UNIVERSITY OF CALIFORNIA,
IRVINE

Position and Orientation Detection of Magnetic Objects Using Optimization Techniques

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Mechanical Engineering

by

Alan Lawrence Mitchell

Thesis Committee:
Assistant Professor Camilo Velez Cuervo, Chair
Distinguished Professor J. Michael McCarthy
Assistant Professor Alexandra Voloshina

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my Academic Advisor, Professor Camilo Velez Cuervo, for introducing me to the world of Magnetics and always providing guidance to help me make the most of my interests and efforts

I would like to thank my committee members, Thesis Committee: Professor Camilo Velez Cuervo, Professor J. Michael McCarthy, and Professor Alexandra Voloshina for setting up and participating in the Mechanical Engineering Systems and Design seminal series, at which I presented my work and received valuable feedback.

I also would like to thank my colleague, Ryan Apalonio, for help creating 3D printed parts and computation of the magnetic moment of the 2mm NdFeB cube magnets I used for my work.

ABSTRACT OF THE THESIS

Position and Orientation Detection of Magnetic Objects and 3D Field Measurement of Magnetic Devices

by

Alan Lawrence Mitchell

Master of Science in Mechanical Engineering

University of California, Irvine, 2023

Assistant Professor Camilo Velez Cuervo, Chair

A system was constructed for measuring magnetic field in 3D space and determining the position and orientation of magnets. Measurements were performed with a Hall-effect Gaussmeter attached to a 3-axis cartesian robot, capable of measuring magnetic fields of ±2 Tesla at spatial increments of as little as 8.75 μm within a working volume of 325mm by 290mm by 100mm. An algorithm was developed for the localization of magnets with known properties based on nonlinear optimization techniques. The Nelder-Mead simplex algorithm and Levenberg-Marquardt nonlinear least-squares algorithm were tested for detection of Neodymium-Iron-Boron magnets. Nelder-Mead performed the best, with a positional error of 2.21mm and orientation error of up to 22.4 degrees.

Introduction

The field of robot guidance has seen great advancement thanks to advances in Machine Vision and Machine Learning. Palletizer systems, comprised of vision guided pick-and-place robots along a conveyor have become commonplace in manufacturing and logistics, reducing labor costs and handling heavier loads than humans are capable of handling [1]. In the field of robotic surgery, neural networks have been developed to automate repetitive tasks based on input from cameras, reducing surgeon fatigue during long procedures [2].

Permeant magnets could be a useful positioning aid in cases where clear line of sight is not available. For example, surgical robots have been incorporated in the insertion of pedicle screws during spinal fusion surgery [3], but only as far as aligning a surgical tool to the spine. The actual insertion of screws is highly dependent on the feel and experience of the surgeon. If some part of the screw could be magnetized, magnetometers could provide useful information about its position in the body.

There has been some work on magnetic object tracking. Wahlstrom [4] used an array of 4 magnetometers to track magnets from the opposite side of a piece of plywood using an Extended Kalman Filter, with RMS position error of 4.95mm and orientation error of 1.85 degrees. This work will attempt to calculate magnet positions with greater accuracy using a larger array of magnetic field readings.

To avoid the increased cost of using a large number of magnetometers simultaneously, one magnetometer is positioned at different locations in 3D space. With readings taken from a large grid of points, existing nonlinear optimization algorithms can be used to compute the position and orientation of the magnets. In order to carry out this task, a system had to be designed and built to position a magnetometer in 3 dimensions.

An alternate use that this system was created for was the characterization of magnetic devices fabricated by other members of the Magnetic Microsystems and Microrobotics lab. Measuring fields surrounding MMM lab devices will help in calculating magnetic forces and experimentally validating simulations.

Requirements

System requirements are driven by both of the project goals:

1. Produce a system capable of measuring magnetic field at desired positions in 3D space with resolution of 0.1mm
2. Use that system to test an algorithm for determining a magnet's position and orientation. Given the existence of systems that track objects in real time that have accuracy of 5-10mm [4][9] accuracy should be better than 5mm.

