MAGNETIC PARTICLE TWO DIMENSIONAL POSITION CONTROL USING MAGNETIC FIELD

By

WENJIE LI

Bachelor of Science in Engineering

Sichuan University

Chengdu, China

1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2006

MAGNETIC PARTICLE TWO DIMENSIONAL
POSITION CONTROL USING
MAGNETIC FIELD

Thesis Approved:

Eduardo A. Misawa

Thesis Adviser

Prabhakar R. Pagilla

Frank W. Chambers

A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my major adviser, Dr. Eduardo A. Misawa for his intelligent supervision, constructive guidance, inspiration and friendship. My sincere appreciation extends to my other committee members Dr. Prabhakar R. Pagilla and Dr. Frank W. Chambers, whose assistance, encouragement, and friendship are also invaluable. I would like to thank Dr. Afshin J. Ghajar for his warmhearted assistance.

I would also like to express my sincere gratitude to those who provided suggestions and assistance for this study: Matthew G. Duvall, Leon E. Boomershine III, Mallikarjuna R. Mudiam, and Yunfei Zou.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Motivation

Small magnetic particles or microdevices can be used in many modern medical applications, such as localized drug delivery [1], hyperthermia therapy to cure cancers [3], catheter ablation for heart procedures, and magnetic stereotaxis for brain procedures [9]. Using an external magnetic field to control these small or even micro sized objects is promising in *in vivo* applications to realize noninvasive or minimally invasive treatment. Not only in medical applications, non-contact magnetic manipulation is also an important factor in high precision machines, especially in micromachine applications because of their small work space [5].

In these applications, external magnetic fields are usually generated by magnets, coils, or superconductive coils. The field manipulation is achieved by controlling currents applied to the coils. The one dimensional (1-D) magnetic particle position control problem is much like the 1-D magnetic levitation problem, which has been thoroughly investigated. There is ongoing research trying to realize 3-D magnetic body/particle position control and put it into practice.

1.2 Related literature

1.2.1 3-D magnetic particle control using MRI system

A promising method for 3-D control of magnetic particles inside the human body utilizes an MRI (Magnetic Resonance Imaging) system [2,4] . Researchers are investigating the use of an MRI system to propel a ferromagnetic core, which in the future may be embedded in a microdevice. The MRI system will impart a three dimensional propulsion force. The long-term goal of such research is to provide surgeons ways of controlling microdevices inside blood vessels.

In an MRI system (Fig. 1), there is one main magnet and three gradient magnets. The main magnet generates a strong uniform background field, the three gradient magnets can generate a field gradient in a local region. The three dimensional field gradient will exert a force on a magnetic particle. The MRI system will also get an *artifact* (imaging distortion caused by the magnetic particle) in the scan image. The position of the artifact could be used as feedback signal to adjust the applied gradient which in turn will control the particle's motion. Research in this field is in an initial stage, and it relies on the expensive MRI system.

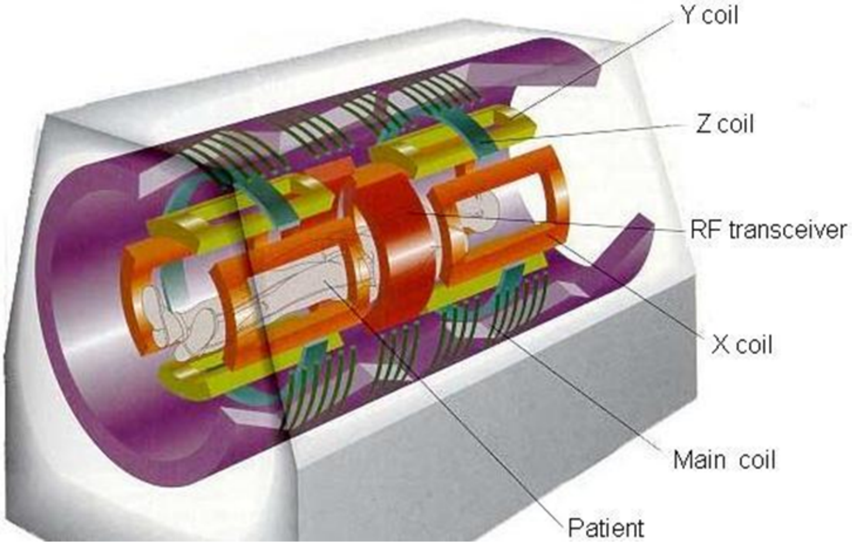

Figure 1: An MRI machine (www.medical-definitions.net/images/mri-machine.jpg)

1.2.2 A mechanism for 3-D levitated movement of a small magnet

There are also researchers working on 3-D levitated movement control of a small magnet (it can be regarded as a small magnetic dipole) that could be used in future micromachines [5,6]. They propose a mechanism that includes four electromagnets and a cross-shaped pole-piece to connect the magnetic poles. The geometric design of the pole-piece (Fig. 2) is the key for this mechanism. The combined pole-piece can generate a unique maximum point of the magnetic flux density B in the horizontal plane containing the magnetic dipole.

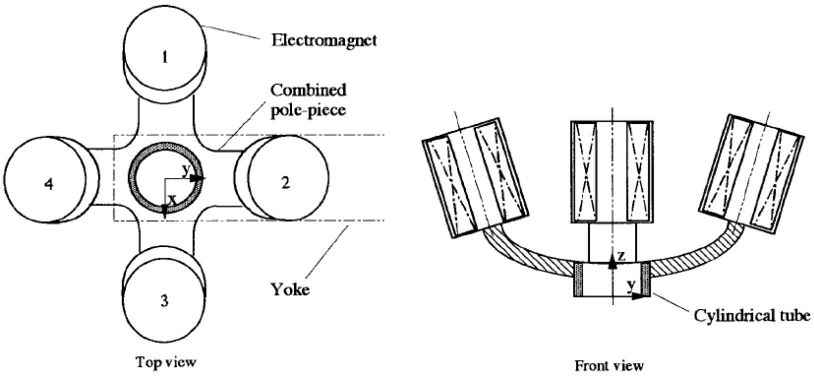

Figure 2: Illustration of the combined pole-piece

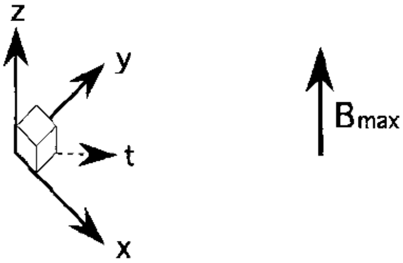

Figure 3: A small magnet in magnetic field B

For a small magnet levitating in the work space (Fig. 3), a vertical magnetic force (generated by additional compensative permanent magnets, not shown in the figure) will balance its gravity. In the horizontal direction the small magnet will be attracted to the B_{max} point (Fig. 3) and stay there, since no external force exists in the horizontal directions at the B_{max} point.

The location of B_{max} is determined by leak fluxes of the electromagnets, and the leak fluxes depend on the magnitude of each coil's current. So by controlling the currents, a small magnet's position can be controlled.

1.2.3 Magnetic stereotaxis system

Magnetic stereotaxis is a novel therapeutic methodology for the treatment of brain tumors and other neurological problems. The basic idea of the magnetic stereotaxis system is that large electromagnetic coils can be used to guide a small piece of implanted permanent magnetic material (a “magnetic seed”) along some arbitrary trajectory through brain tissue. Once the seed has been maneuvered into a tumor, the seed can be heated inductively by applied high-frequency magnetic fields to kill tumor cells, or the seed can be used to guide a tip of catheter to deliver drugs.

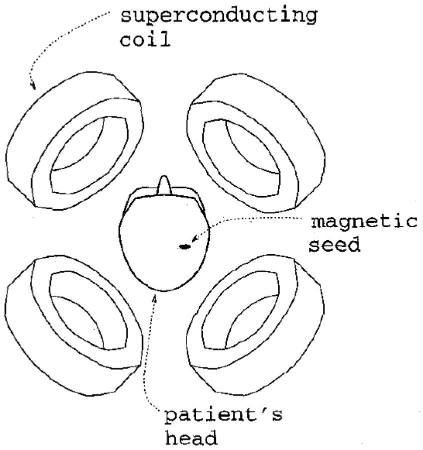

Figure 4: Schematic of multicoil MSS

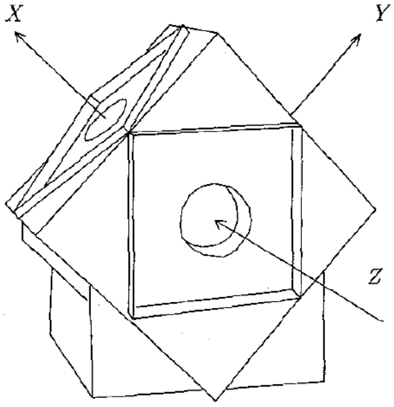

Figure 5: A six-coil MSS

Researchers at the University of Virginia have proposed a multicoil magnetic stereotaxis system [9,10,11] (Fig. 4, Fig. 5). It uses six superconducting coils to generate the desired three dimensional force on the seed. An MSS device has been built and tested for open-loop step motions of a seed.

2 Problem Description

For the problem of one dimensional position control of a magnetic particle, only one control variable is needed. Because a coil always exerts an attractive force to a magnetic particle which is allowed to rotate freely, only one coil is needed to balance a particle against its gravity and control its position for a 1-D magnetic levitation problem. If gravity is not considered, to control a particle's position one dimensionally, two coils must be used (Fig. 6). The currents of the two coils can be controlled by one control variable in a reciprocal way, where the two currents vary inversely and the variation depends on the control input, then a position control can be achieved.

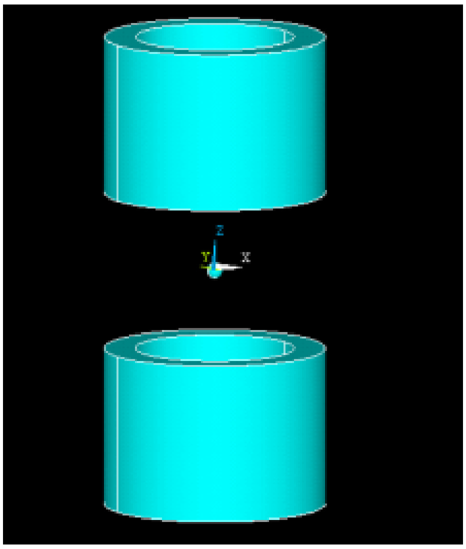

Figure 6: One dimensional control problem

Moving to the two dimensional control problem, it may be desired to control the position of a particle of which motion is confined within a plane by using four coils (Fig. 7). This system is analogous to a two dimensional MSS system. The difference is that we are using four elongated rectangular coils, which generate a field distribution

(Fig. 14) stronger and more even in the central plane ($x - z$ plane) than the field distribution circular coils generate (Fig. 18). This difference is illustrated in chapter 4.

The two dimensional control problem is stated as follows:

In a four-rectangular-coil frame as shown in Fig. 7, it is desired to control the position of a permanent magnetic particle, of which the motion is confined within the $x - z$ plane, by moving it to a target point and stabilizing it there using a magnetic field generated by the four coils. The coils carry direct currents. Gravity is not considered.

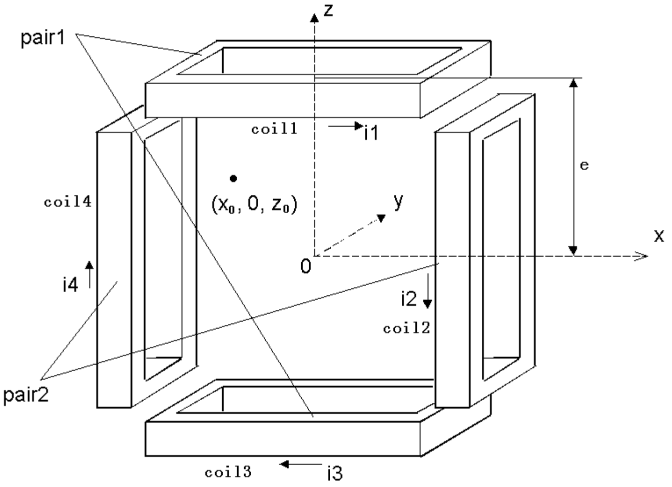

Figure 7: Two dimensional control problem

In this symmetric four-rectangular-coil configuration, there is a parameter e, which is the distance from the center of each coil to the origin.

The parameters of a single coil are shown in Fig. 7 and Table 1.

Table 1: Definition of parameters of the coil

parameter	definition
a	inner length of the coil
b	inner width of the coil
c	cross-section width
d	cross-section height
N	number of turns

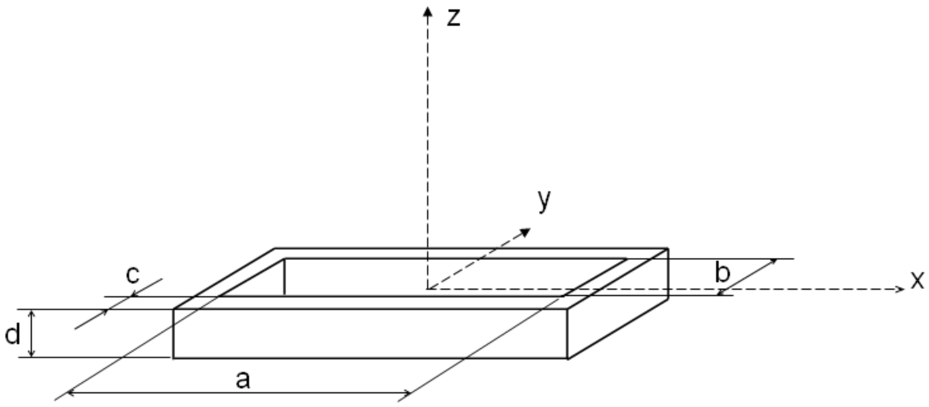

Figure 8: Single rectangular coil

Ferromagnetic materials that are not permanently magnetized may exhibit hysteresis. This may result in their magnetic moments changing under a varying external magnetic field. In this paper we only consider permanent magnetized particles which have constant magnetic moments.

