Magnetic Field Measurement

Larry Hartman

Augusta University
Department of Chemistry and Physics
1120 15th Street
Augusta, Georgia 30912

27 November 2017

In partial fulfillment of a senior thesis project under the supervision of Prof. Trinanjan Datta

Abstract

Viscosity of gelatinous substances are proposed to be calculated by measuring the distance a paramagnetic bead would travel inside the substance when subjected to an external magnetic field. A computational model was generated using the Finite Element Method Magnetics (FEMM) software to gain insights on design and construction of the required equipment for initial experiments involving one- and two-rod experimental setups. Once initial computations were completed construction of two required instruments commenced to build the experimental setups to measure magnetic fields. The first instrument constructed was a precision coil winder. It was designed to produce electromagnetic coils of a variety of specifications, such as wire gauge, number of turns, and physical size. The second was a magnetic field measurement apparatus. It was set on a three-dimensional linear actuator in order to measure the magnetic fields produced by sources with widely differing geometries. An initial experiment was conducted with one-rod setup where a single coil was wound and mounted on a steel rod. The generated field was measured and shown to qualitatively agree with the FEMM-produced model via similarity in shape. This agreement opens the door to conduct more extensive studies of magnetic fields, specifically detailed field mapping of setups involving one and two rods.

Introduction

In early Spring 2015 the Augusta University Chemistry and Physics Department was approached by the University Health Sciences Campus with a project proposal to determine the viscosity of fluid within embryonic cells. Viscosity is a measure of a fluid's internal friction that causes resistance to its flow. Possible pathways to measuring viscosity include optical tweezers and atomic force microscopy (AFM). Optical tweezers is an unlikely method due to impurities in the biological material. The efficiency of AFM is severly hampered by aqueous environments.

Existing literature suggested that viscosity could also be determined by measuring the distance a microscopic paramagnetic bead would travel inside the cell when subjected to a external magnetic field. This method is not hampered by the intrinsic challenges of optical tweezers and AFM. A magnetic field is created by electromagnetic coils arranged in a symmetric pattern, and is commonly referred to as "magnetic tweezers" (Figure 1). Designs for magnetic tweezers can use as few as two identical coils to as many as eight identical coils. The limitation of this method is that it requires video photography to indirectly measure the force (via movement) and engineering of feedback controls.[1]

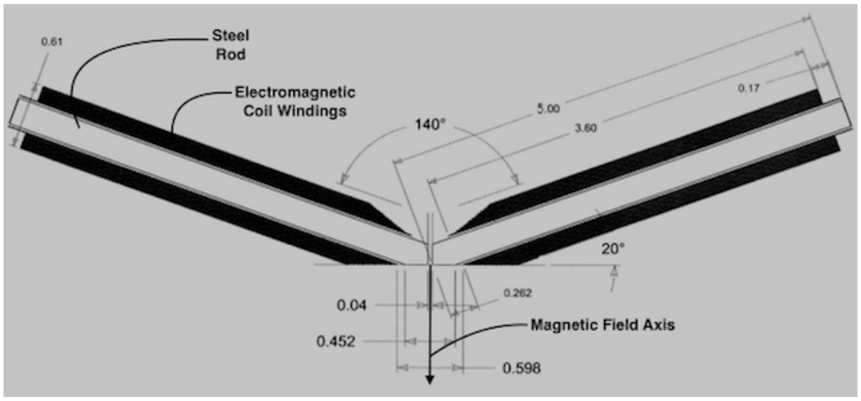

Figure 1: Magnetic Tweezer Schematic. A simple two coil magnetic tweezer setup.[2] Displayed units are millimeters.

Two primary challenges exist concerning the experimental setup pertaining to embryonic cells that could not be overcome via the limitations within the department: 1) physical size (nano- and pico-scale), and 2) the control and impact of temperature variations on the biological material. The proposal requested that the viscosity measurements take place within a cell frozen by liquid nitrogen (approx. 200K) in vitro. The size required that a pico-scale precision force to manipulate the bead without disturbing the cell.[3] Additionally the use of liquid nitrogen necessitated the containment of portions of the experimental setup in a super-cooled atmosphere (possibly microscope, camera, etc). As much as a 1K variance in temperature greatly impacts the physical properties of the biological material by expanding a single cell hundreds of nanometers, compounding the concerns already inherent to the small scale.[1]

Owing to the limitations posed by scale and temperature combined, the project to determine the viscosity of fluids within a cell was set aside. However, a macro-scale derivative of the project was taken up—the measurement of viscosity of a fluid at room temperature in a container with that could accommodate 20 to 30 mm of linear movement. Even though the scale was sufficiently large (paramagnetic beads with diameters from 1.0 to 6.0 mm, Appendix A), the project was still dominated by engineering challenges in building the experimental setup. The required custom-made precision electromagnetic coils cost hundreds of dollars each commercially. Trial and error was expected in determining the correct specifications of coils, potentially necessitating several purchases. An alternative path was undertaken to build a machine to produce the coils in house. A search was also conducted to find a commercial supplier for magnetic field measurement instruments that could fill the need to measure magnetic fields produced in experimental setups under examination. The instruments examined were designed for larger scale applications, were not easily adaptable to these setups, did not meet the needs of experimental precision, or were too costly (over $1000).

The theory involved in both experimental setups (viscosity in a cold cell and viscosity at room temperature) are motivated by the same underlying physical principles. A non-uniform external magnetic field exerts a force upon a para-magnetic bead in a viscous environment, causing the bead to move. The motion is counter-balanced by the viscosity of the fluid. The forces involved (even at smaller nano- and pico-scale setups) can be safely examined using Newtonian equations:

$$\vec{\mathbf{v}} = \frac{\vec{\mathbf{F}}_v}{\gamma} \tag{1}$$

$$\vec{\mathbf{F}}_b = \vec{\boldsymbol{\nabla}}(\vec{\mathbf{m}} \cdot \vec{\mathbf{B}}) \tag{2}$$

$$\gamma = 6\pi\eta r \qquad \text{[7]} \tag{3}$$

Equations 1 & 3 are motivated by Stoke's Law which relates drag force of spherical objects to fluid viscosity. After substitution of Equation (2) into (1) and some algebraic manipulation, the magnetic dipole moment is converted into volume magnetization. Volume magnetization is the amount of magnetic dipole moments per unit of volume:

$$\vec{\mathbf{v}} = \frac{\frac{4}{3}\pi r^3 M_{bead} \vec{\boldsymbol{\nabla}}|\vec{\mathbf{B}}|}{6\pi\eta r} \tag{4}$$

$$\vec{\mathbf{v}} = \frac{d^2 M_{bead} \vec{\boldsymbol{\nabla}}|\vec{\mathbf{B}}|}{18\eta} \qquad \text{[3]} \tag{5}$$

$\vec{\mathbf{v}} =$	velocity	[m/s]
$\vec{\mathbf{F}} =$	force	$[N]$
$\vec{\mathbf{m}} =$	magnetic dipole moment	[Kg m/s^2]
$\vec{\mathbf{B}} =$	magnetic field	$[T]$
$\gamma =$	viscous drag	[Kg/s]
$M =$	Volume Magnetization	[Kg/m^{2}s^2]
$\eta =$	viscosity	[Kg/m s]
$d =$	diameter	[m]

The direction of the force that will be exerted on the para-magnetic bead is typically in a straight line. The objective of the experimental setup is to exert the force along an easily identifiable axis. In this instance the above vector equation would be reduced to a single dimension. Re-arranging an expression for viscosity is obtained:

$$\eta = \frac{d^2 M_{bead}}{18 v_{bead}} \frac{\partial B_z}{\partial z} \tag{6}$$

It is noted that the velocity of the bead in a viscous fluid is directly tied to the slope of the magnetic field produced by the coils. However, the field gradient is non-linear, indicating that the bead is experiencing acceleration along its path. To overcome this challenge, it is necessary to locate the best range along the line of movement where the field gradient is approximately linear–from which to obtain measurements of the bead's average velocity with minimal error.

$$\eta = \frac{d^2 M_{bead}}{18 v_{bead}} k_{B_z} \tag{7}$$

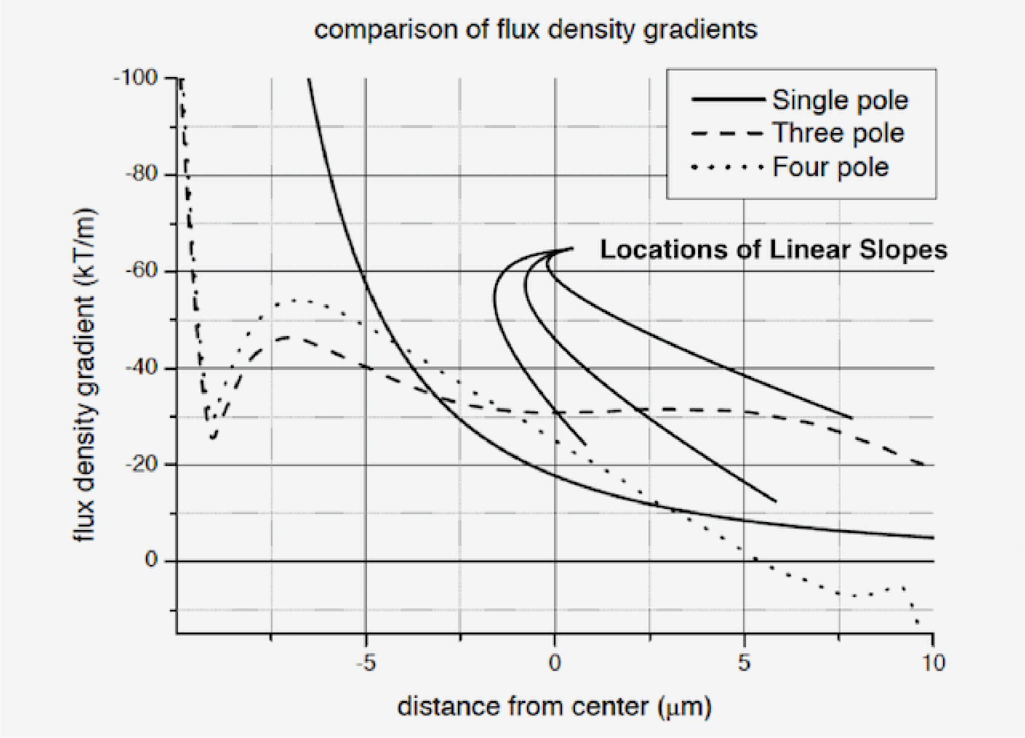

Figure 2: Field Gradients. Experimental data showing field gradients produced by differing numbers of electromagnetic coils.[3] The locations where slopes are approximately linear are in the positive region of the axis are annotated.

The following three figures depict possible arrangements of single, three, and four pole experimental setups referenced in Figure 2. There are other possible arrangements for the three and four pole setups. The goal in any arrangement is to achieve symmetry on the central axis of the setup. The arrow in each figure indicates the axis of symmetry for each setup and the path of the bead.

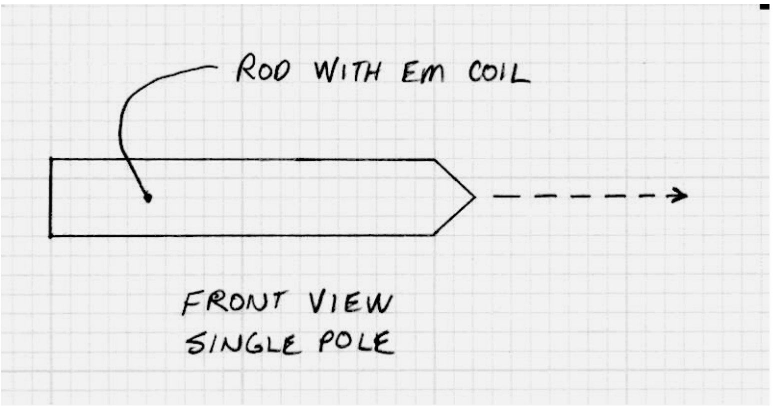

Figure 3: Single Pole Setup.

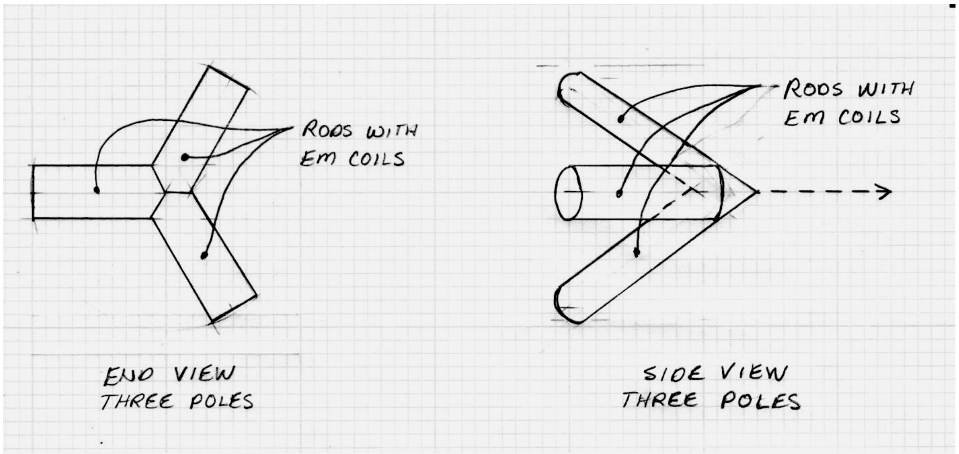

Figure 4: Three Pole Setup.

