

12-31-1991

Computer control for 2-axis magnetic suspension system

Jian Chen
New Jersey Institute of Technology

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Chen, Jian, "Computer control for 2-axis magnetic suspension system" (1991). *Theses*. 2453.
<https://digitalcommons.njit.edu/theses/2453>

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

COMPUTER CONTROL FOR 2-AXIS MAGNETIC SUSPENSION SYSTEM

by

Jian Chen

**Thesis submitted to the Faculty of the Graduate School of
the New Jersey Institute of Technology
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering**

1991

APPROVAL SHEET

Title of Thesis: COMPUTER CONTROL FOR 2-AXIS MAGNETIC
SUSPENSION SYSTEM

Name of Candidate: JIAN CHEN

Master of Science in Electrical Engineering, 1991

Thesis and Abstract Approved

by the Examining Committee: / _____

Dr. Bernard Friedland Date
Distinguished Professor, Thesis Advisor
ECE Department

Dr. Andrew U. Meyer Date
Professor, Coordinator of Control Area
ECE Department

Dr. T. Chang Date
Assistant Professor
ECE Department

Acknowledgement

I would especially like to thank my research advisor, Professor Bernard Friedland, for his efforts put into this research. I would also like to thank the members of NJIT Control Systems Research Laboratory, for the support and ideas offered for this research.

VITA

Name: Jian Chen

Permanent address:

Degree and date to be conferred: M.S.E.E., December 1991

Date of birth:

Place of birth:

Collegiate institutions attended	Dates	Degree	Date of Degree
Zhejiang University	1985-1989	B.S.	July 1989
New Jersey Institute of Technology	1990-1991	M.S.	December 1991

Major: Electrical Engineering

ABSTRACT

Title of Thesis: COMPUTER CONTROL FOR 2-AXIS MAGNETIC
SUSPENSION SYSTEM

Name of Candidate:

Jian Chen
Master of Science
Electrical Engineering Department. 1991

Thesis directed by:

Dr. Bernard Friedland
Distinguished Professor
Department of Electrical Engineering

A two-axis magnetic suspension system, using linear-potentiometer position sensing and DC solenoids, controlled digitally by a 80386 microcomputer has been built and operated in our laboratory. Control algorithms obtained by combining the full-state feedback control law with two methods of estimating velocities are studied. One method uses a nonlinear reduced-order observer; the other uses a finite difference algorithm. The nonlinear reduced-order observer performs better in simulation but the difference algorithm takes less calculation time and is used in the implementation. The system has been operated successfully to suspend weights up to 1.2 lb. and to get a slant angle more than 10 degrees. Stable operation is achieved at sampling rates as low as 100HZ.

Contents

1	INTRODUCTION	1
2	ANALYSIS	7
2.1	Model of 2-Axis Magnetic Suspension System	7
2.2	Algorithm of Full-State Feedback Control Law	10
2.3	Algorithm of Nonlinear, Reduced-Order Observer	14
2.4	The Gain Matrices of Full-State Feedback and Observer . .	18
2.4.1	The Gain Matrix of Observer	18
2.4.2	The Gain Matrix of Full-State Feedback	19
2.5	Additional Consideration on Experimental System	24
2.5.1	Another Way to Estimate Velocities	24
2.5.2	Employ Integral Feedback Control to Achieve High Control Accuracy	24
3	DESIGN AND SIMULATED PERFORMANCE	26
3.1	Basic Character of Magnet	26
3.2	Design Calculations	30

3.3	Numerical Simulations	32
3.3.1	Simulation by Using Observer	33
3.3.2	Comparison of Nonlinear Observer With Difference Algorithm	38
4	IMPLEMENTATION	41
4.1	Hardware	41
4.2	Software	46
4.3	Performance Tests	48
5	CONCLUSIONS & RECOMMENDATIONS	53
A	APPENDICES	58
A.1	Appendix A	58
A.2	Appendix B	62
A.3	Appendix C	68

List of Figures

1.1	2-Axis Magnetic Suspension System Using Computer Control	6
2.1	Model of 2-Axis Magnetic Suspension System	8
2.2	Block diagram of full-state feedback design of 2-axis mag- netic suspension system based on linearization about set point	14
2.3	Block diagram of nonlinear reduced-order observer design of 2-axis magnetic suspension	18
2.4	Block diagram of control algorithm of 2-axis magnetic sus- pension system	23
3.1	Diagram of solenoid test	27
3.2	k_B versus gap y of solenoid	28
3.3	Relationship of i , f and y of solenoid	28
3.4	The inductance of solenoid	29
3.5	A diagram of the suspended object of the experiment	30
3.6	The positions using nonlinear observer when z set (simulated result)	34

3.7	The control currents using nonlinear observer when z set (simulated result)	34
3.8	The estimated and actual velocities using nonlinear observer when z set (simulated result)	35
3.9	Closeup of the estimated and actual velocities using nonlin- ear observer when z set (simulated result)	35
3.10	The positions using nonlinear observer when z not set(simulated result)	36
3.11	The control currents using nonlinear observer when z not set(simulated result)	37
3.12	The estimated and actual velocities using nonlinear observer when z not set (simulated result)	37
3.13	Compare positions when using observer(left) and using dif- ference algorithm(right)	39
3.14	Compare control currents when using observer(left) and us- ing difference algorithm(right)	39
3.15	The estimated velocity(left) and the actual velocity(right) when using observer	40
3.16	The estimated velocity(left) and the actual velocity(right) when using difference algorithm	40
4.1	Experiment in operation	42

4.2	Closeup of suspended object; airgaps in solenoids about 20mm, suspended object total weight 1.16 lb.	42
4.3	A block diagram of the Data Acquisition Adapter	44
4.4	Diagram of power amplifier	45
4.5	The performance of amplifier	45
4.6	Overview flowchart of the software	46
4.7	Timing diagram of the software design	47
4.8	Experimental positions when integral feedback control is present	50
4.9	Simulated positions when integral feedback control is present	50
4.10	Experimental estimated velocities when integral feedback con- trol is present	51
4.11	Simulated estimated velocities when integral feedback con- trol is present	51
4.12	Experimental control currents when integral feedback control is present	52
4.13	Simulated control currents when integral feedback control is present	52

Chapter 1

INTRODUCTION

The phenomenon of levitation has fascinated philosophers through the ages and it has attracted much attention from scientists in recent times as application in scientific instrumentations and a means of eliminating friction or physical contact. Magnetic suspensions act to maintain the position of a body along or about one or more axes attached to a certain reference frame, through the force actions of magnetic fields. Such suspensions generally are statically unstable. Stable action can be achieved only by active feedback control either by sensing the position of the body continuously or at intervals to enable the force fields to be controlled by servo action with sufficient rapidity to prevent the body from departing from its desired position by more than a tolerable amount.

For an active suspension, position may be sensed by a variety of electrical or mechanical means(optical, capacitive, inductive, etc.). With a permanent magnet, or an electromagnet carrying constant current, there

is just one position of the object on the vertical axis at which the magnetic lift force equals the weight, the slightest upward movement causes the object to snatch up to the magnet and the slightest downward movement results in a fall. Therefore, the critical part of the magnetic suspension system is *the feedback compensator*, which takes the position-related signal from the position transducer and suitably processes it to provide a driving signal for the current regulator. The compensator must provide, at least, a signal reversal such that upward movement cause reduced current and downward movement causes increased current, thus always tending to restore the suspended object to the required position.

The first active magnetic suspension system was built and operated in the late 1930's[15]. Since then, a wide variety of analog controller for such systems have been conceived and used. The analog control systems have disadvantages. The most well known is *drift* in both the active and passive components used in construction of the system. Another is the difficulty of *reconfiguring* such systems. For instance, if the corner frequency of one of the signal-processing filters must be changed, components must be replaced (e.g., swapping out resistors in an op-amp feedback network) or more expensive adjustable components must be used (e.g., potentiometers). Moreover, analog signal processing systems have limited capabilities. As an example, it is difficult to synthesize certain kinds of high order filters and nonlinear functions with analog components. These problems, and others,

suggest using a digital controller.

The question of control system design for an operating magnetic suspension can be approached from different directions. The use of *proportional-integral-derivative* (PID) control may be a suitable method for compensating a single-axis magnetic suspension system; applying *state-feedback* may be preferable for a two-axis or multi-axis system.

One of main benefits of state-space methods for design of linear control system is the *separation principle*, which permits the control system to be designed in two stages. First, a “full-state” feedback control law is designed, and then an observer is provided to estimate the states that cannot be measured directly. But the magnetic suspension system is nonlinear . However, the separation principle can often be extended to nonlinear systems and can prove to be an effective design method. This is the method used here.

The magnetic suspension system is designed to maintain the state of the dynamic process at a specified constant value. This value is usually called the *set point*, or *operating point*. Sometimes it is referred to simply as the *referenced input*. The magnetic suspension system can be visualized as consisting of two distinct subsystem, the first to generate the control signal needed to keep the suspended object at the set position once it gets there, and the second to generate the control needed to return the suspended object to the set position if it should deviate from it.

In order to realize the state-feedback control to magnetic suspension

system. we need two groups of state variables. One group are *position-state-variables*, another are *velocity-state-variables*. From the transducers in this system, however, only position-state-variables can be obtained directly. To estimate the velocity-state-variables, an *observer* is used. An observer is dynamic system whose state variables are the estimates of the state variables of another system. Since the magnetic suspension system is nonlinear and the position-state-variables can be measured directly from the transducers, a nonlinear reduced-order observer can be used to estimate velocity. Alternatively, a finite difference method can be used for this purpose.

The essential features of our 2-axis experimental system are shown in Figure 1.1. We employ two DC solenoids to reach the objective of control. The cores of the solenoids are marked with measurements by sticking thin tapes on them. Two linear potentiometers, which are connected with the two cores by two hard sticks, are used as position transducers to produce two output signals proportional to the motion of the suspended object. A hard bar is suspended from the bottoms of the two cores, and two weights can be hung anywhere on it. The two solenoids are coupled through the mass of the hard bar and the two weights. When the weights are hung at different positions, the two solenoids have different working relations. The total weight of the two cores, the hard bar and the two weights is 1.164 Lb. The reference positions of the cores come from two reference pots. When

we turn the reference pots, the reference positions of the cores will be changed correspondingly. The two detected position signals coming from the linear potentiometers and the two reference position signals coming from the reference pots are digitized, operated on by an algorithm in the computer, and reconverted for use in driving the system's power amplifiers to stabilize the suspended object height and the angle of inclination.

The state-space control algorithm design techniques used that referred professor Bernard Friedland's book "Control System Design" and manuscript "Advanced Control System Design" (Chapter 4, Controlling Nonlinear Systems, and Chapter 5, Observers for Nonlinear Systems).

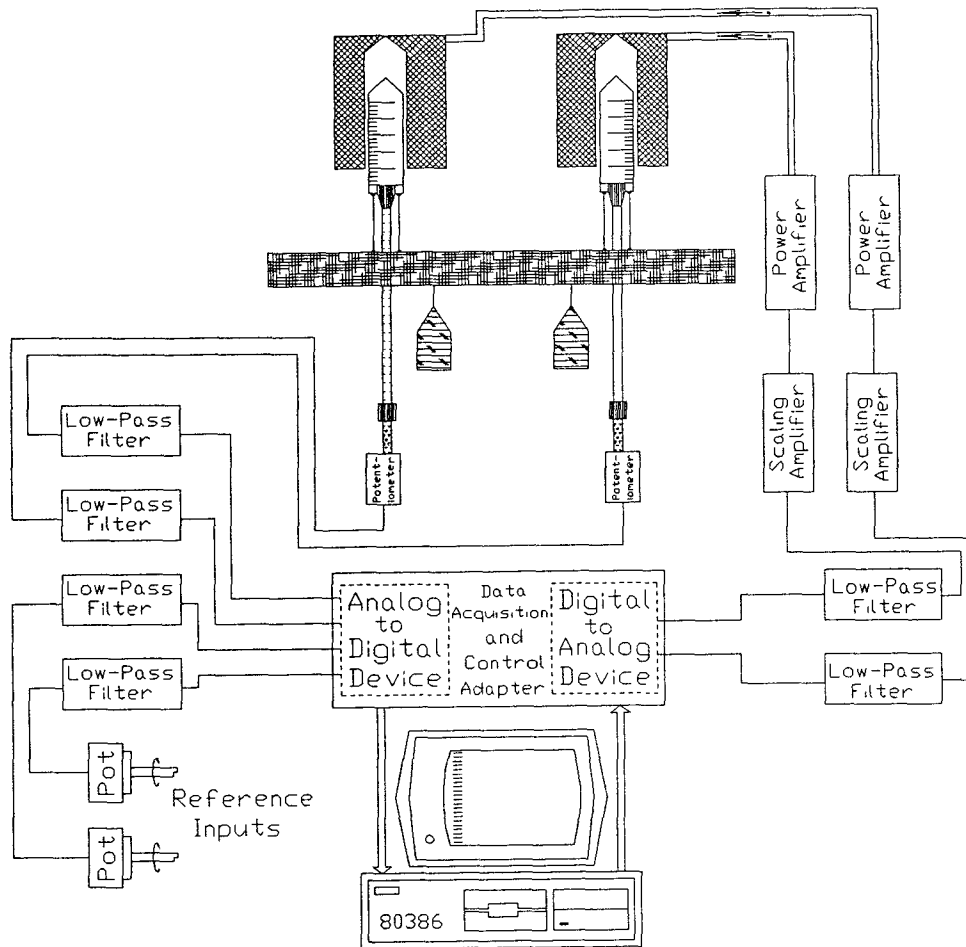


Figure 1.1: 2-Axis Magnetic Suspension System Using Computer Control

Chapter 2

ANALYSIS

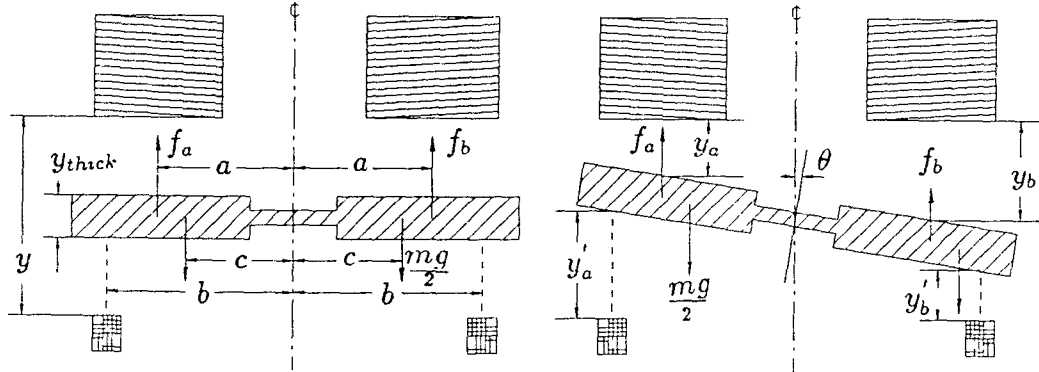
In our experimental system, the input to the algorithm are the data streams from the four-channel analog-to-digital(A/D) converter and the output from the algorithm are the data streams to the two digital-to-analog (D/A) converters, that will be amplified to drive the currents in the two DC magnets. Following is a description of algorithm analysis.

2.1 Model of 2-Axis Magnetic Suspension System

The mathematical model of 2-axis magnetic suspension system is shown in figure 2.1(a)(b). In theory, system is strictly symmetric, although there are slight asymmetries in reality.