In reality, a "rectangular" coil can not be made, so these coils are only approximations to elongated wound coils. If the radius of round corners of a real coil is small compared to the coil's length and width, this approximation should be acceptable for the purpose of field computation.

3 Mathematical Model

3.1 Motion equations

A small permanent magnetic particle can be represented by a magnetic dipole $\mathbf{p}$. The force and torque exerted on a permanent magnetic particle are [13]

$$\mathbf{F} = \nabla(\mathbf{p} \cdot \mathbf{B}) \tag{3.1}$$

$$\mathbf{T} = \mathbf{p} \times \mathbf{B} \tag{3.2}$$

where the torque $\mathbf{T}$ causes the magnetic moment $\mathbf{p}$ to align with the magnetic field $\mathbf{B}$ (Fig. 9). In this paper, a bold character denotes a vector and a plain character denotes its magnitude.

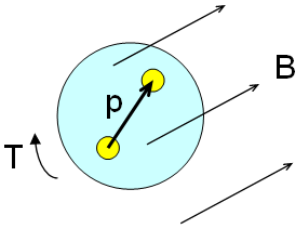

Figure 9: Torque on a magnetic particle

It can be noted that torque on the particle is zero only when $\mathbf{p}$ is parallel or antiparallel to $\mathbf{B}$. Furthermore, the parallel state is a stable rotational equilibrium position of the particle, while the antiparallel state is an unstable rotational equilibrium position. If any directional change in $\mathbf{B}$ produces a torque, the torque tends to re-align $\mathbf{p}$ and $\mathbf{B}$. Assuming that the resistance of the particle to rotation is negligible, then it may be assumed that it takes negligible time for the particle to rotate to its stable parallel state, and the magnetic moment $\mathbf{p}$ always keeps the same orientation as **B** during its motion. Therefore, the dot product in (3.1) equals the product of magnitudes of **p** and **B**. So the force exerted on the particle is

$$\mathbf{F} = \nabla(pB) = p\nabla B,$$

where p is the magnitude of magnetic moment of the particle and B is the magnitude of the field intensity. The components of the force in $x - z$ plane are

$$F_x = p(\frac{\partial B}{\partial x}), \qquad F_z = p(\frac{\partial B}{\partial z}). \tag{3.3}$$

where F_x and F_z are the component of the force along x direction and z direction respectively.

Not considering the gravity and fluid force, the motion equation of a particle along x direction is

$$\ddot{x} = \frac{F_x}{m} = \frac{\partial B}{\partial x}\frac{p}{m}, \tag{3.4}$$

and similarly, the motion equation along z direction is

$$\ddot{z} = \frac{F_z}{m} = \frac{\partial B}{\partial z}\frac{p}{m} \tag{3.5}$$

where m is the mass of the magnetic particle.

3.2 Field equations

In the most general case, the magnetic field (at a point in free space) produced by a current complies with Maxwell's equations for the special case of magnetostatic fields [12,13]:

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} \tag{3.6}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{3.7}$$

where $\mathbf{j}$ is vector current density, μ_0 is the permeability of free space. The field flux density $\mathbf{B}$ can be defined as the curl of vector potential $\mathbf{A}$ of the field, then the magnetostatic Maxwell's equations become:

$$\nabla \times \nabla \times \mathbf{A} = \mu_0 \mathbf{j} \tag{3.8}$$

$$\nabla \cdot \nabla \times \mathbf{A} = 0 \tag{3.9}$$

these two equations are both satisfied if

$$\nabla \cdot \nabla \mathbf{A} = -\mu_0 \mathbf{j} \tag{3.10}$$

The contribution of a differential volume (dV) of current to the vector potential $\mathbf{A}$ has the following solution [9]

$$d\mathbf{A} = \frac{\mu_0}{4\pi} \frac{\mathbf{j}dV}{|\mathbf{R}|} \tag{3.11}$$

where $\mathbf{R}$ is the vector distance from the differential current element to the point of interest.

The field intensity is found by differentiating $\mathbf{B} = \nabla \times \mathbf{A}$, and substituting from (3.11) [9],

$$d\mathbf{B} = \frac{\mu_0}{4\pi} \frac{\mathbf{j}dV \times \mathbf{R}}{|\mathbf{R}|^3} \tag{3.12}$$

For the magnetic field generated by a current-carrying wire (Fig. 10), the above equation takes the following form:

$$d\mathbf{B} = \frac{\mu_0 i}{4\pi} \frac{d\mathbf{L} \times \mathbf{R}}{|\mathbf{R}|^3} \tag{3.13}$$

where i is the current value, $d\mathbf{L}$ is a differential element of the current carrying wire. This relation is also known as the law of Biot-Savart .

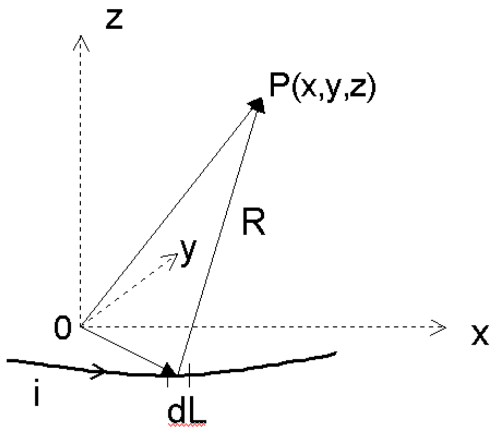

Figure 10: A filamentary current-carrying wire

By applying the Biot-Savart law to a single coil centered at the origin (Fig. 8) with a volume integration, the x component of the field $\mathbf{B}$ at a point (x_0, z_0) in the $x-z$ plane can be represented as

$$B_x = ki\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{z_0-z}{[(x_0-x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}-\frac{z_0-z}{[(x_0+x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}\}dydxdz$$

where $k=\frac{\mu_0 N}{4\pi cd}$, and i is the current input to the coil.

The other two components of the $\mathbf{B}$ field are

$$B_y = ki\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{-\frac{b}{2}-c}^{-\frac{b}{2}}\int_{y-\frac{a-b}{2}}^{\frac{a-b}{2}-y}\{\frac{z-z_0}{[(x_0-x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}-\frac{z-z_0}{[(x_0+x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}\}dxdydz,$$

and

$$B_z = ki\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{-\frac{b}{2}-c}^{-\frac{b}{2}}\int_{y-\frac{a-b}{2}}^{\frac{a-b}{2}-y}\{\frac{y_0-y}{[(x_0-x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}-\frac{y_0-y}{[(x_0+x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}\}dxdydz$$

$$+ki\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{-\frac{b}{2}-c}^{-\frac{b}{2}}\int_{y-\frac{a-b}{2}}^{\frac{a-b}{2}-y}\{\frac{x-x_0}{[(x_0-x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}-\frac{x-x_0}{[(x_0+x)^2+y^2+(z_0-z)^2]^{\frac{3}{2}}}\}dxdydz.$$

For the four-rectangular-coil configuration, through coordinate translation and rotation, according to superposition principle, the field $\mathbf{B}$ at a point (x_0, z_0) in the

$x-z$ plane of the global coordinates system (Fig. 7) can be represented as:

$$B_x = i_1 f_{1x}(x_0, z_0) + i_2 f_{2x}(x_0, z_0) + i_3 f_{3x}(x_0, z_0) + i_4 f_{4x}(x_0, z_0)$$

$$B_z = i_1 f_{1z}(x_0, z_0) + i_2 f_{2z}(x_0, z_0) + i_3 f_{3z}(x_0, z_0) + i_4 f_{4z}(x_0, z_0)$$

where f_{1x}, f_{2x}, f_{3x}, and so on [A] are functions that are determined by the position (x_0, z_0). For succinctness (x_0, z_0) is omitted in following equations. Since the motion of the particle is confined in the $x-z$ plane, we do not need to consider B_y (in fact $B_y = 0$ and $(\nabla B)_y = 0$ in $x-z$ plane).

Rewriting above equations in vector form gives

$$B_x = \mathbf{i}^T \mathbf{f}_x; \quad B_z = \mathbf{i}^T \mathbf{f}_z, \tag{3.14}$$

where

$$\mathbf{i} = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix}; \quad \mathbf{f}_x = \begin{bmatrix} f_{1x} \\ f_{2x} \\ f_{3x} \\ f_{4x} \end{bmatrix}; \quad \mathbf{f}_z = \begin{bmatrix} f_{1z} \\ f_{2z} \\ f_{3z} \\ f_{4z} \end{bmatrix}.$$

The magnitude of vector $\mathbf{B}$ is

$$B = (B_x^2 + B_z^2)^{\frac{1}{2}} = \sqrt{(\mathbf{i}^T \mathbf{f}_x)^2 + (\mathbf{i}^T \mathbf{f}_z)^2}, \tag{3.15}$$

and the field gradient equations are:

$$\frac{\partial B}{\partial x_0} = \frac{(\mathbf{i}^T \mathbf{f}_x)(\mathbf{i}^T \frac{\partial \mathbf{f}_x}{\partial x_0}) + (\mathbf{i}^T \mathbf{f}_z)(\mathbf{i}^T \frac{\partial \mathbf{f}_z}{\partial x_0})}{\sqrt{(\mathbf{i}^T \mathbf{f}_x)^2 + (\mathbf{i}^T \mathbf{f}_z)^2}} \tag{3.16}$$

$$\frac{\partial B}{\partial z_0} = \frac{(\mathbf{i}^T \mathbf{f}_x)(\mathbf{i}^T \frac{\partial \mathbf{f}_x}{\partial z_0}) + (\mathbf{i}^T \mathbf{f}_z)(\mathbf{i}^T \frac{\partial \mathbf{f}_z}{\partial z_0})}{\sqrt{(\mathbf{i}^T \mathbf{f}_x)^2 + (\mathbf{i}^T \mathbf{f}_z)^2}} \tag{3.17}$$

In agreement with equations 3.4 and 3.5, we denote the coordinates of a point in the $x - z$ plane as (x, z) instead of (x_0, z_0), so functions $\mathbf{f}_x, \mathbf{f}_z$ are determined by the position (x, z), then,

$$\frac{\partial B}{\partial x} = \frac{(\mathbf{i}^T \mathbf{f}_x)(\mathbf{i}^T \frac{\partial \mathbf{f}_x}{\partial x}) + (\mathbf{i}^T \mathbf{f}_z)(\mathbf{i}^T \frac{\partial \mathbf{f}_z}{\partial x})}{\sqrt{(\mathbf{i}^T \mathbf{f}_x)^2 + (\mathbf{i}^T \mathbf{f}_z)^2}}$$

$$\frac{\partial B}{\partial z} = \frac{(\mathbf{i}^T \mathbf{f}_x)(\mathbf{i}^T \frac{\partial \mathbf{f}_x}{\partial z}) + (\mathbf{i}^T \mathbf{f}_z)(\mathbf{i}^T \frac{\partial \mathbf{f}_z}{\partial z})}{\sqrt{(\mathbf{i}^T \mathbf{f}_x)^2 + (\mathbf{i}^T \mathbf{f}_z)^2}}.$$

Rearranging above equations,

$$\frac{\partial B}{\partial x} = \frac{\mathbf{i}^T [\mathbf{f}_x, \mathbf{f}_z] \begin{bmatrix} \frac{\partial \mathbf{f}_x^T}{\partial x} \\ \\ \frac{\partial \mathbf{f}_z^T}{\partial x} \end{bmatrix} \mathbf{i}}{\sqrt{\mathbf{i}^T [\mathbf{f}_x, \mathbf{f}_z] \begin{bmatrix} \mathbf{f}_x^T \\ \\ \mathbf{f}_z^T \end{bmatrix} \mathbf{i}}}$$

$$\frac{\partial B}{\partial z} = \frac{\mathbf{i}^T [\mathbf{f}_x, \mathbf{f}_z] \begin{bmatrix} \frac{\partial \mathbf{f}_x^T}{\partial z} \\ \\ \frac{\partial \mathbf{f}_z^T}{\partial z} \end{bmatrix} \mathbf{i}}{\sqrt{\mathbf{i}^T [\mathbf{f}_x, \mathbf{f}_z] \begin{bmatrix} \mathbf{f}_x^T \\ \\ \mathbf{f}_z^T \end{bmatrix} \mathbf{i}}}.$$

Denoting $[\mathbf{f}_x, \mathbf{f}_z]$ as $\mathbf{f}^T$, then,

$$\frac{\partial B}{\partial x} = \frac{\mathbf{i}^T \mathbf{f}^T (\frac{\partial \mathbf{f}}{\partial x})\ \mathbf{i}}{\sqrt{\mathbf{i}^T \mathbf{f}^T \mathbf{f}\ \mathbf{i}}} \tag{3.18}$$

$$\frac{\partial B}{\partial z} = \frac{\mathbf{i}^T \mathbf{f}^T (\frac{\partial \mathbf{f}}{\partial z})\, \mathbf{i}}{\sqrt{\mathbf{i}^T \mathbf{f}^T \mathbf{f}\, \mathbf{i}}}. \tag{3.19}$$

It can be seen that the field gradient components are quadratic forms of $\mathbf{i}$, and their values depend on the currents, the position (x, z), and geometric parameters of the coils.

