Optimal Solutions to the Inverse Problem in Quadratic Magnetic Actuators

A dissertation presented to the faculty

of the

School of Engineering and Applied Science

University of Virginia

In partial fulfillment

of the requirements

for the degree

Doctor of Philosophy, Mechanical, Aerospace, and Nuclear Engineering

by

David C. Meeker

May 1996

Approval Sheet

The dissertation is submitted in partial fulfillment of the requirements for the degree of:

Doctor of Philosophy, Mechanical, Aerospace, and Nuclear Engineering

Author

This dissertation has been read and approved by the Examining Committee:

Dissertation advisor

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering and Applied Science

May, 1996

Abstract

The formulation of current-to-force relationships for magnetic actuators proceeds in a fairly straightforward fashion from Maxwell's equations for magnetostatic problems. However, the inverse problem of determining a set of currents to realize a desired force is less well understood. Historically, this problem has been relatively neglected because actuators were built in symmetric geometries where a viable solution could be intuited. Recently, calls for both optimal actuator performance and fault tolerance have necessitated the formulation of general solution methodologies for magnetic actuators. This dissertation explores such formulations for magnetic actuators whose current-to-force relationships are homogeneous quadratics. Two inverse strategies are considered: a *generalized bias linearization* approach that yields solutions which are easily implemented and fault-tolerant; and a *direct optimal* approach that realizes low power loss. The examples of the class of actuators addressed are radial magnetic bearings and the magnetic stereotaxis system.

Acknowledgments


```
credit  n

4  favorable notice or attention resulting from an action or achievement
        <took all the credit for the idea>
   syn    acknowledgment, recognition
   rel    attention, notice; distinction, fame, honor; glory, kudos
```

How many ways can I say it?

To Eric Maslen for being an advisor and a friend rather than a boss.

To the rest of my committee: Miles Townsend, Hossein Haj-Hariri, Carl Knospe and Gang Tao. Thanks for your help and encouragement along the way, and for putting in a good word for me now and again.

To Chris Sortore for always answering the phone.

To Daniel Noh for his help and insight on a number of different projects.

To Tana Herndon, Cathy Dixon, and Tammy Ramsey for making sure that I have all the right forms filled out.

To Sri Gopalakrishna and Amish Thaker for swapping unix secrets with me.

To A. Peter Allan for cool ideas and strong martinis.

And of course, my love and thanks to my wife, Aimee Dalrymple, for putting up with me (and Arrington) through all of this.

Contents

List of Figures

Nomenclature

a .... Pole face area

A .... Diagonal matrix of pole face areas

A .... Anti-symmetric matrix that reduces the rank of a corresponding M matrix

b .... Column matrix of flux densities

$\underline{b}$ .... Non-dimensional flux density

$\mathbf{b}$ .... Vector flux density

B .... Component of the decomposition of the bias linearization problems that is common to all force directions. In the MSS, B has the physical interpretation of relating coil current to flux density at the seed's location.

BH ...... Function characterizing the B–H curve for a given material.

C_a ...... Collection of the rows of V and V_s corresponding to active constraints in the direct optimization method

d .... Right-hand side of the decomposed bias linearization problem

D .... Force direction-specific component of the decomposition of the bias linearization problem. In the MSS, D has the physical interpretation of derivative of B with respect to spatial coordinates.

D .... Matrix defined as $\mathsf{D} \equiv [D_1'm|D_2'm|D_3'm]^T$

e .... Column matrix whose entries are the element by element square of i

E .... Energy stored in the magnetic field of a magnetic bearing

f .... Column matrix of forces produced by an actuator

$\mathbf{f}$ .... Vector of force on a dipole

F .... A vector function that defines the bias linearization problem

F_{kt} ...... Kuhn-Tucker conditions

$\underline{f}$ .... Non-dimensional force

g .... Air gap length between pole tip and journal surface in a radial magnetic bearing

g_o .... Nominal air gap length

G .... Matrix characterizing the decomposed bias linearization problem

G .... Matrix defined by $G \equiv [M_1 + A_1| \cdots |M_k + A_k]$

h .... Magnetic field intensity

H .... Matrix whose j^{th} row is $i' M_j$

H .... Matrix analagous to H when the bias linearization problem is re-written in terms of an unknown vector rather than an unknown matrix

i .... Coil current vector
$\underline{i}$ .... Nondimensional coil current
i_o .... Non-zero current vector that creates no force
I .... Identity matrix
$\hat{i}$ .... Bias current scaling vector
J .... Cost function for coil currents
$\hat{J}$ .... Modified cost function including Lagrange multipliers
k .... Number of force directions
K .... A matrix whose rows are the diagonals of the M matrices associated with a given actuator
$\mathtt{K}$ .... Matrix relating reduced order to full current vector
L .... Inductance matrix
$\mathbf{m}$ .... Dipole moment vector
m .... Dipole moment column matrix
M .... Symmetric matrix characterizing the relation between applied current and resulting force
$\hat{M}$ .... Symmetric matrix characterizing the relationship between $\hat{i}$ and force
n .... Number of currents supplied to the actuator
$\mathbf{n}$ .... Unit vectors defining a fixed coordinate frame in the MSS
N .... Number of turns in a magnetic bearing winding
$\mathtt{N}$ ...... Windings matrix
$\mathbf{p}$ .... Position vector for a specified point in space
P .... Matrix whose columns span a totally isotropic space corresponding to all M matrices for an actuator
Q .... Matrix pencil formed from M_1 and M_2.
$\mathtt{Q}$ .... Positive definite current weighting matrix
$\mathtt{R}$ .... Air gap reluctance
R .... Air gap reluctance matrix
$\Re$ .... The set of real numbers
$\mathbf{t}$ .... Vector of torque on a magnetic seed
T .... Orthonormal coordinate transformation matrix
$\mathtt{U}$ .... An orthogonal matrix resulting from a singular value decomposition
V .... Matrix relating applied current to flux in the air gaps of a magnetic bearing
$\mathtt{V}$ .... An orthogonal matrix resulting from a singular value decomposition
W .... Bias linearization current matrix
$\mathtt{W}$ ...... Set of all possible forces that can be produced by a given actuator
w .... Dimension of the maximal totally isotropic space of a symmetric matrix
$\hat{w}$ .... Dimension of the null space common to all M matrices for a given actuator
α .... Ratio of flux density to force
γ .... Dipole moment magnitude
ζ .... Scaling factor for biasing current

Θ .... Angle denoting location of a pole in a radial magnetic bearing

λ .... Denotes a Lagrange multiplier

Λ .... Diagonal matrix of non-zero singular values

μ_o .... Magnetic permeability of free space

Υ .... Diagonal matrix relating gap flux density to stored energy

Υ_x .... Derivative of Υ with respect to x

ϕ .... Column matrix denoting magnetic flux in the air gaps of a magnetic bearing

Mathematical Notation

x' .... Denotes transpose of x

$x > 0$ .... Denotes every element in x is greater than zero

$F[x]$ ..... Denotes that F is a function of x

Chapter 1

Introduction

1.1 Problem definition

The technology of active magnetic bearings has now been in existence for several decades. Jesse Beams [BYM46], working at the University of Virginia's Department of Physics in the 1940's, is usually given credit as the "Father of Magnetic Bearings." Since that time, magnetic actuators have been used in a wide range of applications: for support of rotating shafts (perhaps the most common application), for vibration isolation, as robotic "wrists", and for precision pointing applications.

Ultimately, any magnetic actuator is controlled by the variation of voltage across the machine's windings. Typically, a two-level approach is taken in specifying the control voltages. The most basic level is a fast feedback loop that tracks a requested current for the bearing coils, typically via a transconductance switching amplifier. If the amplifiers are adequately sized and operated at an adequately fast switching rate, realization of a desired current will take place at a much smaller time scale than the dynamics of the mechanical system being acted upon. In this case, it can be assumed that the requested currents are realized instantaneously. There then remains the higher-level task of specifying requested coil currents so that a desired set of actuator forces can be realized. The present thesis considers the primarily higher-level problem of the choice appropriate currents; the lower-level control task is discussed at some length by Keith [KMHW90] and Fedigan [Fed93].

Although magnetic bearings have been used for a number of different machines with diverse purposes, nearly all of these devices are composed of pairs of opposed electromagnets, as shown in Figure 1.1. Each magnet is able to pull the suspended object only towards itself; two opposed magnets are a configuration that is sufficient to generate a force of arbitrary sign. The typical actuator is built of sets of opposed magnets with one set for each controlled force direction. For example, a radial bearing used to support a rotating shaft typically has a configuration as pictured in Figure 1.2.

For the opposed-pair actuator in Figure 1.1, the current-to-force relationship is

$$f = c(i_1^2 - i_2^2) \tag{1.1}$$

where c is a constant derived from bearing geometry, and i_1 and i_2 are the currents supplied to the top and bottom coils respectively. For this type of actuator, it is relatively straightforward to invert the relationship between requested current and resulting force so that any desired force can be realized. There are typically two ways in which this inversion can be approached:

1. the bias current linearization method;
2. the direct optimization method.

The bias linearization method is perhaps the most common method of realizing desired forces in a magnetic bearing. Bias linearization is a change of variables in terms of bearing currents. Define variables $\hat{i}_o$

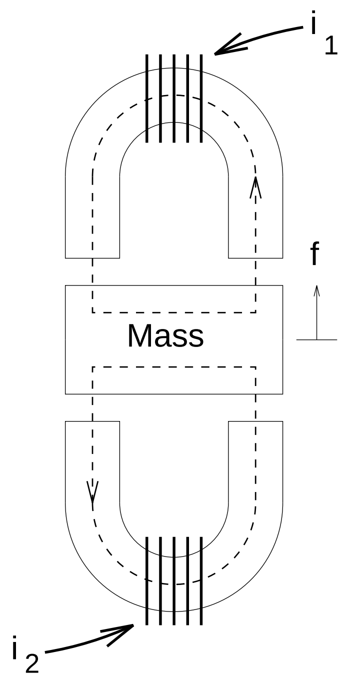


Figure 1.1: A pair of opposed electromagnets.

and $\hat{i}_c$ so that

$$\begin{aligned} i_1 &= (\hat{i}_o + \hat{i}_c)/(2\sqrt{c}) \\ i_2 &= (\hat{i}_o - \hat{i}_c)/(2\sqrt{c}) \end{aligned} \tag{1.2}$$

When transformation (1.2) is substituted into (1.1), force in terms of the new variables is

$$f = \hat{i}_o \hat{i}_c \tag{1.3}$$

Any force can then be realized by holding the magnitude of $\hat{i}_o$ constant and varying $\hat{i}_c$ linearly with the desired force. In terms of the actual coil currents, the rule for realizing any desired force is then

$$\begin{aligned} i_1 &= (\hat{i}_o + \frac{f}{\hat{i}_o})/(2\sqrt{c}) \\ i_2 &= (\hat{i}_o - \frac{f}{\hat{i}_o})/(2\sqrt{c}) \end{aligned} \tag{1.4}$$

Bias current linearization has two main advantages. First, it yields a simple formula for currents to realize any force – this formula is linear in the desired force. Second, the magnitude of the constant (or biasing) component can be chosen to avoid slew rate limiting. This condition occurs when the requested rate of variation of currents is faster than can be realized by the actuator's power amplifiers. Reduced performance or instability can result. Slew rate limiting is considered in detail in Appendix B.

Although bias linearization is easy to implement, it is not optimal in the sense of minimizing the power needed to produce a given force. An alternative to bias linearization is the direct optimization method. The method proceeds by recognizing that (1.1) is linear in (i_1^2) and (i_2^2). Furthermore, to optimally realize a force in a power sense and also be physically realizable, (i_1^2) and (i_2^2) should minimize a cost function J where

$$J = (i_1^2) + (i_2^2) \tag{1.5}$$

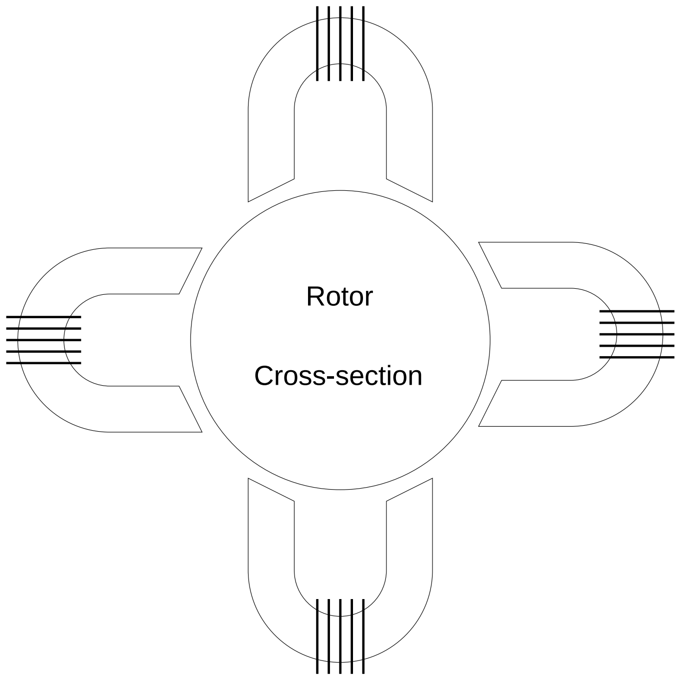


Figure 1.2: Two sets of magnets supporting a rotating shaft.

subject to

$$
\begin{aligned}
i_1^2 &\geq 0 && (1.6)\\
i_2^2 &\geq 0 && (1.7)\\
(i_1^2) - (i_2^2) - \frac{f}{c} &= 0 && (1.8)
\end{aligned}
$$

This is a classical linear programming problem, and it is easy to see that its solution is

$$
\begin{array}{ccc|l}
i_1 & = & \sqrt{f/c} & \multirow{2}{*}{$f \geq 0$} \\
i_2 & = & 0 & \\
\\
i_1 & = & 0 & \multirow{2}{*}{$f < 0$} \\
i_2 & = & \sqrt{f/c} &
\end{array} \tag{1.9}
$$

Although this solution is power-optimal, it has some undesirable characteristics. Consider that

$$\frac{df}{dt} = 2c(i_1\frac{di_1}{dt} + i_2\frac{di_2}{dt}) \tag{1.10}$$

Since (1.9) yields $i_1 = i_2 = 0$ at $f = 0$, (1.10) implies that $\frac{di_1}{dt}$ and $\frac{di_2}{dt}$ must be infinite to realize any $\frac{df}{dt}$; slew rate limiting is inevitable. To eliminate this problem, extra constraints can be imposed that force slightly higher cost solutions with realizable slew rates. One such constraint would be

$$i_1^2 i_2^2 \geq a^2 \tag{1.11}$$

where a is a constant chosen such that the solution has adequate slew rate properties. The problem can still be easily solved. The problem is represented graphically in Figure 1.3. The solution is the intersection of the

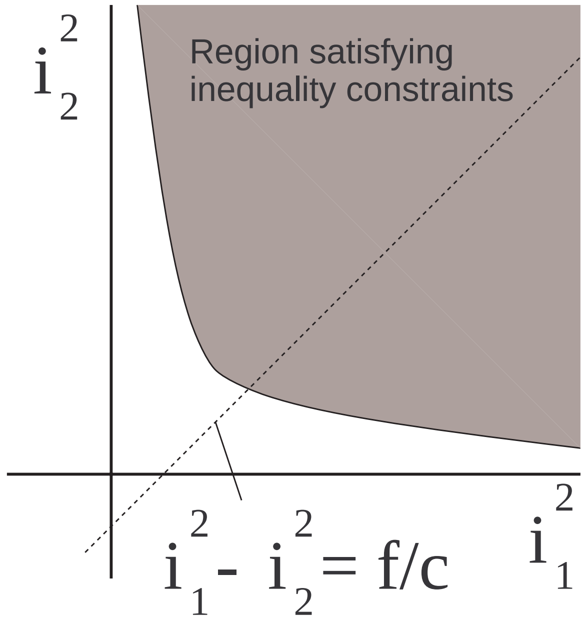


Figure 1.3: Additional constraint for realizable slew rate.

first quadrant part of constraint (1.11) with the required force line (1.8):

$$\begin{aligned} i_1^2 &= \frac{f}{2c} + \frac{1}{2c}\sqrt{f^2 + 4a^2} \\ i_2^2 &= -\frac{f}{2c} + \frac{1}{2c}\sqrt{f^2 + 4a^2} \end{aligned} \tag{1.12}$$

At high forces, (1.12) converges to (1.9). At zero force, the currents are non-zero, guaranteeing adequate slew rate in this crucial region.

In the past, the direct method was less commonly used because of the difficulty of implementing a square-root in analog circuitry. However, schemes of this type are becoming increasingly more attractive now that relatively inexpensive digital controllers are widely available.

Although the above inverse strategies have been used successfully in many devices, these strategies have certain drawbacks.

- They impose artificial constraints on actuator geometry. The actuator must be designed in an opposing horse-shoe configuration if the force directions are to be decoupled so that the above methods can be employed.

- They cannot be used to control more general quadratic actuators in which the force directions are fundamentally coupled due to the mission of the device. A prime example of such a device is the Magnetic Stereotaxis System (MSS). Magnetic stereotaxis is a novel therapeutic methodology for the treatment of brain tumors and other neurological problems. The fundamental idea of magnetic stereotaxis 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. The device is represented schematically in Figure 1.4. Incidental damage is reduced by selecting a path that avoids important brain structures. Once the seed has been maneuvered into a tumor, the seed is heated inductively by high-frequency magnetic fields. This heating results in highly localized cell death. By successive movements and heating, a tumor could be destroyed with little damage to the surrounding tissue

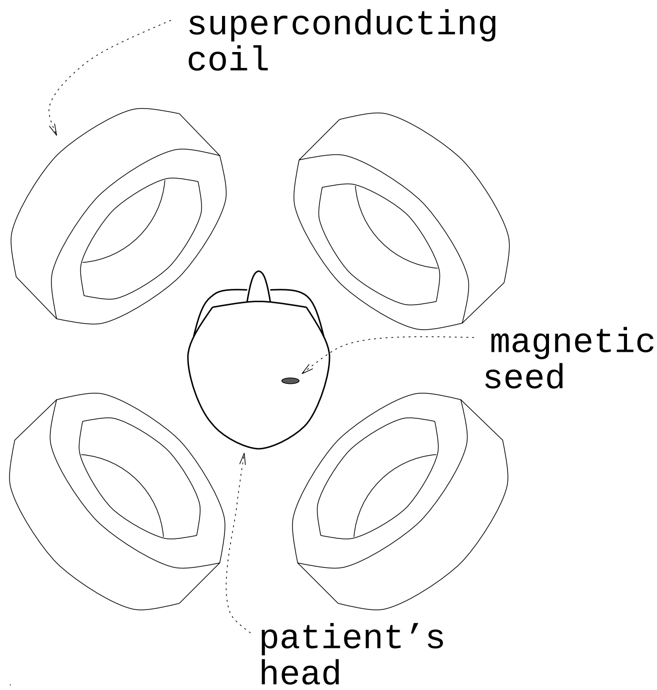


Figure 1.4: Schematic of multi-coil Magnetic Stereotaxis System.

[Mol91]. Alternatively, the magnetic seed could be used to guide the tip of a catheter. This catheter would then be used to deliver drug treatments directly to sites inside the brain [GRB+94] [RGHG92].

- They have little capability of fault-tolerance. If a coil or its associated amplifier fails, symmetry is lost; the bearing cannot be controlled. The only way to compensate in terms of the previous schemes is with redundant horse-shoe pairs [LPGS94].
- The above methods do not necessarily guarantee the best possible performance, particularly in the case of an actuator with multiple degrees of freedom.

All of these above shortcomings can be remedied by a more general formulation of the current-to-force relations combined with approaches to the inverse solution that do not rely on a design in which all force directions are physically decoupled.

Instead of a set of decoupled current-to-force relations like (1.1), a general problem will be considered of the form

$$\begin{aligned} f_1 &= i' M_1 i \\ f_2 &= i' M_2 i \\ &\vdots \\ f_k &= i' M_k i \end{aligned} \tag{1.13}$$

where f_j is the force in the j^{th} force direction, i is a vector of currents requested in the actuator coils, and M_j is a real symmetric matrix deduced from actuator geometry. The dissertation will proceed with a synopsis of past work germane to this problem. The formulation of general current-to-force relations will then be considered. The problem will be formulated for a generic Maxwell-force actuator using magnetic circuit theory. An identical formulation will also be derived for a Lorentz-force machine, the Magnetic Stereotaxis System. Generalized bias current linearization and direct optimization approaches to the inverse problem will be developed without regard to the specific implementation of the solution. Since the definition of “optimal” is solution implementation-specific, the general solution methods will then be applied to both of the specific cases of radial magnetic bearings and the magnetic stereotaxis system to yield optimal solutions for each implementation.

Chapter 2

Literature Review

Literature relevant to the dissertation might be broken down into five categories:

- Mathematical literature on quadratic forms.
- Examples of bias linearization in Maxwell-force actuators.
- Power optimal solutions to the magnetic inverse problem.
- The Magnetic Stereotaxis System.
- Current realization issues.

2.1 Mathematical literature on quadratic forms

The magnetic inverse problem has received some attention in the mathematical literature under the guise of "joint numerical range of hermitian matrices" and the "algebraic theory of quadratic forms." The work on joint numerical range was not written with any physical problem in mind, but it can be interpreted as conditions on the ability of an actuator to produce forces in all possible directions. Papers that study various properties of joint numerical range are [BL91], in which Binding and Li address joint numerical range of more than two matrices; [TU91] on the useful matrix pencil approach to the analysis of the joint numerical range of two matrices; and Yeung and Tsing [AYT84], with a number of theoretical results about joint numerical range. Although the algebraic theory of quadratic forms does not deal directly with the magnetic inverse problem, a number of crucial tools are developed. These include the Witt index of a matrix, and the concept of a totally isotropic space, both important to the interpretation of the generalized bias linearization problem. The two seminal works on the algebraic theory of quadratic forms are by Lam [Lam73] and Scharlau [Sch69].

2.2 Examples of bias linearization

Bias linearization has been widely mentioned in the literature. Usually, however, pairs of opposed horseshoes are considered, and bias linearization is therefore only mentioned in passing. Bias linearization is applied to radial magnetic bearings in works by Imlach [Iml90], Bornstein [Bor91], Chen and Darlow [CD88], Matsumura and Yoshimoto[MY86], Chiba and Rahman [CR91], Burrows *et al.* [Bur88], and Lee and Kim [LK92]. A more elaborate device also controlled via bias linearization is a 6-degree of freedom actuator discussed by Allan and Knospe [AK91]. The machine is composed of three independent actuators, each of which controls a force and a torque. Through appropriate choice of biasing currents, the force and torque can be decoupled and controlled independently. One of the uses of generalized bias linearization is the computation of fault-tolerating current mappings. An alternative approach to fault tolerance is building an actuator with redundant sets of horseshoes, as is considered by Lyons *et al.* [LPGS94].

2.3 Power optimal solutions

The general problem of manipulating an object with an arbitrary array of horseshoe actuators is considered by Iwaki [Iwa90]. Though this work limits itself to horseshoe configurations, it contains necessary and sufficient conditions for the actuator to provide all desired forces, and it addresses how currents should be picked when there is a sufficient arrangement of horseshoes. For the specific problem of radial magnetic bearings, a thesis by Green [Gre96] considers a heuristic "flexible quadrant control" scheme for realizing both low power losses and high load capacity while avoiding slew rate limiting. A number of authors use a square-root function in order to invert the non-linearity of magnetic bearings in an efficient way. This approach is advocated by Mouille and Lottin [ML92], Lottin, Mouille and Ponsart [LMP94] and Charara and Caron[CC92]. Maslen *et al.* [MANS96] consider the square-root scheme and give a detailed examination of why this scheme is subject to slew rate limiting.

Although not applied specifically to magnetic bearings, there are several works considering the application of continuation (or homotopy) methods to constrained optimization problems that are similar to the present inverse problem. In these works, the satisfaction of the Kuhn-Tucker necessary conditions for optimality are tracked from a known starting point to a desired solution. In general, these works deal not only with equality constraints but inequality constraints as well. Most relevant to the present work is a paper by Huneault *et al.* [aAFVJ85], in which a continuation approach is applied to power system optimization. Analogous to the present work, the initial condition is a no-load optimum that is trivial to determine. Load on the power system is then increased while changing the system inputs to maintain optimality. Huneault elaborates further on the application of continuation methods to power systems in [HG90]. Ponrajah and Galiana [PG92] consider power system optimization and develop on a "elevator predictor-corrector" method that is particularly suitable for the numerical integrations involved in optimization by continuation. Theoretical aspects of homotopy and continuation algorithms as applied to optimization applications are considered by Watson and Haftka [WH89] and Allgower and Georg [AG80].

2.4 The Magnetic Stereotaxis System

Another machine that fits into the class addressed by this dissertation is the Magnetic Stereotaxis System. In [GRB+94], Gillies presents a survey of magnetic manipulators for medical uses, including the MSS. Most of the work on the MSS has been done at the University of Virginia Department of Physics since the mid-1980's by group headed by R. Ritter. The early development of the device, when it consisted of a single movable coil, is described in[QWL91], [RGHG92] and [Gra90]. This first device was tested successfully in live dogs. Subsequently, an MSS consisting of 6 fixed coils, the design considered in the present dissertation, was developed when the single coil machine proved unwieldy for use on humans. A detailed description of the present machine and a discussion of previous approaches to the magnetic inverse problem in the MSS are considered in a two-part paper by McNeil *et al.* [MRW95b] [MRW95a].

2.5 Current realization issues

The avoidance of slew rate limiting is a pervasive theme in this dissertation. Slew rate and a number of other practical limitations of magnetic bearings are examined by Maslen *et al.* in [MANS96] and [MHSH89]. Bandwidth limitations arising from eddy currents in magnetic actuators are addressed by Meeker, Maslen and Noh [MMN95], Feeley [Fee96], and Zmood [ZAK87] for radial bearings, and by Kucera and Ahrens in [KA95] for axial bearings. Other issues related to the use of switching amplifiers in magnetic bearings are addressed by Fedigan [Fed93] and in Keith's doctoral dissertation [Kei93].

Chapter 3

Current-to-force relation: Maxwell force actuator

Perhaps the most common type of actuator with a homogeneous quadratic relationship between currents and forces is the radial magnetic bearing. This actuator is an example of a Maxwell force actuator; that is, an actuator where the forces are developed by a magnetic field acting upon a piece of high magnetic permeability material. Assuming negligible eddy current effects and a linear flux density to field intensity relationship with negligible hysteresis effects, a magnetostatic analysis can be employed to obtain the current-to-force relations for these actuators. If losses from flux leakage and fringing are also assumed negligible, the applicable magnetostatic field equations become one dimensional. Flux and field intensity at any point in the bearing can then be solved by circuit theory [Plo78]. An analysis of the magnetic circuits in the actuator yields a quadratic relationship between coil currents and resulting forces. These force relations will be derived with the specific example of the radial magnetic bearing in mind; however, the same technique applies to configurations with other than two degrees of freedom.

An n pole magnetic bearing (as exemplified by Fig. 3.1) is characterized by R_j, Ni_j, ϕ_j, a_j, and Θ_j for $j = 1 \ldots n$, the reluctance, magnetomotive force contribution, flux, pole face area, and orientation angle respectively for each pole. Considering that steel or iron has a relative permeability of greater than 1000, the reluctances of all metal parts of the flux path are neglected; virtually all of the circuit reluctance is due to the air gap associated with each pole. Positive fluxes are directed out of the stator poles into the rotor by the sign convention for this model. Positive coil currents pass counter-clockwise around the stator poles when viewing the pole end from the gap. It is assumed that the only sources of magnetic excitation in the bearing are the coils wound on each pole. This assumption specifically excludes bearings employing permanent magnets from this analysis. An equivalent electrical circuit, useful in understanding the development of the governing magnetic equations, appears in Figure 37.

The application of Ampere's loop law to the magnetic circuit results in $n-1$ independent equations:

$$\mathsf{R}_j\phi_j - \mathsf{R}_{j+1}\phi_{j+1} = N_j i_j - N_{j+1} i_{j+1} \tag{3.1}$$

where the reluctance of the j^{th} gap is

$$\mathsf{R}_j = \frac{g_j}{\mu_o a_j} \tag{3.2}$$

One independent equation results from flux conservation:

$$\sum_{j=1}^{n} \phi_j = 0 \tag{3.3}$$

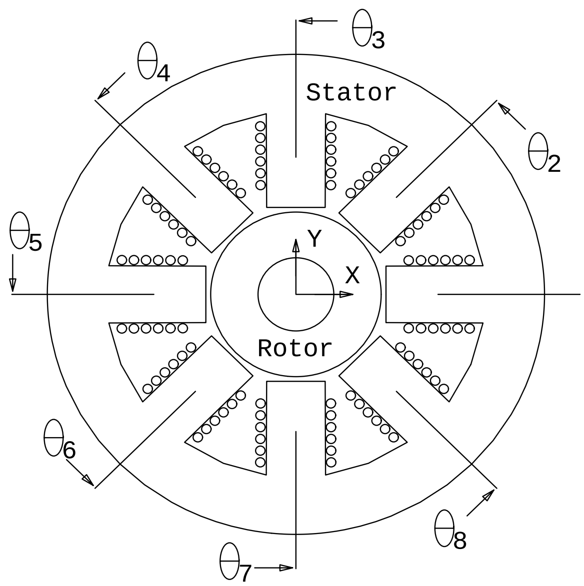


Figure 3.1: Typical bearing arrangement

Arranging these equations in matrix form produces

$$\begin{bmatrix} \mathsf{R}_1 & -\mathsf{R}_2 & 0 & \cdots & 0 \\ 0 & \mathsf{R}_2 & -\mathsf{R}_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathsf{R}_{n-1} & -\mathsf{R}_n \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \phi = \begin{bmatrix} N_1 & -N_2 & 0 & \cdots & 0 \\ 0 & N_2 & -N_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & N_{n-1} & -N_n \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} i \tag{3.4}$$

This matrix relationship is represented more succinctly by

$$R\phi = \mathsf{N}\, i \tag{3.5}$$

where R can easily be shown to be nonsingular. The flux in each leg due to the applied currents is then

$$\phi = R^{-1}\mathsf{N}\, i \tag{3.6}$$

A useful result that can be obtained from (3.6) is the matrix L of self and mutual inductances between the different coils in the bearing:

$$L = \mathrm{diag}[N_1, \ldots, N_n]\, R^{-1}\mathsf{N} \tag{3.7}$$

Assuming uniform flux density in the air gap, flux ϕ_j is related to flux density b_j by $\phi_j = b_j a_j$. In matrix form, this relationship is

$$\phi = \mathsf{A} b \tag{3.8}$$

where A is a diagonal matrix of pole face areas. Re-arranging and substituting from (3.5) and (3.8),

$$b = \mathsf{A}^{-1}R^{-1}\mathsf{N}\, i = Vi,\ V \doteq \mathsf{A}^{-1}R^{-1}\mathsf{N} \tag{3.9}$$

Note from (3.4) that the matrix N has a nullity of 1. Consequently, one of the currents in i is redundant if each leg has an independent coil.

Forces produced by the bearing can be computed by variations of the energy stored in the system or by Maxwell's stress tensor. A complete discussion of the different methods of calculating magnetic force is

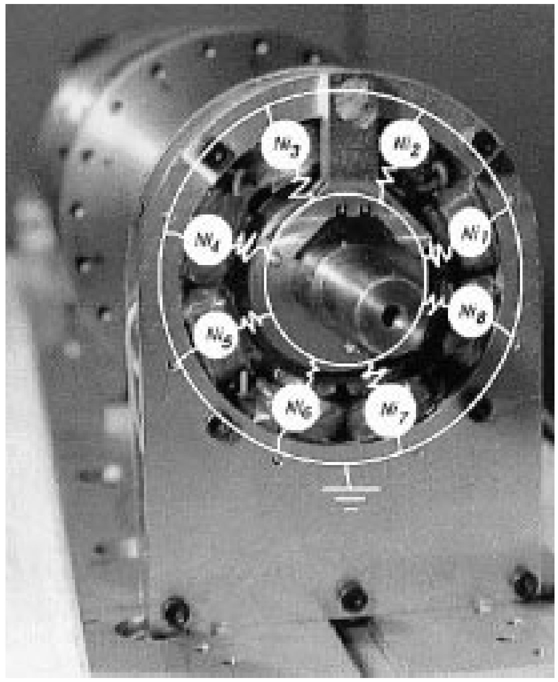


Figure 3.2: Equivalent electrical circuit

contained in [Sad92]. Here, the energy method will be used under the constant excitation voltage assumption. Assuming linear materials, the energy stored in a magnetic field is defined in the general case as

$$E = \int \frac{1}{2\mu} \mathbf{b} \cdot \mathbf{b} \, dV \tag{3.10}$$

where **b** is vector flux density and the integral is taken over all space. In the present one-dimensional analysis, the only component is b along the path direction. Due to the assumptions of no leakage and zero reluctance of the metal sections of the path, all of the energy is stored in the air gaps:

$$E = \sum_{j=1}^{n} \frac{g_j[x] a_j}{2\mu_o} b_j^2 \tag{3.11}$$

The energy can be written in vector form as

$$E = b' \Upsilon b = i' V' \Upsilon[x] V i \tag{3.12}$$

where $\Upsilon[x]$ is a diagonal matrix with the j^{th} entry equal to $g_j[x]a_j/(2\mu_o)$. Note that the $g_j[x]$ are the mean air gap lengths as functions of x, a set of coordinates specifying the rotor's position. Coordinates x_1, x_2 and x_3 might be associated with translations along the X, Y and Z axes that define some fixed coordinate system, whereas x_4, x_5 and x_6 might be associated with infinitesimal rotations about the X, Y and Z axes respectively.

Force is defined as

$$\begin{aligned} f_j &= -\frac{\partial E}{\partial x_j} \\ &= -b' \Upsilon_j b \\ &= -i' V' \Upsilon_j V i \end{aligned} \tag{3.13}$$

where Υ_j denotes $\frac{\partial \Upsilon}{\partial x_j}$. Defining the symmetric matrix M_j as

$$M_j = -V'\Upsilon_j V \tag{3.14}$$

it can be noted that the current-to-force relation is identical in form to 1.13:

$$f_j = i' M_j i \tag{3.15}$$

In the particular case of radial magnetic bearings, the only force components are in the X and Y directions (see Figure 3.1). In this case, the position-dependent gap lengths are

$$g_j[x] = g_{j,o} - x\cos\Theta_j - y\sin\Theta_j \tag{3.16}$$

where $g_{j,o}$ is the length of the j^{th} gap when the rotor is in the centered position. Matrices Υ_x and Υ_y can then be explicitly defined as

$$\Upsilon_x = \text{diag}\left[\frac{-a_j\cos\Theta_j}{2\mu_o}\right] \quad , \quad \Upsilon_y = \text{diag}\left[\frac{-a_j\sin\Theta_j}{2\mu_o}\right] \tag{3.17}$$

where Θ_j is the angular position of the centerline of the j^{th} stator leg.

If one or more of the coils is missing or has failed ($N_j i_j = 0$), then (3.15) still applies. The matrix K is introduced to relate the reduced order current vector of dimension m to the full current vector:

$$i = \mathsf{K}\,\imath \tag{3.18}$$

Matrix K is simply the identity matrix with columns removed corresponding to each failed or missing coil. Substituting into (3.15) yields:

$$f_j = \imath'(\mathsf{K}' M_j \mathsf{K})\imath \tag{3.19}$$

Matrix K can also be used to indicate coils wired in series. In this case, the vector of coil currents can be represented as the product of a matrix times a vector of *independent* coil currents. For instance, assume that coils 1 and 2 are wound in reverse series ($i_2 = -i_1$). The K reflecting this coupling would be

$$\mathsf{K} = \begin{bmatrix} 1 & 0 & & 0 \\ -1 & 0 & \cdots & \vdots \\ 0 & 1 & & \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

It is worth noting that $V'\Upsilon_j V$ has a null space of dimension 1. This singularity can be removed by defining a K with $n-1$ columns whose columns span the row space of N.

3.1 Example 1 – Asymmetric Magnetic Bearing

In the past, the problem of determining bias and control currents was only considered for symmetric cases. Under these conditions, the proper linearizing currents are obtained by inspection. However, when symmetry is lost, determination of the proper currents is no longer a trivial problem. Take for example the bearing pictured in Figure 3.3. The geometry of this bearing is described in Table 3.1, where $a = 1\,cm^2$, $g_o = 1\,mm$ and $N = 200$. The unusual asymmetry of this example is intended only to emphasize the generality of the result: such asymmetry would seldom be encountered in practice. The point to this example is that the usual assumptions concerning symmetry are not needed – a result which is particularly useful in permitting the fault tolerance alluded to in the introduction.

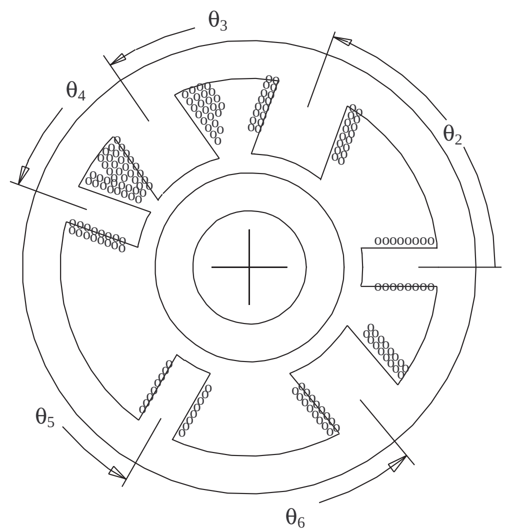


Figure 3.3: Asymmetric bearing.

Leg	θ	Area	Turns	Gap
1	0^o	a	N	g_o
2	70^o	2 a	2 N	g_o
3	125^o	2 a	3 N	g_o
4	160^o	a	2 N	g_o
5	240^o	a	N	g_o
6	310^o	2 a	2 N	g_o

Table 3.1: Asymmetric bearing parameters.

The reluctance of each air gap is determined by (3.2). Substituting the reluctances into (3.4) gives:

$$\frac{g_o}{\mu_o a}\begin{bmatrix} 1. & -0.5 & 0. & 0. & 0. & 0. \\ 0. & 0.5 & -0.5 & 0. & 0. & 0. \\ 0. & 0. & 0.5 & -1. & 0. & 0. \\ 0. & 0. & 0. & 1. & -1. & 0. \\ 0. & 0. & 0. & 0. & 1. & -0.5 \\ 1. & 1. & 1. & 1. & 1. & 1. \end{bmatrix}\phi = N\begin{bmatrix} 1. & -2. & 0. & 0. & 0. & 0. \\ 0. & 2. & -3. & 0. & 0. & 0. \\ 0. & 0. & 3. & -2. & 0. & 0. \\ 0. & 0. & 0. & 2. & -1. & 0. \\ 0. & 0. & 0. & 0. & 1. & -2. \\ 0. & 0. & 0. & 0. & 0. & 0. \end{bmatrix} i \tag{3.20}$$

Re-arranging according to (3.9), the current to flux density relationship is:

$$b = \frac{\mu_o N}{9 g_o}\begin{bmatrix} 8 & -4 & -6 & -2 & -1 & -4 \\ -1 & 14 & -6 & -2 & -1 & -4 \\ -1 & -4 & 21 & -2 & -1 & -4 \\ -1 & -4 & -6 & 16 & -1 & -4 \\ -1 & -4 & -6 & -2 & 8 & -4 \\ -1 & -4 & -6 & -2 & -1 & 14 \end{bmatrix} i \tag{3.21}$$

This example is a radial magnetic bearing; therefore, Υ_x and Υ_y can be obtained directly from (3.17):

$$\Upsilon_x = \frac{a}{2\mu_o} \mathrm{diag}\,[1., 0.6840, -1.1472, -0.9397, -0.5, 1.2856] \tag{3.22}$$

$$\Upsilon_y = \frac{a}{2\mu_o} \mathrm{diag}\,[0, 1.8794, 1.6383, 0.3420, -0.8660, -1.5321] \tag{3.23}$$

Because this stator has an independent coil on each leg, one coil will be redundant; matrices M_x and M_y will both be singular. This singularity can be removed with a suitable K matrix. The K matrix should have columns orthogonal to the null space of V. This null space represents a vector of currents that produces no flux through the gaps. When K is chosen orthogonal to this space, a given flux distribution is then realized with the least possible power dissipation since all portions of the current are contributing to producing flux. One such matrix, derived by Gram–Schmidt orthogonalization [HJ85] of the N matrix, is

$$\mathsf{K} = \begin{bmatrix} 0.447214 & 0.255551 & 0.337645 & 0.487556 & 0.182932 \\ -0.894427 & 0.127775 & 0.168823 & 0.243778 & 0.0914661 \\ 0. & -0.958315 & 0.112548 & 0.162519 & 0.0609774 \\ 0. & 0. & -0.919145 & 0.243778 & 0.0914661 \\ 0. & 0. & 0. & -0.785507 & 0.182932 \\ 0. & 0. & 0. & 0. & -0.955312 \end{bmatrix} \tag{3.24}$$

Note that the choice of this particular matrix is somewhat arbitrary. Any other K whose columns lie perpendicular to the null space of V would give the same power-minimizing properties.

