MODELING AND CONTROL OF A PERMANENTLY MAGNETIZED SPHERE'S MOTION BY FIELD MANIPULATION

By

MATTHEW GLENN DUVALL

Bachelor of Science
Mechanical Engineering
Oklahoma State University
Stillwater, Oklahoma, USA
1994

Master of Science
Mechanical Engineering
Oklahoma State University
Stillwater, Oklahoma, USA
1997

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
Doctor of Philosophy
December, 2006

MODELING AND CONTROL OF A PERMANENTLY MAGNETIZED SPHERE'S MOTION BY FIELD MANIPULATION

Dissertation Approved:

Eduardo A. Misawa

Dissertation Advisor

Prabhakar R. Pagilla

Frank W. Chambers

Martin Hagan

Dean of the Graduate College

ACKNOWLEDGMENTS

The author would like to sincerely thank the following individuals: Firstly, my academic advisor Dr. Eduardo A. Misawa whose support and advice greatly facilitated this project. Secondly, my academic committee, specifically Dr. Prabhakar R. Pagilla. His guidance, mentorship, and not to mention the many hours of time sacrificed to examining draft manuscripts proved to be invaluable to my educational experience. I would also like to thank Dr. Gary Young for his support during my teaching assistantship which provided a crucial aspect of the doctoral candidacy experience. In addition I would like to thank Dr. Kenneth L. Pottebaum of Seagate Technology, LLC for his professional guidance and counseling regarding the balance of school and academia. Moreover, I would like to thank Dr. Ryan T. Ratliff of the Boeing Company for his assistance in, and clarification of, various technical difficulties. Furthermore, I would like to thank Leon E. Boomershine and Wenjie Li. Without assistance from these two individuals it is doubtful that the experimental hardware setup for this project would have been completed in a timely fashion. Finally, I must offer an extreme amount of gratitude to my wife, Heather D. Duvall, for her unflinching support in my academic endeavors and the many sacrifices for which she has had to endure. Without the support of these individuals, it is inconceivable that this body of work would be completed and this particular scholastic goal accomplished.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

Introductory Work

1.1 Impetus for the Research

In recent decades many new technologies have been proposed and investigated in the biomedical sciences. Several studies have suggested the use of magnetic nano-particles *in vivo* to the human body. The benefits of such methods are believed to be invaluable because of the minimally invasive capabilities of anticipated procedures. Envisioned applications of nano-particles include, but are not limited to, cell labeling, magnetic separation, targeted drug delivery, hyperthermic cell treatment, and MRI contrast enhancement [1]. Significant study has been performed regarding these concepts and is available in references [2, 3, 4, 5]. A second novel application of a magnetically reactive device being introduced to the internal workings of a living body and being manipulated by an applied *ex vivo* magnetic field is magnetic implant and catheter guidance. The practice of this procedure on the human brain is referred to as stereotactic neurosurgery and a significant effort has been put forth on this technique [6, 7]. Similarly, a process utilizing nano sized particles and applied external magnetic fields for intraocular retinal repair is being explored as well [8]. Finally, the fascinating and emerging science of nano-scaled machines has offered the potential employment of microdevices in medical procedures. To reduce device complexity, an external magnetic propulsion technique is envisioned [9, 10].

The common attribute present in all of the aforementioned systems is the use of an applied magnetic field. This field is utilized to exert a desired force and/or torque on an object susceptible to such fields. Upon that realization, one may attempt to first recognize the analogy and similarities to the classic magnetic levitation problem. Indeed a parallel may be observed that manipulation of a particle or micro-device along a unidirectional trajectory is comparable to the well-known technique of suspending or manipulating objects by either the application of alternating fields or utilizing a direct field to maintain the static position of the target object [11, 12]. However, in the aforementioned biomedical processes, the ability to either sustain a static position in a moving fluid medium possessing varying components of velocity in multiple coordinate directions, or propel the target object(s) along a predetermined spatial trajectory elevates this problem to one beyond classical levitation. Achieving this type of dynamics requires application of forces with components in multiple coordinate directions. The resulting fields, field gradients, and the means by which to generate them are far more complex than that of the levitation problem, which often requires force generation only along a single direction. A system such as this presents an interesting control problem for which a suitable descriptive model is necessary before a sufficient treatment and analysis may occur.

The technique of moving an implant or micro-device by fields from static electromagnets is not a new concept [6, 7, 13]. Indeed a substantial amount of work has been performed using a cubic arrangement of coils to facilitate three dimensional motion. A recent technique as described in [13] utilizes six active coils to impart a predetermined force and torque on a target object. The studied approach powers all coils simultaneously with the three opposite the primary attractive coils being operated with a lower current of reverse polarity. This would result in an increase in the field gradients at the location of the target and in effect raise the magnetic force on the object. While this technique has been explored, calculation of the required currents is quite tedious. The device operates in an open loop control scheme and employs minimization techniques to numerically solve for the six coil currents. The difficulty in the current solution for this system rests with the fact that it is underdetermined (i.e., there are more equations than unknowns). For the scenario of moving a spherical object where the orientation is unconstrained, a solution of six unknowns (the currents) from three equations is needed. Hence, many solutions exist, requiring the use of optimization methods to single out a minimal current set. This work explores the technique of reducing the number of coils to only those which are necessary to generate the force. This results in a *determined* system, allowing for an exact calculation of the current set. The analysis will be limited to the less complicated problem of moving a sphere two dimensionally in a fluid medium using static electromagnets. The system consists of four electromagnets, of which only two are used at any time to provide planar motion to the particle. The pair of active coils which will be utilized is determined by the force requirement on the sphere. Before exploring the dynamics of this system in detail, a synopsis of the work available in the literature will be presented.

1.2 Overview of Relevant Research

The use of magnetic particles in a biomedical application is not new and the available work is quite extensive. Early concepts consisted of using magnetic fluids to perform magnetic tracing and mapping of blood flow systems [14]. This technique is “passive” in nature relying on the high permeability characteristic of the introduced fluid for detection by an electromagnetic pickup array. A second concept explored by [14] involves actively guiding and controlling an unmixed and undiluted material in the bloodstream. The envisioned technique would be helpful in sealing off defects in the circulatory system allowing for the initiation of clotting and scarring. Work on this has been limited with very little success due to the nature of ferro-fluidic material not being easily controlled thru magnetic propulsion methods. More recent proposals involve manipulation of *particles* rather than a fluid based material.

The envisioned techniques by which an active guidance system utilizing magnetic propulsion may be employed consist of targeted drug delivery and guidance of micro devices in blood vessels. For the technique of targeted drug delivery, the vast majority of present methods utilize uniform magnetic fields in concert with fluid flow dynamics. In this method, the magnetic particles are moved to the desired location by way of fluid system and applied uniform magnetic fields are then applied to hold the particles in the desired location [1, 2, 5]. While these techniques are promising, a more robust solution would be the employment of an active system to guide the particle to the desired location, independent of the flow dynamics. While this would be the ideal system, present technology is somewhat limited due to the fact that extremely small particles require large field gradients to produce a significant degree of body force on the object. While this does not preclude the use of a magnetic propulsion system, the large power requirements necessary for implementation can be a significant engineering challenge. Guidance of larger scale particles and micro-devices is a concept on which magnetic propulsion and guidance techniques are readily applicable.

Previous investigations of magnetic guidance of microdevices have been undertaken. Once such study has been given in [9, 10]. In this work, an initial investigation to evaluate the feasibility of utilizing a conventional magnetic resonance imaging machine to impart large enough forces on a ferromagnetic sphere was performed. It was the intent of this study to assess the potential for employment of a microdevice in a human circulatory system, therefore the size of the sphere was chosen accordingly. The investigators utilized a 1/8” 1010 carbon steel sphere immersed in water and subjected to uniaxial flow in a 1/4” diameter tube. The MRI was utilized to keep the global position of the sphere static. It was determined that with a field gradient of $0.018 T/m$, the sphere could withstatnd a maximum flow of $0.370 \pm 0.0064 l/min$. This value is comparable to that of the human circulatory system a good distance away from the heart, but still significantly larger than the flow rates in regions furthest from the heart. As previously stated, the fluid flow only was present in a single coordinate direction therefore evaluation of device guidance in multiple directions was not achieved in this work. In addition, the static position control was achieved in an open loop fashion with the current in the electromagnet coils adjusted accordingly to arrive at equilibrium of the sphere in the flow. Feedback position control of a microdevice was not demonstrated in this work.

A second novel concept of a microdevice guided by applied magnetic fields investigated in recent years is available in [15, 16]. This work studies motion of a “screw” type device propelled through both fluid and organic tissue by application of a *rotating* magnetic field. The device is essentially comprised of a cylindrical magnet with a spiral blade residing on its surface, analogous to a mechanical screw or auger. The cylindrical magnet is magnetized in a direction perpendicular to its central axis (a radial direction). The magnetization in this coordinate direction allows for a torque about the central axis of the cylinder upon application of a magnetic field perpendicular to the central magnet axis. The torque about the cylinder will be present until the magnetic moment of the cylinder aligns itself with the applied field. Subsequent rotation of the applied field results in a rotation of the cylinder about its central axis. The forward (or backward) motion of the device then occurs in the direction of the central cylindrical axis and is a product of the mechanical interaction of the spiral blade with the medium in which the device is immersed. The mechanism may be viewed much in the same way as that of an auger boring its way through a material body. Directional changes of the device motion are facilitated by a change in direction of the axis of rotation of the rotating magnetic field. Experimental investigations of this concept utilized a $15mm$ long device with a diameter of $1.2mm$ and a $0.15mm$ spiral blade with a pitch of $45°$. Results of this study were quite interesting with maximum velocities achieved valued at approximately $20mm/s$ and $0.022mm/s$ in fluids with kinematic viscosities of $500mm^2/s$ and $5 \times 10^5 mm^2/s$, respectively. Additionally, maneuvering of the auger device was accomplished in a mazed patterned waterway with $15mm$ wide passage and filled with a silicone oil having a viscosity of $1000mm^2/s$. Finally, additional experimental work has been performed evaluating the ability of a device such as this to “bore” its way through gel or a soft solid. Results in this area were quite satisfactory for the initial study into this technique. While this is a novel technique, it differs subtly from the method proposed in this work. The method proposed in [15, 16] utilizes an applied torque and a mechanical “propeller” effect to achieve device motion. This is different than the technique to be investigated by this work in which the direction of motion is determined by the applied field and the device is effectively pulled through the medium in which it resides.

The body of work most closely related to the study presented in this investigation may be found in [17, 6, 7, 13, 18]. The cited research investigates a system known as the magnetic stereotaxis system. This system is capable in theory of moving a small intracranial implant along a path of minimum neurological damage in the brain by utilizing applied field gradients. The insert is then positioned inside a deep-seated tumor and inductively heated, hence heating and eliminating only the tumor. The envisioned implant would have a dimensional scale $< 3mm$. The initial investigations of this system incorporated a single electromagnet situated on the end of a robotic manipulator. The imaging method employed by this technique consist of a bi-planar fluoroscope which was chosen due to its insensitivity to the applied magnetic fields. Further work into this apparatus emphasized the shortcomings of the moveable electromagnet technique and a system employing an array of stationary electromagnets was the selected path of continued study. More specifically, the stationary electromagnet assembly is a cubic arrangement on which the central axis of each electromagnet is collinear with the global coordinate system. This system is similar to the electromagnetic actuator technique explored in this work. Indeed the mechanism by which the magnetized seed is moved is the method studied. The target object is essentially pulled through the medium in which it resides. For this system however, all six electromagnets are energized. This operation technique results in an *underdetermined* system in current in which more variables are present than equations available to solve. As a result, optimization techniques must be employed to evaluate and search for a current solution satisfying the desired force on the target object. One advantage to this approach of utilizing all electromagnets is the ability to increase the field gradient at the target location, however it must be recognized that the overall system power is greatly increased due to utilization of all coils. In contrast, this work will examine a technique which will employ only the minimum number of coils required to generate the desired force on the target object. Another aspect of the magnetic stereotaxis system is that it is only capable of operation in an "open loop" mode. The target object is propelled in an impulsive fashion with the location being reached by just "nudging" the seed into position. This is presumably due to the difficulties associated with the optimal calculation of the current vector for the movement instance. By utilizing the minimal coil set method, it can be shown that a direct formulation giving the required coil currents can be made. This will be demonstrated with more detail in a later passage.

1.3 Contributions of this Study

There are several contributions produced by this body of work. First, a detailed first principle model of the particle dynamics and electromagnetics will be presented. The electromagnetic model will be one formulated from a simplified coil representation consisting of finite current loops. Use of this method allows for incorporation of the model into a controller in which closed loop control may be achieved. Secondly, a new technique based on the minimal coil set will be presented and explored. Advantages to this technique lie in the fact that a direct formulation may be made to calculate the required coil currents for producing the necessary or desired force on the target object. In addition, the coil switching algorithm will be rooted in a method by which the knowledge of the actual coil currents is unnecessary. Only the geometric field relations of the coils and the target object position is required. Thirdly, an experimental apparatus for investigation of this concept is designed and presented. The test configuration is of "bench top" scale and is optimally designed for minimal use of power. Finally, extensive closed loop experiments in which a small magnetized sphere is guided along 2-D trajectories are performed and the results presented. As this is the initial work for achieving closed loop control in 2 dimensions utilizing electromagnetic actuators, a classical control technique is employed along with the minimal coil set current solution method.

1.4 Overview of this Study

As has been previously stated, this study will limit itself to the simplified system of a magnetized sphere residing in a fluid medium. Additionally the sphere will be constrained to have only planar motion. Therefore, this study will examine the details associated with a system of four coil which are engineered to provide controlled motion of the sphere in the plane. This body of work will be presented in six separate chapters. Chapter 1 provides the motivation for the work and any existing relevant research. These details are available in the previous passages. The theoretical derivation of the system model will be presented in Chapter 2. In this chapter, the dynamic model will be derived from first principles. It describes the particle dynamics as well as the electromagnetics with the geometric field vector functions presented in a general form. Additionally, a state space representation will be presented. Also, the inverse current solution will be formulated. This will correspond to the system being studied which consists of the particle being constrained to a 2-D plane. Subsequently, the technique by which the coil switching will be determined and the resulting calculated current trajectories smoothed will be discussed.

A detailed electromagnetic field description will be presented in Chapter 3. This will be a more rigorous treatment of the general geometric vector field functions associated with the electromagnets which were introduced in Chapter 2. The coil field model will be based on a discrete filamentary loop description. Initially, the field resultant from a single filamentary loop will be presented, followed by a multi-turn expansion. Subsequently, a formulation describing the net field and field gradient generated by the quad coil assembly will be presented and discussed. Following that, relationships for calculating the electromagnet resistance, self inductance, and mutual inductance will be derived. Finally, experimental verification of the predicted field will be presented and discussed.

The design and optimization of the experimental electromagnet assembly will be presented in Chapter 4. Calculation of the electromagnetic characteristics used in the optimization will employ the results of Chapters 2 and 3. A functional for minimization based on minimal power required for moving a magnetized sphere in a circular orbit is derived and presented. Additionally, a minimization technique based on steepest descent is discussed and employed. Finally, the details of the optimally designed electromagnetic coil set are presented.

The results of closed loop control of the sphere's 2-D motion are presented in Chapter 5. A closed loop control law utilizing a classical PID controller with and without feedforward in concert with the inverse current solution is presented in Chapter 2. A numerical simulation is formulated and the results utilized to discuss further considerations about the coil switching technique. Additionally, three different path trajectories are examined in simulation. These consist of a spiral beginning at the origin and approaching a circular orbit, a linear path following the outline of a square, and the point regulation problem in which a single coordinate is provided and the path to approach it is not specified. Simulation results for both controllers with all three trajectories are presented and discussed. Finally, the control method was implemented in experiment and the results are presented and discussed.

CHAPTER 2

Dynamic System Model

2.1 Dynamic Model Development

The equation of motion describing the movement of the target object (sphere) may be devised through a Newtonian formulation given here as

$$m\ddot{\vec{S}}(t) = \vec{F}_{Bias} + \vec{F}_{Mag} + \vec{F}_{Drag}, \tag{2.1}$$

where $\vec{F}_{Bias}$ denotes any bias force, $\vec{F}_{Mag}$ denotes the applied magnetic force, $\vec{F}_{Drag}$ denotes a fluid drag force from the motion of the object relative to the fluid medium it resides in, m denotes the mass of the object, and $\ddot{\vec{S}}(t)$ denotes the vector acceleration of the object in an inertial reference frame. The corresponding trajectory $\vec{S}(t)$ describing the object's motion may be ascertained by solving the differential equation (2.1). For the system consisting of a spherical target residing in a fluid medium, the first term on the right hand side of (2.1) may be defined to represent the resulting force of gravity on the sphere and the buoyant force of the fluid medium on the sphere

$$\vec{F}_{Bias} = (\rho_{sph} - \rho_{fl})V_{sph}\vec{g}, \tag{2.2}$$

where ρ_{sph} denotes the density of the sphere, ρ_{fl} denotes the density of the fluid medium in which the sphere resides, V_{sph} denotes the volume of the sphere, and $\vec{g}$ denotes the acceleration of the sphere due to gravitational attraction. The second term on the right hand side of (2.1) may be expressed by approximating the sphere as a magnetic dipole. This approximation for small objects is widely accepted and has seen some experimental correlation [2, 8]. When a magnetic field is applied, a resulting force and torque act on the dipole. In this case, assuming that the fluid medium has a permeability close to that of free space, the force and torque on the sphere are

$$\vec{F}_{Mag}(\mathbf{H}) = \frac{4\pi\mu_0 r^3}{3}\Big(\mathbf{M}\cdot\nabla\Big)\mathbf{H} \tag{2.3a}$$

$$\vec{\tau}_{Mag}(\mathbf{H}) = \frac{4\pi\mu_0 r^3}{3}(\mathbf{M}\times\mathbf{H}), \tag{2.3b}$$

where $\mathbf{M}$ is the magnetization of the body, μ_0 is the permeability of free space (valued at $4\pi \times 10^{-7}\mathrm{T}\cdot\mathrm{m/A}$), r is the radius of the sphere, and $\mathbf{H}$ is the applied magnetic field strength at the location of the sphere [19, 20]. Note that boldface indicates a vector field. In the case of the particle residing in a medium where the relative permeability $\mu_{fl} \gg 1$, then $\mu_0 \to \mu_0\mu_{fl}$. Recognizing that rotational resistance of the spherical object will be small in most fluid media, it is apparent from (2.3b) that the body will have a tendency to align with the applied field $\mathbf{H}$. It may be noted that in some applications, particularly those involving an elongated magnetic object as suggested in [13], the alignment of the object relative to the applied force $\vec{F}_{Mag}$ may be considered as an additional constraint on the target. In this study, however, this requirement is disregarded due to the symmetry of the object.

At this point in the model formulation, the type of material utilized for the spherical target must be addressed, as this affects the magnetization $\mathbf{M}$ achieved when the field $\mathbf{H}$ is applied. Primarily two types of materials are proposed for use, paramagnetic or ferromagnetic. The paramagnetic class of material is characterized by a linear relationship between the resulting magnetization and the internal magnetic field of the object. This linear constant is referred to as the magnetic *susceptibility* and is typically denoted as χ_m. The resulting magnetic force on the paramagnetic sphere is [19]

$$\vec{F}_{Mag,PM}(\mathbf{H}) = \frac{2\pi\mu_0 r^3(\chi_m)}{(\chi_m + 3)}\nabla\big(\mathbf{H}\cdot\mathbf{H}\big).$$

The second class of material, ferromagnetic, is characterized by a nonlinear relationship between the applied magnetic field and the magnetization of the object. More specifically, the $M - H$ curve is sigmoidal in shape indicative of these materials reaching a magnetic saturation. Additionally, the sigmoidal $M - H$ curve may also possess a region of hysteresis about the origin depending on the classification of the ferromagnetic material as "hard" or "soft". An example of a typical ferromagnetic $M - H$ curve is available in Figure 2.1. Hard ferromagnetic materials are typified as those in which the $M - H$ hysteresis loop is large and upon removal of the applied H, the sample is permanently magnetized (e.g. a permanent magnet). This is often referred to as magnetic *remanence*. In contrast, soft ferromagnetic materials have a small hysteresis loop and very little residual magnetization upon removal of any applied magnetic field (e.g. transformer iron) [21]. This work will address only the ferromagnetic material type. For the ferromagnetic sphere residing in a medium with permeability close to that of free space, the resultant magnetic force for a given amount of magnetization is

$$\vec{F}_{Mag,FM}(\mathbf{H}) = \frac{4\pi\mu_0 r^3}{3}\left(\frac{M}{H}\mathbf{H}\cdot\nabla\right)\mathbf{H}, \tag{2.4}$$

where M is the magnitude of magnetization and H is the magnitude of the applied field $\mathbf{H}$ at the sphere's location [19]. For the permanently magnetized sphere, M may be treated as a constant. In

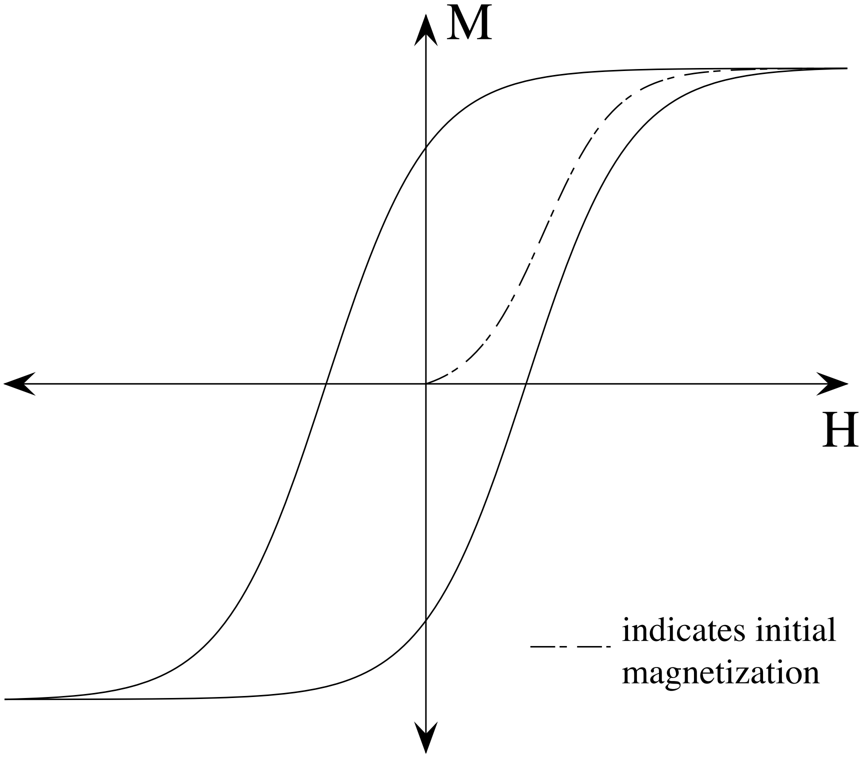

Figure 2.1: Typical $M - H$ curve for a ferromagnetic material.

the Cartesian coordinate system, taking $\mathbf{H} = H_X\hat{\imath} + H_Y\hat{\jmath} + H_Z\hat{k}$ where $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ are unit vectors in the respective coordinate directions, the magnitude of the applied field is given by

$$H = |\mathbf{H}| = \sqrt{H_X^2 + H_Y^2 + H_Z^2}.$$

For the final force on the right hand side of (2.1), it has been readily accepted in the literature [1, 2, 3, 4, 5, 8, 10] that the flow about a spherical particle or object in these applications corresponds to a low Reynolds' number condition (i.e. $Re \ll 1$) and that the drag force may be formulated by applying the Stokes' flow unbounded free stream drag force relation

$$\vec{F}_{Drag} = 6\pi r\mu(\dot{\vec{S}}_{fl} - \dot{\vec{S}}), \tag{2.5}$$

where μ represents the viscosity of the fluid medium, $\dot{\vec{S}}_{fl}$ represents the free stream velocity and the quantity $(\dot{\vec{S}}_{fl} - \dot{\vec{S}})$ denotes the velocity of the moving object relative to the free stream velocity. Indeed there is experimental evidence that, at least for the cases where particles are manipulated in a stagnant or slowly moving medium, this approximation correlates with the observed phenomena [2, 8]. It must be acknowledged, however, that flow scenarios may exist in which the Stokes approximation (2.5) may lose some accuracy. For instance, the Reynolds' number for flow over a sphere is

$$Re = \frac{2\rho_{fl}\dot{S}_{fl}r}{\mu},$$

where ρ_{fl} is the density of the fluid. The extreme smallness of the particle may still allow for a calculation of $Re \ll 1$ even with a significant magnitude of flow velocity around the sphere. If the object is of a larger size (e.g. a microdevice or catheter guide), the result of $Re \ll 1$ may no longer hold, pushing the drag coefficient into a nonlinear function of Re. Additionally, the relative size of the sphere to that of the channel in which it is moving and the geometry of that channel will have an impact on the calculation of the drag coefficient. As an example, consider the system of a sphere moving along the central axis of a cylindrical duct or a square duct. For the cylindrical duct, the drag force on the sphere is given as

$$F_{Drag,cyl} = 6\pi\mu Ur\Big(1 + k\frac{r}{R_0}\Big), \tag{2.6}$$

where U is the velocity of the sphere, R_0 is the radius of the cylindrical duct, and k is a correction coefficient valued at $k = 2.10444$ [22]. For the square duct, the drag force on the sphere is given by

$$F_{Drag,sqr} = 6\pi\mu Ur\Big(1 + k\frac{r}{l}\Big), \tag{2.7}$$

where l is the half width of the square duct and $k = 1.903266$ [22]. Likewise, the wall effects on an object in close proximity to the channel surface have been neglected. These will influence the drag force and often impart lift and moment components on the sphere. Depending on the flow boundary conditions, often the most convenient way to evaluate these effects is through numerical simulation [23, 24]. Additionally, it is well known that fluids in living bodies do not behave in a "Newtonian" fashion. Additional forces may be present, i.e. collisions with other small bodies. Depending on the size of the target object, these dynamics may need to be accounted for. Finally, the condition of turbulent flow in the medium is neglected. Turbulent flow of the fluid medium will lead to significant deviation from the Stokes' approximation. The wall and turbulent flow effects will be ignored for the purposes of this study and left for analysis at a later time. For now, equations (2.2), (2.4), and (2.5) may be used in (2.1) to solve for the motion of the sphere. A component which remains to be addressed is the method to generate the required field $\mathbf{H}$ in (2.4).

The technique utilized in this study for generation of the magnetic field on the sphere is that of an air core electromagnet. It will be assumed that the required motion of the target and hence the required field changes will be slow enough that the magnetic *quasistatic* approximation may be used to ignore any transient effects (e.g. eddy currents, etc.). In this system, stationary electromagnets will be employed. Because of this, multiple coils must be used to facilitate parallel and anti-parallel forces along multiple coordinate directions. This is due to the fact that when the sphere is unrestrained in the rotational sense and allowed to freely align with the field, a single electromagnet will act in an attractive capacity alone. For three dimensional motion an obvious configuration is to make use of six coils arranged in a cubic fashion similar to that of the magnetic implant guidance system by *Stereotaxis, Inc.* [6, 7, 13]. The resulting field from a closed path current density conforms to a solution of Maxwell's field equations which are readily available in [19, 20]. Application of these equations and subsequent reduction of this set to the law of Biot-Savart yields a relation expressing the field $\mathbf{H}$ of the P-th individual coil as a linear function of the current applied to the coil

$$\mathbf{H}_P = j_P[G_X\hat{\imath} + G_Y\hat{\jmath} + G_Z\hat{k}] = j_P\mathbf{G}_P(X,Y,Z), \tag{2.8}$$

where j_P is the applied current, $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ are unit vectors in the respective coordinate directions, and $\mathbf{G}_p$ is a nonlinear vector field multiplier, which is a function of the location of the sphere relative to the coil and the geometry of the coil itself. The combined field from the employment of Q electromagnets may be ascertained through superposition of the individual coil fields

$$\mathbf{H} = [J]^T \begin{bmatrix} \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_Q \end{bmatrix} = [J]^T[\mathbf{G}], \tag{2.9}$$

where $[J]$ is a $Q \times 1$ column vector of the applied currents ($[J] \in \mathbb{R}^Q$) and $\mathbf{G}_P$ is the geometry and position dependent vector field from the P-th coil evaluated in an inertial coordinate system. Using (2.9) the following quantity in (2.4) is derived

$$\begin{aligned}\left(\frac{M}{H}\mathbf{H}\cdot\nabla\right)\mathbf{H} =& \frac{M}{H}[J]^T\left([\mathbf{k}_{FM,X}]\hat{\imath} + [\mathbf{k}_{FM,Y}]\hat{\jmath} + [\mathbf{k}_{FM,Z}]\hat{k}\right)[J] \\ =& \frac{M}{H}[J]^T[\vec{\mathbf{k}}_{FM}][J], \end{aligned} \tag{2.10}$$

where

$$\begin{aligned}
\mathbf{k}_{FM,X} &= [G_X]\left[\frac{\partial G_X}{\partial X}\right]^T + [G_Y]\left[\frac{\partial G_X}{\partial Y}\right]^T + [G_Z]\left[\frac{\partial G_X}{\partial Z}\right]^T, \\
\mathbf{k}_{FM,Y} &= [G_X]\left[\frac{\partial G_Y}{\partial X}\right]^T + [G_Y]\left[\frac{\partial G_Y}{\partial Y}\right]^T + [G_Z]\left[\frac{\partial G_Y}{\partial Z}\right]^T, \\
\mathbf{k}_{FM,Z} &= [G_X]\left[\frac{\partial G_Z}{\partial X}\right]^T + [G_Y]\left[\frac{\partial G_Z}{\partial Y}\right]^T + [G_Z]\left[\frac{\partial G_Z}{\partial Z}\right]^T, \\
H &= \sqrt{[J]^T\Big[[G_X][G_X]^T + [G_Y][G_Y]^T + [G_Z][G_Z]^T\Big][J]},
\end{aligned}$$

$$[G_S] = \begin{bmatrix} G_{S,1} \\ \vdots \\ G_{S,Q} \end{bmatrix}, \tag{2.11a}$$

$$\left[\frac{\partial G_S}{\partial S}\right]^T = \left[\frac{\partial G_{S,1}}{\partial S}, \cdots, \frac{\partial G_{S,Q}}{\partial S}\right]. \tag{2.11b}$$

Note that in equations (2.11a) and (2.11b), $S = X, Y, Z$. All field components and partial derivatives of the field components are evaluated in a global inertial coordinate system. Substituting (2.2), (2.4), (2.5), and (2.10) into (2.1) yields the equation of motion

$$m\ddot{\vec{S}} = (\rho_s - \rho_{fl})V_s\vec{g} + \frac{4\pi\mu_0 r^3}{3}\frac{M}{H}[J]^T[\vec{\mathbf{k}}_{FM}][J] + 6\pi r\mu(\dot{\vec{S}}_{fl} - \dot{\vec{S}}). \tag{2.12}$$

Observe that (2.12) relates a given column vector of input current to the resultant motion of the sphere. Because this system utilizes electromagnet coils, the inductive effects of the electromagnet array must be taken into account. Hence, the equation of motion must be augmented by a suitable description of the electrical dynamics for a complete description of the system. Approximating each coil as an R-L circuit (a resistor in series with an inductor) results in the following relationship between the input voltage and current flow of the electromagnets

$$v_m = R_m j_m + \sum_{n=1}^{Q} L_{m,n}\frac{dj_n}{dt}, \tag{2.13}$$

where Q is the number of coils, $m = 1, 2, \ldots, Q$, $n = 1, 2, \ldots, Q$, v_m denotes the input voltage of coil m, R_m denotes the resistance of coil m, j_m denotes the current in coil m, and dj_n/dt denotes the time rate of change of the current in coil n. Additionally, $L_{m,n}$ signifies the mutual magnetic inductance of coil n on coil m or the self inductance of coil m when $n = m$ [19]. It should be noted that while the self inductance is always positively valued, the sign of the mutual inductance is dependent on the orientation of coil m to n. Observing that (2.13) is a linear system, it may be rewritten in state space form

$$[\dot{J}] = -[L^{-1}][R][J] + [L^{-1}][V] \tag{2.14}$$

where $[V]$ and $[J]$ are $Q \times 1$ column vectors of the voltages and currents in Q coils, $[R]$ is a $Q \times Q$ diagonal matrix of the coil resistances, and $[L]$ is a $Q \times Q$ symmetric matrix of inductance with the diagonal elements being the self inductance terms and the off diagonal elements being the mutual inductance terms. Note that $[L]$ is symmetric because $L_{n,m} = L_{m,n}$ and also note that $[L]$ is invertible [19]. Equation (2.12) is used in concert with (2.14) to provide a full description of the sphere subject to Stokes' flow and a magnetic field generated by Q discrete electromagnets with input voltages $[V]$.

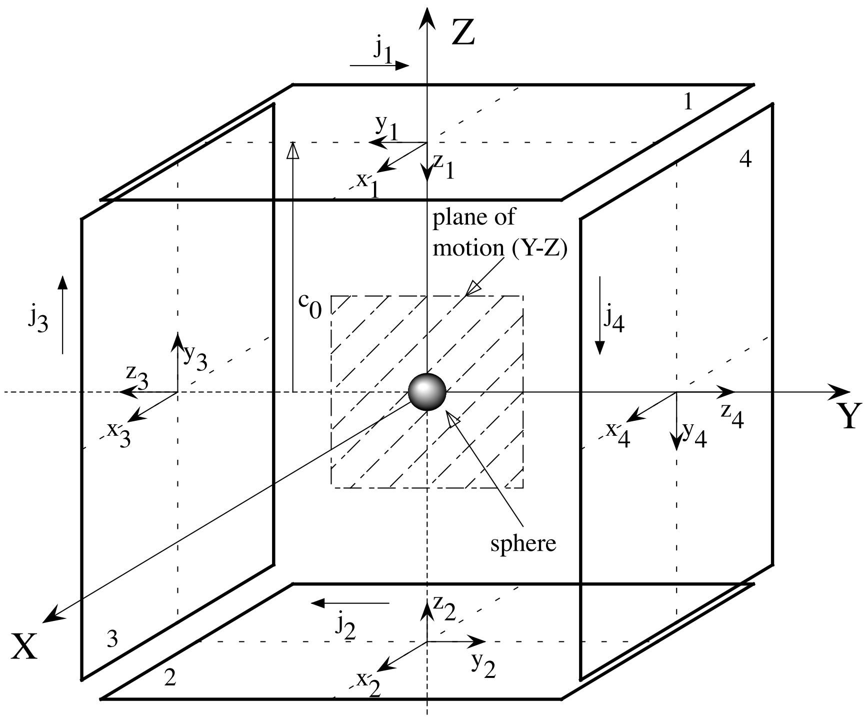

Figure 2.2: Schematic of a four coil arrangement to provide forces for planar motion of a sphere. The individual coil coordinates indicate direction of positive current polarity (right hand rule).

2.2 Inverse Current Solution for Constrained 2-D Motion with a Reduced Coil Set

In an effort to further explore a system using stationary coils to manipulate the position of a magnetized sphere in a fluid medium, this study will examine the two dimensional planar motion case with the sphere constrained to the $Y-Z$ plane (see Fig. 2.2). As previously mentioned, a cubic arrangement of coils can facilitate three dimensional motion of the target sphere [6, 7, 13]. Similarly, a set of four coils arranged on the edges of a plane with the central axes of the coils lying in the plane will facilitate forced motion in that plane. A schematic of this type of system is shown in Fig. 2.2. The inverse current solution entails determining the current trajectories $[J]$ which would cause the sphere to move along a desired motion trajectory $\vec{S}$. Indeed, the inverse current solution may be considered as a more thorough examination of (2.4), acknowledging in this case that $\vec{F}_{Mag}$ is the force required to "offset" the inertial, bias, and drag forces incurred by the sphere moving along $\vec{S}$. It is obvious from (2.4) that $\vec{F}_{Mag,FM}$ is a function linear in $[J]$ but is nonlinear in the components of $\mathbf{H}$, so a direct algebraic solution is not feasible. With this noted, it is evident that in the configuration where the sphere is constrained to a central plane and when the coil configuration to provide both parallel and anti-parallel motion is employed, the system is underdetermined. In other words, a

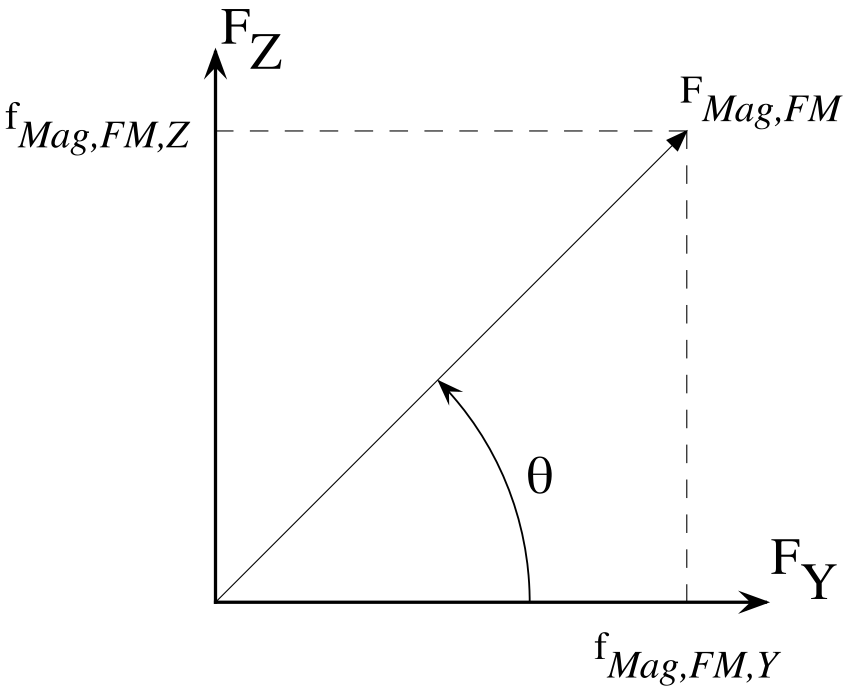

Figure 2.3: Graphical representation of the applied planar magnetic force.

solution in four variables (the coil currents) is to be ascertained from just two equations (the vector components of (2.4) in the Y and Z directions). Due to the fact that electromagnets will act in an attractive capacity alone, one may reduce the number of utilized coils to only those which are necessary for the required force. In the scenario of planar motion utilizing the system in Fig. 2.2, this entails reducing the active coils to that of "corner" combinations (i.e. coils 1&4, 1&3, 2&4, and 2&3), commensurate with utilizing only *adjacent* coils. This will reduce the number of solution variables from four to two and make the system determined. For example, suppose that there is a known required magnetic force $\vec{F}_{Mag,FM}$ in the $Y - Z$ plane. This force may be represented in a cylindrical coordinate system as $\vec{F}_{Mag,FM} = f_{Mag,FM}[\cos(\theta)\hat{\jmath} + \sin(\theta)\hat{k}]$ where $f_{Mag,FM}$ is the force magnitude and θ gives the angular direction of the force measured from the (global) Y axis with the counter clockwise direction taken as positive (see Fig. 2.3). Recognizing that the coils will only attract the sphere, for a required force in the direction of $\theta = 45°$, the necessary coils to achieve this will be numbers 1 and 4 (see Fig. 2.2) with the actual current amplitudes and polarity dependent on the location of the sphere. In other words, the coil combination for a force pointing into a given quadrant will be the coils which bound that quadrant. One may postulate that the corner combinations of 1&4, 1&3, 2&3, and 2&4 correspond to force directions of $0° \leq \theta \leq 90°$, $90° \leq \theta \leq 180°$, $180° \leq \theta \leq 270°$, and $270° \leq \theta \leq 360°$, respectively. However, this quadrant assignment to pairs of coils is approximate due to the nonlinear nature of the attractive magnetic fields.

The components of the required force $\vec{F}_{Mag,FM}$ may be expressed as a function of one another

$$f_{Mag,Z} = \tan(\theta) f_{Mag,Y}, \tag{2.15}$$

where once again θ is the angular direction of the required force measured from the Y axis with the counter clockwise direction taken as positive. As can be seen from (2.4) and (2.12), the Y and Z components of magnetic force for the ferromagnetic sphere are

$$f_{Mag,Y} = \frac{4\pi\mu_0 r^3}{3}\frac{M}{H}[J]^T[\mathbf{k}_{FM,Y}][J], \tag{2.16a}$$

$$f_{Mag,Z} = \frac{4\pi\mu_0 r^3}{3}\frac{M}{H}[J]^T[\mathbf{k}_{FM,Z}][J]. \tag{2.16b}$$

Substituting (2.16a) and (2.16b) into (2.15), rearranging, and then simplifying yields

$$[J]^T[\mathbf{\Omega}][J] = 0, \tag{2.17}$$

where

$$[\mathbf{\Omega}] = \begin{cases} [\mathbf{k}_{FM,Y}], \text{ for } \theta = \frac{\pi}{2} \pm n\pi,\ n = 0, 1, 2.\ldots; \\ [\mathbf{k}_{FM,Z}] - \tan(\theta)[\mathbf{k}_{FM,Y}], \text{ otherwise.} \end{cases}$$

The quantity $[J]$ is a 2×1 vector of the currents in the reduced coil set and the vector fields $[\mathbf{k}_{FM,Y}]$ and $[\mathbf{k}_{FM,Z}]$ are 2×2 matrices containing the $\mathbf{G}$ and $\partial\mathbf{G}/\partial S$ terms of the reduced set in accordance with (2.11a) and (2.11b). The matrices $[\mathbf{k}_{FM,Y}]$ and $[\mathbf{k}_{FM,Z}]$ are given by the quantities in (2.10), however G_S and $\partial G_S/\partial S$ are now 2×1 matrices with the first element corresponding to coil 1 or 2 and the second element corresponding to coil 3 or 4, depending on the coil combination being examined. Once again note that all field components and partial derivatives of the field components are evaluated in the global inertial coordinate system. Equation (2.17) corresponds to a quadratic surface in the J (current) space, centered about the origin. For a current solution other than $J = [0\ 0]^T$ to exist, $[\mathbf{\Omega}]$ must be indefinite or semi-definite [25]. In other words, for a nontrivial solution to exist $[\mathbf{\Omega}]$ cannot be positive or negative definite. Several techniques are available to ascertain the definiteness of $[\mathbf{\Omega}]$. One method involves examining the signs of the principle minors of $[\mathbf{\Omega}]$. If the principle minors are not exclusively < 0 or > 0, then $[\mathbf{\Omega}]$ is semi-definite or indefinite [25]. Alternatively, for the 2×2 matrix $[\mathbf{\Omega}]$ one may examine the product of eigenvalues. If the eigenvalue product is ≤ 0, then $[\mathbf{\Omega}]$ is semi-definite or indefinite. To facilitate the inverse current solution, it is necessary to require the eigenvalues to be real. This may be achieved by forcing symmetry through the relation

$$[J]^T[\mathbf{\Omega}][J] = [J]^T[(\mathbf{\Omega} + \mathbf{\Omega}^T)/2][J] = [J]^T[\mathbf{\Delta}][J].$$

Recognizing that the 2×2 matrix $[\mathbf{\Omega}]$ of the reduced coil set will yield two eigenvalues, the eigenvalues of $[\mathbf{\Delta}]$ will be denoted as λ_1 and λ_2. For a sphere located in the $Y-Z$ plane, a force may be effected on the sphere by any coil combination with an eigenvalue product of $\lambda_1\lambda_2 \leq 0$. As an example, the sign of the eigenvalue products for coil combination 1&4 was examined over a range of $-45^\circ \leq \theta \leq 135^\circ$ for three different locations in the $Y-Z$ plane. The initial system analyzed and demonstrated here consists of four rectangular filamentary current loops with a major dimension of 1 unit, a minor dimension of 0.5 units, and an offset "c" from the global origin (see Fig. 2.2) of 1.5 units. The effective force angle range evaluated at the $Y-Z$ locations of $(0.5, 0.5)$, (0.0), and $(-0.5, -0.5)$ are $-30.51^\circ \leq \theta \leq 120.51^\circ$, $-2.43^\circ \leq \theta \leq 82.43^\circ$, and $10.89^\circ \leq \theta \leq 79.11^\circ$, respectively. These angular ranges indicate the directions of force that coil combination 1&4 is capable of achieving at the given sphere locations. As alluded to earlier, the range of force directions a particular coil combination may impart on the sphere is dependent on the nonlinear field expressions which are functions of the sphere's position as well as coil geometry. The effective range of force direction decreases as the sphere moves away from the electromagnet pair.

To solve for the required currents to achieve the desired (planar) force $\vec{F}_{Mag,FM}$, the orthonormal basis $[\widetilde{\mathbf{Q}}]$ composed of the normalized eigenvectors of $[\mathbf{\Delta}]$ may be used to form the orthogonal transformation

$$[J] = [\widetilde{\mathbf{Q}}][\widetilde{J}].$$

Application of this to

$$[J]^T[\mathbf{\Delta}][J] = 0$$

results now in the diagonal quadratic

$$[\widetilde{J}]^T[\widetilde{\mathbf{D}}][\widetilde{J}] = 0$$

where $[\widetilde{\mathbf{D}}]$ is a diagonal matrix of the eigenvalues of $[\mathbf{\Delta}]$. In the $\widetilde{J}$ space, two line loci may now be formed from the diagonal quadratic

$$\tilde{j}_i^2\lambda_1 + \tilde{j}_{ii}^2\lambda_2 = 0 \begin{cases} \tilde{j}_{ii} = \pm\tilde{j}_i\sqrt{\dfrac{-\lambda_1}{\lambda_2}};\ \lambda_2 \neq 0, & \text{(2.18a)} \\ \tilde{j}_i = \pm\tilde{j}_{ii}\sqrt{\dfrac{-\lambda_2}{\lambda_1}};\ \lambda_1 \neq 0, & \text{(2.18b)} \end{cases}$$

where the subscripts $i = 1, 2$ and $ii = 3, 4$ denote which coil is used in the calculation. As a reminder, recall that $[\mathbf{\Delta}]$ is also a function of which coils are included in the reduced coil set. Note that (2.18) actually provides two separate formulations for the line loci with a caveat on the existence of a non-zero eigenvalue. Utilizing (2.18), the $[\widetilde{J}]$ space current vector may now be rewritten as

$$[\widetilde{J}] = \begin{cases} \tilde{j}_i \begin{bmatrix} 1 \\ \pm\sqrt{\frac{-\lambda_1}{\lambda_2}} \end{bmatrix} = \tilde{j}_i[\xi_1^{\pm}];\ \lambda_2 \neq 0, & (2.19a) \\ \tilde{j}_{ii} \begin{bmatrix} \pm\sqrt{\frac{-\lambda_2}{\lambda_1}} \\ 1 \end{bmatrix} = \tilde{j}_{ii}[\xi_2^{\pm}];\ \lambda_1 \neq 0. & (2.19b) \end{cases}$$

Applying the orthogonal transformation to (2.16a) yields

$$f_{Mag,Y} = \frac{4\pi\mu_0 r^3}{3}\frac{M}{\widetilde{H}}[\widetilde{J}]^T[\widetilde{\mathbf{Q}}]^T[\mathbf{k}_{FM,Y}][\widetilde{\mathbf{Q}}][\widetilde{J}], \tag{2.20}$$

where

$$\widetilde{H} = \sqrt{[\widetilde{J}]^T[\widetilde{\mathbf{Q}}]^T[G][\widetilde{\mathbf{Q}}][\widetilde{J}]},$$

$$[G] = G_X G_X^T + G_Y G_Y^T + G_Z G_Z^T.$$

Substitution of (2.19a) into (2.20), simplifying, and utilizing (2.18a) results in the $[\widetilde{J}]$ solution of

$$\tilde{j}_i = \pm\frac{3f_{Mag,Y}}{4\pi\mu_0 r^3 M}\frac{\sqrt{[\xi_1^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[G][\widetilde{\mathbf{Q}}][\xi_1^{\pm}]}}{[\xi_1^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[\mathbf{k}_{FM,Y}][\widetilde{\mathbf{Q}}][\xi_1^{\pm}]}, \tag{2.21a}$$

$$\tilde{j}_{ii} = \pm\tilde{j}_i\sqrt{\frac{-\lambda_1}{\lambda_2}}. \tag{2.21b}$$

Note that (2.21) must still conform to the caveat of $\lambda_2 \neq 0$. For the condition that $\lambda_2 = 0$, a similar process may be employed utilizing (2.18b) and (2.19b) to provide the solution

$$\tilde{j}_i = \pm\tilde{j}_{ii}\sqrt{\frac{-\lambda_2}{\lambda_1}}, \tag{2.22a}$$

$$\tilde{j}_{ii} = \pm\frac{3f_{Mag,Y}}{4\pi\mu_0 r^3 M}\frac{\sqrt{[\xi_2^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[G][\widetilde{\mathbf{Q}}][\xi_2^{\pm}]}}{[\xi_2^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[\mathbf{k}_{FM,Y}][\widetilde{\mathbf{Q}}][\xi_2^{\pm}]}, \tag{2.22b}$$

where $\lambda_1 \neq 0$. Additionally, solutions may be formulated using (2.16b). This will result in forms similar to (2.21) and (2.22) with $f_{Mag,Y} \rightarrow f_{Mag,Z}$ and $[\mathbf{k}_{FM,Y}] \rightarrow [\mathbf{k}_{FM,Z}]$. To select the proper signs for the solution coordinates, the correct groupings on loci intersections must be observed. These are available in Table 2.1, where in the solution value $\tilde{j}_\delta^{\alpha\beta}$ $(\alpha, \beta = \pm,\ \delta = i, ii)$, $\alpha = \pm$ denotes the sign just to the right of the "=" symbol in (2.21) or (2.22) and $\beta = \pm$ denotes the sign used for $[\xi_\delta^{\beta}]$, as defined in (2.19). It can be seen from (2.21), (2.22), and Table 2.1 that for the planar motion reduced set, there are four different solution coordinates. The coordinate system $\widetilde{J}$ is a transformed system and therefore has no physical meaning. To complete the calculation of the current solution, the $\widetilde{J}$ coordinate solutions must be transformed back to the J space via

$$[J] = [\widetilde{\mathbf{Q}}][\widetilde{J}].$$

Table 2.1: Correct Line Loci Groupings for $\widetilde{J}$ Space Solution Coordinates

solution set	sign
$\widetilde{J}_a$	$(\tilde{\jmath}_i^{++}, \tilde{\jmath}_{ii}^{++})$
$\widetilde{J}_b$	$(\tilde{\jmath}_i^{-+}, \tilde{\jmath}_{ii}^{-+})$
$\widetilde{J}_c$	$(\tilde{\jmath}_i^{+-}, \tilde{\jmath}_{ii}^{--})$
$\widetilde{J}_d$	$(\tilde{\jmath}_i^{--}, \tilde{\jmath}_{ii}^{+-})$

A plot of the intersecting loci indicating the solution coordinates in the J space for the previously mentioned filamentary loop configuration, a sphere with a unit parameter multiplier ($4\pi\mu_0 r^3 M/3 = 1$) located at the global origin, and a required magnetic force of $\vec{F}_{Mag} = 1\hat{\jmath} + 1\hat{k}$ is shown in Fig. 2.4. As can be seen, the four solution coordinates may be ascertained by the intersection of the component loci of equation(s) (2.16) or the intersection of the line loci for equation (2.17) with just one of the component loci of (2.16). Indeed it is the latter technique which yields the results of (2.21) and (2.22). The calculated solution coordinates using (2.21) for solutions J_a, J_b, J_c, and J_d are $(11.208, 11.208)$, $(-11.208, -11.208)$, $(-25.519, 25.519)$, and $(25.519, -25.519)$, respectively. As can be seen, they correlate with the loci intersections.

The inverse current solution presented in this work was derived for the ferromagnetic material type. A similar derivation may be used for the paramagnetic material and will give the same number of solutions, four. However, the solution is of a simpler nature because the shapes of the component loci for a given force are analytically known. In this case the current solution coordinates correspond to intersecting conic sections. Extending this line of thought would result in the inverse current solution for the paramagnetic material being manipulated by a cubic arrangement of coils being the intersection of quadric surfaces. While the inverse current solution given in relations (2.21) and (2.22) do indeed result in coil currents providing the desired magnetic force on the sphere, as will be shown the sign convention given in Table 2.1 is not sufficient to provide a *smooth* current trajectory as a function of time. It is advantageous however for the current profile to be smooth to minimize inductive effects incurred by discontinuities in the desired current profile. Further analysis into this formulation is necessary to achieve this and will be provided in the next section.

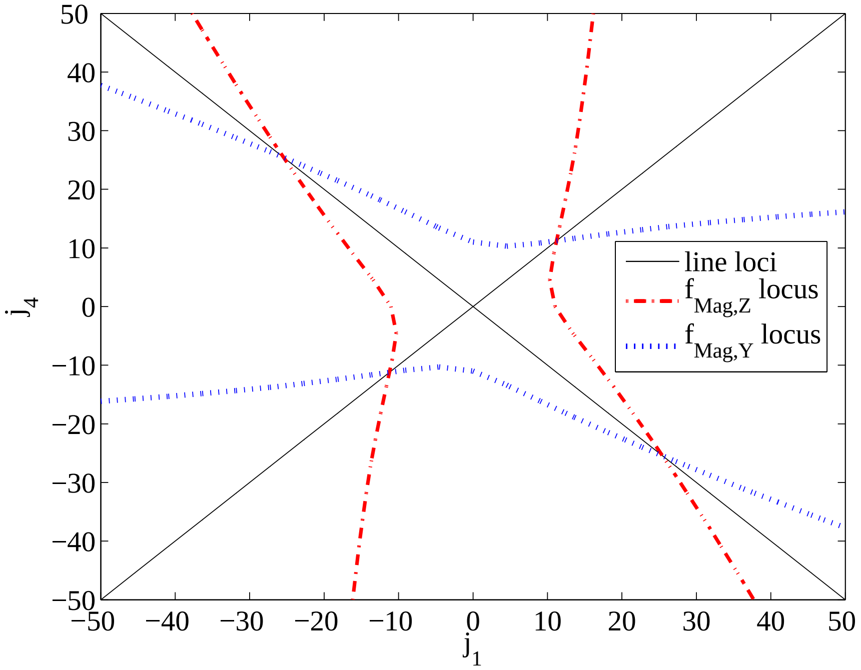

Figure 2.4: Solution loci for the 1&4 coil combination in the J space.

2.3 Current Trajectory Smoothing Technique

To formulate a more detailed analysis of the inverse solution current coordinates using (2.21) and (2.22), a more precise representation of the desired force on the sphere is needed. As previously stated, the required force on the sphere may be regarded as the force required to overcome the inertial, drag, and buoyant forces acting on the sphere. From (2.12), the required force to be provided by the electromagnets for a given trajectory $\vec{S}$ is

$$\vec{F}_{Mag,FM} = m\ddot{\vec{S}} + 6\pi r\mu(\dot{\vec{S}} - \dot{\vec{S}}_{fl}) + (\rho_{fl} - \rho_s)V_s\vec{g}. \tag{2.23}$$

In addition, a trajectory must be examined to fully understand the resultant current profiles calculated using the inverse current formulation presented. For this example, the trajectory to be used was chosen to be a circular path in the $Y - Z$ plane, orbiting the origin with a frequency of $0.1Hz$. This may be defined as

$$\vec{S}(t) = \rho\left[\cos\left(\frac{2\pi}{10}t\right)\hat{j} + \sin\left(\frac{2\pi}{10}t\right)\hat{k}\right], \tag{2.24}$$

where ρ is the radius of the path and t represents time. As can be seen, the first and second time derivatives of (2.24) may be easily evaluated and substituted into (2.23) to calculate the necessary force $\vec{F}_{Mag,FM}$. It will be assumed that the vector acceleration due to gravity, denoted as $\vec{g}$, will be normal to the plane of motion, hence offset by a constraint force and may be subsequently dropped from (2.23). Furthermore, it will also be assumed that this testing apparatus will move the sphere in a *static* fluid, allowing for the dropping of the term $\dot{\vec{S}}_{fl}$ in (2.23) as well. The physical parameters in (2.21) (or (2.22)) and (2.23) which need assignment consist of the mass of the sphere, the radius of the sphere, the viscosity of the fluid medium, the radius of the circular path, and the magnetization of the sphere. In an effort to formulate a more "realistic" example, parameters closer to that of the physical system will be utilized. For this system, $m = 0.1257g$, $r = 1/8$", $\rho = 0.5$", and $\mu = 0.950$. It should be noted that the fluid selected for initial testing is glycerine and the viscosity value is consistent with that found in many fluid property tables. The sphere used in this study is a permanent neodymium rare-earth magnet of grade 40. Accounting for demagnetization effects of the spherical shape results in a permanent magnetization of

$$M = \frac{3B_r}{2\mu_0},$$

where B_r is the remanent flux density in the magnetized sphere [19, 21]. For this study, the remanent flux density will be estimated to have the value of $B_r = 1T$. A selection of this value is consistent with the material type and grade of magnet. Finally, the coil geometry used for this example is that of rectangular coils wound with 12 gauge magnet wire. The coils consist of 21 layers of wire with 18 turns per layer. The inner dimensions of the coils are 0.439" × 0.757", while the distance from each coil face to the global origin is 2.5". A more in depth analysis of this coil configuration will be provided in Chapter 4.