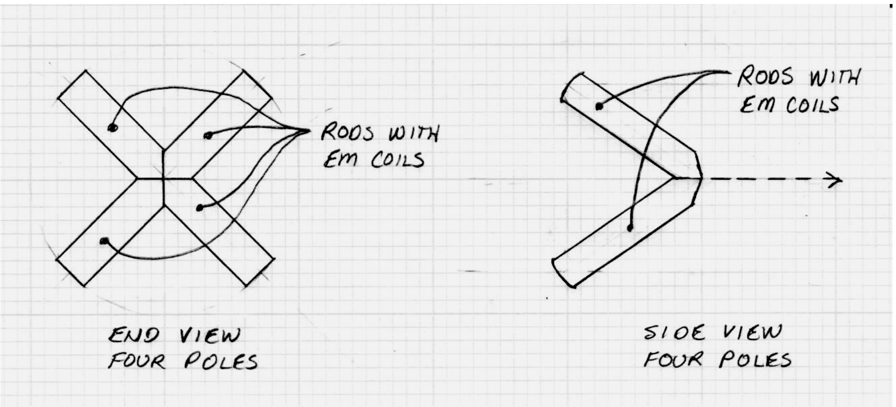

Figure 5: Four Pole Setup.

The magnetic field slope k_{B_z} of Equation (6) would be determined by taking measurements over equidistant intervals using a calibrated Hall sensor. A Hall sensor produces and output voltage as a result of an applied magnetic field. The velocity of the bead v_{bead} can be obtained by inspecting a video of the bead traveling over a certain distance. The approximately linear part of the field gradient must be located. Measurements in this range will be preserved and analyzed, the others discarded. Volume magnetization M_{bead} in the paramagnetic bead is constant.

A computational model was applied to experimental setups of differing geometries (Figure 6) involving one or two steel rods using Finite Element Method Magnetics software.[4] The goal in inspecting these models is to determine the most suitable geometry for experimental investigation by determining which geometry has the largest range of approximately linear field gradient. Field gradient contours were created within the computational model (Figure 7). The linear portion of the single magnet is nearly flat close to the tip, while the linear portions of the two opposing rod setups are fairly identical and more sloped farther away.

The FEMM model was created using magnetic material rather than steel rods with electromagnetic coils. Therefore the field values are different; however, the field vectors and gradients are the same shape. More analysis of experimental setups in the FEMM computational model should reveal a range of angles for two opposing rods that would provide a reasonably linear gradient. The FEMM software setup parameters are listed in Appendix B.

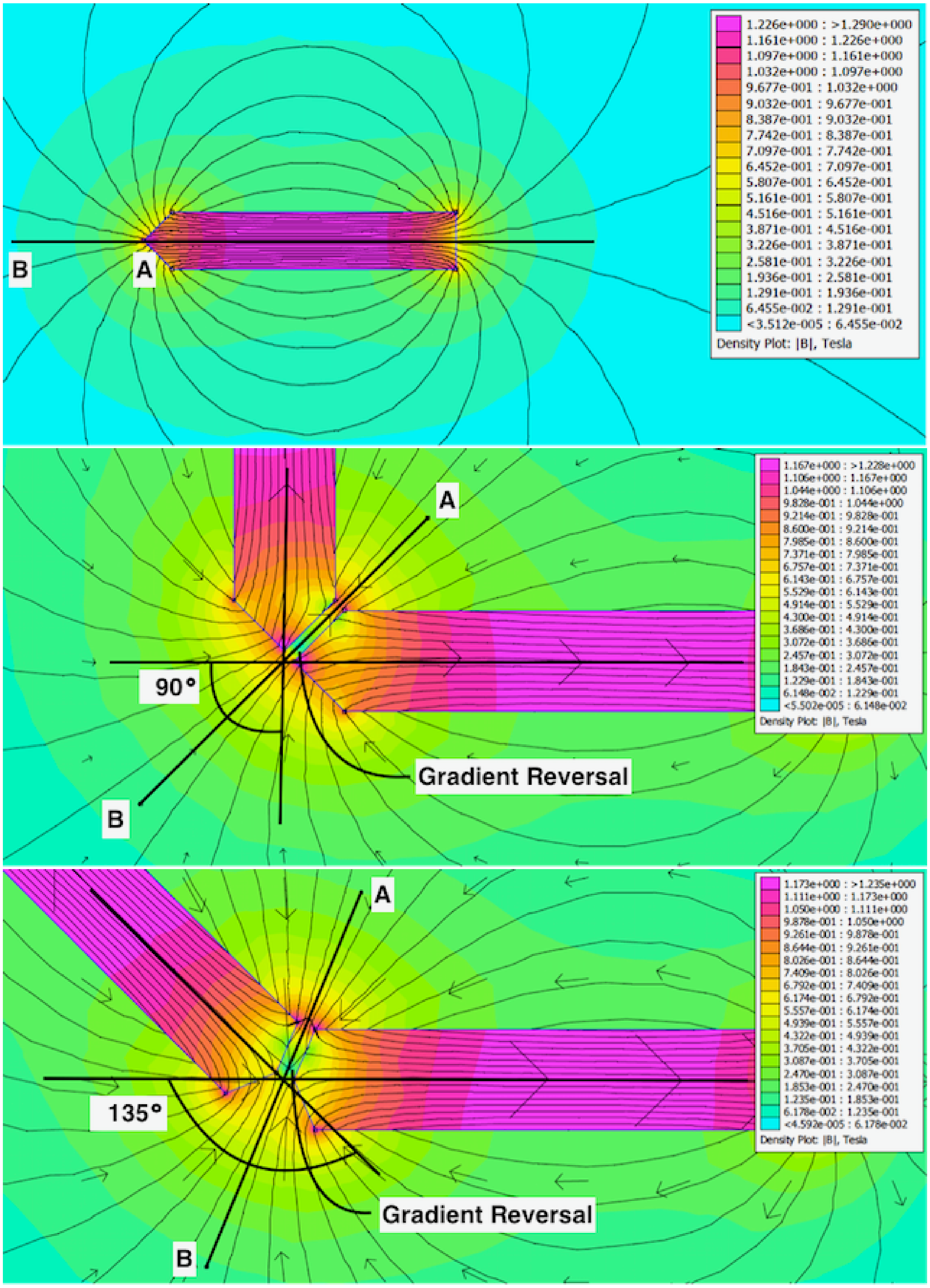

Figure 6: FEMM Model of Magnetic Field Lines. Simulation for different pole geometries. The line labeled "A—B" is the line along which the magnetic field gradient is calculated. *Top:* Single rod with tip, *Middle:* Two rods opposing at 90°, *Bottom:* Two rods opposing at 135°.

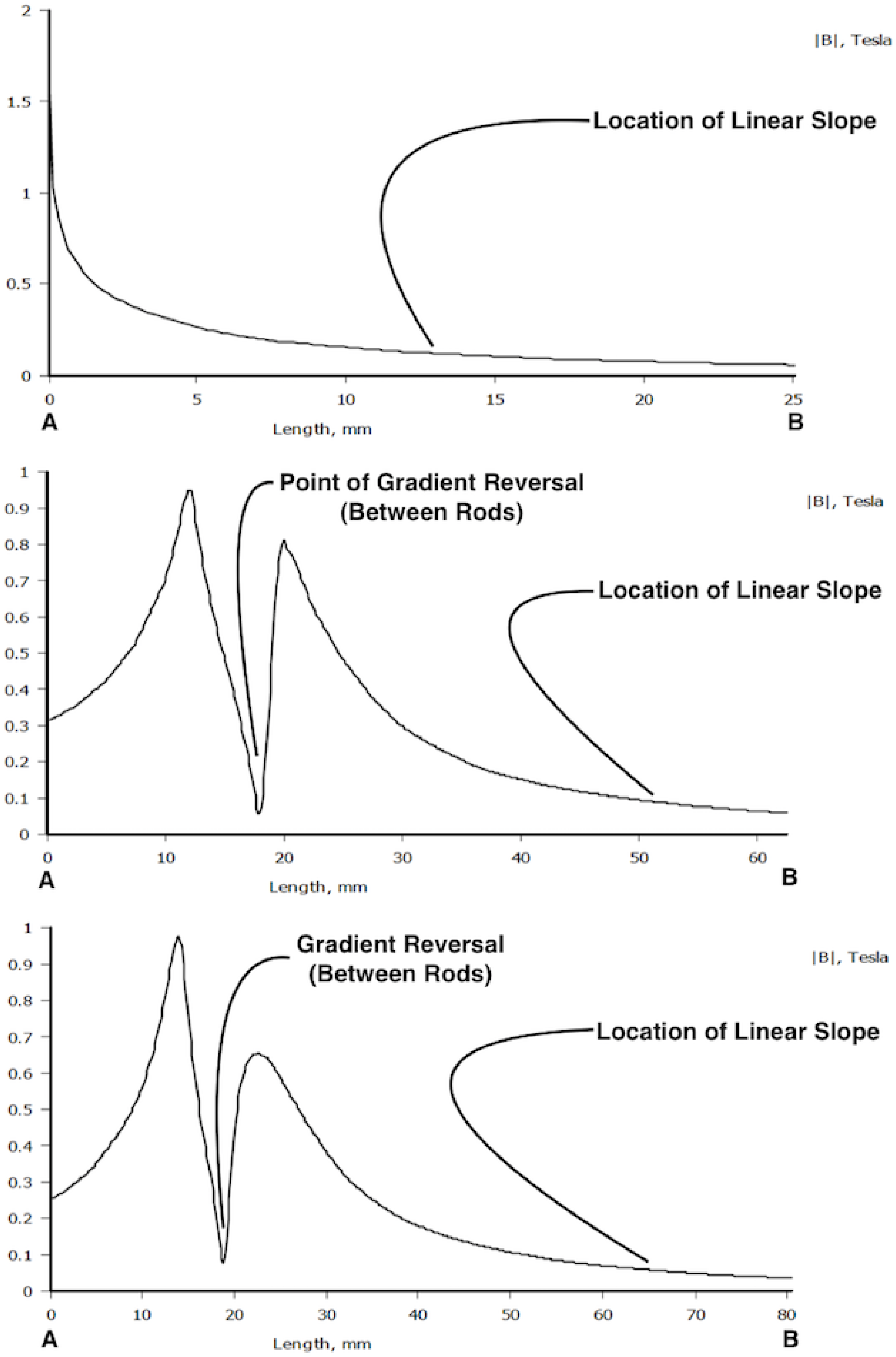

Figure 7: FEMM Model of Magnetic Field Gradient. The magnetic field gradient plot along line labeled "A—B" in Figure 6. *Top:* Single rod with tip, *Middle:* Two rods opposing at $90°$, *Bottom:* Two rods opposing at $135°$.

Engineering and Design

Precision Coil Winder

For an opposing rod setup consisting of two coils (Figure 1, as an example), the key design consideration is that each of the coils used must be identical. Mismatched coils will result in a skewed magnetic field axis, causing the bead to move in a direction not parallel to any measurement reference. Coils are mismatched when either the number of turns differ, or the turns are not evenly wound. This necessitated a machine capable of winding coils that have more than a couple dozen turns. The size of the coil winder was specified by taking into account winding replacement coils for some of the larger coils presently in use within the department. The frame was based on modular components designed to fit and connect together in certain arrangements. The electronics circuitry and programming was kept as simple as feasible.

The frame was built with OpenBuilds aluminum extrusion rails with prefabricated plates and joints (Figure 8).[5] The rails were cut with the compound miter saw, drilled with a hand drill and drill press, then tapped for screw threads. Plates, joints, and t-nuts were used to assemble the parts. The project included a couple of gentry plates which were also purchased through OpenBuilds. During this time, the OpenBuilds company became an important supplier for other physics projects.

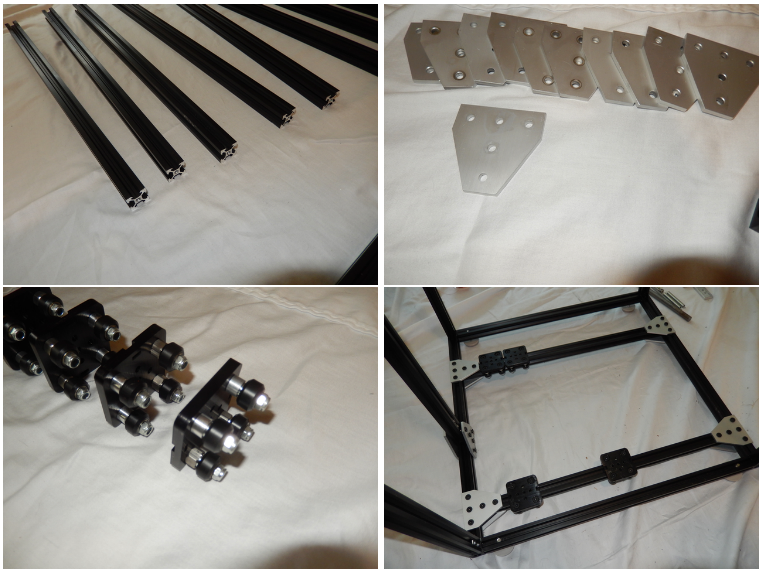

Figure 8: OpenBuilds Components. *Upper Left:* black anodized aluminum extrusion rails cut to length, *Upper Right:* plates for making T-joints, *Lower Left:* gentries, *Lower Right:* Partial assembly of coil winder.

Two aluminum blocks and four bearings were purchased to make two axle holders. The blocks were taken to Augusta University partner, the Textron Easy-Go Test Department, [6] for water jet cutting, milling, and boring. The bearings were inserted with a press (Figure 9). Later it was discovered that bearing units could have been purchased (without the labor and expense) to mount directly to the aluminum rails; however, these would not have held the axles as straight as the ones custom built with the help of Textron.