The force-current relationships are specified by (3.19) as:

$$\mathsf{K}' M_x \mathsf{K} = \left(\frac{a\mu_o N^2}{g_o^2}\right) \begin{bmatrix} 4.762 & 1.865 & 0.983 & -0.395 & -3.105 \\ 1.865 & -12.882 & 5.985 & 2.487 & 0.862 \\ 0.983 & 5.985 & -7.667 & 1.203 & 2.126 \\ -0.395 & 2.487 & 1.203 & -1.196 & 1.731 \\ -3.105 & 0.862 & 2.126 & 1.731 & 8.076 \end{bmatrix} \tag{3.25}$$

$$\mathsf{K}' M_y \mathsf{K} = \left(\frac{a\mu_o N^2}{g_o^2}\right) \begin{bmatrix} 9.681 & -9.828 & -2.484 & -0.175 & -0.533 \\ -9.828 & 23.695 & -2.955 & 0.426 & 0.464 \\ -2.484 & -2.955 & 3.961 & 0.444 & 0.817 \\ -0.175 & 0.426 & 0.444 & -1.891 & 0.673 \\ -0.533 & 0.464 & 0.817 & 0.673 & -8.581 \end{bmatrix} \tag{3.26}$$

3.2 Example 2 – 3 d.o.f. actuator

As an example of a device with more than two degrees of freedom, consider the device pictured in Figure 3.4. This particular magnetic bearing controls X and Y forces as well as a torque about the Z axis. Each coil is wound with N turns and has a pole area of a. The air gaps as a function of rotor position are:

$$\begin{aligned} g_1 &= g_o - x \\ g_2 &= g_o - y - d\beta \\ g_3 &= g_o - y + d\beta \\ g_4 &= g_o + x \\ g_5 &= g_o + y - d\beta \\ g_6 &= g_o + y + d\beta \end{aligned} \tag{3.27}$$

where g_o is nominal gap length.

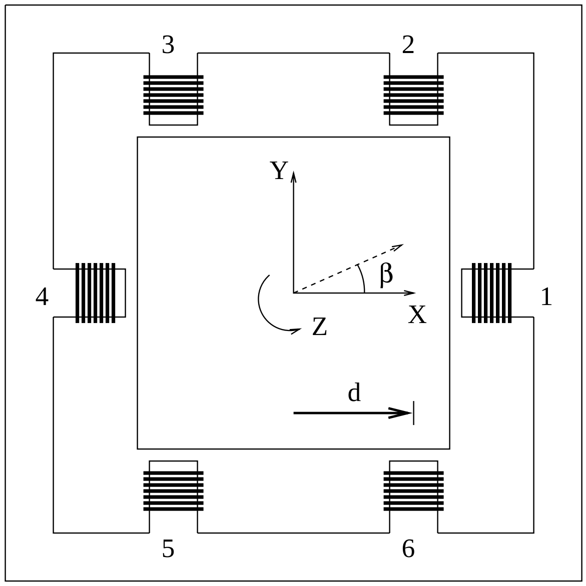


Figure 3.4: 3 d.o.f. magnetic bearing.

From (3.4),

$$\frac{g_o}{\mu_o a}\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}\phi = N\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} i \tag{3.28}$$

Inverting the left-hand side and dividing by a results in (3.9).

$$b = \frac{\mu_o N}{6g_o}\begin{bmatrix} 5 & -1 & -1 & -1 & -1 & -1 \\ -1 & 5 & -1 & -1 & -1 & -1 \\ -1 & -1 & 5 & -1 & -1 & -1 \\ -1 & -1 & -1 & 5 & -1 & -1 \\ -1 & -1 & -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & -1 & -1 & 5 \end{bmatrix} i \tag{3.29}$$

By differentiating (3.27) by the rotor degrees of freedom,

$$\Upsilon_x = \left(\frac{a}{2\mu_o}\right)\mathrm{diag}\{-1,0,0,1,0,0\} \tag{3.30}$$

$$\Upsilon_y = \left(\frac{a}{2\mu_o}\right)\mathrm{diag}\{0,-1,-1,0,1,1\} \tag{3.31}$$

$$\Upsilon_\beta = \left(\frac{a\,d}{2\mu_o}\right)\mathrm{diag}\{0,-1,1,0,-1,1\} \tag{3.32}$$

Matrices M_x, M_y, and M_β characterizing the current-to-force and current-to-torque relationships are formed from (3.14):

$$M_x = -\frac{\mu_o a N^2}{12 g_o^2} \begin{bmatrix} -4 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 4 & -1 & -1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{3.33}$$

$$M_y = -\frac{\mu_o a N^2}{12 g_o^2} \begin{bmatrix} 0 & 1 & 1 & 0 & -1 & -1 \\ 1 & -4 & 2 & 1 & 0 & 0 \\ 1 & 2 & -4 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 4 & -2 \\ -1 & 0 & 0 & -1 & -2 & 4 \end{bmatrix} \tag{3.34}$$

$$M_\beta = -\frac{\mu_o d a N^2}{12 g_o^2} \begin{bmatrix} 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -4 & 0 & 1 & 2 & 0 \\ -1 & 0 & 4 & -1 & 0 & -2 \\ 1 & 2 & 0 & 1 & -4 & 0 \\ -1 & 0 & -2 & -1 & 0 & 4 \end{bmatrix} \tag{3.35}$$

These three matrices completely define the relationship between coil current and output forces/torques.

3.3 Variation of force with position

It is important to note that the current-to-force relations derived above are a function of the position of the suspended object. This dependence arises because the reluctance of each air gap varies linearly with the length of the gap, as shown in (3.2). Consequently, reluctance matrix R is a linear function of rotor position. However, R is inverted to obtain the force relations; the dependence of force on position is therefore not linear.

The position-dependence of each air gap can be specifically included in the R matrix. An exact expression for position-dependent force can then be obtained by inverting this symbolic matrix R and then forming the force relations via (3.15). However, the symbolic inversion of R is only possible for matrices of low dimension. An alternative formulation that does not involve inverting a symbolic matrix is a Taylor series expansion of force in terms of displacements. For small displacements, it is sufficient to truncate this expansion after the first term:

$$f_j = i' \left(M_j\Big|_{x=0} + \sum_{q=1}^{k} x_q \frac{\partial M_j}{\partial x_q}\Big|_{x=0} \right) i \tag{3.36}$$

The derivatives of M_j can be evaluated analytically:

$$\frac{\partial R^{-1}}{\partial x_j} = -R^{-1} \frac{\partial R}{\partial x_j} R^{-1} \tag{3.37}$$

Eq. (3.37) can then be used to evaluate the derivatives of V:

$$\frac{\partial V^{-1}}{\partial x_j} = -R^{-1} \frac{\partial R}{\partial x_j} R^{-1} N = -R^{-1} \frac{\partial R}{\partial x_j} V \tag{3.38}$$

Finally, the derivatives of M_j are

$$\frac{\partial M_j}{\partial x_q} = \frac{\partial}{\partial x_j}(-V' \Upsilon_j V)$$

$$
\begin{aligned}
&= \frac{\partial V}{\partial x_j}' \Upsilon_j V + V' \Upsilon_j \frac{\partial V}{\partial x_j} \\
&= V' \frac{\partial R}{\partial x_j}' R^{-T} \Upsilon_j V + V' \Upsilon_j R^{-1} \frac{\partial R}{\partial x_j} V
\end{aligned}
\tag{3.39}
$$

As an example of the above procedure, a one-term taylor expansion of force is computed for a symmetric 8-pole bearing in Appendix A.

Chapter 4

Current to force relations: Lorentz force actuator

The Magnetic Stereotaxis Machine is an example of a Lorentz force actuator; that is, an actuator wherein forces are derived by an electromagnet acting upon a piece of permanent magnetic material. Although Lorentz-type devices are usually linear, peculiarities of this particular machine give it a set of quadratic current-to-force relationships identical in form to those derived previously for Maxwell force actuators.

4.1 Governing Dipole Equations

Since the dimensions of the permanent magnet seed are very small compared to the size of the superconducting coils, the seed can be idealized as a point dipole. The magnetic properties of the seed are summarized by the seed's dipole moment, **m**. The direction of this vector is the same as the direction of magnetization in the seed. The magnitude of **m** is the product of the magnetization and volume of the seed.

Derivation of the forces and torques on a dipole due to an applied magnetic field can be found in the literature [Jac75]. Defining **p** as a position vector locating the seed relative to the center of the helmet, the force on the seed is

$$\mathbf{f}[\mathbf{p}] = \nabla(\mathbf{m}\cdot\mathbf{b}[\mathbf{p}]) \tag{4.1}$$

and the torque is

$$\mathbf{t}[\mathbf{p}] = \mathbf{m}\times\mathbf{b}[\mathbf{p}] \tag{4.2}$$

4.2 Formulation of Current-to-Force Relationships

Equations (4.1) and (4.2) specify the force and torque respectively on the dipole seed, but they do not imply any particular basis in terms of which these vectors are represented. A computationally useful form of these equations can be obtained by defining a basis of orthogonal vectors $\mathbf{n}_1, \mathbf{n}_2$ and $\mathbf{n}_3$ that are fixed at the center of the helmet. With these vectors, define

$$\begin{aligned} \mathbf{p} &= x\mathbf{n}_1 + y\mathbf{n}_2 + z\mathbf{n}_3 \\ \mathbf{m} &= m_1\mathbf{n}_1 + m_2\mathbf{n}_2 + m_3\mathbf{n}_3 \\ \mathbf{f} &= f_1\mathbf{n}_1 + f_2\mathbf{n}_2 + f_3\mathbf{n}_3 \\ \mathbf{b} &= b_1\mathbf{n}_1 + b_2\mathbf{n}_2 + b_3\mathbf{n}_3 \end{aligned} \tag{4.3}$$

Since **m** is not a function of **p**, (4.1) can be written in matrix form as

$$f_1 = m'\frac{\partial b}{\partial x}$$

$$\begin{aligned} f_2 &= m' \frac{\partial b}{\partial y} \\ f_3 &= m' \frac{\partial b}{\partial z} \end{aligned} \tag{4.4}$$

where $(')$ denotes transpose.

Equation (4.4) defines the field-to-force relation in a particular reference frame, but the connection between coil currents and the magnetic field, b, must still be defined. Since the permeability of the seed is very close to that of free space, it is appropriate to characterize the field at a point as a linear superposition of field contributions created by each of the six coils. Furthermore, the contribution of any particular coil is linearly proportional to the current in that coil. The relationship between coil current and field can then be concisely represented in matrix notation as

$$b[x,y,z] = B[x,y,z]\, i \tag{4.5}$$

where i is a column matrix of coil currents and B is a 3×6 matrix dependent on seed position. The j^{th} column of B represents the contribution of the j^{th} coil to the field at $[x,y,z]$ in response to a unit of current in the coil.

Spatial derivatives of the field also follow the same rule of superposition. Coil currents are related to these derivatives by the differentiation of (4.5):

$$\frac{\partial b}{\partial w} = \left[\frac{\partial B[x,y,z]}{\partial w}\right] i \qquad \text{where} \qquad w = x,y,z \tag{4.6}$$

Defining

$$D_1 = \frac{\partial B}{\partial x}; \qquad D_2 = \frac{\partial B}{\partial y}; \qquad D_3 = \frac{\partial B}{\partial z} \tag{4.7}$$

and referring to (4.4), a particular force component is

$$f_j = m' D_j i \tag{4.8}$$

For a given dipole orientation, all force components can be collected into the expression

$$f = \begin{bmatrix} m'D_1 \\ m'D_2 \\ m'D_3 \end{bmatrix} i \tag{4.9}$$

Eq. (4.8) appears to be linear in i. However, peculiarities of the MSS make m a function of i. There are several crucial observations that can be made about the motions of a permanent magnet seed through neural tissue due to a slowly varying magnetic field. First, there is a certain threshold force that must be exceeded for translational motion to occur. Second, the resistance of the seed to rotation is negligible, as has been observed experimentally [Gra90] [MRW95a]. From (4.2), it can be noted that torque on the seed is zero only when $\mathbf{m}$ is aligned with $\mathbf{b}$. Furthermore, if $\mathbf{m}$ is parallel to $\mathbf{b}$, a small perturbation in $\mathbf{m}$ produces a torque that tends to re-align $\mathbf{m}$ and $\mathbf{b}$. The converse is true if $\mathbf{m}$ is antiparallel to $\mathbf{b}$. Since the seed is free to rotate, it will align with the only stable equilibrium orientation: $\mathbf{m} \| \mathbf{b}$. Since there is a considerable threshold force that must be applied before translational motion begins, the seed is assumed to line up with $\mathbf{b}$ before any change in seed position occurs.

The constraint that the dipole moment is aligned with the flux density field is written as

$$\mathbf{m} = \frac{|\mathbf{m}|}{|\mathbf{b}|}\mathbf{b} \tag{4.10}$$

Substituting from (4.5), the dipole moment in matrix notation is

$$m = \frac{\gamma B i}{\sqrt{i' B' B i}} \tag{4.11}$$

where γ is a constant equal to $|\mathbf{m}|$. Combining (4.11) with the previously developed current to force relation (4.8) yields

$$f_j = \frac{\gamma i' B' D_j i}{\sqrt{i' B' B i}} \tag{4.12}$$

By employing the coordinate transformation

$$i = \frac{\imath}{\gamma}\sqrt{\imath' B' B \imath} \tag{4.13}$$

the denominator of (4.12) is removed, leaving a quadratic in $\imath$:

$$f_j = \imath' B' D_j \imath \tag{4.14}$$

Eq. (4.14) is quadratic in $\imath$. For ease of manipulation, it is preferable to manipulate quadratics in terms of operations on symmetric matrices. Any square matrix can be represented as the sum of a symmetric and an anti-symmetric matrix [HJ85]. A quadratic formed from the anti-symmetric part is always zero; therefore, only the symmetric part of $B'D_j$ contributes to $\imath' B' D_j \imath$. Define M_j as the symmetric part of $B'D_j$:

$$M_j = \frac{1}{2}(D_j' B + B' D_j) \tag{4.15}$$

The force relations can now be written in terms of the symmetric matrices M_j: a homogeneous quadratic form identical to (1.13):

$$f_j = \imath' M_j \imath \tag{4.16}$$

Chapter 5

Realizability of arbitrary forces

This chapter considers conditions ensuring that a force can be produced in every intended direction by some set of input currents. Knowledge of these conditions is vital to the successful design of magnetic actuators. Previously, actuators were designed in opposed pairs, ensuring that all forces could be produced. Although such a design is sufficient to produce all forces, it will be shown that it is not necessary. More general conditions will be presented that also ensure that every force can be realized. These conditions are of special interest in the case of coil failures; they can be used to determine which failure configurations are and are not catastrophic.

The goal of all forces being realizable can be precisely defined in a mathematical sense. Define the set $\mathsf{W}\,[]$ to be

$$\mathsf{W}\,[M_1,\ldots,M_k] = \left\{ \begin{matrix} i'M_1 i \\ \vdots \\ i'M_k i \end{matrix} \right\} \in \Re^k : i \in \Re^n \tag{5.1}$$

$\mathsf{W}\,[]$ is the set of all possible forces that can be produced by a given actuator characterized by $(M_1,\ldots,M_k)$. A successful actuator design should satisfy the following proposition:

Proposition 5.1 $\mathsf{W}\,[M_1,\ldots,M_k] \equiv \Re^k$

The conditions for which Prop. 5.1 is true have been studied in the literature for some special cases. Elegant results have been obtained in [Iwa90] for the case in which all M matrices are diagonal or simultaneously diagonalizable. Results have also been obtained in the literature for 2-d.o.f. actuators with no restrictions on the structure of the M matrices. These two results will be explored; then, necessary and sufficient conditions for the truth of Prop. 5.1 under the constraint of finite slew rate will be explored.

5.1 Diagonal M matrices

If all M matrices are diagonal, the current-to-force relations can be written as

$$f = \begin{bmatrix} \text{diagonal of } M_1 \\ \vdots \\ \text{diagonal of } M_k \end{bmatrix} e = K\,e\,;\;\; K \in \Re^{k\times n} \tag{5.2}$$

where

$$e = \left\{ \begin{matrix} i_1^2 \\ \vdots \\ i_n^2 \end{matrix} \right\} \tag{5.3}$$

Since the M matrices are diagonal, the problem can be converted to a linear problem in terms of the squares of the currents. For Prop. 5.1 to be true, every element in f must be realizable by some $e \geq 0$ in the sense that every component of e is ≥ 0).

Proposition 5.2 *If $Rank[K] \leq k$, Prop. 5.1 is false.*

Proof: If $Rank[K] < k$, the system of equations is underdetermined. Any force with a component perpendicular to the columns of K is not realizable. Since $Rank[K] < k$, there is at least one one-dimensional space of unrealizable forces. If $Rank[K] = k$, K is invertible, yielding a unique, one-to-one mapping between e and f. For any e chosen with a component less than zero, the force $f = Ke$ is not realizable.

For $n > k$, partition K into a set of k linearly independent columns, K_1, and a set of remaining columns, K_2:

$$K = [K_1|K_2] \tag{5.4}$$

Likewise, partition e into e_1 and e_2 corresponding to the partitioning of K so that

$$f = K_1 e_1 + K_2 e_2 \tag{5.5}$$

This equation can be solved for e_1:

$$e_1 = K_1^{-1} f - K_1^{-1} K_2 e_2 \tag{5.6}$$

The possible solutions for e that realize a given force are then

$$e = \begin{bmatrix} K_1^{-1} \\ 0 \end{bmatrix} f + \begin{bmatrix} -K_1^{-1} K_2 \\ I \end{bmatrix} e_2 \tag{5.7}$$

where e_2 is chosen arbitrarily. Equation (118) can be written more succinctly as

$$e = e_f[f] + e_b[e_2] \tag{5.8}$$

where

$$e_f[f] = \begin{bmatrix} K_1^{-1} \\ 0 \end{bmatrix} f \tag{5.9}$$

and

$$e_b[e_2] = \begin{bmatrix} -K_1^{-1} K_2 \\ I \end{bmatrix} e_2 \tag{5.10}$$

The e_f component of e is mandatory for creating the desired force f. However, by direct substitution, it can be seen that the e_b component of e creates no force:

$$Ke_b = K_1(-K_1^{-1} K_2)e_2 + K_2 e_2 = 0$$

Though e_b does not create any force, it is essential to generating a realizable e: all the elements in e_f are not necessarily greater than or equal to zero for a given f, so an appropriate e_b must be included to make all elements in e greater than or equal to zero and therefore realizable.

Proposition 5.3 *The existence of a vector $e_2^* \in \Re^{n-k}$ such that*

$$\begin{aligned} -K_1^{-1} K_2 e_2^* &> 0 \\ e_2^* &\geq 0 \end{aligned}$$

is a necessary and sufficient condition for Prop. 5.1 to be true.

Sufficient: Assumet that there exists a vector e_2^* satisfying the above conditions. Let $e_2 = c_o e_2^*$ where c_o is defined as:

$$c_o = \left| \frac{\min(K_1^{-1} f)}{\min(-K_1^{-1} k_2 e_2^*)} \right| \tag{5.11}$$

Note that since $(-K_1^{-1} k_2 e_2^* > 0)$ by definition, the denominator of c_o is never zero, and c_o is always finite. This choice of e_2 yields:

$$e_1 = K_1^{-1} f - co K_1^{-1} K_2 e_2^* \tag{5.12}$$

Since $c_o \geq 0$, by (5.11), the e_2 component of e must be greater than or equal to zero, since $e_2^* \geq 0$. Furthermore, this choice of c_o guarantees that every element is e_1 must be greater than or equal to zero, since c_o scales $(-K_1^{-1} K_2 e_2^*)$ so that its smallest component is positive and of equal magnitude to the most negative component in $(K_1^{-1} f)$. Since $e_1 \geq 0$ and $e_2 \geq 0$, $e \geq 0$ and is therefore realizable. Since the choice of f is arbitrary, all forces can be produced by a realizable e.

Necessary: Since K^{-1} is non-singular, some f can be chosen so that

$$K^{-1} f = \left\{ \begin{array}{c} -1 \\ \vdots \\ -1 \end{array} \right\}$$

For this particular f,

$$e_1 = \left\{ \begin{array}{c} -1 \\ \vdots \\ -1 \end{array} \right\} - K_1^{-1} K_2 e_2$$

If there is no $(-K_1^{-1} K_2 e_2^* > 0)$, r_1 cannot be made greater than or equalt to zero for this f. The required conditions are therefore necessary.

The required current vector $e_b[e_2^*]$ can be considered a biasing vector; that is, a vector of non-zero currents that produces zero forces in the actuator. This set of currents heuristically "pre-tensions" the system so that forces can be produced in an arbitrary direction, analogous to the way that gravity pre-tensions the system in Figure 5.1. Gravity provides a downward force; by counteracting lesser or greater amounts of the force of gravity, a net force in either an up or down direction can be produced, even though the horse-shoe magnet can only pull upwards.

Although Prop. 122 only applies for a system with diagonal matrices, it does imply an interesting design constraint for any Maxwell-force actuator:

Corollary 5.1 *Any Maxwell-force actuator must have at least $k+1$ poles as a necessary condition for all forces to be realizable.*

Proof: The flux-to-force relationship is characterized in (51) as

$$f_j = -b' \Lambda_j b$$

where Λ_j is a diagonal matrix. In terms of fluxes rather than currents, every actuator assumes a diagonal form regardless of the pattern in which the coils are wound. For fluxes to be chosen that realize all arbitrary forces, exactly the same argument as in Prop. 114 and in the "necessary" part of Prop. 122 can be made, simply substituting fluxes for currents. However, flux conservation constraints preclude all possible sets of gap fluxes from being realized; $k+1$ poles is not a sufficient condition for all forces to be realized.

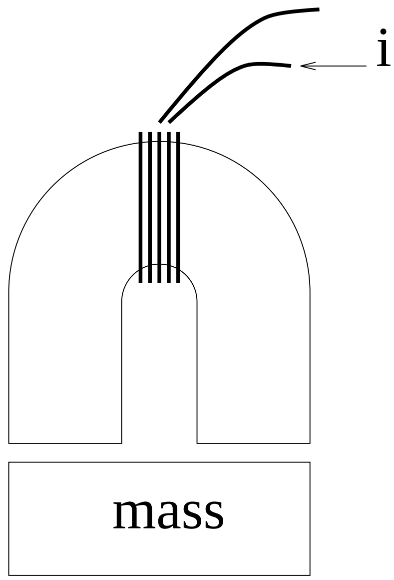


Figure 5.1: Horse-shoe actuator biased by gravity.

5.2 Necessary conditions for realizability of arbitrary forces in the literature

The other relevant case previously considered in the literature is a 2 degree-of-freedom actuator with no restrictions on the form of the M matrices other than symmetry. This problem has been considered in mathematics literature under the guise of "joint numerical range of hermitian matrices," with no connection to a physical problem [AYT84] [TU91] [BL91]. The relevant result is

Theorem 5.1 *Given the two degree-of-freedom force relation*

$$\begin{aligned} f_1 &= i' M_1 i \\ f_2 &= i' M_2 i \end{aligned}$$

Prop. 114 is true only if the matrix pencil $c_1 M_1 + c_2 M_2$ is indefinite for all real c_1, c_2.

In demonstrating this result, it is first important to consider the forces producible for any single matrix M.

Lemma 5.1 *The quadratic form $f = i'Mi$, where M is real and symmetric, can produce $f < 0$ and $f > 0$ if and only if M is indefinite (has both positive and negative eigenvalues).*

Proof: Let $M = \Phi \Lambda \Phi'$ be the eigenvalue decomposition of M. Since M is real and symmetric, all entries in diagonal matrix Λ are real. Since the ordering of the entries in Λ is arbitrary, they can be assumed to be specified in descending order such that Λ can be partitioned as

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\Lambda_2 \end{bmatrix}$$

where Λ_1 represents the positive eigenvalues of M, $-\Lambda_2$ represents the negative eigenvalues of M, and all elements of Λ_1 and Λ_2 are greater than zero. Define $x \equiv \Phi' i$ and let x be partitioned into 3 parts corresponding to the partitioning of Λ. Then, force written in terms of x is

$$f = x_1'\Lambda_1 x_1 - x_2\Lambda_2 x_2$$

In this form, each term $x_1'\Lambda_1 x_1$ and $x_2'\Lambda_2 x_2$ can only produce a result greater than zero for $x_1 \neq 0$ and $x_2 \neq 0$, respectively.

One way to demonstrate the "if" part of the lemma is to pick $x_1 \neq 0$ and $x_2 = 0$ to produce a positive result and $x_2 \neq 0$ and $x_1 = 0$ to produce a negative result.

For the "only if", assume that M is indefinite and results of only one sign can be produced. Then, either Λ_1 or Λ_2 must be of dimension 0, in which case, M is either definite or semidefinite.

The result from Lemma 5.1 can be extended to show the necessity of Theorem 5.1. Parameterize the force f as

$$f = \begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix} \tag{5.13}$$

where c_1 and c_2 are arbitrary real numbers. If there is an i that realizes this force,

$$f'f = c_1 i' M_1 i + c_2 i' M_2 i = c_1^2 + c_2^2 \tag{5.14}$$

Since $(c_1^2 + c_2^2)$ is always greater than or equal to zero, for the above to be true,

$$i'(c_1 M_1 + c_2 M_2)i \geq 0 \tag{5.15}$$

Alternatively, a force of

$$f = -\begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix} \tag{5.16}$$

could also be desired. In this case,

$$f'f = -c_1 i' M_1 i - c_2 i' M_2 i = c_1^2 + c_2^2 \tag{5.17}$$

implying

$$i'(c_1 M_1 + c_2 M_2)i \leq 0 \tag{5.18}$$

By Lemma 5.1, for (5.15) and (5.18) to both be realized, $(c_1 M_1 + c_2 M_2)$ must be indefinite. Since c_1 and c_2 are arbitrary real numbers, $(c_1 M_1 + c_2 M_2)$ must be indefinite for all real c_1, c_2 for all forces to be realized.

Note that the same proof of necessity would also apply to a system with arbitrarily many force directions: If the forces produced by an actuator are characterized by $(M_1, \ldots, M_k)$, every possible linear combination of M matrices must be indefinite for all possible forces to be produced.

5.2.1 Test of indefiniteness for two force directions

Although the extension of Theorem 5.1 to more than two force directions is difficult to test, the 2 d.o.f. case can easily be evaluated. Since the inertia of a matrix is not changed by scaling the matrix by a non-zero constant, an equivalent of Theorem 5.1 is that

$$Q[s] \equiv (M_1 + sM_2) \tag{5.19}$$

must be indefinite for all s. However, $Q[s]$ need not be exhaustively tested over all s. The eigenvalues of a polynomial matrix like Q are known to vary smoothly with s; the only possible changes in sign (and thus changes from indefinite) occur at values of s where $Q[s]$ has a zero eigenvalue. These values of s can be found by solving for s that satisfies

$$\det Q[s] = 0 \tag{5.20}$$

for s. Note that $\det Q[s]$ is at most an n^{th} order polynomial in s. Then, if M_1 and M_2 are indefinite, and every $Q[s]$ for s satisfying (5.20) is indefinite, Q[s] is indefinite for all s and Theorem 5.1 is satisfied.

This test will work unmodified in most cases; however, there is the pathological case in which the determinant of $Q[s]$ is zero for all s. The same sort of test can be used in this case as well, but the zero eigenvalues must be factored out so that the characteristic polynomial is only zero at the zero crossings of the eigenvalues. The zero eigenvalues are factored out by first forming the more general determinant $det(Q[s]-\lambda I)$. This is the characteristic polynomial that would be used to find the eigenvalues of $Q[s]$ at a particular value of s. This polynomial is then divided by λ to eliminate the zero eigenvalue. Since only the zero crossings are of interest, λ is then set to zero, and $Q[s]$ is tested at the roots of the reduced-order polynomial.

For example, consider the case

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad M_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

The matrix $Q[s]$ associated with these two matrices is always singular, but M_1 and M_2 do not share a common zero eigenvalue that might merely be reduced out. Forming $det(Q[s]-\lambda I)$ yields

$$\lambda - \lambda^3 + \lambda s^2 = 0$$

Factoring out a λ and setting λ to zero yields

$$s^2 + 1 = 0$$

The roots are purely imaginary, indicating no zero crossings. Since there are no zero crossings, the signs of the eigenvalues do not change from the ± 1 pair of M_1. M_1 and M_2 satisfy the necessary conditions for all forces to be produced.

5.3 Conditions for a solution realizable with finite current slew rate

Because the actuator currents are realized by applying voltages, the requested change in current with respect to time, $\frac{di}{dt}$, must be finite (see Appendix B). As long as a finite $\frac{df}{dt}$ is requested, $\frac{di}{dt}$ is finite if every element in $\frac{di}{df}$ is finite.

If a proposed inverse $i[f]$ is in hand, it is easy enough to check that the gradient of $i[f]$ is finite everywhere in the range of magnitudes of interest. If the inverse mapping $i[f]$ is not known, one might instead consider if, for a given point, there exists a continuous current trajectory leading away from the point in every possible force direction.

Lemma 5.2 *For an actuator with a current-to-force relation described by (1.13), a current vector i is on a finite current slew rate realizable solution manifold only if the matrix*

$$\frac{df}{di} = 2 \begin{bmatrix} i'M_1 \\ \vdots \\ i'M_k \end{bmatrix} \equiv 2H \tag{5.21}$$

has a rank of k.

If H is not of rank k, then the force cannot be modified in any direction with a component in the null space of HH'.

It has been shown in the literature [AYT84] that Theorem 5.1 is a sufficient as well as necessary condition for all forces for be produced for $k = 2$ and $n > 2$. However, this condition admits many cases that are physically unrealizable. For example, consider

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \qquad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.22}$$

This form arises from three-pole radial magnetic actuators. In this case, the M matrices are 3×3 because an independent coil is wound on each leg. There is a common null space in M_1 and M_2 caused by conservation of flux constraints. The eigenvalues of $Q[s] \equiv M_1 + sM_2$ are 0 and $\pm\sqrt{1+s^2}$. Regardless of the choice of s, there is always one positive and one negative eigenvalue. Matrix $Q[s]$ is indefinite for all s, and all forces are therefore realizable. However, when the force relations

$$\begin{aligned} f_1 &= i_1^2 - i_2^2 \\ f_2 &= 2i_1 i_2 \end{aligned} \tag{5.23}$$

are examined, it is apparent that the only way to realize a zero force is with $i_1 = i_2 = 0$. If i_1 and i_2 are zero, the resulting $H[i]$ is also zero and therefore singular: even though M_1 and M_2 can produce every force, they cannot produce them without requiring an infinite current slew rate at $f = 0$.

5.4 General realizability condition

A more restrictive condition is therefore needed that tests not only if a set of M matrices can realize all forces, but if those forces can be produced by a scheme that requires a finite current slew rate (that is, obeys Lemma 5.2 at every point). This condition is:

Theorem 5.2 *Every possible set of forces can be obtained by a continuous manifold with finite gradients if and only if there exists a current vector i_o such that*

$$\begin{gathered} i_o' M_1 i_o = 0 \\ \vdots \\ i_o' M_k i_o = 0 \end{gathered}$$

and

$$H[i_o] \equiv \begin{bmatrix} i_o' M_1 \\ \vdots \\ i_o' M_k \end{bmatrix}$$

is of rank k.

Necessary: If i_o does not exist, Lemma 5.2 cannot be satisfied at $f = 0$. If Lemma 5.2 is not satisfied, there are some directions in which the differential current required to cause a differential change from zero force is infinte The actuator is subject to slew rate limiting about zero force if no i_o exists.

Sufficient: The strategy for showing sufficiency is to show that if i_o exists, a path can be created from zero force to any force within a finite ball about zero force. If every force inside the ball can be realized, then a force of with a magnitude lying outside the ball can be realized by simply scaling the currents required to produce the largest force inside the ball in the same direction as the desired force.

Define current i to be

$$i = \frac{1}{\varepsilon} i_o + \varepsilon i_1 \tag{5.24}$$

define current i_1 to be

$$i_1 = \frac{1}{2} H[i_o]' (H[i_o] H[i_o]')^{-1} z \equiv \Phi z \tag{5.25}$$

where z is a vector of the same dimension as f. Substitute the definition of i into the force relations (1.13):

$$\begin{aligned} f_j &= i' M_j i \\ &= (\frac{1}{\varepsilon} i_o + \varepsilon i_1)' M_j (\frac{1}{\varepsilon} i_o + \varepsilon i_1) \\ &= (\frac{1}{\varepsilon})^2 i_o' M_j i_o + 2 i_o' M_j i_1 + \varepsilon^2 i_1' M_j i_1 \end{aligned} \tag{5.26}$$

Since $(i_o' M_j i_o = 0)$ by definition,

$$f_j = 2i_o' M_j i_1 + \varepsilon^2 i_1' M_j i_1 \tag{5.27}$$

The current-to-force relationships for each direction characterized by (5.27) can be combined into one expression as:

$$f = 2H[i_o]i_1 + \varepsilon^2 \left\{ \begin{matrix} i_1' M_1 i_1 \\ \vdots \\ i_1' M_k i_1 \end{matrix} \right\} \tag{5.28}$$

Substituting from (5.25) for i_1 yields:

$$f = z + \varepsilon^2 \delta \tag{5.29}$$

where

$$\delta = \left\{ \begin{matrix} z'\Phi' M_1 \Phi z \\ \vdots \\ z'\Phi' M_k \Phi z \end{matrix} \right\} \tag{5.30}$$

The first task is to get a bound on the magnitude of z required to produce a given magnitude of f. Using (5.29) and the triangle inequality,

$$\begin{aligned} |z| &= |f - \varepsilon^2 \delta| \\ &\leq |f| + \varepsilon^2 |\delta| \end{aligned} \tag{5.31}$$

$$\tag{5.32}$$

Any individual entry, δ_j, in δ will be less than or equal to $\bar{\sigma}[\Phi' M_j \Phi]|z|^2$ by definition of maximum singular value. A bound on $|\delta|$ would then be $c|z|^2$, where c^2 is defined as

$$c^2 = k \left(\max_j \bar{\sigma}[\Phi' M_j \Phi] \right) \tag{5.33}$$

Substituting into (5.31),

$$|z| \leq |f| + (\varepsilon c |z|)^2 \tag{5.34}$$

This expression can be solved for $|z|$ to yield

$$\begin{aligned} |z| &\leq \frac{1 - \sqrt{1 - 4c^2\varepsilon^2 |f|}}{2c^2\varepsilon^2} \\ &\leq |f| + c^2\varepsilon^2 |f|^2 \quad \text{for small } \varepsilon \end{aligned} \tag{5.35}$$

The strategy is to realize arbitrary forces by starting from the initial condition $z = 0$ at zero force and integrating dz/df to reach the desired force. The change in force with respect to z is

$$\frac{df}{dz} = \left[I + 2\varepsilon^2 \begin{bmatrix} z'\Phi' M_1 \Phi \\ \vdots \\ z'\Phi' M_k \Phi \end{bmatrix} \right] \tag{5.36}$$

Gradient matrix df/dz must be inverted to obtain dz/df, which is then integrated along the path from initial condition to desired current. If there is no non-zero real vector x for which $x'(df/dz)x = 0$, then df/dz must be invertible. If the condition

$$2\varepsilon^2 \bar{\sigma} \begin{bmatrix} z'\Phi' M_1 \Phi \\ \vdots \\ z'\Phi' M_k \Phi \end{bmatrix} < 1 \tag{5.37}$$

is satisfied, there is no way for $(x'(df/dz)x = 0)$ to be true. Inside some arbitrary ball $|f| \leq f_{max}$, $|z|$ is bounded by (5.35). Knowing this bound, an ε can then be picked so that (5.37) is always true within the ball, and therefore df/dz always invertible inside the ball. It is then possible to start at $f = 0, z = 0$ and march along an arbitrary trajectory inside the ball, moving from force to force with finite gradients. For forces of magnitude greater than f_{max}, a solution with finite gradients results from a simple scaling of the i that realizes $|f| = f_{max}$.

This inverse scheme is most likely not the *best* smooth inverse possible, but the point is to show that the existence of i_o allows all forces to be produced, is necessary for an inverse with finite gradients, and has at least one inverse solution with finite gradients.

A great similarity between Theorem 5.2 for a general actuator and Proposition 122 can be noted. In each case, the existence of a biasing current vector is a necessary and sufficient condition for all forces to be produced.

5.4.1 Numerical discovery of i_o

If a valid i_o does exist, it is relatively easy to find numerically. To satisfy Theorem 5.2, a vector i_o must satisfy:

$$\begin{aligned} i_o' M_1 i_o &= 0 \\ &\vdots \\ i_o' M_k i_o &= 0 \\ i_o' i_o - 1 &= 0 \end{aligned} \tag{5.38}$$

Denote these conditions $F[i_o] = 0$ for short. The last condition is included so that the zero vector is discounted. An i_o that satisfies the conditions can then be found by a modified Newton-Raphson iteration.

Define matrix $\hat{H}$ as the H from Lemma 5.2 augmented by the row i'. If $F[i]$ is linearized about the j^{th} iteration, i_j,

$$F[i_k + \delta i] \approx F[i_k] + 2\hat{H}[i_k]\delta i \tag{5.39}$$

Setting the approximation of $F[i_k + \delta i]$ equal to zero and solving for the smallest δi that will satisfy the conditions yields:

$$\delta i = -\frac{1}{2}\hat{H}'(\hat{H}\hat{H}')^{-1}F[i_k] \tag{5.40}$$

The $k+1$ approximation of i_o is then

$$i_{k+1} = i_k + \delta i \tag{5.41}$$

This iteration usually converges very quickly when an i_o exists.

Note that this iteration relies upon the fact that H is of rank k to compute the subsequent i_{k+1}. Therefore, any vector converged upon by this iteration satisfies the condition that $H[i_o]$ is of rank k automatically.

Chapter 6

Inverse Solution – Bias Linearization

For a typical magnetic actuator, there are many more currents to be specified than forces to be produced. Any desired force might then be realized by many different sets of coil currents. Since many solutions might be possible, the task is not merely to find a set of currents that realizes every desired force, but to produce each force in the "best" possible way. In this chapter, the criteria for a "good" inverse is ease of implementation; an inverse in which each current is merely a linear function of force is desired. This criterion is reduced to a general mathematical problem form, and different methods of solving this problem are explored.

6.1 Formulation of the generalized bias linearization problem

In the Chapter 1, the bias linearization of a 2-horseshoe actuator was considered. By direct substitution, one can verify that the change of variables

$$\begin{aligned} i_1 &= \tfrac{1}{2\sqrt{c}}(\hat{\imath}_o + \hat{\imath}_c) \\ i_2 &= \tfrac{1}{2\sqrt{c}}(\hat{\imath}_o - \hat{\imath}_c) \end{aligned} \tag{6.1}$$

and the current to force relation

$$f = c(i_1^2 - i_2^2) \tag{6.2}$$

yields the bilinear form

$$f = \hat{\imath}_o \hat{\imath}_c \tag{6.3}$$

If $\hat{\imath}_o$ is held constant, force is a linear function of $\hat{\imath}_c$.

For an actuator that produces force in multiple directions, one desires a change in variables that results in a bias current, $\hat{\imath}_o$, and a control current associated with each force direction, $\hat{\imath}_{c1}, \ldots, \hat{\imath}_{ck}$. This change of variables should be chosen to transform the force relations into the form

$$\begin{aligned} f_1 &= \hat{\imath}_o \hat{\imath}_{c1} \\ &\vdots \\ f_k &= \hat{\imath}_o \hat{\imath}_{ck} \end{aligned} \tag{6.4}$$

in terms of the transformed currents. The desired result can be written in matrix form as

$$f_j = \hat{\imath}' \hat{M}_j \hat{\imath} \tag{6.5}$$

for each force direction where

$$\hat{\imath} = \left\{ \begin{array}{c} \hat{\imath}_o \\ \hat{\imath}_{c1} \\ \vdots \\ \hat{\imath}_{ck} \end{array} \right\} \tag{6.6}$$

and $\hat{M}_j$ is a $(k+1)\times(k+1)$ matrix filled with zeros except for the $\{j+1,1\}$ and $\{1,j+1\}$ entries, which are equal to $\frac{1}{2}$.

For example, the $\hat{M}$ matrix corresponding to the 2-horseshoe case is

$$\hat{M}_1 = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}$$

and $\hat{i} = \{\hat{i}_o, \hat{i}_c\}^T$. Substituting into (6.5) yields $f = \hat{i}_o \hat{i}_c$.

Analogous to the 2-horseshoe case, a linear transformation between $\hat{i}$ and i is desired. This transformation will be denoted

$$i = W\hat{i} \tag{6.7}$$

For example, the transformation for the horseshoe case (6.5) can be written in matrix form as

$$i = W\hat{i} = \frac{1}{2\sqrt{c}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \hat{i} \tag{6.8}$$

For the general case, 1 bias current and k control currents are mapped into n coil currents, implying that W is an $n\times(k+1)$ matrix. Substituting (6.7) into the generalized force relations (1.13) yields

$$\begin{aligned} f_1 &= \hat{i}'W'M_1W\hat{i} \\ &\vdots \\ f_k &= \hat{i}'W'M_kW\hat{i} \end{aligned} \tag{6.9}$$

If W transforms the force relations into the desired form (6.5), eqs. (6.5) and (6.9) can set equal to one another, producing

$$\begin{aligned} \hat{i}'W'M_1W\hat{i} = f_1 &= \hat{i}'\hat{M}_1\hat{i} \\ &\vdots \\ \hat{i}'W'M_kW\hat{i} = f_k &= \hat{i}'\hat{M}_k\hat{i} \end{aligned} \tag{6.10}$$

For all forces to be realized, (6.10) must apply regardless of the choice of $\hat{i}$:

$$\begin{aligned} W'M_1W &= \hat{M}_1 \\ &\vdots \\ W'M_kW &= \hat{M}_k \end{aligned} \tag{6.11}$$

Equation (6.11) is the generalized bias linearization problem. To control an actuator using a bias linearization scheme, the task is to find a matrix W that satisfies (6.11). If such a W is found, the desired form (6.4) is realized by transformation (6.7), and an inverse mapping that realizes any force is

$$i = W \begin{Bmatrix} \hat{i}_o \\ f_1/\hat{i}_o \\ \vdots \\ f_k/\hat{i}_o \end{Bmatrix} \tag{6.12}$$

6.1.1 Note on choice of reference frame

If a given actuator can be bias linearized with the M matrices derived in one set of orthogonal coordinates, the actuator can be bias linearized in any set of orthogonal coordinates. This fact can be shown by considering the transformation of force in the "A" reference frame, f_A to force in the "B" reference frame, f_B via the orthonormal transformation matrix ${}^BT^A$:

$$\begin{aligned} f_B &= {}^BT^A f_A \\ &= \hat{i}_o\, {}^BT^A \begin{Bmatrix} \hat{i}_{c1} \\ \vdots \\ \hat{i}_{c_k} \end{Bmatrix} \end{aligned} \tag{6.13}$$

Force in the “B” frame is still linear in the control current magnitudes. Linearization matrix W can then be converted to the new reference frame by

$$W'_B = \begin{bmatrix} 1 & 0 \\ 0 & {}^B T^A \end{bmatrix} W'_A \tag{6.14}$$

6.2 A necessary condition for bias linearization

In the literature on quadratic forms [Lam73], a vector x is known as an *isotropic vector* with respect to a symmetric matrix M if

$$x' M x = 0 \tag{6.15}$$

Likewise, two isotropic vectors, x_1 and x_2 form a *totally isotropic space* if every linear combination of x_1 and x_2 is also an isotropic vector. It is important to note that most collections of isotropic vectors do not form an isotropic space.