The trajectory chosen for this study is that of a circle orbiting the origin with a constant angular velocity. Because of this, symmetry may be used and only one coil combination need be examined. In other words, the force requirements for each coil combination is the same, hence the current solutions are of the same form. Utilizing this, the combination of coils 1 and 3 will be examined initially. A plot of the four solutions given by (2.21) (see Table 2.1) for coil combination 1 and 3 is shown in Figure 2.5. As can be seen, the solution provided by (2.21) for coils 1 and 3 has two distinct regions in which $[\mathbf{\Delta}_{13}]$ is semi-definite or indefinite. However, if this solution is used to calculate the resultant force generated and then compared to the required force it can be seen that one of the current regions is erroneous. This is demonstrated in Figure 2.6. The incorrect solution regions of both force components seen in Figure 2.6 are artifacts of the mathematical formulation of the inverse solution technique. As is evidenced by Figure 2.6, the requirement of $[\mathbf{\Delta}]$ to be semi-definite or indefinite is not "strong enough" to result in a true solution. The resulting inverse solution must also satisfy the desired *direction* of the applied magnetic force. In other words, $\theta_{calculated}$ must

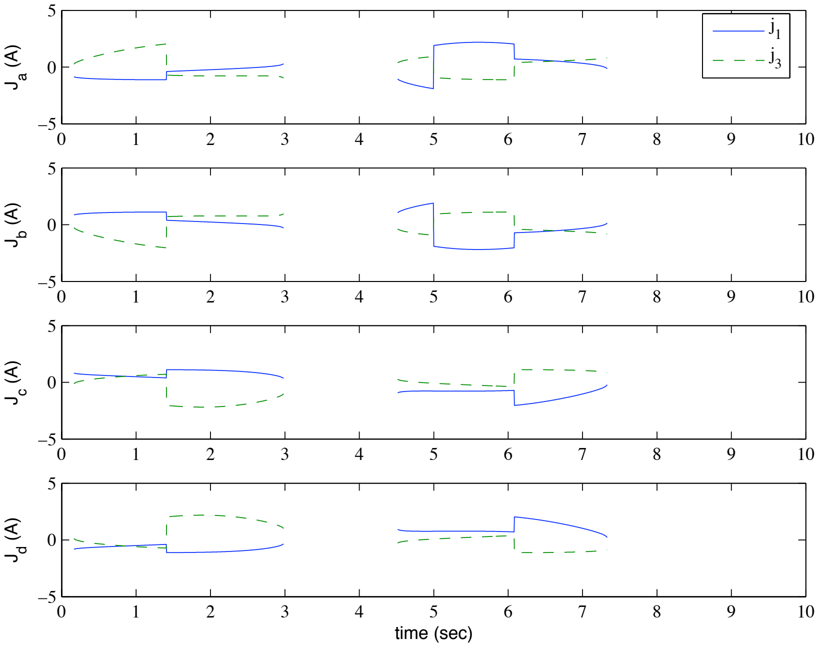

Figure 2.5: Initial current solution for coil combination 1 and 3 during circular trajectory.

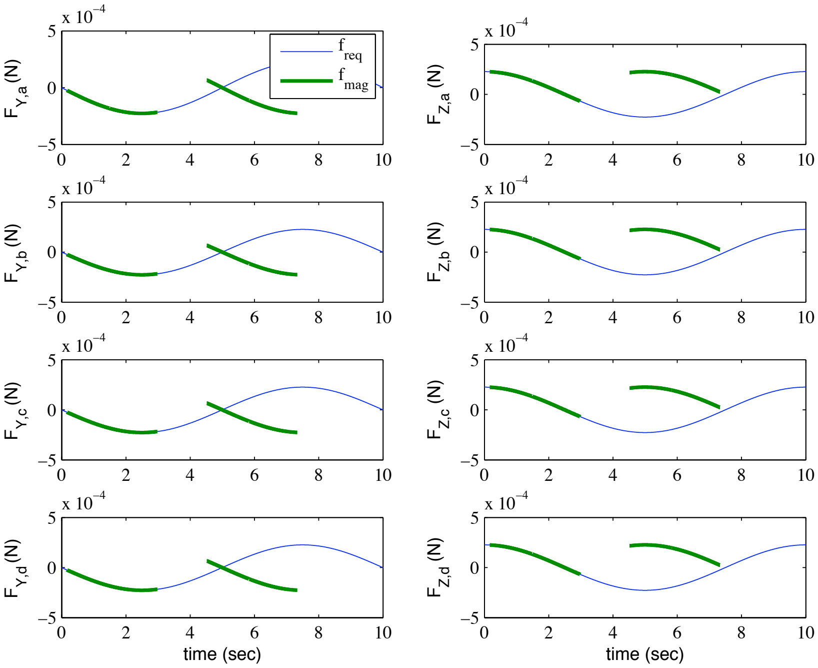

Figure 2.6: Calculated forces using initial current solution compared to required force. Note the regions of erroneous calculated magnetic forces.

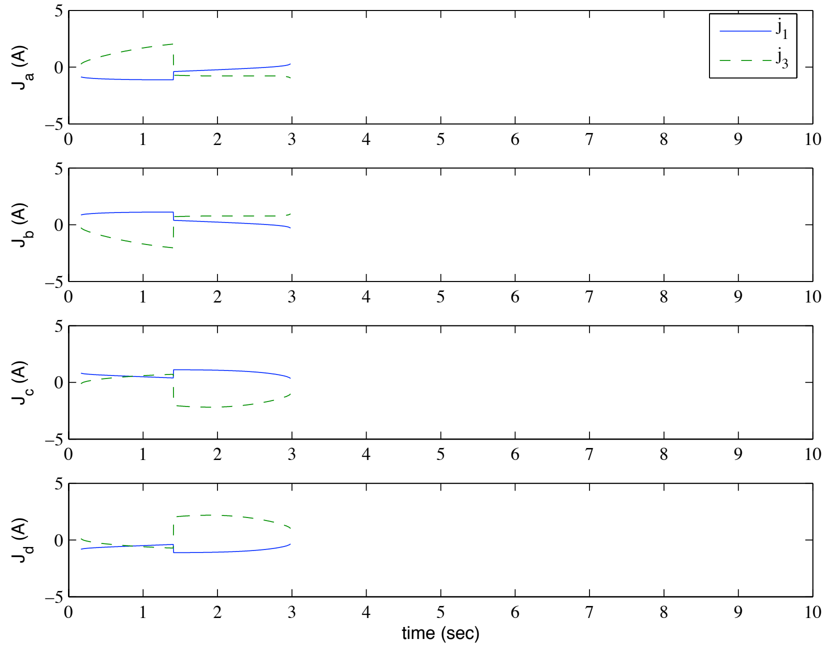

Figure 2.7: Correct current solution for coil combination 1 and 3 during circular trajectory.

equal $\theta_{desired}$. Disregarding the solution regions which do not satisfy this second criteria, Figures 2.7 and 2.8 may be constructed which give the correct solution and calculated resultant forces for coil combination 1 and 3. As can be seen, the region of incorrect force results has vanished from the calculated solution.

Now that a known correct solution with no erroneous sections has been formulated, a second facet of this result may be observed in Figure 2.7, that being the discontinuities present in the current trajectories. Indeed, this aspect when examined at face value may be construed as quite disturbing. While this current trajectory would in fact result in the necessary magnetic force being imparted on the sphere, the ability to achieve this trajectory in an inductive system may or may not be entirely possible. Because of this, it would be desirable for the current trajectories to be "smooth" or continuous. Upon examination of the discontinuity shown in Figure 2.7, it has been observed that the discontinuous point corresponds to the location at which the transformation $[\widetilde{\mathbf{Q}}]$ changes from a rotation to a reflection, or vice versa. It is well known that the particular type of transformation (either rotation or reflection) may be ascertained by examining the sign of the determinant of the transformation [26, 25, 27]. Defining $\mathcal{D} = \det[\widetilde{\mathbf{Q}}]$ and plotting $sgn(\mathcal{D}_{1,3})$ results in Figure 2.9. A comparison of Figures 2.7 and 2.9 shows that the location of the current solution

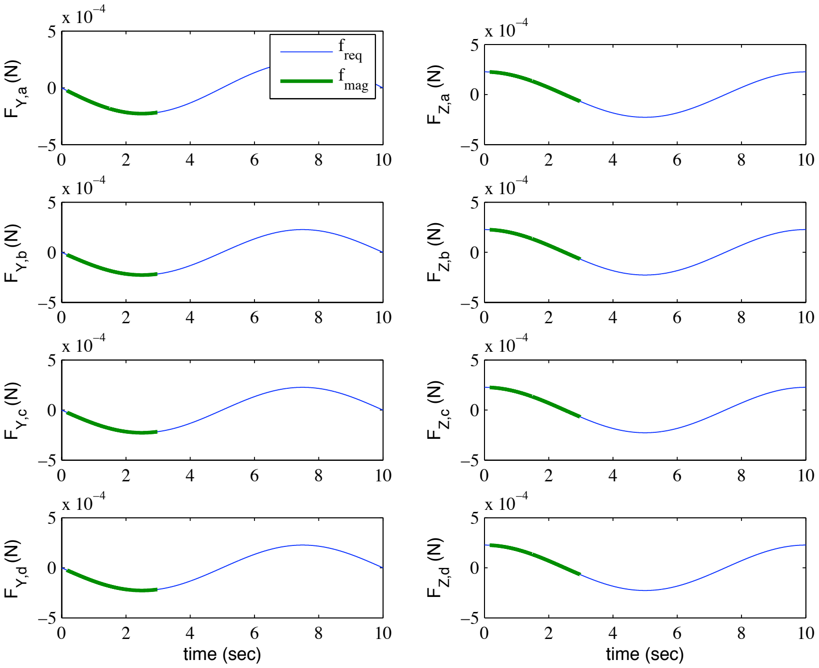

Figure 2.8: Calculated forces using initial current solution compared to required force. Note the regions of erroneous calculated magnetic forces are now absent.

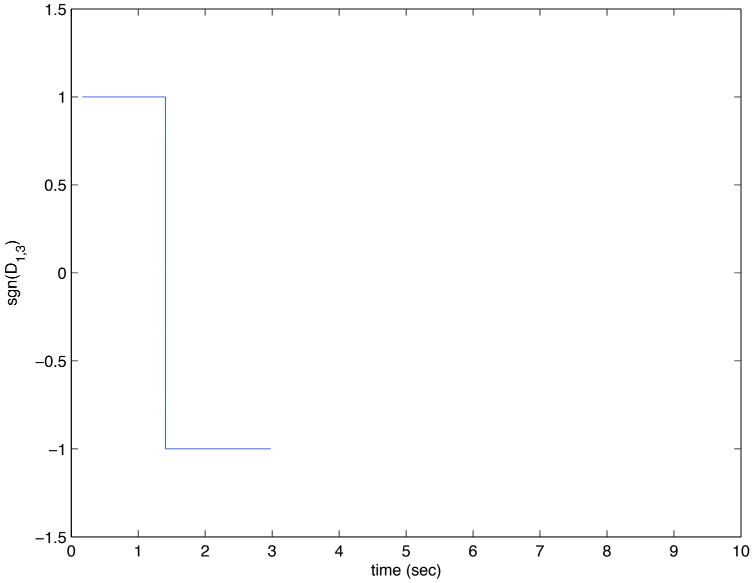

Figure 2.9: A plot of $sgn(\mathcal{D}_{1,3})$, where $\mathcal{D} = \det[\widetilde{\mathbf{Q}}]$. A value of $+1$ indicates that transformation is a *rotation*, while a value of -1 is indicative of a *reflection*.

Table 2.2: Modified Sign Groupings for Smooth $\widetilde{J}_{1,3} = \left(\tilde{j}_1^{\alpha_1\beta_1}, \tilde{j}_3^{\alpha_3\beta_3}\right)$ Solution Coordinates

solution set	α_1	β_1	α_3	β_3
$\widetilde{J}_{a,1,3}$	$+sgn(\mathcal{D}_{1,3})$	$+sgn(\mathcal{D}_{1,3})$	$+$	$+sgn(\mathcal{D}_{1,3})$
$\widetilde{J}_{b,1,3}$	$-sgn(\mathcal{D}_{1,3})$	$+sgn(\mathcal{D}_{1,3})$	$-$	$+sgn(\mathcal{D}_{1,3})$
$\widetilde{J}_{c,1,3}$	$+sgn(\mathcal{D}_{1,3})$	$-sgn(\mathcal{D}_{1,3})$	$-$	$-sgn(\mathcal{D}_{1,3})$
$\widetilde{J}_{d,1,3}$	$-sgn(\mathcal{D}_{1,3})$	$-sgn(\mathcal{D}_{1,3})$	$+$	$-sgn(\mathcal{D}_{1,3})$

discontinuity corresponds to the point at which the transformation $\widetilde{\mathbf{Q}}_{1,3}$ changes from a rotation to a reflection. This dependence may be accounted for by modifying the definitions of the correct solution signs given in Table 2.1. A "smoothed" version of the solution given in Figure 2.7 may be achieved by the solution signs given in Table 2.2. Note that the pattern of ±'s preceding the signum functions matches that seen in the sign pattern given in Table 2.1. Utilizing the signing pattern of Table 2.2, a smooth current solution for coil combination 1 and 3 may be formed. This is shown in Figure 2.10. As can be seen, there are two types of solution, one in which the coil currents of like sign and one in which the coil currents are of opposite sign. Additionally, for each of the two types of solution there are two forms in which the current polarity of each coil is reversed.

As stated earlier, for the generic circular trajectory chosen here, the current trajectory solutions for the other coil combinations will be of similar form to that shown in Figure 2.10. However, the sign convention for providing a *smooth* current trajectory in each coil combination will differ from that given in Table 2.2. For the remaining coil combinations 2&3, 2&4, and 1&4, the resulting inverse solutions using equation (2.21) and the sign convention given in Table 2.1 results in the current trajectories shown in Figures 2.11, 2.12, and 2.13. In addition, the required sign convention needed for each of the combinations to achieve a "smooth" trajectory are given in Tables 2.3, 2.4, and 2.5, with the resulting smooth trajectories also shown in Figures 2.11, 2.12, and 2.13. As can be seen, the discontinuities have been removed by using the sign conventions provided in Tables 2.2, 2.3, 2.4, and 2.5. It should be noted that unlike the solution from equation (2.21), the sign convention given in the tables is not dependent on the solution form used. It has been previously stated that (2.21) is equally valid by substituting $f_{Mag,Y} \rightarrow f_{Mag,Z}$ and $[\mathbf{k}_{FM,Y}] \rightarrow [\mathbf{k}_{FM,Z}]$. In the case of sign convention, $f_{Mag,Y}$ is utilized in the formulation regardless of the use of the Y direction solution versus the Z. Now that smooth current trajectories have been calculated, it is matter of

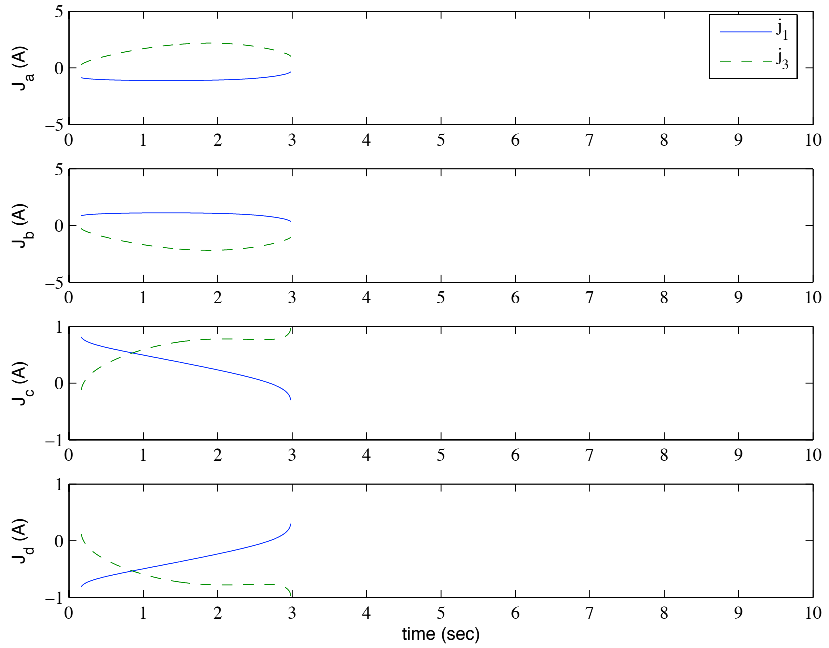

Figure 2.10: Smoothed current solution for coil combination 1 and 3 during circular trajectory.

selecting from the four different solutions provided by (2.21). Any of the four solutions presented will result in the desired force, however one may choose to prefer one solution over another based on the polarity or the total power utilized. For the purposes of experiment design and optimization of this design, symmetry may be leveraged and only analysis of the hardware configuration required to generate one coil combination's solution is necessary. As will be seen in Chapter 4, only the solution for coil combination 1&3 shown in Figure 2.10 will be used in the optimization routine.

2.4 State Space Model and Closed Loop Control Considerations

Implementation of closed loop control on this type of system is challenging and has not been explored in literature yet. Even with the reduced set and the exact calculation of the required currents, it has been shown in this work that for the two dimensional case there will be four possible solutions. For the three dimensional case utilizing six stationary coils, the reduced set would number three and the number of solution coordinates would be eight. Because of this, closed loop control employing the reduced set will be challenging as well. Even with that said, the benefits of being able to exactly calculate the possible solutions are obvious when compared with using optimization methods to search out solutions for an underdetermined system. Feedback control will be further complicated

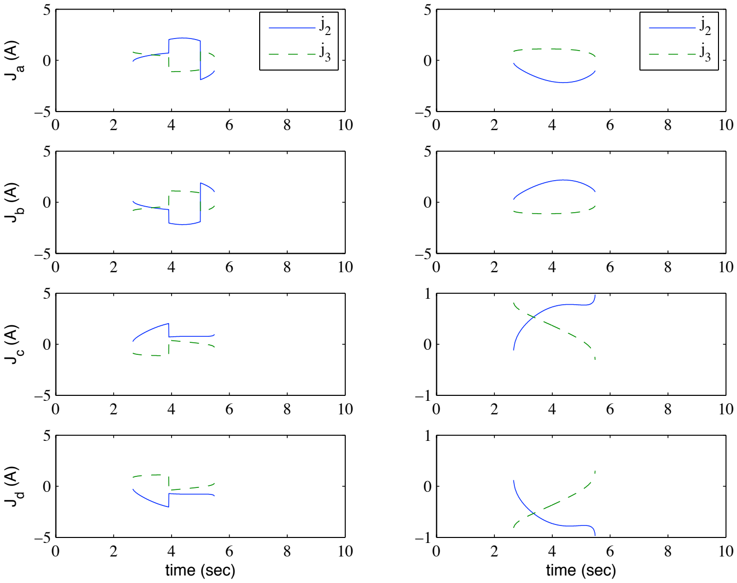

Figure 2.11: Initial and smoothed current solution for coil combination 2 and 3 during circular trajectory.

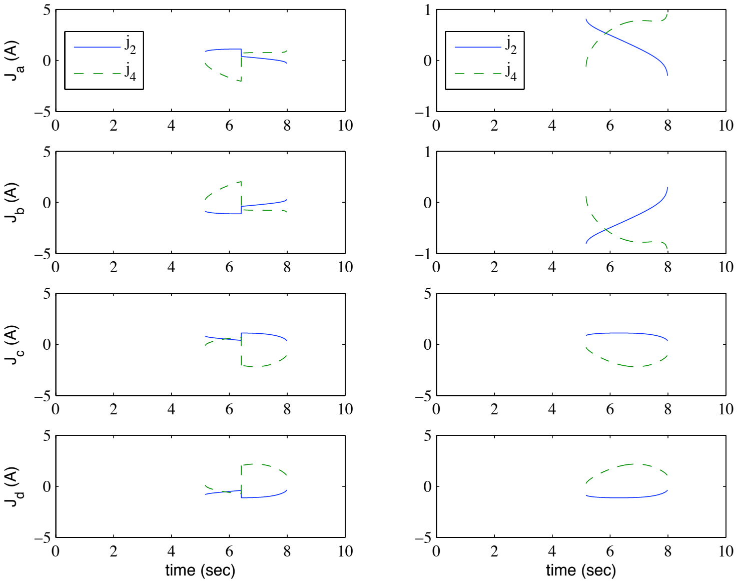

Figure 2.12: Initial and smoothed current solution for coil combination 2 and 4 during circular trajectory.

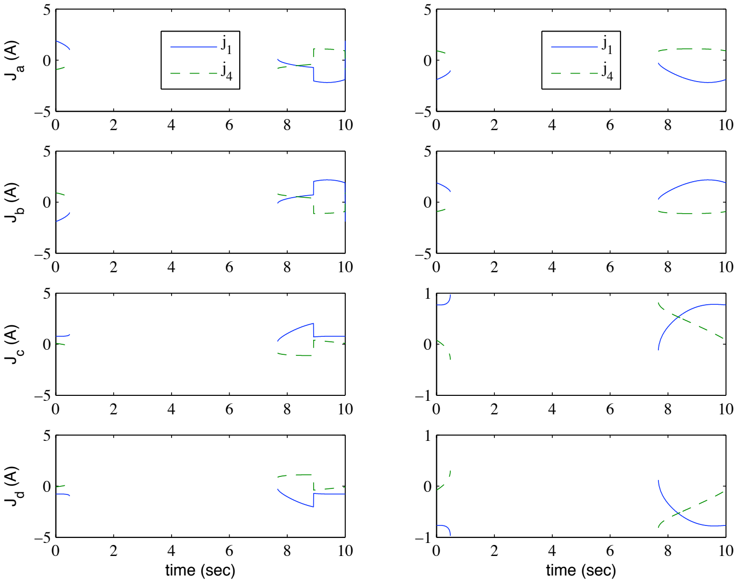

Figure 2.13: Initial and smoothed current solution for coil combination 1 and 4 during circular trajectory.

Table 2.3: Modified Sign Groupings for Smooth $\widetilde{J}_{2,3} = \left(\tilde{j}_2^{\alpha_2\beta_2}, \tilde{j}_3^{\alpha_3\beta_3}\right)$ Solution Coordinates

solution set	α_2	β_2	α_3	β_3
$\widetilde{J}_{a,2,3}$	$+sgn(f_{Mag,Y})$	$+sgn(\mathcal{D}_{2,3}f_{Mag,Y})$	$+sgn(\mathcal{D}_{2,3})$	$+sgn(\mathcal{D}_{2,3}f_{Mag,Y})$
$\widetilde{J}_{b,2,3}$	$-sgn(f_{Mag,Y})$	$+sgn(\mathcal{D}_{2,3}f_{Mag,Y})$	$-sgn(\mathcal{D}_{2,3})$	$+sgn(\mathcal{D}_{2,3}f_{Mag,Y})$
$\widetilde{J}_{c,2,3}$	$+$	$-sgn(\mathcal{D}_{2,3}f_{Mag,Y})$	$-sgn(\mathcal{D}_{2,3}f_{Mag,Y})$	$-sgn(\mathcal{D}_{2,3}f_{Mag,Y})$
$\widetilde{J}_{d,2,3}$	$-$	$-sgn(\mathcal{D}_{2,3}f_{Mag,Y})$	$+sgn(\mathcal{D}_{2,3}f_{Mag,Y})$	$-sgn(\mathcal{D}_{2,3}f_{Mag,Y})$

Table 2.4: Modified Sign Groupings for Smooth $\widetilde{J}_{2,4} = \left(\tilde{j}_2^{\alpha_2\beta_2}, \tilde{j}_4^{\alpha_4\beta_4}\right)$ Solution Coordinates

solution set	α_2	β_2	α_4	β_4
$\widetilde{J}_{a,2,4}$	$+$	$+sgn(\mathcal{D}_{2,4})$	$+sgn(\mathcal{D}_{2,4})$	$+sgn(\mathcal{D}_{2,4})$
$\widetilde{J}_{b,2,4}$	$-$	$+sgn(\mathcal{D}_{2,4})$	$-sgn(\mathcal{D}_{2,4})$	$+sgn(\mathcal{D}_{2,4})$
$\widetilde{J}_{c,2,4}$	$+$	$-sgn(\mathcal{D}_{2,4})$	$-sgn(\mathcal{D}_{2,4})$	$-sgn(\mathcal{D}_{2,4})$
$\widetilde{J}_{d,2,4}$	$-$	$-sgn(\mathcal{D}_{2,4})$	$+sgn(\mathcal{D}_{2,4})$	$-sgn(\mathcal{D}_{2,4})$

Table 2.5: Modified Sign Groupings for Smooth $\widetilde{J}_{1,4} = \left(\tilde{j}_1^{\alpha_1\beta_1}, \tilde{j}_4^{\alpha_4\beta_4}\right)$ Solution Coordinates

solution set	α_1	β_1	α_4	β_4
$\widetilde{J}_{a,1,4}$	$+sgn(\mathcal{D}_{1,4})$	$+sgn(\mathcal{D}_{1,4}f_{Mag,Y})$	$+sgn(f_{Mag,Y})$	$+sgn(\mathcal{D}_{1,4}f_{Mag,Y})$
$\widetilde{J}_{b,1,4}$	$-sgn(\mathcal{D}_{1,4})$	$+sgn(\mathcal{D}_{1,4}f_{Mag,Y})$	$-sgn(f_{Mag,Y})$	$+sgn(\mathcal{D}_{1,4}f_{Mag,Y})$
$\widetilde{J}_{c,1,4}$	$+sgn(\mathcal{D}_{1,4}f_{Mag,Y})$	$-sgn(\mathcal{D}_{1,4}f_{Mag,Y})$	$-$	$-sgn(\mathcal{D}_{1,4}f_{Mag,Y})$
$\widetilde{J}_{d,1,4}$	$-sgn(\mathcal{D}_{1,4}f_{Mag,Y})$	$-sgn(\mathcal{D}_{1,4}f_{Mag,Y})$	$+$	$-sgn(\mathcal{D}_{1,4}f_{Mag,Y})$

for this system due to the inductive effects of the coils. For a given (slow) trajectory, the current solutions to follow that path may be calculated from (2.21) or (2.22). The voltage solutions would then need to be evaluated using (2.13). The method of using a reduced coil set would inherently involve "switching" coils on and off. Inductive effects associated with this type of technique must be accounted and compensated for. One advantage of this system is that for the electrical dynamics, full state feedback is available. Therefore it is conceivable that the switching effects may be taken into account. In the case of large coils with significant inductance parameters, full state feedback of current may prove to be invaluable when dealing with the switching effects.

To synthesize a controller, it is beneficial to first express the entire system model in a state space form. For this system, the state variables will be designated as the sphere's Y and Z positions, the velocity components $\dot{Y}$ and $\dot{Z}$, and currents in the four coils j_1, j_2, j_3, and j_4. The inputs for this system are the coil voltages v_1, v_2, v_3, and v_4. This results in the state vector

$$x = \begin{bmatrix} Y & \dot{Y} & Z & \dot{Z} & j_1 & j_2 & j_3 & j_4 \end{bmatrix}^T, \tag{2.25}$$

and the input vector

$$u = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}^T. \tag{2.26}$$

It will be assumed that the measured outputs are the sphere positions Y and Z, and the four coil currents. As can be seen from (2.12) and (2.13), this system may be represented as the two linear systems of sphere dynamics and electrical dynamics coupled by the nonlinear magnetic force terms. In addition any unknown drag force which deviates from the linear Stokes' static drag may be treated as a disturbance. This leads to the state space representation of

$$\dot{x} = [A]x + f(x) + [B]u + [D], \tag{2.27a}$$

$$y = [C]x, \tag{2.27b}$$

where

$$[A] = \begin{bmatrix} A_s & [\emptyset]_{4\times4} \\ [\emptyset]_{4\times4} & -L^{-1}R \end{bmatrix}, \qquad [B] = \begin{bmatrix} [\emptyset]_{4\times4} \\ L^{-1} \end{bmatrix},$$

$$A_s = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-6\pi r\mu}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{-6\pi r\mu}{m} \end{bmatrix}, \qquad [D] = \begin{bmatrix} 0 \\ d_Y \\ 0 \\ d_Z \\ [\emptyset]_{4\times1} \end{bmatrix},$$

$$[C] = \begin{bmatrix} C_s & [\emptyset]_{4\times 4} \\ [\emptyset]_{4\times 4} & [I]_{4\times 4} \end{bmatrix}, \qquad C_s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$f(x) = \begin{bmatrix} 0 \\ \frac{4\pi\mu_0 r^3}{3m} \frac{M}{\sqrt{x^T \Gamma x}} x^T [\mathbf{K}_{FM,Y}] x \\ 0 \\ \frac{4\pi\mu_0 r^3}{3m} \frac{M}{\sqrt{x^T \Gamma x}} x^T [\mathbf{K}_{FM,Z}] x \\ [\emptyset]_{4\times 1} \end{bmatrix},$$

$$[\mathbf{K}_{FM,S}] = \begin{bmatrix} [\emptyset]_{4\times 4} & [\emptyset]_{4\times 4} \\ [\emptyset]_{4\times 4} & [\mathbf{k}_{FM,S}] \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} [\emptyset]_{4\times 4} & [\emptyset]_{4\times 4} \\ [\emptyset]_{4\times 4} & [G] \end{bmatrix}.$$

Note that in (2.27), $[\emptyset]_{m\times n}$ represents an $m \times n$ matrix of zeros, $[I]_{m\times m}$ represents an $m \times m$ identity matrix, and d_S, $S = X, Y, Z$, represents the unknown component flow disturbances. Equation (2.27) is an eighth order nonlinear state space model which may be used for controller synthesis and design. As can be seen, the system is linear in the input u but contains rather formidable nonlinear terms describing the field forces. With the system now represented in a state space form, the formulation of the field components as they pertain to the coil geometries and the calculation of the mutual inductances must be examined in greater detail. This will be explored in the following chapter.

CHAPTER 3

Electromagnetic Field Description

3.1 The Single Filamentary Loop

While a detailed examination of the planar equation of motion and the relation between all the fields of the individual coils is developed in Chapter 2, the functions describing the resulting field components and gradient field components generated by the rectangular electromagnets is not addressed. Greater insight into this physical system allows for the creation of a model which may then be used in a predictive capacity and allow for design and optimization of an experimental setup for evaluation of this concept. This chapter will expand on the work of Chapter 2 and provide the derivation of a model describing the field generated by elliptical coils wound on a rectangular core, represented as rectangular electromagnets. This geometric simplification allows for the analytic formulation of the descriptive field equations. In addition, the technique used will entail the approximation of the coils as assemblies of discrete filamentary loops [28]. This technique has been selected due to the less difficult analytical treatment when compared to the volumetric coil model [29, 30]. The law of Biot-Savart will be used to calculate the field contribution of each loop.

To develop a relation describing the resulting magnetic field from the quad coil configuration, an analysis of a single filamentary loop will initially be undertaken. The resulting magnetic field for a single (closed) filamentary loop may be ascertained by using the law of Biot-Savart [31],

$$\mathbf{h} = \frac{j}{4\pi} \oint \frac{d\vec{l} \times \vec{r}}{r^3}, \tag{3.1}$$

where $\mathbf{h}$ denotes the magnetic field at a point in 3-D space, j denotes the applied current in the filamentary loop, $d\vec{l}$ denotes a differential element of the loop, $\vec{r}$ denotes the vector from the differential element to the point at which the vector field is being evaluated, and r is the magnitude of $\vec{r}$. In this study, we will first examine a single rectangular filamentary loop as represented in Figure 3.1. Application of (3.1) in a piecewise fashion to the rectangular conductor results in

$$\mathbf{h} = \frac{j}{4\pi} \left[\int_{-b}^{b} \frac{d\vec{l_1} \times \vec{r}_1}{r_1^3} + \int_{-a}^{a} \frac{d\vec{l_2} \times \vec{r}_2}{r_2^3} + \int_{-b}^{b} \frac{d\vec{l_3} \times \vec{r}_3}{r_3^3} + \int_{-a}^{a} \frac{d\vec{l_4} \times \vec{r}_4}{r_4^3} \right], \tag{3.2}$$

where $d\vec{l_m}$ is a differential element of the m side of the rectangular filament, $\vec{r}_m$ is the vector from

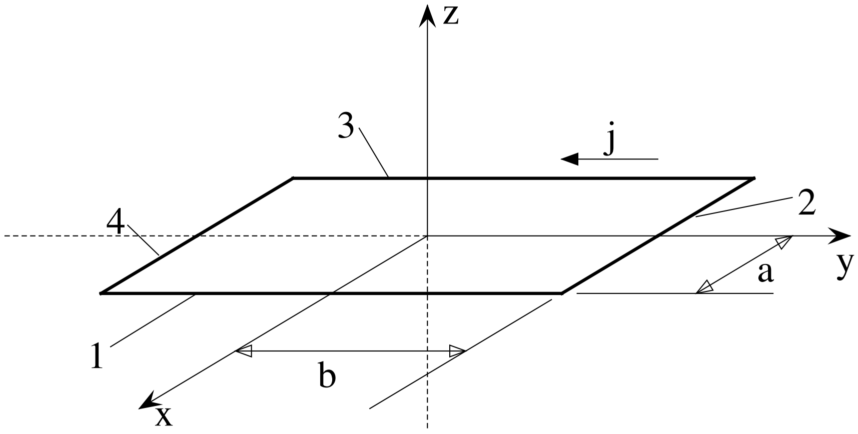

Figure 3.1: Schematic of a single filamentary conducting loop. The local coordinate system is fixed to the center of the loop with the central axis aligned with the z coordinate direction. Positive current flow obeys the right hand rule. The numbers indicate sections of conductor while the letters denote the dimensions of the rectangle.

the m side differential element to the point in 3-D space at which the field is being evaluated, and $m = 1, 2, 3, 4$ indicates a given side of the rectangular filament as shown in Figure 3.1. Note that $\vec{dl_m}$ is in the direction of positive current flow, which in this case obeys the right hand rule. Evaluation of (3.2) in the rectangular coordinate system of Figure 3.1 results in

$$\begin{aligned}\mathbf{h}(x, y, z, j) &= j[g_x(x, y, z, a, b, c)\hat{i} + g_y(x, y, z, a, b, c)\hat{j} + g_z(x, y, z, a, b, c)\hat{k}] \\ &= j\mathbf{g}(x, y, z, a, b, c), \end{aligned} \tag{3.3}$$

where

$$g_x = \frac{z-c}{4\pi}\left[\frac{1}{\varphi_1}\left(\frac{y+b}{\varphi_6} - \frac{y-b}{\varphi_5}\right) - \frac{1}{\varphi_2}\left(\frac{y+b}{\varphi_8} - \frac{y-b}{\varphi_7}\right)\right],$$

$$g_y = \frac{z-c}{4\pi}\left[\frac{1}{\varphi_3}\left(\frac{x+a}{\varphi_7} - \frac{x-a}{\varphi_5}\right) - \frac{1}{\varphi_4}\left(\frac{x+a}{\varphi_8} - \frac{x-a}{\varphi_6}\right)\right],$$

$$\begin{aligned} g_z = \frac{1}{4\pi}\Bigg[&\frac{x+a}{\varphi_1}\left(\frac{y+b}{\varphi_8} - \frac{y-b}{\varphi_7}\right) - \frac{x-a}{\varphi_2}\left(\frac{y+b}{\varphi_6} - \frac{y-b}{\varphi_5}\right) \\ &+ \frac{y+b}{\varphi_3}\left(\frac{x+a}{\varphi_8} - \frac{x-a}{\varphi_6}\right) - \frac{y-b}{\varphi_4}\left(\frac{x+a}{\varphi_7} - \frac{x-a}{\varphi_5}\right)\Bigg],\end{aligned}$$

$$\varphi_1 = (x-a)^2 + (z-c)^2,$$

$$\varphi_2 = (x+a)^2 + (z-c)^2,$$

$$\varphi_3 = (y-b)^2 + (z-c)^2,$$

$$\varphi_4 = (y+b)^2 + (z-c)^2,$$

$$\varphi_5 = [(x-a)^2 + (y-b)^2 + (z-c)^2]^{1/2},$$

$$\varphi_6 = [(x-a)^2 + (y+b)^2 + (z-c)^2]^{1/2},$$

$$\varphi_7 = [(x+a)^2 + (y-b)^2 + (z-c)^2]^{1/2},$$

$$\varphi_8 = [(x+a)^2 + (y+b)^2 + (z-c)^2]^{1/2}.$$

In addition, (x, y, z) is the point in 3-D space at which the field is being evaluated, $\hat{\imath}$, $\hat{\jmath}$ and $\hat{k}$ are unit vectors in the x, y, and z directions, a and b indicate the dimensions of the rectangular loop (see Figure 3.1), and c denotes the offset of the loop from the $x-y$ plane of Figure 3.1. It should be noted that there is a fundamental difference between parameters a, b and c. The parameters a and b denote dimensions, hence $a, b > 0$. However, c denotes an *offset* of the loop from a plane. In this formulation $c > 0$ indicates a *positive* offset, while $c < 0$ indicates a *negative* offset. Additionally it may be observed that if the loop resides in the $x-y$ plane as seen in Figure 3.1, $c = 0$. As can be seen from (3.3), the resultant field from the single rectangular filament is a function of the applied current multiplied by the vector field $\mathbf{g}$. The "$\mathbf{g}$ field" is also seen to be a nonlinear function of the 3-dimensional location of the point at which the vector field $\mathbf{h}$ is being evaluated and the geometry of the filamentary loop.

In addition to the vector field $\mathbf{h}$ (or $\mathbf{g}$), the calculation of the resultant force on a particle requires additional quantities. It is known in [19] that the vector force on a magnetized sphere approximated as a dipole residing in free space is given as

$$\vec{F}_{Mag}(\mathbf{h}) = \frac{4\pi\mu_0 r^3}{3}\Big(\mathbf{M}\cdot\nabla\Big)\mathbf{h}, \tag{3.4}$$

where μ_0 is the permeability of free space, r is the radius of the sphere (not to be confused with the magnitude of the vector $\vec{r}$ given in (3.1)) , $\mathbf{M}$ is the magnetization vector field of the sphere, and $\mathbf{h}$ is the applied magnetic field from the filamentary loop at the location of the particle. It must also be noted that in most fluid media, the spherical object will be unconstrained in the rotational sense resulting in an alignment of the magnetization field with the applied field [32]. Use of this fact along with (3.3) and expansion of the vector quantity $(\mathbf{M}\cdot\nabla)\mathbf{h}$ in (3.4) results in [33]

$$\begin{aligned}\left(\frac{M}{h}\mathbf{h}\cdot\nabla\right)\mathbf{h} = \frac{|j|M}{g}\Bigg[&\left(g_x\frac{\partial g_x}{\partial x} + g_y\frac{\partial g_x}{\partial y} + g_z\frac{\partial g_x}{\partial z}\right)\hat{\imath}\\ &+\left(g_x\frac{\partial g_y}{\partial x} + g_y\frac{\partial g_y}{\partial y} + g_z\frac{\partial g_y}{\partial z}\right)\hat{\jmath}\\ &+\left(g_x\frac{\partial g_z}{\partial x} + g_y\frac{\partial g_z}{\partial y} + g_z\frac{\partial g_z}{\partial z}\right)\hat{k}\Bigg],\end{aligned} \tag{3.5}$$

where M is the magnitude of magnetization and $g = \sqrt{g_x^2 + g_y^2 + g_z^2}$. As can be seen in (3.5), to effectively describe the resultant vector force on the spherical target, not only are the $\mathbf{g}$ field components needed, but the partial derivatives are required as well. It should be noted that for a linear paramagnetic sphere, the force relation may be rewritten as

$$\vec{F}_{Mag,PM}(\mathbf{h}) = \frac{2\pi\mu_0 r^3(\chi_m)}{(\chi_m + 3)}\nabla\big(\mathbf{h}\cdot\mathbf{h}\big),$$

where χ_m is the magnetic susceptibility of the paramagnetic material. The quantity $\nabla(\mathbf{h}\cdot\mathbf{h})$ will then take on the same form as (3.5) with the substitutions of $|j|M/g \to 2j^2$ and $\partial g_m/\partial n \to \partial g_n/\partial m$, $(m, n = x, y, z)$. The partial derivative terms in (3.5) may be directly evaluated from the $\mathbf{g}$ field components given in (3.3). While this has been performed, it proves to be quite tedious and will be omitted here for brevity, however these functions for the rectangular filamentary loop are available in Appendix A.

3.2 Multi-turn Expansion

To evaluate the resultant vector force on a spherical object from a coil with multiple turns, one may utilize the filamentary model of Section 3.1 and the principle of superposition. If each turn of a multi-turn coil is approximated as a single filamentary loop, then the total magnetic field $\mathbf{H}$ at a given point in 3-D space from a multi-turn coil may be given as

$$\mathbf{H} = \mathbf{h}_1 + \mathbf{h}_2 + \cdots + \mathbf{h}_q = \sum_{p=1}^{q} \mathbf{h}_p, \tag{3.6}$$

where q is the total number of turns for the coil. Equation (3.6) is only valid with adherence to a few important caveats. First, the individual fields $\mathbf{h}_1, \cdots, \mathbf{h}_q$ are given in the *same* coordinate system. For the principle of superposition to hold, the resultant field from each turn must be presented in terms of the same basis. Second, the point in 3-D space where the field is being evaluated may not be coincident with a point on a filamentary loop. As can be seen from (3.2), application of the law of Biot-Savart for the coincident loop results in an undefined integrand. This is not expected to be an issue due to the fact that no trajectory of the object can be coincident with the coil volume, hence points coincident with the filamentary loops are not needed in this analysis. Application of (3.3) to (3.6) results in

$$\mathbf{H} = j\mathbf{g}_1 + j\mathbf{g}_2 + \cdots + j\mathbf{g}_q = j\sum_{p=1}^{q} \mathbf{g}_p = j\mathbf{G}, \tag{3.7}$$

where $\mathbf{g}_p$ is the geometry dependent vector field of the p-th loop. Note that in (3.7) the total vector field $\mathbf{H}$ is a function of the current j in each turn multiplied by the vector field $\mathbf{G}$ which is a function

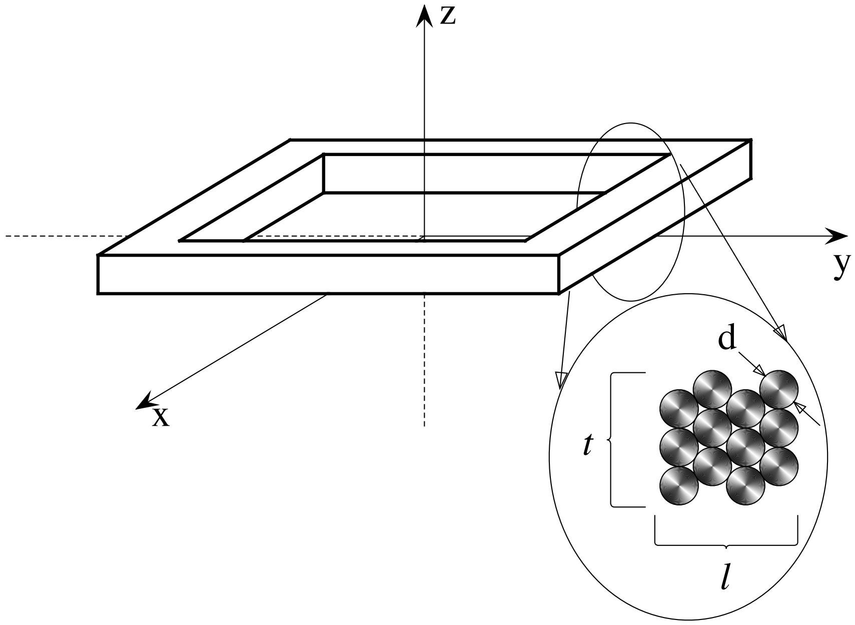

Figure 3.2: Schematic of a rectangular coil with a "perfect wind" packing pattern. The diameter of the wire is denoted as "*d*", while the number of turns/layer and number of layers are denoted t and l, respectively.

of the number of current loops, the geometry of the current loops, and the location of the loops in 3-D space. To formulate the **G** field expression describing the multi-turn coil, use may be made of the expressions in (3.3). As can be seen, (3.3) gives the single loop **g** field components in the coordinate system of Figure 3.1. This same description may be used for each of the the filamentary loops comprising the approximation of the multi-turn coil, provided that the proper values of a, b, and c are used for the loop. Thus to describe the magnetic vector field from a multi-turn coil by using the approximate technique of representing the turns as a finite number of current loops, one must only supply the proper values of a, b, and c in equation (3.3) for each p-th loop and then apply equation (3.7).

If it is assumed that each of the wound coil's turns may be represented by one concentric loop and that the coil is wound with "perfect" packing, then a schematic of this system may be formed in Figure 3.2. The representation of a wound coil as a grouping of discrete loops is indeed an approximation of the true physical system. In reality, the coil consists of a single conductor wrapped multiple times about a given geometry and the previous "wrappings". In fact, for a wound coil to be a continuous conductor there must be a region of wire "crossover" which is obviously being neglected in this technique. Additionally, the wound path is in actuality a *spiral*. This results in an angular component of the current path being present that is not represented by the concentric current loops. This angle can be envisioned as one which would be between the current path and the $x - y$ plane of Figure 3.2. Finally, the ability to fabricate a "rectangular" coil may be quite elusive. While the innermost layers of wire may be very close to a rectangle due to the shape of the bobbin on which the wire is wound, inevitably some rounding of the coil corners will occur. This fact is also being neglected for the analytical model formulation of this study. As is hinted by Figure 3.2, four parameters are necessary to establish the "perfectly" wound, concentric loop model. These consist of the number of wound layers l, the number of turns per layer t, the diameter d of the wire used, and the z direction in which the coil is built. For the representation in Figure 3.2, the coil is built in the positive z direction of the coil's coordinate system. The parameter t may be thought of as the amount of coil build-up in the z direction while l is the amount of coil build-up in the transverse $x - y$ directions. Note that with the cross sectional wind geometry chosen in Figure 3.2, all layers have the same number of turns per layer.

To establish an expression for $\mathbf{g}_p$, $p = 1, \ldots, q$, in (3.7), the following relations may be used:

$$\mathbf{G} = \sum_{p=1}^{q} \mathbf{g}_p = \sum_{\lambda=1}^{l} \sum_{\tau=1}^{t} \mathbf{g}(x, y, z, a_\lambda, b_\lambda, c_\tau^\pm), \tag{3.8}$$

where

$$a_\lambda = a_0 + \frac{d\sqrt{3}}{2}(\lambda - 1)$$

$$b_\lambda = b_0 + \frac{d\sqrt{3}}{2}(\lambda - 1)$$

$$c_\tau^\pm = \begin{cases} \pm[d(\tau - 1)], & \text{for } \lambda = 1, 3, 5, \ldots \text{ (odd);} \\ \pm[\frac{d}{2} + d(\tau - 1)], & \text{for } \lambda = 2, 4, 6, \ldots \text{ (even).} \end{cases}$$

In equation (3.8), a_0 and b_0 denote the initial positions of the innermost layer of the rectangular coil, and the sign of $c_\tau^\pm$ indicates the build direction along the $\pm z$ axis as indicated in Figure 3.2. Note that $q = tl$ and also that this formulation begins the coil in the local $x - y$ plane, building out from there. An analogous expression may be formed for the partial derivative terms needed in equation (3.5), available here as

$$\frac{\partial G_m}{\partial n} = \sum_{p=1}^{q} \frac{\partial g_{m,p}}{\partial n} = \sum_{\lambda=1}^{l} \sum_{\tau=1}^{t} \frac{\partial g_m(x, y, z, a_\lambda, b_\lambda, c_\tau^\pm)}{\partial n}, \tag{3.9}$$

where $m, n = x, y, z$. Expressions (3.8) and (3.9) may now be used to calculate the $\mathbf{G}$ field and partial derivatives of the $\mathbf{G}$ field for a given coil of t turns per layer and l layers ($q = tl$ total turns) at a given point in 3-D space in the local coil coordinate system of Figure 3.2. The configuration proposed to facilitate the particle motion consists of four coils, hence an expansion of the single coil model to the quad-coil model is necessary.

3.3 Quad Coil Model

The expressions derived in Section 3.2 may also be used to describe the multi-coil configuration. A schematic of this system is available in Figure 2.2. Once again using the principle of superposition, the total field resulting from an array of multi-turn coils may be given as

$$\mathbf{H} = \mathbf{H}_1 + \mathbf{H}_2 + \cdots + \mathbf{H}_Q = \sum_{P=1}^{Q} \mathbf{H}_P,$$

where Q represents the number of coils in the array. Note that in system of Figure 2.2, $Q = 4$. Representing the total field in terms of the $\mathbf{G}$ field and current of each coil in the array results in

$$\mathbf{H} = \sum_{P=1}^{Q} j_P \mathbf{G}_P = j_1 \mathbf{G}_1 + \cdots + j_Q \mathbf{G}_Q = [J]^T \begin{bmatrix} \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_Q \end{bmatrix} = [J]^T [\mathbf{G}],$$

where the individual fields $\mathbf{G}_P$ must be given in a common basis. As has been shown in [32] and Chapter 2, the vector field $(\mathbf{M} \cdot \nabla)\mathbf{H}$ proportional to the force resulting on a magnetized sphere for an array of coils is derived as

$$\begin{aligned}\left(\frac{M}{H}\mathbf{H} \cdot \nabla\right)\mathbf{H} = \frac{M[J]^T}{H} \Bigg[&\left([G_X] \left[\frac{\partial G_X}{\partial X}\right]^T + [G_Y] \left[\frac{\partial G_X}{\partial Y}\right]^T + [G_Z] \left[\frac{\partial G_X}{\partial Z}\right]^T \right) \hat{\imath} \\ &+ \left([G_X] \left[\frac{\partial G_Y}{\partial X}\right]^T + [G_Y] \left[\frac{\partial G_Y}{\partial Y}\right]^T + [G_Z] \left[\frac{\partial G_Y}{\partial Z}\right]^T \right) \hat{\jmath} \\ &+ \left([G_X] \left[\frac{\partial G_Z}{\partial X}\right]^T + [G_Y] \left[\frac{\partial G_Z}{\partial Y}\right]^T + [G_Z] \left[\frac{\partial G_Z}{\partial Z}\right]^T \right) \hat{k} \Bigg] [J], \end{aligned} \tag{3.10}$$

where:

$$H = \sqrt{[J]^T \Big[[G_X][G_X]^T + [G_Y][G_Y]^T + [G_Z][G_Z]^T \Big] [J]},$$

$$[G_S] = \begin{bmatrix} G_{S,1} \\ \vdots \\ G_{S,Q} \end{bmatrix},$$

$$\left[\frac{\partial G_S}{\partial S}\right]^T = \begin{bmatrix} \frac{\partial G_{S,1}}{\partial S} & \cdots & \frac{\partial G_{S,Q}}{\partial S} \end{bmatrix}.$$

As previously stated, the field components and partial derivatives of field components in (3.10) are given in the global coordinate system $S = X, Y, Z$. To express the fields in the global coordinate system, homogeneous coordinate transformations may be employed on equations (3.8) and (3.9) to transform the calculated field results from the local coil coordinates to the global system [26].

To utilize the representation (3.10) and transform to the proper location in the global coordinate system, the individual local coordinate systems for each coil must be established. The local coil coordinate locations are available in Figure 2.2. It is important to note that the local coordinates given in Figure 2.2 are positioned in the plane of the turns closest to the global origin. Additionally, the correct build directions in the local systems must be chosen, and hence the proper sign of $c_\tau^\pm$ in (3.8) and (3.9) selected. For this system it is determined by the orientation of the local coordinate systems; coils one and two utilize c_τ^- and coils three and four utilize c_τ^+. As can be seen, local coordinate system 2 is translated and not rotated relative to the global system. Therefore the homogenous coordinate transformation may be given as

$$[A_2] = \begin{bmatrix} [R_2] & [d_2] \\ \emptyset & 1 \end{bmatrix}, \tag{3.11}$$

where

$$[R_2] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad [d_2]^T = \begin{bmatrix} 0 & 0 & -c_0 \end{bmatrix},$$

the quantity $[R_2]$ denotes the rotation component of the transform, $[d_2]$ denotes the translation component of the transform, and c_0 is the *distance* from the global origin to the $x - y$ plane of the closest turns of the coil to the global origin. Note that because coil 2 is only translated in relation to the global coordinates, the rotation matrix is effectively an identity matrix. For the other three coils this will not be the case. The coordinate transforms for coils 1, 3, and 4 may be given as

$$[A_1] = \begin{bmatrix} [R_1] & [d_1] \\ \emptyset & 1 \end{bmatrix}, \tag{3.12}$$

$$[A_3] = \begin{bmatrix} [R_3] & [d_3] \\ \emptyset & 1 \end{bmatrix}, \tag{3.13}$$

and

$$[A_4] = \begin{bmatrix} [R_4] & [d_4] \\ \emptyset & 1 \end{bmatrix}, \tag{3.14}$$

where:

$$[R_1] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \qquad [d_1]^T = \begin{bmatrix} 0 & 0 & c_0 \end{bmatrix} = -[d_2]^T,$$

$$[R_3] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad [d_3]^T = \begin{bmatrix} 0 & -c_0 & 0 \end{bmatrix} = -[d_4]^T,$$

$$[R_4] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}.$$

Equation (3.8) may now be used with the relations in (3.11), (3.12), (3.13), and (3.14) to formulate the **G** field components (given in the global coordinate system) of equation (3.10). This may be accomplished by first representing global point at which the sphere is located in the local basis of each coil, calculating each coil's local **G** field and $\partial\mathbf{G}/\partial s$ contributions (where $s = x, y, z$), and finally transforming those results *back* into the global coordinate system for use in (3.10).

To transform the point $S = (X, Y, Z)$ to a point $s_P = (x_P, y_P, z_P)$ in the P-th coil's local coordinate system, it is beneficial to first present the point in its augmented homogeneous representation [26]

$$[\widetilde{S}]^T = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix},$$
$$[\widetilde{s}_P]^T = \begin{bmatrix} x_P & y_P & z_P & 1 \end{bmatrix}.$$

Transformation of the global point may then be achieved by

$$[\widetilde{s}_P] = [A_P]^{-1}[\widetilde{S}]. \tag{3.15}$$

Once the global coordinate is transformed into the individual coordinate systems, equations (3.3), (3.8), and (3.9) may be used to calculate the local field and spatial partial derivatives of the field. Following this, the local coil field contributions may be transformed back to the global coordinate system by utilizing

$$[\mathbf{G}_P] = [R_P][\mathbb{G}_P], \tag{3.16}$$

where

$$[\mathbf{G}_P]^T = \begin{bmatrix} G_{X,P} & G_{Y,P} & G_{Z,P} \end{bmatrix},$$
$$[\mathbb{G}_P]^T = \begin{bmatrix} G_{x,P} & G_{y,P} & G_{z,P} \end{bmatrix}.$$

The transform of the spatial partial derivatives of $\mathbb{G}_P$ is not as elegant as (3.16). This vector field may be transformed back into the global coordinates using

$$\begin{aligned}
[\nabla_S G_{X,1}] &= [R_1][\nabla_{s,1} G_{x,1}] \\
[\nabla_S G_{Y,1}] &= -[R_1][\nabla_{s,1} G_{y,1}] \\
[\nabla_S G_{Z,1}] &= -[R_1][\nabla_{s,1} G_{z,1}],
\end{aligned} \tag{3.17}$$

$$\begin{aligned}
[\nabla_S G_{X,2}] &= [R_2][\nabla_{s,1} G_{x,2}] \\
[\nabla_S G_{Y,2}] &= [R_2][\nabla_{s,1} G_{y,2}] \\
[\nabla_S G_{Z,2}] &= [R_2][\nabla_{s,1} G_{z,2}],
\end{aligned} \tag{3.18}$$

$$\begin{aligned}
[\nabla_S G_{X,3}] &= [R_3][\nabla_{s,1} G_{x,3}] \\
[\nabla_S G_{Y,3}] &= -[R_3][\nabla_{s,1} G_{y,3}] \\
[\nabla_S G_{Z,3}] &= [R_3][\nabla_{s,1} G_{z,3}],
\end{aligned} \tag{3.19}$$

and

$$\begin{aligned}
[\nabla_S G_{X,4}] &= [R_4][\nabla_{s,1} G_{x,4}] \\
[\nabla_S G_{Y,4}] &= [R_4][\nabla_{s,1} G_{y,4}] \\
[\nabla_S G_{Z,4}] &= -[R_4][\nabla_{s,1} G_{z,4}],
\end{aligned} \tag{3.20}$$

where ∇_S and $\nabla_{s,P}$ denote the gradient operation in the global coordinates S and local coordinates s, P ($P = 1, 2, 3, 4$), respectively. Once the $\mathbf{G}$ field and partial spatial derivatives of $\mathbf{G}$ have been calculated, (3.10) may be utilized to calculate the vector field proportional to the force on the magnetized sphere, provided that the applied current vector $[J]$ is given as well.

3.4 Coil Resistance and Inductance

As has been shown in [32] and Chapter 2, the resistance, self inductance, and mutual inductance are needed for an accurate representation of the electrical dynamics of the system shown in Figure (2.2). For a volumetric coil residing in free space with the resulting vector magnetic potential denoted as $\vec{\mathbb{A}}$, the inductance L may be calculated analytically by

$$L = \frac{1}{j^2} \int_{\mathcal{V}} \vec{\mathbb{A}} \cdot \vec{\mathbf{j}} d\mathcal{V}, \tag{3.21}$$

where

$$\vec{\mathbb{A}} = \frac{1}{4\pi} \int_{\mathcal{V}} \frac{\vec{\mathbf{j}}}{\mathcal{R}} d\mathcal{V},$$

$\vec{\mathbf{j}}$ is the *current density* in the coil, $\mathcal{V}$ is the volume of the coil, and $\mathcal{R}$ is the distance from the volume element $d\mathcal{V}$ to the point at which the vector potential is being evaluated [19]. As can be seen, (3.21) is the result of *two* triple integrals and may prove to be quite challenging. Keeping with the spirit of using a finite loop representation of the field result, the inductance may be formulated in a like manner. Inductance of a coil is defined as

$$L = \frac{q\phi}{j},$$

where ϕ is the total magnetic flux linked by the each turn, q is the total number of turns, and j is the current in the coil [31]. Expanding the definition to include the effect between *two* coils, an examination on a turn by turn basis yields a new definition

$$L_{M,N} = \frac{\Phi_{M,N}}{j_N} = \frac{\sum\limits_{p=1}^{q} \phi_{M,N,p}}{j_N}, \tag{3.22}$$

where $\Phi_{M,N}$ is the total magnetic flux generated by the N-th coil linked by the M-th coil, j_N is the current flowing in the N-th coil, and $\phi_{M,N,p}$ is the total magnetic flux generated by the N-th coil linked by the p-th filamentary loop of the M-th coil, summed up for q total loops. In this system, the self inductance of a coil may be defined as $L_{M,M}$ and the mutual inductance between two coils as $L_{M,N}$ where $M, N = 1, 2, 3, 4$ and $M \neq N$. It is well known that $L_{M,N} = L_{N,M}$ [31, 19, 20]. Through symmetry, it may also be seen that for the quad coil system of Figure 2.2, $L_{1,3} = L_{1,4} = L_{2,3} = L_{2,4}$ and $L_{1,2} = L_{3,4}$. It is obvious that for four identical coils $L_{1,1} = L_{2,2} = L_{3,3} = L_{4,4}$. From these symmetry observations, it is evident that only three formulations are needed to summarize all of the self and mutual inductance values for the system in Figure 2.2. Calculation of $L_{1,1}$, $L_{3,1}$, and $L_{2,1}$ will facilitate a full description of the system inductance.