Requirements for a system capable of meeting these goals are separated into three categories:

- Magnetometer
- Motion System
- Controller and Software

Magnetometer

The driving requirement for the magnetometer was the measurement range. Devices to be characterized under goal #1 will include permeant magnets, so a minimum measurement range was taken to be ±400 mT in X, Y, and Z.

Because of the automated nature of the system, the magnetometer also had to have a way to read and store data programmatically over an interface such as USB (preferred), SPI, I2C, etc.

While a minimum specification was not set, as high a sample rate as possible was desired, as sample rate drives the total scan time.

Motion System

The principal requirement for the motion system was to position the magnetometer in 3 dimensions with resolution better than 0.1mm. To speed up the design and build process, it was decided to pick a DIY kit for 3D printers, CNC machines, laser engravers or similar and add the magnetometer on. For cost reasons it was highly desired just to obtain the X/Y/Z mechanism without any unneeded tooling.

The motion system needed home sensors for automatic referencing of the moving axes. Referencing to a home sensor allows the system to move to repeatable absolute coordinates and allows safeguards against overtravel to be implemented.

The motion system should have stepper motors or NEMA mounting hole patterns to install stepper motors. Steppers are the most cost-effective way to achieve the required motion control.

Controller and Software

The control system must operate the stepper motors and perform data acquisition from the sensor. Furthermore, the magnetometer sampling rate is expected to be the bottleneck for scan time so the control of the motors should not block the magnetometer data acquisition. This means motor control and data acquisition must be done asynchronously on different processors.

Software must be created to orchestrate motion control and data acquisition with relative ease to the end user.

Design and Methodology

Magnetometer

Because other research uses of the system require measurement of strong fields near hard magnets, magnetometer options were limited. A USB magnetometer with ±2 Tesla operating range, USB25103[6] from Multi Dimension Technology, was selected. The magnetometer has a USB command interface for performing measurements and changing settings. The magnetometer can take "one shot" reads on command or continuous reads. The continuous read function had a stated sample rate of 200-250 Hz. In practice, using the magnetometer in "one shot" mode (taking readings only once the magnetometer is in the desired position), the sample rate reached only 8 Hz.

Motion System

OpenBuilds ACRO, a 2-axis (X/Y) laser engraver kit[5] was purchased, and a linear actuator was installed in place of the laser to meet the 3-axis requirement. The kit included two home sensors and all necessary cables. The kit did not include motors, so three NEMA 17 1.8 degree stepper motors were purchased separately. The system is pictured in Figure 1. The engraver kit has a linear travel per motor turn of 28mm, which combined with 200 step per revolution stepper motors and stepper drivers using 16 microsteps per full step gives a resolution of 8.75 μm.

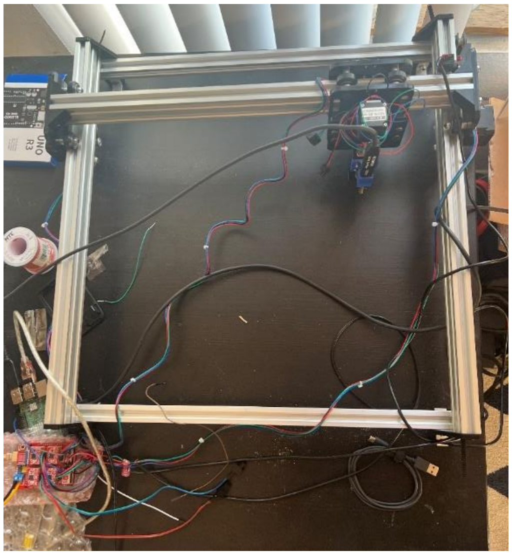

Figure 1: Cartesian 3-axis system with magnetometer probe attached.

