**

We would need to calibrate the sensors initially in order to understand what the maximum force or pressure the patient can exert. This function would request the user squeeze as hard as they can in order to get a maximum force for calibration, and would operate similarly for the minimum force. This would be stored in the sensor data class.

**Testing function**

We would need to create a function that would calculate the force needed from the patient at whatever percentage the doctor conducting the test requests it. The force required would be a percentage of the maximum force that is stored in the sensor data class.
**User Interfaces:**

Once the software code for data collection has been written, we would need to link it to two interfaces. One being a doctor interface in which a technician or doctor can modify the testing and set the parameters for each test. The other interface would be a user prompt for the patients, which would enable them to know what they will have to do for each test.

**Doctor Interface:**

The Doctor Interface (DI) will start the program; it would need to be designed to allow selection between MRI and MEG, start the calibration test for maximum force and minimum force. The DI will have a test protocol for the doctor or technician, allowing them to check number of iterations (number of tests), percentage force they want the patient to use, time in between testing (we do not want patient to develop a rhythm, as this may affect brain signatures), the ability to save user preferences (i.e. testing specifications), and to be able to save patient test logs. This should be designed as open as possible, allowing easy redesign for any specifications.

**Design goals:**
- Selection between MRI or MEG systems.
- Calibration test.
- Test Protocol – Number of tests, force percentages per test, time in between tests.
- Save Test Protocol preferences.
- Save patient test log to file.

**User Prompt:**

In the MEG and MRI systems there is a projection screen that enables doctors to prompt or test the patients. We hope to utilize this option for our Capstone project. We would like to utilize the screen for pressure testing for the patients. We wish to create a step-by-step procedure with examples to show the patient what to do during testing. Such as prompting them to squeeze the device for maximum force input, matching item testing, and examples of how the test will be done. The main test would involve two visual objects on screen, such as two squares. One square would be at a force percentage that the doctor designates; the other would be the square the patient needs to squeeze to make match the doctor square. We would design the prompt to offer visual changes in square size to allow the patient to know when they need to generate more, less, or hold at their force level.

**Design goals:**
- Visual prompt for patient.
- Examples for patient guidance.
- Show patient force percentage they are squeezing to match test objects on screen.
- Show End of Test.
Design Implementation

Hardware

While we were able to implement a working prototype based on our original design we were not able to follow our exact design approach due to time and cost constraints. One important part to our design was the sensor’s compatibility with MRI and MEG scans; which are both very sensitive to magnetic materials. Therefore, the sensor and materials we used were non-ferrous. We found a sensor made by Measurement Specialties, their FC22 line, which is a low cost, piezoresistive load cell (see Fig. A9). This type of sensor functions by using electrical properties of materials and pressure. The resistance of a material will change when pressure is applied to it. In the case of a piezoresistive sensor, a supply voltage is applied to the sensor, and a corresponding output voltage is sent out. As the sensor is squeezed, the resistance of the inner material changes, and the output voltage is changed correspondingly. The voltage relation can then be used to interpret the change in force or pressure. The sensor we purchased is constructed with plastic, silicon, and some stainless steel.

Although this particular sensor has not yet been tested in an MRI or MEG environment, it is our best hypothesis that due to the small amount of stainless used in the sensor and the limited iron content of the steel, that it will be acceptable to use in the MRI magnetic field (Find Articles). After speaking with the Solomon Diamond and the other doctors as Mass General, it was revealed that a previous experiment had been done with accelerometer sensors that were somewhat ferrous, but had passed an acceptability test in the MRI that allowed them to be used.

