

ECE4560

DIGITAL CONTROL LABORATORY

Fall 2002

Discrete-time control systems will be designed and tested using microcomputers, compensators, A/D and D/A converter analog computers. Experiments in the control of discrete and analog systems will be performed. Coreq., ECE4540.

Instructor: TBD

Office: TBD

Phone: TBD

email: TBD

Course web-page: <http://mocha-java.uccs.edu/ECE4560/>

Office Hours: TBD

Summary: This course deals with practical aspects of control engineering. It is intended as a companion course for ECE4540 (Digital Control Systems), and serves to augment and demonstrate concepts presented in the classroom.

Matlab/Simulink: You will use Matlab[®] and Simulink[®] extensively. Prior familiarity with Matlab and Simulink is assumed. You are not required to purchase these packages. Matlab and Simulink run on the control-systems lab computers and on the ECE multimedia room computers.

Grading: 10% of your course grade will depend on your prelab assignments and 90% of your course grade will depend on your lab reports. No lab report will be given a grade better than “F” until you turn in an approved prelab assignment for that lab. See further note regarding grading on the back of this page.

Lab Syllabus

Laboratory Orientation

1. Discrete-time simulation with Simulink.

(Free lab period to write up formal report)

Unit 1: Design by Emulation

2. Time-domain controller emulation.
3. Frequency-domain controller emulation.

(Free lab period to write up formal report)

Unit 2: Digital Effects

4. Sampling, aliasing, zero-order hold.
5. Discrete-time plant modeling.
6. Filter structure and finite-precision effects.

(Free lab period to write up formal report)

Unit 3: Transfer Function Controller Design

7. Frequency-response controller design.
8. Numeric optimal PID controller design.
9. Ragazzini’s direct control design method.

(Free lab period to write up formal report)

Unit 4: State-Space Controller Design

10. State-feedback controller design.
11. State estimation and control design.

(Formal report due before end of exam week)

Work Load: This is an aggressive lab course requiring weekly assignments. On average, expect to spend 3 to 4.5 hours per week *outside* of the lab preparing for the labs and completing lab writeups. This is in accord with UCCS policy relating credit hours for a laboratory course to student workload. Some students will find that more time is required, while others will find that less time is required.

Grading: Due to space and equipment limitations, the labs must be performed in groups of two or three students per group. Your grade will be largely determined by the quality of your report. The reports will be graded based on organization, technical accuracy, neatness, grammar and spelling. If a lab-group submits a single report, then each student will receive the same grade for the lab report. If each student in a lab-group submits his or her own report, each student will receive an individual grade.

Lab Reports: The Department requires formal lab reports which must satisfy the following format rules:

1. Title page: This must include a title, group member names, course and section name and date.
2. Introduction: Explain the background and objective of the lab indicating requirements and desired results.
3. Discussion: Discuss the underlying applicable theory and concepts that support the measurements. Indicate and discuss the measurement set-up and equipment used.
4. Measurement data and/or Results: Present measurement results in tabular, graphical or numeric form. Present results from required lab exercises.
5. Discussion of Measurements: Discuss measured data in context of comparison to expectation, accuracy, difficulties, etc.
6. Summary and Conclusions: Discuss findings, explain errors and unexpected results; summarize and indicate conclusions.

Further requirements on the lab report are:

1. Correct spelling, grammar and punctuation is required.
2. Report must be typed; figures, drawings and equations may be handwritten.
3. Format of references must conform to IEEE (transactions) standards.

Prelab Assignments: Before you enter the laboratory each week, you are expected to thoroughly read and understand the corresponding lab writeup in this lab reader. Each lab has a prelab assignment. This assignment must be completed and turned in to the lab instructor before you will be permitted to start the lab. Each of your prelab assignments must contain:

- Answers to the specific questions given in the “Prelab Assignment” portion of the lab reader for that lab,
- A paragraph describing what you are supposed to do during the lab,
- A paragraph describing expected results.

A portion of your course grade will be assigned based on your prelab assignments. Furthermore, no grade better than an “F” is possible for any lab report until an approved prelab assignment has been turned in for that lab.

Missed Labs: Attendance is your responsibility. Missed labs will count as ZERO without a physician’s documentation of an illness, or other appropriate documentation of an emergency beyond your control and requiring your absence.

Please Note: The instructor reserves the right to change the syllabus and to add or delete lab assignments depending on circumstances. Changes will be announced ahead of time, and handouts will be given out at least one class ahead of the lab.

The Control-Systems Lab and the Lab Reader This material is based upon work supported by the National Science Foundation under Grant No. 9981009. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

The lab reader has been composed using the LyX document processing system and typeset with L^AT_EX₂ ϵ . All original diagrams have been created using either xfig or Matlab. Screen shots were made using xv.

The lab reader has been prepared by Dr. Gregory Plett; however, much has been taken from the manual for the *Model 730 Magnetic Levitation* unit by Educational Control Products (ECP), <http://www.ecpsystems.com>, and the manual for the *Real Time Linux Target (RTLTL)* by Quality Real Time Systems (QRTS), <http://www.qrts.com>. Text and diagrams have been copied from these manuals. Scanned photos from the ECP web site have also been included. All are used with permission of ECP and QRTS.

Discrete-Time Simulation with Simulink

1.1 INTRODUCTION

A fundamental aspect of digital control systems is that they operate in discrete time rather than continuous time. In this lab period you will discover how to implement and simulate discrete-time systems in Simulink. This will be important as the semester progresses since you will use Simulink to control the Lab's Magnetic Levitation (MagLev) systems using digital control methods.

1.2 THIS LAB IS...

During this lab period, two issues will be addressed:

- Orientation: A quick overview of the syllabus and expectations for lab reports.
- Discrete-Time Simulation: How to use Simulink to perform discrete-time simulation.

1.3 ORIENTATION

1.3.1 Prerequisites

The prerequisite for this lab is *ECE4510: Feedback Control Systems*. This is implied by the corequisite *ECE4540: Digital Control Systems*. Specifically, the *ECE4530: Feedback Control Laboratory* is **not** a prerequisite for this lab course. However, if you have not taken ECE4530, you will be at a disadvantage. Please read (very carefully) Apps. A and B. A thorough understanding of the material in App. B is critical.

1.3.2 The Lab Reader

This lab reader¹ contains eleven labs. The first is a Simulink-only lab. The remaining labs are divided up into four "units". The first unit focuses on digital design via emulating a continuous-time design; the second unit addresses digital effects not present in continuous-time design; the third unit concentrates on control design using transfer-function methods; and the fourth unit looks at state-space models and design. Each lab has a detailed discussion in this lab reader, and a consistent format is used throughout:

Introduction: A short introduction to the lab experiment.

This Lab Is...: A summary of the main points of the lab.

¹The reader has been entered using the LyX document processing system and typeset with L^AT_EX_{2 ϵ} on a Pentium class computer running the Linux operating system. All original diagrams have been created using either xfig or Matlab. Screen shots were made using xv.

The lab reader has been prepared by Dr. Gregory Plett; however, much has been taken from the manual for the *Model 730 Magnetic Levitation* unit by Educational Control Products (ECP), <http://www.ecpsystems.com>, and the manual for the *Real Time Linux Target* (RTLTL) by Quality Real Time Systems (QRTS), <http://www.qrts.com>. Text and diagrams have been copied from these manuals. Scanned photos from the ECP web site have also been included. All are used with permission of ECP and QRTS.

Background: Theory and practical information required to perform the lab.

Prelab assignment: An assignment which *must* be completed and turned in before you will be permitted to start the lab.

Laboratory Experiment: The experiment to be performed during the lab period.

Assignment: Items which must be included in the formal lab writeup.

1.3.3 Prelab Assignments

Before you enter the laboratory each week, you are expected to thoroughly read and understand the corresponding lab writeup in this lab reader. Each lab has a prelab assignment. This assignment must be completed and turned in to the lab instructor before you will be permitted to start the lab. Each of your prelab assignments must contain:

- Answers to the specific questions given in the “Prelab Assignment” portion of the lab reader for that lab,
- A paragraph describing what you are supposed to do during the lab,
- A paragraph describing expected results.

A portion of your course grade will be assigned based on your prelab assignments. Furthermore, no grade better than an “F” is possible for any lab report until an approved prelab assignment has been turned in for that lab.

1.3.4 Lab Reports

A syllabus of lab topics is listed on the front page of this lab reader. You will see that the semester is divided into four main units which each contain a number of labs. The report for Lab 1 is due at the beginning of lab period 2; the reports for Labs 2 and 3 are due at the beginning of lab period 4; the reports for Labs 4–6 are due at the beginning of lab period 7; the reports for Labs 7–9 are due at the beginning of lab period 10; the reports for Labs 10–11 are due before the end of finals week.

The Department requires formal lab reports. The format for formal lab reports must comply to the following:

1. Title page: This must include a title, group member names, course and section name and date.
2. Introduction: Explain the background and objective of the lab indicating requirements and desired results.
3. Discussion: Discuss the underlying applicable theory and concepts that support the measurements. Indicate and discuss the measurement set-up and equipment used.
4. Measurement data and/or Results: Present measurement results in tabular, graphical or numeric form. Present results from required lab exercises.
5. Discussion of Measurements: Discuss the measurement data in context of comparison to expectation, accuracy, difficulties, etc.
6. Summary and Conclusions: Discuss findings, explain errors and unexpected results and summarize and indicate conclusions.

Further requirements on the lab report are:

1. Correct spelling, grammar and punctuation is required.
2. Report must be typed; figures, drawings and equations may be handwritten.
3. Format of references must conform to IEEE (transactions) standards.

1.4 BACKGROUND

Control systems implemented on a digital computer are called digital control systems (tricky, huh?). Digital control systems differ from their analog counterparts in two important ways.

1. Digital systems operate in discrete time, not continuous time. That is, control computations do not occur continuously (as they do using op-amps for analog systems), but occur at discrete instants in time. These instants are usually regular periodic times, separated by the *sampling period* T [sec]. The sampling operation introduces artifacts into the signal which are known as signal *aliases*. It also introduces delay into the closed-loop system which tends to destabilize the control.
2. Digital systems do not use infinite-precision mathematics. To store the value of a signal—such as a control signal—in computer memory, the value must be *quantized*. This means that the value is rounded to the nearest number which can be stored. The coefficients of the transfer function of the controller must also be quantized. These are two separate issues: signal quantization and coefficient quantization.

In this lab period you will implement and simulate a discrete-time system in Simulink.

1.4.1 Notation for Sampling

Consider a continuous-time signal $x(t)$. If we look at the signal at discrete points in time that are separated by a constant sampling period T , then we have the set of values $x(kT)$ where k is an integer. To simplify notation, we say that $x[k] = x(kT)$ where $x[k]$ is a discrete-time signal (denoted using square brackets) and $x(t)$ is a continuous-time signal (denoted by parentheses). When you read $x[k]$ you should understand that some sampling period T is implied, which must be somehow specified elsewhere.

1.4.2 Linear Constant Coefficient Difference Equations

The continuous-time systems you have seen are defined by Linear Constant Coefficient Ordinary *Differential* Equations (LCCODEs). Linear, time invariant, lumped discrete-time systems may be defined by Linear Constant Coefficient *Difference* Equations (LCCDEs). A controller would be implemented with a LCCDE such as:

$$\sum_{i=0}^n a_i u[k-i] = \sum_{i=0}^m b_i e[k-i],$$

with input $e[k]$ and output $u[k]$ and a_i and b_i which determine the transfer function of the controller.

An example discrete-time system is a discrete-time integrator. If the sampling period is T , then the output of an integrator may be approximated by (rectangular rule)

$$y[k] = T \sum_{i=0}^k x[i],$$

where $y[k]$ is the output of the integral and $x[k]$ is the input signal. Note: $y[k-1] = T \sum_{i=0}^{k-1} x[i]$ so we can write²

$$y[k] = y[k-1] + Tx[k].$$

1.4.3 Discrete-Time Simulation

When we consider implementing the above equation, we find that we need two quantities: $x[k]$ and $y[k-1]$. The input to the system is $x[k]$, and $y[k-1]$ is a *delayed* version of the system's output. Therefore, we can implement the above equation with a feedback loop which has a delay in it.

²Notice that this is the same as $y[k] - y[k-1] = Tx[k]$, which is of the LCCDE form. However, the form given in the main text is more useful for us.

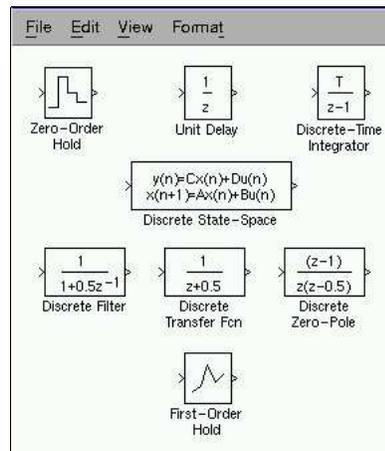


Figure 1.1 Simulink's discrete-time library.

Simulink has a number of blocks which will aid you in making discrete-time simulations. These are part of the Discrete-Time Simulink library, and are shown in Fig. 1.1. The two blocks of interest to us now are the “Zero-Order Hold” and “Unit Delay” blocks. Over the course of the semester, you will learn to use many of the other blocks as well. The reason for the label $1/z$ on the unit-delay will also become apparent.

To continue with the integrator example, consider the block diagram in Fig. 3.1 on page 3-2. A source (in this case a sine wave) generates an input signal $x(t)$. This signal must be sampled to make $x[k]$. The Simulink block which does this function is the “Zero-Order Hold” block. The $x[k]$ signal is scaled by T to produce $Tx[k]$. The output of the integrator is $y[k]$ which is computed as $x[k] + y[k - 1]$. The final remaining signal $y[k - 1]$ is computed from $y[k]$ by passing it through a “Unit Delay” block.

The zero-order hold, amplifier and unit delay blocks require the sampling period T . You will find that it is best to enter these as symbolic T and define the value of T in the Matlab workspace. This allows easy and consistent change of sampling rate.

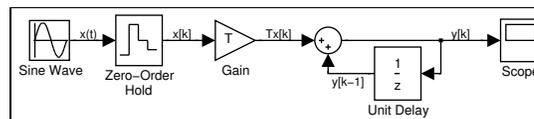


Figure 1.2 Discrete-time integrator implemented in Simulink.

1.5 PRELAB ASSIGNMENT

There are no prelab questions for this lab. You are still expected to submit a prelab containing a paragraph describing what you are supposed to do during the lab, and a paragraph describing expected results.

1.6 LABORATORY EXPERIMENT

A “leaky” integrator is defined by the difference equation $y[k] = ay[k - 1] + Tx[k]$, where usually $0 \leq a < 1$. The system is stable, however, for $-1 < a < 1$.

1. Implement a leaky integrator in Simulink. Plot the step response of the system for a number of values of $0 \leq a \leq 1$. Also plot the step response of the system for a number of values $-1 \leq a < 0$. Use $T = 0.1 \text{ sec}$.

2. Plot the response of the system with the input being a sine wave and $a = 0.9$. Use a sine wave of frequency 1 rad/sec . Use sampling periods $T = 0.1, 0.5, 1 \text{ sec}$.

1.7 ASSIGNMENT

Include your Simulink diagram and plots in your report. Comment on the shapes of the output curves. What prominent features do you see? How do you explain them? Your “leaky integrator” also has another name—suggest what kind of discrete-time filter it is implementing (Hint: It is either a low-pass, band-pass or high-pass filter).

Note: You have the next lab period free; the report for Lab 1 is due at the beginning of the following lab period.

(mostly blank)

Unit 1

Design by Emulation

In the laboratory this semester, there are four major groups of labs. This unit is the first such group and it focuses on digital control design via “emulating” an analog control design for the same plant. In:

- Lab 2 you will use time-domain methods to transform a continuous-time control design to a discrete-time control design; in
- Lab 3 you will use frequency-domain methods to transform a continuous-time control design to a discrete-time control design.

(mostly blank)

Time-Domain Controller Emulation

2.1 INTRODUCTION

Two different approaches may be taken to digital controller design. The first, which you investigate during the next two lab periods, emulates (or simulates) a known analog control design in discrete time. The second, which you will investigate the rest of the semester, uses direct digital design methods.

2.2 THIS LAB IS...

- An introduction to emulation by simulating an LCCODE with an LCCDE.
- Your first discrete-time control system.

2.3 BACKGROUND

Analog control design techniques for single-input single-output (SISO) dynamic systems result in controller transfer functions $D(s)$. The input to the controller is the tracking error $e(t) = r(t) - y(t)$ and the output of the controller is the control effort $u(t)$. Therefore, we can write

$$D(s) = \frac{U(s)}{E(s)}.$$

We are interested in mimicking the performance of $D(s)$ in discrete-time. That is, we would like to produce an output approximating $u(t)$. So, rearrange the above equation:

$$U(s) = D(s)E(s).$$

There are two ways to get a time-domain result from this equation:

1. We could take the inverse-Laplace transform of $U(s)$ to find $u(t)$

$$u(t) = \mathcal{L}^{-1} [D(s)E(s)],$$

but that would require *a priori* knowledge of $E(s)$ to compute $u(t)$. We do not know what $E(s)$ will be since it is influenced by disturbance and plant modeling errors which are unknown.

2. We could take the inverse-Laplace transform of the above equation on a term-by-term basis to come up with an LCCODE for the controller. This exactly reverses the steps normally taken to convert the dynamics of a plant to its transfer function. We then end up with

$$\sum_{k=0}^n a_k \frac{d^k u(t)}{du(t)^k} = \sum_{k=0}^m b_k \frac{d^k e(t)}{de(t)^k}.$$

The second form is useful. It gives the dynamic relationship between $e(t)$ and $u(t)$ regardless of what the time-history of each signal might be.

2.3.1 Approximating a derivative

A derivative operator may be approximated in discrete time using the “backward rectangular rule”

$$\dot{x}(t) \triangleq \lim_{\delta t \rightarrow 0} \frac{\delta x}{\delta t} = \lim_{\delta t \rightarrow 0} \frac{x(t) - x(t - \delta t)}{\delta t}.$$

If the sampling rate T of the discrete-time system is small, we may approximate $\delta t \approx T$

$$\dot{x}(kT) \approx \frac{x(kT) - x((k-1)T)}{T}, \quad \text{or} \quad \dot{x}[k] \approx \frac{x[k] - x[k-1]}{T}.$$

We can perform the same analysis for second- and higher-order derivatives. For example

$$\ddot{x}(t) \triangleq \lim_{\delta t \rightarrow 0} \frac{\delta \dot{x}}{\delta t} = \lim_{\delta t \rightarrow 0} \frac{\dot{x}(t) - \dot{x}(t - \delta t)}{\delta t}$$

$$\ddot{x}(kT) \approx \frac{\dot{x}(kT) - \dot{x}((k-1)T)}{T} \quad \text{or} \quad \ddot{x}[k] \approx \frac{\dot{x}[k] - \dot{x}[k-1]}{T}.$$

Apply the rule once again to $\dot{x}[k]$ and $\dot{x}[k-1]$

$$\begin{aligned} \ddot{x}[k] &\approx \frac{\frac{x[k]-x[k-1]}{T} - \frac{x[k-1]-x[k-2]}{T}}{T} \\ &= \frac{x[k] - 2x[k-1] + x[k-2]}{T^2}. \end{aligned}$$

2.3.2 Putting the pieces together

To see how to use these two facts to convert an analog controller into a digital controller we will examine a concrete example. Consider a lead or lag controller

$$D(s) = K \frac{s+a}{s+b}.$$

Rearranging, and noticing that $D(s) = U(s)/E(s)$, gives us the relationship

$$(s+b)U(s) = K(s+a)E(s).$$

Convert to an LCCODE via $sU(s) \mapsto \dot{u}(t)$, $U(s) \mapsto u(t)$, and so forth; we have

$$\dot{u}(t) + bu(t) = K\dot{e}(t) + Kae(t).$$

Replacing $u(t) \mapsto u[k]$, $\dot{u}(t) \mapsto (u[k] - u[k-1])/T$ and so forth

$$\begin{aligned} \frac{u[k] - u[k-1]}{T} + bu[k] &= K \frac{e[k] - e[k-1]}{T} + Kae[k] \\ u[k] - u[k-1] + bTu[k] &= Ke[k] - Ke[k-1] + KTaek[k] \\ u[k] &= \frac{u[k-1] + K((1+aT)e[k] - e[k-1])}{1+bT}. \end{aligned}$$

This equation says that the control signal at time index k may be computed using the previous control value $u[k-1]$, the current error signal $e[k]$ and the previous error signal $e[k-1]$.