Controller and Software

To accomplish the required asynchronous motion control and data acquisition, two popular microcontroller boards were selected to perform each function. Figure 2 shows the system block diagram. An Arduino UNO with Allegro A4988 stepper motor drivers was used as the motor controller. When the magnetometer is in position, the Arduino sends a signal to a Raspberry Pi via the two boards' GPIO pins to initiate the magnetometer read. The Arduino has X and Y home sensors attached, and homing of the Z axis is performed by simply positioning the magnetometer probe touching the table, so that the z coordinate is always the height of the probe above the table, even when the probe is removed and reinstalled, which is expected to happen often.

The Raspberry Pi runs a Linux OS with a desktop, so it is ideal as a user interface. At this stage, the system is operated by running Python scripts on the Raspberry Pi that perform the most common tasks. These scripts utilize a Python class that handles all of the communication and allows for easy development of new scripts to perform tasks not covered by the provided scripts.

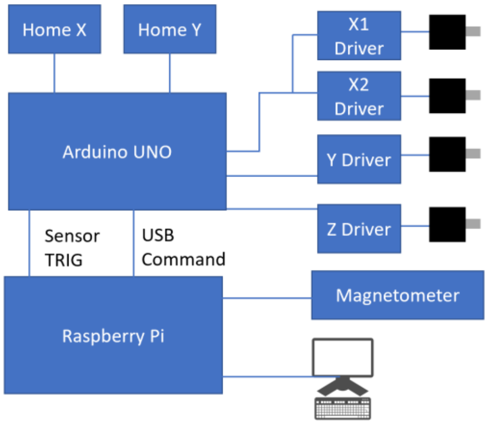

Figure 2: System Block Diagram.

Name	Value	Unit
Linear Travel Per Microstep (X/Y)	8.75	µm
Linear Travel Per Microstep (Z)	5	µm
Max Velocity (Standard)	450	mm/s
Acceleration (Standard)	100	mm/s^2
Travel (X)	290	mm
Travel (Y)	325	mm
Travel (Z)	100	mm
Supply Voltage	24	V
Max Power	30	W
Max Magnetic Field	2	T
Noise level	0.75	G
Sample Rate	3-8	Hz

Table 1:Overall System Design Specifications.

Optimization problem formulation

Before building the system, the approach to position and orientation detection was formulated.

As a point approximation to a real magnet, the dipole equation can be used to model the magnetic field at a position **r** relative to the magnet centroid:

$$\mathbf{B}(\mathbf{r},\mathbf{m}) = \frac{\mu_0}{4\pi}\left[\frac{3\mathbf{r}(\mathbf{m}\cdot\mathbf{r})}{\|\mathbf{r}\|^5} - \frac{\mathbf{m}}{\|\mathbf{r}\|^3}\right] \tag{1}$$

The magnetic moment **m** is a vector along the magnetization direction of the magnet and has a known magnitude which is a property of the magnet. The magnetic moment **m** can be written as:

$$\mathbf{m} = |\mathbf{m}|\begin{bmatrix}\cos\psi\sin\varphi \\ \sin\psi\sin\varphi \\ \cos\varphi\end{bmatrix} \tag{2}$$

φ and ψ describe the orientation of the magnet, according to Figure 3

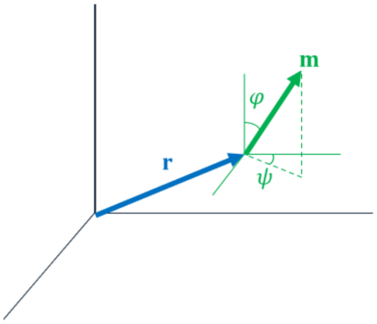

Figure 3: Magnetic moment vector orientation described by φ and ψ.