We had originally planned on using four sensors. The idea was to take a cylinder and cut it in quarters, each quarter would receive a sensor, and the total force would be a combination from all the sensors. The reasoning behind this would be that no matter which way the cylinder was gripped, the sum of the forces would be constant, and the orientation in which the sensor was held would no longer matter. This would increase the accuracy and repeatability of multiple tests runs on different patients. The MSI sensor we chose was the only sensor that fit our budget and our design constraints, the other sensors were either too expensive or had too small of a range to fit our application. Unfortunately, the profile of the sensor we chose was too large to fit multiple sensors inside of a properly sized cylinder for a human hand. We then decided, based on our budget constraint of not being able to purchase a properly sized compatible sensor, that we would add hand straps to our device to help orientate the sensor in the same way for each test and for each patient, thus increasing the accuracy and repeatability of a single sensor hand dynamometer, even though it would not be as accurate as a four sensor model. This single sensor design would act as a prototype that would serve as a proof of concept of our idea as well as allow us to design and implement the software in such a way that the same software could be used with an upgraded hand sensor in the future.

With the sensor and cylindrical shape chosen, we then proceeded to test different diameters of cylinders. The aim was to find the most comfortable diameter that could be
used on a wide range of hand sizes. After a few trials, we felt that a one inch diameter was too small as the fingers may overlap the thumb, and that a two inch diameter was too large as smaller hands might not be able to grip the whole cylinder. The final design consisted of a one and a half inch cylinder.

To further save on budgeting costs, we decided to machine the hand sensor ourselves instead of sending it out to be done professionally. This saved not only monetary costs, but also time because we did not have to create electronic blueprints and measurements of the device to give to a machinist to produce the piece. With some power tools and a few trips to the Home Depot, we were able to construct a hand sensor out of inch and a half PVC pipe, Plastique (a plastic epoxy), silicon glue, silicon caulk, stretch velcro, 5 minute epoxy, and twisted pair wires.

The PVC was cut to three and a quarter inches. This appeared to be the best width to fit all of the fingers comfortably without being too long. If the sensor was too long, torque could be applied to our sensor if the cylinder was gripped on one end or the other and the sensor was located in the middle. Also, if the sensor was too small, all of the fingers would not fit on the cylinder and the sensor may read inaccurate results if all of the fingers were not used to grip.

The PVC was then cut into two hemispheres in order to mount the sensor. The sensor's displacement is very minimal, which was a design constraint; however it still needed to be mounted freely so that the forces measured were on the sensor alone and not any of the PVC touching as the cylinder was squeezed. After the PVC was split, Plastique, a plastic epoxy made for patching PVC pipe was placed on the inside of each half of the PVC. The sensor was then placed in the middle and the two sides were then pushed back together. This in turn made a mold for the sensor to sit in the middle of the cylinder. Two shims made out of strands of wire were placed in between the two halves of PVC so that the halves would not touch each other and the sensor would be the only touching piece between the halves. Once dry, eighth inch holes were drilled in the Plastique that was surrounding where the sensor would sit in order to reduce some of the weight of the unit.

Once the mold dried we needed to figure out a way to hold the two halves together without affecting the sensing. One idea was to drill holes and thread one half of the PVC. A screw could be put through a hole on one side then screwed into the other. This would allow the sensor to be squeezed, but would not fall apart when released because the screw would hold it together. However, with some testing we realized that the same functionality could be achieved using silicon caulk. The silicon was flexible enough to allow the sensor to be squeezed, but durable enough to hold the sensor together when released. The final design used silicon glue on the inside of the cylinder to add more strength to the bond, and white silicon caulk on the outside for aesthetic purposes. The entire unit was then coated in clear spray paint in order to keep it clean from oils and dirt from hands.
The final step to the hand sensor was adding the orientation restricting straps. The loop side of a stretchable Velcro was used for straps. This material is durable and stretchable, yet soft enough for a patient interface. One strap was made to go around the thumb. 5-Minute Epoxy was used to attach the strap to the PVC cylinder. The strap was attached in a way that will conform to any thumb size without being too tight. Two more loops of strap were added to orientate the forefinger. Two were needed so that the sensor could be used for left and right handed patients.

The final result is a sealed inch and a half PVC cylinder that contains a piezoresistive sensor mounted in epoxy in the middle, complete with straps that help to hold the sensor on the patients hand while testing, and also keep it in the same position to help improve accuracy and repeatability to the measurements (see Fig. A10).

