

CORNELL UNIVERSITY, DEPARTMENT OF MECHANICAL & AEROSPACE ENGINEERING

---

# **Magnetic Field Strength Measurement System for Type II Superconductor Flux Pinning**

---

Steven DOURMASHKIN

NetID: SJD227

Adviser: Mason PECK

December 15, 2015

## CONTENTS

1. Abstract	1
2. Introduction	1
3. Design	2
3.a. System Requirements	2
3.b. Hardware Selection	5
3.c. System Integration	7
3.d. Software Design	12
4. Analysis	15
4.a. Calibration and Characterization	15
4.b. Post-Processing	20
5. Results	24
6. Conclusion	27
References	29
7. Appendix - Sensor Trade Studies	30
8. Appendix - Primary MATLAB Code	31

## 1. ABSTRACT

Magnetic flux pinning occurs when a type-II superconductor is cooled below its critical temperature in the presence of one or more bodies generating a magnetic field. In turn, these magnetic fields become “frozen,” creating a passive equilibrium in up to six degrees of freedom for each magnetic body. This passive control of position and orientation has led to the recent research of flux pinning as, among other applications, a favorable mechanism for spacecraft docking. This report describes the design and implementation of an efficient triaxial magnetic field strength measurement system used to gather empirical data of the magnetic interactions caused by flux pinning. The final measurement system consists of a motorized linear translator that is manually, precisely controllable in its other two translational degrees of freedom, with a set of three single-axis hall sensors to measure magnetic field strength along three orthogonal axes as well as three ultrasonic sensors to measure associated position vectors. The experiments that have and will be conducted using this system will enable improvement of the ideal image dipole model currently being used by Cornell University’s magnetic flux pinning research team. In particular, the empirical data will help yield a refined model that accounts for a finite-volume effect of type-II superconductor flux pinning in order to establish appropriate initial conditions for passive spacecraft docking through magnetic flux pinning.

## 2. INTRODUCTION

Magnetic flux pinning allows the relative dynamics between two spacecraft to be altered in a unique way that reduces risk for close proximity spacecraft interactions, which are high-risk and complex yet essential to many missions (e.g., sample return or on-orbit repair) [1]. In a flux-pinned interface (FPI), a type-II superconductor is cooled in the presence of a magnetic field below its critical temperature, which is 88 K for Yttrium Barium Copper Oxide (YBCO): the superconductor used in this research. Known as field-cooling, this process creates supercurrent vortices in the superconductor that react to changes in magnetic flux, in turn creating a potential well that causes the field-cooled magnet(s) to retain an initial position and orientation in up to six degrees of freedom. For example, a uniformly-dense cubic magnet will retain all six degrees of freedom, since any translation or rotation will impart a change in the magnetic field relative to the superconductor; on the other hand, a cylindrical magnetic, which is axis-symmetric, will be free to rotate about its symmetric axis after being field-cooled yet bounded in all other five degrees of freedom. This phenomenon thus creates a potential well that can draw two bodies into equilibrium, correct misalignment, and attenuate kinetic energy to prevent collision without any mechanical contact [2].

When combined with electromagnetic actuators, which can be controlled to alter the induced magnetic flux distribution surrounding the superconductor, the FPI can alter the equilibrium of the field-cooled permanent magnet(s). As a result, electromagnetic actuators enable reorientation and position control of one body relative to another. Referred to as a magnetic array, a collection of permanent magnets and electromagnets can therefore be positioned on one spacecraft while the superconductor is placed on another to control their relative behavior [3] within tens of centimeters. This attitude control design has been studied for a variety of spacecraft applications, including docking [4], formation flight [5], and segmented spacecraft in-flight reconfiguration [6]. In particular, the FPI can be

used for a mars sample-return mission in which it can attract and manipulate an orbiting sample without the use of mechanical contact [2].

In order to assess the use of an FPI in a viable Mars sample-return architecture, the conditions under which objects can successfully enter and be captured by the FPI must be better understood. Specifically, refinements to the existing flux-pinned models must be made to enable a more accurate representation of the potential energy wells induced through flux pinning to, in turn, better predict the capture limits of a practical FPI [2]. These refined models, which take into account the finite-volume effect of type-II superconductor flux pinning, can be validated by measuring magnetic field strength vectors in a three-dimensional grid about field-cooled permanent magnet(s) and electromagnetic actuators. The design and implementation of such a magnetic field strength measurement system, in addition to the initial experiments conducted and empirical data received, will be described throughout this report.

### 3. DESIGN

**3.a. System Requirements.** To help define a list of originating requirements, a context diagram was created for the magnetic field strength measurement system. A context diagram displays the most prominent items, operators, and environmental factors involved with the system in rectangular boxes. Through lines with descriptive text connecting these boxes, this tool illustrates the ways in which these entities interact with the system as well as each other. In turn, the context diagram is used to better understand the environment in which the system exists; in other words, it is used to more thoroughly and accurately define system requirements relevant to both intended and unintended or accidental interactions. In particular, this context diagram helped determine that, in addition to recording magnetic field strength and associated position vectors, the system must also hold the superconductor at a precise, fixed location relative to the measurement locations. The final context diagram for the magnetic measurement system is shown below in Figure 1.

The list of originating requirements was ultimately provided by Ryan Caracciolo, a PhD Candidate in the School of Applied and Engineering Physics leading the flux-pinning research team with Frankces Zhu. These requirements are provided below in Table 1.

To conceptualize the system that would satisfy the listed requirements, an annotated concept sketch was created. An annotated concept sketch is a rough sketch of a system concept with directed text illustrating the key high-level functional or structural components of the system. A functional annotated concept sketch describes the capabilities of particular features, how those features meet particular requirements, and how they interact with other features or sub-systems. On the other hand, a structure annotated concept sketch specifies exact solutions to implementing certain capabilities and interactions as well as satisfying customer needs; in other words, the structural annotated concept sketch provides solutions to the goals indicated in the annotated concept sketch. The functional and structural annotated concept sketches are provided below in Figures 2 and 3.

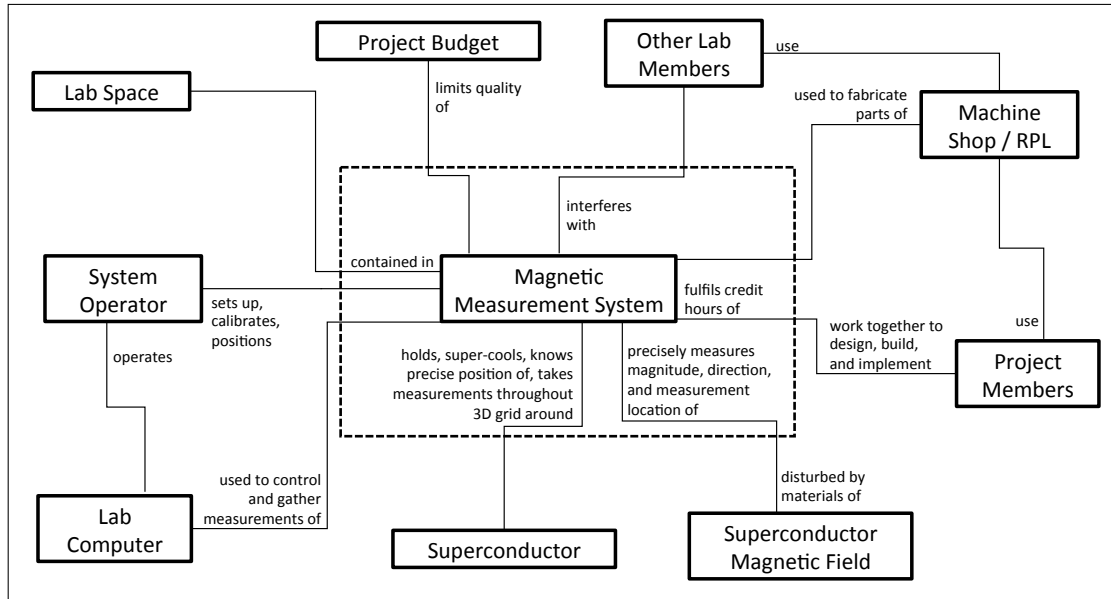


FIGURE 1. Context diagram of magnetic field strength measurement system.

Index	Originating Requirement	Abstract Name
OR.1	The system shall measure magnetic field strength in the orthogonal x, y, and z axes along with its corresponding x, y, and z position	Measurement Types
OR.2	The system shall be able to measure the magnetic field of each superconductor, permanent magnet, and electromagnet	Measure Fields
OR.3	The system shall measure magnetic field strength up to at least 750 G	Magnetic Range
OR.4	The system shall measure magnetic field strength with sensitivity of at most 3 G	Magnetic Sensitivity
OR.5	The system shall measure position with a 20 mm range in all three orthogonal axes	Position Range
OR.6	The system shall measure position along each axis with sensitivity of at most 0.5 mm	Position Sensitivity
OR.7	The system shall have an operating voltage under 18 V	Maximum Voltage
OR.8	The system shall have a maximum current draw under 30 A	Maximum Current
OR.9	The system shall enable experiments to repeated at least 100 times without repair	Repeatability
OR.10	The system shall consist of components purchased under a total of \$1000	Total Cost

TABLE 1. Originating requirements of Magnetic Field Strength Measurement System

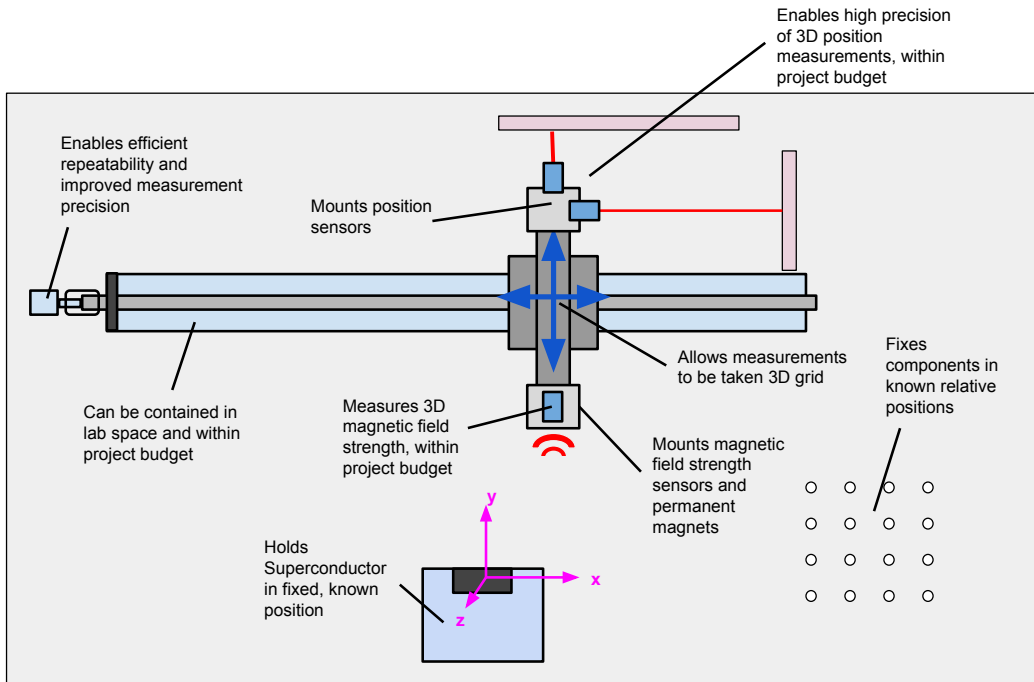


FIGURE 2. Functional annotated concept sketch of magnetic field strength measurement system.

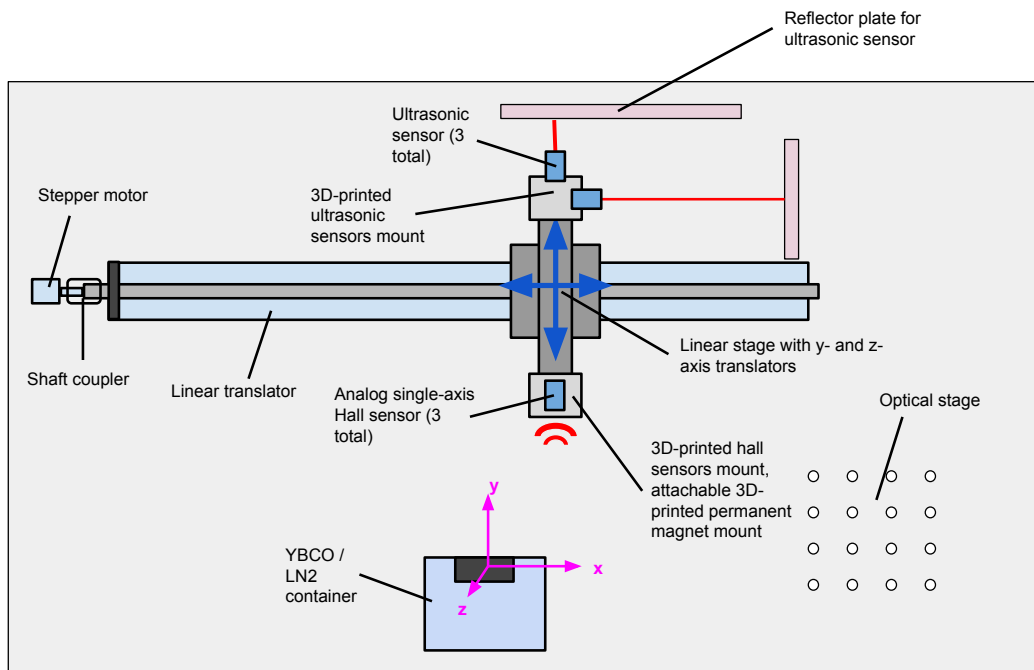


FIGURE 3. Structural annotated concept sketch of magnetic field strength measurement system.