2.3.3 Implementing the difference equation

The above difference equation may be easily implemented in Simulink. In Fig. 2.1 on page 2-3, a subsystem for implementing this controller is drawn. The input port is $e[k]$, the output port is $u[k]$, and the $1/z$ blocks are unit-delays.

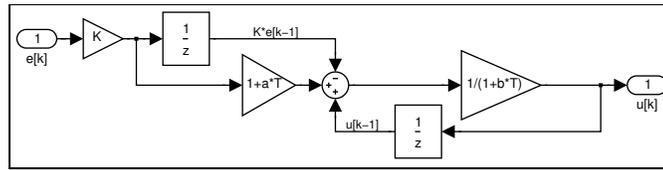


Figure 2.1 Simulink subsystem for implementing a lead or lag controller.

2.4 PRELAB ASSIGNMENT

In this lab you will control the MagLev device, configured with a single disk (lower position). You will control the position of the disk around an equilibrium position of 2.5 cm . The steady-state dc value to achieve the equilibrium height may be calculated as

$$u_{1o} = 100mga(y_{1o} + b)^4,$$

where $m = 0.12\text{ kg}$, $g = 9.8\text{ m/sec}^2$, a and b are the actuator characteristics of your MagLev (consult App. B), and the factor of 100 is to convert m to cm so that the units work out with u_{1o} in V .

The linearized transfer function of the system may be found to be (again, consult App. B)

$$G(s) = \frac{k_{u11}}{ms^2 + m\beta s + k_{11}},$$

where the friction coefficient is approximately 5 sec^{-1} . The k_{u11} and k_{11} coefficients may be calculated for the given equilibrium height using the formulae in App. B.

1. Calculate the equilibrium dc voltage u_{1o} for your MagLev.
2. Calculate the transfer function $G(s)$ for your MagLev. Normalize your transfer function so that the leading coefficient in the denominator is 1. (That is, compute $G(s) = (k_{u11}/m)/(s^2 + \beta s + (k_{11}/m))$.)
3. A PID controller which is known to give reasonable performance on all four MagLevs has transfer function¹

$$D(s) = 0.21 + \frac{19.95}{s} + 0.04s = \frac{0.04s^2 + 0.21s + 19.95}{s}.$$

- (a) Simulate the closed-loop step response for your MagLev using the given $D(s)$ (in continuous-time). If you use Simulink, you will not be able to enter $D(s)$ as a transfer function since the numerator has higher-order power of s than the denominator—use a differentiator block to compute this derivative instead, or combine the plant dynamics and controller dynamics into one transfer-function block.
- (b) Estimate (from your simulation) the rise time, settling time, peak overshoot and steady-state error for the analog control system.
- (c) Convert $D(s)$ into a discrete-time equation that is a function of the sampling period T . (Do not use the canned formula in the ECE4540 course notes; use the method given in this lab writeup. They do not give the same results.)
- (d) Construct a Simulink subsystem which implements your difference equation of part (c).

2.5 LABORATORY EXPERIMENT

During the lab, you will implement a control system using $D(s)$ and a control system using your difference equations found in the Prelab. In Matlab, create a Simulink block diagram by first typing `template`

¹This transfer function was found using a numerical optimization procedure you will investigate in Lab 8.

```
>> template
```

at the Matlab prompt. This opens up a window with the “HW Adapter” block and a “Scope” block already in place, with many of the parameters set to the correct values.

1. Construct a closed-loop control system using $D(s)$ to control the MagLev. Your Simulink diagram will look like the one in Fig. 2.2. You can find the “Fix Sensor” block for your MagLev in the `utility.mdl` library. (That is, type `utility` at the Matlab prompt). The A2D and D2A blocks may be found in the `rtlib.mdl` library. (That is, type `rtlib` at the Matlab prompt). The step time should be at 5 sec .

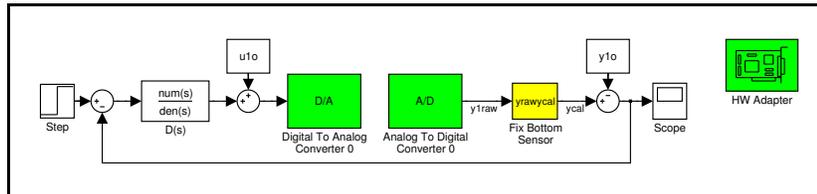


Figure 2.2 Analog closed-loop control system in Simulink. Note that you will need to find a different way of implementing $D(s)$ since it has a pure derivative in it!

2. Run the control system, and record the step response. Use a sampling rate of 2 kHz .
3. Now, construct a discrete-time control system using your difference equations. Your Simulink diagram will look like the one in Fig. 2.3. Be sure to leave T as a variable parameter in your Simulink block diagram so that you can change it easily within the Matlab environment (You will have to hard-code the sampling frequency $1/T$ in the “HW Adapter” block as it does not work if you leave it as a variable. Don’t forget to change the rate here every time you change it in your workspace!).

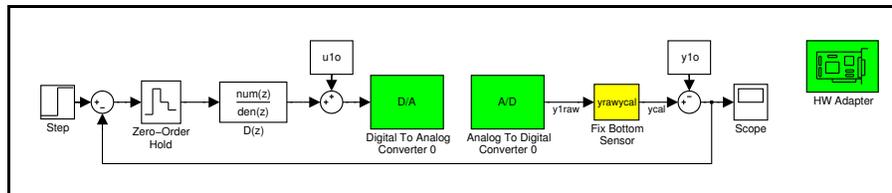


Figure 2.3 Discrete-time closed-loop control system in Simulink. The block denoted $D(z)$ should be replaced by your subsystem which computes your difference equation.

4. Run your emulated control system for different values of T . (Be sure to change the sampling rate in the Hardware Adapter block, and Simulink’s “Simulation > Parameters” too!) Determine a range of T that seems to give performance roughly equivalent to the analog control design. Without destroying the equipment, attempt to find a range of T that gives stable closed-loop response.
5. Record step responses of your discrete-time control system for illustrative values of T to include in your report.

2.6 ASSIGNMENT

In your report, give a full derivation of the difference equations you used for your PID controller. Include all computed results (such as u_{1o} , $G(s)$ and so forth). Give simulated and measured step responses for the analog control system, and compare with the measured step responses of your discrete-time control system for various values of T . Estimate the bandwidth of the controller $D(s)$ and compare to the sampling frequency $1/T$ that you think “works best.” Report the range of T which seems to result in a stable closed-loop system.

Frequency-Domain Controller Emulation

3.1 INTRODUCTION

In the last lab period you simulated an analog PID control design with a discrete-time controller. You performed the conversion by changing derivatives to differences and therefore changing LCCODEs to LCCDEs.

In this lab you will also investigate emulation. This time, you will convert a controller transfer function $D(s)$ directly to a discrete-time transfer function $D(z)$ by recognizing a relationship between s and z . You will also look at a heuristic method of improving performance (improving damping) of your emulated design which tries to undo the harmful effects of the zero-order-hold circuit.

3.2 THIS LAB IS...

- A second look at emulation using frequency-domain methods.
- Experience with a novel way to compensate for zero-order-hold effects.

3.3 BACKGROUND

In Lab 2, you were introduced to emulation with the realization that the Laplace operator “ s ” performs a derivative operation which may be approximated in discrete-time. In this lab, it is more natural to work with the “ $1/s$ ” operator which is an integration. Consider

$$Y(s) = \frac{1}{s}X(s) \quad \Longleftrightarrow \quad y(t) = \int_0^t x(\tau) d\tau.$$

If we sample $y(t)$ every T seconds, we get

$$\begin{aligned} y(kT) &= \int_0^{kT} x(\tau) d\tau \\ &= y((k-1)T) + \int_{(k-1)T}^{kT} x(\tau) d\tau. \end{aligned}$$

So, $y(kT)$ is related to $y((k-1)T)$ plus an integral of the input signal over one time period.

We can think of at least three ways to approximate a continuous-time integral in discrete-time (see Fig. 3.1 on page 3-2). Each of the three resulting difference equations may be z -transformed to get:

$$\begin{aligned} \text{Forward Rectangular Rule : } Y(z) &= \left[\frac{T}{z-1} \right] X(z); \\ \text{Backward Rectangular Rule : } Y(z) &= \left[\frac{Tz}{z-1} \right] X(z); \\ \text{Bilinear Rule : } Y(z) &= \left[\frac{T}{2} \frac{z+1}{z-1} \right] X(z). \end{aligned}$$

In all cases, the term in the square brackets $[\cdot]$ is an approximation to $1/s$ in $Y(s) = \frac{1}{s}X(s)$. So, in order to convert a continuous-time transfer function $D(s)$ to a discrete-time transfer function $D(z)$, use the following design procedure:

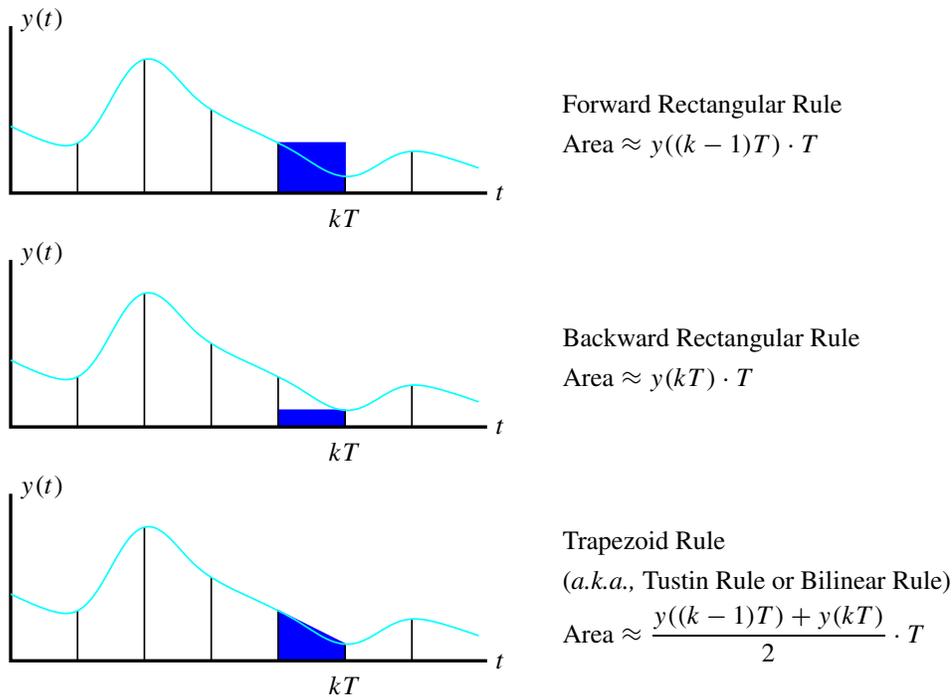


Figure 3.1 Approximating a continuous-time integral in discrete-time.

Design Procedure

1. Start with analog control design $D(s)$.
2. Replace $s \mapsto z$ according to the following table to get $D(z)$.

$$\text{Forward : } s \mapsto \frac{z-1}{T}; \quad \text{Backward : } s \mapsto \frac{z-1}{Tz}; \quad \text{Bilinear : } s \mapsto \frac{2}{T} \frac{z-1}{z+1}.$$

3.3.1 Approximate compensation for ZOH

One reason that design-by-emulation does not work well is the effect of the zero-order-hold (ZOH) which adds a delay of $T/2$ or a phase of $-\omega T/2$ to the open-loop system. This reduces the damping/ stability of the closed-loop system.

We might consider “reversing” or “inverting” the effect of the ZOH by adding a $\frac{2z}{z+1}$ term to our controller. The added term has a phase of $+\omega T/2$ which cancels the ZOH phase. In practice it is better to add a $\frac{2(z-\varepsilon)}{z+1-2\varepsilon}$ term for small ε . A pole added at -1 amplifies high-frequency sensor noise, and by moving the pole we end up with a more stable system.

This heuristic method (which can be examined in further detail in Reference [1]) is not guaranteed to work since a number of approximations have been made along the way whose error accumulates. However, it is always worth trying it out to see if it improves performance. The design procedure is then:

Design Procedure

1. Start with analog control design $D(s)$.
2. Replace $s \mapsto z$ with the forward, backward or bilinear rule to get $D'(z)$.
3. Compute $D(z) = \frac{2(z-\varepsilon)}{z+1-2\varepsilon} D'(z)$, the modified controller which attempts to account for the ZOH.

3.3.2 Implementation in Simulink

To implement your controller, you will use the discrete-time transfer-function block from the Simulink library. This block has built-in sampling, so you do not need the ZOH block preceding your controller as you did in Lab 2. Your control system will look like the one in Fig. 3.2.

3.4 PRELAB ASSIGNMENT

For this lab, the MagLev will be configured in the same way as in Lab 2. You will emulate the same PID controller.

1. Convert $D(s)$ to $D(z)$ using the three methods (but without the ZOH compensation) described in the “Background” section.
2. Which of the three resulting controllers are causal? (If the order of z in the numerator is strictly greater than the order in the denominator, the transfer function is *not* causal).

Leave all results in terms of variable T since you will need to change T when you implement the lab.

3.5 LABORATORY EXPERIMENT

During the lab, you will implement a control system which emulates $D(s)$ using (1) the backward-rectangular rule and (2) the backward-rectangular rule with ZOH compensation. In Matlab, create a Simulink block diagram to implement both methods.

1. Construct a discrete-time control system using $D(z)$ calculated with the backward-rectangular rule (without the ZOH compensation). Your Simulink diagram will look like the one in Fig. 3.2. Be sure to leave T as a variable parameter in your Simulink block diagram so that you can change it easily within the Matlab environment (You will have to hard-code the sampling frequency $1/T$ in the “HW Adapter” block as it does not work if you leave it as a variable. Don’t forget to change the rate here every time you change it in your workspace!). The step time should be at 5 sec.

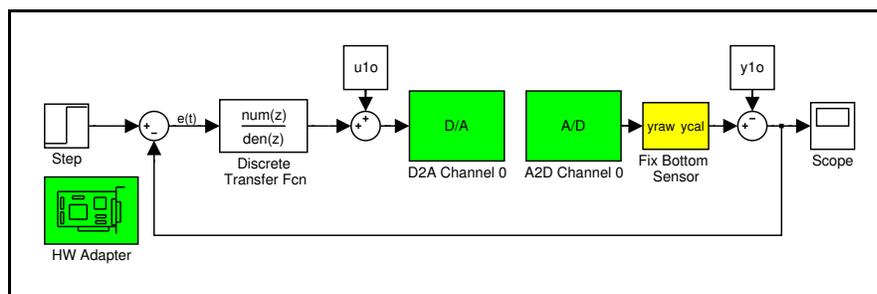


Figure 3.2 Discrete-time closed-loop control system in Simulink.

2. Run your emulated control system for different values of T . (Be sure to change the sampling rate in the Hardware Adapter block, and Simulink’s “Simulation > Parameters” too!) Determine a range of T that seems to give performance roughly equivalent to the analog control design. Without destroying the equipment, attempt to find a range of T that gives stable closed-loop response.
3. Record step responses of your discrete-time control system for illustrative values of T to include in your report.

4. Repeat steps 1–3 using the controller $D(z)$ found using the backward-rectangular rule *with* the ZOH compensation. A ε of about 0.1 seems to work well.
5. Optionally, repeat step 1 using the controller $D(z)$ found using the bilinear rule. Your $D(z)$ will have a pole at -1 which amplifies high-frequency sensor noise, and will not work (at all) well. Move this pole slightly (*e.g.*, to -0.9).

3.6 ASSIGNMENT

In your report, give a full derivation of the three controllers found with the three methods (without ZOH compensation). Which are causal? Show that the controller transfer function obtained with the backward-rectangular rule is the same as the one found using difference equations in Lab 2.

Compare with the measured step responses of your discrete-time control system for various values of T . For the two methods investigated in this lab, report the range of T which seems to result in a stable closed-loop system, and the range of T which seems to give good performance. Compare with the results you found in Lab 2. Does the ZOH-compensation circuit help?

This is the end of Unit 1. You have the next lab period off to give you time to write your lab report. Your formal laboratory report *covering both labs in this unit* is due at the beginning of the subsequent lab period.

References

- [1] D. Raviv and E. Djaja, “*Techniques for Enhancing the Performance of Discretized Controllers,*” IEEE Control Systems Magazine, Vol. 19, No. 3, June 1999, pp. 52–57.

Unit 2

Digital Effects

This unit of labs focuses on artifacts of the digital system not present in an analog control system. In:

- Lab 4 you will investigate sampling, aliasing, anti-aliasing filtering and the zero-order-hold operation; in
- Lab 5 you will make a discrete-time model of the plant and verify it by comparing it to the true system response; in
- Lab 6 you will use fixed-point arithmetic to implement filter coefficients and investigate problems due to round-off.

(mostly blank)

Sampling, Aliasing, Zero-Order-Hold

4.1 INTRODUCTION

In class we have talked about an artifact of digital systems not present in analog systems. When sampling a continuous-time signal, a phenomenon called *aliasing* may occur. High-frequency signals may appear as low-frequency sampled signals. In this lab, you will experience aliasing, and will experiment with an anti-aliasing filter to eliminate it.

4.2 THIS LAB IS...

- A demonstration of aliasing,
- A means of eliminating aliasing via an anti-aliasing filter.

4.3 BACKGROUND

Digital control systems must sample the input signal $e(t)$ in order to generate a compensation signal $u(t)$. This sampling is usually done on a regular (periodic) basis, so we refer to a sampling frequency as being $1/T$ where T is the period between samples. The input to the digital controller is then $e(kT)$ or more simply $e[k]$.

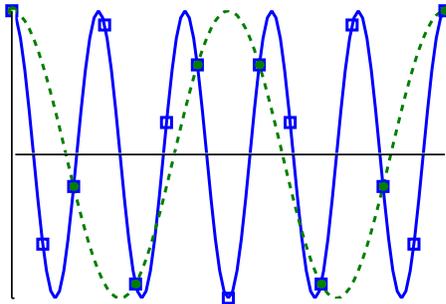


Figure 4.1 Time-domain example of aliasing.

If sampling is not done “quickly enough” a phenomenon called aliasing occurs. This is demonstrated in the time domain in Fig. 4.1. The solid (blue) line indicates a cosine of frequency 5 Hz. If this signal is sampled at 14 Hz the points indicated by (blue) squares are the samples. The signal may be well reconstructed from these samples. If, however, the signal is sampled at 7 Hz, the points indicated by (green) circles are the samples. This signal cannot then be distinguished from the 2 Hz sinusoid shown as the (green) dashed line. Therefore, if we sample too slowly (7 Hz), our 5 Hz signal *masquerades* or *aliases* as a different 2 Hz signal.

If we consider what is happening in the frequency domain, we find that the spectrum of the sampled signal is periodic in f_s . The 5 Hz cosine, which has frequency content at ± 5 Hz, repeats every 14 Hz when sampled at 14 Hz and so has frequency content at ± 5 Hz, ± 9 Hz, ± 19 Hz and so forth. Since we only consider frequencies within the band $\pm f_s/2$, then we are left with a sampled-signal spectrum of ± 5 Hz, which is what we would like.

On the other hand, if we sample the signal at 7 Hz , then the sampled signal has frequency content at $\pm 2\text{ Hz}$, $\pm 5\text{ Hz}$, $\pm 9\text{ Hz}$ and so forth. Since we only consider frequencies within the band $\pm f_s/2$ then we are left with the sampled-signal spectrum of $\pm 2\text{ Hz}$, which is not what we expect.

Shannon/Nyquist showed that in order to avoid aliasing, the sampling frequency must be at least twice the highest frequency present in the input signal. For the cosine example, above, we must sample at a rate of at least 10 Hz . This explains why the 14 Hz sampling rate “worked” and the 7 Hz rate did not. A practical difficulty in applying this rule is that real signals are not strictly band-limited. Therefore, we must make engineering approximations and tolerate some aliasing, or place an analog anti-aliasing low-pass filter placed before the sampler will also help eliminate effects due to aliasing. (A digital filter placed after the sampler will not work—why?)

4.4 PRELAB ASSIGNMENT

During the lab, you will sample three types of signals: sinusoidal, triangle-wave and square-wave.

1. Determine the complex-exponential Fourier series of:

(a) A sinusoidal signal $y_1(t) = \cos(2\pi ft)$.

(b) A triangle wave $y_2(t) = -4f|t| + 1$ for $|t| \leq \frac{1}{2f}$ and periodic thereafter.

(c) A square wave $y_3(t) = \text{sgn}(y_1(t))$.

You may look up the series in a table—you do not need to derive them if you can find a reference somewhere.