An isotropic vector, x, can achieve a zero result with respect to M in two ways:

1. x is an eigenvector of M corresponding to a zero eigenvalue of M.
2. x produces equal and opposite contributions from a positive and a negative eigenvalue of M. For example, consider

$$M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; \qquad x = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$$

   A zero is produced by adding together a 1 from the first diagonal entry and a -1 from the second diagonal entry.

A totally isotropic space results from a combination of sets of paired eigenvalues and zero eigenvalues. For example, consider the M matrix

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

One possible basis that spans a totally isotropic space is the columns of

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The first column takes advantage of the first eigenvalue pair, the second column of the second eigenvalue pair, and the last column the zero eigenvalue. Any linear combination of these vectors is also an isotropic vector.

A relatively intuitive result that is proven in [Lam73] is that the dimension of the largest totally isotropic space realizable for any particular matrix is the sum of the number of positive and negative eigenvalue pairs plus the number of zero eigenvalues. The dimension of the maximal totally isotropic space will be denoted w. For example, if some M matrix had the eigenvalues:

$$\{1, -1, 2, -3, 4, 5, 0, 0\}$$

the dimension of the maximal totally isotropic space, w, would be equal to 4, since there are two $\pm$ eigenvalue pairs and two zero eigenvalues.

Generally, however, there are an infinite number of maximal totally isotropic spaces. Consider the matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

In this case, there is one eigenvalue pair, so the dimension of the maximal totally isotropic space is one. However, the vector

$$x = \begin{Bmatrix} \sin\theta \\ \cos\theta \\ 1 \end{Bmatrix}$$

is isotropic for any choice of θ; there are in infinite number of one-dimensional maximal totally isotropic spaces. However, since the dimension of the maximal totally isotropic space is one, no two of this infinite number of spaces can be combined to form a two dimensional isotropic space.

Upon examining (6.11), it can be seen that the 2^{nd} through $(k+1)^{th}$ columns in W are the basis of a k-dimensional totally isotropic space with respect to every M matrix characterizing a particular actuator. For a k-dimensional isotropic space to exist for every M, every M must have $w \geq k$. Furthermore, it was shown in the previous section that if an actuator can be linearized in one reference frame, it can be linearized in any reference frame; therefore any linear combination of M matrices must have $w \geq k$.

Consider the special case when all the M matrices have a common null space denoted by z. Let the dimension of this common null space be denoted $\hat{w}$. For every vector x in z,

$$M_j x = 0; \quad j = 1, \ldots, k; \quad x \in z$$

The last k columns of W can have no component in z.

> *Proof:* Assume that there is a W that satisfies (6.11) and, without loss of generality, let the second column of W be an element of z. From (6.11), $W_1' M_1 W_2 = 1$. However, since W_2 is in z, $W_1' M_1 W_2 = 0$. There is a contradiction, so the assertion must be true.

If $\hat{w} > 0$, the minimum necessary w for bias linearization must then be $w \geq k + \hat{w}$. Although z is a totally isotropic space, it cannot contribute to W.

These results can be summarized as:

Theorem 6.1 *Consider a set of real symmetric matrices $M_1, \ldots, M_k$ that satisfies Theorem 5.1 necessary for the realizability of all forces. Let w denote the dimension of the maximal totally isotropic space of a particular matrix, and let $\hat{w}$ denote the dimension of any null space common to all of the M matrices. For a set of bias linearizing currents to exist for the set of M's, every linear combination of M matrices must have $w \geq k + \hat{w}$.*

6.2.1 2 d.o.f. testing of the necessary condition

As with Theorem 5.1, only the two degree-of-freedom case is feasible to test. The testing of Theorem 197 proceeds along the same basis as the testing of Theorem 5.1 outlined in § 5.2.1. For the 2-d.o.f. case, a matrix $Q(s)$ is defined as

$$Q(s) \equiv (M_1 + sM_2) \tag{6.16}$$

where s is an arbitrary real number. As in § 5.2.1, if the necessary condition applies for all s, the condition applies for all linear combinations of M_1 and M_2. Since Theorem 5.1 is a precondition for Theorem 197, all values of s where zero crossing occur must be found in accordance with the procedures in § 5.2.1, and Theorem 5.1 must be satisfied at each zero crossing as well as for M_1 and M_2. Then, the conditions for Theorem 197 can be tested. Since, once again, only the signs of the eigenvalues are of interest, Theorem 197 is sufficiently tested at only a finite number of points. Since there are several ways in which w can be composed, either of zero eigenvalues or of paired eigenvalues, Theorem 197 must be tested at every zero crossing s, and at one point between each zero crossings. If $w > k + \hat{w}$ at each one of these test points, Theorem 197 is satisfied.

6.3 Numerical determination of W

A useful way to obtain W matrices satisfying (6.11) is to find them numerically. An obvious but rather expensive way of proceeding is to define the function

$$J(W) = \sum_{j=1}^{k} \|W' M_j W - \hat{M}_j\|_2^2 \tag{6.17}$$

Any W that makes J=0 is clearly a solution to (6.11). Many other such functions are equally valid, but this particular cost function is a fourth-order polynomial in the elements of W. Gradient methods can then be employed to minimize J.

Alternatively, a Newton-Raphson method can be used that closely parallels the method used to find i_o in § 5.4.1. This method finds a valid W without explicitly minimizing (6.17). Presently, this scheme will be developed only for the case of radial magnetic bearings, but the generalization to any number of force components is relatively straightforward.

Equation (6.11) describes $k(k^2+3k+2)/2$ constraint equations on $(k+1)n$ unknowns; almost always, the number of unknowns is more than the number of equations. If there were the same number of unknowns and constraints, the usual Newton-Raphson method could be used. Here, the choice of the smallest possible constraint satisfying step specifies a particular solution out of many possible step directions.

For the two dimensional case, the three columns in W will be represented as

$$W \equiv [W_b | W_{c1} | W_{c2}] \tag{6.18}$$

The columns of W can then be stacked on top of one another to make one vector with $3n$ components. The generalized bias linearization problem (6.11) can then be re-written as a series of quadratic forms in this extended vector. Define H as

$$\mathsf{H} \equiv \begin{bmatrix} W_b' M_1 & 0 & 0 \\ W_b' M_2 & 0 & 0 \\ 0 & W_{c1}' M_1 & 0 \\ 0 & W_{c1}' M_2 & 0 \\ 0 & 0 & W_{c2}' M_1 \\ 0 & 0 & W_{c2}' M_2 \\ 0 & W_{c2}' M_1 & W_{c1}' M_1 \\ 0 & W_{c2}' M_2 & W_{c1}' M_2 \\ W_{c2}' M_1 & 0 & W_b' M_1 \\ W_{c1}' M_2 & W_b' M_2 & 0 \\ W_{c2}' M_2 & 0 & W_b' M_2 \\ W_{c1}' M_1 & W_b' M_1 & 0 \end{bmatrix} \tag{6.19}$$

Then, (6.11) can be written as

$$\mathsf{H} \left\{ \begin{array}{c} W_b \\ W_{c1} \\ W_{c2} \end{array} \right\} - \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \right\} = 0 \tag{6.20}$$

or more succinctly as

$$F[w] = 0 \text{ where } w \equiv \begin{Bmatrix} W_b \\ W_{c1} \\ W_{c2} \end{Bmatrix} \tag{6.21}$$

This set of equations is analogous to the set of equations (5.38) that a valid i_o must satisfy. The Taylor expansion of $\mathsf{H}\,[w]w$ about some particular vector w_j is

$$\mathsf{H}\,[w]w \approx H[w_j]w_j + 2H[w_j]\delta w + \ldots \tag{6.22}$$

The smallest change in w necessary to solve (6.21) is then calculated by setting the first-order expansion of $\mathsf{H}\,[w]$ equal to c and solving for δw using the Moore-Penrose pseudoinverse:

$$\delta w = -\frac{1}{2}\mathsf{H}\,'[w_j]\left(\mathsf{H}\,[w_j]\mathsf{H}\,'[w_j]\right)^{-1}F[w_j] \tag{6.23}$$

The next approximation for i is

$$w_{j+1} = w_j + \Delta\delta w_j \tag{6.24}$$

where Δ is a stepsize less than or equal to 1. This iteration will usually converge quickly with $\Delta = 1$.

As noted in § 5.4.1, this algorithm only converges to solutions for which $\mathsf{H}\,\mathsf{H}\,'$ is of full rank. However, valid solutions for the generalized bias linearization problem can exist where $\mathsf{H}\,\mathsf{H}\,'$ is rank deficient. This occurs if one row of W is coincident with a zero eigenvalue of one of the M matrices. If this occurs, several rows in H become identically equal to zero, and $\mathsf{H}\,\mathsf{H}\,'$ loses rank. In most cases where a solution is sought numerically, there are no solutions for which $\mathsf{H}\,\mathsf{H}\,'$ is rank deficient, and the algorithm works well unmodified. The rank deficiency usually arises in situations where the actuator could be more properly considered as a collection of smaller degree-of-freedom actuators rather than one higher degree-of-freedom actuator. An example would be multiple sets of opposed horseshoes, each pair of which can be considered an independent one-dimensional actuator. Even in this case, however, the algorithm can be modified so that it will converge properly on solutions that have a singular $\mathsf{H}\,\mathsf{H}\,'$.

As the algorithm converges on a solution with a rank-deficient H at the solution, $\mathsf{H}\,\mathsf{H}\,'$ becomes increasingly poorly conditioned with each step. However, the algorithm can converge quite close to a candidate W that nearly satisfies (6.11) without $\mathsf{H}\,\mathsf{H}\,'$ becoming singular. When $\mathsf{H}\,\mathsf{H}\,'$ is judged to be effectively singular (that is, difference between the maximum and minimum singular values is greater than the computer's precision), the iterative scheme should proceed via minimizing (6.17) by a gradient descent rather than by continuing with the modified Newton-Raphson scheme.

6.4 Analytical determination of W

In some special but practically important cases, manifolds of solutions for W can be obtained without the need for a numerical search. Instances where this analytical scheme can be successfully used include symmetric radial magnetic bearings with an even number of legs but having arbitrary windings, and the magnetic stereotaxis system.

Recall that any quadratic formed with an anti-symmetric matrix A is equal to zero:

$$i'Ai = 0 \tag{6.25}$$

Proof: Since $i'Ai$ is a scalar, $i'Ai$ equals its own transpose: $i'Ai = i'A'i$. But since A is anti-symmetric, $i'Ai = -i'Ai$. This can only be true for all i if $iA'i = 0$ for all i.

Therefore, it can be concluded that the quadratic formed from a symmetric matrix M and any anti-symmetric matrix A is equal to the quadratic formed from M alone:

$$i'(M+A)i = i'Mi + i'Ai = i'Mi \tag{6.26}$$

However, $(M+A)$ can have very different properties from M. In particular, an A might be chosen such that $(M+A)$ has a lower rank than M. For example, consider:

$$M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad\qquad A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

In this case, M is clearly of rank 2. However

$$M+A = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

The columns of $(M+A)$ are clearly linearly dependent; the rank has been reduced from 2 to 1.

If an anti-symmetric matrix is found that produces a reduced rank $(M+A)$, the $(M+A)$ can be broken down into 2 components, each of which has less than N rows. Continuing with the previous example,

$$M+A = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \end{bmatrix}$$

To denote this decomposition in the general case, the notation

$$(M+A) = B'D \tag{6.27}$$

will be used.

The existence of this decomposition is crucial to the analytical determination of bias linearization currents. Using this decomposition, the quadratic force relations can be decomposed into a linear system of equations with the quadratic nature of the problem encapsulated in an arbitrarily chosen vector that enters in on both sides of the otherwise linear equation. For the example problem,

$$f = \begin{Bmatrix} i_1 & i_2 \end{Bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{Bmatrix} i_1 \\ i_2 \end{Bmatrix}$$

can be re-written as the linear system of equations

$$\begin{bmatrix} 1 & 1 \\ q & -q \end{bmatrix} \begin{Bmatrix} i_1 \\ i_2 \end{Bmatrix} = \begin{Bmatrix} q \\ f \end{Bmatrix}$$

where q is an arbitrarily chosen number. The left-hand side of the equation can then be inverted to yield:

$$\begin{Bmatrix} i_1 \\ i_2 \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & \frac{1}{q} \\ 1 & -\frac{1}{q} \end{bmatrix} \begin{Bmatrix} q \\ f \end{Bmatrix}$$

A manifold of solutions indexed by q has resulted. By choosing q to be any particular constant, a linear rule for currents that realize any force results. By comparing to (6.12), the W matrix associated with this solution is

$$W = \frac{1}{2} \begin{bmatrix} q & \frac{1}{q} \\ q & -\frac{1}{q} \end{bmatrix}$$

This result is just the same as (6.8), examined earlier in the chapter. The arbitrary constant q simply indexes all possible bias current levels.

The same sort of decomposition used in the one degree-of-freedom case can be used to determine a manifold of solutions for more complicated problems. Consider a higher degree of freedom actuator characterized by $(M_1, \ldots, M_k)$. Assume that anti-symmetric matrices $(A_1, \ldots, A_k)$ can be chosen such that

$$\begin{aligned} M_1 + A_1 &= B'D_1 \\ &\vdots \\ M_k + A_k &= B'D_k \end{aligned} \tag{6.28}$$

The current-to-force relationship could then be re-written as

$$\begin{bmatrix} B \\ q'D_1 \\ \vdots \\ q'D_k \end{bmatrix} i = \begin{Bmatrix} q \\ f_1 \\ \vdots \\ f_k \end{Bmatrix} \tag{6.29}$$

where q is now an arbitrary vector rather than an arbitrary scalar. If the simultaneous rank reduction of each matrix is of k or greater dimension, the left-hand side of (6.29) can be inverted for any q for which the left-hand side is non-singular:

$$i = \mathsf{G}[q]'(\mathsf{G}[q]\mathsf{G}[q]')^{-1} \begin{Bmatrix} q \\ f_1 \\ \vdots \\ f_k \end{Bmatrix} \quad \text{where} \quad \mathsf{G}[q] = \begin{bmatrix} B \\ q'D_1 \\ \vdots \\ q'D_k \end{bmatrix} \tag{6.30}$$

Equation (6.30) is a family of linear inverses to the current to force relations indexed by the choice of q.

The question then becomes: "when and how can a set of anti-symmetric A matrices be chosen that yields the decomposition in (6.28)?" The ability to realize (6.28) is intimately related to the existence of a k or greater dimensional totally isotropic space common to all M characterizing a given actuator.

Theorem 6.2 *A set of current to force realtions characterized by $M_1, \ldots, M_k$ can be decomposed into the linear form*

$$\mathsf{G}[q]i = \begin{Bmatrix} q \\ f_1 \\ \vdots \\ f_k \end{Bmatrix}$$

in which $\mathsf{G}[q]$ has less than ore equal to n rows if and only if there exits a k or greater dimensional subspace of currents i spanned by the rows of a matrix P that satisfies

$$\begin{gathered} P'M_1P = 0 \\ \vdots \\ P'M_kP = 0 \end{gathered}$$

Necessary: Assume that (6.29) can be realized. Matrix B is then a rectangular matrix with l rows and $l \leq n-k$. Let P be an $n \times (n-l)$ matrix whose columns are perpendicular to the rows of B. Such a matrix can always be constructed via Gramm-Schmidt orthogonalization [HJ85]. Since $P'B$ always yields a zero result, every $P'M_jP = \frac{1}{2}(P'BDP + P'DB'P) = 0$.

Sufficient: Let anti-symmetric matrix A_j be

$$A_j = -P(P'P)^{-1}P'M_j + M_jP(P'P)^{-T}P' \tag{6.31}$$

Then,

$$P'(M_j + A_j) = P'M_j - P'P(P'P)^{-1}P'M_j + P'M_jP(P'P)^{-T}P' = 0 \tag{6.32}$$

Now, define the matrix

$$G \equiv [M_1 + A_1 | \cdots | M_k + A_k] \tag{6.33}$$

Note that $P'G = 0$ due to the choice of A's. Let the singular value decomposition of G be denoted

$$G = \mathsf{U}'\Lambda\mathsf{V} \tag{6.34}$$

where the entries in each matrix associated with zero singular values have been omitted. Since the $P'G = 0$, U will have at most $n - \text{rank}(P)$ rows. It can then be noted that defining

$$B \equiv \mathsf{U} \tag{6.35}$$

and partitioning

$$[D_1|\cdots|D_k] \equiv \Lambda\mathsf{V} \tag{6.36}$$

yields the desired form (6.29).

Corollary 6.1 *A solution to the generalized bias linearization problem exists only if the current-to-force relations can be put in the form of (6.29).*

Proof: If a solution to the bias linearization problem exists, the last k columns of W are the basis for a k dimensional totally isotropic space common to all M's. These last k columns can be used as the matrix P described in Theorem 6.2; (6.29) can then be constructed via the "sufficient" part of this theorem.

Basically, if an isotropic space of dimension k or greater is known, a family of solutions to the bias linearization problem can be found merely by the inversion of a small matrix. For the case of radial magnetic bearings, to be discussed in detail in later chapters, high-dimensional totally isotropic spaces can be obtained by inspection. For the case of the magnetic stereotaxis system, the derivation of the force-current relationship serendipitiously leads directly to the decomposed form (6.29) (see § 4.2).

However, formulating the problem in the form of (6.29) has advantages even when a totally isotropic space is not known *a priori*. Even though a totally isotropic space must be found by a numerical search in this case, the result is a family of solutions rather than just the single solution obtained by directly solving (6.11) numerically.

To obtain a common totally isotropic space by a numerical search, it must first be noted that an orthonormal basis, P, for such a space must satisfy:

$$\begin{gathered} P'M_1P = 0 \\ \vdots \\ P'M_kP = 0 \\ P'P = I \end{gathered} \tag{6.37}$$

where I denotes the identity matrix. Matrix P must have at least k columns, but higher dimensional spaces can be sought by adding more columns to P. An initial guess for P is chosen randomly, and P can be determined by exactly the same modified Newton-Raphson iteration described in § 6.3.

In general, if there is one solution to the generalized bias linearization problem, many other solutions also exist. Choosing the "best" set of linearization currents is a highly implementation-specific question and is therefore addressed in subsequent chapters.

Chapter 7

Inverse Solution – Direct Optimization

In the previous chapter, the criterion for a desirable inverse was that the inverse should be easy to implement, *vis.* a linear relationship between the desired forces and the required currents. However, requiring a linear relationship between desired force and currents is overly restrictive; an actuator need not have a linear inverse for all desired forces to be realizable. In addition, bias linearization does not necessarily yield an inverse with optimal performance in terms of maximizing bearing load capacity or minimizing resistive power losses.

An alternate philosophy for choosing a particular inverse is to select the solution that optimizes some measure of performance while also realizing the desired forces. For the inverse to be physically realizable, it should also have the following properties [Gre96]:

- *All currents must go to a nominal bias value when the force requested is zero.* This requirement avoids the slew rate limiting problem at low force levels if the bias currents are appropriately selected (see Appendix B).
- *Coil currents should be a continuous function of force.* This requirement avoids jumps in required currents that would cause slew rate limiting problems away from $f = 0$.
- *The algorithm* [should be] *computationally quick and simple.* For a magnetic actuator to have adequate bandwidth, the throughput rate must be fast. The time spent solving the magnetic inverse problem should therefore not take up a large portion of the sampling interval in a digital controller implementation. An inverse computed off-line and stored in a look-up table for real-time used is assumed to be adequate.

7.1 Formulation of the generalized direct optimization problem

A natural candidate for a cost function to optimize is $i'\mathtt{Q}\,i$ where $\mathtt{Q}$ is a positive definite matrix used for weighting the currents relative to one another. Minimizing this cost would give, in a sense, the smallest current necessary to realize a given force. Another interpretation is that this cost function minimizes the resistive power losses necessary to produce a given force. Using this quadratic cost function, the formal definition of the generalized direct optimization problem is:

$$\min_i J(i) \equiv i'\mathtt{Q}\,i \tag{7.1}$$

$$\text{subject to} \quad \begin{aligned} i'M_1 i &= f_1 \\ &\vdots \\ i'M_k i &= f_k \end{aligned}$$

However, this formulation has an immediately apparent problem. At zero force, $i = 0$ satisfies the constraints while at the same time producing $J = 0$. Since $\mathtt{Q}$ is positive definite, zero is the lowest possible value of J; $i = 0$ is clearly the optimal solution at $f = 0$. The requirement of a non-zero current at zero force, necessary to avoid slew rate limiting, has been violated.

Consider instead the cost function

$$\min_i J(i) \equiv (i - i_o)' \mathsf{Q}\, (i - i_o) \tag{7.2}$$

$$\begin{aligned} \text{subject to} \quad i' M_1 i &= f_1 \\ &\vdots \\ i' M_k i &= f_k \end{aligned}$$

where i_o satisfies the same conditions outlined in § 5.4:

$$i_o' M_j i_o = 0 \;\; \forall \;\; j = 1, \ldots, k$$

and the matrix $H[i_o]$ is of rank k where

$$H[i_o] \equiv \begin{bmatrix} i_o' M_1 \\ \vdots \\ i_o' M_k \end{bmatrix}$$

As shown previously by Theorem 5.2, it is possible to control an actuator with finite current slew rate if and only if such a vector exists. For this cost function, $f = 0$ and $J = 0$ at $i = i_o$; current vector i_o therefore must be the optimal solution of i at $f = 0$. Away from $f = 0$, i_o becomes increasingly insignificant in comparison to i. As i gets larger,

$$(i - i_o)' \mathsf{Q}\, (i - i_o) \approx i' \mathsf{Q}\, i \tag{7.3}$$

The modified cost converges to the power-optimal cost for large i.

The problem defined by (7.2) may be adequate if there is a way of solving (7.2) that yields a smooth inverse mapping. Perhaps the best way to produce an inverse mapping in this case is through a continuation (or homotopy) approach. The optimal solution is known at $f = 0$. The idea is then to make small changes to i that produce a non-zero force but still are optimal in the sense of (7.2). Similar techniques have been used in the literature, particularly in the area of optimal power system studies [aAFVJ85].

The first step in developing this approach is to combine the desired force constraints into the cost function via scaling by Lagrange multipliers, denoted by λ [Fox71]:

$$\hat{J} \equiv (i - i_o)' \mathsf{Q}\, (i - i_o) + \lambda' (H[i] i - f_j) \tag{7.4}$$

The Lagrange multipliers can be thought of heuristically as representing a relative cost of satisfying the constraints. For an optimum, the partial derivatives of $\hat{J}$ with respect to both i and λ must be equal to zero:

$$\begin{array}{rcl} 2\mathsf{Q}\, (i - i_o) + 2H'[i] i & = & 0 \\ H[i] i - f & = & 0 \end{array} \tag{7.5}$$

Equation (7.5) is known as the "Kuhn-Tucker optimality conditions."

If a small change in force is desired, i should change in such a way that the change in force is realized while still satisfying the optimality conditions. Let s denote the distance along an arbitrarily chosen continuous trajectory originating at $f = 0$ in the space of desired forces, as illustrated in Figure 7.1. A small change in forces can be represented now by df/ds.

For the optimality conditions to be satisfied for a given df/ds, the total derivative of (7.5) with respect to s must be zero:

$$2 \begin{bmatrix} \mathsf{Q} + \sum_{j=1}^{k} \lambda_j M_j & H'[i] \\ H[i] & 0 \end{bmatrix} \left\{ \begin{array}{c} \frac{di}{ds} \\ \frac{d\lambda}{ds} \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ \frac{df}{ds} \end{array} \right\} \tag{7.6}$$

Equation (7.6) is a system of ordinary differential equations in s. On the right hand side, df/ds is specified by the choice of path through the k-dimensional space of f. The left-hand side can then be inverted

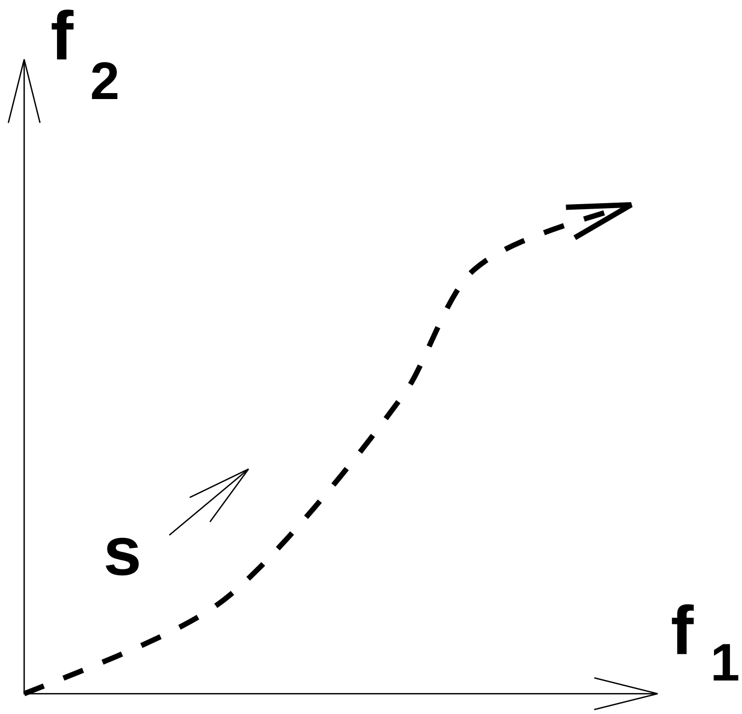


Figure 7.1: 2-d example of a trajectory out of $f = 0$

at any particular i and λ to yield the change in currents and Lagrange multipliers that correspond to any df/ds. An exposition by Bryson and Ho [BH69] indicates that this integration yields the same i and λ for a given f regardless of path as long as the left-hand side of (7.6) is always non-singular and the initial condition is itself a minimum.

Initial conditions must be supplied so that (7.6) can be integrated. The initial condition on current is $i = i_o$ at $f = 0$, since i_o is the optimal solution to (7.2) at zero force. However, the Lagrange multipliers, λ, are also functions of s, and an appropriate condition on λ must also be supplied at $f = 0$. The value of λ can be determined by considering the Kuhn-Tucker conditions (7.5) at the $f = 0$ point. Substituting $f = 0$ and $i = i_o$ into (7.5) yields

$$\begin{aligned} 2H'[i_o]\lambda &= 0 \\ 0 &= 0 \end{aligned} \tag{7.7}$$

The constraint equations in (7.5) are satisfied at $f = 0$ by definition of i_o. Recall that another condition on i_o is that $H[i_o]$ must be of rank k. An equivalent condition is that the columns of $H'[i_o]$ are linearly independent. Since the columns of $H'[i_o]$ must be linearly independent, no non-zero combination of columns can add up to zero; only $\lambda = 0$ will satisfy (7.7). The correct initial condition on λ is therefore $\lambda = 0$ at $f = 0$ so that the manifold tracked out of the zero force solution is an optimum. If some other initial condition is used for λ, a manifold will result that satisfies, the constraint equations; however, a manifold produced by $\lambda[0] \neq 0$ will not be optimal in the sense of (7.2).

An optimal inverse mapping is created by integrating (7.6) numerically along many different paths heading out of the origin, using $i = i_o$, $\lambda = 0$ as the initial condition at $f = 0$. For example, in a 2-force actuator, f can be parameterized in terms of s and an angle θ as

$$\begin{aligned} f_1 &= s\cos\theta \\ f_2 &= s\sin\theta \end{aligned} \tag{7.8}$$

The path is chosen so that the choice of θ corresponds to the direction of the force, and s corresponds to the magnitude of the force along that direction. To create an inverse mapping, (7.6) would be integrated from

$s = 0$ to some desired maximum force at a great enough number of θ's so that the inverse is suitably defined in the range of forces of interest.

This method relies on the fact that the inverse has a finite slope with respect to s to compute the inverse; therefore, any inverse obtained by this method will have the desired property of smoothness along each integration path. Unfortunately, it is not clear that the left hand side of (7.6) will always be non-singular for every possible set of M's and i_o's. However, as shown in examples in Chapter 9, this method can give smooth inverse mappings in the practically important case of 8-pole radial magnetic bearings.

7.2 1 degree of freedom example

As an example of the method, consider the 1 d.o.f. problem

$$f = \{ i_1 \quad i_2 \} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{Bmatrix} i_1 \\ i_2 \end{Bmatrix} \tag{7.9}$$

For this example, one can choose

$$i_o = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{7.10}$$

as a biasing vector. The cost function to be optimized is given by (7.4):

$$\hat{J} = (i_1 - c)^2 + (i_2 - c)^2 + \lambda(i_1^2 - i_2^2 - f) \tag{7.11}$$

where c is a constant that scales the magnitude of the vector. Taking derivatives with respect to i_1, i_2 and λ yields the optimality conditions:

$$\begin{aligned} 2(i_1 - c) + 2\lambda i_1 &= 0 \\ 2(i_2 - c) - 2\lambda i_2 &= 0 \qquad (7.12) \\ i_1^2 - i_2^2 - f &= 0 \qquad (7.13) \end{aligned}$$

Define f to be linear with s:

$$f[s] = s \tag{7.14}$$

Now, taking the total derivative of the optimality conditions with respect to s yields:

$$2 \begin{bmatrix} (1+\lambda) & 0 & i_1 \\ 0 & (1-\lambda) & -i_2 \\ i_1 & -i_2 & 0 \end{bmatrix} \begin{Bmatrix} \frac{di_1}{ds} \\ \frac{di_2}{ds} \\ \frac{d\lambda}{ds} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \tag{7.15}$$

This system of ordinary differential equations is then integrated numerically, using $i_1 = i_2 = c; \lambda = 0$ as the initial condition at $s = 0$.

The resulting currents for i_1 are shown in Figure 7.2. The required i_2 is the same plot reflected about $f = 0$. Several different magnitudes of c are considered: $c = 0.5, 0.375, 0.25, 0.125, 0.05$. As c goes to zero, the solution converges to $i_1 = \sqrt{f}$ for $f \geq 0$ and $i_1 = 0$ for $f < 0$ – the solution from (1.9) based solely on power losses. As the magnitude of i_o increases, the high slopes around $f = 0$ are smoothed out, yielding solutions that require greater current but are physically realizable.

7.2.1 Similarity between solutions for different magnitudes of i_o

If (7.6) is solved for one i_o, the solution for all other scalings of the same i_o can be inferred by rescaling. Consider the class of problems

$$\min_i J(i) \equiv (i - c\, i_o)' \mathsf{Q}\, (i - c\, i_o) \tag{7.16}$$

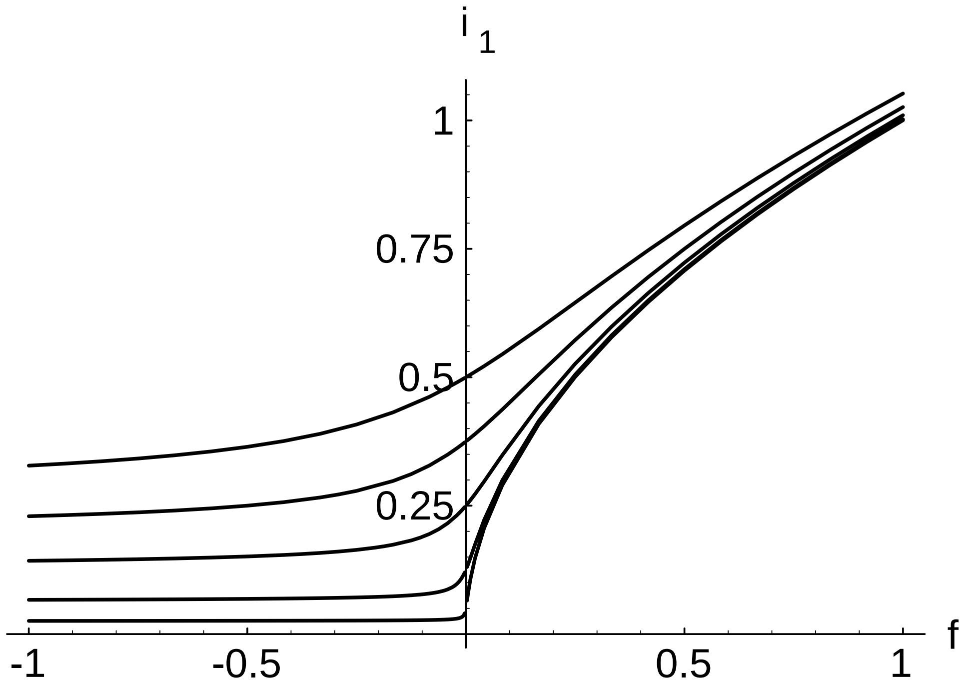


Figure 7.2: Solution for 1-d example at different bias magnitudes.

$$\text{subject to} \quad i'M_1 i = f_1$$
$$\vdots$$
$$i'M_k i = f_k$$

Define $cz = i$ and substitute to obtain

$$\min_i J(i) \equiv c^2 (z - i_o)' \mathsf{Q} \, (z - i_o) \tag{7.17}$$

$$\text{subject to} \quad z'M_1 z = f_1/c^2$$
$$\vdots$$
$$z'M_k z = f_k/c^2$$

By inspection of (7.17), one can conclude that $i[c\, i_o, f] = c\, i[i_o, f/c^2]$. Note, however, that this similarity applies only to actuators that have a linear B-H relationship. If the B-H curve is nonlinear, the current-to-force relations cannot be reduced to the simple $f = i'\, M\, i$ that allows for the similarity.

Chapter 8

Bias Linearization–Magnetic Bearings

In Chapter 6, a general representation of the bias linearization problem and several methods of obtaining solutions were presented without any reference to a specific application. The present chapter will address the application of these tools to the specific problem of magnetic bearings. First, the application of the analytical solution method to symmetric magnetic bearings with an even number of poles is considered. It is shown that all possible configurations of up to $n-1$ failed coils for a symmetric bearing with $2n$ coils and $n>1$ can be decomposed into a linear problem of the form of (6.29). Since there are typically many W matrices that satisfy the generalized bias linearization problem (6.11), a criterion is then presented by which a best W can be selected.

8.1 Analytical solution for symmetric bearings

In § 6.4, an analytical method was presented for solving for a linear inverse given the existence of a matrix P that spans a totally isotropic space common to all M matrices. In the general case, finding a valid P necessary for this method is as difficult as solving for a valid W via the methods addressed in § 6.3. However, for symmetric radial magnetic bearings with an even number of poles, many different candidates for P are available by inspection.

Consider first the symmetric radial case in which each pole is wound with an independent coil and the reluctances of iron sections of the flux path are assumed to be zero. Columns of P can be formed in at least three ways:

1. The first way is by specifying currents of equal magnitude but different sign in two coils that are 180^o apart. This case is illustrated in Figure 8.1. In this picture, there are currents of opposite sign but equal magnitude in coils 1 and 5. The result is that the only flux crossing from the stator onto the rotor goes through poles 1 and 5. Since the poles are located opposite to one another, the magnetic stress on the rotor integrates to zero. The column in P corresponding to this case is

$$\begin{Bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

2. The second way is to have equal currents in one set of opposed coils and equal currents of different

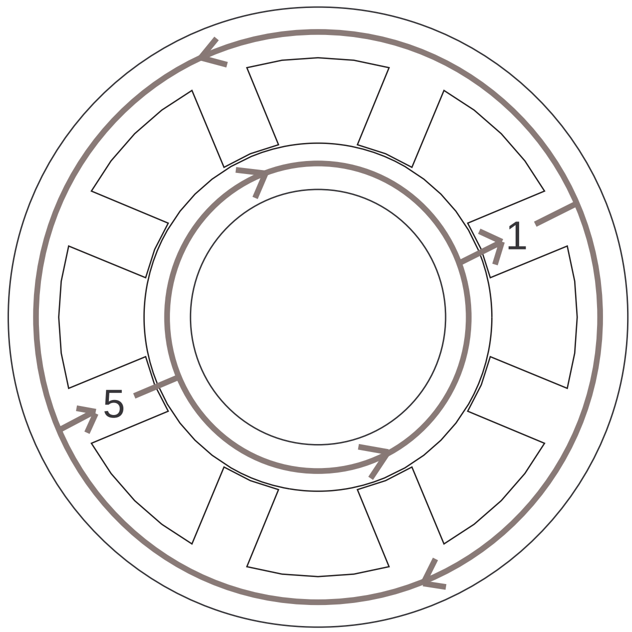


Figure 8.1: Flux resulting from opposite currents in opposed coils.

sign in another set of opposed coils, yielding a column of P of the form:

$$\begin{Bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \end{Bmatrix}$$

Again, the choice of current is such that flux only flows across the air gaps associated with coils that are turned on. This situation is illustrated in Figure 8.2.

3. Lastly, the vector $\{1, \dots, 1\}^T$ produces no flux across any gap due to conservation of flux constraints.

A high-dimensional isotropic space can be formed by including each coil in an isotropic space formed by either case 1 or case 2, and including the vector from case 3.

For example, in the 8-pole case, some valid P matrices would be

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{8.1}$$

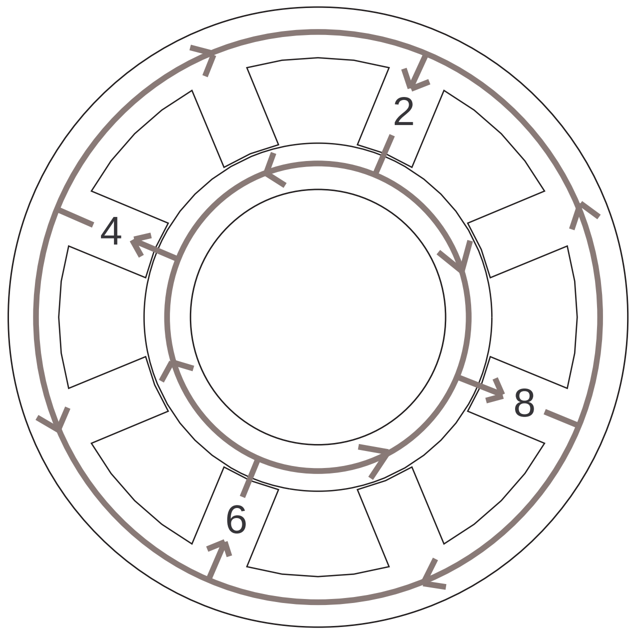


Figure 8.2: Flux resulting from two sets of opposed coils.

$$P = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{8.2}$$

$$P = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} \tag{8.3}$$

Note that different ways of “using” the poles result in totally isotropic spaces of different dimensions. The highest-dimension isotropic space can be achieved using the scheme in (8.1), yielding an $n/2+1$ dimensional totally isotropic space for an n pole bearing. Although a large number of different isotropic spaces can be formed by the above method, it is also possible that there may be other totally isotropic spaces that do not rely on the symmetries in case 1 and case 2; however, the above method yields a broad range of solutions.

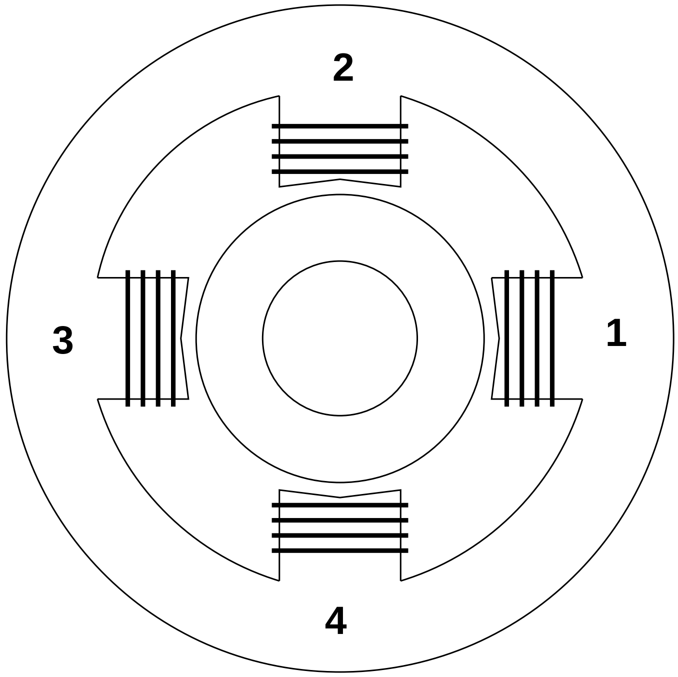


Figure 8.3: 4 pole symmetric radial bearing.

8.1.1 Fault tolerance

Once a particular P is chosen, the quadratic current-to-force relations can be decomposed into the linear form of (6.29):

$$\begin{bmatrix} B \\ q'D_1 \\ \vdots \\ q'D_k \end{bmatrix} i = \begin{Bmatrix} q \\ f_1 \\ \vdots \\ f_k \end{Bmatrix} \tag{6.29}$$

using (6.34)-(6.36). The resulting form (6.29) is a set of at most $(2+n-\mathrm{rank}[P])$ equations for n unknowns. If $\mathrm{rank}[P] > 2$, this system of equations is underdetermined, implying a potential for fault tolerance. If a coil fails, extra constraints can simply be added to this set of equation enforcing zero current in the failed coils. For the biggest dimension P matrix, P is of rank $n/2+1$, implying that extra constraints can be added to account for the failure of up to $n/2-1$ coils.