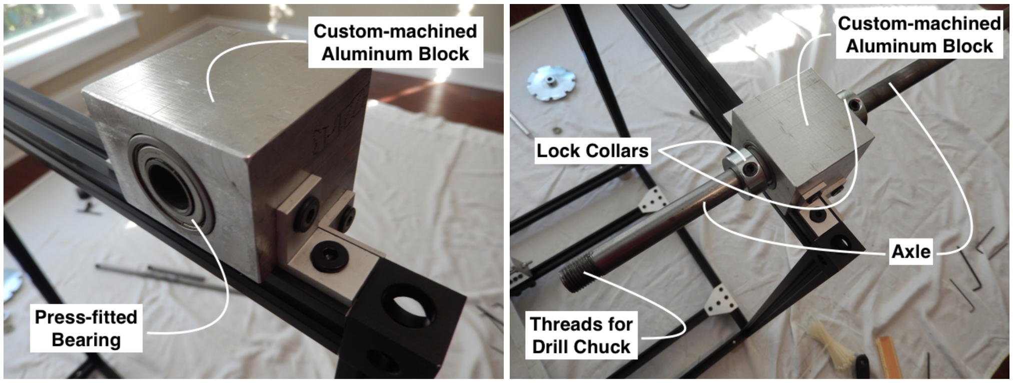
Custom-machined Aluminum Block
Press-fitted Bearing
Custom-machined Aluminum Block
Lock Collars
Axle
Threads for Drill Chuck

Figure 9: Axle Holder. *Left:* Axle holder without axle inserted, *Right:* Axle holder with axle inserted.

The drive train consisted of a 24 V DC high torque gear motor attached to the drive axle via a timing belt and toothed pulleys. Initially a sewing machine motor was chosen. It was discarded soon after assembly when it demonstrated poor control at very low speeds. After the DC gear motor was mounted the drive pulley broke. It was replaced with a new drive pulley. A longer belt with an idler pulley was added to control belt tension in order to prevent future damage to pulleys (Figure 10). The motor was mounted to a piece of acrylic which was laser cut by the Research and Development Department of Janus Research Group.[7] A circular metal plate with equally spaced notches was cut with a water jet and mounted to the drive axle for a rotary counter that could count to $\frac{1}{8}$ rotation. Drill chucks were used for mounting electromagnetic cores and bobbins.

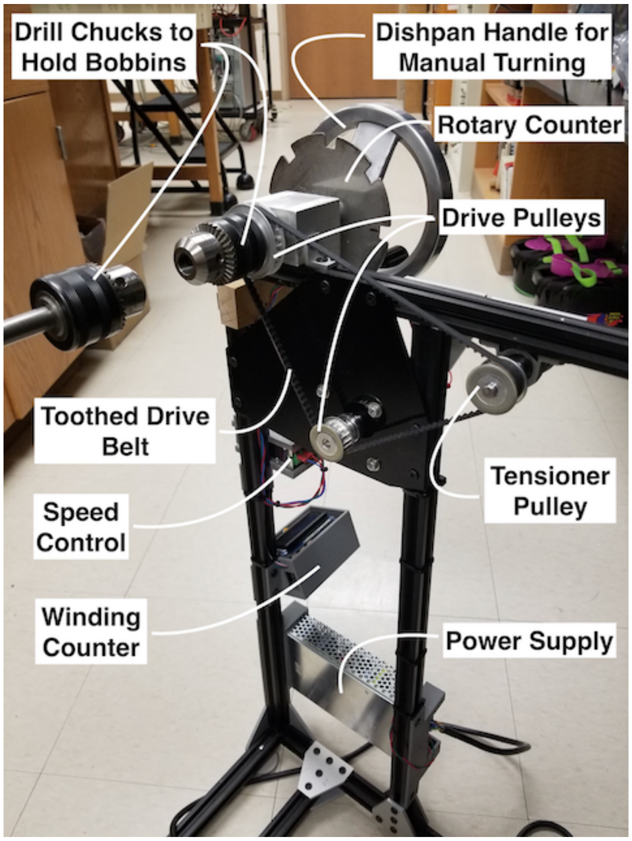

Figure 10: Drive Train. Full drive train depicted with major electronic components installed.

The electronics circuitry is composed of a Raspberry Pi development board with a 2 line RGB LCD display (see Figure 10). It is connected to a photogate sensor on the rotary counter and a DC adjustable-speed PWM motor controller. Maximum speed of the motor is about 15 rpms. The Raspberry PI is powered via a wall wort USB cable. A separate 24 V DC power supply provides power to the motor. Mounting brackets to mount the components to the frame were 3D printed with PLA plastic.

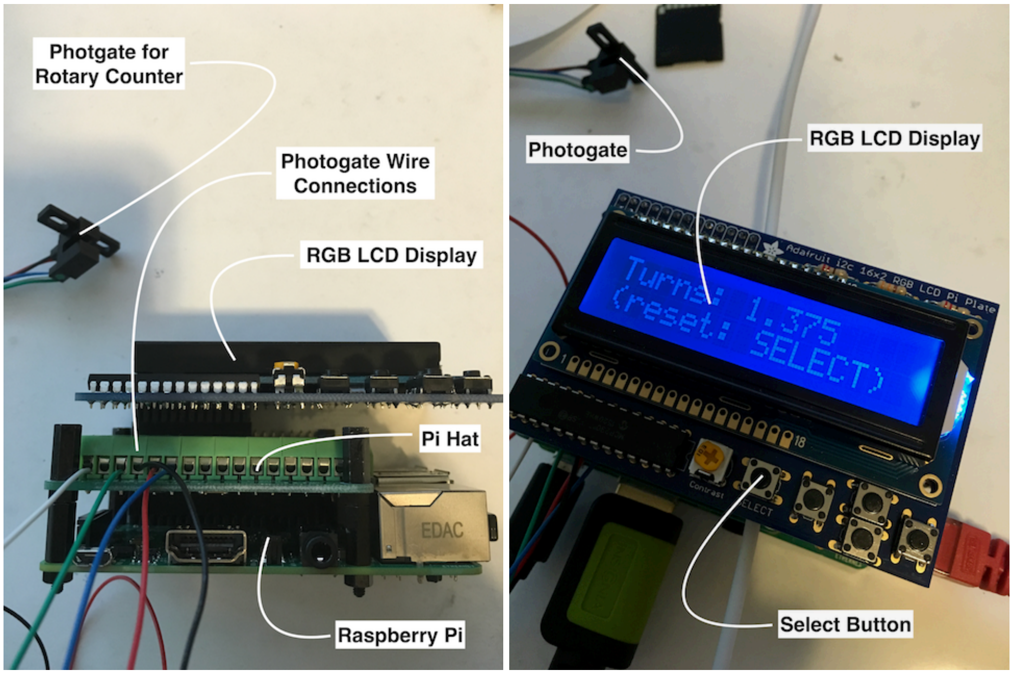

Figure 11: Raspberry Pi. *Left:* RPi shown with photogate sensor connections, *Right:* screen shown with number of turns.

Once construction of the coil winder was complete, two approximately identical 500-turn coils were wound onto 3D printed PLA plastic bobbins. One of the bobbins was then fitted to a 0.5-inch steel rod for initial field measurements (Figure 13).

Electromagnetic Coils, Rods, and Rod Mounts

A smaller engineering project commenced to build mount steel rods and circuitry to control the electromagnetic coils. Two frames (Openbuilds extrusion rails) were build to mount the rods: one where two coils are at an angle of $60°$ to each other, and the second where the coils are at an angle of $90°$ to each other. Both frames can also be used to mount a single coil. The rods were constructed of two steel alloys with high magnetic susceptibility. Several diameters and tips of differing angles were machined at the Textron Easy-Go Test Department. Rod specifications are listed in Appendix C.

3D-printed parts were used to achieve the angles for rod placement. The estimated precision is $\pm 1°$. Each rod is held by two holders. Both holders required identical vertical heights in order to hold the steel rod parallel to the extrusion rail. Additionally, in a two steel rod setup, all four holders required identical heights not only for parallelism, but in order to prevent offsets in rod height from interfering with the symmetry of the electromagnetic field. 3D-printed parts were also used to construct the steel rod holders. Precision was difficult to achieve due to the more complex geometry of the part. The estimated precision in parallelism is $\pm 2°$.

The precision estimates are based upon the calibration procedure used to setup the 3D printer. A large carpenter's square was used to square each of the printer axes to each other with the printer bed as the primary datum. Additionally components were printed with interior and exterior surfaces that were measured with a precision vernier caliper. Using this data, adjustments were made to the stepper motor constants to bring the sizes to the correct measurements within $\pm 0.25\,\mathrm{mm}$ over a distance of $100\,\mathrm{mm}$. The precision estimates given above are somewhat qualitatively derived, but likely higher than actual values.

The coils are powered by a $24\,\mathrm{V}$ automotive battery in order to ensure the cleanest possible DC current source. Current limiting (to match wire gauge) was achieved with resistors.[9]

Measurement Apparatus

The measurement apparatus was also constructed with Openbuilds parts coupled with a power supply, Raspberry Pi and assorted electronic circuitry and sensors. Three 500 mm lead screw linear actuators were connected to form movements on the x-,y-, and z-axes. Motor controllers and drivers connected between the DC stepper motors and the Raspberry Pi formed the basic circuitry. A program was developed in Python to give the user control of movements on each axis, and to execute automated three-dimensional movements (Appendix E). Initially a single Gauss meter sensor (professionally calibrated from a NIST standard) was used to manually collect data from a single 500-turn electromagnetic coil.

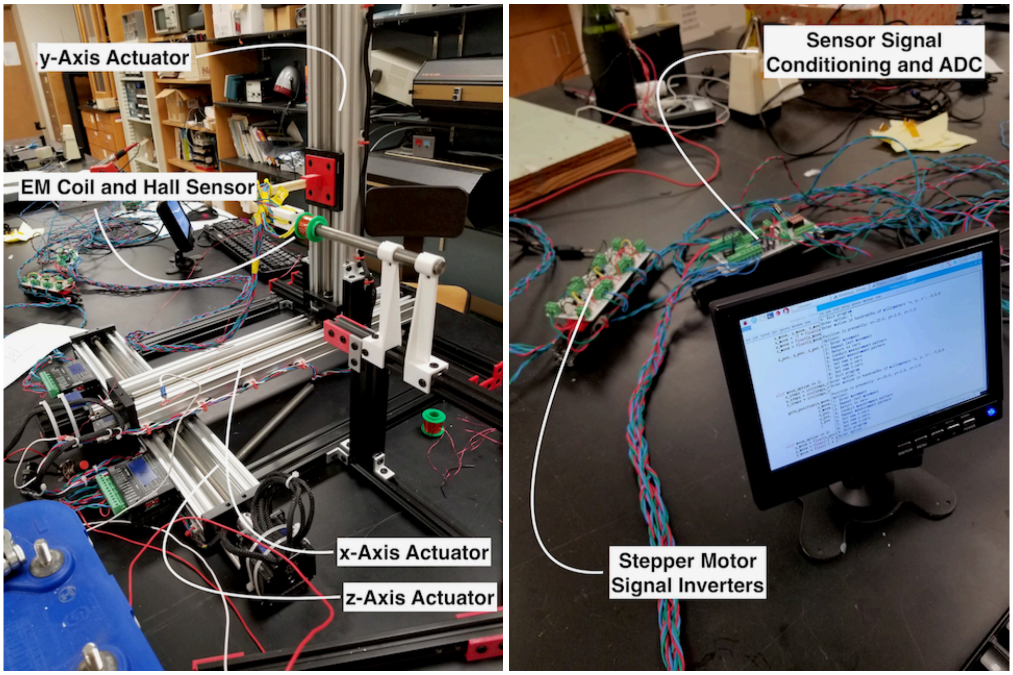

Figure 12: Measurement Apparatus. *Left:* Three-dimensional linear actuator, *Right:* User interface monitor.

Manual data collection proved to laborious, so a 5-sensor SS-49e Hall sensor array (Figure 13) was constructed to fully automate data collection on all three axes. The sensors were mounted to a 3D-printed PLA part glued to a wood square dowel rod. The estimated precision in sensor placement for each sensor is ±3° with respect to assigned axis and ± 0.5 mm to a measurement plane. The printed part requires redesign to fix the sensor wire protrusion that prevents full closure of the center sensor to the tip of the steel rod (Figure 13).