As illustrated in the annotated concept sketches, the magnetic measurement system was determined to consist of a set of three orthogonal ultrasonic sensors and single-axis analog Hall sensors fixed to a linear stage with y- and z-axis translators, in turn controllable by a stepper motor attached to the shaft of an x-axis linear translator. Although optical sensors (e.g., laser distance sensors) would have enabled a higher degree of precision (under 0.1 mm), they are expensive – in general exceeding \$500. As a result, ultrasonic sensors, which are less expensive yet precise enough to satisfy the originating requirements, were ultimately selected. Similarly, while a 3 DoF motorized linear translator would have enabled significantly greater efficiency and autonomy by not requiring manual adjustment of the y- and z-axis translators while sweeping through a three-dimensional grid, such a system costs thousands of dollars and would thus be significantly outside of the project budget. Instead, a linear translator belonging to the Space Systems Design Studio (SSDS) was used; and, the manual turning wheel of the translator was replaced by a stepper motor to increase autonomy, consistency, and position measurement precision along the x-axis. The Hall sensor, ultrasonic sensor, and stepper motor were decided to be fixed to the linear stage and optical table using 3D-printed mounts. While the digital manufacturing capabilities available to students through the Rapid Prototyping Lab (RPL) yields less precision than the tools of the Emerson Machine Shop, 3D-printed parts were ultimately chosen to allow faster prototyping and iteration since our team was on a tight schedule throughout the semester. The selection of the particular Hall sensor, ultrasonic sensor, and stepper motor will be described in the next section of this report.

**3.b. Hardware Selection.** As stated in the list of originating requirements (Table [1](#)), a magnetic field strength sensor capable of measuring up to 750 G (OR.3) with sensitivity of at most 3 G (OR.4) was required. Because of this large magnetic field strength range requirement, the search for an appropriate sensor was fairly limited. Provided in Table [A1](#) of the Appendix, six magnetic field strength sensors were identified as viable solutions to the originating requirements. Ultimately, the Analog Hall Sensor manufactured by Sensor Solutions, as shown below in Figure [4](#), was chosen. The primary attribute that led to the selection of this analog hall sensor was its analog data interface, which could readily be integrated into the National Instruments Data Acquisition Device (DAQ) used in the flux-pinning research lab. Further, it was chosen over the two-point programmable hall analog sensor by Sensor Solutions because of its lower price (\$80 versus \$136) and lower detection range ( $\pm 1$  kG versus  $\pm 1.5$  kG), which enables a higher resolution through the DAQ’s 1024 bit analog-to-digital converter (1.95 G/bit versus 2.93 G/bit). Because the  $\pm 1$  kG Analog Hall Sensor is single-axis, three were required to concurrently measure the magnetic field strength in all three orthogonal axes.

Next, a relative position sensor satisfying OR.5 (“Position Range”) and OR.6 (“Position Sensitivity”) was selected. Chosen from a list of six position sensors provided in Table [A2](#) of the Appendix, the SICK UM12 ultrasonic sensor with a working range of 40 mm to 240 mm (refer to Figure [4](#)) was selected. As previously stated, an ultrasonic sensor was chosen over optical sensors, which are generally much more precise, due to lower cost and low enough sensitivity to satisfy the system requirements. Moreover, the flux-pinning research lab already owned two SICK UM12 ultrasonic sensors; thus, only one additional sensor was required, which greatly reduced purchasing costs. Smooth aluminum plates were determined to be used as reflective surfaces for these ultrasonic sensors. They were

machined and mounted to the optical table such that their faces were parallel to the front faces of each ultrasonic sensor.



FIGURE 4. Sensor Solutions Analog Hall Sensor (left) and SICK UM12 Ultrasonic Sensor (right), both of which were selected as final sensors for magnetic field strength measurement system.

Before selecting a stepper motor to autonomously and precisely control the translation of the linear stage along the x-axis, a simple experiment was conducted to determine the amount of torque required to rotate the shaft of the linear translator. As illustrated in Figure 5, a downward force was applied to the handle of the translation wheel, located at a distance of  $R = 27$  mm from the center of the linear translator shaft, until the shaft began to move. Upon attaching masses to the handle of the wheel using fairly coarse increments, a mass of 225 g was found to turn the handle downward, corresponding to a torque of  $T = 0.059$  N · m.

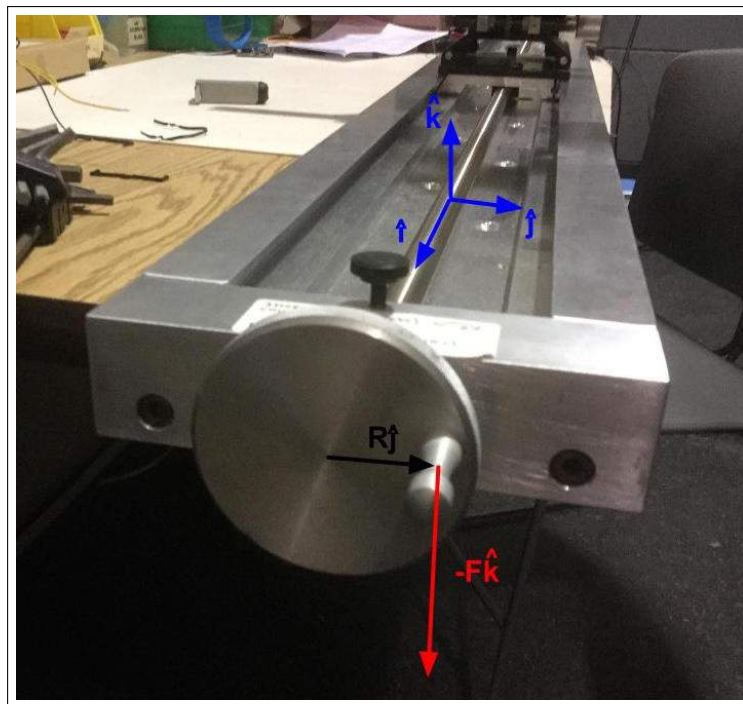


FIGURE 5. Illustrating of motor holding torque experiment.



To overcome this holding torque of  $T = 0.059 \text{ N} \cdot \text{m}$ , the Sparkfun Electronics 400 steps/rev 68 oz-in stepper motor (refer to Figure 6), with a rated holding torque of  $0.48 \text{ N} \cdot \text{m}$ , was selected. Additionally, this motor enabled a great deal of precision; since one rotation of the linear translator shaft is specified to be  $2 \text{ mm}$ , each commanded step of the motor corresponds to  $5 \mu\text{m}$ . Thus, when combined with the external position measurements from the x-axis ultrasonic sensor in a linear dynamics estimation filter (e.g., a Kalman Filter), commanded input to the stepper motor that propagates the linear stage by  $5 \mu\text{m}$  can enable very low estimation uncertainty in the x-position of magnetic field strength measurements.



FIGURE 6. Sparkfun Electronics 400 steps/rev 68 oz-in stepper motor, the selected stepper motor to control linear translator motion.

In order to drive the Sparkfun stepper motor, the Sparkfun Big and Easy Driver was selected. As its name suggests, this driver is simple to work with, commanding a single step to the motor through a change in voltage supplied to the STEP pin of the board. Further, this driver supports micro-stepping, which can divide a step into as little as 16 micro-steps to yield smoother and more precise stepping.

**3.c. System Integration.** The wiring of the magnetic field strength measurement system to the National Instruments SCB-68A DAQ breakout board is provided below in Figure 7. The blue labels in the wiring diagram are connected to the DAQ breakout board, while the red labels (i.e., the  $+10 \text{ V}$  supply) are connected to the SPS-30DM power supply. In addition, all corresponding negative analog reference terminals on the SCB-68A breakout board are connected to ground. Note that the  $500 \Omega$  resistor converts the output of the z-axis ultrasonic sensor from current on a zero to  $20 \text{ mA}$  scale to voltage on a zero to  $10 \text{ V}$  scale (i.e., the same scale as the other two ultrasonic sensors). Also note that the green unconnected wire of each ultrasonic sensor is its control pin, which is used to set the sensing window and is thus not need for operation of the magnetic field strength measurement system, assuming that the sensing window remains unaltered. If necessary, instructions for re-calibrating a sonar's sensing window are provided in the SICK UM12 data sheet.

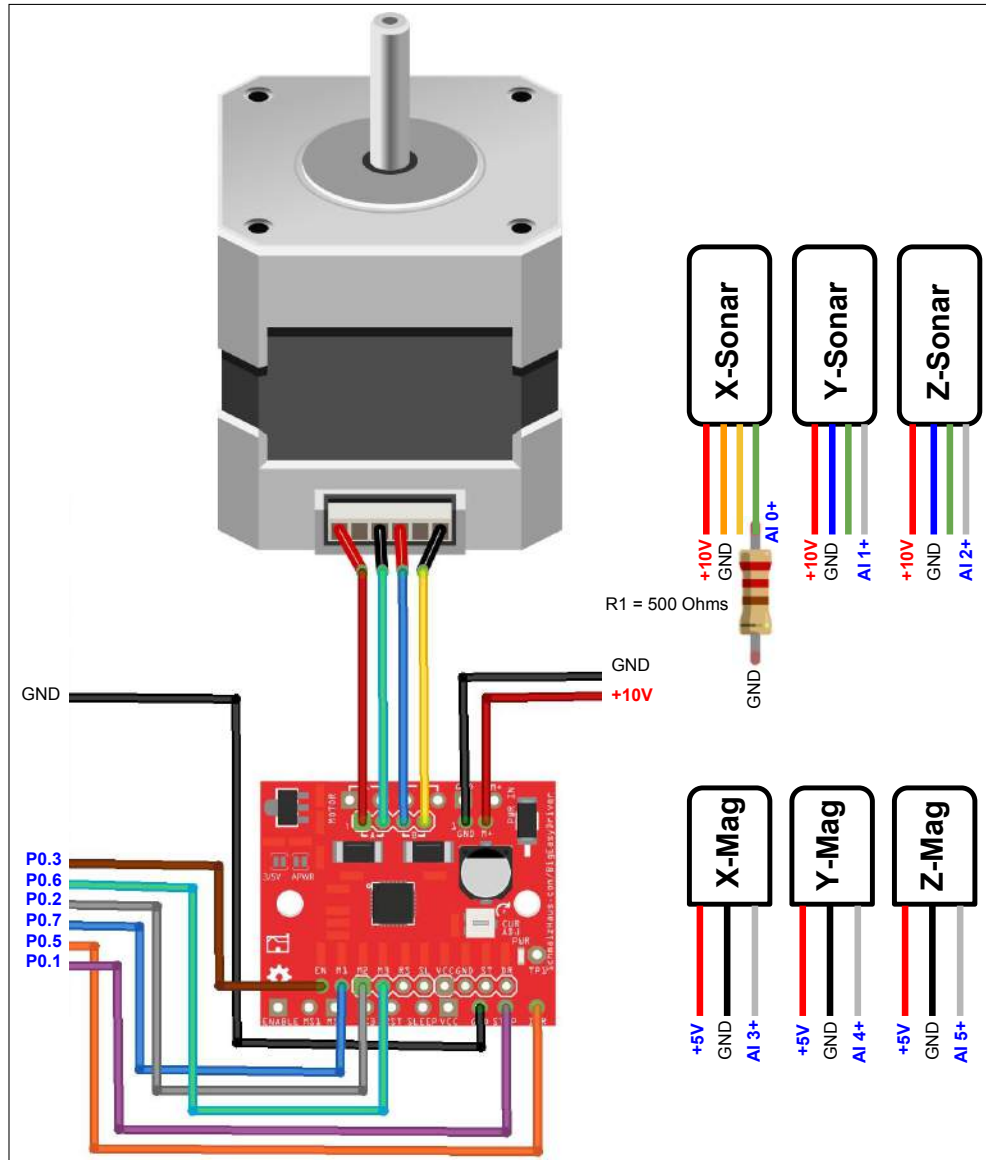


FIGURE 7. Wiring between magnetic field strength measurement system and National Instruments SCB-68A DAQ.