As a demonstration of the capacity for fault tolerance, consider the symmetric four-pole bearing pictured in Figure 8.3. Each pole has an area of a, a nominal gap of g_o, and a coil of n turns. For this actuator,

$$M_x = c \begin{bmatrix} 0.25 & -0.125 & 0 & -0.125 \\ -0.125 & 0 & 0.125 & 0 \\ 0 & 0.125 & -0.25 & 0.125 \\ -0.125 & 0 & 0.125 & 0 \end{bmatrix}$$

$$M_y = c \begin{bmatrix} 0 & -0.125 & 0 & 0.125 \\ -0.125 & 0.25 & -0.125 & 0 \\ 0 & -0.125 & 0 & 0.125 \\ 0.125 & 0 & 0.125 & -0.25 \end{bmatrix}$$

where c is a constant containing the bearing geometry:

$$c = \frac{an^2\mu_o}{g_o^2}$$

One could select a P of the form:

$$P = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix}$$

which would yield the corresponding anti-symmetric matrices via (6.32):

$$A_x = c\begin{bmatrix} 0. & 0.125 & -0.25 & 0.125 \\ -0.125 & 0. & 0.125 & 0. \\ 0.25 & -0.125 & 0. & -0.125 \\ -0.125 & 0. & 0.125 & 0. \end{bmatrix}$$

$$A_y = c\begin{bmatrix} 0. & -0.125 & 0. & 0.125 \\ 0.125 & 0. & 0.125 & -0.25 \\ 0. & -0.125 & 0. & 0.125 \\ -0.125 & 0.25 & -0.125 & 0. \end{bmatrix}$$

The matrix $G \equiv [M_x + A_x | M_y + A_y]$ is then

$$G = c\begin{bmatrix} 0.25 & 0. & -0.25 & 0. & 0. & -0.25 & 0. & 0.25 \\ -0.25 & 0. & 0.25 & 0. & 0. & 0.25 & 0. & -0.25 \\ 0.25 & 0. & -0.25 & 0. & 0. & -0.25 & 0. & 0.25 \\ -0.25 & 0. & 0.25 & 0. & 0. & 0.25 & 0. & -0.25 \end{bmatrix}$$

A singular value decomposition of G yields

$$B = [0.5 \quad -0.5 \quad 0.5 \quad -0.5]$$
$$D_x = c\,[0.5 \quad 0. \quad -0.5 \quad 0.]$$
$$D_y = c\,[0. \quad -0.5 \quad 0. \quad 0.5]$$

The linear equation analogous to (6.29) is then

$$\begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5qc & 0. & -0.5qc & 0. \\ 0. & -0.5qc & 0. & 0.5qc \end{bmatrix} i = \begin{Bmatrix} q \\ f_x \\ f_y \end{Bmatrix} \tag{8.4}$$

where q can be chosen as any number that makes the left-hand side of full rank. Eq. (8.4) specifies three equations for four unknowns, leaving open the possibility of one coil failure. If, for example, the coil on pole 1 were to fail, the additional constraint

$$[1 \quad 0 \quad 0 \quad 0]\,i = 0$$

could also be included in (8.4). The solution is found by inverting the left-hand side of (8.4) augmented by the additional constraint:

$$i = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -1 & \frac{-1}{cq} & \frac{-1}{cq} & 1 \\ 0 & \frac{-2}{cq} & 0 & 1 \\ -1 & \frac{-1}{cq} & \frac{1}{cq} & 1 \end{bmatrix} \begin{Bmatrix} q \\ f_x \\ f_y \\ 0 \end{Bmatrix}$$

The W matrix corresponding to this example is

$$W = \begin{bmatrix} 0 & 0 & 0 \\ -q & \frac{-1}{cq} & \frac{-1}{cq} \\ 0 & \frac{-2}{cq} & 0 \\ -q & \frac{-1}{cq} & \frac{1}{cq} \end{bmatrix}$$

8.2 Numerical determination of linearizing currents in fault configurations

The task of finding linearizing currents for a fault configuration can also be approached on a numerical basis. Using the matrix K introduced in (3.18), the reduced order current vector can be mapped onto the full current vector. In this case, the generalized bias linearization problem is

$$\begin{aligned} \mathsf{K}'M_x\mathsf{K} &= \hat{M}_x \\ \mathsf{K}'M_y\mathsf{K} &= \hat{M}_y \end{aligned} \tag{8.5}$$

for radial magnetic bearings. This problem is then solved using the numerical method presented in § 6.3.

8.3 Asymmetric bearings

In the case of asymmetric bearings, there may still be linearizing solutions, even though the isotropic spaces that lead to these solutions are not at all obvious. An example is the asymmetric bearing considered in § 3.1. The bearing will be linearized if there can be found a 5×3 matrix W such that (6.11) is satisfied:

$$W'M_xW = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \hat{M}_x \tag{8.6}$$

$$W'M_yW = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix} = \hat{M}_y \tag{8.7}$$

A suitable linearizing set of currents is then obtainable through the numerical search described in § 6.3. For example,

$$W = \begin{bmatrix} 0.136293 & -0.314259 & 0.610881 \\ 0.205067 & -0.0433335 & 0.233348 \\ 0.461968 & -0.409403 & 0.67415 \\ 0.697068 & 0.261145 & -0.475498 \\ -0.0278216 & 0.227161 & -0.158715 \end{bmatrix} \tag{8.8}$$

is one linearizing solution satisfying (8.6) and (8.7). The first column of (8.8) is the biasing current vector. The second and third columns represent the $X-$ and $Y-$ direction control vectors respectively. The physical coil currents are then specified by (6.12).

$$\begin{aligned} i_1 &= 0.6041\hat{i}_o & -0.1210\,f_x/\hat{i}_o & \quad +0.2996\,f_y/\hat{i}_o \\ i_2 &= 0.1497\hat{i}_o & +0.2909\,f_x/\hat{i}_o & \quad -0.5332\,f_y/\hat{i}_o \\ i_3 &= -0.0329\hat{i}_o & +0.0517\,f_x/\hat{i}_o & \quad -0.2347\,f_y/\hat{i}_o \\ i_4 &= -0.2572\hat{i}_o & +0.4607\,f_x/\hat{i}_o & \quad -0.7501\,f_y/\hat{i}_o \\ i_5 &= -0.5526\hat{i}_o & -0.1636\,f_x/\hat{i}_o & \quad +0.3445\,f_y/\hat{i}_o \\ i_6 &= 0.0266\hat{i}_o & -0.2170\,f_x/\hat{i}_o & \quad +0.1516\,f_y/\hat{i}_o \end{aligned} \tag{8.9}$$

8.4 Criterion for Optimal W

In general, the problem defined by (6.11) has many solutions. Therefore, a criterion must be established for selecting the best solution. While many possible quality measures can be devised, possibly the most useful is the maximum load which the bearing can generate before magnetic saturation occurs at some point in the actuator. Again, radial magnetic bearings will be considered in particular. It is, however, straightforward to formulate the same type of cost function for more elaborate Maxwell force actuators.

To determine when saturation occurs in the stator, the flux densities in the legs, back-iron, and journal iron (denoted b_p, b_b, and b_j respectively) must all be computed. If the pole areas are equal to the air gap areas, then the pole flux densities are simply equal to the gap densities:

$$b_p = b \tag{8.10}$$

Most of the back iron flux densities can be found from the $n-1$ independent conservation of flux conditions:

$$a_{b,j}b_{b,j} - a_{p,j}b_{p,j} - a_{b,j+1}b_{b,j+1} = 0 \tag{8.11}$$

The one remaining equation required is most properly obtained by applying Ampere's loop law to the back iron:

$$\sum_{j=1}^{n} b_{b,j}l_j = 0 \tag{8.12}$$

where l_j is the length of the j^{th} section. However, as the circuit begins to saturate, the permeabilities of the back iron sections with higher flux density will begin to decrease. This will produce a redistribution of flux density which tends to minimize the peak flux density in the back iron, subject to conservation of flux. (Of course, as the iron starts to saturate, flux leakage will also increase, reducing the validity of the simple conservation of flux conditions used here.) On the basis of this heuristic argument, it may be best to solve these equations in such a manner as to minimize the peak flux density. The simplest approximation to this kind of solution is provided by the Moore-Penrose pseudoinverse. Summarize (8.11) as

$$V_b b_b = V_p b_p \tag{8.13}$$

Using the Moore-Penrose pseudoinverse results in

$$b_b = V_b^{\dagger} V_p b_p \,, \; V_b^{\dagger} \doteq V_b^T (V_b V_b^T)^{-1} \tag{8.14}$$

The journal flux densities can be computed in a similar manner, leading to

$$b_s = \left\{ \begin{array}{c} b_p \\ b_b \\ b_j \end{array} \right\} = \left[\begin{array}{c} I \\ V_b^{\dagger} V_p \\ V_j^{\dagger} V_p \end{array} \right] b = \left[\begin{array}{c} I \\ V_b^{\dagger} V_p \\ V_j^{\dagger} V_p \end{array} \right] V i \tag{8.15}$$

The transformation from the reduced order current vector to the distribution of flux densities throughout the stator can then be defined as:

$$V_s \doteq \left[\begin{array}{c} I \\ V_b^{\dagger} V_p \\ V_j^{\dagger} V_p \end{array} \right] V \tag{8.16}$$

Now consider the particular case of a 2 degree of freedom radial bearing. Rather than computing the saturation load directly, compute the flux density distribution for a force of magnitude 1.0 and arbitrary orientation Θ:

$$f_x = \cos\Theta \qquad f_y = \sin\Theta \tag{8.17}$$

If the parameters $\hat{i}_o, \hat{i}_x$, and $\hat{i}_y$ are chosen according to

$$\hat{i}_o = \zeta, \qquad \hat{i}_x = \frac{\cos\Theta}{\zeta}, \qquad \hat{i}_y = \frac{\sin\Theta}{\zeta} \tag{8.18}$$

then the desired force of magnitude 1.0 and direction Θ will result. The flux distribution throughout the stator resulting from any selection of ζ and Θ is given by

$$b_s[\zeta,\Theta,W] = V_s i = V_s W \left\{ \begin{array}{c} \zeta \\ \cos\Theta/\zeta \\ \sin\Theta/\zeta \end{array} \right\} \tag{8.19}$$

The maximum magnitude of the resulting flux density distribution is

$$b_{max}[\zeta,\Theta,W] = |b_s[\zeta,\Theta,W]|_\infty \tag{8.20}$$

The achievable load capacity is then

$$f_{max}[\zeta,\Theta,W] = \left(\frac{b_{sat}}{b_{max}[\zeta,\Theta,W]}\right)^2 \tag{8.21}$$

where b_{sat} is the saturation flux density of the magnet iron.

The achievable load capacity is dependent upon the choice of ζ and Θ. Typically, it is conservative to base the load capacity upon the worst case orientation:

$$b_{max}[\zeta,W] = \max_{\Theta} |b_s[\zeta,\Theta,W]|_\infty \tag{8.22}$$

This choice might be modified for systems where a gravity load or some other load with fixed orientation is significant. Further, the choice of ζ is essentially free: it is the square root of the ratio between biasing field and control field and has no effect on the magnitude or orientation of the field generated. This parameter should be chosen in such a manner as to minimize the peak flux density (and thereby maximize the load capacity):

$$b_{max}[W] = \min_{\zeta}\max_{\Theta} |b_s[\zeta,\Theta,W]|_\infty \tag{8.23}$$

In this manner, the best solution W^* is that which minimizes b_{max} (or maximizes f_{max}):

$$b_{max} = \min_{W\to W^*}\min_{\zeta}\max_{\Theta} |b_s[\zeta,\Theta,W]|_\infty \tag{8.24}$$

The minimax problem defined by (8.24) along with the constraint equation (6.11) forms a nonlinear optimization problem for selecting W. However, it is unlikely that a single gradient descent optimization will yield a global optimum because the solutions for W are not necessarily connected. For symmetric radial bearings with an even number of poles, manifolds of solutions can be obtained analytically. The task is then to find the particular manifold and arbitrary vector q that gives the best performance relative to (8.24). A globally optimal solution is still not guaranteed, because it is possible that the best q found on the best manifold found is only locally optimal, or that the global optimum might lie on an unconsidered manifold.

8.5 Optimization of W on the basis of maximized load capacity

The criterion for an optimal W presented in § 8.4 can be combined with the method presented in § 6.3 for obtaining feasible solutions to yield a locally optimal solution. The simplest way to proceed is to repetitively apply the procedure in § 6.3 starting from different randomly generated seeds to obtain many feasible solutions for the bias linearization problem in question. Then, on the basis of (8.24), the W is picked that delivers the best bearing load capacity.

An alternative procedure that tends to yield better solutions is to use the modified Newton-Raphson method to first yield a feasible solution, denoted w_o. It can then be supposed that there is a manifold of solutions connected to the feasible solution. The idea is to then move along this manifold of solutions in a direction that improves the quality of the solution. Movement should proceed along this manifold until further moves do not improve the quality of W. This procedure is known as the *reduced gradient method*, and was first developed in [AC69].

Recall that the conditions that W must satisfy can be written as

$$F[w] = 0 \text{ where } w \equiv \begin{Bmatrix} W_b \\ W_{c1} \\ W_{c2} \end{Bmatrix} \tag{6.21}$$

The first-order Taylor expansion of F about a W that satisfies (6.21) is then

$$F[w] \approx 2\mathsf{H}\,[w_o]\delta w + \ldots \tag{8.25}$$

If a small change away from w_o, denoted by δw, is made that is orthogonal to each row in $\mathsf{H}\,[w]$, the change in $F[w]$ will be very close to zero.

Using Gramm-Schmidt orthogonalization, one can create a matrix $\mathsf{H}_o[W]$ whose rows are orthonormal and orthogonal to the rows of $\mathsf{H}\,[w]$. In a region local to feasible solution w_o, the manifold of solutions can be approximated by

$$w = w_o + \mathsf{H}_o\nu \tag{8.26}$$

where ν is a set of coordinates that parameterizes the local solution manifold. The task is to find a direction in the space parameterized by ν that gives the best improvement in b_{max}.

Since the $b_{max}[w]$ cannot be differentiated analytically, and since the gradient of $b_{max}[w]$ with respect to w may be discontinuous, the best way to find a direction that improves $b_{max}[w]$ is approximating the gradient, $\partial b_{max}/\partial\nu$ by two-point numerical derivatives. This approximation may not yield the *best* search direction, but it should yield a direction that improves $b_{max}[w]$. Once an approximate $\partial b_{max}/\partial\nu$ is computed, a new w, denoted w_1 is created by a step along the gradient direction:

$$w_1 = w_o - c\frac{\partial b_{max}}{\partial\nu} \tag{8.27}$$

where c is a small number controlling the length of the step. Typically, c should be chosen such that the length of $c\frac{\partial b_{max}}{\partial\nu}$ is small compared to the length of w_o ($< 1\%$).

Vector w_1, may not, however, satisfy $F[w_1] = 0$. Therefore, w_1 should be used as the starting point of a new modified Newton-Raphson search. Typically, only one or two Newton-Raphson steps are necessary to bring w back onto the solution manifold. The entire process is repeated until additional steps along the solution manifold bring no improvement (or only trivial improvements) in b_{max}.

The above procedure has been used successfully to find locally optimal solutions for each failure configuration for an 8-pole radial magnetic bearing for up to 3 coils failed. These solutions are detailed in Appendix A. A program that implements the reduced gradient search is included as Appendix A.2.

8.6 Modification of W with change in position

Up to this point, the position of the rotor has been assumed constant. If the rotor is allowed to change position, the force-to-current relationships vary as well, as described in § 3.3. Fortunately, if a set of linearizing currents is known for the centered position, it is relatively easy to compute position-dependent corrections to these currents via a continuation strategy. Consider the radial magnetic bearing for which a W matrix has been chosen that satisfies (8.6) and (8.7):

$$W'M_x[x,y]W - \hat{M}_x = 0$$
$$W'M_y[x,y]W - \hat{M}_y = 0$$

Assume that a matrix $W[0,0]$ has been chosen so that these equations are satisfied at $x = y = 0$. The change in these equations with respect to position should then be equal to zero if $W[x,y]$ is correctly chosen:

$$\frac{\partial W'}{\partial x}M_xW + W'M_x\frac{\partial W}{\partial x} + W'\frac{\partial M_x}{\partial x}W = 0 \tag{8.28}$$

$$\frac{\partial W'}{\partial x}M_yW + W'M_y\frac{\partial W}{\partial x} + W'\frac{\partial M_y}{\partial x}W = 0 \tag{8.29}$$

These equations represent 12 linear conditions that $\partial W/\partial x$ must obey so that the bias linearizing conditions are still satisfied. For small displacements, it is sufficient to linearize W about $x = y = 0$. In this case, the current as a function of position as well as force is approximated as

$$\left(W + x\frac{\partial W}{\partial x} + y\frac{\partial W}{\partial y}\right)\begin{Bmatrix} \hat{i}_o \\ f_1/\hat{i}_o \\ f_2/\hat{i}_o \end{Bmatrix} \tag{8.30}$$

To find $\frac{\partial W}{\partial x}$ and $\frac{\partial W}{\partial y}$, (8.28) and (8.29) should be solved using a Moore-Penrose pseudoinverse to yield the smallest possible changes in W that still satisfy the generalized bias linearization conditions.

Chapter 9

Direct Optimization–Magnetic Bearings

Consider a brief re-examination of the motivation behind the direct optimal method for magnetic bearings. For generality, an 8-pole bearing with N turns wound on each leg and adjacent coils connected in a series configuration can be represented non-dimensionally using

$$\underline{b} = \frac{b}{b_{sat}} \tag{9.1}$$

$$\underline{i} = \left(\frac{\mu_o N}{g_o b_{sat}}\right) i \tag{9.2}$$

$$\underline{f} = \left(\frac{\mu_o}{\cos(\pi/8)\, a\, b_{sat}^2}\right) f \tag{9.3}$$

For each force direction

$$\underline{f} = \underline{i}' \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \underline{i} \tag{9.4}$$

With this particular non-dimensionalization, an $\underline{i}$ of 1 induces the saturation flux density in the legs. The highest load is then attained by $\underline{i}_1 = 1$ and $\underline{i}_2 = 0$, yielding a non-dimensional force of $\underline{f} = 1$. To realize this maximum load with a bias linearization scheme, the bearing is biased at $\frac{1}{2}$ the saturation flux density:

$$\underline{i}_1 = \frac{1}{2} + \frac{1}{2}\underline{f} \tag{9.5}$$

$$\underline{i}_2 = \frac{1}{2} - \frac{1}{2}\underline{f}$$

Although this bias flux level is necessary to get the maximum load, it may exceed the minimum bias necessary to avoid slew rate limiting. Either load capacity or efficiency (in the sense of resistive power losses) must be sacrificed.

The aim of the direct optimal method is to simultaneously achieve both aims by realizing $\underline{f} = 0$ with the smallest possible currents that satisfy slew rate constraints and by minimizing the currents required at high forces to achieve acceptable load capacity. For the simple 1-d case, Figure 299 compares the currents required for bias linearization to the solution for a $\underline{i} = 0.25$ bias level from § 7.2. Substantially lower current levels occur around zero force, and the maximum force of 1 is realized with a current very close to the optimal value of 1. The savings are more apparent in Figure 9.2, a plot of power loss for each scheme. Although both schemes perform about equally well at high force levels, the direct optimization result is much better at the low force levels, with only one quarter of the power required of the bias linearization scheme at zero force.

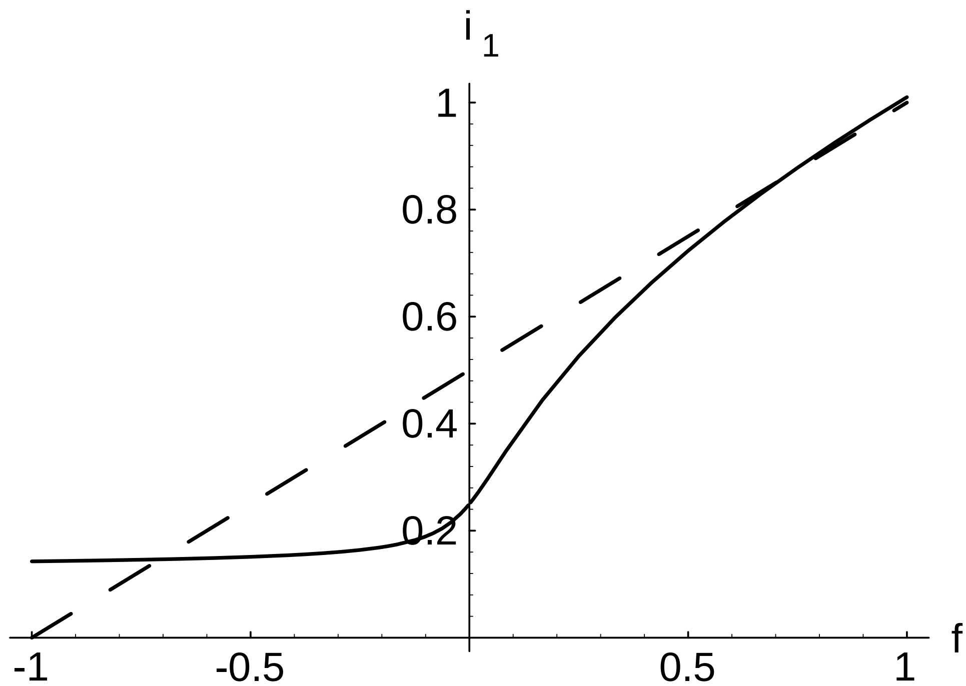


Figure 9.1: Comparison of bias linearization and direct optimal method in a 1-d.o.f. actuator.

9.1 Choice of i_o

The bias vector, i_o, is necessary to avoid slew rate limiting near zero force. In a 1-d.o.f. actuator, choice of a bias vector is relatively obvious. For more general actuators, there are many possible candidates for i_o. Some criterion must be defined by which a particular i_o can be chosen. One way of rating the efficacy of a particular i_o might then be on the basis of the worst slew rate required to move away from the $f = 0$ point. At $f = 0$, the slew rate for the worst direction is easy to compute. At this point, Lagrange multipliers, λ, are equal to zero, making λ disappear from the left-hand side of (7.6). The solution for $\frac{di}{dt}$ is then given through the Moore-Penrose pseudoinverse:

$$\frac{di}{dt} = \mathsf{Q}^{-1}H'(H\mathsf{Q}^{-1}H')^{-1}\frac{df}{dt} \tag{9.6}$$

For the worst possible direction,

$$\left\|\frac{di}{dt}\right\|_2 = \bar{\sigma}[\mathsf{Q}^{-1}H'(H\mathsf{Q}^{-1}H')^{-1}]\left\|\frac{df}{dt}\right\|_2 \tag{9.7}$$

The most desirable i_o would give the smallest possible di/dt per unit length of i_o:

$$\min_{i_o} \bar{\sigma}[\mathsf{Q}^{-1}H'(H\mathsf{Q}^{-1}H')^{-1}] \tag{9.8}$$

$$\text{subject to} \quad i_o' i_o = 1$$

For the case where $\mathsf{Q} = I$, (304) reduces to

$$\min_{i_o} \frac{1}{\underline{\sigma}[H]} \tag{9.9}$$

$$\text{subject to} \quad i_o' i_o = 1$$

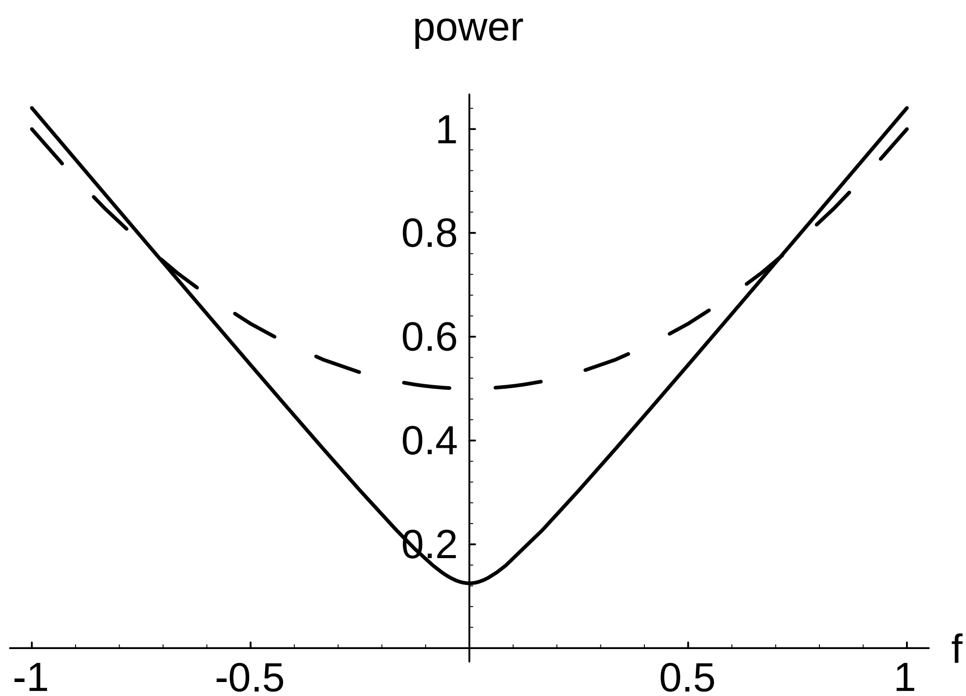


Figure 9.2: Power loss for bias linearization and direct optimization in a 1-d.o.f. actuator.

Once a particular direction for i_o is chosen on the basis of (304), a magnitude must be selected that yields a high enough maximum slew rate at $f = 0$. From Appendix B, a sufficient condition for realizable slew rate is

$$\left\| \frac{di}{dt} \right\|_2 \leq \frac{v_o}{\bar{\sigma}[L]} \qquad \text{(B.4)}$$

The maximum force slew rate, $|df/dt|_{max}$ should be a design specification of the actuator based on the maximum force that the actuator is desired to produce at each frequency. Combining (B.4) and (303) gives, for realizability

$$\frac{v_o}{\bar{\sigma}[L]} \geq \frac{|df/dt|_{max}}{c\,\underline{\sigma}[H]} \qquad (9.10)$$

where c is the magnitude associated with a unit vector i_o, $H \equiv H[i_o]$ and Q is assumed to be the identity matrix. Solving for c,

$$c \geq \frac{|df/dt|_{max}\,\bar{\sigma}[L]}{v_o\,\underline{\sigma}[H]} \qquad (9.11)$$

9.2 Symmetric 8-pole bearing

Of practical interest is the performance of the direct optimal solution on an 8-pole symmetric bearing. The non-dimensionalization for a general 8-pole bearing and the matrices characterizing the arrangement are detailed in Appendix A. For comparison with bias linearization, the direct optimization solution can be contrasted with the all-coils-active bias linearization presented in § A.1.1.

The two obvious candidates for i_o are of the form $\{1,-1,1,-1,1,-1,1,-1\}^T$ and $\{1,1,-1,-1,1,1,-1,-1\}^T$ which correspond to the NSNS and NNSS biasing schemes typically used in 8-pole bearings. Of these two options, the NSNS scheme has been observed to yield consistently lower power

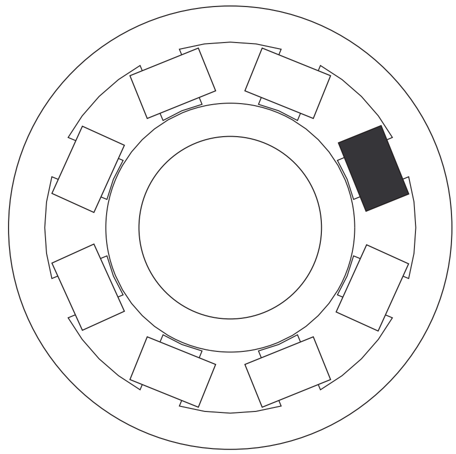


Figure 9.3: Coil 1 in an 8-pole bearing

losses and maximum flux densities in the stator when used as i_o; therefore, the NSNS will be exclusively considered here.

For the 8-pole bearing, the optimization problem to be solved is

$$\min_i J(i) \equiv (i - i_o)' \mathsf{Q}\, (i - i_o) \qquad (7.2)$$

$$\text{subject to} \qquad i' M_1 i = f_1$$
$$\vdots$$
$$i' M_k i = f_k$$
$$\text{where} \qquad i_o = c\{1,-1,1,-1,1,-1,1,-1\}^T$$

The particular scaling $c = 0.25$ is about half of the bias level used by the 8-pole bias linearization in § A.1.1. This optimization leads to the set of ordinary differential equations

$$2\begin{bmatrix} \mathsf{Q} + \sum_{j=1}^{k} \lambda_j M_j & H'[i] \\ H[i] & 0 \end{bmatrix} \begin{Bmatrix} \frac{di}{ds} \\ \frac{d\lambda}{ds} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \frac{df}{ds} \end{Bmatrix} \qquad (7.6)$$

with the initial condition $i = i_o$, $\lambda = 0$. Although more elaborate methods exist, the Euler method was found to be adequately accurate for integrating these equations from $|\underline{f}| = 0$ to $|\underline{f}| = 1$. A short *Mathematica* program used for performing the integration is included in Appendix A.3. Because of symmetry it is sufficient to consider the mapping between force and current for only one coil. For coil "1", illustrated in Figure 9.3 the results of the integration are shown in Figure 9.4. For comparison purposes, the equivalent mapping between force and current for the bias linearization scheme is shown in Figure 9.5. Upon inspection of Figure 9.4, it can be seen that the direct optimal scheme realizes forces by trying to "pull" only with the coil closest to the direction of the force. The result is substantially lower power consumption. A plot of resistive losses for the direct optimal scheme is shown in Figure 9.6, and for the bias linearization scheme in Figure 9.7. The direct optimal scheme starts at a low power level, and the losses increase linearly with force magnitude. The bias linearization scheme starts at a high loss level, and the losses increase quadratically with force magnitude.

Even though resistive losses are greatly decreased by the direct optimization scheme, the cost function does not necessarily perform well with regards to maximum force before saturation. This measure of performance is an infinity norm on flux density in the bearing, as shown previously. Because the direct optimal method causes one pole to create most of the force, saturation can occur at lower force levels than for bias linearization with a carefully chosen bias level. Full back iron and journal iron must be used to avoid

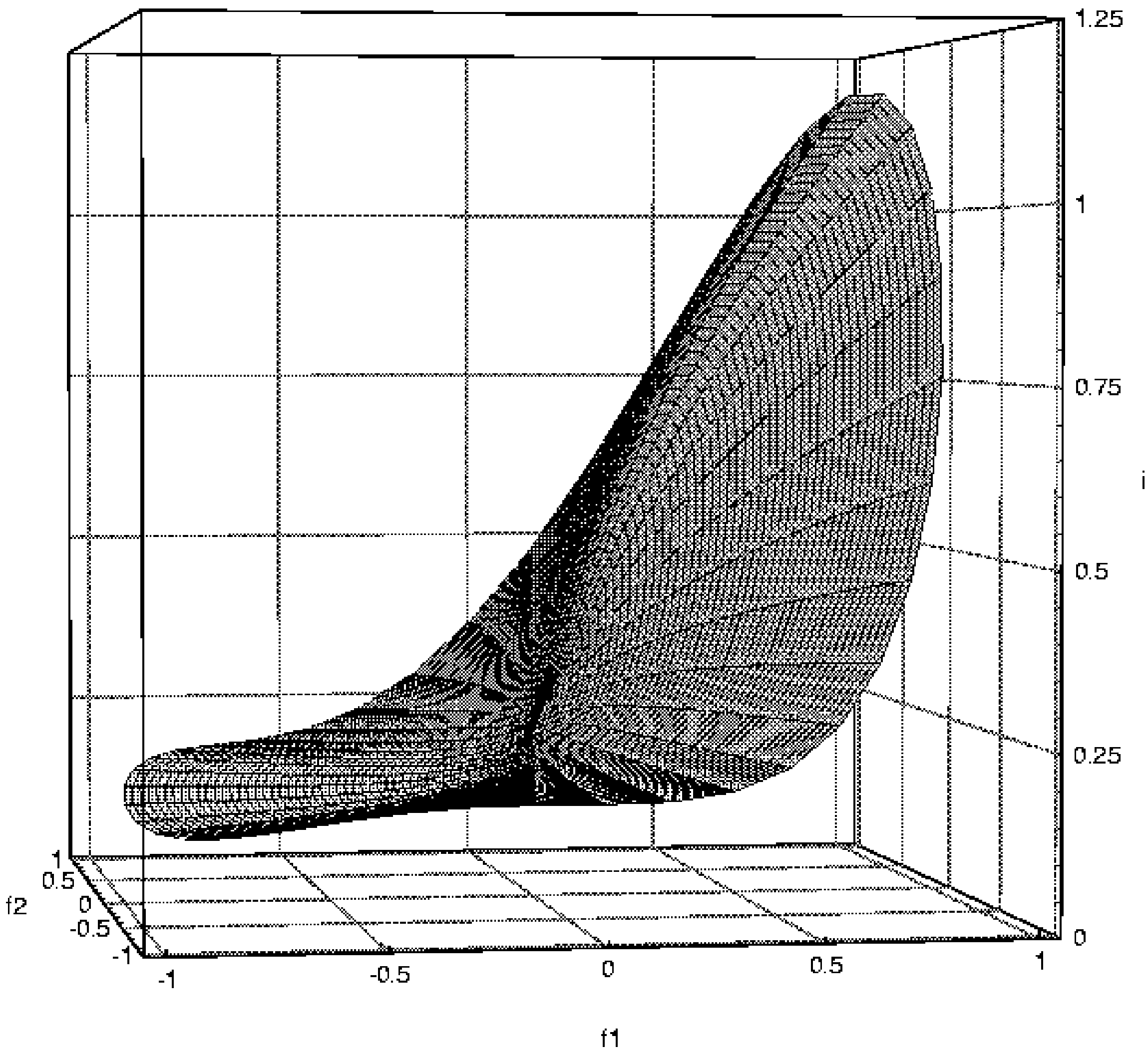


Figure 9.4: Direct optimal force-to-current mapping, c=0.25

pre-mature saturation. For the above example with $c = 0.25$ and assuming full back iron, a saturation flux density is achieved at $\underline{f} = 0.713$, compared with 1.0 for the bias linearization scheme with full back iron. A plot of load capacity before saturation as a function of bias level c is shown in Figure 9.8.

It appears that there is still some trade-off between load capacity and minimized power losses. There are three options:

- Tolerate decreased load capacity resulting from the direct optimal scheme for an 8-pole bearing.
- Enforce an opposed-horseshoe winding for the bearing. The direct optimal scheme can be used on each axis, per Figure 299, requiring slightly higher power losses but achieving the same load capacity as opposed horseshoes.
- Extend the direct optimal formulation to include saturation effects. At high force levels, flux would be redistributed away from the saturated sections, raising the bearing's load capacity.

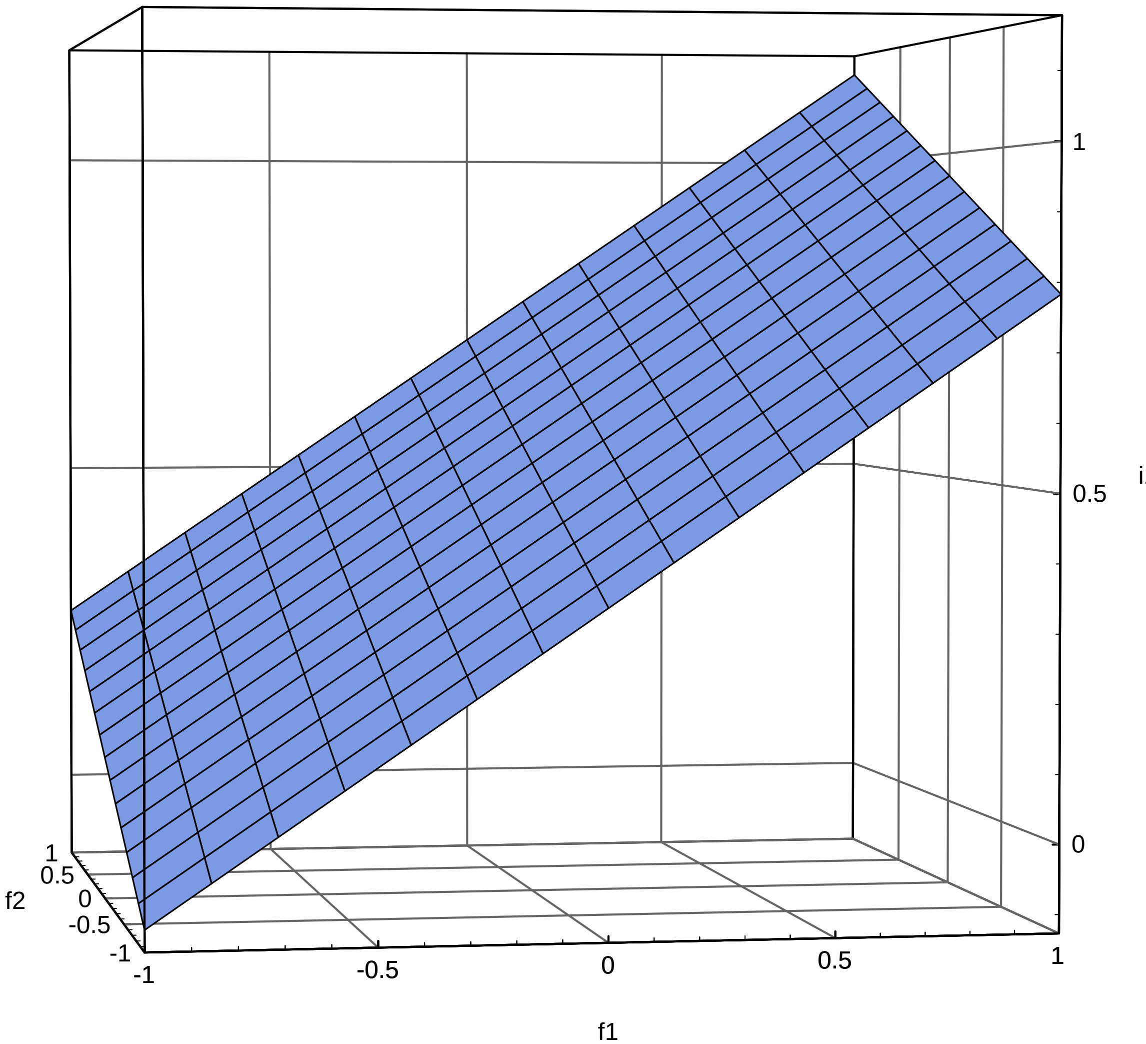


Figure 9.5: Load optimal bias linearization force-to-current mapping

9.3 Direct optimal method including smooth saturation effects

Saturation effects can be included by two different methods. The first and most elegant way to include saturation effects is to write a full set of nonlinear circuit equations for every section of flux path in the actuator. These nonlinear equations are then incorporated as extra constraints in the optimization. The optimization problem becomes:

$$J = \min_i (i - i_o)' \mathsf{Q} \, (i - i_o) \tag{9.12}$$

$$\begin{gathered} b'\Upsilon_1 b + f_1 = 0 \\ b'\Upsilon_2 b + f_2 = 0 \\ C_1 b + C_2 h + C_3 i = 0 \\ b - BH[h] = 0 \end{gathered}$$

where $BH[h]$ is a function representing the virgin magnetization curve of the actuator material, and C_1, C_2, and C_3 are matrices describing the magnetic circuit equations for the actuator. Each of the contraint equations is then combined into cost function J via Lagrange multipliers. The optimality conditions are then

$$\begin{aligned} \frac{\partial J}{\partial i} &= 0 \\ \frac{\partial J}{\partial b} &= 0 \\ \frac{\partial J}{\partial h} &= 0 \\ \frac{\partial J}{\partial \lambda} &= 0 \end{aligned} \tag{9.13}$$

A set of ordinary differential equations results from taking the total derivative of (9.13) with respect to s. The initial condition is found by solving the constraint equations for the b and h that result from $i = i_o$ and taking $\lambda = 0$ at $f = 0$.

9.3.1 2-horseshoe example

As a simple example of including saturation by writing a set of nonlinear circuit equations, consider the 2-horseshoe actuator considered in Figure 9.9. This example is analogous to one axis of a radial magnetic bearing composed of horseshoes. The rotor will be assumed to be infinitely permeable, but each horseshoe has a $B-H$ relationship of

$$b = BH[h] = -0.15\tanh[0.05\,h] + 1.8\tanh[0.005\,h] + \mu_o h \tag{9.14}$$

which is approximately the form of the $B-H$ curve for a typical silicon iron. This equation is plotted in Figure 321 The constraint equations for the optimization are:

$$h_1\,l + b_1\frac{2g_o}{\mu_o} - N\,i_1 = 0 \tag{9.15}$$

$$h_2\,l + b_2\frac{2g_o}{\mu_o} - N\,i_2 = 0 \tag{9.16}$$

$$b_1 - BH[h_1] = 0 \tag{9.17}$$

$$b_2 - BH[h_2] = 0 \tag{9.18}$$

$$\cos\theta\frac{2a}{\mu_o}(b_1^2 - b_2^2) - f = 0 \tag{9.19}$$

where $\theta = \pi/8$, $a = (0.01\,m)^2$, $N = 100$ turns, $g_o = 0.4\,mm$, and the iron length of each horseshoe is $l = 10\,cm$. Each one of these constraints is then included in the cost function with a Lagrange multiplier corresponding to each constraint equation. A bias level approximately one quarter of the way to saturation is

$$i_o = 0.25\left(\frac{2\,g_o\,b_{sat}}{\mu_o N}\right) = 2.62\,\mathrm{A} \tag{9.20}$$

with $b_{sat} = 1.65$ Tesla.