Thus, the magnetic field measured at any sensor position depends on the sensor's position relative to the magnet, $\mathbf{r} = [x, y, z]^T$, and the magnet's orientation, described by φ and ψ. The decision variable $\mathbf{x}$ is defined as:

$$\mathbf{x} = [x, y, z, \varphi, \psi]^T$$

(3)

A grid of m by n magnetometer readings, $\mathbf{B}_{actual\ at\ position\ i}$, is taken at positions $\mathbf{r}_{sensor,i}$, and the error for a guessed magnet position $\mathbf{x}$ can be quantified as simply the differences between the Magnetic field at $\mathbf{r}_{sensor,i}$ predicted by (1) and the measured magnetic field, added up across all $m \times n$ measurements. The unconstrained optimization problem can thus be stated as:

$$\min_{\mathbf{x} \in \mathbb{R}^5} \sum_{i=1}^{mn} \|\boldsymbol{B}_{model\ at\ sensor\ i}(\mathbf{x}) - \boldsymbol{B}_{actual\ at\ sensor\ i}\|$$

(4)

Nonlinear Least-Squares Problem Formulation

The problem can be posed slightly differently to use nonlinear least squares algorithms. Rather than adding all error terms, these algorithms use a vector of residuals:

$$\mathbf{R}(\mathbf{x}) = \begin{bmatrix} \|\boldsymbol{B}_{model\ at\ sensor\ 1}(\mathbf{x}) - \boldsymbol{B}_{actual\ at\ sensor\ 1}\| \\ \vdots \\ \|\boldsymbol{B}_{model\ at\ sensor\ mn}(\mathbf{x}) - \boldsymbol{B}_{actual\ at\ sensor\ mn}\| \end{bmatrix}$$

(5)

Prototype

System Hardware

The OpenBuilds ACRO system proved to be a good foundation for the system. Under control of the Arduino motor controller, it operated smoothly and precisely. Motor controller software was developed first. It accepts text commands to perform any motion task, including homing, moving, setting velocity and acceleration, and performing scans. This command interface was tested by sending commands using a terminal emulator (PuTTY on Windows or screen on Linux) and verifying correct execution of those commands. With the motor control software validated, the higher level Python class to run on the Raspberry Pi was written and debugged. The major issue encountered in the higher level software was that the signal from the Arduino that triggers the Raspberry Pi to read the magnetometer was picking up noise and the Raspberry Pi was detecting rising edges when it was not the time to read the magnetometer. These spurious triggers were eliminated using the following 3 actions:

1. Re-routing the signal cable away from the motor power cables.
2. Keep the Arduino GPIO pin HIGH normally, and pulse it LOW when the magnetometer is in position for a read.
3. Raspberry Pi data acquisition software was modified to reject detected edges if they were not detected within a time window of when the edge was expected based on motor speed and acceleration.

Magnet Position and Orientation Detection

A 2mm cube Neodymium-Iron-Boron magnet was used in all position/orientation detection tests. In order to use the techniques described in Design and Methodology, the magnetic dipole moment $|\mathbf{m}|$ must be known. A value of 0.009072 A m^2 was determined from the manufacturer's website [6]. To validate this value, a simple comparison between the dipole equation and actual measurements at various horizontal positions at heights of 5.75 and 8.63mm above a magnet with its north pole facing up was performed. Figure 4 shows the results and a schematic of the measurement positions.