Images of the magnetic field strength measurement system wiring are provided in Figures 8 and 9. As shown in both Figures, a long, flexible ribbon cable connects the base breadboard next to the SCB-68A DAQ breakout board to the mobile one on the linear stage, to which the three ultrasonic sensors and hall sensors are wired to. This ribbon cable prevents the motion of the linear stage from interfering with the system's wiring. Next, an image of the fully integrated system (with the x-axis linear translator motor detached) is provided in Figure 10. Close-up images of the sonars and hall sensors, as well as their 3D-printed mounts, are also provided in Figures 11 and 12, respectively. Last, an image of a permanent magnet being field-cooled during an experiment is provided in Figure 13.

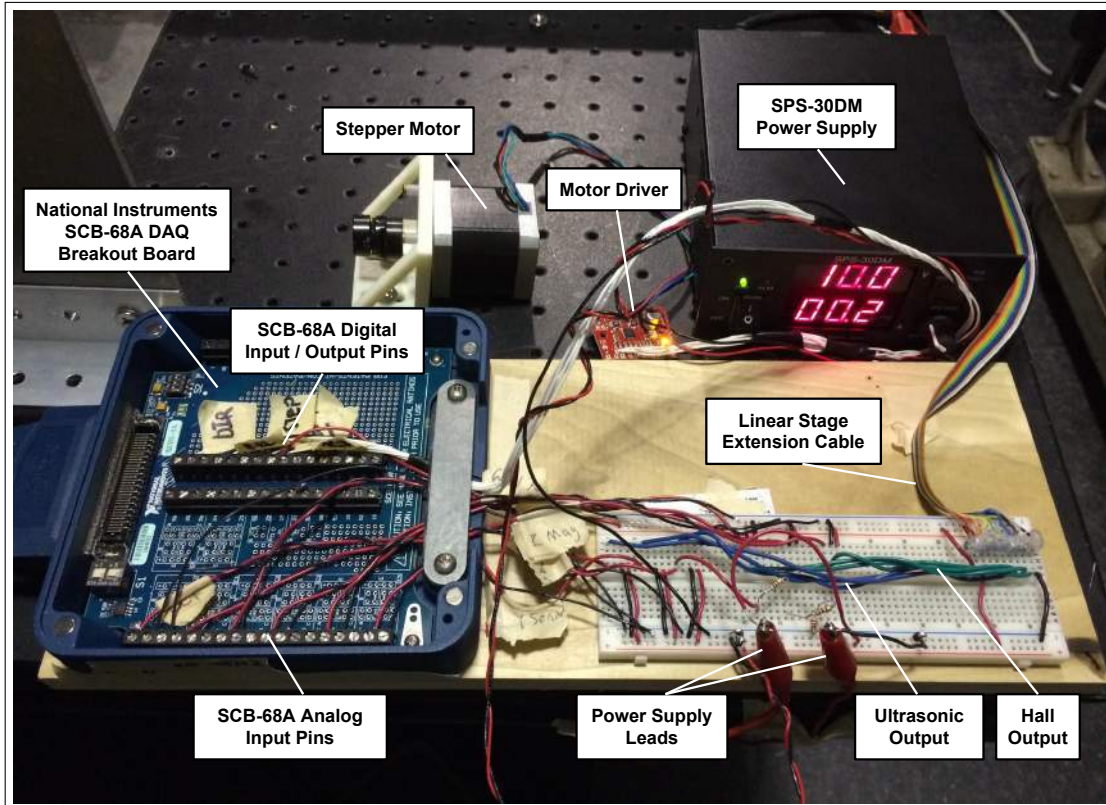


FIGURE 8. Wiring between magnetic field strength measurement system and National Instruments SCB-68A DAQ.

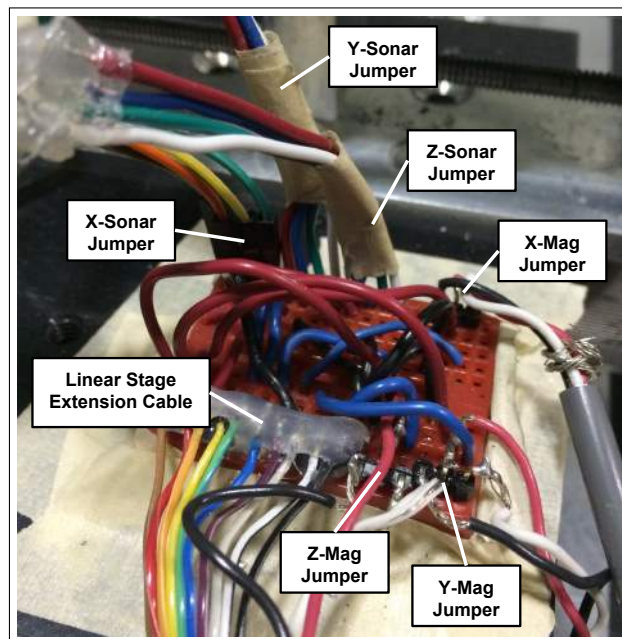


FIGURE 9. Wiring of the mobile breadboard on the linear stage, connected to the base breadboard via the linear stage extension cable.

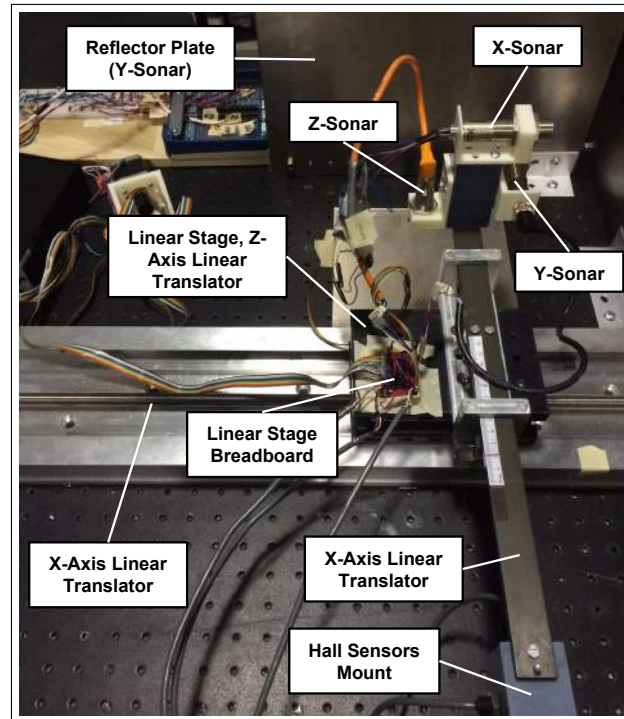


FIGURE 10. Fully integrated system (with the x-axis linear translator motor detached).

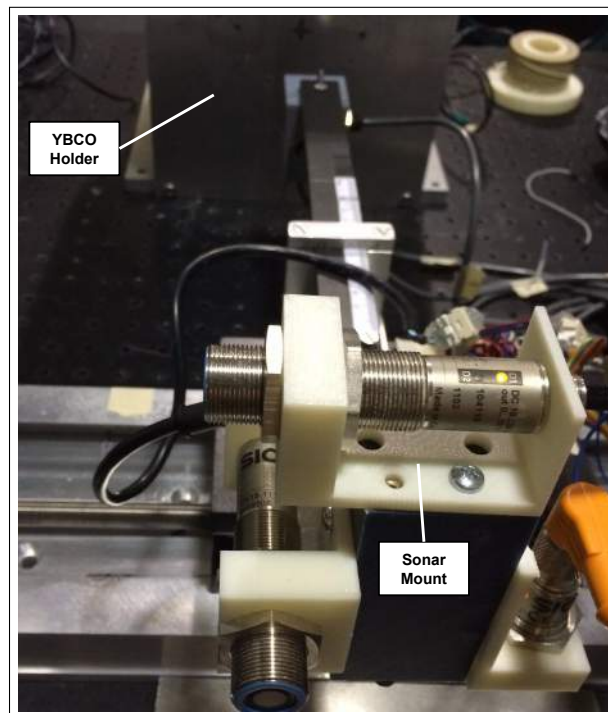


FIGURE 11. Close-up image of all three ultrasonic sensors and their 3D-printed mounts, designed by Kate Zhou. The YBCO holder, designed by McKenzie Read.

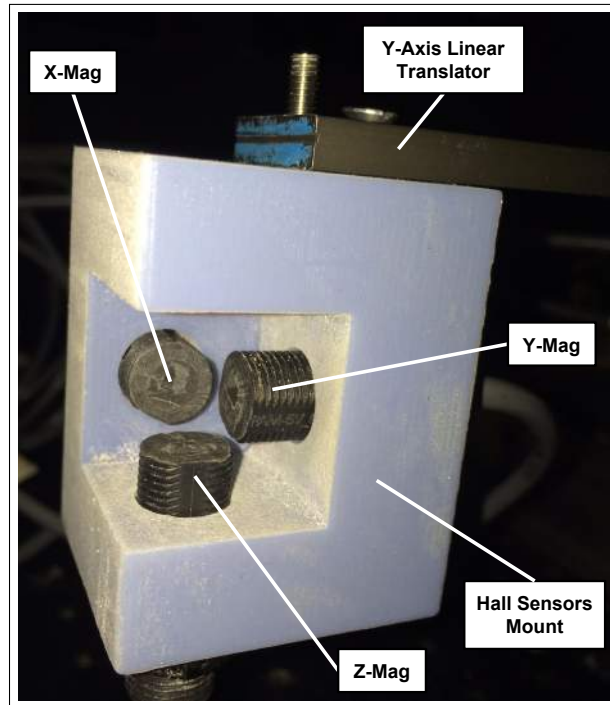


FIGURE 12. Close-up image of all three hall sensors and their 3D-printed mount, designed by Reginald Lin.

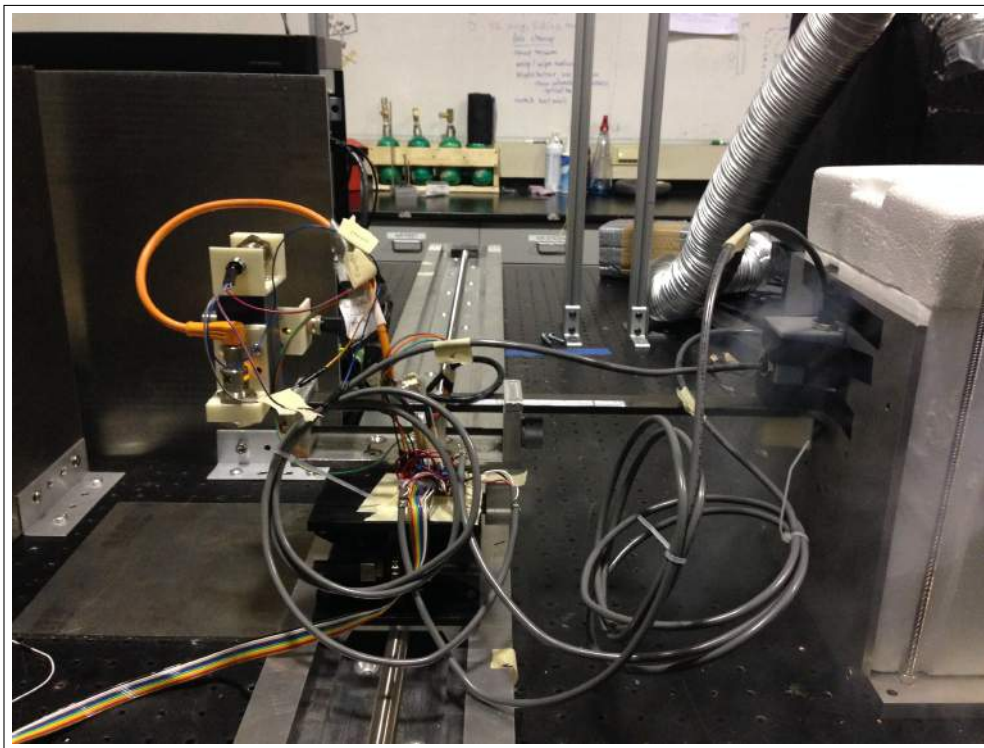


FIGURE 13. Image of permanent magnet being field-cooled during an experiment.

3.d. **Software Design.** The magnetic field strength measurement system is operated through a MATLAB graphical user interface (GUI) named `Mag_Meas_GUI`, as shown in Figure 14. Seven global structures, each holding unique data, are referenced throughout the callback functions of the GUI: `PARAMS`, `DAQD`, `DATA`, `GRAPHICS`, `TIMERS`, `SONARCAL`, and `MAGCAL`. These global structures come to define the functionality and organization of the GUI code. The data stored in each global structure, as well as the callback functions that reference these global structures, are thoroughly explained throughout the next subsections of this report.