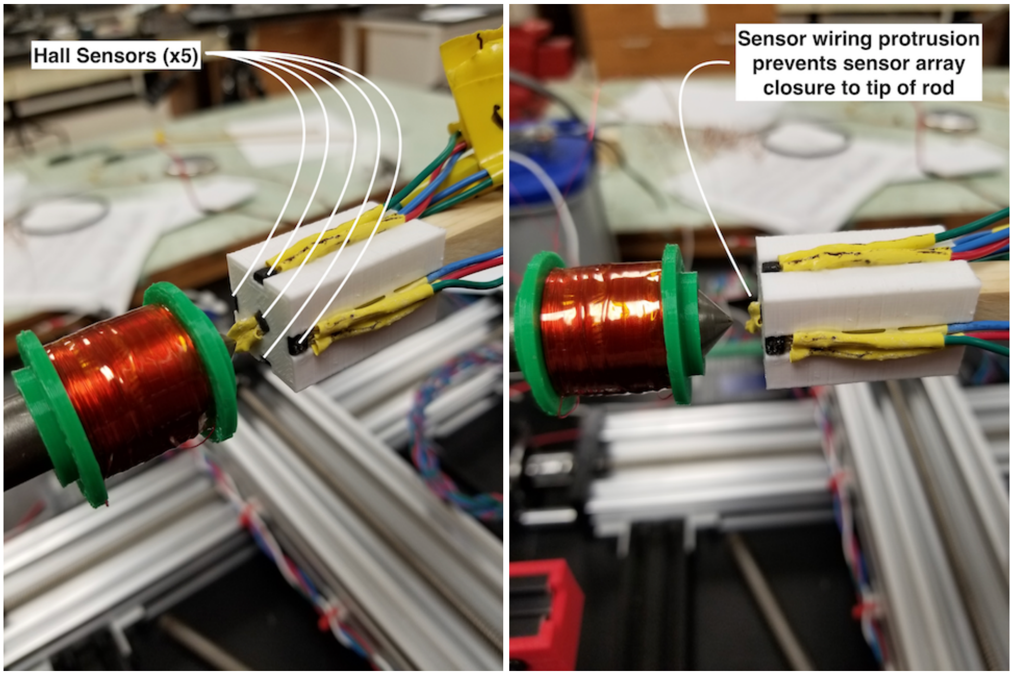
Hall Sensors (x5)
Sensor wiring protrusion prevents sensor array closure to tip of rod

Figure 13: Sensor Array. *Left:* View angle to show placement of five sensors, *Right:* View angle to show gap between array and metal rod.

The Hall sensors are connected to the Raspberry Pi via an Analog-Digital Converter (ADC) and a bidirectional level shifter. The output of the ADC is 5 V, while the input of the Raspberry Pi is 3.3V. Communications with the ADC takes place via Serial Peripheral Interface (SPI) protocol. A schematic of this circuit is provided in Appendix D. The below equations were used to determine the low-pass RC values for signal conditioning to filter out $60\,\mathrm{Hz}$ components that could exist from using an AC power source. The signal reactance and output voltages are:

$$R = \frac{1}{2\pi f C} \tag{8}$$

$$V_{out} = V_{in}\frac{X_C}{\sqrt{R^2 + X_C^2}} \tag{9}$$

$R =$	resistance	$[\Omega]$
$f =$	frequency	[Hz]
$X_C =$	capacitative reactance	$[\Omega]$
$V_{in} =$	input voltage	[V]
$V_{out} =$	output voltage	[V]

A cutoff frequency of $48\,\mathrm{Hz}$ and a standard capacitor size of $22\,\mu\mathrm{F}$ were chosen. The data rate for the sensor is set to 10 Hz in order to keep the data signal below the low-pass cutoff frequency.

$$R = \frac{1}{2\pi 48 * 0.000022} \tag{10}$$

$$R = 150\,\Omega \tag{11}$$

$$X_C = \frac{1}{2\pi 10 * 0.000022} \tag{12}$$

$$X_C = 723.4\,\Omega \tag{13}$$

$$V_{out} = 5.0\,\mathrm{V}\frac{723.4}{\sqrt{723.4^2 + 150^2}} \tag{14}$$

$$V_{out} = 0.979 * 5.0\,\mathrm{V} = 4.90\,\mathrm{V} \tag{15}$$

At $10\,\mathrm{Hz}$, there is very little voltage loss ($0.10\,\mathrm{V}$).

Programming

The three major modules of the program (linear actuator movement, SPI communications, and data dump to file) are complete and function as individual units. The code itself was programmed with Python v3. The major blocks of the code are located in Appendix E.

Calibration

Calibration of the linear actuator stepper motors was conducted by entering a command to move $100\,\mathrm{mm}$ in positive and negative directions on each axis, then measuring the actual distance traveled with precision vernier calipers. The ratio of distance commanded vs. distance traveled was then hard-coded into the linear actuator programming (Appendix E). The precision prior to calibration was $\pm 1.00\,\mathrm{mm}$ per $100\,\mathrm{mm}$ traveled in most cases. After calibration the precision was increased to an estimated $\pm 0.25\,\mathrm{mm}$ per $100\,\mathrm{mm}$. On short $1.00\,\mathrm{mm}$ movements required to map a magnetic field, the precision is negligible.

The SS-49e Hall sensors are considered linear devices; therefore, each was subjected to a two-point calibration with a known magnetic field that was measured with the NIST-calibrated Gauss meter (mentioned above). First, a location was marked on a small bar magnet and measured with the NIST-calibrated meter (about 925 gauss). Second, another location was marked on the table surface away from known magnetic fields and also measured (about 3 gauss). The second point proved to be only slightly above the noise threshold of the SS-49e sensors with fluctuations $\pm 3\,\mathrm{gauss}$, while the higher field measured $\pm 0.5\,\mathrm{gauss}$. From these two points the slopes and intercepts were determined and hard coded into the programming. Each sensor required a separate calibration process. Future recalibration of the sensors will prove to be difficult since they are super-glued to the 3D-printed PLA part.

Timelines

Both of these engineering tasks (combined) required just over two years to complete. Even so, the construction of the precision coil winder took one full year. The construction time for electromagnetic coils, rods, and rod mounts took a few weeks. The construction and programming time for the measurement apparatus took 9 months.

Experimental Procedure

Initial Field Measurements

FEMM models (Figure 14) were built using rods of the same dimensions available for data collection. The difference is that the FEMM models used rods formed of magnetic material, while the actual rods used to take initial field measurements was a steel alloy with an electromagnetic coil attached (Figure 13 for picture of actual rod and mounted coil).

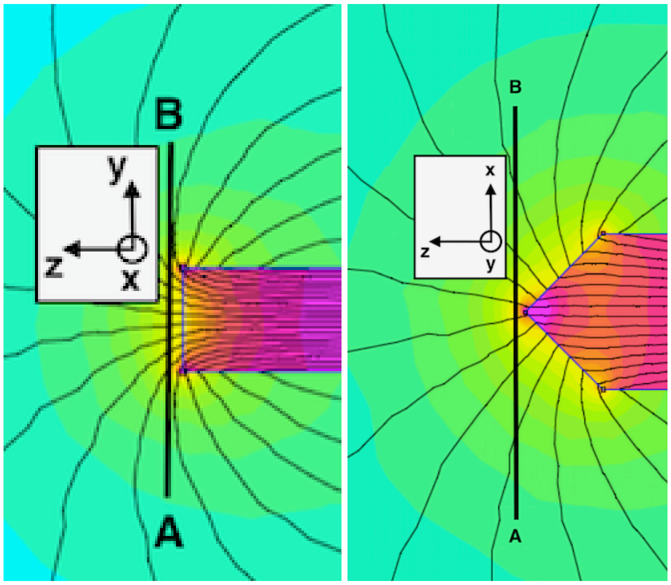

Figure 14: Field Measurement Paths. FEMM depicting measurement path for single Hall sensor sweep. Black contours are the field vector lines. The line labeled "A—B" in front of the rod tip is the data collection path. *Left:* Blunt rod, *Right:* Tipped rod.

The primary axis of symmetry is considered the z-axis for both experiments. A single NIST standard calibrated directional Hall sensor was mounted to a square wood dowel and measurements taken with a commercial Gauss meter.

Data was collected in four sweeps for the blunt rod. The first two sweeps were along the x- and y-axes of the rod and collected field strength in the z-direction (Figure 15). The third and fourth sweeps were conducted along the x- and y-axes and collected field strengths in the x- and y-directions (Figure 16). For the third and fourth sweeps, each half of the sweep required that the sensor be flipped to sense field vectors heading in opposite directions. Data was manually recorded (Appendix F) over a period of a couple of hours.

The magnetic center of the rod along each axis was determined by measuring the field for the lowest value. The datasheet for the SS-49e Hall sensor does indicate the location in the sensor package of the sensing plate; however, the size of the plate is not defined. If the datasheet drawing is to scale, the plate is about 0.75 mm by 0.50 mm. Based on this information, the estimation on the location of the two axes are about ± 0.5 mm from the true location. This situation was compounded by the requirement to flip the Hall sensor for each half of the sweep.

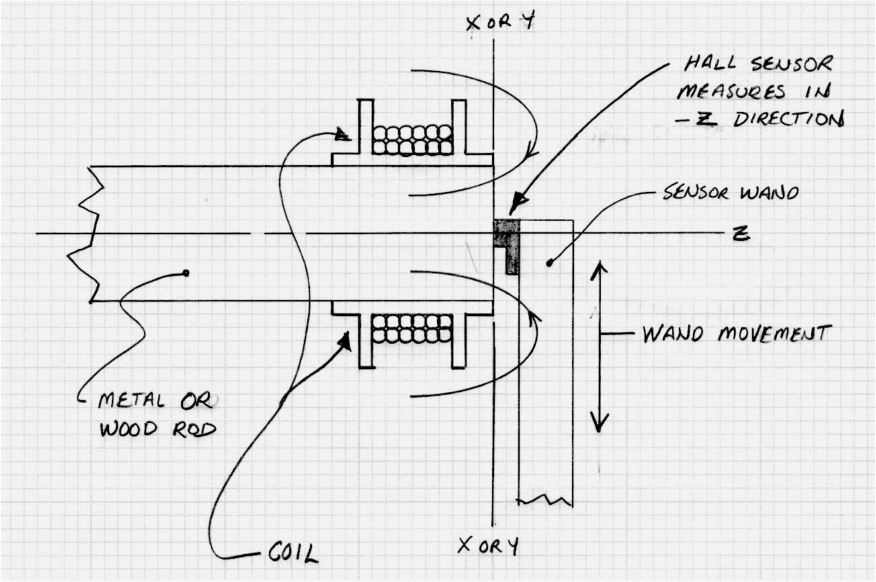

Figure 15: Setup for z-Component Measurement. The rod depicted in the graphic is blunt tipped, while the rod used is pointed. Otherwise the setup is identical.

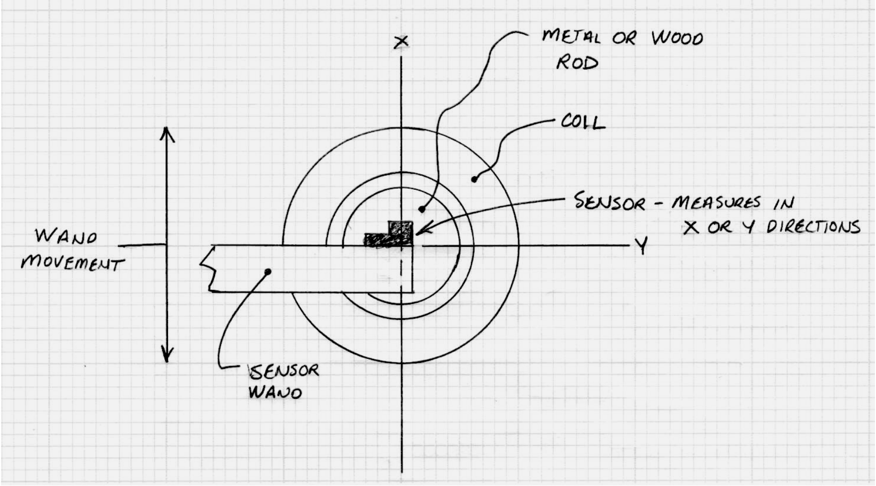

Figure 16: Setup for x- and y-Component Measurement.

Data was collected in three sweeps for the tipped rod. The first two sweeps were along the x- and y-axes of the rod and collected field strength in the z-direction. The third sweep to collect field strengths in the y-direction was conducted only along half of the y-axis. All data was recorded manually (Appendix G). The setups depicted in Figures 15 and 16 are the same as used for tipped rods, even though the depiction is for blunt rods.

After results of data collection from the blunt rod showed qualitative conformance to the model, data collection was halted on the tipped rod to begin pursuit of automating future data collection, flipping the sensor and getting the correct position proved too cumbersome for efficient and accurate data collection.

Results

z-Component data along both the x- and y-axis show remarkable symmetry and likeness of shape (Figure 17). The slight offset shown between the two curves is likely due to the imprecision in axis location. Another noteworthy characteristic of the curve is that the field is stronger near the cylindrical edges of the rod than the central axis.

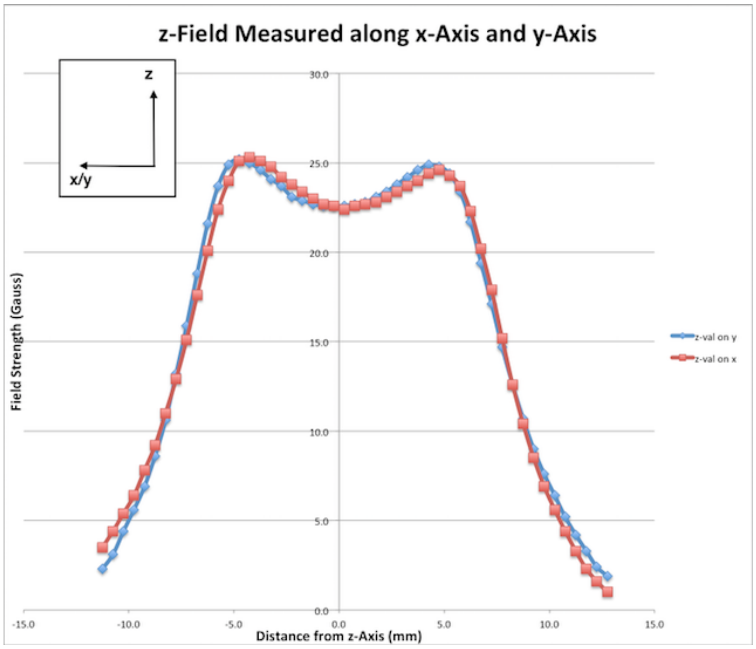

Figure 17: Blunt Rod z-Components. z-Component measurements along path labeled "A—B" near blunt rod. Gauss meter error in this range is ±2 %.