The dynamic equations of the suspended object are:

$$m \left(\frac{\vec{y}'_a + \vec{y}'_b}{2} \right) = f_a + f_b - mg \quad (2.1)$$



- m : mass of the suspended object
- g : acceleration due to gravity
- θ : inclined angle of object, $< 5^\circ$
- y_{thick} : thick of object
- y : distance from magnet to sensor
- y_a, y_b : gaps between magnets and object
- y'_a, y'_b : distance from sensors to object
- f_a, f_b : magnetic forces between magnets and object
- a : distance from magnetic forces to symmetric axis
- b : distance from detective points to symmetric axis
- c : distance from core of halves of object to symmetric axis

Figure 2.1: Model of 2-Axis Magnetic Suspension System

$$J \left(\frac{\ddot{y}'_a - \ddot{y}'_b}{2b} \right) = f_a a - f_b a \quad (2.2)$$

since the angle of suspended object will be small, $\theta \doteq \sin \theta$

where:

$$J = 2 \cdot \frac{m}{2} \cdot c^2 = mc^2 \quad (2.3)$$

$$f_a = k_B \frac{i_a^2}{y_a^2} \quad (2.4)$$

$$f_b = k_B \frac{i_b^2}{y_b^2} \quad (2.5)$$

$$y_a = y - y_{thick} - y'_a \quad (2.6)$$

$$y_b = y - y_{thick} - y'_b \quad (2.7)$$

From equations (2.6) (2.7), we have

$$\ddot{y}_a = -\ddot{y}'_a$$

$$\ddot{y}_b = -\ddot{y}'_b$$

The dynamic equations become:

$$-m \left(\frac{\ddot{y}_a + \ddot{y}_b}{2} \right) = k_B \frac{i_a^2}{y_a^2} + k_B \frac{i_b^2}{y_b^2} - mg \quad (2.8)$$

$$-mc^2 \left(\frac{\ddot{y}_a - \ddot{y}_b}{2b} \right) = a \cdot k_B \frac{i_a^2}{y_a^2} + a \cdot k_B \frac{i_b^2}{y_b^2} \quad (2.9)$$

Rewrite the dynamic equations:

$$\ddot{y}_a = \alpha \frac{i_a^2}{y_a^2} + \beta \frac{i_b^2}{y_b^2} + g \quad (2.10)$$

$$\ddot{y}_b = \beta \frac{i_a^2}{y_a^2} + \alpha \frac{i_b^2}{y_b^2} + g \quad (2.11)$$

where

$$\alpha = \left(-\frac{1}{m} - \frac{ab}{mc^2} \right) \cdot k_B$$

$$\beta = \left(-\frac{1}{m} + \frac{ab}{mc^2} \right) \cdot k_B$$

Express in the form of state-space:

$$\dot{y}_a = v_a \quad (2.12)$$

$$\dot{y}_b = v_b \quad (2.13)$$

$$\dot{v}_a = \alpha \frac{i_a^2}{y_a^2} + \beta \frac{i_b^2}{y_b^2} + g \quad (2.14)$$

$$\dot{v}_b = \beta \frac{i_a^2}{y_a^2} + \alpha \frac{i_b^2}{y_b^2} + g \quad (2.15)$$

with

$$x = \begin{bmatrix} y_a & y_b & v_a & v_b \end{bmatrix}^T \quad \text{state variables}$$

$$u = \begin{bmatrix} i_a & i_b \end{bmatrix}^T \quad \text{controller}$$

$$y = \begin{bmatrix} y_a & y_b \end{bmatrix}^T \quad \text{observation}$$

2.2 Algorithm of Full-State Feedback Control Law

As mentioned in the Introduction, the magnetic suspension system can be visualized as consisting of two distinct subsystems, the first to generate the control signal needed to keep the suspended object at the set point once it gets there, and the second to generate the control needed to return the

suspended object to the set position if it should deviate from it. So, the total state x is the sum of the reference state \bar{x} and a perturbation δx :

$$x = \bar{x} + \delta x$$

similarly:

$$u = \bar{u} + \delta u$$

In our experimental system, in order to maintain a steady-state airgap \bar{y}_a and \bar{y}_b (i.e., to maintain a desired position of suspended object), steady-state currents \bar{i}_a and \bar{i}_b are required to keep the acceleration zero.

Since

$$\ddot{\bar{y}}_a = 0 \quad \ddot{\bar{y}}_b = 0$$

from equations (2.10) (2.11), we get the steady-state currents:

$$0 = \alpha \frac{\bar{i}_a^2}{\bar{y}_a^2} + \beta \frac{\bar{i}_b^2}{\bar{y}_b^2} + g \quad (2.16)$$

$$0 = \beta \frac{\bar{i}_a^2}{\bar{y}_a^2} + \alpha \frac{\bar{i}_b^2}{\bar{y}_b^2} + g \quad (2.17)$$

Thus, we get:

$$\bar{i}_a = \pm \sqrt{k} \cdot \bar{y}_a \quad (2.18)$$

$$\bar{i}_b = \pm \sqrt{k} \cdot \bar{y}_b \quad (2.19)$$

where

$$k = \frac{-g}{\alpha + \beta}$$

From equations (2.16) (2.17), we also can obtain the *steady-state condition* of the experimental system:

$$\frac{\bar{i}_a^2}{\bar{y}_a^2} = \frac{\bar{i}_b^2}{\bar{y}_b^2} \quad (2.20)$$

i.e.,

$$\frac{\bar{i}_a}{\bar{i}_b} = \frac{\bar{y}_a}{\bar{y}_b} \quad (2.21)$$

The next step is to design a control law by linearizing about the desired position of suspended object (i.e., the operating point). Let:

$$y_a = \bar{y}_a + \delta y_a$$

$$y_b = \bar{y}_b + \delta y_b$$

$$i_a = \bar{i}_a + \delta i_a$$

$$i_b = \bar{i}_b + \delta i_b$$

From the state equations of system model (2.12) (2.13) (2.14) (2.15), we linearize at the operating points $\bar{y}_a \bar{y}_b$, get the perturbation equation:

$$\delta \dot{y} = A(\bar{y}_a \bar{y}_b) \delta y + B(\bar{i}_a \bar{i}_b) \delta i \quad (2.22)$$

where:

$$A(\bar{y}_a \bar{y}_b) = \begin{bmatrix} \frac{\partial \dot{y}_a}{\partial y_a} & \frac{\partial \dot{y}_a}{\partial y_b} & \frac{\partial \dot{y}_a}{\partial v_a} & \frac{\partial \dot{y}_a}{\partial v_b} \\ \frac{\partial \dot{y}_b}{\partial y_a} & \frac{\partial \dot{y}_b}{\partial y_b} & \frac{\partial \dot{y}_b}{\partial v_a} & \frac{\partial \dot{y}_b}{\partial v_b} \\ \frac{\partial \dot{v}_a}{\partial y_a} & \frac{\partial \dot{v}_a}{\partial y_b} & \frac{\partial \dot{v}_a}{\partial v_a} & \frac{\partial \dot{v}_a}{\partial v_b} \\ \frac{\partial \dot{v}_b}{\partial y_a} & \frac{\partial \dot{v}_b}{\partial y_b} & \frac{\partial \dot{v}_b}{\partial v_a} & \frac{\partial \dot{v}_b}{\partial v_b} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-2\alpha k}{\bar{y}_a} & \frac{-2\beta k}{\bar{y}_b} & 0 & 0 \\ \frac{-2\beta k}{\bar{y}_a} & \frac{-2\alpha k}{\bar{y}_b} & 0 & 0 \end{bmatrix}$$

$$B(\bar{i}_a \bar{i}_b) = \begin{bmatrix} \frac{\partial \dot{y}_a}{\partial i_a} & \frac{\partial \dot{y}_a}{\partial i_b} \\ \frac{\partial \dot{y}_b}{\partial i_a} & \frac{\partial \dot{y}_b}{\partial i_b} \\ \frac{\partial \dot{v}_a}{\partial i_a} & \frac{\partial \dot{v}_a}{\partial i_b} \\ \frac{\partial \dot{v}_b}{\partial i_a} & \frac{\partial \dot{v}_b}{\partial i_b} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{2\alpha k}{\bar{i}_a} & \frac{2\beta k}{\bar{i}_b} \\ \frac{2\beta k}{\bar{i}_a} & \frac{2\alpha k}{\bar{i}_b} \end{bmatrix}$$

$$\delta y = \begin{bmatrix} y_a - \bar{y}_a \\ y_b - \bar{y}_b \\ v_a - \bar{v}_a \\ v_b - \bar{v}_b \end{bmatrix} = \begin{bmatrix} \delta y_a \\ \delta y_b \\ v_a \\ v_b \end{bmatrix}$$

$$\delta i = \begin{bmatrix} \delta i_a \\ \delta i_b \end{bmatrix}$$

A suitable linear, full-state feedback control law is:

$$\delta i = -G \cdot \delta y \quad (2.23)$$

where:

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \end{bmatrix}$$

We can solve for the gain matrix G numerically using an appropriate algorithm as discusses in section 2.4.

Consequently, get the full-state feedback control law at the set point:

$$\begin{aligned} i_a = & \sqrt{\frac{-g}{\alpha + \beta}} \bar{y}_a + g_{11}(\bar{y}_a - y_a) + g_{12}(\bar{y}_b - y_b) \\ & + g_{13}(-v_a) + g_{14}(-v_b) \end{aligned} \quad (2.24)$$

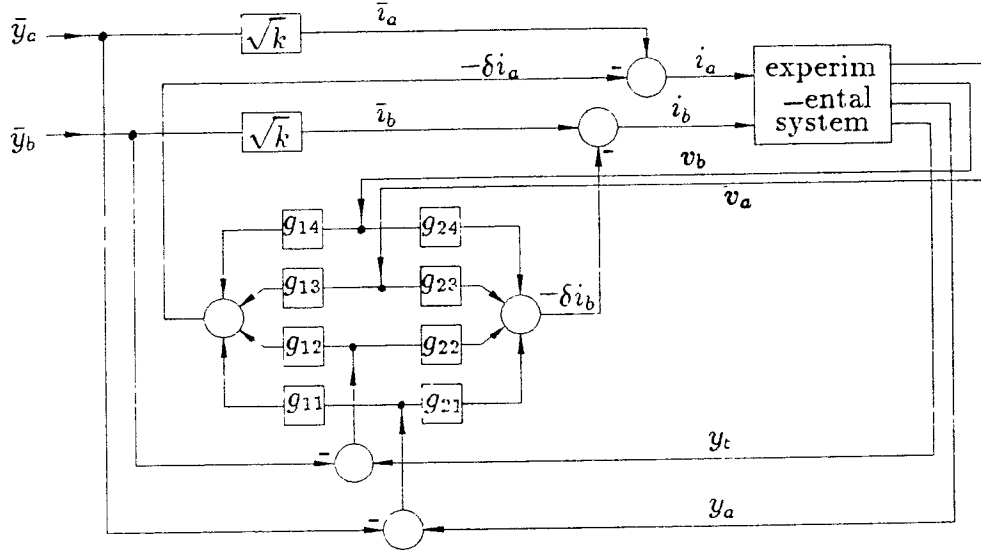


Figure 2.2: Block diagram of full-state feedback design of 2-axis magnetic suspension system based on linearization about set point

$$\begin{aligned}
 i_b = & \sqrt{\frac{-g}{\alpha + \beta}} \bar{y}_b + g_{21}(\bar{y}_a - y_a) + g_{22}(\bar{y}_b - y_b) \\
 & + g_{23}(-v_a) + g_{24}(-v_b)
 \end{aligned} \tag{2.25}$$

The block diagram of this algorithm is show in Figure 2.2

2.3 Algorithm of Nonlinear, Reduced-Order Observer

Reduced-order observers have two benefits over full-order observers: reduced computational requirements and superior performance, because it usually turns out that a control law based on a reduced-order observer is more robust than one using a full-order observer.

To design the reduced-order observer, rewrite the state equation of the

system:

$$\dot{x}_1 = x_2 \quad (2.26)$$

$$\dot{x}_2 = f(x_1, u) \quad (2.27)$$

where:

$$x_1 = \begin{bmatrix} y_a \\ y_b \end{bmatrix} \quad \text{observed variables}$$

$$x_2 = \begin{bmatrix} v_a \\ v_b \end{bmatrix} \quad \text{unobserved variables}$$

$$u = \begin{bmatrix} i_a \\ i_b \end{bmatrix} \quad \text{controller}$$

$$f(x_1, u) = \begin{bmatrix} \alpha \frac{i_a^2}{y_a^2} + \beta \frac{i_b^2}{y_b^2} + g \\ \beta \frac{i_a^2}{y_a^2} + \alpha \frac{i_b^2}{y_b^2} + g \end{bmatrix}$$

The observation is:

$$y = \begin{bmatrix} y_a \\ y_b \end{bmatrix} \quad (2.28)$$

Following the theory in *Advanced Control System Design* [7], the non-linear, reduced-order observer is assumed to have the same structure as the corresponding linear observer.

For the estimate of the substate x_1 , we use the observation itself:

$$\hat{x}_1 = y \quad (2.29)$$

while the substate x_2 is estimated using an observer of the form:

$$\hat{x}_2 = ky + z \quad (2.30)$$

where:

$$\hat{x}_2 = \begin{bmatrix} \hat{v}_a \\ \hat{v}_b \end{bmatrix} \quad \text{is estimated states}$$

$$k = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \quad \text{is the gain matrix}$$

$$y = \begin{bmatrix} y_a \\ y_b \end{bmatrix} \quad \text{is observation}$$

z is the state of a dynamic system of the same order as the dimension of the subvector x_2 and is given by:

$$\dot{z} = \phi(y, \hat{x}_2, u)$$

The object of the observer design is the determination of the gain matrix k and the nonlinear function ϕ . These are to be selected such that:

- The steady-state error in estimating x_2 converges to zero, independent of x_1 and u . (The error in estimating x_1 is already zero when $\hat{x}_1 = y$)
- The observer is asymptotically stable.

since:

$$e = x_2 - \hat{x}_2$$

we have:

$$\begin{aligned} \dot{e} &= \dot{x}_2 - \dot{\hat{x}}_2 \\ &= \dot{x}_2 - k\dot{y} - \dot{z} \end{aligned}$$

$$\begin{aligned}
&= f(x_1, u) - kx_2 - \phi(y, \hat{x}_2, u) \\
&= f(y, u) - kx_2 - \phi(y, x_2 - e, u)
\end{aligned} \tag{2.31}$$

In order for the right-hand side of equation (2.31) to vanish when $e = 0$, it is necessary that the function $\phi(y, x_2, u)$ satisfy:

$$\phi(y, x_2, u) = f(y, u) - kx_2 \tag{2.32}$$

for all values of y , x_2 and u .

So, we have:

$$\begin{aligned}
\dot{z} &= \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} = \phi(y, x_2, u) \\
&= \begin{bmatrix} \alpha \frac{\dot{z}_a^2}{y_a^2} + \beta \frac{\dot{z}_b^2}{y_b^2} + g - k_{11}v_a - k_{12}v_b \\ \beta \frac{\dot{z}_a^2}{y_a^2} + \alpha \frac{\dot{z}_b^2}{y_b^2} + g - k_{21}v_a - k_{22}v_b \end{bmatrix}
\end{aligned} \tag{2.33}$$

To achieve asymptotic stability, the linearized system,

$$\dot{e} = A(x_2) \cdot e \tag{2.34}$$

with

$$A(x_2) = (\partial\phi/\partial x_2)$$

In order for the error to approach zero asymptotically it is necessary that the Jacobian Matrix $A(x_2)$ be a stability matrix, i.e., that the eigenvalues of the matrix $A(x_2)$ all be in the left half-plane.

The Jacobian Matrix for 2-axis magnetic suspension system is:

$$\begin{aligned}
A(x_2) &= \begin{bmatrix} \frac{\partial\varphi_1}{\partial v_a} & \frac{\partial\varphi_1}{\partial v_b} \\ \frac{\partial\varphi_2}{\partial v_a} & \frac{\partial\varphi_2}{\partial v_b} \end{bmatrix} \\
&= \begin{bmatrix} -k_{11} & -k_{12} \\ -k_{21} & -k_{22} \end{bmatrix}
\end{aligned} \tag{2.35}$$

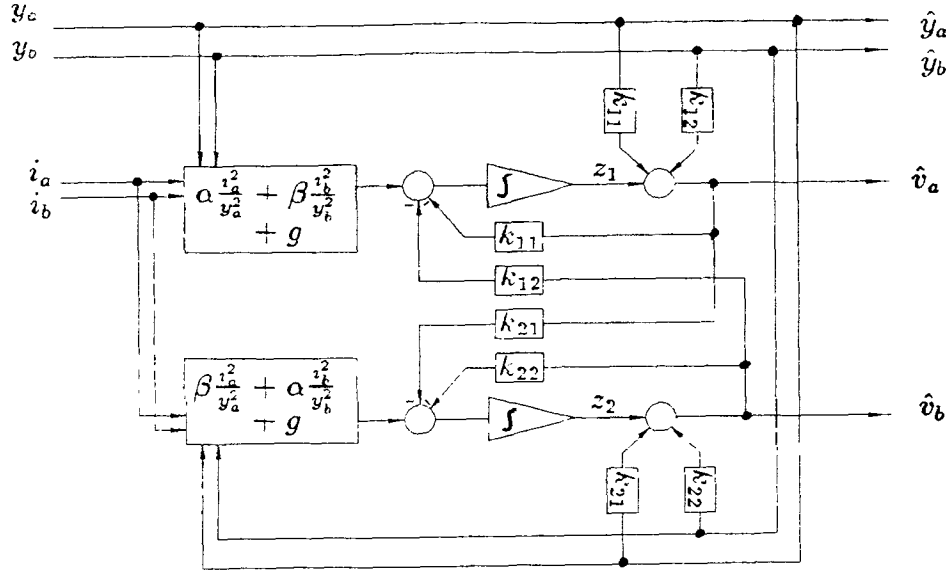


Figure 2.3: Block diagram of nonlinear reduced-order observer design of 2-axis magnetic suspension

We can employ pole-placement to determine the gain matrix k to place the two eigenvalues of observer at desired locations. The location of the closed-loop poles is discussed in section 2.4.

The block diagram of this observer is shown in Figure 2.3.

2.4 The Gain Matrices of Full-State Feedback and Observer

2.4.1 The Gain Matrix of Observer

Considerations of symmetry suggest that the observer gain matrix should be of the form:

$$K = \begin{bmatrix} k_1 & k_2 \\ k_2 & k_1 \end{bmatrix}$$

The characteristic equation of observer becomes:

$$\begin{aligned} |SI - A(X_2)| &= \begin{vmatrix} s + k_1 & k_2 \\ k_2 & s + k_1 \end{vmatrix} \\ &= s^2 + 2k_1s + (k_1^2 + k_2^2) = 0 \end{aligned} \quad (2.36)$$

This means that there are only two parameters to be determined for a second-order observer, and hence that a unique design can be achieved by pole-placement.

