12-2020

Modeling the Torque of Ferrite Nano-Particles as a Ferrofluid Suspended in Liquid to Determine the Electromagnetic Response to Angular Displacement

Jackson Tyler Brennecke
Grand Valley State University

Modeling the Torque of Ferrite Nano-Particles as a Ferrofluid Suspended in Liquid to Determine the Electromagnetic Response to Angular Displacement

Jackson Tyler Brennecke

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Engineering

Padnos College of Engineering and Computing

December 2020

Acknowledgements

I would like to thank my parents, Andrew and Jessica, for their support and encouragement throughout my college education. They are the reason that I was able to be a part of this academic program, and I am incredibly grateful for everything that they have done. I also need to thank my fiance Jenna, because I would not be here without her love and support.

I also want to thank my thesis advisor, Dr. Brian Krug, who worked with me through every step of this research. Dr. Krug routinely dedicated his time and effort to provide me with guidance leading up to and during this research process. I could not have asked for a more reliable mentor. I am also thankful to my other thesis committee members Dr. Joshua Veazey and Dr. Karl Brakora, who have dedicated their time and effort as well.

Lastly, I want to thank my graduate advisor Dr. Shabbir Choudhrui for going above and beyond to make sure that I succeeded in my academic and professional endeavors. From helping me schedule and plan course loads, to finding grants and scholarships, to answering questions about real world applications, Dr. Choudhuri has been one of the most valuable mentors that I have had.

Jackson Brennecke

Preface

This thesis was completed almost entirely during the national COVID-19 pandemic. The circumstances presented by the COVID-19 pandemic posed some challenges to the research, specifically the empirical testing and data collection.

Material from this thesis was also presented publicly at the Michigan Space Grant Consortium conference in October 2020.

Abstract

Inertial sensing is an important part of engineering and technology, especially for determining spatial orientation. Most modern inertial sensing units rely on MicroElectroMechanical systems (MEMS) style gyroscopic sensors to determine angular acceleration. This research investigates a novel gyroscopic sensing technology that uses mechanical precession of magnetic nanoparticles, instead of MEMS, to determine inertial measurements. The only other study on this novel technology proposed a scalar set of equations for relating magnetic field and torque magnitude to the magnitude of angular displacement of the sensor. This research develops the theoretical model into a set of full vector equations, so that the magnetic field and torque can be related to both the magnitude and direction of angular displacement of the sensor. It was determined that inertial components of nanoparticle torque in the original model are negligible due to scaling laws at the nanoscale, and that the only significant contributions are due to viscous fluid drag, which changed the theoretical equations considerably. Euler rotation angles were used to derive a decomposed 3D vector that represents the torque and magnetic field of the nanoparticle response to angular displacement. Simulations verified the assumptions made in the model, and overall it was concluded that, theoretically, the sensor technology could work and is viable for further applications. However, improvements should be made to the sensor design in order to improve electromagnetic immunity to exterior sources.

Table of Contents

List of Tables

List of Figures

Key to Symbols

Symbol	Description
$B_{\text{precessed}}$	Magnetic field magnitude of precessed nanoparticles
$B_{\text{not\_recessed}}$	Magnetic field magnitude of steady state not precessed nanoparticles
I_{sphere}	Moment of inertia for spherical object
$\boldsymbol{\Omega}$	Angular displacement (sensor tilt) rate (rad/s)
$\boldsymbol{\omega}$	Rotational velocity (nanoparticle spin rate) (rad/s)
$\boldsymbol{\eta}$	Fluid viscosity
$\boldsymbol{V}$	Volume
M	Single domain magnetic nanoparticle magnetic moment
θ_1	Difference between nanoparticle magnetic moment and applied magnetic field
θ_2	Tilt angle
$\boldsymbol{\alpha_a}$	Angular acceleration (rad/s^2)
$\boldsymbol{\tau}_{\text{precessed}}$	Nanoparticle torque when precessed
$\boldsymbol{\tau}_{\text{not\_precessed}}$	Nanoparticle torque when not precessed
$\overline{m}$	Magnetic moment of nanoparticle (vector)
m	Mass
r	Radius
$\boldsymbol{\alpha}$	Euler rotation angle - Z1
$\boldsymbol{\beta}$	Euler rotation angle - X
$\boldsymbol{\gamma}$	Euler rotation angle - Z2
$\overline{B}$	Magnetic field vector
$\boldsymbol{\mu}_0$	Vacuum permeability of free space

I	Conventional electrical current
$\overline{dL}$	Differential wire segment length vector
$\bar{r}$	Distance vector between wire and point
$\hat{r}$	Unit vector between wire and point
ϱ	Density
d	Diameter
D	Characteristic dimension
θy	Single axis angular displacement angle
f	Sinusoidal frequency (of AC electrical current)
Bx	X component of magnetic field vector
By	Y component of magnetic field vector
Bz	Z component of magnetic field vector
N	Number of wire (solenoid) coils
R	Radius
Ix	Electrical current induced from magnetic field in X direction
Iy	Electrical current induced from magnetic field in Y direction
Iz	Electrical current induced from magnetic field in Z direction
χ	Magnetic susceptibility

Chapter 1: Introduction

1.1) Introduction

Inertial sensing is an important part of engineering and technology, especially for determining spatial orientation. One of the most important aspects of inertial sensing is being able to translate a measurable input signal to its corresponding inertial movement. It is important to know not just that something moved, but in what direction and in what period of time that it moved. One component that is used to determine inertial information is the gyroscopic sensor. Gyroscopes are used in many electronics applications today, ranging from mobile cell phones, to marine navigation units, to missile or rocket ship control units [1]. At their core, all gyroscopes are devices that sense changes in angular velocity. The first gyroscopes were purely mechanical, but due to advances in technology there are now a handful of different types of gyroscopic sensors available. The most common gyroscopic sensors used today are the MEMS type, due to their relatively good performance and low cost, made possible through advancements in manufacturing [2].

Even though MEMS gyroscopic sensors are widely used for many applications, they do have some drawbacks and they are not suitable for every application. Most notably, MEMS gyroscope sensors can be less accurate over longer periods of time than other types of gyroscopic sensors because biasing can vary with voltage and temperature, which results in drift [18]. When measuring Bias Stability, specifically, other types of sensors can obtain up to 5 orders of magnitude better ratings than MEMS [3]. This is illustrated in figure 1.

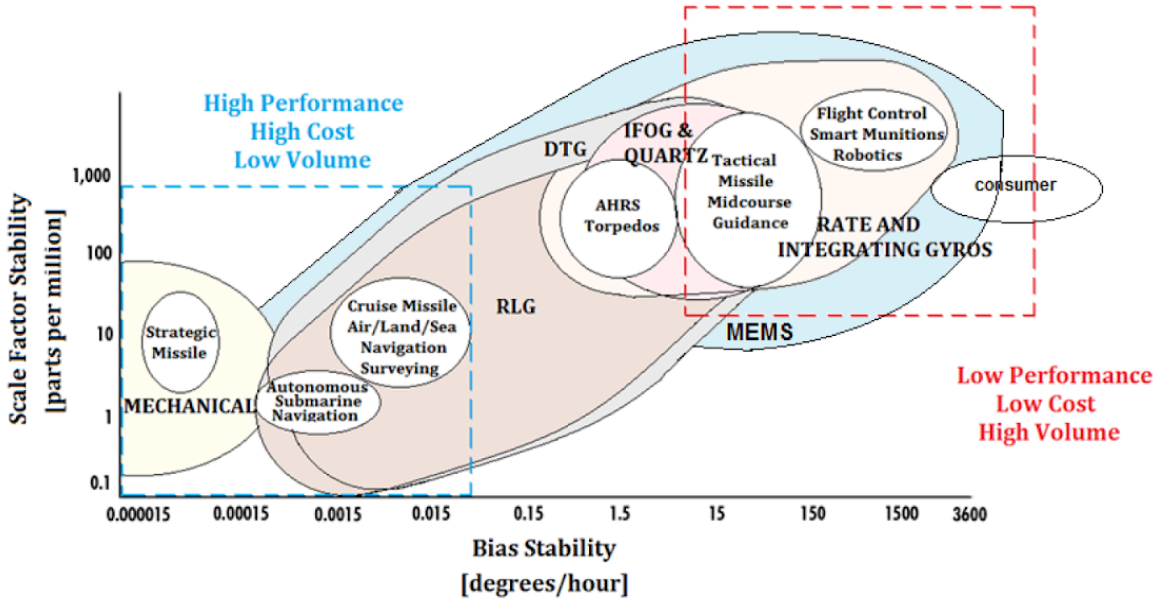

Figure 1: Gyroscope technology performance chart [1]

A novel inertial sensor technology that relies on the gyroscopic properties of magnetic nanoparticles is investigated here [4]. This technology utilizes ferrite nanoparticles with an average radius of about 15nm, which is smaller than the finest features found in MEMS style gyroscopes. The working principle of this technology, however, is mechanical precession, rather than the vibrating phenomena employed by MEMS style gyroscopes. The goal of this technology is to provide an alternative to MEMS gyroscopes that provides increased immunity to environmental factors, such as temperature and pressure, while also providing accurate inertial measurements.

1.2) Purpose

The goal of this research is to model the 3D torque of nano-particles in this sensor technology to determine the electromagnetic response to angular displacement. The primary objectives of this research are:

i) To theorize a model consistent with ferrofluid mechanics to describe the torque of ferrite nano-particles as a 3D vector.

ii) To simulate the electromagnetic fields, and effects thereof, for the ferrite nano-particle model with an alternating current supply.

iii) To compare simulated and theorized results to empirically collected data to determine whether this technology is viable for use in a gyroscopic sensing application.

1.3) Scope

This study analyses the specific nano-particle sensor technology introduced in a single previous study. This research is focused on developing the theoretical model of this sensor technology to model the response as a 3D vector, rather than just a magnitude scalar.

1.4) Assumptions

The assumptions made for this research are largely based on the specific sensor technology used. Assumptions are made about the overall sensor size, geometry, and composition, as well as the nano-particle properties of magnetization and spin. These assumptions are further explained in section 2.1.

1.5) Research Question

The sensor technology studied has been previously proven to exhibit measurable electrical current (on the order of milliamps) corresponding to the magnitude of angular displacement. The research question here is whether or not these measurements can be expanded, by modeling as a vector,

to determine information about the direction of angular displacement, rather than just the magnitude, and whether or not this technology is viable for inertial sensing applications.

1.6) Significance

This research will improve the understanding of this novel inertial sensing technology, and help to determine whether or not it is viable for real-world applications.

1.7) Definitions

A relationship exists for this novel sensing technology [4] to calculate the magnitude of the magnetic field response to angular displacement, provided in equations 1 and 2.

$$\boldsymbol{B}_{\text{precessed}} = (\boldsymbol{I\Omega\omega} - 6\boldsymbol{\eta V\omega}) / M(\sin(\theta_1)) = (2mr^2\boldsymbol{\omega\Omega} / 5 - 6\boldsymbol{\eta V\omega}) / M(\sin(\theta_1)) \quad \textbf{\textit{(eqn 1)}} \textbf{ [4]}$$

$$\boldsymbol{B}_{\text{not\_precessed}} = (\boldsymbol{I\alpha}_a - 6\boldsymbol{\eta V\omega}) / M(\sin(\theta_2)) = (2mr^2\boldsymbol{\alpha}_a / 5 - 6\boldsymbol{\eta V\omega}) / M(\sin(\theta_2)) \quad \textbf{\textit{(eqn 2)}} \textbf{ [4]}$$

This relationship can also be expressed in terms of the physical torque experienced by the particles as provided in equations 3 and 4.

$$\boldsymbol{\tau}_{\text{precessed}} = \boldsymbol{\Omega I\omega} - 6\boldsymbol{\eta V\omega} = \boldsymbol{MB}\sin(\theta_1) \quad \textbf{\textit{(eqn 3)}} \textbf{ [4]}$$

$$\boldsymbol{\tau}_{\text{not\_precessed}} = \boldsymbol{I\alpha}_a - 6\boldsymbol{\eta V\omega} = \boldsymbol{MB}\sin(\theta_2) \quad \textbf{\textit{(eqn 4)}} \textbf{ [4]}$$

In these relationships, “precessed” and “not precessed” can be defined as follows:

Precessed: When the nanoparticles are spinning (momentarily) about an axis that is offset from the axis by which the supplied spinning force is applied. This occurs

when the sensor is tilted and before the nanoparticles shift to realign their rotational axis.

Not Precessed: When the nanoparticles are spinning about the same axis that the supplied spinning force is applied. This represents the steady-state of the nano-particles when spinning within the sensor.

Due to the superparamagnetic properties of the magnetite nanoparticles, the magnetic moment of the nanoparticles, $\overline{m}$, only exists in the presence of a magnetic field. It is important to note the difference here between the magnetic moment, $\overline{m}$, of the nanoparticles and the magnetization, M. The magnetic moment, $\overline{m}$, can be expressed as provided in equation 5.

$$\overline{m} = \int\int\int M dV \qquad \textbf{(eqn 5)}$$

Because we are describing the orientation of a rigid body (the sensor) with respect to a fixed coordinate system, it is also helpful to use Euler Angles in order to decompose the angular components of the system into the fixed coordinate system. Figure 2 shows the geometrical definition of the euler angles that will be used. The Euler angles help to describe the system for more complex cases when the sensor is tilted about multiple spatial axes simultaneously (or concurrently).

Because the particles in the sensor are only nanometers in diameter, scaling laws must be considered. Specifically, the scaling laws for moment of inertia of a sphere, since the nanoparticles are spherical. In order to assess these scaling laws, the moment of inertia for a sphere is provided in equation 6.

$$I_{sphere} = \frac{2}{5} m r^2 \qquad \textbf{(eqn 6)}$$

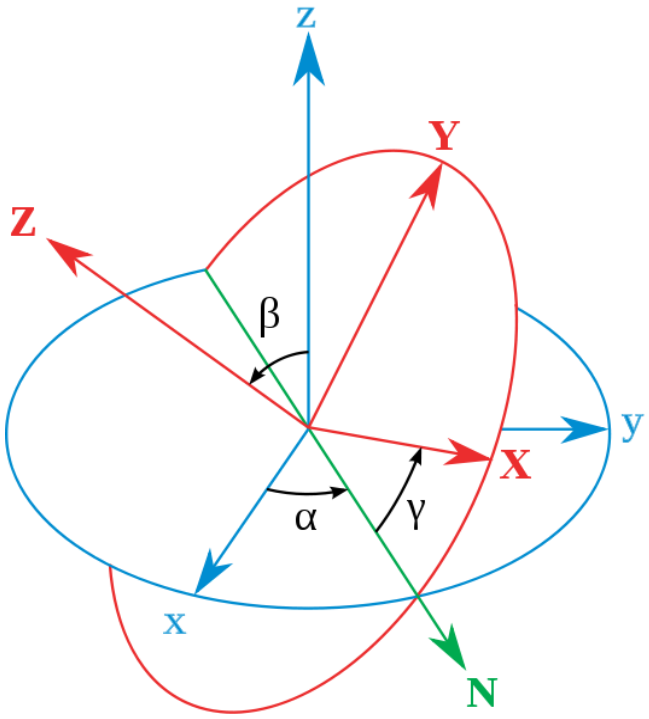

Figure 2: Euler Angles [5]

These Euler angles are applied only for the case where the sensor is tilted, and the nanoparticle spin axis is momentarily misaligned with the applied rotating magnetic field axis. Looking at figure 2, the blue components represent the fixed reference frame (the sensor and coil windings) and the red components represent displaced components (the nanoparticles when precessed) following 3 consecutive rotations about each of the 3 fixed axes. For the specific Euler angle representation used here, a 3-1-3, or Z-X-Z, rotation sequence is applied. This is because the nanoparticle is always rotating about the fixed Z axis, until it is precessed, at which point it experiences an angular displacement (which can be on either the X or Y axis). $\boldsymbol{\alpha}$ represents the first rotation about the fixed Z axis, $\boldsymbol{\beta}$ represents the second rotation about the fixed X axis, and $\boldsymbol{\gamma}$ represents the third rotation again about the fixed Z axis. $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ can take any angle value from 0° to 360°, while $\boldsymbol{\beta}$ is limited between -90° and 90°, or 0° and 180°.

Due to the symmetry of the sensor and nanoparticles, the Z-X-Z rotation convention can also be applied to rotations about the Y axis. Using this convention, the angular velocity vector of the rotation sequence can be decomposed into a 3D vector in the fixed coordinate system. This angular velocity vector is provided in equation 8.

$$\overline{\omega_{euler}} = \; < \frac{d\alpha}{dt} sin(\beta)sin(\gamma) + \frac{d\beta}{dt} cos(\gamma), \; \frac{d\alpha}{dt} sin(\beta)cos(\gamma) - \frac{d\beta}{dt} sin(\gamma), \; \frac{d\alpha}{dt} cos(\beta) + \frac{d\gamma}{dt} > \quad \textbf{(eqn 7) [11]}$$

In order to determine the magnetic field applied to rotate the nanoparticles, coming from the driving sensor coils, the Biot-Savart law is applied. The Biot-Savart law states that the magnetic intensity at any point due to a steady current in an infinitely long straight wire is directly proportional to the current and inversely proportional to the distance from point to wire [6]. This law is provided in equations 8 and 9 [7].

$$\overline{dB} = \frac{\mu_0 I \, \overline{dL} \times \hat{r}}{4\pi \bar{r}^2} \quad \textbf{(eqn 8)}$$

$$\overline{B} = \frac{\mu_0}{4\pi} \int_C \frac{I \, \overline{dL} \times \hat{r}}{\bar{r}^2} \quad \textbf{(eqn 9)}$$

The sensor technology used in this paper relies on a ferrofluid of Fe_3O_4 nanoparticles suspended in water. The relative magnetic permeability of water (μ_r) is known to be very close to 1, which means that it is appropriate to just use the magnetic permeability of free space (μ_0). However, because the ferrofluid contains Fe_3O_4 nanoparticles as well, that must be considered when assessing the magnetic permeability coefficient. This is discussed further in chapter 2.