2. In the lab, you will use signal frequencies $f = 25\text{ Hz}$ for most experiments. Plot the line spectra for the above three signals with this f .
3. If you sample at a rate of $f_s = 100\text{ Hz}$, draw the *aliased* line spectra for the above three signals.
4. You will use an anti-aliasing low-pass filter with transfer function

$$H(s) = \frac{25000}{s^2 + 200s + 25000}$$

after the signal source and *before* the sampler. Plot the line spectra for the three signals after filtering (Magnitude only). Should this help eliminate aliasing when the signals are sampled? How do you expect the shape of the three signals to be changed by the anti-aliasing filter?

4.5 LABORATORY EXPERIMENT

For the first experiment, hook up the apparatus as shown in Fig. B.10 on page B–11. Construct a Simulink block diagram as shown in Fig. 4.3. The frequency generator supplies the source signal, which is displayed on the scope and input to the computer via the A2D and A2D/ ports of the breakout box. The computer passes the signal straight through (sampling it). Connect the computer output D2A and D2A/ back to another channel of the scope.

In the following “oscilloscope” refers to the hardware box dedicated to this task, and “scope” refers to the Simulink block which records data.

1. Set the frequency generator to generate sinusoidal signals of amplitude 5 V and no dc-offset. (**Check the amplitude on the oscilloscope before connecting to the computer so that you do not damage the computer!**)
2. Construct the Simulink block diagram of Fig. 4.3. Set the sampling rate to 100 Hz .
3. Generate frequencies between 0 Hz and 50 Hz and run them through the computer. Do the signals recorded by Simulink’s “scope” block look correct? How does the computer output signal on the oscilloscope correspond to the signal from the signal generator? Record a few representative results from Simulink’s “scope” block.

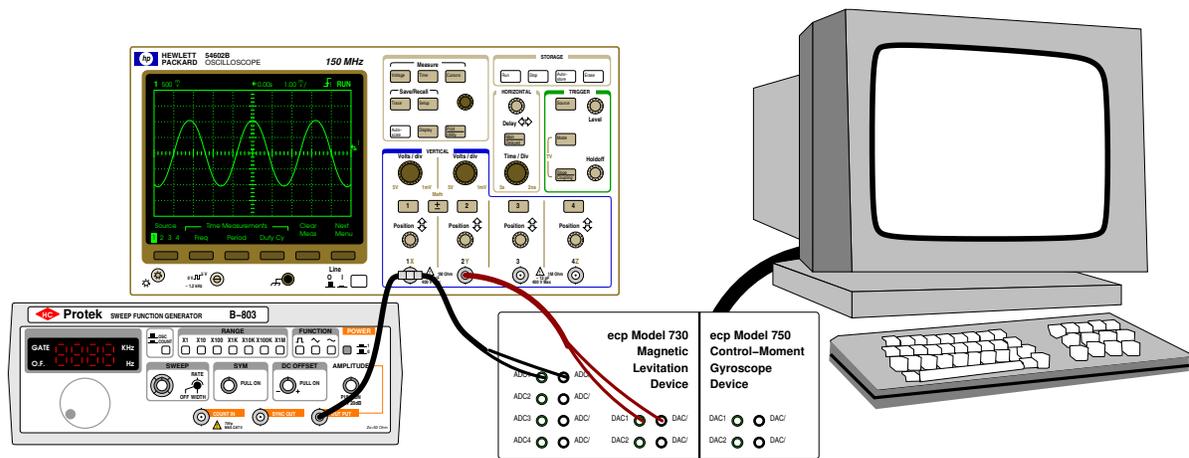


Figure 4.2 Setup for aliasing experiment.

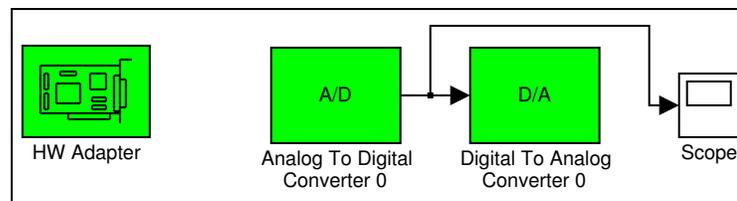


Figure 4.3 Simulink diagram to sample and pass-through samples.

4. Generate frequencies between 50 Hz and 200 Hz and run them through the computer. Do the signals recorded by Simulink's "scope" block look correct? How does the computer output signal on the oscilloscope correspond to the signal from the signal generator? Record a few representative results from Simulink's "scope" block. Sketch output from the oscilloscope to show aliasing occurring.

An anti-aliasing filter has been constructed for you and implemented on a GP-6 analog computer (see Fig. 4.4 on page 4-4). It has nominal transfer function $H(s)$ as mentioned in the prelab. The input leads and output leads from the GP-6 filter are labeled.

5. Hook up the anti-aliasing filter between the frequency generator and the oscilloscope. That is, run the output of the signal generator to the input of the GP-6 filter; run the output of the GP-6 filter to the oscilloscope and into the computer. View the oscilloscope and visually compare the output of the frequency generator and the output of the anti-aliasing filter as you vary the frequency from 0 Hz up to 200 Hz . Repeat steps 3-4. You may need to set the frequency-generator output to about 3 V to avoid overflow on the GP-6 computer.
6. Set the frequency generator to generate 25 Hz triangle-wave signals of amplitude 3 V and no dc-offset. Run the frequency-generator output through the computer. Do the signals recorded by Simulink's "scope" block look correct? How does the computer output signal on the oscilloscope correspond to the (filtered) signal from the signal generator? Record a few representative results from Simulink's "scope" block. **Do this step both with and without the anti-aliasing filter.**
7. Set the frequency generator to generate 25 Hz square-wave signals of amplitude 3 V and no dc-offset. Run the frequency-generator output through the computer. Do the signals recorded by Simulink's "scope" block look correct? How does the computer output signal on the oscilloscope correspond to the (filtered) signal from the signal generator? Record a few representative results from Simulink's "scope" block. **Do this step both with and without the anti-aliasing filter.**

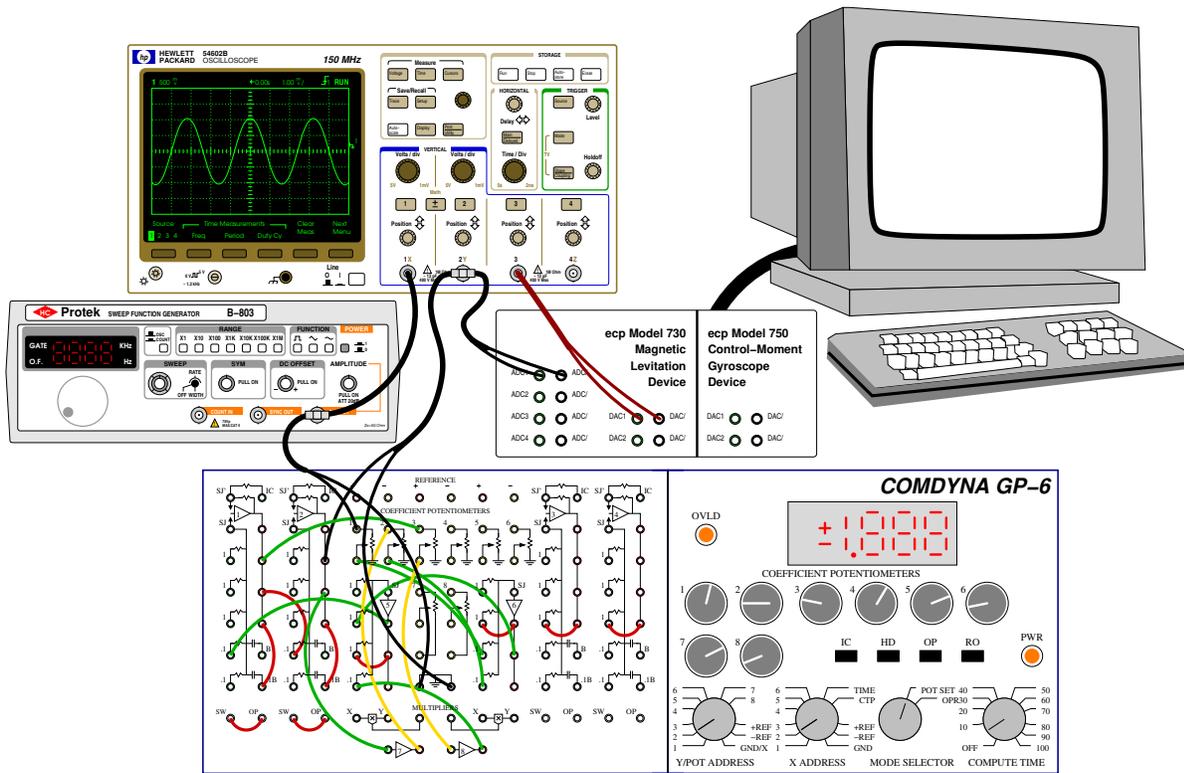


Figure 4.4 Setup for aliasing experiment, with anti-aliasing filter.

4.6 ASSIGNMENT

In this lab, you will collect a lot of data. Organize your Prelab and Experiment results and indicate correspondences with theory, and any unusual or unexplained results.

Discrete-Time Plant Modeling

5.1 INTRODUCTION

From now on, all of our control-system analysis and design will be done using z -transforms and w -transforms. Both of these require that a discrete-time plant model be made.

A discrete-time plant model combines the LTI dynamics of the zero-order-hold and the analog plant, and relates the plant output *at the sample instants* with the digital controller output *at the sample instants*. The plant model may be represented as a transfer function $G(z)$ or—using the bilinear transformation—as a transfer function $G(w)$.

In this lab, you will create discrete-time plant models $G(z)$ and $G(w)$ of the MagLev device which are functions of the sample period T . You will test these models at different sampling frequencies to verify that they reasonably predict the MagLev's behavior.

5.2 THIS LAB IS...

- A demonstration of discrete-time transfer functions,
- Experience in both the z - and w -domains.
- More fun with the MagLev!

5.3 BACKGROUND

5.3.1 Hybrid systems analyzed at sampling instants

Digital control systems are neither completely digital nor completely analog. They are *hybrid* systems which contain both digital and analog elements. (Another name which is used to describe hybrid systems like this is “*sampled-data systems*.”) The compensator $D(z)$ is a digital system, and the plant $G_p(s)$ is an analog system. The sampler and zero-order-hold devices are a bit of each.

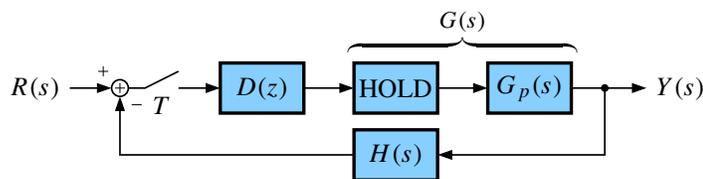


Figure 5.1 Closed-loop hybrid control system.

We have seen that in order to analyze hybrid systems in continuous time, the starred Laplace transform needs to be used; *e.g.*, $X^*(s)$. If we only need to evaluate system response at the sampling instants, the z -transform may be used. This is generally much simpler. Furthermore, the bilinear transform relates the z -domain to the w -domain, which allows frequency-response methods of Laplace analysis and design to be used on digital/hybrid systems.

Figure 5.1 shows the general configuration used for digital control. We have seen in class that the closed-loop transfer function for this system is

$$T(z) = \frac{D(z)G(z)}{1 + D(z)\overline{GH}(z)},$$

where

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left\{ \frac{G_p(s)}{s} \right\} \quad \text{and} \quad \overline{GH}(z) = \frac{z-1}{z} \mathcal{Z} \left\{ \frac{G_p(s)H(s)}{s} \right\}.$$

Therefore, we can analyze the system if we know $G(z)$ and $\overline{GH}(z)$. Furthermore, we will find that we can analyze and design control systems in the w -domain by using the bilinear transform

$$T(w) = T(z) \Big|_{z = \frac{1+(T/2)w}{1-(T/2)w}}.$$

We can also find $T(w)$ by individually converting $D(z) \mapsto D(w)$, $G(z) \mapsto G(w)$ and $\overline{GH}(z) \mapsto \overline{GH}(w)$ and computing

$$T(w) = \frac{D(w)G(w)}{1 + D(w)\overline{GH}(w)}.$$

In this lab, you will compute $G(z)$ and $G(w)$ for future reference.

5.3.2 Computing $G(z)$ and $G(w)$

Recall that the analog transfer function of the MagLev, operating with a single disk (lower position) is

$$G_p(s) = \frac{k_{u11}}{ms^2 + m\beta s + k_{11}}.$$

We could use the Matlab `c2d` command to find the transfer function $G(z)$ for a specific value of T . Alternately, we could convert the plant dynamics to the z -domain for symbolic T . To see how to do this, we need to “complete-the-square” in the plant’s transfer function denominator, and write the result as

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left\{ \frac{G_p(s)}{s} \right\} = \frac{z-1}{z} \mathcal{Z} \left\{ g \frac{a^2 + b^2}{s((s+a)^2 + b^2)} \right\},$$

where g is the dc-gain of the analog plant. The z -transform of this result may be found using the table (entry 22) on the *FPW* front flyleaf:

$$G(z) = g \frac{Az + B}{z^2 - 2e^{-aT}(\cos bT)z + e^{-2aT}},$$

where

$$A = 1 - e^{-aT} \cos bT - \frac{a}{b} e^{-aT} \sin bT;$$

$$B = e^{-2aT} + \frac{a}{b} e^{-aT} \sin bT - e^{-aT} \cos bT.$$

We can then find

$$G(w) = G(z) \Big|_{z = \frac{1+(T/2)w}{1-(T/2)w}}$$

$$= g \frac{A \frac{1+(T/2)w}{1-(T/2)w} + B}{\left(\frac{1+(T/2)w}{1-(T/2)w} \right)^2 - 2e^{-aT}(\cos bT) \left(\frac{1+(T/2)w}{1-(T/2)w} \right) + e^{-2aT}}.$$

5.3.3 In Matlab?

In Matlab, you can convert a continuous-time transfer function to a discrete-time transfer function in z via

```
sysc=tf(num,den);           % Continuous-time transfer function.
sysd=c2d(sysc,Ts,'zoh');   % Ts=sampling period.
```

Converting from a transfer-function in z to a transfer function in w is not quite so smooth as Matlab does not directly understand the w -domain. The following command

```
sysw=d2c(sysd,'tustin'); % Be careful to understand!
```

converts from the z -domain back to a transfer function written in s , but the numbers are the same as if you had done the bilinear transform to the w -domain. That is, the `sysw` transfer function is written as a function of s in Matlab, but if you replace s with w you have done the w -transform.

5.4 PRELAB ASSIGNMENT

1. For your MagLev, and for sampling rates of 100 Hz , 150 Hz and 200 Hz find $G(z)$ and $G(w)$. Do by hand for one sampling rate, and verify with Matlab. You may use Matlab to automatically do the other two.
2. Plot the Bode-magnitude plot for $G_p(s)$.
3. Plot the Bode-magnitude plot for $G(w)$ for all three sampling rates.

5.5 LABORATORY EXPERIMENT

Generate a Simulink model as shown in Fig. 5.2. Levitate the magnet around an equilibrium height of 2.5 cm .

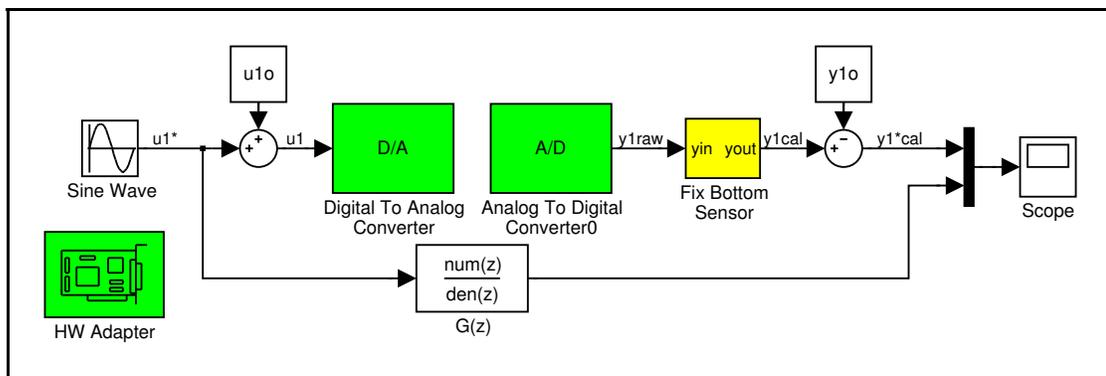


Figure 5.2 Setup for testing $G(z)$.

1. Use a sampling frequency of 100 Hz .
2. For sinusoidal frequencies between 0 rads/sec and 40 rads/sec and appropriate magnitudes (so that the magnet moves, but does not slam against the lower stop) record system output and model output.
3. Optional: Experiment with a random signal source (replace the sine-wave generator). Compare model output with system output at the sampling points.
4. Repeat step 2 for sampling frequencies 150 Hz and 200 Hz .

5.6 ASSIGNMENT

Comment on how well your discrete-time model $G(z)$ agrees with the sinusoidal responses you measured, for the three different sampling frequencies and at different test frequencies. Make a rough Bode-magnitude plot of system response from your results, and compare with the Bode-magnitude plot of $G(w)$. Note that $G(w)$ has a “warped” frequency axis. Comment on similarities and differences. Try to explain any differences.

Filter Structure and Finite-Precision Effects

6.1 INTRODUCTION

Digital computers are finite-precision machines. A signal value can only take on one of a finite set of possible values; a transfer-function coefficient can also only take on one of a finite set of possible values. In the lab so far, we have used floating-point numbers as signal values and as coefficient values and this has largely mitigated finite-precision effects. However, floating point processors are relatively expensive, so it is important to understand finite-precision effects and how to design for finite-precision fixed-point machines.

6.2 THIS LAB IS...

- An introduction to finite-precision effects,
- One method to implement controller structures which help reduce sensitivity to finite-precision effects.

6.3 BACKGROUND

Inexpensive micro-controllers, which are often used to implement digital control systems, are *fixed-point* processors. A certain number of binary digits are used to represent each number; some of these digits are assumed to be to the left of the binary point and some are assumed to be to the right of the binary point. An example four-bit number is then

$$\boxed{b_0 \mid b_1 \mid . \mid b_2 \mid b_3}$$

which is represented in decimal as $b_0 \times 2^1 + b_1 \times 2^0 + b_2 \times 2^{-1} + b_3 \times 2^{-2}$. Sixteen possible values may be represented: 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5 and 3.75. If signed numbers are desired, then one bit is used as the sign value. The sixteen possible values are then: $-2, -1.75, -1.5, -1.25, -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5$ and 1.75. With a four-bit number we could also assume that the fixed point was in the following configurations

$$\boxed{b_0 \mid . \mid b_1 \mid b_2 \mid b_3} \quad \text{or} \quad \boxed{b_0 \mid b_1 \mid b_2 \mid . \mid b_3}$$

A leading fixed point does not make sense as the b_0 bit is generally the sign bit.

Since only a finite set of numbers may be represented with a fixed-point value, the following effects occur:

1. Quantization noise at A2D: When real-world signals enter the computer at the A2D, the real-valued signal must take on one of a finite set of values. The analog value is generally *rounded* to the nearest digital value. The difference between the true analog value and the digital value is *quantization error*. This quantization error may be modeled as a noise source.

2. Quantization noise at D2A: The internal representation of signals often has a greater number of bits than the D2A is able to handle. For example, internal computations may be performed with 32-bit numbers, but the D2A is only able to handle 16-bit numbers. Converting the 32-bit numbers to 16-bit numbers introduces another source of quantization error.
3. Internal overflow: When performing computations with fixed-point numbers, it is possible to obtain a result which cannot be correctly represented. For example, when using the first four-bit scheme above, we could add 3 plus 3, but not get the value 6 since we can only represent numbers up to a maximum of 3.75. Due to the vagaries of twos-complement arithmetic, we would get the value 2, which makes little sense.
4. Limit cycles: A strange thing indeed! A perfectly stable system with zero input can have oscillating output which does not decrease in amplitude.
5. Coefficient quantization changes filter transfer function: We know that transfer functions may be represented as rational polynomials in z , and the coefficients of the polynomials are also the coefficients of the difference equations used to implement the controller. These coefficients must be represented in our fixed-point number scheme, so round-off occurs when implementing the coefficients of the difference equation. This in turn moves the poles and zeros of our compensator, so that we implement a transfer function that is different from what we intended to implement.

During this lab, you will investigate this final effect.