We choose a second-order Butterworth configuration of radius r . The corresponding characteristic equation of the observer is:

$$D(s) = \left(\frac{s}{r}\right)^2 + \sqrt{2}\left(\frac{s}{r}\right) + 1 = 0$$

i.e.

$$s^2 + \sqrt{2}r \cdot s + r^2 = 0 \quad (2.37)$$

Comparing equations (2.36) (2.37), we get:

$$k_1 = \frac{\sqrt{2}}{2}r \quad (2.38)$$

$$k_2 = \sqrt{r^2 - k_1^2} = \frac{\sqrt{2}}{2}r \quad (2.39)$$

The dynamics of the observer are linear and time-invariant.

2.4.2 The Gain Matrix of Full-State Feedback

We also can employ the pole-placement technique to determine the gain matrix G . Considerations of symmetry suggest that the full-state feedback

gain matrix should be of the form:

$$\begin{bmatrix} g_1 & g_2 & g_3 & g_4 \\ g_2 & g_1 & g_4 & g_3 \end{bmatrix}$$

Since we have four parameters for a fourth-order full-state feedback, a unique design can be achieved by pole-placement.

But a better way to solve this problem may be using linear, quadratic optimum regulator (LQR) theory.

To use LQR theory, consider dynamic equation:

$$\dot{x} = Ax + Bu$$

and control law:

$$u(t) = -Gx(t)$$

Instead of seeking a gain matrix G to achieve specified closed-loop pole locations, we seek a gain to minimize a specified performance criterion V (also called “cost function”) expressed as the integral of a quadratic form in the state x plus a second quadratic form in the control u ; i.e.,

$$V = \int_t^\infty (x' Q x + u' R u) d\tau$$

where Q and R are symmetric matrices.

The objective can be achieved by solving *algebraic Riccati equation*(ARE),

$$0 = \bar{M}A + A'\bar{M} - \bar{M}BR^{-1}B'\bar{M} + Q$$

and the optimum gain in the steady state is given by,

$$G = R^{-1}B'\bar{M}$$

where

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-2\alpha k}{\bar{y}_a} & \frac{-2\beta k}{\bar{y}_b} & 0 & 0 \\ \frac{-2\beta k}{\bar{y}_a} & \frac{-2\alpha k}{\bar{y}_b} & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{2\alpha k}{\bar{y}_a} & \frac{2\beta k}{\bar{y}_b} \\ \frac{2\beta k}{\bar{y}_a} & \frac{2\alpha k}{\bar{y}_b} \end{bmatrix}$$

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \end{bmatrix}$$

$$Q = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

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

with:

$$k = \frac{-g}{\alpha + \beta}$$

$$\bar{y}_a = \sqrt{k} \bar{y}_a \quad \bar{y}_b = \sqrt{k} \bar{y}_b$$

$$\alpha = \left(-\frac{1}{m} - \frac{ab}{mc^2}\right)k_B \quad \beta = \left(-\frac{1}{m} + \frac{ab}{mc^2}\right)k_B$$

- a : distance from magnetic forces to symmetric axis
- b : distance from detective points to symmetric axis
- c : distance from core of halves of suspended object to symmetric axis
- d : constant, need to be decided
- k_B : coefficient of the magnetic force, depending on the structure of magnet

We can solve for the gain matrix G that results for a range of d , and simulate the corresponding closed-loop response.

The block diagram of whole control algorithm is shown in Figure 2.4

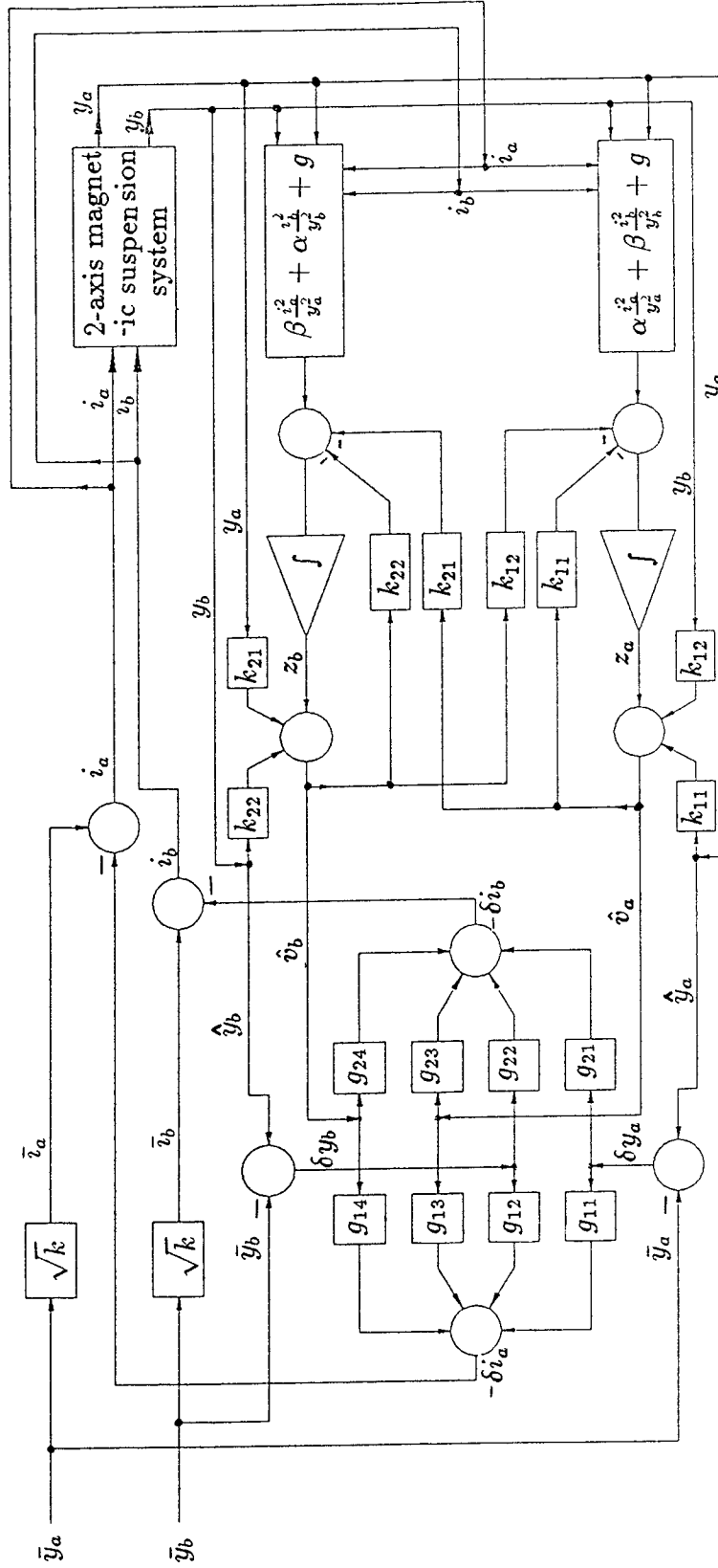


Figure 2.4: Block diagram of control algorithm of 2-axis magnetic suspension system

2.5 Additional Considerations for Experimental System

2.5.1 Another Way to Estimate Velocities

Besides using a nonlinear, reduced-order observer to estimate velocities, another way is to use a finite difference algorithm. One could take just the last two points along with the current point to approximate time derivatives:

$$v(i) = \frac{1.5y(i) - 2y(i-1) + 0.5y(i-2)}{T}$$

$v(i)$: velocity at time i
 $y(i)$: position at time i
 $y(i-1)$: position of last sampling (at time $i-1$)
 $y(i-2)$: position at time $i-2$
 T : sample period

To apply the difference algorithm to estimate velocity, the computer needs only to do simple algebraic operations. A lot of CPU time can be saved, which is very important in real-time computer control.

2.5.2 Employ Integral Feedback Control to Achieve High Control Accuracy

In our experimental system, we use the linear potentiometers as position sensors and employ the DC solenoids as electromagnets. There is a small uncertain friction when the core of solenoid and the bar of potentiometer move. Although the design of system is robust enough to overcome the

friction, the control accuracy degrades. A fancy way to deal with this problem is to re-design the nonlinear, reduced-order observer, to include the estimate of the friction. This may not be practical with our apparatus, because solving the differential equations of the observer may require too much CPU time.

A customary way of dealing with the friction problem is to introduce position error integral feedback control into the control law. The control law becomes:

$$\begin{aligned} i_a = & \sqrt{\frac{-g}{\alpha + \beta}} \bar{y}_a + g_{11}(\bar{y}_a - y_a) + g_{12}(\bar{y}_b - y_b) \\ & + g_{13}(-v_a) + g_{14}(-v_b) + \gamma \int_0^t (\bar{y}_a - y_a) d\tau \end{aligned} \quad (2.40)$$

$$\begin{aligned} i_b = & \sqrt{\frac{-g}{\alpha + \beta}} \bar{y}_b + g_{21}(\bar{y}_a - y_a) + g_{22}(\bar{y}_b - y_b) \\ & + g_{23}(-v_a) + g_{24}(-v_b) + \gamma \int_0^t (\bar{y}_b - y_b) d\tau \end{aligned} \quad (2.41)$$

To calculate the integral term, the trapezoid rule was used in the following form:

$$\int_t^{t+T} f(\tau) d\tau \approx \frac{T}{2} [f(t+T) + f(t)]$$

This operation is also easily implemented by computer.

Chapter 3

DESIGN AND SIMULATED PERFORMANCE

This chapter is concerned with details of the control system design, using actual measured hardware characters and simulated performance using these characteristics.

3.1 Basic Character of Magnet

The magnets used in the suspension are commercial solenoids. They are chosen for availability and cost. The relationship between force, current and core displacement is determined experimentally using the set up shown in Figure 3.1. We assume that $f = k_B \frac{i^2}{y^2}$ and calculate k_B at different position of core (suspended object) according to corresponding values of i , y , f .

The experiment process is to create a known displacement by putting some non-magnetic material between core and solenoid, such as, paper, plastic, etc., and attaching a known mass; then applying just enough current

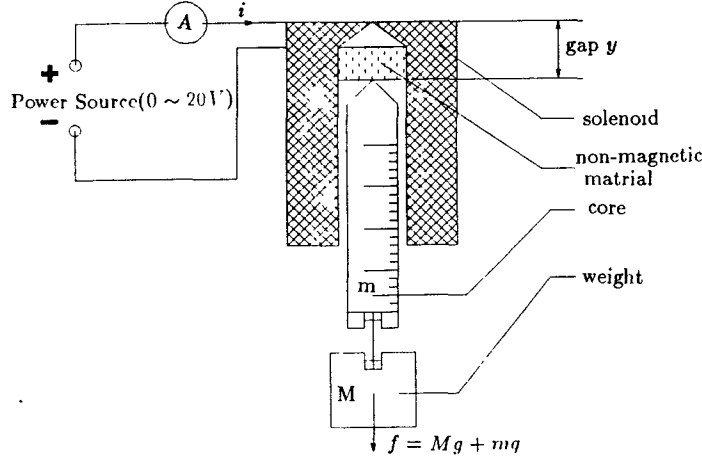


Figure 3.1: Diagram of solenoid test

to keep the core from falling, recording the current and the value $f = mg + Mg$ (m is the mass of core and M is the mass of weight). The results of experiment are shown in Figure 3.2 and 3.3.

The inductance of the solenoid is measured, although it does not appear in the dynamic equations. Actually, we can use L, y, i to replace k_B, y, i to obtain another expression of magnetic force f .

From the view of conservation of energy,

$$\int f dy = e = \frac{1}{2} Li^2$$

we get,

$$f = \frac{i^2}{2} \frac{dL}{dy}$$

Since in our experiment, the solenoid works in DC state, the low frequency behavior of the inductance is of interest. The experiment result is

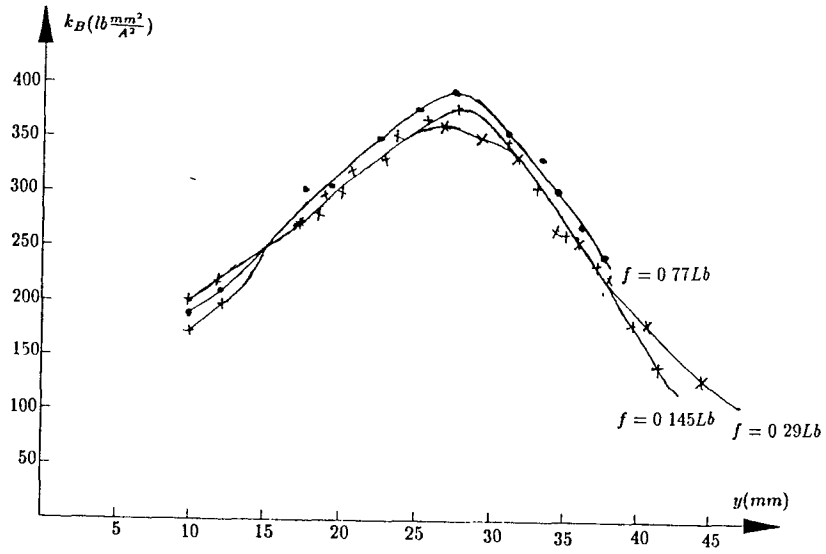


Figure 3.2: k_B versus gap y of solenoid

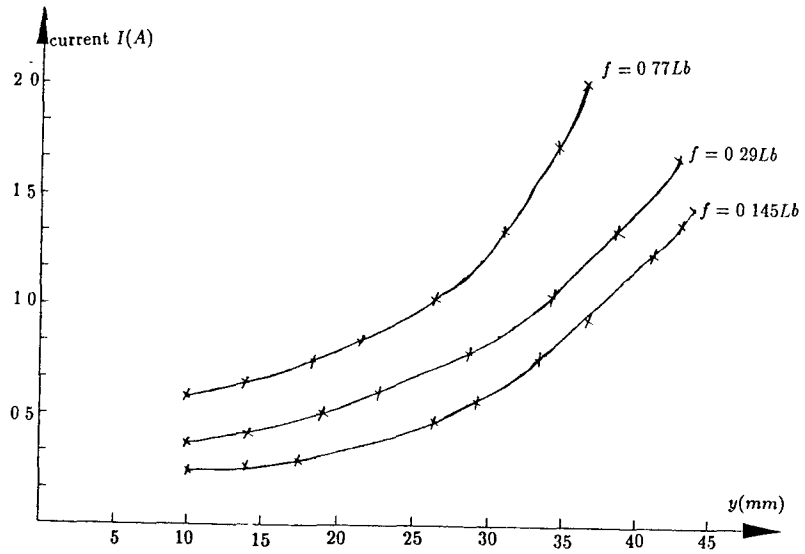


Figure 3.3: Relationship of i , f and y of solenoid

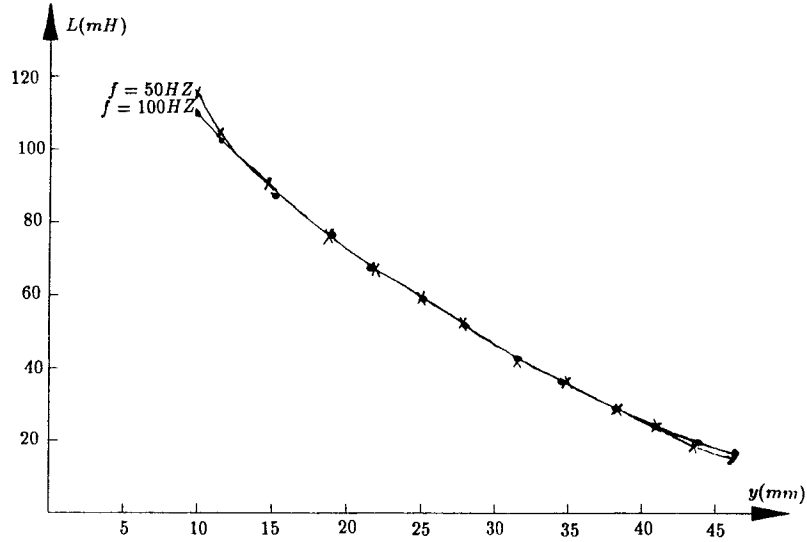


Figure 3.4: The inductance of solenoid

shown in Figure 3.4.

The resistance of the solenoid also is an important factor. In our experiment, the resistance of solenoid about 9.3Ω . In order that the power dissipation of solenoid will be less than 40 W, the control current should be below 2.0 A.

In our experiment, we will control the airgaps in the two solenoids from 15mm to 25mm (actually, a wider range also can be chosen, but the two linear potentiometers, which we employ as position sensors, only possess 10mm effective range) and the maximum weight of suspended object will be 1.2 Lb.

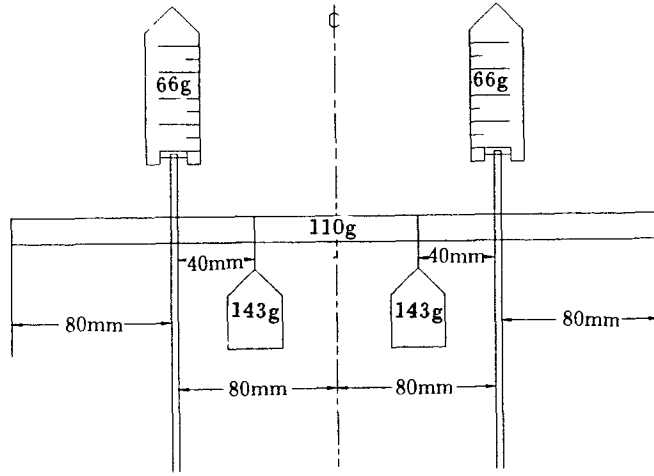


Figure 3.5: A diagram of the suspended object of the experiment

3.2 Design Calculations

The experiment is set up as figure 1.1. Figure 3.5 is a diagram of the suspended object. The system parameters are chosen and calculated as below. The notation follows section 2.4.