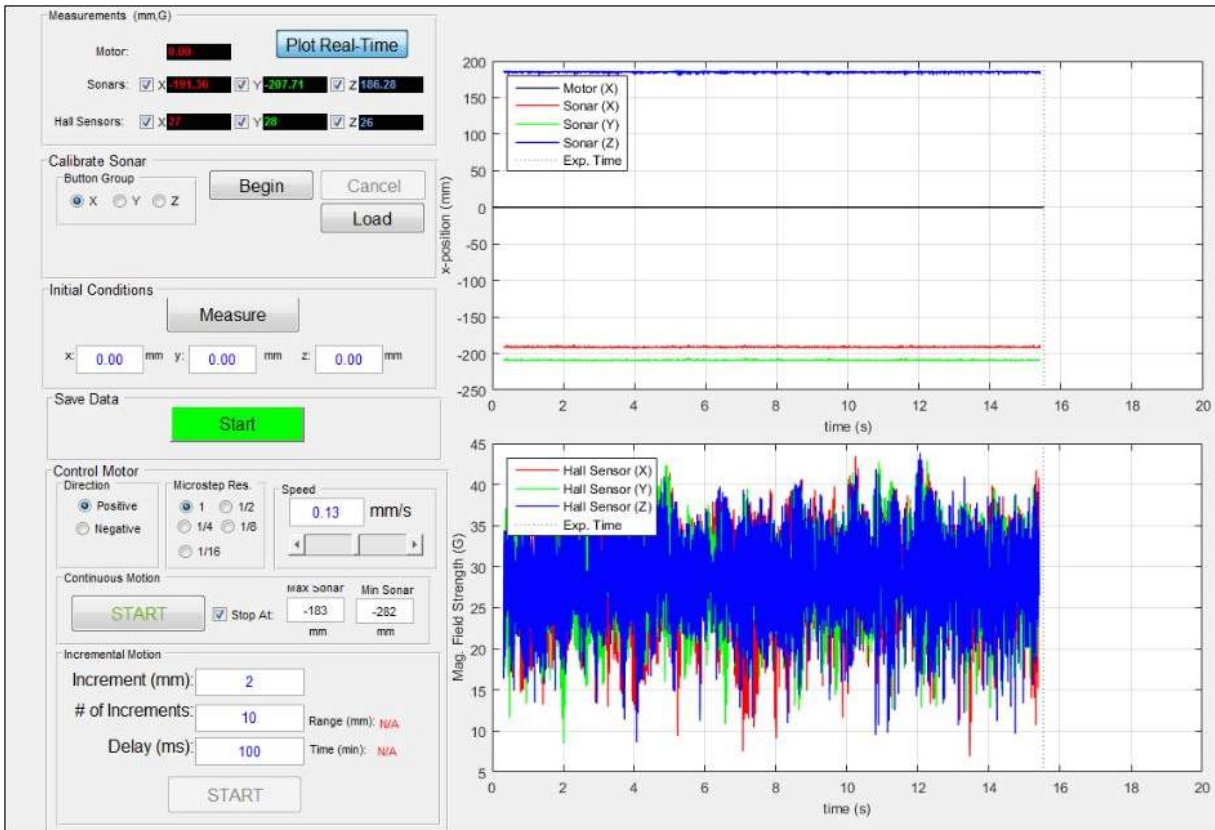


FIGURE 14. Magnetic field strength measurement system MATLAB graphical user interface (GUI).

3.d.1. *PARAMS.* The `PARAMS` global structure contains all of the user-defined parameters involved with the magnetic measurement system. Called during each instantiation of the GUI, `initParams.m` initializes the global parameters that should rarely change while using the system, yet might while, for example, re-configuring or adding functionality to the setup. In other words, `initParams.m` should be the only GUI function that a general user might alter while running an experiment in order to change specific parameters. For instance, `initParams.m` specifies a structure indicating the device and pin address of each DAQ signal (`p.daqSig`, declared in lines 5 through 16 of `initParams.m`), which could vary if the DAQ device and/or pins need to be changed. In addition, this function specifies the direction measured by each sonar with respect to the reflector plate

(`p.sonar.dir`, declared in lines 60 through 62), which would change if the sonar mount were to be reversed.

The other user-defined parameters are specified through the user interface window. These parameters might often change during the magnetic measurement experiments. They include the option of using each ultrasonic and hall sensor (i.e., reading the voltage of their associated analog input pin and plotting and saving the corresponding measurements), the direction and microstep resolution of the motor, and the lower and upper limits of the sonar measurements that enable each run to stop autonomously for increased efficiency. The initial user inputs of the GUI are read at the instantiation of the GUI through the call to `initUserInputs.m` and updated in the callback function of the GUI feature corresponding to each user-defined parameter.

3.d.2. *DAQD*. The *DAQD* global structure contains references to the analog and digital data acquisition sessions as well as pin numbers and values associated with those sessions. Like the other global structures, *DAQD* is initialized when the GUI is opened (i.e., in the sub-function `MagMeas_GUI_OpeningFcn` beginning in line 48 of `MagMeas_GUI.m`), through the call to `initDAQD.m`. *DAQ* sessions are created by passing the input 'ni' (i.e., the "National Instruments" vendor hardware id) into the function `daq.createSession` and adding digital input/output and/or analog input channels for each pin [7]. A snippet of code initializing the *DAQ* session for controlling the stepper motor driver is provided below in Figure 15.

```
% Initialize digital input/output pins
s.DIO = daq.createSession('ni');
addDigitalChannel(s.DIO, params.daqSig.DIR.device, ...
    params.daqSig.DIR.pin, 'OutputOnly');
addDigitalChannel(s.DIO, params.daqSig.STEP.device, ...
    params.daqSig.STEP.pin, 'OutputOnly');
addDigitalChannel(s.DIO, params.daqSig.M3.device, ...
    params.daqSig.M3.pin, 'OutputOnly');
addDigitalChannel(s.DIO, params.daqSig.M2.device, ...
    params.daqSig.M2.pin, 'OutputOnly');
addDigitalChannel(s.DIO, params.daqSig.M1.device, ...
    params.daqSig.M1.pin, 'OutputOnly');
addDigitalChannel(s.DIO, params.daqSig.EN.device, ...
    params.daqSig.EN.pin, 'OutputOnly');
```

FIGURE 15. Code snippet that creates the *DAQ* session used to control the stepper motor driver (lines 7 through 13 of `initDAQD.m`).

Because the stepper motor is driven at a different frequency than the sensor measurements, two *DAQ* sessions must be created: one that sends digital output commands to the stepper motor driver (`DAQD.sesh.DIO`), and another that periodically reads the analog input pins corresponding to the ultrasonic and Hall sensors (`DAQD.sesh.AI`). In lines 41 through 44 of `initDAQD.m`, the analog *DAQ* session is declared to read analog input

data at a frequency of `params.AI.sampleRate` Hz (as specified in `initParams.m`) and pass the available data to the listener function `daqDataReceived.m`. This listener solves for the sensor measurement values corresponding to the measured voltages of each analog input pin, and packs the results in the global structure `DATA`, which will be described in the next section of this report. When the motor is commanded to run, the Digital DAQ session periodically outputs motor commands through a timer function specified in the global structure `TIMERS`, which will also be described later in this report.

3.d.3. *DATA*. All of the relevant data collected using the magnetic measurement system are stored in the `DATA` global structure. In particular, this global structure stores the position measurements from the ultrasonic sensors, dead-reckoning propagation of the x-axis position from motor commands, and magnetic field strength measurements from the Hall sensors, as well as all corresponding time vectors. However, because new data is collected at a very high rate, old data must be regularly cleared to prevent memory overload. In fact, data stored in the `DATA` structure is cleared immediately after being plotted and saved to a csv file, as explained in the `TIMERS` subsection of this report. Data that must be temporarily held (e.g., during calibration, when hundreds of ultrasonic sensor measurements end up being taken) is stored in the field `DATA.hold` until the particular process is complete.

3.d.4. *GRAPHICS*. Declared in `initGraphics`, the `GRAPHICS` global structure specifies all parameters associated with data visualization in the MATLAB GUI. Most importantly, the `GRAPHICS` structure specifies the time window for which live data should be displayed (through `GRAPHICS.window.t`, declared to be 20s in line 4 of `initGraphics.m`). The user may wish to vary this parameter in order to display measured data for a longer period of time during an experiment. Other data saved in `GRAPHICS` include plotting and text-display handles as well as line style parameters.

3.d.5. *TIMERS*. The `TIMERS` global structure is used to initialize and control the two timers used in the GUI. The first timer, `TIMERS.motor`, sends step commands to the motor driver periodically (through the call to `motorDriver.m`), with a fixed spacing of `PARAMS.timer.lag` seconds when the motor is enabled. The user enables motor control by clicking the “Start” button within the “Control Motor” panel of the GUI, and can specify positive or negative x-axis linear motion, the micro-step resolution, and the speed of the motor. As the motor sends step commands, the motor position is propagated via dead reckoning and displayed on the upper axis of the GUI.

The second timer function, `TIMERS.plotting`, is also called periodically with a fixed spacing of `params.timer.delay` seconds in the function `initTimers.m`. This timer uses the function `plottingTimer.m`, which plots incoming sensor measurements on the GUI axes, prints the data to an output file if the “Start” button was pressed in the “Save Data” panel, and resets the `DATA` arrays to prepare for new incoming data. Because the dead-reckoning performed while commanding motor steps requires knowledge of the previous position, the last element of the motor data arrays is saved (in lines 148-150 of `plottingTimer.m`), while all of the previous data is removed from the global `DATA` structure.

Although implemented efficiently, plotting requires a great deal of memory, primarily due to the large amounts of data being plotted at a relatively high rate. Thus, when



the motor is being ran during an experiment, live plotting should be disabled by toggling the button “Plot Real-Time” in the “Measurements” panel. This button starts and stops `TIMERS.plotting` in order to allocate more processing power to `TIMERS.motor` to increase the rate that it calls `motorDriver.m`. This allocation, in turn, increases the speed of the motor.

3.d.6. *SONARCAL & MAGCAL*. The global data structures `SONARCAL` and `MAGCAL` are used to store the calibration results – in particular, the gain and offset – of each ultrasonic and Hall sensor. By making these variables global, calibration must only be ran once per MATLAB session. Further, to load ultrasonic calibration results from a previous MATLAB session, the user can select a particular axis in the “Calibrate Sonar” panel, click the “Load” button, and find the `.mat` file automatically saved after calibration was performed for the ultrasonic sensor of that axis. On the other hand, `MAGCAL` simply uses the gain and offset values specified on the Sensor Solutions  $\pm 1$  kG hall sensor data sheet: gain = 0.4 G/mV, offset = -1000 G. To correct for sensor offset and drift, “zero-field” measurements should be taken at the start of each set of experiments, in which magnetic field strength measurements are taken about the desired three-dimensional grid without any magnetic sources present. The measured zero-field grid is then subtracted from the magnetic field strength measurements of each following experiment run.

## 4. ANALYSIS

### 4.a. Calibration and Characterization.

4.a.1. *Ultrasonic Sensors*. To calibrate an ultrasonic sensor, the user clicks the “Begin” button in the “Calibrate Sonar” panel of the magnetic field strength measurement system GUI and follows the instructions provided within the panel. These instructions direct the user to mount the sonar at six locations on the optical table with respect to a reflector plate. Referring to Figure [16](#), these distances, labeled  $d_{sonar}$ , are defined by the center of the sonar mounting hole (i.e., the reference point to which the ultrasonic distance measurements are calibrated) and the center of the reflector plate mounting hole. By using the known distance between the reflector plate center and its mounting hole,  $d_{offset}$ , as well as the reflector plate thickness,  $t_{plate}$ , the relative position between the reflector plate and sonar reference point,  $x$ , can be solved for using the following equation:

$$(1) \quad |\vec{x}| = |\vec{d}_{sonar} + \vec{d}_{offset}| - \frac{t_{plate}}{2}$$

Making sure that  $|\vec{x}|$  lies within the ultrasonic sensor measurement window, the  $d_{sonar}$  calibration distances were determined to be incremented by 1 in. between 4 in. and 9 in., inclusively. Example ultrasonic calibration results are provided below in Figure [17](#).

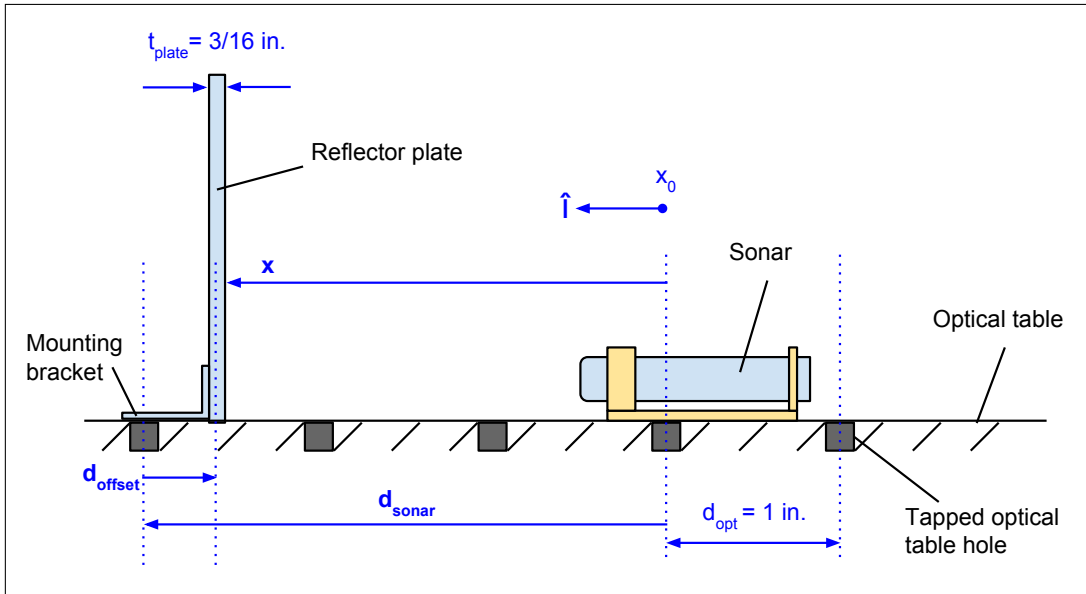


FIGURE 16. Illustration of sonar calibration position vectors.