**Software**

**Hardware to Software Interface**

In order to get the sensor data into the computer where we could record and analyze it, we needed some sort of Analog to Digital converter and computer interface. After doing some research on products that have already been designed to do this, we decided to purchase a MiniLab 1008 USB-DAQ system. This is a USB powered acquisition system that would easily interface our sensor with our computer. This model was inexpensive and had multiple channels so that we could expand to multiple sensor hand dynamometers in the future. This DAQ was selected due to its low cost and flexibility. At the time the DAQ was purchased it was unknown whether we would need 1 sensor or 4 separate sensors. This DAQ gave us that flexibility. The low cost was also a factor. Its cost of 130 dollars was significantly cheaper than most DAQs that would have gone beyond our budget. This DAQ also served to power our sensor using the power supplied by the computer USB connection.

**Software Architecture**

On the software side of things, we needed to design a system that would capture the data taken by the DAQ system, store it, timestamp it, and also provide an interface for the doctors performing the experiments to set them up and start them, and also a visual prompt for the patient to use to guide them through the experiment.

We chose to use MATLAB to do our programming in due to its powerful mathematical functions and its easily integrated GUI programming. MATLAB was also advantageous due to how it handles graphing and vectors. We were able to set up a GUI that would be easy for the Doctors to use to perform the experiments as well as a visual prompt that would appear on a separate screen that would be shown in the MRI or MEG rooms for the patients to follow. The software was designed in a modular fashion so that it could be easily modified and adapted to different applications, both for our group and for future groups that may work on this project. Our data was parsed into text files to allow for the saving of data as well as easy integration into Excel spreadsheets or other data analysis tools.
The researchers at MGH gave us a few guidelines to follow while creating the procedure. Among the constraints were that the patient must keep their eyes focused on the center of the screen. The eye motor function area in the brain is very close to the hand motor function area, by keeping the eyes in one place, cross correlation error can be reduced between these two exercises.

The test begins with a small red circle in the middle of a gray screen. This is for the patient to focus on. The software then guides the patient through a calibration process. This is a two step process. First a measurement is taken with no force applied to the sensor. Second a measurement is taken at the patients maximal grip strength. This provides a range of force for that particular patient. The following tests will then be a percentage of the user’s maximal force, 15, 20, or 30 percent for example. Each test will have a tolerance that can be set which is the range or width between two visible red bars on a graph. The patient will have to squeeze the sensor and try to get a graphed line to stay in between these two red set points. Once this is achieved, the test is passed. The doctor can set how many tests to perform and at what percent of maximal force they will be at. Before the test begins there are three colored circles that appear with the words “Ready, Set, GO!” This provides ample time for the patient to ready themselves for the next test.

In order to give the software greater flexibility we tried to separate a lot of the customizable options into several different functions.

**DAQ Setup**

The DAQ setup function allowed the future programmers adjust the settings so that the software could work with a variety of DAQ devices from different manufacturers that operate at different frequencies. The software can also be modified to work with multiple sensors on the same DAQ using this function. The DAQ is initialized when the program is executed. If the DAQ is not plugged in, this function will not work and thus the program will not run.

**Storage of Data/Doctor Interface**

Due to how the MATLAB GUI operates, we needed to design a lot of data to be stored in the “background” of MATLAB. This was accomplished by using the getappdata and setappdata built-in MATLAB functions (for more information consult the MATLAB documentation). The following information was stored from the Doctor’s information using this protocol:

1. Data file Name
2. Test Percentages (1-10)
3. Test Tolerance
4. DAQ information
5. DAQ channel that is read
6. Array of information gathered from DAQ
7. Tolerances Range (One vector for high and one for low)

**Patient Interface**

For the Patient, this program has two main functions. The Calibrate Function, and the Run Test Function.

The Calibration function is used before any testing is done. Since patients will have varying degrees of grip force we decided to design the calibration to take the minimum value of the data while the device is at rest, and then take the maximum of the patient’s grip force by having them apply their maximum force to the device.