Radial component data along both the x- and y-axis also show symmetry and likeness of shape (Figure 18). The vertical center line of each pattern is fairly precise to the center of the rod. However, the field along the y-axis is significantly higher on one side of the rod than the other. It is also suspected that sensor misalignment is the significant cause for this apparent asymmetry.

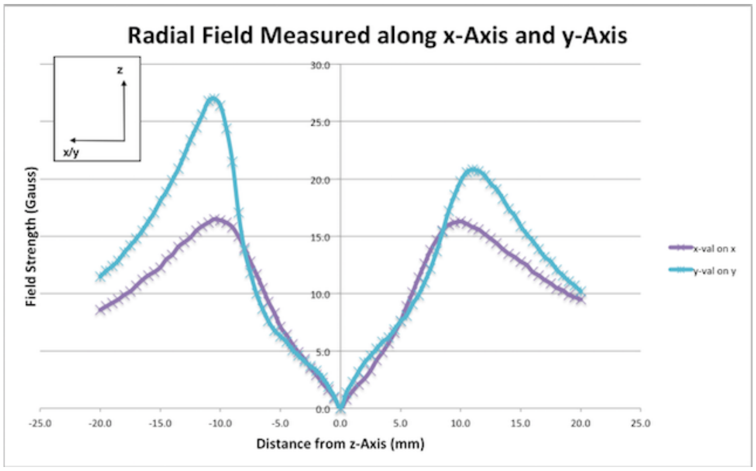

Figure 18: Blunt Rod Radial Components. Radial component measurements along path labeled “A—B” near blunt rod. Gauss meter error in this range is $\pm 2\,\%$.

The composite magnetic field strength is calculated from Tables 2 through 5 of Appendix F, using the following two equations. The results are found in Tables 6 and 7 of the same appendix and graphically depicted along with corresponding FEMM model in Figure 19. Comparison of the two pictures in Figure 19 shows excellent qualitative agreement between the measurements taken from the blunt rod and the corresponding FEMM computations.

$$B = \sqrt{b_x^2 + b_z^2} \tag{16}$$

$$\theta = \arctan \frac{b_x}{b_z} \tag{17}$$

$B =$	Composite Magnetic Field	[T]
$b_x =$	Radial Magnetic Field	[T]
$b_z =$	Axial Magnetic Field	[T]
$\theta =$	Angle	[deg]

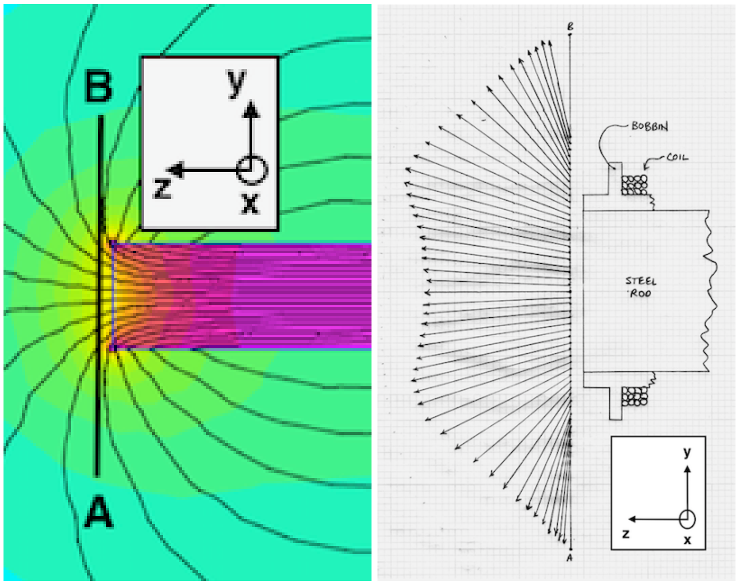

Figure 19: Blunt Rod FEMM vs Experimental Values. *Left:* FEMM model. *Right:* Composite field vectors (measured) along path.

The measurement of the tipped rod yielded similar symmetry and shape for the z-Component vectors (Figure 20). However, the tipped rod produced a clearly different radial field shape than the blunt rod (Figure 21).

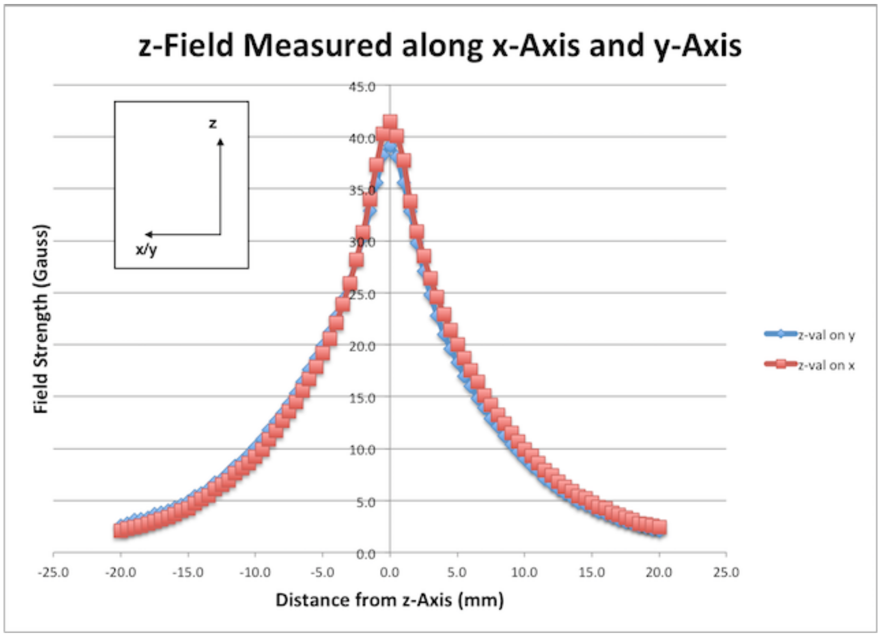

Figure 20: Tipped Rod z-Components. z-Component measurements along path labeled "A—B" near tipped rod. Gauss meter error in this range is ±2 %.

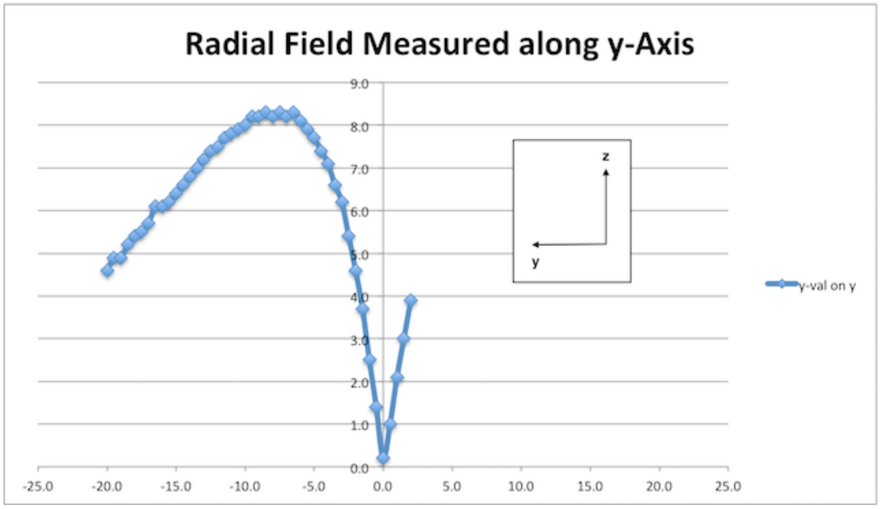

Figure 21: Tipped Rod Radial Components. Radial component measurements along path labeled "A—B" near tipped rod. Gauss meter error in this range is ±2 %.

The composite magnetic field strength is calculated from Tables 8 and 9 of Appendix G. The results are found in Table 10 of the same appendix and graphically depicted along with corresponding FEMM model in Figure 22. Again–for the data collected–their is an excellent qualitative agreement between the measurements taken from the tipped rod and the corresponding FEMM model.

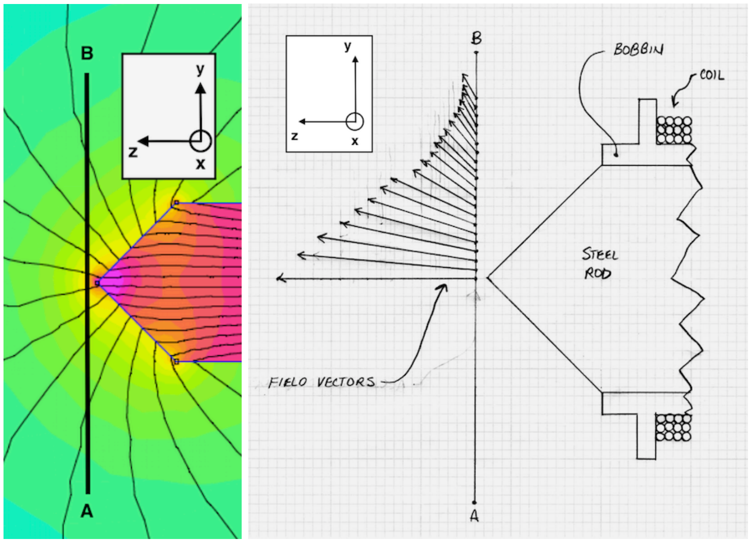

Figure 22: Tipped Rod FEMM vs Experimental Values. *Left:* FEMM model. *Right:* Composite field vectors (measured) along path labeled "A—B".

Conclusion

This project started over two years ago with the goal of building a magnetic tweezers setup to measure the viscosity of fluids with a macro-scale paramagnetic bead. A computational model produced by FEMM was employed to study a small handful of experimental setups involving one and two rods. The computations provided magnetic field vector and field gradient data that was analyzed to to determine a course of action for construction of an experimental setup.

Afterwards it was determined that construction of a precision coil winder was necessary to cut down on expenses associated with purchasing custom-made coils. Once completed two coils were wound for use to conduct initial data collection. After the two coils were completed the measurement apparatus was constructed and the linear actuators were calibrated.

Initial data collection was conducted on two experimental setups. Both used a single rod with an electromagnetic coil: one was a blunt rod, the other a tipped rod. A single NIST standard calibrated Hall sensor was used to manually collect the magnetic field data in each case. The data were compared to the FEMM computational model and found to have qualitative correlation in field shape. While the shapes agreed, magnitudes differed—the difference understandable considering that the computational model used a magnetic rod while the experimental setups used a steel rod with an electromagnetic coil.

Limitations in time prevented continuation of the project beyond an initial confirmation of the FEMM model. Recommendations for future steps in the project are discussed in the "Future Direction" Section. The steps laid out will finish up the construction of the measurement apparatus and place a future researcher in a good position to proceed to 3-dimensional mapping of experimental setups.

Future Direction

Linear Actuator Stability. An issue that became apparent during actuator calibration is the stability of the system. The z-Axis (parallel to the table surface) has the actuator only on one side of the apparatus. The friction of movement on the parallel extrusion rail causes a momentary bind when the actuator engages resulting in a slight lag of one side of the apparatus to the other. This can be easily fixed by purchasing another actuator assembly from OpenBuilds to install in the place of the parallel extrusion rails. The stepper motor would be wired in parallel with only very slight changes in circuitry and programming. Another stability concern is that the y-axis actuator shows slight wobble when bumped. This is may be improved by designing a better system of attaching the y-axis actuator to the x-axis actuator.

Linear Actuator Travel Range. At present the linear actuator software contains an entry that allows the user to set an arbitrary "zero" reference point at any place in the range of the actuator on each axis. However, the actuator system does not know where the limits of its travel begin and end. A possible solution is to install microswitches at the end points of each axis that will signal the stepper motors to stop when triggered. These switches (6 total, 2 for each axis) should be wired into Raspberry Pi GPIO ports. The Raspberry Pi should have software interrupts enabled on each port and associated code to call a stepper motor stop command for the corresponding axis. until this issue is resolved users will need to exercise care to not operate the stepper motors near the travel limits, otherwise the actuator system could be damaged.

Linear Actuator Calibration. The calibration of the linear actuator system used a set of precision vernier calipers that was manually placed along each axis to take measurements. The precision estimates therefore relied upon how well the calipers were placed along the axis. A more true calibration and assessment of actuator precision can be obtained via optical time-of-flight techniques using a HeNe laser with a photosensor mounted on the actuator.

Hall Sensor Array. The SS-49e Hall sensors on the sensor array should be replaced with sensors that have better precision at a lower range. The output signal fluctuates too greatly at less than 5 gauss to build a precise picture of the magnetic field in that range. Once new sensors are identified, they should be individually calibrated and then mounted on a sensor array block that will allow measurements closer to the end of the rod. Changing sensors may require an adjustment in the signal conditioning circuitry between the sensors and the ADC. Additionally the ADC in use is an MCP-3008, 10-bit ADC. The MCP-3208, 12-bit ADC should be investigated as an approach to gain more precision.

Automated Data Collection. While the three major blocks of programming are complete (Appendix E), the task remains to integrate the three modules in order to fully automate the data collection. The programming needed should be straight forward. The programmer will need to locate the correct placement of SPI communication function calls in the main body then execute the correct file operations function call to dump the incoming data into a file. File format will need to be determined for efficient data analysis using Mathematic, MatLab, Excel, or other analysis software.