3.3 About the analytic form of the field gradient

In the above field gradient equations, the functions $\mathbf{f}_x$ and $\mathbf{f}_z$ and their partial derivatives do not have an analytic form. For example,

$$\frac{\partial f_{1x}}{\partial x_0} = k\frac{\partial\Big\{ \int_{-\frac{d}{2}}^{\frac{d}{2}} \int_{\frac{a}{2}}^{\frac{a}{2}+c} \int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}} \{\frac{z_0-e-z}{[(x_0-x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}} - \frac{z_0-e-z}{[(x_0+x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}}\} dydxdz\Big\}}{\partial x_0}$$

does not have an analytic form.

For one rectangular filamentary current loop (Fig. 12), however, an analytic form of the gradient can be obtained by working out the loop integral and taking derivatives [20]. The gradient components (the two partial derivatives) also follow the rule of superposition. So we may get the analytic form of the gradient produced by each turn of a coil, then add them together to get an analytic form of the gradient produced by one coil (Fig. 11). That is, for this coil, the field gradient produced at a point of interest

$$\nabla B = \sum_{j=1}^{m} \sum_{k=1}^{n} \nabla B_{jk}$$

where n is the number of layers of the coils, m is the number of turns per layer, and ∇B_{jk} is the gradient generated by the jth turn on the kth layer.

Through coordinate translation and rotation, the field gradient produced by each coil in the global coordinates system can be obtained. Then the total gradient can reach an analytic form by superposition.

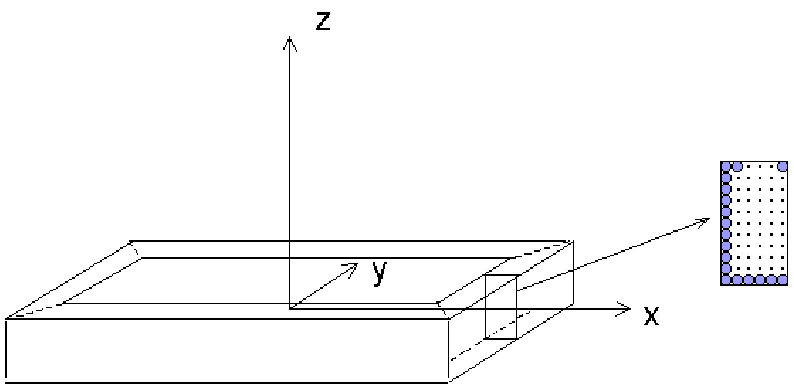

Figure 11: Turns of one rectangular coil

A coil usually has many turns, so it is cumbersome to get an analytic form of the total field gradient by this approach. Although we can write a Matlab program to compute the total gradient at a point of interest analytically through this method, it takes very long time. As a quicker approach, numerical computation needs to be applied.

4 Position Control Scheme

4.1 Numerical computation of the magnetic field

For a filamentary current-carrying loop (Fig. 12), the field intensity at a point in free space can be calculated through a loop integration using the law of Biot-Savart,

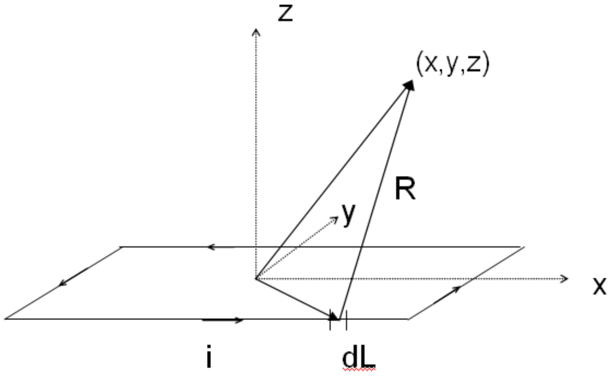

Figure 12: A rectangular filamentary current loop

$$\mathbf{B} = \frac{\mu_0 i}{4\pi} \oint \frac{d\mathbf{L} \times \mathbf{R}}{|\mathbf{R}|^3}.$$

For each segment of the loop, the integrand is different. Using Simpson's rule of numerical integration, the field intensity $\mathbf{B}$ generated by a rectangular loop may be computed.

For one rectangular coil (Fig. 13), its cross-sectional area may be divided into many small areas, and the coil may be viewed as being composed of closed loops of wire which has the small cross-section area. The current flowing through these loops will be the product of the current density and the small area. For each loop, its contribution to the field intensity at the point of interest is computed, and contributions from all the loops are added together to get the total field intensity generated by this coil.

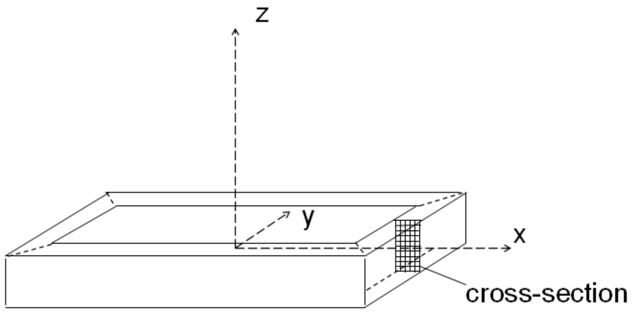

Figure 13: Numerical computation method

For the 4-rectangular-coil configuration, first the field intensity generated by each coil is calculated in its own local coordinate system. Then coordinate translation and rotation are performed to get the contribution of each coil to the total field intensity in the global coordinate system.

The total field intensity at the point of interest is obtained by superposition. The gradient of B is also computed based on values of B using the following midpoint numerical differentiation rule,

$$f^{'}(x) = \frac{f(x+h) - f(x-h)}{2h}.$$

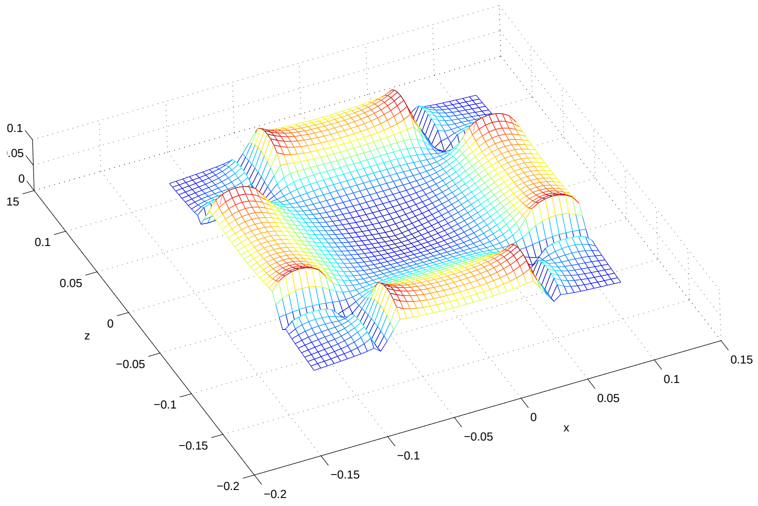

Figure 14: Field magnitude map produced by a four-rectangular-coil config.

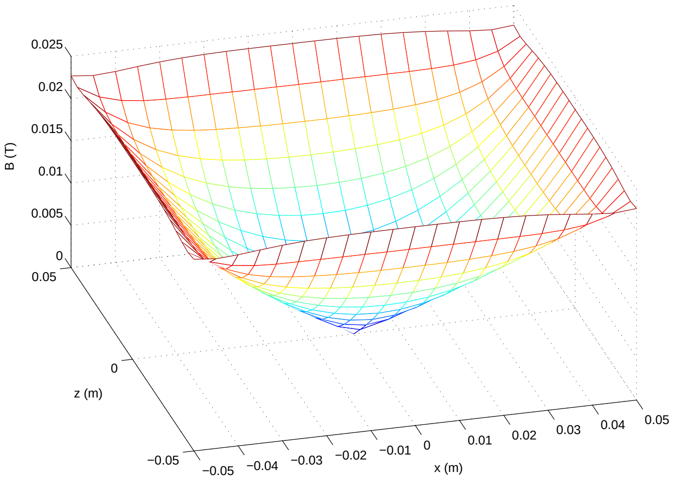

Figure 15: A sharp minimum point

The field and gradient computing code was embedded into a simulink model as an S-function for the purpose of simulation. The field intensity at any point in free space can be obtained using this numerical method. The results of this method were compared to the results calculated by Maple (in which the field components are calculated using the volume integration equations in Chapter 3), for a number of points picked in free space. The accuracy of this method increases as the cross-section area is divided into smaller areas, and the results of this method converge to the results from Maple. By computing B values at many grid points in the $x - z$ plane, we can obtain a map of the magnitude of B in the $x - z$ plane, as shown in Fig. 14. This map is generated under the following conditions:

the four currents are equal to 4A;

the directions of the currents are as shown in Fig. 7;

geometric parameters (in Fig. 7 and Fig. 8) are:

$a = 0.1m, \quad b = 0.05m, \quad c = 0.02m, \quad d = 0.04m, \quad e = 0.095m, \quad N = 2500.$

This magnetic field map has a sharp minimum point at origin, as shown in Fig. 15. If the current directions' configuration is changed to the one as shown in Fig. 16, where the two coils in each pair have the same current direction, then the magnetic field map has a smooth minimum point.

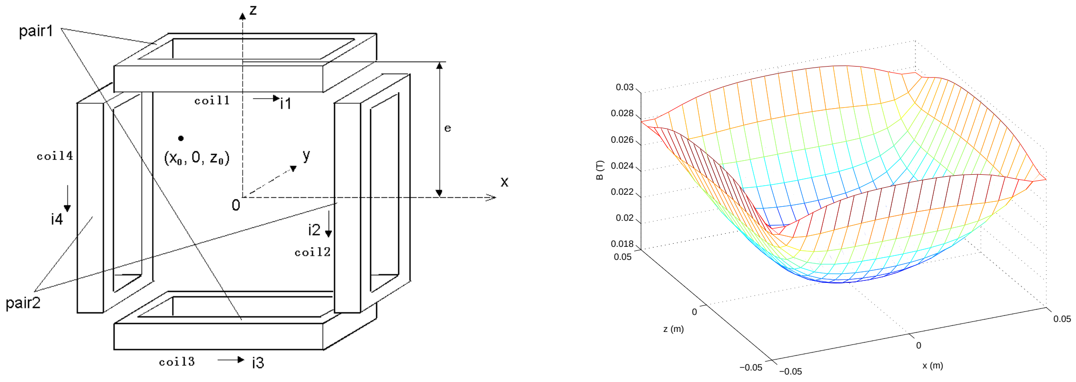

Figure 16: One config. of current directions that produce a smooth bottom

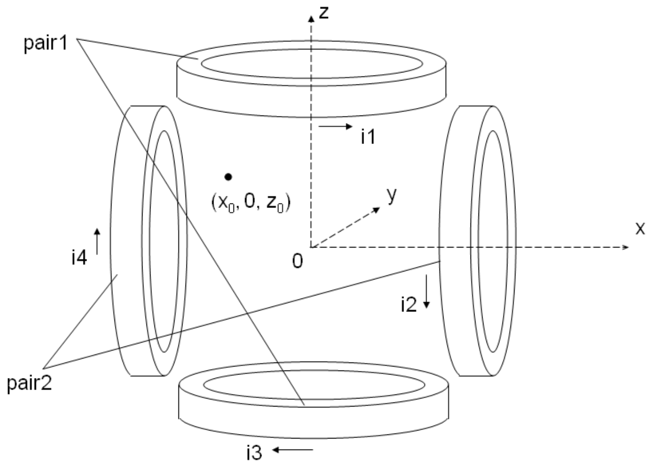

Figure 17: The four-circular-coil configuration

The same numerical computation method can be used to compute field intensity generated by circular coils. In a 4-circular-coil configuration (Fig. 17), four circular coils which have the same current, cross-sectional area, and inner diameter (corresponding to inner length) as those coils in Fig. 14 will produce a B magnitude distribution in the $x - z$ plane as shown in Fig. 18. It can be seen that the region of the map around the origin is not as regular as that in a 4-rectangular-coil map, where the central part looks like a bowl with a minimal point at the bottom. This field shape character is critical to fulfill our control scheme that will be brought up in section 4.3. This is also the reason why rectangular coils are adopted in this research.

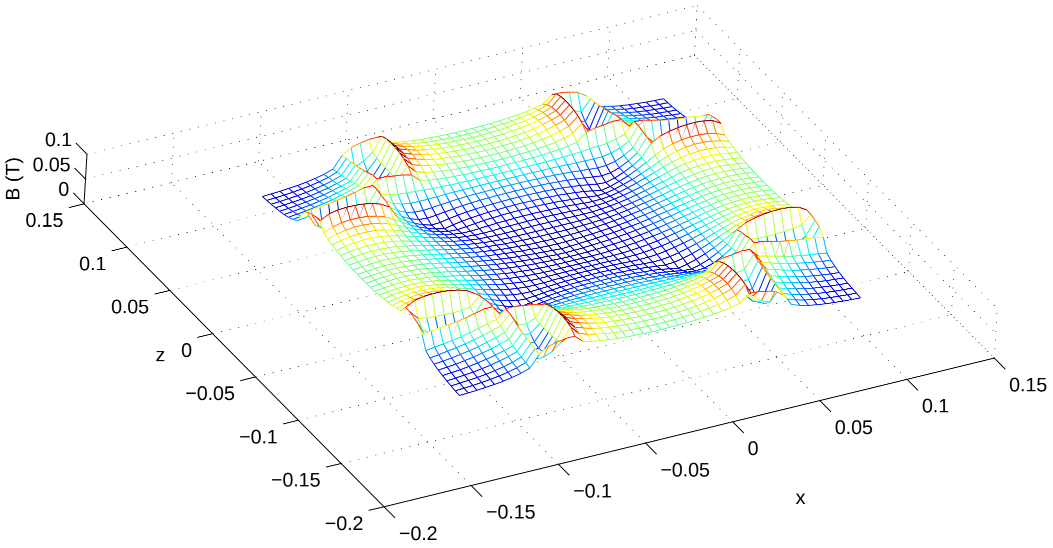

Figure 18: B magnitude map for four-circular-coil config.