To design enable this feature we needed to have a tolerance for noise (to enable slight changes in force to be smoothed out and not constantly restart the calibration) and to be able to compare previous values to new values. We do this by reading the data from the DAQ at 10 Hz and storing it in an array (20 values). This array is then averaged into one value and stored for comparison. Then we use an if/else statement to compare the previous value to the new value of data coming in. As long as these values stay within the tolerance range set in the program (+- 0.1 volts in our prototype) for 20 consecutive values this will pass the Calibration function. To assist the patient in knowing when to calibrate the device, we have on screen prompting to let them know when to release and squeeze the device. They will also see their input displayed on a graph on screen during testing to know when they are done calibrating. This results in the output of CalMax and CalMin values for the Run Test Function.

For the Run Test Function, we first take in 4 input values. First are the CalMax value and the CalMin outputs from the Calibration function, which are subtracted from each other to give the actual maximum force the patient squeezed. Then the TestPercentage input, which takes a percentage the doctor, chooses such as 15% and sets the value the patient wants to match from their maximum calibration. So if their maximum was calculated to be 3.5 volts and minimum at 0 volts, the value the patient will try to match will be 0.525 volts. To help with noise control, we also have the doctor input a Tolerance value. The Tolerance value sets upper and lower bounds in which the patient has to consistently stay in the middle of. So if the tolerance is set to 5% in the previous example, the patient now has to squeeze anywhere between 0.35 volts and 0.7 volts for a certain set number of runs to pass (needs to be in the center for 10 consecutive runs in prototype) to pass the test.

The way this is shown to the user is on the GUI Force graph. Where the data being received inputs to the center of the graph and is then shifted left and right for 10 points (so the graph is 20 points long with the newest data being outputted to the center of this graph) and has a vertical max value at 60 lbs (.1volts = 1 lbs. conversion). This concept makes it so that the patient’s do not shift their eyes to follow the graph data. We also placed a centerline on the graph to help the patient keep his or her focus in the exact center of the graph. All the data received is stored in a text document with a timestamp.
and a ‘P’ or ‘F’, P being if the data was within the upper and lower bounds and an F for when the data was outside of these bounds.

*Test_Panel.m Graphical Updates*

The Patient Interface used some data that was saved in the “background” of MATLAB (Plothigh, Plotlow, and DataArray). Other than those, it contained a few built in functions:

1. **WaitForNextTest**
   - This function sets a Red dot graphic in the center of the screen that the patient can focus on

2. **UpdateAxesOneGraph**
   - This function updates the graph with only one updating array. This is used to initiate measure. It uses the stored data from DataArray1.

3. **UpdateAxes**
   - This function updates the Axes along with tolerances. It uses the stored data from DataArray1, PlotLow and PlotHigh.

4. **ReadySetGo**
   - This function cues up a set of graphics that run before the user starts testing.

*Design Analysis*

*Hardware*

We were able to conduct some testing of our prototype to determine the linearity of the measurements as well as to correlate the voltage output of the devise to a force in pounds. Additionally, we completed some testing to determine the noise margin of our device.

The linearity testing was done by suspending measured weights by a wire that was draped over the center of our device. We then recorded the voltage value for that particular weight. We used weights ranging from .2 lbs to 35 lbs and found our device to be reasonable linear (see Fig. A11). We were also able to determine that the voltage to pound relationship was .1 volt per 1 pound.

Using the oscilloscope and known weights we measured the relative noise margin of our device. We measured at ten pound intervals from zero to thirty pounds the max noise, min noise, as well as the RMS voltage value.