Choose the control range: $15mm < y_a, y_b < 25mm$

Choose the reference positions: $\bar{y}_a = 20mm, \quad \bar{y}_b = 20mm$

Acceleration due to gravity: $g = 9.8m/s^2$

From figure 3.5: $a = 80mm, \quad b = 80mm, \quad m = 0.528Kg$

Calculate from figure 3.5: $c = 52.8mm$

From figure 3.2: $k_B = 300lb \frac{mm^2}{A^2} = 1.335 \times 10^{-3} Kg \frac{m^2}{A^2}$

Calculate coefficient: $\alpha = (-\frac{1}{m} - \frac{ab}{mc^2})k_B = -7.33 \times 10^{-3}$

Calculate coefficient: $\beta = (-\frac{1}{m} + \frac{ab}{mc^2})k_B = 2.275 \times 10^{-3}$

Choose: $d = 1, \quad r = 3$

Calculate matrices at set point:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1421 & -441 & 0 & 0 \\ -441 & 1421 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -32.3 & 10 \\ 10 & -32.3 \end{bmatrix}$$

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

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

$$G = \begin{bmatrix} -87.976 & 7.1919 \times 10^{-2} & -2.4241 & -0.38374 \\ 7.311 \times 10^{-2} & -87.977 & -0.38371 & -2.4241 \end{bmatrix}$$

$$K = \begin{bmatrix} 2.121 & 2.121 \\ 2.121 & 2.121 \end{bmatrix}$$

A linear, fixed gain control law may be robust enough to be valid over the normal range of operation of the system.

3.3 Numerical Simulations

Base on the analysis in Chapter 2, the dynamics of the overall system is described by:

Plant dynamics:

$$\dot{y}_a = v_a \quad (3.1)$$

$$\dot{y}_b = v_b \quad (3.2)$$

$$\dot{i}_a = \alpha \frac{i_a^2}{y_a^2} + \beta \frac{i_b^2}{y_b^2} + g \quad (3.3)$$

$$\dot{i}_b = \beta \frac{i_a^2}{y_a^2} + \alpha \frac{i_b^2}{y_b^2} + g \quad (3.4)$$

Control law:

$$\begin{aligned} i_a = & \sqrt{\frac{-g}{\alpha + \beta}} \bar{y}_a + g_{11}(\bar{y}_a - y_a) + g_{12}(\bar{y}_b - y_b) \\ & + g_{13}(-v_a) + g_{14}(-v_b) + \gamma \int_0^t (\bar{y}_a - y_a) d\tau \end{aligned} \quad (3.5)$$

$$\begin{aligned} i_b = & \sqrt{\frac{-g}{\alpha + \beta}} \bar{y}_b + g_{21}(\bar{y}_a - y_a) + g_{22}(\bar{y}_b - y_b) \\ & + g_{23}(-v_a) + g_{24}(-v_b) + \gamma \int_0^t (\bar{y}_b - y_b) d\tau \end{aligned} \quad (3.6)$$

Nonlinear reduced-order observer:

$$\dot{z}_a = \alpha \frac{i_a^2}{y_a^2} + \beta \frac{i_b^2}{y_b^2} + g - k_{11}\hat{v}_a - k_{12}\hat{v}_b \quad (3.7)$$

$$\dot{z}_b = \beta \frac{i_a^2}{y_a^2} + \alpha \frac{i_b^2}{y_b^2} + g - k_{21}\hat{v}_a - k_{22}\hat{v}_b \quad (3.8)$$

$$\hat{v}_a = z_a + y_a k_{11} + y_b k_{12} \quad (3.9)$$

$$\hat{v}_b = z_b + y_a k_{21} + y_b k_{22} \quad (3.10)$$

The dynamics of the system is simulated by using ALSIM. Since the integral terms in the control law are aimed at the uncertain friction in our experimental system, they are introduced only from the view of engineering to achieve high control accuracy. If the friction does not exist, the integral terms are unnecessary. Here, we present the simulated results of the experimental system by using the nonlinear observer to estimate the velocities and by using the difference algorithm to approximate the time derivatives under the assumption that the friction does not exist. Typical simulated results are presented below.

3.3.1 Simulation by Using Observer

We choose initial positions: $y_a = 17mm$, $y_b = 23mm$

Since we assume that the friction is not in existence, choose $\gamma = 0$

The initial states of the nonlinear observer should be set as follows:

$$z_1 = -0.017 \times 2.121 - 0.023 \times 2.121 = -0.08484$$

$$z_2 = -0.017 \times 2.121 - 0.023 \times 2.121 = -0.08484$$

The results are shown in figures 3.6 3.7 3.8 3.9. The simulation code is listed in appendix A.

Because the nonlinear observer works so well, the estimated velocities \hat{v}_a \hat{v}_b are almost the same as the actual velocities v_a v_b . In figure 3.8, \hat{v}_a \hat{v}_b and v_a v_b overlap. In figure 3.9, there are close similarity between \hat{v}_a \hat{v}_b and v_a v_b .

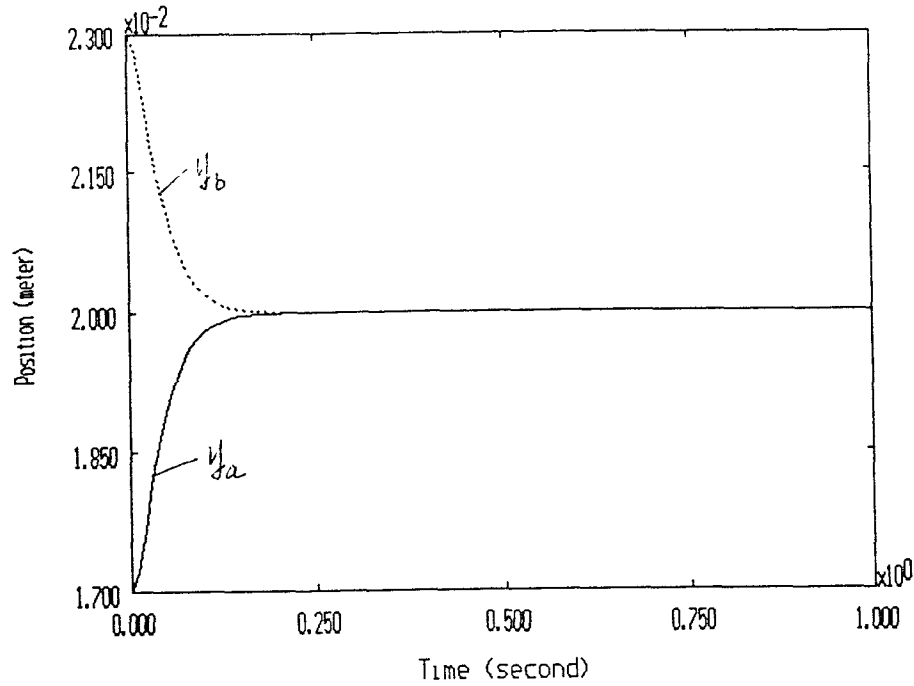


Figure 3.6: The positions using nonlinear observer when z set (simulated result)

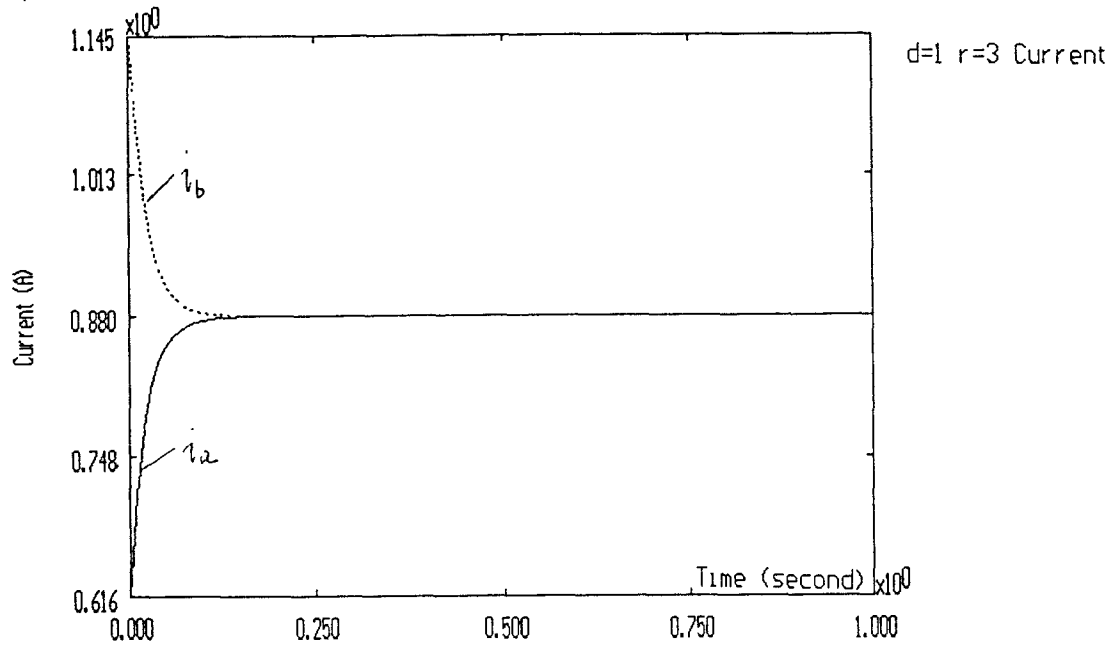


Figure 3.7: The control currents using nonlinear observer when z set (simulated result)

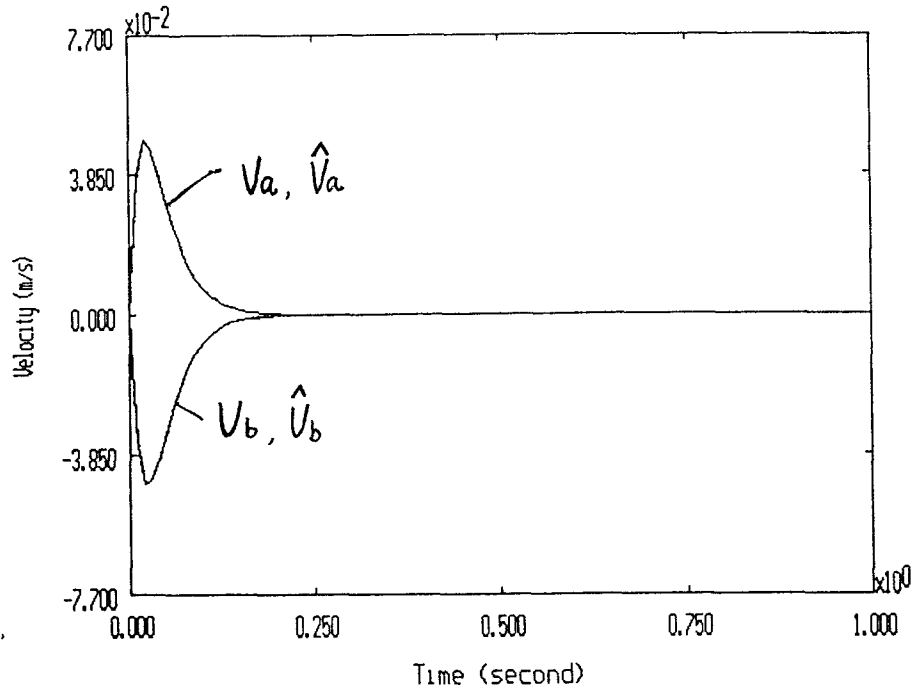


Figure 3.8: The estimated and actual velocities using nonlinear observer when z set (simulated result)

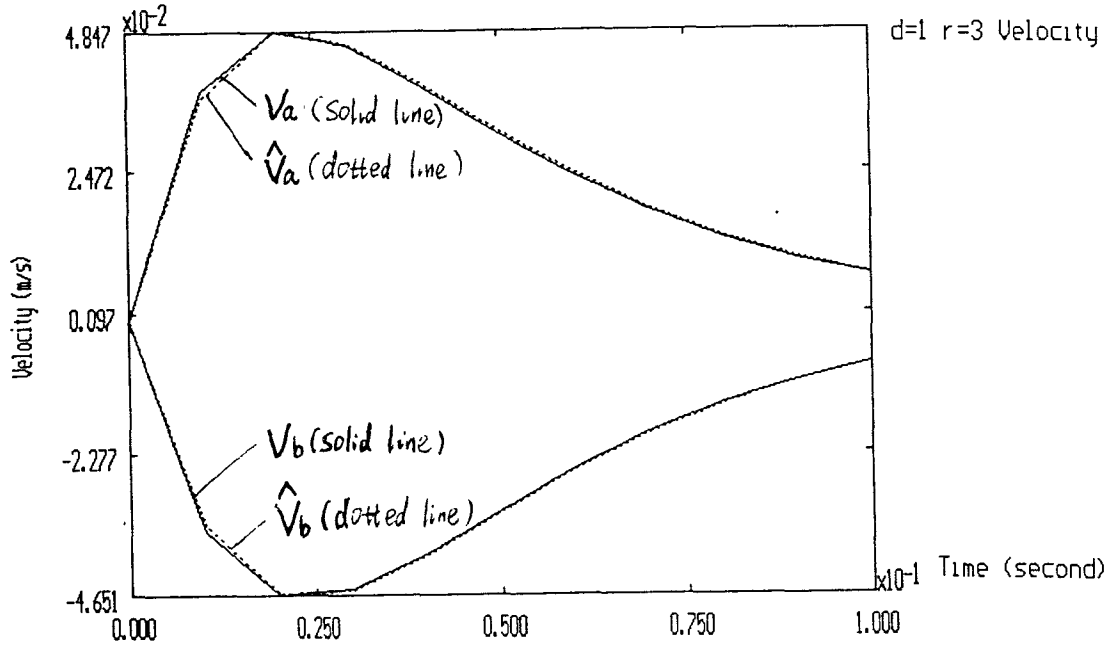


Figure 3.9: Closeup of the estimated and actual velocities using nonlinear observer when z set (simulated result)

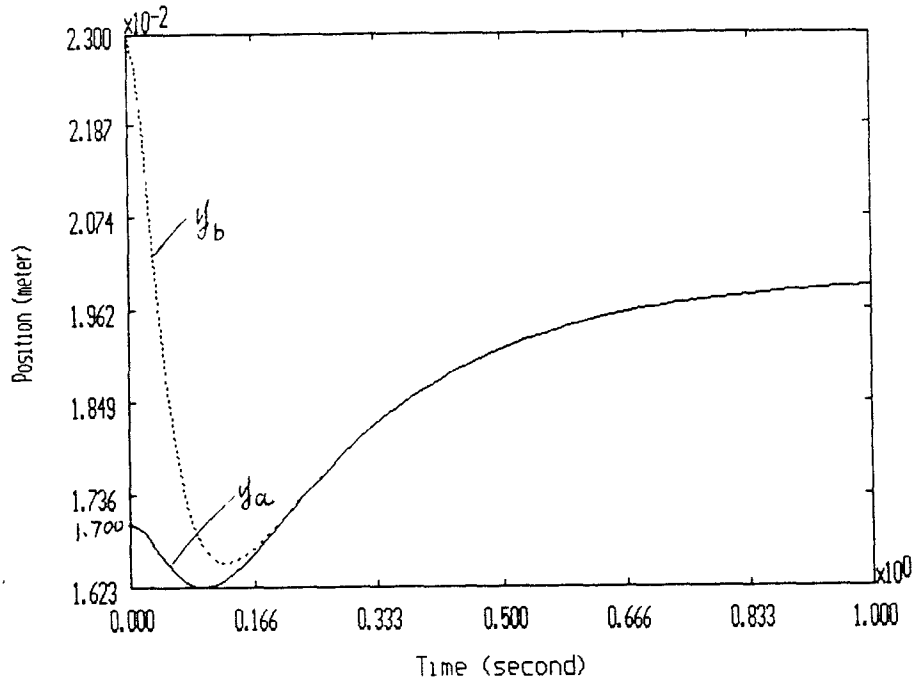


Figure 3.10: The positions using nonlinear observer when z not set(simulated result)

If we set the initial states of the observer as zero, the initial estimated velocities will be 84.84 mm/s . The results are shown in Figures 3.10 3.11 3.12. The estimated velocities will tend to reach the actual velocities after 1 second.

The results of the numerical simulation show that the system model works pretty well. It also proves that the performance of the nonlinear reduced-order observer is quite satisfactory.