The initial b_1 and h_1 are approximately $b_1 = b_{sat}/4 = 0.41$T and $h_1 = b_{sat}/(4 * 5000\,\mu_o) = 65.6$A/m in response to i_o. The exact initial condition must be found by iteratively solving the constraint equations for $f = 0, i = i_o$ given this guess for an initial condition. About 3 steps of a Newton-Raphson iteration are needed to converge to $b_1 = b_2 = 0.403$, $h_1 = h_2 = 63.36$. Taking the appropriate derivatives and integrating from $f = 0$ to the maximum force $f_{max} = 400$N yields the inverse relationship pictured as the solid line in Figure 328. For comparison, the inverse relationship derived using a constant $\mu_r = 5000$ for the iron is denoted by the dashed line. A rather abrupt deviation from the linear model can be noted near the maximum force due to the saturation effects. However, the smoothness of the inverse mapping is maintained in the saturation region.

Although this method of including saturation yields a smooth inverse mapping, it can be relatively costly to implement in more general bearings. For each section of flux path, a b, an h, and two Lagrange multipliers must be included in the vector of dependent variables to be integrated. For an 8-pole actuator, the result is 106 variables, implying the inversion of a 106×106 matrix at each step. Although this dimensionality is certainly not prohibitive, it may be unnecessarily inconvenient.

9.4 Direct optimal method including hard saturation effects

A more computationally efficient way to include saturation effects is by imposing a set of inequality constraints on the flux in every part of the stator. The advantage of this approach is that the current-to-force relationships need not be broken down into their component equations to include saturation. The problem is still amenable to a non-dimensionalized form, and the computational effort involved is only mildly greater than without saturation. The form of the problem including these equality constraints is, for the non-dimensional 8-pole radial journal bearing:

$$\min_i J(i) \equiv (i-i_o)'\mathsf{Q}\,(i-i_o) \tag{9.21}$$

$$\begin{array}{rc} \text{subject to} & i'M_1 i = f_1 \\ & i'M_2 i = f_2 \\ & V_s i - b_{sat} \leq 0 \\ & -V_s i - b_{sat} \leq 0 \\ \text{where} & i_o = c\{1,-1,1,-1,1,-1,1,-1\}^T \end{array}$$

where V_s is the matrix developed in (8.16) that relates current to flux density in every section of the stator. The inequality constraints simply enforce that absolute value of flux density is everywhere less than or equal to the saturation flux density.

Let C_a denote the collection of rows of V_s and $-V_s$ corresponding to active constraints at a particular step in the integration. Using just the active constraints, (9.21) can be written as a problem with only equality constraints:

$$\min_i J(i) \equiv (i-i_o)'\mathsf{Q}\,(i-i_o) \tag{9.22}$$

$$\begin{array}{rc} \text{subject to} & i'M_1 i = f_1 \\ & i'M_2 i = f_2 \\ & C_a i - b_{sat} = 0 \\ \text{where} & i_o = c\{1,-1,1,-1,1,-1,1,-1\}^T \end{array}$$

Including the additional equality constraints into the cost function with Lagrange multipliers yields the Kuhn-Tucker optimality conditions:

$$\begin{aligned} 2\mathsf{Q}\,(i-i_o)+2\lambda_1 M_1 i+2\lambda_2 M_2 i+C_a'\lambda_3 &= 0 \\ i'M_1 i - f_1 &= 0 \\ i'M_k i - f_k &= 0 \\ C_a i - b_{sat} &= 0 \end{aligned} \tag{9.23}$$

or more succinctly as

$$F_{kt}[i,f,\lambda] = 0 \tag{9.24}$$

where λ_3 is a vector of Lagrange multipliers associated with the cost of the active constraints. Differentiating with respect to the variable s denoting force magnitude yields the system of ordinary differential equations:

$$\begin{bmatrix} 2(I+2\lambda_a M_1+2\lambda_2 M_2) & 2M_1 i & 2M_2 i & C_a' \\ 2i'M_1 & 0 & 0 & 0 \\ 2i'M_2 & 0 & 0 & 0 \\ C_a & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \frac{di}{ds} \\ \frac{d\lambda_1}{ds} \\ \frac{d\lambda_2}{ds} \\ \frac{d\lambda_3}{ds} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \cos\theta \\ \sin\theta \\ 0 \end{Bmatrix} \tag{9.25}$$

These equations can be represented more succinctly as

$$\frac{\partial F_{kt}}{\partial x}\frac{dx}{ds} = -\frac{\partial F_{kt}}{\partial s} \tag{9.26}$$

where $x = \{i,\lambda\}^T$.

In this case, the integration must be done in a more accurate way than the Euler method so that constraints can be picked up and dropped appropriately. The integration of these equations should proceed in the following way, similar to the "elevator predictor-corrector" method described in [PG92]:

1. Start at $f = 0$ with $i = i_o$, $\lambda_1 = \lambda_2 = 0$. None of the inequality constraints should be active at this initial point, so C_a is empty, and λ_3 has a dimension of 0.

2. Take a prediction step via Euler integration:

$$x[s+ds] = x[s] + ds\left(\frac{\partial F_{kt}[x]}{\partial x}\right)^{-1}\frac{\partial F_{kt}}{\partial s} \tag{9.27}$$

3. Correct with a Newton-Raphson step to make sure that the Kuhn-Tucker conditions are satisfied exactly:

$$x^*[s+ds] = x[s+ds] - (\frac{\partial F_{kt}[x]}{\partial x})^{-1}F_{kt}[x+ds] \tag{9.28}$$

   where $x^*[s+ds]$ represents the corrected value of x at $(s+ds)$. Usually only one correction step is sufficient, given an adequately small step size ds.

4. Check for constraint violation. If constraints have been violated, add the violated constraints to C_a, and add new elements to λ_3 corresponding to the newly imposed constraints. The value of the new elements of λ_3 is estimated to be 0, since the inequality constraints are close to being met. A series of Newton-Raphson steps should be taken via (9.28) to accurately determine the values of the new entries of λ_3 with the newly imposed constraints in place. Again, usually one Newton step is sufficient to correct.

5. Check for newly inactivated constraints. If an element in λ_3 becomes negative, the constraint does not impede progress and should be taken out of the list of active constraints. A Newton correction step should be taken when a constraint is dropped to make sure that the Kuhn-Tucker conditions are precisely satisfied.

6. Repeat, taking another Euler prediction step.

This procedure has been applied to the non-dimensionalized 8-pole bearing described in Appendix A. The source code used to implement the procedure is included as § A.4. Inequality constraints were applied so that the flux density in any pole is limited to $|\underline{b}| \leq 1$. A bias level of $i_o = 0.25 * \{1,-1,1,-1,1,-1,1,-1\}^T$ is employed, identical to the unsaturated case considered earlier. The resulting inverse mapping is pictured in Figure 9.12. Again, only the mapping for the coil on pole "1" is pictured, since the mappings for other poles are merely rotations of this one. Comparing to Figure 9.4, the inverse with saturation loses some of its smoothness as the saturation constraints are imposed. However, the mapping is still continuous with finite slope, and therefore realizable. The full load realizable by the bias linearization scheme ($\underline{f} = 1.0$) has been attained in every direction. Power loss, pictured in Figure 9.13, is only slightly worse than in the unconstrained case.

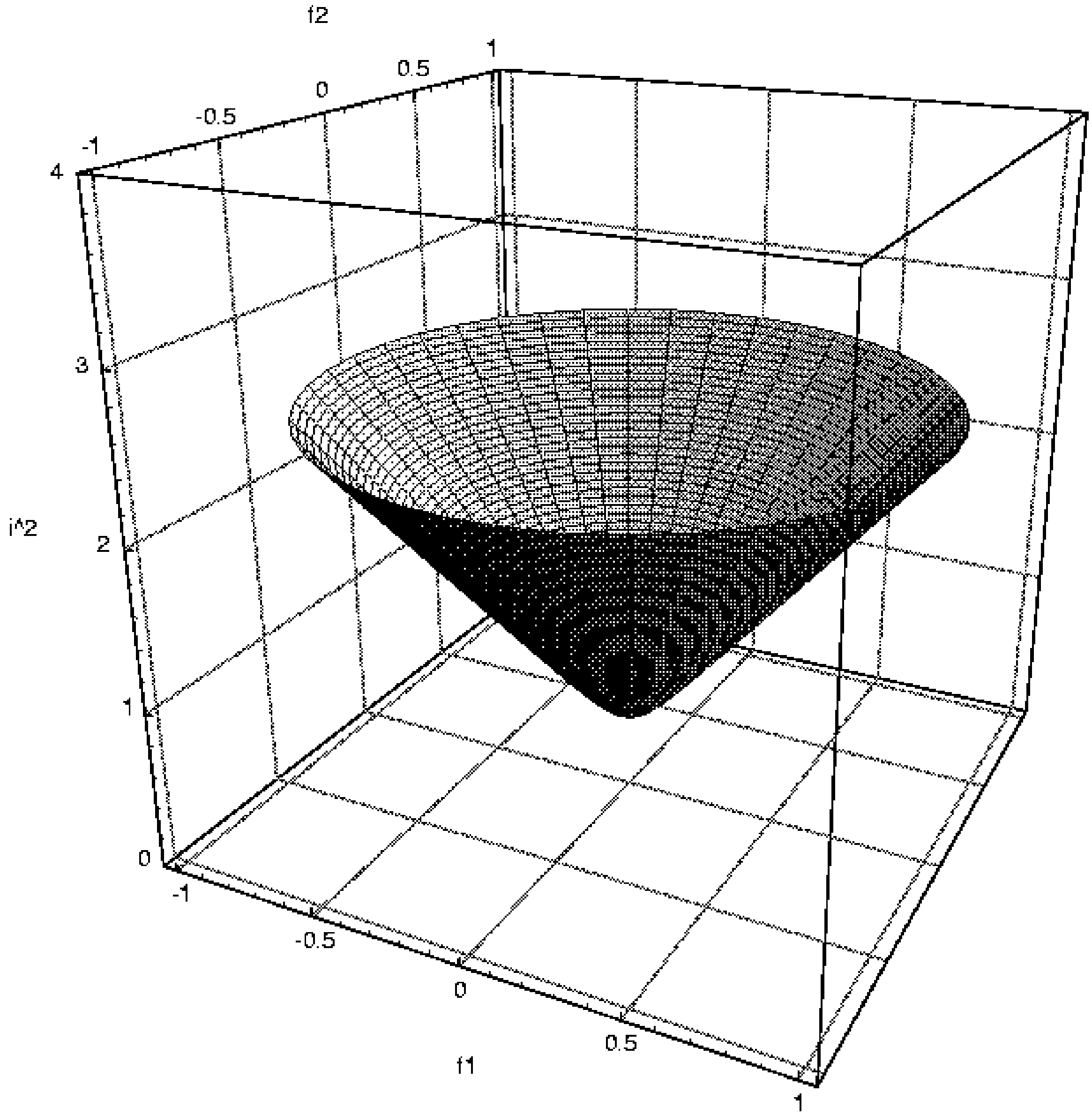


Figure 9.6: Direct optimal resistive losses

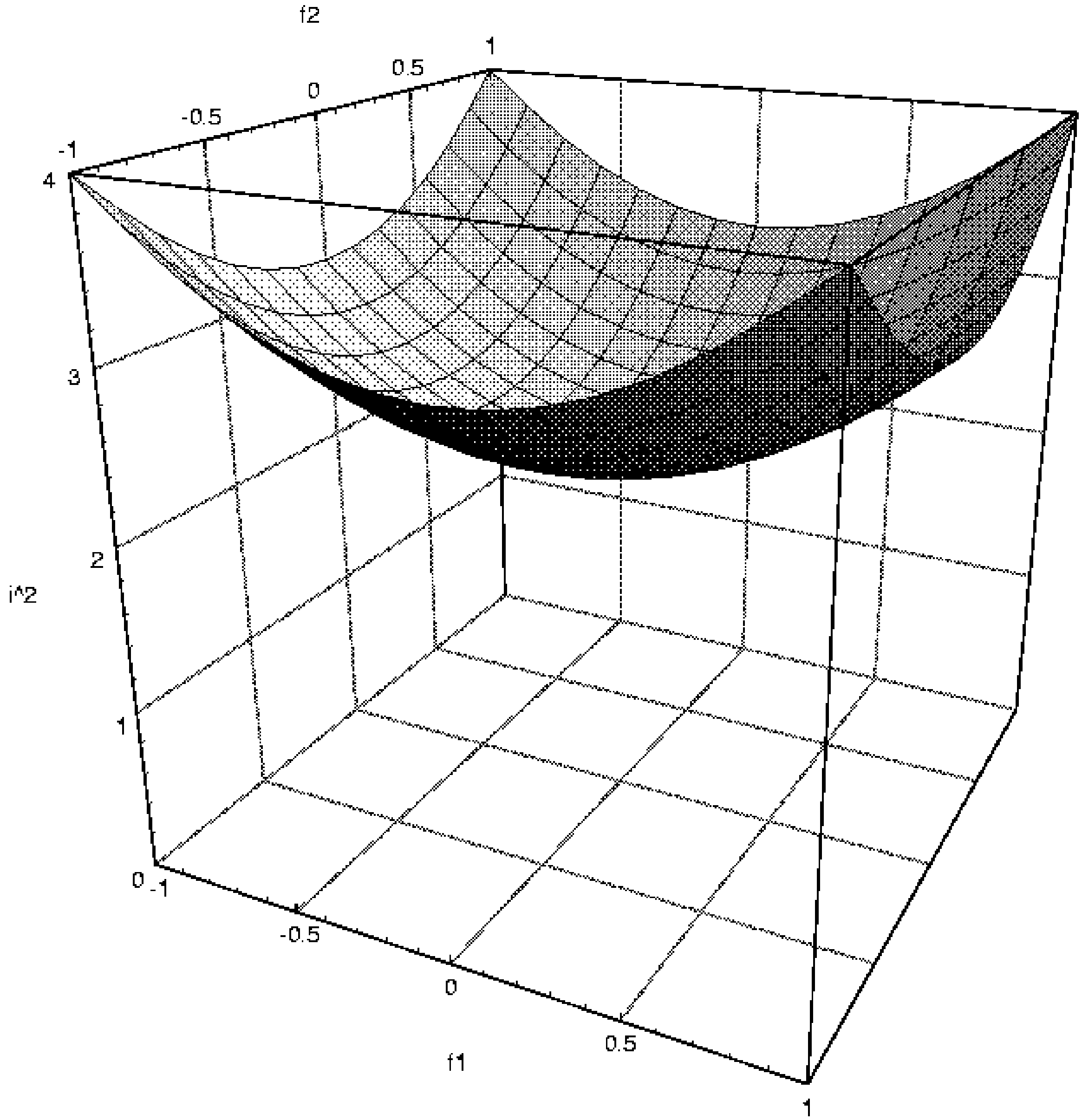


Figure 9.7: Load optimal bias linearization resistive losses

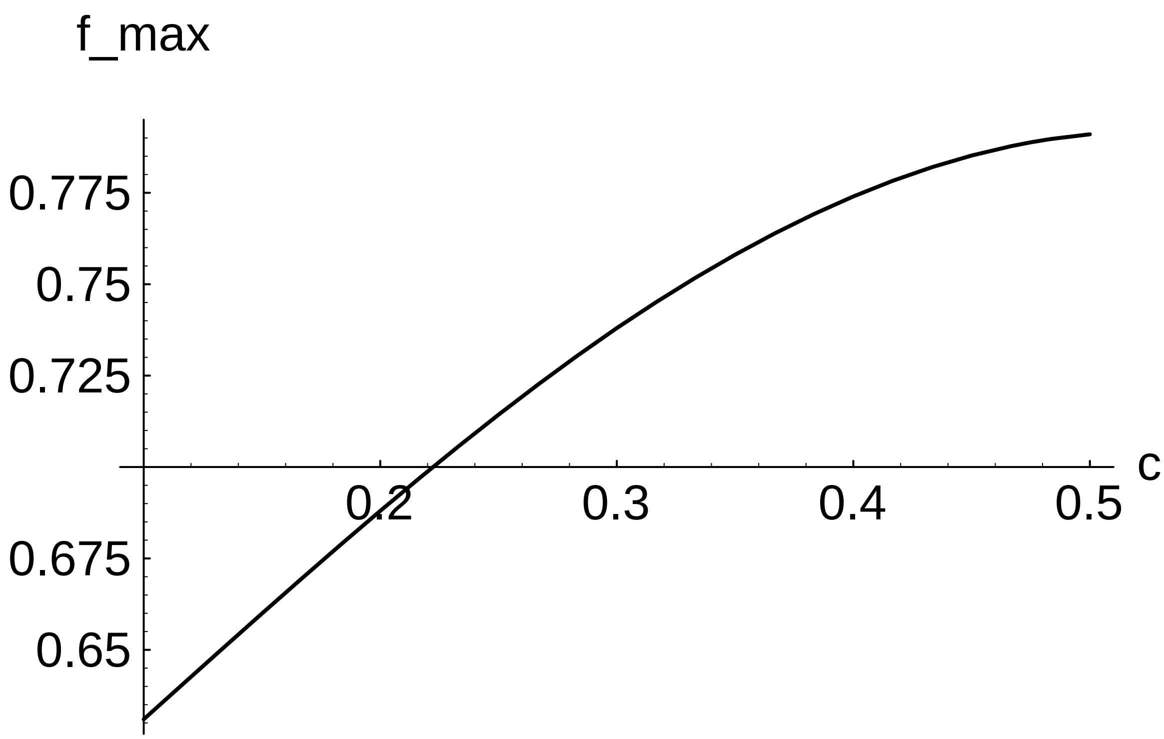


Figure 9.8: Load capacity versus bias level for direct optimal method.

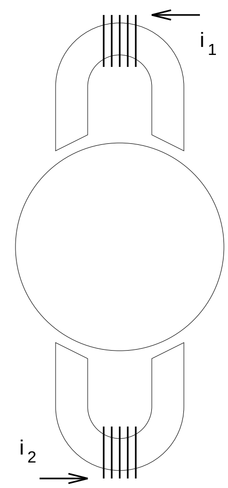


Figure 9.9: 2-horseshoe saturating actuator

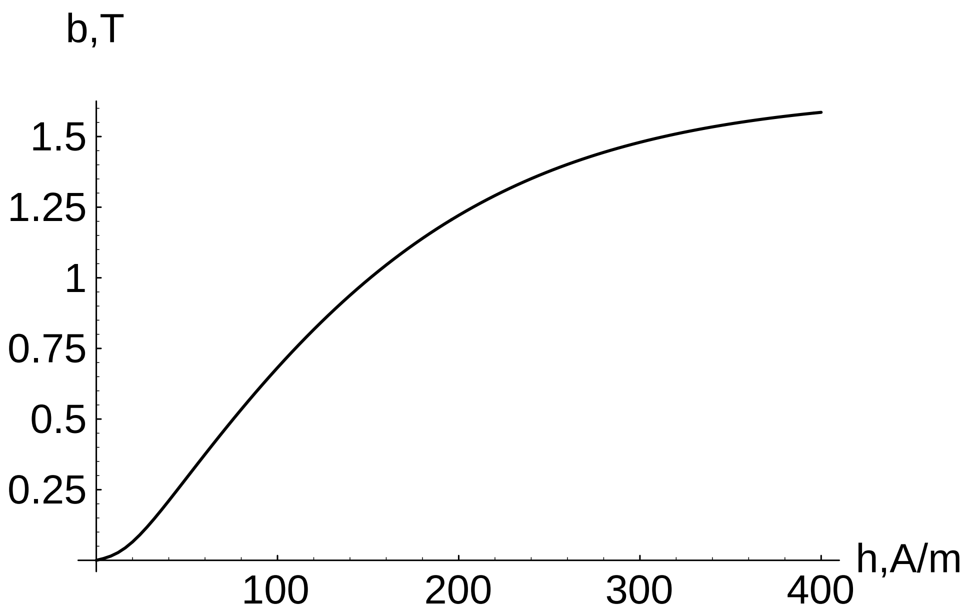


Figure 9.10: Approximate B-H curve for silicon iron

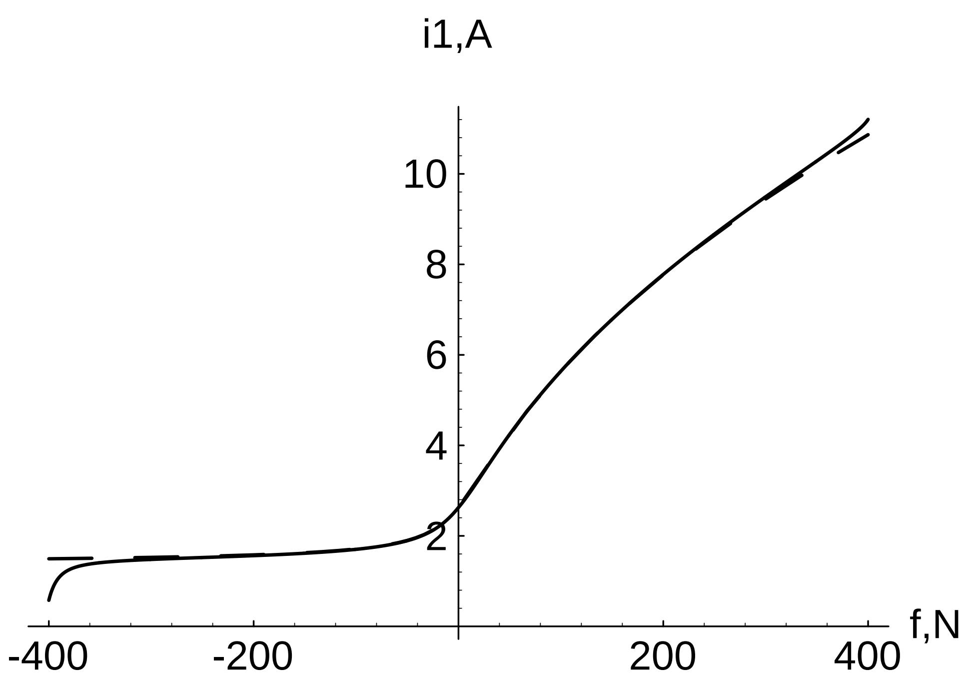


Figure 9.11: 2-horseshoe force-to-current including saturation

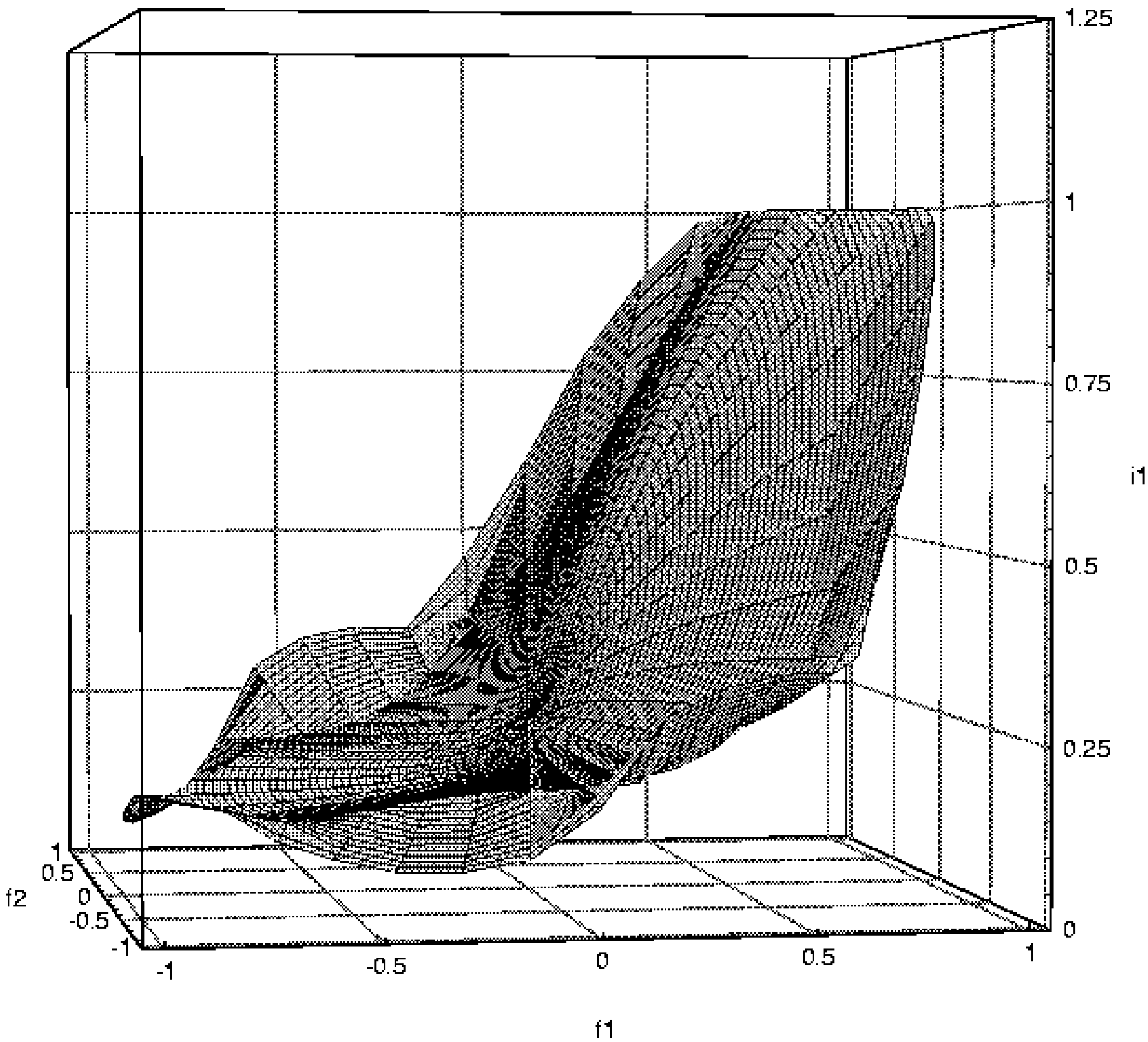


Figure 9.12: 8 pole inverse mapping with saturation

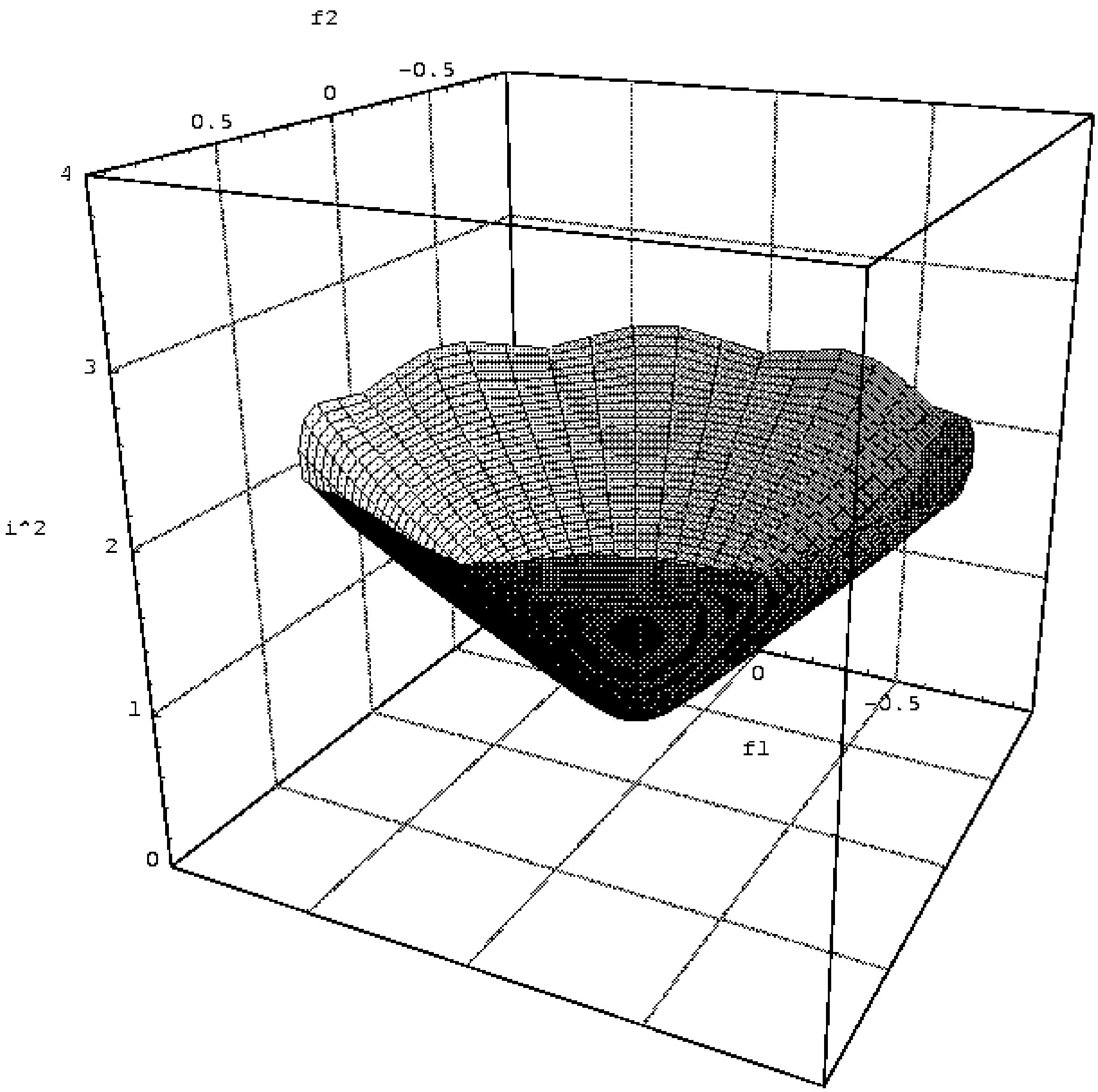


Figure 9.13: 8 pole bearing with saturation–power loss.

Chapter 10

Coil Current Solution–Magnetic Stereotaxis System

In Chapter 4, the relationship between current and force was derived for the magnetic stereotaxis machine. Besides merely solving this equation, currents must also be chosen so that the magnetic seed is aligned in a desirable direction. Generally, the seed is prolate, and the magnetization is aligned along the major axis. If a catheter is pulled, it is attached on one of the ends of the major axis. To minimize tissue damage, the field and force should be directed along the same line so that the seed presents the smallest possible profile and minimizes damage to tissue. Misalignment between the force and field vectors is called "skidding." Although no skidding at all is desirable, the currents required to realize this constraint are often unacceptably large. However, some skidding might be tolerable in exchange for lower current requirements. Three scenarios are therefore considered which impose different constraints on the amount of skidding allowed.

Three different scenarios will be considered; each approach considers different constraints on the seed orientation.

1. *Unlimited Skid Seed.* Since the seed attitude is not constrained, the desired force can be created with the dipole oriented in any position. This configuration represents the most economical force for current case, since the best dipole orientation can be used.

2. *No Skid Seed.* The seed is prolate with the magnetization aligned along the major axis for this case. To cause the least damage to tissue as the seed moves, the dipole must be anti-parallel to the desired force direction.

3. *Limited Skid Seed.* The seed is again prolate. However, misalignment (skidding) bounded by a specified maximum angle is tolerated in exchange for a solution requiring lower currents.

10.1 Unlimited Skid Seed

The problem of producing an arbitrary force on the seed is now that of finding an i that satisfies (4.12):

$$f_j = \frac{\gamma i' M_j i}{\sqrt{i' B' B i}} \tag{4.12}$$

for an arbitrary f; seed orientation is of no particular concern. Generally, there exist many possible i that satisfy (4.12), since (4.12) represents three equations for six unknowns. This implies up to a three-dimensional manifold of solutions for i. More equations must be specified to rate the desirability of the valid solutions so that the most efficient solution is chosen.

One possible measure of solution quality is cost function C:

$$C = \frac{i'i}{\sqrt{i'B'Bi}} \tag{10.1}$$

Minimization of C over the set of i that satisfies (4.12) is a trade-off between minimizing the currents needed to produce a force and strongly aligning the seed. In the transformed coordinates,

$$i = \frac{\imath}{\gamma}\sqrt{\imath'B'B\imath} \tag{4.13}$$

The cost function to be minimized is:

$$J = \imath'\imath \tag{10.2}$$

$$\text{subject to } \imath M_j \imath = f_j \; ; \; j = 1,2,3$$

Eq. (10.2) is the same form minimized by the direct optimal method in Chapter 7. Since there is a substantial threshold force that must be overcome before motion occurs, force levels need never be lowered entirely to zero. Slew rate limiting around zero force is a non-issue, and $i_o = 0$ can be used. If $i_o = 0$, it is easy to see that the optimal current for realizing a force at one magnitude scales linearly to produce optimal forces of the same direction but different magnitude. Determining a path from $f = 0$ is then determined solely by computing the optimal currents necessary to produce a unit force in a given direction. The cost function that must be minimized is then

$$\hat{J} = \imath'\imath + \sum_{j=1}^{3} \lambda_j (\imath' M_j \imath - f_j) \tag{10.3}$$

Conditions for an optimum are then

$$\begin{array}{lcl} \imath + (\lambda_1 M_1 + \lambda_2 M_2 + \lambda_3 M_3)\hat{\imath} & = & 0 \\ \imath' M_1 \imath - f_1 & = & 0 \\ \imath' M_2 \imath - f_2 & = & 0 \\ \imath' M_3 \imath - f_3 & = & 0 \end{array} \tag{10.4}$$

Equations (10.4) are a set of 9 equations for 9 unknowns. This system can be solved for $\hat{\imath}$ and λ using a Newton-Raphson iteration starting from a randomly chosen seed $\imath$. The continuation approach is not mandatory in this case because only a single set of currents is desired, rather than a continuous manifold. However, a solution found via Newton-Raphson iteration is not necessarily a global minimum; it could also be a maximum, a saddle point, or a local minimum. It is therefore necessary to solve several times starting from several starting points to make sure that the actual minimum is found.

10.1.1 No-Skid Seed

Recall the current-to-force relationship:

$$f_j = \imath' B' D_j \imath \tag{4.14}$$

Both the B matrix and all D matrices have at most 3 rows. By virtue of the way that these equations are derived, they naturally fall into the form of (6.28). Eq. (4.14) can be directly transformed into the decomposed form:

$$\begin{bmatrix} B \\ q'D_1 \\ q'D_2 \\ q'D_3 \end{bmatrix} \imath = \begin{Bmatrix} q \\ f_1 \\ f_2 \\ f_3 \end{Bmatrix} \tag{6.29}$$

In terms of physical currents i, the decomposed form is

$$\begin{bmatrix} B \\ \frac{\gamma}{|b|} b' D_1 \\ \frac{\gamma}{|b|} b' D_2 \\ \frac{\gamma}{|b|} b' D_3 \end{bmatrix} i = \left\{ \begin{array}{c} b \\ f_1 \\ f_2 \\ f_3 \end{array} \right\} \tag{10.5}$$

This time, however, the decomposed form has a meaningful physical interpretation. Bi specifies the direction and magnitude of the flux density at the seed location. Since it is assumed that the seed turns to align stably with the field direction, the direction of Bi also specifies the alignment of the dipole.

The "no-skid" constraint additionally requires that the dipole must be stably aligned along the direction of motion. As a matter of convention, $\mathbf{m}$ is chosen to be anti-parallel to $\mathbf{f}$ in the stable configuration, implying that $\mathbf{b}$ should also be anti-parallel to $\mathbf{f}$. This convention has been adopted because it matches the design of existing seeds used for pulling catheters in the MSS. However, the solution for the opposite convention is easily obtainable. The quadratic form of (4.12) guarantees that the same force is produced regardless of the sign of i. However, from (4.5), a change in the sign of i results in a reversal of field direction. The same force is produced, but the stable dipole orientation is rotated 180^o. Therefore, the negative of the solution under the present convention is the solution for the same force result with $\mathbf{m}$ stably oriented parallel to $\mathbf{b}$ rather than anti-parallel.

From (10.5), f goes linearly with current. It is then sufficient to consider only $\mathbf{f}_d$, a unit force in the desired force direction. All other force magnitudes can be achieved by a linear scaling of the currents required for f_d. If the seed is properly aligned,

$$\begin{aligned} \mathbf{m} &= -\gamma \mathbf{f}_d \\ \mathbf{b} &= -\alpha \mathbf{f}_d \end{aligned} \tag{10.6}$$

where α is some presently unknown but positive real number, ensuring that the seed is stably aligned. This orientation is pictured in Figure 10.1.

For ease of notation, define

$$\mathsf{D}[p, m] \equiv \begin{bmatrix} m' D_1 \\ m' D_2 \\ m' D_3 \end{bmatrix} \tag{10.7}$$

We can substitute from (10.7) and (4.5) to form the following system of equations that is linear in i:

$$\begin{bmatrix} \mathsf{D}[\mathbf{p}, -\gamma \mathbf{f}_d] \\ B[\mathbf{p}] \end{bmatrix} i = \left\{ \begin{array}{c} f_d \\ -\alpha f_d \end{array} \right\} \tag{10.8}$$

or more concisely as

$$\mathsf{G}\, i = d \quad \text{where} \quad \mathsf{G}[\mathbf{p}, \mathbf{f}_d] = \begin{bmatrix} \mathsf{D}[\mathbf{p}, -\gamma \mathbf{f}_d] \\ B[\mathbf{p}] \end{bmatrix} \quad \text{and} \quad d[\alpha] = \left\{ \begin{array}{c} f_d \\ -\alpha f_d \end{array} \right\} \tag{10.9}$$

if the seed is properly aligned and the correct force is produced. For a given position specified by $\mathbf{p}$ and a desired force direction $\mathbf{f}$, G is a uniquely determined 6×6 matrix. If G is nonsingular, the set of i that satisfies (10.8) is

$$i = \mathsf{G}^{-1} d[\alpha] \tag{10.10}$$

However, an appropriate value for α has yet to be determined. The only constraint on α is that it must be positive so that the stable alignment of the seed matches the attitude assumed in (10.8) for forming F. (If α is negative, the stable orientation rotates by 180^o, changing the sign of the F matrix. The resulting force is then the opposite of the desired force.) Any arbitrary positive real α is a valid solution, but some solutions are more economical than others to realize. One way to choose α is such that α minimizes $i'i$ over the set of valid i. Equation (10.10) can be directly substituted into cost function $i'i$:

$$i'i = d[\alpha]' (\mathsf{G}^{-1})' \mathsf{G}^{-1} d[\alpha] \tag{10.11}$$

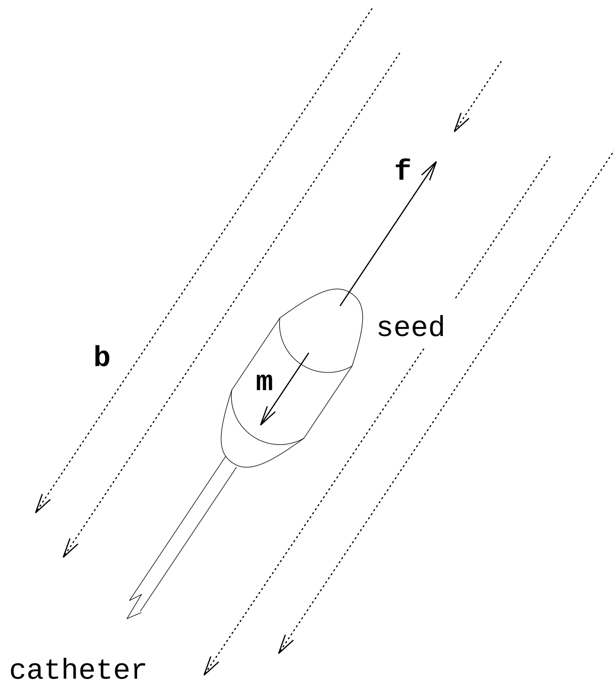


Figure 10.1: MSS seed in stable no-skid orientation.

Equation (10.11) is a quadratic in one variable, α. The extremum of this quadratic must be a minimum since $i'i$ goes to $+\infty$ as α goes to $\pm\infty$.

Perhaps the simplest closed-form representation of the optimal α uses the singular value decomposition [HJ85] of G:

$$\mathsf{G} = \mathsf{U}'\Lambda\mathsf{W} \tag{10.12}$$

U and W are rank$[\mathsf{G}] \times n$ orthonormal matrices. Λ is a diagonal matrix of dimension rank$[\mathsf{G}]$ whose entries are all greater than zero and are arranged in non-increasing order. The rows of U represent a basis for the output of $\mathsf{G}i$, and the diagonal entry corresponding to a row of U represents the inverse "cost" of realizing a unit output parallel to that column. Substituting (10.12) into (10.11) and minimizing with respect to α yields:

$$\alpha_{opt} = \frac{\left\{ \begin{array}{cc} f_d' & 0 \end{array} \right\} \mathsf{U}'\Lambda^{-2}\mathsf{U} \left\{ \begin{array}{c} 0 \\ f_d \end{array} \right\}}{\left\{ \begin{array}{cc} 0 & f_d' \end{array} \right\} \mathsf{U}'\Lambda^{-2}\mathsf{U} \left\{ \begin{array}{c} 0 \\ f_d \end{array} \right\}} \tag{10.13}$$

There is, however, no guarantee that α_{opt} is positive. Parameter α should then be chosen as

$$\alpha = \max[\alpha_{opt}, \alpha_{min}] \tag{10.14}$$

where α_{min} is an arbitrarily chosen minimum acceptable positive value, thereby ensuring that the dipole is properly oriented and adequately aligned.