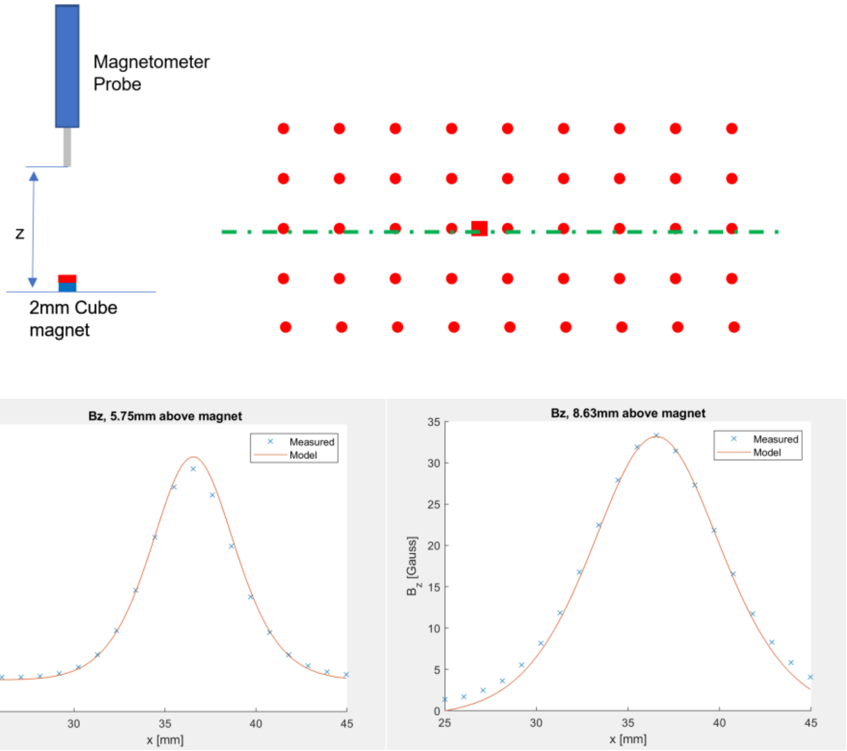

Figure 4: Top – Schematic of measurement setup – measurement points (red dots) selected to be plotted were the measurements passing directly above the magnet. Middle – Comparison of values from equation 1 to measured values at 5.75mm. Bottom – Comparison of values from equation 1 to measured values at 8.63mm.

For the full test of position and orientation detection, two magnets were positioned 50mm apart. One had the north pole facing down ($\varphi = \pi$) and the other facing perpendicular to the line between the two magnets ($\varphi = \frac{\pi}{2}, \psi = \frac{\pi}{2}$). The arrangement is shown in Figure 5. A 10 by 40 grid of measurements were taken at 10mm above the magnets. Table 2 shows the initial guesses used.

Magnet	$\mathbf{x}_0$
Bottom, North down	x=15, y=15, z=10, φ=0, ψ=0
Top, North to the side	x=15, y=90, z=10, φ=0, ψ=0

*Units of **x** are (x[mm], y[mm], z[mm], φ [radians], ψ [radians])

Table 2: Initial guesses used for optimization algorithms.

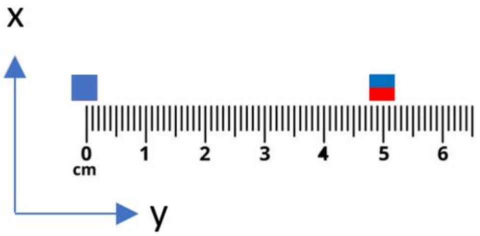

Figure 5: Magnet configuration for position/orientation detection test.

These sensor values were then used with MATLAB's implementation of the Nelder-Mead simplex algorithm, fminsearch. Table 3 shows the results for fminsearch.

The same data was used with MATLAB's Levenberg-Marquardt nonlinear least squares solver, lsqnonlin. The results are in Table 4.

For both of the above cases, the stopping conditions were relative change in the objective function less than 10^{-8} and relative change in the decision variable less than 10^{-6}

Magnet	$\mathbf{x}_{converged}$
Bottom, North down	x=22.28525, y=54.93237, z=11.18055, φ=0.8563 π, ψ= -0.42814 π
Top, North to the side	x=23.73272, y=102.73465, z=10.56913, φ =0.48660 π, ψ =-0.51532 π

* Units of **x** are (x[mm], y[mm], z[mm], φ [radians], ψ [radians])

Table 3: Converged values for magnet configuration shown in Figure 5 using Nelder-Mead Simplex algorithm.