In order to evaluate the relative permeability of a ferrofluid, it is important to relate the magnetic permeability to the magnetic susceptibility. This relationship is provided in equations 38 and 39.

$$\mu_r = \frac{\mu}{\mu_0} \quad \therefore \ \mu = \mu_r \mu_0 \qquad \textbf{(eqn 38) [18]}$$

$$\chi = \mu_r - 1 \qquad \textbf{(eqn 39) [18]}$$

In order to determine the induced current from the magnetic fields generated when the nanoparticles within the sensor are displaced, Faraday's law is applied. Faraday's law states that the electromotive force induced in a circuit by variation of the magnetic flux through the circuit is proportional to the negative of the time rate of change of the magnetic linkage [20]. This law is provided in equations 40 and 41 [21]. The equations provided by this law are for a generalized case, and must be evaluated for the specific geometry and conditions of the sensor technology. This is discussed in chapter 2.

$$V_{emf} = -N\frac{d\int \overline{B}d\overline{A}}{dt} \qquad \textbf{(eqn 40) [21]}$$

$$I_{induced} = \frac{V_{emf}}{R_{coil}} = -\frac{N}{R_{coil}}\frac{d\int \overline{B}d\overline{A}}{dt} \qquad \textbf{(eqn 41)}$$

The Biot-Savart law can be applied to this technology because an alternating current is being supplied through the driving coils, resulting in a changing magnetic field being generated and interacting with the magnetic nanoparticles. Faraday's law can be applied because moving (precession) nanoparticles generate a changing magnetic field that results in current being induced back onto the driving coils.

Chapter 2: Manuscript

2.1) Theoretical Modeling

Most modern inertial measurement units (IMUs) rely on accelerometers and gyroscopes to determine linear and rotational acceleration in spatial directions [8]. The novel technology used for this thesis relies on induced magnetic fields from rotating nanoparticles to achieve measurements.

The specific design of the sensing unit in question relies on a rectangular volume of magnetic nanoparticles suspended in a liquid. This volume then has coils of low impedance motor wire wrapped around it, similar to solenoid coils, in two orthogonal directions. In order to maintain a consistent frame of reference, the two spatial axes which the coils are wrapped around can be considered to be the x and y axes. Because the coils are wrapped around the entire volume containing the nanoparticles, the generated magnetic field within the coils (where the nanoparticles are present) can be held constant. A sinusoidal AC signal is supplied to each of the two driving coils, with each signal being of the same amplitude but with different phase. Specifically, the amplitudes are offset by a phase of 90 degrees, which allows the resultant force to spin the nanoparticles about the z axis.

Specifically, this relationship describes spinning magnetite (Fe_3O_4) nanoparticles and their resultant torque when tilted off of their axis of spin. This relationship includes effects from the magnetic forces from the applied spin field as well as the frictional drag forces from the liquid suspension agent (H_2O). Some assumptions made in equations 1-4 are that: (i) the magnetite nanoparticles are spherical; (ii) the nanoparticles are spinning about a stationary axis (not wobbling); (iii) the rate at which the entire sensor body tilts ($\boldsymbol{\Omega}$) is much smaller than the rate at which the nanoparticles are spinning ($\boldsymbol{\omega}$); (iv) the magnitude of the magnetic moment of the magnetite nanoparticles (M) is approximately 0.025×10^{-6} Am^2 [9]; (v) when returning to the steady state

position after being momentarily precessed, the particles reorient their axes in a linear motion; (vi) the dynamic or absolute viscosity is used for the suspension agent (H_2O); (vii) the magnetic permeability of free space (μ_0) can be used when evaluating the sensor technology; and (viii) the magnetic field produced by rotating nanoparticles is geometrically uniform.

These relationships presented in equations 1-4 are useful for determining the magnitude of magnetic flux that is generated when the sensor is displaced (tilted). When current is supplied to the driving coils of the sensor, the nanoparticles will start to spin because of their susceptibility to the applied magnetic field. Once these particles are spinning they will reach a steady state which is represented by the "not precessed" relationships. Then, if the sensor (and thus the nanoparticles) is tilted, the nanoparticles will temporarily be misaligned and will quickly move back into alignment with the axis of spin due to precession. The torque required to move these particles back into alignment is captured by the "precessed" equations. Thus, the difference between the precessed and not precessed relationships can be used to determine the angular displacement information.

When the sensor is displaced, and the nanoparticles are momentarily misaligned on their axis of rotation, which results in a reduction of nanoparticle spin torque relative to the original spin axis. This reduction in torque manifests as a reduction in the magnetic field magnitude, which will result in a drop in the induced current in each of the coil wrappings, which can then be measured. A previous study has shown that for a driving current on the order of a hundred milliamps, the measured induced current drop is on the order of a few milliamps [4].

Before going further into analyzing the electromagnetic response of this sensor, we must take a look at the existing relationship from equations 1-4. It can be seen that both the magnetic field and torque equations for the precessed state contain the same terms: $\boldsymbol{\Omega}\, \mathrm{I}\, \boldsymbol{\omega} - 6\eta \mathrm{V} \boldsymbol{\omega}$. For the not precessed

state, a similar pair of terms is also present in both sets of equations: $\boldsymbol{I\alpha}$ - $6\boldsymbol{\eta V\omega}$. In both the precessed and not precessed states, these two terms represent effects due to inertial precession and viscous fluid drag, respectively. Because the particles used in this sensor are only nanometers in diameter, scaling laws must be considered. At the nanoscale, inertia and gravity make virtually no difference, but viscous fluid forces are still significant [10]. This can be verified by deriving the scaling law for the moment of inertia of a sphere. By replacing the dimensional components of the moment of inertia equation, equation 6, with the characteristic dimensions, D, this scaling law is derived and provided in equation 10.

$$m = \varrho V$$
$$V_{sphere} = \tfrac{1}{6}\pi d^3$$
$$\therefore m = \varrho\tfrac{1}{6}\pi d^3 \sim D^3$$

$$r = \tfrac{d}{2} \sim D$$

$$I_{sphere} = \tfrac{2}{5}\,\varrho\,\tfrac{1}{6}\,\pi D^3\,\tfrac{1}{4}\,D^2 = \varrho\pi\tfrac{1}{60}D^5 \sim D^5 \qquad \textbf{(eqn 10)}$$

It can be seen that the moment of inertia of a sphere is proportional to D^5, which means that the moment of inertia for a spherical object with a diameter of 10nm would be approximately 10^{30} times smaller than an object with a moment of inertia of 1cm.

Computing the torque components due to inertial precession and viscous fluid drag for a typical precessed sensor condition, found in table 1, using equation 1 yields the results of: $\boldsymbol{\Omega}$ I $\boldsymbol{\omega}$ = 3.995 x 10^{-32} N*m and $6\boldsymbol{\eta V\omega}$ = 4.544 x 10^{-22} N*m. It can be seen that the viscous fluid drag effects are 10 orders of magnitude greater than the effects due to inertial precession. This is largely due to the fact that the moment of inertia of the nanoparticle is proportional to the particle volume and the particle

radius squared, whereas the fluid drag is only proportional to the particle volume. This confirms the result of the scaling laws

Table 1: Typical Sensor Values

Item	Symbol	Value
nanoparticle mass	m	7.065 x 10^{-20} kg
nanoparticle radius	r	1.5 x 10^{-8} m
spin velocity	$\boldsymbol{\omega}$	6283 rad/s
tilt velocity	$\boldsymbol{\Omega}$	1 rad/s
fluid viscosity (water @300K)	$\boldsymbol{\eta}$	8.53 x 10^{-4} N*s / m^2
nanoparticle volume	V	1.413 x 10^{-23} m^3
driving current phase difference	$\theta 1$	90 degrees
magnetization of nanoparticle	$\boldsymbol{M}$	2.5 x 10^{-8} A*m^2

This drastically changes the model, because the inertial precession components of equations 1-4 are the only components that relate angular displacement ($\boldsymbol{\Omega}$) to the magnetic field. Given that the inertial contributions of the model are insignificant compared to the fluid drag effects, the previous equations for the magnetic field and torque can be re-written, as provided in equations 11-14.

$$\boldsymbol{B}_{\text{precessed}} = (- 6\boldsymbol{\eta V\omega}) \,/\, M(\sin(\theta_1)) = (- 6\boldsymbol{\eta V\omega}) \,/\, M(\sin(\theta_1)) \qquad \textit{(eqn 11)}$$

$$\boldsymbol{B}_{\text{not\_precessed}} = (- 6\boldsymbol{\eta V\omega}) \,/\, M(\sin(\theta_2)) = (- 6\boldsymbol{\eta V\omega}) \,/\, M(\sin(\theta_2)) \qquad \textit{(eqn 12)}$$

$$\boldsymbol{\tau}_{\text{precessed}} = - 6\boldsymbol{\eta V\omega} = \boldsymbol{M B}\sin(\theta_1) \qquad \textit{(eqn 13)}$$

$$\boldsymbol{\tau}_{\text{not\_precessed}} = - 6\boldsymbol{\eta V\omega} = \boldsymbol{M B}\sin(\theta_2) \qquad \textit{(eqn 14)}$$

At first, it appears as though the equations for the precessed and not precessed states are now equivalent to each other, which would indicate that there is no difference in the magnetic field

generated when the particles are precessed or not. However, it is important to note that these equations represent individual nanoparticles within the sensor volume. Therefore, when the sensor is tilted, the nanoparticles will momentarily be rotating on a different axis, which means that the angular velocity will decompose differently. By decomposing the angular velocity vector for the precessed case, we can establish a relationship between the applied angle of tilt and the magnetic field response.

As previously stated, in order to examine spatial components of this sensor, it is important to establish a frame of reference. Figure 3 provides a simplified model of the sensor with labeled axes. It is important to note that the entire sensor is not rotating about the Z-axis, but the applied rotating magnetic field generated from the driving coils is always rotating about this axis. The frame of reference is established such that the sensor body, and the driving coil wrappings, are always considered to be the fixed reference. Using this frame of reference, it can be seen that rotating the sensor about the Z-axis will have no effect on the precession of the nanoparticles, because the Z-axis is the axis of rotation for the nanoparticles. Further, it can be observed that due to the symmetry of the sensor (and uniform distribution of spherical nanoparticles within the sensor volume), that rotating about either the X-axis or Y-axis should have the same effect on precession of the nanoparticles. This can be observed in figures 4 and 5. Rotating about either the X or Y axes will cause a momentary displacement between the spin axis of the nanoparticles and the axis of rotation of the applied magnetic field.

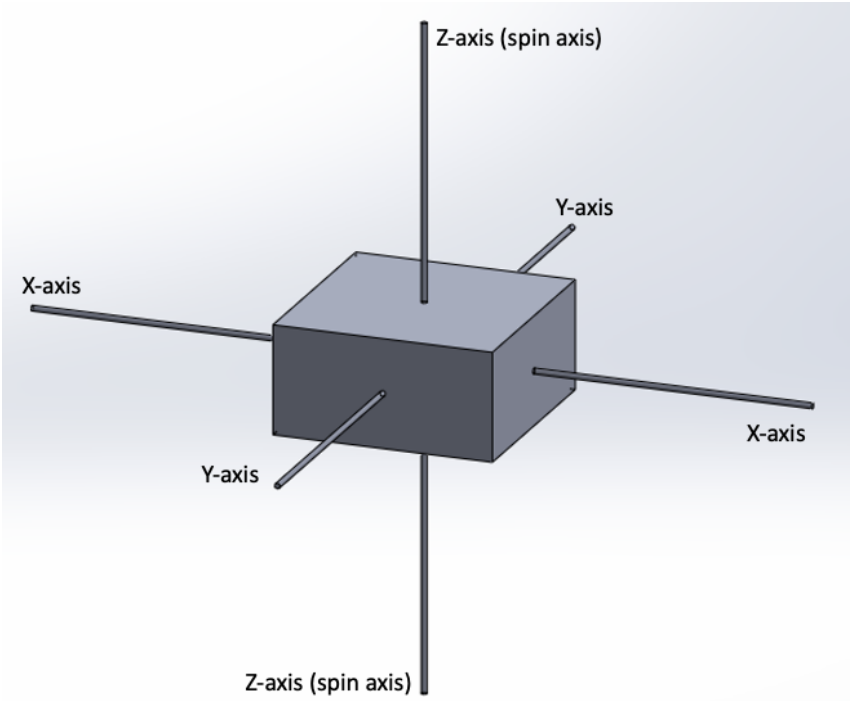

Figure 3: Simplified Sensor Model Frame of Reference

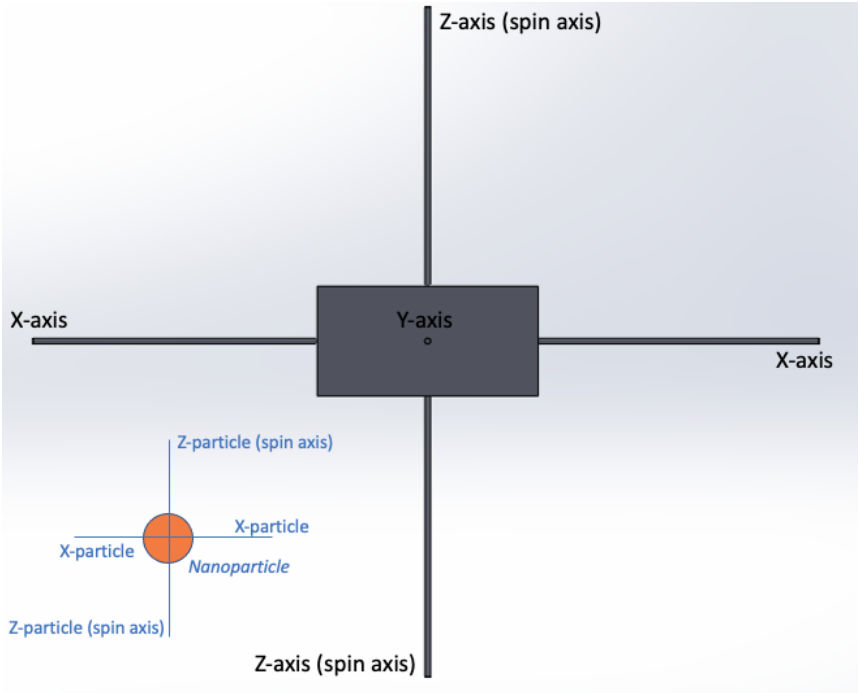

Figure 4: Simplified Sensor Model Not Displaced

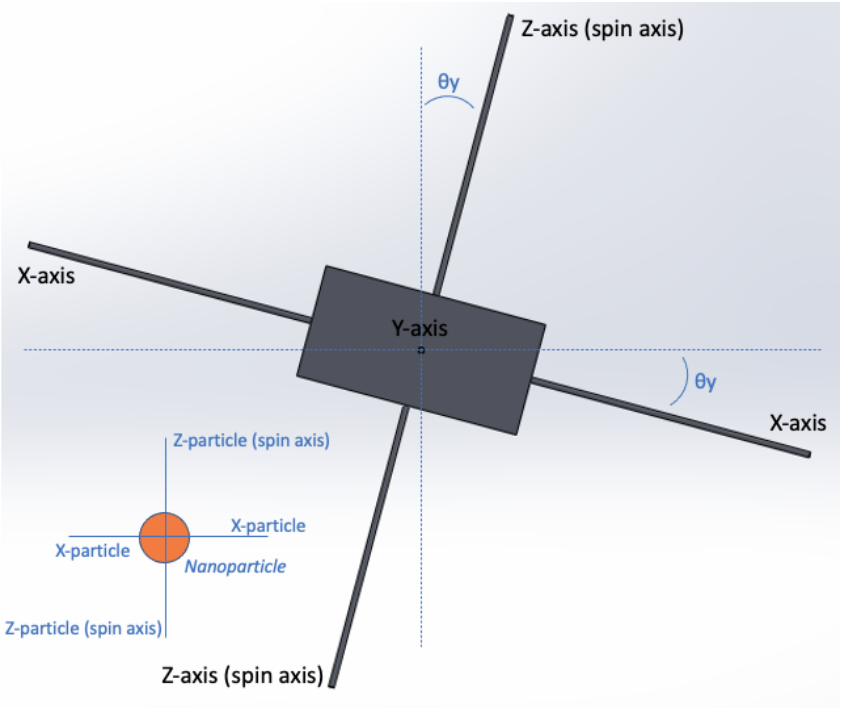

Figure 5: Simplified Sensor Model Displaced

The angular velocity of the displaced nanoparticles can be decomposed, using the Euler angular velocity vector from equation 7, as provided in equation 16. Equation 15 provides the angular velocity of the nanoparticles when the axes of rotation are aligned.

$$\overline{\omega_{not\_precessed}} = \ < 0,\ 0,\ 2\pi f > \qquad \textit{(eqn 15)}$$

$$\overline{\omega_{precessed}} = \ < \frac{d\alpha}{dt} sin(\beta)sin(\gamma) \ + \ \frac{d\beta}{dt} cos(\gamma),\ \frac{d\alpha}{dt} sin(\beta)cos(\gamma) \ - \ \frac{d\beta}{dt} sin(\gamma),\ \frac{d\alpha}{dt} cos(\beta) + \ \frac{d\gamma}{dt} > \qquad \textit{(eqn 16)}$$

Because of the assumed spherical geometry of the nanoparticles, we can see that regardless of the nanoparticle orientation, the magnetic moment magnitude should remain constant. This allows us

to adjust equations 13 and 14 to those provided in equations 17 and 18 which include the vector components of the B field as well. It is noticed that the equations are now identical.

$$\boldsymbol{\tau}_{\text{precessed}} = -6\eta V\overline{\omega} = M < Bx,\ By,\ Bz > \qquad \textit{(eqn 17)}$$

$$\boldsymbol{\tau}_{\text{not\_precessed}} = -6\eta V\overline{\omega} = M < Bx,\ By,\ Bz > \qquad \textit{(eqn 18)}$$

Because the novel sensing unit relies on wire wrappings, similar to solenoid coils, the Biot-Savart law can be evaluated accordingly to determine the applied magnetic field from the driving coils to spin the nanoparticles. Because solenoid coils have a uniform cylindrical geometry, the curve integral can be evaluated to obtain the closed form solution provided in equation 19.

$$\overline{B} = \frac{\mu_0 NI}{L} \qquad \textit{(eqn 19)}\ \textbf{[12]}$$

Looking at equation 19, it can be seen that the magnetic permeability coefficient of free space (μ_0) is being used, per the initial assumptions made. Because of the presence of Fe_3O_4 particles suspended in the water, however, the magnetic susceptibility changes. This is because permeability (μ) in ferrofluids is a complex quantity dependent upon the frequency of the supplied magnetic field [18]. However, because the supplied signal frequencies used in this technology are relatively low (1 kHz), these effects are not significant. A study on the magnetic susceptibility of Fe_3O_4 nanoparticles found that for frequency ranges from 10 Hz to 10 kHz, the magnetic susceptibility was approximately the same as a function of AC field frequency, and almost exactly the same when at room temperature (which is the case for this technology)[19]. Given the operating conditions of the sensor, specifically relatively low AC signal frequency and constant room temperature, a static magnetic permeability coefficient can be reasonably assumed because the magnetic susceptibility is not a function of applied

magnetic field frequency. If this technology were to be employed with higher signal frequencies, especially in the MHz and GHz range, then this assumption would need to be reevaluated.