In certain pathological cases, G is not invertible. These instances most often occur along lines of symmetry. For some of these cases, however, an answer satisfying (10.8) may still exist. If G is of rank 5 rather than of rank 6 (and therefore singular), matrix U that forms a basis for the output of $\mathsf{G}i$ is not square. The cost to realize an output with any component orthogonal to the rows of U is therefore infinite. For a solution to exist, α must be picked so that $d[\alpha]$ is spanned by U (the vector d must be perpendicular to the unrealizable output vector):

$$0 = u'd[\alpha] \qquad \Rightarrow \qquad \alpha = \frac{u'\left\{\begin{array}{c} f_d \\ 0 \end{array}\right\}}{u'\left\{\begin{array}{c} 0 \\ f_d \end{array}\right\}} \tag{10.15}$$

where u is the vector orthogonal to the rows of U. If (10.15) yields $\alpha > 0$, a solution that satisfies the no-skid conditions (and is perpendicular to the unrealizable space of outputs) exists. It is also interesting to note that α_{opt} (10.13) converges to (10.15) as G becomes singular.

In the singular case, there are multiple ways to produce any realizable output, since there is a six-dimensional space of inputs and a five-dimensional space of outputs. It can be shown that the most efficient way (in an $i'i$ sense) to produce the solution for the singular case is

$$i = \mathsf{W}\,'\Lambda^{-1}\mathsf{U}\,d[\alpha] \tag{10.16}$$

where α satisfies (10.15).

10.1.2 Limited-Skid Seed

Under limited skid conditions, some misalignment between **m** and $-$**f** is considered acceptable. In this case, **b** is chosen so that the dipole is stably oriented at a small misalignment with **-f**. The desired **b** direction can be thought of as the direction of **f** rotated through some small angles about vectors perpendicular to **f**.

Define a new reference frame E such that

$$\left\{\begin{array}{c} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{array}\right\} = T\left\{\begin{array}{c} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{n}_3 \end{array}\right\} \tag{10.17}$$

where

$$T = \left[\begin{array}{ccc} e_{11} & e_{12} & e_{13} \\ f_{d,1} & f_{d,2} & f_{d,3} \\ (e_{12}f_{d,3} - e_{13}f_{d,2}) & (e_{13}f_{d,1} - e_{11}f_{d,3}) & (e_{11}f_{d,2} - e_{12}f_{d,1}) \end{array}\right] \tag{10.18}$$

and $\mathbf{e}_1$ is some arbitrarily chosen unit vector such that

$$e_{11}f_{d,1} + e_{12}f_{d,2} + e_{13}f_{d,3} = 0 \tag{10.19}$$

Unit vectors $\mathbf{e}_1$ and $\mathbf{e}_3$ are perpendicular and $\mathbf{e}_2$ is parallel to the desired force direction. Define

$$\hat{\mathbf{f}}_d \equiv (-\sin\varepsilon_2)\mathbf{e}_1 + (\cos\varepsilon_1\cos\varepsilon_2)\mathbf{e}_2 + (\sin\varepsilon_1\cos\varepsilon_2)\mathbf{e}_3 \tag{10.20}$$

Angles ε_1 and ε_2 represent small rotations about $\mathbf{e}_1$ and $\mathbf{e}_3$ respectively. Vector $\hat{\mathbf{f}}_d$ is $\mathbf{f}_d$ misaligned by ε_1 and ε_2. With respect to the E frame:

$$\hat{f}_d(\varepsilon_1, \varepsilon_3) = T'\left\{\begin{array}{c} -\sin\varepsilon_2 \\ \cos\varepsilon_1\cos\varepsilon_2 \\ \sin\varepsilon_1\cos\varepsilon_2 \end{array}\right\} \tag{10.21}$$

Instead of (10.6), the conditions now required are

$$\begin{array}{rcl} \mathbf{m} & = & -\gamma\hat{\mathbf{f}}_d \\ \mathbf{b} & = & -\alpha\hat{\mathbf{f}}_d \end{array} \tag{10.22}$$

so that the dipole is stably positioned and misaligned with $-\mathbf{f}_d$ by a small amount. These conditions imply the linear system

$$\mathsf{G}\,i = \left\{\begin{array}{c} f_d \\ -\alpha\hat{f}_d \end{array}\right\} \quad \text{where} \quad \mathsf{G}\,[\mathbf{p}, \hat{\mathbf{f}}_d] = \left[\begin{array}{c} \mathsf{D}\,[\mathbf{p}, -\gamma\hat{\mathbf{f}}_d] \\ B[\mathbf{p}] \end{array}\right] \tag{10.23}$$

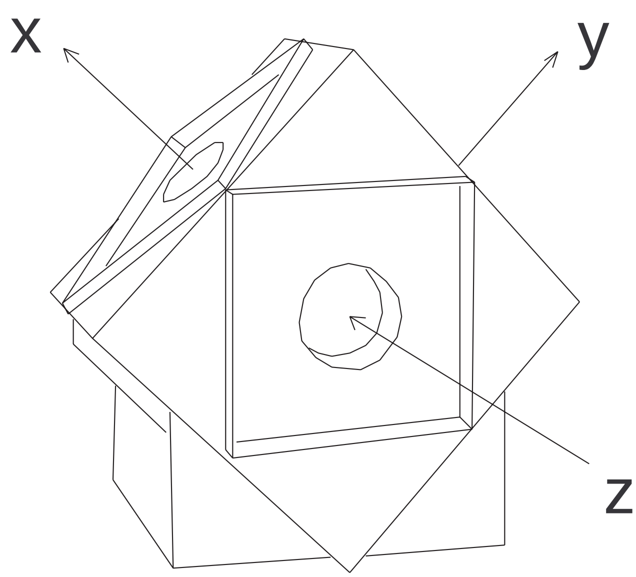


Figure 10.2: Magnetic Stereotaxis System.

The solutions for i and an appropriate α proceed exactly as for the no-skid case, given some arbitrary values of ε_1 and ε_2.

A convenient limit on the allowable skid is

$$\varepsilon_1^2 + \varepsilon_2^2 \leq \varepsilon_{max}^2 \tag{10.24}$$

where ε_{max} is the largest allowable skid angle. A search can then be made over the allowable region of ε's for the most economical solution.

10.2 Examples

As a demonstration of the above solutions, the procedures will be appled to the Magnetic Stereotaxis System (MSS). Since each problem is posed over a five-dimensional domain space of independent variables (consisting of seed location coordinates and two degrees of freedom to orient the force direction), an exhaustive comparison under all operating conditions is impractical. However, merely examining several randomly chosen points in the operating region is sufficient to reveal some of the character of each solution.

Before specific operating points can be tested, the general layout and specifications of the MSS must be described. This machine consists of six large superconducting coils arranged on the faces of a flattened cubical structure, as depicted in Figure 10.2. A right-handed coordinate system is defined as shown in the figure. Each axis of the coordinate system extends through the center of a pair of coils. The origin of the coordinate system is located at the center of the machine. Fluoroscopes for sensing seed position are aligned along the $X-$ and $Y-$axes, and the patient's head enters the machine along the $Z-$axis.

Due to ergonomic constraints, the coils centered on the $X-$ and $Y-$axes are identical, whereas the two coils centered on the $Z-$axis are slightly flattened and closer together. The physical dimensions of these coils are given in Table 10.1. One seed employed in the MSS has a strength of 0.016 A·m^2. This seed is a circular cylinder approximately 3 mm in diameter and 3 mm tall. Rounded plastic end pieces are usually

Coil Dimensions

	X,Y axes	Z axis
Inner Dia.	28.00 cm	32.10 cm
Outer Dia.	37.20 cm	41.10 cm
Thickness	7.01 cm	3.72 cm
Distance between coil faces	45.69 cm	29.38 cm
turns/cm^2	207.2	207.2
Max. current	100 A	100 A

Table 10.1: MSS Coil Dimensions

#	Point			Force direction		
1	{ 1.50,	7.40,	-4.68}	{ -0.936,	0.338,	-0.094}
2	{ 2.88,	2.09,	-2.89}	{ 0.736,	-0.624,	0.259}
3	{ -5.78,	0.28,	-4.34}	{ -0.101,	-0.678,	-0.727}
4	{ 4.15,	-7.29,	6.79}	{ 0.456,	-0.234,	-0.858}
5	{ 5.00,	5.00,	0.00}	{ 0.707,	0.707,	0.00 }
6	{ -3.30,	-4.23,	5.04}	{ 0.506,	-0.046,	0.860 }

Table 10.2: Test points and desired force directions

attached to decrease resistance to seed motion. More detailed descriptions of the device, associated power electronics, and performance are contained in [GRB+94], [MRW95b] and [MRW95a].

The operating region of the MSS lies within a box that extends from -10 cm to 10 cm on the X- and Y-axes and from -14 cm to 6 cm on the Z-axis. Inside this region, a set of four random points and force directions have been chosen. Two more specifically chosen points are included because they represent interesting special cases where the G matrix is singular in the no-skid case. These points are summarized in Table 10.2.

For the no-skid and limited-skid cases, there is no *a priori* basis upon which to assume the minimum acceptable value of α necessary to properly align the dipole. This value should most properly be determined experimentally. In the absence of these experimental results, α_{min} will be assumed zero arbitrarily, since this is the lowest possible value for maintaining a proper dipole alignment once that alignment is achieved.

10.2.1 No-Skid Case

Under the No-Skid conditions, currents are determined by solving the linear equation (10.8). The value of the arbitrary parameter α is chosen to be the greater of the α that minimizes (10.11) or α_{min}. Each of the points in Table 10.2 are considered under the No-Skid conditions, and the solution currents are summarized in Table 10.3. Examples (1.1-1.4) represent the typical solutions to (10.8). Example (1.1) is found to require enormously high currents, whereas other positions are less expensive. It is interesting to note that there is no particular propensity for α_{opt} as solved by (10.13) to turn out positive. In three of these examples, the arbitrary value of α_{min} has to be imposed for a properly aligned solution.

In examples (1.5) and (1.6), the G matrix of (10.8) is singular. These two are specifically chosen because (1.5) is a well-behaved singularity, and (1.6) is ill-behaved. Example (1.5) is typical of singularities in the MSS that arise from symmetry. These singularities occur when the seed is located on a plane of symmetry and the desired direction of motion is also within the plane of symmetry. In these cases, the G matrix is singular, but the unachievable space is perpendicular to all $d[\alpha]$ vectors, implying (10.15) is uniformly satisfied for all alpha. The choice of α is again arbitrary, and a good solution results. Example

	Currents, A/N						
#	-X	+X	-Y	+Y	-Z	+Z	α_{opt}
1.1	20161	20165	1389	13041	14300	15803	0.
1.2	-3847	4082	5616	-1675	-990	639	40.7
1.3	-3735	-5304	-3057	-3112	-2348	-1571	0.
1.4	6496	6580	5000	2805	2506	4002	0.
1.5	669	-16	669	-16	-1359	-1359	0.85
1.6	-	-	-	-	-	-	-4.18

Table 10.3: Coil currents, no-skid case.

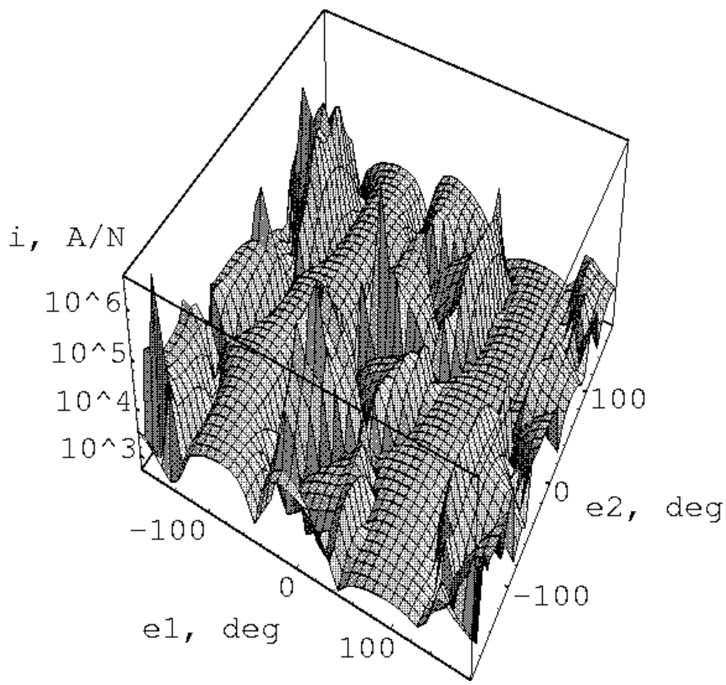


Figure 10.3: 2-norm of coil currents versus force orientation for example 1.2.

(1.6), however, does not yield a usable solution. When (10.15) is solved for the α that yields an answer perpendicular to the unachievable space, that α is negative – the only realizable alignment of the **b** vector with **f** has the wrong orientation.

If singularities such as (1.6) were a rare occurrence, a strategy would be to simply catalog and avoid them. However, this is not the case. Every point has some directions that are singular or badly conditioned in a MSS with six coils. For example, consider rotations of the desired force direction in (1.2). Let angles ε_1 and ε_2 represent rotation angles away from a nominal position, as detailed in Section 10.1.2. Figure 10.3 represents the 2-norm of the coil currents required to realize a no-skid force in the direction of (1.2) rotated by ε_1 and ε_2. From this figure, it is evident that even though the specific direction considered in (1.2) is nonsingular and well-behaved, many other possible force directions produced from the same point are very badly behaved.

	Currents, A/N							
#	-X	+X	-Y	+Y	-Z	+Z	α_{opt}	ε
2.1	1196	855	856	191	-477	-1717	0.	19.9^o
2.2	-431	1068	1721	154	-778	-1671	8.05	19.9^o
2.3	-610	-2755	971	730	441	917	0.94	8.49^o
2.4	3052	2709	1533	-1073	1994	1698	0.	18.04^o
2.5	669	-16	669	-16	-1360	-1360	0.85	$0.^o$
2.6	600	1282	-574	-2089	1214	577	0.97	8.94^o

Table 10.4: Coil currents, limited skid case.

10.2.2 Limited-Skid Case

For the limited-skid case, a misalignment of up to 20^o is arbitrarily deemed tolerable. Under the Limited-Skid conditions, currents are determined by solving (10.23) on a fine grid of perturbed seed orientations inside the allowable skid region. At any particular seed orientation, the solution only requires the inversion of a 6×6 matrix. At each orientation, the value of the arbitrary parameter α is chosen to be the greater of the α that minimizes (10.11) or α_{min}. The orientation with the lowest required $i'i$ is then chosen as the solution. Each of the seed location and force direction pairs in Table 10.2 is considered under the Limited-Skid conditions, and the solution currents are summarized in Table 389.

In general, there is a marked improvement in the current levels required for a given force. For instance, example (2.1) shows an order-of-magnitude decrease in the peak required current. The other examples exhibit a similar improvement in performance. Only (2.5) remains unchanged; the locally most efficient orientation is the no-skid orientation for this particular example. Of special note is example (2.6). In the no-skid configuration, no solution existed. With less than 10^o of misalignment, however, this example has a fairly economical solution.

The improvements in solution economy rely on the fact that economical orientations are often quite close to orientations that are prohibitively expensive to realize. In the particular case of (2.1), the no-skid solution is quite expensive. In Figure 10.4, it can be seen that the no-skid orientation (at the center of the figure) lies very close to a ridge of singularities. By allowing misalignment, an attitude at the far edge of the figure and away from the singularities is used.

10.2.3 Unlimited-Skid Case

The dipole orientation will now be considered unconstrained. Instead of simply solving linear equations, the unconstrained dipole orientation requires the solution of (10.4). A Newton-Raphson iteration starting from a randomly chosen set of currents was used to solve this equation. There are typically a finite number of local minima (usually about five) to which this iteration can converge, so the iteration was run several times to ensure a global minimum. The results for this scenario corresponding to the test points in Table 10.3 are summarized in Table 10.5.

As with the limited-skid case, the unlimited skid case produced a valid result for each case. The current magnitudes are roughly equivalent to the results of the limited-skid case, but each example now has an α of around 10; to realize these α_{min} of 10 in the limited skid case would increase the required current magnitudes, possibly considerably.

A similar solution to the unlimited skid case would be obtained by applying the limited-skid conditions over ε's ranging from -180^o to 180^o. The value of α_{min} would again be explicitly chosen, an option that does not exist in the quadratic formulation.

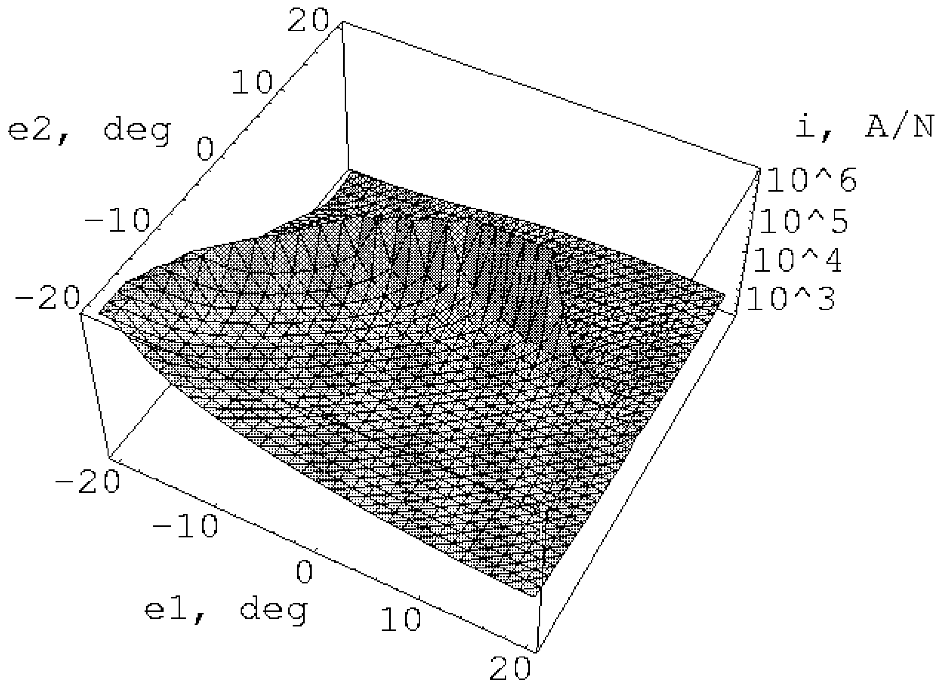


Figure 10.4: 2-norm of coil currents versus dipole orientation for example 2.1.

	Currents, A/N							
#	-X	+X	-Y	+Y	-Z	+Z	α_{opt}	ε
3.1	-1832	-273	69	1116	1025	226	9.73	90.58^o
3.2	135	-1442	1334	-251	-226	2290	9.08	103.68^o
3.3	362	-298	1537	-277	-953	-784	8.69	83.75^o
3.4	-1132	942	-1423	993	2210	-724	16.78	57.81^o
3.5	-31	-849	31	849	0.	0.	8.09	$90.^o$
3.6	-459	1439	254	-297	91	-1186	10.8	118.02^o

Table 10.5: Coil currents, unlimited skid case.

Chapter 11

Conclusions

This dissertation has presented a general form for considering the magnetic inverse problem in magnetic actuators. Specifically considered were inverse problems that have a homogeneous quadratic relationship between applied current and resulting force. Typically, many possible inverse mappings exist. The task is then not only to find an inverse, but to find, in some sense, the *best* inverse. Two different methods of inverting the current-to-force relationship were considered based on different definitions of an "optimal" inverse:

- A generalized bias linearization approach. Necessary conditions for bias linearization were considered. A numerical method for finding linearizing currents was developed, and an analytical method that can be used in some special but practical cases were also presented.

- A direct optimal method for minimizing a quadratic cost on the input currents. A continuation method was presented for solving this optimization problem. Extensions to the case of a saturating actuator were developed.

These methodologies were successfully applied to the specific examples of magnetic bearings and the magnetic stereotaxis system.

Several important results come from the application of these methods. The first is that magnetic bearings can be made fault-tolerant through the use of generalized bias linearization without any physical design change in the bearings themselves. This result may be crucial to the use of magnetic bearings in aircraft engine applications, where fault-tolerance is required.

The direct optimal methods may also be important to specific magnetic bearing applications. In precision applications, changes in physical dimensions due to thermal expansion may be unacceptable. The direct optimal methods keep power losses as small as possible while still achieving maximum load capacity.

The methods developed in this dissertation are also an important step in the development of the Magnetic Stereotaxis System. Previous controllers relied on the fact that the seed was in the exact center of the MSS so that the problem could be decomposed into 3 decoupled problems; however, this scheme breaks down rapidly as the seed moves away from the center of the helmet. The generalized bias linearization scheme as applied to the MSS is not limited to any specific coil geometry. This scheme implies that a MSS could be readily controlled with many small coils located close to the head, rather than a few large coils located farther away. The use of many coils would also help to eliminate the singularity problems that arise in the present MSS along lines of symmetry. Acceptable currents might then be found for any seed orientation, thus avoiding the need for "skidding" altogether.

Although not considered at length in this dissertation, the methods developed here could be used as a valuable tool in evaluating the utility of proposed actuator designs. Actuators could then be designed to give the best power loss performance or fault-tolerance in the case of magnetic bearings, or robustness to sensor error and seed position variation in the Magnetic Stereotaxis System.

There are many directions of further inquiry in which the present work could be extended. Specifically, the treatment could be extended to address the non-homogeneous quadratic current-to-force relationships that arise in machines employing permanent magnet biasing and in systems that are biased by gravity loading.

As of this writing, only limited portions of the theory developed in this dissertation have been implemented in hardware. A number of issues arising from implementation may need to be addressed. These issues might include:

- Details of transition between sets of linearizing currents when failures are detected. There may be problems with slew rate limiting during the transition.

- Robustness issues. Degradation of the inverse mappings in the presence of modeling errors has not been addressed, and could be of concern. Specifically in the case of the MSS, sensitivity to errors in the sensed position of the seed and to materials properties of the brain tissue have not been addressed.

With regard to bias linearization, several interesting theoretical questions remain unanswered. An interesting line of inquiry might be the discovery of a way of deriving the equations for a device characterized by magnetic circuits that yields a “natural” decomposed form, as occurs serendipitously for the MSS. Failing that, a better way of finding common totally isotropic spaces rather than by either intuiton or numerical search would be in order. Although tighter necessary conditions for linearizability have been developed in the present work, useful necessary and sufficient conditions for linearizability have yet to be found.

The application of continuation methods to the magnetic inverse problem is new, and much extension might be done with this approach. It would be interesting to attempt to prove that a continuation algorithm started from a valid i_o does not run into a singular point. Although no ill-conditioned cases have yet been encountered, it is not at all clear that the problem is well-conditioned for every device having one or more valid i_o vector. An interesting extension would also be the on-line integration of (7.6) in terms of time, rather than the artificial variable s. This might be included in some sort of feedback linearization controller and eliminate the need for a look-up table.

Bibliography

[aAFVJ85] M. Huneault amd A. Fahmideh-Vojdani and M. Juman. The continuation method in power system optimization: applications to economy-security functions. *IEEE Transactions on Power Apparatus and Systems*, 104:114–123, 1985.

[AC69] J. Abadie and J. Carpentier. Generalization of the wolfe reduced gradient method to the case of nonlinear constraints. In R. Fletcher, editor, *Optimization*, pages 34–47. Academic Press, 1969.

[AG80] E. Allgower and K Georg. Simplical and continuation methods for approximating fixed points. *SIAM Review*, 22:28–85, 1980.

[AK91] A. P. Allan and C. Knospe. A six-degree of freedom magnetic bearing for microgravity isolation. In *International Symposium on Magnetic Suspension Technology*, Hampton, VA, August 19-23 1991.

[AYT84] Y. H. Au-Yeung and N. K. Tsing. Some theorems on the generalized numerical ranges. *Linear and Multilinear Algebra*, 15:3–11, 1984.

[BH69] A. Bryson and Y. Ho. *Applied optimal control*. Blaisdell Publishing, 1969.

[BL91] P. Binding and C. K. Li. Joint ranges of hermitian matrices and simultaneous diagonalization. *Linear Algebra and its Applications*, 151:157–167, 1991.

[Bor91] K. R. Bornstein. Dynamic load capabilities of active electromagnetic bearings. *Journal of Tribology*, 113:598–603, July 1991.

[Bur88] C. R. Burrows. Design and application of a magnetic bearing for vibration control and stabilization of a flexible rotor. In *Proceedings of the First International Symposium on Magnetic Bearings*, Zurich, June 6-8 1988.

[BYM46] J. W. Beams, J. L. Young, and J. W. Moore. The production of high centrifugal fields. *Journal of Applied Physics*, 17:886–890, 1946.

[CC92] A. Charara and B. Caron. Magnetic bearings: comparison between linear and nonlinear functioning. In *Proceedings of the Third International Symposium on Magnetic Bearings*, Alexandria, VA, July 29-31 1992.

[CD88] H. M. Chen and M. S. Darlow. Magnetic bearings with rotating force control. *Journal of Tribology*, 110:100–105, January 1988.

[CR91] A. Chiba and M. Rahman. Radial force in a bearingless reluctance motor. *IEEE Transactions on Magnetics*, MAG-27:786–790, March 1991.

[Fed93] S. J. Fedigan. A real-time operating system for a magnetic bearings digital controller. Master's thesis, University of Virginia, 1993.

[Fee96] J. J. Feeley. A simple dynamic model for eddy currents in a magnetic actuator. *IEEE Transactions on Magnetics*, 32(2):453–458, March 1996.

[Fox71] R. L. Fox. *Optimization methods for engineering design*. Addison-Wesley, 1971.

[Gra90] M. S. Grady. Nonlinear magnetic stereotaxis: three-dimensional, in vivo remote magnetic manipulation of a small object in canine brain. *Medical Physics*, 17:405–415, 1990.

[GRB+94] G. T. Gillies, R. C. Ritter, W. C. Broaddus, M. S. Grady, M A. Howard, and R. G. McNeil. Magnetic manipulation instrumentation for medical physics research. *Review of Scietific Instruments*, 65:533–562, 1994.

[Gre96] E. W. Green. Control of magnetic actuators for increased force capability. Master's thesis, University of Virginia, 1996.

[HG90] M. Huneault and F. D. Galiana. An investigation of the solution to the optimal power flow problem incorporating continuation methods. *IEEE Transactions on Power Systems*, 5:103–110, 1990.

[HJ85] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1985.

[Iml90] J. Imlach. *The design and testing of radial magnetic bearings for an industrial canned motor pump*. PhD thesis, University of Virginia, 1990.

[Iwa90] S. Iwaki. The optimal location of electromagnets in multiple degree-of-freedom magnetically suspended actuators. *Transactions of the ASME,* 112:690–695, December 1990.

[Jac75] J. D. Jackson. *Classical electrodynamics.* Wiley, 2^{nd} edition, 1975.

[KA95] L. Kucera and M. Ahrens. A model for axial magnetic bearings including eddy currents. In *Proceedings of the Third International Symposium on Magnetic Suspension Technology*, Tallahasse, FL, December 13-15 1995.

[Kei93] F. J. Keith. *Implicit flux feedback control for magnetic bearings*. PhD thesis, University of Virginia, 1993.

[KMHW90] F. J. Keith, E. H. Maslen, R. R. Humphris, and R. D. Williams. Switching amplifier design for magnetic bearings. In *Proceedings of the Second International Symposium on Magnetic Bearings*, Tokyo, July 1990.

[Lam73] T. Y. Lam. *The algebraic theory of quadratic forms*. W. A. Benjamin, Inc., 1973.

[LK92] C. W. Lee and J. S. Kim. Modal testing and suboptimal vibration control of flexible rotor bearing system by using a magnetic bearing. *Journal of Dynamic Systems, Measurement, and Control*, 114:244–252, June 1992.

[LMP94] J. Lottin, P. Mouille, and J. C. Ponsart. Nonlinear control of active magnetic bearings. In *Proceedings of the Fourth International Symposium on Magnetic Bearings*, Zurich, 1994.

[LPGS94] J. P. Lyons, M. A. Preston, R. Gurumoorthy, and P. M. Szczesny. Design and control of a fault-tolerant active magnetic bearing system for aircraft engines. In *Proceedings of the Fourth International Symposium on Magnetic Bearings*, Zurich, 1994.

[MANS96] E. H. Maslen, P. E. Allaire, M. D. Noh, and C. K. Sortore. Magnetic bearings for reduced power consumption. *Journal of Tribology*, 118:To be published, July 1996.

[MHSH89] E. Maslen, P. Hermann, M. Scott, and R. Humphris. Practical limits to the performance of magnetic bearings: peak force, slew rate, and displacement sensitivity. *ASME Journal of Tribology*, 111:331–336, April 1989.

[ML92] P. Mouille and J. Lottin. Digital control of magnetic bearings: inversion of actuator model. In *Proceedings of the Third International Symposium on Magnetic Bearings*, Alexandria, VA, July 29-31 1992.

[MMN95] D. C. Meeker, E. H. Maslen, and M. D. Noh. A wide bandwidth model for the electrical impedance of magnetic bearings. In *Proceedings of the Third International Symposium on Magnetic Suspension Technology*, Tallahasse, FL, December 13-15 1995.

[Mol91] J. A. Molloy. Thermodynamics of moveable inductively heated seeds for the treatment of brain tumors. *Medical Physics*, 18:794–803, 1991.

[MRW95a] R. G. McNeil, R. C. Ritter, and B. Wang. Characteristics of an improved magnetic-implant guidance system. *IEEE Transaction on Biomedical Engineering*, 42:802–808, August 1995.

[MRW95b] R. G. McNeil, R. C. Ritter, and B. Wang. Functional design features and initial performance characteristics of a magnetic implant guidance system for stereotactic neurosurgery. *IEEE Transaction on Biomedical Engineering*, 42:793–801, August 1995.

[MY86] F. Matsumura and T. Yoshimoto. System modeling and control of a horizontal-shaft magnetic-bearing system. *IEEE Transactions on Magnetics*, MAG-22:197–206, May 1986.

[PG92] R. A. Ponrajah and F. D. Galiana. The minimum cost optimal power flow problem solved via the restart homotopy continuation method. *IEEE Transaction on Power Systems*, 4:138–148, 1992.

[Plo78] M. Plonus. *Applied electromagnetics*. McGraw-Hill, 1978.

[QWL91] E. G. Quate, K. G. Wika, and M. A. Lawson. Goniometric motion controller for the superconducting coil in a magnetic stereotaxis system. *IEEE Transaction on Biomedical Engineering*, 38:899–905, 1991.

[RGHG92] R. C. Ritter, M. S. Grady, M. A. Howard, and G. T. Gillies. Magnetic stereotaxis: computer-assisted, image-guided remote movement of implants in the brain. *Innovation et Technologie en Biologie et Medecine*, 13:437–449, 1992.

[Sad92] M. Sadiku. *Numerical techniques in electromagnetics*. CRC Press, 1992.

[Sch69] W. Scharlau. *Quadratic forms*. Queen's University Press, 1969.

[Sto74] R. L. Stoll. *The analysis of eddy currents*. Oxford University Press, 1974.

[TU91] N. H. Tsing and F. Uhlig. Inertia, numerical range, and zeros of quadratic forms for matrix pencils. *SIAM Journal of Matrix Analysis Applications*, 12(1):146–159, January 1991.

[WH89] L. T. Watson and R. T. Haftka. Modern homotopy methods in optimization. *Computer Methods in Applied Mechanics and Engineering*, 74:289–305, 1989.

[ZAK87] R. B. Zmood, D. K. Anand, and J. A. Kirk. The influence of eddy currents of magnetic actuator performance. *Proceedings of the IEEE*, 8(2):259–260, 1987.

Appendix A

Symmetric 8-pole magnetic bearing

The symmetric eight-pole bearing (pictured in Figure A.1) currently is common in moderate capacity radial bearing design; an examination of the characteristics of this design is therefore of great practical value.

To deal with this bearing in as general a way as possible, it is desirable to define a non-dimensionalized bearing. Define non-dimensional current, flux density, force, and displacement as respectively:

$$\underline{b} = \frac{b}{b_{sat}} \tag{A.1}$$

$$\underline{i} = \left(\frac{\mu_o N}{g_o b_{sat}}\right) i \tag{A.2}$$

$$\underline{f} = \left(\frac{\mu_o}{a b_{sat}^2}\right) f \tag{A.3}$$

$$\underline{x} = \frac{x}{g_o};\ \underline{y} = \frac{y}{g_o} \tag{A.4}$$

where a is pole area, g_o is nominal air gap, and N turns of wire are wound independently around each pole. Analyzing the magnetic circuits for this non-dimensional bearing yields the following matrices:

$$R = \begin{bmatrix} g_1 & -g_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_2 & -g_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_3 & -g_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_4 & -g_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_5 & -g_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & g_6 & -g_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_7 & -g_8 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathtt{N} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where

$$g_j = 1 - \underline{x}\cos\left(\frac{j\pi}{4} + \frac{\pi}{8}\right) - \underline{y}\sin\left(\frac{j\pi}{4} + \frac{\pi}{8}\right)$$

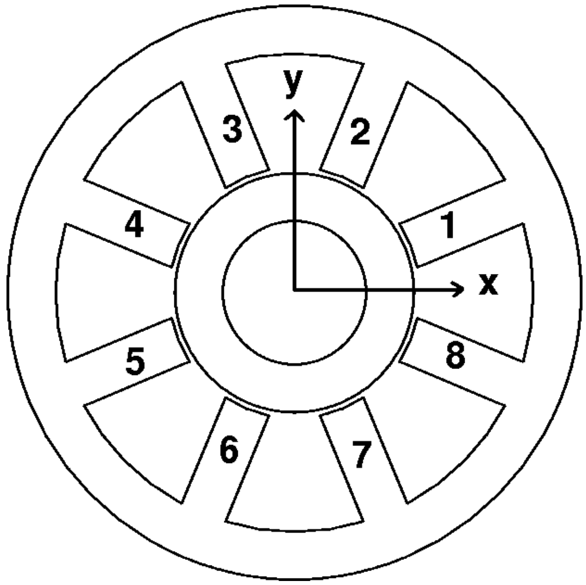


Figure A.1: 8-pole symmetric bearing.

For the centered position, the flux in each gap is determined by

$$\underline{b} = V\,\underline{i} = \begin{bmatrix} 7 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 7 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 7 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 7 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 7 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 7 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 7 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 7 \end{bmatrix} \underline{i}$$

The M matrices for each force direction can now be built via (3.14):

$$M_x = \begin{bmatrix} 0.346455 & -0.0816602 & -0.0338248 & 0 & 0 & -0.0338248 & -0.0816602 & -0.115485 \\ -0.0816602 & 0.143506 & 0 & 0.0338248 & 0.0338248 & 0 & -0.0478354 & -0.0816602 \\ -0.0338248 & 0 & -0.143506 & 0.0816602 & 0.0816602 & 0.0478354 & 0 & -0.0338248 \\ 0 & 0.0338248 & 0.0816602 & -0.346455 & 0.115485 & 0.0816602 & 0.0338248 & 0 \\ 0 & 0.0338248 & 0.0816602 & 0.115485 & -0.346455 & 0.0816602 & 0.0338248 & 0 \\ -0.0338248 & 0 & 0.0478354 & 0.0816602 & 0.0816602 & -0.143506 & 0 & -0.0338248 \\ -0.0816602 & -0.0478354 & 0 & 0.0338248 & 0.0338248 & 0 & 0.143506 & -0.0816602 \\ -0.115485 & -0.0816602 & -0.0338248 & 0 & 0 & -0.0338248 & -0.0816602 & 0.346455 \end{bmatrix}$$

$$M_y = \begin{bmatrix} 0.143506 & -0.0816602 & -0.0816602 & -0.0478354 & 0 & 0.0338248 & 0.0338248 & 0 \\ -0.0816602 & 0.346455 & -0.115485 & -0.0816602 & -0.0338248 & 0 & 0 & -0.0338248 \\ -0.0816602 & -0.115485 & 0.346455 & -0.0816602 & -0.0338248 & 0 & 0 & -0.0338248 \\ -0.0478354 & -0.0816602 & -0.0816602 & 0.143506 & 0 & 0.0338248 & 0.0338248 & 0 \\ 0 & -0.0338248 & -0.0338248 & 0 & -0.143506 & 0.0816602 & 0.0816602 & 0.0478354 \\ 0.0338248 & 0 & 0 & 0.0338248 & 0.0816602 & -0.346455 & 0.115485 & 0.0816602 \\ 0.0338248 & 0 & 0 & 0.0338248 & 0.0816602 & 0.115485 & -0.346455 & 0.0816602 \\ 0 & -0.0338248 & -0.0338248 & 0 & 0.0478354 & 0.0816602 & 0.0816602 & -0.143506 \end{bmatrix}$$

Position-dependence of these matrices can be approximated by the one-term Taylor series described in (3.36). Necessities for computing derivatives of the force matrices with respect to displacement are the derivatives of the reluctance matrix R with respect to position:

$$\frac{\partial R}{\partial \underline{x}} = \begin{bmatrix} -\cos\frac{\pi}{8} & \cos\frac{3\pi}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\cos\frac{3\pi}{8} & \cos\frac{5\pi}{8} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\cos\frac{5\pi}{8} & \cos\frac{7\pi}{8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\cos\frac{7\pi}{8} & \cos\frac{9\pi}{8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\cos\frac{9\pi}{8} & \cos\frac{11\pi}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\cos\frac{11\pi}{8} & \cos\frac{13\pi}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\cos\frac{13\pi}{8} & \cos\frac{15\pi}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial R}{\partial \underline{y}} = \begin{bmatrix} -\sin\frac{\pi}{8} & \sin\frac{3\pi}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\sin\frac{3\pi}{8} & \sin\frac{5\pi}{8} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sin\frac{5\pi}{8} & \sin\frac{7\pi}{8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\sin\frac{7\pi}{8} & \sin\frac{9\pi}{8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sin\frac{9\pi}{8} & \sin\frac{11\pi}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\sin\frac{11\pi}{8} & \sin\frac{13\pi}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\sin\frac{13\pi}{8} & \sin\frac{15\pi}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The derivatives of M_x and M_y are then computed by substituting into (3.39) to yield:

$$\frac{\partial M_x}{\partial \underline{x}} = \begin{bmatrix} 0.595971 & -0.106694 & -0.0183058 & -0.0441942 & -0.0441942 & -0.0183058 & -0.106694 & -0.257583 \\ -0.106694 & 0.154029 & 0.0441942 & -0.0183058 & -0.0183058 & 0.0441942 & 0.00758252 & -0.106694 \\ -0.0183058 & 0.0441942 & 0.154029 & -0.106694 & -0.106694 & 0.00758252 & 0.0441942 & -0.0183058 \\ -0.0441942 & -0.0183058 & -0.106694 & 0.595971 & -0.257583 & -0.106694 & -0.0183058 & -0.0441942 \\ -0.0441942 & -0.0183058 & -0.106694 & -0.257583 & 0.595971 & -0.106694 & -0.0183058 & -0.0441942 \\ -0.0183058 & 0.0441942 & 0.00758252 & -0.106694 & -0.106694 & 0.154029 & 0.0441942 & -0.0183058 \\ -0.106694 & 0.00758252 & 0.0441942 & -0.0183058 & -0.0183058 & 0.0441942 & 0.154029 & -0.106694 \\ -0.257583 & -0.106694 & -0.0183058 & -0.0441942 & -0.0441942 & -0.0183058 & -0.106694 & 0.595971 \end{bmatrix}$$

$$\frac{\partial M_x}{\partial \underline{y}} = \begin{bmatrix} 0.220971 & -0.150888 & -0.0441942 & 0 & -0.0441942 & -0.0258883 & 0.0441942 & 0 \\ -0.150888 & 0.220971 & 0 & 0.0441942 & -0.0258883 & -0.0441942 & 0 & -0.0441942 \\ -0.0441942 & 0 & -0.220971 & 0.150888 & 0.0441942 & 0 & 0.0441942 & 0.0258883 \\ 0 & 0.0441942 & 0.150888 & -0.220971 & 0 & -0.0441942 & 0.0258883 & 0.0441942 \\ -0.0441942 & -0.0258883 & 0.0441942 & 0 & 0.220971 & -0.150888 & -0.0441942 & 0 \\ -0.0258883 & -0.0441942 & 0 & -0.0441942 & -0.150888 & 0.220971 & 0 & 0.0441942 \\ 0.0441942 & 0 & 0.0441942 & 0.0258883 & -0.0441942 & 0 & -0.220971 & 0.150888 \\ 0 & -0.0441942 & 0.0258883 & 0.0441942 & 0 & 0.0441942 & 0.150888 & -0.220971 \end{bmatrix}$$

$$\frac{\partial M_y}{\partial \underline{x}} = \begin{bmatrix} 0.220971 & -0.150888 & -0.0441942 & 0 & -0.0441942 & -0.0258883 & 0.0441942 & 0 \\ -0.150888 & 0.220971 & 0 & 0.0441942 & -0.0258883 & -0.0441942 & 0 & -0.0441942 \\ -0.0441942 & 0 & -0.220971 & 0.150888 & 0.0441942 & 0 & 0.0441942 & 0.0258883 \\ 0 & 0.0441942 & 0.150888 & -0.220971 & 0 & -0.0441942 & 0.0258883 & 0.0441942 \\ -0.0441942 & -0.0258883 & 0.0441942 & 0 & 0.220971 & -0.150888 & -0.0441942 & 0 \\ -0.0258883 & -0.0441942 & 0 & -0.0441942 & -0.150888 & 0.220971 & 0 & 0.0441942 \\ 0.0441942 & 0 & 0.0441942 & 0.0258883 & -0.0441942 & 0 & -0.220971 & 0.150888 \\ 0 & -0.0441942 & 0.0258883 & 0.0441942 & 0 & 0.0441942 & 0.150888 & -0.220971 \end{bmatrix}$$

$$\frac{\partial M_y}{\partial \underline{y}} = \begin{bmatrix} 0.154029 & -0.106694 & -0.106694 & 0.00758252 & 0.0441942 & -0.0183058 & -0.0183058 & 0.0441942 \\ -0.106694 & 0.595971 & -0.257583 & -0.106694 & -0.0183058 & -0.0441942 & -0.0441942 & -0.0183058 \\ -0.106694 & -0.257583 & 0.595971 & -0.106694 & -0.0183058 & -0.0441942 & -0.0441942 & -0.0183058 \\ 0.00758252 & -0.106694 & -0.106694 & 0.154029 & 0.0441942 & -0.0183058 & -0.0183058 & 0.0441942 \\ 0.0441942 & -0.0183058 & -0.0183058 & 0.0441942 & 0.154029 & -0.106694 & -0.106694 & 0.00758252 \\ -0.0183058 & -0.0441942 & -0.0441942 & -0.0183058 & -0.106694 & 0.595971 & -0.257583 & -0.106694 \\ -0.0183058 & -0.0441942 & -0.0441942 & -0.0183058 & -0.106694 & -0.257583 & 0.595971 & -0.106694 \\ 0.0441942 & -0.0183058 & -0.0183058 & 0.0441942 & 0.00758252 & -0.106694 & -0.106694 & 0.154029 \end{bmatrix}$$

A.1 Failure configuration bias linearization currents

In all, there are 93 different ways that a bearing can fail between 0 and 3 coils. All of these failures are, however, described by only 11 unique failure maps due to the symmetry of the bearing; all other mappings are simply rotations and permutations of these unique maps. If a "1" represents an active coil, and a "0" represents an inactive coil, the unique configurations are:

No.	Coils On
1	11111111
2	01111111
3	00111111
4	01011111
5	01101111
6	01110111
7	00011111
8	00101111
9	00110111
10	01010111
11	01011011

As explored in Chapter 249, an analytical method can be used to decompose the bias linearization problem and yield manifolds of solutions. Alternatively, a purely numerical approach (the reduced gradient method) can be used to solve for linearizing currents. Whichever method is used, the problem ultimately becomes a numerical search. If the analytical method is used, it is not clear *a priori* which manifold results in the best solution. It is also not clear which arbitrarily chosen coefficients q yield the best solution. The best q must be found by some sort of search. Because no method for finding an optimal set of linearizing currents is clearly the most efficient, the reduced gradient method was used to find a W for each failure configuration. A description and listing of the program that implements the reduced gradient method is given in § A.2. For the purposes of computing b_{max}, the width of the back iron and journal iron was assumed to be $\frac{1}{2}$ the width of the legs. This is a typical design for 8-pole bearings run with a *NSNS* biasing scheme.