6.3.1 Two Filter Structures

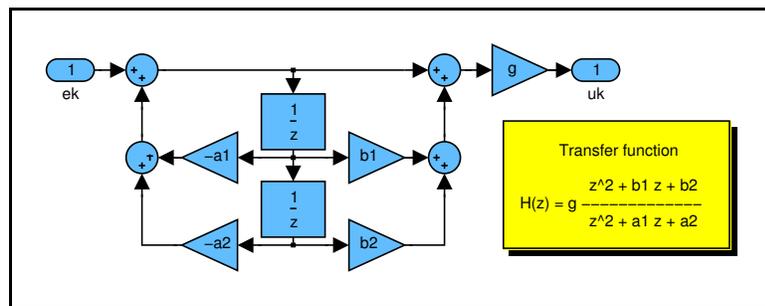


Figure 6.1 Direct Form II second-order section.

For any given controller transfer function $D(z)$ there are a wide variety of ways to implement it using unit delays, multipliers and summation units. Often, $D(z)$ is broken down into a cascade of “second-order sections” $D(z) = D_1(z)D_2(z) \cdots D_n(z)$ where each second-order section has two zeros and two poles. One way to implement a second-order section (called “Direct Form II”) is shown in Fig. 6.1.

The coefficients of the multipliers in this form are g , b_1 , b_2 , a_1 and a_2 . If the coefficients of the filter can only take on a finite set of possible values due to quantization, this will affect the poles and zeros implemented by the filter. That is, the poles and zeros implemented will be different from those designed. To see what happens, consider one section

$$D_1(z) = \frac{z^2 - 2r_1 \cos \theta_1 z + r_1^2}{z^2 - 2r_2 \cos \theta_2 z + r_2^2}$$

where r_1 is the radius to the zeros, $\pm\theta_1$ is the angle to the zeros, r_2 is the radius to the poles and $\pm\theta_2$ is the angle to the poles. In a Direct-Form-II section, $b_1 = -2r_1 \cos \theta_1$, $b_2 = r_1^2$ and so forth. Therefore, if we quantize b_1 and b_2 uniformly, we are actually quantizing the radius and angles non-uniformly. A sample grid showing quantized radii and $r_1 \cos \theta_1$ (real part of pole location) is shown in Fig. 6.2 on page 6-3(a). Poles and zeros may only be implemented at the intersection of the vertical grid lines and radial grid lines.

Some observations may be made:

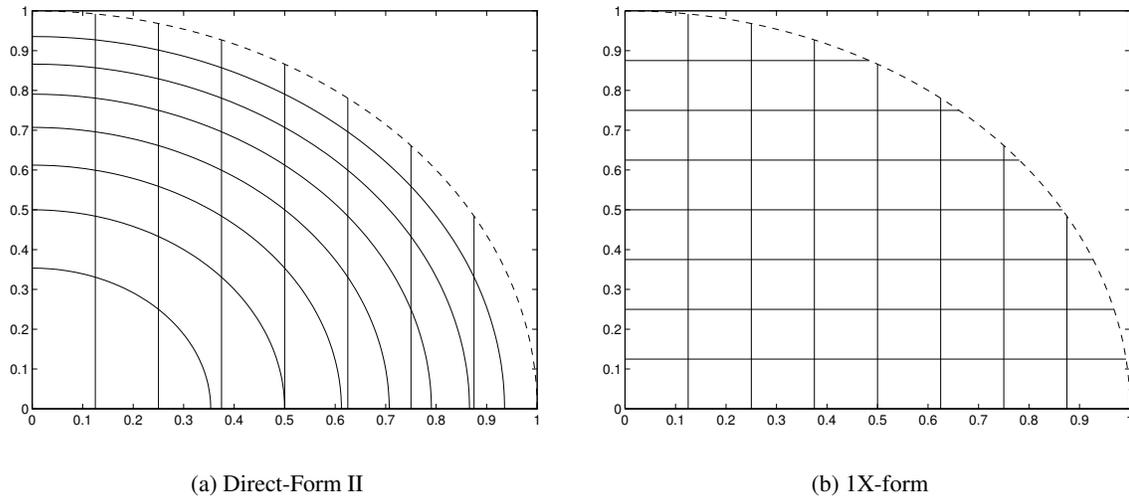


Figure 6.2 Quantized pole locations for Direct-Form II and 1X-Form.

- Narrow-band lowpass and narrow-band highpass filters are most sensitive (since they cluster poles around ± 1) and hence require more bits to implement.
- Sampling at too high a rate is bad, since it pushes poles closer to $+1$ (where pole density is smallest.)

Because of these undesirable features, we can use an alternative filter implementation, called the *1X-form*. This form is illustrated in Fig. 6.3 on page 6–4. The filter parameters are calculated by converting $D_1(z)$ to a partial-fraction expansion:

$$D_1(z) = a_0 + \frac{A}{s - p} + \frac{A^*}{s - p^*},$$

where

$$\begin{aligned} g_1 &= \Re(p); & g_2 &= \Im(p); \\ g_3 &= 2\Im(A); & g_4 &= 2\Re(A), \end{aligned}$$

in the diagram. A Matlab code segment to find a_0 , g_1 , g_2 , g_3 and g_4 from a given second-order $D(z)$ is:

```
[num,den]=tfdata(d); num=num{:}; den=den{:};
[r,p,a0]=residue(num,den);
g1=real(p(1)); g2=imag(p(1)); g3=2*imag(r(1)); g4=2*real(r(1));
```

All filter coefficients in the 1X-form second-order section that are related to the pole locations are of the form $\Re(p)$ or $\Im(p)$. Therefore, quantizing g_1 and g_2 uniformly leads to uniform pole density in the real and imaginary-axis directions. A sample grid showing quantized locations is shown in Fig. 6.2(b). Poles may only be implemented at the intersection of the vertical grid lines and horizontal grid lines.

This form often leads to more precise implementation with fewer bits precision. The price we pay for improved fidelity is the increased complexity of the second-order section.

6.4 PRELAB ASSIGNMENT

Verify the transfer functions of the Direct-Form II section in Fig. 6.1 on page 6–2 and the 1X-section in Fig. 6.3 on page 6–4.

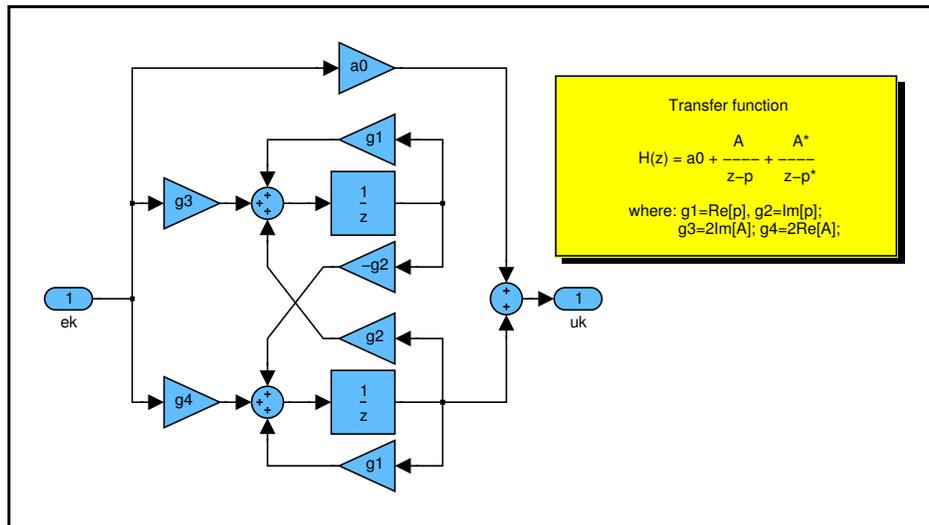


Figure 6.3 1X Form second-order section.

6.5 LABORATORY EXPERIMENT

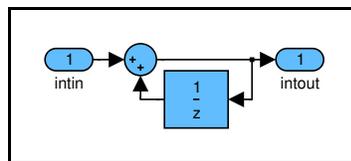


Figure 6.4 Integrator form without coefficient quantization.

For this experiment, you will use a sampling rate of 50 Hz. A controller which provides acceptable performance for all four MagLevs is

$$D(z) = 2.7896 \frac{z^2 - 1.6923z + 0.8918}{z^2 + 0.2114z + 0.5108} \frac{z}{z-1}.$$

The integrator term, $z/(z-1)$, may be implemented as shown in Fig. 6.4, without worry of coefficient quantization. The constant gain 2.7896 may be implemented as a multiply by four (shift left two bits, without quantization) followed by a multiply by 0.6974. In practice, the constant 0.6974 will need to be quantized to fit the number of bits available. Finally, the second-order section may be realized in either Direct-Form-II or in 1X-Form. All coefficients must be quantized to the number of bits available. Coefficients greater than one may be implemented in a similar way to the constant gain of 2.7896, above.

1. Implement the controller $D(z)$ using the Simulink discrete-time transfer-function block (without coefficient quantization—this is the baseline test). Use an equilibrium height of 2.5 cm. Plot the step response of the system (with the step starting at 5 sec).
2. Implement the controller $D(z)$ using Direct Form II, and quantized coefficients. Record step responses for decreasing precision until performance is badly degraded.
3. Repeat step 2 using 1X-Form.

Note, in order to quantize a number in Matlab,

```
quantval = quant( preciseval, 2^-(nobits-1) );
```

where: `preciseval` is the precise value to be quantized, `quantval` is the output quantized value and `nobits` is the number of bits available to store the number. The term $(nobits-1)$ reflects the fact that one of the available bits is used as sign information.

6.6 ASSIGNMENT

Discuss all results. Particularly, do you notice any benefit using 1X-Form versus Direct Form II? How many bits precision did you need when implementing coefficients? Is floating point “overkill” from the point of view of this control design (Note that Matlab’s double-precision values require 64 bits to store)? Plot the design pole-zero mapping and the implemented pole-zero mapping for each resolution attempted.

This is the end of Unit 2. You have the next lab period off to give you time to write your lab report. Your formal laboratory report *covering all three labs in this unit* is due at the beginning of the subsequent lab period.

(mostly blank)

Unit 3

Transfer Function Controller Design

This unit of labs focuses on digital control system design using transfer-function design methods. The MagLev will be configured in a two-disk single-input multi-output (SIMO) configuration. In:

- Lab 7 you will use standard frequency-response design methods; in
- Lab 8 you will design a controller using numeric optimization methods; in
- Lab 9 you will use Ragazzini's direct method to design a controller.

Remember that if a control-system design is properly done, testing is very fast. The majority of your time will be spent on the Prelab assignment, designing the controller in the first place.

(mostly blank)

Frequency-Response Controller Design

7.1 INTRODUCTION

In this unit of labs you will consider the MagLev configured with two magnets stacked on the glass rod. Both magnets are free to move but you will only be interested in controlling the position of the lower magnet, and will use the lower sensor as the feedback information.

In mechanical systems when the primary feedback sensor is rigidly coupled to the actuator, the system is referred to as being collocated. When some flexible structural member exists between the actuator and the sensor,¹ the system is referred to as being noncollocated. Systems with flexibly coupled inertia occur commonly in industry. Examples include flexible drive shafts, conveyor belts, and other mechanical linkages. An example of collocated control in such cases is when the feedback sensor is integral with the drive motor even though there may be flexible elements that are driven by the motor. An example of noncollocated control is when a flexible conveyor belt position is fed back for closed loop control. The configuration you will be investigating is collocated control.

7.2 THIS LAB IS...

- Control of a higher-order system, using frequency-response methods,
- An example of single-input multi-output (SIMO) control.

7.3 BACKGROUND

7.3.1 Model of the two-disk system

In previous labs, you controlled the MagLev with a single disk on the glass rod. In this and future labs, two disks will be placed on the glass rod. We must modify the plant model to consider the full dynamic interaction between them. The linearized equations from App. B give us a system model:

$$\begin{aligned} m\ddot{y}_1^* + m\beta\dot{y}_1^* + (k_{11} - k_{21} + k_{m12})y_1^* - k_{m12}y_2^* &= k_{u11}u_1^* + k_{u12}u_2^* \\ m\ddot{y}_2^* + m\beta\dot{y}_2^* + (k_{22} - k_{12} + k_{m12})y_2^* - k_{m12}y_1^* &= k_{u22}u_2^* + k_{u21}u_1^*. \end{aligned}$$

Let $u_2^* = 0$ since we are only using the lower coil as input. An additional effect of $u_2^* = 0$ is that $k_{21} = k_{22} = 0$.

Take Laplace transforms:

$$\begin{aligned} ms^2Y_1(s) + m\beta sY_1(s) + (k_{11} + k_{m12})Y_1(s) - k_{m12}Y_2(s) &= k_{u11}U_1(s) \\ ms^2Y_2(s) + m\beta sY_2(s) + (k_{m12} - k_{12})Y_2(s) - k_{m12}Y_1(s) &= k_{u21}U_1(s) \\ Y_2(s) &= \frac{k_{u21}U_1(s) + k_{m12}Y_1(s)}{ms^2 + m\beta s + k_{m12} - k_{12}}. \end{aligned}$$

Substituting into the first equation,

¹There may be multiple sensors used including ones that are both collocated and noncollocated relative to the actuator. If the location of the control objective is not collocated with the actuator, then the plant is typically referred to as being noncollocated.

$$\left(ms^2 + m\beta s + k_{11} + k_{m12} - \frac{(k_{m12})^2}{ms^2 + m\beta s + k_{m12} - k_{12}} \right) Y_1(s) = \left(k_{u1} + \frac{k_{m12}k_{u21}}{ms^2 + m\beta s + k_{m12} - k_{12}} \right) U_1(s).$$

This gives us the transfer function

$$\frac{Y_1(s)}{U_1(s)} = \frac{k_{u11}ms^2 + k_{u11}m\beta s + k_{u11}k_{m12} - k_{u11}k_{12} + k_{m12}k_{u21}}{m^2s^4 + 2m^2\beta s^3 + m(m\beta^2 + k_{11} + 2k_{m12} - k_{12})s^2 + m\beta(k_{11} + 2k_{m12} - k_{12})s + k_{11}k_{m12} - k_{11}k_{12} - k_{m12}k_{12}}.$$

The open-loop system is analogous to that of Fig. 7.1. The effective “spring” k_{m12} is the linearized term resulting from the forces of gravity and magnetic repulsion acting on the second magnet. Recall however that this term is in fact quite nonlinear. The addition of the spring and second inertia to the rigid body case studied above increases the plant order by two and adds an oscillatory mode to the plant dynamics.

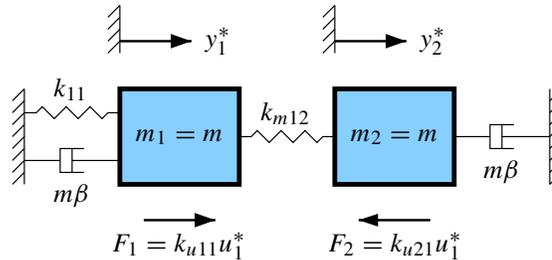


Figure 7.1 Analogous system to idealized SIMO plant.

7.4 PRELAB ASSIGNMENT

In this lab, the MagLev will be operated around the equilibrium point $y_{1o} = 1 \text{ cm}$. The inter-magnet separation is nominally $y_{12o} = 8 \text{ cm}$. $y_{2o} = y_{1o} + y_{12o} - 13$.

1. Compute u_{1o} such that $y_{1o} = 1 \text{ cm}$. **It is very important in this configuration to keep y_{1o} small. The weight of both magnets plus the control actuation force is supplied by the lower coil. The required current increases dramatically with height of the lower magnet. Excessive amplitude for sustained duration will overheat the coil and damage the apparatus.** Note: from the equilibrium condition in App. B, and multiplying by 100 to get the correct units, we know that

$$u_{1o} = a(y_{1o} + b)^4 \left(100mg + \frac{c}{(y_{12o} + d)^4} \right).$$

2. Determine the transfer function $G(s)$ of the two-disk system; that is, fill in the numbers and be careful with units. Normalize the transfer function (divide numerator and denominator by m^2) so that the leading coefficient of the denominator is one.

In the following parts, automate your design using Matlab m-file(s) so that you can easily redesign for different sampling rates, etc.

3. Determine the transfer function $G(z)$ of the two-disk system (You do not have to do this by hand). Use a sampling rate of 50 Hz .
4. Your z -plane controller will have the form

$$D(z) = K_d \frac{z - z_0}{z - z_p} \frac{Tz}{z - 1},$$

which is a lead controller cascaded with an integrator. The integrator eliminates steady-state error and the lead controller changes the dynamics. For the purpose of designing the lead controller, the fixed $Tz/(z - 1)$ term is added to the “plant” dynamics. State the new $G'(z)$ (with the integrator added), and convert $G'(z)$ to $G'(w)$.

7.6 ASSIGNMENT

Report your controller design. Discuss the transient and steady-state response of your closed-loop system. Explain the behavior of the system when you manually disturb the lower disk for a lengthy period of time. Discuss the disturbance-canceling performance of your system. How slowly can you sample and still get reasonable performance? What sampling rate do you recommend using?

Numeric Optimal PID Controller Design

8.1 INTRODUCTION

A generic PID controller implements a difference equation of the form

$$u[k] = u[k - 1] + q_0 e[k] + q_1 e[k - 1] + q_2 e[k - 2].$$

The values $\{q_0, q_1, q_2\}$ represent some set of PID coefficients $\{K, T_D, T_I\}$. In order to make a PID control design to optimize some performance criteria, we need to optimize the values of $\{K, T_D, T_I\}$ or, equivalently, the values of $\{q_0, q_1, q_2\}$. In this lab you will use a numeric PID optimization routine to find values of $\{q_0, q_1, q_2\}$ to give good control results.¹

8.2 THIS LAB IS...

- An introduction to numeric control-design techniques.
- Another example of single-input multi-output (SIMO) control.

8.3 BACKGROUND

8.3.1 Numeric Design of PID Controllers

Any digital controller is implemented via difference equations. For example, a PID controller is implemented via the equations

$$u[k] = u[k - 1] + K \left[\left(1 + \frac{T}{T_I} + \frac{T_D}{T} \right) e[k] - \left(1 + \frac{2T_D}{T} \right) e[k - 1] + \frac{T_D}{T} e[k - 2] \right] \quad (8.1)$$

or,

$$u[k] = u[k - 1] + q_0 e[k] + q_1 e[k - 1] + q_2 e[k - 2]$$

where q_0, q_1 , and q_2 are constants. In order to optimize the controller to meet some desired specifications, we must choose the coefficients $\{q_0, q_1, q_2\}$.

One method to do this is to “fiddle” with the coefficients in an intelligent way in order to get the closed-loop transfer function to resemble some desired closed-loop transfer function as closely as possible. We compare the output of our controlled system with the desired system, and the difference in the output at each time sample is $e[k]$. We can then compute a “cost function” J such as

$$J = \sum_k |e[k]|^2.$$

If we design our controller to minimize J then the output of our controlled system approximates the desired output as closely as possible.

¹Incidentally, this is how the controllers in Labs 2 and 6 were developed. They were designed to optimize the average performance of the controller over the four MagLevs.

8.3.2 Function Minimization in Three(+) Dimensions

Our PID controller design problem is an optimization problem in three variables. We start out with an initial guess as to good values for $q = \{q_0, q_1, q_2\}$. Then, we approximate the gradient ∇J_q of J with respect to q . If we change our q values in the direction of the negative gradient, we are descending down the error surface toward a minimum. Specifically:

1. Start with a guess. $q = \{q_0, q_1, q_2\}$.
2. Set the “step size” to a small value. Say $\gamma = 10^{-3}$, and set the gradient “test step sizes” $\delta q_0 = \delta q_1 = \delta q_2$ to a value smaller than γ .
3. Estimate the slope in the q_0 -direction:

$$\frac{\partial J}{\partial q_0} \approx \frac{J(q_0 + \delta q_0, q_1, q_2) - J(q_0, q_1, q_2)}{\delta q_0}$$

4. Estimate the slope in the q_1 -direction:

$$\frac{\partial J}{\partial q_1} \approx \frac{J(q_0, q_1 + \delta q_1, q_2) - J(q_0, q_1, q_2)}{\delta q_1}$$

5. Estimate the slope in the q_2 -direction:

$$\frac{\partial J}{\partial q_2} \approx \frac{J(q_0, q_1, q_2 + \delta q_2) - J(q_0, q_1, q_2)}{\delta q_2}$$

6. Normalize the gradient for better performance

$$\Delta = \sqrt{\left(\frac{\partial J}{\partial q_0}\right)^2 + \left(\frac{\partial J}{\partial q_1}\right)^2 + \left(\frac{\partial J}{\partial q_2}\right)^2}$$

7. Update q_0, q_1 and q_2 :

$$q_0 \leftarrow q_0 - \frac{\gamma}{\Delta} \left(\frac{\partial J}{\partial q_0}\right); \quad q_1 \leftarrow q_1 - \frac{\gamma}{\Delta} \left(\frac{\partial J}{\partial q_1}\right); \quad q_2 \leftarrow q_2 - \frac{\gamma}{\Delta} \left(\frac{\partial J}{\partial q_2}\right).$$

8. Repeat from (3) until convergence.

Note that every evaluation of $J(x, y, z)$ requires that you evaluate the system performance for the controller defined by the parameters $\{x, y, z\}$. Four separate evaluations are required per iteration of the above loop ($J(q_0, q_1, q_2)$ needs be evaluated only once). In the case of trying to match a step-response design criteria, four step responses need be simulated per iteration.