For the first treatment, the self inductance ($M = N$) will be examined and the indices M and N will be temporarily dropped. In a general sense, the total flux linked by the p-th loop of a coil may be defined as the surface integral

$$\phi_p = \int_{\partial S_p} \mathbf{B} \cdot d\vec{S_p}, \tag{3.23}$$

where $\mathbf{B}$ is the *total flux density* vector field and ∂S_p is the surface outlined by the p-th loop. For the single filamentary loop of Figure 3.1, the total flux linked may be rewritten as

$$\phi = \int_{\partial A} B_z dA, \tag{3.24}$$

where B_z is the component of the total flux density in the z direction linked by the loop and ∂A is the area outlined by the loop. One may suggest now that (3.24) be solved directly using the relation

$\mathbf{B} = \mu_0 \mathbf{H} \implies B_z = \mu_0 H_z$. For the multi-turn coil with q loops, the z component of the total flux density may be formulated from (3.6) and (3.7) as

$$B_z = \mu_0 H_z = j\mu_0 G_z = j\mu_0 \sum_{p=1}^{q} g_{z,p}.$$

This may be used with (3.24) and (3.8) to formulate the total flux (produced by a coil having $q = tl$ turns) linked by the same coil's $\tilde{p}$-th loop as

$$\phi_{\tilde{p}} = j\mu_0 \sum_{\lambda=1}^{l} \sum_{\tau=1}^{t} \int_{-b_{\tilde{p}}}^{b_{\tilde{p}}} \int_{-a_{\tilde{p}}}^{a_{\tilde{p}}} g_{z,p}(x, y, z_{\tilde{p}}, a_\lambda, b_\lambda, c_\tau^+) dx dy, \tag{3.25}$$

where $z_{\tilde{p}}$ is the z location of the $\tilde{p}$-th loop in the local coordinate system of Figure 3.2. It must also be noted that for the self inductance formulation, a_λ, b_λ, and c_τ^+ obey the definitions provided in (3.8).

Substitution of the relation for g_z from (3.3) into the integrand of (3.25) makes for a demanding exercise in calculus. The evaluation of the total flux can be simplified by utilizing the magnetic vector potential. For the single filamentary loop of Figure 3.1, the magnetic vector potential may be formulated as

$$\vec{\mathbb{A}} = \frac{j}{4\pi} \oint \frac{d\vec{l}}{r} = j\mathbf{A}, \tag{3.26}$$

where $\mathbf{A}$ may be thought of as a *geometric* vector potential. The magnetic vector potential is related to the magnetic field through the relation $\mathbf{H} = \nabla \times \vec{\mathbb{A}}$, therefore the flux density may be related to the geometric vector potential through $\mathbf{B} = j\mu_0(\nabla \times \mathbf{A})$. This may be substituted into (3.23), resulting in a formula describing the total flux linked by the p-th loop in terms of total vector potential

$$\phi_p = j\mu_0 \int_{\partial S_p} (\nabla \times \mathbf{A}) \cdot d\vec{S_p}. \tag{3.27}$$

The integration of (3.27) is no less complicated than (3.23), however (3.27) may now be reduced from a surface integral to a *path* integral by use of Stokes' theorem [33]. Application of this theorem results in the total flux linkage being defined as

$$\phi_p = j\mu_0 \oint_{\partial s_p} \mathbf{A} \cdot d\vec{s_p}, \tag{3.28}$$

where ∂s_p is the closed path around the pth loop. Clearly, this is a more mathematically tractable problem.

Before (3.28) may be used, the quantity $\mathbf{A}$ must be evaluated. Initially, the vector potential of the single filamentary loop, denoted by the vector field $\mathbf{a}$, will be examined and then superposed with all loops comprising a coil to form $\mathbf{A}$. The vector field $\mathbf{a}$ may be evaluated by using (3.26) and a piecewise integration similar to the technique used in (3.2),

$$\mathbf{a} = \frac{1}{4\pi}\left[\int_{-b}^{b}\frac{d\vec{l_1}}{r_1} + \int_{a}^{-a}\frac{d\vec{l_2}}{r_2} + \int_{b}^{-b}\frac{d\vec{l_3}}{r_3} + \int_{-a}^{a}\frac{d\vec{l_4}}{r_4}\right]. \tag{3.29}$$

Note that the vector field $\mathbf{a}$ exists only in the $\hat{\imath}$ and $\hat{\jmath}$ directions. Evaluation of (3.29) in the rectangular coordinate system of Figure 3.1 yields

$$\mathbf{a}(x,y,z,a,b,c) = a_x(x,y,z,a,b,c)\hat{\imath} + a_y(x,y,z,a,b,c)\hat{\jmath}, \tag{3.30}$$

where

$$\begin{aligned}
a_x &= \frac{1}{4\pi}\ln\left|\frac{\vartheta_2\vartheta_3}{\vartheta_1\vartheta_4}\right|,\\
a_y &= \frac{1}{4\pi}\ln\left|\frac{\vartheta_6\vartheta_7}{\vartheta_5\vartheta_8}\right|,\\
\vartheta_1 &= (x+a)+\varphi_7,\\
\vartheta_2 &= (x-a)+\varphi_5,\\
\vartheta_3 &= (x+a)+\varphi_8,\\
\vartheta_4 &= (x-a)+\varphi_6,\\
\vartheta_5 &= (y-b)+\varphi_5,\\
\vartheta_6 &= (y+b)+\varphi_6,\\
\vartheta_7 &= (y-b)+\varphi_7,\\
\vartheta_8 &= (y+b)+\varphi_8.
\end{aligned}$$

In a fashion similar to that of (3.8), the total geometric vector potential for the coil in Figure 3.2 with q turns consisting of l layers and t turns per layer is

$$\mathbf{A} = \sum_{p=1}^{q}\mathbf{a}_p = \sum_{\lambda=1}^{l}\sum_{\tau=1}^{t}\mathbf{a}(x,y,z,a_\lambda,b_\lambda,c_\tau^{+}). \tag{3.31}$$

Equation (3.31) may be substituted into (3.28) to calculate the total flux linked by the path ∂s_p. The flux linked by the $\tilde{t}$th turn in the $\tilde{l}$th layer may now be ascertained through the piecewise integration

$$\phi_{\tilde{t},\tilde{l}} = j\mu_0\sum_{\lambda=1}^{l}\sum_{\tau=1}^{t}\left[\int_{-b_{\tilde{l}}}^{b_{\tilde{l}}} a_y(a_{\tilde{l}},y,c_{\tilde{t}},a_\lambda,b_\lambda,c_\tau^{+})dy + \int_{a_{\tilde{l}}}^{-a_{\tilde{l}}} a_x(x,b_{\tilde{l}},c_{\tilde{t}},a_\lambda,b_\lambda,c_\tau^{+})dx + \right.$$
$$\left. \int_{b_{\tilde{l}}}^{-b_{\tilde{l}}} a_y(-a_{\tilde{l}},y,c_{\tilde{t}},a_\lambda,b_\lambda,c_\tau^{+})dy + \int_{-a_{\tilde{l}}}^{a_{\tilde{l}}} a_x(x,-b_{\tilde{l}},c_{\tilde{t}},a_\lambda,b_\lambda,c_\tau^{+})dx\right]. \tag{3.32}$$

Solution of (3.32) yields

$$\phi_{\tilde{t},\tilde{l}} = \frac{j\mu_0}{4\pi}\sum_{\lambda=1}^{l}\sum_{\tau=1}^{t}\Theta_{\tilde{t},\tilde{l},\tau,\lambda}, \tag{3.33}$$

where

$$\Theta_{\tilde{t},\tilde{l},\tau,\lambda} = b_{\tilde{l}} \ln\left| \prod_{m=5}^{8} \frac{\vartheta_m(\mathcal{I}_1)\vartheta_m(\mathcal{I}_3)}{\vartheta_m(\mathcal{I}_2)\vartheta_m(\mathcal{I}_4)} \right| + a_{\tilde{l}} \ln\left| \prod_{m=1}^{4} \frac{\vartheta_m(\mathcal{I}_3)\vartheta_m(\mathcal{I}_4)}{\vartheta_m(\mathcal{I}_1)\vartheta_m(\mathcal{I}_2)} \right|$$
$$+ b_\lambda \ln\left| \prod_{m=5}^{8} \left[\frac{\vartheta_m(\mathcal{I}_1)\vartheta_m(\mathcal{I}_4)}{\vartheta_m(\mathcal{I}_2)\vartheta_m(\mathcal{I}_3)}\right]^{(-1)^m} \right| + a_\lambda \ln\left| \prod_{m=1}^{4} \left[\frac{\vartheta_m(\mathcal{I}_1)\vartheta_m(\mathcal{I}_4)}{\vartheta_m(\mathcal{I}_2)\vartheta_m(\mathcal{I}_3)}\right]^{(-1)^m} \right|,$$

$$\mathcal{I}_1 = (a_{\tilde{l}}, b_{\tilde{l}}, c^+_{\tilde{t}}, a_\lambda, b_\lambda, c^+_\tau),$$

$$\mathcal{I}_2 = (-a_{\tilde{l}}, b_{\tilde{l}}, c^+_{\tilde{t}}, a_\lambda, b_\lambda, c^+_\tau),$$

$$\mathcal{I}_3 = (a_{\tilde{l}}, -b_{\tilde{l}}, c^+_{\tilde{t}}, a_\lambda, b_\lambda, c^+_\tau),$$

$$\mathcal{I}_4 = (-a_{\tilde{l}}, -b_{\tilde{l}}, c^+_{\tilde{t}}, a_\lambda, b_\lambda, c^+_\tau).$$

Note that in (3.33), $\mathcal{I}$ denotes the variable assignments in the functions ϑ and φ. Additionally, $a_{\tilde{l}} = a_\lambda$, $b_{\tilde{l}} = b_\lambda$, and $c^+_{\tilde{t}} = c^+_\tau$, for $\tilde{l} = \lambda$, and $\tilde{t} = \tau$. Use of (3.32) and (3.22) allows for the formulation of the self inductance $L_{1,1}$ as

$$L_{1,1} = \frac{1}{j} \sum_{\tilde{\lambda}=1}^{\tilde{l}} \sum_{\tilde{\tau}=1}^{\tilde{t}} \phi_{\tilde{\tau}\tilde{\lambda}} = \frac{\mu_0}{4\pi} \sum_{\tilde{\lambda}=1}^{\tilde{l}} \sum_{\tilde{\tau}=1}^{\tilde{t}} \sum_{\lambda=1}^{l} \sum_{\tau=1}^{t} \Theta_{\tilde{\tau},\tilde{\lambda},\tau,\lambda}.$$

While (3.33) is convenient, it does have limitations. The integrands of (3.32) are *undefined* when $\tilde{t} = \tau$ and $\tilde{l} = \lambda$, and hence must be ignored when calculating (3.33), or in other words

$$\widetilde{\phi}_{\tilde{t},\tilde{l}} = \begin{cases} 0, & \text{for } \tilde{t} = \tau \textit{ and } \tilde{l} = \lambda; \\ \frac{j\mu_0}{4\pi} \sum\limits_{\lambda=1}^{l} \sum\limits_{\tau=1}^{t} \Theta_{\tilde{t},\tilde{l},\tau,\lambda}, & \text{for } \tilde{t} \neq \tau \textit{ or } \tilde{l} \neq \lambda. \end{cases} \tag{3.34}$$

A modified self inductance may now be formed which neglects the undefined integrands,

$$\widetilde{L}_{1,1} = \frac{1}{j} \sum_{\tilde{\lambda}=1}^{\tilde{l}} \sum_{\tilde{\tau}=1}^{\tilde{t}} \widetilde{\phi}_{\tilde{\tau}\tilde{\lambda}}. \tag{3.35}$$

This will result in error being introduced to the calculation, however if the coil windings are assumed to be closely packed, an estimate of the error may be obtained. Assuming that the coil windings are closely packed allows for the approximations

$$\mathbf{A} = \mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_q \approx q\mathbf{a},$$

$$\phi_p = j\mu_0 \oint_{\partial s_p} \mathbf{A} \cdot d\vec{s_p} \approx qj\mu_0 \oint_{\partial s} \mathbf{a} \cdot d\vec{s}.$$

The total flux linked by the coil would approximately be

$$\Phi = \sum_{p=1}^{q} \phi_p \approx q^2 j\mu_0 \oint_{\partial s} \mathbf{a} \cdot d\vec{s}.$$

If the indefinite path integral is neglected, then

$$\widetilde{\mathbf{A}} \approx (q-1)\mathbf{a},$$

$$\widetilde{\phi}_p \approx (q-1)j\mu_0 \oint_{\partial s} \mathbf{a}\cdot d\vec{s}.$$

The total flux linked by the coil when neglecting the undefined integrands is then

$$\widetilde{\Phi} = \sum_{p=1}^{q} \widetilde{\phi}_p \approx q(q-1)j\mu_0 \oint_{\partial s} \mathbf{a}\cdot d\vec{s}.$$

Using (3.22), the error estimate $\mathcal{E}_s$ of the self inductance resulting from neglecting the undefined integrands may be formed as

$$\mathcal{E}_s \approx \frac{L-\widetilde{L}}{L} = \frac{1}{q}.$$

As can be seen, the error in the self inductance calculation utilizing finite loops and neglecting the undefined integrands is proportional to the inverse of the number of turns in the coil. A corrected self inductance calculation may then be formed as

$$\widehat{L}_{1,1} = \widetilde{L}_{1,1}\left(1+\frac{1}{q}\right).$$

To calculate the mutual inductances $L_{3,1}$ and $L_{2,1}$, a similar approach may be used. Additionally, for the mutual inductance formulations it will not be necessary to neglect some of the integrations, as there will no longer be any overlapping integration paths and hence no undefined integrands. Performing a similar derivation to that of the self inductance formulation yields the mutual inductance of two adjacent coils as

$$L_{3,1} = \frac{\mu_0}{4\pi}\sum_{\tilde{\lambda}=1}^{\tilde{l}}\sum_{\tilde{\tau}=1}^{\tilde{t}}\sum_{\lambda=1}^{l}\sum_{\tau=1}^{t}\Psi_{\tilde{\tau},\tilde{\lambda},\tau,\lambda}, \tag{3.36}$$

where

$$\Psi_{\tilde{\tau},\tilde{\lambda},\tau,\lambda} = (a_{\tilde{\lambda}})_{3,1}\ln\left|\prod_{m=1}^{4}\frac{\vartheta_m(\mathcal{L}_3)\vartheta_m(\mathcal{L}_4)}{\vartheta_m(\mathcal{L}_1)\vartheta_m(\mathcal{L}_2)}\right| + a_\lambda \ln\left|\prod_{m=1}^{4}\left[\frac{\vartheta_m(\mathcal{L}_1)\vartheta_m(\mathcal{L}_4)}{\vartheta_m(\mathcal{L}_2)\vartheta_m(\mathcal{L}_3)}\right]^{(-1)^m}\right|$$
$$+\sum_{m=5}^{8}[\varphi_m(\mathcal{L}_4)-\varphi_m(\mathcal{L}_3)+\varphi_m(\mathcal{L}_1)-\varphi_m(\mathcal{L}_2)],$$

$$\mathcal{L}_1 = ((a_{\tilde{\lambda}})_{3,1},(b_{\tilde{\tau}})_{3,1},(c_{\tilde{\lambda}})_{3,1}^{-},a_\lambda,b_\lambda,c_\tau^{-}),$$

$$\mathcal{L}_2 = (-(a_{\tilde{\lambda}})_{3,1},(b_{\tilde{\tau}})_{3,1},(c_{\tilde{\lambda}})_{3,1}^{-},a_\lambda,b_\lambda,c_\tau^{-}),$$

$$\mathcal{L}_3 = ((a_{\tilde{\lambda}})_{3,1},-(b_{\tilde{\tau}})_{3,1},(c_{\tilde{\lambda}})_{3,1}^{+},a_\lambda,b_\lambda,c_\tau^{-}),$$

$$\mathcal{L}_4 = (-(a_{\tilde{\lambda}})_{3,1},-(b_{\tilde{\tau}})_{3,1},(c_{\tilde{\lambda}})_{3,1}^{+},a_\lambda,b_\lambda,c_\tau^{-}),$$

$$(a_{\tilde{\lambda}})_{3,1} = a_0 + \frac{d\sqrt{3}}{2}(\tilde{\lambda}-1),$$

$$(b_{\tilde{\tau}})_{3,1} = \begin{cases} c_0 + (\tilde{\tau} - 1)d; & \text{if } \tilde{\lambda} \text{ is odd,} \\ c_0 + \frac{d}{2} + (\tilde{\tau} - 1)d; & \text{if } \tilde{\lambda} \text{ is even,} \end{cases}$$

$$(c_{\tilde{\lambda}})^{\pm}_{3,1} = c_0 \pm \left[b_0 + \frac{d\sqrt{3}}{2}(\tilde{\lambda} - 1) \right],$$

and the mutual inductance of two opposite coils as

$$L_{2,1} = \frac{\mu_0}{4\pi} \sum_{\tilde{\lambda}=1}^{\tilde{l}} \sum_{\tilde{\tau}=1}^{\tilde{t}} \sum_{\lambda=1}^{l} \sum_{\tau=1}^{t} \Xi_{\tilde{\tau},\tilde{\lambda},\tau,\lambda}, \tag{3.37}$$

where

$$\begin{aligned} \Xi_{\tilde{\tau},\tilde{\lambda},\tau,\lambda} = b_{\tilde{\lambda}} \ln & \left| \prod_{m=5}^{8} \frac{\vartheta_m(\mathcal{J}_2)\vartheta_m(\mathcal{J}_4)}{\vartheta_m(\mathcal{J}_1)\vartheta_m(\mathcal{J}_3)} \right| + a_{\tilde{\lambda}} \ln \left| \prod_{m=1}^{4} \frac{\vartheta_m(\mathcal{J}_1)\vartheta_m(\mathcal{J}_2)}{\vartheta_m(\mathcal{J}_3)\vartheta_m(\mathcal{J}_4)} \right| \\ & + b_{\lambda} \ln \left| \prod_{m=5}^{8} \left[\frac{\vartheta_m(\mathcal{J}_2)\vartheta_m(\mathcal{J}_3)}{\vartheta_m(\mathcal{J}_1)\vartheta_m(\mathcal{J}_4)} \right]^{(-1)^m} \right| + a_{\lambda} \ln \left| \prod_{m=1}^{4} \left[\frac{\vartheta_m(\mathcal{J}_2)\vartheta_m(\mathcal{J}_3)}{\vartheta_m(\mathcal{J}_1)\vartheta_m(\mathcal{J}_4)} \right]^{(-1)^m} \right|, \end{aligned}$$

$$\mathcal{J}_1 = (a_{\tilde{\lambda}}, b_{\tilde{\lambda}}, (c_{\tilde{\tau}})_{2,1}, a_{\lambda}, b_{\lambda}, c^{-}_{\tau}),$$

$$\mathcal{J}_2 = (-a_{\tilde{\lambda}}, b_{\tilde{\lambda}}, (c_{\tilde{\tau}})_{2,1}, a_{\lambda}, b_{\lambda}, c^{-}_{\tau}),$$

$$\mathcal{J}_3 = (a_{\tilde{\lambda}}, -b_{\tilde{\lambda}}, (c_{\tilde{\tau}})_{2,1}, a_{\lambda}, b_{\lambda}, c^{-}_{\tau}),$$

$$\mathcal{J}_4 = (-a_{\tilde{\lambda}}-, b_{\tilde{\lambda}}, (c_{\tilde{\tau}})_{2,1}, a_{\lambda}, b_{\lambda}, c^{-}_{\tau}),$$

$$(c_{\tilde{\tau}})_{2,1} = \begin{cases} 2c_0 + [d(\tilde{\tau} - 1)], & \text{for } \tilde{\lambda} = 1, 3, 5, \ldots \text{ (odd);} \\ 2c_0 + [\frac{d}{2} + d(\tilde{\tau} - 1)], & \text{for } \tilde{\lambda} = 2, 4, 6, \ldots \text{ (even).} \end{cases}$$

The coil resistance calculation is quite straightforward. If the total length of the wire in the coil is known, then this length may be multiplied by the resistance per length $\mathcal{R}_{length}$ of the wire used to wind the coil. As in the previous field calculations and inductance formulations, the coil is approximated as a series of finite loops. This results in the calculation of resistance as

$$R = 4tl\mathcal{R}_{length} \left[a_0 + b_0 + \frac{d\sqrt{3}}{2}(l - 1) \right]. \tag{3.38}$$

Alternatively, coil resistance may be calculated on a *weight* W basis, where the parameter of resistance per unit weight $\mathcal{R}_{wgt}$ is utilized to calculate the resistance $R = \mathcal{R}_{wgt}W$. While (3.38) is a useful formulation for use in a predictive model or optimization routine, the weight based calculation is much more practical when winding the coil.

CHAPTER 4

Design and Optimization of Experiment

4.1 Functional for Minimization and Algorithm

In a system such as the one proposed in Figure 2.2, there are many design parameters which must be determined before fabrication of an experimental setup can proceed. It may be assumed for the purposes of this study that certain aspects of the system are "fixed" and are not considered variables to be changed or adjusted in any kind of optimization procedure. The fixed characteristics are listed as follows: sphere radius, sphere magnetization, fluid viscosity, and sphere 2D trajectory. Additionally, a large amount of 12 gauge heavy insulation magnet wire was already in hand, resulting in wire diameter being fixed as well. In contrast, certain parameters of this system will be used as variables in an optimization process. These parameters are given in Table 4.1. Upon first observation, one may reason that the coil resistance should be among the variables listed in Table 4.1, however this is not the case. Fabrication of the experimental apparatus will utilize a magnet wire already in hand (12 gauge), thus establishing the wire diameter. The coil resistance is a function of wire diameter, electrical conductivity of the wire utilized, and length of the wire wound into coil form. Of these three criteria, only the wire length will be varied during the optimization and it is a function of the first four parameters of Table 4.1.

In devising a "bench top" test apparatus, it is desirable to have a hardware setup which will move the sphere with the least amount of effort. Hence a minimization of some sort is in order. From the standpoint of the physical system, the attribute which must be minimized is the power applied to the electromagnet coils. The *instantaneous* power supplied to any reactive electrical component is defined as

$$\mathcal{P} = vj,$$

where v is the potential across the device and j is current flowing in the device [34]. Adopting the approach used in [32] and representing the coil as an RL circuit, the instantaneous power in the mth coil may be stated as

$$\mathcal{P}_m = R_m j_m^2 + j_m \sum_{n=1}^{Q} L_{m,n} \frac{dj_n}{dt}, \tag{4.1}$$

Table 4.1: System Parameters to be Used for Optimization

parameter	symbol	unit
number of wire turns/layer	t	- - -
number of wire layers	l	- - -
coil inner "x" dimension	a_{in}	*inch*
coil inner "y" dimension	b_{in}	*inch*
distance of coil face from global origin	c_0	*inch*

where R_m and j_m are the resistance and current for the mth coil, $L_{m,n}$ is the mutual $(m = n)$ or self $(m \neq n)$ inductance of the mth coil on the nth coil, and Q is the total number of coils. Consistent with Chapter 2, the optimization problem will be based the ability of the hardware system to move a sphere about a circular orbit. Hence, symmetry may be leveraged and only coils 1 and 3 examined. To calculate the instantaneous power of the coil combination depicted in Figure 2.10, denoted here as $\Pi_{1,3} = (\mathcal{P}_1, \mathcal{P}_3)$, the formulations for coil resistance and inductance (both self and mutual) developed in Chapter 3 may be employed. The analytical evaluation of the time derivative of current would prove to be quite tedious, if even possible, due to the required formulation of the partial derivatives of the vector field term $\vec{\mathbf{k}}_{FM}$. A complete mathematical description of the vector field term is available in Chapter 2. Because of this, the time derivative of the required currents in coils 1 and 3 will be calculated numerically from discrete points comprising the inverse current solution of a given system configuration. Obviously in a reactive system such as an RL ciruit, achieving the *instantaneous* current change at the beginning and end of the inverse current solution as depicted in Figure 2.10 would be impossible. Indeed an inductive "switching effect" would be inherent in the proposed technique, however for the purposes of optimizing an experimental design this will be neglected.

For the set of discrete points making up the inverse current solution $J_{\gamma,1,3}$, where $\gamma = a, b, c, d$, a forward difference approximation was used to calculate the derivative at the initial point while a backward difference approximation was used to calculate the derivative at the last point. For all other points making up the set, a central difference technique was used. This will provide a reasonable approximation of the function $\dot{J}_{\gamma,1,3}$, provided that the time step spacing is small [35]. The time derivative of current was calculated using points spaced at $10\,msec$ intervals. An

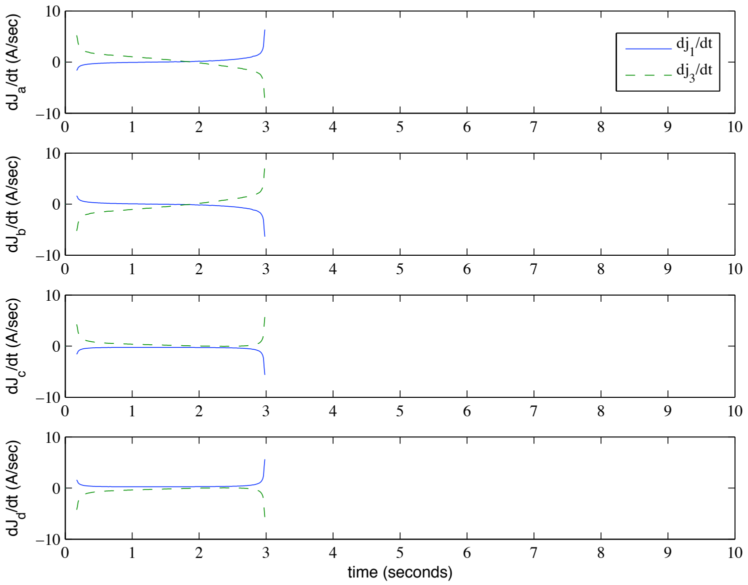

Figure 4.1: Calculated current derivative from inverse solution of coil combination 1&3. Displayed results correspond to system with $R_{1,3} = 0.547\,\Omega$, $L_{1,1} = 6.79\,mH$, and $L_{1,3} = 0.58\,mH$ (calculated).

example calculation on the current trajectory depicted in Figure 2.10 is shown in Figure 4.1. It should be noted that the trace shown in Figure 2.10 corresponds to a system where the resistance, inductance and mutual inductance have been calculated to be $R_{1,3} = 0.547\,\Omega$, $L_{1,1} = 6.79\,mH$, and $L_{1,3} = 0.58\,mH$. Utilizing equation (4.1) with the discrete traces of $j_{1,3}$ and $dj_{1,3}/dt$ along with the calculated values for resistance and inductance allows for the estimation of the instantaneous power required by the coil combination which results in the inverse current solution. The instantaneous power calculation for this configuration is shown in Figure 4.2. As can be seen, the instantaneous power has two distinct forms rather than four. This is indicative of the same power being needed for the solutions corresponding to opposite polarities.

With the proposed minimal coil set technique, two coils will be powered at any one given time. Hence, it would be advantageous to utilize an expression as the functional for minimization which incorporates the power applied to *both* members of the coil set. The Euclidean norm was employed to satisfy this convenience and is defined here as

$$\|\mathit{\Pi}_{m,n}\| = \sqrt{\mathcal{P}_m^2 + \mathcal{P}_n^2},$$

where $m, n = 1, 2, 3, 4$. Obvioulsy $\|\mathit{\Pi}_{m,n}\| = \|\mathit{\Pi}_{n,m}\|$ and for the case being examined $m = 1$ and $n = 3$. The norms of the two instantaneous power traces are shown in Figure 4.3. From

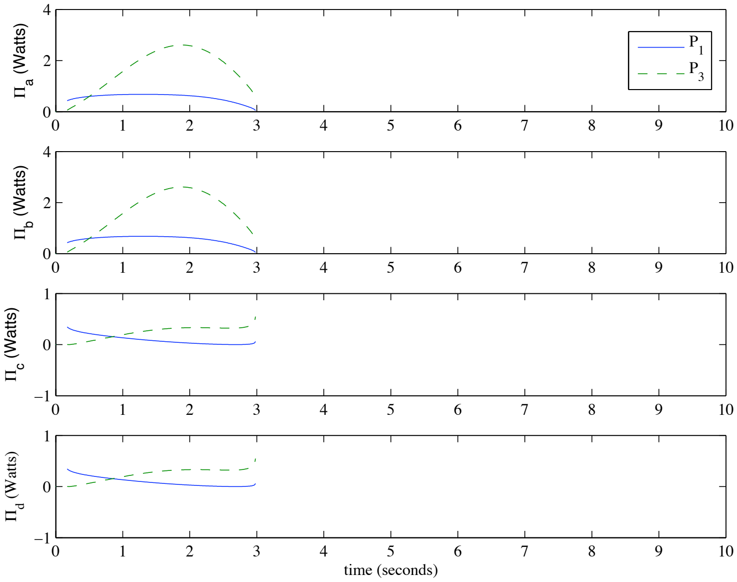

Figure 4.2: Instantaneous power required for inverse current solution of coil combination 1&3.

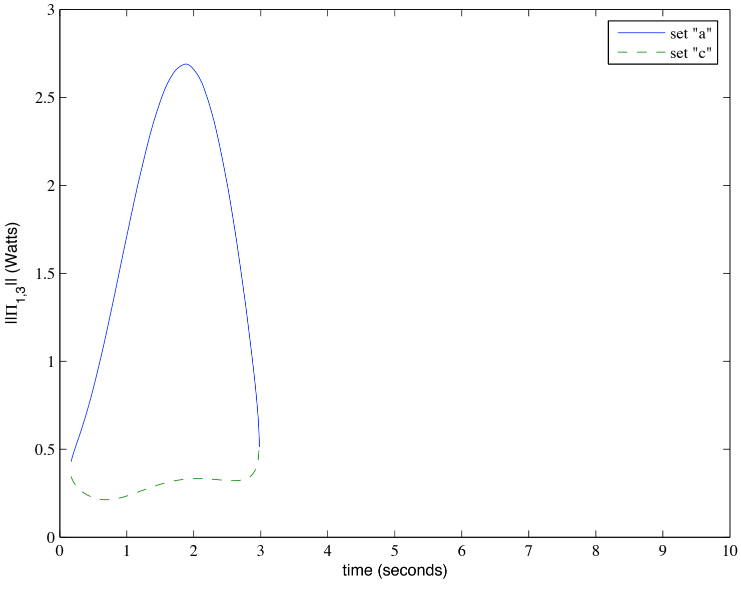

Figure 4.3: Plot of $\|\mathit{\Pi}_{1,3}\|$ for solution sets a and c. The norm of solution sets b and d are identical to sets a and c, respectively.

the power norm, a single metric may be defined to describe the performance of a coil system with the parameters seen in Table 4.1. If the maximum value of the current norm, defined here as $\mathcal{M} = \max \|\mathit{\Pi}_{m,n}\|$, is used as criteria for minimization, then the minimized configuration $\mathcal{S}_{min}$ would be the set of parameters $\mathcal{S} = \{t, l, a_0, b_0, c_0\}$ which results in the minimum $\mathcal{M}$ when compared to all parameter values possible. More precisely, $\{t, l, a_{in}, b_{in}, c_0\} = \mathcal{S} = \mathcal{S}_{min}$ if $\exists \{t, l, a_{in}, b_{in}, c_0\}$: $\|\mathit{\Pi}_{m,n}(t, l, a_{in}, b_{in}, c_0)\| \leq \inf\{\mathcal{M}(\mathcal{S})\} \forall \mathcal{S} \in \mathbb{R}$. In an effort to reduce the number of parameters utilized in the search routine, it was decided to make c_0, the distance from the face of the coils to the origin (see Figure 2.2), a function of the parameters l and b_{in}. To achieve this, a spacing criteria c_{sp} is defined as

$$c_{sp} = b_{in} + d\left[\frac{2 + (l-1)\sqrt{3}}{2}\right] + \delta_{sep}, \tag{4.2}$$

where d is the wound wire diameter, b_{in} is the inner Z dimension of the rectangular coil, δ_{sep} is a separation constant, and l is the number of layers of wire wound on the coil. It should be noted that the formulation of (4.2) is based upon a coil having a "perfect" winding pattern as defined in Chapter 3. The minimum coil to coil spacing for this experiment was chosen to be 5", resulting in a minimum coil spacing from the origin of $c_{min} = 2.5$". In addition, the separation constant in (4.2) was chosen to be $\delta_{sep} = 0.25$". To reduce the number of parameters for the optimization problem, c_0 may then be defined as

$$c_0 = \begin{cases} c_{min} & \text{if } c_{sp} \leq c_{min}, \\ c_{sp} & \text{otherwise.} \end{cases}$$

This relationship sets the coil spacing as 2.5 inches unless the coil physical dimensions do not allow for them to be positioned this closely. In that case, the coil spacing from the origin is dictated by (4.2).

The technique selected to find the minimal parameter set is a basic gradient method: steepest descent [36, 37]. More specifically, a *discrete* steepest descent algorithm was used because the system model was evaluated at specific *points* in the parameter space. This work will utilize a rather straigtforward version of this gradient method, which will be presented in two parts. The first component may be summarized as a method by which, for a given a_{in} and b_{in}, the number of turns per layer t and the number of layers l resulting in the minimum $\mathcal{M}$ is found. To achieve this, we may look at $\mathcal{M}$ as a function of the minimizing variables t, l, a_{in}, and b_{in}, hence $\mathcal{M} = \mathcal{M}(t, l, a_{in}, b_{in})$. For a given initial guess (t_i, l_i) and a given (a_{in}, b_{in}), the initial value of $\mathcal{M} = \mathcal{M}(t_i, l_i, a_{in}, b_{in})$ is calculated. The points "around" the initial guess are utilized to then calculate the values of $\mathcal{M}$ around the initial guess. These points make up the ordered set $\mathbb{S}_u^i = \{(t_i + 1, l_i), (t_i + 1, l_i - 1), (t_i, l_i -$

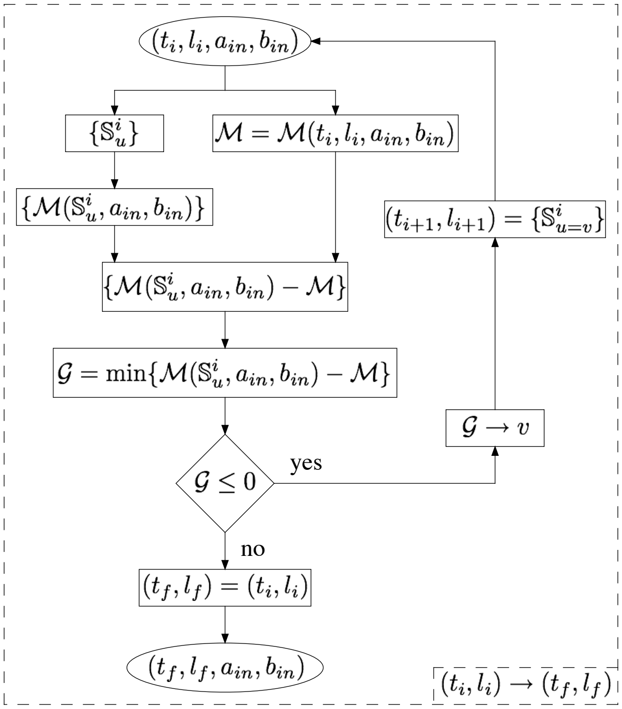
$(t_i, l_i, a_{in}, b_{in})$
$\{\mathbb{S}_u^i\}$
$\mathcal{M} = \mathcal{M}(t_i, l_i, a_{in}, b_{in})$
$\{\mathcal{M}(\mathbb{S}_u^i, a_{in}, b_{in})\}$
$(t_{i+1}, l_{i+1}) = \{\mathbb{S}_{u=v}^i\}$
$\{\mathcal{M}(\mathbb{S}_u^i, a_{in}, b_{in}) - \mathcal{M}\}$
$\mathcal{G} = \min\{\mathcal{M}(\mathbb{S}_u^i, a_{in}, b_{in}) - \mathcal{M}\}$
$\mathcal{G} \to v$
$\mathcal{G} \leq 0$
yes
no
$(t_f, l_f) = (t_i, l_i)$
$(t_f, l_f, a_{in}, b_{in})$
$(t_i, l_i) \to (t_f, l_f)$

Figure 4.4: A flowchart illustrating the algorithm used to find the number of turns t_f and number of layers l_f which results in minimal power used for a given a_{in} and b_{in}.

$1), (t_i - 1, l_i - 1), (t_i - 1, l_i), (t_i - 1, l_i + 1), (t_i, l_i + 1), (t_i + 1, l_i + 1)\}$, where $u = 1, 2, \ldots, 8$ denotes a particular ordered set member. This set may then be used to calculate the values of $\mathcal{M}$ at each set point, $\{\mathcal{M}(\mathbb{S}_u^i, a_{in}, b_{in})\} = \{\mathcal{M}(\mathbb{S}_1^i, a_{in}, b_{in}), \ldots, \mathcal{M}(\mathbb{S}_8^i, a_{in}, b_{in})\}$. The "next step" (t_{i+1}, l_{i+1}) may then be determined as the vth element of $\mathbb{S}_u^i$ which results in $\mathcal{G} = \min\{\mathcal{M}(\mathbb{S}_u^i, a_{in}, b_{in}) - \mathcal{M}(t_i, l_i, a_{in}, b_{in})\}$ and $\mathcal{G} \leq 0$. This process may be repeated until a point (t_f, l_f) is found such that $\mathcal{G} > 0$. At which time, a minimizing t and l have been found. This process is illustrated in the flow chart shown in Figure 4.4.

The second component of the complete minimization algorithm may be viewed as an analogy of the first component discussed in the previous paragraph and shown in Figure 4.4. It may be summarized as a method by which, for a given t and l, the inner x coil dimension a_{in} and the inner y coil dimension b_{in} resulting in the minimum $\mathcal{M}$ is found. As before, this may be achieved by regarding $\mathcal{M}$ as a function of the variables which are being "varied". For a given initial guess $(a_{in,j}, b_{in,j})$ and a known value (t, l), the initial value of $\mathcal{M} = \mathcal{M}(t, l, a_{in,j}, b_{in,j})$ is calculated. Once again, the points "around" the initial guess are utilized to calculate the values of $\mathcal{M}$ around the initial guess. These points make up the ordered set $\mathbb{T}_u^j = \{(a_{in,j} + \alpha, b_{in,j}), (a_{in,j} + \alpha, b_{in,j} - \alpha), (a_{in,j}, b_{in,j} - \alpha), (a_{in,j} - \alpha, b_{in,j} - \alpha), (a_{in,j} - \alpha, b_{in,j}), (a_{in,j} - \alpha, b_{in,j} + \alpha), (a_{in,j}, b_{in,j} + \alpha), (a_{in,j} + \alpha, b_{in,j} + \alpha)\}$, where $u =$

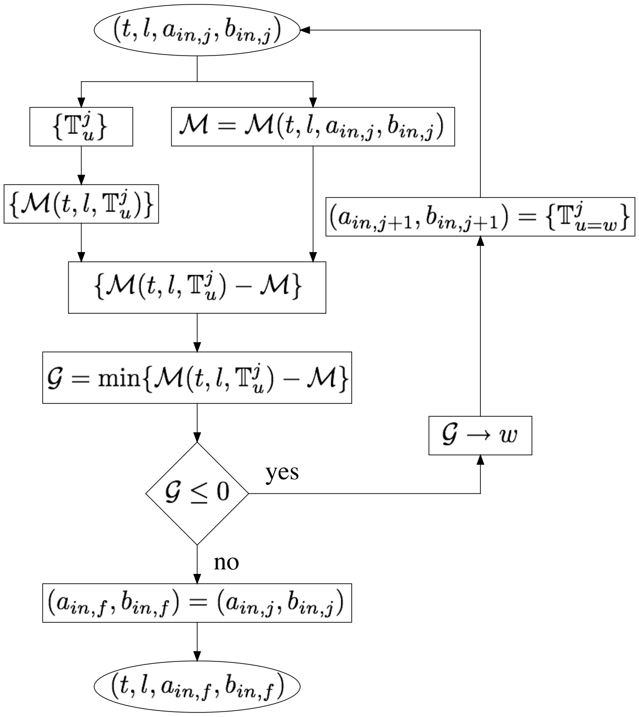
$(t, l, a_{in,j}, b_{in,j})$
$\{\mathbb{T}_u^j\}$
$\mathcal{M} = \mathcal{M}(t, l, a_{in,j}, b_{in,j})$
$\{\mathcal{M}(t, l, \mathbb{T}_u^j)\}$
$(a_{in,j+1}, b_{in,j+1}) = \{\mathbb{T}_{u=w}^j\}$
$\{\mathcal{M}(t, l, \mathbb{T}_u^j) - \mathcal{M}\}$
$\mathcal{G} = \min\{\mathcal{M}(t, l, \mathbb{T}_u^j) - \mathcal{M}\}$
$\mathcal{G} \to w$
$\mathcal{G} \leq 0$
yes
no
$(a_{in,f}, b_{in,f}) = (a_{in,j}, b_{in,j})$
$(t, l, a_{in,f}, b_{in,f})$

Figure 4.5: A flowchart illustrating the algorithm used to find the inner x coil dimension $a_{in,f}$ and inner y coil dimension $b_{in,f}$ which results in minimal power used for a given t, l, and α.

$1, 2, \ldots, 8$ denotes a particular ordered set member and α denotes a *search length*. This set may then be used to calculate the values of $\mathcal{M}$ at each set point, $\{\mathcal{M}(t, l, \mathbb{T}_u^j)\} = \{\mathcal{M}(t, l, \mathbb{T}_1^j), \ldots, \mathcal{M}(t, l, \mathbb{T}_8^j)\}$. The "next step" $(a_{in,j+1}, b_{in,j+1})$ may then be determined as the wth element of $\mathbb{T}_u^j$ which results in $\mathcal{G} = \min\{\mathcal{M}(t, l, \mathbb{T}_u^j) - \mathcal{M}(t, l, a_{in,j}, b_{in,j})\}$ and $\mathcal{G} \leq 0$. This process may be repeated until a point $(a_{in,f}, b_{in,f})$ is found such that $\mathcal{G} > 0$. At which time, a minimizing a_{in} and b_{in} have been found. This process is illustrated in the flow chart shown in Figure 4.5. As can be seen, Figures 4.4 and 4.5 are similar in structure. To achieve a single algorithm which finds a minimizing point $(t_f, l_f, a_{in,f}, b_{in,f})$, one algorithm will be "nested" in the other.

For this study, it was decided to nest the algorithm depicted in Figure 4.4 into the algorithm depicted in Figure 4.5. The flow of the complete process is similar to that of the procedures previously outlined. For an initial guess $(t_i, l_i, a_{in,j}, b_{in,j})$, the initial value of $\mathcal{M} = \mathcal{M}(t_i, l_i, a_{in,j}, b_{in,j})$ must be calculated. Subsequently, $\mathcal{M} = \mathcal{M}(t_\delta, l_\delta, a_{in,j}, b_{in,j})$ which has been minimized on (t, l) may be calculated using a nested algorithm similar to that in Figure 4.4. In this case, δ represents an arbitrary number of iterations required for convergence. Similarly, the minimal $\{(t_\epsilon, l_\epsilon)_u\}$ for *each* set member of $\mathbb{T}_u^j$ may be calculated as well by a nested (t, l) minimization algorithm, where ϵ represents an arbitrary number of iterations required for convergence. These results are then used to check

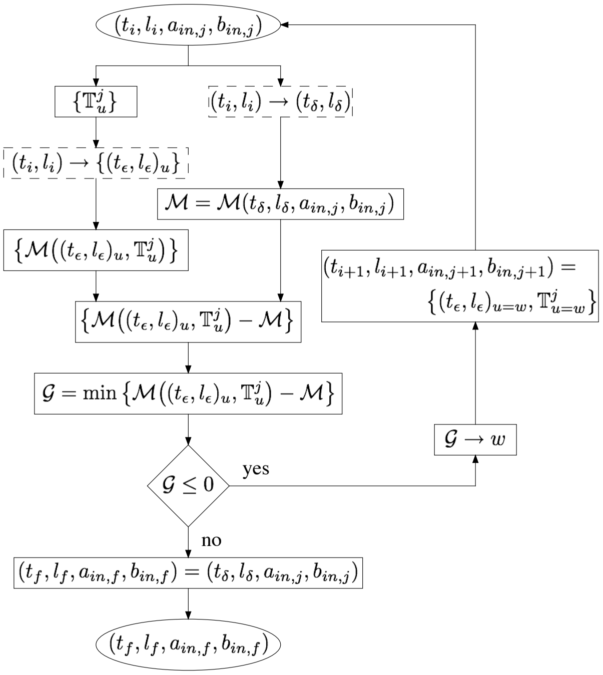
$(t_i, l_i, a_{in,j}, b_{in,j})$
$\{\mathbb{T}_u^j\}$
$(t_i, l_i) \rightarrow (t_\delta, l_\delta)$
$(t_i, l_i) \rightarrow \{(t_\epsilon, l_\epsilon)_u\}$
$\mathcal{M} = \mathcal{M}(t_\delta, l_\delta, a_{in,j}, b_{in,j})$
$\{\mathcal{M}((t_\epsilon, l_\epsilon)_u, \mathbb{T}_u^j)\}$
$(t_{i+1}, l_{i+1}, a_{in,j+1}, b_{in,j+1}) = \{(t_\epsilon, l_\epsilon)_{u=w}, \mathbb{T}_{u=w}^j\}$
$\{\mathcal{M}((t_\epsilon, l_\epsilon)_u, \mathbb{T}_u^j) - \mathcal{M}\}$
$\mathcal{G} = \min \{\mathcal{M}((t_\epsilon, l_\epsilon)_u, \mathbb{T}_u^j) - \mathcal{M}\}$
$\mathcal{G} \rightarrow w$
$\mathcal{G} \leq 0$
yes
no
$(t_f, l_f, a_{in,f}, b_{in,f}) = (t_\delta, l_\delta, a_{in,j}, b_{in,j})$
$(t_f, l_f, a_{in,f}, b_{in,f})$

Figure 4.6: A flowchart illustrating the complete algorithm used to find the parameters $(t_f, l_f, a_{in,f}, b_{in,f})$ resulting in minimal power. The nested algorithms are indicated by "broken line" boxes.

for a descent direction in the variable space. This procedure is then repeated until a minimizing configuration $(t_f, l_f, a_{in,f}, b_{in,f})$ is found. The complete algorithm is shown in Figure 4.6. It must be noted that while not explicitly depicted in Figure 4.6, a value for the search length α must be selected before using the algorithm. The final aspect to be resolved before the minimization routine may be implemented is which norm set will be used (see Figure 4.3). For this study, "a" was selected as the set utilized in the functional evaluation. With the minimization routine developed, and the system model in place, a minimizing parameter set may be calculated.

4.2 Minimization Results and Experimental Setup

A minimizing parameter set $(t_f, l_f, a_{in,f}, b_{in,f})$ was indeed found from utilizing the routines developed in Section 4.1. Three different search efforts were performed with search lengths valued as $\alpha = 0.1\text{"}, 0.01\text{"}$, and 0.001". This method may be viewed as one in which an initial "coarse" search is performed to establish a general area of the extremum and then each successive search utilizes a finer search grid. The optimized design parameters for the quad coil assembly are given in Table 4.2. As can be seen, the optimal coil design corresponds to one in which the coil faces are placed as close

Table 4.2: Optimized System Parameters

parameter	symbol	value	unit
number of wire turns/layer	t	18	- - -
number of wire layers	l	21	- - -
coil inner “x” dimension	a_{in}	0.439	*inch*
coil inner “y” dimension	b_{in}	0.757	*inch*
distance of coil face from global origin	c_0	2.501	*inch*

to the origin as allowed, in this case 2.5”. This is indicative of the proximity of the particle to the electromagnet as being a major contributor to the amount of power required for particle movement. A coil assembly along with the tray providing the plane of motion for the sphere was constructed to these specifications and is shown in Figure 4.10. As expected, the geometry of the wound coil is not “rectangular” even though the spool on which the wire was wound was of the dimensions given in Table 4.2. In fact, the model only *approximates* the coil geometry as that of a rectangular shape, however it will be shown in a later passage that this approximation closely predicts the field characteristics of this system. Indeed, the parameters calculated with the optimization routine were utilized in the system model to generate the traces seen in Figures 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13, 4.1, 4.2, and 4.3 and it is expected that these traces will closely predict the performance of the hardware.

With the design for the coil set firmly established, the remainder of the experimental setup needs to be conceived. The system will be composed of: the coil set, the sphere and tray, a sensor system consisting of sense resistors for current measurement and a digital camera/optical amplifier for sphere position measurement, a power system consisting of DC supplies and power operational amplifiers, a signal conditioning system to properly scale the measured voltage signals, and a data acquisition system comprised of a computer and a dSPACE 1104 acquisition card for implementing control and logging data. A schematic of this system is shown in Figure 4.8. This experimental setup will facilitate open loop and closed loop control while measuring sphere position and coil voltage and current. A more detailed discussion of this system will be provided in the section to follow.

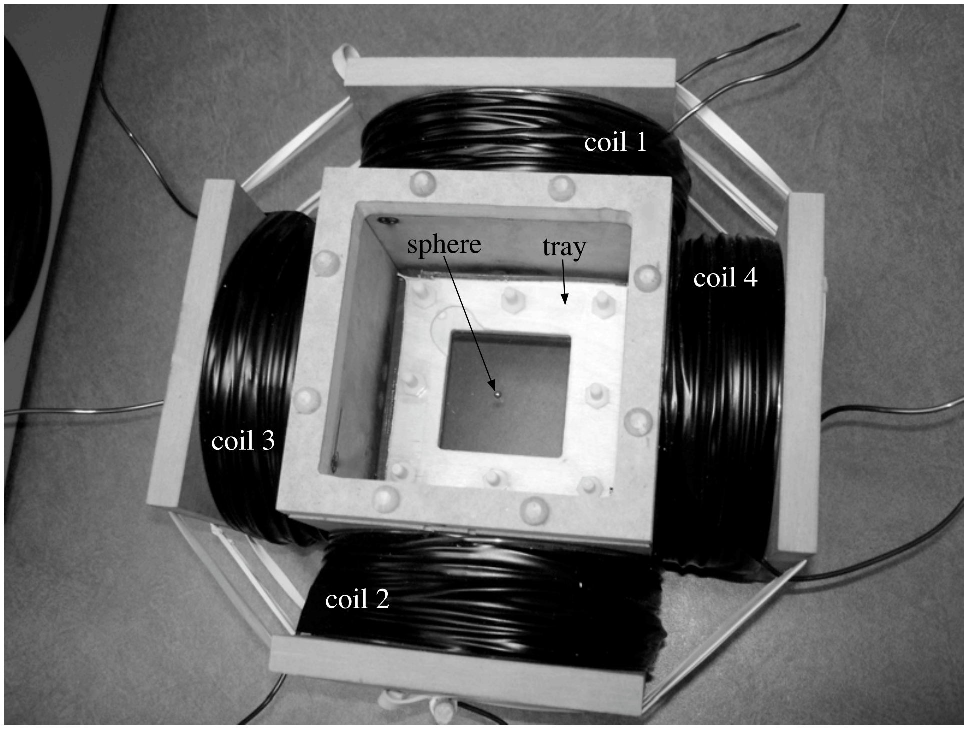

Figure 4.7: Experimental apparatus

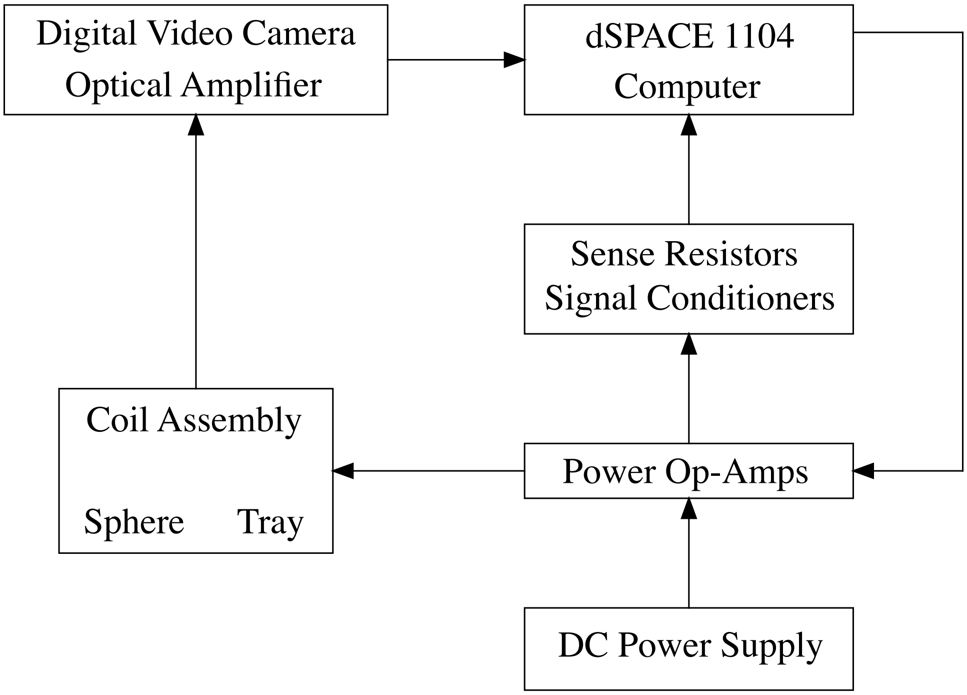

Figure 4.8: Schematic of experimental set-up and signal flow.

4.3 Experimental Platform

In the simplest terms, the hardware for this experimental work consists of the actuator, the plant, the sensor system, and the controller. To gain an understanding of the physical make up of these elements, the four component classes will be examined in more specific terms. The actuator of this system consists of the coil assembly. Orientation of the assembly shown in Figure 2.2, will be such that X coordinate direction will be aligned with the gravitational bias.

The plant of this system consists of the sphere and fluid medium. The sphere used is a permanent rare earth magnet, neodymium of grade 40 with a diameter of 1/8". No instrumentation was in hand which would facilitate a measurement of the sphere's magnetization, therefore it was assumed that the remanent flux density in the sphere was $B_r \approx 1T$. The magnetization of the sphere may then be estimated as $M \approx 3B_r/(2\mu_0)$, where μ_0 is the permeability constant of free space and the multiplying factor 3/2 adjusts for the demagnetization effect of the spherical geometry. The tray in which the sphere resides is constructed of a wooden square annulus with similarly shaped rubber gaskets placed between two square plexiglas plates. The entire assembly was clamped together by way of nylon fasteners around the perimeter of the wooden spacer. The area of allowable sphere motion is limited to $4in^2$, with this area to be centered in the coil assembly. The total distance between the two plexiglas plates was determined by the wooden spacer thickness of 3/16" and rubber gasket material of 1/16" (each), bringing the total spacing of the plates to 0.3125". This allows around 3/32" clearance above and below the sphere, however it is assumed that the out-of-plane magnetic force which will be generated by the coil assembly is not sufficient to "levitate" the ball in the X direction. Location of the tray in the electromagnet assembly will be such that the center of the ball aligns with the central plane of the coil assembly, denoted the "plane of motion" in Figure 2.2. Because of this, the sphere will be *off center* in the fluid gap, with the lower plate providing the constraint force keeping the sphere "on plane". A schematic depicting this is available in 4.9. The static fluid the sphere is immersed in is glycerine, with an assumed value for viscosity of $\mu = 0.950 N \cdot s/m^2$ [38]. A photograph of the fabricated actuator and plant for this experiment is shown in Figure 4.10.

The sensor system for this experiment consists of hardware to measure position of the sphere in the 2-D plane, voltage applied to the coils, and current flowing in the coils. For position measurement, a digital video camera was employed. Measurement of coil voltage was directly measured by the data acquisition system which will be discussed in more detail in a later passage. The final general system component is the controller. Hardware for the controller consists of the data acquisition card

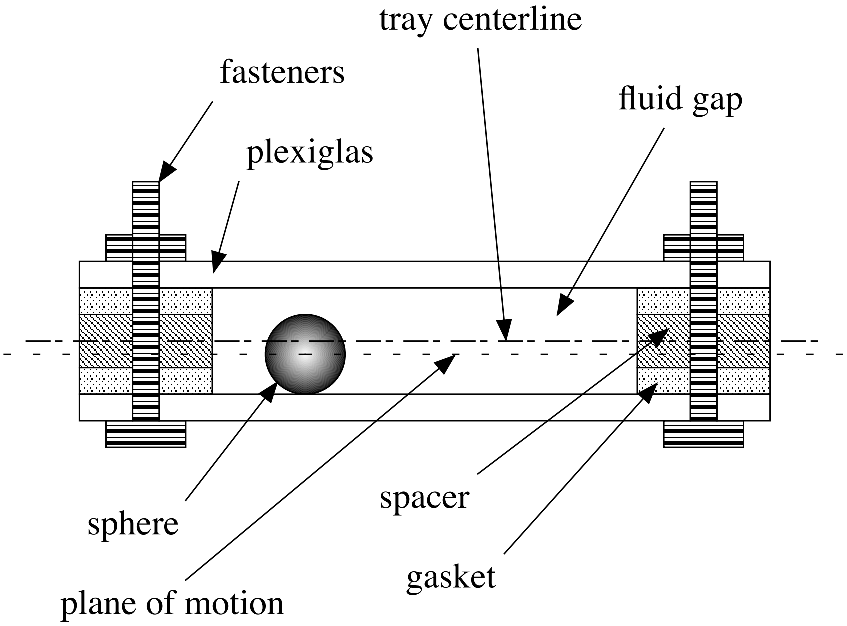

Figure 4.9: Schematic of tray cross section. The sphere resides below the centerline of the fluid gap.

and PC, DC power supplies, and power operational amplifiers. The data acquisition system used in this experiment was a dSPACE 1104 card installed in an Intel based PC.