New Electromagnetic Coils. Before precise fields are mapped on any experimental setup, the two coils that have been wound should be discarded. These are insufficient for precise data collection because the windings are not sufficiently uniform. Coil bobbins of sufficent precision can be 3D printed. One guide to help with winding new coils on the precision coil winder can be accessed here: https://www.youtube.com/watch?v=yfppXiNRnrc. The only exception to this guide is that after each layer is complete a thin sheet of cardboard or mylar should be wrapped around the coil before starting another layer to prevent the layers from merging. The guide noted that a coil would take only 15 minutes to complete; however, the author's experience is that hours should be set aside to produce one coil.

References

(1) Neuman, K. C.; Nagy, A.; Single-molecule Force Spectroscopy: Optical Tweezers, Magnetic Tweezers and Atomic Force Microscopy, *Nature Methods*, **June 2008**, *V5 N6*, 491-505.
(2) Desai, M. Z.; Hashemi, A. B.; Hirota, K.; Wong, A. J.; *Design of Magnetic Tweezers*, undated slide presentation.
(3) de Vries, A.; High Force Magnetic Tweezers for Molecular Manipulation inside Living Cells, **2004**
(4) Finite Element Method Magnetics: http://www.femm.info/wiki/HomePage/
(5) OpenBuilds Part Store: http://openbuildspartstore.com/
(6) Textron Easy-Go: http://www.ezgo.com/
(7) Janus Research Group: http://www.janusresearch.com/
(8) Precision Wafers Inc., *World's Largest Database of Soft Magnetic Materials*: http://magweb.us
(9) The Engineering Toolbox, *Wire Gauges - Current Ratings*: https://www.engineeringtoolbox.com/wire-gauges-d_419.html

Acknowledgments

Great appreciation is extended to Dr. Trevor Roebuck (Director of Testing at Textron Easy-Go) for use of the extensive machine shop for part fabrication. Dr Rodney Roeber (Chief Scientist, Janus Research Group) and Mr Scott Martin (Director of Reseach and Development, Janus Research Group) mentored in electronic circuit design and component selection. Prof. Andrew Hauger (Augusta University) also guided in electronic component selection and provided “seed” code for linear actuator movements. Prof. Thomas Colbert (Augusta University) gave numerous thoughts on how to calibrate and properly measure forces on small objects that requires deep consideration. Lastly, great thanks goes to Prof. Trinanjan Datta for patiently trusting me with getting this project off the ground—more engineering than science was accomplished in this effort; however, (hopefully) the next round of effort will be less engineering and a great deal more science.

Biography

Larry Hartman retired from the U.S. Air Force after 22 years of active duty on January 1, 2014 and immediately enrolled as a post-baccalaureate student at Augusta University to pursue a B.S. in Physics with minors in Chemistry and Mathematics. He is expected to graduate from Augusta University with Magna cum Laude honors in December 2017. He has earned diplomas from the Defense Language Institute in Basic and Intermediate Modern Standard Arabic, an A.A.S. in Communications Technology from the Community College of the Air Force, and a B.S. (Summa cum Laude) from DeVry in Electronics Engineering Technology. He is a (student) member of the American Physical Society (APS), American Chemical Society (ACS), the Optical Society of America (OSA), the International Photonics Society (SPIE), the Laser Institute of America (LIA), and the Institute of Electrical and Electronics Engineers (IEEE). Larry is presently employed as a part-time electronics engineer at Janus Research Group in Evans, GA.

Appendix A: Bead Specifications

The paramagnetic beads are grade 25 (±0.0025 mm, SAE 52100) chromium steel bearings.

- **Mo:** 0.08%
- **Mn:** 0.25-0.45%
- **Ni:** 0.25%
- **C:** 0.98-0.35%
- **Cr:** 1.30-1.60%
- **S:** 0.025%
- **Si:** 0.15-0.35%
- **P:** 0.025%
- **Fe:** 96.50-97.20%
- **Density:** 7.81 $\mathrm{g/cm^3}$
- **Susceptibility** > 300

Diameter (mm)	Volume $(\mathrm{mm})^3$	Average Mass (g)
1.0000 ± 0.0025	4.1888 ± 0.0105	0.00410
2.0000 ± 0.0050	33.5103 ± 0.0838	0.03278
2.5000 ± 0.0063	65.4498 ± 0.1636	0.06377
3.0000 ± 0.0075	113.0973 ± 0.2827	0.11011
3.5000 ± 0.0088	179.5944 ± 0.4490	0.17531
4.0000 ± 0.0100	268.0826 ± 0.6702	0.26133
4.5000 ± 0.0113	381.7035 ± 0.9563	0.37187
5.0000 ± 0.0125	523.5988 ± 1.3090	0.51043
6.0000 ± 0.0150	904.7787 ± 2.2619	0.88122

Table 1: Bead Dimensions. Masses were measured on a digital scale with a ten-bead sample size.

Appendix B: FEMM Software Setup Parameters

It is beyond the scope of this paper to provide instructions on the use of FEMM; however, the reader can access the user guides at the following location: http://www.femm.info/wiki/Documentation/. Useful instructions for making magnetic models can be found in the FEMM Reference Manual Magnetics Preprocessor and Postprocessor sections. Instructions for setting up electromagnetic coils can be found in the Magnetics Tutorial.

The following properties for "Problem Definition" were used in setting up models for this project:

- **Problem Type:** Planar
- **Length Units:** Millimeter
- **Frequency (Hz):** 0
- **Depth:** 1
- **Solver Precision:** 1e-008
- **Min Angle:** 30
- **AC Solver:** Succ. Approx.

The following properties for "Property Definition" was used in setting up models for this project:

- **Magnetic Rods:** NdFeB 52 MGOe
- **Interior Atmosphere:** Air
- **Boundaries:** u1 to u7

The geometry used for the rods matched the overall dimensions of rods manufactured for the project.

Appendix C: Rod Specifications

The two types of steel alloys chosen for rod construction are 1018 and 12L14. These types were chosen because they have stronger magnetic properties than other common steel alloys.

1018 Steel Alloy

- **Mn:** 0.60-0.90%
- **C:** 0.14-0.20%
- **S:** 0.05% Max
- **P:** 0.04% Max
- **Fe:** 98.81-99.26%
- **Density:** 7.87 g/cm^3

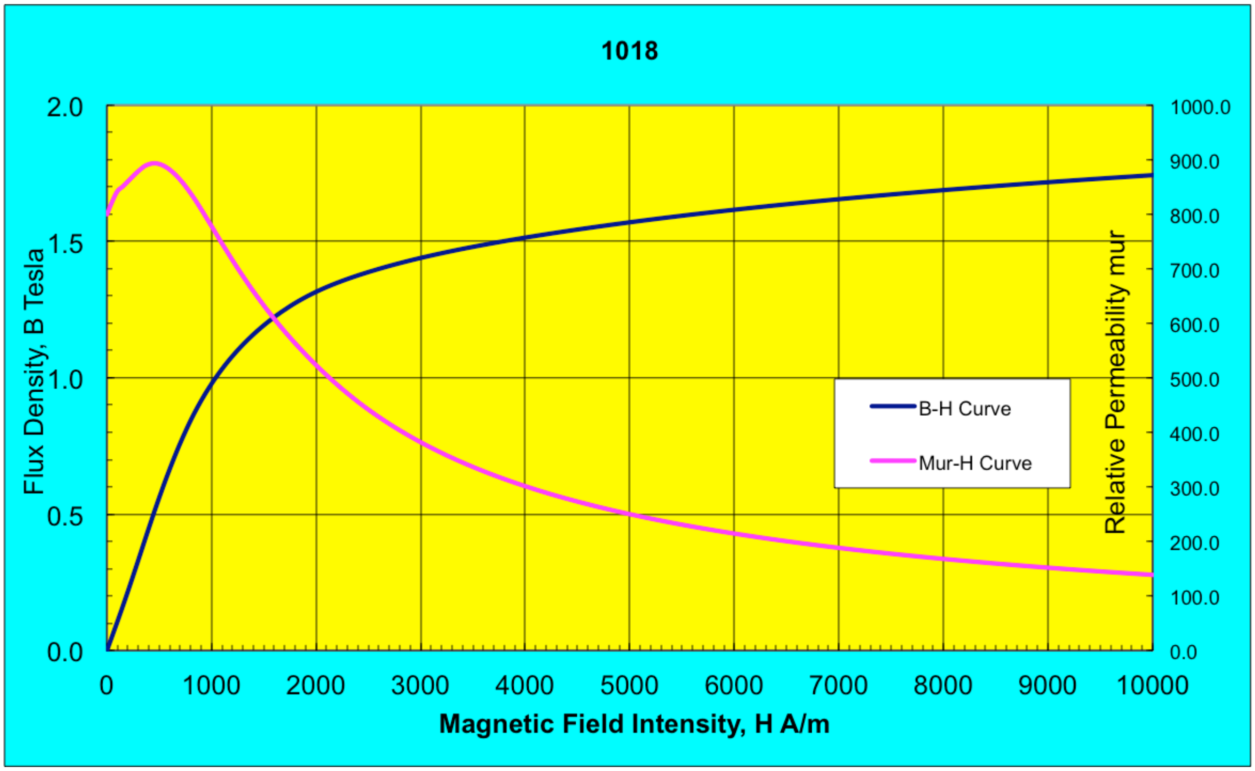

Figure 23: 1018 Magnetic Properties. Curves generated by and used with permission from Precision Wafers Inc.[8]

12L14 Steel Alloy

- **Mn:** 0.85-1.15%
- **C:** 0.15%
- **S:** 0.26-0.35%
- **P:** 0.04-0.09%
- **Pb:** 0.15-0.35%
- **Fe:** 97.91-98.70%
- **Density:** 7.87 g/cm^3

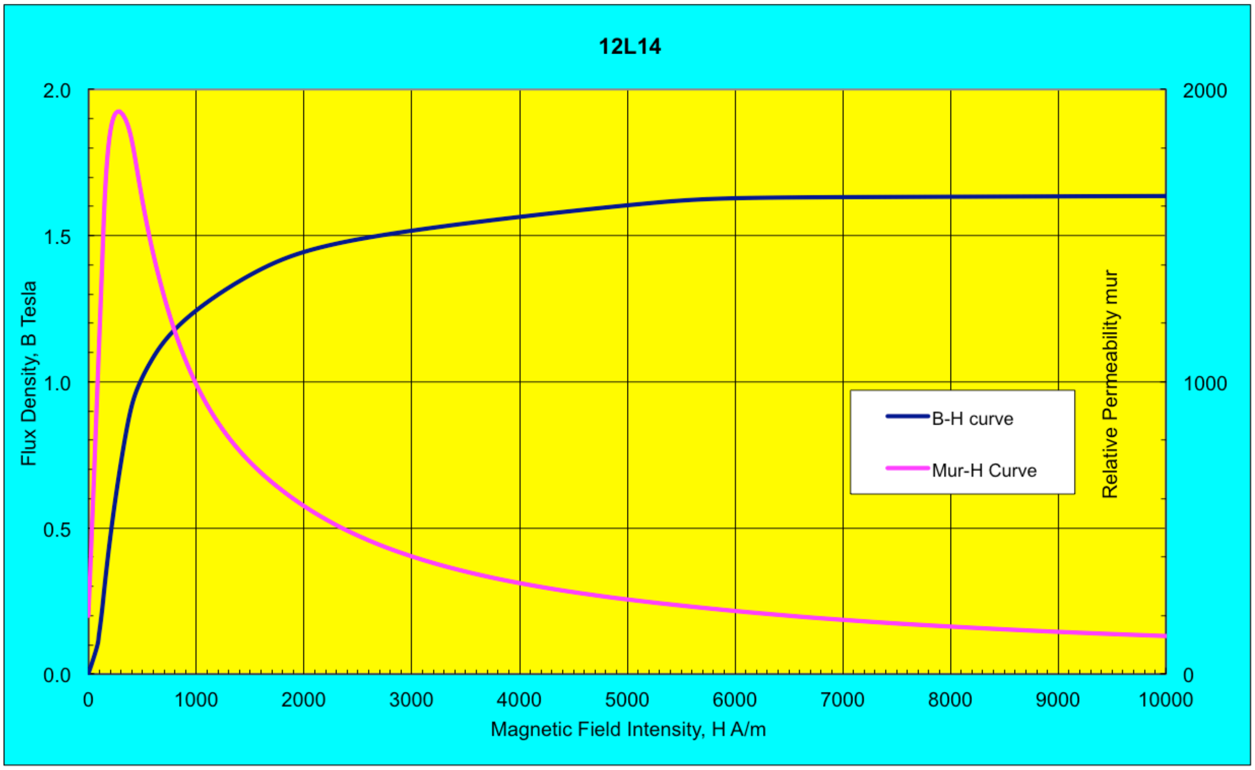

Figure 24: 12L14 Magnetic Properties. Curves generated by and used with permission from Precision Wafers Inc..[8]

Rod Dimensions

All rods are cut to 200mm. The remaining size specifications for each rod produced are:

- **Diam.:** 0.25 in, **Tip:** Blunt
- **Diam.:** 0.375 in, **Tip:** Blunt
- **Diam.:** 0.50 in, **Tip:** Blunt
- **Diam.:** 0.25 in, **Tip:** 90°
- **Diam.:** 0.375 in, **Tip:** 90°
- **Diam.:** 0.50 in, **Tip:** 90°
- **Diam.:** 0.25 in, **Tip:** 120°
- **Diam.:** 0.375 in, **Tip:** 120°
- **Diam.:** 0.50 in, **Tip:** 120°