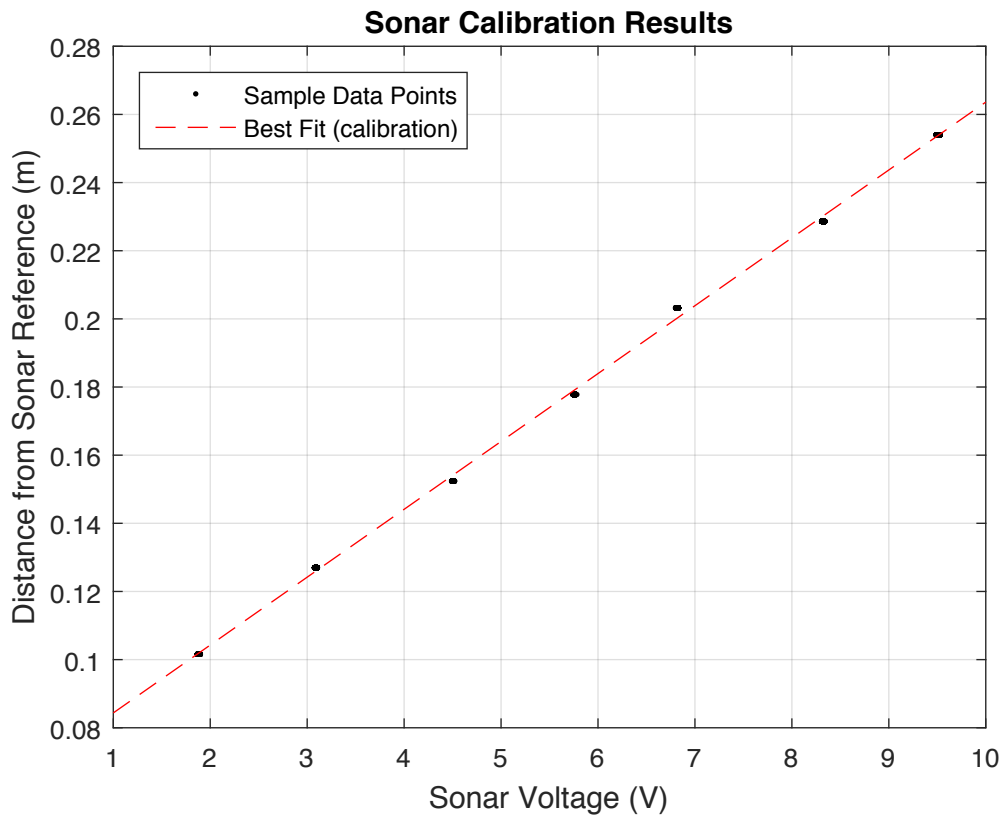


FIGURE 17. Example ultrasonic calibration results, indicating a gain of 19.92 mm/V and offset of 64.4 mm.

According to its data sheet, the SICK UM12 ultrasonic sensor has a repeatability rating of  $\pm 15\%$ . Since the working range of this sensor is specified to be 40 mm to 240 mm, this rating corresponds to  $200 \text{ mm} \cdot 0.0015 = 0.3 \text{ mm}$ . When a calibrated sonar was tested at a mean measurement distance of 221.8 mm, the standard deviation of measurements taken over approximately 100 seconds was found to be 0.123 mm, as shown below in Figure 18, which does in fact fall within the repeatability rating.

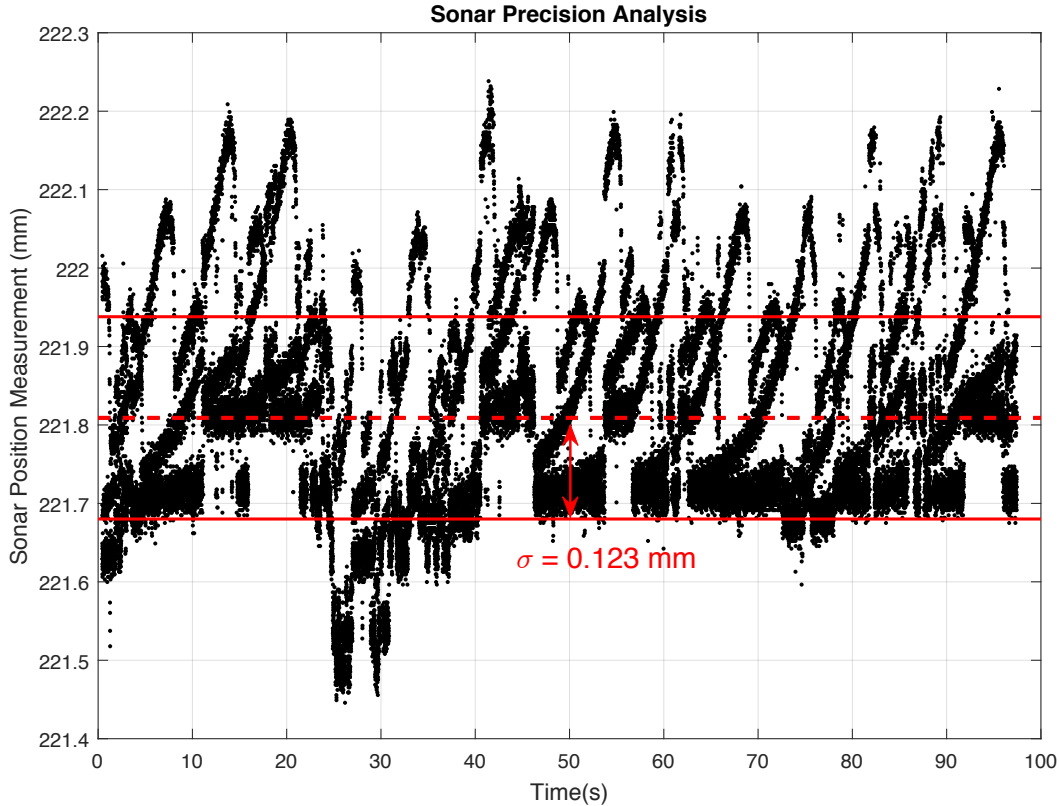


FIGURE 18. Example measurements taken using a calibrated sonar directed at a stationary reflector plate, indicating a measurement precision with  $\sigma = 0.123 \text{ mm}$ .

4.a.2. *Stepper Motor*. Although one rotation of the linear translator shaft is specified to translate the linear stage by 0.2 mm, there are several reasons why the x-axis distance propagation resulting from stepper motor commands should be calibrated. First and foremost, the accuracy and precision of this number is not known; it was obtained simply by a label on the linear stage indicating that five rotations correspond to 1 cm. Further, calibration would take into account general inconsistencies in the motion of the shaft caused by, for example, friction in the shaft and slippage of the shaft coupling. Last, through multiple calibration runs, a better estimation of the process noise could be found, which would enable more effective filter gains and, in turn, a better estimate of x-axis position and uncertainty.

As illustrated below in Figure 19, the initial overall trend between the error in position between measurements of a recently-calibrated ultrasonic sensor and predictions derived

from the number of motor steps was found to be  $0.07291 \mu\text{m} / \text{step}$ . Given that the motor undergoes 400 steps/rev, and assuming that each rotation of the shaft causes approximately 0.2 mm of linear translation, this error corresponds to 0.029 mm / rotation, or about 15% of the linear stage's translation. To account for this error, then,  $0.07291 \mu\text{m}$  was added to `PARAMS.motor.distPerStep` in line 51 of `initParams.m`.

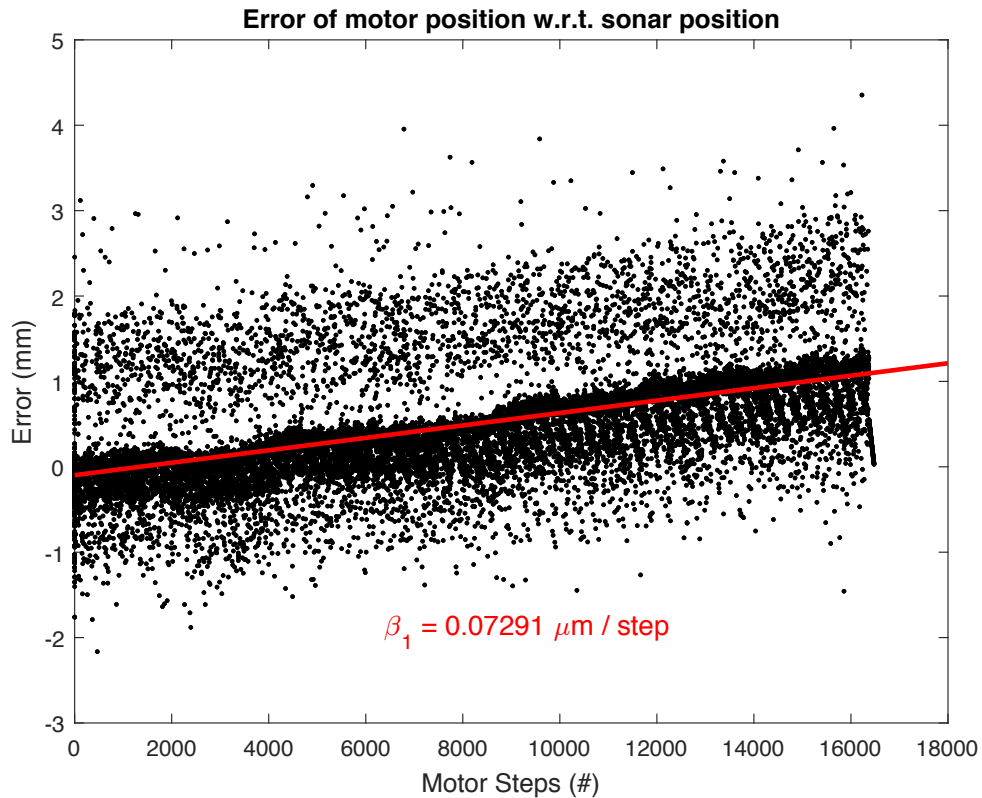


FIGURE 19. Initial trend in position error between measurements of a recently-calibrated ultrasonic sensor and predictions derived from the number of motor steps.

As illustrated below in Figure 20, which was taken using x-axis ultrasonic measurements and stepper motor prediction estimates from a future run after the above adjustment was made, this modification had essentially mitigated the discrepancies between sonar and stepper motor position measurements, which was calculated to be  $-0.0058 \mu\text{m} / \text{step}$  (i.e., about 1.2% of the stage's linear motion). This error is most likely attributed to other sources of experimental error besides that between dead-reckoning and the actual linear motion. In fact, as illustrated in Figure 20, there seems to be an abnormal bump in position error at around 8000 steps. Because the stepper motor could not impart such a jump in position, this must be attributed to the measurements of the ultrasonic sensor. Such an error could have been caused by the reflector plate wobbling from, for example, a lab member accidentally shaking the optical table during the experiment.

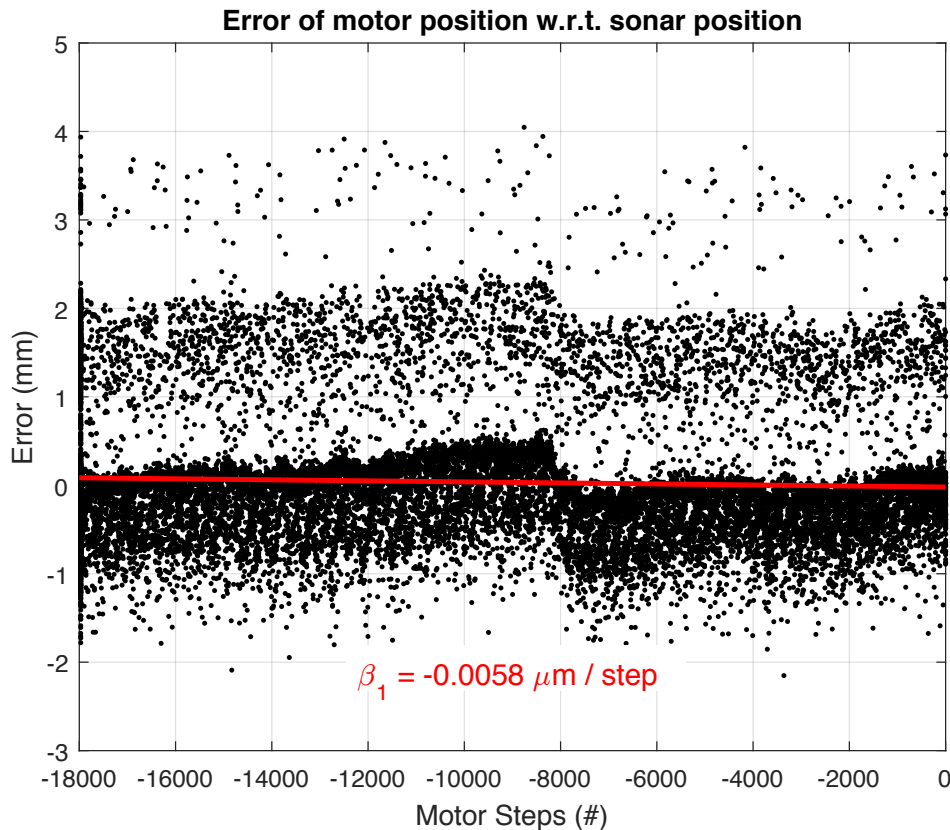


FIGURE 20. Trend in position error between measurements of a recently-calibrated ultrasonic sensor and predictions derived from the number of motor steps after adjustment to `PARAMS.motor.distPerStep`.