With a typical *NSNS* control scheme with all coils active and limiting saturation occurring in the legs, the flux levels in the gap would be set to $\frac{1}{2}$ the saturation level, implying the magnitude of $\underline{i} = 0.5$ in each coil. The maximum load would then be approximately $\underline{f}_{max} = 1$.

Through use of W that do not employ certain coils, the occurrence of faults can be tolerated. The price for these failures is a decrease in bearing load capacity. The relative load capacity for each failure configuration using the best discovered W for each set and using (8.21) to define load capacity is:

No.	Relative Load Capacity
1	100%
2	100%
3	48.0%
4	44.0%
5	51.3%
6	56.5%
7	14.0%
8	26.0%
9	36.7%
10	41.0%
11	31.3%

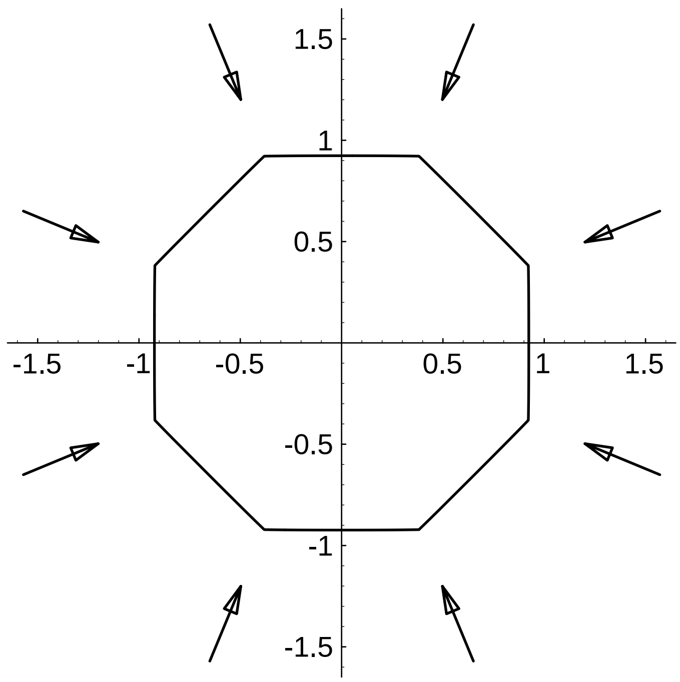
For each set of failure currents, the accompanying plot represents load capacity versus force orientation. Active coil locations are represented by arrows on these plots. The reported load capacity, $\underline{f}_{max}$, is load capacity for the worst force orientation, as specified by (8.21).

To convert $\underline{W}$ into dimensional units,

$$W = \underline{W} \cdot \begin{bmatrix} \left(\frac{g_o\, b_{sat}}{\mu_o\, N}\right) & 0 & 0 \\ 0 & \left(\frac{g_o}{a\, b_{sat}\, N}\right) & 0 \\ 0 & 0 & \left(\frac{g_o}{a\, b_{sat}\, N}\right) \end{bmatrix} \tag{A.5}$$

This conversion puts the first column of W into units of current, and the second and third columns into current per unit force. To achieve the optimal load capacity, the value of $\hat{i}_o$ should be 1.

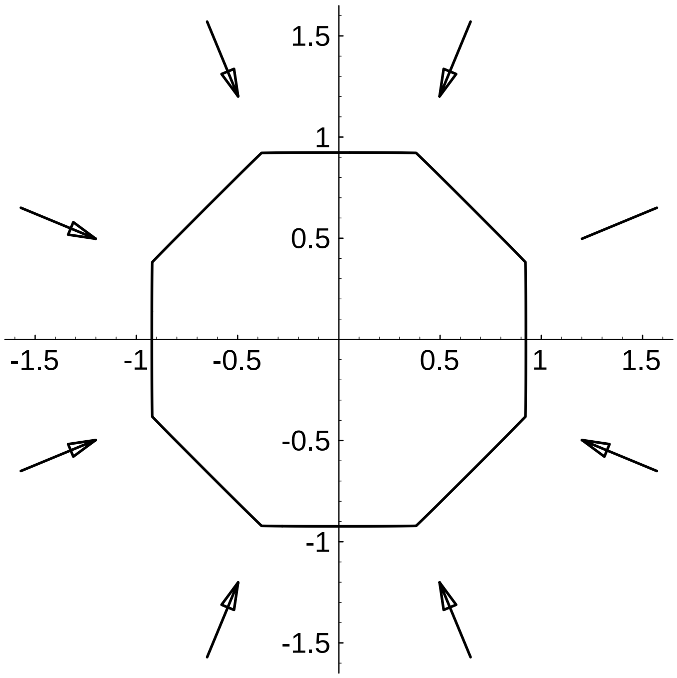
A.1.1 Case 1



$$\underline{W} = \begin{bmatrix} 0.505134 & 0.457245 & 0.189397 \\ -0.505134 & -0.189397 & -0.457245 \\ 0.505134 & -0.189397 & 0.457245 \\ -0.505134 & 0.457245 & -0.189397 \\ 0.505134 & -0.457245 & -0.189397 \\ -0.505134 & 0.189397 & 0.457245 \\ 0.505134 & 0.189397 & -0.457245 \\ -0.505134 & -0.457245 & 0.189397 \end{bmatrix}$$

$$\underline{f}_{max} = 0.923782$$

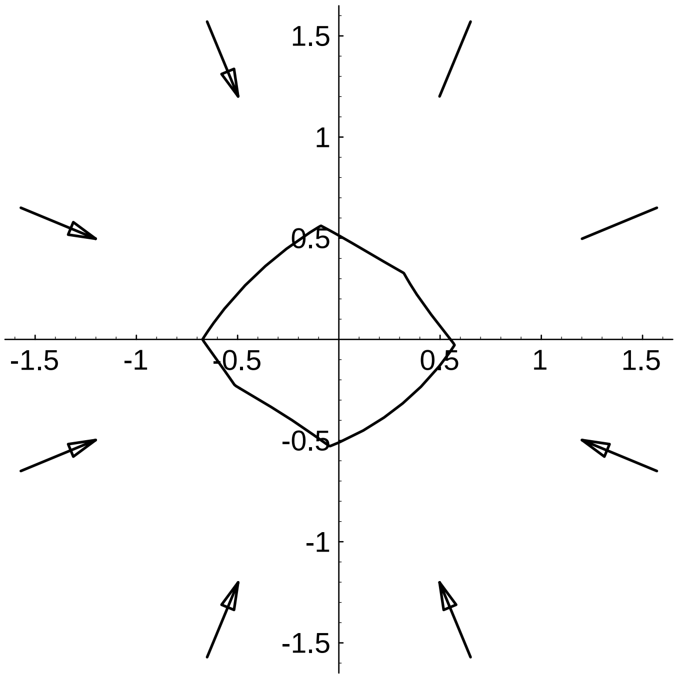
A.1.2 Case 2



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ -1.01027 & -0.646641 & -0.646641 \\ 0. & -0.646641 & 0.267848 \\ -1.01027 & 0. & -0.378793 \\ 0. & -0.914488 & -0.378793 \\ -1.01027 & -0.267848 & 0.267848 \\ 0. & -0.267848 & -0.646641 \\ -1.01027 & -0.914488 & 0. \end{bmatrix}$$

$$\underline{f}_{max} = 0.923782$$

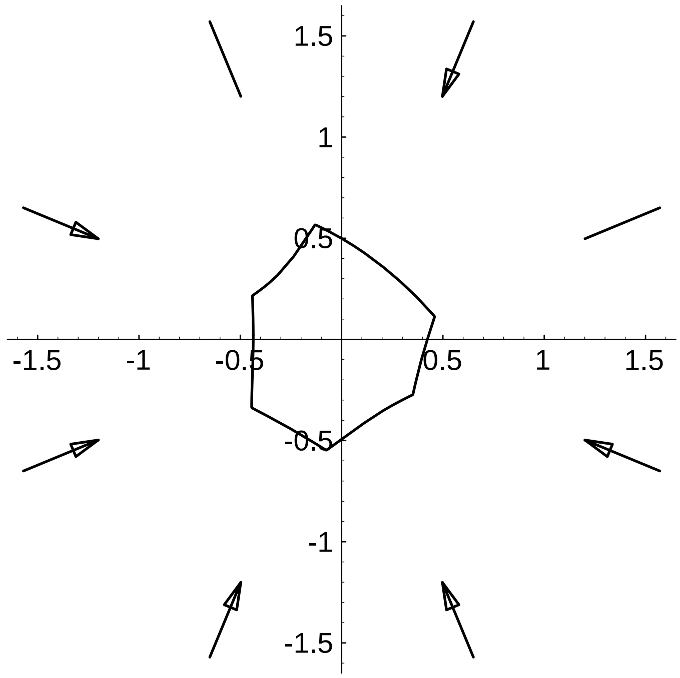
A.1.3 Case 3



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ 0. & 0. & 0. \\ -0.745322 & -0.505285 & -1.24809 \\ 0.274924 & -1.19563 & -0.482923 \\ -0.500997 & 0.207802 & 0.0736056 \\ -0.30642 & -0.270924 & -0.119114 \\ 0.333943 & -0.576502 & -1.28494 \\ -0.717777 & -1.29388 & -0.547609 \end{bmatrix}$$

$$\underline{f}_{max} = 0.443783$$

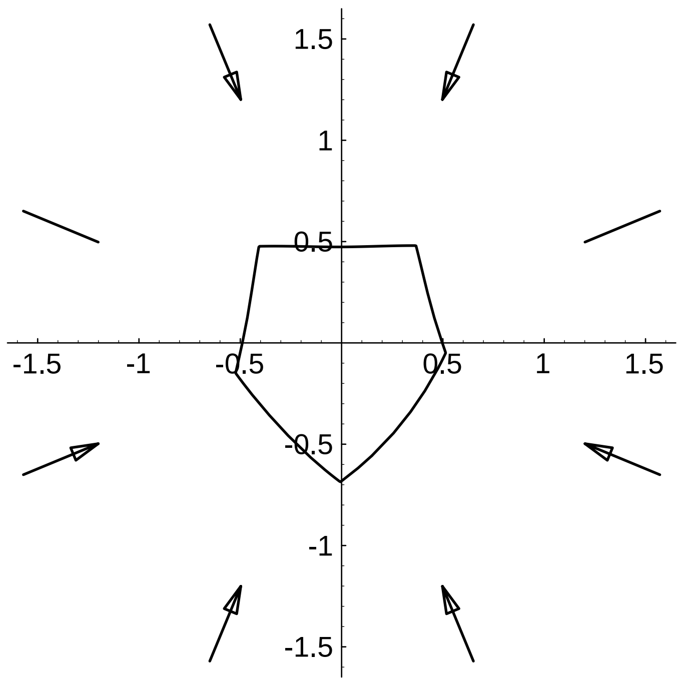
A.1.4 Case 4



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ 0.4572 & 0.829465 & 1.79418 \\ 0. & 0. & 0. \\ 0.420565 & -1.42794 & 1.73208 \\ -0.168895 & 0.334289 & 1.78272 \\ 0.392786 & -0.147955 & -0.231423 \\ 0.1394 & 0.922727 & 1.40995 \\ -0.263455 & -1.18527 & 1.91783 \end{bmatrix}$$

$$\underline{f}_{max} = 0.40625$$

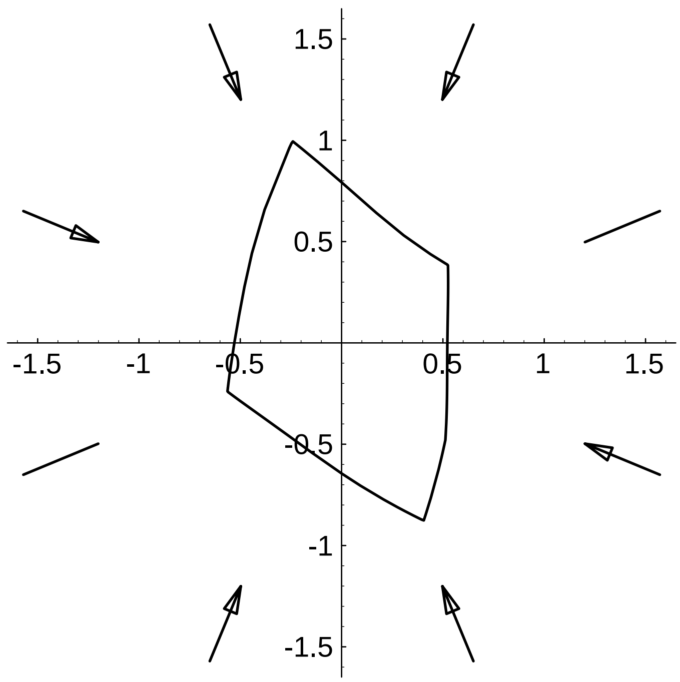
A.1.5 Case 5



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ 0.445917 & 0.731063 & 0.828046 \\ -0.420705 & 0.727704 & -0.623231 \\ 0. & 0. & 0. \\ 0.527528 & -0.640086 & -0.0244798 \\ -0.250088 & 0.483385 & 0.82879 \\ 0.31721 & 0.441614 & -0.619126 \\ -0.48957 & -0.646243 & 0.180955 \end{bmatrix}$$

$$\underline{f}_{max} = 0.473622$$

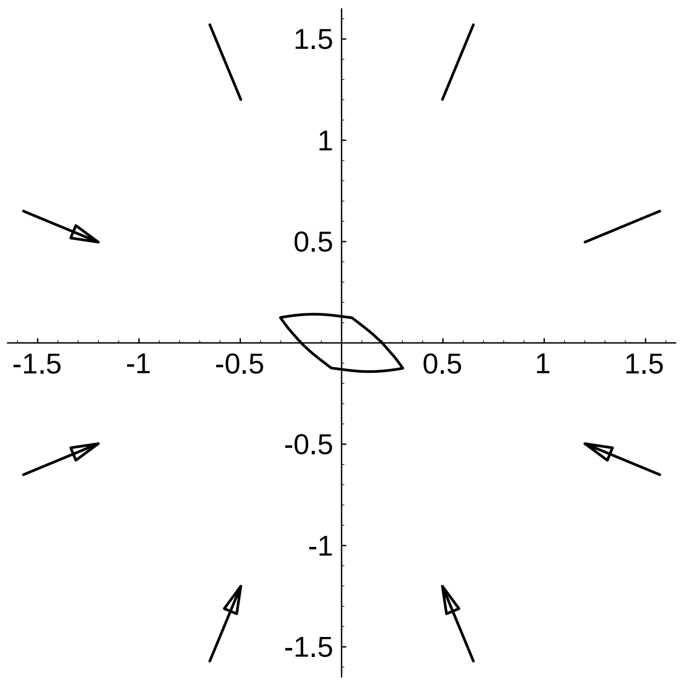
A.1.6 Case 6



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ -0.265606 & -0.416743 & -1.28104 \\ 0.485629 & 0.102261 & 0.146081 \\ -0.3711 & 1.1636 & -0.490013 \\ 0. & 0. & 0. \\ 0.439899 & -0.416743 & -1.28104 \\ -0.299692 & 0.102261 & 0.146081 \\ 0.535831 & 1.1636 & -0.490013 \end{bmatrix}$$

$$\underline{f}_{max} = 0.521902$$

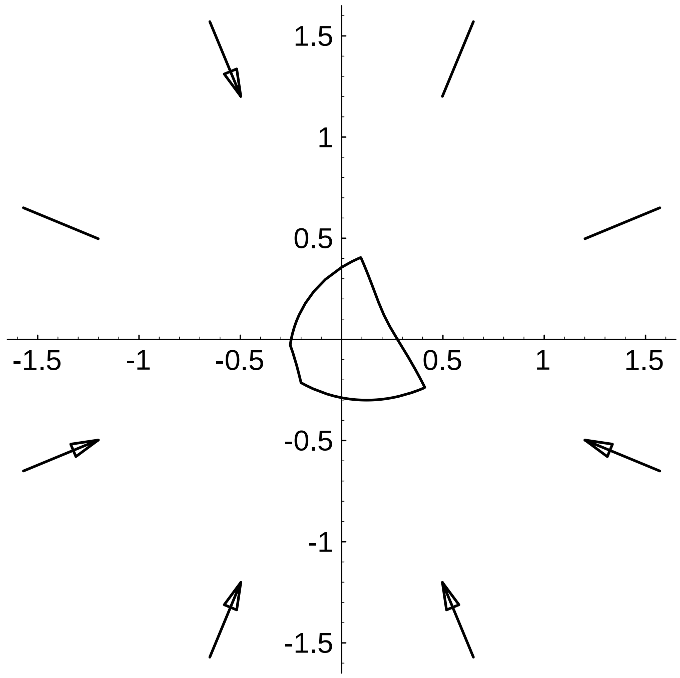
A.1.7 Case 7



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ -0.666665 & 0.448349 & -1.53073 \\ 0. & -0.951079 & -2.29611 \\ 0. & -0.951079 & -2.29611 \\ 0. & -0.951079 & -2.29611 \\ -0.666662 & -1.39943 & -0.76538 \end{bmatrix}$$

$$\underline{f}_{max} = 0.129543$$

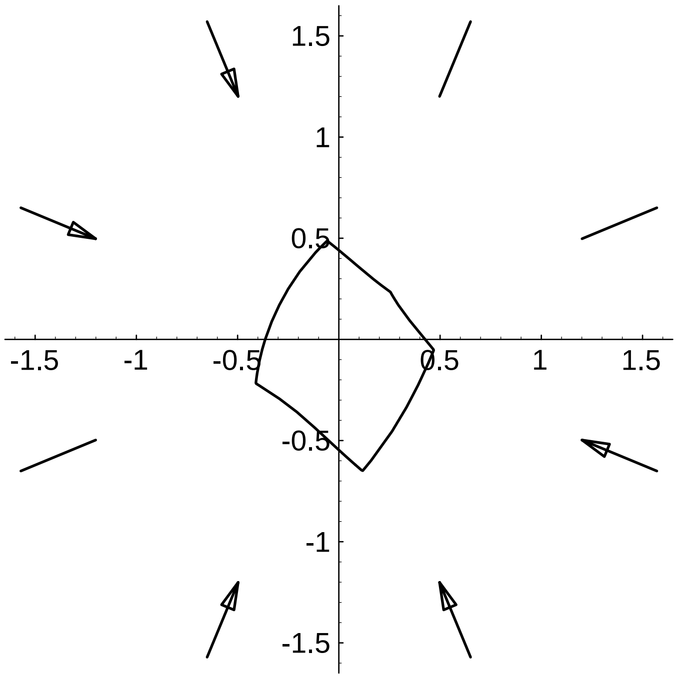
A.1.8 Case 8



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ 0. & 0. & 0. \\ -0.710937 & -0.234225 & -1.34591 \\ 0. & 0. & 0. \\ 0.241507 & -1.75379 & -1.0097 \\ -0.405378 & -0.748979 & -0.108603 \\ 0.162386 & -0.864742 & -1.53023 \\ -0.678722 & -1.69752 & -0.654697 \end{bmatrix}$$

$$\underline{f}_{max} = 0.240411$$

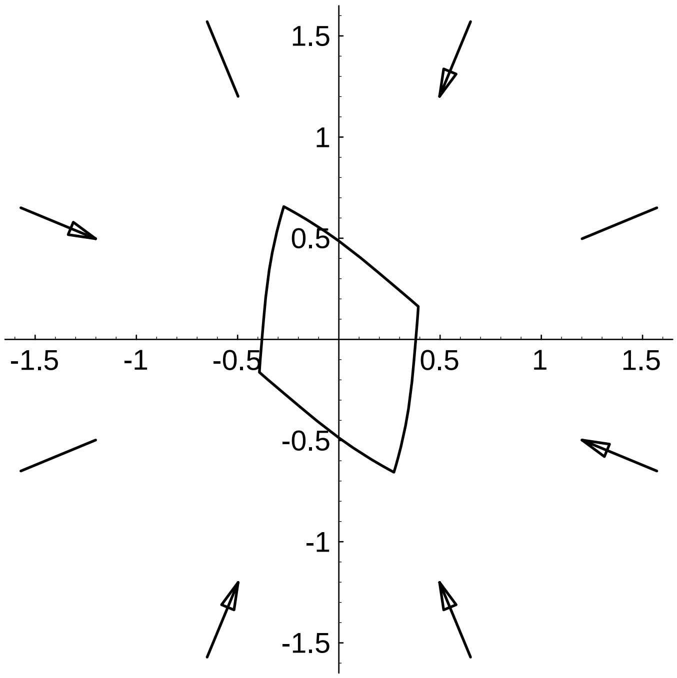
A.1.9 Case 9



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0.700709 & 0.74353 & 1.54867 \\ -0.128455 & 2.02657 & 0.939184 \\ 0. & 0. & 0. \\ 0.504082 & 0. & 0. \\ -0.230737 & 0.74353 & 1.54867 \\ 0.589266 & 2.02657 & 0.939184 \end{bmatrix}$$

$$\underline{f}_{max} = 0.338875$$

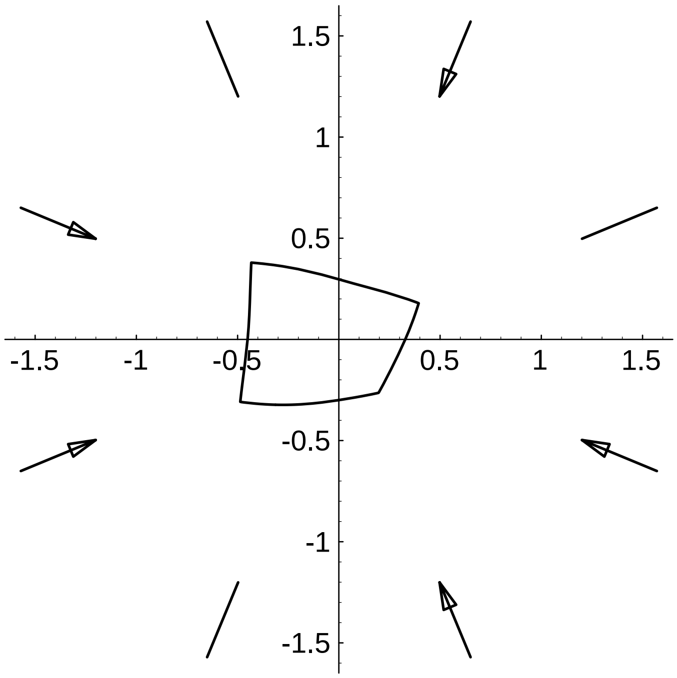
A.1.10 Case 10



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ -0.691558 & -0.618983 & -1.5709 \\ 0. & 0. & 0. \\ -0.691552 & 1.54853 & -0.673114 \\ 0. & 0. & 0. \\ -0.0887534 & -0.618983 & -1.5709 \\ -0.794338 & 0. & 0. \\ -0.0887668 & 1.54853 & -0.673114 \end{bmatrix}$$

$$\underline{f}_{max} = 0.378526$$

A.1.11 Case 11



$$\underline{W} = \begin{bmatrix} 0. & 0. & 0. \\ -0.644074 & -1.05866 & -1.11147 \\ 0. & 0. & 0. \\ -0.0497678 & -2.14374 & 0.963628 \\ -0.658303 & -0.106088 & 0.729123 \\ 0. & 0. & 0. \\ 0.0724449 & -0.731037 & -1.11456 \\ -0.542448 & -2.25897 & 0.623658 \end{bmatrix}$$

$$\underline{f}_{max} = 0.289147$$

A.2 C code for determining W via the reduced gradient method

A.2.1 Program *findw.c*

This program implements a reduced gradient method search for a locally optimal W matrix in the sense of giving the highest possible load capacity. This program specifically addresses radial magnetic bearings in which each pole has the same face area, nominal gap, and number of turns. This program treats bearings in the same non-dimensional formulation developed earlier in this appendix.

The program first reads a data file, specified as a command line option, that describes the geometry of the particular bearing in question. Then, a number of feasible solutions (specified by `HOW_MANY`) are found using the modified Newton-Raphson method. After the specified number of feasible solutions are found, the best feasible solution is used as the starting point of a reduced gradient descent. The length of each gradient step is specified by `GradStepLength`. A value of 0.01 has been found to work well. A number of gradient steps are taken until further steps bring no improvement in solution cost. The program is then terminated, and the best W found is reported along with the nondimensional load capacity associated with that W. The output is correctly scaled so that (451) can be used to convert the output to dimensional units.

Specific subroutines employed in the program are as follows:

GetMaxB This subroutine takes a set of linearizing currents and a specific biasing level and returns the absolute value of the worst-case flux density over all θ resulting from a force of magnitude 1. Since each particular flux density is the sum of a constant, a sine, and a cosine, the worst-case magnitude is easily determined by adding the absolute value of the constant component to the square root of the sum of the squared coefficients of the sine and cosine terms.

Rule This subroutine evaluates the $\underline{b}_m ax$ produced for a given bias linearization solution for the best case biasing level ζ. A golden section search is performed to find the best ζ.

IRule An alternate entry into `Rule`.

ReadEm Reads the specified data file and forms the matrices M_x, M_y, and V_s necessary to find and rank various W matrices.

MakeH Creates the H matrix.

```
#include<math.h>
#include<stdio.h>
#include<stdlib.h>

#define DT 0.5          /* modified Newton-Raphson step size */
#define HOW_MANY 1000   /* number of solutions to be found */
#define MAXSTEPS 1000   /* maximum iterations allowed for any one soln.   */
#define VERB FALSE      /* flag for verbose mode */
#define Del 0.0001      /* step length used to determine reduced gradient */
#define GradStepLength        0.01

struct Entry{
        int co,ve,ma;
};

#include"mathstuf.c"

double GetMaxB(I,V,Vb,k,dim)
        double **I,**V,**Vb;
        double k;
        int dim;
```

```
{
        int i;
        double max,*rb,*rx,*ry,*res;

        res=(double *)calloc(dim,sizeof(double));
        rx=(double *)calloc(dim,sizeof(double));
        ry=(double *)calloc(dim,sizeof(double));
        rb=(double *)calloc(dim,sizeof(double));

                max=0.;
                vTimes(V,I[0],rb,dim);
                vTimes(V,I[1],rx,dim);
                vTimes(V,I[2],ry,dim);
                for(i=0;i<dim;i++)
                        res[i]=k*fabs(rb[i]) + sqrt(rx[i]*rx[i]+ry[i]*ry[i])/k;
                for(i=0;i<dim;i++)
                        if(res[i]>max) max=res[i];

                vTimes(Vb,I[0],rb,dim);
                vTimes(Vb,I[1],rx,dim);
                vTimes(Vb,I[2],ry,dim);
                for(i=0;i<dim;i++)
                        res[i]=k*fabs(rb[i]) + sqrt(rx[i]*rx[i]+ry[i]*ry[i])/k;
                for(i=0;i<dim;i++)
                        if(res[i]>max) max=res[i];

        free(res); free(rx); free(ry); free(rb);
        return max;
}

double Rule(I,V,Vb,dim,zeta)
        double **I,**V,**Vb,*zeta;
        int dim;
{
        int i,j,l;
        double c1=0,c2=0,c3=0,c4=0,H,max=0,t,Bm,z0,z1;
        double q[4],k[4];

/*        c1 = Ibias . I bias
        c2 = Ix . Ix
        c3 = Iy . Iy
        c4 = Ix . Iy
*/

        for(i=0;i<dim;i++)
        {
                c1+=(I[0][i]*I[0][i]);
                c2+=(I[1][i]*I[1][i]);
                c3+=(I[2][i]*I[2][i]);
                c4+=(I[1][i]*I[2][i]);
        }

        /* get an initial estimate for bias level */
        for(t=0.0,j=0,H=0.0;t<2*Pi;t+=Pi/300.)
        {
```

```
                j++;
                H+=sqrt(sqrt((2*cos(t)*sin(t)*c4 + cos(t)*cos(t)*c2 +
                                   sin(t)*sin(t)*c3)/c1));
        }
        H/=((double) j);

        k[0]=0.5*H;k[1]=0.8*H;k[2]=1.2*H;k[3]=1.5*H;
        for(j=0;j<4;j++) q[j]=GetMaxB(I,V,Vb,k[j],dim);

        for(j=0;j<40;j++)
        {
                for(i=1,l=0;i<4;i++)
                        if(q[i]<q[l]) l=i;

                if(l<2){
                        k[3]=k[2];q[3]=q[2];
                }
                else{
                        k[0]=k[1];q[0]=q[1];
                }

                k[1]=k[0]+0.3*(k[3]-k[0])/2.;
                k[2]=k[3]-0.3*(k[3]-k[0])/2.;
                q[1]=GetMaxB(I,V,Vb,k[1],dim);
                q[2]=GetMaxB(I,V,Vb,k[2],dim);
        }

        for(i=1,l=0;i<4;i++) if(q[i]<q[l]) l=i;
        *zeta=k[l];
        return q[l];
}

double IRule(I,V,Vb,ncoils,dim,zeta,out)
        double *I,**V,**Vb,*zeta,**out;
        int ncoils,dim;
{
        int i,j;

        for(i=0;i<ncoils;i++)
                for(j=0;j<3;j++)
                        if(i<dim) out[j][i]=I[j*dim+i];
                        else out[j][i]=0.;
        return Rule(out,V,Vb,ncoils,&zeta);
}

int ReadEm(fi,x,y,p,v,vb,Ncoils,Dim)
        char *fi;
        double ***x,***y,***p,***v,***vb;
        int *Ncoils,*Dim;
{

        int i,j,k,dim,ncoils,*onflag,errflag=FALSE;
        double **Mx,**My,**I,**R,**T,**V,**Vt,**P;
        double *an,*e;
        double J;
```

```
	char st[80];
	FILE *fp;

	if ((fp=fopen(fi,"rt"))!=NULL)
	{
		ncoils=0;
		while(fgets(st,80,fp)!=NULL)
			if (strlen(st)>2) ncoils++;
		if (VERB==TRUE) printf("%i coils.\n",ncoils);
		fclose(fp);
		fp=fopen(fi,"rt");
		an=(double *)calloc(ncoils,sizeof(double));
		onflag=(int *)calloc(ncoils,sizeof(int));
		for(i=0;i<ncoils;i++)
		{
			fscanf(fp,"%lf",&an[i]);
			an[i]*=(Pi/180.);
			fscanf(fp,"%i",&onflag[i]);
			if (VERB==TRUE)
				printf("%f %i\n",an[i],onflag[i]);
		}
		fclose(fp);

		/* sort so that active coils come first */
		P=MatrixAlloc(ncoils,ncoils);
		for(dim=0,i=0;i<ncoils;i++){
			if (onflag[i]==TRUE) dim++;
			P[i][i]=1.0;
		}
		for(i=0;i<ncoils-1;i++)
			for(j=0;j<ncoils-1;j++)
			{
				if (onflag[j+1]>onflag[j])
				{
					e=P[j];P[j]=P[j+1];P[j+1]=e;
					J=an[j];an[j]=an[j+1];an[j+1]=J;
					k=onflag[j];onflag[j]=onflag[j+1];
					onflag[j+1]=k;
				}
			}

		/* create Mx,My matrices */
		R=MatrixAlloc(ncoils,ncoils);
		T=MatrixAlloc(ncoils,ncoils);
		Mx=MatrixAlloc(ncoils,ncoils);
		My=MatrixAlloc(ncoils,ncoils);
		V=MatrixAlloc(ncoils,ncoils);
		Vt=MatrixAlloc(ncoils,ncoils);

		for(i=0;i<(ncoils-1);i++)
		{
			R[i][i]=1.;
			R[i][i+1]=-1.;
			T[i][i]=1.;
			T[i][i+1]=-1.;
```

```
        }
for(i=0;i<ncoils;i++) R[ncoils-1][i]=1.;
Invert(R,ncoils);
Times(R,T,V,ncoils);

for(i=0;i<ncoils;i++)
        for(j=0;j<ncoils;j++)
        {
                if(i!=j)
                {
                        R[i][j]=0.;
                        T[i][j]=0.;
                }
                else
                {
                        R[i][i]=cos(an[i])/2.;
                        T[i][i]=sin(an[i])/2.;
                }
                Vt[j][i]=V[i][j];
        }

/* form Mx matrix */
Times(R,V,My,ncoils);
Times(Vt,My,Mx,ncoils);

/* form My matrix */
Times(T,V,R,ncoils);
Times(Vt,R,My,ncoils);

Invert(P,ncoils);

/* form backiron flux matrix */
for(i=0;i<ncoils;i++)
        for(j=0;j<ncoils;j++)
                if (i!=ncoils-1) R[i][j]=0.;
                else R[i][j]=1.;
for(i=0;i<(ncoils-1);i++)
{
        R[i][i]=0.5;
        R[i][i+1]=-0.5;
}
Invert(R,ncoils);
for(i=0;i<ncoils;i++)
        for(j=0;j<ncoils;j++)
                if((i!=j)) Vt[i][j]=0.;
                else Vt[i][j]=1.;
Vt[ncoils-1][ncoils-1]=0.;
Times(R,Vt,T,ncoils);
Times(T,P,R,ncoils);
Times(R,V,Vt,ncoils);
/* backiron flux density in Vt */

if (VERB==TRUE)
{
        printf("\nMx matrix:\n");
```

```
				for(i=0;i<ncoils;i++)
				{
					for(j=0;j<ncoils;j++)
						printf("%f ",Mx[i][j]);
					printf("\n");
				}

				printf("\nMy matrix:\n");
				for(i=0;i<ncoils;i++)
				{
					for(j=0;j<ncoils;j++)
						printf("%f ",My[i][j]);
					printf("\n");
				}
			}

			/* cleanup */
			*x=Mx;
			*y=My;
			*p=P;
			*v=V;
			*vb=Vt;
			*Ncoils=ncoils;
			*Dim=dim;
			MatrixFree(R,ncoils);
			MatrixFree(T,ncoils);
			free(an);
		}
		else{
			fclose(fp);
			errflag=TRUE;
		}
		return errflag;
}

void MakeH(H,s,I,E,S,Mx,My,dim)
		double **H,*s,**I,**S,**Mx,**My;
		int dim;
		struct Entry **E;
{
		int i,j,k;

		/* make H matrix */
		for(i=0;i<12;i++)
		{
			for(j=0;j<3*dim;j++) s[j]=0.0;
			for(j=0;j<3;j++)
			{
				if(E[i][j].co==TRUE)
				{
					if(E[i][j].ma==0)
					  vTimes(Mx,I[E[i][j].ve],S[j],dim);
					else
					  vTimes(My,I[E[i][j].ve],S[j],dim);
				}
```

```
                }
                for(j=0;j<3*dim;j++) H[i][j]=s[j];
        }
}

main(argc,argv)
        int argc;
        char *argv[];
{
        int i,j,k,count;        /* iterators */
        int GradStep,Done;
        long steps;
        int dim,ncoils,errflag=0;
        double u,cost,z,zeta,bestcost,lastcost,dt;
        double *Io,*In,*Ip,*s,*cn,*cc,*di,*best;
        double **Mx,**My,**P,**V,**Vb,**H,**HHt,**I,**S,**out,**m;
        struct Entry **E;
        FILE *fp;

        /* note --

                Mx,My correspond to the M_x,M_y matrices of theory
                permuted so that active coils are in the upper left
                dim x dim block.

                P is the matrix by which Mx and My were permuted,
                necessary so that output can be un-permuted for
                printout. */

        srand((int)time(0));
        if (argc!=2) errflag=1;
        if (errflag==0)
        {
                if(ReadEm(argv[1],&Mx,&My,&P,&V,&Vb,&ncoils,&dim)==TRUE)
                        errflag=2;
        }
        if (errflag==0)
        {
                if((fp=fopen("matset.dat","rt"))!=NULL)
                {
                        E=(struct Entry **)calloc(12,sizeof(struct Entry *));
                        for(i=0;i<12;i++)
                        {
                                E[i]=(struct Entry *)calloc(3,
                                      sizeof(struct Entry));
                                for(j=0;j<3;j++)
                                {
                                        fscanf(fp,"%i",&E[i][j].co);
                                        fscanf(fp,"%i", &E[i][j].ve);
                                        fscanf(fp,"%i", &E[i][j].ma);
                                }
                        }
                        fclose(fp);
                }
                else errflag=3;
```

```
}
if (errflag==0)
{
        /* Allocate matrices needed */
        Io=(double *)calloc(3*dim,sizeof(double));
        In=(double *)calloc(3*dim,sizeof(double));
        Ip=(double *)calloc(3*dim,sizeof(double));
        best=(double *)calloc(3*dim,sizeof(double));
        I =(double **)calloc(3,sizeof(double *));
        S =(double **)calloc(3,sizeof(double *));
        I[0]=In; I[1]=In+dim; I[2]=In+2*dim;
        s =(double *)calloc(3*dim,sizeof(double));
        S[0]=s; S[1]=s+dim; S[2]=s+2*dim;
        H=MatrixAlloc(12,3*dim);
        HHt=MatrixAlloc(12,12);
        m=MatrixAlloc(3*dim,3*dim);
        out=MatrixAlloc(4,ncoils);
        cn=(double *)calloc(12,sizeof(double));
        cc=(double *)calloc(12,sizeof(double));
        di=(double *)calloc(3*dim-12,sizeof(double));
        count=0;

    do{
        /* make an initial random guess */
        for(i=0;i<3*dim;i++){
                if (count<HOW_MANY) In[i]=Random();
                else In[i]=best[i];
        }

        /* update currents vector */
        steps=0;GradStep=FALSE;Done=FALSE;dt=DT;lastcost=pow(10.,10.);
        do{
                MakeH(H,s,I,E,S,Mx,My,dim);
                if(GradStep==TRUE){

                        /* get a basis for the feasible space */
                        for(i=0;i<3*dim;i++){
                                for(j=0;j<3*dim;j++){
                                        if (i<12) m[i][j]=H[i][j];
                                        else m[i][j]=Random();
                                }
                        }
                        Gramm(m,3*dim,3*dim);

                        /* compute a numerical derivative associated
                           with each direction of feasible space */
                        for(i=12,j=0;i<3*dim;i++,j++){
                                for(k=0;k<3*dim;k++)
                                    Ip[k]=In[k]+Del*m[i][k];
                                di[j] =IRule(Ip,V,Vb,ncoils,dim,&zeta,out);
                                for(k=0;k<3*dim;k++)
                                    Ip[k]=In[k]-Del*m[i][k];
                                di[j]-=IRule(Ip,V,Vb,ncoils,dim,&zeta,out);
                                di[j]=di[j]/(2.*Del);
                        }
```

```
        /* scale the derivative vector */
        for(j=0,zeta=0.;j<3*dim-12;j++)
                zeta+=di[j]*di[j];
        zeta=sqrt(zeta);
        for(j=0;j<3*dim-12;j++)
                di[j]=GradStepLength*di[j]/zeta;

        /* add gradient step to In */
        for(i=12,k=0;i<3*dim;i++,k++)
                for(j=0;j<3*dim;j++)
                        In[j]-=di[k]*m[i][j];

        /* Make H matrix for the new In */
        MakeH(H,s,I,E,S,Mx,My,dim);
        GradStep=FALSE;
}

/* make H . H(transpose) matrix */
for(i=0;i<12;i++)
{
        for(j=0;j<12;j++)
        {
                z=0.;
                for(k=0;k<3*dim;k++)
                        z+=H[i][k]*H[j][k];
                HHt[i][j]=z;
        }
}
Invert(HHt,12);

/* make RHS */
for(i=0;i<12;i++)
{
        cn[i]=0.;
        for(j=0;j<3*dim;j++)
                cn[i]+=H[i][j]*In[j];
        if(i>9) cn[i]-= 1.;
}

vTimes(HHt,cn,cc,12);

/* figure new current */
for(i=0;i<3*dim;i++)
{
        Io[i]=In[i];
        for(j=0;j<12;j++)
                In[i]-=0.5*dt*H[j][i]*cc[j];
}

/* figure how convergence is doing */
z=0;
for(i=0;i<12;i++)
{
        cn[i]=0.;
```

```
                for(j=0;j<3*dim;j++)
                        cn[i]+=H[i][j]*In[j];
                if(i>9) cn[i]-=1.0;
                z+=cn[i]*cn[i];
        }
        z=sqrt(z);
        if (z<pow(10.,-8.)){
                if (count!=(HOW_MANY)) Done=TRUE;
                else{
                        cost=IRule(In,V,Vb,ncoils,dim,&zeta,out);
                        if(VERB==TRUE)
                         printf("f_max = %f\n",pow(cost,-2.));
                        GradStep=TRUE;
                        dt=1.;
                        if (cost>lastcost) Done=TRUE;
                        lastcost=cost;
                        steps=0;
                }
        }
        if (VERB==TRUE) printf("convergence %e\n",z);
        steps++;
} while((Done==FALSE) && (steps<MAXSTEPS));

if (steps<MAXSTEPS)
{
        for(i=0;i<ncoils;i++)
                for(j=0;j<3;j++)
                        if(i<dim) out[j][i]=I[j][i];
                        else out[j][i]=0.;
        cost=Rule(out,V,Vb,ncoils,&zeta);
        if ((cost<bestcost) || (count==0))
        {
                for(i=0;i<3*dim;i++)
                        best[i]=In[i];
                bestcost=cost;
                printf("%i        f_max = %f\n",count,
                       pow(bestcost,-2.));
        }
}
else count--;

}while(HOW_MANY>count++);

cost=Rule(out,V,Vb,ncoils,&zeta);
printf("\nf_max = %f\n",pow(cost,-2.));
for(i=0;i<3;i++){
        vTimes(P,out[i],out[3],ncoils);
        Cp(out[i],out[3],ncoils);
}

printf("{");
for(i=0;i<ncoils;i++){
/* output is scaled so that the best \hat{i}_o is 1
   and saturation occurs at f=f_max        */
printf("{%f,        %f,      %f}",
```

```
                        out[0][i]*zeta/cost,cost*out[1][i]/zeta,
                        cost*out[2][i]/zeta);
                if (i<(ncoils-1)) printf(",\n");
                }
                printf("}\n");
        }

        /* catch any errors */
        if (errflag==1){
                printf("%s - finds control currents for a magnetic\n",argv[0]);
                printf("       bearing specified in a given datafile:\n\n");
                printf("%s [datafile]\n",argv[0]);
        }
        if (errflag==2) printf("File Not Found.\n");
        if (errflag==3) printf("Data file \"matset.dat\" not found.\n");

}
```