Appendix D: Hall Sensor Schematic

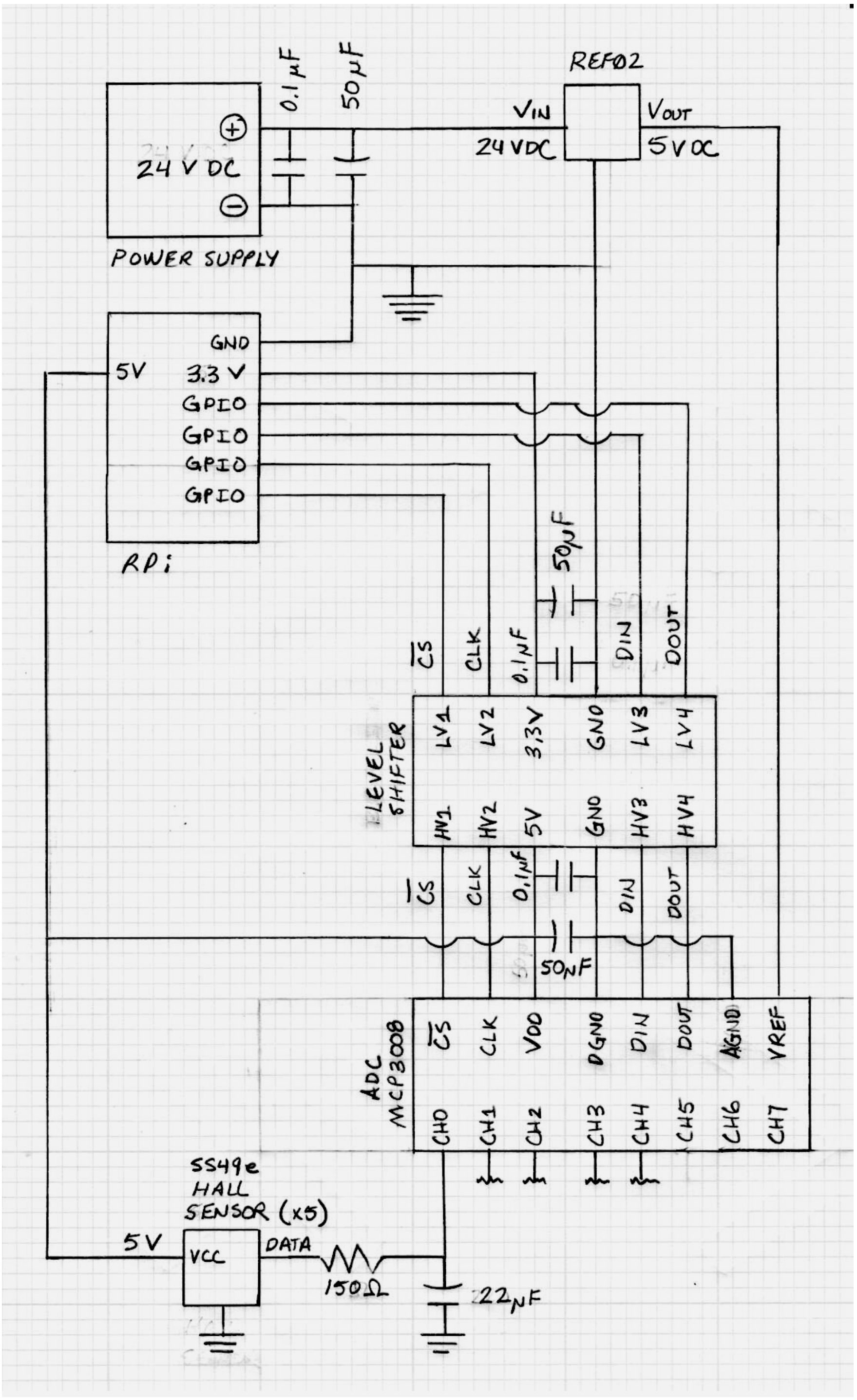

Figure 25: Hall Sensor Schematics. Designed by the author with significant input from Prof. Andrew Hauger (Augusta University) and Rodney Roeber (Janus Research Group)

Appendix E: Measurement Apparatus Code Algorithms

Actuator Movements in Python (for Raspberry Pi)

Original code to move a single-axis linear actuator was provided by Prof. Andrew Hauger (Augusta University) and expanded by the author for movement along three axes in separate movements. Included in the code are the calibration constants for the linear actuator stepper motors.

```
import time
import RPi.GPIO as GPIO
import sys

# GPIO Setup
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Direction and pulse pins
x_dirpin = 16
x_pulpin = 5
y_dirpin = 20
y_pulpin = 6
z_dirpin = 21
z_pulpin = 13

# Set GPIO pin modes
GPIO.setup(x_dirpin, GPIO.OUT)
GPIO.setup(x_pulpin, GPIO.OUT)
GPIO.setup(y_dirpin, GPIO.OUT)
GPIO.setup(y_pulpin, GPIO.OUT)
GPIO.setup(z_dirpin, GPIO.OUT)
GPIO.setup(z_pulpin, GPIO.OUT)

# Pitch: millimeters moved per single revolution
pitch = 8.0

# TODO Check value with switches on controller
# Stepper Controller switches: S1 = OFF, S2 = OFF, S3 = OFF
steps_rev = 6400

pulse_freq = 10000.0
pulse_delay = 1.0 / pulse_freq

x_move = 0
y_move = 0
z_move = 0

x_pos = 0
y_pos = 0
z_pos = 0

movement_interval = 0
```

```
x_movement = {'enable': False,
              'start': 0,
              'end': 0
              }

y_movement = {'enable': False,
              'start': 0,
              'end': 0
              }

z_movement = {'enable': False,
              'start': 0,
              'end': 0
              }

# Calibration constants determined 21 Aug 17
x_cal_neg = 0.9891  # Traveled 101.10 mm commanded 100 mm
x_cal_pos = 0.9901  # Traveled 100.95 mm commanded 100 mm
y_cal_neg = 0.9900  # Traveled 100.60 mm commanded 100 mm
y_cal_pos = 0.9881  # Traveled 100.95 mm commanded 100 mm
z_cal_neg = 0.9887  # Traveled 100.15 mm commanded 100 mm
z_cal_pos = 0.9867  # Traveled 101.30 mm commanded 100 mm

def move_motor(dirpin: int,
               pulpin: int,
               steps: int,
               delay: float,
               move: float,
               cal_neg: float,
               cal_pos: float
               ):
    """
    Moves motor

    :param dirpin: int
    :param pulpin: int
    :param steps: int
    :param delay: float
    :param move: float
    :param cal_neg: float
    :param cal_pos: float
    """
    # Less than 0, direction:
    # x-axis left
    # y-axis front
    # z-axis down
    #
    # Greater than 0, direction:
    # x-axis right
    # y-axis back
    # z-axis up
    if move < 0:
        GPIO.output(dirpin, True)
        cal_steps = int(cal_neg * float(steps))
    elif move > 0:
        GPIO.output(dirpin, False)
        cal_steps = int(cal_pos * float(steps))

    # Move axis
    for d in range(cal_steps):
        GPIO.output(pulpin, True)
        time.sleep(pulse_delay)
        GPIO.output(pulpin, False)
        time.sleep(pulse_delay)

def goto_position(x_move: float,
                  y_move: float,
                  z_move: float,
                  x_pos: float,
                  y_pos: float,
                  z_pos: float
                  ):
    """
    Transforms movement inputs into motor commands

    :param x_move: float
    :param y_move: float
    :param z_move: float
    :param x_pos: float
    :param y_pos: float
    :param z_pos: float
    """
    x_steps = int(steps_rev * abs(x_move) / pitch)
    y_steps = int(steps_rev * abs(y_move) / pitch)
    z_steps = int(steps_rev * abs(z_move) / pitch)

    if x_steps != 0:
        GPIO.output(y_pulpin, False)
        GPIO.output(z_pulpin, False)
        move_motor(x_dirpin,
                   x_pulpin,
                   x_steps,
                   pulse_delay,
                   x_move,
                   x_cal_neg,
                   x_cal_pos
                   )
        x_pos += -1 * x_move

    if y_steps != 0:
        GPIO.output(x_pulpin, False)
        GPIO.output(z_pulpin, False)
        move_motor(y_dirpin,
                   y_pulpin,
                   y_steps,
                   pulse_delay,
                   y_move,
                   y_cal_neg,
                   y_cal_pos
                   )
        y_pos += y_move

    if z_steps != 0:
        GPIO.output(x_pulpin, False)
        GPIO.output(y_pulpin, False)
        move_motor(z_dirpin,
                   z_pulpin,
                   z_steps,
                   pulse_delay,
                   z_move,
                   z_cal_neg,
                   z_cal_pos
                   )
        z_pos += z_move

    return x_pos, y_pos, z_pos
```

SPI Communications in Python (for Raspberry Pi)

Original code was accessed at https://raspberrypi-aa.github.io/session3/spi.html. It was heavily modified for use to automate data collection in this project. The calibration lines near the top of the code block are the values for each of the five Hall sensors presently attached to the sensor array.

```
import time
import sys
import spidev

spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz = 200000
spi.mode = 0b01
calibration = [
    {
        'ref_voltage': 5.03,
        'zero_voltage': 2.565,
        'sensitivity': 0.00119,
        'precision': 1
    },
    {
        'ref_voltage': 5.03,
        'zero_voltage': 2.539,
        'sensitivity': 0.00132,
        'precision': 1
    },
    {
        'ref_voltage': 5.03,
        'zero_voltage': 2.569,
        'sensitivity': 0.00131,
        'precision': 1
    },
    {
        'ref_voltage': 5.03,
        'zero_voltage': 2.55,
        'sensitivity': 0.00135,
        'precision': 1
    },
    {
        'ref_voltage': 5.03,
        'zero_voltage': 2.578,
        'sensitivity': 0.00128,
        'precision': 1
    }
]

def buildReadCommand(channel):
    startBit = 0x01
    singleEnded = (channel << 4) | 0b10000000

    # Return python list of 3 bytes
    #   Build a python list using [1, 2, 3]
    #   First byte is the start bit
    #   Second byte contains single ended along with channel #
    #   3rd byte is 0
    return [startBit, singleEnded, 0]

def readADC(calibration, channel):
    if ((channel > 7) or (channel < 0)):
        return -1
    result = spi.xfer2(buildReadCommand(channel))
    value_bit_raw = (result[1] << 8) + result[2]
    value_10bit = value_bit_raw & 0b1111111111
    value_deltavolt = (((value_10bit * calibration['ref_voltage']) / 1024) - calibration['z
    value_gauss = value_deltavolt / calibration['sensitivity']
    value_gauss_rounded = round(value_gauss, 1)
    return value_gauss_rounded

def processSensorValues():
    values = [0,0,0,0,0,0,0,0,0,0,0]
    for reading in range(0, 11):
        values[reading] = readAdc(calibration[sensor], sensor)
        time.sleep(0.5)
    del values[0]
    print(values)
    values_avg = round(sum(values)/float(len(values)), 1)
    print(values_avg)
```

File Operations in Python (for Raspberry Pi)

Original code was programmed by the author for use at Janus Research Group. It was modified to remove proprietary elements and adapted for use to automate data collection in this project.

```
import logging
import time
from datetime import datetime
from os import close, remove
from random import randint
from shutil import move
from tempfile import mkstemp


def request(
    file_cmd: str,
    file_name: str,
    data_loc: int,
    num_bytes: int,
    data_file_in: str,
    search_field: str='',
    replace_line: str='',
):
    """
    Executes file transaction

    :param file_cmd: str
    :param file_name: str
    :param data_loc: int
    :param num_bytes: int
    :param search_field: str,
    :param replace_line: str,
    :param data_file_in: str

    :return read_err: bool
    :return data: 0 (read_err = True)
    :return data: str (read_err = False)
    """
    data_file_out = 0
    file_mode = ''
    field_dict = {}
    temp_path = None
    file_hdlr = None
    file_temp = None

    if file_cmd == 'data_read':
        file_mode = 'r'

    elif file_cmd == 'line_app':
        file_mode = 'a+'

    elif file_cmd == 'fld_read':
```

```
    file_mode = 'r'

elif file_cmd == 'fld_read_all':
    file_mode = 'r'

elif file_cmd == 'fld_edit':
    file_mode = 'r'

elif file_cmd == 'data_wrt':
    file_mode = 'r+'

elif file_cmd == 'file_replace':
    file_mode = 'w'

# Open file in specified mode, utf-8
file_open = open(
    file_name,
    mode=file_mode,
    encoding='utf-8'
)

if file_cmd == 'data_read':
    file_open.seek(data_loc)
    data_file_out = file_open.read(num_bytes)

elif file_cmd == 'line_app':
    file_open.write(data_file_in + '\n')
    file_open.flush()

elif file_cmd == 'fld_read':
    for line in file_open:
        line = line.split('\n')[0]
        key = line.split('=')[0]
        if key == search_field:
            data_file_out = line.split('=')[1]

elif file_cmd == 'fld_read_all':
    for line in file_open:
        line = line.split('\n')[0]
        key, val = line.split('=')
        field_dict[key] = str(val)

elif file_cmd == 'fld_edit':
    file_hdlr, temp_path = mkstemp()
    file_temp = open(temp_path,
                     mode='w',
                     encoding='utf-8'
                     )

    found_field = False
    for line in file_temp:
        if search_field in line:
            file_temp.write(replace_line + '\n')
            found_field = True
        else:
            file_temp.write(line)

   if not found_field:
        file_temp.write(replace_line)

    file_temp.flush()

elif file_cmd == 'data_wrt':
    file_open.seek(data_loc)
    file_open.write(data_file_in)
    file_open.flush()

elif file_cmd == 'file_replace':
    file_open.seek(0)
    file_open.write(data_file_in)
    file_open.flush()

# Close file
file_open.close()

if file_cmd == 'fld_edit':
    remove(file_name)
    move(temp_path, file_name)
    file_temp.close()
    close(file_hdlr)

return data_file_out
```

Appendix E: Data from Single Hall Sensor on Blunt Rod

Position (mm)	Field Strength (G)	Position (mm)	Field Strength (G)
−12.0	3.5	0.5	22.7
−11.5	4.4	1.0	22.8
−11.0	5.4	1.5	23.1
−10.5	6.4	2.0	23.4
−10.0	7.8	2.5	23.7
−9.5	9.2	3.0	24.0
−9.0	11.0	3.5	24.4
−8.5	12.9	4.0	24.6
−8.0	15.1	4.5	24.3
−7.5	17.6	5.0	23.7
−7.0	20.1	5.5	22.3
−6.5	22.4	6.0	20.2
−6.0	24.0	6.5	17.9
−5.5	25.1	7.0	15.2
−5.0	25.3	7.5	12.6
−4.5	25.1	8.0	10.4
−4.0	24.8	8.5	8.5
−3.5	24.2	9.0	6.9
−3.0	23.8	9.5	5.6
−2.5	23.4	10.0	4.4
−2.0	23.0	10.5	3.3
−1.5	22.7	11.0	2.3
−1.0	22.6	11.5	1.6
−0.5	22.4	12.0	1.0
0.0	22.6		

Table 2: x-Direction Field Strength along x-Axis.