Finally, power operational amplifiers were utilized to facilitate a variable voltage/current being applied to the coils and sense resistors. The particular model of chip used was the Burr-Brown OPA548 high voltage, high current power op amp. This chip is rated for $3A$ continuous, $5A$ peak output current. In addition to high output current, this chip has a second desirable feature. The chip design incorporates an "on/off" switch enabled by a threshold voltage applied to a dedicated pin out. Because the proposed minimal coil set method will inherently utilize the switching of coils on and off, this feature will be used to switch the coil's current source off in addition to just commanding a zero current. A complete circuit diagram for the power op amp and output signal conditioning amplifiers is shown in Appendix B.

4.4 Experimental Validation of Platform Design

A "benchtop" version of the apparatus depicted in Figure 2.2 has been constructed. The purpose of this experimental setup is to explore the technique proposed in [32] by moving a small, 1/8" diameter, spherical magnet constrained to a plane. The hardware was designed using the models developed in Sections 3.2 and 3.4, the inverse current solution of Section 2 and [32], and a discrete steepest decent gradient search technique. A picture of the coil assembly is available in Figure 4.10. The

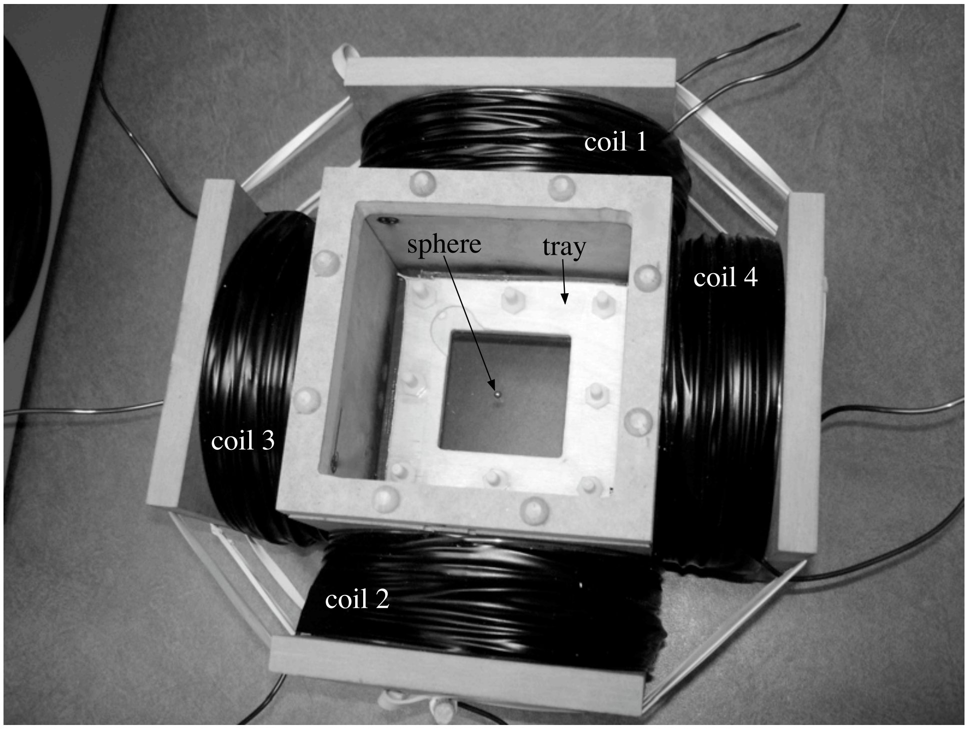

Figure 4.10: Experimental apparatus

minimization sought to find the inner coil dimensions a_0 and b_0, along with the winding parameters t and l, such that minimal power was used to follow a given path. It was decided to utilize a magnet wire of gauge 12 (0.0838" diameter) because a quantity of this material was already in hand. For the minimization process, the path chosen was a circular path orbiting the origin with a frequency of $0.1Hz$. Such a slow trajectory was selected to guarantee operation at a current level lower than $3A$, the maximum achievable current output of power supplies already in hand. Additionally, to maintain a low power requirement the radius of the circular path was limited to 0.5".

The four coils were wound on rectangular bobbins and field measurements in a local $y-z$ plane were performed. These measurement planes correspond to the plane in which the magnetized sphere would reside when the four coils are assembled in a configuration as seen in Figures 2.2 and 4.10. The targeted coil parameters are given in Table 4.3. The coils for this study were wound by hand and it was observed that while the perfect winding pattern displayed in Figure 3.2 was achieved for the initial layers, as the number of layers increased there was difficulty in maintaining this pattern. In fact, only the first eight layers (approximately) of the 21 targeted were wrapped in a perfect fashion. The remaining layers deviated from this with some layers possessing larger spacings between the wires than would exist in the ideal winding pattern. As a result, it was not

parameter	variable	target	unit
turns per layer	t	18	n/a
layers	l	21	n/a
wire diameter	d	0.0838 (12)	$inch$ ($Gauge$)
inner minor length	a_0	0.481	$inch$
inner major length	b_0	0.799	$inch$
weight	W	6.897	lb
resistance	R	0.547	Ω
self inductance	$\widehat{L}_{1,1}$	6.81	mH

Table 4.3: Targeted wound coil parameters

possible to accurately track the number of turns per layer or the number of layers. Additionally, as expected it was not possible to wind a "rectangular" coil. Although the coil was wound on a spool with a rectangular cross section, the final geometry resembles that of a cylinder with an elliptical (rather than circular) shape. It should be noted that the spools were constructed of wood to be non-magnetic and electrically non-conductive. The imperfect winding technique resulted in difficulties in accurately tracking the number of layers wound. Because of this, the four coils were wound on a *weight* basis. Each coil was wound to a weight of $6lb$ $15oz$. The accuracy of the digital scale employed allowed for an error of $\pm 0.5oz$. Resistance measurements on the four wound coils resulted in an average resistance of 0.545Ω with a $\pm 3\sigma$ value of 0.03Ω. Self inductance measurements yielded an average value of $7.708mH$ with a $\pm 3\sigma$ value of $1.329mH$. As expected, the resistance values for the four wound coils are quite close to the targeted value given in Table 4.3. The average self inductance, however, is higher than the targeted value with approximately a $\pm 17\%$ distribution. This is presumably due to variations in wound geometry from coil to coil and the fact that the model is representing an oval shaped coil as a rectangle. A Gauss probe was also utilized to perform flux density measurements in the $y - z$ plane. Data was taken in both the y and z directions at spatial points on a $1in^2$ grid. So as not to disturb the field with the introduction of any ferromagnetic material, wooden gauge blocks and a perforated probe positioning plate also constructed of wood were machined to facilitate the spacing. Pictures of the measurement technique utilized in the Y and Z directions on a single coil are shown in Figures 4.11 and 4.12. The applied current and the resulting field components $B_{y,z}$ were recorded and then converted to magnetic field strength by approximating the permeability of the air/wood medium as free space (μ_0). The magnetic field strength may then be divided by applied current to calculate the geometric field components $G_{y,z}$.

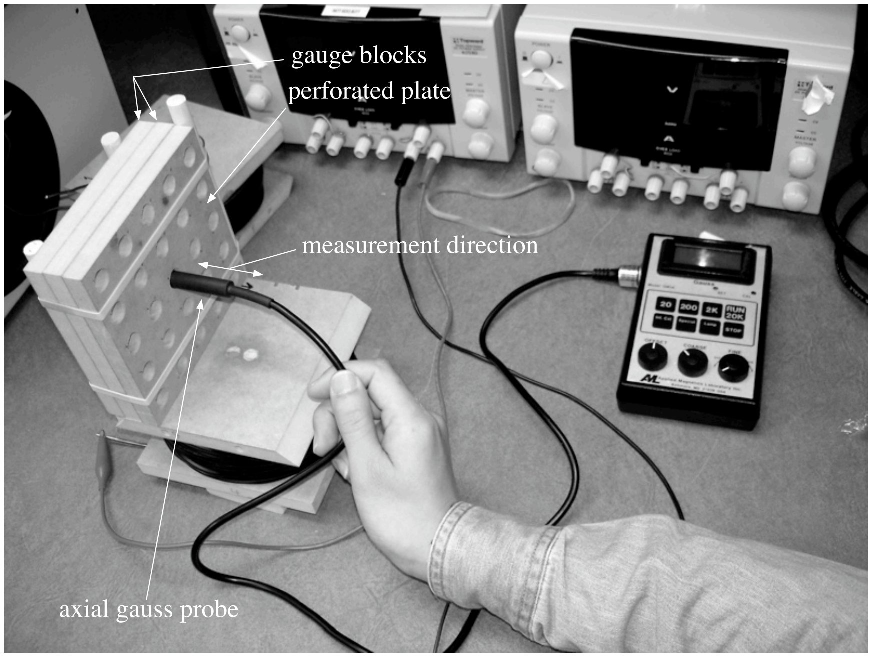

Figure 4.11: Experimental setup for measurement of the flux density produced by a single coil in the y direction.

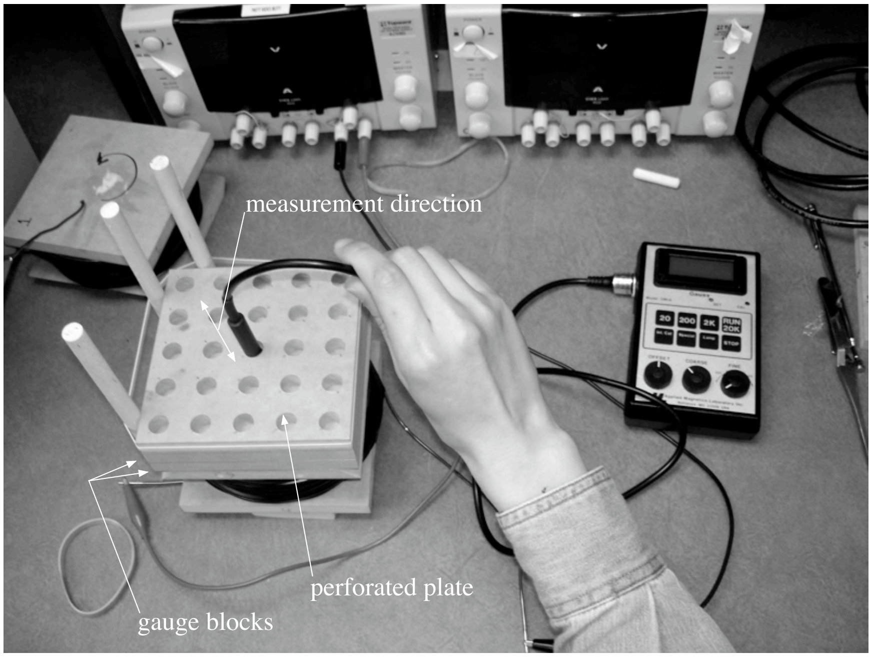

Figure 4.12: Experimental setup for measurement of the flux density produced by a single coil in the z direction.

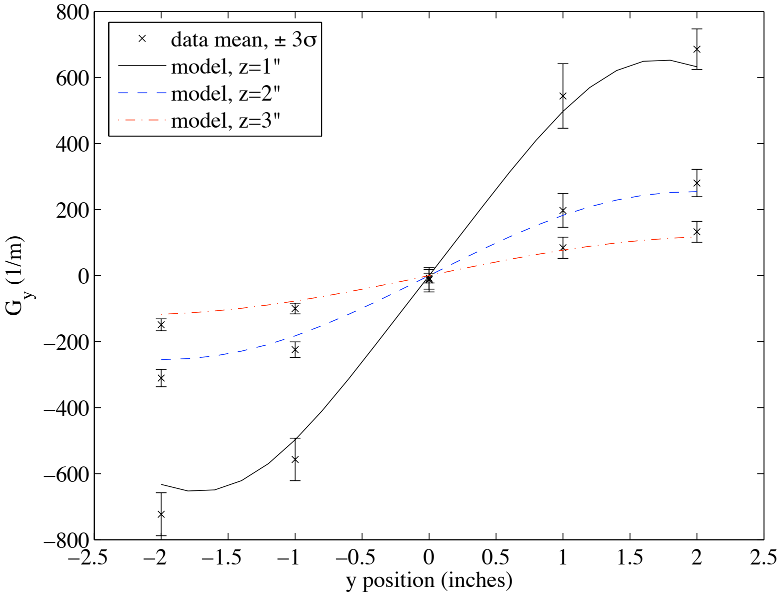

Figure 4.13: A comparison of the predictive model for the field component G_y to measured data points. The brackets around the points indicate $\pm 3\sigma$ distributions of the measurement technique at that point.

At each spatial location, the measurement was performed five times to ascertain the variation of the measurement method. Plots of this data compared to the predictive model in the y and z directions are shown in Figures 4.13 and 4.14.

As can be seen, the predictive model tracks the measured data quite well, especially in the case of the z direction. The model curves fall well within the 3σ error-bars at most points. Some regions on the y direction plot do not follow the measured data. In fact, the measured data exhibits a degree of asymmetry from the first to third quadrants in Figure 4.13, which for an ideally wound coil should not be present. This points to the possibility that in addition to winding discrepancies, probe fixture may be an issue in this measurement as well. Indeed, the gauge block system used resulted in a more accurate positioning of the probe for the z direction measurement versus the y direction. Additionally, assembly tolerance of the coil bobbins may also account for some of the symmetry deviations. The probe is aligned with datums located on the bobbin edge plates, not on the wire itself. As the spools were constructed with hand tools from two edge-plates connected with a central rectangular bobbin for the wire to be wrapped around, there is the possibility that the wire is not *exactly* centered on the edge-plates. Finally, the final shape of the coil is not just elliptical, it turns out to be slightly *ovoid*, with one end slightly longer and more pointed due to the attempted

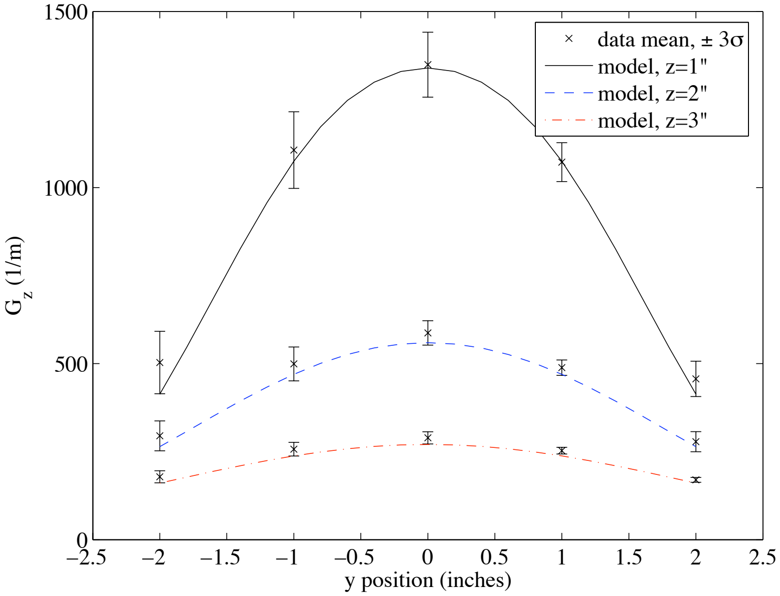

Figure 4.14: A comparison of the predictive model for the field component G_z to measured data points. The brackets around the points indicate $\pm 3\sigma$ distributions of the measurement technique at that point.

perfect wind resulting in a consistent wire crossover side during the winding process.

CHAPTER 5

Closed Loop Control of Sphere Motion

5.1 Coil Switching Considerations

By utilizing the new minimal coil set technique, a coil “switching” methodology must be applied in the control algorithm. Indeed, by employing the minimal coil set concept some advantages may be realized. Firstly, application of a full set of energized coils results in an *underdetermined* system in which a multitude of current solutions are available for the same resultant force vector. Resolving this issue is a challenging problem and drawback of a full set technique. Reducing the number of active coils to only the minimum greatly reduces the number of solutions. Secondly, any energized coil is not trying to overcome the effects of the coil opposite of it in the assembly or array, as would be the case for a set in which all coils are active simultaneously. For a fully active set of coils, two opposite coils may adversely effect the field gradient resulting in no motion of the particle. Thirdly, a minimal amount of power is needed for a coil set which is comprised of the *minimum* number of coils necessary to achieve the desired force on the spherical particle. This strategy has been achieved with the results presented at a later point in the work. The method by which the switching is achieved is presented first.

As described in Chapter 2 and [32], the determination of which coils should be energized is achieved by first grouping them into adjacent combinations. These may be viewed as combinations 1&3, 2&3, 1&4, and 2&4 (see Figure 2.2). Utilizing the desired force direction and the *geometric* vector field associated with the coil magnetics, an orthonormal basis may be formed which allows for an evaluation of whether or not a viable solution would even exist for a given coil combination. To achieve this, the components of the desired magnetic force may be expressed as a function of one another

$$f_{Mag,Z} = \tan(\theta) f_{Mag,Y},$$

where θ is the angular direction of the required force measured from the Y axis with the counter clockwise direction taken as positive. Utilizing the Y and Z components of (2.4) allows for the formation of a matrix equation in coil currents given by

$$[J]^T[\mathbf{\Omega}][J] = 0,$$

where

$$[\mathbf{\Omega}] = \begin{cases} [\mathbf{k}_{FM,Y}], \text{ for } \theta = \frac{\pi}{2} \pm n\pi, \; n = 0, 1, 2.\dots \\ \\ [\mathbf{k}_{FM,Z}] - \tan(\theta)[\mathbf{k}_{FM,Y}], \text{ otherwise.} \end{cases}$$

The quantity $[J]$ is a 2×1 vector of the currents in the reduced coil set and the $[\mathbf{k}_{FM,Y}]$ and $[\mathbf{k}_{FM,Z}]$ are 2×2 matrices containing the $\mathbf{G}$ and $\partial\mathbf{G}/\partial S$ terms of the reduced set in accordance with (2.11a) and (2.11b). The matrices $[\mathbf{k}_{FM,Y}]$ and $[\mathbf{k}_{FM,Z}]$ are given by the quantities in (2.10), however G_S and $\partial G_S/\partial S$ are now 2×1 matrices with the first element corresponding to coil 1 or 2 and the second element corresponding to coil 3 or 4, depending on the coil combination being examined. To deduce the existence of a current solution for a given coil combination, one may now just examine the definiteness of $[\mathbf{\Omega}]$. For this solution, an indefinite or semi-definite $[\mathbf{\Omega}]$ corresponds to the existence of a solution.

To facilitate the inverse solution, it is necessary that the eigenvalues of $[\mathbf{\Omega}]$ be real. This may be achieved by forcing symmetry through the relation $[J]^T[\mathbf{\Omega}][J] = [J]^T[(\mathbf{\Omega} + \mathbf{\Omega}^T)/2][J] = [J]^T[\mathbf{\Delta}][J]$. Once symmetry is established, an orthonormal basis $[\widetilde{\mathbf{Q}}]$ composed of the normalized eigenvectors of $[\mathbf{\Delta}]$ may be used to form the orthogonal transformation $[J] = [\widetilde{\mathbf{Q}}][\widetilde{J}]$. This transformation in concert with the eigenvalues of $[\mathbf{\Delta}]$ may be used to solve for the $[\widetilde{J}]$ solution as given in Chapter 2 and summarized here for convenience

$$\begin{aligned} \tilde{j}_i &= \pm\frac{3f_{Mag,Y}}{4\pi\mu_0 r^3 M}\frac{\sqrt{[\xi_i^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[G][\widetilde{\mathbf{Q}}][\xi_i^{\pm}]}}{[\xi_i^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[\mathbf{k}_{FM,Y}][\widetilde{\mathbf{Q}}][\xi_i^{\pm}]}, \\ \tilde{j}_{ii} &= \pm\tilde{j}_i\sqrt{\frac{-\lambda_i}{\lambda_{ii}}}, \end{aligned}$$

where

$$[\xi_i^{\pm}] = \begin{bmatrix} 1 & \pm\sqrt{-\lambda_i/\lambda_{ii}} \end{bmatrix}^T,$$

λ_i is the first eigenvalue, and λ_{ii} is the second eigenvalue of $[\mathbf{\Delta}]$, respectively. It should also be noted that the subscripts $i = 1, 2$ and $ii = 3, 4$ and these represent the solutions for different coil combinations. As can be seen, (2.21) requires that $\lambda_{ii} \neq 0$ for a solution to exist. A similar formulation may be made in which $\lambda_i \neq 0$ is necessary and is given here as

$$\begin{aligned} \tilde{j}_i &= \pm\tilde{j}_{ii}\sqrt{\frac{-\lambda_{ii}}{\lambda_i}}, \\ \tilde{j}_{ii} &= \pm\frac{3f_{Mag,Y}}{4\pi\mu_0 r^3 M}\frac{\sqrt{[\xi_{ii}^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[G][\widetilde{\mathbf{Q}}][\xi_{ii}^{\pm}]}}{[\xi_{ii}^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[\mathbf{k}_{FM,Y}][\widetilde{\mathbf{Q}}][\xi_{ii}^{\pm}]}, \end{aligned}$$

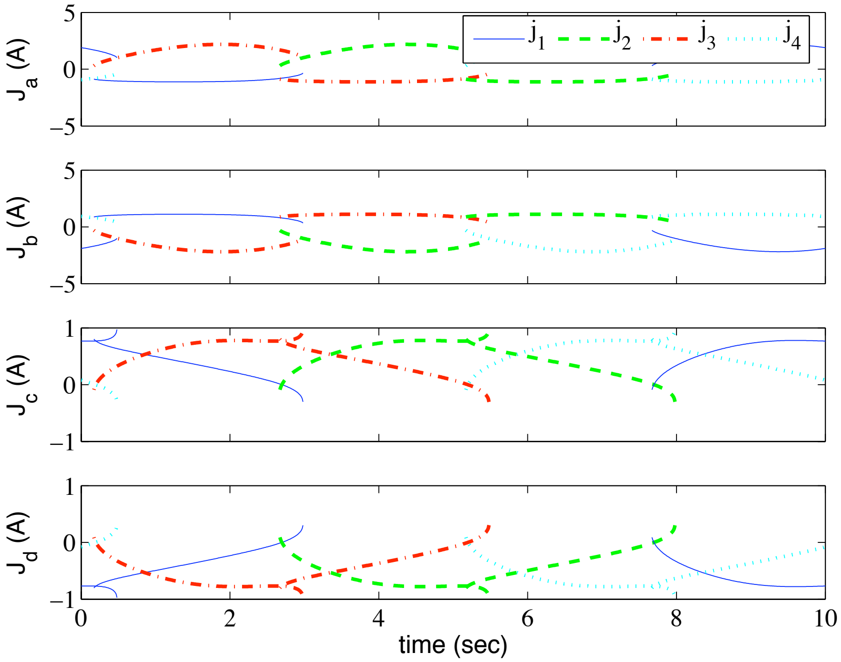

Figure 5.1: Calculated current trajectories for different coil combinations.

where

$$[\xi_{ii}^{\pm}] = \left[\begin{array}{cc} \pm\sqrt{-\lambda_{ii}/\lambda_i} & 1 \end{array} \right]^T .$$

Additionally, solutions may be formulated using $f_{Mag,Z}$. This will result in forms similar to (2.21) and (2.22) with $f_{Mag,Y} \rightarrow f_{Mag,Z}$ and $[\mathbf{k}_{FM,Y}] \rightarrow [\mathbf{k}_{FM,Z}]$. As has also been shown in Chapter 2, once the $[\widetilde{J}]$ solution has been calculated, it must be transformed back into the $[J]$ space via the transformation $[J] = [\widetilde{\mathbf{Q}}][\widetilde{J}]$. The remaining aspect of the inverse solution is to establish the proper signs. As a reminder these are available in Section 2.3, tables 2.2, 2.3, 2.4, and 2.5.

As an example of the inverse current solution, Figure 5.1 shows the four possible calculated current trajectories necessary for the spherical particle shown in Figure 4.10 to follow a 1" diameter circular path orbiting the origin once every 10 seconds. As can be seen, for the majority of instances there is only one coil combination which will satisfy the force requirement. However, as evidenced from the plot this is not the case at the ends of traces. Each coil combination solution *overlaps*. Because of this, a stronger switching algorithm beyond just examining the definiteness of the orthonormal basis must be used. To achieve a single coil set solution for all instances, it was decided to use the coil set solution with the smallest norm, but still greater than zero for the instances where overlapping solutions exist.

5.2 Closed Loop Control Law

The closed loop control law will utilize the inverse current solution in the controller. The basic concept consists of employing the error signal as the required force input to the inverse current solution. One may extrapolate this even further and utilize a classical "PD" control block feeding into an inverse current solution block. Indeed, this is the first of the control strategies implemented in hardware for this experiment. The classical technique was chosen as an obvious initial control strategy for evaluation of this electromagnetic system utilizing the minimal coil set. Additionally, because the system hardware works in a voltage control mode and the inverse solution calculates a current, the required *voltage* to achieve the necessary applied force on the spherical particle must be calculated. This is achieved by an inverse electrical dynamics calculation. This formulation utilized the coil self inductance and mutual inductance of the reduced set as well as the coil resistance to evaluate the necessary applied voltage to the coils. To conceptualize this controller, one may look at the system in the most simple terms. The sphere moving in a linear Newtonian fluid subject to a magnetic body force may be described with the non-homogenous differential equation

$$m\ddot{\vec{S}} + c\dot{\vec{S}} = \vec{F}_{Mag},$$

where m is the mass of the sphere, c is a constant proportional to the fluid drag on the sphere, and $\vec{F}_{Mag}$ is the applied magnetic force on the sphere. If the applied magnetic force is represented as a linear function of the position error $\vec{e} = \vec{S}_r - \vec{S}$ and velocity error $\dot{\vec{e}} = \dot{\vec{S}}_r - \dot{\vec{S}}$, where $\vec{S}_r$ and $\dot{\vec{S}}_r$ are the desired position and velocity, respectively, then the non-homogeneous system may be rewritten as

$$\begin{aligned} m\ddot{\vec{S}} + c\dot{\vec{S}} &= k_P\vec{e} + k_D\dot{\vec{e}}, \\ &= k_P(\vec{S}_r - \vec{S}) + k_D(\dot{\vec{S}}_r - \dot{\vec{S}}), \end{aligned} \tag{5.1}$$

where k_p and k_d are constant gains. Rearranging (5.1) results in the non-homogenous differential equation

$$m\ddot{\vec{S}} + (c + k_D)\dot{\vec{S}} + k_P\vec{S} = k_p\vec{S}_r + k_d\dot{\vec{S}}_r. \tag{5.2}$$

As can be seen, the homogeneous system in (5.2) is asymptotically stable because the coefficients m, c, k_P, and k_D are guaranteed to always be positive. Hence, the total stability of the system is guaranteed by the unperturbed linear system provided that the non-homogeneous forcing function of (5.2) is bounded [39].

The control law for the PD controller utilizing the inverse current solution may be formulated by same procedure adhered to in the previous passage. The desired magnetic force is assigned to be a function proportional to the position and velocity error of the magnetized sphere,

$$\begin{aligned}\vec{F}_{Mag,FM} &= k_P \vec{e} + k_D \dot{\vec{e}}, \\ &= k_P \begin{bmatrix} e_Y \\ e_Z \end{bmatrix} + k_D \begin{bmatrix} \dot{e}_Y \\ \dot{e}_Z \end{bmatrix}.\end{aligned}$$

This may then be used with (2.15) and (2.17) to establish the correct coil combination to effect the desired force. Once the correct coil combination has been selected, (2.21) and (2.22) may be used along with the observation that $f_{Mag,Y} \rightarrow f_{Mag,Z}$ and $[\mathbf{k}_{FM,Y}] \rightarrow [\mathbf{k}_{FM,Z}]$ to formulate the reduced coil set current solution

$$[J] = [\widetilde{\mathbf{Q}}][\mathbf{\Upsilon}][E] \begin{bmatrix} k_P \\ k_D \end{bmatrix}, \tag{5.3}$$

where $[\widetilde{\mathbf{Q}}]$ is the orthogonal coordinate transform corresponding to the selected coil combination,

$$[\mathbf{\Upsilon}] = \begin{bmatrix} \pm\frac{3}{4\pi\mu_0 r^3 M} \frac{\sqrt{[\xi_1^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[G][\widetilde{\mathbf{Q}}][\xi_1^{\pm}]}}{[\xi_1^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[\mathbf{k}_{FM,Y}][\widetilde{\mathbf{Q}}][\xi_1^{\pm}]} & 0 \\ 0 & \pm\frac{3}{4\pi\mu_0 r^3 M} \frac{\sqrt{[\xi_2^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[G][\widetilde{\mathbf{Q}}][\xi_2^{\pm}]}}{[\xi_2^{\pm}]^T[\widetilde{\mathbf{Q}}]^T[\mathbf{k}_{FM,Z}][\widetilde{\mathbf{Q}}][\xi_2^{\pm}]} \end{bmatrix},$$

$$[E] = \begin{bmatrix} e_Y & \dot{e}_Y \\ e_Z & \dot{e}_Z \end{bmatrix}.$$

It should be noted that in the event of a zero valued eigenvalue of the corresponding matrix $[\mathbf{\Delta}]$, different forms of the diagonal elements may be used and are available from (2.21) and (2.22). As can be seen, (5.3) provides the resulting current applied to the reduced coil set as a function of the position and velocity errors. Because the system will operated in a *voltage* control mode, the calculated current must be translated to a required voltage to be supplied to the electromagnets. The reduced coil set voltage vector may be achieved by first calculating the time derivative of the current vector $[J]$ and then applying the relationship

$$[V] = [R][J] + [L][\dot{J}] \tag{5.4}$$

to calculate the required voltage to be applied to the electromagnets. It should be noted that the matrices $[R]$ and $[L]$ in (5.4) contain only the resistance and inductance terms corresponding to the reduced coil set. The formulation of $[\dot{J}]$ can prove to be quite complicated, therefore in this study it will be evaluated numerically. To apply the control law to the state space model given in (2.27), the voltage relation given in (5.4) will need to be modified by adding a column and row of zeros corresponding to which electromagnets are absent in the reduced coil set. With the modification of (5.4), the closed loop PD controller utilizing the inverse current solution has been forumulated. A block diagram for the PD based control system is shown in Figure 5.2.

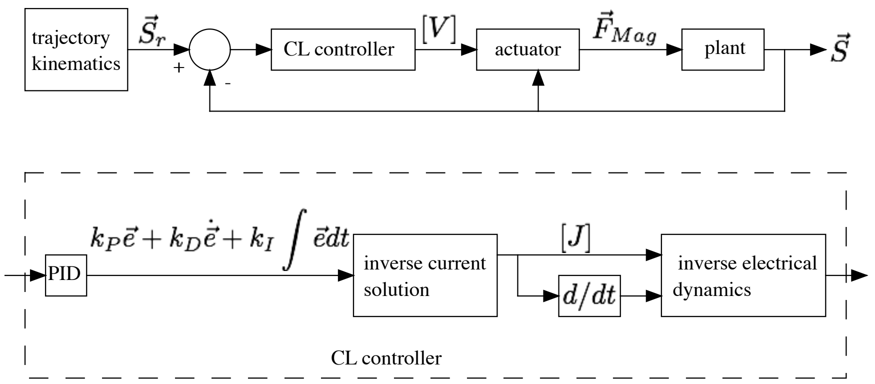

Figure 5.2: Block diagram of closed loop control simulation with a classical "proportional" controller is implemented. The upper diagram is the closed loop system while the lower diagram is a detailed view of the controller.

The second control strategy is similar to the strategy outlined in the previous passage, however with the addition of a feedforward block to estimate the required forces based on the inverse kinetics of the particle/fluid dynamics. Additionally, a simpler classical control block consisting of proportional control alone will be implemented. Following the same logical progression as that employed for the PD based controller, the non-homogeneous differential equation describing this system where the applied magnetic body force is defined as a function of the position error and a prediction of the particle dynamics based on a Stokes' flow approximation may be given as

$$\begin{aligned} m\ddot{\vec{S}}+c\dot{\vec{S}} &= k_P\vec{e}+m\ddot{\vec{S}}_r+6\pi r\mu\dot{\vec{S}}_r, \\ &= k_P(\vec{S}_r-\vec{S})+m\ddot{\vec{S}}_r+6\pi r\mu\dot{\vec{S}}_r. \end{aligned} \tag{5.5}$$

Rearrangement of (5.5) yields

$$m\ddot{\vec{e}}+(6\pi r\mu-c)\dot{\vec{S}}_r+k_P\vec{e}=0 \tag{5.6}$$

As can be seen from (5.6), the closed loop dynamics are asymptotically stable if the feedforward model damping coefficient value is larger than the actual fluid damping of the physical system. A block diagram for the feedforward based control system is shown in Figure 5.3.

The feedforward proportional controller may be formulated by first identifying the applied magnetic force as a function of the error signal and feedforward quantities, given here as

$$\begin{aligned} \vec{F}_{Mag,FM} &= k_P\vec{e}+m\ddot{\vec{S}}_r+6\pi r\mu\dot{\vec{S}}_r, \\ &= k_P\begin{bmatrix} e_Y \\ e_Z \end{bmatrix}+m\ddot{\vec{S}}_r+6\pi r\mu\dot{\vec{S}}_r. \end{aligned}$$

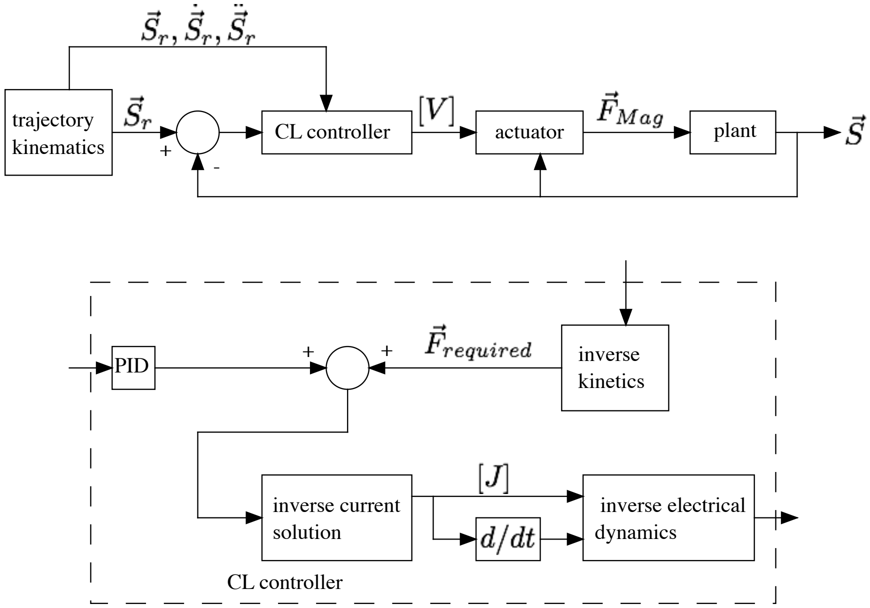

Figure 5.3: Block diagram of closed loop control simulation with a feedforward block. A classical “proportional” controller is implemented in concert with a feedforward block which estimates the desired forces on the spherical particle. The upper diagram is the closed loop system while the lower diagram is a detailed view of the controller.

As before, this may then be used with (2.15) and (2.17) to establish the correct coil combination to effect the desired force. The inverse current using a proportional controller in concert with feedforward may then be defined as

$$[J]_{ff} = [\widetilde{\mathbf{Q}}][\boldsymbol{\Upsilon}][\vec{e}]k_P + [\widetilde{\mathbf{Q}}][\boldsymbol{\Upsilon}][\ddot{\vec{S}}_r]m + [\widetilde{\mathbf{Q}}][\boldsymbol{\Upsilon}][\dot{\vec{S}}_r]6\pi r\mu. \tag{5.7}$$

As before, the required voltage for the proportional/feedforward controller may be formulated as

$$[V]_{ff} = [R][J]_{ff} + [L][\dot{J}]_{ff}. \tag{5.8}$$

The system was simulated using *MATLAB/Simulink*. Even though the inverse kinetics block *approximates* the flow as “Stokes” and neglects the friction effects, in this physical system friction and wall effects are present and therefore must be accounted for in the plant model for a simulation that is representative of the true system dynamics. For this system, the fluid drag force on the spherical particle deviates from the Stokes flow approximation due to the wall effect of the constraining plate on which the spherical particle sits and slides. An analytical formulation for this effect has been investigated in [40, 41, 22, 24]. Additionally, this formulation has been numerically and experimentally validated in [42, 23]. Application of this allows for an approximation of the fluid

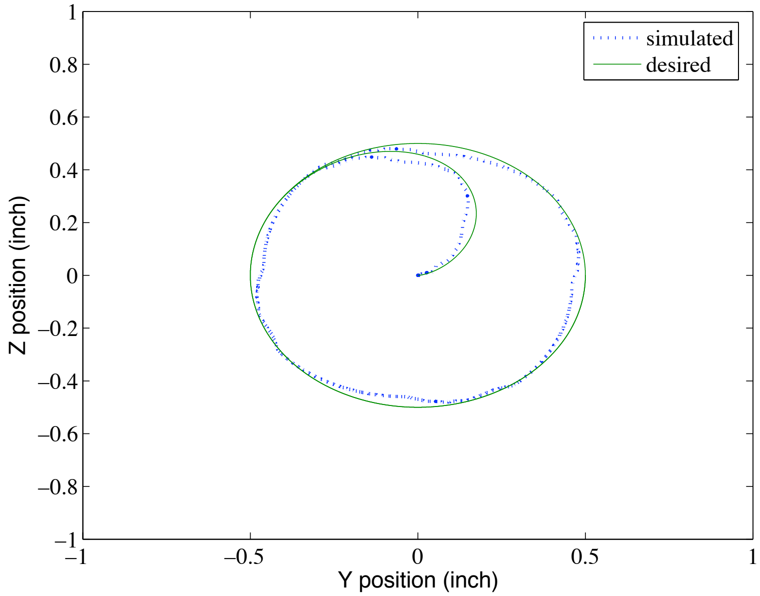

Figure 5.4: Simulation of closed loop PD controller resultant trajectory following a spiral reference.

drag force to be $\vec{F}_d \approx -6\pi\mu r\dot{\vec{S}}(1.5625)$. The friction force from the spherical particle sliding along the constraining plate will be represented as $\vec{F}_{fr} \approx -\mu_k mg(\dot{\vec{S}}/|\dot{\vec{S}}|)$ when the spherical particle is moving and $\vec{F}_{fr} = 0$ when the spherical particle is not. Note that μ_k is the kinetic coefficient of friction and for this study it is approximately valued at $\mu_k \approx 0.3$.

The desired trajectory utilized in the closed loop simulation is that of a spiral beginning from the origin with zero velocity and asymptotically approaching a circular orbit. This trajectory was chosen to evaluate this method because of the constant directional change of spherical particle motion and the challenge incurred upon an attempted tracking. So as not to exceed the capabilities of the hardware, a twenty second orbit trajectory will be examined. This corresponds to steady state angular velocity of $\dot{\theta}_{ss} = 2\pi/20\, rad/s$. The particular current solution selected for use in the controller was the set corresponding to the least current needed with both powered coils having positive current. This correlates with set "d" shown in Figure 5.1. The simulation results for the PD based control are shown in Figures 5.4 and 5.5. Additionally, a simulation of the controller utilizing a feedforward block (see Figure 5.3) with proportional control alone ($k_P = 0.3$) has been performed. These results are shown in Figures 5.6 and 5.7. From these plots it is evident that track following is achievable utilizing the reduced coil set in concert with a simple classical controller. However, implementation of this concept in hardware is critical to ultimately proving the concept.

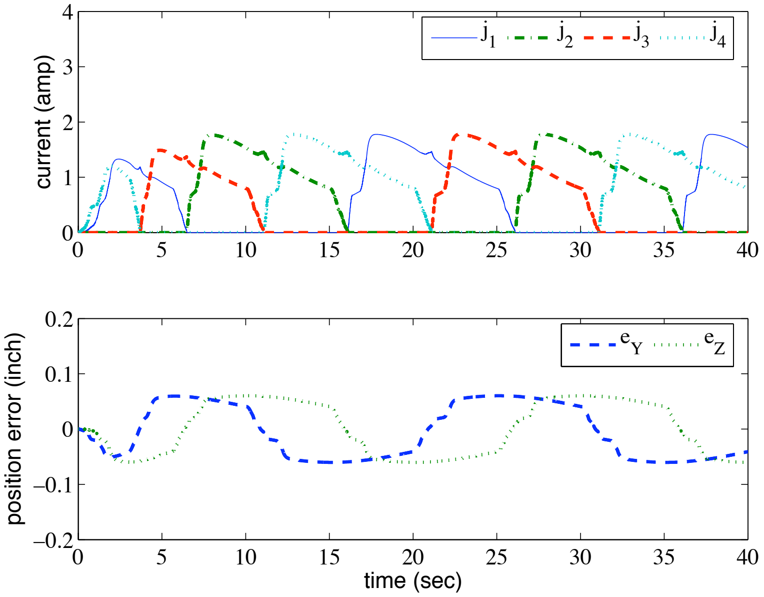

Figure 5.5: Simulation of closed loop PD controller resultant current and position error following a spiral reference.

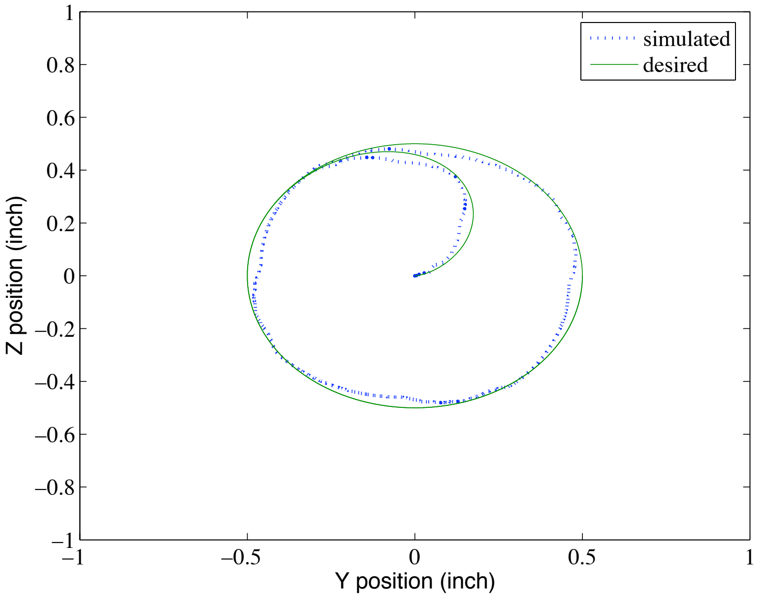

Figure 5.6: Simulation of closed loop feedforward controller resultant trajectory following a spiral reference.

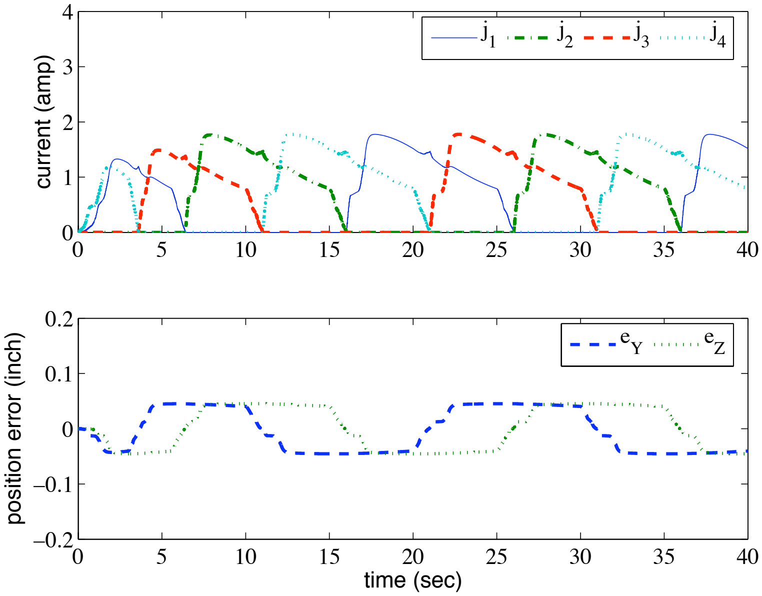

Figure 5.7: Simulation of closed loop feedforward controller resultant current and position error following a spiral reference.

5.3 Closed Loop Experimental Results

Two closed loop controllers, one with and one without a feedforward block, were implemented on the hardware setup. The results for an attempted tracking of the spiral trajectory for the non-feedforward controller with gains valued at $k_P = 0.3$ and $k_D = 0.001$ are available in Figures 5.8 and 5.9. As is evidenced from the plots, reasonable trajectory tracking was accomplished. The gains utilized were initially inferred from simulation results and then verified experimentally. As can be seen, gains established from the simulation result in a stable control system. Additionally, this control scheme results in a maximum error approximately equal to one particle diameter (≈ 0.125”). A second experiment utilizing proportional control alone in concert with a feedforward inverse kinetics block was performed. Once again, the gain value was inferred from the simulation results and implemented in experiment. As before, the desired trajectory is that of the spiral beginning at the origin and having a 20 second orbit. The results from this experiment are shown in Figures 5.10 and 5.11. Indeed the experimental results correlate well with the simulation results of Section 5.2 indicating that the system model reflects the actual plant. The more elementary proportional control was chosen to emphasize the improvement achieved by feedforward with this particular system. As can be seen, the position error is significantly reduced when using feedforward action by comparing

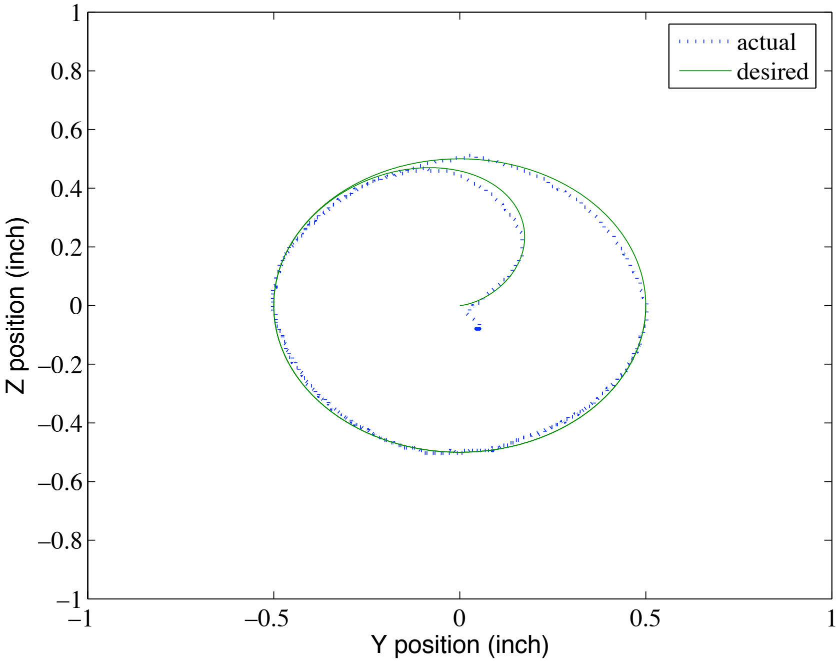

Figure 5.8: Experimental closed loop controller resultant trajectory following a spiral reference.

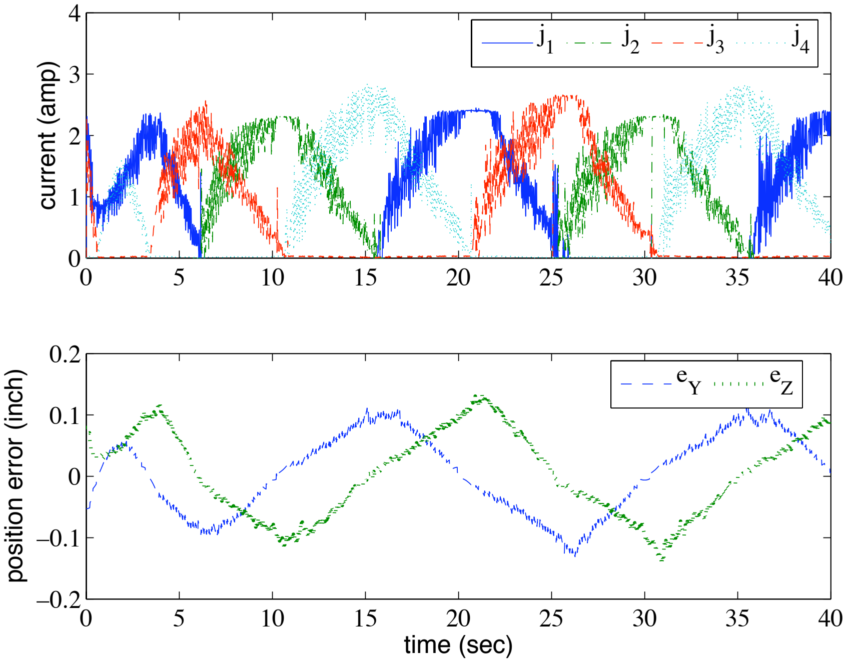

Figure 5.9: Experimental closed loop controller resultant current and position error following a spiral reference.

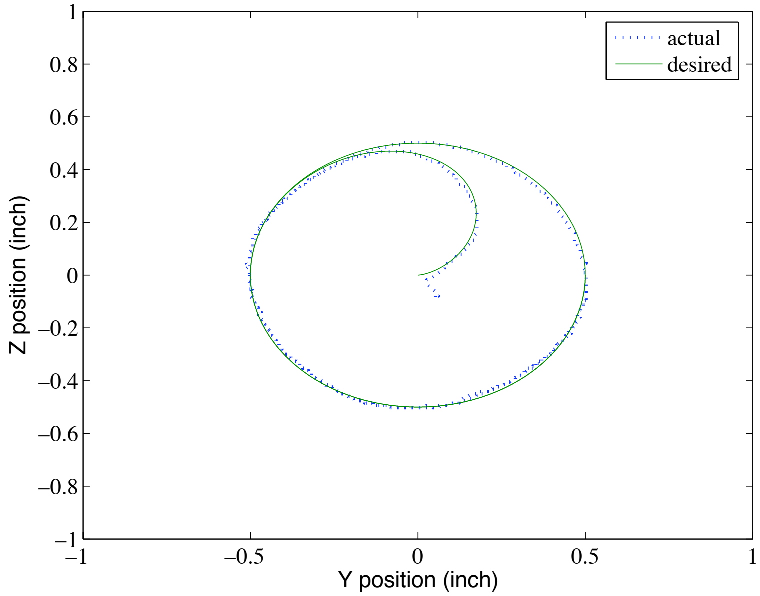

Figure 5.10: Experimental closed loop controller with feedforward block resultant trajectory following a spiral reference.

the results shown in Figures 5.9 and 5.11. In addition, it can be observed that this improvement comes with little change in the required current magnitude.

In an effort to further explore the performance capabilities of these control strategies, some additional trajectories were examined experimentally. The next trajectory examined is that of a square. The sphere begins at the origin and moves to the corners of a "box", following a linear path to each corner point. The trajectory also calls for the sphere to come to a stop at each corner location. Each linear segment of the square trajectory is traversed in 20 seconds. The performance results of the PD controller used in concert with the inverse solution are shown in Figures 5.12 and 5.13. The gains utilized are the same that were applied to the spiral trajectory, $k_P = 0.3$ and $k_D = 0.001$. As can be seen, a stable control condition was achieved again and the sphere tracked the square path quite well with a maximum position error less than a diameter in value. The proportional/feedforward controller used in concert with the inverse current solution was also applied to the square trajectory. In a similar fashion, the proportional gain is the same as that used in the spiral trajectory, $k_P = 0.3$. The results of this experiment are shown in Figures 5.14 and 5.15. Once again stable tracking is achieved and an even better error performance attained by implementation of a feedforward block. This highlights once again the model closely resembling the physical system.

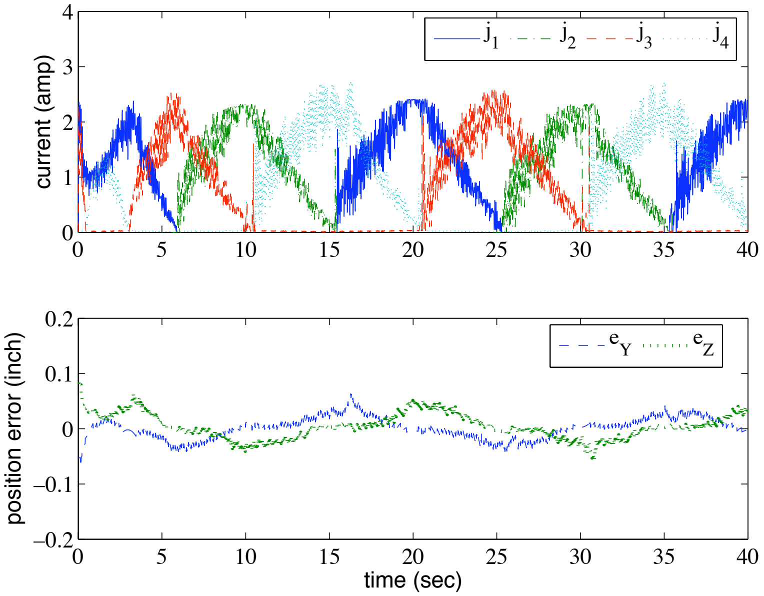

Figure 5.11: Experimental closed loop controller with feed forward block resultant current and position error following a spiral reference.

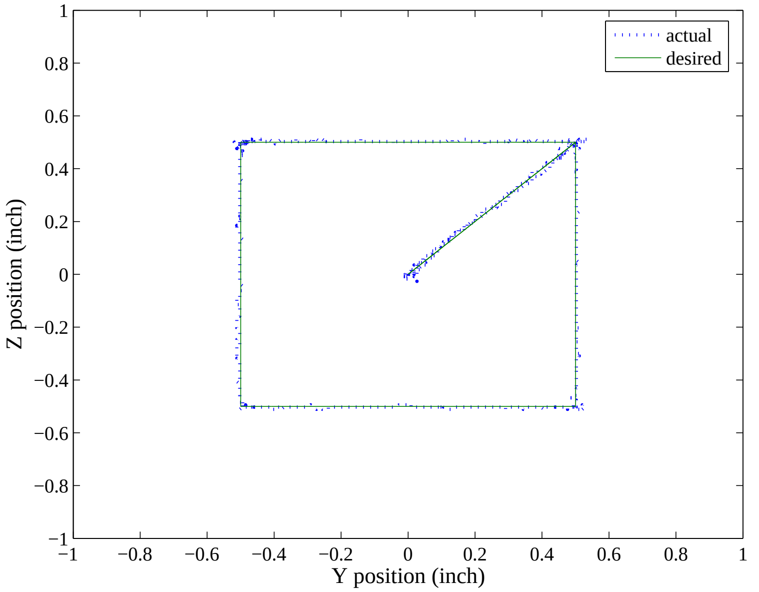

Figure 5.12: Experimental closed loop controller resultant trajectory following a square reference.

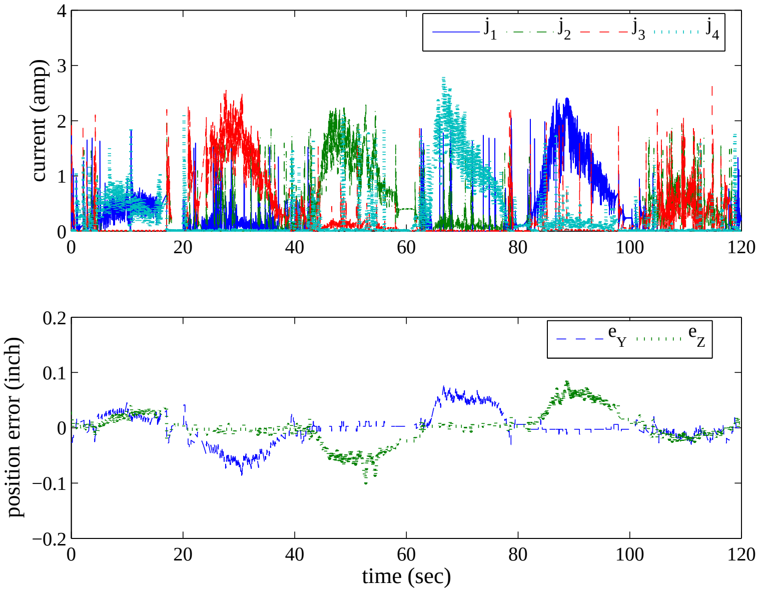

Figure 5.13: Experimental closed loop controller resultant current and position error following a square reference.