A.2.2 Mathematics library *mathstuf.c*

This library contains various subroutines that facilitate the creation of and operation on matrices in C.

```
#define Pi 3.141592653589793
#define TRUE 1
#define FALSE 0

void MatrixFree();
void Times();
void vTimes();
double Dot();
void TransposeTimes();
char Solve();
char Invert();
double **MatrixAlloc();
double Random();
void MatrixClear();
void PlusMat();
void CpMat();
void Cp();
void Plus();
void Scale();

double Random()
{
        /* returns a random number between -1 and 1 */

        double x;
        x=(double)rand();
        return (2.*x/((double) RAND_MAX) - 1.);
}

void MatrixFree(M,n)
        double **M;
        int n;
{
        /* frees up a matrix allocated with MatrixAlloc */
        int i;

        for(i=0;i<n;i++) free(M[i]);
        free(M);
}

void MatrixClear(M,n)
        double **M;
        int n;
{
        /* fills up a square matrix with zeros in every entry */
        int i,j;

        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                        M[i][j]=0.;
}
```

```
void Times(a,b,c,n)
        double **a,**b,**c;
        int n;
{
        /* multiplies together 2 square matrices a and b, puts result in c */
        int i,j,k;

        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                {
                        c[i][j]=0.;
                        for(k=0;k<n;k++) c[i][j]+=a[i][k]*b[k][j];
                }
}

void PlusMat(out,in,scalar,n)
        double **out,**in;
        double scalar;
        int n;
{
        /* adds matrix ''in'' times the scalar ''scalar'' to the matrix
           ''out''.  The result is returned in ''out'' */
        int i,j;

        for(i=0;i<n;i++)
                for (j=0;j<n;j++)
                        out[i][j]=out[i][j]+scalar*in[i][j];
}

void CpMat(out,in,n)
        double **out,**in;
        int n;
{
        /* copies square matrix in ''in'' into ''out'' */
        int i,j;

        for(i=0;i<n;i++)
                for (j=0;j<n;j++)
                        out[i][j]=in[i][j];
}

double Dot(b,c,dim)
        double *b,*c;
        int dim;
{
        /* returns the dot product of vectors b and c */

        int i;
        double res=0.;

        for(i=0;i<dim;i++) res+=b[i]*c[i];
        return res;
}

void Cp(b,c,dim)
```

```
        double *b,*c;
        int dim;
{
        /* copies entries from b into c */

        int i;

        for(i=0;i<dim;i++) b[i]=c[i];
}

void Plus(z,b,c,dim)
        double *b,*c, *z;
        int dim;
{
        /* adds vectors b and c, puts the result in z */

        int i;

        for(i=0;i<dim;i++) z[i]=b[i]+c[i];
}

void Scale(z,s,dim)
        double *z,s;
        int dim;
{
        /* scales every entry in z by the constant s */

        int i;

        for(i=0;i<dim;i++) z[i]=s*z[i];
}


void vTimes(M,x,b,dim)
        double **M,*x,*b;
        int dim;
{
        /* multiplies vector x times square matrix M.  Result in b */

        int i,j;

        for(i=0;i<dim;i++)
        {
                b[i]=0;
                for(j=0;j<dim;j++) b[i]+=(M[i][j] * x[j]);
        }
}

void TransposeTimes(a,b,c,n)
        double **a,**b,**c;
        int n;
{
        /* multiplies Transpose[a] and b, puts the result in c */
        int i,j,k;
```

```
        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                {
                        c[i][j]=0.;
                        for(k=0;k<n;k++) c[i][j]+=a[k][i]*b[k][j];
                }
}

double **MatrixAlloc(rows,cols)
        int rows,cols;
{
        /* allocates a matrix with dimensions specified by rows and cols */

        double **matrix;
        int i;

        matrix=(double **)calloc(rows,sizeof(matrix));
        if (matrix==NULL) return NULL;
        for(i=0;i<rows;i++){
                matrix[i]=(double *)calloc(cols,sizeof(double));
                if (matrix[i]==NULL) return NULL;
        }
        return matrix;
}

char Solve(m,b,dim)
        double **m,*b;
        int dim;
{
        /* solves the linear system m x = b for x.  The result is returned
           in b, m is destroyed in the process */

        int i,j,k;
        double *z;
        double max,f;
        int n;

        for(i=0;i<dim;i++)
        {
                for(j=i,max=0;j<dim;j++)
                        if (fabs(m[j][i])>fabs(max))
                        {
                                max=m[j][i];
                                n=j;
                        }
                if(max==0) return FALSE;
                z=m[i];m[i]=m[n];m[n]=z;
                f=b[i];b[i]=b[n];b[n]=f;
                for(j=i+1;j<dim;j++)
                {
                        f=m[j][i]/m[i][i];
                        b[j]=b[j]-f*b[i];
                        for (k=i;k<dim;k++)
                                m[j][k]-=(f*m[i][k]);
                }
```

```
        }
        for(i=dim-1;i>=0;i--)
        {
                for(j=dim-1,f=0;j>i;j--)
                        f+=m[i][j]*b[j];
                b[i]=(b[i]-f)/m[i][i];
        }
        return TRUE;
}

char Invert(double **m, int dim)
{
        /* replaces square matrix m with its inverse */
        int i,j,k;
        double **x;
        double *z;
        double max,f;
        int n;

        x=MatrixAlloc(dim,2*dim);
        for(i=0;i<dim;i++)
        {
                for(j=0;j<dim;j++) x[i][j]=m[i][j];
                x[i][dim+i]=1.0;
        }

        for(i=0;i<dim;i++)
        {
                for(j=i,n=i,max=0;j<dim;j++)
                        if (fabs(x[j][i])>fabs(max))
                        {
                                max=x[j][i];
                                n=j;
                        }
                if(max==0) return TRUE;
                z=x[i];x[i]=x[n];x[n]=z;
                for(j=i;j<2*dim;j++) x[i][j]=x[i][j]/max;
                for(j=0;j<dim;j++)
                {
                        if (j!=i)
                        {
                                f=x[j][i];
                                for (k=i;k<2*dim;k++)
                                        x[j][k]=x[j][k]-f*x[i][k];
                        }
                }
        }
        for(i=0;i<dim;i++) for(j=0;j<dim;j++) m[i][j]=x[i][dim+j];
        MatrixFree(x,dim);
        return FALSE;
}

int Gramm(m,row,col)
        double **m;
        int row, col;
```

```
{
        /* does a Gramm-Schmidt orthonormalization of the rows of m */

        int i,j,k;
        int flag=FALSE;
        double x;

        for(k=0;k<row;k++){
                for(j=0,x=0.;j<col;j++) x+=(m[k][j]*m[k][j]);
                x=sqrt(x);
                for(j=0;j<col;j++) m[k][j]=m[k][j]/x;

                for(i=0;i<k;i++){
                        for(j=0,x=0.;j<col;j++) x+=m[k][j]*m[i][j];
                        for(j=0;j<col;j++) m[k][j]-=(x*m[i][j]);
                }

                for(j=0,x=0.;j<col;j++) x+=(m[k][j]*m[k][j]);
                x=sqrt(x);

                if(x<pow(10.,-12)){
                        flag=TRUE;
                        for(j=0;j<col;j++) m[k][j]=0.;
                }
                else for(j=0;j<col;j++) m[k][j]=m[k][j]/x;
        }
        return flag;
}
```

A.2.3 Sample bearing data file *brg.dat*

This file specifies the location of each pole and which poles are active. The first column corresponds to the angle of the center of each pole as measured in degrees from the “X” axis as defined in Figure A.1. The second column denotes which coils have active poles. A “1” denotes that the coil on the corresponding pole is active, and a “0” that the coil is inactive. This particular data file corresponds to Case 11.

```
22.5       0
67.5       1
112.5      0
157.5      1
202.5      1
247.5      0
292.5      1
337.5      1
```

A.2.4 Data file *matset.dat* needed by *findw.c*

This file contains information that `findw` needs to build the `H` matrix defined in eq. (6.19). This file should not be modified.

```
1  0  0  0  0  0  0  0  0
1  0  1  0  0  0  0  0  0
0  0  0  1  1  0  0  0  0
0  0  0  1  1  1  0  0  0
0  0  0  0  0  0  1  2  0
0  0  0  0  0  0  1  2  1
0  0  0  1  2  0  1  1  0
0  0  0  1  2  1  1  1  1
1  2  0  0  0  0  1  0  0
1  1  1  1  0  1  0  0  0
1  2  1  0  0  0  1  0  1
1  1  0  1  0  0  0  0  0
```

A.2.5 Sample run of *findw.c*

As a demonstration, the program was run using the previously presented file `brg.dat` as its input. Each time the program finds a new best solution, it reports the iteration number, and the f_{max} produced by that solution. In this case, 1000 feasible solutions were found starting from random seeds. In the 1000^{th} iteration, the best feasible solution is improved through the application of the reduced gradient method. When further gradient steps yield no improvement, the best solution is reported in a form that can be conveniently imported into *Mathematica*.

This particular execution was run through the `time` command so that the time taken to run the program was reported at the end of the run. This run took about $4\frac{1}{2}$ minutes of time on an IBM RS6000. A listing of the run follows.

```
romac2: /home/dcm3c/diss $ time findw brg.dat
0       f_max = 0.078109
1       f_max = 0.090939
6       f_max = 0.164693
24      f_max = 0.183287
33      f_max = 0.213848
36      f_max = 0.228720
49      f_max = 0.232852
230     f_max = 0.238559
274     f_max = 0.263614
812     f_max = 0.276421
1000    f_max = 0.290706

f_max = 0.290706
{{0.000000,     0.000000,      0.000000},
{-0.642233,     -0.019391,      -1.541162},
{0.000000,      0.000000,      0.000000},
{-0.530964,     2.098009,      -1.183733},
{0.078041,      -0.261275,      -1.308100},
{0.000000,      0.000000,      0.000000},
{-0.653558,     0.598380,      0.436182},
{-0.055513,     2.256527,      -0.862093}}

real    5m11.89s
user    4m36.06s
sys     0m0.07s
```

A.3 *Mathematica* program for solution of direct optimization problem

This program finds the solution to the direct optimization problem for the specific case of a non-dimensionalized 8-pole radial bearing with all coils active. Variables `m1` and `m2` contain matrices M_x and M_y respectively. The specific i_o to be used in the solution is specified by the variable `io`.

After the M matrices and i_o are defined, the program integrates the system of differential equations (7.6) at 40 different angles spaced evenly between 0 and 2π. An Euler integration is performed from $\underline{f} = 0$ to $\underline{f} = 1$ with a step size of 0.02.

The currents required for coil "1" are stored as an array of points contained in `data`. After all integrations are done, the information in `data` is used to form a graphical representation of the solution surface. The graphics primitives that define the surface are contained in the variable `li`. The last statement of the program plots the resulting surface with the appropriate labeling.

```
Rr={{1,-1,0,0,0,0,0,0},
    {0,1,-1,0,0,0,0,0},
    {0,0,1,-1,0,0,0,0},
    {0,0,0,1,-1,0,0,0},
    {0,0,0,0,1,-1,0,0},
    {0,0,0,0,0,1,-1,0},
    {0,0,0,0,0,0,1,-1},
    {1,1,1,1,1,1,1,1}};
Nn={{1,-1,0,0,0,0,0,0},
    {0,1,-1,0,0,0,0,0},
    {0,0,1,-1,0,0,0,0},
    {0,0,0,1,-1,0,0,0},
    {0,0,0,0,1,-1,0,0},
    {0,0,0,0,0,1,-1,0},
    {0,0,0,0,0,0,1,-1},
    {0,0,0,0,0,0,0,0}};
V=Inverse[Rr] . Nn;
m1=Chop[N[Transpose[V].DiagonalMatrix[
        (1/2)*Table[Cos[Pi(n/4 + 1/8)],{n,0,7}]].V]];
m2=Chop[N[Transpose[V].DiagonalMatrix[
        (1/2)*Table[Sin[Pi(n/4 + 1/8)],{n,0,7}]].V]];
io=0.25*{1.,-1.,1.,-1.,1.,-1.,1.,-1.};
d=8;

data=Table[,{41}];
co=0;
For[t=0,N[t]<=N[2 Pi],t+=Pi/20,
        ds=1/50;
        q=Table[,{1/ds+1}];k=0;
        x=io; z1=0.; z2=0.;
        For[j=0,j<=1,j+=ds,
                A=IdentityMatrix[d]+2(z1 m1 + z2 m2);
                Ap=2*{m1 . x,m2 . x};
                A=Transpose[Join[A,Ap]];
                Ap[[1]]=Join[Ap[[1]],{0,0}];
                Ap[[2]]=Join[Ap[[2]],{0,0}];
                A=Join[A,Ap];
                b=Join[Table[0,{d}],{N[Cos[t]],N[Sin[t]]}];
                ans=Inverse[A].b;
                k++;
                (* q[[k]]=Join[{N[j]},x,{z1,z2}]; *)
```

```
                q[[k]]={N[j Cos[t]],N[j Sin[t]],x[[1]]};
                x+=(N[ds]*ans[[Range[1,d]]]);
                z1+=(N[ds]*ans[[d+1]]);
                z2+=(N[ds]*ans[[d+2]])
        ];
        co++;
        data[[co]]=q;
        Print[N[t]]
]

li=Table[,{40*50}];k=0;
For[i=1,i<41,i++,
        For[j=1,j<51,j++,
                k++;
                li[[k]]=Polygon[{data[[i,j]],data[[i+1,j]],
                                data[[i+1,j+1]],data[[i,j+1]]}]
        ];
        Print[i];
]

Show[Graphics3D[{EdgeForm[{Thickness[0.001]}],li}],
     FaceGrids->{{0,0,-1},{1,0,0},{0,1,0}},
     AxesLabel->{f1,f2,i1},BoxRatios->{1,1,1},Axes->True,
     ViewPoint->{-1,-6,0.2},Lighting->True,AxesEdge->{{-1,-1},
     Automatic,Automatic}]
```

A.4 C code for saturating 8-pole bearing

This C code is for the specific case of an 8-pole nondimensionalized radial magnetic bearing subject to saturation. All parameters are hard-coded into the source so no external data file is necessary. These parameters are:

ThetaDivs Defines the number of angles at which the system of equations specified by (9.25) will be integrated.

MaxForce Denotes the limit of integration to which the force will be integrated.

ForceDivs Specifies the size of each step in the integration by specifying the number of steps between 0 and `MaxForce`.

BiasLevel scales the vector $i_o = \{1,1,1,1,1,1,1,1\}^T$ so that solutions can be found for various bias levels.

The output of this program is a set of *Mathematica* graphics primitives that can be plotted to produce a portrait of the required currents analogous to the plot created by the previous *Mathematica* program.

The algorithm employed by this program is explained in detail in § 9.4. Specific subroutines used in the program are

MakeA Forms the matrix on the left-hand side of (9.25).

EvalF Evaluates the Kuhn-Tucker conditions. This evaluation is used to take Newton-Raphson steps to make sure that the Kuhn-Tucker conditions are satisfied.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define ThetaDivs 40
#define MaxForce  1.00
#define ForceDivs 50
#define BiasLevel 0.25

#include"mathstuf.c"

void MakeA(A,s,t,M1,M2,Ca,x,a)
        double **A,s,t,**M1,**M2,**Ca,*x;
        int a;
{
        int i,j,k,p,q;
        double c;

        MatrixClear(A,16);

        for(p=0;p<8;p++){
                for(q=0;q<8;q++){
                        if (p==q) c=1.; else c=0.;
                        A[p][q]=2*(c + x[8]*M1[p][q]
                                     + x[9]*M2[p][q]);
                        A[8][p]+=2*M1[p][q]*x[q];
                        A[9][p]+=2*M2[p][q]*x[q];
                }
                A[p][8]=A[8][p]; A[p][9]=A[9][p];
        }
```

```
        for(i=0,k=10;i<a;i++,k++)
                for(j=0;j<8;j++){
                        A[k][j]=Ca[i][j];
                        A[j][k]=Ca[i][j];
                }
}

void EvalF(b,s,t,x,M1,M2,Ca,na,io)
        double *b,s,t,*x,**M1,**M2,**Ca,*io;
        int na;
/* puts evaluated K-T conditions into b */
{
        int i,j,k;
        double f1,f2;

        for(k=0;k<16;k++) b[k]=0.;

        for(i=0;i<8;i++) b[i]=2.*(x[i] - BiasLevel*io[i]);

        for(f1=0.,f2=0.,i=0;i<8;i++){
                for(j=0;j<8;j++){
                        f1+=x[i]*M1[i][j]*x[j];
                        f2+=x[i]*M2[i][j]*x[j];
                        b[i]+=2.*(x[8]*M1[i][j]*x[j] + x[9]*M2[i][j]*x[j]);
                }
        }

        for(i=0;i<8;i++)
                for(j=0;j<na;j++)
                        b[i]+=(Ca[j][i]*x[10+j]);

        for(i=0;i<na;i++){
                for(j=0,b[10+i]=0.;j<8;j++) b[10+i]+=Ca[i][j]*x[j];
                b[10+i]-=1.;
        }

        b[8]=f1-s*cos(t); b[9]=f2-s*sin(t);
}


main()
{
        int i,j,k,p,q,flag;
        int na,np,nc;
        double **A,**V,**Ca,**Cp,**G,**H,**M1,**M2,**R,**N,**Q,**d;
        double io[8]={1.,-1.,1.,-1.,1.,-1.,1.,-1.};
        double t,s,dt,ds,c,maxb,x[16],b[16];

        A =MatrixAlloc(16,16);
        V =MatrixAlloc(8,8);
        R =MatrixAlloc(8,8);
        N =MatrixAlloc(8,8);
        M1=MatrixAlloc(8,8);
        M2=MatrixAlloc(8,8);
        Q =MatrixAlloc(8,8);
```

```
G =MatrixAlloc(2,8);
H =MatrixAlloc(6,8);

Cp=MatrixAlloc(16,8);
Ca=(double **)calloc(16,sizeof(double *));
np=16; na=0; nc=16;

for(i=0;i<7;i++){
        R[i][i]=1.; R[i][i+1]=-1.;
        N[i][i]=1.; N[i][i+1]=-1.;
}
for(i=0;i<8;i++){
        R[7][i]=1.;
        M1[i][i]=0.5*cos(Pi*(((double) i)/4. + 0.125));
        M2[i][i]=0.5*sin(Pi*(((double) i)/4. + 0.125));
}

Invert(R,8);
Times(R,N,V,8);
Times(M1,V,Q,8);
TransposeTimes(Q,V,M1,8);
Times(M2,V,Q,8);
TransposeTimes(Q,V,M2,8);

for(i=0;i<8;i++)
        for(j=0;j<8;j++){
                Cp[i][j]  =V[i][j];
                Cp[8+i][j]=-V[i][j];
        }

dt=2.*Pi/((double) ThetaDivs);
ds=MaxForce/((double) ForceDivs);
d=MatrixAlloc(ThetaDivs+1,ForceDivs+1);

for(i=0,t=0.;i<=ThetaDivs;i++,t+=dt){

        for(j=0;j<16;j++){
                if(j<8){
                        x[j]=BiasLevel*io[j];
                }
                else x[j]=0.;
        }

        for(j=0;j<na;j++){
                Cp[np]=Ca[j];
                np++;
        }
        na=0;

        for(j=0,s=0.;j<ForceDivs;j++,s+=ds){
                d[i][j]=x[0];

                /* make b for prediction step */
                for(k=0;k<16;k++) b[k]=0.;
                b[8]=cos(t); b[9]=sin(t);
```

```
                /* compute prediction step */
                MakeA(A,s,t,M1,M2,Ca,x,na);
                Solve(A,b,10+na);
                for(k=0;k<16;k++) x[k]+=ds*b[k];

                /* do newton cleanup step */
                MakeA(A,s,t,M1,M2,Ca,x,na);
                EvalF(b,s,t,x,M1,M2,Ca,na,io);
                Solve(A,b,10+na);
                for(p=0;p<10+na;p++)
                        x[p]=x[p]-b[p];

                /* check for constraint violation */
                flag=FALSE;
                for(p=0;p<np;p++){
                        for(k=0,c=0.;k<8;k++) c+=Cp[p][k]*x[k];
                        if (c>=1.){
                                flag=TRUE;
                        /*        printf("*");        */
                                Ca[na]=Cp[p];
                                x[10+na]=0.;
                                na++; np--;
                                for(k=p;k<np;k++) Cp[k]=Cp[k+1];
                        }
                }

                /* if new constraints have been imposed,
                   clean up with a newton step to meet constraint
                   exactly. */

                if(flag==TRUE){
                        MakeA(A,s,t,M1,M2,Ca,x,na);
                        EvalF(b,s,t,x,M1,M2,Ca,na,io);
                        Solve(A,b,10+na);
                        for(p=0;p<10+na;p++)
                                x[p]=x[p]-b[p];
                }
        }
        d[i][j]=x[0];
}

printf("li={");
for(i=0,t=0;i<ThetaDivs;i++,t+=dt){
        for(j=0,s=0;j<ForceDivs;j++,s+=ds){
              printf(
              "Polygon[{{%f,%f,%f},{%f,%f,%f},{%f,%f,%f},{%f,%f,%f}}]",
                       s*cos(t),s*sin(t),d[i][j],
                       s*cos(t+dt),s*sin(t+dt),d[i+1][j],
                       (s+ds)*cos(t+dt),(s+ds)*sin(t+dt),d[i+1][j+1],
                       (s+ds)*cos(t),(s+ds)*sin(t),d[i][j+1]);
                if((i==ThetaDivs-1) && (j==ForceDivs-1)) printf("}\n");
                else printf(",\n");
        }
}
```

```
}
```

Appendix B

Slew rate limiting.

B.1 Bounds on current slew rate.

In this dissertation, it is assumed that any requested set of currents is realized virtually instantaneously. However, some type of amplifier, in conjunction with a very fast feedback loop, is actually used to obtain the requested currents in a short but not infinitesimal time. Practical limitations of magnetic bearings are addressed in detail in [MHSH89]. The goal of this appendix is to provide a brief synopsis of the limitations due to amplifiers so that control currents can be chosen in a way so that nominal bearing performance is not affected.

The amplifier realizes the requested currents by controlling the voltage drop across each coil. The current flowing in each coil is sensed and used by the amplifier control scheme to choose voltages that drive the coil currents to the desired values.

Two types of amplifier are generally used in magnetic bearing applications: linear and switching amplifiers. Linear amplifiers can create a voltage anywhere between two extreme voltages, v_o and $-v_o$. Switching amplifiers, however, alternate only between the two extreme voltages and spend almost no time at intermediate voltages. Switching amplifiers are used predominantly due to their low cost and high efficiency as compared to linear amplifiers.

Regardless of amplifier type, the finite maximum amplifier voltage limits the currents in the coils to a finite time rate of change. The rate of current change in a coil is known as "slew rate." For any bearing, the electrical circuit equations are:

$$L\frac{di}{dt} + Ri = v \tag{B.1}$$

where L is the inductance matrix for the bearing as defined in (3.7), R is a diagonal matrix of coil resistances, and v is a vector of amplifier voltages. Typically, the resistance of the coils is small in comparison to the impedance due to the coils' inductance; it is then appropriate to approximate:

$$L\frac{di}{dt} = v \tag{B.2}$$

It is useful to derive one number as a limiting slew rate. In the case of a single horseshoe bearing, the limiting slew rate is trivial to compute. In this case, i, L, and v are all scalars; by inspection of (B.1), the largest possible slew rate magnitude is

$$\frac{di}{dt}_{max} = \frac{v_o}{L} \tag{B.3}$$

For a system with cross-coupling terms in the inductance matrix (such as occurs with an 8-pole radial bearing with all coils driven independently), a similar value can be derived.

Taking the 2-norm of each side of (B.1) yields

$$\begin{aligned} \left\| L\frac{di}{dt} \right\|_2 &= \|v\|_2 \\ \bar{\sigma}[L]\left\| \frac{di}{dt} \right\|_2 &\geq \|v\|_2 \\ \left\| \frac{di}{dt} \right\|_2 &\geq \frac{\|v\|_2}{\bar{\sigma}[L]} \end{aligned}$$

It can be noted that $\|v\|_\infty \leq \|v\|_2$:

$$\left\| \frac{di}{dt} \right\|_2 \geq \frac{\|v\|_\infty}{\bar{\sigma}[L]}$$

For the limiting case, $\|v\|_\infty = v_o$.

$$\left\| \frac{di}{dt} \right\|_2 \geq \frac{v_o}{\bar{\sigma}[L]}$$

A sufficient condition for a requested slew rate to be realizable is then

$$\left\| \frac{di}{dt} \right\|_2 \leq \frac{v_o}{\bar{\sigma}[L]} \tag{B.4}$$

Slew rate limiting is the condition where the requested slew rate is greater than can be realized. One scenario in which this problem can occur is at low force levels. Consider

$$f_j = i' M_j i \,;\; j = 1, \ldots k$$

The derivative of force with respect to time is

$$\frac{df}{dt} = 2i' M_j \frac{di}{dt} \tag{B.5}$$

If zero force is realized by $i = 0$, $df/dt = 0$ regardless of di/dt. An infinite current slew rate is required to get a finite df/dt, and slew rate limiting is inevitable. Slew rate limiting can be avoided at low force levels by requiring non-zero currents at zero force such that $(B.5)$ is a set of k independent linear equations in di/dt.

Another potential cause of slew rate limiting is a discontinuous inverse mapping. Any discontinuities in $i(t)$ require an infinite di/dt. This problem is avoided by requiring an inverse mapping that is continuous and has finite gradients everywhere.

Another, and more serious, cause of slew rate limiting can arise due to mis-specification of the amplifier switching voltage. If the switching voltage is not high enough, requested di/dt can be outside of the range of realizable values during normal operation. Perhaps the only remedy for this type of slew rate limiting is a change to amplifiers with higher switching voltages.

B.2 Decoupling of electrical circuits in a radial magnetic bearing

In a typical magnetic bearing wound in a horseshoe configuration, the requested currents are realized by a switching amplifier controlled by a fast feedback loop. With horseshoe windings, each pair of coils is magnetically isolated from all other coils; current in horseshoe's winding produces flux only in the legs associated with that horseshoe. This configuration is illustrated in Figure B.1.

However, the same is not the case for a bearing with an independently wound coil on each leg. A current applied to one leg causes flux in all legs of the bearing. Moreover, the inductance matrix L is singular due to conservation of flux constraints; the current vector $i = \{1, \ldots, 1\}$ produces no flux and is therefore associated with a zero inductance. This current vector associated with no flux can create problems if the usual

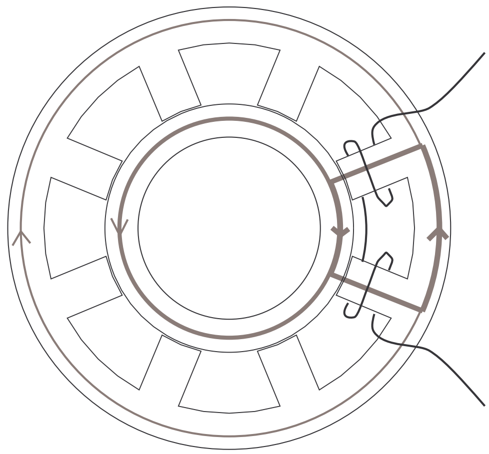


Figure B.1: Flux induced by a horseshoe winding

switching algorithm for realizing currents in horseshoes is naively applied to a bearing with independently wound coils.

For a bearing wound in a horseshoe scheme, the electric circuit equation for an individual horseshoe is

$$l\frac{di}{dt} + ri = v \tag{B.6}$$

The applied voltage, v, is supplied by a switching amplifier and can take on the value of either v_o or $-v_o$. The control law that realizes a desired current can be idealized as

$$\begin{array}{rcl} v = v_o & ; & i < i_d \\ v = -v_o & ; & i > i_d \end{array} \tag{B.7}$$

where i_d is the desired current. This system is known to give good tracking performance as long as $|i_d| < \frac{v_o}{r}$ and $|\frac{di_d}{dt}| < \frac{v_o}{l}$ (restrictions on the achievable magnitude and slew rate of i).

Consider the use of the same switching algorithm on a 4-pole bearing with independently wound coils. The electric circuit equations are:

$$L\frac{di}{dt} + ri = v \tag{B.8}$$

where

$$L = \left(\frac{N^2 a\mu_o}{g_o}\right)\begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix} \tag{B.9}$$

Inductance matrix L is singular; its eigenvalues are $(\frac{N^2 a\mu_o}{g_o})\{1,1,1,0\}$. Since L is singular, there are only 3 states to the system, even though there are four currents and four inputs. Writing this system in standard form via a singular value decomposition yields:

$$\frac{dx}{dt} = -\left(\frac{g_o r}{N^2 a \mu_o}\right)x + \left(\frac{g_o r}{N^2 a \mu_o}\right)\begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} v \tag{B.10}$$

$$i = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{2} \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{2} \\ 0 & -\frac{1}{\sqrt{2}} & -\frac{1}{2} \end{bmatrix} x + \left(\frac{1}{4r}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} v \tag{B.11}$$

Any component of v along $\{1,1,1,1\}$ is fed instantaneously into i, creating a large magnitude step in coil currents. Since half of the possible switching states have a component along $\{1,1,1,1\}$, exciting the zero-inductance vector is unavoidable with switching amplifiers.

In the failed coil case, L is no longer singular and no part of v feeds directly into i, but a similar problem remains. For the four-pole bearing, for example, the eigenvalues of L in the one-coil failed case are $\{1,1,\frac{1}{4}\}$. One eigenvalue is much smaller than the others, implying that one vector of currents changes nearly instantaneously. Though all desired currents may be realizable, excessive chatter will result by exciting the low-inductance current vector.

One solution to these problems is to add extra inductance to the electrical circuit equations associated only with the component of i along the null space of L. The result is that the electrical circuit equations associated with each coil again become decoupled; the same current control scheme used for horseshoes can then be used for the independent coil actuator. To achieve this decoupling, each bearing coil is also attached in series to windings around a laminated slotted ring. Each electric circuit has the same number of turns wound in the same direction around the ring; therefore, flux is only induced in the ring if i has a component along $\{1,1,\ldots,1\}$. Schematically, the arrangement is illustrated in Figure B.2. If the ring is designed so that the self-inductance of the ring for each electric circuit, l_p, has a value of

$$l_p = \frac{1}{n}\left(\frac{a\mu_o N^2}{g_o}\right) \tag{B.12}$$

the negative off-diagonal mutual inductances in L from the bearing are exactly canceled out by the positive mutual inductances from the ring. The electric circuit equations become:

$$\left(\frac{a\mu_o N^2}{g_o}\right)\frac{di}{dt} + ri = v \tag{B.13}$$

Although the inductance matrix has been changed with respect to the electric circuit, the bearing still has all of the coupled magnetic properties that allow low power loss performance and fault-tolerance.

Since the ring only adds inductance along the null vector of L, adding this extra ring does not adversely influence the slew rate limiting properties of the bearing. In addition, with l_p included in the electrical circuit, the slew rate limiting characteristics of the bearing become much easier to assess. In this case, the change in the desired current must obey

$$\|\frac{di}{dt}\|_\infty \leq \frac{v_o g_o}{a\mu_o N^2} \tag{B.14}$$

Note that since each coil is decoupled, this condition is not only sufficient for the avoidance of slew rate limiting but, unlike (B.4), also necessary.

B.2.1 Design of the decoupling ring

To mask the effects of eddy currents, the ring should should have an air gap in the magnetic circuit. In a laminated ring without an air gap, eddy currents can cause a large deviation from the desired inductance at relatively low frequencies [Sto74]. If an air gap is included in the ring, nearly all of the reluctance of the magnetic circuit is due to the air gap; changes in the effective reluctance of the iron sections due to eddy currents can be neglected.

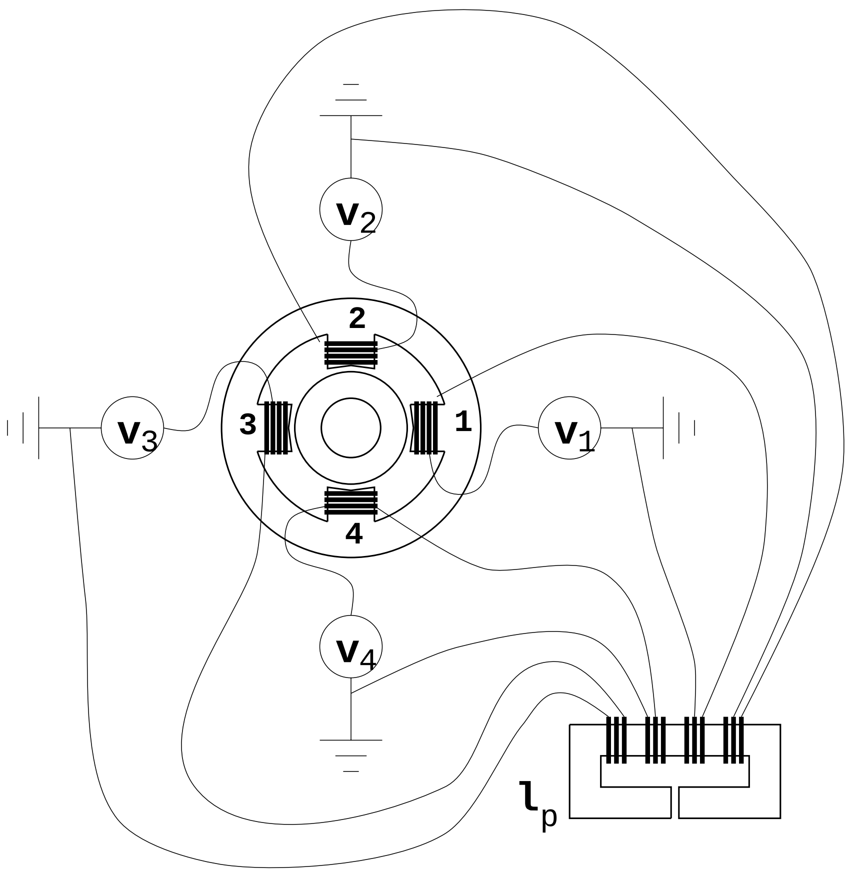


Figure B.2: Circuits including l_p to cancel mutual inductance.

The dimensions to be chosen for the ring are then g_p, the air gap in the ring; a_p, the cross-sectional area of the ring; and N_p, the number of turns from each electric circuit wound around the ring. The first constraint on the choice of these parameters is that (B.12) must be satisfied:

$$\frac{1}{n}\left(\frac{a\mu_o N^2}{g_o}\right) = \left(\frac{a_p\mu_o N_p^2}{g_p}\right) \tag{B.15}$$

If the bearing is only meant to be used in the all-coils-active case, very little flux is expected to be induced in the ring. In this case, the desired currents are generally orthogonal to the null space of L, because currents along this vector produce no force. The only flux produced in the ring is due to switching noise, of low magnitude, and transient in nature. The design parameters can then be chosen solely to minimize the size of the ring while satisfying (B.15). If the bearing is meant to be used in failure configurations as well, additional constraints on the choice of ring dimensions arise. In many failure cases, it is often not possible to have a desired set of currents that produce no flux in the ring. The ring must be sized so that the failure configuration currents do not saturate the ring and cause a premature loss of load capacity.

The flux density in the ring is easily show to be:

$$\left(\frac{N_p\mu_o}{g_p}\right)\{1,\ldots,1\}\,i \tag{B.16}$$

This expression can be non-dimensionalized using (A.1)-(A.4):

$$\underline{b}_{ring} = \left(\frac{N_p}{N}\right)\left(\frac{g_o}{g_p}\right)\{1,\ldots,1\}\,\underline{i} \tag{B.17}$$

For the worst-case orientation for each failure configuration, the ring should not saturate. The worst case is characterized by

$$\underline{i}_{max} = \max_{\underline{W}}\max_{\theta}\{1,\ldots,1\}\underline{W}\begin{Bmatrix} 1 \\ \underline{f}_{max}\sin\theta \\ \underline{f}_{max}\cos\theta \end{Bmatrix} \tag{B.18}$$

Then, the worst-case flux density is

$$\underline{b}_{ring} = \left(\frac{N_p}{N}\right)\left(\frac{g_o}{g_p}\right)\underline{i}_{max} \tag{B.19}$$

At this worst case, $\underline{b}_{ring}$ should be 1, denoting that the ring is bordering upon saturation. Substituting $\underline{b}_{ring} = 1$ into (B.19) gives a second design constraint:

$$\left(\frac{g_p}{g_o}\right) = \left(\frac{N_p}{N}\right)\underline{i}_{max} \tag{B.20}$$

If g_p is to be the arbitrarily chosen parameter, (B.15) and (B.20) can be solved for N_p and a_p in terms of g_p:

$$N_p = \frac{N}{\underline{i}_{max}}\left(\frac{g_p}{g_o}\right) \tag{B.21}$$

$$a_p = \frac{a\,\underline{i}_{max}^2}{n}\left(\frac{g_o}{g_p}\right) \tag{B.22}$$

For the 8-pole symmetric bearing using the failure currents described in Appendix A, $\underline{i}_{max} = 7.7$. It is interesting to note that this current level occurs on the one-coil-failed case. If this case is neglected, the worst case is $\underline{i}_{max} = 4.4$. For the one-pole failure case, it may be better to use the numerically determined current set:

$$\underline{W} = \begin{bmatrix} 0.000000 & 0.000000 & 0.000000 \\ -0.392383 & -0.643138 & -0.801866 \\ 0.284591 & -0.317674 & 0.766932 \\ -0.392383 & 1.021772 & -0.112238 \\ 0.000000 & 0.000000 & 0.000000 \\ 0.392383 & -0.643138 & -0.801866 \\ -0.284591 & -0.317674 & 0.766932 \\ 0.392383 & 1.021772 & -0.112238 \end{bmatrix}$$

The load capacity is reduced to $\underline{f}_{max} = 0.554178$ for a half back iron bearing, but this particular solution has no component along the $\{1,\ldots,1\}$ vector and therefore produces no flux in the ring.

An alternative approach to the ring design is to pick a ring geometry before linearizing currents are determined. Equation (B.19) would then be incorporated into the equation $b = V_s\,i$ used to determine the peak flux density in the bearing for purposes of rating bearing load capacity. Flux density in the ring is treated just like flux density in an section of the stator. Currents are chosen such that flux density in the ring is taken into account when selecting a best linearizing current set.

Once an appropriate g_p, a_p and N_p are selected, the ring still must be designed mechanically. One possible approach is pictured in Figure B.3. The ring is split into two "L"-shaped parts. Coils would be wound on spools and slid onto L's during assembly. The L's would then be separated by non-magnetic shims of thickness $g_p/2$.

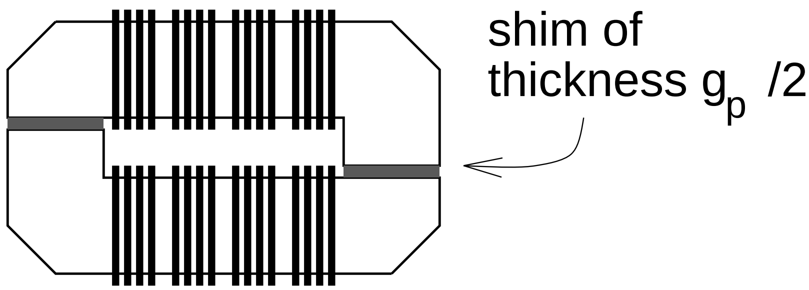


Figure B.3: Schematic of decoupling ring.