8.3.3 Example

The following code segment is similar to what you will need to write for this lab. The goal is to make the step response of the digital plant $G(z) = \frac{z^2+0.2z+0.02}{(z+0.8)(z^2+0.6z+0.36)}$ resemble the ideal unit step $1[k]$ as closely as possible.

```
g=zpk([0.1+0.1*j 0.1-0.1*j],[0.3+0.5*j 0.3-0.5*j 0.8],1,-1);
gamma=0.01; dq=0.001; T=20;
des = ones([T+1 1]); % desired step response

q0 = 0; q1 = 0; q2 = 0
```

```

maxiter=410; J=zeros([1 maxiter]);

for i=1:maxiter,
    d=tf([q0+dq q1 q2],[1 -1 0],-1);
    t=feedback(d*g,1); s=step(t,T+1); J1=norm(des-s);

    d=tf([q0 q1+dq q2],[1 -1 0],-1);
    t=feedback(d*g,1); s=step(t,T+1); J2=norm(des-s);

    d=tf([q0 q1 q2+dq],[1 -1 0],-1);
    t=feedback(d*g,1); s=step(t,T+1); J3=norm(des-s);

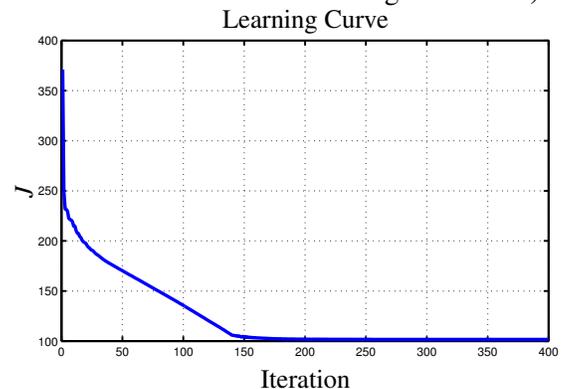
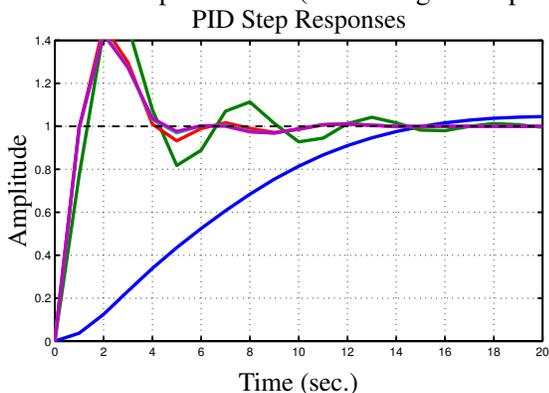
    d=tf([q0 q1 q2],[1 -1 0],-1);
    t=feedback(d*g,1); s=step(t,T+1); J(i)=norm(des-s);
    if (mod(i-10,100)==0),
        J(i), plot(0:T,s); axis([0 T 0 1.4]); drawnow;
    end;

    p1=(J1-J(i))/dq; p2=(J2-J(i))/dq; p3=(J3-J(i))/dq;
    Delta=sqrt(p1*p1+p2*p2+p3*p3);

    q0=q0-(gamma/Delta)*p1; q1=q1-(gamma/Delta)*p2; q2=q2-(gamma/Delta)*p3;
end;

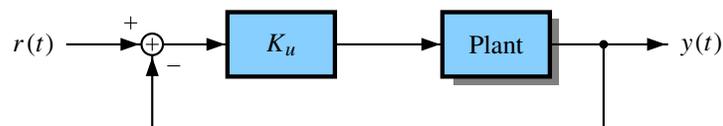
```

The left plot below shows the step response of the system evolving over time, and the right plot shows the “learning curve” of the optimization (A learning curve plots J versus iteration to show how the cost is being minimized).

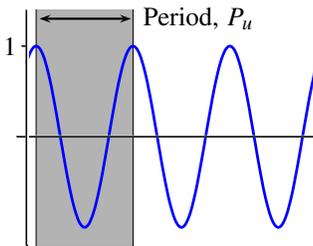


8.3.4 Initial Guess for K , T_I and T_D :

In the case of designing a PID controller numerically, the Ziegler-Nichols rules may be used to provide a reasonable initial guess for $\{q_0, q_1, q_2\}$. Recall the “second method” of Ziegler and Nichols: Configure your plant as:



Turn up gain K_u until system produces oscillations (on stability boundary) $K_u =$ “ultimate gain.” Measure the period of oscillations: P_u is the period. The K , T_I and T_D parameters may then be chosen as shown in the table below (Convert to q_0, q_1, q_2 using Eq. (8.1).)



RESULTING TUNING RULES:

P	PI	PID
$K = 0.5K_u$	$K = 0.45K_u$	$K = 0.6K_u$
$T_I = \infty$	$T_I = \frac{1}{1.2}P_u$	$T_I = 0.5P_u$
$T_D = 0$	$T_D = 0$	$T_D = \frac{P_u}{8}$

8.4 PRELAB ASSIGNMENT

In this lab, the MagLev will be configured identically to Lab 7. Again, it will be operated around the equilibrium point $y_{1o} = 1 \text{ cm}$. The inter-magnet separation is nominally $y_{12o} = 8 \text{ cm}$.

1. Recall u_{1o} such that $y_{1o} = 1 \text{ cm}$; the transfer function $G(s)$ and the transfer function $G(z)$ from Lab 7. Use a sampling rate of 50 Hz .
2. Determine initial conditions for the numeric optimization $\{q_0, q_1, q_2\}$ using the approximate values $K_u = 0.4$ and $P_u = 0.3 \text{ sec}$.
3. Your controller will implement the difference equation

$$u[k] = u[k - 1] + q_0e[k] + q_1e[k - 1] + q_2e[k - 2].$$

Use a numeric optimization routine to find $\{q_0, q_1, q_2\}$ which give closed-loop step response most closely resembling the step response of the analog prototype system

$$T(s) = \frac{20}{s + 20}.$$

Minimize the cost function $J = \sum_k |e_k|^2$.

- (a) State the transfer function $D(z)$ of the controller you find.
- (b) Plot the step response of your optimized system.
- (c) Plot the “learning curve” showing how J decreases versus iteration.

8.5 LABORATORY EXPERIMENT

1. Implement your feedback control system in Simulink.
2. For a reference input equal to a unit step at time 5 sec , record your system output.
3. See how well your design rejects disturbance by repeating Lab 7 experimental step 6.

8.6 ASSIGNMENT

Report your controller design. Discuss the transient and steady-state response of your closed-loop system. How well does the physical system correspond with your design?

Ragazzini's Direct Control Design Method

9.1 INTRODUCTION

Many of the transfer-function design techniques we have discussed (lead/lag, etc) grew out of the limitations of technology to implement continuous-time compensators using resistors, capacitors and the like. With controllers realized by digital computer, such limitations are not relevant any more. An alternative design method is the direct control design method of Ragazzini which directly *computes* the controller $D(z)$ to achieve some desired closed-loop transfer function $T(z)$.

9.2 THIS LAB IS...

- Practice using Ragazzini's direct control design method.
- Control system design via pole-placement and prototype emulation.

9.3 BACKGROUND

9.3.1 Direct Design Method of Ragazzini

The closed-loop transfer function of a unity-feedback digital control system is

$$T(z) = \frac{D(z)G(z)}{1 + D(z)G(z)}.$$

Note that we can re-arrange this expression and solve for $D(z)$ to give this closed-loop transfer function.

$$\begin{aligned} (1 + D(z)G(z))T(z) &= D(z)G(z) \\ D(z)G(z)(T(z) - 1) &= -T(z) \\ D(z) &= \frac{1}{G(z)} \frac{T(z)}{1 - T(z)}. \end{aligned}$$

From this equation, it appears as if we can specify a desired $T(z)$ and simply compute $D(z)$ to achieve it! Can control design be this simple?

In general, the answer is "not quite". The problem is that this technique may ask for the impossible: (non-causal, unstable ...). We need to impose some constraints on $T(z)$ for the control system to be implementable.

Causality: If $D(z)$ is causal, then it has no poles at ∞ (equivalently, no zeros at the origin). Therefore, $T(z)$ must have enough zeros at ∞ to cancel out poles at ∞ from $\frac{1}{G(z)}$. Put another way, the delay in $T(z)$ must be at least as long as the delay in $G(z)$. The constraint on $T(z)$ is then

- $T(z)$ must have a zero at infinity of the same order as the order of the zero of $G(z)$ at infinity.

Stability: If $G(z)$ has unstable poles, they cannot be canceled directly by $D(z)$ or there will be trouble! To show this, consider the characteristic equation of the closed-loop system

$$1 + D(z)G(z) = 0.$$

Let $D(z) = c(z)/d(z)$ and $G(z) = b(z)/a(z)$. Then

$$1 + \frac{c(z)}{d(z)} \frac{b(z)}{a(z)} = 0.$$

Let the unstable pole in $G(z)$ be at α , so $a(z) = (z - \alpha)\bar{a}(z)$. To cancel it, $c(z) = (z - \alpha)\bar{c}(z)$, and

$$\begin{aligned} (z - \alpha)\bar{a}(z)d(z) + (z - \alpha)\bar{c}(z)b(z) &= 0 \\ (z - \alpha)[\bar{a}(z)d(z) + \bar{c}(z)b(z)] &= 0. \end{aligned}$$

The unstable root is still a factor of the characteristic equation! (oops). We conclude that unstable poles may not be directly canceled, but must be moved via the feedback mechanism. This imposes two constraints on $T(z)$.

- $[1 - T(z)]$ must contain as zeros all the poles of $G(z)$ outside the unit circle.
- $T(z)$ must contain as zeros all the zeros of $G(z)$ outside the unit circle.

Steady-State Accuracy: Finally, requirements for steady-state accuracy will impose some constraints on $T(z)$. Assume that the system is to be of type-I with velocity constant K_v . Note:

$$E(z) = [1 - T(z)]R(z).$$

We must have zero steady-state error to a step. Therefore

$$\lim_{z \rightarrow 1} (z - 1)[1 - T(z)] \frac{z}{(z - 1)} = 0$$

or $T(1) = 1$. Furthermore, we must have $1/K_v$ error to a unit ramp. Therefore

$$\lim_{z \rightarrow 1} (z - 1)[1 - T(z)] \frac{Tz}{(z - 1)^2} = \frac{1}{K_v}.$$

Use l'hôpital's rule to evaluate:

$$-T \left. \frac{dT(z)}{dz} \right|_{z=1} = \frac{1}{K_v}.$$

So, the final constraints on $T(z)$ are

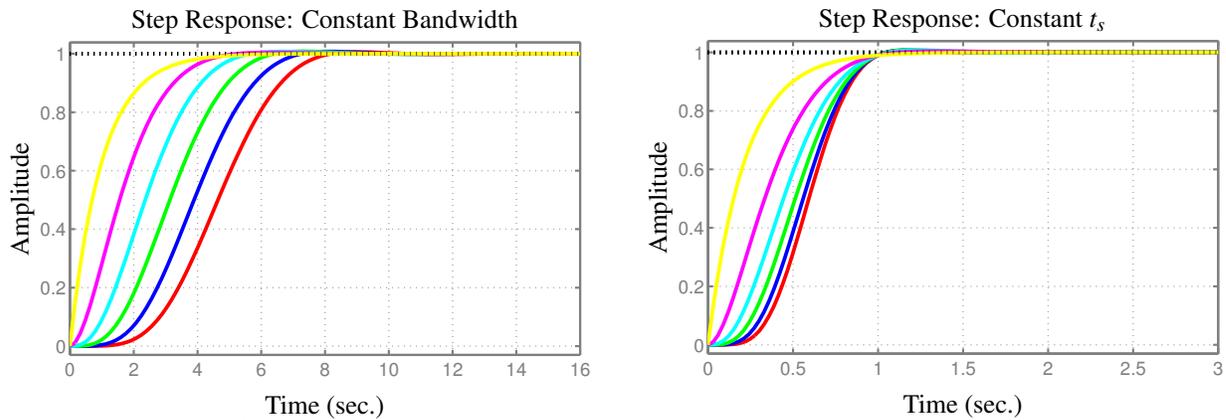
- $T(1) = 1$.
- $\left. \frac{dT(z)}{dz} \right|_{z=1} = -\frac{1}{TK_v}$.

Any $T(z)$ which meets these requirements for causality, stability and steady-state accuracy is one which will provide a functional $D(z)$.

9.3.2 Pole Placement

Now that we have this considerable tool at our disposal, we need to determine where we should place the closed-loop poles of $T(z)$. One possibility is to choose closed-loop poles to mimic a system that has performance that you like. Set closed-loop poles equal to this prototype system.

Prototypes are tabulated with poles scaled to give settling time of 1 sec. or bandwidth of $\omega = 1$ rad/sec. For example, the following two sets of step responses have constant settling time and constant bandwidth:



The following tables list the normalized s -plane pole locations.

Bessel pole locations for $t_s = 1$ sec.

1:	-4.6200		
2:	-4.0530 ± 2.3400 j		
3:	-3.9668 ± 3.7845 j	-5.0093	
4:	-4.0156 ± 5.0723 j	-5.5281 ± 1.6553 j	
5:	-4.1104 ± 6.3142 j	-5.9268 ± 3.0813 j	-6.4480
6:	-4.2169 ± 7.5300 j	-6.2613 ± 4.4018 j	-7.1205 ± 1.4540 j

Bessel pole locations for $\omega_o = 1$ rad/sec.

1:	-1.0000		
2:	-0.8660 ± 0.5000 j		
3:	-0.7455 ± 0.7112 j	-0.9420	
4:	-0.6573 ± 0.8302 j	-0.9047 ± 0.2711 j	
5:	-0.5906 ± 0.9072 j	-0.8516 ± 0.4427 j	-0.9264
6:	-0.5385 ± 0.9617 j	-0.7998 ± 0.5622 j	-0.9093 ± 0.1856 j

In order to convert these pole locations to something useful, use the following procedures:

PROCEDURE: For n th-order system—desired bandwidth.

1. Determine desired bandwidth ω_d .
2. Find the n th-order poles from the table of constant bandwidth, and *multiply* pole locations by ω_d .
3. Convert s -plane locations to z -plane locations using $z = e^{sT}$.
4. Ensure $T(z)$ meets design criteria and use Ragazzini to compute controller.

PROCEDURE: For n th-order system—desired settling time.

1. Determine desired settling time t_d .
2. Find the n th-order poles from the table of constant settling time, and *divide* pole locations by t_d .
3. Convert s -plane locations to z -plane locations using $z = e^{sT}$.
4. Ensure $T(z)$ meets design criteria and use Ragazzini to compute controller.

9.4 PRELAB ASSIGNMENT

In this lab you will again control the two-disk MagLev in the configuration of Labs 7 and 8.

1. Recall u_{1o} such that $y_{1o} = 1 \text{ cm}$; the transfer function $G(s)$ and the transfer function $G(z)$ from Lab 7. Use a sampling rate of 50 Hz .
2. Design a $T(z)$ for settling time 0.2 sec and of appropriate order (number of poles). Make sure that all the requirements for stability, causality and steady-state error are met (choose an appropriate value of K_v).
3. Compute $D(z)$. Simulate and plot the step response of the closed-loop system. Plot the step response of $T(z)$. Do they match?

9.5 LABORATORY EXPERIMENT

1. Implement your feedback control system in Simulink.
2. For a reference input equal to a unit step at time 5 sec , record your system output.

9.6 ASSIGNMENT

Report your controller design. Discuss the transient and steady-state response of your closed-loop system. How well does the physical system correspond with your design?

This is the end of Unit 3. You have the next lab period off to give you time to write your lab report. Your formal laboratory report *covering all three labs in this unit* is due at the beginning of the subsequent lab period.

Unit 4

State-Space Controller Design

This final unit of labs focuses on digital control system design using state-space design methods. The MagLev will be configured in a two-disk multi-input multi-output (MIMO) configuration. In:

- Lab 10 you will use state feedback to control the MagLev; in
- Lab 11 you will use output feedback to control the MagLev.

Remember that if a control-system design is properly done, testing is very fast. The majority of your time will be spent on the Prelab assignment, designing the controller in the first place.

(mostly blank)

State-Feedback Controller Design

10.1 INTRODUCTION

In this experiment, you will implement full multi-input multi-output (MIMO) control on the two magnet MagLev system. The physical plant configuration is again that of the previous three labs except that now you will apply inputs at both the upper and lower magnet locations. You will use state-feedback control design methods in order to design the controller for this system.

10.2 THIS LAB IS...

- An introduction to MIMO state-space control.
- Another instance of control system design via pole-placement and prototype emulation.

10.3 BACKGROUND

10.3.1 A full MIMO state-space model of the MagLev (Continuous-time)

Linear, lumped, time invariant systems may be represented in state-space form as

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t),\end{aligned}$$

where A , B , C and D are constant matrices, $x(t)$ is a vector function of time, and $u(t)$ is a vector of inputs. We wish to represent the MagLev dynamics in state-space form; that is, we must define a state vector $x(t)$ and find matrices A , B , C and D .

From App. B we know that the full model of the MagLev is:

$$\begin{aligned}m\ddot{y}_1^* + m\beta\dot{y}_1^* + (k_{11} - k_{21} + k_{m12})y_1^* - k_{m12}y_2^* &= k_{u11}u_1^* + k_{u12}u_2^* \\ m\ddot{y}_2^* + m\beta\dot{y}_2^* + (k_{22} - k_{12} + k_{m12})y_2^* - k_{m12}y_1^* &= k_{u22}u_2^* + k_{u21}u_1^*.\end{aligned}$$

where the appropriate constants are defined in App. B.

If we define the state vector $x(t)$ to be

$$x(t) = [y_1^*(t) \quad y_2^*(t) \quad \dot{y}_1^*(t) \quad \dot{y}_2^*(t)]^T,$$

then we can represent the equations of motion for the MagLev in state space as

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ (k_{21} - k_{11} - k_{m12})/m & k_{m12}/m & -\beta & 0 \\ k_{m12}/m & (k_{12} - k_{22} - k_{m12})/m & 0 & -\beta \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{u11}/m & k_{u12}/m \\ k_{u21}/m & k_{u22}/m \end{bmatrix} \begin{bmatrix} u_1^*(t) \\ u_2^*(t) \end{bmatrix} \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x(t).\end{aligned}$$

Therefore, the full MIMO model is:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ (k_{21} - k_{11} - k_{m12})/m & k_{m12}/m & -\beta & 0 \\ k_{m12}/m & (k_{12} - k_{22} - k_{m12})/m & 0 & -\beta \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{u11}/m & k_{u12}/m \\ k_{u21}/m & k_{u22}/m \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

10.3.2 A full MIMO state-space model of the MagLev (Discrete-time)

Recall from class that in order to convert the MagLev dynamics to discrete time, we must perform the following computations:

$$A_d = e^{AT}, \quad B_d = A^{-1}(A_d - I)B,$$

$$C_d = C, \quad D_d = D.$$

This gives us the discrete time system

$$x[k+1] = A_d x[k] + B_d u[k]$$

$$y[k] = C_d x[k] + D_d u[k].$$

The interpretation of the state is

$$x[k] = \begin{bmatrix} y_1^*(kT) \\ y_2^*(kT) \\ \dot{y}_1^*(kT) \\ \dot{y}_2^*(kT) \end{bmatrix}.$$

10.3.3 Calculating Equilibrium u_{1o} and u_{2o}

We may calculate the steady-state control effort u_{1o} and u_{2o} for a specific operating point $\{y_{1o}, y_{2o}\}$ by evaluating the equilibrium equations in App. B. These give the two equations and two unknowns which may be combined into a matrix equation:

$$\begin{bmatrix} \frac{1}{a(y_{1o}+b)^4} & \frac{1}{a(13-y_{1o}+b)^4} \\ \frac{-1}{a(13+y_{2o}+b)^4} & \frac{-1}{a(-y_{2o}+b)^4} \end{bmatrix} \begin{bmatrix} u_{1o} \\ u_{2o} \end{bmatrix} = \begin{bmatrix} mg + \frac{c}{(y_{12o}+d)^4} \\ mg - \frac{c}{(y_{12o}+d)^4} \end{bmatrix}$$

$$\begin{bmatrix} u_{1o} \\ u_{2o} \end{bmatrix} = \begin{bmatrix} \frac{1}{a(y_{1o}+b)^4} & \frac{1}{a(13-y_{1o}+b)^4} \\ \frac{-1}{a(13+y_{2o}+b)^4} & \frac{-1}{a(-y_{2o}+b)^4} \end{bmatrix}^{-1} \begin{bmatrix} mg + \frac{c}{(y_{12o}+d)^4} \\ mg - \frac{c}{(y_{12o}+d)^4} \end{bmatrix}$$

Note that in order for the units to properly agree, a factor of 100 must be introduced:

$$\begin{bmatrix} u_{1o} \\ u_{2o} \end{bmatrix} = \begin{bmatrix} \frac{1}{a(y_{1o}+b)^4} & \frac{1}{a(13-y_{1o}+b)^4} \\ \frac{-1}{a(13+y_{2o}+b)^4} & \frac{-1}{a(-y_{2o}+b)^4} \end{bmatrix}^{-1} \begin{bmatrix} 100mg + \frac{c}{(y_{12o}+d)^4} \\ 100mg - \frac{c}{(y_{12o}+d)^4} \end{bmatrix}.$$

10.3.4 State feedback control via pole placement

Control of systems in state-space form uses state feedback rather than output feedback. The controller is designed by choosing a K matrix such that

$$u[k] = r[k] - Kx[k].$$

The poles of a state-space system are the eigenvalues of the A_d matrix. By employing state feedback, we change the state equation to

$$x[k+1] = (A_d - B_d K)x[k] + B_d r[k]$$

and therefore change the effective “ A_d ” matrix to $A_d - B_d K$. For many systems, we may choose K to place the eigenvalues of $A_d - B_d K$ anywhere we like (in complex-conjugate pairs, that is). The Matlab command `place()` may be used to compute the desired K matrix.