4.a.3. *Reflector Plate Tilt Planes.* Because the x- and y-axis ultrasonic aluminum reflector plates tend to slightly tilt when mounted vertically onto the optical table, the raw ultrasonic measurements are slightly unreliable. For example, if the y-position of the linear stage were to remain constant while the z-position is changed, the y-axis ultrasonic measurement would vary if the y-axis sonar plate were tilted forwards or backwards, in turn appearing as though the y-position of the linear stage is changing. To account for these false variations, the tilt planes of each reflector plate were characterized. The challenge in performing this characterization was that, in order to characterize a particular plate in the system's defined coordinate system, the location in the plate's intended plane must be known to relate the ultrasonic measurements to plane coordinates. For example, when characterizing the y-axis reflector plate, the exact x- and z-positions must be known to correctly correlate the y-axis ultrasonic measurement with the plane's true y value. However, the other ultrasonic sensors take measurements from other potentially tilted plates; in other words, they require prior knowledge of the other reflector plates' tilt planes to correct their measurements. This co-dependence leads to an iterative solution in which all plates are initially assumed to be perfectly aligned with the system's coordinate system and adjusted iteratively in a way that allows the plane equations to converge [8]. The final characterized tilt planes for each ultrasonic reflector are provided below in Figure 21

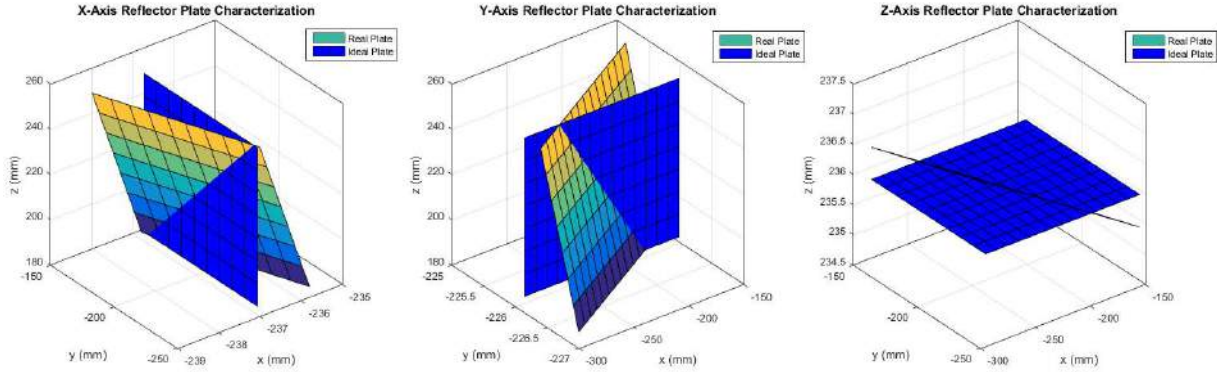


FIGURE 21. Characterization of ultrasonic reflector plate tilt planes.

4.b. **Post-Processing.** The post-processing of the data collected throughout a set of experiments conducted with the magnetic field strength measurement system is summarized in the IDEF0 diagram provided below in Figures 22 and 23. An IDEF0 follows a particular format: a particular function block takes in an input (e.g., material, information), satisfies controls / constraints (i.e., informational input that influences activity), utilizes mechanisms (i.e., subsidiary functions that will not be analyzed further), and returns an output (e.g., alternate material, information). They are also used to more clearly illustrate a use case or other functional processes, which can help identify additional system requirements needed to execute the function block or intermediate / lower-level processes.

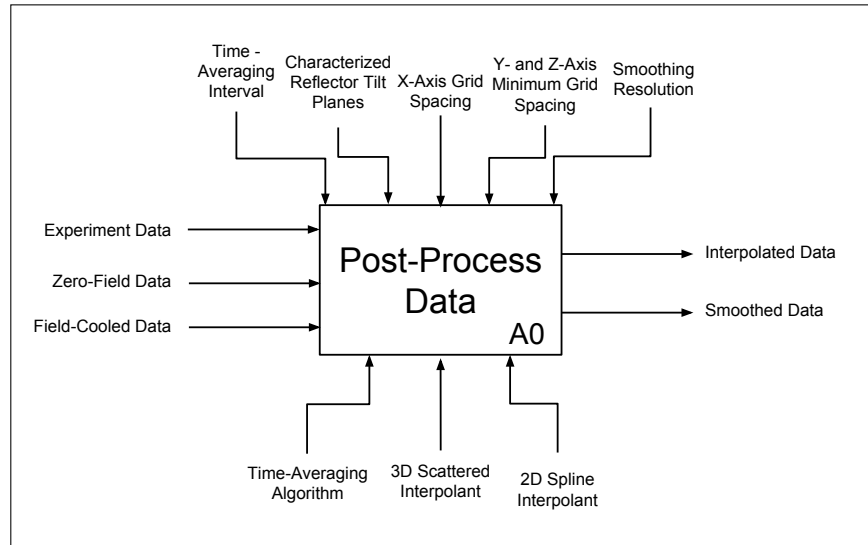


FIGURE 22. First level of post-processing IDEF0 diagram.

The three sets of input data to the post-processing algorithm – “Experimental Data”, “Zero-Field Data”, and “Field-Cooled Data” – are selected through running the function `file_selector.m`. This function prompts the user to select all related files using MATLAB’s `uigetfile()` function, and packs all file names into a `.mat` file saved under the name `p.file.names.all_files` declared in line 14 of `initialize_params.m`.

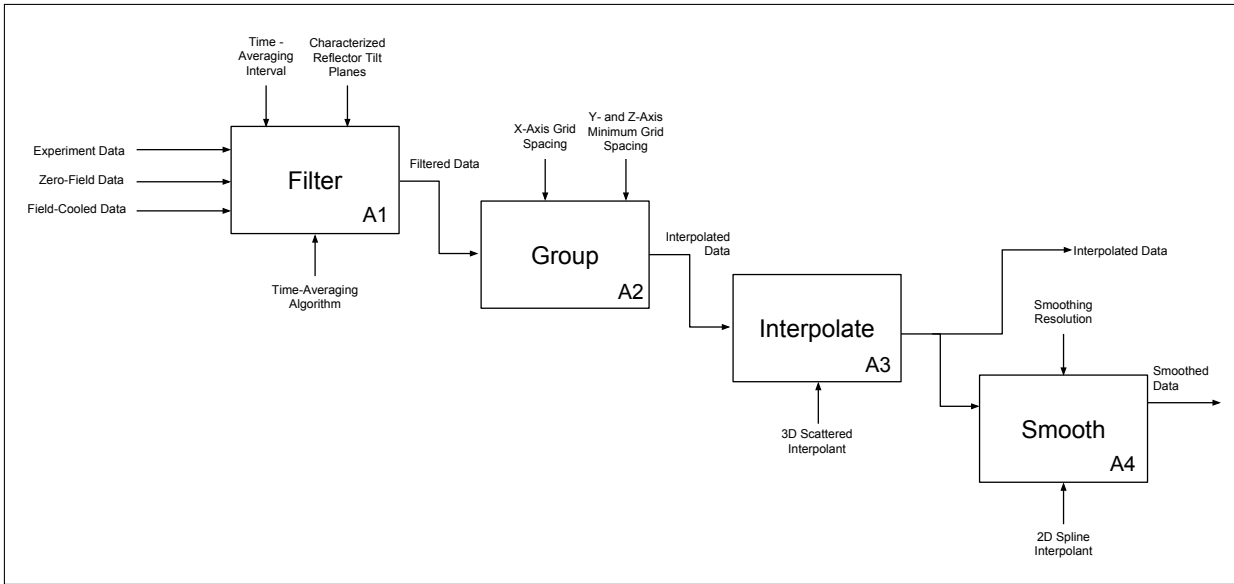


FIGURE 23. Second level of post-processing IDEF0 diagram.

Note that, for all experiment files, both the “\_DIO” and “\_AI” extensions must be selected to gather the digital input/output and analog input data of each experiment, respectively.

4.b.1. *Filter*. The first process of the post-processing algorithm consists of gathering and filtering the data. As depicted by the “Filter” block in Figure 23 and implemented in `filter_data.m`, this function is responsible for looping through each experiment file and extracting all of its data, adjusting ultrasonic measurements based on the inputted field-cooled position data as well as the reflector tilt planes characterization, and filtering the data by performing a time-average of window `p.time_avg_interval` seconds across all data. Like every other separate post-processing MATLAB function, the filtering function calls `initialize_params.m` if no inputs are specified, and can thus automatically locate the specified input data files, assuming the name and locations of these files do not change. The output data structure of `filter_data.m` – `filtered_data`, which is automatically saved to `p.file_names.filtered_data` – is depicted below in Figure 24. Formatted as a struct array in which each element corresponds to the data set of a particular experiment run, `filtered_data` stores the mean and standard deviation from each time-averaged interval of magnetic field strength and corresponding  $(x, y, z)$  position of each Hall sensor (i.e., `mag.x`, `mag.y`, and `mag.z`). Note that each Hall sensor has a different  $(x, y, z)$  position for the same magnetic field strength measurements since, as shown in Figure 12, each is slightly offset from one-another. Also note that the dotted symbols used in Figure 24 indicate that the associated struct field follows the same format as the field labeled to its right.

4.b.2. *Group*. Once the experimental data is filtered, it must be grouped into particular XY, YZ, and XZ planes in order to be useful for analysis and visualization; in other words, because data cannot be collected along a perfect 3D grid, and since the ultrasonic sensor measurements are inherently noisy, the position data must be “de-scattered”

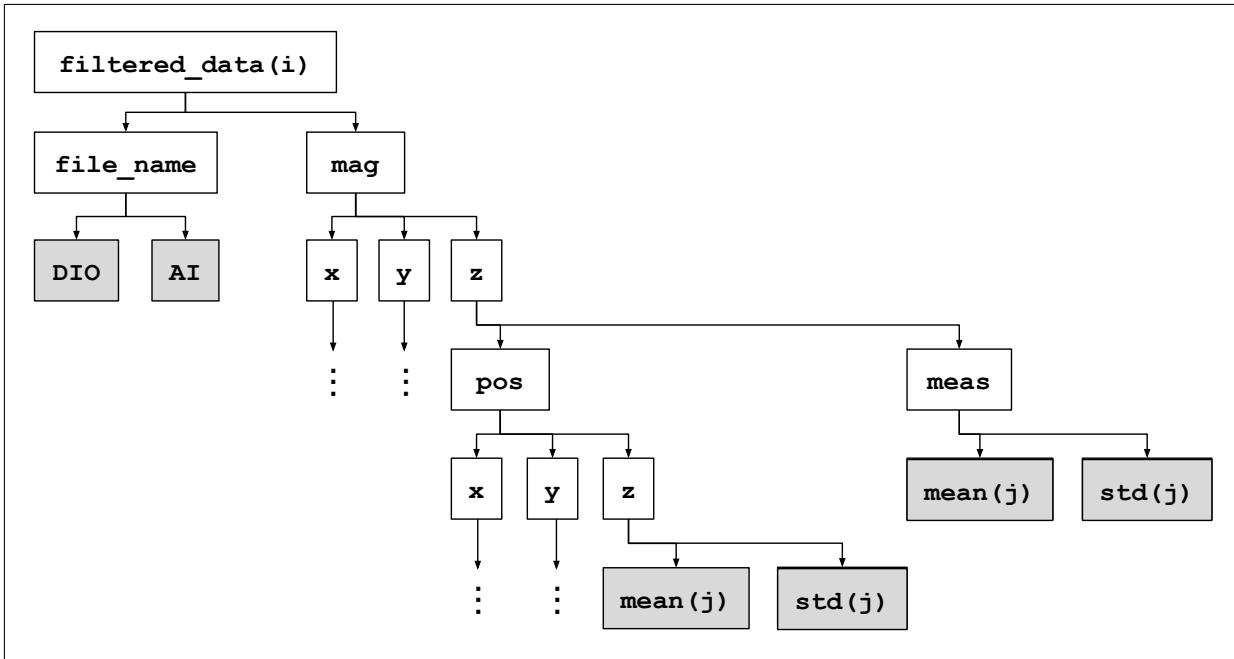


FIGURE 24. Depiction of output structure array `filtered_data`.

into common grid planes, and magnetic field strength measurements must be interpolated to match these slightly shifted positions. This entire process is captured in the “Group” and “Interpolate” blocks of the IDEF0, which occurs through a call to the function `interpolate_data.m`. The “Group” block, which is executed in the function `initialize_interpolated_data.m`, first calculates the mean y- and z-position of each run (which is held constant throughout the run). Next, this mean y- and/or z-position is added to the list of other mean y- or z-positions with a difference in overall mean below `p.interpolate.max_diff`, or added to a new list (i.e., a new grouping) if none meets this criteria. After looping through each run, the grouping function sets grid coordinates at the overall means of each y- and z-mean value list, and splits up the x-coordinates using an interval size of `p.interpolate.x_spacing` between the largest minimum and smallest maximum x-position of all experiment data sets (to prevent extrapolation). Finally, each XY, YZ, and XZ plane is defined using these grid coordinates along with MATLAB’s `meshgrid()` function in lines 70 through 81 of `initialize_interpolated_data`. For each plane, each out-of-plane grid coordinate is stored as a matrix of the same dimensions as that plane’s `meshgrid`. Similarly, the magnetic field strength measurements, which will be interpolated at each (x,y,z) grid coordinate through the “Interpolate” post-processing function, is stored in a cell array for each plane in which each element corresponds to the matching out-of-plane value. The output of `initialize_interpolated_data.m` (`interpolated_data`) is illustrated below in Figure 25.