4.2 Inverse linearization method

From equations 3.4 and 3.5, if we take the two force components as system control inputs, the system is

$$\begin{cases} \ddot{x} = \frac{1}{m}F_x \\ \ddot{z} = \frac{1}{m}F_z \end{cases}$$

This is a linear system, the position control can be realized by a feedback controller (Fig. 19), such as a proportional controller.

In practice we can not control the force directly, we can only control the currents. The relationship between the currents and the force can be obtained from equations 3.3, 3.18, and 3.19. In general the force is a function of the four currents and position (x, z), that is

$$\mathbf{F} = h(\mathbf{i}, x, z). \tag{4.20}$$

Conversely, given x, z, if we can find a solution of $\mathbf{i}$ to an arbitrary desired force, then we can use these currents as control inputs to achieve our control goal of the

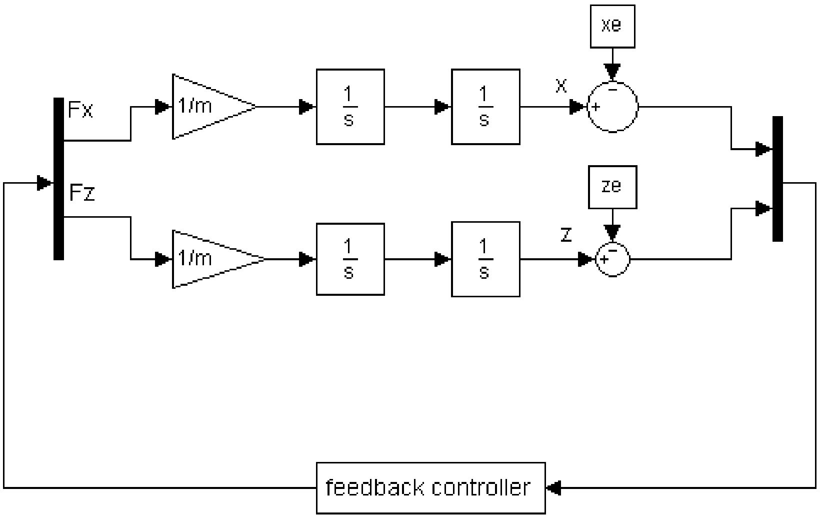

Figure 19: A linear system controlled by force

system. This method is illustrated in Fig. 20, where the 'inverse solution' block solves for currents **i** based on x, z, and desired force. Then these currents are input to the four coils to generate the desired ∇B, and the control goal can be achieved.

The problem of finding an **i** for an arbitrary desired force was solved by Meeker, et al [9]. Generally, there exist infinite possible **i** that satisfy equation (4.20) for a desired force. So a cost function was introduced to form an optimization problem which takes equation (4.20) as constraints, and this optimization problem can be solved numerically [9].

Although this inverse linearization control method is theoretically feasible, it uses four control inputs for this 2-D problem, and it involves intensive numerical computational work. This method will not be implemented in this paper.

4.3 Two control variables method

4.3.1 Unstable equilibrium target point

For this 2-D control problem, we may need only two control variables. Just as the way two coils are used for 1-D problem (chapter 2), in the 4-rectangular-coil configuration (Fig. 7) each pair of coils can be controlled by one variable. The currents of a pair of coils can be changed in a reciprocal way by the control variable.

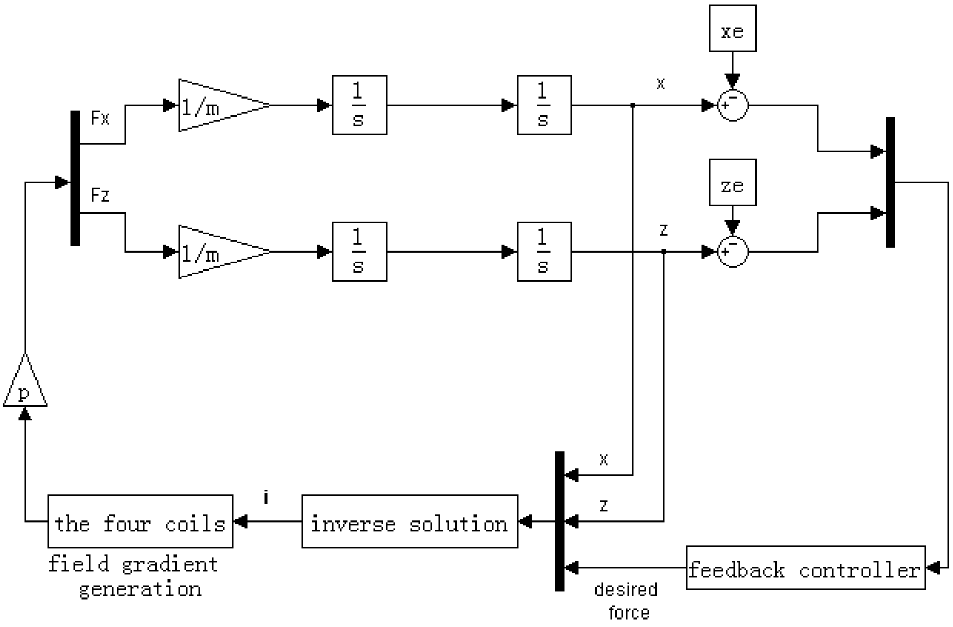

Figure 20: Inverse linearization control method

Let us examine the case when the particle is in its steady state at the target point first. If the target point is at the origin, and the four currents are of the same value, from the field maps in section 4.1 it can be seen that the origin is an unstable equilibrium point where the gradient of B is zero (the slope of the surface of the B map is ∇B). For a target point other than the origin, we may find a combination of the four currents (*nominal currents*) that makes the target point an unstable equilibrium point. Then we can apply some control law to stabilize a magnetic particle to this equilibrium point.

Now it is similar to the optimization problem mentioned above, there are infinite solutions of **i** which generate a zero gradient of field at a target point in the $x-z$ plane. Constraints need to be added. Let us require the two currents of each pair change inversely about a central value I_c (for example, $I_c = 4A$) when we are searching for the four nominal currents numerically, so the sum of each pair of currents is always $2I_c$ (8A). The nominal currents searching code was embedded in our simulink model.

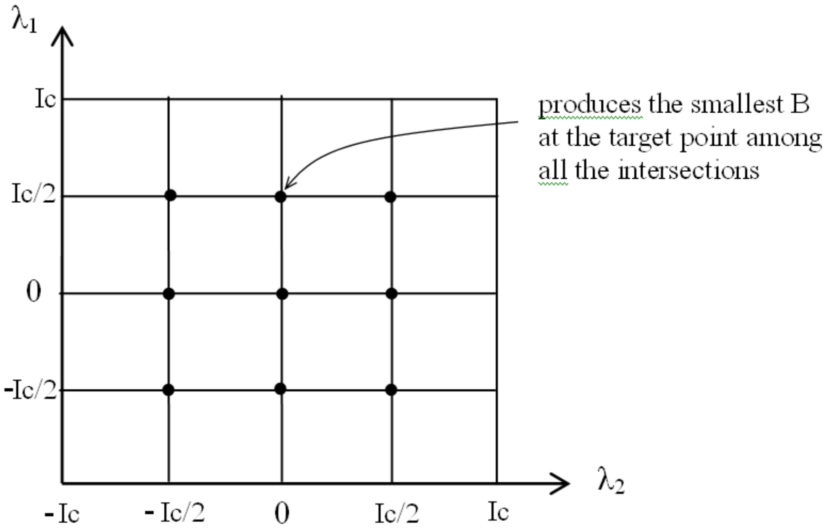
λ1
Ic
Ic/2
0
-Ic/2
-Ic
- Ic/2
0
Ic/2
Ic
λ2
produces the smallest B at the target point among all the intersections

Figure 21: First iteration step

4.3.2 Nominal currents search

The above constraints for nominal currents are represented as:

$$pair1:\ i_1 = I_c - \lambda_1,\ i_3 = I_c + \lambda_1$$

$$pair2:\ i_2 = I_c - \lambda_2,\ i_4 = I_c + \lambda_2$$

where the central current value $I_c = 4A$, $\lambda_1 \in (-I_c, I_c)$, $\lambda_2 \in (-I_c, I_c)$.

Now we need to search for suitable λ_1 and λ_2 values that make the four currents nominal for a given target point.

It can be seen that a point (x, z) inside the four-rectangular-coil frame at which $\nabla B = 0$ is a minimum point of B (in Fig. 14, the field is not smooth at the equilibrium point, theoretically the gradient of B is undefined at that point, but regardless of the current direction configuration, the equilibrium point is always a minimum point). We may use an iterative method to search for the suitable λ_1 and λ_2 values which produce a minimum B value at the target point. This iterative method is illustrated in Fig. 21, Fig. 22, Fig. 23.

First we divide the changing range $(-I_c, I_c)$ of λ_1 and λ_2 by a given number (for example, 4), to form grids as shown in Fig. 21. Of the intersection points inside boundary (there are 9 points here), each point represents a combination of λ_1 and

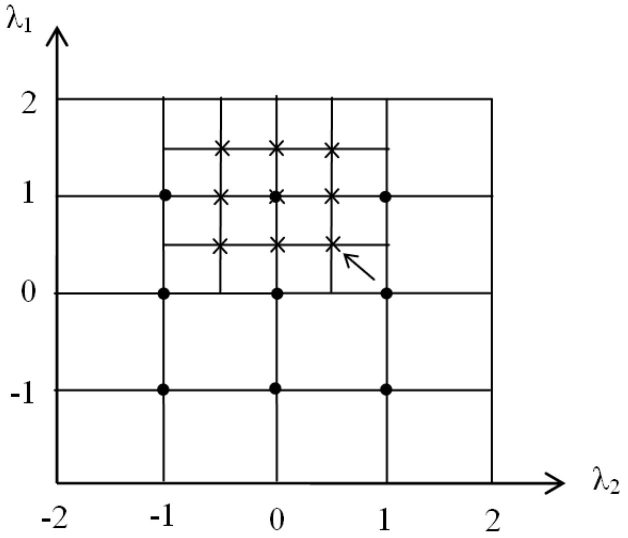

Figure 22: Second iteration step

λ_2 values. For each combination, the produced B value and ∇B at the target point (x_e, z_e) are computed. Then a test is made to see if $|\nabla B| < \epsilon$, where ϵ is a small number adjustable for accuracy. If this combination satisfied the accuracy requirement, we get our nominal values and quit. If not, compare the B values for all intersection points to find a smallest one, for example, the middle upper one in Fig. 21. Take this point as the center point for next iteration, in a range of the four panels around this point, redo the above iteration step.

As shown in Fig. 22, no combination is satisfactory and the smallest B occurs at the right lower point, take this point as the center point for next iteration, redo the iteration (Fig. 23), until a satisfactory combination of λ_1 and λ_2 is found.

This numerical searching method can be viewed as a kind of directional searching method. This method provides accurate results in a short time period.

Here is an example of the searching result, for a target point at (0.01, 0.01) and $I_c = 4A$, the four nominal currents are:

$i_{1n} = i_{2n} = 3.0263671875\ A$

$i_{3n} = i_{4n} = 4.9736328125\ A$,

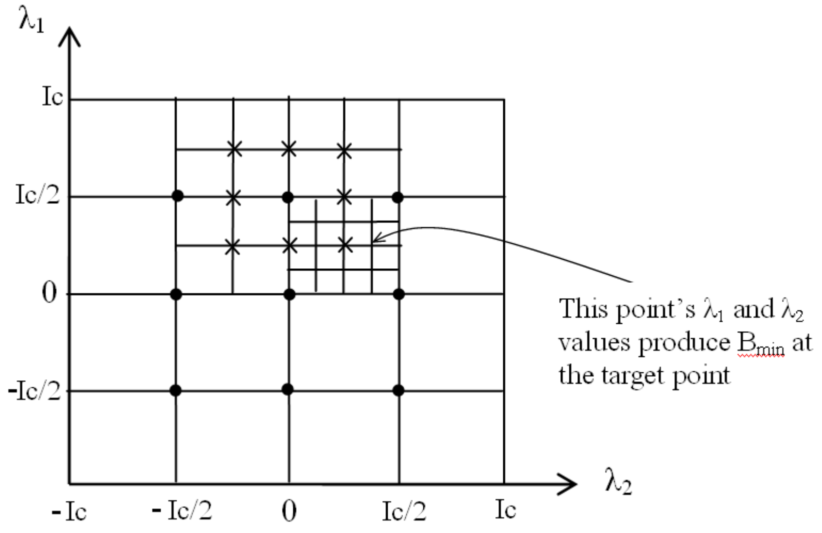
λ1
Ic
Ic/2
0
-Ic/2
λ2
- Ic
- Ic/2
0
Ic/2
Ic
This point's λ1 and λ2 values produce Bmin at the target point

Figure 23: Third iteration step

and the field magnitude map on the $x-z$ plane under these nominal currents is shown in Fig. 24.

4.3.3 Control scheme

The control scheme for the two control variables method is proposed as follows:

To stabilize a magnetic particle at a target point (x_e, z_e), first we set this point as an unstable equilibrium point to get the four nominal currents $i_{1n}, i_{2n}, i_{3n}, i_{4n}$, under these currents the gradient of B equals zero. Then we use two control currents α, β, to vary the currents of pair1 and pair2 (around each coil's nominal value) separately, in a reciprocal pattern:

$$pair1: \; i_1 = i_{1n} - \alpha, \; i_3 = i_{3n} + \alpha;$$

$$pair2: \; i_2 = i_{2n} - \beta, \; i_4 = i_{4n} + \beta.$$

The control law to generate the two control currents would be a feedback controller.