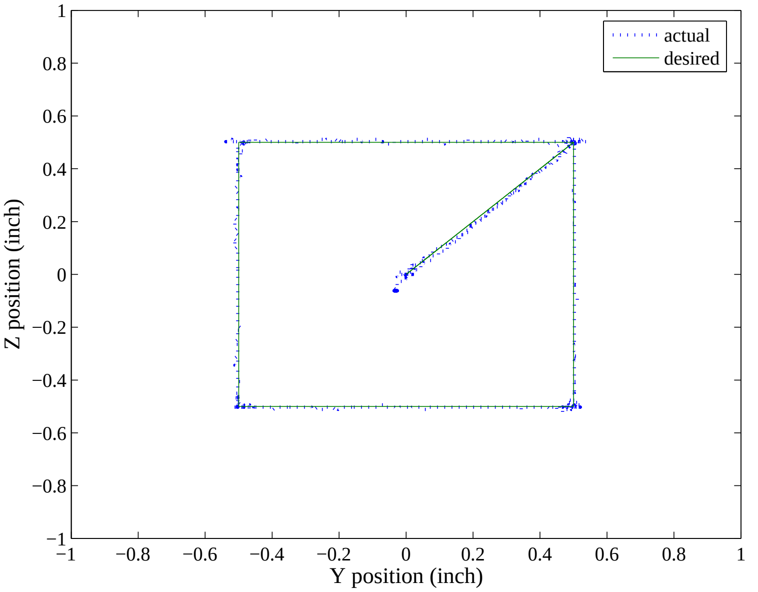

Figure 5.14: Experimental closed loop controller with feedforward resultant trajectory following a square reference.

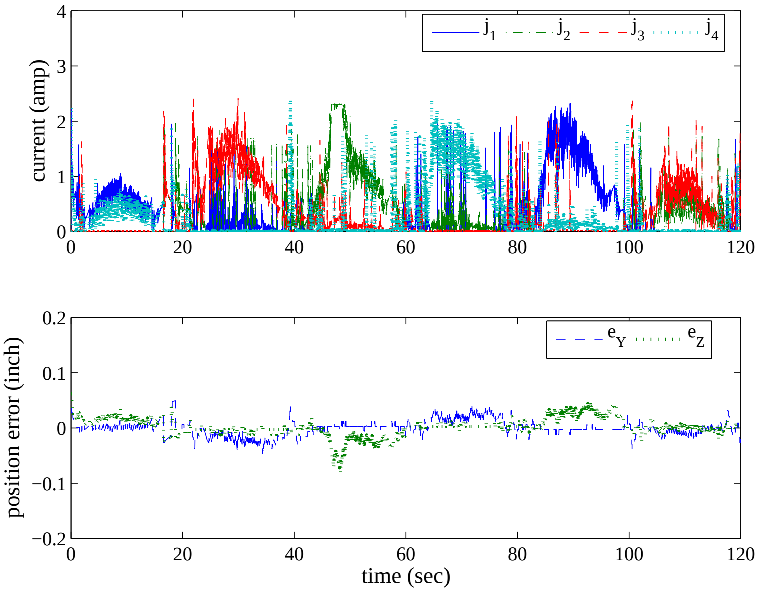

Figure 5.15: Experimental closed loop controller with feedforward resultant current and position error following a square reference.

A third trajectory type was also examined experimentally. This problem is analogous to the point regulation and tracking problem in which a point position in 2-D space is commanded and the sphere is allowed to move toward this position irrespective of the resulting path undertaken to approach the desired position. The commanded points correspond to the origin and corners of the previously investigated square path, however the linear paths between the points will no longer be used as reference trajectories. The sphere is commanded to "dwell" at the point locations for 10 seconds until a new location is commanded. It is obvious that the feedforward based control method may not be used in this problem as the path in which the sphere will undertake is not defined and hence may not be modeled and fed forward in the control loop. Because of this, only the PD controller used in concert with the inverse current solution will be attempted. The gains will be the same as the two previously mentioned experiments, $k_P = 0.3$ and $k_D = 0.001$. The results from the point regulation problem are shown in Figures 5.16 and 5.17. As can be seen in the figures, the nonlinear nature of this electromagnet system can be seen. It is evident in the unusual path taken by the sphere to the corner points of the square trajectory. Although notwithstanding the erratic path, the desired points were reached and the sphere remained at those points in a stable fashion until the next point location was commanded. Additionally, an interesting observation may be made from Figure 5.17. As can be seen, the current limit of the hardware was reached as evidenced by

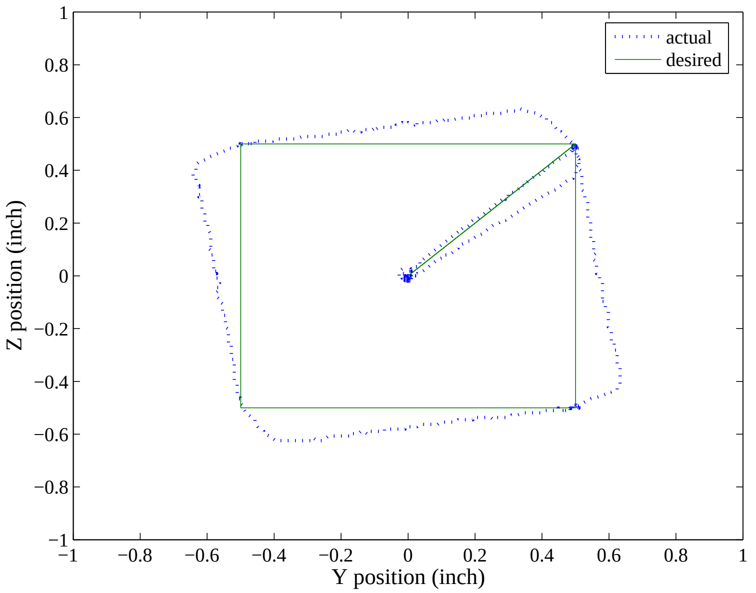

Figure 5.16: Experimental closed loop controller resultant trajectory attempting a point regulation.

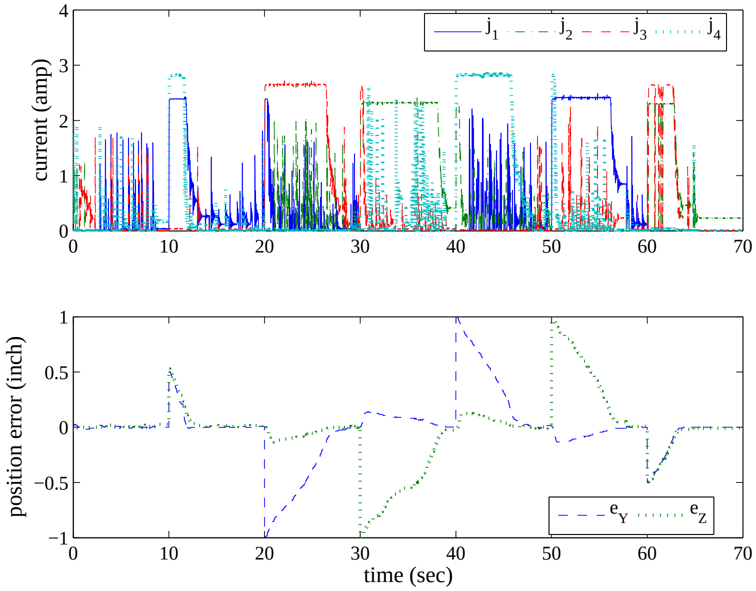

Figure 5.17: Experimental closed loop controller resultant current and position error attempting a point regulation.

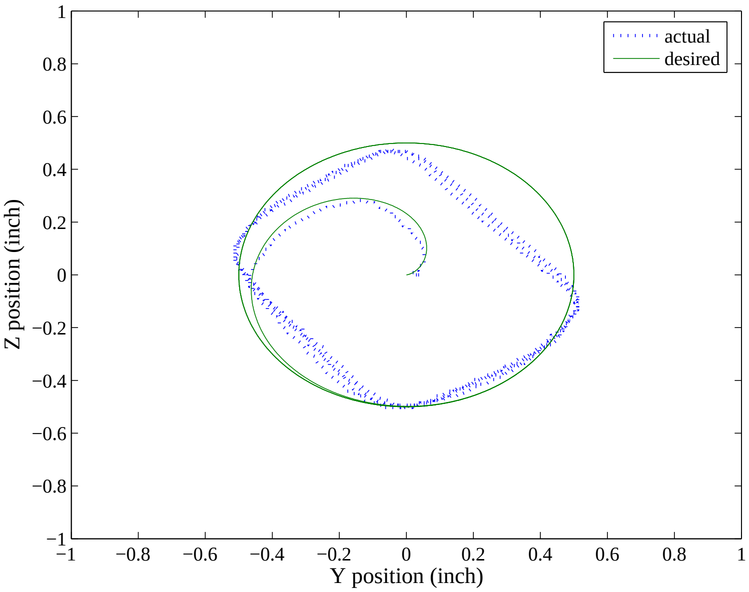

Figure 5.18: Experimental closed loop controller resultant trajectory attempting to track the fast spiral trajectory.

the “clipping” of the current signal. Further study may indicate that this has contributed to the irregular path taken by the sphere in the process of tracking to the desired point.

To gain a more in depth understanding of the capabilities of the hardware developed and the effects of hardware limitations on controller performance, the spiral and square trajectory experiments were also performed with the same controller but on a *faster* position trajectory. For the spiral trajectory, the orbit time was decreased from 20 seconds to 10 seconds. Additionally, for the square trajectory the rate at which the linear segments were traversed was decreased from 20 seconds to 10 seconds as well. The PD controller applied to the fast spiral trajectory results are shown in Figures 5.18 and 5.19. As can be seen, the PD based controller has a large amount of difficulty tracking the fast spiral. This is not surprising as the current limits of the hardware are evident in Figure 5.19. Upon inspection, one may see the subtle differences in the winding conditions and power circuits resulting in different upper end current capabilities between the four coils. For example, coil 1 obviously has a poorer performance than coil 4. Slightly better performance was achieved by the proportional/feedfoward based controller with the gain $k_P = 0.3$, however this appears to only have significant improvement in the error in one coordinate direction. This is evidenced in Figures 5.20 and 5.21. In dealing with the hardware limitations, little improvement may be achieved by utilizing a feedforward component with the gain of $k_P = 0.3$, however significant improvement was

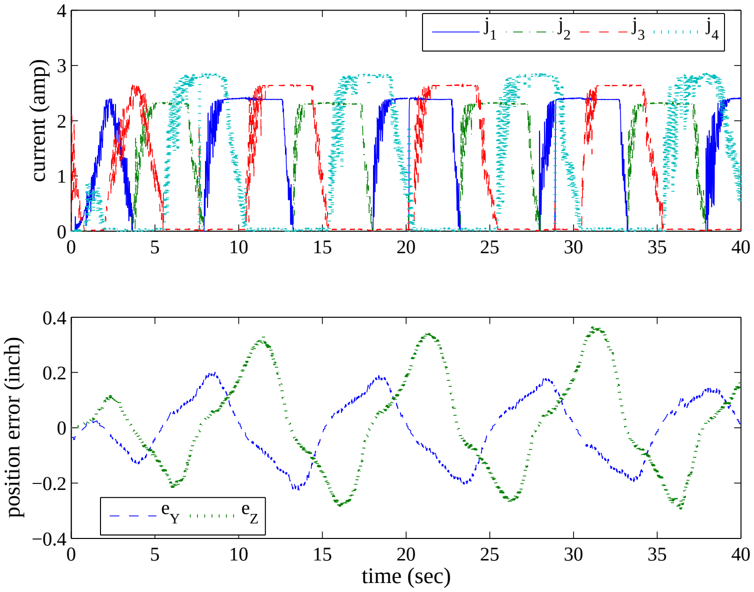

Figure 5.19: Experimental closed loop controller resultant current and position error attempting to track the fast spiral trajectory.

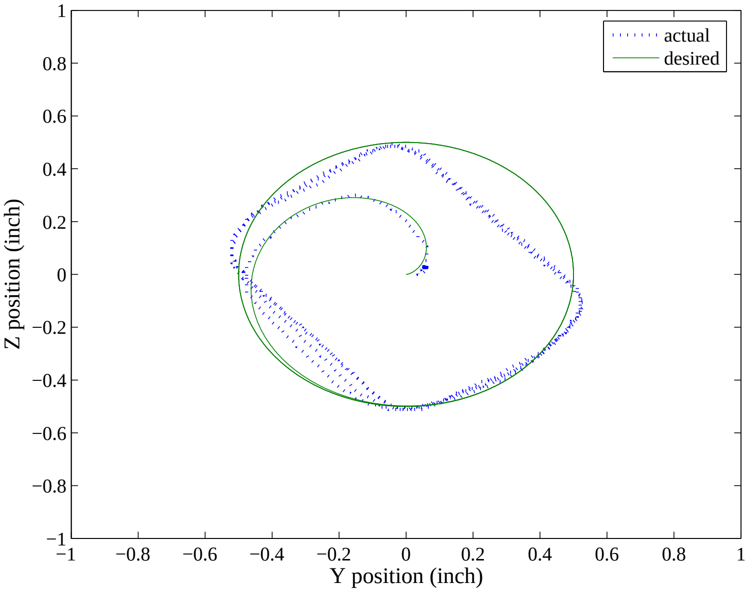

Figure 5.20: Experimental closed loop feedforward controller resultant trajectory attempting to track the fast spiral trajectory.

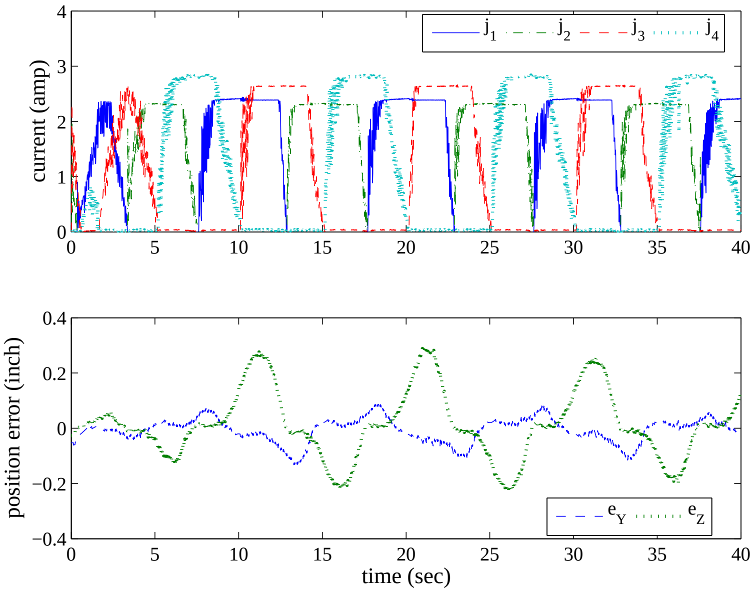

Figure 5.21: Experimental closed loop feedforward controller resultant current and position error attempting to track the fast spiral trajectory.

attained by adjusting this gain value to $k_P = 0.5$. The results of this experiment are available in Figures 5.22 and 5.23. As can be seen, the feedforward controller with the increased proportional gain actually makes a significant improvement of the system performance in spite of the hardware limitations. While the performance is not quite as satisfactory of that of the slower trajectory, a substantial performance gain is evident when Figure 5.23 is examined. The performance of this system on the faster square trajectory is slightly better. This is indicative of the square path chosen in this study as being less demanding than the spiral path. Performance of the PD based controller when tracking the fast square path with the original gains is available in Figures 5.24 and 5.25. As can be seen, the PD based controller tracks the square trajectory quite well. However, when Figures 5.25 and 5.13 are compared, the faster trajectory results in a slightly larger position error. The results for the proportional/feedforward based controller on the fast square trajectory are also quite good. These are shown in Figures 5.26 and 5.27. Upon comparing Figures 5.27, 5.25, and 5.15, one may observe the superior robustness of the feedforward based controller to that of the PD based system. As can be inferred, the increased speed of the fast square trajectory degrades the performance of the PD based controller much more than the feedforward system.

The final experimental result to be presented in this work is that of the performance of the *open loop* system. This is done to emphasize the importance and the need for implementation of a

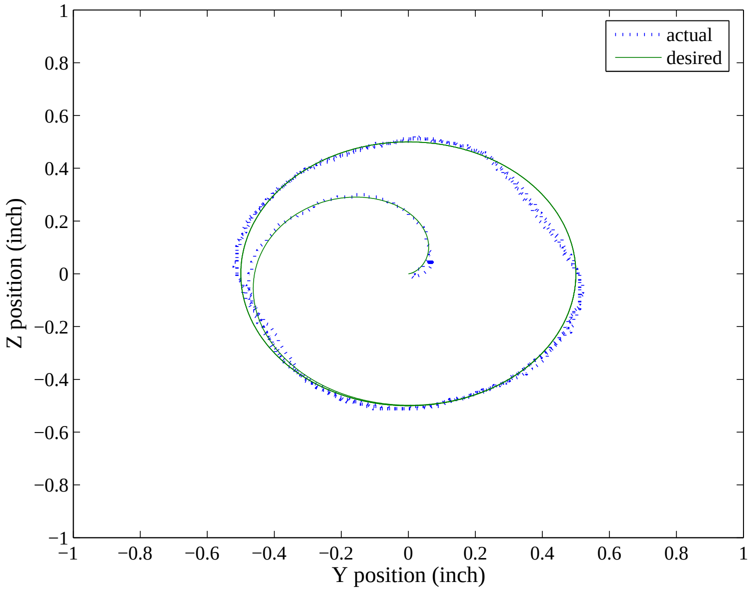

Figure 5.22: Experimental closed loop feedforward controller resultant trajectory attempting to track the fast spiral trajectory. ($k_P = 0.5$)

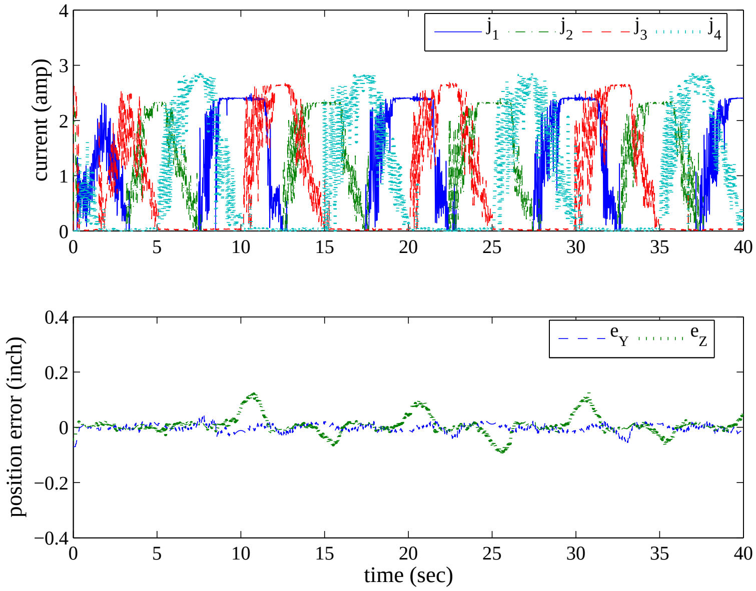

Figure 5.23: Experimental closed loop feedforward controller resultant current and position error attempting to track the fast spiral trajectory. ($k_P = 0.5$)

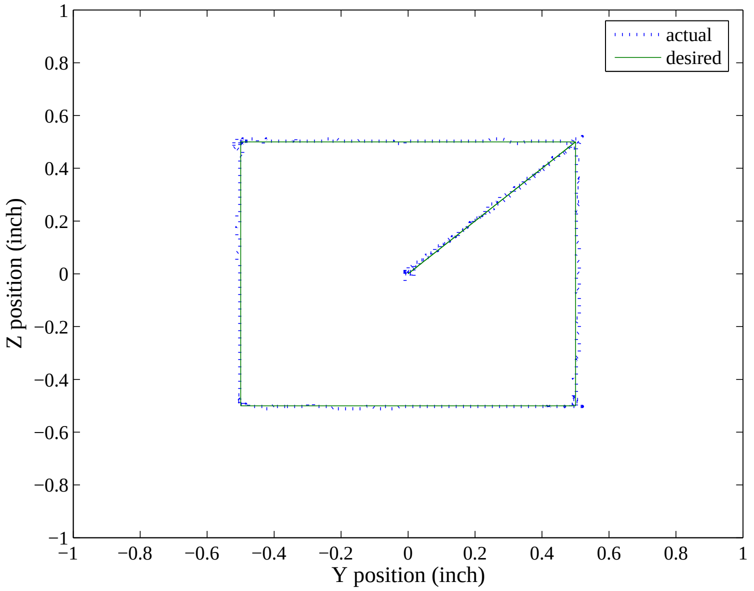

Figure 5.24: Experimental closed loop controller resultant trajectory following the fast square reference.

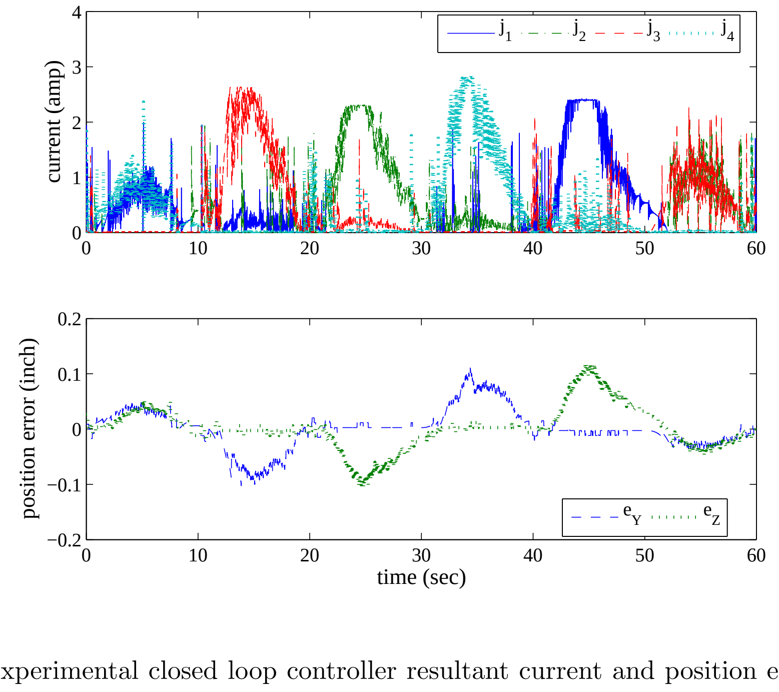

Figure 5.25: Experimental closed loop controller resultant current and position error following the fast square reference.

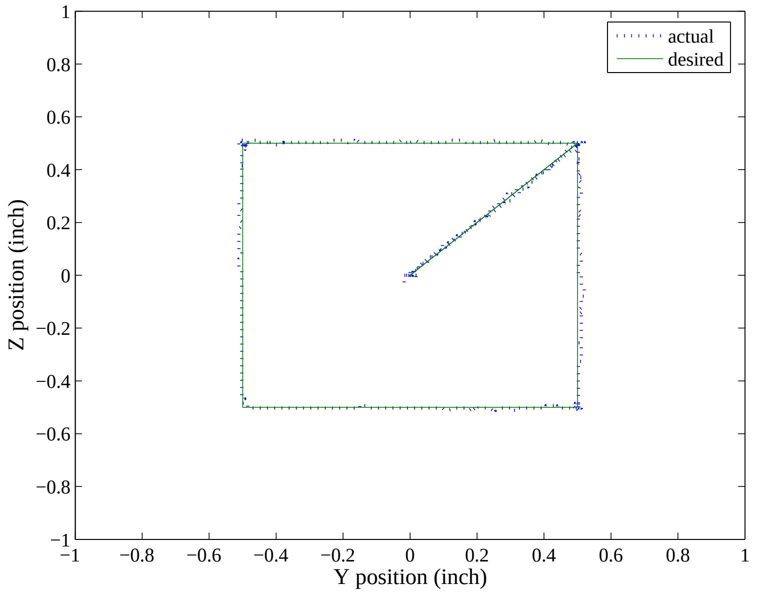

Figure 5.26: Experimental closed loop feedforward controller resultant trajectory following the fast square reference.

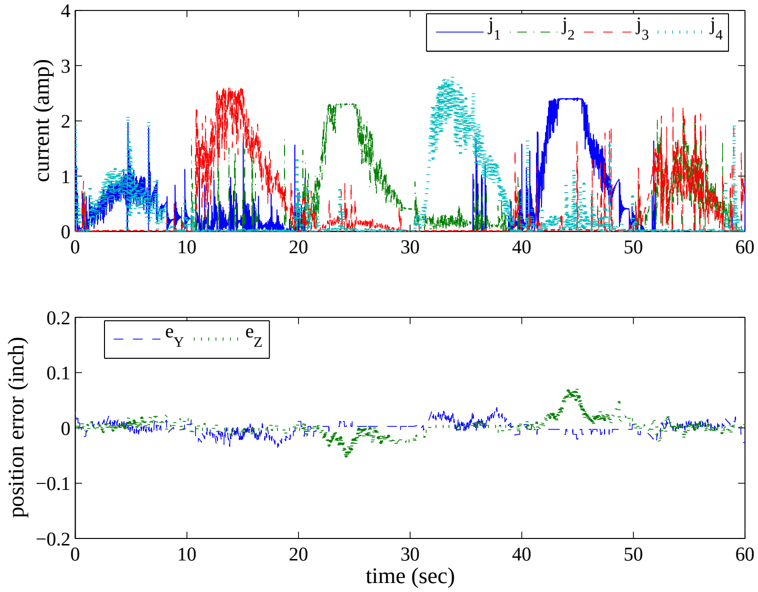

Figure 5.27: Experimental closed loop feedforward controller resultant current and position error following the square reference.

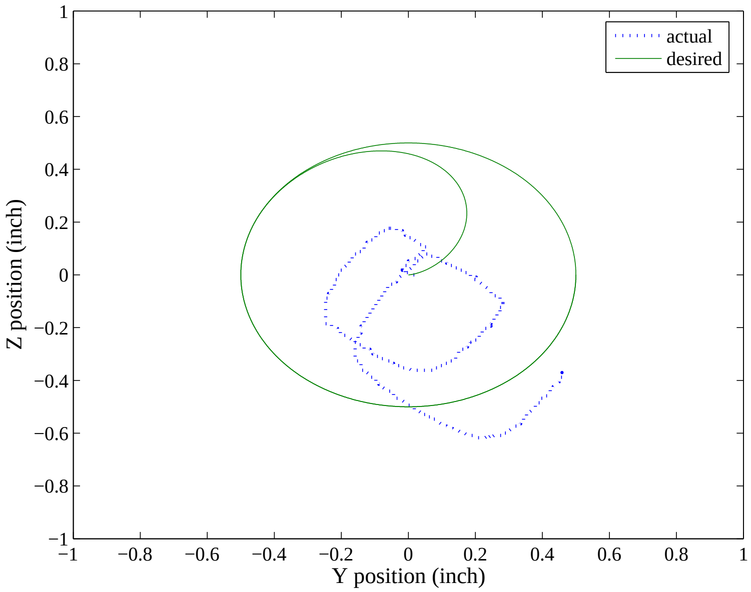

Figure 5.28: Experimental open loop controller resultant trajectory attempting to follow a spiral.

closed loop control on a system such as this. It may be hypothesized that for this simple system one may be able to "adjust" the fluid viscosity coefficient in the model and predict the required current trajectories resulting in a reasonable tracking of the desired position trajectory. This was performed and it was iteratively deduced that the best open loop tracking performance for the 20 second orbit spiral trajectory was achieved with a modification of the fluid kinematic viscosity in the model to be $\mu = 3.5N - s/m^2$. The results of this experiment are available in Figures 5.28 and 5.29. As can be seen the open loop tracking is quite poor. Even with what is arguably the simplest system for a technique such as this, a purely model based open loop controller is insufficient for affecting even a modest amount of trajectory tracking. The subtle deviations from the ideal model parameters warrants and demands the implementation of a feedback control method.

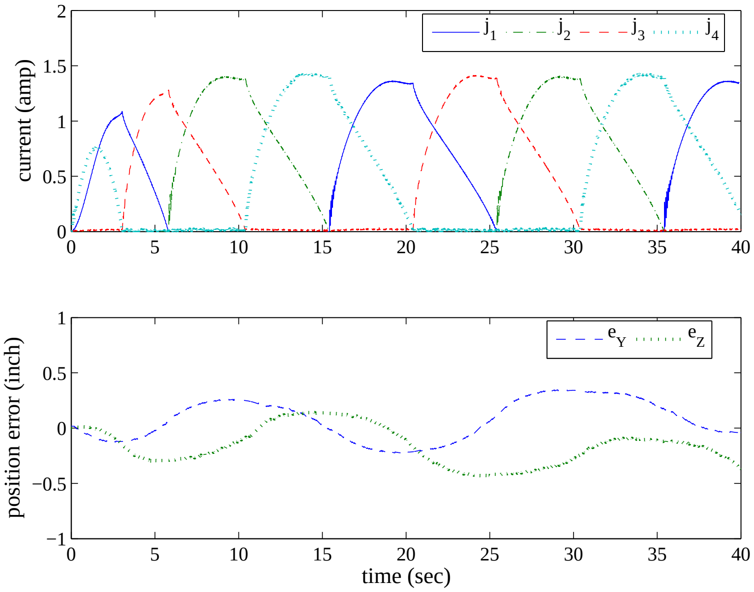

Figure 5.29: Experimental open loop controller resultant current and position error attempting to follow a spiral.

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions from this Study

A model describing a system to manipulate the position of a magnetic sphere in a fluid using stationary electromagnets has been developed. From an electromagnetic standpoint, the “attractive only” nature leads to an underdetermined system with multiple solutions. By reducing the number of active coils to only those necessary to produce the desired force, solution of the coil currents is feasible. However, reduction of the set to the minimal number of coils can result in regions of the trajectory space where certain directions of force may not be capable due to the narrowing of the force angle range. Nevertheless, if this characteristic is firmly understood it may be possible to design trajectories avoiding this condition. A technique for determining the coil combination candidates for a required force has also been developed. By examining the sign definiteness of a coil combination’s superpositioned geometric field function, one may establish whether a current solution for that coil combination even exists. In this work, this quantity is referred to as the $[\mathbf{\Omega}]$ matrix, and is proportional to a desired magnetic force at a given location in the plane. An interesting aspect of this technique is that it does not involve explicit calculation of the currents themselves. The $[\mathbf{\Omega}]$ matrix is a function of the desired magnetic force vector, geometric characteristics of the coils, and position of the sphere. The current flowing in the coils is absent from the $[\mathbf{\Omega}]$ matrix formulation.

Upon examination it is clear that with respect to the fluid drag force being represented as Stokes flow in the cited literature, this approximation may not be entirely accurate for all of the envisioned applications. When the target size approaches larger dimensions, as would be the case in the employment of a micro-robot or device, the Reynolds number describing the flow about the object may deviate from the linear region of drag force. Additionally the larger object may cause the assumption of an unbounded free stream to break down, requiring the model to account for channel flow effects. In these cases, accurate calculation or prediction of these dynamics may require modification of the model describing fluid drag similar to equations (2.6) and (2.7). In some flow conditions, a numerical model may be required to achieve the desired accuracy. Even with improved model precision, open loop control of this type of system proves to be inadequate or even impossible to achieve due to the inherent instability present when using an electromagnet as an actuation device. With that said, further exploration into closed loop control as a means to overcome flow disturbances and innate modeling errors is necessary, attempted, and achieved in this work. The state space model presented in this study is a firm footing to begin the closed loop system dynamic analysis.

An analytical model representing an elliptical coil wound on a rectangular bobbin as a series of rectangular filamentary conductors has been derived and presented. This formulation predicts the magnetic field and the partial derivative of the magnetic field with respect to the spatial variables in a local coordinate system attached to the coil. Homogeneous coordinate transforms may then be used to express the fields and field partial derivatives in a global coordinate system. This representation may then be used to calculate the resultant force on a permanently magnetized sphere. In addition, formulations for calculating the self inductance, mutual inductance, and resistance of the coils in a quad-coil configuration have been derived also based on a finite rectangular loop formulation. To gauge the accuracy of these formulations, the calculation results for magnetic field, resistance, and self inductance were compared to measurements made of these parameters on four coils which were wound to meet a specific inner dimension and weight. Results of these measurements were quite good. Modeled field values agreed quite well with the measured field. Indeed for most regions, the model resided within the $\pm 3\sigma$ error of the measurement. The calculated inductance value was lower than the average of the four coils, however it did fall within the $\pm 3\sigma$ distribution of the coil inductance measurements. The results of this study show that for the quad coil configuration to be used to impart planar motion of a magnetized sphere, the finite rectangular loop formulation is sufficient for a reasonable approximation to the physical system. This model may then be used as a predictive model to formulate an optimal design of the coil system for planar motion of the sphere.

While the model developed in [32] (Chapter 2) is sufficient for predicting an inverse current requirement for the reduced coil 2-D motion problem, it is insufficient for predicting a smooth current trajectory over time. Discontinuities will be present in the calculated trajectory if the type of transformation used in the inverse current solution is not accounted for and the set groupings given in Section 2.3 not observed. If care is not taken in this area, an unobtainable current trajectory will be required due to the fact that an instantaneous change in current is not achievable in most inductive systems. In addition, the calculated resultant magnetic force must be compared to the desired force to ensure that an erroneous result is not obtained. It may therefore be stated that the inverse solution alone is not “strong” enough. The inverse solution must be implemented in concert with a check of correct force direction for a complete solution. Also, an algorithm based on the gradient method of a discrete steepest descent was developed and realized. The functional used was the maximum norm of the instantaneous power required to move a sphere in a circular trajectory. The norm was chosen as a satisfactory metric to evaluate power applied to two coils at one time. The quadratic nature of the solution, in addition to the capability to reverse polarity of the coils results in four different current trajectories for any one coil combination. Hence, two possible maximal power norms are calculable. While either may be used in the minimization algorithm, the larger of the two was utilized and a minimal coil parameter set was evaluated. The minimization also hinted to the attribute that proximity of the coil to the object being propelled may be a significant factor. The optimization calculated a coil spacing which is as close to the origin as the constraint allowed, and coil dimensions were sized accordingly to a "close" electromagnet. This parameter set will provide the basis for a coil design which will enable open loop and closed loop control of sphere's motion in a 2-D plane and for the verification/evaluation of the reduced coil set concept. Fabrication and assembly of the hardware components to achieve planar motion was accomplished and implementation of the dynamic system experiment was performed.

In this study, an experimental setup to examine closed loop control for the 2D position of a magnetized spherical particle has been investigated. The electromagnet configuration consists of four coils arranged around a planar movement space. A technique in which only the minimal number of coils resulting in the desired force on the spherical particle is utilized and is central to the control technique. This method results in only *two* coils being powered at any one instant. A spiral trajectory with an initial particle position located at the origin and with zero initial velocity was initially chosen to evaluate the minimal coil set method due to the difficulty associated with a path which constantly changes direction. The system has been numerically simulated utilizing a controller with a feedforward action and a classical proportional control technique. The feedforward action is based on the inverse kinetics of the particle inertia and fluid dynamics, where the fluid drag model is based on the flow about the particle being approximated as Stokes flow. The simulation results show that trajectory tracking utilizing the coil switching technique is achievable, even with flow disturbances resulting in a deviation from the Stokes flow approximation. The technique of reducing the coil set to the minimally required number of coils results in stable tracking performance.

Experimental results are equally encouraging. Closed loop control was achieved with a performance that was quite satisfactory. A classical PD controller with no feedforward component was implemented in an experiment in which the spiral path was tracked. Results show that trajectory tracking was achieved with a peak error less than 0.125" for the trajectory chosen. Enhanced performance was also achieved by the implementation of feedforward action based on the developed model where the fluid flow dynamics is approximated as Stokes flow. The experimental results are indicative of the fact that while the system model may not be exact, its structure appears to match the real plant. Indeed, the performance improvement achieved by the application of a feedforward component were achieved with proportional control alone. In addition, the superior performance was attained with no increase in current magnitude. Additionally, other spatial trajectories were examined in experiment. The first consisted of a square path with a predetermined dwell time at each corner of the square. Good tracking performance was achieved with similar results as seen on the spiral path. The feedforward controller consistently outperformed the PD based control strategy. A point regulation problem was also examined in experiment. The corners of the square trajectory were input to the PD based controller and movement of the sphere to these points was achieved. However, a very erratic path was taken to the set points which was indicative of the nonlinear nature of the electromagnetic system and also highlighted the introduction of hardware limiting effects. Faster versions of the spiral and square trajectories were also examined in experiment. It was shown that while tracking of both “faster” trajectories was achieved, the robustness of the feedforward based controller was superior to that which possessed no feedforward component. These observations indicate significant improvement of performance for this system by utilizing a feedforward component in the control method. Satisfactory results from this initial experiment validate the minimal coil set concept.

6.2 Recommended Future Work

The solution developed in this Study is valid for the 2-D planar motion problem, however it is conceivable to expand this work to the full 3-D motion system. An extension into this case would entail increasing the reduced minimal coil set from two coils to three. In addition the current solution would be modified from an intersection of curves similar to conic sections to the intersection of quadric surfaces. While this may prove to be analytically and computationally challenging, the broadening of this technique to the 3-D case is believed achievable. Indeed, a three dimensional motion hardware system has been implemented as seen in [6, 7]. However, this system relies on numerical search techniques to deduce the necessary current vector and is also only operated in an open loop fashion. Implementation of the reduced coil set technique put forth in this study may prove to be quite advantageous. In addition, an examination of the number of coils utilized must be performed. As seen in this work, the number of coils employed for 2-D motion numbered four. In actuality, the minimal number of coils necessary to be present in the physical set is only *three*. In this case, the arrangement of the coils would not be on the edges of a square as seen in Figure 2.2, but on the edges of a *triangle*. For the three dimensional case, the minimal physical set population would number *four*. For this configuration, the electromagnets would be arranged on the surfaces of a *tetrahedral* pyramid with the inner volume of the pyramid being the motion space of the magnetic target object. As was observed however in this study, an examination of the effectiveness or volume where an effective force may be employed must be examined for these geometric arrangements of electromagnets.

In addition to dimensional expansion of the reduced coil technique, the study needs to be expatiated on *scale*, both in particle size and achievable motion area/volume of the target. The work done in this initial evaluation utilized a magnetized sphere with a diameter of 1/8". This would be considered to be on the upper end of the size spectrum pertaining to actual biomedical applications. A particle size significantly smaller than this needs to be employed in a further study. The complication associated with this lies in the fact that the achievable magnetic force on the target is a function of the object's volume, hence a significant reduction in the volume of the target results in a geometric increase in the required field and field gradients to produce a significant force on the target. Compounding the problem is the need to expand the target movement area. An increase in this plane/volume invariably results in an increase in the electromagnet spacing and the proximity of the target to the magnetizing and field generating coils. As has been shown in this study, the proximity of the target to the electromagnet is of upmost importance when considering the amount of power needed to facilitate motion. The attractive nature between the target and the electromagnets can be viewed as an "inverse square" relation, hence a geometric increase in the required field and field gradient results when the movement space is increased and/or the target size is decreased.

Furthermore, an expansion of this study to fluids of different viscosities and flow characteristics is necessary. This study utilized the rather viscous fluid medium of glycerine as the test base. In addition this fluid medium was static for all analysis and experimental results. Examination of a system in which different viscosities are present is necessary to further understand the capabilities of this hardware configuration in the biomedical application. Additionally, system performance with time invariant fluid flow patterns needs to be further investigated. Flows such as this are more representative of biomedical systems and must be examined in greater detail.

Finally, further system control improvements should be possible with an expansion from the classical controller to more advanced control techniques. As has been demonstrated in this study, the reduced coil concept is entirely feasible. Closed loop control has been accomplished utilizing classical methods as well, however these techniques may not suffice when the scale of the system is expanded. Increased electromagnet size and capabilities will no doubt result in larger inductive effects which can negatively affect the switching physics of the system. Additionally, the required responsiveness of the target object will inevitably be elevated when a flow disturbance is introduced. These reasons coupled with the fact that the 2-D and 3-D configurations of this system are inherently multi-input, multi-output systems warrant a further investigation into the application of more sophisticated control methodologies.

BIBLIOGRAPHY

[1] Q. A. Pankhurst, J. Connolly, S. Jones, and J. Dobson, "Applications of magnetic nanoparticles in biomedicine," *Journal of Physics D: Applied Physics*, vol. 36, pp. R167–R181, 2003.

[2] A. Senyei, K. Widder, and G. Czerlinski, "Magnetic guidance of drug-carrying microspheres," *Journal of Applied Physics*, vol. 49, no. 6, pp. 3578–3583, 1978.

[3] M. Zborowski, C. B. Fuh, R. Green, L. Sun, and J. J. Chalmers, "Analytical magnetapheresis of ferritin-labeled lymphocytes," *Analytical Chemistry*, vol. 67, no. 20, pp. 3702–3712, 1995.

[4] S. Østergaard, G. Blankenstein, H. Dirac, and O. Leistiko, "A novel approach to the automation of clinical chemistry by controlled manipulation of magnetic particles," *Journal of Magnetism and Magnetic Materials*, vol. 194, pp. 156–162, 1999.

[5] Z. G. Forbes, B. B. Yellen, K. A. Barbee, and G. Friedman, "An approach to targeted drug delivery based on uniform magnetic fields," *IEEE Transactions on Magnetics*, vol. 39, no. 5, pp. 3372–3377, 2003.

[6] R. G. McNeil, R. C. Ritter, B. Wang, M. A. Lawson, G. T. Gillies, K. G. Wika, E. G. Quate, M. A. Howard, and M. S. Grady, "Functional design features and initial performance characteristics of a magnetic-implant guidance system for stereotactic neurosurgery," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 8, pp. 793–801, 1995.

[7] R. G. McNeil, R. C. Ritter, B. Wang, M. A. Lawson, G. T. Gillies, K. G. Wika, E. G. Quate, M. A. Howard, and M. S. Grady, "Characteristics of an improved magnetic-implant guidance system," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 8, pp. 802–808, 1995.

[8] D. L. Holligan, G. T. Gillies, and J. P. Dailey, "Magnetic guidance of ferrofluidic nanoparticles in an *in vitro* model of intraocular retinal repair," *Nanotechnology*, vol. 14, pp. 661–666, 2003.

[9] J.-B. Mathieu, S. Martel, L. Yahia, G. Soulez, and G. Beaudoin, "Mri systems as a mean of propulsion for a microdevice in blood vessels," in *Proceedings of the 25th Annual International Conference of the IEEE EMBS*, September 2003.

[10] J.-B. Mathieu, S. Martel, L. Yahia, G. Soulez, and G. Beaudoin, “Preliminary investigation of the feasibility of magnetic propulsion for future microdevices in blood vessels,” *Bio-Medical Materials and Engineering*, vol. 15, pp. 367–374, 2005.

[11] B. Ciocirlan, D. G. Beale, and R. A. Overfelt, “Simulation of motion of an electromagnetically levitated sphere,” *Journal of Sound and Vibration*, vol. 242, no. 4, pp. 559–575, 2001.

[12] U. B. Sathuvalli and Y. Bayazitoglu, “The lorentz forces on an electrically conducting sphere in an alternating magnetic field,” *IEEE Transactions on Magnetics*, vol. 32, no. 2, pp. 386–399, 1996.

[13] D. C. Meeker, E. H. Maslen, R. C. Ritter, and F. M. Creighton, “Optimal realization of arbitrary forces in a magnetic stereotaxis system,” *IEEE Transactions on Magnetics*, vol. 32, no. 2, pp. 320–328, 1996.

[14] R. S. Newbower, “Magnetic fluids in the blood,” *IEEE Transactions on Magnetics*, vol. 9, no. 3, pp. 447–450, 1973.

[15] K. Ishiyama, K. Arai, M. Sendoh, and A. Yamazaki, “Spiral-type micro-machine for medical applications,” in *International Symposium on Micromechatronics and Human Science*, 2000.

[16] M. Sendoh, K. Ishiyama, and K. I. Arai, “Direction and individual control of magnetic micromachine,” in *Digests of the Intermag Conference*, 2002.

[17] M. Grady, M. A. H. III, R. C. Ritter, E. G. Quate, J. A. Molloy, M. A. Lawson, K. G. Wika, P. A. Ramos, and G. T. Gillies, “Magnetic stereotaxis system for neurosurgical procedures,” in *Proceedings of the International Instrumentation Symposium*, 1991.

[18] F. Creighton, *Control of Magnetomotive Actuators for an Implanted Object in Brain and Phantom Materials.* Ph.D. dissertation, University of Virginia, May 1996.

[19] H. E. Knoepfel, *Magnetic Fields.* John Wiley and Sons, 2000.

[20] H. A. Haus and J. R. Melcher, *Electromagnetic Fields and Energy.* Prentice Hall, 1989.

[21] S. Chikazumi, *Physics of Ferromagnetism.* Oxford University Press, 1997.

[22] J. Happel and E. Bart, “The settling of a sphere along the axis of a long square duct at low Reynolds’ number,” *Applied Scientific Research*, vol. 29, pp. 241–258, 1974.

[23] L. Zeng, S. Balachandar, and P. Fischer, "Wall-induced forces on a rigid sphere at finite Reynolds number," *Journal of Fluid Mechanics*, vol. 536, pp. 1–25, 2005.

[24] R. M. Wham, O. A. Basaran, and C. H. Byers, "Wall effects on flow past fluid spheres at finite Reynolds number: Wake structure and drag correlations," *Chemical Engineering Science*, vol. 52, no. 19, pp. 3345–3367, 1997.

[25] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra.* Society for Industrial and Applied Mathematics (SIAM), 2000.

[26] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control.* John Wiley and Sons, 1989.

[27] L. W. Johnson, R. D. Reiss, and J. T. Arnold, *Introduction to Linear Algebra.* Addison Wesley, second ed., 1989.

[28] M.-Y. Chen, C.-F. Tsai, and L.-C. Fu, "Design and control of a 2-dimensional electro-magnetic suspension actuator," in *Proceedings of the 2004 IEEE International Conference on Control Applications*, September 2004.

[29] M. Berlec, "An integral approach for 3d magnetostatic field calculation," *COMPEL The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 17, pp. 55–58, 1998.

[30] I. R. Ciric, "Simple analytical expressions for the magnetic field of current coils," *IEEE Transactions on Magnetics*, vol. 27, no. 1, pp. 669–673, 1991.

[31] J. William H. Hayt and J. A. Buck, *Engineering Electromagnetics.* McGraw-Hill, sixth ed., 2001.

[32] M. G. Duvall, P. R. Pagilla, and E. A. Misawa, "Modeling and analysis of a permanently magnetized spheres motion facilitated by field manipulation," in *Proceedings of the 45th IEEE Conference on Decision and Control*, December 2006. available on website: pagilla.ceat.okstate.edu.

[33] J. E. Marsden and A. J. Tromba, *Vector Calculus.* W. H. Freeman and Company/New York, third ed., 1988.

[34] P. Horowitz and W. Hill, *The Art of Electronics.* Cambrige University Press, second ed., 1989.

[35] R. W. Hornbeck, *Numerical Methods.* Prentice-Hall, Inc., 1975.

[36] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design.* PWS Publishing Company, 1996.

[37] D. A. Pierre, *Optimization Theory with Applications.* Dover Publications Inc., New York, 1986.

[38] W. S. Janna, *Introduction to Fluid Mechanics.* PWS Publishing Company, third ed., 1993.

[39] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Prentice Hall, 1991.

[40] P. Vasseur and R. Cox, "The lateral migration of spherical particles sedimenting in a stagnant bounded fluid," *Journal of Fluid Mechanics*, vol. 80, pp. 561–591, 1977.

[41] R. Cox and S. Hsu, "The lateral migration of solid particles in a laminar flow near a plane," *International Journal of Multiphase Flow*, vol. 3, pp. 201–222, 1977.

[42] A. Ambari, B. Gauthier-Manuel, and E. Guyon, "Wall effects on a sphere translating at constant velocity," *Journal of Fluid Mechanics*, vol. 149, pp. 235–253, 1984.

APPENDIX A

Partial Spatial Derivatives of g Field Components Resulting From a Single Rectangular Filamentary Loop

$$\begin{aligned}
\frac{\partial g_x}{\partial x} = \frac{1}{4\pi}\Bigg[& \frac{2z(x+a)(y+b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} + \frac{z(x+a)(y+b)}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
& + \frac{2z(x-a)(y-b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z(x-a)(y-b)}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{2z(x+a)(y-b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} - \frac{z(x+a)(y-b)}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{2z(x-a)(y+b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} - \frac{z(x-a)(y+b)}{[(x-a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}}\Bigg]
\end{aligned}$$

$$\begin{aligned}
\frac{\partial g_x}{\partial y} = \frac{1}{4\pi}\Bigg[& \frac{z}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} - \frac{z(y+b)^2}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \\
& - \frac{z}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z(y-b)^2}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{z}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} + \frac{z(y+b)^2}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
& + \frac{z}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} - \frac{z(y-b)^2}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}}\Bigg]
\end{aligned}$$

$$\begin{aligned}
\frac{\partial g_x}{\partial z} = \frac{1}{4\pi}\Bigg[& \frac{y+b}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} - \frac{2z^2(y+b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} \\
& - \frac{z^2(y+b)}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} - \frac{y-b}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} \\
& + \frac{2z^2(y-b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z^2(y-b)}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{y+b}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} + \frac{2z^2(y+b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} \\
& + \frac{z^2(y+b)}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} + \frac{y-b}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} \\
& - \frac{2z^2(y-b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} - \frac{z^2(y-b)}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}}\Bigg]
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial g_y}{\partial x} = \frac{1}{4\pi}\Bigg[& \frac{z}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} - \frac{z(x+a)^2}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{z}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z(x-a)^2}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{z}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} + \frac{z(x+a)^2}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
& + \frac{z}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} - \frac{z(x-a)^2}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial g_y}{\partial y} = \frac{1}{4\pi}\Bigg[& \frac{2z(x+a)(y+b)}{[(y+b)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} + \frac{z(x+a)(y+b)}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
& + \frac{2z(x-a)(y-b)}{[(y-b)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z(x-a)(y-b)}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{2z(x-a)(y+b)}{[(y+b)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} - \frac{z(x-a)(y+b)}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \\
& - \frac{2z(x+a)(y-b)}{[(y-b)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} - \frac{z(x+a)(y-b)}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial g_y}{\partial z} = \frac{1}{4\pi}\Bigg[& \frac{x+a}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} - \frac{2z^2(x+a)}{[(y-b)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} \\
& - \frac{z^2(x+a)}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} - \frac{x-a}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} \\
& + \frac{2z^2(x-a)}{[(y-b)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z^2(x-a)}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{x+a}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} + \frac{2z^2(x+a)}{[(y+b)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} \\
& + \frac{z^2(x+a)}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} + \frac{x-a}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} \\
& - \frac{2z^2(x-a)}{[(y+b)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} - \frac{z^2(x-a)}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial g_z}{\partial x} = \frac{1}{4\pi}\Bigg[& \frac{y+b}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} - \frac{2(x+a)^2(y+b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} \\
& - \frac{(x+a)^2(y+b)}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} - \frac{y-b}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} \\
& + \frac{2(x+a)^2(y-b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} + \frac{(x+a)^2(y-b)}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{y+b}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} + \frac{2(x-a)^2(y+b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} \\
& + \frac{(x-a)^2(y+b)}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} + \frac{y-b}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} \\
& - \frac{2(x-a)^2(y-b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} - \frac{(x-a)^2(y-b)}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& + \frac{y+b}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} - \frac{(x+a)^2(y+b)}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
& - \frac{y+b}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} + \frac{(x-a)^2(y+b)}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \\
& - \frac{y-b}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} + \frac{(x+a)^2(y-b)}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \\
& + \frac{y-b}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} - \frac{(x-a)^2(y-b)}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}}\Bigg]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial g_z}{\partial y} = \frac{1}{4\pi}\Bigg[& \frac{x+a}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} - \frac{(x+a)(y+b)^2}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
& - \frac{x+a}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} + \frac{(x+a)(y-b)^2}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \\
& - \frac{x-a}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} + \frac{(x-a)(y+b)^2}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \\
& + \frac{x-a}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} - \frac{(x-a)(y-b)^2}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
& + \frac{x+a}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{1/2}} - \frac{2(x+a)(y+b)^2}{[(y+b)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} \\
& - \frac{(x+a)(y+b)^2}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} - \frac{x-a}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{1/2}} \\
& + \frac{2(x-a)(y+b)^2}{[(y+b)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} + \frac{(x-a)(y+b)^2}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \\
& - \frac{x+a}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{1/2}} + \frac{2(x+a)(y-b)^2}{[(y-b)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} \\
& + \frac{(x+a)(y-b)^2}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} + \frac{x-a}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{1/2}} \\
& - \frac{2(x-a)(y-b)^2}{[(y-b)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} - \frac{(x-a)(y-b)^2}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}}\Bigg]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial g_z}{\partial z} = \frac{1}{4\pi}\Bigg[&- \frac{2z(x+a)(y+b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} - \frac{z(x+a)(y+b)}{[(x+a)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
&+ \frac{2z(x+a)(y-b)}{[(x+a)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z(x+a)(y-b)}{[(x+a)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \\
&+ \frac{2z(x-a)(y+b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} + \frac{z(x-a)(y+b)}{[(x-a)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \\
&- \frac{2z(x-a)(y-b)}{[(x-a)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} - \frac{z(x-a)(y-b)}{[(x-a)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}} \\
&- \frac{2z(x+a)(y+b)}{[(y+b)^2+z^2]^2[(x+a)^2+(y+b)^2+z^2]^{1/2}} - \frac{z(x+a)(y+b)}{[(y+b)^2+z^2][(x+a)^2+(y+b)^2+z^2]^{3/2}} \\
&+ \frac{2z(x-a)(y+b)}{[(y+b)^2+z^2]^2[(x-a)^2+(y+b)^2+z^2]^{1/2}} + \frac{z(x-a)(y+b)}{[(y+b)^2+z^2][(x-a)^2+(y+b)^2+z^2]^{3/2}} \\
&+ \frac{2z(x+a)(y-b)}{[(y-b)^2+z^2]^2[(x+a)^2+(y-b)^2+z^2]^{1/2}} + \frac{z(x+a)(y-b)}{[(y-b)^2+z^2][(x+a)^2+(y-b)^2+z^2]^{3/2}} \\
&- \frac{2z(x-a)(y-b)}{[(y-b)^2+z^2]^2[(x-a)^2+(y-b)^2+z^2]^{1/2}} - \frac{z(x-a)(y-b)}{[(y-b)^2+z^2][(x-a)^2+(y-b)^2+z^2]^{3/2}}\Bigg]
\end{aligned}
$$

APPENDIX B

Power Circuit Diagram

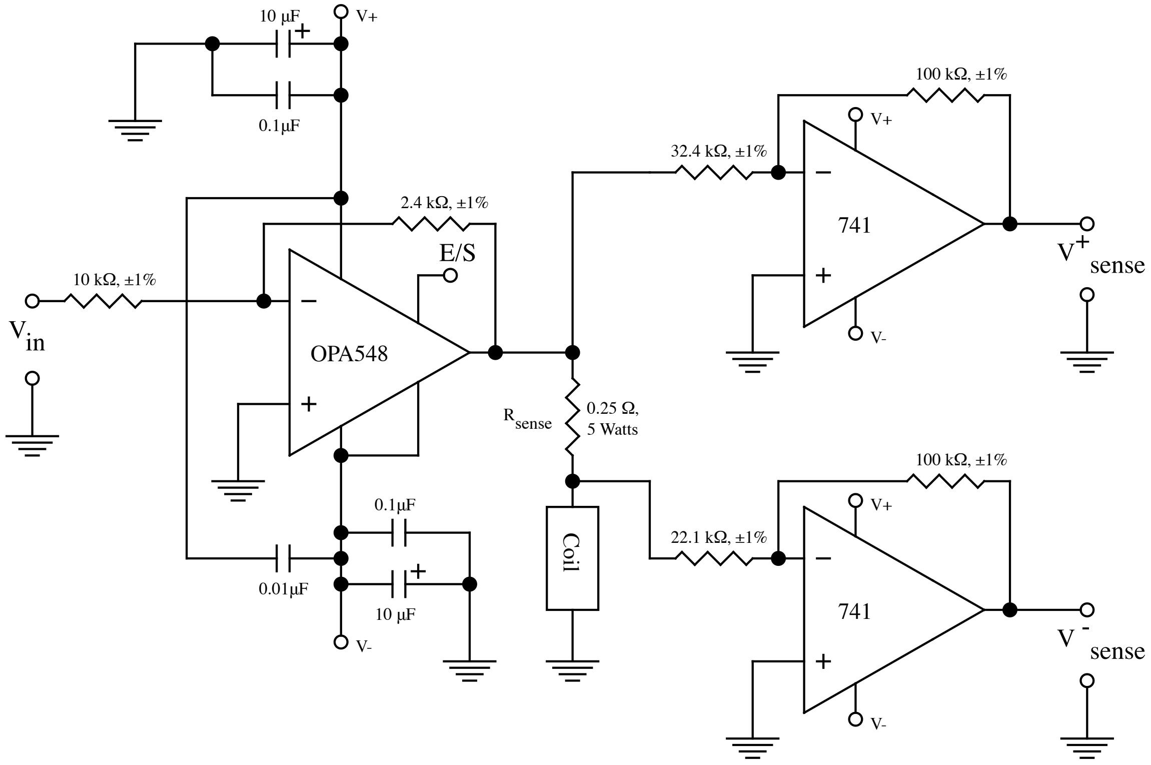

Figure B.1: Circuit diagram for coil power op amps and signal conditional op amps. Note that these are inverting amplifiers and that *each* coil has a power circuit.