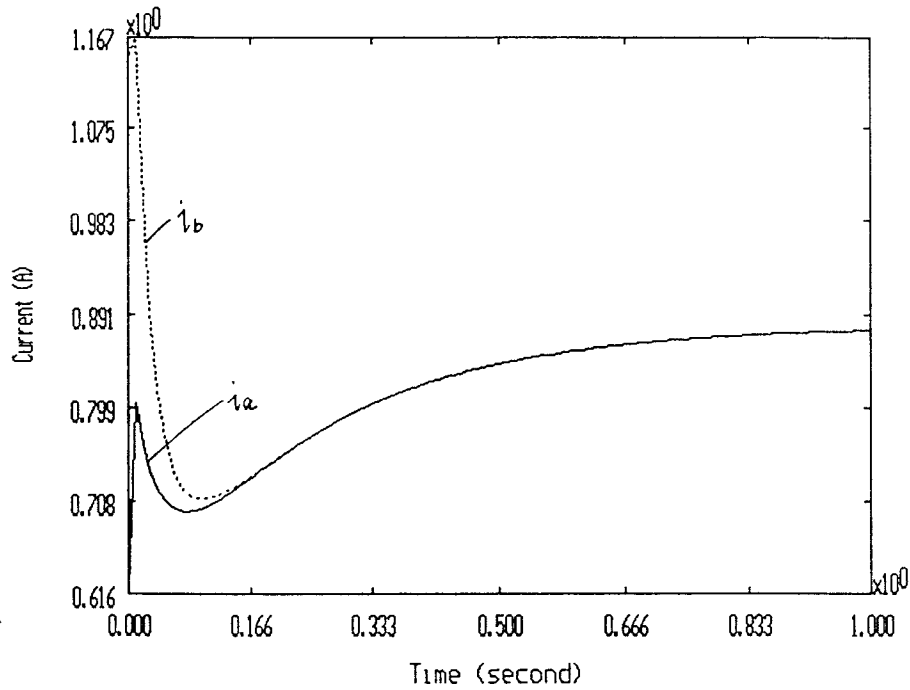


Figure 3.11: The control currents using nonlinear observer when z not set(simulated result)

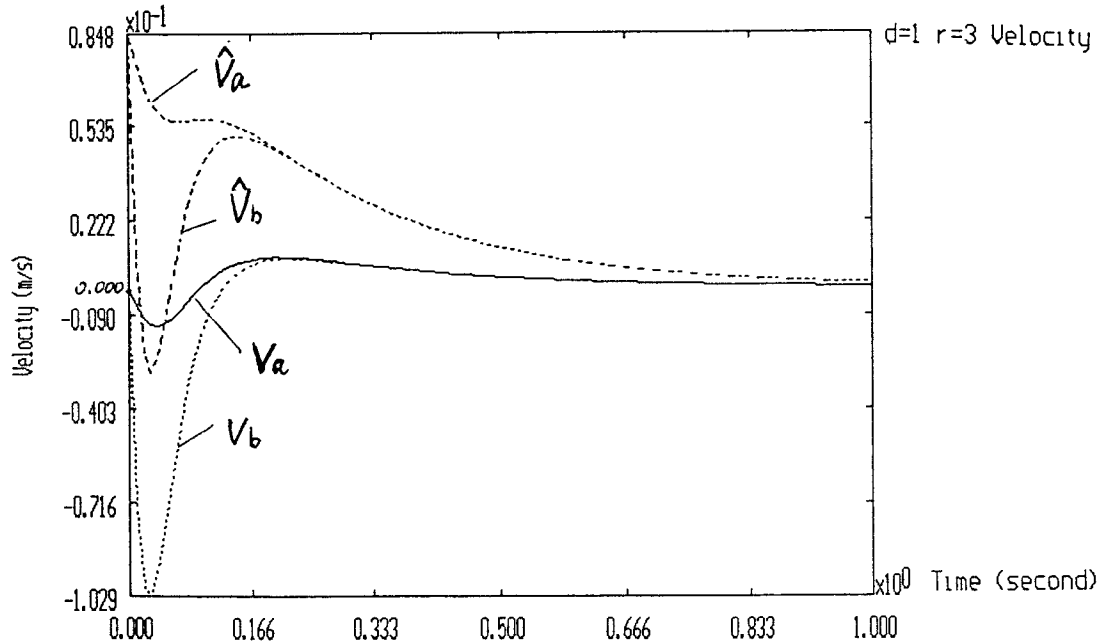


Figure 3.12: The estimated and actual velocities using nonlinear observer when z not set (simulated result)

3.3.2 Comparison of Nonlinear Observer With Difference Algorithm

In order to compare the nonlinear observer with the difference algorithm, we will present the simulated results of the experimental system by using the nonlinear observer and by using the difference algorithm to estimate the velocities.

We choose initial positions: $y_a = 25mm$, $y_b = 25mm$

Assume that the friction is not present, we choose $\gamma = 0$

The results are shown in Figures 3.13, 3.14, 3.15, and 3.16. We choose the positions, velocities, and currents at the magnet A for comparison. The simulation code using difference algorithm is listed in appendix B.

The comparison of the simulated results shows that the full-state feedback of magnetic suspension system have a better performance when we use the nonlinear reduced-order observer to estimate the velocities.

The next step, we will set up the experimental system and test performance.

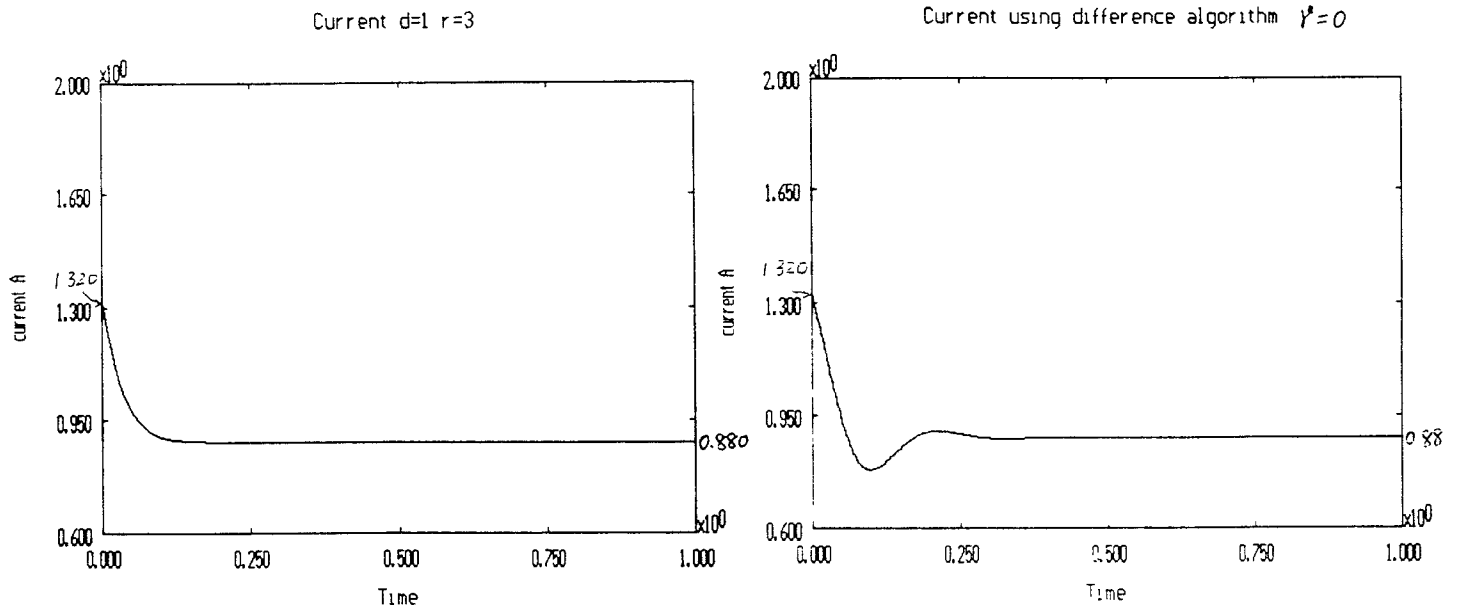


Figure 3.13: Comparison of positions when using observer(left) and using difference algorithm(right)

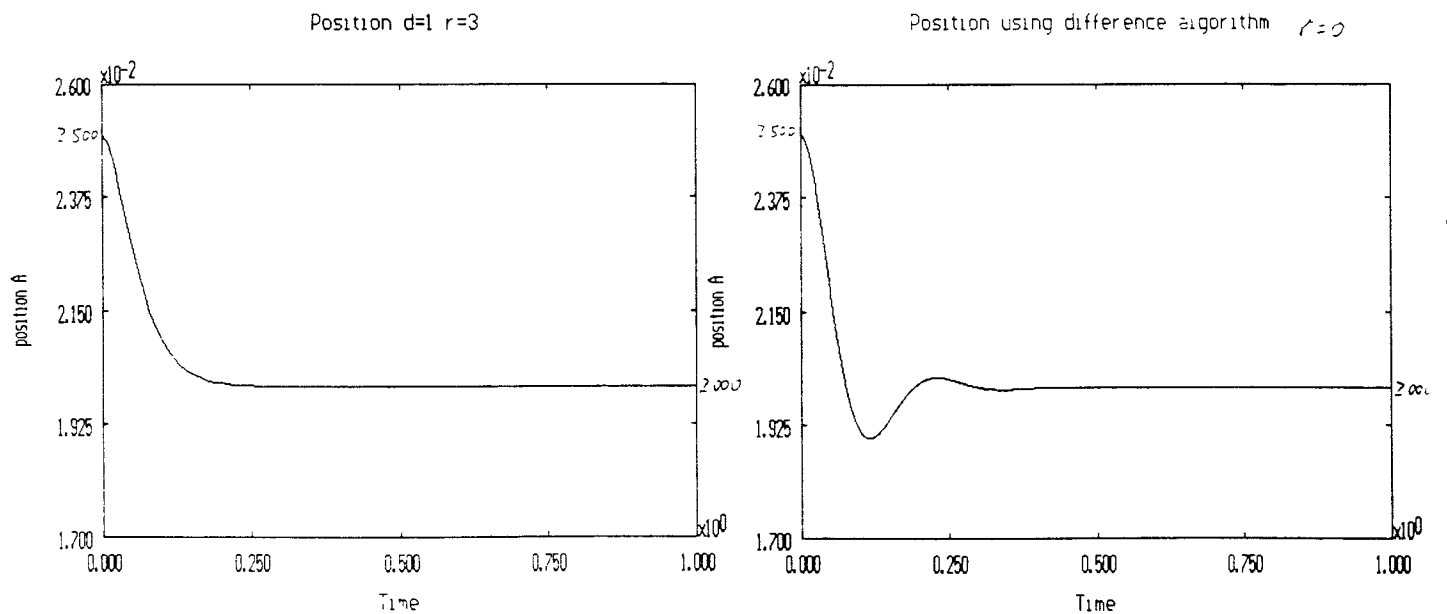


Figure 3.14: Comparison of control currents when using observer(left) and using difference algorithm(right)

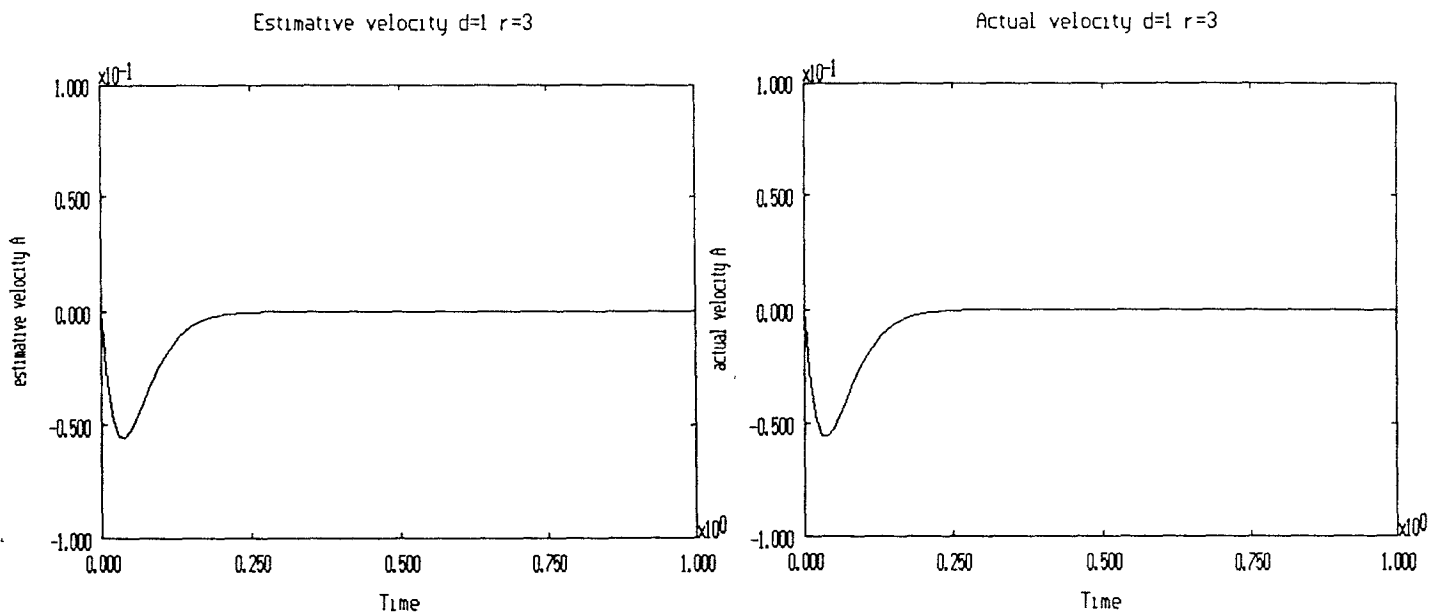


Figure 3.15: The estimated velocity(left) and the actual velocity(right) when using observer

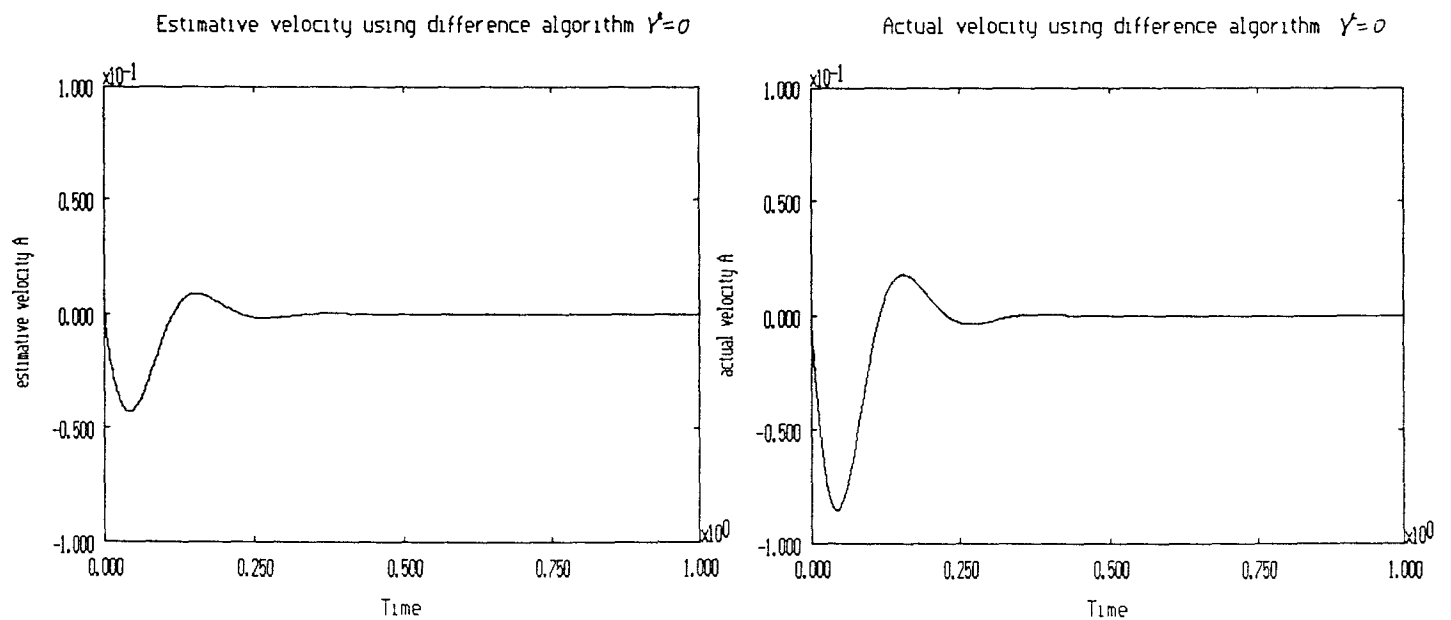


Figure 3.16: The estimated velocity(left) and the actual velocity(right) when using difference algorithm

Chapter 4

IMPLEMENTATION

In what follows, we describe the functional features of the apparatus and present the results of experiment. Some mechanical structure and properties of the experimental system and apparatus have been described in previous chapters.

4.1 Hardware

Figures 5.1 and 5.2 show the experimental setup. The frame is made of wood. The distance between two solenoids is 160mm. Two signal transfer and power amplifier circuits are located on both sides of the bottom board. A 80386 is employed.

Several kinds of position sensors are available for magnetic suspension system. Here, we use two linear potentiometers, which are very simple and cheap.