<table>
<thead>
<tr>
<th></th>
<th>0 lb</th>
<th>10 lb</th>
<th>20 lb</th>
<th>30 lb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min (V)</td>
<td>.501</td>
<td>.994</td>
<td>2.001</td>
<td>2.994</td>
</tr>
<tr>
<td>Max (V)</td>
<td>.516</td>
<td>1.015</td>
<td>2.010</td>
<td>3.011</td>
</tr>
<tr>
<td>RMS (V)</td>
<td>.508</td>
<td>1.000</td>
<td>2.004</td>
<td>3.008</td>
</tr>
</tbody>
</table>

Our average noise margin is .016 V, a very reasonable number for the voltage levels that we are dealing with.
Software

We had an opportunity to examine whether the data that was saved to the text file matched the data we saw on the screen. Initially we thought that when we had set the data transfer via MATLAB to 20 Hz but the data written to the text file was less than 10 Hz (we saw less than 10 points plotted per second). In order to verify whether the test was skipping writing points to the data file we used the “tic” and “toc” functions before and after the “start(ai)” command to analyze the data rate of starting and stopping the data acquisition. We found that the “start” function itself took 5 ms, with a 5 ms pause on top of this it took 13-15 ms each time we gathered the data, plotted and wrote it to the text file. This verifies that the data plotted did match the data that was written to the file.

While the documentation states that the DAQ has a transfer rate of 50 Hz we found that in MATLAB it was well under half of this. In order for the DAQ to acquire one data point it took a little over 1/10th of a second (a little over 10 Hz). We are fairly sure that this is due to the DAQ and not our software. Experimenting with different DAQs could help to speed up our data transfer rate.
Division of Tasks

Eric Krejci (leader):

- Coordinate Technical Writing and Presentation
- Hardware/Software Interfaces Design and Coordination
- Hardware build support
- Software coding support

Ben Pinkus

- Sensor selection and placement
- Overall hardware device design/modeling
- Hardware build

Jason Trinque

- Sensor selection and placement
- Hardware device design
- Hardware build

Oshin Karamian

- Software Back End Design
- Software Coding

John Michaud

- Software User Interface Design
- Software Coding

Arian Kamali

- Materials Selection
- Wire and Connector selection
- Hardware build
Timeline

July
- 5 – Capstone project begins; form groups
- 6 – Discuss possible project ideas with group members
- 12 – 1st meeting with professor Shafai; discuss possible project ideas with Prof. Shafai.
- 19 – 2nd meeting with professor Shafai; short meeting, topics not discussed due to MGH trip following day.
- 20 – Jay, Oshin, Eric, and Ben met with project coordinators and Prof. Shafai at MGH. Discussed scope of project.
- 26 – 3rd meeting with professor Shafai; decide to pursue MGH research project.

August
- 2 – 4th meeting with professor Shafai. Shortly discussed sensors for application; Following meeting Eric, Jay, Ben, John and Oshin went to MGH to further discuss project with MGH project coordinators.
- 9 -5th meeting with professor Shafai. Discuss project presentation and proposal.
- 10 – Group meeting to discuss division of tasks for proposal and project presentation.
- 15 – Group meeting to combine project proposal and presentation.
- 16 – 6th meeting with professor Shafai. PowerPoint presentation due.
- 17 – Present project and proposal to Capstone Design class

Fall
- Undergo MRI and MEG training. Gather materials and prepare software for fall.

January
- 8 – Spring semester begins.
- 20 – Have all necessary equipment to begin building MGH prototype.

February
- 10 – Backend of software complete

March
- 15 – Complete Hardware construction of handheld device (not including interface to computer)
- 22 – Complete User interfacing software.
- 22 – Complete Hardware construction to interface with computer.

April
- 3 – Work on hardware/software interfacing. Work on any and all project bugs.
- 10 - Project complete.
- 17 - Final Design and Report Due.
Cost Analysis