Magnet	$\mathbf{x}_{converged}$
Bottom, North down	x = 22.11911, y=55.71739, z=11.85632, φ=0.87543 π, ψ = -0.44973 π
Top, North to the side	x = 23.69636, y = 102.72475, z = 10.56853, φ = 0.48642π, ψ = 0.4853 π

* Units of **x** are (x[mm], y[mm], z[mm], φ [radians], ψ [radians])

Table 4: Converged values for magnet configuration shown in Figure 5 using Levenberg-Marquardt nonlinear least-squares algorithm.

The results for the Nelder-Mead simplex algorithm and the Levenberg-Marquardt algorithm least-squares algorithm both converged to very similar values. Nelder-Mead gave a magnet spacing of 47.82mm while Levenberg-Marquardt gave a spacing of 47.05mm. For Nelder-Mead, the biggest orientation error was 0.45 radians or 25.8 degrees. For Levenberg-Marquardt, the worst-case orientation error was 22.4 degrees. Figure 6 shows the positional results overlaid on the magnetic field measurements.

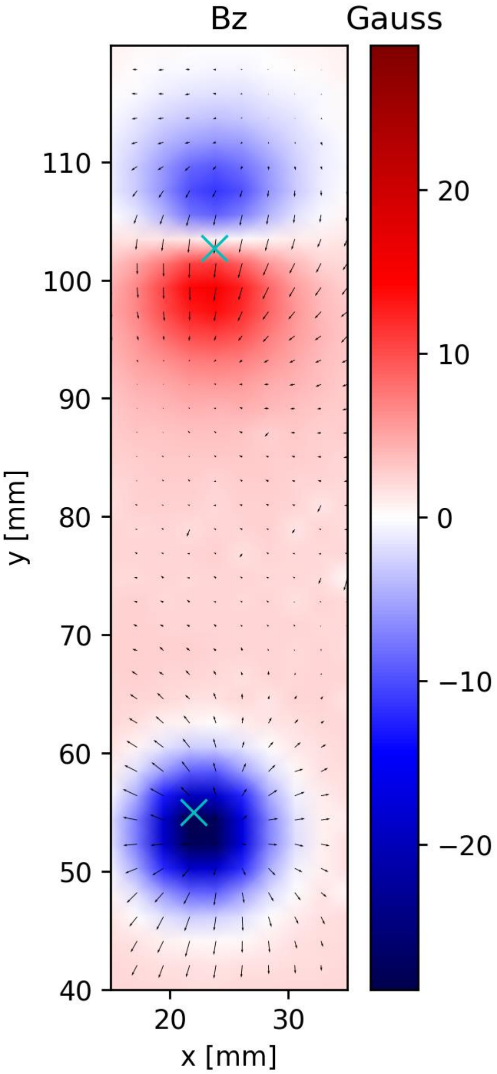

Figure 6: Solutions for magnet X/Y position (blue "X") overlaid on the magnetic field results. The Magnetic field perpendicular to the table, B_z, is represented by color while B_x and B_y are represented by the size and direction of the arrows

While the magnet positioning accuracy was strong, with error of 2.21mm for Nelder-Mead, there were significant errors in the orientation of the magnet oriented North-Pole down. This does not appear to be a case of the solver getting stuck in a local minimum before reaching the correct value of π radians, because giving when the solver the correct value in the initial guess, it still converges to the value of 0.8563 π radians. There are two likely causes for the error:

1. Measurement noise from the magnetometer. The standard deviation on reference readings with no magnet present was 0.375 Gauss, and deviations as high as 3.69 Gauss were observed. Given that the largest measured value, right above the north pole-down magnet, was -29 Gauss, this noise is significant. Noise may be reduced by changing ADC oversampling settings on the sensor. A higher sensitivity sensor, like a magnetoresistive sensor, with a smaller operating range would likely be better suited to this application
2. A simple point approximation for the magnetic field was used in the objective function. Accuracy of the model-estimated fields can be improved by moving to a model of a discrete-sized magnet such as the model derived by Derby [7]

The system was quite successful in characterizing the magnetic fields of one of the lab's devices. A small Magnetohydrodynamic (MHD) motor designed for microrobot propulsion in salt water was measured with positional resolution of 0.2mm. The Magnetic fields in X, Y, and Z are shown below in Figure 7.