The outbound signals, which come from the two position detectors and

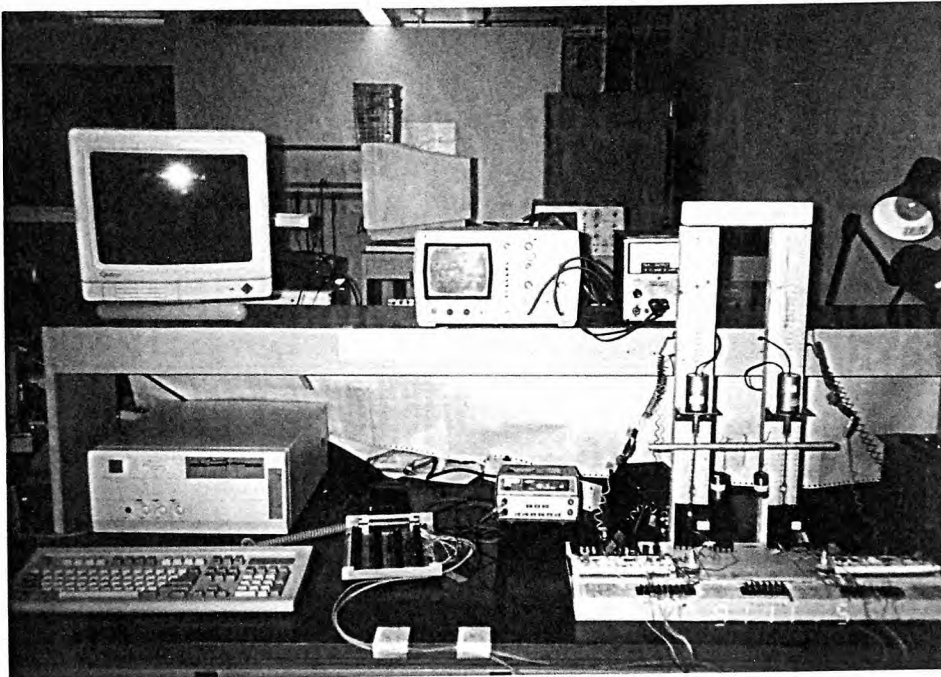


Figure 4.1: Experiment in operation

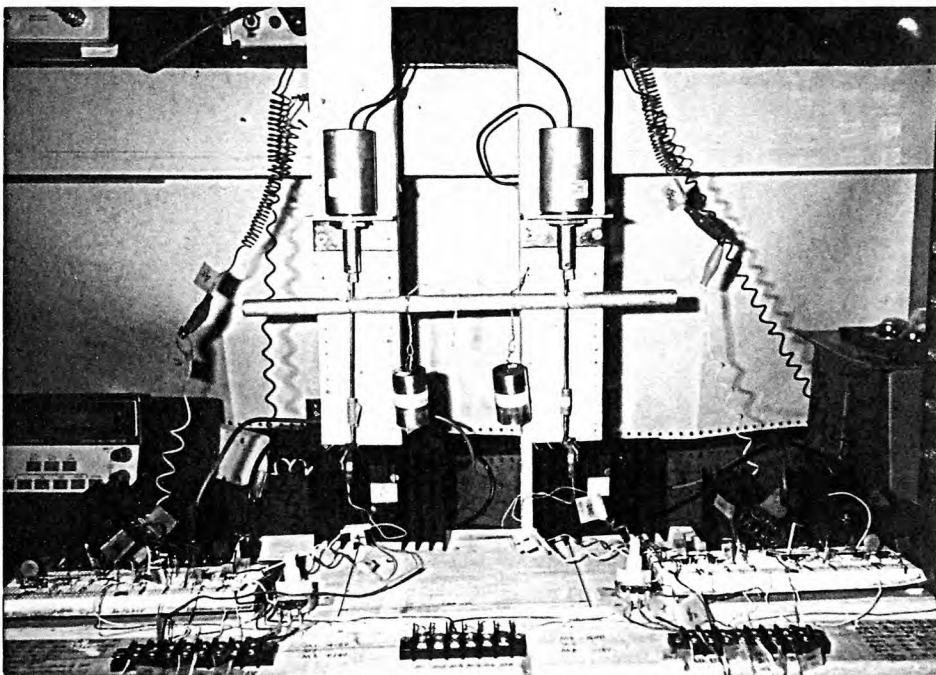


Figure 4.2: Closeup of suspended object; airgaps in solenoids about 20mm, suspended object total weight 1.16 lb.

the two reference position pots. and send to the two power amplifiers. are accessible to the computer through a IBM PC DACA card(Data Acquisition and Control Adapter). The adapter provides: four analog input channels; two analog output channels; a 16-bit digital input port; a 16-bit digital output port; a 32-bit timer; a 16-bit, externally-clocked, timer/counter. The detailed information on the operation of DACA can be found in reference manuals. Figure 5.3 shows a block diagram of DACA.

Two groups of low-pass filters are added to the both sides of the DACA. to reduce the steady-state error and long-term drift of the suspension. The filters, which are first-order, unity-gain active networks, between analog-in and the DACA possess a -3dB point of 500HZ, and between analog-out and the DACA possess a -3dB point of 1000HZ. The construction details of filters in the suspension system are available elsewhere[11].

The computer will read the two position states of suspended object through A/D0 and A/D1, read the two reference position states from A/D2 and A/D3, and send the final current control signals to D/A0 and D/A1.

The current control signals($0 \sim 10V$), through the low-pass filters, are sent to the power voltage-to-current amplifiers. The circuit design of power amplifier and experimental test are shown in Figures 4.4 and 4.5.

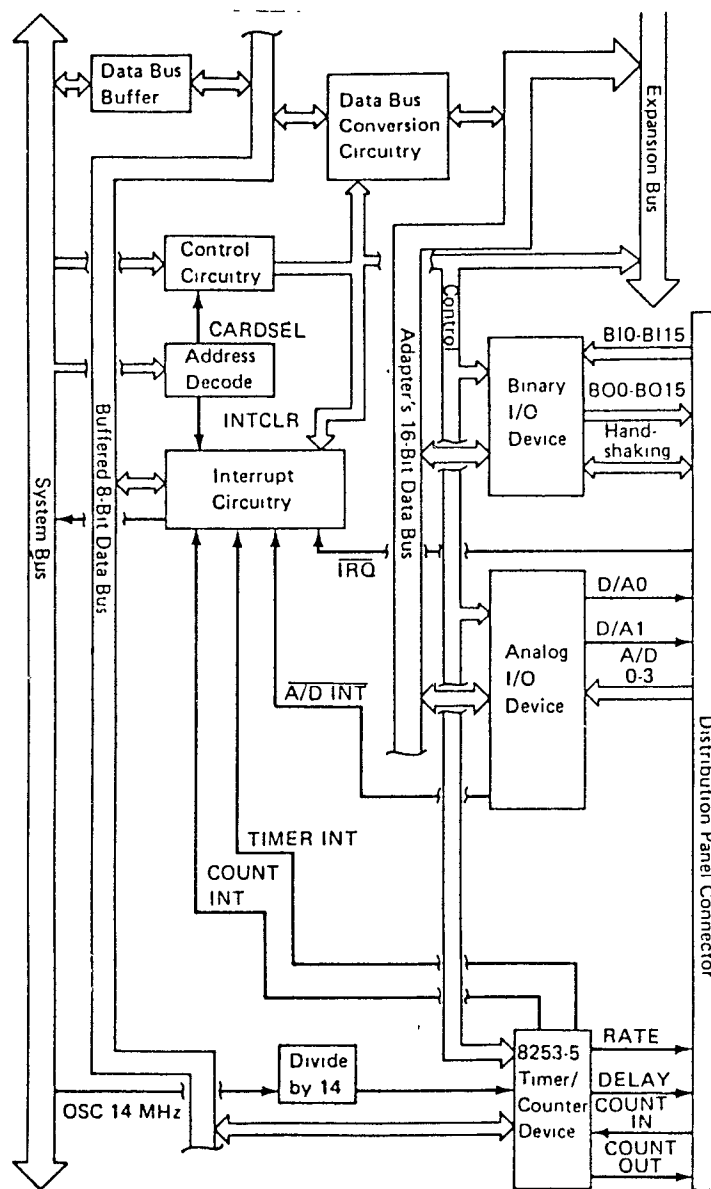


Figure 4.3: A block diagram of the Data Acquisition Adapter

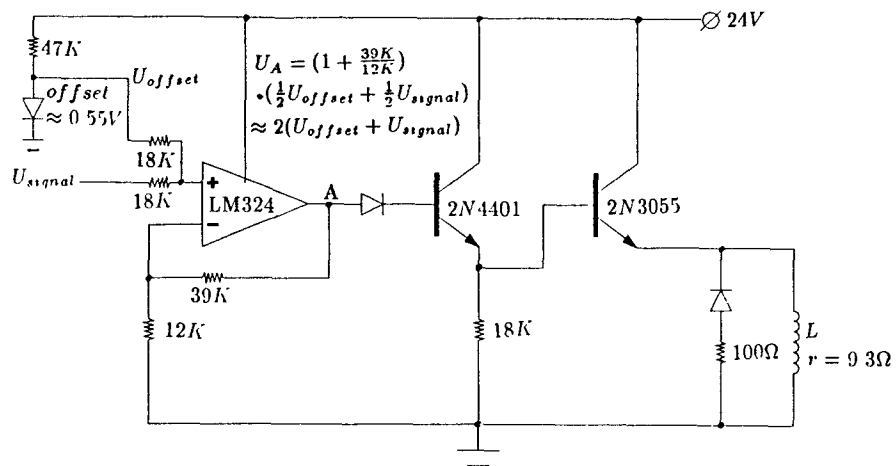


Figure 4.4: Diagram of power amplifier

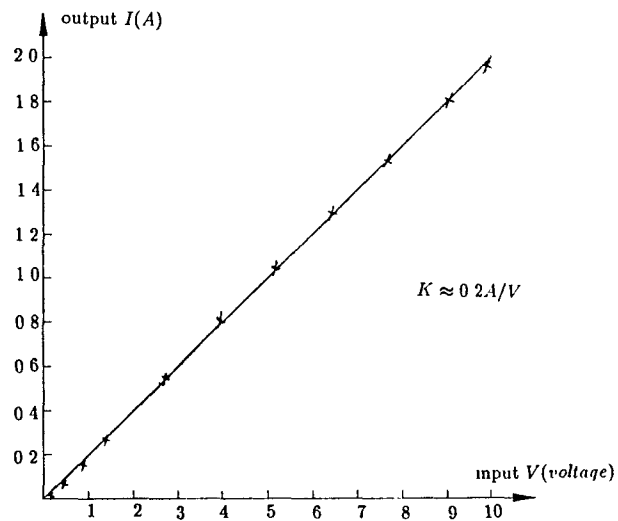
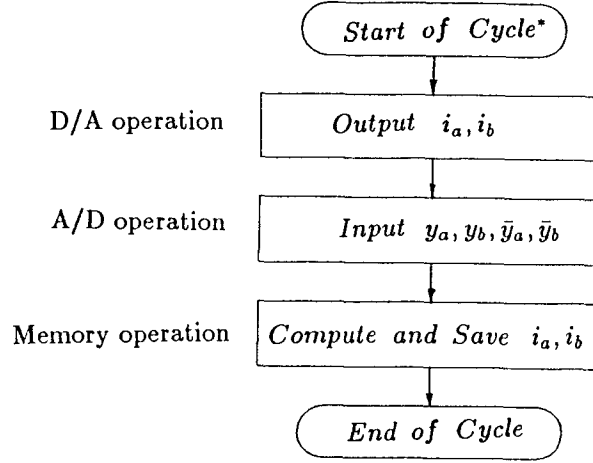


Figure 4.5: The performance of amplifier



* cycle is started by interrupt triggered by clock.

Figure 4.6: Overview flowchart of the software

4.2 Software

State-space representation translates directly into computer program for implementation. The software for this research is developed using C language. An overview flowchart of the program is shown in figure 4.6.

Figure 4.7 shows the timing diagram of the software design. This sequence minimizes time delay between beginning of the cycle and the output of two new control current values. In order for implementation to be feasible, we must have sampling time:

$$T < t_1 + t_2 + t_3 + t_4 + t_5$$

The software developed here only provides some basic functions of the experimental system. A lot of improvement to this software can be made.

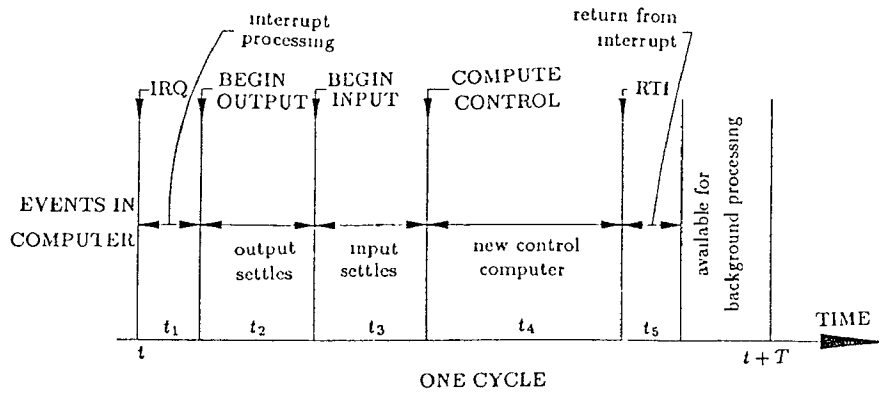


Figure 4.7: Timing diagram of the software design

The complete source code is listed in appendix C.

The software provides: entering time increment from keyboard, showing control executed times on-line, recording the experimental data for later print out, stopping control process at any time by hitting any key on the keyboard.

In the control algorithm of the source code, in order to simplify editing process and save CPU time in real-time control, we employ the difference algorithm to approximate time derivatives instead of the observer. If one wants to try the observer to estimate the velocities, which has been proved in the numerical simulation to be working very well, a subroutine of solving the differential equations of the observer should be developed to replace the difference algorithm.

The software also can be developed from many ways, such as adding on-line graphics, using keyboard inputs instead of the reference pots inputs, etc.

4.3 Performance Tests

The experimental system is set in the initial positions:

$$y_a = 25mm, \quad y_b = 25mm$$

Integral coefficient: $\gamma = -100$

Sampling frequency: 333 HZ

Other initial positions, integral coefficient and sampling frequency can also be chosen. The experimental results are shown in figures 4.8 4.10 4.12. We also present the simulated results of experimental system in figures 4.9 4.11 4.13.

Since the uncertain friction exists in our experimental system, the experimental results show clearly that the system overcomes the uncertain friction to achieve the reference positions. It shows the attribution of the position error integral feedback control. Figure 4.8 shows that the control accuracy can approach the position errors less than 0.1mm (i.e. 1%) after 3 second.

Since it is difficult to obtain the actual velocities of the suspended object in our experiment, the only way is to estimate them from the detected positions. Figure 4.10 shows that there are a lot of small ripples on the estimated velocity curves, which is the noise introduced by the difference algorithm. Figure 4.8 shows that the experimental position curves are smooth.

Figure 4.12 shows that there are also lots of small ripples on the control

current curves, which are caused by the feedback of the estimated velocities.

The comparison between the simulated results and experimental results shows that design of 2-axis magnetic suspension system is quite successful.