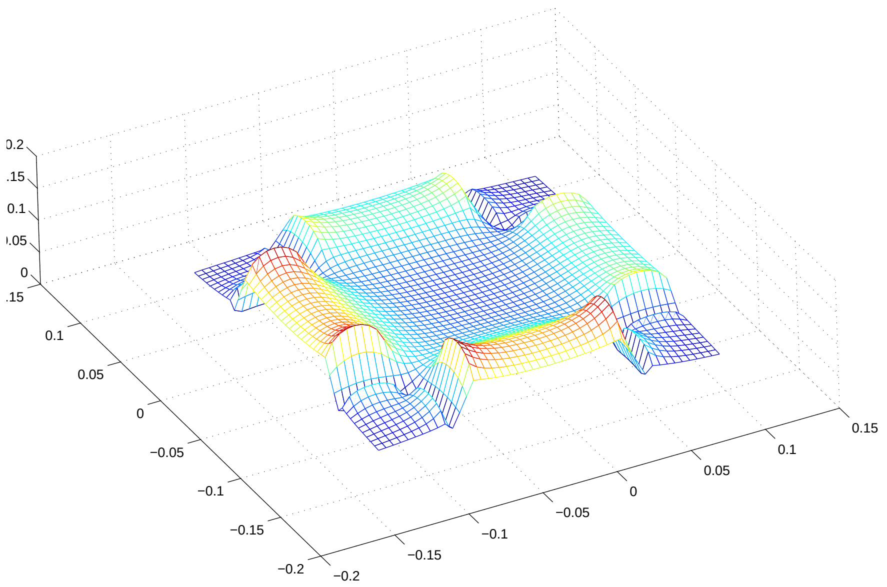

Figure 24: B map where (0.01, 0.01) is an equilibrium point

4.3.4 System representation

Now let us take a look on the system representation in terms of the two control inputs (α and β). From equation 3.14, $\mathbf{i} = [i_{1n} - \alpha, i_{2n} - \beta, i_{3n} + \alpha, i_{4n} + \beta]^T$ and

$$\begin{aligned} B_x &= (i_{1n} - \alpha) f_{1x} + (i_{2n} - \beta) f_{2x} + (i_{3n} + \alpha) f_{3x} + (i_{4n} + \beta) f_{4x} \qquad (4.21) \\ &= i_{1n} f_{1x} + i_{2n} f_{2x} + i_{3n} f_{3x} + i_{4n} f_{4x} - \alpha f_{1x} - \beta f_{2x} + \alpha f_{3x} + \beta f_{4x} \\ &= B_{nx} + \alpha (f_{3x} - f_{1x}) + \beta (f_{4x} - f_{2x}) \\ &= B_{nx} + \alpha g_{1x} + \beta g_{2x}; \end{aligned}$$

where B_{nx} is the x component of the nominal $\mathbf{B}$ at (x, z) and we denote $(f_{3x} - f_{1x})$ as g_{1x}, $(f_{4x} - f_{2x})$ as g_{2x}. Similarly,

$$B_z = B_{nz} + \alpha g_{1z} + \beta g_{2z}; \qquad (4.22)$$

The magnitude of vector $\mathbf{B}$ is

$$B = (B_x^2 + B_z^2)^{\frac{1}{2}} = [(B_{nx} + \alpha g_{1x} + \beta g_{2x})^2 + (B_{nz} + \alpha g_{1z} + \beta g_{2z})^2]^{\frac{1}{2}}, \tag{4.23}$$

then the x component of the field gradient is:

$$\frac{\partial B}{\partial x} = \frac{(B_{nx}+\alpha g_{1x}+\beta g_{2x})(\frac{\partial B_{nx}}{\partial x}+\alpha\frac{\partial g_{1x}}{\partial x}+\beta\frac{\partial g_{2x}}{\partial x})+(B_{nz}+\alpha g_{1z}+\beta g_{2z})(\frac{\partial B_{nz}}{\partial x}+\alpha\frac{\partial g_{1z}}{\partial x}+\beta\frac{\partial g_{2z}}{\partial x})}{[(B_{nx}+\alpha g_{1x}+\beta g_{2x})^2+(B_{nz}+\alpha g_{1z}+\beta g_{2z})^2]^{\frac{1}{2}}}, \tag{4.24}$$

and the z component of the field gradient is:

$$\frac{\partial B}{\partial z} = \frac{(B_{nx}+\alpha g_{1x}+\beta g_{2x})(\frac{\partial B_{nx}}{\partial z}+\alpha\frac{\partial g_{1x}}{\partial z}+\beta\frac{\partial g_{2x}}{\partial z})+(B_{nz}+\alpha g_{1z}+\beta g_{2z})(\frac{\partial B_{nz}}{\partial z}+\alpha\frac{\partial g_{1z}}{\partial z}+\beta\frac{\partial g_{2z}}{\partial z})}{[(B_{nx}+\alpha g_{1x}+\beta g_{2x})^2+(B_{nz}+\alpha g_{1z}+\beta g_{2z})^2]^{\frac{1}{2}}}. \tag{4.25}$$

Let $x_1 = x,\ x_2 = \dot{x},\ x_3 = z,\ x_4 = \dot{z}$, the state equations for a magnetic particle moving in $x - z$ plane are:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = (\frac{p}{m})\frac{\partial B}{\partial x_1} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = (\frac{p}{m})\frac{\partial B}{\partial x_3} \end{cases}$$

where $\frac{\partial B}{\partial x_1}$ and $\frac{\partial B}{\partial x_3}$ are determined by equations 4.24, 4.25.

4.3.5 Controllability of the linearized system

The above system equations can be put into the form:

$$\begin{cases} \dot{x}_1 = x_2 \\ \\ \dot{x}_2 = r(x_1, x_3, \alpha, \beta) \\ \\ \dot{x}_3 = x_4 \\ \\ \dot{x}_4 = q(x_1, x_3, \alpha, \beta) \end{cases}$$

At a target point (x_{1e}, x_{3e}), a direct linearization of the system around the nominal currents gives the following linearized model,

$$\dot{\delta \mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial r}{\partial x_1} & 0 & \frac{\partial r}{\partial x_3} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\partial q}{\partial x_1} & 0 & \frac{\partial q}{\partial x_3} & 0 \end{bmatrix}_{\substack{\mathbf{x}=\mathbf{x}_e \\ \alpha=0 \\ \beta=0}} \delta \mathbf{x} \; + \begin{bmatrix} 0 & 0 \\ \frac{\partial r}{\partial \alpha} & \frac{\partial r}{\partial \beta} \\ 0 & 0 \\ \frac{\partial q}{\partial \alpha} & \frac{\partial q}{\partial \beta} \end{bmatrix}_{\substack{\mathbf{x}=\mathbf{x}_e \\ \alpha=0 \\ \beta=0}} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = A\delta\mathbf{x} \; + \; B \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

where

$$\delta \mathbf{x} = \mathbf{x} - \mathbf{x}_e = \begin{bmatrix} x_1 - x_{1e} \\ x_2 - 0 \\ x_3 - x_{3e} \\ x_4 - 0 \end{bmatrix}.$$

The controllability matrix of this linearized system is

$$\mathbf{C} = [B|AB|A^2B|A^3B],$$

then

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & \frac{\partial r}{\partial \alpha} & \frac{\partial r}{\partial \beta} & 0 & 0 & \cdots & \cdots \\ \frac{\partial r}{\partial \alpha} & \frac{\partial r}{\partial \beta} & 0 & 0 & (\frac{\partial r}{\partial x_1}\frac{\partial r}{\partial \alpha} + \frac{\partial r}{\partial x_3}\frac{\partial q}{\partial \alpha}) & (\frac{\partial r}{\partial x_1}\frac{\partial r}{\partial \beta} + \frac{\partial r}{\partial x_3}\frac{\partial q}{\partial \beta}) & \cdots & \cdots \\ 0 & 0 & \frac{\partial q}{\partial \alpha} & \frac{\partial q}{\partial \beta} & 0 & 0 & \cdots & \cdots \\ \frac{\partial q}{\partial \alpha} & \frac{\partial q}{\partial \beta} & 0 & 0 & (\frac{\partial q}{\partial x_1}\frac{\partial r}{\partial \alpha} + \frac{\partial q}{\partial x_3}\frac{\partial q}{\partial \alpha}) & (\frac{\partial q}{\partial x_1}\frac{\partial r}{\partial \beta} + \frac{\partial q}{\partial x_3}\frac{\partial q}{\partial \beta}) & \cdots & \cdots \end{bmatrix}_{\substack{\mathbf{x}=\mathbf{x}_e \\ \alpha=0 \\ \beta=0}}$$

This controllability matrix has a rank of 4, therefore the linearized system is controllable with the two inputs α and β.

4.3.6 The PID feedback controller

To generate the two controls, a state feedback PID controller is used as following,

$$\alpha = K_p \cdot (z - z_e) + K_i \cdot \textstyle\int_0^t (z - z_e) d\tau + K_d \cdot \dot{z},$$

$$\beta = K_p \cdot (x - x_e) + K_i \cdot \textstyle\int_0^t (x - x_e) d\tau + K_d \cdot \dot{x},$$

where K_p, K_d, K_i are gains for proportional, derivative, and integral control respectively.

5 Simulation

5.1 Position control simulation result

The simulation of this closed-loop system is a computationally intensive work, the simulink models are shown in Fig. 25 and 26. They contain two steps. The first step is the nominal currents searching, where the S-function 'rect4_nominalCurrents' takes x_e, z_e as inputs and obtains four nominal currents for this target point (Fig. 25). The second step is the simulation of the closed-loop control system, where the two controls α and β operate with the four nominal currents separately, generating four currents for the coils (Fig. 26). The S-function 'rectcoil4_cloop_sfun' takes x, z, and $\mathbf{i}$ as its inputs and computes the gradient of B at (x, z).

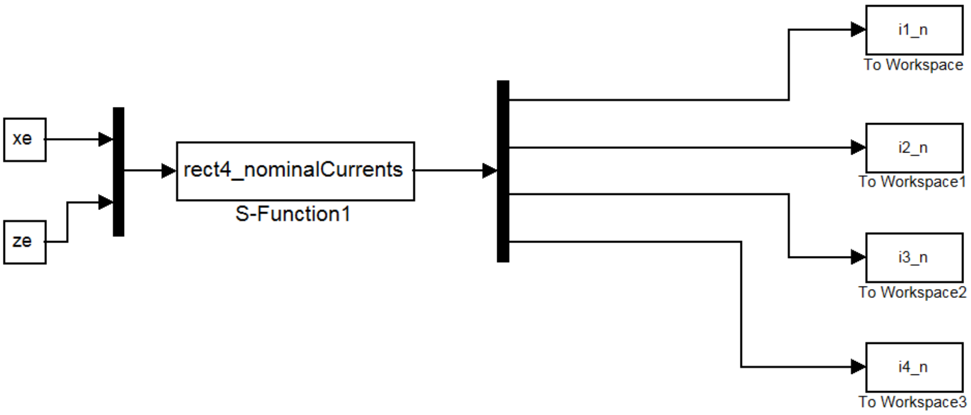

Figure 25: Nominal currents search

It can be seen that under a given ∇B, the acceleration of the particle is only determined by the ratio p/m. In the simulation the particle is assumed to have a property of $p/m = 0.02 wb \cdot m/kg$ (for instance, the mass of the particle could be

Table 2: Coil parameters for the simulation

parameter	value	unit
a	0.1	m
b	0.05	m
c	0.02	m
d	0.04	m
e	0.095	m
N	2500	$no.$
I_c	4	Amp

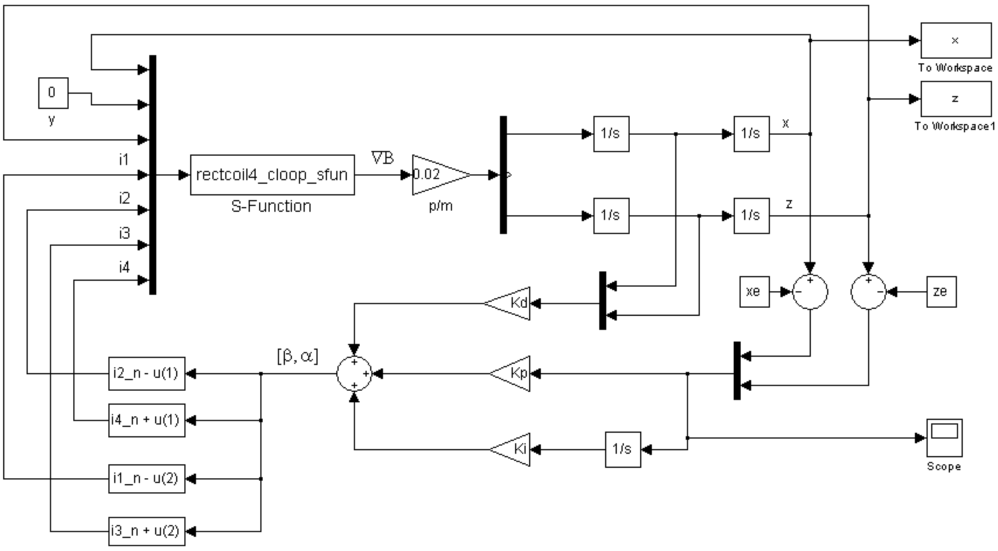

Figure 26: The closed-loop position control system

$m = 1 \times 10^{-6}kg$, the magnetic moment of the particle could be $p = 2 \times 10^{-8}wb \cdot m$). The parameters (defined in Fig. 7 and Fig. 8) of the coils are shown in table 2.