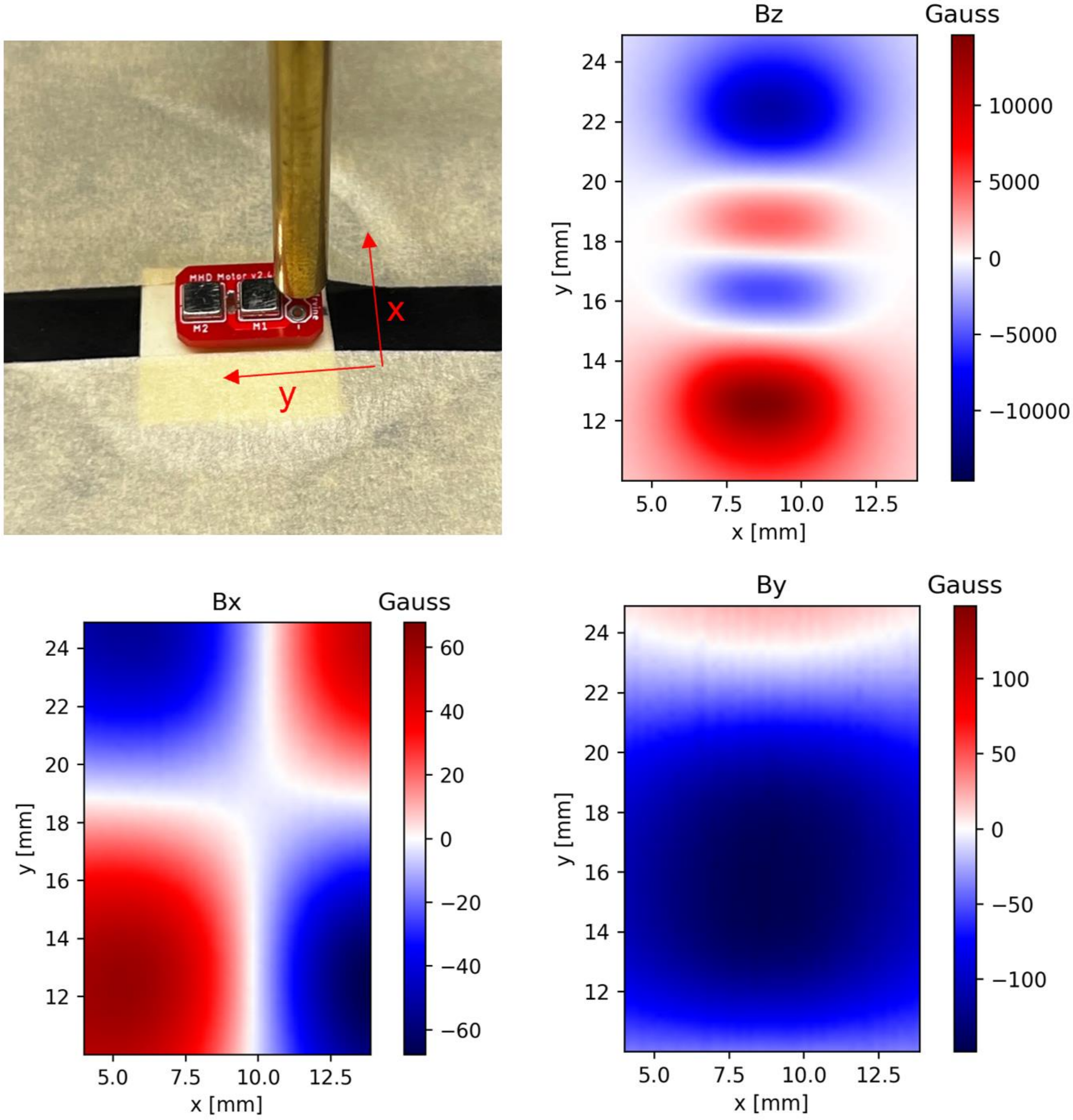

Figure 7: MHD motor fabricated by Magnetic Microsystems and Microrobotics Lab, with plots of Magnetic field in X, Y, and Z. Note that the color scales are different. The magnetic field in Z (perpendicular to the table) is dominant. This scan was taken 1.5mm from the top surface of the magnets.