10.3.5 Steady-state accuracy

State-space design via pole placement addresses the transient-response of the system but does not directly address the steady-state response. For instance, it does not ensure zero steady-state error to step-like control inputs. There are a number of ways to improve steady-state response and the simplest is discussed here (also the least accurate!).

The dc-gain of the state-space system may be calculated to be

$$K_{dc} = C_d(I - A_d + B_d K)^{-1} B_d + D_d.$$

If we multiply the reference input by the inverse of the K_{dc} matrix, we compensate for any aberration from unity dc-gain.

10.4 PRELAB ASSIGNMENT

1. Determine the full state-space model of the continuous-time MagLev system. That is, fill in all numbers.
2. Convert the plant model to discrete-time. Use a sampling frequency of 50 Hz. You may use either the `expm` command or the `c2d` command in Matlab.
3. Compute the u_{1o} and u_{2o} voltages for nominal operating points $y_{1o} = 1.5 \text{ cm}$ and $y_{2o} = -2.0 \text{ cm}$. In this configuration, $y_{12o} = 9.5 \text{ cm}$.
4. You will need to place four poles using state feedback. Use a prototype from Lab 9 and design for a settling time of 0.5 sec. State the four desired pole locations.
5. Use Matlab’s `place` command to determine a state-feedback gain matrix

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \end{bmatrix}.$$

6. Compute the dc-gain of your compensated system K_{dc} . Also compute the inverse: K_{dc}^{-1} .

10.5 LABORATORY EXPERIMENT

Consider the state-feedback MIMO controller as illustrated in Fig. 10.1 on page 10–4. The state is composed for feedback by actually measuring $y_1^*[k]$ and $y_2^*[k]$, and by estimating the derivatives $\dot{y}_1^*[k]$ and $\dot{y}_2^*[k]$ using Euler’s backward rule:

$$\dot{y}_1^*[k] \approx \frac{y_1^*[k] - y_1^*[k-1]}{T} \quad \text{and} \quad \dot{y}_2^*[k] \approx \frac{y_2^*[k] - y_2^*[k-1]}{T}.$$

The four components of the state vector are passed through a multiplexer to make the vector signal $x[k]$, which is multiplied by the matrix K (be sure to use a matrix gain block from the Math Panel, and not a regular scalar gain), and demultiplexed into the two components which modify $u[k]$. The reference inputs are multiplexed into a single $r[k]$ vector, which is multiplied by K_{dc}^{-1} to remove steady-state error, and demultiplexed into individual components which are used for each control input.

1. Set the nominal operating points to $y_{1o} = 1.5 \text{ cm}$ and $y_{2o} = -2.0 \text{ cm}$.

State Estimation and Control Design

11.1 INTRODUCTION

State feedback control is a very powerful design technique as it often allows us to place closed-loop poles anywhere we desire. The price we pay for this power is that we must measure all of the system states. In Lab 10, measuring the two states $y_1^*[k]$ and $y_2^*[k]$ was simple (although even these are corrupted by sensor noise). However, the state derivatives had to be approximated by

$$\dot{y}_1^*[k] \approx \frac{y_1^*[k] - y_1^*[k-1]}{T} \quad \text{and} \quad \dot{y}_2^*[k] \approx \frac{y_2^*[k] - y_2^*[k-1]}{T}.$$

State-space design may be accomplished more accurately if the state of the system is estimated using a special circuit called an *estimator* or an *observer*. The output of the estimator is the state estimate $\hat{x}[k]$. Feedback control is accomplished via $u[k] = r[k] - K\hat{x}[k]$ rather than $u[k] = r[k] - Kx[k]$. Even though a state approximation is used, stable control is guaranteed if the original control poles were stable and the estimator poles are stable.

11.2 THIS LAB IS...

- Your final control design for this course, incorporating many aspects of state-space theory including state estimation.

11.3 BACKGROUND

11.3.1 Estimator Design

We seldom measure all of the states of $x[k]$ in order to accomplish state-feedback. Instead, we use our knowledge of the plant dynamics and the plant input and output in order to estimate the state of the plant by simulating the plant behavior in real-time. Let $\hat{x}[k]$ be our estimate of the plant state. If we feed back $u[k] = r[k] - K\hat{x}[k]$ the poles of the closed-loop system are moved to the same place as if we feed back $u[k] = r[k] - Kx[k]$.

How do we design an estimator? Consider the following:

$$\hat{x}[k+1] = A_d\hat{x}[k] + B_d u[k] + L(y[k] - C_d\hat{x}[k])$$

where L is a $n \times m$ (number-of-states by number-of-outputs) matrix of gains chosen by the control-system designer. Call the error between the actual state and the estimated state $\tilde{x}[k] = x[k] - \hat{x}[k]$. Then, the error has dynamics

$$\begin{aligned} \tilde{x}[k+1] &= A_d x[k] + B_d u[k] - A_d \hat{x}[k] - B_d u[k] - L(C_d x[k] - C_d \hat{x}[k]) \\ &= (A_d - LC_d)\tilde{x}[k]. \end{aligned}$$

So, the estimator error has poles determined by the eigenvalues of the $(A_d - LC_d)$ matrix. By choosing the L gains, we can make the estimator error converge to zero as quickly as we desire. Often, the poles of the estimator are chosen to be slightly faster than the poles of the closed-loop plant. The estimator gain vector L is designed in exactly the same way as the controller gain vector K , but with A_d replaced by A_d^T , B_d replaced by C_d^T and finally by computing the transpose of the result. Specifically, for a set of desired pole locations `des_poles`,

$$L = \text{place}(A_d', C_d', \text{des_poles})';$$

11.4 PRELAB ASSIGNMENT

1. Recall your discrete-time system model, u_{1o} , u_{2o} , K and K_{dc} from Lab 10. You will need these again.
2. You will need to place four poles for the estimator design. Use a prototype from Lab 9 and design for a settling time of 0.1 sec. State the four desired pole locations and your estimator gain matrix L .

11.5 LABORATORY EXPERIMENT

Consider the MIMO controller with estimated-state feedback as illustrated in Fig. 11.1. The estimator at the bottom of the figure implements

$$\hat{x}[k+1] = A_d \hat{x}[k] + B_d u[k] - L(y[k] - C_d \hat{x}[k]).$$

The feedback is computed as $u[k] = K_{dc}^{-1} r[k] - K \hat{x}[k]$.

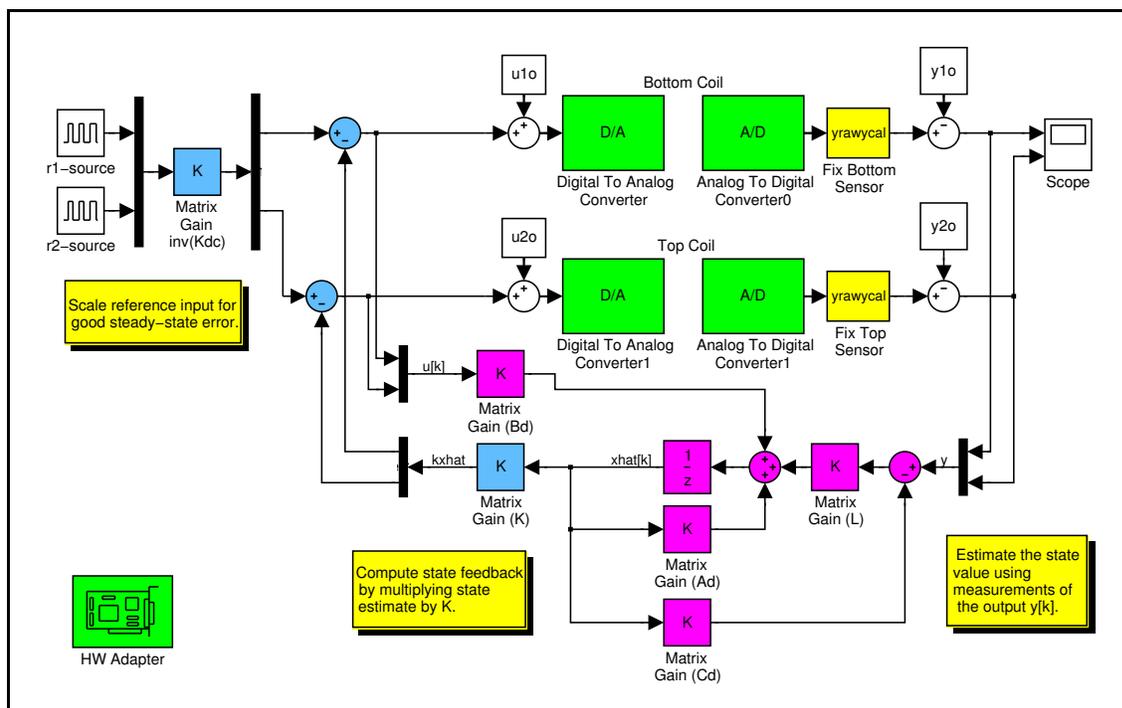


Figure 11.1 Diagram of state-feedback MIMO system.

1. Set the nominal operating points to $y_{1o} = 1.5 \text{ cm}$ and $y_{2o} = -2.0 \text{ cm}$.
2. Use the plastic safety clip so that the upper magnet rests at approximately -2.5 cm . Implement your controller (for zero reference input), and check that the system is stable and functioning properly (e.g., the magnets are levitating at roughly their proper heights).
3. Set up the following multivariable input: Lower-disk trajectory: Pulse Generator input, 1 cm amplitude, 4 sec period, 0 sec offset; Upper-disk trajectory: Pulse Generator input, 1 cm amplitude, 4 sec period, 1 sec offset. Execute these two trajectories. Notice the response of the magnets and the cross coupling between them (e.g., how much the stationary magnet moves as the moving one changes position). Plot your commanded position and y_j^* , $j = 1, 2$, data for each axis. Notice the cross coupling in the output and in which magnet positions it is most pronounced.

11.6 ASSIGNMENT

Report all Prelab and measured Experiment results. Describe the motion of the two magnets under MIMO state-space control from your pulse responses in terms of their relative interference when the two magnets are closest together and when they are furthest apart. Can you see the nonlinear effect of the “spring” in the outputs? Explain. Compare the results of this lab with those of Lab 10. Which are better? Explain.

This is the final lab. Your formal laboratory report *covering both labs in this unit* is due at 5:00p.m. Thursday 10 May 2001.

(mostly blank)

Congratulations!

You have just completed a rigorous lab course in digital control. You have studied discrete-time simulation; two methods of control design which emulate analog systems; effects of digital systems such as aliasing, discrete-time models and finite-precision effects; three methods of direct digital control design; and state-space controller and estimator design. You have implemented SISO, SIMO and MIMO controllers.

More importantly, you have implemented all these methods on a physical system. You have discovered that control experiments focus attention on performance and implementation issues that may be overlooked and at the least are difficult to capture by “theory.” For example:

- Control experiments highlight implementation issues (such as integrator overload, sensor noise, dynamic nonlinearities and actuator saturation) that are easy for theory to overlook.
- The difference between a “toy” control assignment and a “real” control experiment is whether the disturbance is of your own construction or is thrown at you by Mother Nature herself, and if unmodeled nonlinear effects are present.
- Control experiments provide a quick way to identify control methods that seem to work under real-world conditions and those that clearly don’t.

The last item—finding what does and does not work—leads to real learning.

I hope that you have enjoyed this lab, have found it interesting, and that you have learned a great deal.

Gregory Plett

August 2002

(mostly blank)

App. A: Magnetic Levitation Principles

This appendix gives a brief overview of key concepts in magnetism in general and magnetic levitation in particular. It includes equations and illustrations that show how magnetic fields and forces are generated within the Model 730 apparatus and concludes with a reference section. For a more rigorous treatment of these topics, the reader is referred to the literature listed at the end of the appendix.

A.1 Introduction

Magnetic fields are used to describe forces at a distance from electric currents. These currents are of two types:

1. Free, or Amperian, currents as drawn from a battery pack, power supply, or an electrical outlet, and
2. Bound currents as in permanent magnet materials.

The forces come in three variations:

1. An electrical current feels a force from another current,
2. A current feels a force from a permanent magnet, and
3. A permanent magnet feels a force from another permanent magnet.

This action at a distance is described by saying a magnetic field exists created by one of the bodies at the location of the other body. The magnetic field is the medium by which the force is transferred.

In this section, a brief discussion concerning the magnetic fields caused by magnetized materials (*i.e.*, permanent magnets) is presented. By demonstrating that magnetic materials can be reduced to effective current distributions, this discussion forms the basis for calculating the forces on permanent magnets. The magnetic fields due to free current distributions are calculated next. These fields are used to calculate the forces felt by current-carrying conductors. Time-varying currents cause time-varying magnetic fields. These changing magnetic fields induce electric currents that, in turn, experience a force.

Maglev systems utilize the fundamental physics of electric currents experiencing forces-at-a-distance. These systems are most often described in terms of the interaction of electrical current with magnetic fields. Because the masses of the vehicles are large, large forces are required for magnetic suspension. These large forces are provided by the high magnetic fields of either large superconducting currents or small air gaps in normal ferromagnetic circuits.

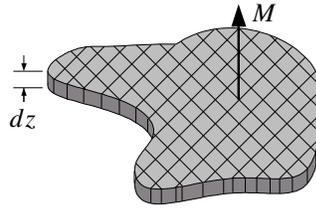
A dictionary of common terms is provided at the end of the section and a reference list is included featuring books, Maglev studies and Maglev URLs.

A.2 Magnetic Fields & Forces

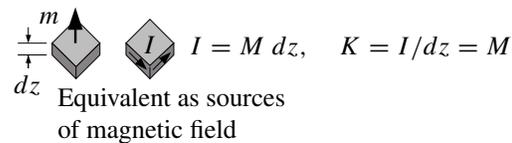
A.2.1 Magnetic Fields Caused by Magnetized Materials

Electron spin is a quantum mechanical phenomenon. Its significance here is the fact that there is associated with the electron spin a magnetic moment of fixed magnitude. To determine the forces on magnetic materials, we use the fact that the magnetic moment of the electron spin *acts as if* it is a current loop.

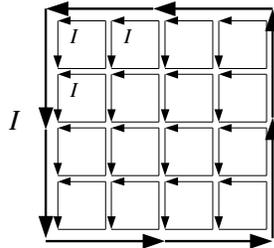
A volume of magnetized material contains a very large number of aligned electron spins. This is illustrated schematically in the figure below. A grid pattern was superposed onto the material to indicate small volumes of materials which can be analyzed discretely.



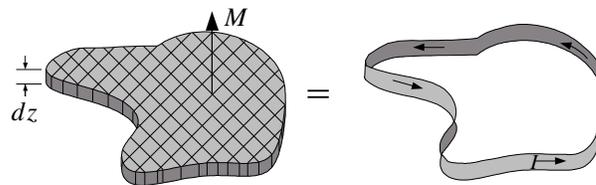
The figure below isolates two small volumes. The one on the left generates a magnetic field due to its magnetic moment, m . The one on the right generates a magnetic field by virtue of the current wrapping the volume (this is commonly referred to as a “current sheet.”), and otherwise ignores the presence of the magnetic material. **The two magnetic fields are entirely equivalent** (external to the material). The material property M describes the strength of the magnetic material and the amount of current required per unit distance of height. The parameter K describes the current per unit height within the current sheet. For modern high performance neodymium-iron-boron permanent magnets, $K \approx 900,000$ amps/meter.



Assembling many small volumes in juxtaposition, one can see in the figure below the result of substituting the current loop for the magnetic material; the internal currents cancel while the surface currents do not cancel. Hence, no matter the shape of the material, the external magnetic field can be exactly reproduced by a current stripe along the material perimeter. (This is true if the magnetization is uniform. If the magnetization is uniform then the currents are equal and cancel. If the magnetization is not uniform, then the equivalent current distribution can be calculated by $J = \nabla \times M$, where J is the volume current density.)



In summary, the magnetic field of a uniformly magnetized permanent magnet can be exactly reproduced by a current sheet along the perimeter. This equality is indicated graphically in the figure below. The magnitude of the current is proportional to the material thickness with the constant of proportionality dependent upon the material itself. This equivalence of magnetic fields is commonly exploited to calculate the forces on permanent magnet materials.



A.2.2 Calculating Magnetic Fields

The Biot-Savart Law is the fundamental relationship between current and magnetic field:

$$d\vec{B} = \frac{\mu}{4\pi} \frac{I d\vec{l} \times \vec{r}}{r^3}, \quad (\text{A.1})$$

where,

$d\vec{B}$ = differential magnetic field, tesla,

μ = permeability, Henry/m,

I = current, amps,

$d\vec{l}$ = differential length of current-carrying element, m,

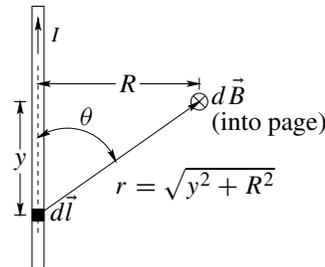
\vec{r} = vector distance from current element to field point, m.

Two simple geometries for calculating the magnetic field are an infinitely long, straight conductor and a circular loop of conductor.

Straight Conductor

For the straight conductor carrying current upwards along the y-direction, the magnetic field can be calculated at any point due to a differential current element as:

$$d\vec{B} = \frac{\mu}{4\pi} \frac{I d\vec{l} \sin \theta}{r^2}, \quad (\text{A.2})$$



By integrating along y from $-\infty$ to ∞ , the result is:

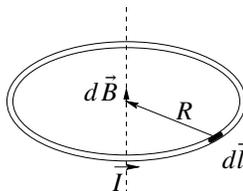
$$B = \frac{\mu I}{2\pi R},$$

where the direction of the magnetic field is determined by the right hand rule (point the thumb of the right hand along the direction of current and the fingers curl in the direction of the magnetic field).

Circular Loop at Center

The magnetic field at the center of a circular loop of wire can also be calculated from the Biot-Savart Law. In this case, the angle between the current element and field point is a constant (90°) so the vector cross product always yields unity. Then the magnetic field is:

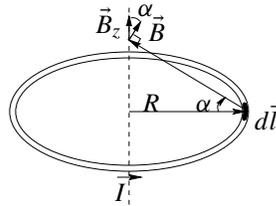
$$d\vec{B} = \frac{\mu}{4\pi} \frac{I 2\pi R}{R^2} = \frac{\mu I}{2R}. \quad (\text{A.3})$$



Circular Loop Anywhere Along Axis

The magnetic field along the axis of a circular loop of wire can also be calculated from the Biot-Savart Law. In this case, the angle between the current element and field point is still a constant (90°) so the vector cross product always yields unity. The net magnetic field from any current element has vertical and horizontal components. However, as the current element follows the conductor path, the horizontal components of the field cancel while the vertical components add. Then the axial magnetic field is:

$$B_z = \frac{\mu}{4\pi} \frac{I 2\pi R}{z^2 + R^2} \cos \alpha = \frac{\mu I R}{2(z^2 + R^2)} \frac{R}{\sqrt{z^2 + R^2}} = \frac{\mu I R^2}{2(z^2 + R^2)^{3/2}}. \quad (\text{A.4})$$



Due to the common occurrence of circular coils, this relationship is very important. Along the axis the magnetic field is purely vertical. For values of height, z , above the plane of the coil, which are small compared to the radius, R , ($z \ll R$) the vertical magnetic field is insensitive to z . This is due to the finite radius of the coil. For heights z much larger than R ($z \gg R$) the axial field decreases as the reciprocal third power of height. Off-axis, the radial magnetic field decreases as the reciprocal fourth power of height.