The simulation initial conditions are:

$x0 = -0.01m$,

$z0 = -0.01m$,

$vx0 = 0.01m/s$ (initial velocity in x direction),

$vz0 = 0\ m/s$ (initial velocity in z direction).

The feedback gains are: $K_p = 100, K_d = 10, K_i = 5$.

The simulation result (Fig. 27, Fig. 28) shows that the particle can be stabilized at this target point. In Fig. 28 the z position error $(z - z_e)$ has a larger overshoot.

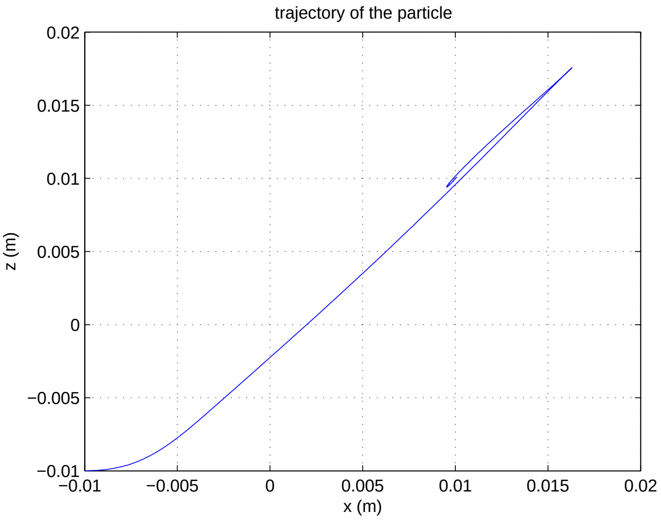

Figure 27: Trajectory of the particle

5.2 Simulation of motion along a circular path

To move the particle along a desired path, the position control scheme can be exploited in a stepwise way. A number of fixed points on the desired path could be selected, then the particle can be moved from one point to the next point using the position control scheme, and step by step, the full path would be fulfilled. The more points selected along the path, the more accurate the fulfilled trajectory would be.

The stepwise path tracking method was simulated for a desired circular path with a radius of 0.01m, centered at the origin. The PID controller parameters are chosen as: $K_p = 100, K_d = 30, K_i = 5$. Here the derivative gain is chosen to be bigger than the derivative gain in the last position control simulation, to suppress the overshoot of each step so that the fulfilled trajectory would be smoother.

For each step, when the particle gets close to the target point and its position error to the target point is below a constant ϵ ($\epsilon = 2\%$ in this simulation), the current step of simulation stops, the target point is set to be the next point on the path, the current position and velocity are set to be the initial values for the next step of simulation, then next step starts.

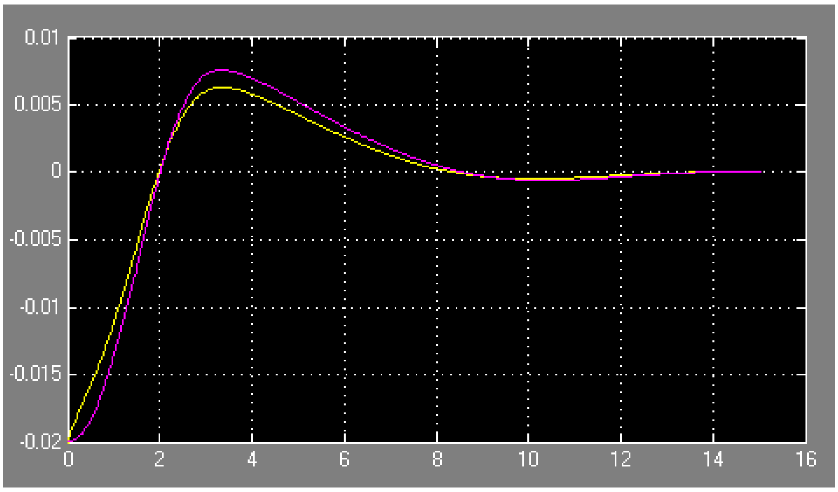

Figure 28: Position error $(x - x_e)$ and $(z - z_e)$ vs time (s)

In this simulation 32 equidistant points are selected on the desired circular path (Fig. 29), the starting position of the particle is at (-0.01, 0), the simulation result is shown in (Fig. 30).

5.3 Limitation on two variables control scheme

There are limitations on the proposed two variables control scheme. First, only in a certain range around the origin in the $x - z$ plane may this method be implemented. If the target point is too far from the origin, there may be no suitable current combination to make this point an equilibrium point. The initial position of the particle should not be too far from the target point, otherwise the linearized model may fail and the system may become uncontrollable.

Secondly, each coil's current is changing by the control α or β about its own nominal current, for example, $i_2 = i_{2n} - \alpha$. If the nominal current $i_{2n} = 2A$ and the control $\alpha > 2A$, then i_1 becomes negative, that means the direction of current of coil 2 will be reversed. The coil with a reversed current will still attract a magnetic particle, and the attraction will get stronger as α gets bigger, then the control scheme will fail. So in implementation, the initial conditions and the feedback gains are all factors which need to be considered to ensure that the controls will not become too big and all currents stay in their original direction.

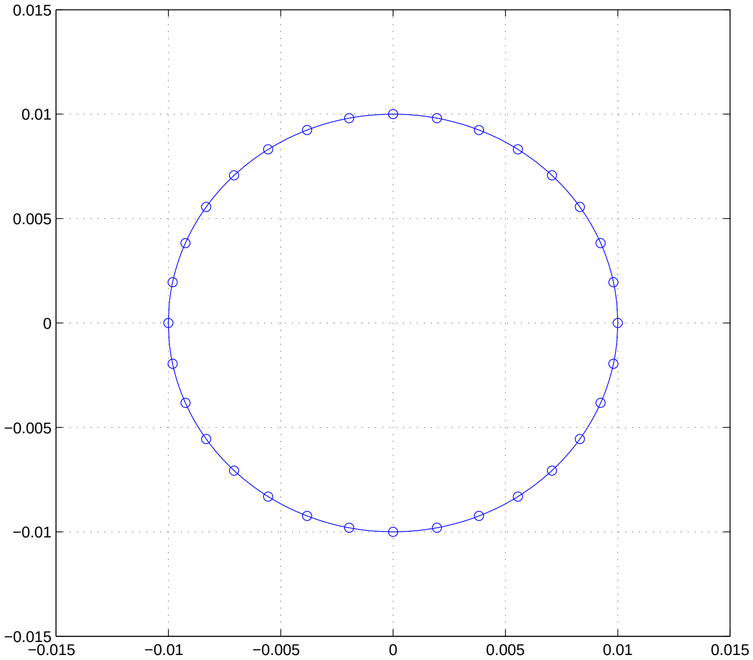

Figure 29: Selected points on a desired circular path

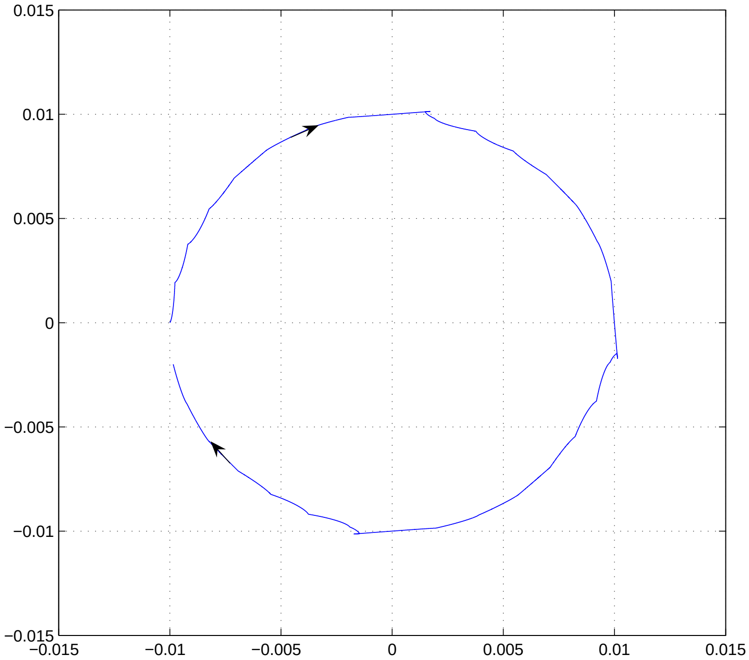

Figure 30: Simulation of motion along a circular path

6 Conclusion

This thesis studies the two dimensional position control problem of a small permanent magnetic particle. External magnetic field generated by four elongated wound coils is used to control the position of the particle. The force imparted on the magnetic particle is directly related to the gradient of the field strength.

A two control variable method is proposed and verified by simulation. In this method a set of nominal currents which make the target position an unstable equilibrium point are searched, then two control variables are applied to the two pairs of the coils respectively. Each control variable varies the currents in each pair in a reciprocal way. The currents varies around each coil's nominal current. A PID feedback controller is used to generate the two control variables. The simulation result shows that this control method can stabilize the particle at a desired point two dimensionally within a certain region of the configuration of the four elongated coils.

The controllability analysis of the system is based on a linearized model where the system is linearized at the target position and with the nominal currents. This causes limitations on the two variable method. The target position should not be too far from the origin, otherwise there may be no suitable current combination to make the target position an equilibrium point. The initial position of the particle should not be too far from the target position, in case the linearized model fails. Furthermore, the initial conditions and the feedback gains need to be considered together to ensure that all currents stay in their nominal directions during the motion of the particle when this method is implemented.

Due to the existence of mutual inductance and self inductance, the realizable current slew rate of the four coils is limited. So the two control inputs α and β should not change too rapidly. A suggestion is that the slew rate of α and β should not exceed $10A/ms$ for coils with a size comparable to the coils used in this thesis.

Coils which have a similar size to the coils in this thesis usually have a resistance less than 100Ω. With a central current set to $1A$, the average power consumption will be less than 400 Watts. This is realizable for a lab environment. So it is suggested that the central current value be set to $1A$ when the control scheme is implemented.

Future work of this study may include:

- Experimental implementation of the control scheme.

- Fluid force needs to be considered in the motion equations, for the simulation of the particle motion in a fluid environment.

- More accurate rectangular coil model could be built by considering fillets on the four corners, and field computation codes need to be rewritten to contain integration on the corners.

- The specific region of controllability of this position control scheme needs to be investigated or tested.

REFERENCES

[1] J.H. Leach, Magnetic Targeted Drug Delivery, Master of Science Thesis, February 2003.

[2] J.B.Mathieu, S.Martel, L.H.Yahia, G.Soulez, G.Beaudoin, MRI systems as a mean of propulsion for a microdevice in blood vessels, *Proceedings of the 25th Annual International Conference of the IEEE EMBS*, pp.3419-3422, 2003

[3] S.Ueno, Recent advances in biomagnetics , *IEEE EMBS Asian-Pacific Conference on Biomedical Engineering 2003*, pp.10-13, Oct. 2003

[4] J.B.Mathieu, S. Martel, L. Yahia, G.Soulez, G.Beaudoin, Preliminary studies for using magnetic resonance imaging systems as a mean of propulsion for microrobots in blood vessels and evaluation of ferromagnetic artefacts , *IEEE Canadian Conference on Electrical and Computer Engineering 2003*, Vol.2, pp.835 - 838, May 2003

[5] T.Nakamura, M.B. Khamesee, A mechanism for three dimensional levitated movement of a small magnet, *IEEE/ASME Transactions on Mechatronics*, vol.2, no.1, pp.41-50, 1997

[6] M.B. Khamesee, T.Nakamura, N. Kato, S. Guo, K. Mizutani, A remote micromanipulator for teleoperation using magnetic levitation, *Micromechatronics and Human Science, Proceedings of the 1997 International Symposium*, pp.169 - 176, 1997

[7] I.J.Busch-Vishniac, S.J.Chen, M.C.Jeong, S.H.Li, I.Y.Wang, Magnetic levitation-based micro-automation of mechanical processes in semiconductor fabrication, *IEEE Micro Electro Mechanical Systems 'Proceedings, An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots'*, pp.142 - 146, 1990

[8] K. S. Jung, Y. S. Baek, Precision stage using a non-contact planar actuator based on magnetic suspension technology, *Mechatronics*, vol.13, no. 8, pp.981-999, 2003

[9] D.C. Meeker, E. H. Maslen, R. C. Ritter, 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

[10] R.G. McNeil, R.C. Ritter,. Wang, M.A. Lawson, G.T. Gillies, et al, 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

[11] R.G. McNeil, R.C. Ritter,. Wang, M.A. Lawson, G.T. Gillies, et al, Characteristics of an improved magnetic-implant guidance system, *IEEE Transactions on Biomedical Engineering*, vol.42, no. 8, pp.802 - 808 1995

[12] N. Ida, J.P. Bastos, *Electromagnetics and calculation of fields*, New York, NY: Springer, 1997.

[13] R.M.Whitmer, *Electromagnetics*, Englewood Cliffs, NJ: Prentice-Hall, 1962

[14] G.M.Phillips, P.J.Taylor, *Theory and Applications of numerical analysis,* San Diego, CA: Academic press, 1996

[15] R.J.Vaccaro, *Digital control a state-space approach*, McGraw-Hill, 1995.