<table>
<thead>
<tr>
<th>Proposed Cost for Grip Force Dynamometer</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Quantity</td>
<td>Cost</td>
<td>Total Cost</td>
</tr>
<tr>
<td>Sensors</td>
<td>4</td>
<td>$75</td>
<td>$300</td>
</tr>
<tr>
<td>Plastics</td>
<td>5' length</td>
<td>$50</td>
<td>$50</td>
</tr>
<tr>
<td>DB9 Solder Connector</td>
<td>2</td>
<td>$2.50</td>
<td>$5.00</td>
</tr>
<tr>
<td>Wires</td>
<td>150ft</td>
<td>$0.50/ft</td>
<td>$75</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>$430</td>
</tr>
<tr>
<td>Machining Costs (Requires quote after specific parts are chosen)</td>
<td>estimate</td>
<td>$100</td>
<td></td>
</tr>
<tr>
<td>Grand Total</td>
<td></td>
<td></td>
<td>$530</td>
</tr>
</tbody>
</table>
BLAH BLAH insert printed xls here!
Conclusion

Our device will enable researchers at the Martinos Center for Biomedical Imaging to measure hand grip force during MRI and MEG scans. This device will track software directed hand grip strength during both an fMRI and MEG brain scan. Our design will include pressure sensors that will be held by a patient’s hand which will track directed hand activities. It will also feature control software for the hand tracking device as well as a software interface that will prompt the patient and time synchronize our device with data from the MRI/MEG devices. This will allow the researchers to better understand stroke patient recovery by giving them the ability to correlate hand grip strength data with brain scan data from MRI and MEG devices.
References

Grip Force Vectors for Varying Handle Diameters and Hand Sizes.
http://eadc.engr.wisc.edu/Web_Documents/Edgren%202004.pdf

http://www.sensorland.com/HowPage004.html

Wire Source
http://www.wildtracks.cihost.com/homewire/wg_types.html#wire_cost

Connector Source

Previous Motor function experiments
Schaechter, Judith. “Finger motion Sensors for fMRI motor studies”.
Appendix:

Fig. A1. Comparison of Magnetic Fields

**Magnetic Fields**

B (Teslas)

- Earth's Field
- Urban Noise
- Car @ 50 m
- Screwdriver @ 5 m
- Transistor, IC chip @ 2 m
- Transistor die @ 1 m

**Biomagnetic Fields**

- Lung particles
- Human heart
- Skeletal muscles
- Fetal heart
- Human eye
- Human Brain ($\alpha$)
- Human Brain (evoked response)
- SQUID System Noise level
Fig A2. Software Flow Chart

Fig. A3. Velocity Sensing Finger Sensor Device
Fig. A5. DB9 Solder Connector – Male

Fig. A6. Conductor Shielded Cable/Twisted Shielded Pairs
Fig. A7. Grip Force Dynamometer Early Design Concept

Fig. A8. Grip Force Dynamometer Final Design
Fig. A9. Load Cell (sensor) Spec Sheet
FC22 Compression Load Cell

LOW COST
Compression Ranges: 10, 25, 50 and 100 Lbf Compression
High Level or Millivolt Outputs
Interchangeable
Compact Easy to Fixture Design

DESCRIPTION

The FC22 Series low cost compression load cells create new markets previously unrealizable due to cost and performance constraints. This series provides a new level of performance at very low cost. Measurement Specialties' proprietary Microfused™ technology, derived from demanding aerospace applications, employs micro-
machined silicon piezoresistive strain gages fused with high temperature glass to a high performance stainless steel substrate. Microfused™ technology eliminates age-sensitive organic epoxies used in traditional load cell designs providing excellent long term span and zero stability.

Operating at very low strains, Microfused™ technology provides gauge factors greater than 100, an essentially unlimited cycle life expectancy, superior resolution, exceedingly high overrange margins without the need for steps and a ratiometric span of up to 4V. Microfused™ sensors are used in a variety of applications including bathroom scales, paint sprayers and safety-critical automotive stability control.

Measurement Specialties' model FC22 is appropriate for use in all types of OEM weighing and force measurement applications where high reliability and accuracy are critical. From appliance controls to biomechanical force feedback, FC22 is the OEM designer's dream-come-true: cost-optimized to bring your OEM products to life whether you need thousands or millions of load cells annually. Although the standard model is ideal for a wide range of applications, the design team at our Load Cell Engineering Center can provide custom designs for your OEM applications. The FC22 is fully thermally compensated for changes in zero and span with respect to temperature and offer normalized zero and span for interchangeability. Consult Measurement Specialties for uncompensated super low cost variants of the FC22 load cell.