Calculating Magnetic Forces

The force between current carrying conductors is given by the Lorentz Law:

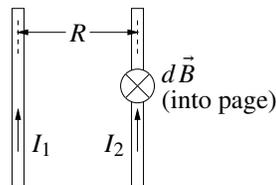
$$d\vec{F} = I d\vec{l} \times \vec{B}, \quad (\text{A.5})$$

where, $d\vec{F}$ = differential force, in Newtons.

One simple geometry for calculating the magnetic force per unit length is two infinitely long, straight conductors parallel to each other.

$$\frac{F}{l} = -\frac{\mu I_1 I_2}{2\pi R}.$$

The negative sign means the two parallel currents attract one another. If the direction of one of the currents were to be reversed (anti-parallel currents), the force would also reverse, and the currents would repel. Reversing *both* currents, of course, once again produces an attractive force.



To evaluate the magnetic field at off-axis points, the same formula (A.1) can be used. The mathematics quickly becomes intractable and the solution is usually implemented numerically. Alternatively, specialized computer aided design software can be used to calculate the magnetic field at arbitrary points in space. The two methods approach the calculation differently (the former is the idealized magnetic field numerically approximated while the other is the high fidelity numerical model of the detailed system). Either method will, of course, yield the same result.

The force due to a current-field interaction off-axis can be calculated according to (A.5). Thus, in the case of a Neodymium Iron Boron permanent magnet positioned above a coil, the magnet can be modeled as a current sheet of thickness equal to that of the magnet. This “current” creates a magnet field at the location of the coil conductors.

The force on the coil and, by Newton's third law, the force on the magnet is simply the product of the magnetic field, current and conductor path length.

Since the equivalent current of the permanent magnet is in the theta direction (circumferential around the magnet), the vector cross product in the force equation suggests that to get an axial force, F_z we must have a radial magnetic field, B_r ($-\hat{z} = \hat{\theta} \times \hat{r}$). In developing the control system, it is convenient to express the radial magnetic field equation in the following form:

$$B_r = \frac{c_1}{(z + c_2)^N},$$

where c_1 and c_2 are constants depending upon the geometry of the permanent magnets and N is a parameter describing the decrease in magnetic field with increasing axial distance. For the case of the axial magnetic field, we saw above c_1 is the surface current density multiplied by the magnet thickness, c_2 is related to the square of the magnet radius, and N is three or less, depending upon the relative axial distance. The radial magnetic field decreases more rapidly with distance than the axial magnetic field by approximately one more power of the denominator: $3 < N < 4$.¹

In order to suit the control law, the following form of the force equation for a magnet-coil interaction is sought:

$$F = \frac{K}{(z + D)^N} \cdot \frac{I}{I_{\max}}.$$

The various magnet-coil interactions have been analyzed and the appropriate constants have been determined. Because of the inherent non-linearity of magnetic fields, these constants can vary when the excursion from the calculated system is large; that is, the equations have been approximated by constant parameters in the region of interest. That is, at very large axial heights or very low heights the constants will differ from those calculated.

The form of the force is the same for interactions between permanent magnets due to the equivalent current concept discussed above. In this case, however, the current is not a free parameter for control but is determined by the geometry.

Calculating Induced Currents

A time-varying magnetic field induces voltages in a closed loop according to Faraday's Law:

$$V = -N_t \frac{\partial \phi}{\partial t},$$

where,

V is the induced voltage around a closed loop,

N_t is the number of turns of conductor around the loop, and,

$\partial \phi / \partial t$ is the flux rate of change through that loop.

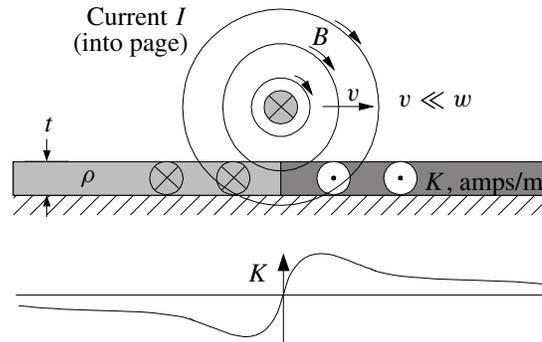
The negative sign in Faraday's Law is the manifestation of Lenz's Law. Lenz's Law states that when currents are induced in bodies due to a changing magnetic field, the currents are in such a direction as to cancel the change in magnetic field experienced by the body. For instance, if no field is present and suddenly a field is applied, the induced currents tend to circulate to cancel the magnetic field. If, however, the magnetic field has previously existed, removal of the magnetic field causes currents to flow in an attempt to maintain the field.

Consider a single current-carrying conductor moving relative to a conductive sheet. It can be shown [Magneto-Solid Mechanics, pg. 339 ff] that there is a characteristic velocity of the motion, w :

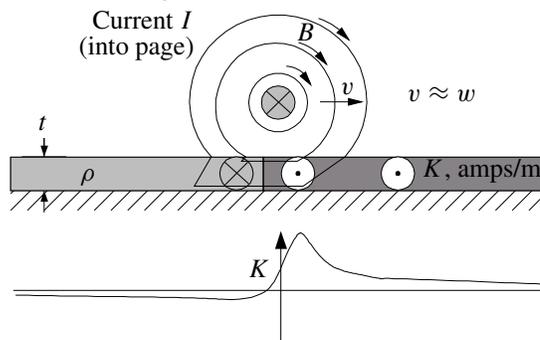
$$w = \frac{2\rho}{\mu_0 t},$$

¹In evaluating specific geometries and using numerical curve fitting, exponents of value $N > 4.0$ can result. For the Model 730 Apparatus, $N \leq 4.3$.

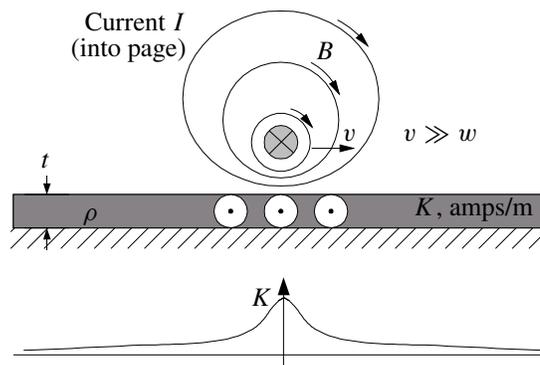
where, ρ is the sheet material resistivity, t is the thickness of the sheet and μ_0 is the permeability of free space. At standstill, the magnetic field of the current will fully permeate the sheet conductor. The magnetic field lines will be perfect circles about the current center. At very low speeds, ($v \ll w$) the field still permeates the sheet and the field lines will still be very nearly circular. This situation is shown in the figure below, the induced current in the sheet is K amps/meter.



As the conductor speed is increased to approximately the characteristic velocity ($v \approx w$), the movement of the magnetic field through the sheet causes induced currents to flow. According to Lenz's Law, these currents flow in such a manner so as to cancel the effect of the approaching field. Nevertheless, due to finite resistance, the magnetic field will still penetrate the conductive sheet to an extent and as the conductor leaves the region of magnetic field additional currents are induced to maintain the presence of the field. This situation is shown in the figure below. Notice the shear effect of the motion on the magnetic field.



When the speed is increased to substantially above the characteristics velocity, the conductivity of the sheet prevents the magnetic field from any significant penetration. The conductor is moving sufficiently fast that significant resistive dissipation does not occur. Each section of sheet generates the exact required current to perfectly shield the interior of the conductive sheet from the magnetic field. This situation is shown in the figure below. Notice that the magnetic field lines do not enter the sheet.



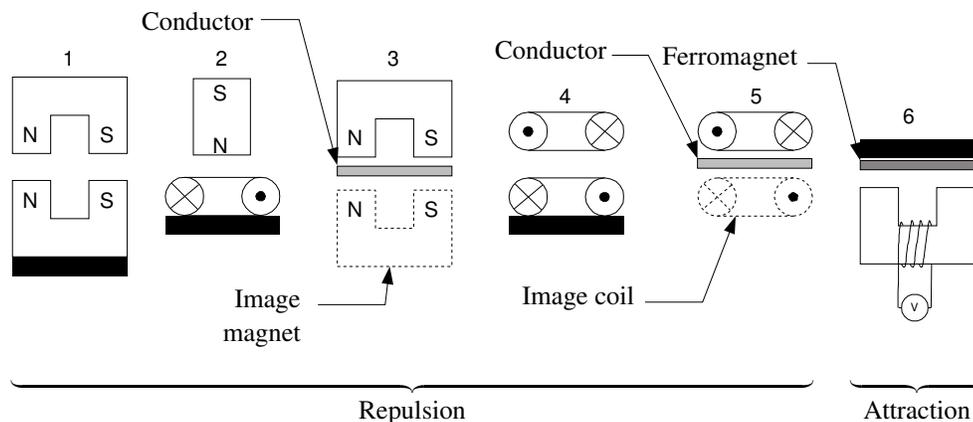
For application to the demonstration unit, the current element is the equivalent current of the permanent magnet edge. The permanent magnet flying above a rotating conductive sheet introduces several details different from that

described above. The physical principle remains the same; that is, the currents are induced in the sheet in response to the apparent change in magnetic field, and these currents tend to cancel the change in magnetic field. The interaction of the (magnetic field from) the induced currents and the original permanent magnetic field produce a repulsion force which levitates the magnet. These fundamentals are the basis of several different configurations of Maglev systems.

A.3 Maglev Applications

The figure below shows six arrangements used in magnetic levitation of moving vehicles. Five of the arrangements rely on repulsive forces. The lower elements are fixed (say, with respect to the earth) and the upper elements levitate. The first arrangement (permanent magnet like poles) is the common one for demonstrating like magnetic poles repel. The second and fourth arrangements (permanent magnet and/or superconducting magnet flying over a normal copper lower coil) is similar to that used for the Japanese superconducting Maglev system. The third and fifth systems are similar to the Magneplane system where permanent magnets or superconducting coils fly over normal sheet conductor. Notably, this has been variously proposed as an inexpensive method to attain levitation.

The sixth arrangement (electromagnet under a ferromagnet) is quite different and is the basis for the German Transrapid Maglev system. The fixed element is the upper ferromagnetic material and the lower electromagnet is actively controlled. If the current in the electromagnet is too large, the electromagnet feels a net upward force until it contacts the ferromagnetic material. If the current is too small, then insufficient force is available and the electromagnet falls. Hence, the current in the electromagnet must be continuously adjusted to enable levitation without contact.



For applications involving moving vehicles, all maglev designs share a common trait: while generating magnetic lift (in the direction perpendicular to travel) there is also generated magnetic drag (opposing the direction of travel). The details of the lift and drag forces, of course, depend upon the configuration.

A.4 Reference

News Item: 14 April 1999 Japanese Maglev vehicle demonstrates top speed of 552 km/hr (345 mph).

A.4.1 Current Status of Maglev in the United States and Internationally

Superconducting Maglev technology was initiated in the late 1960s and early 1970s in the United States in 1969 when Drs. James Powell and Gordon Danby of New York's Brookhaven National Laboratory invented the concept of a repulsive magnetic suspension using superconducting magnets. In the mid-1970s the US stopped Maglev development due to funding problems. Other countries, however, continued to develop Maglev and today have viable systems. In the early 1990s Maglev research was rekindled at a Federal government level. At various times, the Department of Transportation's Federal Railroad Administration and Federal Transit Administration, NASA, Department of the Air Force, and Department of the Navy have joined resources for the purpose of developing

Maglev and Linear Motor technology. Although each agency had its own specific application in mind, a loose consortium seemed to provide the best bang for the buck. The situation has progressed where today there are several high speed (≈ 300 mph) and low speed (≈ 100 mph) regions in this country where Maglev is thought to be a viable alternative means of rapid public transportation—yet it is still unproven in the United States.

Germany's Transrapid vehicle has been extensively tested and has been proposed for use on several projects in this country. The Germans are presently constructing a Transrapid route from Hamburg to Berlin.

Japan is developing a system that uses superconducting magnets and is currently constructing a major test route that will ultimately be incorporated into a revenue-producing route. Approximately 80% of this system will be in tunnels cut into mountains. This has greatly increased the construction cost but has decreased the cost of land acquisition for the Maglev right-of-way.

A.4.2 Glossary

Eddy Currents: Induced currents in conductors by changing magnetic fields. Since the currents flow in closed paths within the materials, they are similar to eddies in rivers. In accordance with Lenz's Law, these currents flow in such a manner as to oppose the change in magnetic field inducing them. Frequently, eddy currents are undesirable, but they can be desired in cases of induction and microwave heating for industrial and consumer purposes.

EDS: Electrodynamic System

Electrodynamic System: Magnetic forces based upon repulsion from induced currents. Inherently stable can be lightly damped or even negatively damped at sufficiently high frequencies.

Electromagnetic System: Magnetic forces based upon attraction from applied currents. Inherently unstable and currents must be controlled.

EMS: Electromagnetic System

HSST: High Speed Surface Transportation (not as high speed as the name sounds). Japanese EMS Maglev.

Inductance, L : $L = N_t \phi / I$, Total flux linked per unit current. Units are Henries, H.

Maglev: Generic term for magnetic suspension. Sometimes refers to both levitation (vertical forces) and guidance (side-to-side forces).

Magnetic Drag: Magnetic force opposing propulsion, due to resistive dissipation. Units are Newtons.

Magnetic Field, \mathbf{H} : Three-dimensional vector used to calculate the enclosed current as used in Ampere's Law $\nabla \times \mathbf{H} = \mu_0 \mathbf{J}_{free}$. This parameter is independent of the material properties. Units are Amperes/meter, A/m.

Magnetic Flux Density, \mathbf{B} : Three-dimensional vector used to calculate the force on conductors and magnetic materials. Units are Wb/m² or Tesla, T. $\nabla \times \mathbf{B} = \mu_0 (\mathbf{J}_{free} + \mathbf{J}_{bound})$. Recall also $\mathbf{F} = I d\mathbf{l} \times \mathbf{B}$.

Magnetic Flux, ϕ : $\phi = \iint \vec{B} \cdot d\vec{A}$, the integral of the vector dot product of the flux density and area. Units are Webers, Wb.

Magnetic Lift: Magnetic force opposing gravity.

Permeability: $\mu = \mathbf{B}/\mathbf{H}$, Ratio of flux density to magnetic field. Units are Henries/meter, H/m.

Reluctance: Ratio of total magnetomotive force to total flux through a circuit. Units are inverse Henries, H⁻¹.

RTRI: Railway Technical Research Institute. Japanese research arm to develop EDS Maglev.

Superconductivity: The material property of the complete lack of electric resistance, obtained for special materials only under conditions of refrigeration. Two classes of materials are low critical temperature superconductors ($T \leq 10K$) and high critical temperature superconductors (HTSC) ($T \leq 77K$).

Transrapid: German EMS Maglev, Hamburg to Berlin route under construction.

Magnetomotive force, mmf: $MMF = \oint \vec{H} \cdot d\vec{l} = N_r I$. Intermediate quantity used for magnetic circuits. Conceptually similar to the electromotive force or voltage of an electric circuit. Units of mmf are Ampere-turns.

Lenz's Law: A law stating that induced currents resulting from a change in magnetic field are in such a direction as to oppose the change in magnetic field. Hence, if a magnetic field is increased, current is induced to cancel it. If the magnetic field is decreased, current is induced to preserve it.

A.4.3 Reference Literature:

Books

1. *Superconducting Levitation: Applications to Bearings and Magnetic Transportation*, Francis Moon, Wiley & Sons, New York, 1994.
2. *Case Studies in Superconducting Magnets: Design and Operational Issues*, Yukikazu Iwasa, Plenum Press, New York, 1994.
3. *Magneto-Solid Mechanics*, Francis Moon, Wiley & Sons, New York, 1984.
4. *Electromagnetic Levitation and Suspension Techniques*, B.V. Jayawant, Edward Arnold Publishers, London, 1981.
5. *Electromagnetics*, J.D. Kraus, McGraw Hill Companies, New York, 1992, p. 288.
6. *Electricity & Magnetism*, E. Purcell, McGraw-Hill Book Companies, New York, 1965.

Maglev Studies

1. "Study of Japanese Electrodynamic Suspension Maglev Systems," J.L. He, D.M. Rote, H.T. Coffey, Argonne National Lab, Argonne, IL, 60439, April 1994.
2. "New York State Technical and Economic Maglev Evaluation," Michael Proise, Grumman Space and Electronics Division, New York, June 1991.
3. "Electrodynamic Suspension and Linear Synchronous Motor Propulsion for High Speed Guided Ground Transportation," David Atherton, Canadian Maglev Group, CIGGT Report No. 77-13, September 1977.
4. "Conceptual Design and Analysis of The Tracked Magnetically Levitated Vehicle Technology Program Repulsion Scheme, Volume I—Technical Studies, Ford Motor Co., US DOT/FRA, Report PB247931, Feb. 1975.

Maglev Web Sites

1. <http://bmes.ece.utexas.edu/~jcamp/physics/index.html> Descriptive link of Maglev technologies and basic physics.
2. <http://eb-p5.eb.uah.edu/maglev/maglev.html> Univ. of Alabama Huntsville College of Engineering WWW Archive and Information service for High Speed and Automated Transportation Technologies. (This site is relatively new and has unfinished links.)
3. <http://www.rtri.or.jp/> RTRI Home page. Japanese EDS Maglev.
4. <http://www.mvp.de/english.htm> Transrapid Official Homepage. German EMS Maglev.

5. <http://weber.u.washington.edu/~jbs/itrans/hsst.htm> **HSST** unofficial descriptive page.
6. <http://www.meitetsu.co.jp/chsst/mecha.html> **Mechanism of HSST**. Japanese EMS.
7. <http://popularmechanics.com/popmech/sci/9805STTRP.html> “Track to the Future” article. Descriptive article of one recent Maglev approach.
8. <http://www.llnl.gov/str/Post.html> “A New Approach to Levitating Trains—and Rockets.” Elaboration of the previous link for a Maglev approach.

App. B: The Control-Systems Laboratory

B.1 THE LABORATORY¹

The Control-Systems Laboratory comprises four student workstations, each having a computer and one or more devices to control.

B.1.1 The Lab Computers

The lab computers are Pentium-III 700 MHz machines which include special hardware for data I/O. The computers run the Real-Time Linux operating system. No previous student knowledge of Linux is expected.

When the computers are turned on, a log-in screen will be displayed. Enter the class username and password:

Username: ece4560

Password: ece4560

(real secure, huh?) It is *NOT* wise to save information permanently on the lab computers. When you log on, the screen shown in Fig. B.1 on page B-2 will appear.

You will use Matlab extensively. You can start Matlab by clicking on the Matlab icon to the left of the screen. Once started, Matlab will run as it does on an NT machine. In particular, “print” will print; “edit” will edit a file, and so forth.

You can easily “link” your ECEDOMAIN NT account to the lab computer. Click on the “NT network” icon to the left of the desktop. Enter your NT username and password and click on “OK”. This will load your NT account. You can read and write files to your account just as you would write to this directory. Note that the mechanism to link your NT account to the lab computer is not particularly secure. Use at your own risk.

To log out, click on the “X” in the menubar at the bottom of the desktop, or select “Logout” from the “K” menu in the same menubar.

B.1.2 The Devices to Control

The lab experiments will focus on one physical device to control—the the Magnetic Levitation (MagLev) device. The lab also has Control Moment Gyroscope devices which will be used in other control courses. These two devices are depicted in Fig. B.2.

Some cautions are in order for the safety of the equipment:

1. **Do not touch the glass rod on the MagLev system.** Oils break down the lubricant in the magnet bushings, increase friction and decrease the lifetime of the equipment.
2. **Do not touch the white surface of the magnets.** The surface must be clean for proper functioning of the laser sensor.

¹The Control-Systems Laboratory has been made possible by a grant from the National Science Foundation directorate of undergraduate education.