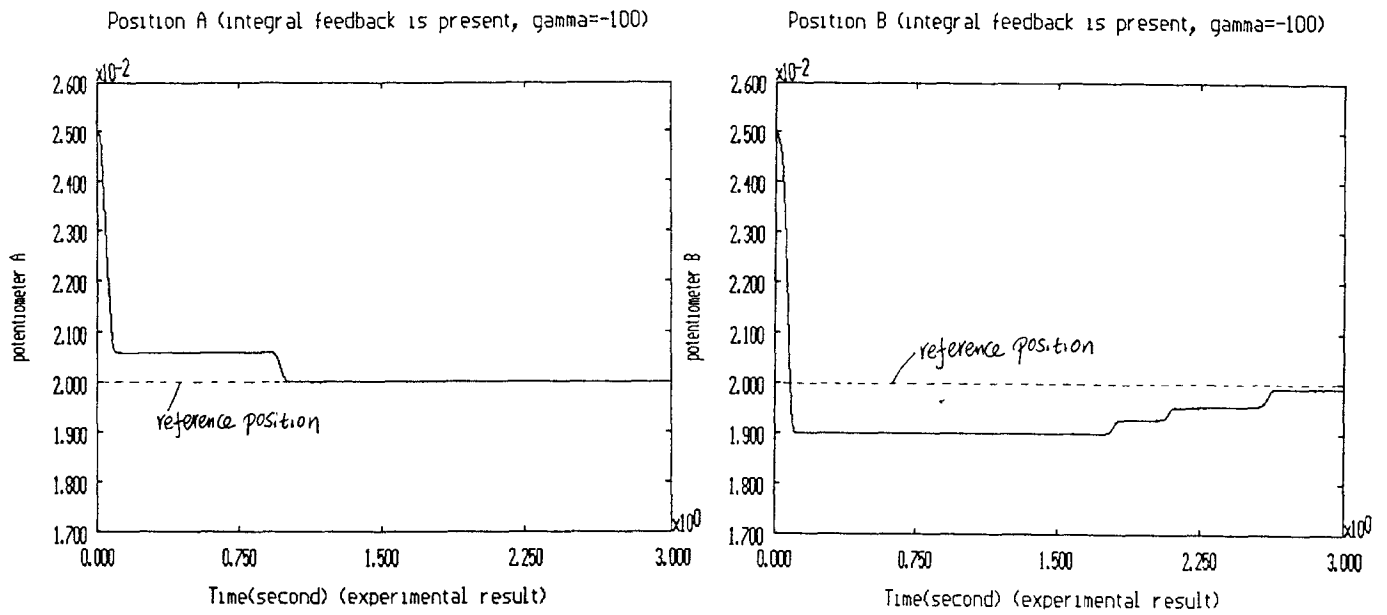


Figure 4.8: Experimental positions when integral feedback control is present

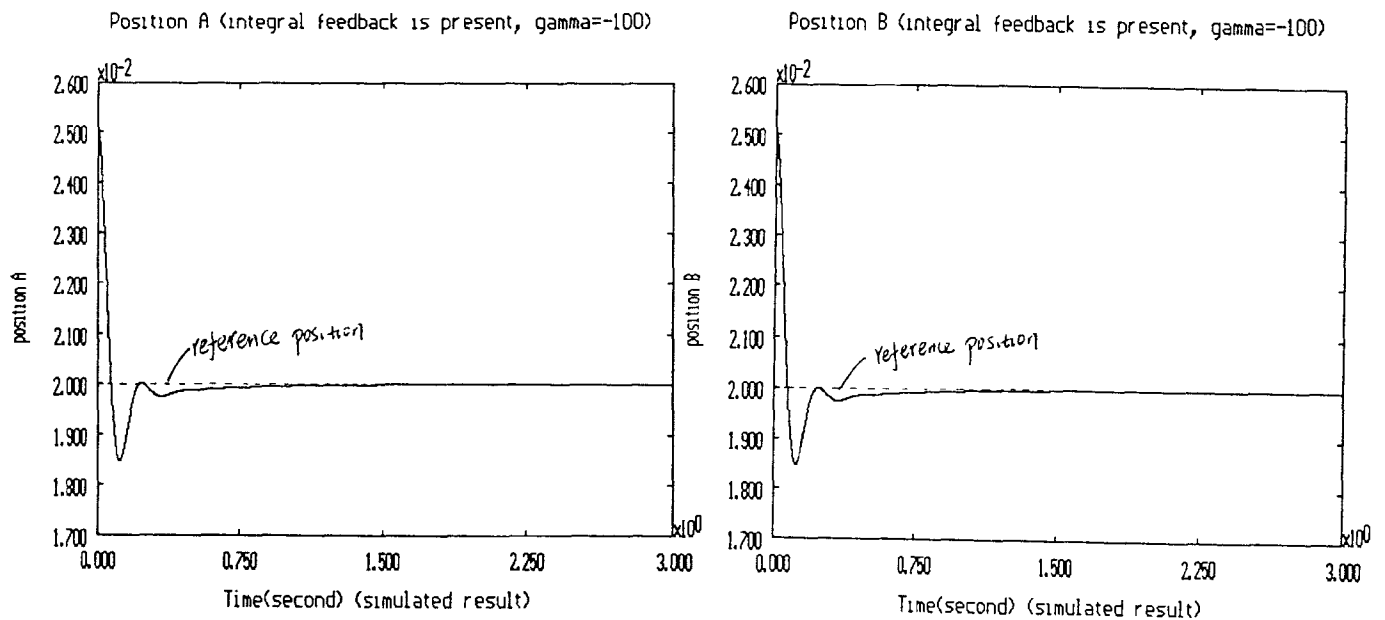


Figure 4.9: Simulated positions when integral feedback control is present

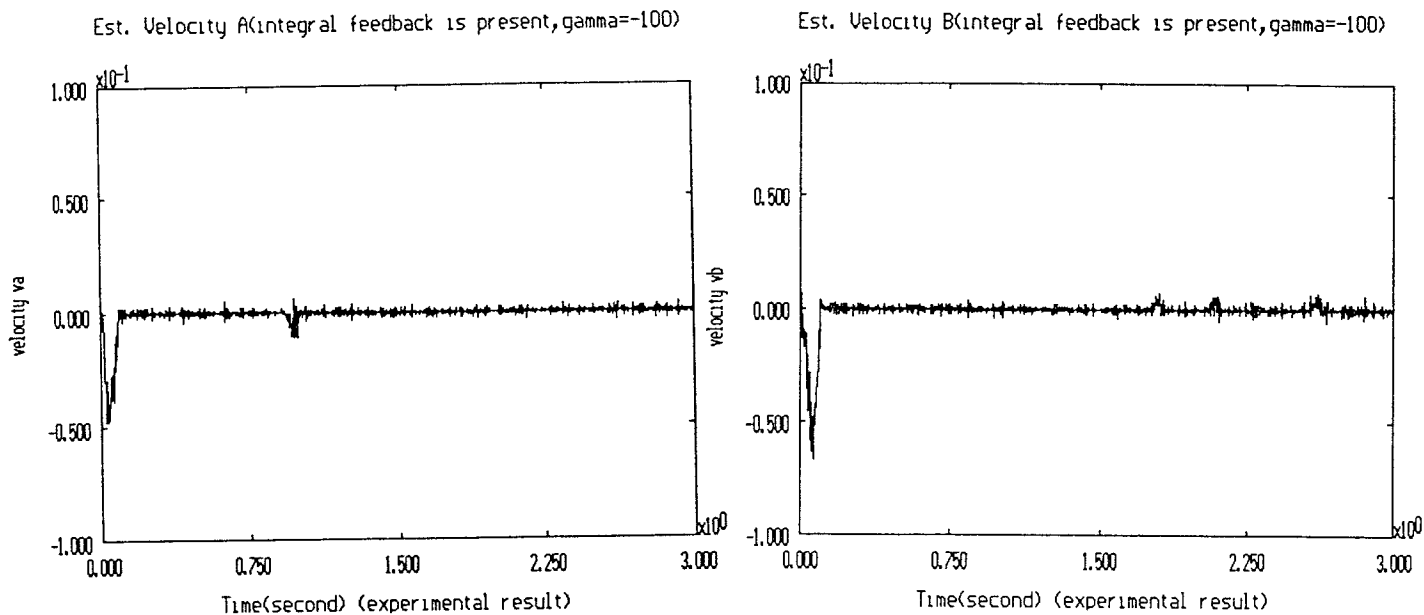


Figure 4.10: Experimental estimated velocities when integral feedback control is present

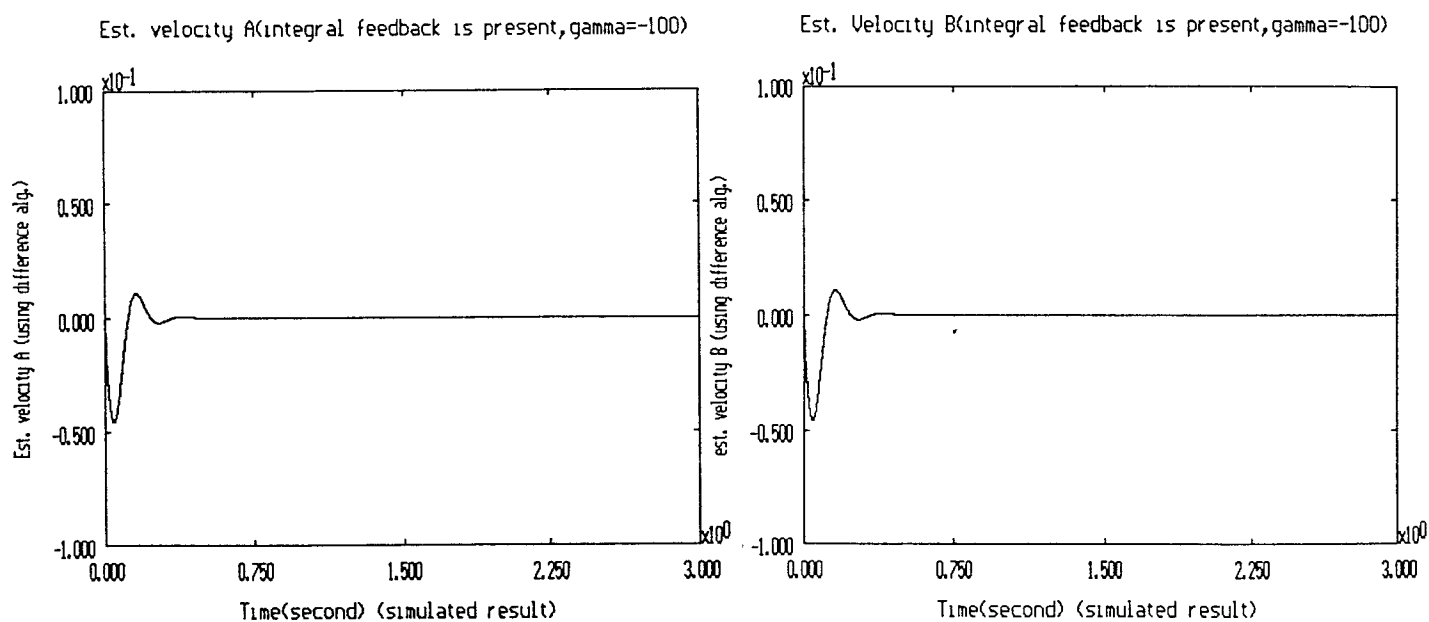


Figure 4.11: Simulated estimated velocities when integral feedback control is present

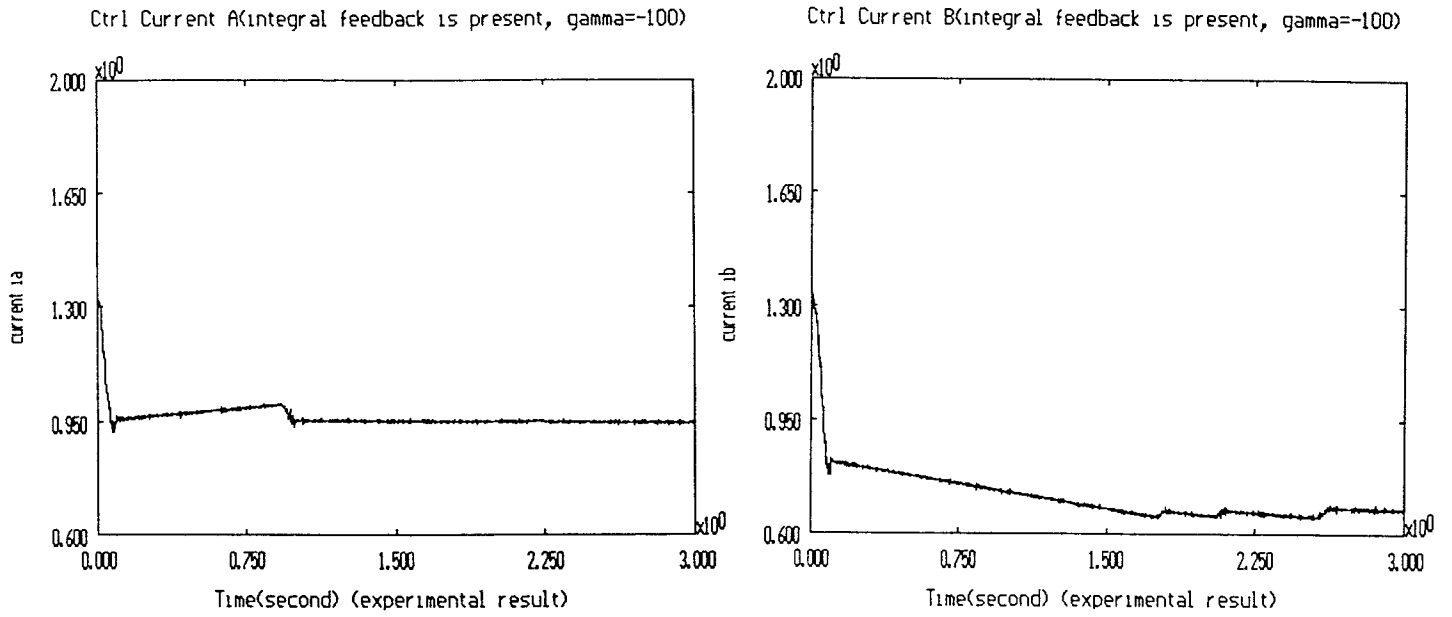


Figure 4.12: Experimental control currents when integral feedback control is present

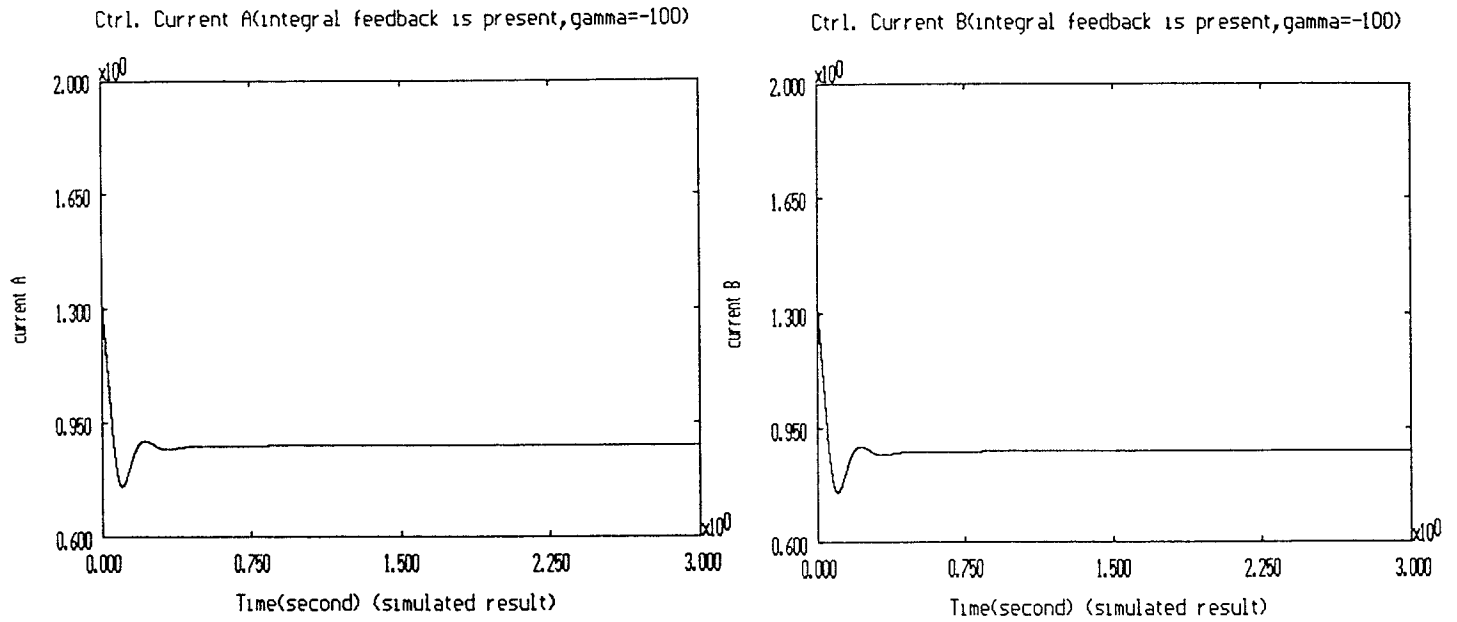


Figure 4.13: Simulated control currents when integral feedback control is present

Chapter 5

CONCLUSIONS & RECOMMENDATIONS

In this investigation, a two-degree-of-freedom magnetic suspension was designed, breadboarded, and successfully tested. There is room for improvement, however. To improve the magnetic suspension experiment will involve changes in the control algorithm, hardware, and software.

With regard to the algorithm, the nonlinear reduced-order observer can be extended for estimating the uncertain friction when the core of the solenoids and the bar of potentiometer move. Theoretically, the friction can be estimated and cancelled.

With regard to software, the control program can be implement the nonlinear observer. In addition, the reference position can be entered at keyboard instead of by setting of potentiometer.

With regard to hardware, the linear potentiometer can be replaced by better sensors (e.g., LVDT, optical, etc.), and a more permanent apparatus can be constructed.

Bibliography

- [1] Carmichael, A.T. and Hinchliffe, S. "Magnetic Suspension Systems with Digital Controller" IEEE, Rev. Sci. Instrum. 57(8), August 1986.
- [2] Defrancesco, S. and Zanetti, V. "Experiment on Magnetic Repulsion" Am. J. Phys. 51(11), November 1983.
- [3] Dunn, John "Big Lift for Magnetic Suspension" The Engineer, 10 January 1985.
- [4] Ethridge, E.C. and Curreri, P.A. "Technique for the Efficient and Reproducible Fabrication of Electromagnetic Levitation Coils" IEEE, Rev. Sci. Instrum. 55(11), November 1984.
- [5] Frazier, Richard H. and Philip J. Gilinson, Jr. "Magnetic and Electric Suspension" MIT, 1974.
- [6] Friedland, Bernard *Control System Design: An Introduction to State Space Methods* New York: McGraw-Hill, 1986.
- [7] Friedland, Bernard *Advanced Control System Design* In preparation.