APPENDIX C

Closed Loop Matlab/Simulink Simulation Source Code

C.1 Model Parameters Script Code

This script file is needed to establish the workspace variables necessary for the Simulink model files. The functions in this script are available in the subsections of this section.

```
%experiment_simulation_param.m
%m-file to enter the parameters to the simulink model
clear;

r=(0.125/2)*0.0254;              %ball radius in meters
density=7.5*(1/1000)*(100^3);    %ball density in kg/m^3
M=((4*pi*r^3)/3)*density;        %ball mass in kg
mu_0=4*pi*10^(-7);               %permeability of free space in T-m/A
Br=1;                            %remnance in ball in Tesla
nu=0.950;                        %viscosity of fluid
Mag=Br*3/(2*mu_0);               %magnitization of sphere in A/m

% coil dimensions and assembly dimensions
a_0_inch=0.481;
b_0_inch=0.799;
c_0_inch=2.5;                    %minimum coil offset
t=18;                            %number of turns per layer
l=21;                            %number of layers
diam_inch=0.0838;                %12 gauge,assumming heavy insulation
WpFT=1/49.9;                     %weight per foot
RpFT=1.589/1000;                 %12 gauge
spacing_inch=0.25;               %length added on for clearance
```

```
wind=0;                         %wind type wind=0 is perfect, wind=1 is stacked
filter=1;

if wind == 0
    c_inch=b_0_inch+(l-1)*diam_inch*sqrt(3)/2+spacing_inch;
elseif wind == 1
    c_inch=b_0_inch+(l-1)*diam_inch+spacing_inch;
end

% change to SI units
a_0=a_0_inch*0.0254;
b_0=b_0_inch*0.0254;
c_0=c_0_inch*0.0254;
diam=diam_inch*0.0254;
spacing=spacing_inch*0.0254;

%get coil offset
if c_0>=c_inch*0.0254
    c=c_0;
else
    c=c_inch*0.0254;
end

% calculate coil resistance and weight
[Resistance Weight]=resistance(t,l,a_0,b_0,diam,RpFT,WpFT,wind);   %ohms and pounds

%calculate coil self inductance
[L L_c]=self_inductance_mex(t,l,a_0,b_0,diam,wind); %Henrys

% assign build directions to individual coils
gd_1=-1;
```

```
gd_2=-1;
gd_3=1;
gd_4=1;
gd=1;

%calculate mutual inductance
L_m=mutual_inductance_mex(t,l,a_0,b_0,c,diam,gd_1,gd_3,wind);   %Henrys

% homogeneous coordinate transformations
R_1=[1 0 0; 0 -1 0; 0 0 -1];
R_2=[1 0 0; 0 1 0; 0 0 1];
R_3=[1 0 0; 0 0 -1; 0 1 0];
R_4=[1 0 0; 0 0 1; 0 -1 0];
d_1=[0; 0; c];
d_2=-d_1;
d_4=[0; c; 0];
d_3=-d_4;
A_1=[R_1 d_1; 0 0 0 1];
A_2=[R_2 d_2; 0 0 0 1];
A_3=[R_3 d_3; 0 0 0 1];
A_4=[R_4 d_4; 0 0 0 1];
A_1_inv=inv(A_1);
A_2_inv=inv(A_2);
A_3_inv=inv(A_3);
A_4_inv=inv(A_4);

R_mtrx=(Resistance+0.42)*eye(4);    %add in sense resistor value to coil resistance

M_mtrx=[L_c -L_m L_m L_m; -L_m L_c L_m L_m; L_m L_m L_c -L_m; L_m L_m -L_m L_c];

%Define spiral trajectory parameters
rho=0.5*0.0254;            %radius of circular path
theta_dot=2*pi/10;         %angular velocity
```

```
C_tau=10;

density_glyc=1263;          %density of glycerine kg/m^3

Reynolds=r*rho*theta_dot*density_glyc/nu;

%Define square trajectory parameters

T_mult=10;

T=[0 1 2 3 4 5 6]*T_mult;

Y_square=[0 0.5 -0.5 -0.5 0.5 0.5 0];

Z_square=[0 0.5 0.5 -0.5 -0.5 0.5 0];


%define field grid for interpolation

Y_inch=-1:0.05:1;

Z_inch=-1:0.05:1;


%convert to SI

X=0;

Y=Y_inch*0.0254;

Z=Z_inch*0.0254;


%generate field components

for m=1:length(Z)

    for n=1:length(Y)

        %convert to local coordinate

        s_1=inv(A_1)*[X; Y(n); Z(m); 1];

        s_2=inv(A_2)*[X; Y(n); Z(m); 1];

        s_3=inv(A_3)*[X; Y(n); Z(m); 1];

        s_4=inv(A_4)*[X; Y(n); Z(m); 1];

        % calculate field effects

        [Gx1_int(m,n) Gy1_int(m,n) Gz1_int(m,n) dGx_dx1_int(m,n) dGx_dy1_int(m,n)

        dGx_dz1_int(m,n) dGy_dx1_int(m,n) dGy_dy1_int(m,n) dGy_dz1_int(m,n)

        dGz_dx1_int(m,n) dGz_dy1_int(m,n) dGz_dz1_int(m,n)]=

        coil_G_winding_mex(s_1(1),s_1(2),s_1(3),t,l,gd_1,a_0,b_0,diam,wind);

        [Gx2_int(m,n) Gy2_int(m,n) Gz2_int(m,n) dGx_dx2_int(m,n) dGx_dy2_int(m,n)

        dGx_dz2_int(m,n) dGy_dx2_int(m,n) dGy_dy2_int(m,n) dGy_dz2_int(m,n)
        dGz_dx2_int(m,n) dGz_dy2_int(m,n) dGz_dz2_int(m,n)]=
        coil_G_winding_mex(s_2(1),s_2(2),s_2(3),t,l,gd_2,a_0,b_0,diam,wind);
        [Gx3_int(m,n) Gy3_int(m,n) Gz3_int(m,n) dGx_dx3_int(m,n) dGx_dy3_int(m,n)
        dGx_dz3_int(m,n) dGy_dx3_int(m,n) dGy_dy3_int(m,n) dGy_dz3_int(m,n)
        dGz_dx3_int(m,n) dGz_dy3_int(m,n) dGz_dz3_int(m,n)]=
        coil_G_winding_mex(s_3(1),s_3(2),s_3(3),t,l,gd_3,a_0,b_0,diam,wind);
        [Gx4_int(m,n) Gy4_int(m,n) Gz4_int(m,n) dGx_dx4_int(m,n) dGx_dy4_int(m,n)
        dGx_dz4_int(m,n) dGy_dx4_int(m,n) dGy_dy4_int(m,n) dGy_dz4_int(m,n)
        dGz_dx4_int(m,n) dGz_dy4_int(m,n) dGz_dz4_int(m,n)]=
        coil_G_winding_mex(s_4(1),s_4(2),s_4(3),t,l,gd_4,a_0,b_0,diam,wind);
        count=[m n];
        count
    end
end
save experiment_simulation_param.mat
```

C.1.1 Coil Resistance and Weight Calculation Script

```
function [Resistance, Weight]=resistance(t,l,a_0,b_0,d,RpFT,WpFT,wind)
% function to calculate the resistance of a coil in a coil
%determine if using a perfect wind or a stacked wind, wind=0 is perfect,
%wind=0 is stacked


if wind == 0
    L_total=4*t*l*(a_0+b_0+d*(l-1)*sqrt(3)/2);
    L_total_feet=L_total/0.3048;
elseif wind == 1
    % calculate coil wire length in meters and convert to feet
    L_total=4*t*l*(a_0+b_0+d*(l-1));
    L_total_feet=L_total/0.3048;
end
Resistance=L_total_feet*RpFT;
Weight=L_total_feet*WpFT;
```

C.1.2 Coil Self Inductance Calculation MEX Script

```
/* c-code to calculate the self inductance for a solid coil */


#include "mex.h"

#include "math.h"

#include "matrix.h"


#define pi 3.14159265359

#define mu_0 4*pi*1e-7

#define gd 1


/* do calculations */

void self_inductance_mex(double *L, double *L_c, int t, int l, double a_0,

double b_0, double d, int wind)


{

int lambda, tau, m, n, p, q;

double a_lambda, b_lambda, a_l, b_l, c_tau, c_t;

double a_y_1_1, a_y_1_2, a_y_1_3, a_y_1_4, a_y_2_1, a_y_2_2, a_y_2_3, a_y_2_4;

double a_x_1_1, a_x_1_2, a_x_1_3, a_x_1_4, a_x_2_1, a_x_2_2, a_x_2_3, a_x_2_4;

double a_tot, A_tot = 0, A_y_1, A_y_2, A_x_1, A_x_2;


    for(p=1;p<=t;p++){

        for(q=1;q<=l;q++){

            for(m=1;m<=t;m++){

                for(n=1;n<=l;n++){

                    if (wind == 0){

                        a_lambda=a_0+(q-1)*d*sqrt(3)/2;

                        b_lambda=b_0+(q-1)*d*sqrt(3)/2;

                        a_l=a_0+(n-1)*d*sqrt(3)/2;

                        b_l=b_0+(n-1)*d*sqrt(3)/2;

                        if (floor(q/2.0) == ceil(q/2.0)){
```

```
        c_tau=gd*((d/2)+(p-1)*d);
    }
    else{
        c_tau=gd*(p-1)*d;
    }
    if (floor(n/2.0) == ceil(n/2.0)){
        c_t=gd*((d/2)+(m-1)*d);
    }
    else{
        c_t=gd*(m-1)*d;
    }
}
else if (wind == 1){
    a_lambda=a_0+(q-1)*d;
    b_lambda=b_0+(q-1)*d;
    a_l=a_0+(n-1)*d;
    b_l=b_0+(n-1)*d;
    c_tau=gd*(p-1)*d;
    c_t=gd*(m-1)*d;
}

if (p == m && q == n){
    a_y_1_1=0.0;
    a_y_1_2=0.0;
    a_y_1_3=0.0;
    a_y_1_4=0.0;
    a_y_2_1=0.0;
    a_y_2_2=0.0;
    a_y_2_3=0.0;
    a_y_2_4=0.0;
    a_x_1_1=0.0;
    a_x_1_2=0.0;
    a_x_1_3=0.0;
    a_x_1_4=0.0;
    a_x_2_1=0.0;
    a_x_2_2=0.0;
    a_x_2_3=0.0;
    a_x_2_4=0.0;
}
else{
    a_y_1_1=(b_lambda+b_l)*log(fabs((b_lambda+b_l)+
    sqrt(pow((a_lambda-a_l),2.0)+pow((b_lambda+b_l),2.0)+
    pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda-a_l),2.0)+
    pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
    (-b_lambda+b_l)*log(fabs((-b_lambda+b_l)+
    sqrt(pow((a_lambda-a_l),2.0)+pow((-b_lambda+b_l),2.0)+
    pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda-a_l),2.0)+
    pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));
    a_y_1_2=(b_lambda-b_l)*log(fabs((b_lambda-b_l)+
    sqrt(pow((a_lambda-a_l),2.0)+pow((b_lambda-b_l),2.0)+
    pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda-a_l),2.0)+
    pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
    (-b_lambda-b_l)*log(fabs((-b_lambda-b_l)+
    sqrt(pow((a_lambda-a_l),2.0)+pow((-b_lambda-b_l),2.0)+
    pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda-a_l),2.0)+
    pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
    a_y_1_3=(b_lambda-b_l)*log(fabs((b_lambda-b_l)+
    sqrt(pow((a_lambda+a_l),2.0)+pow((b_lambda-b_l),2.0)+
    pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda+a_l),2.0)+
    pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
    (-b_lambda-b_l)*log(fabs((-b_lambda-b_l)+
    sqrt(pow((a_lambda+a_l),2.0)+pow((-b_lambda-b_l),2.0)+
    pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda+a_l),2.0)+
    pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
    a_y_1_4=(b_lambda+b_l)*log(fabs((b_lambda+b_l)+
    sqrt(pow((a_lambda+a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda+a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(-b_lambda+b_l)*log(fabs((-b_lambda+b_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((-b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda+a_l),2.0)+
pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));

a_y_2_1=(-b_lambda+b_l)*log(fabs((-b_lambda+b_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((-b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda-a_l),2.0)+
pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(b_lambda+b_l)*log(fabs((b_lambda+b_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda-a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));
a_y_2_2=(-b_lambda-b_l)*log(fabs((-b_lambda-b_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda-a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(b_lambda-b_l)*log(fabs((b_lambda-b_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda-a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_y_2_3=(-b_lambda-b_l)*log(fabs((-b_lambda-b_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda+a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(b_lambda-b_l)*log(fabs((b_lambda-b_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda+a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_y_2_4=(-b_lambda+b_l)*log(fabs((-b_lambda+b_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((-b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda+a_l),2.0)+
pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(b_lambda+b_l)*log(fabs((b_lambda+b_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda+a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));

a_x_1_1=(-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda-a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
sqrt(pow((a_lambda-a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda-a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_1_2=(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda+a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda+a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_1_3=(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda+a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda+a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_1_4=(-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda-a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
sqrt(pow((a_lambda-a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda-a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));

a_x_2_1=(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
sqrt(pow((a_lambda-a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda-a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda-a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_2_2=(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda+a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda+a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_2_3=(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((-b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda+a_l),2.0)+
pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((-b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda+a_l),2.0)+
pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_2_4=(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
sqrt(pow((a_lambda-a_l),2.0)+pow((-b_lambda+b_l),2.0)+
                        pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda-a_l),2.0)+
                        pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
                        (-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
                        sqrt(pow((-a_lambda-a_l),2.0)+pow((-b_lambda+b_l),2.0)+
                        pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda-a_l),2.0)+
                        pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));

                        A_y_1=(a_y_1_1-a_y_1_2+a_y_1_3-a_y_1_4);
                        A_y_2=(a_y_2_1-a_y_2_2+a_y_2_3-a_y_2_4);
                        A_x_1=(a_x_1_1-a_x_1_2+a_x_1_3-a_x_1_4);
                        A_x_2=(a_x_2_1-a_x_2_2+a_x_2_3-a_x_2_4);



                        a_tot=A_y_1+A_y_2+A_x_1+A_x_2;
                        A_tot=A_tot+a_tot;
                    }
                }
            }
        }
    }

*L=(mu_0*A_tot/(4*pi));
*L_c=(*L)*(1+1.0/(t*l));

}


/* The gateway routine */
void mexFunction(int nlhs, mxArray *plhs[],
                 int nrhs, const mxArray *prhs[])
{
    double *L, *L_c;
    int t, l, wind;
    double a_0, b_0, d;
```

```
/* Create a 1-by-1 matrices for the return arguments. */
plhs[0] = mxCreateDoubleMatrix(1, 1, mxREAL);
plhs[1] = mxCreateDoubleMatrix(1, 1, mxREAL);

/* get the scalar inputs */
t = mxGetScalar(prhs[0]);
l = mxGetScalar(prhs[1]);
a_0 = mxGetScalar(prhs[2]);
b_0 = mxGetScalar(prhs[3]);
d = mxGetScalar(prhs[4]);
wind = mxGetScalar(prhs[5]);

/* assign a pointer to the output */
L = mxGetPr(plhs[0]);
L_c = mxGetPr(plhs[1]);

/* Call the self_inductance_mex subroutine. */
self_inductance_mex(L,L_c,t,l,a_0,b_0,d,wind);
}
```

C.1.3 Coil Mutual Inductance MEX Script

```
/* c-code to calculate the mutual inductance for two coils oriented adjacently */

#include "mex.h"
#include "math.h"
#include "matrix.h"

#define pi 3.14159265359
#define mu_0 4*pi*1e-7

/* do calculations */
void mutual_inductance_mex(double *L_31, int t, int l, double a_0, double b_0,
double c, double d, int gd_1, int gd_3, int wind)

{
int m, n, p, q;
double a_lambda, b_lambda, a_l, b_l, c_tau, c_t, c_tau_plus, c_tau_minus;
double a_x_1_1, a_x_1_2, a_x_1_3, a_x_1_4, a_x_2_1, a_x_2_2, a_x_2_3, a_x_2_4;
double a_tot, A_tot = 0, A_x_1, A_x_2;


    for(p=1;p<=t;p++){
        for(q=1;q<=l;q++){
            for(m=1;m<=t;m++){
                for(n=1;n<=l;n++){
                    if (wind == 0){
                        a_l=a_0+(n-1)*d*sqrt(3)/2;
                        b_l=b_0+(n-1)*d*sqrt(3)/2;
                        a_lambda=a_0+(q-1)*d*sqrt(3)/2;
                        c_tau_plus=c+b_0+(q-1)*d*sqrt(3)/2;
                        c_tau_minus=c-b_0-(q-1)*d*sqrt(3)/2;
                        if (floor(n/2.0) == ceil(n/2.0)){
                            c_t=gd_1*((d/2)+(m-1)*d);
                        }
                        else{
                            c_t=gd_1*(m-1)*d;
                        }
                        if (floor(q/2.0) == ceil (q/2.0)){
                            b_lambda=c+gd_3*((d/2)+(p-1)*d);
                        }
                        else{
                            b_lambda=c+gd_3*(p-1)*d;
                        }
                    }
                    else if (wind == 1){
```

```
    a_lambda=a_0+(q-1)*d;
    b_lambda=c+gd_3*(p-1)*d;
    a_l=a_0+(n-1)*d;
    b_l=b_0+(n-1)*d;
    c_tau_plus=c+b_0+(q-1)*d;
    c_tau_minus=c-b_0-(q-1)*d;
    c_t=gd_1*(m-1)*d;
}

c_tau=c_tau_minus;
a_x_1_1=(-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda-a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
sqrt(pow((a_lambda-a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda-a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_1_2=(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda+a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda+a_l),2.0)+
pow((b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_1_3=(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda+a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda+a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_1_4=(-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((-a_lambda-a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
sqrt(pow((a_lambda-a_l),2.0)+pow((b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((a_lambda-a_l),2.0)+
pow((b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));

c_tau=c_tau_plus;
a_x_2_1=(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
sqrt(pow((a_lambda-a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda-a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
sqrt(pow((-a_lambda-a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda-a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_2_2=(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda+a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0))-
(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((-b_lambda-b_l),2.0)+
pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda+a_l),2.0)+
pow((-b_lambda-b_l),2.0)+pow((c_tau-c_t),2.0));
a_x_2_3=(a_lambda+a_l)*log(fabs((a_lambda+a_l)+
sqrt(pow((a_lambda+a_l),2.0)+pow((-b_lambda+b_l),2.0)+
pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda+a_l),2.0)+
pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
(-a_lambda+a_l)*log(fabs((-a_lambda+a_l)+
sqrt(pow((-a_lambda+a_l),2.0)+pow((-b_lambda+b_l),2.0)+
                    pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda+a_l),2.0)+
                    pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));
                    a_x_2_4=(a_lambda-a_l)*log(fabs((a_lambda-a_l)+
                    sqrt(pow((a_lambda-a_l),2.0)+pow((-b_lambda+b_l),2.0)+
                    pow((c_tau-c_t),2.0))))-sqrt(pow((a_lambda-a_l),2.0)+
                    pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0))-
                    (-a_lambda-a_l)*log(fabs((-a_lambda-a_l)+
                    sqrt(pow((-a_lambda-a_l),2.0)+pow((-b_lambda+b_l),2.0)+
                    pow((c_tau-c_t),2.0))))+sqrt(pow((-a_lambda-a_l),2.0)+
                    pow((-b_lambda+b_l),2.0)+pow((c_tau-c_t),2.0));

                    A_x_1=(a_x_1_1-a_x_1_2+a_x_1_3-a_x_1_4);
                    A_x_2=(a_x_2_1-a_x_2_2+a_x_2_3-a_x_2_4);

                    a_tot=A_x_1+A_x_2;
                    A_tot=A_tot+a_tot;
                }
            }
        }
    }
*L_31=(mu_0*A_tot/(4*pi));
}

/* The gateway routine */
void mexFunction(int nlhs, mxArray *plhs[],
                 int nrhs, const mxArray *prhs[])
{
    double *L_31;
    int t, l, gd_1, gd_3, wind;
    double a_0, b_0, c, d;

/* Create a 1-by-1 matrices for the return argument. */
plhs[0] = mxCreateDoubleMatrix(1, 1, mxREAL);
```

```
/* get the scalar inputs */
t = mxGetScalar(prhs[0]);
l = mxGetScalar(prhs[1]);
a_0 = mxGetScalar(prhs[2]);
b_0 = mxGetScalar(prhs[3]);
c = mxGetScalar(prhs[4]);
d = mxGetScalar(prhs[5]);
gd_1 = mxGetScalar(prhs[6]);
gd_3 = mxGetScalar(prhs[7]);
wind = mxGetScalar(prhs[8]);

/* assign a pointer to the output */
L_31 = mxGetPr(plhs[0]);

/* Call the mutual_inductance_mex subroutine. */
mutual_inductance_mex(L_31,t,l,a_0,b_0,c,d,gd_1,gd_3,wind);
}
```

C.1.4 G Field Calculation MEX Script

```
/* c-code to calculate the g-field for a wound coil */

#include "mex.h"
#include "math.h"
#include "matrix.h"

#define pi 3.14159265359

/* do calculations */
void coil_G_mex(double *Gx_c, double *Gy_c, double *Gz_c, double *dGx_dx_c,
       double *dGx_dy_c, double *dGx_dz_c, double *dGy_dx_c,
                double *dGy_dy_c, double *dGy_dz_c, double *dGz_dx_c,
                double *dGz_dy_c, double *dGz_dz_c, double x, double y,
                double z, int t, int l, int gd, double a_0, double b_0,
                double diam, int wind)
{
int lambda, tau;
double a, b, z_g;
double Gx_w, Gy_w, Gz_w, dGx_dx_w, dGx_dy_w, dGx_dz_w, dGy_dx_w, dGy_dy_w,
dGy_dz_w, dGz_dx_w, dGz_dy_w, dGz_dz_w;
double I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8;

double delGz_delx1, delGz_delx2, delGz_delx3, delGz_delx4, delGz_delx5, delGz_delx6,
delGz_delx7, delGz_delx8;
double delGz_dely1, delGz_dely2, delGz_dely3, delGz_dely4, delGz_dely5, delGz_dely6,
delGz_dely7, delGz_dely8, delGz_dely9, delGz_dely10;
double delGz_dely11, delGz_dely12, delGz_dely13, delGz_dely14, delGz_dely15,
delGz_dely16, delGz_dely17, delGz_dely18, delGz_dely19, delGz_dely20;
double delGz_delz1, delGz_delz2, delGz_delz3, delGz_delz4;

double delGx_delx1, delGx_delx2, delGx_delx3, delGx_delx4, delGx_delx5, delGx_delx6,
delGx_delx7, delGx_delx8;
double delGx_dely1, delGx_dely2, delGx_dely3, delGx_dely4, delGx_dely5, delGx_dely6,
delGx_dely7, delGx_dely8;
double delGx_delz1, delGx_delz2, delGx_delz3, delGx_delz4, delGx_delz5, delGx_delz6,
delGx_delz7, delGx_delz8, delGx_delz9;
double delGx_delz10, delGx_delz11, delGx_delz12;

double delGy_delx1, delGy_delx2, delGy_delx3, delGy_delx4, delGy_delx5, delGy_delx6,
delGy_delx7, delGy_delx8;
double delGy_dely1, delGy_dely2, delGy_dely3, delGy_dely4, delGy_dely5, delGy_dely6,
delGy_dely7, delGy_dely8;
double delGy_delz1, delGy_delz2, delGy_delz3, delGy_delz4, delGy_delz5, delGy_delz6,
delGy_delz7, delGy_delz8, delGy_delz9;
double delGy_delz10, delGy_delz11, delGy_delz12;
```

```
for (lambda=1;lambda<=l;lambda++){
    for (tau=1;tau<=t;tau++){
        if (wind == 1){
            z_g=z-gd*(tau-1)*diam;
            a=a_0+(lambda-1)*diam;
            b=b_0+(lambda-1)*diam;
        }
        else if (wind ==0){
            if (floor(lambda/2.0) == ceil(lambda/2.0)){
                z_g=z-gd*((diam/2)+(tau-1)*diam);
            }
            else{
                z_g=z-gd*(tau-1)*diam;
            }
            a=a_0+(lambda-1)*diam*sqrt(3)/2;
            b=b_0+(lambda-1)*diam*sqrt(3)/2;
        }

        /* other function here */
        I_1=((x+a)/(pow(x+a,2.0)+pow(z_g,2.0)))*((y+b)/(sqrt(pow(x+a,2.0)+
        pow(y+b,2.0)+pow(z_g,2.0)))-(y-b)/(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+
        pow(z_g,2.0))));
        I_2=-((x-a)/(pow(x-a,2.0)+pow(z_g,2.0)))*((y+b)/(sqrt(pow(x-a,2.0)+
        pow(y+b,2.0)+pow(z_g,2.0)))-(y-b)/(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+
        pow(z_g,2.0))));
        I_3=((y+b)/(pow(y+b,2.0)+pow(z_g,2.0)))*((x+a)/(sqrt(pow(x+a,2.0)+
        pow(y+b,2.0)+pow(z_g,2.0)))-(x-a)/(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+
        pow(z_g,2.0))));
        I_4=-((y-b)/(pow(y-b,2.0)+pow(z_g,2.0)))*((x+a)/(sqrt(pow(x+a,2.0)+
        pow(y-b,2.0)+pow(z_g,2.0)))-(x-a)/(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+
        pow(z_g,2.0))));
        Gz_w=(1.0/(4*pi))*(I_1+I_2+I_3+I_4);
```

```
delGz_delx1=(y-b)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))))-(2*(pow(x-a,2.0))*(y-b))/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))))-
((pow(x-a,2.0))*(y-b))/((pow(x-a,2.0)+pow(z_g,2.0))*(pow(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_delx2=-(y+b)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))))+(2*(pow(x-a,2.0))*(y+b))/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))))+
((pow(x-a,2.0))*(y+b))/((pow(x-a,2.0)+pow(z_g,2.0))*(pow(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_delx3=(y-b)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))))-((pow(x-a,2.0))*(y-b))/((pow(y-b,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_delx4=-(y-b)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))))+((pow(x+a,2.0))*(y-b))/((pow(y-b,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_delx5=(y+b)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))))-(2*(pow(x+a,2.0))*(y+b))/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))))-
((pow(x+a,2.0))*(y+b))/((pow(x+a,2.0)+pow(z_g,2.0))*(pow(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_delx6=-(y-b)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))))+(2*(pow(x+a,2.0))*(y-b))/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))))+
((pow(x+a,2.0))*(y-b))/((pow(x+a,2.0)+pow(z_g,2.0))*(pow(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_delx7=(y+b)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))))-((pow(x+a,2.0))*(y+b))/((pow(y+b,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_delx8=-(y+b)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))))+((pow(x-a,2.0))*(y+b))/((pow(y+b,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
dGz_dx_w=(1.0/(4*pi))*(delGz_delx1+delGz_delx2+delGz_delx3+
delGz_delx4+delGz_delx5+delGz_delx6+delGz_delx7+delGz_delx8);


delGz_dely1=(x+a)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGz_dely2=((x+a)*pow(y+b,2.0))/((pow(x+a,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_dely3=(x+a)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGz_dely4=((x+a)*pow(y-b,2.0))/((pow(x+a,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2),1.5)));
delGz_dely5=(x-a)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGz_dely6=((x-a)*pow(y+b,2.0))/((pow(x-a,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_dely7=(x-a)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGz_dely8=((x-a)*pow(y-b,2.0))/((pow(x-a,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_dely9=(x+a)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGz_dely10=(2*(x+a)*pow(y+b,2.0))/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGz_dely11=((x+a)*pow(y+b,2.0))/(((pow(y+b,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_dely12=(x-a)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGz_dely13=(2*(x-a)*pow(y+b,2.0))/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGz_dely14=((x-a)*pow(y+b,2.0))/(((pow(y+b,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_dely15=(x+a)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGz_dely16=(2*(x+a)*pow(y-b,2.0))/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGz_dely17=((x+a)*pow(y-b,2.0))/(((pow(y-b,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_dely18=(x-a)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2))));
delGz_dely19=(2*(x-a)*pow(y-b,2.0))/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGz_dely20=((x-a)*pow(y-b,2.0))/(((pow(y-b,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
dGz_dy_w=(1.0/(4*pi))*(delGz_dely1-delGz_dely2-delGz_dely3+
delGz_dely4-delGz_dely5+delGz_dely6+delGz_dely7-delGz_dely8+
delGz_dely9-delGz_dely10-delGz_dely11-delGz_dely12+
delGz_dely13+delGz_dely14-delGz_dely15+delGz_dely16+
delGz_dely17+delGz_dely18-delGz_dely19-delGz_dely20);

delGz_delz1=-((x-a)*(y-b)*2*z_g)/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))))-
((x-a)*(y-b)*z_g)/((pow(x-a,2.0)+pow(z_g,2.0))*(pow(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0),1.5)))+((x-a)*(y+b)*2*z_g)/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))))+
((x-a)*(y+b)*z_g)/((pow(x-a,2.0)+pow(z_g,2.0))*(pow(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGz_delz2=-((x-a)*(y-b)*2*z_g)/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))))-
((x-a)*(y-b)*z_g)/((pow(y-b,2.0)+pow(z_g,2.0))*(pow(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0),1.5)))+((x+a)*(y-b)*2*z_g)/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))))+
((x+a)*(y-b)*z_g)/((pow(y-b,2.0)+pow(z_g,2.0))*(pow(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_delz3=-((x+a)*(y+b)*2*z_g)/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))))-
((x+a)*(y+b)*z_g)/((pow(x+a,2.0)+pow(z_g,2.0))*(pow(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0),1.5)))+((x+a)*(y-b)*2*z_g)/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))))+
((x+a)*(y-b)*z_g)/((pow(x+a,2.0)+pow(z_g,2.0))*(pow(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGz_delz4=-((x+a)*(y+b)*2*z_g)/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))))-
((x+a)*(y+b)*z_g)/((pow(y+b,2.0)+pow(z_g,2.0))*(pow(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0),1.5)))+((x-a)*(y+b)*2*z_g)/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))))+
((x-a)*(y+b)*z_g)/((pow(y+b,2.0)+pow(z_g,2.0))*(pow(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0),1.5)));
dGz_dz_w=(1.0/(4*pi))*(delGz_delz1+delGz_delz2+delGz_delz3+delGz_delz4);

I_5=((1)/(pow(x-a,2.0)+pow(z_g,2.0)))*((y+b)/sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))-(y-b)/sqrt(pow(x-a,2.0)+pow(y-b,2.0)+
pow(z_g,2.0)));
I_6=((1)/(pow(x+a,2.0)+pow(z_g,2.0)))*((y+b)/sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))-(y-b)/sqrt(pow(x+a,2.0)+pow(y-b,2.0)+
pow(z_g,2.0)));
Gx_w=((z_g)/(4*pi))*(I_5-I_6);

delGx_delx1=(2*z_g*(x+a)*(y+b))/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGx_delx2=(z_g*(x+a)*(y+b))/(((pow(x+a,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGx_delx3=(2*z_g*(x-a)*(y-b))/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGx_delx4=(z_g*(x-a)*(y-b))/(((pow(x-a,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGx_delx5=(2*z_g*(x+a)*(y-b))/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGx_delx6=(z_g*(x+a)*(y-b))/(((pow(x+a,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGx_delx7=(2*z_g*(x-a)*(y+b))/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGx_delx8=(z_g*(x-a)*(y+b))/(((pow(x-a,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
dGx_dx_w=(1.0/(4*pi))*(delGx_delx1+delGx_delx2+delGx_delx3+
delGx_delx4-delGx_delx5-delGx_delx6-delGx_delx7-delGx_delx8);

delGx_dely1=(z_g)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGx_dely2=(z_g*pow(y+b,2.0))/((pow(x-a,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGx_dely3=(z_g)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGx_dely4=(z_g*pow(y-b,2.0))/((pow(x-a,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGx_dely5=(z_g)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGx_dely6=(z_g*pow(y+b,2.0))/((pow(x+a,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGx_dely7=(z_g)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGx_dely8=(z_g*pow(y-b,2.0))/((pow(x+a,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
dGx_dy_w=(1.0/(4*pi))*(delGx_dely1-delGx_dely2-delGx_dely3+
delGx_dely4-delGx_dely5+delGx_dely6+delGx_dely7-delGx_dely8);

delGx_delz1=(y+b)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGx_delz2=(2*pow(z_g,2.0)*(y+b))/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGx_delz3=(pow(z_g,2.0)*(y+b))/(((pow(x-a,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGx_delz4=(y-b)/((pow(x-a,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
```

```
delGx_delz5=(2*pow(z_g,2.0)*(y-b))/((pow(pow(x-a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGx_delz6=(pow(z_g,2.0)*(y-b))/(((pow(x-a,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGx_delz7=(y+b)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGx_delz8=(2*pow(z_g,2.0)*(y+b))/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGx_delz9=(pow(z_g,2.0)*(y+b))/(((pow(x+a,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGx_delz10=(y-b)/((pow(x+a,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGx_delz11=(2*pow(z_g,2.0)*(y-b))/((pow(pow(x+a,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGx_delz12=(pow(z_g,2.0)*(y-b))/(((pow(x+a,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
dGx_dz_w=(1.0/(4*pi))*(delGx_delz1-delGx_delz2-delGx_delz3-
delGx_delz4+delGx_delz5+delGx_delz6-delGx_delz7+delGx_delz8+
delGx_delz9+delGx_delz10-delGx_delz11-delGx_delz12);


I_7=((1)/(pow(y-b,2.0)+pow(z_g,2.0)))*((x+a)/sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))-(x-a)/sqrt(pow(x-a,2.0)+pow(y-b,2.0)+
pow(z_g,2.0)));
I_8=((1)/(pow(y+b,2.0)+pow(z_g,2.0)))*((x+a)/sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))-(x-a)/sqrt(pow(x-a,2.0)+pow(y+b,2.0)+
pow(z_g,2.0)));
Gy_w=((z_g)/(4*pi))*(I_7-I_8);


delGy_delx1=(z_g)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGy_delx2=(z_g*pow(x+a,2.0))/((pow(y-b,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGy_delx3=(z_g)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGy_delx4=(z_g*pow(x-a,2.0))/((pow(y-b,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGy_delx5=(z_g)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGy_delx6=(z_g*pow(x+a,2.0))/((pow(y+b,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGy_delx7=(z_g)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGy_delx8=(z_g*pow(x-a,2.0))/((pow(y+b,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
dGy_dx_w=(1.0/(4*pi))*(delGy_delx1-delGy_delx2-delGy_delx3+
delGy_delx4-delGy_delx5+delGy_delx6+delGy_delx7-delGy_delx8);

delGy_dely1=(2*z_g*(x+a)*(y+b))/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGy_dely2=(z_g*(x+a)*(y+b))/(((pow(y+b,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGy_dely3=(2*z_g*(x-a)*(y-b))/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGy_dely4=(z_g*(x-a)*(y-b))/(((pow(y-b,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGy_dely5=(2*z_g*(x-a)*(y+b))/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGy_dely6=(z_g*(x-a)*(y+b))/(((pow(y+b,2.0)+
pow(z_g,2.0)))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGy_dely7=(2*z_g*(x+a)*(y-b))/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGy_dely8=(z_g*(x+a)*(y-b))/(((pow(y-b,2.0)+
pow(z_g,2.0)))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
dGy_dy_w=(1.0/(4*pi))*(delGy_dely1+delGy_dely2+delGy_dely3+
delGy_dely4-delGy_dely5-delGy_dely6-delGy_dely7-delGy_dely8);
```

```
delGy_delz1=(x+a)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGy_delz2=(2*pow(z_g,2.0)*(x+a))/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGy_delz3=(pow(z_g,2.0)*(x+a))/((pow(y-b,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGy_delz4=(x-a)/((pow(y-b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y-b,2.0)+pow(z_g,2.0))));
delGy_delz5=(2*pow(z_g,2.0)*(x-a))/((pow(pow(y-b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0))));
delGy_delz6=(pow(z_g,2.0)*(x-a))/((pow(y-b,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y-b,2.0)+pow(z_g,2.0),1.5)));
delGy_delz7=(x+a)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x+a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGy_delz8=(2*pow(z_g,2.0)*(x+a))/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGy_delz9=(pow(z_g,2.0)*(x+a))/((pow(y+b,2.0)+
pow(z_g,2.0))*(pow(pow(x+a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
delGy_delz10=(x-a)/((pow(y+b,2.0)+pow(z_g,2.0))*(sqrt(pow(x-a,2.0)+
pow(y+b,2.0)+pow(z_g,2.0))));
delGy_delz11=(2*pow(z_g,2.0)*(x-a))/((pow(pow(y+b,2.0)+
pow(z_g,2.0),2.0))*(sqrt(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0))));
delGy_delz12=(pow(z_g,2.0)*(x-a))/((pow(y+b,2.0)+
pow(z_g,2.0))*(pow(pow(x-a,2.0)+pow(y+b,2.0)+pow(z_g,2.0),1.5)));
dGy_dz_w=(1.0/(4*pi))*(delGy_delz1-delGy_delz2-delGy_delz3-
delGy_delz4+delGy_delz5+delGy_delz6-delGy_delz7+delGy_delz8+
delGy_delz9+delGy_delz10-delGy_delz11-delGy_delz12);

/* other function here */

*Gx_c=*Gx_c+Gx_w;
*Gy_c=*Gy_c+Gy_w;
*Gz_c=*Gz_c+Gz_w;
```

```
        *dGx_dx_c=*dGx_dx_c+dGx_dx_w;

        *dGx_dy_c=*dGx_dy_c+dGx_dy_w;

        *dGx_dz_c=*dGx_dz_c+dGx_dz_w;

        *dGy_dx_c=*dGy_dx_c+dGy_dx_w;

        *dGy_dy_c=*dGy_dy_c+dGy_dy_w;

        *dGy_dz_c=*dGy_dz_c+dGy_dz_w;

        *dGz_dx_c=*dGz_dx_c+dGz_dx_w;

        *dGz_dy_c=*dGz_dy_c+dGz_dy_w;

        *dGz_dz_c=*dGz_dz_c+dGz_dz_w;

        }

    }

}


/* The gateway routine */

void mexFunction(int nlhs, mxArray *plhs[],

                 int nrhs, const mxArray *prhs[])

{

double *Gx_c, *Gy_c, *Gz_c, *dGx_dx_c, *dGx_dy_c, *dGx_dz_c, *dGy_dx_c,

*dGy_dy_c, *dGy_dz_c, *dGz_dx_c, *dGz_dy_c, *dGz_dz_c;

int t, l, gd, wind;

double x, y, z, a_0, b_0, diam;


/* Create a 1-by-1 matrices for the return arguments. */

plhs[0] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[1] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[2] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[3] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[4] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[5] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[6] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[7] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[8] = mxCreateDoubleMatrix(1, 1, mxREAL);

plhs[9] = mxCreateDoubleMatrix(1, 1, mxREAL);
```

```
plhs[10] = mxCreateDoubleMatrix(1, 1, mxREAL);
plhs[11] = mxCreateDoubleMatrix(1, 1, mxREAL);

/* get the scalar inputs */
x = mxGetScalar(prhs[0]);
y = mxGetScalar(prhs[1]);
z = mxGetScalar(prhs[2]);
t = mxGetScalar(prhs[3]);
l = mxGetScalar(prhs[4]);
gd = mxGetScalar(prhs[5]);
a_0 = mxGetScalar(prhs[6]);
b_0 = mxGetScalar(prhs[7]);
diam = mxGetScalar(prhs[8]);
wind = mxGetScalar(prhs[9]);

/* assign a pointer to the output */
Gx_c = mxGetPr(plhs[0]);
Gy_c = mxGetPr(plhs[1]);
Gz_c = mxGetPr(plhs[2]);
dGx_dx_c = mxGetPr(plhs[3]);
dGx_dy_c = mxGetPr(plhs[4]);
dGx_dz_c = mxGetPr(plhs[5]);
dGy_dx_c = mxGetPr(plhs[6]);
dGy_dy_c = mxGetPr(plhs[7]);
dGy_dz_c = mxGetPr(plhs[8]);
dGz_dx_c = mxGetPr(plhs[9]);
dGz_dy_c = mxGetPr(plhs[10]);
dGz_dz_c = mxGetPr(plhs[11]);

/* Call the coil_G_mex subroutine. */
coil_G_mex(Gx_c,Gy_c,Gz_c,dGx_dx_c,dGx_dy_c,dGx_dz_c,dGy_dx_c,dGy_dy_c,dGy_dz_c,
dGz_dx_c,dGz_dy_c,dGz_dz_c,x,y,z,t,l,gd,a_0,b_0,diam,wind);
}
```

C.2 Simulink Top level Model and Subsystems

The following figures depict the Simulink block diagrams. The location of the block in the figure tree is given in the figure caption.

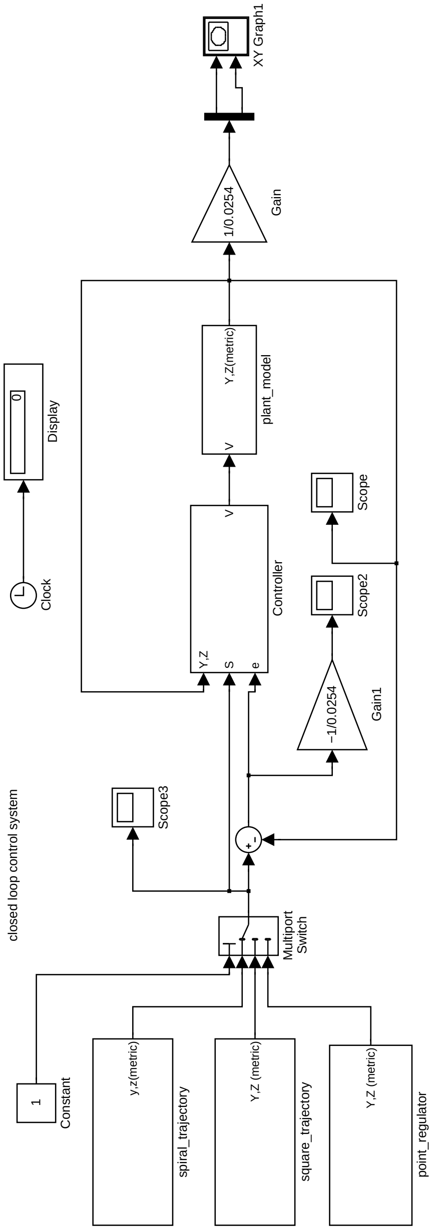

Figure C.1: Top level Simulink Model (“closed_loop”).

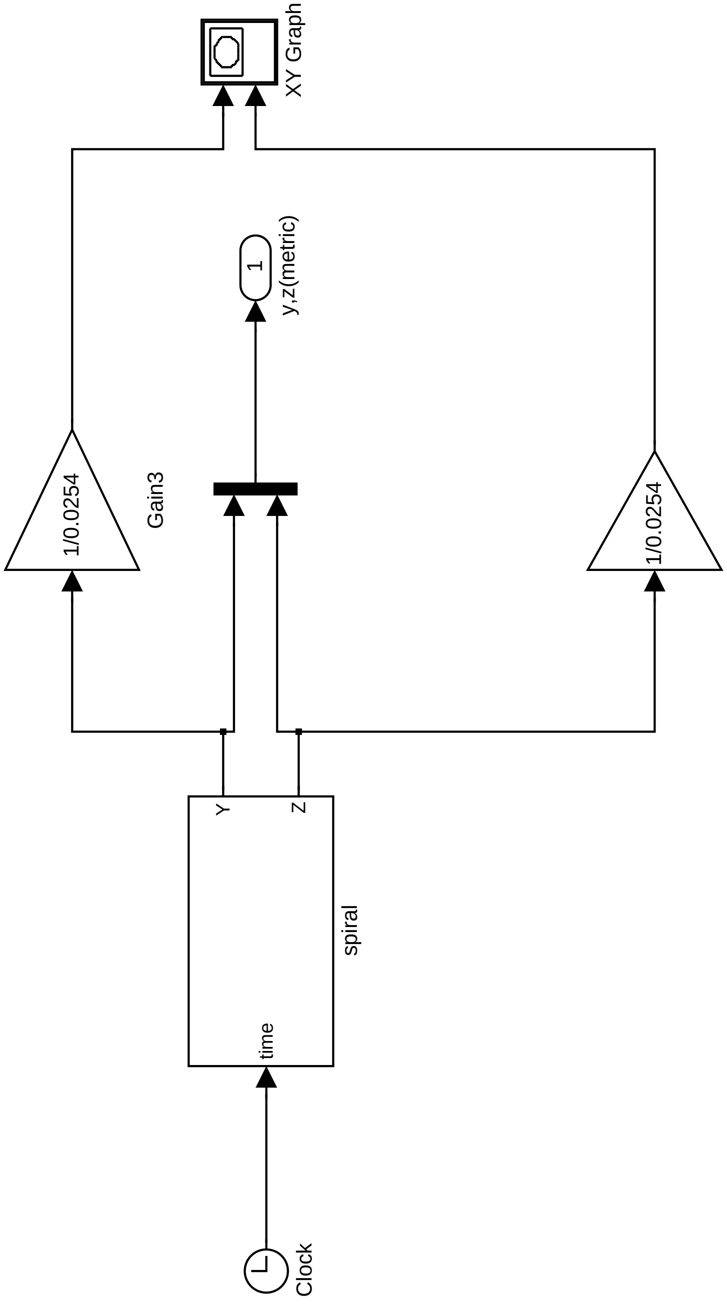

Figure C.2: Spiral trajectory subsystem (closed_loop→spiral_trajectory).

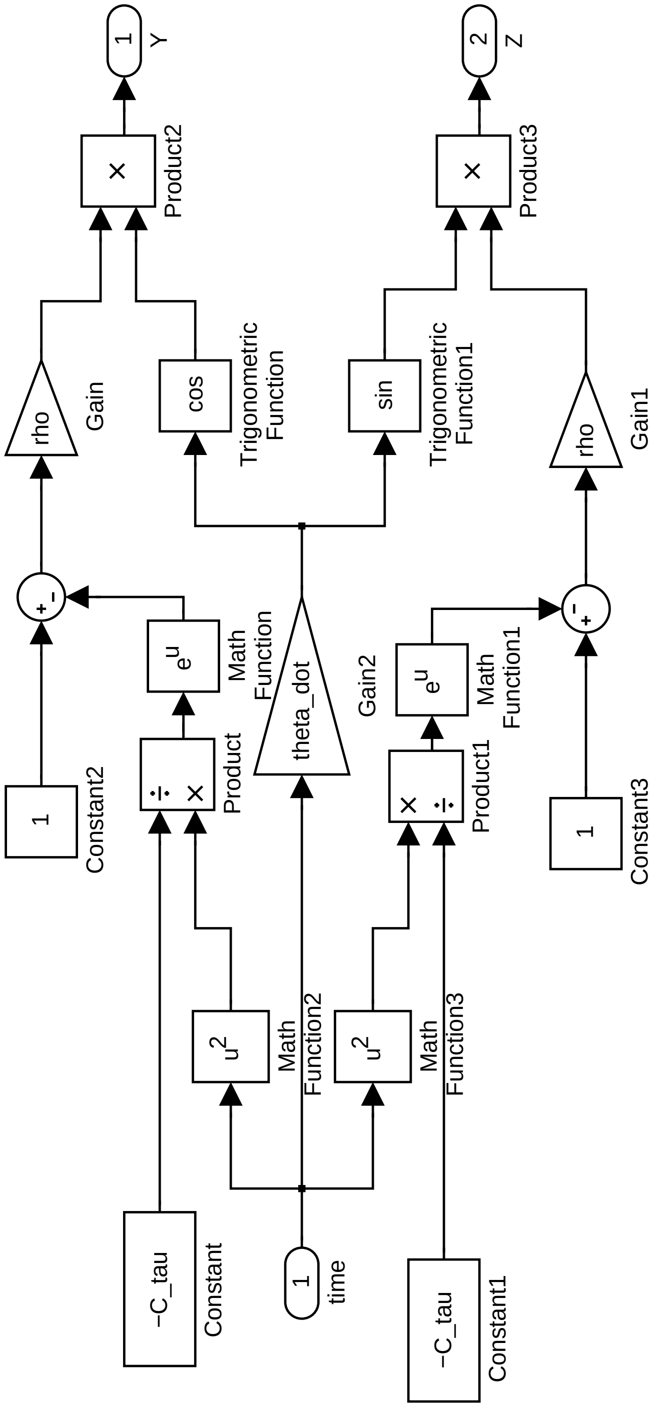

Figure C.3: Spiral trajectory path definition (closed_loop→spiral_trajectory→spiral).

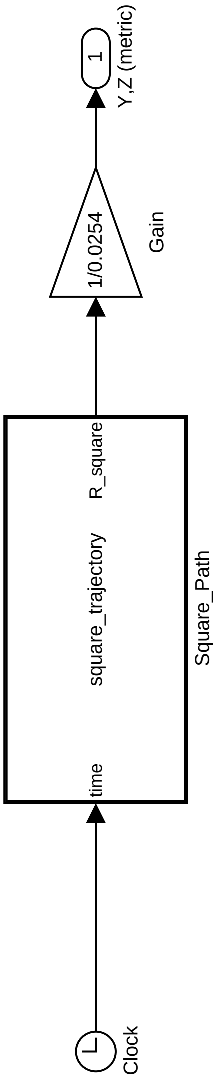

Figure C.4: Square trajectory subsystem (closed_loop→square_trajectory).

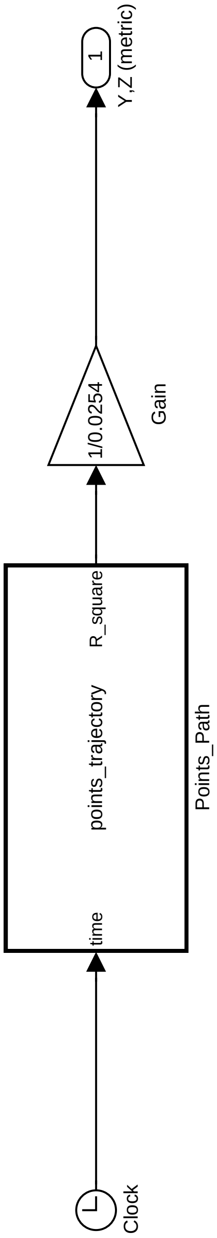

Figure C.5: Point regulator subsystem (closed_loop→point_regulator).

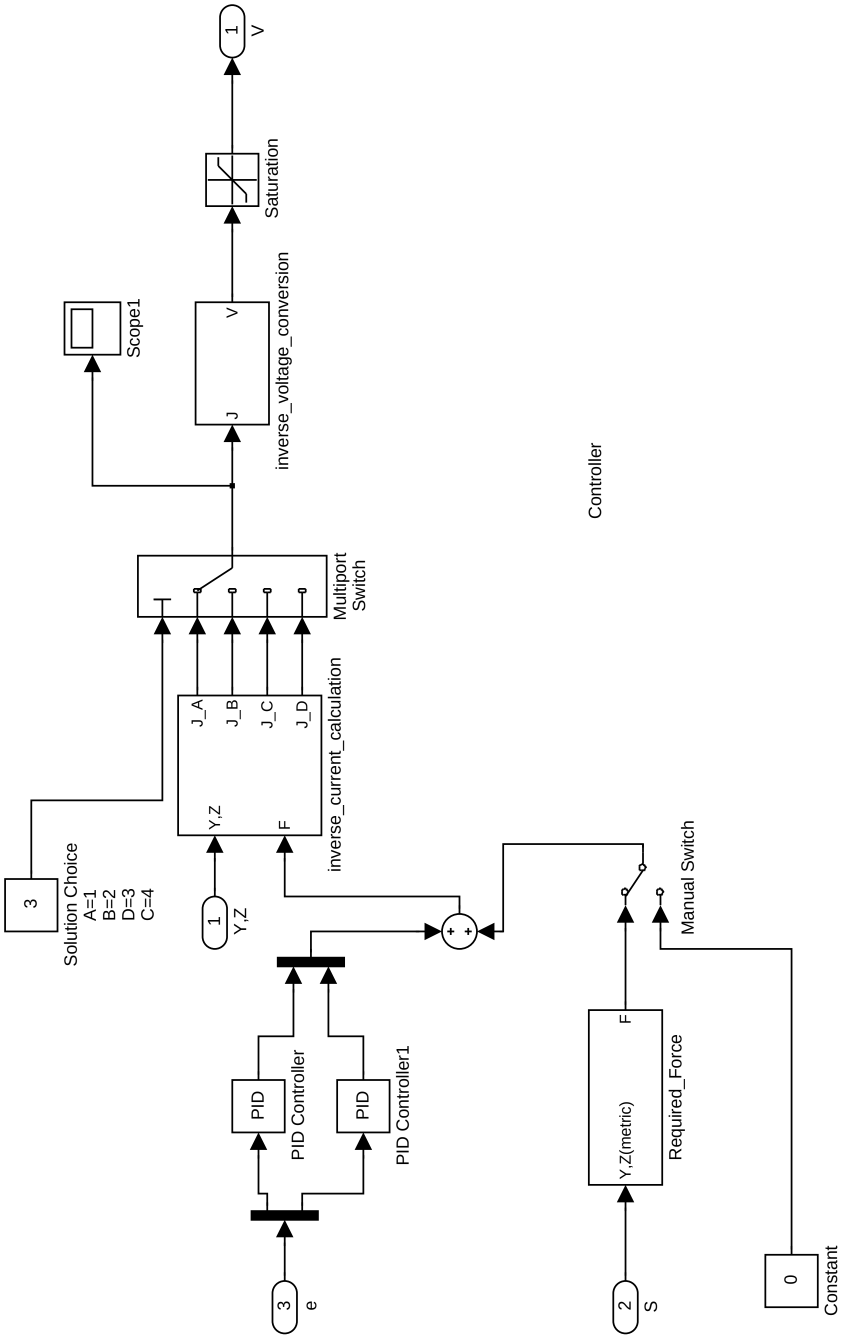

Figure C.6: Controller subsystem (closed_loop→Controller).

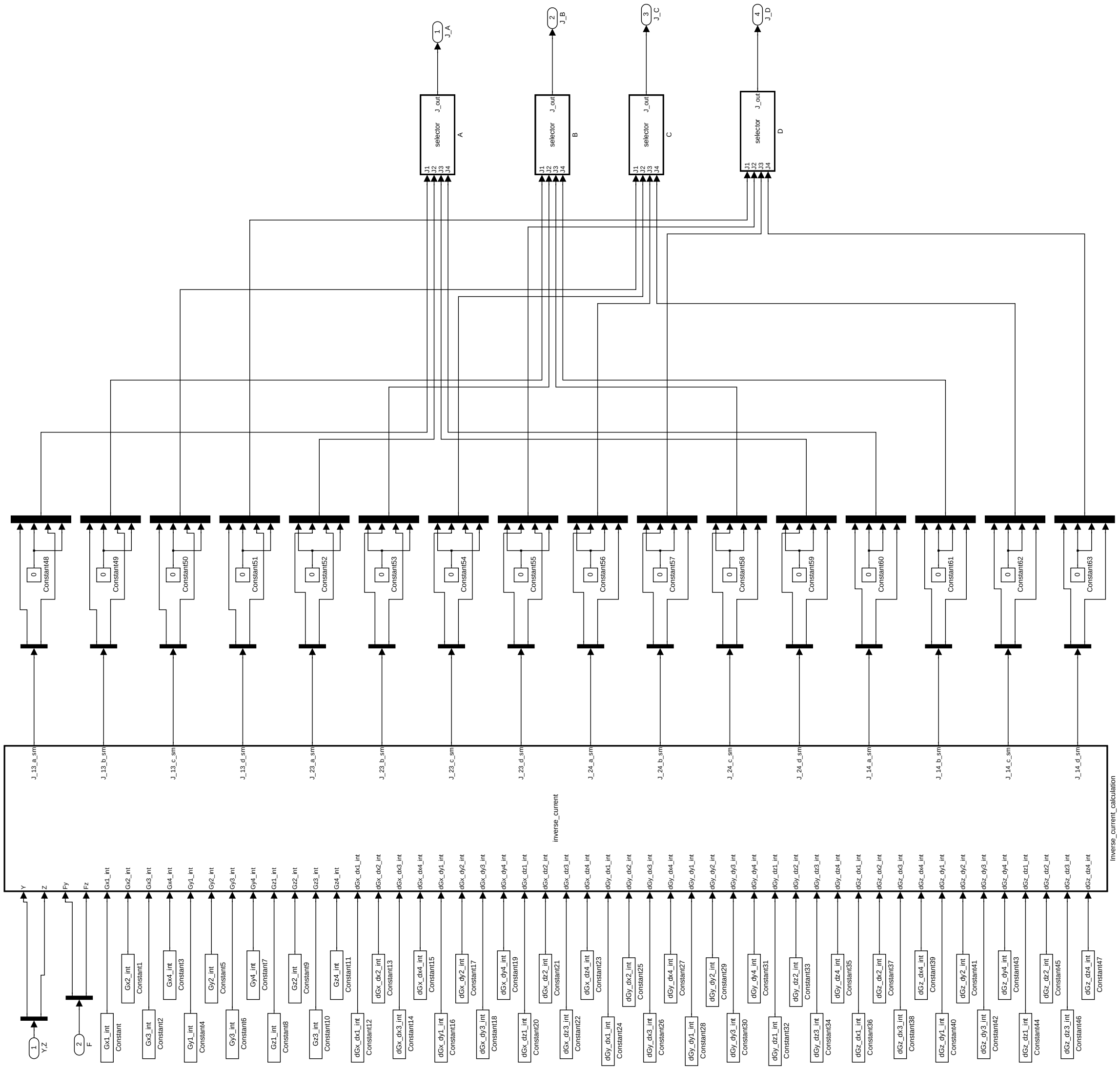

Figure C.7: Inverse current calculation (closed_loop→Controller→inverse_current_calculation).

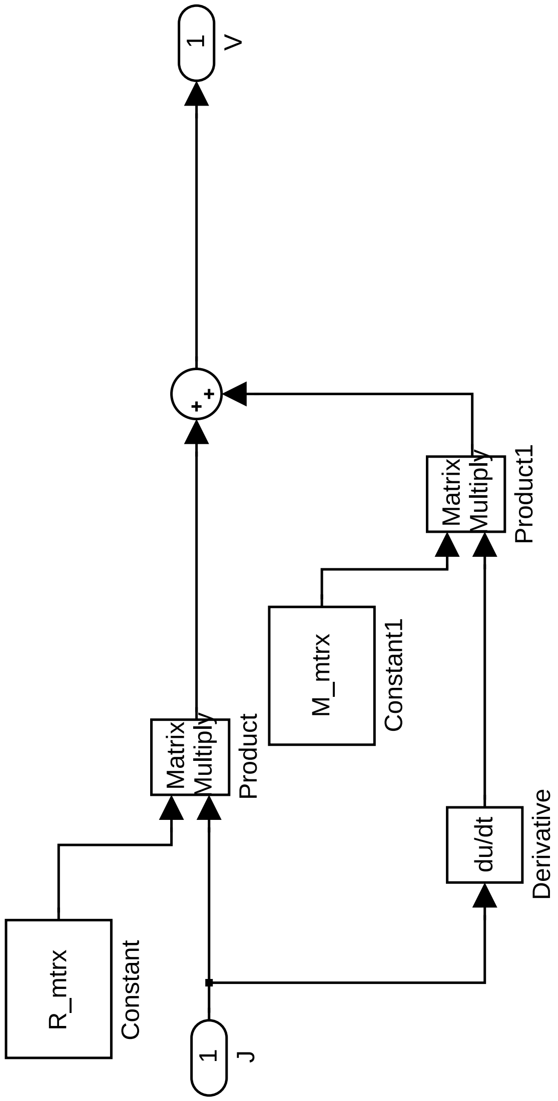

Figure C.8: Inverse voltage conversion (closed_loop→Controller→inverse_voltage_conversion).