Given that the magnetic permeability can be considered to be constant, equations 38 and 39 can be applied to determine the approximate value of the magnetic permeability coefficient. This is done using the empirically derived values for magnetization of Fe_3O_4 nanoparticles [19].

$$\chi = \mu_r - 1 = 0.0096 \quad \therefore \ \mu_r = 1.0096$$

$$\mu = \mu_r \mu_0 = 1.0096 \ \mu_0$$

It can be seen that the relative magnetic permeability of the ferrite nanoparticles can be seen to be very close to 1. This, combined with the fact that the relative permeability of water is also very close to (but lower than) 1, indicates that it is appropriate to continue to equate the magnetic permeability of the ferrofluid to the magnetic permeability of free space.

The Biot-Savart law can be applied to determine the magnetic field at any given point within the sensor (where the nanoparticles are) from the driving current through the coils. This magnetic field contribution can then be combined with the contribution from the precession effects in order to determine the total resultant field.

A critical aspect of this theory is to represent the induced magnetic field from angular precession of the sensor as a vector, rather than a scalar. The current scalar based equations only allow for information to be garnered regarding the magnitude of angular displacement, not the direction. Additional information provided from a full vector model will allow for this direction to be determined. This allows for the sensor measurements from each independent driving coil to be used to provide additional information about the direction of rotation. Because the inertial contribution to

the precession torque equation was proven to be insignificant, however, this becomes less straightforward.

First, the relationship for the non precessed state of the nanoparticle sensor can be established as a vector. By combining equations 15 and 18 we can see the torque that is required to overcome fluid drag in order to spin the nanoparticles at the steady state velocity. This equation is provided in equation 20.

$$\boldsymbol{\tau}_{\text{not\_precessed}} = -6\eta V\langle 0,\ 0,\ 2\pi f \rangle = M\langle Bx,\ By,\ Bz \rangle \qquad \textbf{(eqn 20)}$$

Decomposing equation 20 into its components in the fixed frame of reference is done next, and the results are provided in equations 21, 22, and 23.

$$\boldsymbol{\tau}_{\text{not\_precessed\_X}} = 0 = MBx \qquad \textbf{(eqn 21)}$$

$$\boldsymbol{\tau}_{\text{not\_precessed\_Y}} = 0 = MBy \qquad \textbf{(eqn 22)}$$

$$\boldsymbol{\tau}_{\text{not\_precessed\_Z}} = -6\eta V 2\pi f = MBz \qquad \textbf{(eqn 23)}$$

Looking at equations 21, 22, and 23, we can see that in the steady non precessed state, when the nanoparticles are spinning, there is only a magnetic field in the Z direction, or in the direction of the spin axis.

Using the frame of reference from figure 3, we can apply Lenz's law to determine which components of the magnetic field will result in induced current on the driving coils. Since the driving coils are wrapped around the X and Y axes, that means that the magnetic field components in the X direction will contribute to the current in one coil, and the magnetic field components in the Y

direction will contribute to the current in the other coil. We can refer to the coils wrapped on their respective axes as CoilX and CoilY.

In order to determine the induced current from the rotating nanoparticle magnetic field, Faraday's law must first be evaluated for the sensor geometry. Looking at equation 41 it can be seen that finding a closed form solution of Faraday's law requires integrating with respect to the magnetic field as well as the cross sectional area of the wire loop (coil) exposed to the magnetic field. For the not precessed case, the strength of the magnetic field, as well as the area, should remain constant, which allows the equation to be simplified to the one provided in equation 42. This solution relies on the assumption that the magnetic field produced from rotating nanoparticles is fairly uniform, which can be assumed due to the fact that the wire coils are wrapped closely to the fluid.

$$I_{induced} = \frac{V_{emf}}{R_{coil}} = -\frac{N}{R_{coil}} \frac{d\int \bar{B}d\bar{A}}{dt} = -\frac{N}{R_{coil}} \frac{BA}{\Delta t} \qquad \textbf{(eqn 42)}$$

Because this induced magnetic field is solely in the Z direction, Lenz's law can be applied to find that the magnetic flux should not result in an induced current on the driving coils. Even though the steady state induced current will be 0, it is still useful to quantify current that could be induced if another coil was present. This current can be quantified by applying Faraday's law by combining equation 23 with equation 42, into the resultant equation 25.

$$\frac{-6\eta V 2\pi f}{M} = Bz = -\frac{I_{induced} R_{emf} \Delta t}{NA} \qquad \textbf{(eqn 24)}$$

$$\frac{6\eta V 2\pi f NA}{M \Delta t} = I_{not\ precessed} \qquad \textbf{(eqn 25)}$$

For a typical supply signal of 1000 Hz, and a typical sensor with 50 windings of a coil with an equivalent solenoid radius of 0.5cm, the induced current in the non precessed state would be on the

order of femtoamps per nanoparticle (71.4 fA). For a typical sensor, with water as the suspension agent, there are approximately 10^{16} nanoparticles present within the sensor volume, which means that the total current value can be increased. Due to magnetic susceptibility, interaction between particles, and thermal effects, however, this gain is not exactly equal to the number of particles. A previous study found measurable values to be on the order of milliamps for an equivalent sensor [4]. Regardless, as previously stated, the steady state non precessed magnetic field produced from nanoparticle precession will not be induced onto the driving coil currents.

In order to relate the magnetic field to an angle of tilt, we will need to assess the sensor response when the nanoparticles are precessed. First, we can combine the decomposed angular velocity vector from equation 16 with the precessed torque equation from equation 17. This combination is provided in equation 26.

$$\boldsymbol{\tau}_{\text{precessed}} = -6\eta V < \frac{d\alpha}{dt} sin(\beta)sin(\gamma) + \frac{d\beta}{dt} cos(\gamma),\ \frac{d\alpha}{dt} sin(\beta)cos(\gamma) - \frac{d\beta}{dt} sin(\gamma),\ \frac{d\alpha}{dt} cos(\beta) + \frac{d\gamma}{dt} > = M < Bx,\ By,\ Bz > \quad \textbf{(eqn 26)}$$

Decomposing equation 26 into its components in the fixed frame of reference is done next, and the results are provided in equations 27, 28, and 29.

$$\boldsymbol{\tau}_{\text{precessed\_X}} = -6\eta V \left(\frac{d\alpha}{dt} sin(\beta)sin(\gamma) + \frac{d\beta}{dt} cos(\gamma)\right) = MBx \quad \textbf{(eqn 27)}$$

$$\boldsymbol{\tau}_{\text{precessed\_Y}} = -6\eta V \left(\frac{d\alpha}{dt} sin(\beta)cos(\gamma) - \frac{d\beta}{dt} sin(\gamma)\right) = MBy \quad \textbf{(eqn 28)}$$

$$\boldsymbol{\tau}_{\text{precessed\_Z}} = -6\eta V \left(\frac{d\alpha}{dt} cos(\beta) + \frac{d\gamma}{dt}\right) = MBz \quad \textbf{(eqn 29)}$$

Looking at equations 27, 28, and 29, we can see that when the nanoparticle is precessed, the angular velocity will have components in all three of the fixed reference frame directions, when tilted

about a single axis. These components have magnitudes that directly relate to the angle and rate of tilt. Because this induced magnetic field is in the X, Y, and Z directions, Lenz's law can be applied to find that the magnetic field will result in an induced current on the driving coils.

In order to determine the induced current from the rotating nanoparticle magnetic field, Faraday's law must again be evaluated for the sensor geometry. Looking at equation 41 it can be seen that finding a closed form solution of Faraday's law requires integrating with respect to the magnetic field as well as the cross sectional area of the wire loop (coil) exposed to the magnetic field. For the precessed case, the strength of the magnetic field, as well as the area, will not remain constant. This makes it more difficult to evaluate the integral. In order to arrive at the solution for a changing magnetic field and a changing area, two solutions can be combined. Specifically, solving the equation for the case of a fixed coil (static area) with a changing magnetic field, and the equation for the case of a rotating coil (changing area) in a uniform magnetic field. These solutions are presented and combined in equations 43-45. These solutions again rely on the assumption that the magnetic field produced from rotating nanoparticles is fairly uniform.

$$I_{fixedA} = \frac{V_{emf}}{R_{coil}} = -\frac{N}{R_{coil}} \frac{d\int \overline{B} d\overline{A}}{dt} = -\frac{N}{R_{coil}} A \frac{\Delta B}{\Delta t} \qquad \textbf{(eqn 43) [21]}$$

$$I_{fixedB} = \frac{V_{emf}}{R_{coil}} = -\frac{N}{R_{coil}} \frac{d\int \overline{B} d\overline{A}}{dt} = -\frac{N}{R_{coil}} B \frac{\Delta A}{\Delta t} \qquad \textbf{(eqn 44) [21]}$$

$$I_{induced} = \frac{V_{emf}}{R_{coil}} = -\frac{N}{R_{coil}} \frac{d\int \overline{B} d\overline{A}}{dt} = -\frac{N}{R_{coil}} \frac{\Delta A \, \Delta B}{\Delta t} \qquad \textbf{(eqn 45)}$$

This current can be quantified by applying Faraday's law by combining equations 27, 28, and 29 with equation 45, into the resultant equations 33, 34, and 35. In order to determine the differential

B field element, ΔB , we can observe the difference between the precessed and not processed state. For now can consider the differential area of the coil (ΔA) to be the entire area, however, it is possible (and likely) that this area is less, which would be equivalent to scaling the component. This could be further determined through empirical verification.

$$((-6\eta V(\frac{d\alpha}{dt}sin(\beta)sin(\gamma) + \frac{d\beta}{dt}cos(\gamma)))/M) - 0 = \Delta Bx = -\frac{I_{induced}\,\Delta t\,R_{coil}}{N\,\Delta A} \quad \textbf{(eqn 30)}$$

$$((-6\eta V(\frac{d\alpha}{dt}sin(\beta)cos(\gamma) - \frac{d\beta}{dt}sin(\gamma)))/M) - 0 = \Delta By = -\frac{I_{induced}\,\Delta t\,R_{coil}}{N\,\Delta A} \quad \textbf{(eqn 31)}$$

$$((-6\eta V(\frac{d\alpha}{dt}cos(\beta) + \frac{d\gamma}{dt}))/M) - (-6\,\eta\,V2\pi f) = \Delta Bz = -\frac{I_{induced}\,\Delta t\,R_{coil}}{N\,\Delta A} \quad \textbf{(eqn 32)}$$

$$((6\eta V\ N\Delta A(\frac{d\alpha}{dt}sin(\beta)sin(\gamma) + \frac{d\beta}{dt}cos(\gamma)))/(M\ \Delta t\,R_{coil})) = I_{precessed_X} \quad \textbf{(eqn 33)}$$

$$((6\eta V\ N\Delta A(\frac{d\alpha}{dt}sin(\beta)cos(\gamma) - \frac{d\beta}{dt}sin(\gamma)))/(M\ \Delta t\,R_{coil})) = I_{precessed_Y} \quad \textbf{(eqn 34)}$$

$$((6\eta V\ N\Delta A(\frac{d\alpha}{dt}cos(\beta) + \frac{d\gamma}{dt}))/(M\ \Delta t\,R_{coil})) + (6\,\eta\ V2\pi f\,N\Delta A/(\Delta t\,R_{coil})) = I_{precessed_Z} \quad \textbf{(eqn 35)}$$

Looking at equations 33, 34, and 35, we can see that there will be an induced current on the driving current coils wrapped around the X and Y axes. This induced current will be at its maximum value when the sensor has just stopped tilting, and then it will return to 0 as the rational axes of the nanoparticles realign and the induced current shifts back to the Z axis. This is because the induced current on the driving coils is only due to the X and Y components of the magnetic field, which are only present when the nanoparticles are precessed. Therefore, $I_{x\ precessed}$ and $I_{y\ precessed}$ represent the maximum values of the induced current contribution from each nanoparticle, and $I_{z\ precessed}$

represents the minimum value of the potential current contribution that is at its maximum at the steady non precessed state.

For a typical supply signal of 1000 Hz, and a typical sensor with 50 windings of a coil with an equivalent solenoid radius of 0.5cm, the induced current in the precessed state can be computed for a handful of angular displacements. A simplified example can be used, where tilt is only applied about a single axis, as illustrated in figures 4 and 5. This can be modeled using the Euler angle representation from equations 33-35. Using the Z-X-Z Euler convention, we know the angular velocity about the fixed Z axis, which is represented by $\frac{d\alpha}{dt}$ and $\frac{d\gamma}{dt}$. Tilting the sensor by an angle $\boldsymbol{\beta}$ and at a rate of $\frac{d\beta}{dt}$ will produce an induced current value using equations 33-35. Induced current for a handful of these tilt cases was computed for a variety of angles and tabulated in table 2. Because of the assumption that the tilt rate, $\frac{d\beta}{dt}$, will be much smaller than the spin rate, $\frac{d\alpha}{dt}$ and $\frac{d\gamma}{dt}$, we can determine the amount of Z axis displacement, $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, using the formula in equations 36 and 37. The spin rate is fixed at $2\pi f = 6283$ rad/s.

$$\alpha = ((\frac{d\alpha}{dt}/\frac{d\beta}{dt})\,\beta)\ \%\ 360° \qquad \textit{(eqn 36)}$$

$$\gamma = ((\frac{d\gamma}{dt}/\frac{d\beta}{dt})\,\beta)\ \%\ 360° \qquad \textit{(eqn 37)}$$

Table 2: Induced Current Results (Tilted About X Axis)

Angular Displacement, $\boldsymbol{\beta}$, degrees	**Tilt Rate, $\frac{d\beta}{dt}$, rad/s**	**Z Twist Angle, $\boldsymbol{\alpha} = \boldsymbol{\gamma}$, degrees**	**Maximum $I_{x\ precessed}$, A**	**Maximum $I_{y\ precessed}$, A**	**Minimum $I_{z\ precessed}$, A**
30	349	180.1	-2.01E-15	-1.78E-14	6.66E-14
60	349	0.2	2.09E-15	3.09E-14	5.35E-14
90	349	180.3	-2.17E-15	-3.57E-14	3.57E-14
120	349	0.4	2.20E-15	3.09E-14	1.78E-14

The results from table 2 match the expected behavior of the system. As the sensor is tilted about it's X axis, the current $I_{y\ precessed}$ is approximately 10 times greater than the current $I_{x\ precessed}$. For a rotation about the Y axis, rather than the X axis, this means that the current $I_{x\ precessed}$ would be approximately 10 times greater than the current $I_{y\ precessed}$ as well. This means that the equations can be used to determine not just the magnitude of the angular displacement, but also which axis about which the system was tilted. For a combination of tilt about the X and Y axes, the currents $I_{x\ precessed}$ and $I_{y\ precessed}$ would be more similar to each other.

Further, it can be seen that the magnitude of the current contribution for an individual nanoparticle is on the order of femtoAmps again, which is consistent with the magnitude of current found for the not precessed state.

2.2) Simulation and Analysis

In order to verify that the theoretical model for the sensor is accurate, simulations were performed using MATLAB. The first simulated aspect of the sensor was the magnetic field from the driving coils to spin the nanoparticles. The representation of the Biot Savart law in equation 9 relies on differential magnetic field elements corresponding to differential length segments of a conducting wire. This was simulated in MATLAB by generating a series of points in a plane (to represent nanoparticles within the sensor), and then computing differential magnetic field contributions for each segment of the wire wrappings. The simulated wire coil vectors were plotted and are provided in figure 6, so that the simulation can be better visualized. It can be seen in figure 6 that there are 50 coil wrappings simulated for both the X and Y axes.

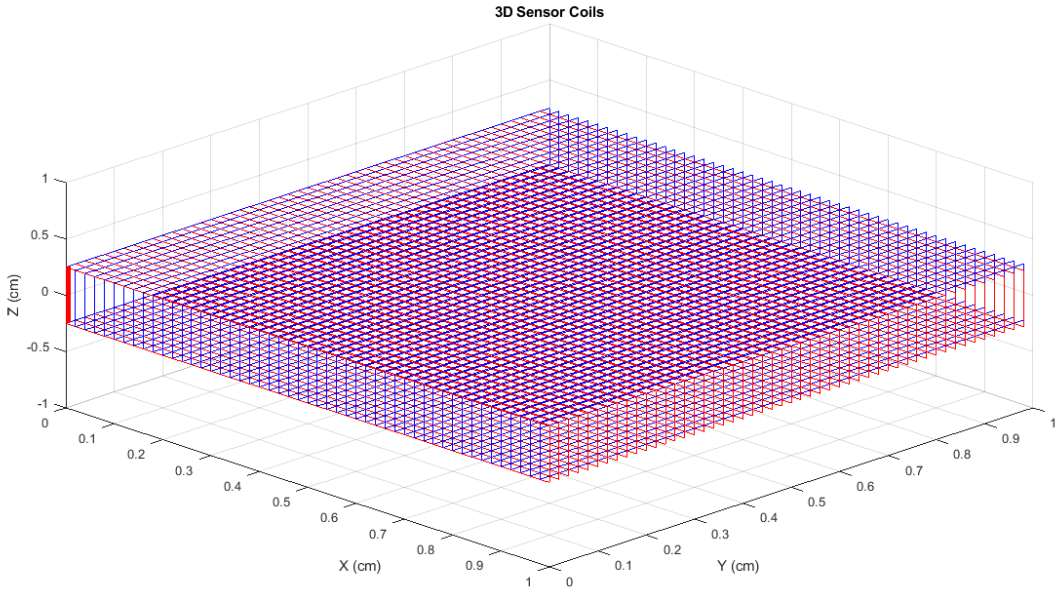

Figure 6: MATLAB Simulated Wire Coils Plot

The simulation of the Biot-Savart law was performed for a few different Z planes, representing different points within the sensor. Because the typical sensor dimensions are 1cm x 1cm x 0.5cm, the magnetic field strength was simulated for Z values of 0cm, 0.1cm, 0.2cm, and 0.2475cm above and below the center of the sensor volume. These simulation results are provided in figures 7-13.