[16] C.S.Lee, H.Lee, R.M.Westervelt, Microelectromagnets for the control of magnetic nanoparticles, *Applied Physics Letters*, vol.79, no.20, pp.3308 - 3311, 2001

[17] Q.A.Pankhurst, J. Connolly, S.K.Jones, J.Dobson, Application of magnetic nanoparticles in biomedicine, *Journal of Physics D: Applied Physics*, vol.36, pp.R167 - R181, 2003

[18] R. R. Saldanha, S. Pelissier, K. Kadded, Y. P. Yonnet, J.L. Coulomb, Nonlinear Optimization Methods Applied to Magnetic Actuators Design, IEEE Transactions ON Magnetics, vol. 28, no.2, pp.1581-1584, 1992

[19] S.Østergaard, G. Blankenstein, H.Dirac, 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

[20] M.G.Duvall, Magnetic Particle Motion by Field Manipulation, Qualifying Examination Report, pp.15-17, 2005

A Appendix

$$f_{1x} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{z_0-e-z}{[(x_0-x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}} - \frac{z_0-e-z}{[(x_0+x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

$$f_{2x} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{-\frac{b}{2}-c}^{-\frac{b}{2}}\int_{y-\frac{a-b}{2}}^{\frac{a-b}{2}-y}\{\frac{-y}{[(-z_0-x)^2+y^2+(x_0-e-z)^2]^{\frac{3}{2}}} - \frac{-y}{[(-z_0+x)^2+y^2+(x_0-e-z)^2]^{\frac{3}{2}}}\}dxdydz$$

$$+k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{x+z_0}{[(-z_0-x)^2+y^2+(x_0-e-z)^2]^{\frac{3}{2}}} - \frac{x+z_0}{[(-z_0-x)^2+y^2+(x_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

$$f_{3x} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{z_0+e+z}{[(-x_0-x)^2+y^2+(-z_0-e-z)^2]^{\frac{3}{2}}}-\frac{z_0+e+z}{[(-x_0+x)^2+y^2+(-z_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

$$f_{4x} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{-\frac{b}{2}-c}^{-\frac{b}{2}}\int_{y-\frac{a-b}{2}}^{\frac{a-b}{2}-y}\{\frac{-y}{[(z_0-x)^2+y^2+(-x_0-e-z)^2]^{\frac{3}{2}}} - \frac{-y}{[(z_0+x)^2+y^2+(-x_0-e-z)^2]^{\frac{3}{2}}}\}dxdydz$$

$$+k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{x-z_0}{[(z_0-x)^2+y^2+(-x_0-e-z)^2]^{\frac{3}{2}}} - \frac{x-z_0}{[(z_0+x)^2+y^2+(-x_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

$$f_{1z} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{-\frac{b}{2}-c}^{-\frac{b}{2}}\int_{y-\frac{a-b}{2}}^{\frac{a-b}{2}-y}\{\frac{-y}{[(x_0-x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}} - \frac{-y}{[(x_0+x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}}\}dxdydz$$

$$+k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{x-x_0}{[(x_0-x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}} - \frac{x-x_0}{[(x_0+x)^2+y^2+(z_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

$$f_{2z} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{-x_0+e+z}{[(-z_0-x)^2+y^2+(x_0-e-z)^2]^{\frac{3}{2}}} - \frac{-x_0+e+z}{[(-z_0+x)^2+y^2+(x_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

$$f_{3z} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{-\frac{b}{2}-c}^{-\frac{b}{2}}\int_{y-\frac{a-b}{2}}^{\frac{a-b}{2}-y}\{\frac{y}{[(-x_0-x)^2+y^2+(-z_0-e-z)^2]^{\frac{3}{2}}}-\frac{y}{[(-x_0+x)^2+y^2+(-z_0-e-z)^2]^{\frac{3}{2}}}\}dxdydz$$

$$+k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{-x-x_0}{[(-x_0-x)^2+y^2+(-z_0-e-z)^2]^{\frac{3}{2}}}-\frac{-x-x_0}{[(-x_0+x)^2+y^2+(-z_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

$$f_{4z} = k\int_{-\frac{d}{2}}^{\frac{d}{2}}\int_{\frac{a}{2}}^{\frac{a}{2}+c}\int_{\frac{a-b}{2}-x}^{x-\frac{a-b}{2}}\{\frac{-x_0-e-z}{[(z_0-x)^2+y^2+(-x_0-e-z)^2]^{\frac{3}{2}}}-\frac{-x_0-e-z}{[(z_0+x)^2+y^2+(-x_0-e-z)^2]^{\frac{3}{2}}}\}dydxdz$$

Source code

rectcoil4_cloop_sfun.cpp:

```
#include "stdio.h"
#include "rect_coil.h"
#include "math.h"

struct Bfield{              //the magnetic field intensity B
     double x;
     double y;
     double z;
            };

struct P_position{          //the position of the particle
     double x;
     double y;
     double z;
            };

//----------------------------------rect_coil class-----------------------------------------------
class rect_coil {

 protected:
    double inner_width;
    double inner_length;
    double coil_xsection_width;
    double coil_xsection_height;
```

```
   int Num_turns;
   double i_n;         //the central current
   double i_density;
   double I;           //current of subdivided circular loop

private:
   double x;
   double y;
   double z;
   double x0;   //the coordinates of the point to be calculated
   double y0;
   double z0;
   double a;
   double b;
   double ra1;  //current_ratio of coil1
   double ra2;  //current_ratio of coil2
   double ra3;  //current_ratio of coil3
   double ra4;  //current_ratio of coil4

 double func_Bx_fore(double l) //the integrant function for Bx, fore segment for a single loop
 {
     return 0;
 }

 double func_By_fore(double l) // for By, fore segment
 {
   return -I*z/(10000000*pow((x-l)*(x-l)+(y+b)*(y+b)+z*z,1.5));  //'mu0/4pi'=10^-7
 }

 double func_Bz_fore(double l) // for Bz, fore segment
 {
   return I*(y+b)/(10000000*pow((x-l)*(x-l)+(y+b)*(y+b)+z*z,1.5));
 }

 double func_Bx_right(double l) // for Bx, right segment
 {
   return I*z/(10000000*pow((x-a)*(x-a)+(y-l)*(y-l)+z*z,1.5));
 }

 double func_By_right(double l) // for By, right segment
 {
   return 0;
 }
```

```
  double func_Bz_right(double l) // for Bz, right segment
  {
    return I*(a-x)/(10000000*pow((x-a)*(x-a)+(y-l)*(y-l)+z*z,1.5));
  }

double half_loop_Bx(double a_loop, double b_loop)
//integration using Simpson's rule, to get the Bx generated by the fore and right segment
{
    a=a_loop;   //a, b are the geometric parameters of the rect. subdivided coil (wire)
    b=b_loop;   //a, b will be used inside the functions func_Bx_fore(double l) and so on
    int k;
    double integ=0.0;    //integration result with fore and right segments
    double dx=2*b/Num_subdiv_right; //dx is the dL in the Biot-Savart law
    double Sub_integ[Num_subdiv];   //Num_subdiv is the subdivision number along the fore segment
    //Num_subdiv_right is the subdiv. number along the right segment, Num_subdiv_right<= Num_subdiv.

    for(k=0;k<Num_subdiv_right;k++)
    {
    Sub_integ[k]=(dx/6)*(func_Bx_right(-b+k*dx)+4*func_Bx_right(((-b+k*dx)+(-b+(k+1)*dx))/2)
                  +func_Bx_right(-b+(k+1)*dx));
    integ=integ+Sub_integ[k];
    }
    return integ;
}

double  half_loop_By(double a_loop, double b_loop)
//integration using Simpson's rule, to get the By generated by the fore and right segment
{
    a=a_loop;   //a, b are the geometric parameters of the rect. subdivided coil (wire)
    b=b_loop;   //a, b will be used inside the functions func_Bx_fore(double l) and so on
    int k;
    double integ=0.0;    //integration result with fore and right segments
    double dx=2*a/Num_subdiv;
    double Sub_integ[Num_subdiv];    //Num_subdiv is the subdivision number along the fore segment

    for(k=0;k<Num_subdiv;k++)
    {
        Sub_integ[k]=(dx/6)*(func_By_fore(-a+k*dx)+4*func_By_fore(((-a+k*dx)+(-a+(k+1)*dx))/2)
                    +func_By_fore(-a+(k+1)*dx));
        integ=integ+Sub_integ[k];
    }
    return integ;
```

```
}

double  half_loop_Bz(double a_loop, double b_loop)
//integration using Simpson’s rule, to get the Bz generated by the fore and right segment
{
    a=a_loop;   //a, b are the geometric parameters of the rect. subdivided coil (wire)
    b=b_loop;   //a, b will be used inside the functions func_Bx_fore(double l) and so on
    int k;
    double integ=0.0;    //integ. with fore and right segments
    double dx=2*a/Num_subdiv;
    double Sub_integ[Num_subdiv];   //Num_subdiv is the subdivision number along the fore segment
    //Num_subdiv_right is the subdiv. number along the right segment, Num_subdiv_right<= Num_subdiv.

    for(k=0;k<Num_subdiv;k++)
    {
        Sub_integ[k]=(dx/6)*(func_Bz_fore(-a+k*dx)+4*func_Bz_fore(((-a+k*dx)+(-a+(k+1)*dx))/2)
                      +func_Bz_fore(-a+(k+1)*dx));
        integ=integ+Sub_integ[k];
    }

    dx=2*b/Num_subdiv_right;
    for(k=0;k<Num_subdiv_right;k++)
    {
        Sub_integ[k]=(dx/6)*(func_Bz_right(-b+k*dx)+4*func_Bz_right(((-b+k*dx)+(-b+(k+1)*dx))/2)
                      +func_Bz_right(-b+(k+1)*dx));
        integ=integ+Sub_integ[k];
    }
    return integ;
}

double  other_half_loop_Bx(double a, double b, double c, double
  a_loop, double b_loop)
{
  x=-a;   //change coordinates
  y=-b;
  z=c;
  return -half_loop_Bx(a_loop, b_loop);
          //Bx of the other half loop, the minus sign change it back to the current coordinates.
}

double  other_half_loop_By(double a,double b, double c, double
  a_loop, double b_loop)
{
  x=-a;   //change coordinates
  y=-b;
  z=c;
  return -half_loop_By(a_loop, b_loop);
          //By of the other half loop, the minus sigh change it back to the current coordinates.
}

double  other_half_loop_Bz(double a,double b, double c, double
  a_loop,double b_loop)
{
  x=-a;   //change coordinates
  y=-b;
  z=c;
  return half_loop_Bz(a_loop, b_loop);
          //Bz of the other half loop, has the same sign as in the current coordinates.
}

public:
////--------------------------B calculation for a single rect coil---------------------------------

double  total_Bx(double a,double b, double c) {
  double a_loop,b_loop;

  x=a;
  y=b;
  double Bx=0.0; //temporary variable for Bx
  int j;
  int k;

  for(j=0;j<num_coil_subdiv*(coil_xsection_height/coil_xsection_width);j++)
  {
   z=c+(coil_xsection_height/2)-(coil_xsection_width/num_coil_subdiv)*(0.5+j);
                                      //z coordinate according to the subdivided loop
   for(k=0;k<num_coil_subdiv;k++)     //num_coil_subdive is along the width of the cross-section
   {
    a_loop=(inner_length/2)+(coil_xsection_width/num_coil_subdiv)*(0.5+k);
    b_loop=(inner_width/2)+(coil_xsection_width/num_coil_subdiv)*(0.5+k);
    Bx=Bx+half_loop_Bx(a_loop,b_loop)+other_half_loop_Bx(x,y,z,a_loop,b_loop);
   }
  }
  return Bx;
}
```

```
double  total_By(double a,double b, double c)
{
  double a_loop,b_loop;
  x=a;
  y=b;
  double By=0.0; //temporary variable for By
  int j;
  int k;

  for(j=0;j<num_coil_subdiv*(coil_xsection_height/coil_xsection_width);j++)
  {
   z=c+(coil_xsection_height/2)-(coil_xsection_width/num_coil_subdiv)*(0.5+j);
                                   //z coordinate according to the subdivided loop
   for(k=0;k<num_coil_subdiv;k++)     //num_coil_subdive is along the width of the cross-section
   {
    a_loop=(inner_length/2)+(coil_xsection_width/num_coil_subdiv)*(0.5+k);
    b_loop=(inner_width/2)+(coil_xsection_width/num_coil_subdiv)*(0.5+k);
    By=By+half_loop_By(a_loop,b_loop)+other_half_loop_By(x,y,z,a_loop,b_loop);
   }
  }
  return By;
}

double  total_Bz(double a,double b, double c) {
  double a_loop,b_loop;
  x=a;
  y=b;
  double Bz=0.0; //temporary variable for Bz
  int j;
  int k;

  for(j=0;j<num_coil_subdiv*(coil_xsection_height/coil_xsection_width);j++)
  {
   z=c+(coil_xsection_height/2)-(coil_xsection_width/num_coil_subdiv)*(0.5+j);
                                   //z coordinate according to the subdivided loop
   for(k=0;k<num_coil_subdiv;k++)     //num_coil_subdive is along the width of the cross-section
   {
    a_loop=(inner_length/2)+(coil_xsection_width/num_coil_subdiv)*(0.5+k);
    b_loop=(inner_width/2)+(coil_xsection_width/num_coil_subdiv)*(0.5+k);
    Bz=Bz+half_loop_Bz(a_loop,b_loop)+other_half_loop_Bz(x,y,z,a_loop,b_loop);
   }
  }
  return Bz;
}

//////---B calculation in global coordinates system, using coordinates translation and rotation------

Bfield  B_global(double x0, double y0, double z0, double xc, double yc, double zc, double alpha)
//(x0,y0,z0) is the point in global coords to be calculated, (xc,yc,zc) is the center of the coil
//alpha is the rotation angle of the coil (around the line passing the center and parallel to y axis)
{
  double x_loc,y_loc,z_loc;    //local coordinates
  Bfield B_loc;
  Bfield B;    // B = {Bx_glo,By_glo,Bz_glo}

  x_loc=(x0-xc)*cos(alpha)-(z0-zc)*sin(alpha);
  y_loc=(y0-yc);
  z_loc=(x0-xc)*sin(alpha)+(z0-zc)*cos(alpha);
  B_loc.x=total_Bx(x_loc, y_loc,z_loc);    //B value in local coord. system
  B_loc.y=total_By(x_loc, y_loc,z_loc);
  B_loc.z=total_Bz(x_loc, y_loc,z_loc);
  B.x= (B_loc.x*cos(alpha)+B_loc.z*sin(alpha));    //B value in global coord. system
  B.y= B_loc.y;
  B.z= (-B_loc.x*sin(alpha)+B_loc.z*cos(alpha));

  return B;
}

Bfield  B4coils(double x0, double y0, double z0)
  //compute the field generated by the 4 rect. coils
{
    Bfield B4coils;
    Bfield B1, B2, B3, B4;

    B1=B_global(x0,y0,z0, 0, 0, 0.09, 0);  //0.09 is the z coordinate of the center of the 1st coil
    B2=B_global(x0,y0,z0, 0.09*sin(PI/2), 0, 0.09*cos(PI/2), PI/2);
    B3=B_global(x0,y0,z0, 0.09*sin(2*PI/2), 0, 0.09*cos(2*PI/2), 2*PI/2);
    B4=B_global(x0,y0,z0, 0.09*sin(3*PI/2), 0, 0.09*cos(3*PI/2), 3*PI/2);
    B4coils.x= B1.x*ra1 + B2.x*ra2 + B3.x*ra3 + B4.x*ra4;
    B4coils.y= B1.y*ra1 + B2.y*ra2 + B3.y*ra3 + B4.y*ra4;
    B4coils.z= B1.z*ra1 + B2.z*ra2 + B3.z*ra3 + B4.z*ra4;

    return B4coils;
}
```

```
double  Bsum4coils(double x0, double y0, double z0)
  //compute the field magnitude generated by the 4 rect. coils
{
    double Bsum;
    Bfield B4_coils;

    B4_coils=B4coils(x0,y0,z0);
    Bsum= sqrt(B4_coils.x*B4_coils.x + B4_coils.y*B4_coils.y + B4_coils.z*B4_coils.z);
    return Bsum;
}

double  gradB4coils_x(double a, double b, double c,double i1, double i2, double i3, double i4)
  //return the x component of the gradient
{
  ra1=i1/i_n;
  ra2=i2/i_n;
  ra3=i3/i_n;
  ra4=i4/i_n;

  return (Bsum4coils(a+h,b,c )-Bsum4coils(a-h,b,c ))/(2*h);
}

double  gradB4coils_y(double a, double b, double c, double i1, doublei2, double i3, double i4)
  //return the y component of the gradient
{
  ra1=i1/i_n;
  ra2=i2/i_n;
  ra3=i3/i_n;
  ra4=i4/i_n;

  return (Bsum4coils(a,b+h,c )-Bsum4coils(a,b-h,c ))/(2*h);
}

double  gradB4coils_z(double a, double b, double c,double i1, double i2, double i3, double i4)
  //return the z component of the gradient
{
  ra1=i1/i_n;
  ra2=i2/i_n;
  ra3=i3/i_n;
  ra4=i4/i_n;

  return (Bsum4coils(a,b,c+h )-Bsum4coils(a,b,c-h ))/(2*h);
}
```

```
//--------------------------------constructor of rect_coil----------------------------------
rect_coil() {
  inner_length=0.1;
  inner_width=0.05;
  coil_xsection_width=0.02;
  coil_xsection_height=0.04;
  Num_turns=2500;
  i_n=4;          //i_n is the central current

  i_density
  =i_n*Num_turns/(coil_xsection_width*coil_xsection_height);//coil's current density

  //current of subdivided circular loop, ampere
  I=i_density*pow(double(coil_xsection_width/num_coil_subdiv),2.0);
}
};   //--------------------------end of the class 'rect_coil'----------------------------------


#ifdef __cplusplus extern "C" { // use the C fcn-call standard for all functions
#endif       // defined within this scope
#define S_FUNCTION_LEVEL 2
#define S_FUNCTION_NAME rectcoil4_cloop_sfun

/*
 * Need to include simstruc.h for the definition of the SimStruct and
 * its associated macro definitions.
 */
#include "simstruc.h"

#define IS_PARAM_DOUBLE(pVal) (mxIsNumeric(pVal) &&
!mxIsLogical(pVal) &&\ !mxIsEmpty(pVal) && !mxIsSparse(pVal) &&
!mxIsComplex(pVal) && mxIsDouble(pVal))



/*====================*
 * S-function methods *
 *====================*/

#define MDL_CHECK_PARAMETERS #if defined(MDL_CHECK_PARAMETERS)  &&
defined(MATLAB_MEX_FILE) /*
 * Check to make sure that each parameter is 1-d and positive
 */
```

```
static void mdlCheckParameters(SimStruct *S) {
 /*
    const mxArray *pVal0 = ssGetSFcnParam(S,0);
    if ( !IS_PARAM_DOUBLE(pVal0)) {
        ssSetErrorStatus(S, "Parameter to S-function must be a double scalar");
        return;
    }
 */
} #endif


/* Function: mdlInitializeSizes
===============================================
 * Abstract:
 *    The sizes information is used by Simulink to determine the S-function
 *    block's characteristics (number of inputs, outputs, states, etc.).
 */
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, 0);  /* Number of expected parameters */

#if defined(MATLAB_MEX_FILE)
    if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S)) {
        mdlCheckParameters(S);
        if (ssGetErrorStatus(S) != NULL) {
            return;
        }
    } else {
        return; /* Parameter mismatch will be reported by Simulink */
    }
#endif

    ssSetNumContStates(S, 0);
    ssSetNumDiscStates(S, 0);

    if (!ssSetNumInputPorts(S, 1)) return;

    ssSetInputPortWidth(S, 0, 7);
    ssSetInputPortDirectFeedThrough(S, 0, 1);

    if (!ssSetNumOutputPorts(S, 1)) return;

    ssSetOutputPortWidth(S, 0, 3);
    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, 0);
    ssSetNumIWork(S, 0);
    ssSetNumPWork(S, 1); // reserve element in the pointers vector
    ssSetNumModes(S, 0); // to store a C++ object
    ssSetNumNonsampledZCs(S, 0);
    ssSetOptions(S, 0);
}

/* Function: mdlInitializeSampleTimes
=========================================
 * Abstract:
 *    This function is used to specify the sample time(s) for your
 *    S-function. You must register the same number of sample times as
 *    specified in ssSetNumSampleTimes.
 */
static void mdlInitializeSampleTimes(SimStruct *S) {
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}



#define MDL_START  /* Change to #undef to remove function */
#if defined(MDL_START)
  /* Function: mdlStart =======================================================
   * Abstract:
   *    This function is called once at start of model execution.
   */
  static void mdlStart(SimStruct *S)
  {
      ssGetPWork(S)[0] = (void *) new rect_coil; // store new C++ object in the
  }                                            // pointers vector
#endif /*  MDL_START */

/* Function: mdlOutputs
=======================================================
 * Abstract:
 *    In this function, you compute the outputs of your S-function
 *    block. Generally outputs are placed in the output vector, ssGetY(S).
 */
static void mdlOutputs(SimStruct *S, int_T tid)
{
    InputRealPtrsType uPtrs  = ssGetInputPortRealSignalPtrs(S,0);
    rect_coil *rc = (rect_coil *) ssGetPWork(S)[0];   // retrieve C++ object from
    real_T  *y = ssGetOutputPortRealSignal(S,0); // the pointers vector and use

    y[0]= rc->gradB4coils_x(*uPtrs[0],*uPtrs[1],*uPtrs[2],*uPtrs[3],*uPtrs[4],*uPtrs[5],*uPtrs[6]);
                                                                  // member functions of the object
    y[1]= rc->gradB4coils_y(*uPtrs[0],*uPtrs[1],*uPtrs[2],*uPtrs[3],*uPtrs[4],*uPtrs[5],*uPtrs[6]);
    y[2]= rc->gradB4coils_z(*uPtrs[0],*uPtrs[1],*uPtrs[2],*uPtrs[3],*uPtrs[4],*uPtrs[5],*uPtrs[6]);
}

/* Function: mdlTerminate
=====================================================
 * Abstract:
 *    In this function, perform any actions that are necessary at
 *    the termination of a simulation.  For example, if memory was
 *    allocated in mdlStart, this is the place to free it.
 */
static void mdlTerminate(SimStruct *S) {
    rect_coil *rc = (rect_coil *) ssGetPWork(S)[0]; // retrieve and destroy C++
    delete rc;                                      // object in the termination
}                                                   // function

/*=============================*
 * Required S-function trailer *
 *=============================*/

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
#ifdef __cplusplus }       // end of extern "C" scope
#endif
```