4.b.3. *Interpolate*. As introduced above, the “Interpolate” block performs trilinear interpolation to evaluate the magnetic field strength at the (x, y, z) grid coordinates of every grouped XY, YZ, and XZ plane using all experimental data from each Hall sensor. The



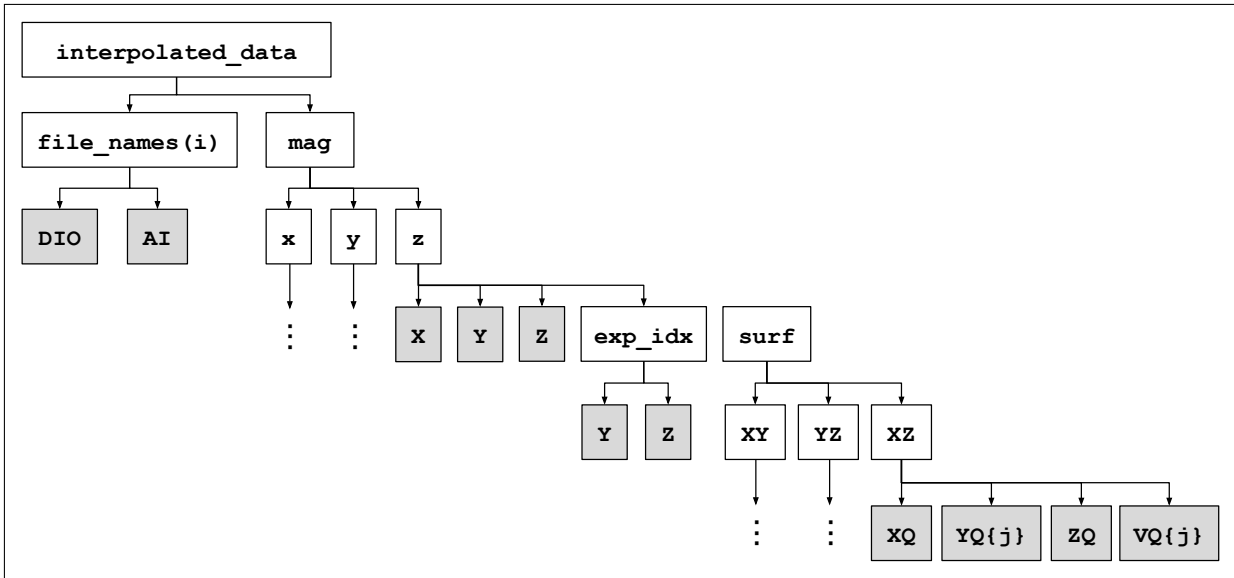


FIGURE 25. Depiction of the interpolated\_data data structure

trilinear interpolation code performed for the  $k^{\text{th}}$  out-of-plane value, *plane*-plane, and *ax*-axis magnetometer is provided below in Figure 26. In this code snippet, which corresponds to lines 66-67 of `interpolate_data.m`, the variables *x*, *y*, *z*, and *v* are all corresponding raw position and magnetometer field strength vectors of the *ax*-axis Hall sensor, while *Q.X*, *Q.Y*, and *Q.Z* are the grid coordinates of the particular plane and out-of-plane value, formatted as a mesh grid. The 'linear' and 'nearest' inputs specify the interpolation and extrapolation methods, respectively.

```

% Execute Trilinear Interpolation
F = scatteredInterpolant(x,y,z,v,'linear','nearest');
interpolated_data.mag.(ax).surf.(plane).VQ{k} = F(Q.X,Q.Y,Q.Z);

```

FIGURE 26. Code snippet to perform trilinear interpolation.

4.b.4. *Smooth & Visualization GUI*. In order to visualize the trilinearly interpolated data, another GUI was developed: `GUI_interpolation.m`. This GUI, which will be illustrated in the following section of this report, not only plots the filtered, interpolated magnetic field strength measurements for each Hall sensor and grouped grid plane, but also enables the user to smooth the interpolated data – the last block of the IDEF0 specified in Figure 23. The data is smoothed by applying a spline fit to a uniform meshgrid with a resolution controllable by the user. Smoothing and its visualization is performed in the subfunction `spline_and_plot_data()` within `GUI_interpolation.m`, while visualization of the interpolated data occurs in the subfunction `plot_interpolated_data()`.

## 5. RESULTS

Visualizations of the interpolated magnetic field strength measurements using the GUI opened through running `GUI_interpolation.m` are provided below in Figures 27 through 30 for an experiment in which a spherical permanent magnet was placed at the center of the YBCO holder, to the right of the linear translator, in the negative y-hat direction. The button “Interpolate Data Sets” in the GUI allows the user to select all data required for post-processing the experiment data and performs the entire post-processing algorithm, while “Load Interpolation Data” loads and plots saved interpolation data (which is done at the end of the call to `interpolate_data.m`). Using the visualization GUI, the user is able to select the magnetometer, plane, and out-of-plane value to display the corresponding results. Further, the user has the option of smoothing the data by selecting the “Spline Fit” check box, and can also set the resolution of the smoothing grid. The effect of smoothing the interpolated data can be observed by comparing Figures 28 and 29, which illustrate interpolated and smooth-interpolated magnetic field strength measurements, respectively, for the z-axis accelerometer in the XY plane. Note that the positions provided in these plots are measured with respect to the reflector plates.