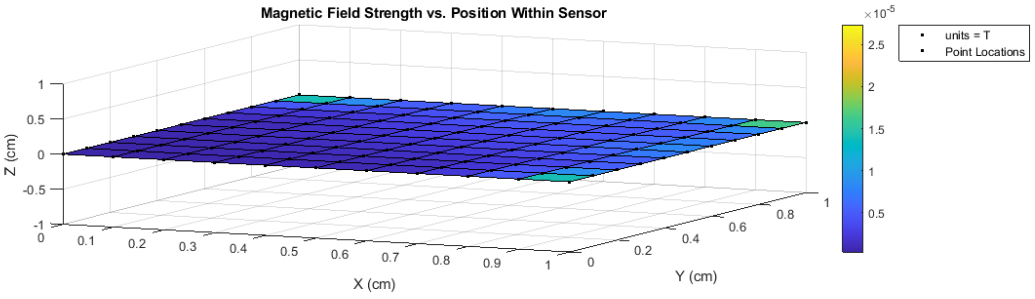

Figure 7: Applied Magnetic Field from Sensor Driving Coils, Z = 0 (center of sensor)

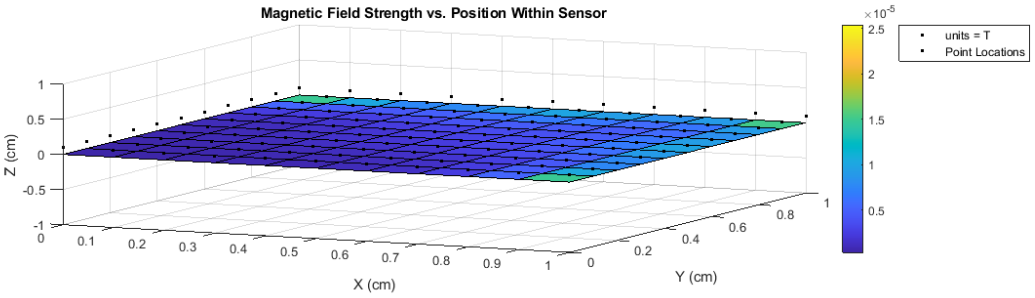

Figure 8: Applied Magnetic Field from Sensor Driving Coils, Z = 0.1cm

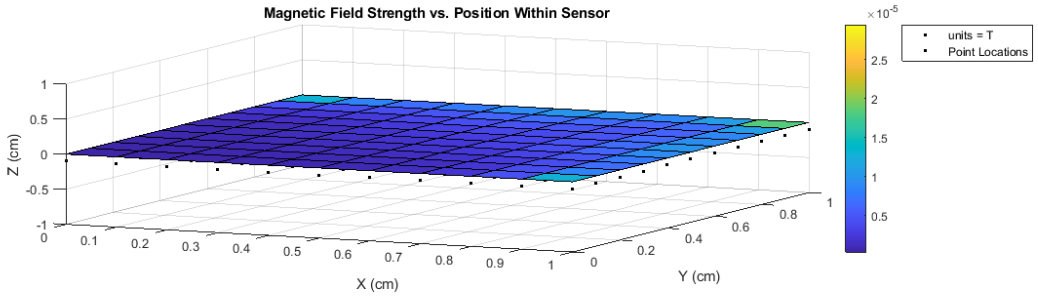

Figure 9: Applied Magnetic Field from Sensor Driving Coils, Z = -0.1cm

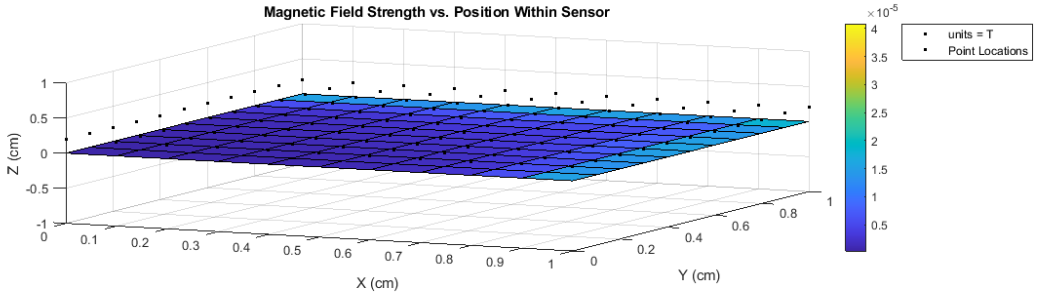

Figure 10: Applied Magnetic Field from Sensor Driving Coils, Z = 0.2cm

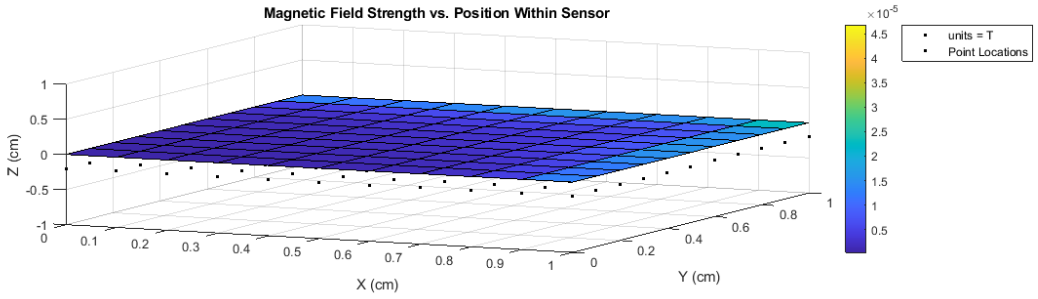

Figure 11: Applied Magnetic Field from Sensor Driving Coils, Z = -0.2cm

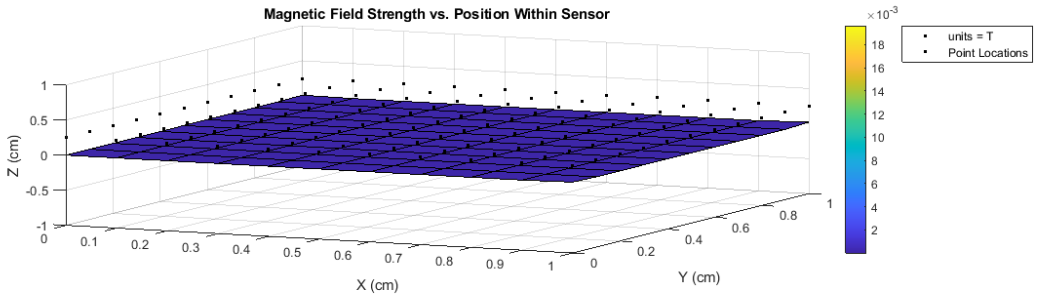

Figure 12: Applied Magnetic Field from Sensor Driving Coils, Z = 0.2475cm

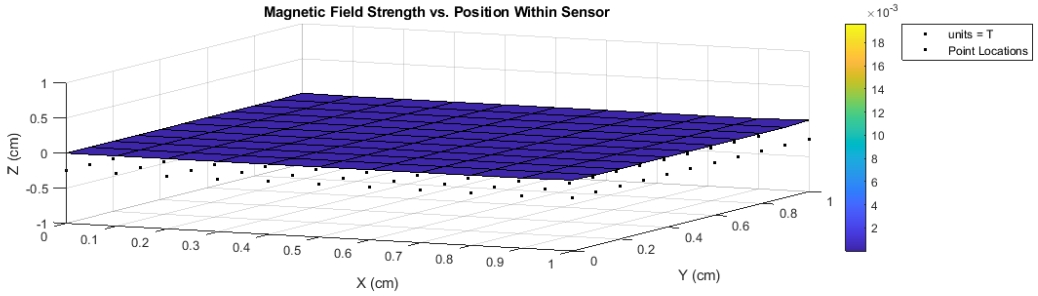

Figure 13: Applied Magnetic Field from Sensor Driving Coils, Z = -0.2475cm

Looking at figure 7 it can be seen that the magnetic field from the driving coils, intended to spin the nanoparticles, remains fairly uniform in the middle with slight variation at the corners. Variation at the corners of the sensor is expected, due to wire wrappings not reaching there, and is mitigated by the fact the nanoparticles are mostly located away from the very edges of the sensor volume.

From figures 8-13, it can be seen that the same pattern is exhibited throughout the volume of the sensor: the applied magnetic field is very uniform throughout, with the exception of the corners. Because the nanoparticles are contained within glass vials within the sensor, it is impossible for the particles to directly touch the wires because of the thickness of the vials. Therefore, it is reasonable to

assume that the applied magnetic field experienced by the nanoparticles within the sensor should be uniform. In order to magnify the patterns observed, to verify that the non-uniformities are negligible, the log (base 10) of the plots from figures 7-13 was also plotted. Figure 14 shows one of these plots (for Z=0.2cm).

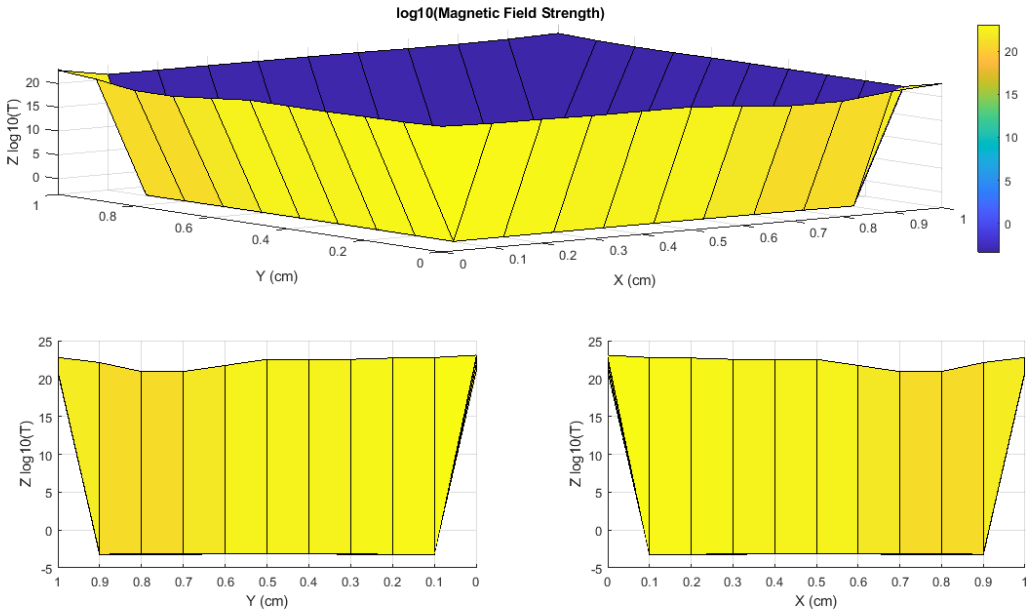

Figure 14: Logarithmic Applied Magnetic Field from Sensor Driving Coils, Z = 0.2cm

Looking at figure 14 it can be seen that the field is still very uniform in the middle with higher magnitudes at the edges and corners.

The simulations also provided a numerical output of the total magnetic field magnitude in the center of the sensor (X=0.5cm, Y=0.5cm) for a handful of different heights (Z values). These values are provided in table 3.

Table 3: Applied Magnetic Field Magnitude within Sensor Volume

X, cm	Y, cm	Z, cm	B Field Magnitude, T
0.5	0.5	0	2.8277E-06
0.5	0.5	0.1	2.7794E-06
0.5	0.5	-0.1	2.7794E-06
0.5	0.5	0.2	2.6295E-06
0.5	0.5	-0.2	2.6295E-06
0.5	0.5	0.225	2.5754E-06
0.5	0.5	-0.225	2.5754E-06
0.5	0.5	0.235	2.5519E-06
0.5	0.5	-0.235	2.5519E-06
0.5	0.5	0.24	2.5398E-06
0.5	0.5	-0.24	2.5398E-06
0.5	0.5	0.2475	2.5212E-06
0.5	0.5	-0.2475	2.5212E-06
0.5	0.5	0.25	2.5148E-06
0.5	0.5	-0.25	2.5148E-06

The values from table 3 were also plotted and fitted with a 2nd order polynomial curve. This plot is provided in figure 15. Looking at figure 15, it can be seen that the strongest magnetic field is present at the center of the sensor, while the weakest field is present at the very top and bottom (closest to the driving coil wires). There is very little change in the magnitude of the applied magnetic field throughout the Z axis of the sensor, as the difference between the smallest and largest magnetic field magnitudes is only 0.313μT (11%). This again validates the assumption of a uniform magnetic field being applied to all nanoparticles within the suspension volume.

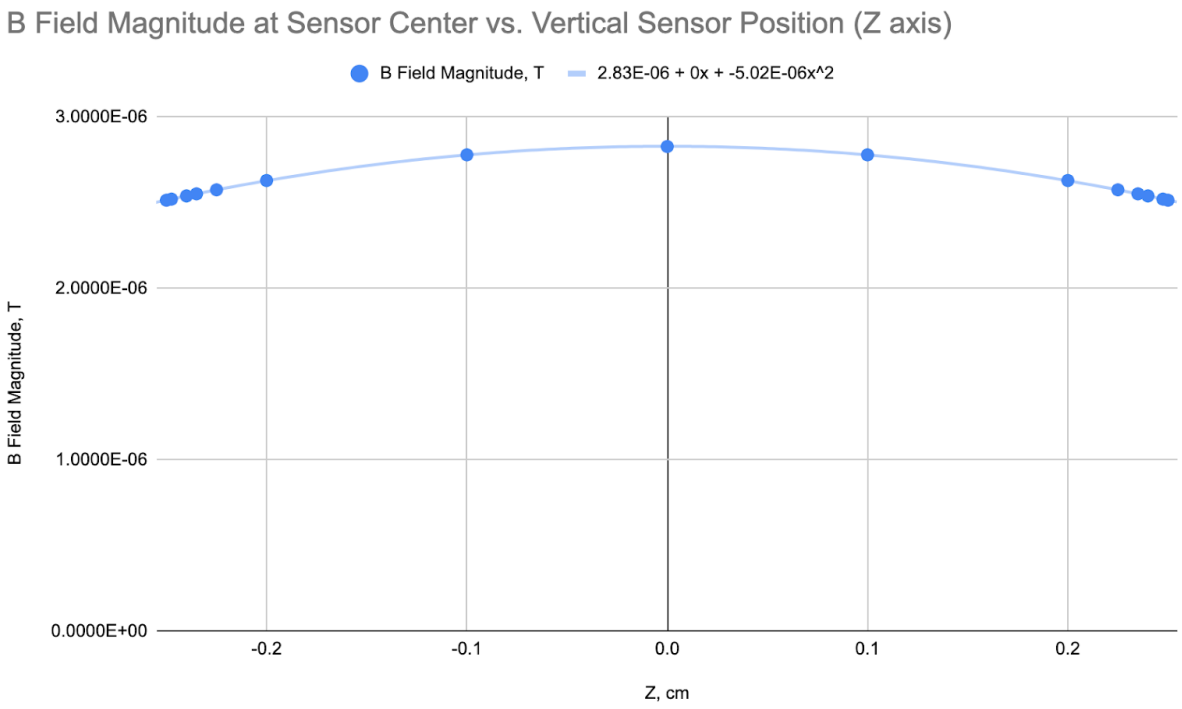

Figure 15: Applied Magnetic Field Magnitude within Sensor Volume

Simulation results were dependent on the assumptions that (i) the magnetite nanoparticles are spherical; (vi) the dynamic or absolute viscosity is used for the suspension agent (H_2O); and (vii) the magnetic permeability of free space (μ_0) can be used when evaluating the sensor technology.

Moving forward, more simulations could be done to assess the assumptions of uniform magnetic fields produced by spinning nanoparticles. Numerically solving the integral for Faraday's law would be one way to validate this assumption.

2.3) Experimental Results

The circuit schematic for the experimental nanoparticle sensor is provided in figure 16. The capacitor and inductor values were selected to obtain an overall phase shift as close as possible to 90 degrees. The driving coils for the nanoparticle sensor are modeled as an RL load, which was verified with an impedance measurement. Because of the capacitive and inductive elements, and the fact that the supplied current is sinusoidal, the impedance of the sensor coil lines changes as the frequency of the supplied signal changes. The two channel amplifier is required in order to ensure that the current passing through each of the two sensor wire coils is equal in amplitude, in order to ensure that the nanoparticles are spun evenly. Mismatched current amplitudes, or a phase angle of not 90 degrees results in an ellipsoidal or "wobbly" spin pattern, rather than the desired circular spin pattern.

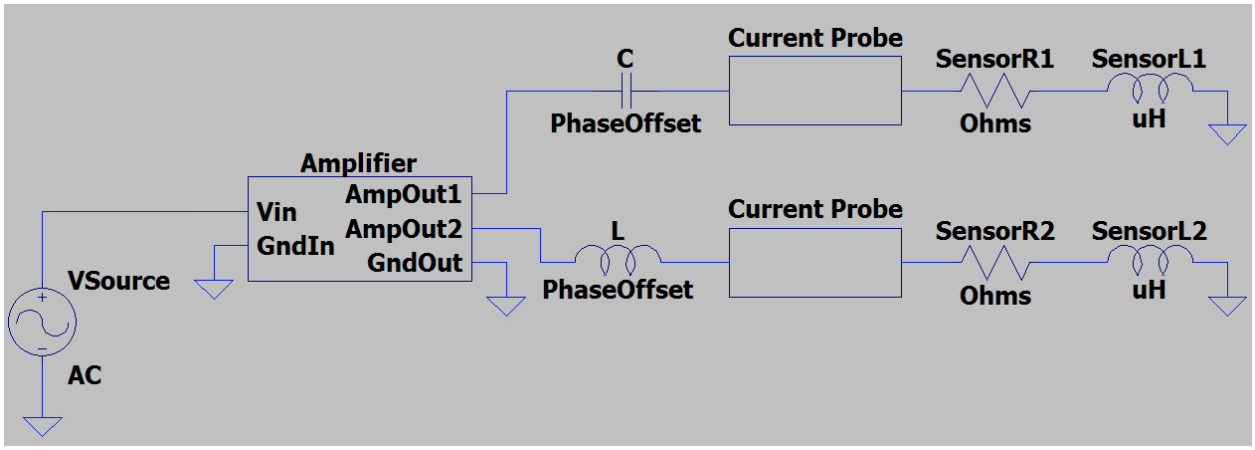

Figure 16: Magnetite Nanoparticle Gyroscopic Sensor Schematic

Previous studies with this novel sensing technology have employed similar measurement methods with similar circuit configurations [4]. Some notable improvements of the experimental testing from this research were that current sensors with higher resolution were employed and angular

displacement control has been automated with a very robust and precise motor (Aerotech ADRS100). As mentioned in the preface, the circumstances presented by the COVID-19 pandemic presented some challenges, most notably in limitations to laboratory access.

Numerous tests were carried out using the nanoparticle sensor, all measuring the overall current in each of the coils as a response to angular displacement. The sensor was connected to the control motor as pictured in figure 17.