Conclusion

Good estimates of magnet positions could be obtained from a distance of 10mm using a single magnetometer mounted on a 3-axis cartesian robot which takes large numbers of readings. Using the Nelder-Mead simplex algorithm, minimizing the objective function (4) gave position values with error of 2.21mm. Levenberg-Marquardt least squares yielded similar results when minimizing the residuals (5). The method had some difficulty determining orientation with a wors case orientation error of 22.4 degrees. To improve these results, a more accurate magnetic field model should be employed, and sensor noise needs to be reduced.

The system succeeded at characterization of magnetic devices. There are two aspects of the system that should be improved going forward:

1. Ease of use: the Python scripts created for this project are fairly simple, but to polish the system a Graphical User Interface should be created. The Python class handling all communication and data acquisition should make it easy for an experienced GUI programmer to create one.
2. Scan time: as the desired resolution gets finer, the scan time rapidly increases. Figure 7 took about 30 minutes to capture, for example. Scan time could be improved massively if the sample rate could be improved from the 8 Hz seen in the current implementation to something closer to the 200Hz that is obtained in continuous read mode.

References

[1] Thompson, Steve. “Benefits of Automated Palletizing Systems.” Benefits of Automated Palletizing Systems, 27 Aug. 2021, www.palletmarketinc.com/blog/benefits-of-automated-palletizing-systems.

[2] Saeidi, H. et al. “Autonomous Robotic Laparoscopic Surgery for Intestinal Anastomosis.” Science robotics 7.62 (2022): eabj2908–eabj2908. Web

[3] Lieberman IH, Kisinde S, Hesselbacher S. Robotic-Assisted Pedicle Screw Placement During Spine Surgery. JBJS Essent Surg Tech. 2020 May 21;10(2):e0020. doi: 10.2106/JBJS.ST.19.00020. PMID: 32944411; PMCID: PMC7478327.

[4] Wahlstrom Niklas, and Fredrik Gustafsson. “Tracking Position and Orientation of Magnetic Objects Using Magnetometer Networks.” Linköping University Electronic Press.

[5] Carew, Mark. “OpenBuilds ACRO System.” OpenBuilds, 22 July 2020, openbuilds.com/builds/openbuilds-acro-system.5416.

[6] “K&J Magnetics - Demagnetization Curves.” K&J Magnetics - Demagnetization Curves, www.kjmagnetics.com/bhcurves.asp. Accessed 18 Dec. 2022.

[7] “USB25103 - USB Magnetometer - Sensors - MultiDimension Technology | Magnetic SensorManufacturer(TMR·GMR·AMR) - MultiDimension Technology Co., Ltd.” USB25103 - USB Magnetometer - Sensors - MultiDimension Technology | Magnetic SensorManufacturer(TMR·GMR·AMR) - MultiDimension Technology Co., Ltd., www.dowaytech.com/en/2089.html. Accessed 18 Dec. 2022.

[8] Derby, Norman, and Stanislaw Olbert. “Cylindrical Magnets and Ideal Solenoids.” American Journal of Physics, vol. 78, no. 3, American Association of Physics Teachers (AAPT), Mar. 2010, pp. 229–35. Crossref, https://doi.org/10.1119/1.3256157.

[9] Antoš, Jaroslav, and Pavel Fišer. “Robotic Palletization with Camera Position Determination.” Mechanisms and Machine Science, vol. 88, pp. 468–476.