- [8] Fong, Paidi *Technique of Computer Control* Zhejiang University, China, 1990
- [9] Jayawant, B.V. *Electromagnetic Levitation and Suspension Techniques* Edward Arnold Ltd, 1981.
- [10] Jayawant, B.V. "Electromagnetic Suspension and Levitation" Rep. Prog. Phys., Vol. 44, 1981.
- [11] Lawson, M.A. and Gillies, G.T. "Interrupt-Drive Digital Controller for A Magnetic Suspension System" IEEE, Rev. Sci Instrum. 60(3), March 1989.
- [12] Murgatroyd, P.N. and Carmichael, A.T. "Improved Differential Optical Position Detector for Magnetic Suspension System" IEEE, Rev. Sci. Instrum. 56(8), August 1985.
- [13] Ogata, K. *Modern Control Engineering* Prentice-Hall Inc., Englewood Cliffs, NJ, 1970.
- [14] Perez, C.C. " A Feedback System for a Stable Magnetic Suspension in Air" Thesis, MIT.
- [15] Scudiere, M.B. and Willems, R.A. "Digital Controller for a Magnetic Suspension System" IEEE, Rev. Sci. Instrum. 57(8), August 1986.
- [16] Shiota, F. and Nakayama, K. "Improvement of the Superconducting Magnetic Levitation System for the Determination of the Magnetic

Flux Quantum" IEEE, Trans. on Instrum. and Measur., Vol. 38, April 1989.

- [17] Sinha, P.K. *Electromagnetic Suspension Dynamics & Control* Peter Peregrinus Ltd., 1987.
- [18] Sridharan, G. "Optical Reflectance Sensor Measures Displacement in Magnetic Suspension" IEEE, Tran. on Instrum. and Measur., Vol. IM-34, March 1985.
- [19] Sridharan, G. "Simple Calibrator for Optical Reflective Sensor" IEEE, Rev. Sci. Instrum. 54(10), October 1983.
- [20] Sridharan, G. "Versatile Damper for Magnetic Suspension Systems" IEEE, Rev. Sci. Instrum. 55(5), May 1984.
- [21] Sridharan, G. "Improved Active Magnetic Suspension" IEEE, Rev. Sci. Instrum. 54(10), October 1983.
- [22] Upton, S.R. and Murgatroyd, P.N. "Magnetic Levitation with Thyristor Control" IEEE, Rev. Sci. Instrum. 52(1), January 1981.

APPENDIX A

Presented here are the code and data for simulation of continuous-time control of 2-axis magnetic suspension by using ALSIM.

```

/*
**      2-Axis Magnetic Suspension Model
*/

#include "\ALSIM\ALSIM.H"

#define g      fpar[1]
#define alpha  fpar[2]
#define beta   fpar[3]
#define YA     fpar[4]
#define YB     fpar[5]
#define g11    fpar[6]
#define g12    fpar[7]
#define g13    fpar[8]
#define g14    fpar[9]
#define g21    fpar[10]
#define g22    fpar[11]
#define g23    fpar[12]
#define g24    fpar[13]
#define k11    fpar[14]
#define k12    fpar[15]
#define k21    fpar[16]
#define k22    fpar[17]

derv(t, x, dxdt)
double t, *x, *dxdt;
{
/*-----Control    Law-----*/

u[1] = sqrt(-g/(alpha + beta))*YA
      + g11*(YA-x[1]) + g12*(YB-x[2])
      + g13*(-x[7]) + g14*(-x[8]);

u[2] = sqrt(-g/(alpha + beta))*YB
      + g21*(YA-x[1]) + g22*(YB-x[2])
      + g23*(-x[7]) + g24*(-x[8]);

/*-----Plant    Dynamic-----*/

y[1] = x[1];
y[2] = x[2];

dxdt[1] = x[3];
dxdt[2] = x[4];

dxdt[3] = alpha*u[1]*u[1]/(x[1]*x[1])
          + beta*u[2]*u[2]/(x[2]*x[2]) + g;

dxdt[4] = beta*u[1]*u[1]/(x[1]*x[1])
          + alpha*u[2]*u[2]/(x[2]*x[2]) + g;

/*-----Nonlinear Reduced-Observer-----*/

```

```

dxdt[5] = alpha*u[1]*u[1]/(x[1]*x[1])
          + beta*u[2]*u[2]/(x[2]*x[2])
          + g - k11*x[7] -k12*x[8];

dxdt[6] = beta*u[1]*u[1]/(x[1]*x[1])
          + alpha*u[2]*u[2]/(x[2]*x[2])
          + g - k21*x[7] -k22*x[8];

x[7] = x[5] + x[1]*k11 + x[2]*k12;

x[8] = x[6] + x[1]*k21 + x[2]*k22;
}

```



```

;Continuous-Time Control of 2-Axis Magnetic Suspension
0.0      ;initial time
2.0      ;final time
0.01     ;maximum stepsize
1.0e-3   ;minimum stepsize
0.001    ;fractional error criterion

80       ;multiple of maximum stepsize for print output
1        ;multiple of maximum stepsize for plot output

8        ;number of plant states
2        ;number of plant inputs
2        ;number of plant outputs
0        ;number of controller states

0        ;size of user defined plot vector
0        ;size of user common area
0        ;size of gaussian random number vector
         ;vector multiplied by sqrt(hmax) to provide approx. uniform
         ;variance for variable stepsize
318      ;random number seed
272      ;random number seed
190      ;random number seed

0        ;number of user defined integer input parameters
0,0      ;end integer input parameters

17       ;number of user defined floating point input parameters
1,9.8    ;g
2,-7.33e-3 ;alpha
3,2.275e-3 ;beta
4,0.02    ;YA
5,0.02    ;YB
6,-87.976 ;g11
7,7.1919e-2 ;g12
8,-2.4241 ;g13
9,-0.38374 ;g14
10,7.311e-2 ;g21
11,-87.977 ;g22
12,-0.38371 ;g23
13,-2.4241 ;g24
14,2.121   ;k11
15,2.121   ;k12
16,2.121   ;k21
17,2.121   ;k22
0,0        ;end floating point input parameters

1,0.017   ;initial position of x[1] i.e. ya
2,0.023   ;initial position of x[2] i.e. yb
5,-0.0848 ;initial state of observer i.e. x[5]
6,-0.0848 ;initial state of observer i.e. x[6]
0,0       ;end plant initial conditions

```

```
0,0      ;end controller initial conditions
```

APPENDIX B

Presented here are the code and data for simulation of discrete-time control of 2-axis magnetic suspension by using ALSIM.

```

/*
** Discrete-Time Control of 2-Axis Magnetic Suspension
*/

#include "\ALSIM\ALSIM.H"

#define g      fpar[1]
#define alpha  fpar[2]
#define beta   fpar[3]
#define YA     fpar[4]
#define YB     fpar[5]

derv(t, x, dxdt)
double t, *x, *dxdt;
{
/*-----Plant    Dynamic-----*/

y[1] = x[1];
y[2] = x[2];

dxdt[1] = x[3];
dxdt[2] = x[4];

dxdt[3] = alpha*u[1]*u[1]/(x[1]*x[1])
          + beta*u[2]*u[2]/(x[2]*x[2]) + g;

dxdt[4] = beta*u[1]*u[1]/(x[1]*x[1])
          + alpha*u[2]*u[2]/(x[2]*x[2]) + g;
}

```

```

/*
** Discrete-Time Control of 2-Axis Magnetic Suspension
*/

#include "\ALSIM\ALSIM.H"
#define g      fpar[1]
#define alpha  fpar[2]
#define beta   fpar[3]
#define YA     fpar[4]
#define YB     fpar[5]
#define g11    fpar[6]
#define g12    fpar[7]
#define g13    fpar[8]
#define g14    fpar[9]
#define g21    fpar[10]
#define g22    fpar[11]
#define g23    fpar[12]
#define g24    fpar[13]
#define k11    fpar[14]
#define k12    fpar[15]
#define k21    fpar[16]
#define k22    fpar[17]
#define T      fpar[18]
#define gamma  fpar[19]

void control(t)
double t;
{
    static double
    tnext = 0,
    errA = -0.005,
    errB = -0.005;

    static double
    YA3=0.025,
    YA2=0.025,
    YA1=0.025,
    YB3=0.025,
    YB2=0.025,
    YB1=0.025,
    IEA=0.,
    IEB=0.,
    IA1=-0.005,
    IA2=-0.005,
    IB1=-0.005,
    IB2=-0.005;

    if(t >= tnext)
    {

/*-----Estimate of Velocities-----*/

YA3 = YA2;
YA2 = YA1;

```

```

YA1 = x[1];
YB3 = YB2;
YB2 = YB1;
YB1 = x[2];

c[1] =(YA1-1.5*YA2+0.5*YA3)/T;
c[2] =(YB1-1.5*YB2+0.5*YB3)/T;

/*-----Calculate The Integral Term-----*/

errA = YA - x[1];
errB = YB - x[2];

IA1=IA2;
IA2=errA;
IB1=IB2;
IB2=errB;

IEA = IEA + (IA1 + IA2)*T/2;
IEB = IEB + (IB1 + IB2)*T/2;

/*-----Control Law-----*/

u[1] = sqrt(-g/(alpha + beta))*YA
      + g11*(YA-x[1]) + g12*(YB-x[2])
      + g13*(-c[1]) + g14*(-c[2]) + gamma*IEA;

u[2] = sqrt(-g/(alpha + beta))*YB
      + g21*(YA-x[1]) + g22*(YB-x[2])
      + g23*(-c[1]) + g24*(-c[2]) + gamma*IEB;

tnext += T;
    }
}

```

```

;Discrete-Time Control of 2-Axis magnetic suspension
0.0      ;initial time
2.0      ;final time
0.001    ;maximum stepsize
1.0e-6   ;minimum stepsize
0.001    ;fractional error criterion

80       ;multiple of maximum stepsize for print output
1        ;multiple of maximum stepsize for plot output

4        ;number of plant states
2        ;number of plant inputs
2        ;number of plant outputs
2        ;number of controller states

0        ;size of user defined plot vector
0        ;size of user common area
0        ;size of gaussian random number vector
         ;vector multiplied by sqrt(hmax) to provide approx. uniform
         ;variance for variable stepsize
318      ;random number seed
272      ;random number seed
190      ;random number seed

0        ;number of user defined integer input parameters
0,0      ;end integer input parameters

19       ;number of user defined floating point input parameters
1,0.8    ;g
2,-7.33e-3 ;alpha
3,2.275e-3 ;beta
4,0.02    ;YA
5,0.02    ;YB
6,-87.976 ;g11
7,7.1919e-2 ;g12
8,-2.4241 ;g13
9,-0.38374 ;g14
10,7.311e-2 ;g21
11,-87.977 ;g22
12,-0.38371 ;g23
13,-2.4241 ;g24
14,2.121   ;k11
15,2.121   ;k12
16,2.121   ;k21
17,2.121   ;k22
18,0.003   ;T
19,-100.   ;gamma
0,0        ;end floating point input parameters

1,0.025    ;initial position of x[1] i.e. ya
2,0.025    ;initial position of x[2] i.e. yb
0,0        ;end plant initial conditions

1,0.0      ;initial state of c[1] i.e. hat{yav}

```

2,0.0
0,0

```
;initial state of c[2] i.e. hat{ybv}  
;end controller initial conditions
```


APPENDIX C

This appendix presents the source code listings for the controller software used in this research. The code is written in Turbo C++.

```
/*
```

```
=====
```

```
CONTROL PROGRAM:
```

```
2-Axis Magnetic Suspension System
```

```
=====
```

```
*/
```

```
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <sys\timeb.h>
#include "dacamu.h"
```

```
#define TWELVEBITS 4096.0
#define OUTSCALE 4095
#define g 9.8
#define alpha -7.33e-3
#define beta 2.275e-3
#define g11 -87.976
#define g12 7.1919e-2
#define g13 -2.4241
#define g14 -0.38374
#define g21 7.311e-2
#define g22 -87.977
#define g23 -0.38371
#define g24 -2.4241
#define r -100.0
```

```
char test;
```

```
int cflag = 0;
```

```
unsigned long
    nc = 0;
```

```
double T;
```

```
double
    YA3=0.025,
    YA2=0.025,
    YA1=0.025,
    YB3=0.025,
    YB2=0.025,
    YB1=0.025,
    IEA=0.,
    IEB=0.,
    IA1=-0.005,
    IA2=-0.005,
    IB1=-0.005,
    IB2=-0.005.,
    ya=0.,
    yb=0.,
```

```

    yas=0.,
    ybs=0.,
    ia=0.,
    ib=0.,
    yav=0.,
    ybv=0.;

static float
    tp[1024],
    yap[1024],
    ybp[1024],
    yasp[1024],
    ybsp[1024],
    iap[1024],
    ibp[1024],
    yavp[1024],
    ybvp[1024],
    errap[1024],
    errbp[1024];

/***** CONTROL LAW AS AN INTERRUPT SERVICE ROUTINE *****/
void
MSA(void)
{
    register int i;

static unsigned
    currentA=0,
    currentB=0,
    positionA=0,
    positionB=0,
    setpointA=0,
    setpointB=0;

static double
    ttime = 0.,
    ia = 0.,
    ib = 0.,
    errA=-0.005,
    errB=-0.005;

    positionA = AnalogRead(0);
    positionB = AnalogRead(1);
    setpointA = AnalogRead(2);
    setpointB = AnalogRead(3);

    AnalogWrite(0, currentA);
    AnalogWrite(1, currentB);

    ya = ((double)positionA/TWELVEBITS + 1.5)/100;
    yb = ((double)positionB/TWELVEBITS + 1.5)/100;
    yas = ((double)setpointA/TWELVEBITS + 1.5)/100;
    ybs = ((double)setpointB/TWELVEBITS + 1.5)/100;

```

```

        ttime += T;          /* update real time */

        errA=yas-ya;
        errB=ybs-yb;        /*calculate errors */

/***** Control Algorithm *****/

YA1 = YA2;
YA2 = YA3;
YA3 = ya;
YB1 = YB2;
YB2 = YB3;
YB3 = yb;

IA1=IA2;
IA2=errA;
IB1=IB2;
IB2=errB;

yav = (YA3 - 1.5*YA2 + 0.5*YA1)/T; /* estimate velocities */
ybv = (YB3 - 1.5*YB2 + 0.5*YB1)/T;

IEA += (IA1 + IA2)*T/2;           /* calculate integral */
IEB += (IB1 + IB2)*T/2;           /* terms */

ia = sqrt(-g/(alpha + beta))*yas
    + g11*(yas - ya) + g12*(ybs - yb)
    + g13*(-yav) + g14*(-ybv) + r*IEA;
ib = sqrt(-g/(alpha + beta))*ybs
    + g21*(yas - ya) + g22*(ybs - yb)
    + g23*(-yav) + g24*(-ybv) + r*IEB;

/***** End of Control Algorithm *****/

currentA =(unsigned)( TWELVEBITS*ia/2);           /* Scale output */
if (currentA > OUTSCALE) currentA = OUTSCALE;      /* Limiter */
else if (currentA < 0) currentA = 0;

currentB =(unsigned)( TWELVEBITS*ib/2);           /* Scale output */
if (currentB > OUTSCALE) currentB = OUTSCALE;      /* Limiter */
else if (currentB < 0) currentB = 0;

if(nc<1024)                                       /* Put first 1024 records */
{
    tp[nc]=(float)(ttime);                       /* into array for later */
    yap[nc]=(float)(ya);                          /* printout. */
    ybp[nc]=(float)(yb);
    yasp[nc]=(float)(yas);
    ybsp[nc]=(float)(ybs);
    iap[nc]=(float)(ia);
    ibp[nc]=(float)(ib);
    yavp[nc]=(float)(yav);

```

```

        ybvp[nc]=(float)(ybv);
        errap[nc]=(float)(errA);
        errbp[nc]=(float)(errB);
    }

    nc++;                                /* Update step counter */
}
/***** BACKGROUND PROGRAM *****/
void
main(void)
{
    FILE *out;

    register int i;

    unsigned Interrupt_level = 7;

    AnalogWrite(0, 0);
    AnalogWrite(1, 0);

    for(i=0; i<1024; i++)                /* Zero out array (for insurance) */
    {
        tp[i]=0.;
        yap[i]=0.;
        ybp[i]=0.;
        yasp[i]=0.;
        ybsp[i]=0.;
        iap[i]=0.;
        ibp[i]=0.;
        yavp[i]=0.;
        ybvp[i]=0.;
        errap[i]=0.;
        errbp[i]=0.;
    }

    clrscr();

    cprintf("\r\n 2-axis magnetic suspension control \r\n" );
    printf("Enter time increment: ");
    scanf("%lf", &T);
    printf("\n\n\n");

    SetIntLevel(Interrupt_level);
    EnableISR(MSA, TIMER, T);           /* Start real-time control */

    while(!bioskey(1)) cprintf("\n\r %ld", nc);

    DisableISR();                       /* Stop real-time control */
    enable();
    AnalogWrite(0, 0);
    AnalogWrite(1, 0);

    cprintf("\r\n\r\n\r\n\r\nControl executed %ld times\r\n", nc);
    cscanf("%x", &test);

```

```

out = fopen("msa.out", "w+b");

for(i=0; i<min(nc,1024); i++)
{
    fwrite(&tp[i],4,1,out);
    fwrite(&yap[i],4,1,out);
    fwrite(&ybp[i],4,1,out);
    fwrite(&yasp[i],4,1,out);
    fwrite(&ybsp[i],4,1,out);
    fwrite(&iap[i],4,1,out);
    fwrite(&ibp[i],4,1,out);
    fwrite(&yavp[i],4,1,out);
    fwrite(&ybvp[i],4,1,out);
    fwrite(&errap[i],4,1,out);
    fwrite(&errbp[i],4,1,out);
}
fclose(out);
cprintf("\r\n msa.out file written");
exit;
}

```