Figure 17: Sensor Mounted to Angular Displacement Control Motor

The first results collected using the sensor appeared promising, displaying results that appeared to fit the theoretical model and were consistent with previous studies. Figures 18-20 shows the results of these experiments.

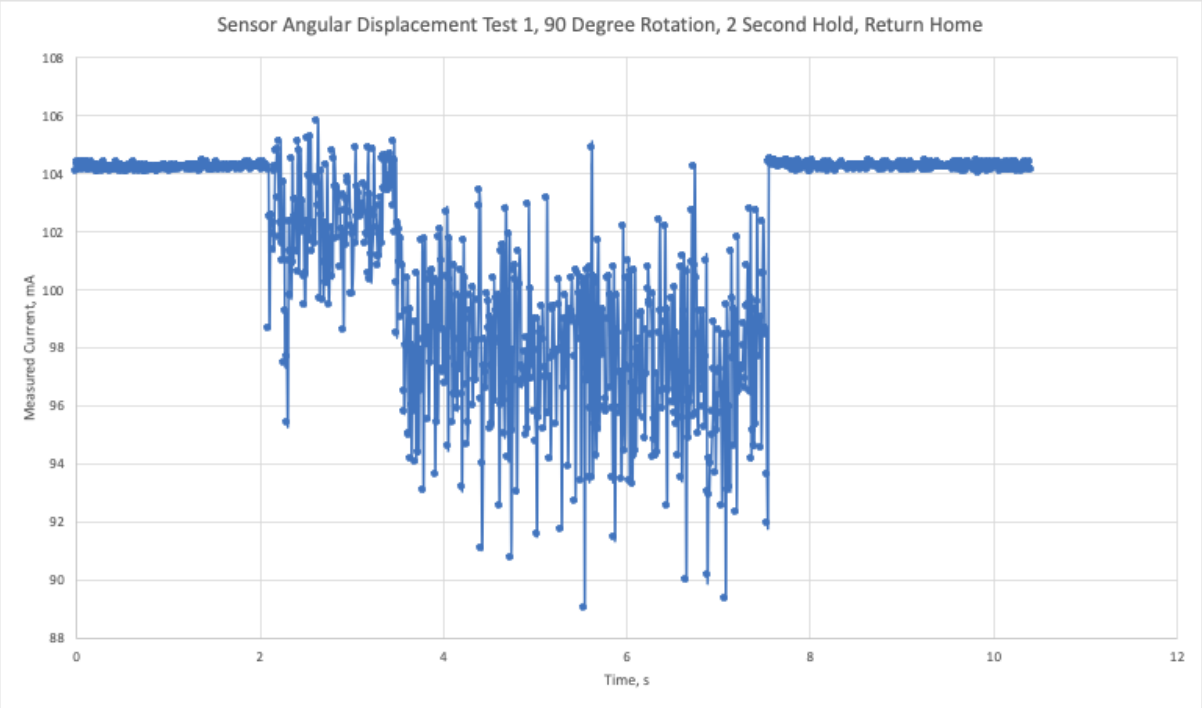

Figure 18: Initial Angular Displacement Experiment Results, Test 1

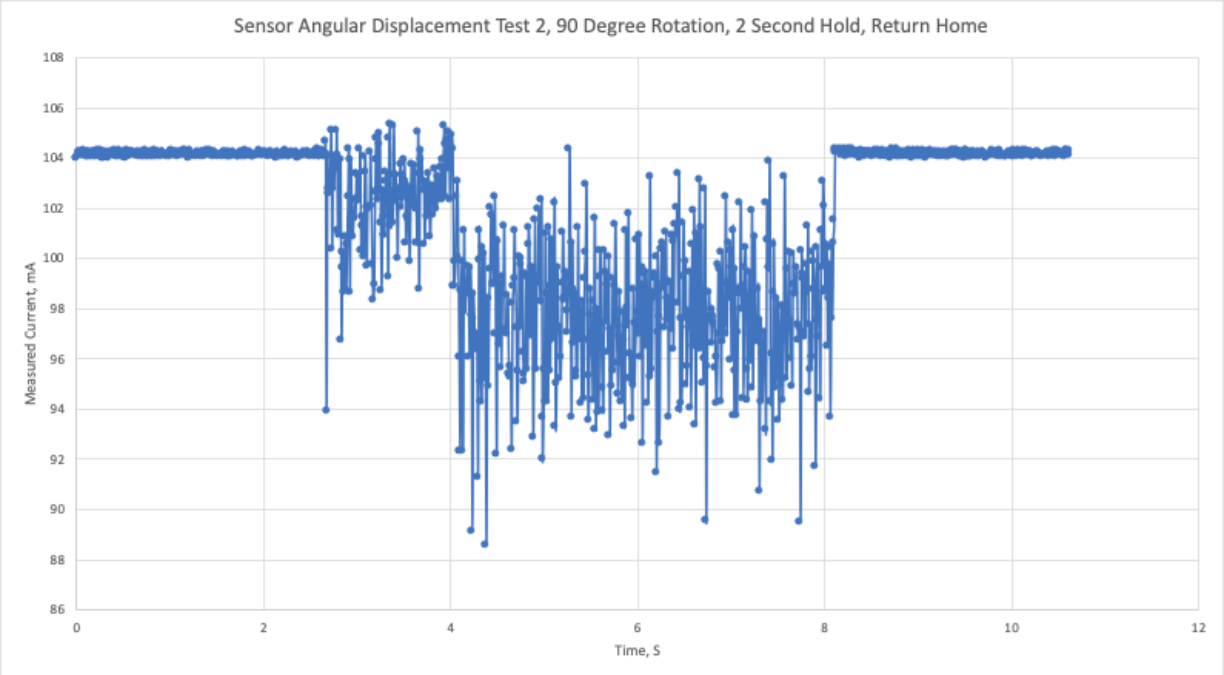

Figure 19: Initial Angular Displacement Experiment Results, Test 2

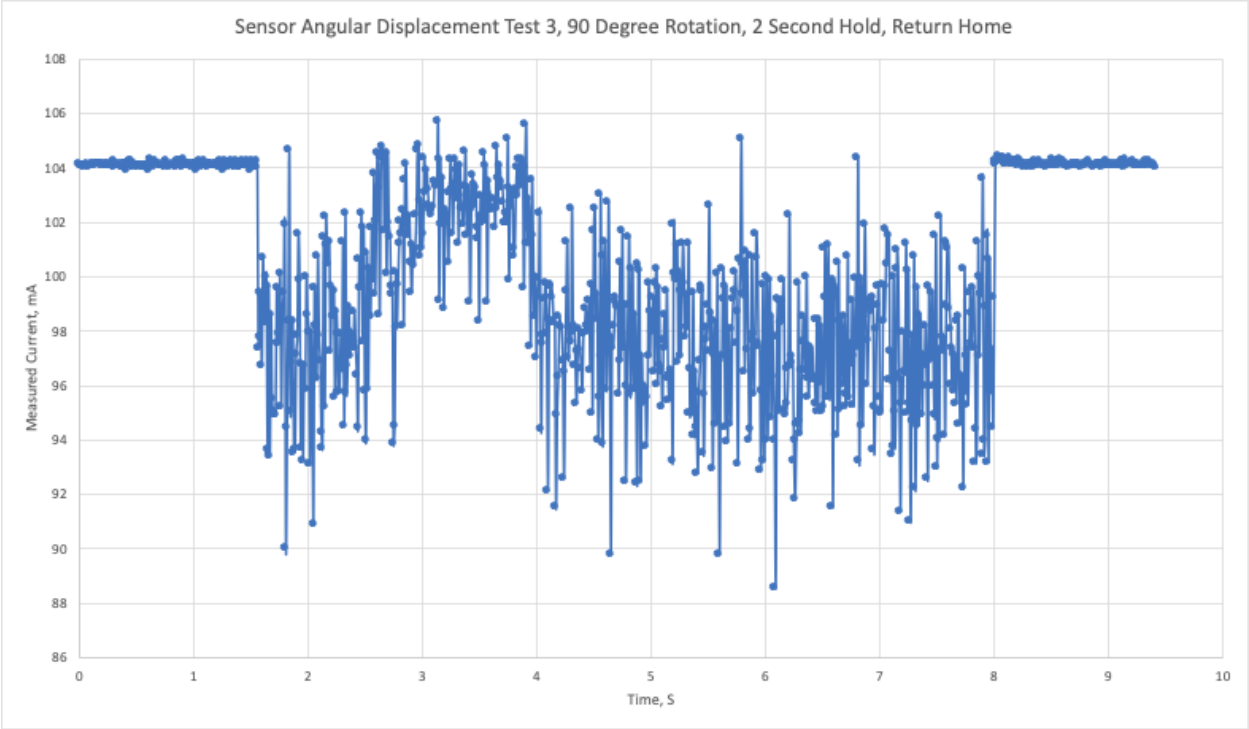

Figure 20: Initial Angular Displacement Experiment Results, Test 3

The initial experiments utilized the automated rotation table to tilt the sensor to a specified angle (90 degrees) at a rate of 20,000 deg/s (349 rad/s). Currents through the driving coils were measured using the current sensor and tabulated and plotted as a function of time.

It can be seen by looking at figures 18-20, that a measurable current spike on the order of milliamps was observed when the sensor was tilted, and then as the nanoparticles realigned their spin axis this current returned to steady state. However, after further testing and inspection of the measurement equipment, it was determined that these results were due to electromagnetic effects from the motor, not the sensor. The same test was performed again, however, the angular displacement was applied manually (tilting the sensor by hand), and the sensor response was not replicated. The results of this experiment are provided in figure 21.

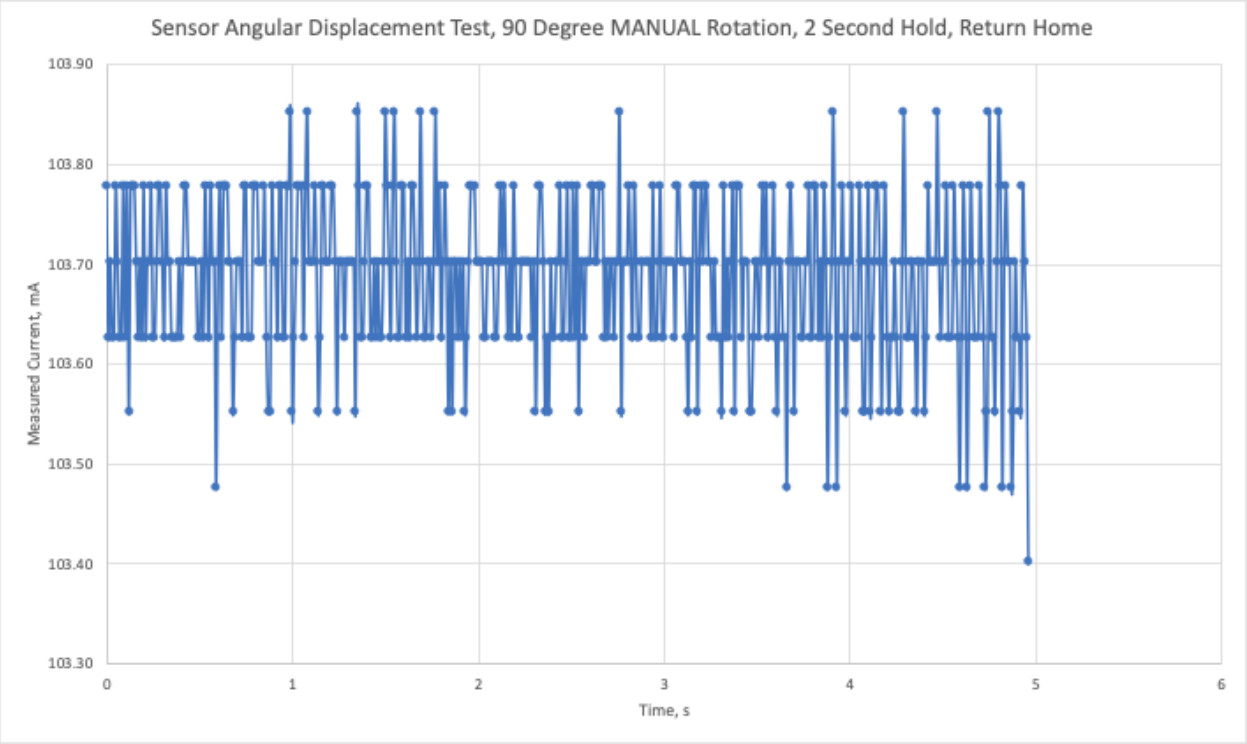

Figure 21: Initial Angular Displacement Experiment Results, Manual Displacement

Looking at figure 21, it can be seen that there was no measurable difference in the coil current when the sensor was manually displaced. The resolution of the current sensor in these experiments (figures 18-21) was 77μA / bit. All of the measured values were within a few bits of resolution from the median, which is not a significant amount, and does not exhibit the expected value of milliamps. It is possible that the manual angular displacement didn't yield the same results as the automated angular displacement because of the rate of angular displacement, or tilt speed. The automated motor displaced the sensor at a rate of 20,000 deg/s, and it is very likely that the rate at which the sensor was tilted by hand was significantly less than that. In order to ensure that the measurements from initial experimentation could not be attributed to nanoparticle precession in the sensor, another set of experiments was run.

The automated motor used to control the angular displacement of the sensor is very robust and has numerous features and capabilities. One of these features is that the rotational axis can be "enabled" or "disabled", which powers the motor coils without rotating it. This feature proved to be useful, because it allowed for power to be supplied to the motor without applying any angular displacement to the sensor. Experiments were run where the sensor current was measured as the motor axis was enabled and disabled, without tilting the sensor at all, to determine whether or not the same phenomena was observed. These experiments are provided in figures 22 and 23.

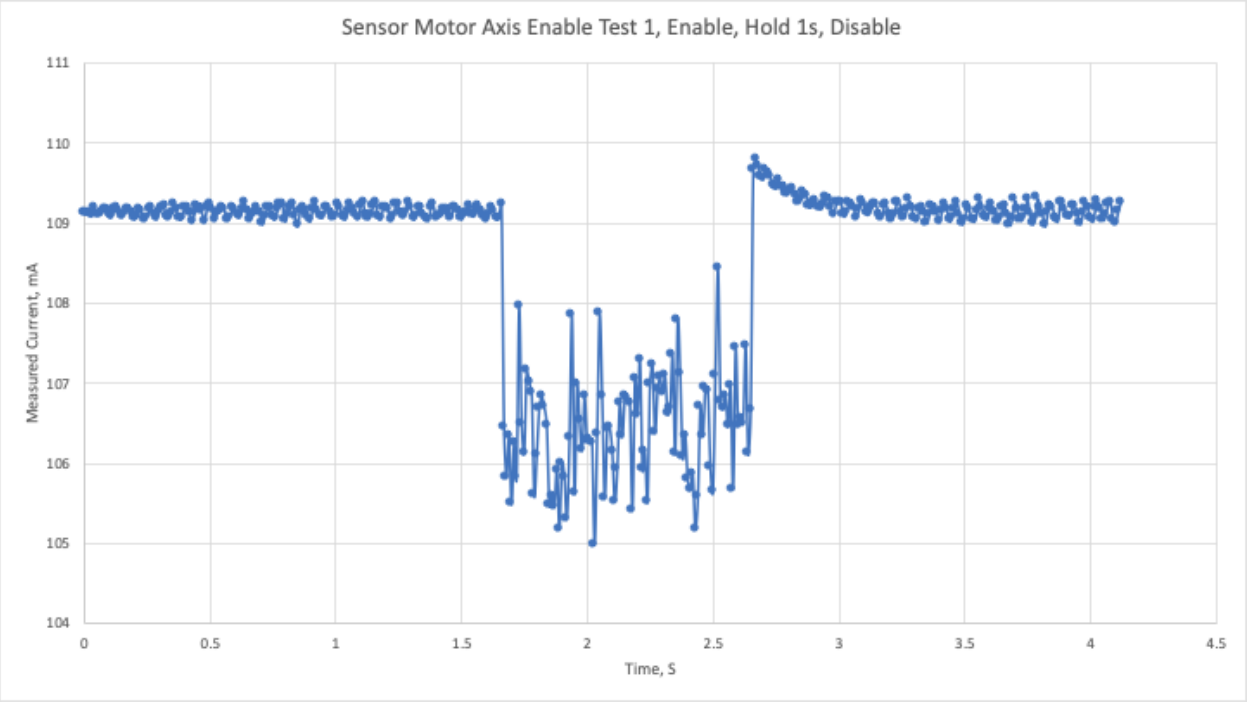

Figure 22: Motor Axis Enable Experiment Results

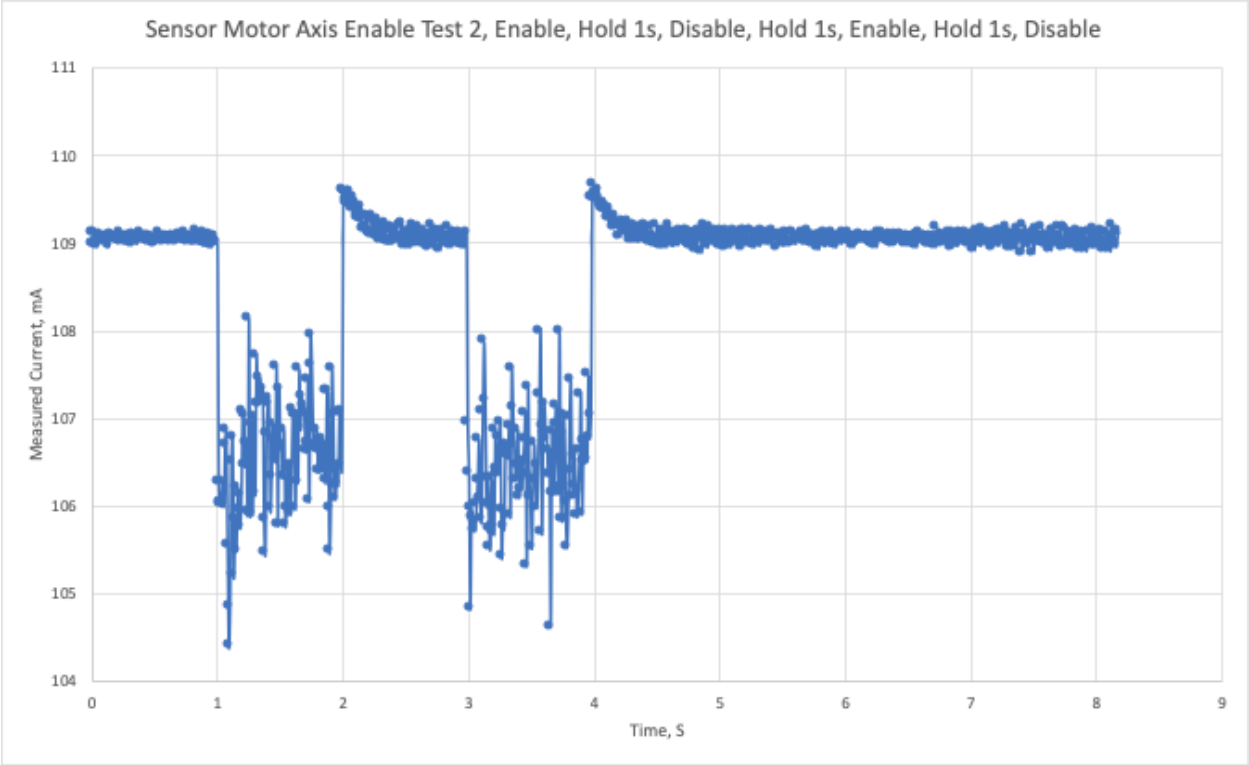

Figure 23: Motor Axis Enable Experiment Results

It can be seen from figures 22 and 23 that the motor was producing electromagnetic fields that were being coupled into the sensor coils. The observed phenomena that was initially thought to be due to nanoparticle precession, was actually just the motor.