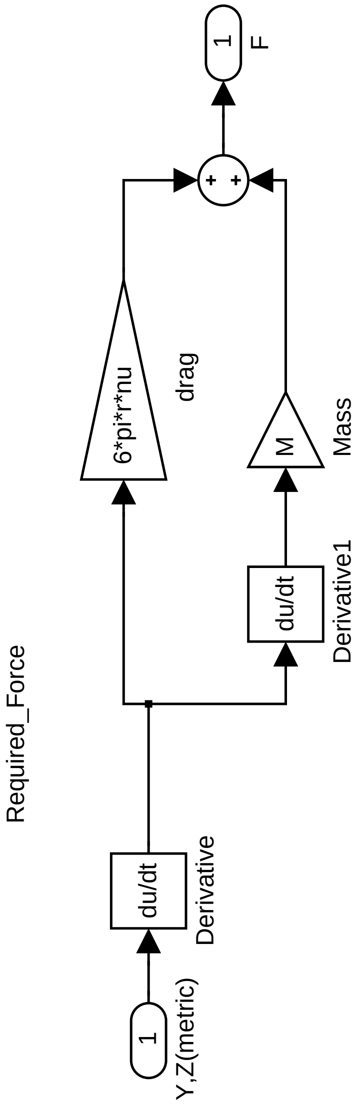

Figure C.9: Required force subsystem (closed_loop→Controller→required_force).

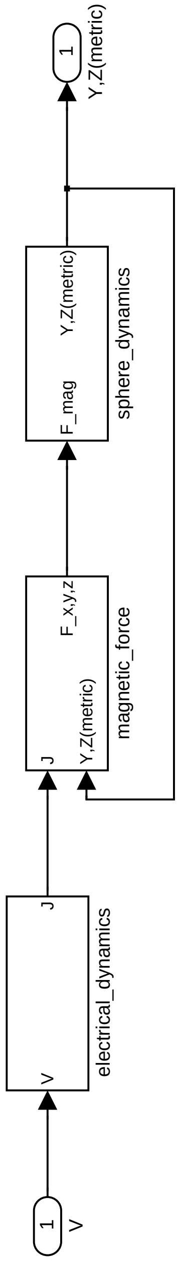

Figure C.10: Plant model subsystem (closed_loop→plant_model).

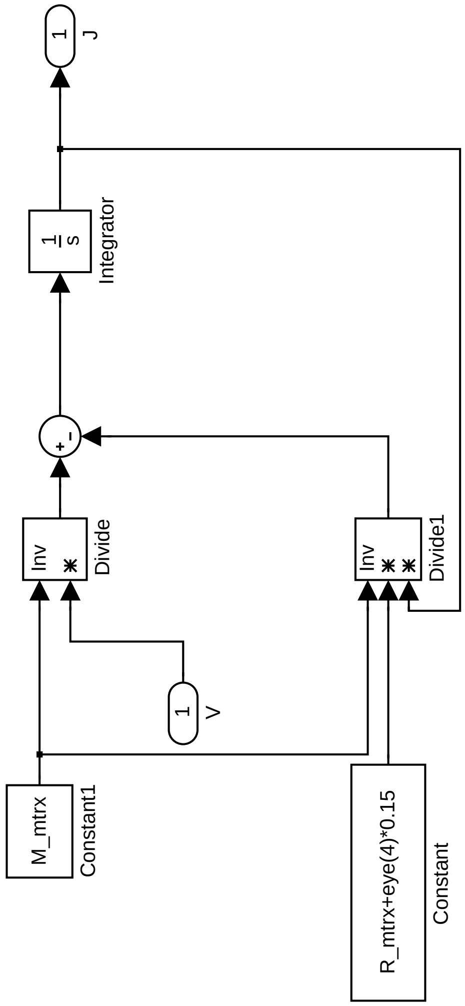

Figure C.11: Electrical dynamics subsystem (closed_loop→plant_model→electrical_dynamics).

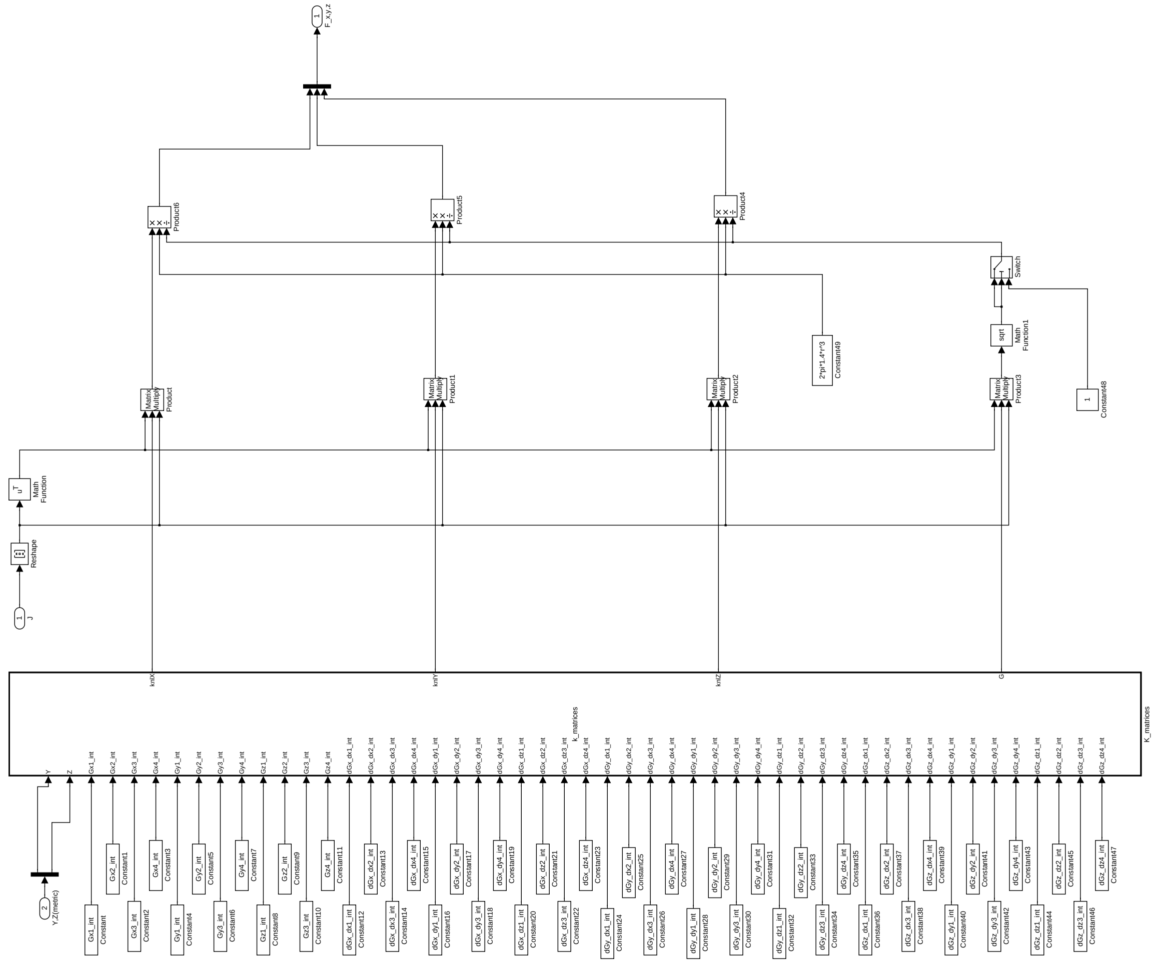

Figure C.12: Magnetic force subsystem (closed_loop→plant_model→magnetic_force).

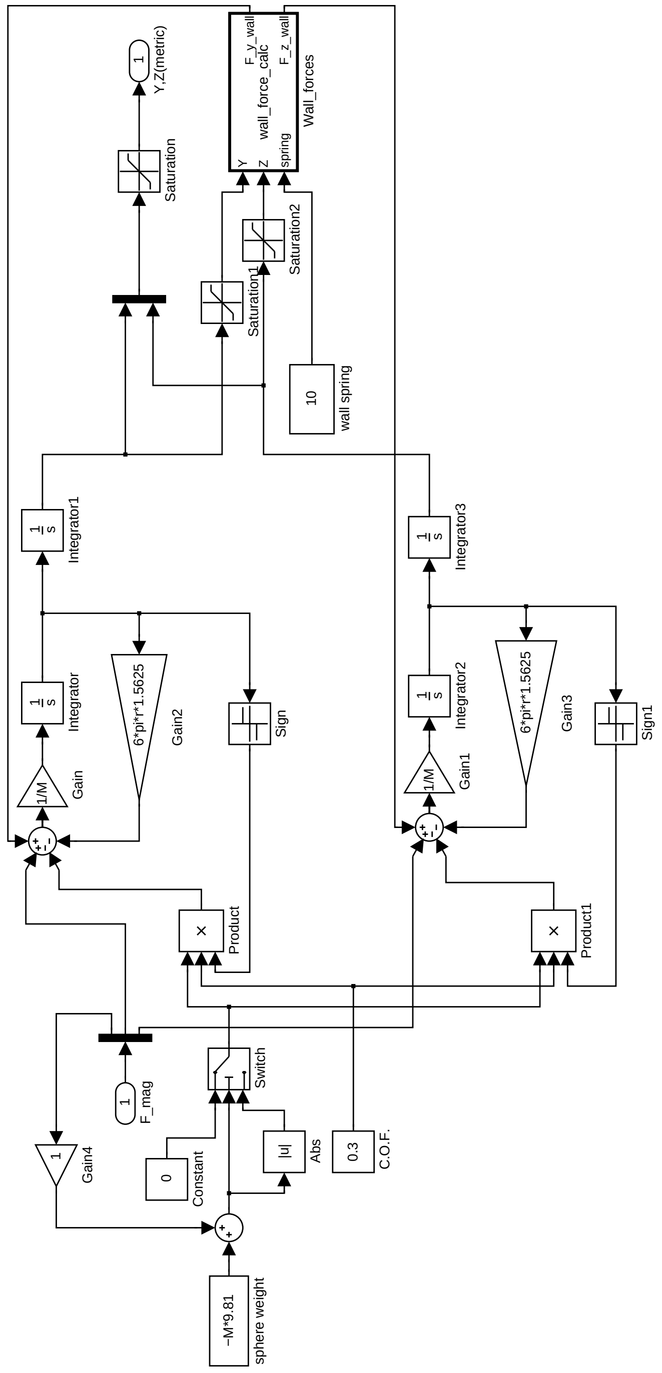

Figure C.13: Sphere dynamics subsystem (closed_loop→plant_model→sphere_dynamics).

C.3 Embedded Matlab Function Block Scripts

The following scripts are located in the embedded matlab function blocks. Location of the embedded function block before the script.

C.3.1 Square path embedded function.

(closed_loop→square_trajectory→Square_Path)

```
function R_square = square_trajectory(time,T_mult,Y_square,Z_square)
T=[0 1 2 3 4 5 6]*T_mult;
alpha=2*pi/(T(2));
CY=[0;0];
CZ=[0;0];
if time >= T(1) && time < T(2)
    CY=inv([T(1)-(1/alpha)*sin(alpha*T(1)) 1;
    T(2)-(1/alpha)*sin(alpha*T(2)) 1])*[Y_square(1);Y_square(2)];
    CZ=inv([T(1)-(1/alpha)*sin(alpha*T(1)) 1;
    T(2)-(1/alpha)*sin(alpha*T(2)) 1])*[Z_square(1);Z_square(2)];
elseif time >= T(2) && time < T(3)
    CY=inv([T(2)-(1/alpha)*sin(alpha*T(2)) 1;
    T(3)-(1/alpha)*sin(alpha*T(3)) 1])*[Y_square(2);Y_square(3)];
    CZ=inv([T(2)-(1/alpha)*sin(alpha*T(2)) 1;
    T(3)-(1/alpha)*sin(alpha*T(3)) 1])*[Z_square(2);Z_square(3)];
elseif time >= T(3) && time < T(4)
    CY=inv([T(3)-(1/alpha)*sin(alpha*T(3)) 1;
    T(4)-(1/alpha)*sin(alpha*T(4)) 1])*[Y_square(3);Y_square(4)];
    CZ=inv([T(3)-(1/alpha)*sin(alpha*T(3)) 1;
    T(4)-(1/alpha)*sin(alpha*T(4)) 1])*[Z_square(3);Z_square(4)];
elseif time >= T(4) && time < T(5)
    CY=inv([T(4)-(1/alpha)*sin(alpha*T(4)) 1;
    T(5)-(1/alpha)*sin(alpha*T(5)) 1])*[Y_square(4);Y_square(5)];
    CZ=inv([T(4)-(1/alpha)*sin(alpha*T(4)) 1;
    T(5)-(1/alpha)*sin(alpha*T(5)) 1])*[Z_square(4);Z_square(5)];
elseif time >= T(5) && time < T(6)
```

```
    CY=inv([T(5)-(1/alpha)*sin(alpha*T(5)) 1;
    T(6)-(1/alpha)*sin(alpha*T(6)) 1])*[Y_square(5);Y_square(6)];
    CZ=inv([T(5)-(1/alpha)*sin(alpha*T(5)) 1;
    T(6)-(1/alpha)*sin(alpha*T(6)) 1])*[Z_square(5);Z_square(6)];
elseif time >= T(6) && time < T(7)
    CY=inv([T(6)-(1/alpha)*sin(alpha*T(6)) 1;
    T(7)-(1/alpha)*sin(alpha*T(7)) 1])*[Y_square(6);Y_square(7)];
    CZ=inv([T(6)-(1/alpha)*sin(alpha*T(6)) 1;
    T(7)-(1/alpha)*sin(alpha*T(7)) 1])*[Z_square(6);Z_square(7)];
elseif time >= T(7)
    CY=[0;0];
    CZ=[0;0];
end
R_square=[CY(1);CZ(1)]*(time-(1/alpha)*sin(alpha*time))+[CY(2);CZ(2)];
```

C.3.2 Point path embedded function.

(closed_loop→point_regulator→Points_Path)

```
 function R_square = points_trajectory(time,T,Y_square,Z_square)
CY=0;
CZ=0;
if time >= T(1) && time < T(2)
    CY=Y_square(2);
    CZ=Z_square(2);
elseif time >= T(2) && time < T(3)
    CY=Y_square(3);
    CZ=Z_square(3);
elseif time >= T(3) && time < T(4)
    CY=Y_square(4);
    CZ=Z_square(4);
elseif time >= T(4) && time < T(5)
    CY=Y_square(5);
    CZ=Z_square(5);
```

```
elseif time >= T(5) && time < T(6)
    CY=Y_square(6);
    CZ=Z_square(6);
elseif time >= T(6) && time < T(7)
    CY=Y_square(7);
    CZ=Z_square(7);
elseif time >= T(7)
    CY=0;
    CZ=0;
end
R_square=[CY;CZ];
```

C.3.3 Inverse current calculation embedded function.

(closed_loop→Controller→inverse_current_calculation→Inverse_current_calculation)

```
function [J_13_a_sm,J_13_b_sm,J_13_c_sm,J_13_d_sm,J_23_a_sm,J_23_b_sm,J_23_c_sm,
J_23_d_sm,J_24_a_sm,J_24_b_sm,J_24_c_sm,J_24_d_sm,J_14_a_sm,J_14_b_sm,
J_14_c_sm,J_14_d_sm]=inverse_current(Y,Z,Fy,Fz,Br,Gx1_int, Gx2_int, Gx3_int,
Gx4_int, Gy1_int, Gy2_int, Gy3_int, Gy4_int, Gz1_int, Gz2_int, Gz3_int, Gz4_int,
dGx_dx1_int, dGx_dx2_int, dGx_dx3_int, dGx_dx4_int, dGx_dy1_int, dGx_dy2_int,
dGx_dy3_int, dGx_dy4_int, dGx_dz1_int, dGx_dz2_int, dGx_dz3_int, dGx_dz4_int,
dGy_dx1_int, dGy_dx2_int, dGy_dx3_int, dGy_dx4_int, dGy_dy1_int, dGy_dy2_int,
dGy_dy3_int, dGy_dy4_int, dGy_dz1_int, dGy_dz2_int, dGy_dz3_int, dGy_dz4_int,
dGz_dx1_int, dGz_dx2_int, dGz_dx3_int, dGz_dx4_int, dGz_dy1_int, dGz_dy2_int,
dGz_dy3_int, dGz_dy4_int, dGz_dz1_int, dGz_dz2_int, dGz_dz3_int, dGz_dz4_int,
R_1, R_2, R_3, R_4, r)
%function for use in the simulink model of the quad coil ball movement


theta_direc=atan2(Fz,Fy);
Cy=Fy/(2*pi*Br*r^3);
Cz=Fz/(2*pi*Br*r^3);
%get m and n of point
y_p_inch=Y/0.0254;
```

```
z_p_inch=Z/0.0254;
n_p=(y_p_inch+1)/0.05+1;
m_p=(z_p_inch+1)/0.05+1;

%put new interpolation technqnique here
Gx1=interpolation(n_p,m_p,Gx1_int);
Gy1=interpolation(n_p,m_p,Gy1_int);
Gz1=interpolation(n_p,m_p,Gz1_int);
dGx_dx1=interpolation(n_p,m_p,dGx_dx1_int);
dGx_dy1=interpolation(n_p,m_p,dGx_dy1_int);
dGx_dz1=interpolation(n_p,m_p,dGx_dz1_int);
dGy_dx1=interpolation(n_p,m_p,dGy_dx1_int);
dGy_dy1=interpolation(n_p,m_p,dGy_dy1_int);
dGy_dz1=interpolation(n_p,m_p,dGy_dz1_int);
dGz_dx1=interpolation(n_p,m_p,dGz_dx1_int);
dGz_dy1=interpolation(n_p,m_p,dGz_dy1_int);
dGz_dz1=interpolation(n_p,m_p,dGz_dz1_int);

Gx2=interpolation(n_p,m_p,Gx2_int);
Gy2=interpolation(n_p,m_p,Gy2_int);
Gz2=interpolation(n_p,m_p,Gz2_int);
dGx_dx2=interpolation(n_p,m_p,dGx_dx2_int);
dGx_dy2=interpolation(n_p,m_p,dGx_dy2_int);
dGx_dz2=interpolation(n_p,m_p,dGx_dz2_int);
dGy_dx2=interpolation(n_p,m_p,dGy_dx2_int);
dGy_dy2=interpolation(n_p,m_p,dGy_dy2_int);
dGy_dz2=interpolation(n_p,m_p,dGy_dz2_int);
dGz_dx2=interpolation(n_p,m_p,dGz_dx2_int);
dGz_dy2=interpolation(n_p,m_p,dGz_dy2_int);
dGz_dz2=interpolation(n_p,m_p,dGz_dz2_int);

Gx3=interpolation(n_p,m_p,Gx3_int);
Gy3=interpolation(n_p,m_p,Gy3_int);
```

```
Gz3=interpolation(n_p,m_p,Gz3_int);
dGx_dx3=interpolation(n_p,m_p,dGx_dx3_int);
dGx_dy3=interpolation(n_p,m_p,dGx_dy3_int);
dGx_dz3=interpolation(n_p,m_p,dGx_dz3_int);
dGy_dx3=interpolation(n_p,m_p,dGy_dx3_int);
dGy_dy3=interpolation(n_p,m_p,dGy_dy3_int);
dGy_dz3=interpolation(n_p,m_p,dGy_dz3_int);
dGz_dx3=interpolation(n_p,m_p,dGz_dx3_int);
dGz_dy3=interpolation(n_p,m_p,dGz_dy3_int);
dGz_dz3=interpolation(n_p,m_p,dGz_dz3_int);

Gx4=interpolation(n_p,m_p,Gx4_int);
Gy4=interpolation(n_p,m_p,Gy4_int);
Gz4=interpolation(n_p,m_p,Gz4_int);
dGx_dx4=interpolation(n_p,m_p,dGx_dx4_int);
dGx_dy4=interpolation(n_p,m_p,dGx_dy4_int);
dGx_dz4=interpolation(n_p,m_p,dGx_dz4_int);
dGy_dx4=interpolation(n_p,m_p,dGy_dx4_int);
dGy_dy4=interpolation(n_p,m_p,dGy_dy4_int);
dGy_dz4=interpolation(n_p,m_p,dGy_dz4_int);
dGz_dx4=interpolation(n_p,m_p,dGz_dx4_int);
dGz_dy4=interpolation(n_p,m_p,dGz_dy4_int);
dGz_dz4=interpolation(n_p,m_p,dGz_dz4_int);
% transform back into global coordinates
Gamma1=R_1*[Gx1; Gy1; Gz1];
Gamma2=R_2*[Gx2; Gy2; Gz2];
Gamma3=R_3*[Gx3; Gy3; Gz3];
Gamma4=R_4*[Gx4; Gy4; Gz4];

dGX1=R_1*[dGx_dx1; dGx_dy1; dGx_dz1];
dGY1=-R_1*[dGy_dx1; dGy_dy1; dGy_dz1];
dGZ1=-R_1*[dGz_dx1; dGz_dy1; dGz_dz1];
```

```
dGX2=R_2*[dGx_dx2; dGx_dy2; dGx_dz2];

dGY2=R_2*[dGy_dx2; dGy_dy2; dGy_dz2];

dGZ2=R_2*[dGz_dx2; dGz_dy2; dGz_dz2];


dGX3=R_3*[dGx_dx3; dGx_dy3; dGx_dz3];

dGY3=-R_3*[dGz_dx3; dGz_dy3; dGz_dz3];

dGZ3=R_3*[dGy_dx3; dGy_dy3; dGy_dz3];


dGX4=R_4*[dGx_dx4; dGx_dy4; dGx_dz4];

dGY4=R_4*[dGz_dx4; dGz_dy4; dGz_dz4];

dGZ4=-R_4*[dGy_dx4; dGy_dy4; dGy_dz4];

% make "G" and "dG/dn" vectors for each coil combination

%1 and 4

GX_14=[Gamma1(1); Gamma4(1)];

GY_14=[Gamma1(2); Gamma4(2)];

GZ_14=[Gamma1(3); Gamma4(3)];


dGX_dX_14=[dGX1(1); dGX4(1)];

dGX_dY_14=[dGX1(2); dGX4(2)];

dGX_dZ_14=[dGX1(3); dGX4(3)];


dGY_dX_14=[dGY1(1); dGY4(1)];

dGY_dY_14=[dGY1(2); dGY4(2)];

dGY_dZ_14=[dGY1(3); dGY4(3)];


dGZ_dX_14=[dGZ1(1); dGZ4(1)];

dGZ_dY_14=[dGZ1(2); dGZ4(2)];

dGZ_dZ_14=[dGZ1(3); dGZ4(3)];

% 1 and 3

GX_13=[Gamma1(1); Gamma3(1)];

GY_13=[Gamma1(2); Gamma3(2)];

GZ_13=[Gamma1(3); Gamma3(3)];
```

```
dGX_dX_13=[dGX1(1); dGX3(1)];
dGX_dY_13=[dGX1(2); dGX3(2)];
dGX_dZ_13=[dGX1(3); dGX3(3)];

dGY_dX_13=[dGY1(1); dGY3(1)];
dGY_dY_13=[dGY1(2); dGY3(2)];
dGY_dZ_13=[dGY1(3); dGY3(3)];

dGZ_dX_13=[dGZ1(1); dGZ3(1)];
dGZ_dY_13=[dGZ1(2); dGZ3(2)];
dGZ_dZ_13=[dGZ1(3); dGZ3(3)];
% 2 and 3
GX_23=[Gamma2(1); Gamma3(1)];
GY_23=[Gamma2(2); Gamma3(2)];
GZ_23=[Gamma2(3); Gamma3(3)];

dGX_dX_23=[dGX2(1); dGX3(1)];
dGX_dY_23=[dGX2(2); dGX3(2)];
dGX_dZ_23=[dGX2(3); dGX3(3)];

dGY_dX_23=[dGY2(1); dGY3(1)];
dGY_dY_23=[dGY2(2); dGY3(2)];
dGY_dZ_23=[dGY2(3); dGY3(3)];

dGZ_dX_23=[dGZ2(1); dGZ3(1)];
dGZ_dY_23=[dGZ2(2); dGZ3(2)];
dGZ_dZ_23=[dGZ2(3); dGZ3(3)];
% 2 and 4
GX_24=[Gamma2(1); Gamma4(1)];
GY_24=[Gamma2(2); Gamma4(2)];
GZ_24=[Gamma2(3); Gamma4(3)];

dGX_dX_24=[dGX2(1); dGX4(1)];
```

```
dGX_dY_24=[dGX2(2); dGX4(2)];
dGX_dZ_24=[dGX2(3); dGX4(3)];

dGY_dX_24=[dGY2(1); dGY4(1)];
dGY_dY_24=[dGY2(2); dGY4(2)];
dGY_dZ_24=[dGY2(3); dGY4(3)];

dGZ_dX_24=[dGZ2(1); dGZ4(1)];
dGZ_dY_24=[dGZ2(2); dGZ4(2)];
dGZ_dZ_24=[dGZ2(3); dGZ4(3)];

%Get "k" matrices for different coil combinations also make the
    %non-linear material "k"
knlX_14=GX_14*(dGX_dX_14’)+GY_14*(dGX_dY_14’)+GZ_14*(dGX_dZ_14’);
knlY_14=GX_14*(dGY_dX_14’)+GY_14*(dGY_dY_14’)+GZ_14*(dGY_dZ_14’);
knlZ_14=GX_14*(dGZ_dX_14’)+GY_14*(dGZ_dY_14’)+GZ_14*(dGZ_dZ_14’);

knlX_13=GX_13*(dGX_dX_13’)+GY_13*(dGX_dY_13’)+GZ_13*(dGX_dZ_13’);
knlY_13=GX_13*(dGY_dX_13’)+GY_13*(dGY_dY_13’)+GZ_13*(dGY_dZ_13’);
knlZ_13=GX_13*(dGZ_dX_13’)+GY_13*(dGZ_dY_13’)+GZ_13*(dGZ_dZ_13’);

knlX_23=GX_23*(dGX_dX_23’)+GY_23*(dGX_dY_23’)+GZ_23*(dGX_dZ_23’);
knlY_23=GX_23*(dGY_dX_23’)+GY_23*(dGY_dY_23’)+GZ_23*(dGY_dZ_23’);
knlZ_23=GX_23*(dGZ_dX_23’)+GY_23*(dGZ_dY_23’)+GZ_23*(dGZ_dZ_23’);

knlX_24=GX_24*(dGX_dX_24’)+GY_24*(dGX_dY_24’)+GZ_24*(dGX_dZ_24’);
knlY_24=GX_24*(dGY_dX_24’)+GY_24*(dGY_dY_24’)+GZ_24*(dGY_dZ_24’);
knlZ_24=GX_24*(dGZ_dX_24’)+GY_24*(dGZ_dY_24’)+GZ_24*(dGZ_dZ_24’);

% make the "g" matrices
g_14=GX_14*GX_14’+GY_14*GY_14’+GZ_14*GZ_14’;
g_13=GX_13*GX_13’+GY_13*GY_13’+GZ_13*GZ_13’;
g_23=GX_23*GX_23’+GY_23*GY_23’+GZ_23*GZ_23’;
```

```
g_24=GX_24*GX_24'+GY_24*GY_24'+GZ_24*GZ_24';

%define the "Omega" matrices
if Fy ~= 0
    Omega_14=knlZ_14-tan(theta_direc)*knlY_14;
    Omega_13=knlZ_13-tan(theta_direc)*knlY_13;
    Omega_23=knlZ_23-tan(theta_direc)*knlY_23;
    Omega_24=knlZ_24-tan(theta_direc)*knlY_24;
else
    Omega_14=knlY_14;
    Omega_13=knlY_13;
    Omega_23=knlY_23;
    Omega_24=knlY_24;
end
%make "omega" symmetric
Delta_tilde_14=(Omega_14+Omega_14')/2;
Delta_tilde_13=(Omega_13+Omega_13')/2;
Delta_tilde_23=(Omega_23+Omega_23')/2;
Delta_tilde_24=(Omega_24+Omega_24')/2;
    %find eigenvalues and eigenvectors
%analyticaly calculate the eigenvalues using characteristic equation
[lambda_14_1 lambda_14_2]=eigenvalue_calc(Delta_tilde_14(1,1),Delta_tilde_14(2,2),
Delta_tilde_14(1,2));
[lambda_13_1 lambda_13_2]=eigenvalue_calc(Delta_tilde_13(1,1),Delta_tilde_13(2,2),
Delta_tilde_13(1,2));
[lambda_23_1 lambda_23_2]=eigenvalue_calc(Delta_tilde_23(1,1),Delta_tilde_23(2,2),
Delta_tilde_23(1,2));
[lambda_24_1 lambda_24_2]=eigenvalue_calc(Delta_tilde_24(1,1),Delta_tilde_24(2,2),
Delta_tilde_24(1,2));
D_tilde_14=[lambda_14_1 0;0 lambda_14_2];
D_tilde_13=[lambda_13_1 0;0 lambda_13_2];
D_tilde_23=[lambda_23_1 0;0 lambda_23_2];
D_tilde_24=[lambda_24_1 0;0 lambda_24_2];
```

```
%analytically calculate the eigenvectors
[E_vec_14_1 E_vec_14_2]=eigenvector_calc(D_tilde_14(1,1),D_tilde_14(2,2),
Delta_tilde_14);
[E_vec_13_1 E_vec_13_2]=eigenvector_calc(D_tilde_13(1,1),D_tilde_13(2,2),
Delta_tilde_13);
[E_vec_23_1 E_vec_23_2]=eigenvector_calc(D_tilde_23(1,1),D_tilde_23(2,2),
Delta_tilde_23);
[E_vec_24_1 E_vec_24_2]=eigenvector_calc(D_tilde_24(1,1),D_tilde_24(2,2),
Delta_tilde_24);
Q_tilde_14=[E_vec_14_1 E_vec_14_2];
Q_tilde_13=[E_vec_13_1 E_vec_13_2];
Q_tilde_23=[E_vec_23_1 E_vec_23_2];
Q_tilde_24=[E_vec_24_1 E_vec_24_2];
%analytically calculate the determinant of Q_tilde
det_14=determinant_calc(Q_tilde_14);
det_13=determinant_calc(Q_tilde_13);
det_23=determinant_calc(Q_tilde_23);
det_24=determinant_calc(Q_tilde_24);
%define the "RY" and "RZ" matrices
RY_14=Q_tilde_14’*knlY_14*Q_tilde_14;
RY_13=Q_tilde_13’*knlY_13*Q_tilde_13;
RY_23=Q_tilde_23’*knlY_23*Q_tilde_23;
RY_24=Q_tilde_24’*knlY_24*Q_tilde_24;
RZ_14=Q_tilde_14’*knlZ_14*Q_tilde_14;
RZ_13=Q_tilde_13’*knlZ_13*Q_tilde_13;
RZ_23=Q_tilde_23’*knlZ_23*Q_tilde_23;
RZ_24=Q_tilde_24’*knlZ_24*Q_tilde_24;
%define the "xi" vectors
xi_14_1_plus=[1;sqrt(-D_tilde_14(1,1)/D_tilde_14(2,2))];
xi_14_1_minus=[1;-sqrt(-D_tilde_14(1,1)/D_tilde_14(2,2))];
xi_13_1_plus=[1;sqrt(-D_tilde_13(1,1)/D_tilde_13(2,2))];
xi_13_1_minus=[1;-sqrt(-D_tilde_13(1,1)/D_tilde_13(2,2))];
xi_23_1_plus=[1;sqrt(-D_tilde_23(1,1)/D_tilde_23(2,2))];
```

```
xi_23_1_minus=[1;-sqrt(-D_tilde_23(1,1)/D_tilde_23(2,2))];
xi_24_1_plus=[1;sqrt(-D_tilde_24(1,1)/D_tilde_24(2,2))];
xi_24_1_minus=[1;-sqrt(-D_tilde_24(1,1)/D_tilde_24(2,2))];

xi_14_2_plus=[sqrt(-D_tilde_14(2,2)/D_tilde_14(1,1));1];
xi_14_2_minus=[-sqrt(-D_tilde_14(2,2)/D_tilde_14(1,1));1];
xi_13_2_plus=[sqrt(-D_tilde_13(2,2)/D_tilde_13(1,1));1];
xi_13_2_minus=[-sqrt(-D_tilde_13(2,2)/D_tilde_13(1,1));1];
xi_23_2_plus=[sqrt(-D_tilde_23(2,2)/D_tilde_23(1,1));1];
xi_23_2_minus=[-sqrt(-D_tilde_23(2,2)/D_tilde_23(1,1));1];
xi_24_2_plus=[sqrt(-D_tilde_24(2,2)/D_tilde_24(1,1));1];
xi_24_2_minus=[-sqrt(-D_tilde_24(2,2)/D_tilde_24(1,1));1];

filt=1;   %set this to a non-zero value to filt out the incorrect solutions
%solution
if Fy~=0    %this will use the Y solution when Fy is /neq 0
    [J_13_a_sm J_13_b_sm J_13_c_sm J_13_d_sm]=time_step_currents_13_a(D_tilde_13,
    xi_13_1_plus,xi_13_1_minus,xi_13_2_plus,xi_13_2_minus,g_13,Cy,Fy,Fz,
    Q_tilde_13,RY_13,knlY_13,knlZ_13,Br,r,filt);
    [J_14_a_sm J_14_b_sm J_14_c_sm J_14_d_sm]=time_step_currents_14_a(D_tilde_14,
    xi_14_1_plus,xi_14_1_minus,xi_14_2_plus,xi_14_2_minus,g_14,Cy,Fy,Fz,
    Q_tilde_14,RY_14,knlY_14,knlZ_14,Br,r,filt);
    [J_23_a_sm J_23_b_sm J_23_c_sm J_23_d_sm]=time_step_currents_23_a(D_tilde_23,
    xi_23_1_plus,xi_23_1_minus,xi_23_2_plus,xi_23_2_minus,g_23,Cy,Fy,Fz,
    Q_tilde_23,RY_23,knlY_23,knlZ_23,Br,r,filt);
    [J_24_a_sm J_24_b_sm J_24_c_sm J_24_d_sm]=time_step_currents_24_a(D_tilde_24,
    xi_24_1_plus,xi_24_1_minus,xi_24_2_plus,xi_24_2_minus,g_24,Cy,Fy,Fz,
    Q_tilde_24,RY_24,knlY_24,knlZ_24,Br,r,filt);

elseif Fz~=0    %this will use the Z solution when Fz is /neq 0 and Fy=0
    [J_13_a_sm J_13_b_sm J_13_c_sm J_13_d_sm]=time_step_currents_13_a(D_tilde_13,
    xi_13_1_plus,xi_13_1_minus,xi_13_2_plus,xi_13_2_minus,g_13,Cz,Fy,Fz,
    Q_tilde_13,RZ_13,knlY_13,knlZ_13,Br,r,filt);
    [J_14_a_sm J_14_b_sm J_14_c_sm J_14_d_sm]=time_step_currents_14_a(D_tilde_14,
    xi_14_1_plus,xi_14_1_minus,xi_14_2_plus,xi_14_2_minus,g_14,Cz,Fy,Fz,
    Q_tilde_14,RZ_14,knlY_14,knlZ_14,Br,r,filt);
    [J_23_a_sm J_23_b_sm J_23_c_sm J_23_d_sm]=time_step_currents_23_a(D_tilde_23,
    xi_23_1_plus,xi_23_1_minus,xi_23_2_plus,xi_23_2_minus,g_23,Cz,Fy,Fz,
    Q_tilde_23,RZ_23,knlY_23,knlZ_23,Br,r,filt);
    [J_24_a_sm J_24_b_sm J_24_c_sm J_24_d_sm]=time_step_currents_24_a(D_tilde_24,
    xi_24_1_plus,xi_24_1_minus,xi_24_2_plus,xi_24_2_minus,g_24,Cz,Fy,Fz,
    Q_tilde_24,RZ_24,knlY_24,knlZ_24,Br,r,filt);

else
    J_13_a_sm=[0; 0];
    J_13_b_sm=[0; 0];
    J_13_c_sm=[0; 0];
    J_13_d_sm=[0; 0];

    J_14_a_sm=[0; 0];
    J_14_b_sm=[0; 0];
    J_14_c_sm=[0; 0];
    J_14_d_sm=[0; 0];

    J_23_a_sm=[0; 0];
    J_23_b_sm=[0; 0];
    J_23_c_sm=[0; 0];
    J_23_d_sm=[0; 0];

    J_24_a_sm=[0; 0];
    J_24_b_sm=[0; 0];
    J_24_c_sm=[0; 0];
    J_24_d_sm=[0; 0];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function f=interpolation(n_p,m_p,A)
%function to interpolate from the offline field calculations.  A is a mXn
%matrix and y_p,z_p is the coordinate to be interplated at

%step 1, Identify the box the point is in
%y_p_inch=y_p/0.0254;
%z_p_inch=z_p/0.0254;
%n_p=(y_p_inch+2)/0.1+1;
%m_p=(z_p_inch+2)/0.1+1;
%n_p and m_p will most likely NOT be an integer, so find box corners
n_up=ceil(n_p);
n_bot=floor(n_p);
m_up=ceil(m_p);
m_bot=floor(m_p);
if m_bot==m_up && n_bot==n_up
    f=A(m_p,n_p);
elseif m_bot==m_up
    s=(A(m_bot,n_up)-A(m_bot,n_bot))/(n_up-n_bot);
    b=A(m_bot,n_bot)-s*n_bot;
    f=s*n_p+b;
elseif n_bot==n_up
    s=(A(m_up,n_bot)-A(m_bot,n_bot))/(m_up-m_bot);
    b=A(m_bot,n_bot)-s*m_bot;
    f=s*m_p+b;
elseif -n_p+(m_up+n_bot)>=m_p
    v1=[n_bot m_bot A(m_bot,n_bot)];
    v2=[n_up m_bot A(m_bot,n_up)];
    v3=[n_bot m_up A(m_up,n_bot)];
    C1=(v3(2)-v1(2))*(v2(3)-v1(3))-(v3(3)-v1(3))*(v2(2)-v1(2));
    C2=(v3(3)-v1(3))*(v2(1)-v1(1))-(v3(1)-v1(1))*(v2(3)-v1(3));
    C3=(v3(1)-v1(1))*(v2(2)-v1(2))-(v3(2)-v1(2))*(v2(1)-v1(1));
    C=[C1 C2 C3];
    D=-C(1)*v1(1)-C(2)*v1(2)-C(3)*v1(3);
    f=(-D-C(1)*(n_p)-C(2)*(m_p))/C(3);
else
    v1=[n_up m_up A(m_up,n_up)];
    v3=[n_up m_bot A(m_bot,n_up)];
    v2=[n_bot m_up A(m_up,n_bot)];
    C1=(v3(2)-v1(2))*(v2(3)-v1(3))-(v3(3)-v1(3))*(v2(2)-v1(2));
    C2=(v3(3)-v1(3))*(v2(1)-v1(1))-(v3(1)-v1(1))*(v2(3)-v1(3));
    C3=(v3(1)-v1(1))*(v2(2)-v1(2))-(v3(2)-v1(2))*(v2(1)-v1(1));
    C=[C1 C2 C3];
    D=-C(1)*v1(1)-C(2)*v1(2)-C(3)*v1(3);
    f=(-D-C(1)*(n_p)-C(2)*(m_p))/C(3);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [lambda_1, lambda_2]=eigenvalue_calc(a_1,a_2,a_3)
%custom function for calculating the eigenvalues of a 2x2 symmetric matrix.  This
%uses the characteristic equation for the matrix A=[a_1 a_3;a_3 a_2].
lambda_2=((a_1+a_2)+sqrt((a_1+a_2)^2-4*(a_1*a_2-a_3^2)))/2;
lambda_1=((a_1+a_2)-sqrt((a_1+a_2)^2-4*(a_1*a_2-a_3^2)))/2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [eigenvec_1,eigenvec_2]=eigenvector_calc(lambda_1,lambda_2,A)
%custom function to calculate the normalized eigenvectors of a 2x2 symmetric matrix.
%The technique uses the adjoint of (A-I*lambda).
Adjoint_1=[A(2,2)-lambda_1 -A(1,2);-A(2,1) A(1,1)-lambda_1];
Adjoint_2=[A(2,2)-lambda_2 -A(1,2);-A(2,1) A(1,1)-lambda_2];
%initialize outputs
eigenvec_1=[0;0];
eigenvec_2=[0;0];
if Adjoint_1(1,1)~=0
    eigenvec_1=(1/sqrt(Adjoint_1(1,1)^2+Adjoint_1(2,1)^2))*[Adjoint_1(1,1);
    Adjoint_1(2,1)];
elseif Adjoint_1(2,1)~=0
```

```
    eigenvec_1=(1/sqrt(Adjoint_1(1,1)^2+Adjoint_1(2,1)^2))*[Adjoint_1(1,1);
    Adjoint_1(2,1)];
elseif Adjoint_1(1,2)~=0
    eigenvec_1=(1/sqrt(Adjoint_1(1,2)^2+Adjoint_1(2,2)^2))*[Adjoint_1(1,2);
    Adjoint_1(2,2)];
elseif Adjoint_1(2,2)~=0
    eigenvec_1=(1/sqrt(Adjoint_1(1,2)^2+Adjoint_1(2,2)^2))*[Adjoint_1(1,2);
    Adjoint_1(2,2)];
end
if Adjoint_2(1,1)~=0
    eigenvec_2=(1/sqrt(Adjoint_2(1,1)^2+Adjoint_2(2,1)^2))*[Adjoint_2(1,1);
    Adjoint_2(2,1)];
elseif Adjoint_2(2,1)~=0
    eigenvec_2=(1/sqrt(Adjoint_2(1,1)^2+Adjoint_2(2,1)^2))*[Adjoint_2(1,1);
    Adjoint_2(2,1)];
elseif Adjoint_2(1,2)~=0
    eigenvec_2=(1/sqrt(Adjoint_2(1,2)^2+Adjoint_2(2,2)^2))*[Adjoint_2(1,2);
    Adjoint_2(2,2)];
elseif Adjoint_2(2,2)~=0
    eigenvec_2=(1/sqrt(Adjoint_2(1,2)^2+Adjoint_2(2,2)^2))*[Adjoint_2(1,2);
    Adjoint_2(2,2)];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function file to calculate currents at each time step
function [J_a_sm,J_b_sm,J_c_sm,J_d_sm]=time_step_currents_13_a(D_tilde,xi_1_plus,
xi_1_minus,xi_2_plus,xi_2_minus,g,Cy,Fy,Fz,Q_tilde,RY,knlY,knlZ,Br,r,filt)
        if D_tilde(1,1)*D_tilde(2,2)>0   %check for definiteness (eigenvalue products)
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
            J_b_sm=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            J_tilde_a=[0; 0];
            J_tilde_b=[0; 0];
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            d=0;
        elseif D_tilde(2,2)~=0   %this loop will use formulation when second
        %eigenvalue is /neq 0
            xi_1=[1;sqrt(-D_tilde(1,1)/D_tilde(2,2))];
            if xi_1_plus'*g*xi_1_plus>0    %get solutions a and b, check for
            %the possible imaginary result and skip if necessary
                j_tilde_1_plus_plus=Cy*sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)
                /(xi_1_plus'*RY*xi_1_plus);
                j_tilde_1_minus_plus=-Cy*sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)
                /(xi_1_plus'*RY*xi_1_plus);
                j_tilde_2_plus_plus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*Cy*
                sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)/
                (xi_1_plus'*RY*xi_1_plus);
                j_tilde_2_minus_plus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*Cy*
                sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)/
                (xi_1_plus'*RY*xi_1_plus);
                d=determinant_calc(Q_tilde);
                %%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
                j_tilde_1_plus_plus_sm=sign(d)*Cy*sqrt([xi_1(1);sign(d)*xi_1(2)]'*
                Q_tilde'*g*Q_tilde*[xi_1(1);sign(d)*xi_1(2)])/([xi_1(1);
                sign(d)*xi_1(2)]'*RY*[xi_1(1);sign(d)*xi_1(2)]);
                j_tilde_1_minus_plus_sm=-sign(d)*Cy*sqrt([xi_1(1);
                sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
                sign(d)*xi_1(2)])/([xi_1(1);sign(d)*xi_1(2)]'*RY*[xi_1(1);
                sign(d)*xi_1(2)]);
                j_tilde_2_plus_plus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
```

```
Cy*sqrt([xi_1(1);sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
sign(d)*xi_1(2)])/([xi_1(1);sign(d)*xi_1(2)]'*RY*[xi_1(1);
sign(d)*xi_1(2)]);
j_tilde_2_minus_plus_sm=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*Cy*
sqrt([xi_1(1);sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*
[xi_1(1);sign(d)*xi_1(2)])/([xi_1(1);sign(d)*xi_1(2)]'*RY*[xi_1(1);
sign(d)*xi_1(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%


J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;
%plug back into magnetic force formulation to evaluate validity
%of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];


F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
```

```
    [J_a(1); J_a(2)];
    F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
    [J_b(1); J_b(2)];
    %F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
    %[J_c(1); J_c(2)];
    %F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
    %[J_d(1); J_d(2)];

    %get force direction and compare to desired
    F_dir=(180/pi)*atan2(Fz,Fy);
    F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
    F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
    %F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
    %F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
    if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
    abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
        J_tilde_a=[0; 0];
        J_tilde_b=[0; 0];
        J_a=[0; 0];
        J_b=[0; 0];
        J_a_sm=[0; 0];
        J_b_sm=[0; 0];
        d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
```

```
if xi_1_minus'*g*xi_1_minus>0    %get solutions c and d, check for
%the possible imaginary result and skip if necessary
    j_tilde_1_plus_minus=Cy*sqrt(xi_1_minus'*Q_tilde'*g*
    Q_tilde*xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
    j_tilde_1_minus_minus=-Cy*sqrt(xi_1_minus'*Q_tilde'*g*
    Q_tilde*xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
    j_tilde_2_minus_minus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
    Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
    (xi_1_minus'*RY*xi_1_minus);
    j_tilde_2_plus_minus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
    Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
    (xi_1_minus'*RY*xi_1_minus);
    d=determinant_calc(Q_tilde);
    %%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
    j_tilde_1_plus_minus_sm=sign(d)*Cy*sqrt([xi_1(1);
    -sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
    -sign(d)*xi_1(2)])/([xi_1(1);-sign(d)*xi_1(2)]'*RY*[xi_1(1);
    -sign(d)*xi_1(2)]);
    j_tilde_1_minus_minus_sm=-sign(d)*Cy*sqrt([xi_1(1);
    -sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
    -sign(d)*xi_1(2)])/([xi_1(1);-sign(d)*xi_1(2)]'*RY*[xi_1(1);
    -sign(d)*xi_1(2)]);
    j_tilde_2_minus_minus_sm=-sqrt(-D_tilde(1,1)/
    D_tilde(2,2))*Cy*sqrt([xi_1(1);-sign(d)*xi_1(2)]'*Q_tilde'*
    g*Q_tilde*[xi_1(1);-sign(d)*xi_1(2)])/([xi_1(1);
    -sign(d)*xi_1(2)]'*RY*[xi_1(1);-sign(d)*xi_1(2)]);
    j_tilde_2_plus_minus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
    Cy*sqrt([xi_1(1);-sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*
    [xi_1(1);-sign(d)*xi_1(2)])/([xi_1(1);-sign(d)*xi_1(2)]'*
    RY*[xi_1(1);-sign(d)*xi_1(2)]);
    J_tilde_c_sm=[j_tilde_1_plus_minus_sm;
    j_tilde_2_minus_minus_sm];
    J_tilde_d_sm=[j_tilde_1_minus_minus_sm;
j_tilde_2_plus_minus_sm];
J_c_sm=Q_tilde*J_tilde_c_sm;
J_d_sm=Q_tilde*J_tilde_d_sm;
%%%%%%%%%%%%%%%%%%%%%%%


J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;


%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
[J_c(1); J_c(2)];
F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
[J_d(1); J_d(2)];


%F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
%[J_a(1); J_a(2)];
%F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
%[J_b(1); J_b(2)];
F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
[J_c(1); J_c(2)];
F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
[J_d(1); J_d(2)];
```

```
        %get force direction and compare to desired
        F_dir=(180/pi)*atan2(Fz,Fy);
        %F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
        %F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
        F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
        F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
        if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
        abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            d=0;
        end
    else
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
elseif D_tilde(1,1)~=0   %this loop will use formulation when first
%eigenvalue is /neq 0
    xi_2=[sqrt(-D_tilde(2,2)/D_tilde(1,1));1];
    if xi_2_plus'*g*xi_2_plus>0    %get solutions a and b, check for the
    %possible imaginary result and skip if necessary
        j_tilde_2_plus_plus=Cy*sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*
        xi_2_plus)/(xi_2_plus'*RY*xi_2_plus);
```

```
j_tilde_2_minus_plus=-Cy*sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*
xi_2_plus)/(xi_2_plus'*RY*xi_2_plus);
j_tilde_1_plus_plus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*xi_2_plus)/(xi_2_plus'*
RY*xi_2_plus);
j_tilde_1_minus_plus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*xi_2_plus)/(xi_2_plus'*
RY*xi_2_plus);
d=determinant_calc(Q_tilde);
%%%%%%%%%%%%%%%%%%%%%%%get smooth currents now
j_tilde_2_plus_plus_sm=Cy*sqrt([sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*
g*Q_tilde*[sign(d)*xi_2(1);xi_2(2)])/([sign(d)*xi_2(1);xi_2(2)]'*RY*
[sign(d)*xi_2(1);xi_2(2)]);
j_tilde_2_minus_plus_sm=-Cy*sqrt([sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*
g*Q_tilde*[sign(d)*xi_2(1);xi_2(2)])/([sign(d)*xi_2(1);xi_2(2)]'*RY*
[sign(d)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_plus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*sign(d)*
Cy*sqrt([sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d)*xi_2(1);
xi_2(2)])/([sign(d)*xi_2(1);xi_2(2)]'*RY*[sign(d)*xi_2(1);xi_2(2)]);
j_tilde_1_minus_plus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*sign(d)*
Cy*sqrt([sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d)*xi_2(1);
xi_2(2)])/([sign(d)*xi_2(1);xi_2(2)]'*RY*[sign(d)*xi_2(1);xi_2(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%
J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;


%plug back into magnetic force formulation to evaluate
```

```
%validity of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];

F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
[J_a(1); J_a(2)];
F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
[J_b(1); J_b(2)];
%F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
%[J_c(1); J_c(2)];
%F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
%[J_d(1); J_d(2)];

%get force direction and compare to desired
F_dir=(180/pi)*atan2(Fz,Fy);
F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
%F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
%F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
```

```
        J_a=[0; 0];
        J_b=[0; 0];
        J_a_sm=[0; 0];
        J_b_sm=[0; 0];
        d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
if xi_2_minus'*g*xi_2_minus>0    %get solutions c and d, check for the
%possible imaginary result and skip if necessary
    j_tilde_2_plus_minus=Cy*sqrt(xi_2_minus'*Q_tilde'*g*
    Q_tilde*xi_2_minus)/(xi_2_minus'*RY*xi_2_minus);
    j_tilde_2_minus_minus=-Cy*sqrt(xi_2_minus'*Q_tilde'*g*
    Q_tilde*xi_2_minus)/(xi_2_minus'*RY*xi_2_minus);
    j_tilde_1_minus_minus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
    Cy*sqrt(xi_2_minus'*Q_tilde'*g*Q_tilde*xi_2_minus)/
    (xi_2_minus'*RY*xi_2_minus);
    j_tilde_1_plus_minus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
    Cy*sqrt(xi_2_minus'*Q_tilde'*g*Q_tilde*xi_2_minus)/
    (xi_2_minus'*RY*xi_2_minus);
    d=determinant_calc(Q_tilde);
    %%%%%%%%%%%%%%%%%%%%%%get smooth trajectories now
    j_tilde_2_plus_minus_sm=Cy*sqrt([-sign(d)*xi_2(1);xi_2(2)]'*
    Q_tilde'*g*Q_tilde*[-sign(d)*xi_2(1);xi_2(2)])/([-sign(d)*xi_2(1);
    xi_2(2)]'*RY*[-sign(d)*xi_2(1);xi_2(2)]);
    j_tilde_2_minus_minus_sm=-Cy*sqrt([-sign(d)*xi_2(1);xi_2(2)]'*
Q_tilde'*g*Q_tilde*[-sign(d)*xi_2(1);xi_2(2)])/([-sign(d)*xi_2(1);
xi_2(2)]'*RY*[-sign(d)*xi_2(1);xi_2(2)]);
j_tilde_1_minus_minus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
sign(d)*Cy*sqrt([-sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*
[-sign(d)*xi_2(1);xi_2(2)])/([-sign(d)*xi_2(1);xi_2(2)]'*RY*
[-sign(d)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_minus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
sign(d)*Cy*sqrt([-sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*
[-sign(d)*xi_2(1);xi_2(2)])/([-sign(d)*xi_2(1);xi_2(2)]'*RY*
[-sign(d)*xi_2(1);xi_2(2)]);
J_tilde_c_sm=[j_tilde_1_plus_minus_sm;
j_tilde_2_minus_minus_sm];
J_tilde_d_sm=[j_tilde_1_minus_minus_sm;
j_tilde_2_plus_minus_sm];
J_c_sm=Q_tilde*J_tilde_c_sm;
J_d_sm=Q_tilde*J_tilde_d_sm;
%%%%%%%%%%%%%%%%%%%%%%
J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;

%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
```

```
    [J_c(1); J_c(2)];
    F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
    [J_d(1); J_d(2)];

    %F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
    %[J_a(1); J_a(2)];
    %F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
    %[J_b(1); J_b(2)];
    F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
    [J_c(1); J_c(2)];
    F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
    [J_d(1); J_d(2)];

    %get force direction and compare to desired
    F_dir=(180/pi)*atan2(Fz,Fy);
    %F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
    %F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
    F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
    F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
    if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
    abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
else
    J_tilde_c=[0; 0];
    J_tilde_d=[0; 0];
    J_c=[0; 0];
```

```
                J_d=[0; 0];
                J_c_sm=[0; 0];
                J_d_sm=[0; 0];
                d=0;
            end
        else
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
            J_b_sm=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            J_tilde_a=[0; 0];
            J_tilde_b=[0; 0];
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            d=0;
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function file to calculate currents at each time step
function [J_a_sm,J_b_sm,J_c_sm,J_d_sm]=time_step_currents_14_a(D_tilde,
xi_1_plus,xi_1_minus,xi_2_plus,xi_2_minus,g,Cy,Fy,Fz,Q_tilde,
RY,knlY,knlZ,Br,r,filt)
        if D_tilde(1,1)*D_tilde(2,2)>0   %check for definiteness
        %(eigenvalue products)
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
```

```
    J_b_sm=[0; 0];
    J_c_sm=[0; 0];
    J_d_sm=[0; 0];
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_tilde_c=[0; 0];
    J_tilde_d=[0; 0];
    d=0;
elseif D_tilde(2,2)~=0   %this loop will use formulation when second
%eigenvalue is /neq 0
    xi_1=[1;sqrt(-D_tilde(1,1)/D_tilde(2,2))];
    if xi_1_plus'*g*xi_1_plus>0    %get solutions a and b, check for the
    %possible imaginary result and skip if necessary
        j_tilde_1_plus_plus=Cy*sqrt(xi_1_plus'*Q_tilde'*g*
        Q_tilde*xi_1_plus)/(xi_1_plus'*RY*xi_1_plus);
        j_tilde_1_minus_plus=-Cy*sqrt(xi_1_plus'*Q_tilde'*g*
        Q_tilde*xi_1_plus)/(xi_1_plus'*RY*xi_1_plus);
        j_tilde_2_plus_plus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
        Cy*sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)/
        (xi_1_plus'*RY*xi_1_plus);
        j_tilde_2_minus_plus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
        Cy*sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)/
        (xi_1_plus'*RY*xi_1_plus);
        d=determinant_calc(Q_tilde);
        %%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
        j_tilde_1_plus_plus_sm=sign(d)*Cy*sqrt([xi_1(1);sign(d*Fy)*
        xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);sign(d*Fy)*xi_1(2)])/([xi_1(1);
        sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);sign(d*Fy)*xi_1(2)]);
        j_tilde_1_minus_plus_sm=-sign(d)*Cy*sqrt([xi_1(1);sign(d*Fy)*
        xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);sign(d*Fy)*xi_1(2)])/([xi_1(1);
        sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);sign(d*Fy)*xi_1(2)]);
        j_tilde_2_plus_plus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*sign(Fy)*
        Cy*sqrt([xi_1(1);sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
sign(d*Fy)*xi_1(2)])/([xi_1(1);sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
sign(d*Fy)*xi_1(2)]);
j_tilde_2_minus_plus_sm=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*sign(Fy)*
Cy*sqrt([xi_1(1);sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
sign(d*Fy)*xi_1(2)])/([xi_1(1);sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
sign(d*Fy)*xi_1(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%


J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;
%plug back into magnetic force formulation to
%evaluate validity of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];


F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
[J_a(1); J_a(2)];
```

```
    F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
    [J_b(1); J_b(2)];
    %F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
    %[J_c(1); J_c(2)];
    %F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
    %[J_d(1); J_d(2)];

    %get force direction and compare to desired
    F_dir=(180/pi)*atan2(Fz,Fy);
    F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
    F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
    %F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
    %F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
    if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
    abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
        J_tilde_a=[0; 0];
        J_tilde_b=[0; 0];
        J_a=[0; 0];
        J_b=[0; 0];
        J_a_sm=[0; 0];
        J_b_sm=[0; 0];
        d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
if xi_1_minus'*g*xi_1_minus>0    %get solutions c and d, check for
```

```
%the possible imaginary result and skip if necessary
    j_tilde_1_plus_minus=Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*
    xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
    j_tilde_1_minus_minus=-Cy*sqrt(xi_1_minus'*Q_tilde'*g*
    Q_tilde*xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
    j_tilde_2_minus_minus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
    Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
    (xi_1_minus'*RY*xi_1_minus);
    j_tilde_2_plus_minus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*Cy*
    sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
    (xi_1_minus'*RY*xi_1_minus);
    d=determinant_calc(Q_tilde);
    %%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
    j_tilde_1_plus_minus_sm=sign(d*Fy)*Cy*sqrt([xi_1(1);
    -sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);-sign(d*Fy)*
    xi_1(2)])/([xi_1(1);-sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
    -sign(d*Fy)*xi_1(2)]);
    j_tilde_1_minus_minus_sm=-sign(d*Fy)*Cy*sqrt([xi_1(1);
    -sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);-sign(d*Fy)*
    xi_1(2)])/([xi_1(1);-sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
    -sign(d*Fy)*xi_1(2)]);
    j_tilde_2_minus_minus_sm=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
    Cy*sqrt([xi_1(1);-sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*
    [xi_1(1);-sign(d*Fy)*xi_1(2)])/([xi_1(1);-sign(d*Fy)*xi_1(2)]'*
    RY*[xi_1(1);-sign(d*Fy)*xi_1(2)]);
    j_tilde_2_plus_minus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*Cy*
    sqrt([xi_1(1);-sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
    -sign(d*Fy)*xi_1(2)])/([xi_1(1);-sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
    -sign(d*Fy)*xi_1(2)]);
    J_tilde_c_sm=[j_tilde_1_plus_minus_sm;j_tilde_2_minus_minus_sm];
    J_tilde_d_sm=[j_tilde_1_minus_minus_sm;j_tilde_2_plus_minus_sm];
    J_c_sm=Q_tilde*J_tilde_c_sm;
    J_d_sm=Q_tilde*J_tilde_d_sm;
```

```
%%%%%%%%%%%%%%%%%%%%%%%

J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;

%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
[J_c(1); J_c(2)];
F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
[J_d(1); J_d(2)];

%F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
%[J_a(1); J_a(2)];
%F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
%[J_b(1); J_b(2)];
F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
[J_c(1); J_c(2)];
F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
[J_d(1); J_d(2)];

%get force direction and compare to desired
F_dir=(180/pi)*atan2(Fz,Fy);
```

```
        %F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
        %F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
        F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
        F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
        if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
        abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            d=0;
        end
    else
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
elseif D_tilde(1,1)~=0   %this loop will use formulation when first
%eigenvalue is /neq 0
    xi_2=[sqrt(-D_tilde(2,2)/D_tilde(1,1));1];
    if xi_2_plus’*g*xi_2_plus>0    %get solutions a and b, check for the
    %possible imaginary result and skip if necessary
        j_tilde_2_plus_plus=Cy*sqrt(xi_2_plus’*Q_tilde’*g*Q_tilde*xi_2_plus)/
        (xi_2_plus’*RY*xi_2_plus);
        j_tilde_2_minus_plus=-Cy*sqrt(xi_2_plus’*Q_tilde’*g*
        Q_tilde*xi_2_plus)/(xi_2_plus’*RY*xi_2_plus);
        j_tilde_1_plus_plus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
```

```
sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*xi_2_plus)/
(xi_2_plus'*RY*xi_2_plus);
j_tilde_1_minus_plus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*xi_2_plus)/
(xi_2_plus'*RY*xi_2_plus);
d=determinant_calc(Q_tilde);
%%%%%%%%%%%%%%%%%%%%%%%get smooth currents now
j_tilde_2_plus_plus_sm=sign(Fy)*Cy*sqrt([sign(d*Fy)*xi_2(1);
xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d*Fy)*xi_2(1);xi_2(2)])/
([sign(d*Fy)*
xi_2(1);xi_2(2)]'*RY*[sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_2_minus_plus_sm=-sign(Fy)*Cy*sqrt([sign(d*Fy)*
xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d*Fy)*xi_2(1);xi_2(2)])/
([sign(d*Fy)*xi_2(1);xi_2(2)]'*RY*[sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_plus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*sign(d)*Cy*
sqrt([sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d*Fy)*
xi_2(1);
xi_2(2)])/([sign(d*Fy)*xi_2(1);xi_2(2)]'*RY*[sign(d*Fy)*xi_2(1);
xi_2(2)]);
j_tilde_1_minus_plus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*sign(d)*Cy*
sqrt([sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d*Fy)*
xi_2(1);xi_2(2)])/([sign(d*Fy)*xi_2(1);xi_2(2)]'*RY*[sign(d*Fy)*
xi_2(1);xi_2(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%
J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;
```

```
%plug back into magnetic force formulation to evaluate
%validity of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];

F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
[J_a(1); J_a(2)];
F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
[J_b(1); J_b(2)];
%F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
%[J_c(1); J_c(2)];
%F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
%[J_d(1); J_d(2)];

%get force direction and compare to desired
F_dir=(180/pi)*atan2(Fz,Fy);
F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
%F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
%F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
    J_tilde_a=[0; 0];
```

```
        J_tilde_b=[0; 0];
        J_a=[0; 0];
        J_b=[0; 0];
        J_a_sm=[0; 0];
        J_b_sm=[0; 0];
        d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
if xi_2_minus'*g*xi_2_minus>0    %get solutions c and d, check for the
%possible imaginary result and skip if necessary
    j_tilde_2_plus_minus=Cy*sqrt(xi_2_minus'*
    Q_tilde'*g*Q_tilde*xi_2_minus)/(xi_2_minus'*RY*xi_2_minus);
    j_tilde_2_minus_minus=-Cy*sqrt(xi_2_minus'*Q_tilde'*g*
    Q_tilde*xi_2_minus)/(xi_2_minus'*RY*xi_2_minus);
    j_tilde_1_minus_minus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
    sqrt(xi_2_minus'*Q_tilde'*g*Q_tilde*xi_2_minus)/(xi_2_minus'*
    RY*xi_2_minus);
    j_tilde_1_plus_minus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
    sqrt(xi_2_minus'*Q_tilde'*g*Q_tilde*xi_2_minus)/
    (xi_2_minus'*RY*xi_2_minus);
    d=determinant_calc(Q_tilde);
    %%%%%%%%%%%%%%%%%%%%%%get smooth trajectories now
    j_tilde_2_plus_minus_sm=Cy*sqrt([-sign(d*Fy)*xi_2(1);xi_2(2)]'*
    Q_tilde'*g*Q_tilde*[-sign(d*Fy)*xi_2(1);xi_2(2)])/([-sign(d*Fy)*
    xi_2(1);xi_2(2)]'*RY*[-sign(d*Fy)*xi_2(1);xi_2(2)]);
```

```
j_tilde_2_minus_minus_sm=-Cy*sqrt([-sign(d*Fy)*xi_2(1);
xi_2(2)]'*Q_tilde'*g*Q_tilde*[-sign(d*Fy)*xi_2(1);xi_2(2)])/
([-sign(d*Fy)*xi_2(1);xi_2(2)]'*RY*[-sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_1_minus_minus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
sign(d*Fy)*Cy*sqrt([-sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*
Q_tilde*[-sign(d*Fy)*xi_2(1);xi_2(2)])/([-sign(d*Fy)*xi_2(1);
xi_2(2)]'*RY*[-sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_minus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
sign(d*Fy)*Cy*sqrt([-sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*
Q_tilde*[-sign(d*Fy)*xi_2(1);xi_2(2)])/([-sign(d*Fy)*xi_2(1);
xi_2(2)]'*RY*[-sign(d*Fy)*xi_2(1);xi_2(2)]);
J_tilde_c_sm=[j_tilde_1_plus_minus_sm;j_tilde_2_minus_minus_sm];
J_tilde_d_sm=[j_tilde_1_minus_minus_sm;j_tilde_2_plus_minus_sm];
J_c_sm=Q_tilde*J_tilde_c_sm;
J_d_sm=Q_tilde*J_tilde_d_sm;
%%%%%%%%%%%%%%%%%%%%%%
J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;

%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
[J_c(1); J_c(2)];
```

```
    F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
    [J_d(1); J_d(2)];

    %F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
    %[J_a(1); J_a(2)];
    %F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
    %[J_b(1); J_b(2)];
    F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
    [J_c(1); J_c(2)];
    F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
    [J_d(1); J_d(2)];

    %get force direction and compare to desired
    F_dir=(180/pi)*atan2(Fz,Fy);
    %F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
    %F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
    F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
    F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
    if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
    abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
else
    J_tilde_c=[0; 0];
    J_tilde_d=[0; 0];
    J_c=[0; 0];
    J_d=[0; 0];
```

```
                J_c_sm=[0; 0];
                J_d_sm=[0; 0];
                d=0;
            end
        else
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
            J_b_sm=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            J_tilde_a=[0; 0];
            J_tilde_b=[0; 0];
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            d=0;
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function file to calculate currents at each time step
function [J_a_sm,J_b_sm,J_c_sm,J_d_sm]=time_step_currents_23_a(D_tilde,xi_1_plus,
xi_1_minus,xi_2_plus,xi_2_minus,g,Cy,Fy,Fz,Q_tilde,RY,knlY,knlZ,Br,r,filt)
        if D_tilde(1,1)*D_tilde(2,2)>0   %check for definiteness
        %(eigenvalue products)
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
            J_b_sm=[0; 0];
            J_c_sm=[0; 0];
```

```
    J_d_sm=[0; 0];
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_tilde_c=[0; 0];
    J_tilde_d=[0; 0];
    d=0;
elseif D_tilde(2,2)~=0   %this loop will use formulation when second
%eigenvalue is /neq 0
    xi_1=[1;sqrt(-D_tilde(1,1)/D_tilde(2,2))];
    if xi_1_plus’*g*xi_1_plus>0    %get solutions a and b, check for the
    %possible imaginary result and skip if necessary
        j_tilde_1_plus_plus=Cy*sqrt(xi_1_plus’*Q_tilde’*g*Q_tilde*
        xi_1_plus)/(xi_1_plus’*RY*xi_1_plus);
        j_tilde_1_minus_plus=-Cy*sqrt(xi_1_plus’*Q_tilde’*g*Q_tilde*
        xi_1_plus)/(xi_1_plus’*RY*xi_1_plus);
        j_tilde_2_plus_plus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*Cy*
        sqrt(xi_1_plus’*Q_tilde’*g*Q_tilde*xi_1_plus)/
        (xi_1_plus’*RY*xi_1_plus);
        j_tilde_2_minus_plus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*Cy*
        sqrt(xi_1_plus’*Q_tilde’*g*Q_tilde*xi_1_plus)/(xi_1_plus’*
        RY*xi_1_plus);
        d=determinant_calc(Q_tilde);
        %%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
        j_tilde_1_plus_plus_sm=sign(Fy)*Cy*sqrt([xi_1(1);sign(d*Fy)*
        xi_1(2)]’*Q_tilde’*g*Q_tilde*[xi_1(1);sign(d*Fy)*xi_1(2)])/([xi_1(1);
        sign(d*Fy)*xi_1(2)]’*RY*[xi_1(1);sign(d*Fy)*xi_1(2)]);
        j_tilde_1_minus_plus_sm=-sign(Fy)*Cy*sqrt([xi_1(1);sign(d*Fy)*
        xi_1(2)]’*Q_tilde’*g*Q_tilde*[xi_1(1);sign(d*Fy)*xi_1(2)])/([xi_1(1);
        sign(d*Fy)*xi_1(2)]’*RY*[xi_1(1);sign(d*Fy)*xi_1(2)]);
        j_tilde_2_plus_plus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*sign(d)*
        Cy*sqrt([xi_1(1);sign(d*Fy)*xi_1(2)]’*Q_tilde’*g*Q_tilde*[xi_1(1);
        sign(d*Fy)*xi_1(2)])/([xi_1(1);sign(d*Fy)*xi_1(2)]’*RY*[xi_1(1);
        sign(d*Fy)*xi_1(2)]);
```

```
j_tilde_2_minus_plus_sm=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*sign(d)*
Cy*sqrt([xi_1(1);sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
sign(d*Fy)*xi_1(2)])/([xi_1(1);sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
sign(d*Fy)*xi_1(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%

J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;
%plug back into magnetic force formulation to evaluate
%validity of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];

F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
[J_a(1); J_a(2)];
F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
[J_b(1); J_b(2)];
```

```
    %F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
    %[J_c(1); J_c(2)];
    %F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
    %[J_d(1); J_d(2)];

    %get force direction and compare to desired
    F_dir=(180/pi)*atan2(Fz,Fy);
    F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
    F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
    %F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
    %F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
    if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
    abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
        J_tilde_a=[0; 0];
        J_tilde_b=[0; 0];
        J_a=[0; 0];
        J_b=[0; 0];
        J_a_sm=[0; 0];
        J_b_sm=[0; 0];
        d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
if xi_1_minus'*g*xi_1_minus>0    %get solutions c and d, check for the
%possible imaginary result and skip if necessary
    j_tilde_1_plus_minus=Cy*sqrt(xi_1_minus'*Q_tilde'*g*
Q_tilde*xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
j_tilde_1_minus_minus=-Cy*sqrt(xi_1_minus'*Q_tilde'*g*
Q_tilde*xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
j_tilde_2_minus_minus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
(xi_1_minus'*RY*xi_1_minus);
j_tilde_2_plus_minus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
(xi_1_minus'*RY*xi_1_minus);
d=determinant_calc(Q_tilde);
%%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
j_tilde_1_plus_minus_sm=Cy*sqrt([xi_1(1);-sign(d*Fy)*xi_1(2)]'*
Q_tilde'*g*Q_tilde*[xi_1(1);-sign(d*Fy)*xi_1(2)])/([xi_1(1);
-sign(d*Fy)*
xi_1(2)]'*RY*[xi_1(1);-sign(d*Fy)*xi_1(2)]);
j_tilde_1_minus_minus_sm=-Cy*sqrt([xi_1(1);-sign(d*Fy)*xi_1(2)]'*
Q_tilde'*g*Q_tilde*[xi_1(1);-sign(d*Fy)*xi_1(2)])/([xi_1(1);
-sign(d*Fy)*
xi_1(2)]'*RY*[xi_1(1);-sign(d*Fy)*xi_1(2)]);
j_tilde_2_minus_minus_sm=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
sign(d*Fy)*
Cy*sqrt([xi_1(1);-sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
-sign(d*Fy)*xi_1(2)])/([xi_1(1);-sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
-sign(d*Fy)*xi_1(2)]);
j_tilde_2_plus_minus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*sign(d*Fy)*
Cy*sqrt([xi_1(1);-sign(d*Fy)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
-sign(d*Fy)*xi_1(2)])/([xi_1(1);-sign(d*Fy)*xi_1(2)]'*RY*[xi_1(1);
-sign(d*Fy)*xi_1(2)]);
J_tilde_c_sm=[j_tilde_1_plus_minus_sm;j_tilde_2_minus_minus_sm];
J_tilde_d_sm=[j_tilde_1_minus_minus_sm;j_tilde_2_plus_minus_sm];
J_c_sm=Q_tilde*J_tilde_c_sm;
J_d_sm=Q_tilde*J_tilde_d_sm;
%%%%%%%%%%%%%%%%%%%%%%%
```

```
J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;

%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
[J_c(1); J_c(2)];
F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
[J_d(1); J_d(2)];

%F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
%[J_a(1); J_a(2)];
%F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
%[J_b(1); J_b(2)];
F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
[J_c(1); J_c(2)];
F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
[J_d(1); J_d(2)];

%get force direction and compare to desired
F_dir=(180/pi)*atan2(Fz,Fy);
%F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
```

```
        %F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
        F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
        F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
        if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
        abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            d=0;
        end
    else
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
elseif D_tilde(1,1)~=0   %this loop will use formulation when first
%eigenvalue is /neq 0
    xi_2=[sqrt(-D_tilde(2,2)/D_tilde(1,1));1];
    if xi_2_plus'*g*xi_2_plus>0    %get solutions a and b, check for the
    %possible imaginary result and skip if necessary
        j_tilde_2_plus_plus=Cy*sqrt(xi_2_plus'*Q_tilde'*g*
        Q_tilde*xi_2_plus)/(xi_2_plus'*RY*xi_2_plus);
        j_tilde_2_minus_plus=-Cy*sqrt(xi_2_plus'*Q_tilde'*g*
        Q_tilde*xi_2_plus)/(xi_2_plus'*RY*xi_2_plus);
        j_tilde_1_plus_plus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
        Cy*sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*xi_2_plus)/
```

```
(xi_2_plus'*RY*xi_2_plus);
j_tilde_1_minus_plus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
Cy*sqrt(xi_2_plus'*Q_tilde'*g*Q_tilde*xi_2_plus)/
(xi_2_plus'*RY*xi_2_plus);
d=determinant_calc(Q_tilde);
%%%%%%%%%%%%%%%%%%%%%%%get smooth currents now
j_tilde_2_plus_plus_sm=sign(d)*Cy*sqrt([sign(d*Fy)*xi_2(1);
xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d*Fy)*xi_2(1);xi_2(2)])/
([sign(d*Fy)*xi_2(1);xi_2(2)]'*RY*[sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_2_minus_plus_sm=-sign(d)*Cy*sqrt([sign(d*Fy)*xi_2(1);
xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d*Fy)*xi_2(1);xi_2(2)])/
([sign(d*Fy)*xi_2(1);xi_2(2)]'*RY*[sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_plus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
sign(Fy)*Cy*sqrt([sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*
Q_tilde*[sign(d*Fy)*xi_2(1);xi_2(2)])/([sign(d*Fy)*xi_2(1);
xi_2(2)]'*RY*[sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_1_minus_plus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
sign(Fy)*Cy*sqrt([sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*
Q_tilde*[sign(d*Fy)*xi_2(1);xi_2(2)])/([sign(d*Fy)*xi_2(1);
xi_2(2)]'*RY*[sign(d*Fy)*xi_2(1);xi_2(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%
J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;


%plug back into magnetic force formulation to evaluate
%validity of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
```

```
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];

F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
[J_a(1); J_a(2)];
F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
[J_b(1); J_b(2)];
%F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
%[J_c(1); J_c(2)];
%F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
%[J_d(1); J_d(2)];

%get force direction and compare to desired
F_dir=(180/pi)*atan2(Fz,Fy);
F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
%F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
%F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
```

```
        J_a_sm=[0; 0];
        J_b_sm=[0; 0];
        d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
if xi_2_minus'*g*xi_2_minus>0    %get solutions c and d, check for
%the possible imaginary result and skip if necessary
    j_tilde_2_plus_minus=Cy*sqrt(xi_2_minus'*Q_tilde'*g*
    Q_tilde*xi_2_minus)/(xi_2_minus'*RY*xi_2_minus);
    j_tilde_2_minus_minus=-Cy*sqrt(xi_2_minus'*Q_tilde'*g*
    Q_tilde*xi_2_minus)/(xi_2_minus'*RY*xi_2_minus);
    j_tilde_1_minus_minus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
    Cy*sqrt(xi_2_minus'*Q_tilde'*g*Q_tilde*xi_2_minus)/
    (xi_2_minus'*RY*xi_2_minus);
    j_tilde_1_plus_minus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
    Cy*sqrt(xi_2_minus'*Q_tilde'*g*Q_tilde*xi_2_minus)/
    (xi_2_minus'*RY*xi_2_minus);
    d=determinant_calc(Q_tilde);
    %%%%%%%%%%%%%%%%%%%%%get smooth trajectories now
    j_tilde_2_plus_minus_sm=sign(d*Fy)*Cy*
    sqrt([-sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*
    [-sign(d*Fy)*xi_2(1);xi_2(2)])/([-sign(d*Fy)*xi_2(1);xi_2(2)]'*
    RY*[-sign(d*Fy)*xi_2(1);xi_2(2)]);
    j_tilde_2_minus_minus_sm=-sign(d*Fy)*Cy*sqrt([-sign(d*Fy)*
    xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[-sign(d*Fy)*xi_2(1);
xi_2(2)])/([-sign(d*Fy)*xi_2(1);xi_2(2)]'*RY*[-sign(d*Fy)*
xi_2(1);xi_2(2)]);
j_tilde_1_minus_minus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
Cy*sqrt([-sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*
Q_tilde*[-sign(d*Fy)*xi_2(1);xi_2(2)])/([-sign(d*Fy)*xi_2(1);
xi_2(2)]'*RY*[-sign(d*Fy)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_minus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
Cy*sqrt([-sign(d*Fy)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*
[-sign(d*Fy)*xi_2(1);xi_2(2)])/([-sign(d*Fy)*xi_2(1);
xi_2(2)]'*RY*[-sign(d*Fy)*xi_2(1);xi_2(2)]);
J_tilde_c_sm=[j_tilde_1_plus_minus_sm;j_tilde_2_minus_minus_sm];
J_tilde_d_sm=[j_tilde_1_minus_minus_sm;j_tilde_2_plus_minus_sm];
J_c_sm=Q_tilde*J_tilde_c_sm;
J_d_sm=Q_tilde*J_tilde_d_sm;
%%%%%%%%%%%%%%%%%%%%%%
J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;


%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
[J_c(1); J_c(2)];
F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
```

```
    [J_d(1); J_d(2)];

    %F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
    %[J_a(1); J_a(2)];
    %F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
    %[J_b(1); J_b(2)];
    F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
    [J_c(1); J_c(2)];
    F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
    [J_d(1); J_d(2)];

    %get force direction and compare to desired
    F_dir=(180/pi)*atan2(Fz,Fy);
    %F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
    %F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
    F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
    F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
    if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
    abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
else
    J_tilde_c=[0; 0];
    J_tilde_d=[0; 0];
    J_c=[0; 0];
    J_d=[0; 0];
    J_c_sm=[0; 0];
```

```
                J_d_sm=[0; 0];
                d=0;
            end
        else
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
            J_b_sm=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            J_tilde_a=[0; 0];
            J_tilde_b=[0; 0];
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            d=0;
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function file to calculate currents at each time step
function [J_a_sm,J_b_sm,J_c_sm,J_d_sm]=time_step_currents_24_a(D_tilde,
xi_1_plus,xi_1_minus,xi_2_plus,xi_2_minus,g,Cy,Fy,Fz,Q_tilde,
RY,knlY,knlZ,Br,r,filt)
        if D_tilde(1,1)*D_tilde(2,2)>0   %check for definiteness
        %(eigenvalue products)
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
            J_b_sm=[0; 0];
            J_c_sm=[0; 0];
```

```
    J_d_sm=[0; 0];
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_tilde_c=[0; 0];
    J_tilde_d=[0; 0];
    d=0;
elseif D_tilde(2,2)~=0   %this loop will use formulation when second
%eigenvalue is /neq 0
    xi_1=[1;sqrt(-D_tilde(1,1)/D_tilde(2,2))];
    if xi_1_plus'*g*xi_1_plus>0    %get solutions a and b, check for the
    %possible imaginary result and skip if necessary
        j_tilde_1_plus_plus=Cy*sqrt(xi_1_plus'*Q_tilde'*g*
        Q_tilde*xi_1_plus)/(xi_1_plus'*RY*xi_1_plus);
        j_tilde_1_minus_plus=-Cy*sqrt(xi_1_plus'*Q_tilde'*g*
        Q_tilde*xi_1_plus)/(xi_1_plus'*RY*xi_1_plus);
        j_tilde_2_plus_plus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
        Cy*sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)/
        (xi_1_plus'*RY*xi_1_plus);
        j_tilde_2_minus_plus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
        Cy*sqrt(xi_1_plus'*Q_tilde'*g*Q_tilde*xi_1_plus)/
        (xi_1_plus'*RY*xi_1_plus);
        d=determinant_calc(Q_tilde);
        %%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
        j_tilde_1_plus_plus_sm=Cy*sqrt([xi_1(1);sign(d)*xi_1(2)]'*
        Q_tilde'*g*Q_tilde*[xi_1(1);sign(d)*xi_1(2)])/([xi_1(1);sign(d)*
        xi_1(2)]'*RY*[xi_1(1);sign(d)*xi_1(2)]);
        j_tilde_1_minus_plus_sm=-Cy*sqrt([xi_1(1);sign(d)*xi_1(2)]'*
        Q_tilde'*g*Q_tilde*[xi_1(1);sign(d)*xi_1(2)])/([xi_1(1);sign(d)*
        xi_1(2)]'*RY*[xi_1(1);sign(d)*xi_1(2)]);
        j_tilde_2_plus_plus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*sign(d)*
        Cy*sqrt([xi_1(1);sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*[xi_1(1);
        sign(d)*xi_1(2)])/([xi_1(1);sign(d)*xi_1(2)]'*RY*[xi_1(1);sign(d)*
        xi_1(2)]);
```

```
j_tilde_2_minus_plus_sm=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
sign(d)*Cy*sqrt([xi_1(1);sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*
[xi_1(1);sign(d)*xi_1(2)])/([xi_1(1);sign(d)*xi_1(2)]'*RY*[xi_1(1);
sign(d)*xi_1(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%


J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;
%plug back into magnetic force formulation to evaluate
%validity of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];


F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
[J_a(1); J_a(2)];
F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
[J_b(1); J_b(2)];
```

```
    %F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
    %[J_c(1); J_c(2)];
    %F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
    %[J_d(1); J_d(2)];

    %get force direction and compare to desired
    F_dir=(180/pi)*atan2(Fz,Fy);
    F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
    F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
    %F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
    %F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
    if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
    abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
        J_tilde_a=[0; 0];
        J_tilde_b=[0; 0];
        J_a=[0; 0];
        J_b=[0; 0];
        J_a_sm=[0; 0];
        J_b_sm=[0; 0];
        d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
if xi_1_minus'*g*xi_1_minus>0    %get solutions c and d, check for the
%possible imaginary result and skip if necessary
    j_tilde_1_plus_minus=Cy*sqrt(xi_1_minus'*Q_tilde'*g*
Q_tilde*xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
j_tilde_1_minus_minus=-Cy*sqrt(xi_1_minus'*Q_tilde'*g*
Q_tilde*xi_1_minus)/(xi_1_minus'*RY*xi_1_minus);
j_tilde_2_minus_minus=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
(xi_1_minus'*RY*xi_1_minus);
j_tilde_2_plus_minus=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
Cy*sqrt(xi_1_minus'*Q_tilde'*g*Q_tilde*xi_1_minus)/
(xi_1_minus'*RY*xi_1_minus);
d=determinant_calc(Q_tilde);
%%%%%%%%%%%%%%%%%%%%%%%make smooth trajectories now
j_tilde_1_plus_minus_sm=Cy*sqrt([xi_1(1);-sign(d)*xi_1(2)]'*
Q_tilde'*g*Q_tilde*[xi_1(1);-sign(d)*xi_1(2)])/([xi_1(1);-sign(d)*
xi_1(2)]'*RY*[xi_1(1);-sign(d)*xi_1(2)]);
j_tilde_1_minus_minus_sm=-Cy*sqrt([xi_1(1);-sign(d)*xi_1(2)]'*
Q_tilde'*g*Q_tilde*[xi_1(1);-sign(d)*xi_1(2)])/([xi_1(1);-sign(d)*
xi_1(2)]'*RY*[xi_1(1);-sign(d)*xi_1(2)]);
j_tilde_2_minus_minus_sm=-sqrt(-D_tilde(1,1)/D_tilde(2,2))*
sign(d)*Cy*sqrt([xi_1(1);-sign(d)*xi_1(2)]'*Q_tilde'*g*Q_tilde*
[xi_1(1);-sign(d)*xi_1(2)])/([xi_1(1);-sign(d)*xi_1(2)]'*RY*
[xi_1(1);-sign(d)*xi_1(2)]);
j_tilde_2_plus_minus_sm=sqrt(-D_tilde(1,1)/D_tilde(2,2))*
sign(d)*Cy*sqrt([xi_1(1);-sign(d)*xi_1(2)]'*Q_tilde'*g*
Q_tilde*[xi_1(1);-sign(d)*xi_1(2)])/([xi_1(1);-sign(d)*xi_1(2)]'*
RY*[xi_1(1);-sign(d)*xi_1(2)]);
J_tilde_c_sm=[j_tilde_1_plus_minus_sm;j_tilde_2_minus_minus_sm];
J_tilde_d_sm=[j_tilde_1_minus_minus_sm;j_tilde_2_plus_minus_sm];
J_c_sm=Q_tilde*J_tilde_c_sm;
J_d_sm=Q_tilde*J_tilde_d_sm;
%%%%%%%%%%%%%%%%%%%%%%%

J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
```

```
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;

%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
[J_c(1); J_c(2)];
F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
[J_d(1); J_d(2)];

%F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
%[J_a(1); J_a(2)];
%F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
%[J_b(1); J_b(2)];
F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
[J_c(1); J_c(2)];
F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
[J_d(1); J_d(2)];

%get force direction and compare to desired
F_dir=(180/pi)*atan2(Fz,Fy);
%F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
%F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
```

```
        if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
        abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            d=0;
        end
    else
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
elseif D_tilde(1,1)~=0   %this loop will use formulation when first
%eigenvalue is /neq 0
    xi_2=[sqrt(-D_tilde(2,2)/D_tilde(1,1));1];
    if xi_2_plus’*g*xi_2_plus>0    %get solutions a and b, check for the
    %possible imaginary result and skip if necessary
        j_tilde_2_plus_plus=Cy*sqrt(xi_2_plus’*Q_tilde’*g*
        Q_tilde*xi_2_plus)/(xi_2_plus’*RY*xi_2_plus);
        j_tilde_2_minus_plus=-Cy*sqrt(xi_2_plus’*Q_tilde’*g*
        Q_tilde*xi_2_plus)/(xi_2_plus’*RY*xi_2_plus);
        j_tilde_1_plus_plus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
        Cy*sqrt(xi_2_plus’*Q_tilde’*g*Q_tilde*xi_2_plus)/
        (xi_2_plus’*RY*xi_2_plus);
        j_tilde_1_minus_plus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
        Cy*sqrt(xi_2_plus’*Q_tilde’*g*Q_tilde*xi_2_plus)/
```

```
(xi_2_plus'*RY*xi_2_plus);
d=determinant_calc(Q_tilde);
%%%%%%%%%%%%%%%%%%%%%%%get smooth currents now
j_tilde_2_plus_plus_sm=sign(d)*Cy*sqrt([sign(d)*xi_2(1);
xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d)*xi_2(1);xi_2(2)])/([sign(d)*
xi_2(1);xi_2(2)]'*RY*[sign(d)*xi_2(1);xi_2(2)]);
j_tilde_2_minus_plus_sm=-sign(d)*Cy*sqrt([sign(d)*xi_2(1);
xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d)*xi_2(1);xi_2(2)])/([sign(d)*
xi_2(1);xi_2(2)]'*RY*[sign(d)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_plus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
sqrt([sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d)*
xi_2(1);xi_2(2)])/([sign(d)*xi_2(1);xi_2(2)]'*RY*[sign(d)*
xi_2(1);xi_2(2)]);
j_tilde_1_minus_plus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*Cy*
sqrt([sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*[sign(d)*
xi_2(1);xi_2(2)])/([sign(d)*xi_2(1);xi_2(2)]'*RY*[sign(d)*
xi_2(1);xi_2(2)]);
J_tilde_a_sm=[j_tilde_1_plus_plus_sm;j_tilde_2_plus_plus_sm];
J_tilde_b_sm=[j_tilde_1_minus_plus_sm;j_tilde_2_minus_plus_sm];
J_a_sm=Q_tilde*J_tilde_a_sm;
J_b_sm=Q_tilde*J_tilde_b_sm;
%%%%%%%%%%%%%%%%%%%%%%%
J_tilde_a=[j_tilde_1_plus_plus;j_tilde_2_plus_plus];
J_tilde_b=[j_tilde_1_minus_plus;j_tilde_2_minus_plus];
J_a=Q_tilde*J_tilde_a;
J_b=Q_tilde*J_tilde_b;

%plug back into magnetic force formulation to evaluate
%validity of solution
H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
%H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
%H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
```

```
F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
[J_a(1); J_a(2)];
F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
[J_b(1); J_b(2)];
%F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
%[J_c(1); J_c(2)];
%F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
%[J_d(1); J_d(2)];

F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
[J_a(1); J_a(2)];
F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
[J_b(1); J_b(2)];
%F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
%[J_c(1); J_c(2)];
%F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
%[J_d(1); J_d(2)];

%get force direction and compare to desired
F_dir=(180/pi)*atan2(Fz,Fy);
F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
%F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
%F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
if (filt~=0) && (abs((F_dir-F_mag_dir_a)/F_dir)>=0.01 ||
abs((F_dir-F_mag_dir_b)/F_dir)>=0.01)
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
    end
else
    J_tilde_a=[0; 0];
    J_tilde_b=[0; 0];
    J_a=[0; 0];
    J_b=[0; 0];
    J_a_sm=[0; 0];
    J_b_sm=[0; 0];
    d=0;
end
if xi_2_minus’*g*xi_2_minus>0    %get solutions c and d, check for the
%possible imaginary result and skip if necessary
    j_tilde_2_plus_minus=Cy*sqrt(xi_2_minus’*Q_tilde’*g*
    Q_tilde*xi_2_minus)/(xi_2_minus’*RY*xi_2_minus);
    j_tilde_2_minus_minus=-Cy*sqrt(xi_2_minus’*Q_tilde’*g*
    Q_tilde*xi_2_minus)/(xi_2_minus’*RY*xi_2_minus);
    j_tilde_1_minus_minus=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
    Cy*sqrt(xi_2_minus’*Q_tilde’*g*Q_tilde*xi_2_minus)/
    (xi_2_minus’*RY*xi_2_minus);
    j_tilde_1_plus_minus=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
    Cy*sqrt(xi_2_minus’*Q_tilde’*g*Q_tilde*xi_2_minus)/
    (xi_2_minus’*RY*xi_2_minus);
    d=determinant_calc(Q_tilde);
    %%%%%%%%%%%%%%%%%%%%%%get smooth trajectories now
    j_tilde_2_plus_minus_sm=sign(d)*Cy*sqrt([-sign(d)*xi_2(1);
    xi_2(2)]’*Q_tilde’*g*Q_tilde*[-sign(d)*xi_2(1);xi_2(2)])/
    ([-sign(d)*xi_2(1);xi_2(2)]’*RY*[-sign(d)*xi_2(1);xi_2(2)]);
    j_tilde_2_minus_minus_sm=-sign(d)*Cy*sqrt([-sign(d)*
    xi_2(1);xi_2(2)]’*Q_tilde’*g*Q_tilde*[-sign(d)*xi_2(1);
    xi_2(2)])/([-sign(d)*xi_2(1);xi_2(2)]’*RY*[-sign(d)*xi_2(1);
    xi_2(2)]);
    j_tilde_1_minus_minus_sm=-sqrt(-D_tilde(2,2)/D_tilde(1,1))*
    Cy*sqrt([-sign(d)*xi_2(1);xi_2(2)]’*Q_tilde’*g*Q_tilde*
[-sign(d)*xi_2(1);xi_2(2)])/([-sign(d)*xi_2(1);xi_2(2)]'*RY*
[-sign(d)*xi_2(1);xi_2(2)]);
j_tilde_1_plus_minus_sm=sqrt(-D_tilde(2,2)/D_tilde(1,1))*
Cy*sqrt([-sign(d)*xi_2(1);xi_2(2)]'*Q_tilde'*g*Q_tilde*
[-sign(d)*xi_2(1);xi_2(2)])/([-sign(d)*xi_2(1);xi_2(2)]'*RY*
[-sign(d)*xi_2(1);xi_2(2)]);
J_tilde_c_sm=[j_tilde_1_plus_minus_sm;j_tilde_2_minus_minus_sm];
J_tilde_d_sm=[j_tilde_1_minus_minus_sm;j_tilde_2_plus_minus_sm];
J_c_sm=Q_tilde*J_tilde_c_sm;
J_d_sm=Q_tilde*J_tilde_d_sm;
%%%%%%%%%%%%%%%%%%%%%%
J_tilde_c=[j_tilde_1_plus_minus;j_tilde_2_minus_minus];
J_tilde_d=[j_tilde_1_minus_minus;j_tilde_2_plus_minus];
J_c=Q_tilde*J_tilde_c;
J_d=Q_tilde*J_tilde_d;

%plug back into magnetic force formulation to evaluate
%validity of solution
%H_a=sqrt([J_a(1) J_a(2)]*g*[J_a(1); J_a(2)]);
%H_b=sqrt([J_b(1) J_b(2)]*g*[J_b(1); J_b(2)]);
H_c=sqrt([J_c(1) J_c(2)]*g*[J_c(1); J_c(2)]);
H_d=sqrt([J_d(1) J_d(2)]*g*[J_d(1); J_d(2)]);
%F_mag_Y_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlY*
%[J_a(1); J_a(2)];
%F_mag_Y_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlY*
%[J_b(1); J_b(2)];
F_mag_Y_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlY*
[J_c(1); J_c(2)];
F_mag_Y_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlY*
[J_d(1); J_d(2)];

%F_mag_Z_a=2*pi*(r^3)*Br*(1/H_a)*[J_a(1) J_a(2)]*knlZ*
%[J_a(1); J_a(2)];
```

```
        %F_mag_Z_b=2*pi*(r^3)*Br*(1/H_b)*[J_b(1) J_b(2)]*knlZ*
        %[J_b(1); J_b(2)];
        F_mag_Z_c=2*pi*(r^3)*Br*(1/H_c)*[J_c(1) J_c(2)]*knlZ*
        [J_c(1); J_c(2)];
        F_mag_Z_d=2*pi*(r^3)*Br*(1/H_d)*[J_d(1) J_d(2)]*knlZ*
        [J_d(1); J_d(2)];

        %get force direction and compare to desired
        F_dir=(180/pi)*atan2(Fz,Fy);
        %F_mag_dir_a=(180/pi)*atan2(F_mag_Z_a,F_mag_Y_a);
        %F_mag_dir_b=(180/pi)*atan2(F_mag_Z_b,F_mag_Y_b);
        F_mag_dir_c=(180/pi)*atan2(F_mag_Z_c,F_mag_Y_c);
        F_mag_dir_d=(180/pi)*atan2(F_mag_Z_d,F_mag_Y_d);
        if (filt~=0) && (abs((F_dir-F_mag_dir_c)/F_dir)>=0.01 ||
        abs((F_dir-F_mag_dir_d)/F_dir)>=0.01)
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            d=0;
        end
    else
        J_tilde_c=[0; 0];
        J_tilde_d=[0; 0];
        J_c=[0; 0];
        J_d=[0; 0];
        J_c_sm=[0; 0];
        J_d_sm=[0; 0];
        d=0;
    end
else
```

```
            J_a=[0; 0];
            J_b=[0; 0];
            J_c=[0; 0];
            J_d=[0; 0];
            J_a_sm=[0; 0];
            J_b_sm=[0; 0];
            J_c_sm=[0; 0];
            J_d_sm=[0; 0];
            J_tilde_a=[0; 0];
            J_tilde_b=[0; 0];
            J_tilde_c=[0; 0];
            J_tilde_d=[0; 0];
            d=0;
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function determinant=determinant_calc(A)
%function for calculating the determinant of a 2x2 matrix
determinant=A(1,1)*A(2,2)-A(1,2)*A(2,1);
```