FEATURES
- Low cost
- Small size
- Low noise
- Robust high overrange capability
- High reliability
- Low deflection
- Essentially unlimited cycle life expectancy
- Low off center errors
- Fast response time
- From 10 to 100 lbf ranges
- Reverse polarity protected

APPLICATIONS
- Medical infusion pumps
- Robotics end-effectors
- Variable force control
- Load and compression sensing
- Exercise machines
- Pumps
- Contact sensing
- Weighing
- Household appliances

CE compliant per the following specifications:

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEC61000-4-2</td>
<td>1st Kv Kv (Air/Contact)</td>
</tr>
<tr>
<td>IEC61000-4-3</td>
<td>3rd Kv (Air)</td>
</tr>
<tr>
<td>IEC622</td>
<td>Class A</td>
</tr>
</tbody>
</table>

### FC22 Compression Load Cell

#### Performance Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Ranges: FC22</td>
<td>10/22, 50, 100 lbf/compression</td>
</tr>
<tr>
<td>Maximum overload</td>
<td>250% of range</td>
</tr>
<tr>
<td>Recommended Excitation Voltage (Amplified)</td>
<td>3.2 V to 5 V DC (1)</td>
</tr>
<tr>
<td>Recommended Excitation Voltage (Bridge Only)</td>
<td>5 V DC</td>
</tr>
<tr>
<td>Full Scale Output Voltage (Bridge Only)</td>
<td>0.5 to 4.5 V +/− 3% of span at 5 VDC excitation</td>
</tr>
<tr>
<td>Output Voltage at No Load (Zero output)</td>
<td>+/−5% FSDC (2)</td>
</tr>
<tr>
<td>Combined Non-linearity, Hysteresis, and Nonrepeatability</td>
<td>+/− 1% FS (typical)</td>
</tr>
<tr>
<td>Temperature Compensation</td>
<td>0°−50°C</td>
</tr>
<tr>
<td>Thermal Sensitivity Shift</td>
<td>+/− 0.03%/°C</td>
</tr>
<tr>
<td>Operating and Storage Temperature Range</td>
<td>−40°C to 85°C</td>
</tr>
<tr>
<td>Humidity</td>
<td>0−95% RH</td>
</tr>
<tr>
<td>Input Resistance (Bridge Only)</td>
<td>3K ohms (nominal)</td>
</tr>
<tr>
<td>Output Resistance (Bridge Only)</td>
<td>2.2K ohms (nominal)</td>
</tr>
<tr>
<td>Deflection at Full Load</td>
<td>&lt;0.05 mm</td>
</tr>
<tr>
<td>Isolation Resistance</td>
<td>≥ 50 M ohms @ 250VDC</td>
</tr>
</tbody>
</table>

1) Higher excitation voltages available on request
2) Lower trim values available on request (FS = Full Scale Output)

#### Dimensions

![Dimensions Diagram](image)

#### Ordering Information

<table>
<thead>
<tr>
<th>Family</th>
<th>Body</th>
<th>Output</th>
<th>Connection</th>
<th>Specials</th>
<th>Range</th>
<th>Multiplier</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample PK</td>
<td>Z2</td>
<td>3</td>
<td>1</td>
<td>0000</td>
<td>0010</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>FC22 Compression</td>
<td>1 mW ISO</td>
<td>0.545 V FS @ 0 VDC input</td>
<td>1 m Cable output</td>
<td>Reserved for custom design</td>
<td>100 Lbf, 20, 50, 100 Lbf</td>
<td>n/a</td>
<td>Lbf</td>
</tr>
</tbody>
</table>

Note: L = Newton
Fig. A10. Functional Prototype of Hardware Design
Fig. A11. Linearity of Prototype

\[ y = 0.0916x + 0.5366 \]

\[ R^2 = 0.9976 \]