Attempts were made to eliminate these effects from the motor to try and isolate the nanoparticle sensor to see if consistent results could be achieved. The sensor coils and driving circuitry were physically moved away from the motor and motor controller and additional ground connections were added in order to reduce loops in the circuit that might couple any electromagnetic fields from the motor. After making these adjustments, the initial experiment was repeated, and the results are provided in figures 24, 25, and 26.

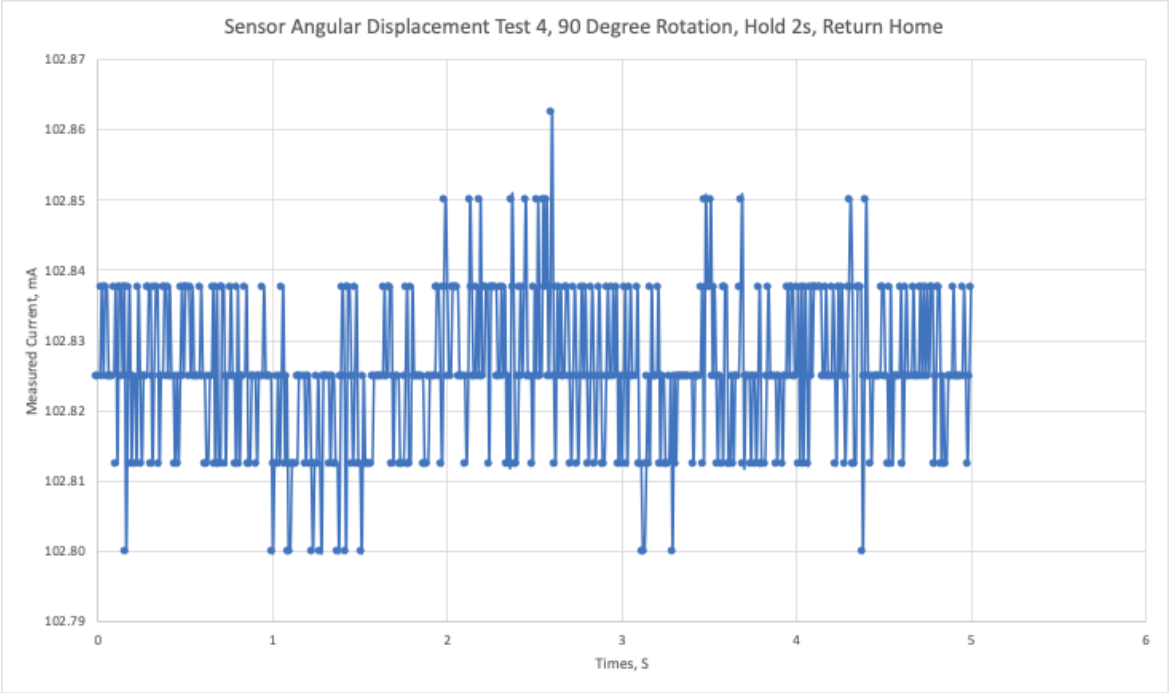

Figure 24: Angular Displacement Experiment Results, Test 4

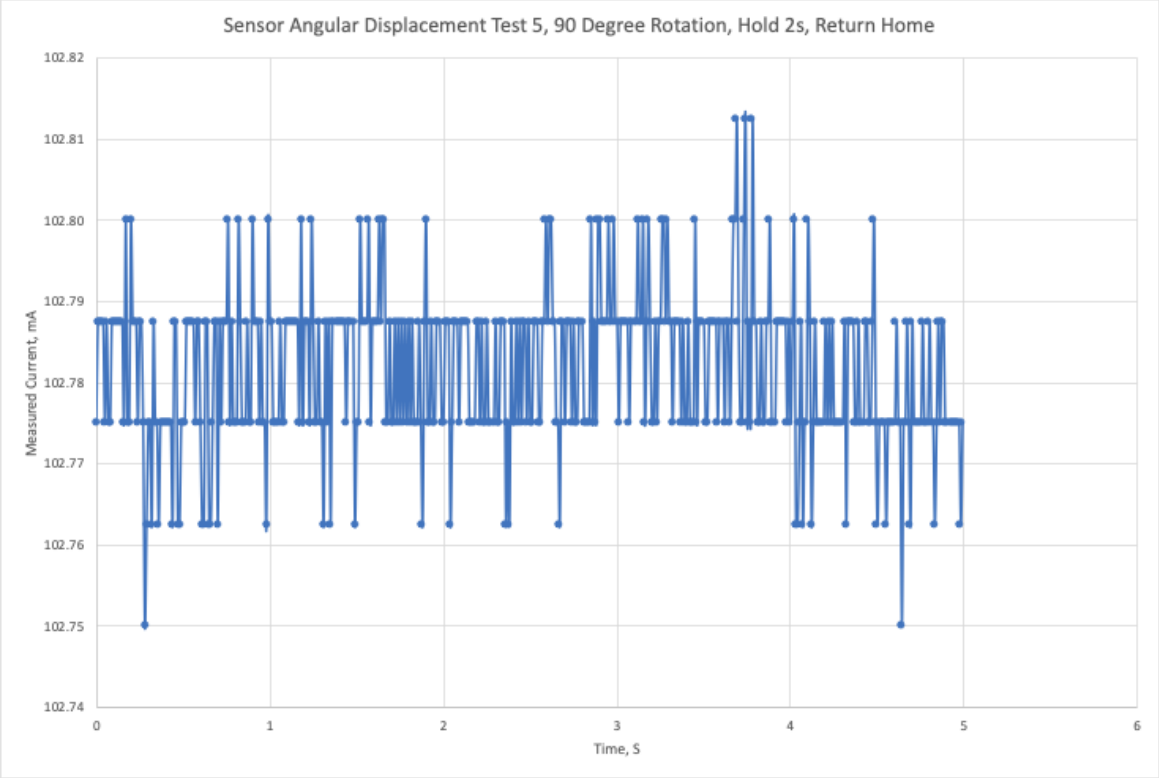

Figure 25: Angular Displacement Experiment Results, Test 5

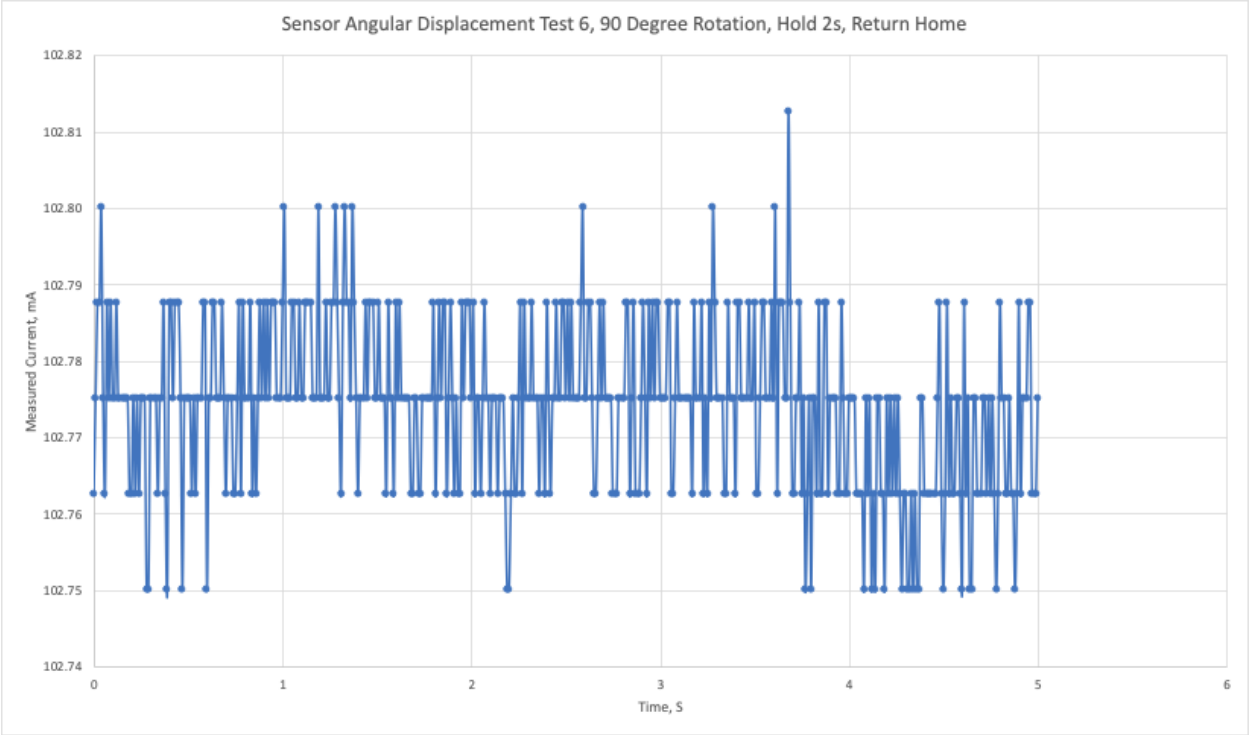

Figure 26: Angular Displacement Experiment Results, Test 6

It can be seen from looking at figures 24-26 that there is no measurable current when the sensor is tilted. The resolution of the current sensor for these experiments was reconfigured and set at 51.2μA / bit, and it can be seen again that all of the measured values were within a few bits of resolution from the median. Therefore, the induced currents from nanoparticle precession of the sensor were not measurable.

Theoretically, the sensor technology could work and is viable for further applications. However, improvements should be made to the sensor design in order to improve electromagnetic immunity to exterior sources. Adding a ferrous casing could be one way to improve upon this.

It is also possible that the effects of nanoparticle precession are too small to measure with the resolution of the current sensors used. Using current sensors with improved resolution (μA) could

resolve this problem. Because the geometry of the sensor involves large lengths of wire wrapped into loops, with long leads running back to the sensor, it is relatively easy to couple unwanted current into the system. Adding a ferrous casing to enclose the sensor would also be helpful to further shield the sensor from unwanted exterior magnetic fields.

Another item that could improve the empirical data collection system would be a custom printed circuit board (PCB) to manage all connections. Using a PCB could reduce the overall parasitic impedance and shorten the path for current to flow to ground. This could help to reduce the overall noise and make it easier to distinguish sensor measurements.

Further, magnetic susceptibility could be empirically measured for the specific ferrofluid vial in use, to verify the assumptions made that magnetic permeability of the sensor is equivalent to the magnetic permeability of free space.

2.4) Conclusions

It was determined that inertial components of nanoparticle torque are negligible due to scaling laws at the nanoscale. This means that a significant component of the previously proposed inertial nanoparticle sensor model does not actually contribute. However, viscous drag effects can still be seen to have a significant effect, which can still be used to derive information regarding orientation of the nanoparticle. Significant energy is still required to overcome the viscous fluid forces in order to realign rotational axes of precessed nanoparticles. Thus, the equations determined for the sensor can still be used to determine angular displacement of the sensor, but the primary contributor to this is the viscous fluid drag between the nanoparticles and the water, not the rotational inertia of the nanoparticles.

The equations determined for relating the angular displacement (both angle and rate) were expanded into a full vector representation that allows for components of the current to be separately determined. This allows for the determination of not just the magnitude of angular displacement, but also the direction (about which axis the sensor was tilted).

The simulations proved that the assumption of uniform magnetic fields being applied to the nanoparticles is valid. The nanoparticles within the sensor volume can be assumed to all be spinning in the same direction and at the same speed, regardless of their position within the sensor volume.

The experimental results were inconclusive, being unable to reproduce the magnitudes of response that were present in previous studies [4]. It was determined that unwanted coupling with the motor produced results similar to those expected, which could indicate that the other study (which used a similar motor) may have also coupled currents from the motor.

It should also be considered that there could be problems with the model presented. For one, the magnetic permeability of the ferrofluid is assumed to be static and equivalent to the permeability of free space. This should be verified with empirical data, and it is possible that a significantly different value could be obtained for the permeability coefficient. In this case, the result would scale the magnetic field applied to spin the nanoparticles, which could have an effect on the overall current measurement. It has been observed in other studies that ferrofluids using ferrite nanoparticles can exhibit magnetic permeability coefficients in the range of 1.5-5 [22], so it is possible that this could occur in this case. Another possible source of error in the model is that the magnetic field generated by the precessed rotating nanoparticles could be of a different form than the one assumed. The case of the sensor technology involves a changing magnetic field (due to nanoparticle precession) as well as a changing area (due to rotating nanoparticles). The solution employed simplified the integral solution by combining solutions for a rotating coil in a uniform magnetic field and a uniform coil in a changing magnetic field, however a different solution could be implemented.

Overall, the purpose of this research was to model the 3D torque of nano-particles in this sensor technology to determine the electromagnetic response to angular displacement. This was successfully accomplished as can be seen by the equations and results presented in section 2.1. These equations allow for the modeling of the torque of the nanoparticles as a 3D vector. Further, section 2.2 successfully showed the simulation of applied magnetic fields in the sensor technology, verifying the assumptions made of a uniform magnetic field. Lastly, the goal of comparing simulated and theorized results to empirically collected data was partially met, however, the empirical data collected was limited. The overall viability of the technology for use in a gyroscopic sensing application can be theoretically assessed, but further research will need to be done to assess the empirical response of the sensor.