Position (mm)	Field Strength (G)	Position (mm)	Field Strength (G)
−12.0	2.3	0.5	22.8
−11.5	3.1	1.0	23.1
−11.0	4.4	1.5	23.4
−10.5	5.6	2.0	23.8
−10.0	6.9	2.5	24.2
−9.5	8.6	3.0	24.6
−9.0	10.6	3.5	24.9
−8.5	13.2	4.0	24.8
−8.0	15.9	4.5	24.4
−7.5	18.8	5.0	23.4
−7.0	21.6	5.5	21.7
−6.5	23.7	6.0	19.4
−6.0	24.9	6.5	17.1
−5.5	25.2	7.0	14.7
−5.0	25.0	7.5	12.6
−4.5	24.6	8.0	10.7
−4.0	24.1	8.5	9.0
−3.5	23.7	9.0	7.6
−3.0	23.1	9.5	6.4
−2.5	22.9	10.0	5.2
−2.0	22.7	10.5	4.2
−1.5	22.6	11.0	3.3
−1.0	22.6	11.5	2.4
−0.5	22.6	12.0	1.9
0.0	22.7		

Table 3: y-Direction Field Strength along y-Axis.

Position (mm)	Field Strength (G)	Position (mm)	Field Strength (G)
−12.0	15.5	0.5	0.8
−11.5	15.9	1.0	1.5
−11.0	16.2	1.5	2.1
−10.5	16.5	2.0	2.6
−10.0	16.4	2.5	3.3
−9.5	16.2	3.0	4.2
−9.0	15.8	3.5	4.9
−8.5	15.0	4.0	5.7
−8.0	14.0	4.5	6.6
−7.5	12.8	5.0	7.6
−7.0	11.7	5.5	8.9
−6.5	10.5	6.0	10.1
−6.0	9.3	6.5	11.4
−5.5	8.3	7.0	12.6
−5.0	7.1	7.5	13.8
−4.5	6.4	8.0	14.7
−4.0	5.6	8.5	15.5
−3.5	4.9	9.0	16.0
−3.0	4.3	9.5	16.2
−2.5	3.5	10.0	16.3
−2.0	2.9	10.5	16.1
−1.5	2.2	11.0	15.8
−1.0	1.6	11.5	15.6
−0.5	0.9	12.0	15.2
0.0	0.0		

Table 4: z-Direction Field Strength along x-Axis.

Position (mm)	Field Strength (G)	Position (mm)	Field Strength (G)
−12.0	24.5	0.5	1.4
−11.5	25.6	1.0	2.3
−11.0	26.8	1.5	3.2
−10.5	27.0	2.0	4.0
−10.0	26.4	2.5	4.6
−9.5	24.4	3.0	5.2
−9.0	21.5	3.5	5.8
−8.5	17.1	4.0	6.2
−8.0	13.8	4.5	6.9
−7.5	11.8	5.0	7.6
−7.0	10.0	5.5	8.1
−6.5	8.7	6.0	9.1
−6.0	7.6	6.5	9.9
−5.5	6.8	7.0	11.0
−5.0	6.3	7.5	12.2
−4.5	5.8	8.0	13.7
−4.0	5.1	8.5	15.4
−3.5	4.6	9.0	17.2
−3.0	4.1	9.5	18.6
−2.5	3.7	10.0	19.8
−2.0	3.3	10.5	20.6
−1.5	2.7	11.0	20.8
−1.0	1.9	11.5	20.7
−0.5	1.0	12.0	20.3
0.0	0.0		

Table 5: z-Direction Field Strength along y-Axis.

Position (mm)	Field Strength (G)	Angle	Position (mm)	Field Strength (G)	Angle
−12.0	15.9	77.3	0.5	22.7	2.0
−11.5	16.5	74.5	1.0	22.8	3.8
−11.0	17.1	71.6	1.5	23.2	5.2
−10.5	17.7	68.8	2.0	23.5	6.3
−10.0	18.2	64.6	2.5	23.9	7.9
−9.5	18.6	60.4	3.0	24.4	9.9
−9.0	19.3	55.2	3.5	24.9	11.4
−8.5	19.8	49.3	4.0	25.3	13.0
−8.0	20.6	42.8	4.5	25.2	15.2
−7.5	21.8	36.0	5.0	24.9	17.8
−7.0	23.3	30.2	5.5	24.0	21.8
−6.5	24.7	25.1	6.0	22.6	26.6
−6.0	25.7	21.2	6.5	21.2	32.5
−5.5	26.4	18.3	7.0	19.7	39.7
−5.0	26.3	15.7	7.5	18.7	47.6
−4.5	25.9	14.3	8.0	18.0	54.7
−4.0	25.4	12.7	8.5	17.7	61.2
−3.5	24.7	11.4	9.0	17.4	66.7
−3.0	24.2	10.2	9.5	17.1	70.9
−2.5	23.7	8.5	10.0	16.9	74.9
−2.0	23.2	7.2	10.5	16.4	78.4
−1.5	22.8	5.5	11.0	16.0	81.8
−1.0	22.7	4.0	11.5	15.7	84.1
−0.5	22.4	2.3	12.0	15.2	86.2
0.0	22.6	0.0			

Table 6: Composite Vector Values along x-Axis.

Position (mm)	Field Strength (G)	Angle	Position (mm)	Field Strength (G)	Angle
−12.0	24.6	84.6	0.5	22.8	3.5
−11.5	25.8	83.1	1.0	23.2	5.7
−11.0	27.2	80.7	1.5	23.6	7.8
−10.5	27.6	78.3	2.0	24.1	9.5
−10.0	27.3	75.4	2.5	24.6	10.8
−9.5	25.9	70.6	3.0	25.1	11.9
−9.0	24.0	63.8	3.5	25.6	13.1
−8.5	21.6	52.3	4.0	25.6	14.0
−8.0	21.1	41.0	4.5	25.4	15.8
−7.5	22.2	32.1	5.0	24.6	18.0
−7.0	23.8	24.8	5.5	23.2	20.5
−6.5	25.2	20.2	6.0	21.4	25.1
−6.0	26.0	17.0	6.5	19.8	30.1
−5.5	26.1	15.1	7.0	18.4	36.8
−5.0	25.8	14.1	7.5	17.5	44.1
−4.5	25.3	13.3	8.0	17.4	52.0
−4.0	24.6	11.9	8.5	17.8	59.7
−3.5	24.1	11.0	9.0	18.8	66.2
−3.0	23.5	10.1	9.5	19.7	71.0
−2.5	23.2	9.2	10.0	20.5	75.3
−2.0	22.9	8.3	10.5	21.0	78.5
−1.5	22.8	6.8	11.0	21.1	81.0
−1.0	22.7	4.8	11.5	20.8	83.4
−0.5	22.6	2.5	12.0	20.4	84.7
0.0	22.7	0.0			

Table 7: Composite Vector Values along y-Axis.

Appendix F: Data from Single Hall Sensor on Tipped Rod

Position (mm)	z-Field Strength on x-Axis(G)	z-Field Strength on y-Axis(G)	Position (mm)	z-Field Strength on x-Axis(G))	z-Field Strength on y-Axis(G)
−20.0	2.1	2.6	0.5	40.1	38.1
−19.5	2.3	2.8	1.0	37.7	35.6
−19.0	2.4	3.1	1.5	33.8	32.8
−18.5	2.6	3.2	2.0	30.9	29.8
−18.0	2.8	3.3	2.5	28.5	27.1
−17.5	3.0	3.7	3.0	26.4	24.8
−17.0	3.2	3.8	3.5	24.6	22.8
−16.5	3.4	4.0	4.0	22.9	21.0
−16.0	3.7	4.3	4.5	21.4	19.6
−15.5	4.0	4.6	5.0	20.0	18.3
−15.0	4.2	4.9	5.5	18.7	17.0
−14.5	4.7	5.4	6.0	17.5	16.0
−14.0	5.1	5.7	6.5	16.4	14.8
−13.5	5.5	6.1	7.0	15.1	13.9
−13.0	6.1	6.7	7.5	14.2	12.9
−12.5	6.5	7.1	8.0	13.2	12.1
−12.0	6.9	7.7	8.5	12.4	11.2
−11.5	7.6	8.3	9.0	11.5	10.4
−11.0	8.1	8.8	9.5	10.7	9.7
−10.5	8.6	9.4	10.0	9.9	9.0
−10.0	9.2	10.1	10.5	9.3	8.3
−9.5	9.9	10.9	11.0	8.6	7.8
−9.0	10.9	11.8	11.5	7.9	7.1
−8.5	11.7	12.6	12.0	7.4	6.6
−8.0	12.7	13.4	12.5	6.8	6.1
−7.5	13.6	14.3	13.0	6.3	5.7
−7.0	14.5	15.3	13.5	5.9	5.3
−6.5	15.6	16.4	14.0	5.4	4.8
−6.0	16.7	17.6	14.5	5.2	4.6
−5.5	17.9	18.7	15.0	4.8	4.3
−5.0	19.2	19.9	15.5	4.4	3.9
−4.5	20.6	21.2	16.0	4.2	3.7
−4.0	22.1	22.7	16.5	3.8	3.5
−3.5	23.9	24.3	17.0	3.6	3.1
−3.0	25.9	26.1	17.5	3.3	2.9
−2.5	28.2	28.1	18.0	3.1	2.8
−2.0	30.8	30.4	18.5	2.8	2.6
−1.5	34.0	32.9	19.0	2.7	2.4
−1.0	37.3	35.6	19.5	2.6	2.2
−0.5	40.3	38.2	20.0	2.4	2.1
0.0	41.5	39.1			

Table 8: z-Field Strength on x-Axis and y-Axis.

Position (mm)	Field Strength (G)	Position (mm)	Field Strength (G)
−20.0	4.6	−8.5	8.3
−19.5	4.9	−8.0	8.2
−19.0	4.9	−7.5	8.3
−18.5	5.2	−7.0	8.2
−18.0	5.4	−6.5	8.3
−17.5	5.5	−6.0	8.1
−17.0	5.7	−5.5	7.9
−16.5	6.1	−5.0	7.7
−16.0	6.1	−4.5	7.4
−15.5	6.2	−4.0	7.1
−15.0	6.4	−3.5	6.6
−14.5	6.6	−3.0	6.2
−14.0	6.8	−2.5	5.4
−13.5	7.0	−2.0	4.6
−13.0	7.2	−1.5	3.7
−12.5	7.4	−1.0	2.5
−12.0	7.5	−0.5	1.4
−11.5	7.7	0.0	0.2
−11.0	7.8	0.5	1.0
−10.5	7.9	1.0	2.1
−10.0	8.0	1.5	3.0
−9.5	8.2	2.0	3.9
−9.0	8.2		

Table 9: y-Direction Field Strength along y-Axis.

Position (mm)	Field Strength (G)	Angle	Position (mm)	Field Strength (G)	Angle
−20.0	5.3	60.5	−8.5	15.1	33.4
−19.5	5.6	60.3	−8.0	15.7	31.5
−19.0	5.8	57.7	−7.5	16.5	30.1
−18.5	6.1	58.4	−7.0	17.4	28.2
−18.0	6.3	58.6	−6.5	18.4	26.8
−17.5	6.6	56.1	−6.0	19.4	24.7
−17.0	6.9	56.3	−5.5	20.3	22.9
−16.5	7.3	56.7	−5.0	21.3	21.2
−16.0	7.5	54.8	−4.5	22.5	19.2
−15.5	7.7	53.4	−4.0	23.8	17.4
−15.0	8.1	52.7	−3.5	25.2	15.2
−14.5	8.5	50.7	−3.0	26.8	13.4
−14.0	8.9	50.0	−2.5	28.6	10.9
−13.5	9.3	48.9	−2.0	30.7	8.6
−13.0	9.8	47.1	−1.5	33.1	6.4
−12.5	10.3	46.2	−1.0	35.7	4.0
−12.0	10.7	44.2	−0.5	38.2	2.1
−11.5	11.3	42.9	0.0	39.1	0.3
−11.0	11.8	41.6	0.5	38.1	1.5
−10.5	12.3	40.0	1.0	35.7	3.4
−10.0	12.9	38.4	1.5	32.9	5.2
−9.5	13.6	37.0	2.0	30.0	7.5
−9.0	14.4	34.8			

Table 10: Composite Vector Values along y-Axis.