the function 'nominal_currents' in rect4_nominalCurrents.cpp:

```
Currents  nominal_currents(double xe, double ze)  //only in x-z plane
{
    double oldStepSize, newStepSize;
    int zSteps;
    int xSteps;
    Currents nominalCurrents;
    double i1,i2,i3,i4,i1_n,i2_n,i3_n,i4_n; //i1_n... are currents that
                                            //generate the min. B
    double B_min, Bsum;
    int j,k;

    i1_n= i_n;
    i2_n= i_n;
    i3_n= i_n;
    i4_n= i_n;         //initialize the four nominal currents

    ra1=1;
    ra2=1;
    ra3=1;
    ra4=1;
    B_min= Bsum4coils(xe,0,ze);  //initialize B_min

    if(xe==0 && ze==0)
         {
          nominalCurrents.i1=i_n;
          nominalCurrents.i2=i_n;
          nominalCurrents.i3=i_n;
          nominalCurrents.i4=i_n;
          return nominalCurrents;   //if (xe,ze) is the origin, the four
                                    //are all i_n.
         }
    else
    {
     double i1_temp= i_n;
     double i2_temp= i_n;
     double i3_temp= i_n;
     double i4_temp= i_n;      //these temporary currents are used to record those currents
                               //for smaller B.
     oldStepSize= i_n;         //initial oldStepSize
     newStepSize= i_n/2;

     while(true)
     {
      zSteps = int(2*oldStepSize/newStepSize);  // =4
      xSteps = int(2*oldStepSize/newStepSize);

      for(j=1;j<zSteps;j++)
       {
        for (k=1;k<xSteps;k++)
        {
         i1= (i1_n-oldStepSize)+j*newStepSize;
         i3= (i3_n+oldStepSize)-j*newStepSize;
         i2= (i2_n-oldStepSize)+k*newStepSize;
         i4= (i4_n+oldStepSize)-k*newStepSize;

         ra1=i1/i_n;
         ra2=i2/i_n;
         ra3=i3/i_n;
         ra4=i4/i_n;

         //compute B under these current ratios
         Bsum= Bsum4coils(xe,0,ze);

         if(Bsum<B_min)
         {
           B_min= Bsum;
           i1_temp= i1;
           i2_temp= i2;
           i3_temp= i3;
           i4_temp= i4;   //record the currents for this smaller Bsum.

          if( Bsum<Bsum4coils(xe+h,0,ze)
           && Bsum<Bsum4coils(xe-h,0,ze)
           && Bsum<Bsum4coils(xe,0,ze+h)
           && Bsum<Bsum4coils(xe,0,ze-h) )
           {
            nominalCurrents.i1=i1;
            nominalCurrents.i2=i2;
            nominalCurrents.i3=i3;
            nominalCurrents.i4=i4;
            return nominalCurrents; //if gradB==0,wet get it,and return.
           }
         }
       }  //end of 'for k' loop
      }   //end of 'for j' loop

        //----- if no currents meet the '0 gradB' criterion--------
        i1_n= i1_temp;
        i2_n= i2_temp;
        i3_n= i3_temp;
```

```
        i4_n= i4_temp;   //renew those currents which generate min.gradB

        oldStepSize = newStepSize;     //current StepSize became oldStepSize
        newStepSize = oldStepSize/2;   //new stepsize for next ’while’ loop

    }   //end of the ’while’ loop
  }  //end of ’else’
}
```

VITA

Wenjie Li

Candidate for the Degree of

Master of Science

Thesis: MAGNETIC PARTICLE TWO DIMENSIONAL POSITION CONTROL USING MAGNETIC FIELD

Major Field: Mechanical Engineering

Biographical:

Education: Received a Bachelor of Science degree in Engineering from Sichuan University, Chengdu, China in July 1993. Completed the requirements for the Master of Science degree with a major in Mechanical Engineering at Oklahoma State University in May, 2006

Experience: Employed by Institute of Applied Lasers, Sichuan University, Chengdu, China; employed by Department of Mechanical and Aerospace Engieering, Oklahoma State University as a graduate teaching assistant.

Professional Memberships:

Name: Wenjie Li Date of Degree: May, 2006

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: MAGNETIC PARTICLE TWO DIMENSIONAL POSITION CONTROL USING MAGNETIC FIELD

Pages in Study: 55 Candidate for the Degree of Master of Science

Major Field: Mechanical Engineering

Scope and Method of Study: The goal of this study is to control the position of a small permanent magnetic particle two dimensionally by means of manipulating external magnetic field strength in the vicinity of the particle. The magnetic field is generated by four elongated wound coils, and the currents of the coils are used as control inputs. The force imparted on the magnetic particle is directly related to the gradient of the field strength. The magnetic field strength and its gradient are computed numerically. A Simulink model is built for the simulation of the motion of the particle.

Findings and Conclusions: A two control variable method is proposed and verified by the simulation. In this method two control variables are applied to the two pairs of the coils respectively, to vary the currents of the coils in each pair reciprocally. A PID feedback controller is used to generate the two controls. The simulation result shows that this control method can stabilize the particle at a desired point two dimensionally within a certain region of the configuration of the four elongated coils.

ADVISERS' APPROVAL: Eduardo A. Misawa