Appendix A

MATLAB Simulation Code

Magnetic_Field_Model_Z_Planes.m

```
%%% Title: Magnetic Field Biot-Savart Law Model
%%% Author: Jackson Brennecke
%%% Date: September 15, 2020
%%% Master's Thesis Research
%%% Description: This code simulates the electromagnetic response
%%%              of some ferrite nano-particles in a sensor. The
%%%              sensor is modeled by the wire wrappings that provide
%%%              the driving current to spin the nanoparticles.
%%%              The electromagnetic B field is determined
%%%              according to the Biot-Savart law and then computed
%%%              for every point in the array. These points are then
%%%              plotted in surface response plots.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Start %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%clear variables
clearvars;

%Data Inputs
u0 = 4*pi*10^(-7); %Vacuum Permissivity
I = 0.1; %100mA Supplied Current Magnitude, Amps
vectLen = 1000; %Number of data points
wireLen = 0.01; %1cm Length of Wire, meters
numPoints = 10; %Number of Points in each direction
setZ = -0.0025; %Where to evaluate the nanoparticles
planeStart = [0,0,setZ]; %XY start point
planeLim = [0.01,0.01,setZ]; %XY end point
pointLoc = zeros(3,(numPoints+1)^2); %array to hold all points
num_wires = 50;

wire1_bottom_start = [0,0,-0.0025];
wire1_bottom_end = [0,0.01,-0.0025];

wire1_side1_start = [0,0.01,-0.0025];
wire1_side1_end = [0,0.01,0.0025];

wire1_top_start = [0,0.01,0.0025];
wire1_top_end = [0,0,0.0025];

wire1_side2_start = [0,0,0.0025];
wire1_side2_end = [0,0,-0.0025];

wire2_bottom_start = [0,0,-0.0025];
wire2_bottom_end = [0.01,0,-0.0025];

wire2_side1_start = [0.01,0,-0.0025];
wire2_side1_end = [0.01,0,0.0025];
```

```
wire2_top_start = [0.01,0,0.0025];
wire2_top_end = [0,0,0.0025];

wire2_side2_start = [0,0,0.0025];
wire2_side2_end = [0,0,-0.0025];

XY_num = 1000;
vectLen = XY_num; %Number of data points
Z_num = XY_num/2;
wire_iterator = planeLim(1)/num_wires;

wire1_bottom_vect = zeros(3,(XY_num + 1));
wire1_side1_vect = zeros(3,(Z_num + 1));
wire1_top_vect = zeros(3,(XY_num + 1));
wire1_side2_vect = zeros(3,(Z_num + 1));

wire2_bottom_vect = zeros(3,(XY_num + 1));
wire2_side1_vect = zeros(3,(Z_num + 1));
wire2_top_vect = zeros(3,(XY_num + 1));
wire2_side2_vect = zeros(3,(Z_num + 1));

wire1_XY_vect_step = abs((wire1_bottom_end(2) -
wire1_bottom_start(2))/XY_num);
wire1_Z_vect_step = abs((wire1_side1_end(3) - wire1_side1_start(3))/Z_num);

wire2_XY_vect_step = abs((wire2_bottom_end(1) -
wire2_bottom_start(1))/XY_num);
wire2_Z_vect_step = abs((wire2_side2_end(3) - wire2_side2_start(3))/Z_num);

for i = 1:(XY_num+1)
    if i == 1
        wire1_bottom_vect(:,i) = wire1_bottom_start;
        wire1_top_vect(:,i) = wire1_top_start;

        wire2_bottom_vect(:,i) = wire2_bottom_start;
        wire2_top_vect(:,i) = wire2_top_start;
    else
        wire1_bottom_vect(:,i) = wire1_bottom_start;
        wire1_top_vect(:,i) = wire1_top_start;

        wire2_bottom_vect(:,i) = wire2_bottom_start;
        wire2_top_vect(:,i) = wire2_top_start;

        wire1_bottom_vect(2,i) = wire1_bottom_vect(2,i-1) +
wire1_XY_vect_step;
        wire1_top_vect(2,i) = wire1_top_vect(2,i-1) - wire1_XY_vect_step;

        wire2_bottom_vect(1,i) = wire2_bottom_vect(1,i-1) +
wire2_XY_vect_step;
        wire2_top_vect(1,i) = wire2_top_vect(1,i-1) - wire2_XY_vect_step;
    end
end

for i = 1:(Z_num+1)
```

```
    if i == 1
        wire1_side1_vect(:,i) = wire1_side1_start;
        wire1_side2_vect(:,i) = wire1_side2_start;

        wire2_side1_vect(:,i) = wire2_side1_start;
        wire2_side2_vect(:,i) = wire2_side2_start;
    else
        wire1_side1_vect(:,i) = wire1_side1_start;
        wire1_side2_vect(:,i) = wire1_side2_start;

        wire2_side1_vect(:,i) = wire2_side1_start;
        wire2_side2_vect(:,i) = wire2_side2_start;

        wire1_side1_vect(3,i) = wire1_side1_vect(3,i-1) + wire1_Z_vect_step;
        wire1_side2_vect(3,i) = wire1_side2_vect(3,i-1) - wire1_Z_vect_step;

        wire2_side1_vect(3,i) =  wire2_side1_vect(3,i-1) + wire2_Z_vect_step;
        wire2_side2_vect(3,i) = wire2_side2_vect(3,i-1) - wire2_Z_vect_step;
    end
end

%Graph Limits
Plot_X_Lim = [0,1]; %0 to 1cm X
Plot_Y_Lim = [0,1]; %0 to 1cm Y
Plot_XY_Lim = [0,1,0,1];
Plot_Z_Lim = [-1,1]; %-1 to 1cm Z

%Generate coordinates
counter = 1;
maxNum = (numPoints+1)^2;
%Loop to place at point at specified intervals in XYZ plane
for i = 1:(numPoints+1)
    setX = ((i-1)*(planeLim(1)-planeStart(1))/numPoints) + planeStart(1);
    pointLoc(1,counter:maxNum) = setX;
    for j = 1:(numPoints+1)
        setY = ((j-1)*(planeLim(2)-planeStart(2))/numPoints) + planeStart(2);
        pointLoc(2,counter:maxNum) = setY;
        pointLoc(3,counter) = setZ;
        counter = counter + 1;
    end
end

%Calculated Values
dL = wireLen / vectLen; %Differential Wire Length

dL_Vect_wire_1_top = (wire1_top_end - wire1_top_start)/XY_num;
dL_Vect_wire_1_side1 = (wire1_side1_end - wire1_side1_start)/Z_num;
dL_Vect_wire_1_bottom = (wire1_bottom_end - wire1_bottom_start)/XY_num;
dL_Vect_wire_1_side2 = (wire1_side2_end - wire1_side2_start)/Z_num;
dL_Vect_wire_2_top = (wire2_top_end - wire2_top_start)/XY_num;
dL_Vect_wire_2_side1 = (wire2_side1_end - wire2_side1_start)/Z_num;
dL_Vect_wire_2_bottom = (wire2_bottom_end - wire2_bottom_start)/XY_num;
dL_Vect_wire_2_side2 = (wire2_side2_end - wire2_side2_start)/Z_num;
```

```
position = zeros(3,vectLen); %3D Position Vector
position(2,:) = linspace(0,wireLen,vectLen); %1000 points between 0 and 1cm
wire_shift = planeLim(1)/num_wires;

r_Vect_wire_1_top = zeros(3,XY_num+1); %Distance Vector between Point and
Position
r_Vect_wire_1_side1 = zeros(3,Z_num+1); %Distance Vector between Point and
Position
r_Vect_wire_1_bottom = zeros(3,XY_num+1); %Distance Vector between Point and
Position
r_Vect_wire_1_side2 = zeros(3,Z_num+1); %Distance Vector between Point and
Position
r_Vect_wire_2_top = zeros(3,XY_num+1); %Distance Vector between Point and
Position
r_Vect_wire_2_side1 = zeros(3,Z_num+1); %Distance Vector between Point and
Position
r_Vect_wire_2_bottom = zeros(3,XY_num+1); %Distance Vector between Point and
Position
r_Vect_wire_2_side2 = zeros(3,Z_num+1); %Distance Vector between Point and
Position

r_Mag_wire_1_top = zeros(1,XY_num+1); %Distance Magnitude between Point and
Position
r_Mag_wire_1_side1 = zeros(1,Z_num+1); %Distance Magnitude between Point and
Position
r_Mag_wire_1_bottom = zeros(1,XY_num+1); %Distance Magnitude between Point and
Position
r_Mag_wire_1_side2 = zeros(1,Z_num+1); %Distance Magnitude between Point and
Position
r_Mag_wire_2_top = zeros(1,XY_num+1); %Distance Magnitude between Point and
Position
r_Mag_wire_2_side1 = zeros(1,Z_num+1); %Distance Magnitude between Point and
Position
r_Mag_wire_2_bottom = zeros(1,XY_num+1); %Distance Magnitude between Point and
Position
r_Mag_wire_2_side2 = zeros(1,Z_num+1); %Distance Magnitude between Point and
Position

r_Unit_wire_1_top = zeros(3,XY_num+1); %Distance Unit Vector between Point and
Position
r_Unit_wire_1_side1 = zeros(3,Z_num+1); %Distance Unit Vector between Point
and Position
r_Unit_wire_1_bottom = zeros(3,XY_num+1); %Distance Unit Vector between Point
and Position
r_Unit_wire_1_side2 = zeros(3,Z_num+1); %Distance Unit Vector between Point
and Position
r_Unit_wire_2_top = zeros(3,XY_num+1); %Distance Unit Vector between Point and
Position
r_Unit_wire_2_side1 = zeros(3,Z_num+1); %Distance Unit Vector between Point
and Position
r_Unit_wire_2_bottom = zeros(3,XY_num+1); %Distance Unit Vector between Point
and Position
r_Unit_wire_2_side2 = zeros(3,Z_num+1); %Distance Unit Vector between Point
and Position
```

```
r2_Mag_wire_1_top = zeros(1,XY_num+1); %^Squared
r2_Mag_wire_1_side1 = zeros(1,Z_num+1); %^Squared
r2_Mag_wire_1_bottom = zeros(1,XY_num+1); %^Squared
r2_Mag_wire_1_side2 = zeros(1,Z_num+1); %^Squared
r2_Mag_wire_2_top = zeros(1,XY_num+1); %^Squared
r2_Mag_wire_2_side1 = zeros(1,Z_num+1); %^Squared
r2_Mag_wire_2_bottom = zeros(1,XY_num+1); %^Squared
r2_Mag_wire_2_side2 = zeros(1,Z_num+1); %^Squared

vectLen = XY_num+1;

dB = zeros(3,vectLen); %Differential B field
dBvect = zeros(3,vectLen); %Differential B field
dBMag = zeros(1,vectLen); %Differential B field Magnitude
dBPlotter = zeros(3,vectLen); %Differential B field Magnitude
pointLocMag = sqrt(pointLoc(1)^2 + pointLoc(2)^2 + pointLoc(3)^2);
pointLocUnit = pointLoc/pointLocMag; %Unit Vector pointing from origin to
point
B_Vect = zeros(4,maxNum); %B field vector
B_Field = zeros(3,maxNum); %2D B field vector
B_Field_Plot = zeros(3,maxNum); %2D B field vector
B_display = "n/a";
B_Vect_wires = zeros(4,maxNum,num_wires); %B field vector
B_Field_wires = zeros(3,maxNum,num_wires); %2D B field vector
B_Vect_wires_sum = zeros(4,maxNum); %B field vector
B_Field_wires_sum = zeros(3,maxNum); %2D B field vector

%Loop through all points in plane
for k = 1:num_wires+1
    %Loop through all points in plane
    for j = 1:maxNum
        %For loop to compute differentials
        for i = 1:XY_num+1
            if i <= Z_num+1
                r2_Mag_wire_1_side1(i) = (pointLoc(1,j) -
wire1_side1_vect(1,i))^2 + ...
                    (pointLoc(2,j) - wire1_side1_vect(2,i))^2 + (pointLoc(3,j)
- wire1_side1_vect(3,i))^2;

                r2_Mag_wire_1_side2(i) = (pointLoc(1,j) -
wire1_side2_vect(1,i))^2 + ...
                    (pointLoc(2,j) - wire1_side2_vect(2,i))^2 + (pointLoc(3,j)
- wire1_side2_vect(3,i))^2;

                r2_Mag_wire_2_side1(i) = (pointLoc(1,j) -
wire2_side1_vect(1,i))^2 + ...
                    (pointLoc(2,j) - wire2_side1_vect(2,i))^2 + (pointLoc(3,j)
- wire2_side1_vect(3,i))^2;

                r2_Mag_wire_2_side2(i) = (pointLoc(1,j) -
wire2_side2_vect(1,i))^2 + ...
                    (pointLoc(2,j) - wire2_side2_vect(2,i))^2 + (pointLoc(3,j)
- wire2_side2_vect(3,i))^2;
```

```
            end
            %Distance vector between Point and Position
            r2_Mag_wire_1_top(i) = (pointLoc(1,j) - wire1_top_vect(1,i))^2 +
...
                (pointLoc(2,j) - wire1_top_vect(2,i))^2 + (pointLoc(3,j) -
wire1_top_vect(3,i))^2;

            r2_Mag_wire_1_bottom(i) = (pointLoc(1,j) -
wire1_bottom_vect(1,i))^2 + ...
                (pointLoc(2,j) - wire1_bottom_vect(2,i))^2 + (pointLoc(3,j) -
wire1_bottom_vect(3,i))^2;

            r2_Mag_wire_2_top(i) = (pointLoc(1,j) - wire2_top_vect(1,i))^2 +
...
                (pointLoc(2,j) - wire2_top_vect(2,i))^2 + (pointLoc(3,j) -
wire2_top_vect(3,i))^2;

            r2_Mag_wire_2_bottom(i) = (pointLoc(1,j) -
wire2_bottom_vect(1,i))^2 + ...
                (pointLoc(2,j) - wire2_bottom_vect(2,i))^2 + (pointLoc(3,j) -
wire2_bottom_vect(3,i))^2;


            %Distance Between Point and Position
            r_Mag_wire_1_top(i) = sqrt(r2_Mag_wire_1_top(i));
            r_Mag_wire_1_bottom(i) = sqrt(r2_Mag_wire_1_bottom(i));
            r_Mag_wire_2_top(i) = sqrt(r2_Mag_wire_2_top(i));
            r_Mag_wire_2_bottom(i) = sqrt(r2_Mag_wire_2_bottom(i));

            if i <= Z_num+1
                r_Mag_wire_1_side1(i) = sqrt(r2_Mag_wire_1_side1(i));
                r_Mag_wire_1_side2(i) = sqrt(r2_Mag_wire_1_side2(i));
                r_Mag_wire_2_side1(i) = sqrt(r2_Mag_wire_2_side1(i));
                r_Mag_wire_2_side2(i) = sqrt(r2_Mag_wire_2_side2(i));
            end

            %Distance vector between Point and Position
            if i <= Z_num+1
                r_Vect_wire_1_side1(:,i) = [pointLoc(1,j) -
wire1_side1_vect(1,i), ...
                    pointLoc(2,j) - wire1_side1_vect(2,i), pointLoc(3,j) -
wire1_side1_vect(3,i)];

                r_Vect_wire_1_side2(:,i) = [pointLoc(1,j) -
wire1_side2_vect(1,i), ...
                    pointLoc(2,j) - wire1_side2_vect(2,i), pointLoc(3,j) -
wire1_side2_vect(3,i)];

                r_Vect_wire_2_side1(:,i) = [pointLoc(1,j) -
wire2_side1_vect(1,i), ...
                    pointLoc(2,j) - wire2_side1_vect(2,i), pointLoc(3,j) -
wire2_side1_vect(3,i)];
```

```
                r_Vect_wire_2_side2(:,i) = [pointLoc(1,j) -
wire2_side2_vect(1,i), ...
                   pointLoc(2,j) - wire2_side2_vect(2,i), pointLoc(3,j) -
wire2_side2_vect(3,i)];

            end

            r_Vect_wire_1_top(:,i) = [pointLoc(1,j) - wire1_top_vect(1,i), ...
                pointLoc(2,j) - wire1_top_vect(2,i), pointLoc(3,j) -
wire1_top_vect(3,i)];

            r_Vect_wire_1_bottom(:,i) = [pointLoc(1,j) -
wire1_bottom_vect(1,i), ...
                pointLoc(2,j) - wire1_bottom_vect(2,i), pointLoc(3,j) -
wire1_bottom_vect(3,i)];

            r_Vect_wire_2_top(:,i) = [pointLoc(1,j) - wire2_top_vect(1,i), ...
                pointLoc(2,j) - wire2_top_vect(2,i), pointLoc(3,j) -
wire2_top_vect(3,i)];

            r_Vect_wire_2_bottom(:,i) = [pointLoc(1,j) -
wire2_bottom_vect(1,i), ...
                pointLoc(2,j) - wire2_bottom_vect(2,i), pointLoc(3,j) -
wire2_bottom_vect(3,i)];

            %Unit Vector from Position to Point
            if i <= Z_num+1
                r_Unit_wire_1_side1(:,i) = r_Vect_wire_1_side1(:,i) /
r_Mag_wire_1_side1(i);
                r_Unit_wire_1_side2(:,i) = r_Vect_wire_1_side2(:,i) /
r_Mag_wire_1_side2(i);
                r_Unit_wire_2_side1(:,i) = r_Vect_wire_2_side1(:,i) /
r_Mag_wire_2_side1(i);
                r_Unit_wire_2_side2(:,i) = r_Vect_wire_2_side2(:,i) /
r_Mag_wire_2_side2(i);
            end

            r_Unit_wire_1_top(:,i) = r_Vect_wire_1_top(:,i) /
r_Mag_wire_1_top(i);

            r_Unit_wire_1_bottom(:,i) = r_Vect_wire_1_bottom(:,i) /
r_Mag_wire_1_bottom(i);

            r_Unit_wire_2_top(:,i) = r_Vect_wire_2_top(:,i) /
r_Mag_wire_2_top(i);

            r_Unit_wire_2_bottom(:,i) = r_Vect_wire_2_bottom(:,i) /
r_Mag_wire_2_bottom(i);


            %Compute X, Y, and Z components of dB
            dBvect(1,i) = 0;
            if i <= Z_num+1
```

```
                dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_1_side1(1,i)*dL_Vect_wire_1_side1(1) / r2_Mag_wire_1_side1(i));
%X
                dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_1_side2(1,i)*dL_Vect_wire_1_side2(1) / r2_Mag_wire_1_side2(i));
%X
                dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_2_side1(1,i)*dL_Vect_wire_2_side1(1) / r2_Mag_wire_2_side1(i));
%X
                dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_2_side2(1,i)*dL_Vect_wire_2_side2(1) / r2_Mag_wire_2_side2(i));
%X

            end

            dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_1_top(1,i)*dL_Vect_wire_1_top(1) / r2_Mag_wire_1_top(i)); %X
            dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_1_bottom(1,i)*dL_Vect_wire_1_bottom(1) /
r2_Mag_wire_1_bottom(i)); %X
            dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_2_top(1,i)*dL_Vect_wire_2_top(1) / r2_Mag_wire_2_top(i)); %X
            dBvect(1,i) = dBvect(1,i) +
(r_Unit_wire_2_bottom(1,i)*dL_Vect_wire_2_bottom(1) /
r2_Mag_wire_2_bottom(i)); %X

            dBvect(2,i) = 0;
            if i <= Z_num+1
                dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_1_side1(2,i)*dL_Vect_wire_1_side1(2) / r2_Mag_wire_1_side1(i));
%Y
                dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_1_side2(2,i)*dL_Vect_wire_1_side2(2) / r2_Mag_wire_1_side2(i));
%Y
                dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_2_side1(2,i)*dL_Vect_wire_2_side1(2) / r2_Mag_wire_2_side1(i));
%Y
                dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_2_side2(2,i)*dL_Vect_wire_2_side2(2) / r2_Mag_wire_2_side2(i));
%Y

            end

            dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_1_top(2,i)*dL_Vect_wire_1_top(2) / r2_Mag_wire_1_top(i)); %Y
            dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_1_bottom(2,i)*dL_Vect_wire_1_bottom(2) /
r2_Mag_wire_1_bottom(i)); %Y
            dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_2_top(2,i)*dL_Vect_wire_2_top(2) / r2_Mag_wire_2_top(i)); %Y
            dBvect(2,i) = dBvect(2,i) +
(r_Unit_wire_2_bottom(2,i)*dL_Vect_wire_2_bottom(2) /
r2_Mag_wire_2_bottom(i)); %Y
```

```
            dBvect(3,i) = 0;
            if i <= Z_num+1
                dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_1_side1(3,i)*dL_Vect_wire_1_side1(3) / r2_Mag_wire_1_side1(i));
%Z
                dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_1_side2(3,i)*dL_Vect_wire_1_side2(3) / r2_Mag_wire_1_side2(i));
%Z
                dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_2_side1(3,i)*dL_Vect_wire_2_side1(3) / r2_Mag_wire_2_side1(i));
%Z
                dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_2_side2(3,i)*dL_Vect_wire_2_side2(3) / r2_Mag_wire_2_side2(i));
%Z

            end
            dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_1_top(3,i)*dL_Vect_wire_1_top(3) / r2_Mag_wire_1_top(i)); %Z
            dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_1_bottom(3,i)*dL_Vect_wire_1_bottom(3) /
r2_Mag_wire_1_bottom(i)); %Z
            dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_2_top(3,i)*dL_Vect_wire_2_top(3) / r2_Mag_wire_2_top(i)); %Z
            dBvect(3,i) = dBvect(3,i) +
(r_Unit_wire_2_bottom(3,i)*dL_Vect_wire_2_bottom(3) /
r2_Mag_wire_2_bottom(i)); %Z

            %Scale dB by appropriate values per Biot-Savart Law Equation
            dB(:,i) = dBvect(:,i)*(u0*I)/(4*pi);

            %Magnitude of dB Vector
            dBMag(i) = sqrt(dB(1,i)^2 + dB(2,i)^2 + dB(3,i)^2);
        end

        B_sum = sum(dBMag); %Total B field magnitude at point

        %Output text
        if pointLoc(1,j) == 0.005
            if pointLoc(2,j) == 0.005
                B_display = num2str(B_sum);
            end
        end

        %Fill arrays
        B_Vect(4,j) = B_sum;
        B_Vect(3,j) = pointLoc(3,j);
        B_Vect(2,j) = pointLoc(2,j);
        B_Vect(1,j) = pointLoc(1,j);

        B_Vect_wires(4,j,k) = B_sum;
        B_Vect_wires(3,j,k) = pointLoc(3,j);
        B_Vect_wires(2,j,k) = pointLoc(2,j);
        B_Vect_wires(1,j,k) = pointLoc(1,j);
```

```
        B_Vect_wires_sum(4,j) = B_Vect_wires_sum(4,j) + B_Vect(4,j);
        B_Vect_wires_sum(3,j) = B_Vect_wires_sum(3,j) + B_Vect(3,j);
        B_Vect_wires_sum(2,j) = B_Vect_wires_sum(2,j) + B_Vect(2,j);
        B_Vect_wires_sum(1,j) = B_Vect_wires_sum(1,j) + B_Vect(1,j);

        B_Field(3,j) = B_sum;
        B_Field(2,j) = pointLoc(2,j);
        B_Field(1,j) = pointLoc(1,j);

        B_Field_wires(3,j,k) = B_sum;
        B_Field_wires(2,j,k) = pointLoc(2,j);
        B_Field_wires(1,j,k) = pointLoc(1,j);

        B_Field_wires_sum(3,j) = B_Field_wires_sum(3,j) + B_Field(3,j);
        B_Field_wires_sum(2,j) = B_Field_wires_sum(2,j) + B_Field(2,j);
        B_Field_wires_sum(1,j) = B_Field_wires_sum(1,j) + B_Field(1,j);
    end

    position(1,:) = position(1,:) + wire_shift;

    wire1_bottom_start(1) = wire1_bottom_start(1) + wire_iterator;
    wire1_bottom_end(1) = wire1_bottom_end(1) + wire_iterator;
    wire1_top_start(1) = wire1_top_start(1) + wire_iterator;
    wire1_top_end(1) = wire1_top_end(1) + wire_iterator;
    wire1_side1_start(1) = wire1_side1_start(1) + wire_iterator;
    wire1_side1_end(1) = wire1_side1_end(1) + wire_iterator;
    wire1_side2_start(1) = wire1_side2_start(1) + wire_iterator;
    wire1_side2_end(1) = wire1_side2_end(1) + wire_iterator;

    wire2_bottom_start(2) = wire2_bottom_start(2) + wire_iterator;
    wire2_bottom_end(2) = wire2_bottom_end(2) + wire_iterator;
    wire2_top_start(2) = wire2_top_start(2) + wire_iterator;
    wire2_top_end(2) = wire2_top_end(2) + wire_iterator;
    wire2_side1_start(2) = wire2_side1_start(2) + wire_iterator;
    wire2_side1_end(2) = wire2_side1_end(2) + wire_iterator;
    wire2_side2_start(2) = wire2_side2_start(2) + wire_iterator;
    wire2_side2_end(2) = wire2_side2_end(2) + wire_iterator;

    for q = 1:(XY_num+1)
        if q == 1
            wire1_bottom_vect(:,q) = wire1_bottom_start;
            wire1_top_vect(:,q) = wire1_top_start;

            wire2_bottom_vect(:,q) = wire2_bottom_start;
            wire2_top_vect(:,q) = wire2_top_start;
        else
            wire1_bottom_vect(:,q) = wire1_bottom_start;
            wire1_top_vect(:,q) = wire1_top_start;

            wire2_bottom_vect(:,q) = wire2_bottom_start;
            wire2_top_vect(:,q) = wire2_top_start;

            wire1_bottom_vect(2,q) = wire1_bottom_vect(2,q-1) +
wire1_XY_vect_step;
```

```
            wire1_top_vect(2,q) = wire1_top_vect(2,q-1) - wire1_XY_vect_step;

            wire2_bottom_vect(1,q) = wire2_bottom_vect(1,q-1) +
wire2_XY_vect_step;
            wire2_top_vect(1,q) = wire2_top_vect(1,q-1) - wire2_XY_vect_step;
        end
    end

    for q = 1:(Z_num+1)
        if q == 1
            wire1_side1_vect(:,q) = wire1_side1_start;
            wire1_side2_vect(:,q) = wire1_side2_start;

            wire2_side1_vect(:,q) = wire2_side1_start;
            wire2_side2_vect(:,q) = wire2_side2_start;
        else
            wire1_side1_vect(:,q) = wire1_side1_start;
            wire1_side2_vect(:,q) = wire1_side2_start;

            wire2_side1_vect(:,q) = wire2_side1_start;
            wire2_side2_vect(:,q) = wire2_side2_start;

            wire1_side1_vect(3,q) = wire1_side1_vect(3,q-1) +
wire1_Z_vect_step;
            wire1_side2_vect(3,q) = wire1_side2_vect(3,q-1) -
wire1_Z_vect_step;

            wire2_side1_vect(3,q) =  wire2_side1_vect(3,q-1) +
wire2_Z_vect_step;
            wire2_side2_vect(3,q) = wire2_side2_vect(3,q-1) -
wire2_Z_vect_step;
        end
    end
end

%Re-Format B Field for Plotting
B_Field_Surf = zeros(numPoints+1,numPoints+1);
B_Field_Plot(1,:) = B_Field_wires_sum(1,:)*100;
B_Field_Plot(2,:) = B_Field_wires_sum(2,:)*100;
B_Field_Plot(3,:) = B_Field_wires_sum(3,:);
counter = 1;
for i = 1:maxNum
    if isnan(B_Field_Plot(3,i))
        B_Field_Plot(3,i) = max(B_Field_Plot(3,:));
    end
end
for i = 1:numPoints+1
    for j = 1:numPoints+1
        B_Field_Surf(j,i) = B_Field_Plot(3,counter);
        counter = counter + 1;
    end
end

%Results
```

```
disp("dB Sum (B field magnitude) at X="+ B_Field(1,61)*100 +"cm, Y="+
B_Field(2,61)*100 +"cm, Z=" + setZ*100 + "cm is: " + B_Field(3,61) + " T");

%Plot dB magnitude along wire
figure(2);
subplot(2,1,1);
hold on
for i = 1:maxNum
    plot3(100*pointLoc(1,i),100*pointLoc(2,i),100*pointLoc(3,i),'k.');
end
surf(linspace(planeStart(1),planeLim(1)*100,numPoints+1),...
    linspace(planeStart(2),planeLim(2)*100,numPoints+1),B_Field_Surf);
colorbar;
xlabel('X (cm)');
ylabel('Y (cm)');
zlabel('Z (cm)');
xlim(Plot_X_Lim);
ylim(Plot_Y_Lim);
zlim(Plot_Z_Lim);
grid on;
legend('.','Point Locations','Location',...
    'NorthEastOutside');
title('Magnetic Field Strength vs. Position Within Sensor');
view(25,25);
hold off

%Add second plot for 2D detail
subplot(2,2,3);
plot3(100*position(1,:),100*position(2,:),100*position(3,:));
hold on
for i = 1:maxNum
    plot3(100*pointLoc(1,i),100*pointLoc(2,i),100*pointLoc(3,i),'k.');
end
%
xlabel('X (cm)');
ylabel('Y (cm)');
zlabel('Z (cm)');
xlim(Plot_X_Lim);
ylim(Plot_Y_Lim);
zlim(Plot_Z_Lim);
grid on;
title('2D Planar View - YZ Plane');
hold off
view(270,0);

%Add third plot for 2D detail
subplot(2,2,4);
plot3(100*position(1,:),100*position(2,:),100*position(3,:));
hold on
for i = 1:maxNum
    plot3(100*pointLoc(1,i),100*pointLoc(2,i),100*pointLoc(3,i),'k.');
end
%
xlabel('X (cm)');
```

```
ylabel('Y (cm)');
zlabel('Z (cm)');
xlim(Plot_X_Lim);
ylim(Plot_Y_Lim);
zlim(Plot_Z_Lim);
grid on;
title('2D Planar View - XZ Plane');
hold off
view(0,0);

%Add another figure for log10 detailed view of field
figure(3);
subplot(2,1,1);
surf(linspace(planeStart(1),planeLim(1)*100,numPoints+1),...
    linspace(planeStart(2),planeLim(2)*100,numPoints+1),log10(B_Field_Surf));
colorbar;
xlabel('X (cm)');
ylabel('Y (cm)');
zlabel('Z log10(T)');
grid on;
title('log10(Magnetic Field Strength)');

%Add second plot for 2D detail
subplot(2,2,3);
surf(linspace(planeStart(1),planeLim(1)*100,numPoints+1),...
    linspace(planeStart(2),planeLim(2)*100,numPoints+1),log10(B_Field_Surf));
view(270,0);
xlabel('X (cm)');
ylabel('Y (cm)');
zlabel('Z log10(T)');
grid on;

%Add third plot for 2D detail
subplot(2,2,4);
surf(linspace(planeStart(1),planeLim(1)*100,numPoints+1),...
    linspace(planeStart(2),planeLim(2)*100,numPoints+1),log10(B_Field_Surf));
view(0,0);
xlabel('X (cm)');
ylabel('Y (cm)');
zlabel('Z log10(T)');
grid on;


%Add a final plot showing the sensor coils
figure(4);

wire1_bottom_start(1) = 0;
wire1_bottom_end(1) = 0;
wire1_top_start(1) = 0;
wire1_top_end(1) = 0;
wire1_side1_start(1) = 0;
wire1_side1_end(1) = 0;
wire1_side2_start(1) = 0;
wire1_side2_end(1) = 0;
```

```
wire2_bottom_start(2) = 0;
wire2_bottom_end(2) = 0;
wire2_top_start(2) = 0;
wire2_top_end(2) = 0;
wire2_side1_start(2) = 0;
wire2_side1_end(2) = 0;
wire2_side2_start(2) = 0;
wire2_side2_end(2) = 0;

subplot(1,1,1);
hold on
for i = 1:num_wires

    for q = 1:(XY_num+1)
        if q == 1
            wire1_bottom_vect(:,q) = wire1_bottom_start;
            wire1_top_vect(:,q) = wire1_top_start;

            wire2_bottom_vect(:,q) = wire2_bottom_start;
            wire2_top_vect(:,q) = wire2_top_start;
        else
            wire1_bottom_vect(:,q) = wire1_bottom_start;
            wire1_top_vect(:,q) = wire1_top_start;

            wire2_bottom_vect(:,q) = wire2_bottom_start;
            wire2_top_vect(:,q) = wire2_top_start;

            wire1_bottom_vect(2,q) = wire1_bottom_vect(2,q-1) +
wire1_XY_vect_step;
            wire1_top_vect(2,q) = wire1_top_vect(2,q-1) - wire1_XY_vect_step;

            wire2_bottom_vect(1,q) = wire2_bottom_vect(1,q-1) +
wire2_XY_vect_step;
            wire2_top_vect(1,q) = wire2_top_vect(1,q-1) - wire2_XY_vect_step;
        end
    end

    for q = 1:(Z_num+1)
        if q == 1
            wire1_side1_vect(:,q) = wire1_side1_start;
            wire1_side2_vect(:,q) = wire1_side2_start;

            wire2_side1_vect(:,q) = wire2_side1_start;
            wire2_side2_vect(:,q) = wire2_side2_start;
        else
            wire1_side1_vect(:,q) = wire1_side1_start;
            wire1_side2_vect(:,q) = wire1_side2_start;

            wire2_side1_vect(:,q) = wire2_side1_start;
            wire2_side2_vect(:,q) = wire2_side2_start;

            wire1_side1_vect(3,q) = wire1_side1_vect(3,q-1) +
wire1_Z_vect_step;
```

```
            wire1_side2_vect(3,q) = wire1_side2_vect(3,q-1) -
wire1_Z_vect_step;

            wire2_side1_vect(3,q) =  wire2_side1_vect(3,q-1) +
wire2_Z_vect_step;
            wire2_side2_vect(3,q) = wire2_side2_vect(3,q-1) -
wire2_Z_vect_step;
        end
    end

    plot3( 100*wire1_side1_vect(1,:), 100*wire1_side1_vect(2,:),
100*wire1_side1_vect(3,:), 'b-');
    plot3( 100*wire1_side2_vect(1,:), 100*wire1_side2_vect(2,:),
100*wire1_side2_vect(3,:), 'b-');

    plot3( 100*wire2_side2_vect(1,:),wire2_side2_vect(2,:),
100*wire2_side2_vect(3,:), 'r-');
    plot3( 100*wire2_side1_vect(1,:), 100*wire2_side1_vect(2,:),
100*wire2_side1_vect(3,:), 'r-');

    plot3( 100*wire1_bottom_vect(1,:), 100*wire1_bottom_vect(2,:),
100*wire1_bottom_vect(3,:), 'b-');
    plot3( 100*wire1_top_vect(1,:), 100*wire1_top_vect(2,:),
100*wire1_top_vect(3,:), 'b-');

    plot3( 100*wire2_bottom_vect(1,:), 100*wire2_bottom_vect(2,:),
100*wire2_bottom_vect(3,:), 'r-');
    plot3( 100*wire2_top_vect(1,:), 100*wire2_top_vect(2,:),
100*wire2_top_vect(3,:), 'r-');

    wire1_bottom_start(1) = wire1_bottom_start(1) + wire_iterator;
    wire1_bottom_end(1) = wire1_bottom_end(1) + wire_iterator;
    wire1_top_start(1) = wire1_top_start(1) + wire_iterator;
    wire1_top_end(1) = wire1_top_end(1) + wire_iterator;
    wire1_side1_start(1) = wire1_side1_start(1) + wire_iterator;
    wire1_side1_end(1) = wire1_side1_end(1) + wire_iterator;
    wire1_side2_start(1) = wire1_side2_start(1) + wire_iterator;
    wire1_side2_end(1) = wire1_side2_end(1) + wire_iterator;

    wire2_bottom_start(2) = wire2_bottom_start(2) + wire_iterator;
    wire2_bottom_end(2) = wire2_bottom_end(2) + wire_iterator;
    wire2_top_start(2) = wire2_top_start(2) + wire_iterator;
    wire2_top_end(2) = wire2_top_end(2) + wire_iterator;
    wire2_side1_start(2) = wire2_side1_start(2) + wire_iterator;
    wire2_side1_end(2) = wire2_side1_end(2) + wire_iterator;
    wire2_side2_start(2) = wire2_side2_start(2) + wire_iterator;
    wire2_side2_end(2) = wire2_side2_end(2) + wire_iterator;

end

xlabel('X (cm)');
ylabel('Y (cm)');
zlabel('Z (cm)');
xlim(Plot_X_Lim);
```

```
ylim(Plot_Y_Lim);
zlim(Plot_Z_Lim);
grid on;
title('3D Sensor Coils');
view(45,45);
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% End %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

References

[1] Vittorio M. N. Passaro et al. “Gyroscope Technology and Applications: A Review in the Industrial Perspective”, Sensors Journal, Volume 17, Number 2284, October 2017, DOI 10.3390/s17102284

[2] S. Alper, Y. Temiz, and T. Akin, “A Compact Angular Rate Sensor System Using a Fully Decoupled Silicon-on-Glass MEMS Gyroscope,” Journal of Microelectromechanical Systems, Vol. 17, No. 6, pp. 1418-1429, December 2008, doi: 10.1109/JMEMS.2008.2007274.

[3] Olaf Deppe et al. “MEMS and FOG Technologies for Tactical and Navigation Grade Inertial Sensors—Recent Improvements and Comparison”, Sensors Journal, Volume 17, Number 3, March 2017, DOI 10.3390/s17030567

[4] Krug, Brian Gerald, "Sensing Gyroscopic Properties of Rotating Magnetic Nanoparticles in Solution", PhD dissertation, Graduate College, Western Michigan University, Kalamazoo, MI, USA, 2016.

[5] L. Brits, *Euler Angles*, 2008, [Online], Available: https://commons.wikimedia.org/wiki/File:Euler.png

[6] "Biot-Savart Law," in Merriam-Webster.com dictionary

[7] I. S. Grant and W. R. Phillips, Electromagnetism, 2nd Edition. Chichester, West Sussex, UK: John Wiley & Sons, 2008.

[8] Poddar, S., Kumar, V., and Kumar, A., "A Comprehensive Overview of Inertial Sensor Calibration Techniques." *ASME. J. Dyn. Sys., Meas., Control.*; vol. 139, no. 1, January 2017, doi: https://doi.org/10.1115/1.4034419

[9] Borochin, A. "How a new generation of highly integrated MEMS sensors opens up new possibilities for the application of motion-sensing and location- tracking". *Future Technology Magazine*, (July, 2014), 26-27.

[10] B. Rogers, J. Adams, and S. Pennathur, "Nanobiotechnology" in *Nanotechnology, Understanding Small Systems*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2015, ch. 10, pp. 336.

[11] University of California, Berkeley, "The Euler angle parameterization", https://rotations.berkeley.edu/the-euler-angle-parameterization/

[12] Griffiths, David J., Introduction to Electrodynamics, 3rd Edition. Upper Saddle River, New Jersey, United States: Prentice Hall, 2007, ch. 5, problem 44.

[13] Massachusetts Institute of Technology, "Force and Torque on a Magnetic Dipole", http://web.mit.edu/8.02t/www/mitxmaterials/Presentations/Presentation_W07D2.pdf

[14] AD Elster, “Magnetic Moments and Dipoles”, http://mriquestions.com/magnetic-dipole-moment.html

[15] Rochester Institute of Technology, “Torque as a Vector Quantity”, http://spiff.rit.edu/classes/phys312/workshops/w2c/angmom/vec_torque.html

[16] Rochester Institute of Technology, “Magnetic Torques and Ampere’s Law”, http://spiff.rit.edu/classes/phys213/lectures/amp/amp_long.html

[17] F. Gulmammadov, "Analysis, modeling and compensation of bias drift in MEMS inertial sensors," *2009 4th International Conference on Recent Advances in Space Technologies*, Istanbul, 2009, pp. 591-596, doi: 10.1109/RAST.2009.5158260.

[18] S. Odenbach, *Ferrofluids: Magnetically Controllable Fluids and Their Applications*, New York, New York, USA: Springer, 2002.

[19] Yunze Long, Zhaojia Chen, Jean Luc Duvail, Zhiming Zhang, Meixiang Wan, *Electrical and magnetic properties of polyaniline/Fe_3O_4 nanostructures*, Physica B: Condensed Matter, Volume 370, Issues 1–4, 2005, Pages 121-130, ISSN 0921-4526, https://doi.org/10.1016/j.physb.2005.09.009.

[20] "Faraday's law," in Merriam-Webster.com dictionary

[21] Georgia State University, “Faraday’s Law”,

http://hyperphysics.phy-astr.gsu.edu/hbase/electric/farlaw.html

[22] Ivanov, Alexey & Kuznetsova, O. & Subbotin, Igor. (2013). *Magnetic properties of ferrofluid emulsions: the effect of droplet elongation*. Magnetohydrodynamics. 49. 287-292.

10.22364/mhd.49.3-4.7.