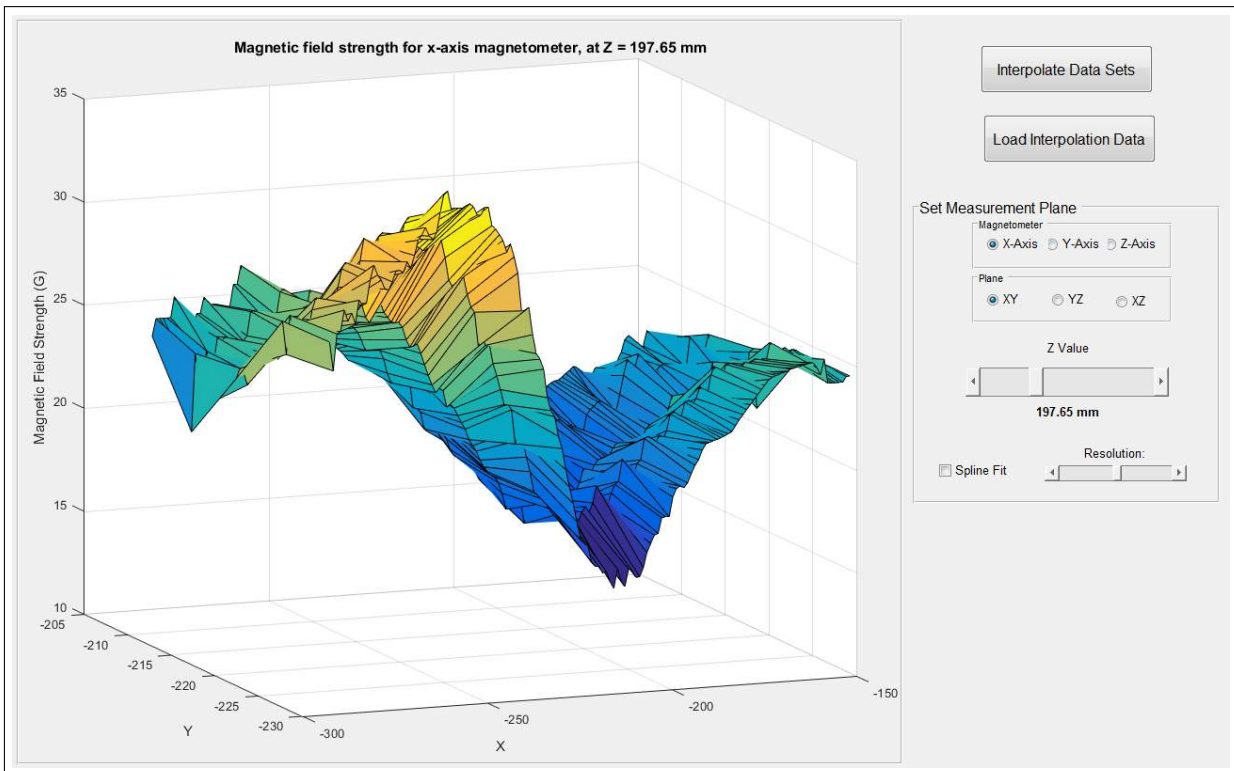


FIGURE 27. Visualization of x-axis magnetic field strength in the XY plane

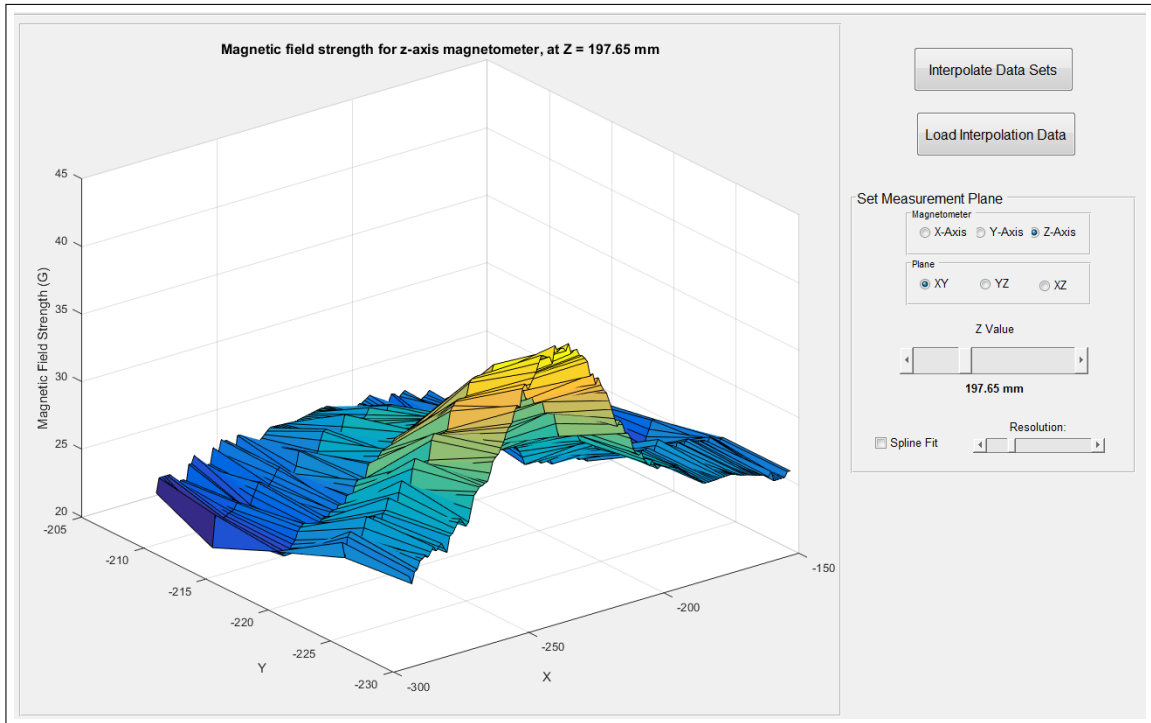


FIGURE 28. Visualization of z-axis magnetic field strength in the XY plane

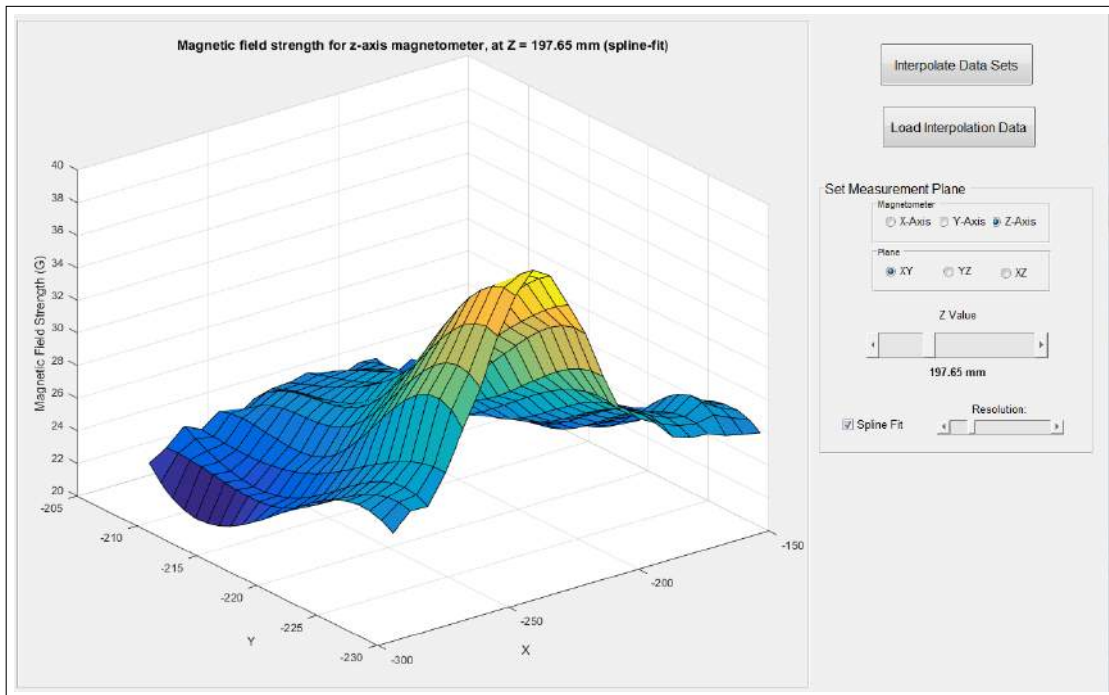


FIGURE 29. Visualization of smoothed z-axis magnetic field strength in the XY plane

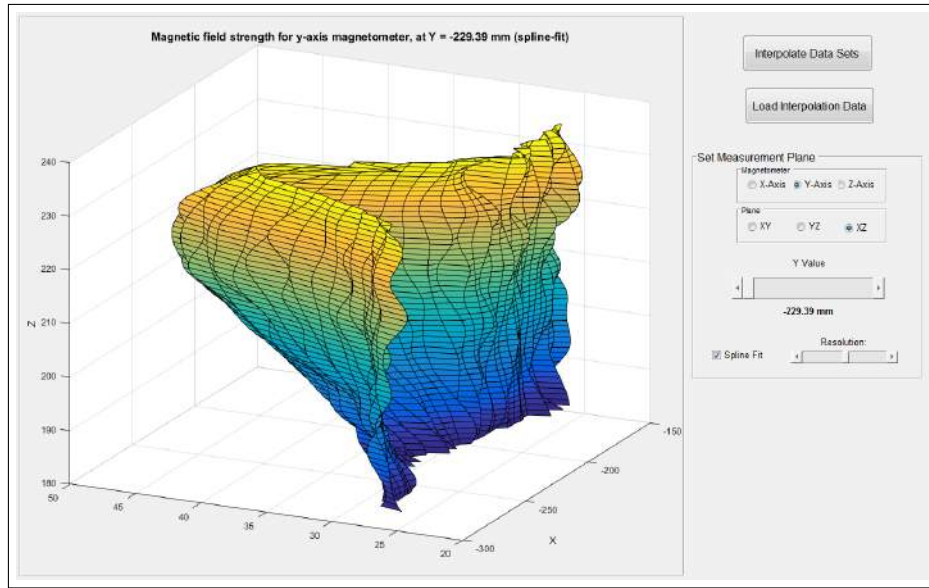


FIGURE 30. Visualization of smoothed z-axis magnetic field strength in the XZ plane

To further smooth the interpolated data, code was written to automatically fit a multivariate normal distribution to an interpolated set of magnetic field strength data in a particular plane. By running the script `gaussian_fit.m`, the Gaussian fits shown below in Figures [31](#) and [32](#) were achieved.

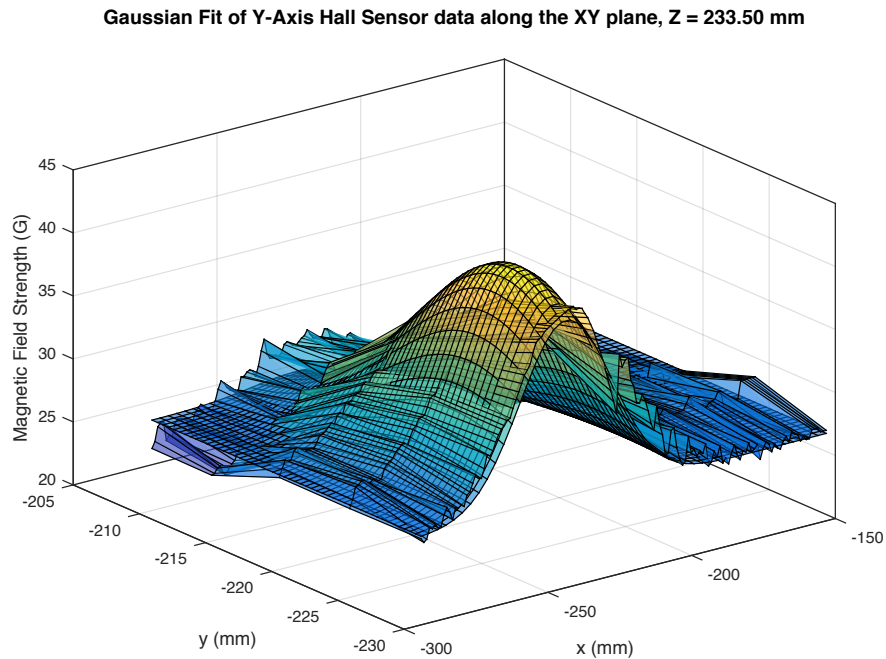


FIGURE 31. Gaussian fit overlaying interpolated y-axis Hall sensor data in the XY plane.

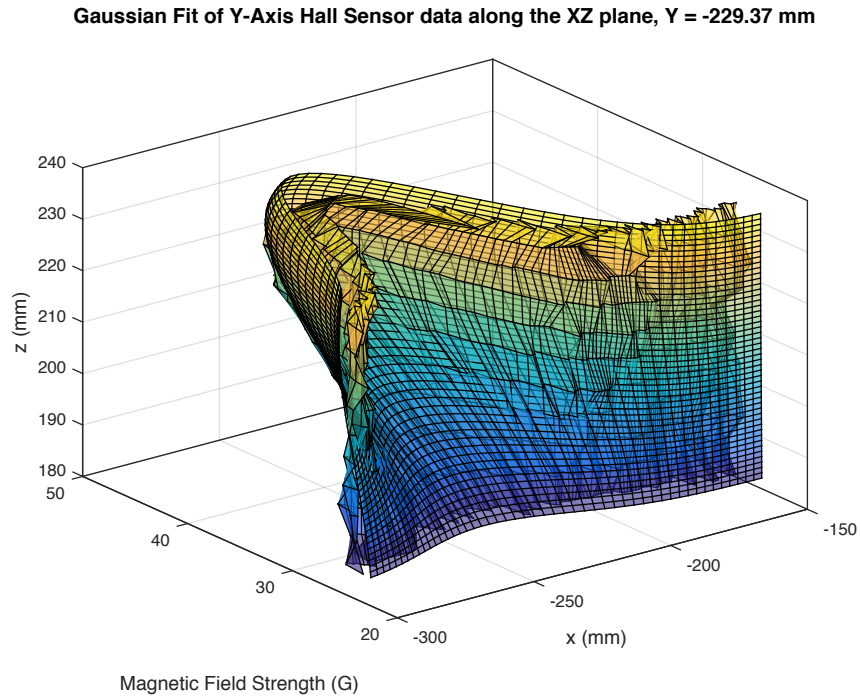


FIGURE 32. Gaussian fit overlaying interpolated y-axis Hall sensor data in the XZ plane.

## 6. CONCLUSION

The results above confirm that the magnetic field strength measurement system is capable of successfully measuring the magnetic field strength about the YBCO holder induced by a permanent magnet as well as corresponding position vectors. Through the use of the developed MATLAB GUI and motorized linear translator, experiments conducted using the magnetic field strength measurement system are able to be ran easily and efficiently, enabling a great deal of empirical data to be collected to help Cornell's flux-pinning research team make refinements to the ideal image dipole model to – in particular, to help understand the finite-volume effect of type-II superconductor flux-pinning. Further, the trilinear interpolation and smoothing algorithms described in the Analysis section of this report serve to enable the initial visualization and analysis of the measured data. Experiments conducted towards the end of this semester in which permanent magnets were field-cooled and exposed to electromagnetic actuators will be analyzed using the tri-interpolation method and used to validate the team's refined flux-pinning models.

While the design and implementation of a working magnetic field measurement system was successful, several key steps must be executed to fully complete the system and the analysis of its results. First, although the visualizations provided in the Results section of this report follow the expected trends of a single dipole's magnetic field, further analysis must be conducted to validate and verify the system's design. In particular, the interpolated magnetic field strength measurements must be compared to theoretical values predicted for a single dipole at the measurement position vectors. Second, positioning relative to the field-cooled position and magnetic field strength relative to the zero-field measurements must be implemented in `filter_data.m`. Further relative position adjustments must also be made due to the different offsets of each Hall

sensor with respect to the field-cooled permanent magnet; currently, the position vectors are relative to the ultrasonic reflective plates, and thus absolute position is not completely informative. Last, a more accurate Gaussian fitting method must be implemented to allow for more a effective visualization and representation of the measured magnetic field strength data.

## REFERENCES

- [1] Shoer, J. and Peck, M. *Fluxed-Pinned Interfaces for the Assembly, Manipulation, and Reconfiguration of Modular Space Systems*. AIAA Guidance, Navigation, and Control Conference, Honolulu, HI. August, 2008.
- [2] Zhu, F., Caracciolo, R., and Peck, M. *Spacecraft Docking with Type II Superconductor Flux Pinning and Potential Energy Capture*. AIAA Guidance, Navigation, and Control Conference. 2015.
- [3] Jones, L., Wilson, W., and Peck, M. *Design Parameters and Validation for a Non-Contacting Flux-Pinned Docking Interface*. AIAA Guidance, Navigation, and Control Conference, Portland, OR.
- [4] Lu, Y., Zhang, M., and Gao, D. *Connection Stiffness and Dynamical Docking Process of Flux Pinned Spacecraft Modules*. 2014.
- [5] Shoer, J., and Peck, M. *Stationkeeping of a Flux-Pinned Satellite Network*. 2010.
- [6] Shoer, J., and Peck, M. *Reconfigurable Spacecraft as Kinematic Mechanisms Based on Flux-Pinning Interactions*. *Journal of Spacecraft and Rockets*, Vol. 46, No. 2, 2009, pp. 466-469.
- [7] MATLAB DAQ Documentation: <http://www.mathworks.com/products/daq/>.
- [8] Zhou, K. End-of-Semester Report for Flux-Pinning Research Team. Fall 2015.

## 7. APPENDIX - SENSOR TRADE STUDIES

Name	Axes	Sensitivity	Range	Data face	Inter-	Price (USD)
Analog Hall Sensor	1	2.5 mV/G	1 kG	Analog		79.70 (39.85 for $\geq 10$ )
5070 FW Bell Gaussmeter	3	1G	2 kG	Scope		500
Vector/Magnitude Gaussmeter Model VGM	3	0.01G	800 G	USB Serial		710
Hall-effect gaussmeter and kilo gaussmeter	1	0.001G	20 kG	scope		500-700
DC Gaussmeter Model GM2	1	0.1 G/step	20 kG	probe		560
Two-point programmable hall analog sensor	1	1.5mV/G	1500 G	analog		135.56

TABLE A1. Trade study of magnetic field strength sensors, with selected sensor highlighted.

Name	Sensitivity	Range	Data face	Inter-	Price (USD)
VL6180 Range Finder Sensor	2mm	0-100mm	I2C		24.95
micro-epsilon laser sensors	0.1mm	2-750mm	RS422, ethernet		>500
Q4X Versatile, Rugged Laser Distance Sensor	0.27 mm/LSB	25-300mm	Analog		449
Ultrasonic sensor UM 12	0.22 mm/LSB	20-250 mm	analog		298
MB1003 HRLV-MaxSonar EZ0	4.9mm/LSB	30-5000mm	analog		37.95
UK6 Series Ultrasonic Sensors	2.5mm/LSB	40-300 mm	analog		99

TABLE A2. Trade study of position sensors, with selected sensor highlighted.