C.3.4 Current selector embedded function.

(closed_loop→Controller→inverse_current_calculation→selector)

```
function J_out = selector(J1,J2,J3,J4)
% This block supports an embeddable subset of the MATLAB language.
% See the help menu for details.
N1=sqrt(J1(1)^2+J1(2)^2+J1(3)^2+J1(4)^2);
if N1~=0
    S1=1;
else
    S1=0;
end
N2=sqrt(J2(1)^2+J2(2)^2+J2(3)^2+J2(4)^2);
```

```
if N2~=0
    S2=1;
else
    S2=0;
end
N3=sqrt(J3(1)^2+J3(2)^2+J3(3)^2+J3(4)^2);
if N3~=0
    S3=1;
else
    S3=0;
end
N4=sqrt(J4(1)^2+J4(2)^2+J4(3)^2+J4(4)^2);
if N4~=0
    S4=1;
else
    S4=0;
end

S=[S1 S2 S3 S4];

if S1==1 && S2==0 && S3==0 && S4==0
    J_out=J1;
elseif S1==0 && S2==1 && S3==0 && S4==0
    J_out=J2;
elseif S1==0 && S2==0 && S3==1 && S4==0
    J_out=J3;
elseif S1==0 && S2==0 && S3==0 && S4==1
    J_out=J4;
elseif S1==1 && S2==1 && S3==0 && S4==0
    if N1<=N2
        J_out=J1;
    else
        J_out=J2;
    end
elseif S1==1 && S2==0 && S3==1 && S4==0
    if N1<=N3
        J_out=J1;
    else
        J_out=J3;
    end
elseif S1==1 && S2==0 && S3==0 && S4==1
    if N1<=N4
        J_out=J1;
    else
        J_out=J4;
    end
elseif S1==0 && S2==1 && S3==1 && S4==0
    if N2<=N3
        J_out=J2;
    else
        J_out=J3;
    end
elseif S1==0 && S2==1 && S3==0 && S4==1
    if N2<=N4
        J_out=J2;
    else
        J_out=J4;
    end
elseif S1==0 && S2==0 && S3==1 && S4==1
    if N3<=N4
        J_out=J3;
    else
        J_out=J4;
    end
else
    J_out=[0; 0; 0; 0];
```

```
end
```

C.3.5 K matrices embedded function.

(closed_loop→plant_model→magnetic_force→K_matrices)

```
function [knlX,knlY,knlZ,G] = k_matrices(Y,Z,Gx1_int, Gx2_int, Gx3_int,
Gx4_int, Gy1_int, Gy2_int, Gy3_int, Gy4_int, Gz1_int, Gz2_int, Gz3_int,
Gz4_int, dGx_dx1_int, dGx_dx2_int, dGx_dx3_int, dGx_dx4_int,
dGx_dy1_int, dGx_dy2_int, dGx_dy3_int, dGx_dy4_int, dGx_dz1_int,
dGx_dz2_int, dGx_dz3_int, dGx_dz4_int, dGy_dx1_int, dGy_dx2_int,
dGy_dx3_int, dGy_dx4_int, dGy_dy1_int, dGy_dy2_int, dGy_dy3_int,
dGy_dy4_int, dGy_dz1_int, dGy_dz2_int, dGy_dz3_int, dGy_dz4_int,
dGz_dx1_int, dGz_dx2_int, dGz_dx3_int, dGz_dx4_int, dGz_dy1_int,
dGz_dy2_int, dGz_dy3_int, dGz_dy4_int, dGz_dz1_int, dGz_dz2_int,
dGz_dz3_int, dGz_dz4_int, R_1, R_2, R_3, R_4)
%get m and n of point
    y_p_inch=Y/0.0254;
    z_p_inch=Z/0.0254;
    n_p=(y_p_inch+1)/0.05+1;
    m_p=(z_p_inch+1)/0.05+1;
    %put new interpolation technqnique here
    Gx1=interpolation(n_p,m_p,Gx1_int);
    Gy1=interpolation(n_p,m_p,Gy1_int);
    Gz1=interpolation(n_p,m_p,Gz1_int);
    dGx_dx1=interpolation(n_p,m_p,dGx_dx1_int);
    dGx_dy1=interpolation(n_p,m_p,dGx_dy1_int);
    dGx_dz1=interpolation(n_p,m_p,dGx_dz1_int);
    dGy_dx1=interpolation(n_p,m_p,dGy_dx1_int);
    dGy_dy1=interpolation(n_p,m_p,dGy_dy1_int);
    dGy_dz1=interpolation(n_p,m_p,dGy_dz1_int);
    dGz_dx1=interpolation(n_p,m_p,dGz_dx1_int);
    dGz_dy1=interpolation(n_p,m_p,dGz_dy1_int);
    dGz_dz1=interpolation(n_p,m_p,dGz_dz1_int);
```

```
Gx2=interpolation(n_p,m_p,Gx2_int);
Gy2=interpolation(n_p,m_p,Gy2_int);
Gz2=interpolation(n_p,m_p,Gz2_int);
dGx_dx2=interpolation(n_p,m_p,dGx_dx2_int);
dGx_dy2=interpolation(n_p,m_p,dGx_dy2_int);
dGx_dz2=interpolation(n_p,m_p,dGx_dz2_int);
dGy_dx2=interpolation(n_p,m_p,dGy_dx2_int);
dGy_dy2=interpolation(n_p,m_p,dGy_dy2_int);
dGy_dz2=interpolation(n_p,m_p,dGy_dz2_int);
dGz_dx2=interpolation(n_p,m_p,dGz_dx2_int);
dGz_dy2=interpolation(n_p,m_p,dGz_dy2_int);
dGz_dz2=interpolation(n_p,m_p,dGz_dz2_int);

Gx3=interpolation(n_p,m_p,Gx3_int);
Gy3=interpolation(n_p,m_p,Gy3_int);
Gz3=interpolation(n_p,m_p,Gz3_int);
dGx_dx3=interpolation(n_p,m_p,dGx_dx3_int);
dGx_dy3=interpolation(n_p,m_p,dGx_dy3_int);
dGx_dz3=interpolation(n_p,m_p,dGx_dz3_int);
dGy_dx3=interpolation(n_p,m_p,dGy_dx3_int);
dGy_dy3=interpolation(n_p,m_p,dGy_dy3_int);
dGy_dz3=interpolation(n_p,m_p,dGy_dz3_int);
dGz_dx3=interpolation(n_p,m_p,dGz_dx3_int);
dGz_dy3=interpolation(n_p,m_p,dGz_dy3_int);
dGz_dz3=interpolation(n_p,m_p,dGz_dz3_int);

Gx4=interpolation(n_p,m_p,Gx4_int);
Gy4=interpolation(n_p,m_p,Gy4_int);
Gz4=interpolation(n_p,m_p,Gz4_int);
dGx_dx4=interpolation(n_p,m_p,dGx_dx4_int);
dGx_dy4=interpolation(n_p,m_p,dGx_dy4_int);
dGx_dz4=interpolation(n_p,m_p,dGx_dz4_int);
```

```
dGy_dx4=interpolation(n_p,m_p,dGy_dx4_int);
dGy_dy4=interpolation(n_p,m_p,dGy_dy4_int);
dGy_dz4=interpolation(n_p,m_p,dGy_dz4_int);
dGz_dx4=interpolation(n_p,m_p,dGz_dx4_int);
dGz_dy4=interpolation(n_p,m_p,dGz_dy4_int);
dGz_dz4=interpolation(n_p,m_p,dGz_dz4_int);

% transform back into global coordinates
Gamma1=R_1*[Gx1; Gy1; Gz1];
Gamma2=R_2*[Gx2; Gy2; Gz2];
Gamma3=R_3*[Gx3; Gy3; Gz3];
Gamma4=R_4*[Gx4; Gy4; Gz4];

dGX1=R_1*[dGx_dx1; dGx_dy1; dGx_dz1];
dGY1=-R_1*[dGy_dx1; dGy_dy1; dGy_dz1];
dGZ1=-R_1*[dGz_dx1; dGz_dy1; dGz_dz1];

dGX2=R_2*[dGx_dx2; dGx_dy2; dGx_dz2];
dGY2=R_2*[dGy_dx2; dGy_dy2; dGy_dz2];
dGZ2=R_2*[dGz_dx2; dGz_dy2; dGz_dz2];

dGX3=R_3*[dGx_dx3; dGx_dy3; dGx_dz3];
dGY3=-R_3*[dGz_dx3; dGz_dy3; dGz_dz3];
dGZ3=R_3*[dGy_dx3; dGy_dy3; dGy_dz3];

dGX4=R_4*[dGx_dx4; dGx_dy4; dGx_dz4];
dGY4=R_4*[dGz_dx4; dGz_dy4; dGz_dz4];
dGZ4=-R_4*[dGy_dx4; dGy_dy4; dGy_dz4];

% make "G" and "dG/dn" vectors
GX=[Gamma1(1); Gamma2(1); Gamma3(1); Gamma4(1)];
GY=[Gamma1(2); Gamma2(2); Gamma3(2); Gamma4(2)];
GZ=[Gamma1(3); Gamma2(3); Gamma3(3); Gamma4(3)];
```

```
    dGX_dX=[dGX1(1); dGX2(1); dGX3(1); dGX4(1)];
    dGX_dY=[dGX1(2); dGX2(2); dGX3(2); dGX4(2)];
    dGX_dZ=[dGX1(3); dGX2(3); dGX3(3); dGX4(3)];

    dGY_dX=[dGY1(1); dGY2(1); dGY3(1); dGY4(1)];
    dGY_dY=[dGY1(2); dGY2(2); dGY3(2); dGY4(2)];
    dGY_dZ=[dGY1(3); dGY2(3); dGY3(3); dGY4(3)];

    dGZ_dX=[dGZ1(1); dGZ2(1); dGZ3(1); dGZ4(1)];
    dGZ_dY=[dGZ1(2); dGZ2(2); dGZ3(2); dGZ4(2)];
    dGZ_dZ=[dGZ1(3); dGZ2(3); dGZ3(3); dGZ4(3)];

    %Get "k" matrices
    knlX=GX*(dGX_dX')+GY*(dGX_dY')+GZ*(dGX_dZ');
    knlY=GX*(dGY_dX')+GY*(dGY_dY')+GZ*(dGY_dZ');
    knlZ=GX*(dGZ_dX')+GY*(dGZ_dY')+GZ*(dGZ_dZ');

    %form the total G matrix
    G=GX*GX'+GY*GY'+GZ*GZ';


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f=interpolation(n_p,m_p,A)
%function to interpolate from the offline field calculations.  A is a mXn
%matrix and y_p,z_p is the coordinate to be interplated at

%step 1, Identify the box the point is in
%y_p_inch=y_p/0.0254;
%z_p_inch=z_p/0.0254;
%n_p=(y_p_inch+2)/0.1+1;
%m_p=(z_p_inch+2)/0.1+1;
%n_p and m_p will most likely NOT be an integer, so find box corners
```

```
n_up=ceil(n_p);

n_bot=floor(n_p);

m_up=ceil(m_p);

m_bot=floor(m_p);

if m_bot==m_up && n_bot==n_up

    f=A(m_p,n_p);

elseif m_bot==m_up

    s=(A(m_bot,n_up)-A(m_bot,n_bot))/(n_up-n_bot);

    b=A(m_bot,n_bot)-s*n_bot;

    f=s*n_p+b;

elseif n_bot==n_up

    s=(A(m_up,n_bot)-A(m_bot,n_bot))/(m_up-m_bot);

    b=A(m_bot,n_bot)-s*m_bot;

    f=s*m_p+b;

elseif -n_p+(m_up+n_bot)>=m_p

    v1=[n_bot m_bot A(m_bot,n_bot)];

    v2=[n_up m_bot A(m_bot,n_up)];

    v3=[n_bot m_up A(m_up,n_bot)];

    C1=(v3(2)-v1(2))*(v2(3)-v1(3))-(v3(3)-v1(3))*(v2(2)-v1(2));

    C2=(v3(3)-v1(3))*(v2(1)-v1(1))-(v3(1)-v1(1))*(v2(3)-v1(3));

    C3=(v3(1)-v1(1))*(v2(2)-v1(2))-(v3(2)-v1(2))*(v2(1)-v1(1));

    C=[C1 C2 C3];

    D=-C(1)*v1(1)-C(2)*v1(2)-C(3)*v1(3);

    f=(-D-C(1)*(n_p)-C(2)*(m_p))/C(3);

else

    v1=[n_up m_up A(m_up,n_up)];

    v3=[n_up m_bot A(m_bot,n_up)];

    v2=[n_bot m_up A(m_up,n_bot)];

    C1=(v3(2)-v1(2))*(v2(3)-v1(3))-(v3(3)-v1(3))*(v2(2)-v1(2));

    C2=(v3(3)-v1(3))*(v2(1)-v1(1))-(v3(1)-v1(1))*(v2(3)-v1(3));

    C3=(v3(1)-v1(1))*(v2(2)-v1(2))-(v3(2)-v1(2))*(v2(1)-v1(1));

    C=[C1 C2 C3];

    D=-C(1)*v1(1)-C(2)*v1(2)-C(3)*v1(3);
    f=(-D-C(1)*(n_p)-C(2)*(m_p))/C(3);
end
```

C.3.6 Wall forces embedded function.

(closed_loop→plant_model→sphere_dynamics→wall_forces)

```
function [F_y_wall,F_z_wall] = wall_force_calc(Y,Z,spring)
% This block supports an embeddable subset of the MATLAB language.
% See the help menu for details.
%convert to inches
y_inch=Y/0.0254;
z_inch=Z/0.0254;
if y_inch>=(1-1/16)
    F_y_wall=(1-(1/16)-y_inch)*spring;
elseif y_inch<=-(1-1/16)
    F_y_wall=(-(1-(1/16))-y_inch)*spring;
else
    F_y_wall=0;
end
if z_inch>=(1-1/16)
    F_z_wall=(1-(1/16)-z_inch)*spring;
elseif z_inch<=-(1-1/16)
    F_z_wall=(-(1-(1/16))-z_inch)*spring;
else
    F_z_wall=0;
end
```

VITA

Matthew Glenn Duvall

Candidate for the Degree of

Doctor of Philosophy

Thesis: MODELING AND CONTROL OF A PERMANENTLY MAGNETIZED SPHERE'S MOTION BY FIELD MANIPULATION

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Okmulgee, Oklahoma, USA on October 19, 1971.

Education: Graduated from Putnam City North High School, Oklahoma City, Oklahoma in May, 1989; Completed the requirements for the Bachelor of Science degree with a major in Mechanical Engineering at Oklahoma State University in December, 1994; Completed the requirements for the Master of Science degree with a major in Mechanical Engineering at Oklahoma State University in May, 1997; Completed the requirements for the Doctor of Philosophy degree with a major in Mechanical Engineering at Oklahoma State University in December, 2006.

Experience: Research Assistant, Web Handling Research Center, Oklahoma State University, Stillwater, Oklahoma, 1995 to 1996; Process Engineer, Slitting Department, Exxon/Mobil Chemical Company, Shawnee, Oklahoma, 1996 to 1997; Senior Design Engineer, Mechanical Design Department, Seagate Technology, LLC, Oklahoma City, Oklahoma 1997 to 2003; Teaching Assistant, Mechanical and Aerospace Engineering Department, Oklahoma State University, Stillwater, Oklahoma, 2004 to 2005; Principle Research and Development Engineer, R&D Department, BD Medical, Sandy, Utah, 2006 to present.

Professional Memberships: Society of Industrial and Applied Mathematics (SIAM)

Name: Matthew Glenn Duvall Date of Degree: December, 2006

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: MODELING AND CONTROL OF A PERMANENTLY MAGNETIZED SPHERE'S MOTION BY FIELD MANIPULATION

Pages in Study: 217 Candidate for the Degree of Doctor of Philosophy

Major Field: Mechanical Engineering

A model describing the motion of a magnetized sphere in a fluid medium subject to an applied field is developed. The inspiration for this system is rooted in technologies and procedures being explored in the biomedical community. In these systems, the accepted approximation of Stokes' flow about the sphere is discussed and limitations to this are identified with drag force relations accounting for wall effects presented. For this system, the sphere is approximated as a point dipole and the resulting electromagnetic force model is presented. For motion of the sphere using static electromagnets, a reduced coil set technique is proposed. The technique consists of utilizing the minimum number of coils for generation of the desired force on the sphere. This approach is in contrast to the existing solutions which use the full coil set for parallel and anti-parallel motion and single out a solution for the underdetermined system. A technique for determining the proper coil combination is developed from examining the definiteness of the geometric field functions. A coil array configuration consisting of a four coil assembly is investigated to facilitate planar motion of a magnetized spherical particle in a fluid medium. For this configuration, the reduced coil set consists of only two adjacent coils being active at any given time. The exact inverse current solution for the minimized coil set is derived and presented. The system dynamic model is formulated and represented in a nonlinear state space system. This study also derives analytical expressions describing the magnetic field and field gradient terms generated by an assembly of four coils grouped about a coordinate origin with the central axes of the coils all residing in the same plane. This model will prove to be quite useful for optimal design of an apparatus to serve as an experimental platform for such a system. The model predicts the magnetic field produced by an elliptically shaped coil, approximated as a finite number of rectangularly shaped loops. The law of Biot-Savart is utilized to calculate the magnetic field and field gradient terms for each loop. The loops are then used to approximate each turn of a wound coil and are superposed to calculate the total field generated by a single coil. Homogenous coordinate transforms are utilized to describe the field generated by the quad coil configuration. An analytical formulation based on finite loops for calculation of self inductance, mutual inductance, and resistance is also derived and presented. Magnetic field measurements of wound coils are performed and compared to the predictive model, with satisfactory results being demonstrated. The technique of representing the wound coils as an assemblage of finite loops is found sufficient to formulate a model which may be used for predictive modeling of a system to facilitate two dimensional motion of a magnetized sphere. A minimization routine based on discrete steepest descent gradient method is developed and presented. The functional used for the minimization routine has been developed and is established as the norm of instantaneous power applied to an adjacent set of coils. This allows for summary of the power in two coils by one calculation. The routine calculates the required coil spacing and coil geometry necessary to achieve motion of the magnetized sphere along a given circular trajectory with minimal power dissipation. Control of a small magnetized spherical particle in 2D space using a square arrangement of electromagnets is presented. A classical controller with and without feedforward is implemented in concert with a novel switching approach. A sample of the experimental results for motion of the particle along a desired trajectory in a fluid medium are presented and discussed. Further, the experimental results are compared with computer simulation results to get insights into the accuracy of the dynamic model.

ADVISOR'S APPROVAL: ______________________________