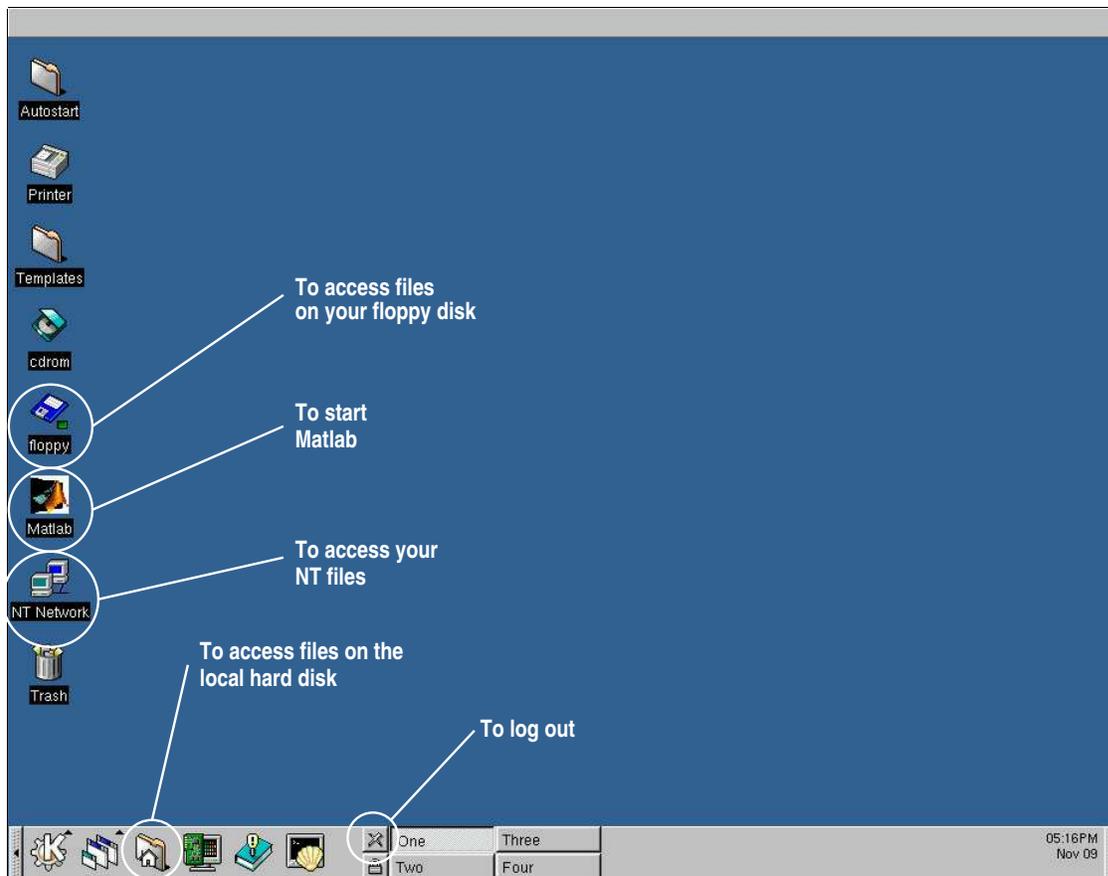


Figure B.1 Desktop screen.



Figure B.2 The two lab devices. The MagLev device is to the left; the Gyro device is to the right.

3. **Do not get magnetic (i.e., computer) media near the MagLev magnets.** The magnets are extremely strong and *will* erase computer media. You have been warned.
4. **Do not attempt to move the Gyro axes manually.** These devices have “brakes” which are activated when the device is off and when switches on the control box are turned on. These brakes will quickly wear out if forced.

B.2 Matlab, Simulink, RTW, RTLT

B.2.1 Simulink

To start Simulink, you start Matlab, and then enter “`simulink`” at the Matlab “> >” prompt.

```
>> simulink
```

Simulink operates the same way on the lab machines as it does on an NT machine.

B.2.2 The Real Time Workshop

The *Real Time Workshop (RTW)* is a part of Matlab which interfaces Simulink with the “real world.” Simulink by itself allows simulation of dynamic systems. With the Real Time Workshop, however, you can include physical devices inside the “simulation”. Signals from your Simulink block diagram may be output through digital to analog (D2A) converters, and external signals may be read into your Simulink block diagram through analog to digital (A2D) and optical encoder inputs.

The Real Time Workshop can operate with many different input-output (I/O) platforms, some having their own DSP on board to run the resulting code. The setup in the lab uses the *ServoToGo Model 2 I/O* board, which includes eight A2Ds, eight D2As, and eight quadrature optical encoder inputs (and a variety of digital I/O and other features we will not use). Many of these features are connected to the breakout box next to the computer so that you can simply wire up experiments using banana plugs.

The ServoToGo I/O board does not have its own DSP, so the real-time software must run on the lab computer’s own Pentium CPU. The Real Time Workshop, in conjunction with a piece of software called the *Real Time Linux Target (RTLT)* (by Quality Real Time Systems), automatically generates C code to implement your Simulink diagram, compiles the C code into executable code, and runs the code on the Pentium CPU in the computer. The *Real Time Linux* operating system is used because it guarantees performance required for real-time processing of control algorithms and data I/O.

The beauty of this arrangement, used frequently in industry to test and prototype control systems, is that you can test controllers in a “hardware in the loop” configuration without ever coding an interrupt service routine or writing a single line of C or assembler code! (much less, debugging same) Designs may be prototyped rapidly resulting in great economies of time and money. You will see that there is a learning curve the first time you use this particular scheme, but afterward it becomes completely natural.

B.2.3 Using the Real Time Workshop

In order to use the Real Time Workshop, you must know how to construct a Simulink block diagram containing RTW elements, be able to set all parameters for these elements and for Simulink/ RTW, and be able to properly compile/build the code and run it. These topics are discussed in the following sections.

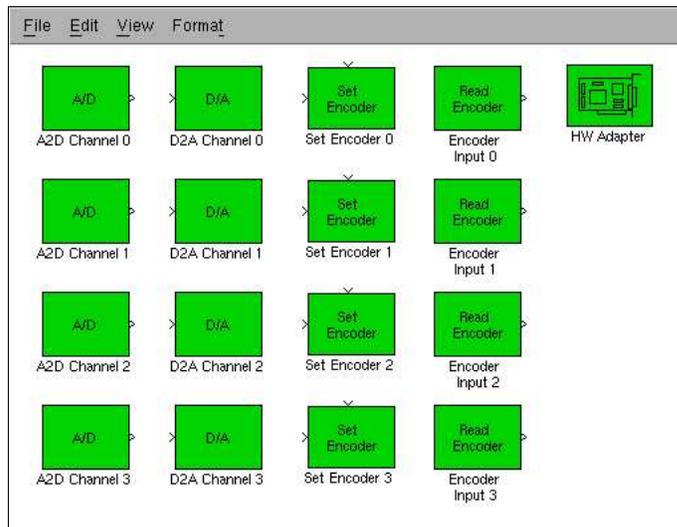


Figure B.3 The rtlib library.

Creating the Real Time Model

In order to start using the Real Time Workshop, in the Matlab window type

```
>> rtlib
```

“rtlib” (Fig. B.3) is a Simulink library file containing blocks useful for I/O.

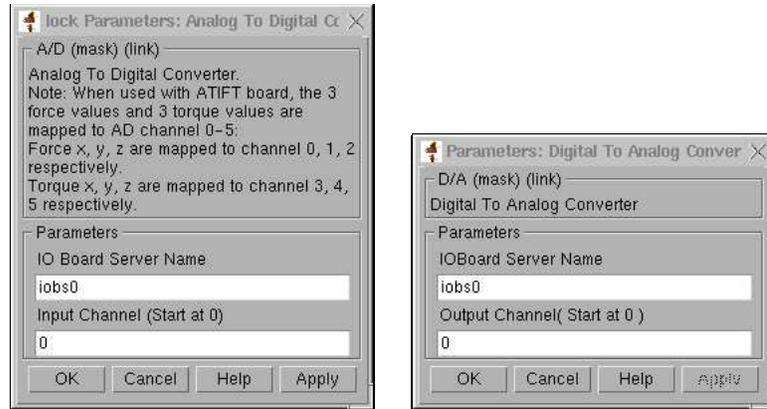
You can drag a desired block into a Simulink model you are working on, and then make the necessary connections.

In the `rtlib` library there are four A2D input blocks. These are actually identical except that their settings have been preset to correspond to the four A2D inputs on the MagLev side of the breakout box. The top set of banana plugs on the breakout box is A2D1 (channel 0) and so forth through to the bottom set which is A2D4 (channel 3). A2D1 is generally used as the input from the sensor on the bottom coil; A2D2 is generally used as the input from the sensor on the top coil. The other two inputs are the temperature readings of the two coils, and may be used to provide more accurate sensor calibration when the sensor heats up.

There are four D2A output blocks. Again, these are identical except that their settings have been preset to correspond to the four D2A outputs on the breakout box. D2A1 (channel 0) and D2A2 (channel 1) correspond to the MagLev drive coils (bottom and top, respectively). D2A3 (channel 2) and D2A4 (channel 3) correspond to the Gyro D2A1 and D2A2 on the breakout box, and are used to drive Axis 1 and Axis 2 of the Gyro.

The final block in the window is the hardware adapter block. **The HW Adapter block must be part of your Simulink diagram, and is not wired to anything else.** Its presence alone is enough to let Simulink/ RTW/ RTLT know which board is being used. By double clicking on the Hardware Adapter block, you can access its properties; the only one which you might consider changing is the sampling frequency. A value of 2 kHz should be sufficient for all lab purposes.

The A2D Converter Blocks: The parameters of the A2D converters may also be changed by double clicking on the block in your Simulink diagram. The window in Fig. B.4(a) will be displayed. If you selected the correct block from the `rtlib` menu, you should never need to change a parameter in this window. If you pulled the wrong block off for the desired channel, you may change the input channel number in this window. For convenience, the associations between channel number and breakout-box labeling are repeated here:



(a) A2D parameters.

(b) D2A parameters.

Figure B.4 I/O Box Parameters.

Channel	Breakout Box	Typical Use
A2D Channel 0	MagLev A2D1	Bottom Coil Sensor
A2D Channel 1	MagLev A2D2	Top Coil Sensor
A2D Channel 2	MagLev A2D3	Bottom Temp Sensor
A2D Channel 3	MagLev A2D4	Top Temp Sensor

The D2A Converter Blocks: The parameters of the D2A converters may be changed by double clicking on the block in your Simulink diagram. The window in Fig. B.4(b) will be displayed. If you selected the correct block from the `rtlib` menu, you should never need to change a parameter in this window. If you pulled the wrong block off for the desired channel, you may change the input channel number in this window. For convenience, the associations between channel number and breakout-box labeling are repeated here:

Channel	Breakout Box	Typical Use
D2A Channel 0	MagLev D2A1	Bottom Coil Drive
D2A Channel 1	MagLev D2A2	Top Coil Drive
D2A Channel 2	Gyro D2A1	Axis 1 Motor Drive
D2A Channel 3	Gyro D2A2	Axis 2 Motor Drive

The Encoder Input Blocks: We will not require encoder input blocks for this lab course. The encoder inputs are used when controlling the gyroscope unit.

Setting the Real-Time Options

There are a number of options, buried deep in a series of menus, which must be set before you can properly build and run your code. Most of these options have already been set properly for you if you use the `template.mdl` file as instructed. There are still some parameters which you must set, and these are located in the “solver page”.

Solver Page: To access this page, you must select **Solver** in the **Simulation Parameters** property sheet (Fig. B.5 on page B-6). Before being able to use your model, you must set the fixed-step solver step size: you must click on the pop-up menu next to **Type** under **Solver options** group title and select “fixed-step”. An appropriate value for the

Fixed step size edit box is the sampling period. Thus, if the control algorithm is executed at 2 kHz , then you should set the **Fixed step size** edit box to “0.0005”.

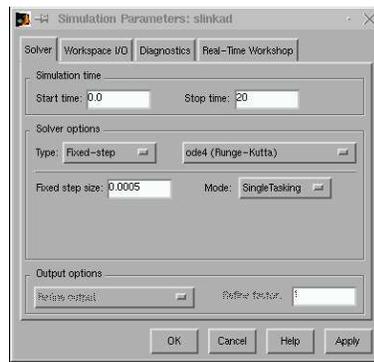


Figure B.5 The Solver property sheet.

Currently, the software does not support multi-tasking; hence, the **Mode** pop-up menu must be set to “Single Tasking”. You can also set the **Start time** and **Stop time** by accessing the appropriate edit boxes under the **Simulation time** group title. By default these values are set to “0.0” and “10.0”, respectively.

There are some correspondences between values in this solver page and other values entered elsewhere which must be observed. If they are not (at best) error messages or (at worst) chaos will ensue:

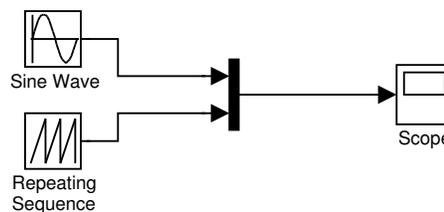
Correspondences	
Solver: Fixed step size	⇔ HW Adapter: Sampling rate
Solver: Stop time	⇔ Scope Properties: Time range

Building the Real-Time Model

From the prior discussion, you should be able to construct a Simulink block diagram including physical I/O and be able to set all necessary parameters. It remains to see how to run the resulting simulation and view the output (log data). Here we discuss data logging.

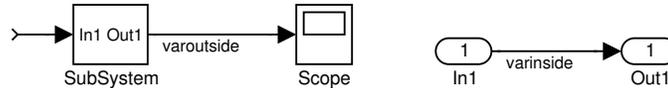
When using RTW/RTL, data logging is accomplished *only* via the scope block. The standard Simulink X-Y Graph, Display, To File and To Workspace blocks are not supported. **In order to display or log any simulation result, a scope block must be used.**

Several signals may be displayed on the same scope graph using the Mux block.



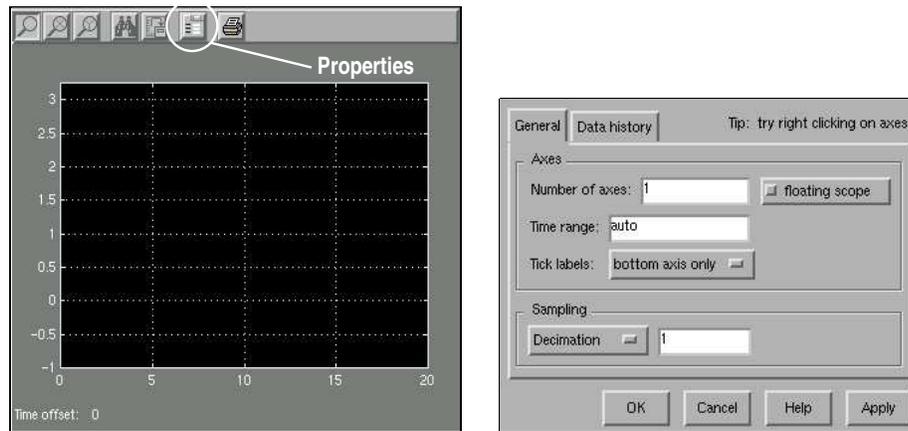
For the purpose of logging data to a file, signals may be given names. You can do this by double clicking on the line that connects the variable to the scope. Enter a valid variable name (no spaces are allowed). If you do not specify a

name, the default Simulink variable name (e.g., “root_Gain1”) will be used. A very subtle point is that output signals of subsystems must be labeled *inside* the subsystem in order for the variable name to stick.



In the above figure, the block diagram on the right constitutes the subsystem in the block diagram on the left. Contrary to intuition, the variable name of the logged data is “varinside” not “varoutside”.

Configuring the Scope: In order to configure the Scope, you must double click on the Scope block in the Simulink model (Fig. B.6(a)). The Properties button in the Scope toolbar is used to access the scope properties. In the **General** page (Fig. B.6(b)) the **Number of axes** refers to the number of subplots in the scope; the **Time Range** specifies how many seconds of real time are displayed on the scope; the **Decimation** box may be used to plot every n data samples for faster plotting/ printing.



(a) Scope window

(b) General scope parameters

Figure B.6 Scope parameter pages.

Options for Viewing Data: To set the data logging options, you must access the **External Mode Control Panel** from your model’s **Tools** menu. You can then access the **External Signal & Triggering** property sheet (Fig. B.7) by clicking on the **Signal & Triggering** button. Under the **Signal selection** group title, you can click on the **Select all** button to select all the scopes, or highlight the scope that you want to access and then enable the **on** check box. **If no data is logged from your simulation run, you have most likely forgotten to perform this step!** A scope is selected if an “X” appears in the leftmost column under the **Signal selection** group title. The **Source** pop-up menu under the Trigger group title should be set to “manual”. The **Mode** pop-up menu should be set to “one-shot”. The **Delay** edit box is usually set to 0.

The **Duration** edit box specifies the number of time intervals for which the external mode logs data after logging starts. The value of the **Duration** edit box is equal to $(\text{stop time} - \text{start time}) / (\text{Fixed step size})$. For example, if you execute a real-time model at 2 kHz for 60 seconds and desire to log data for the entire 60 seconds, then **Duration** is set to 120000.

You should enable the **Arm when connect to target** check box.

Saving Simulation Data to Disk: Simulation data may be saved automatically to a file in a “.m” file format. This is accomplished via the data archiving option of the scope block(s). To set the data archiving options, you must access the **Tools** menu **External Mode Control Panel**. Click on the **Data archiving** button to open the **Data Archiving** property sheet (Fig. B.8 on page B-8). Highlight the **Enable archiving** check box to activate the automated data archiving feature of the external mode. If the **Enable archiving** check box is disabled, no log file

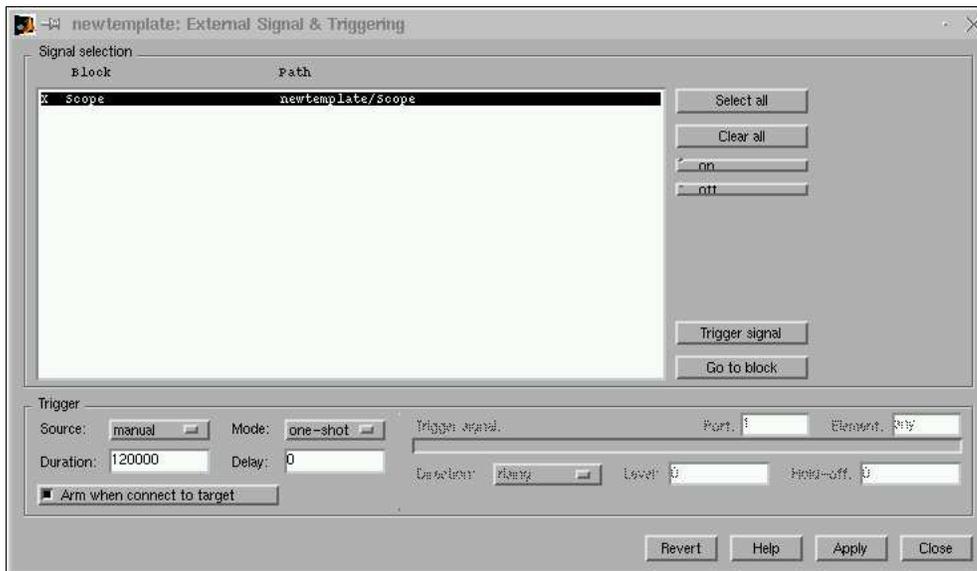


Figure B.7 External Signal and Triggering property sheet.

will be created. If the check box is enabled, you must provide a valid **Directory** name and a **File** name for archiving purposes. If the **Increment file after one-shot** check box is enabled, the filename will be automatically incremented after each run. (You must re-build your model after changing any information in this property sheet for the changes to take effect.)

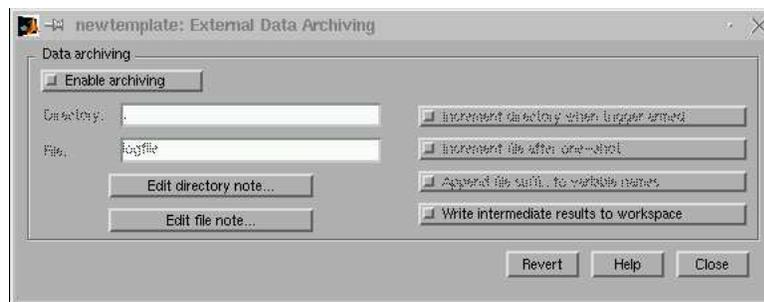


Figure B.8 Data Archiving property sheet.

To load the archived data after a simulation run, if your log file was called “logfile.m” then in Matlab type

```
>> clear; logfile
```

and your data will be loaded under the variable names assigned to the signals. The “clear” command ensures that previous Matlab variables are cleared before you load the log file.

Building and Executing the Real-Time Model

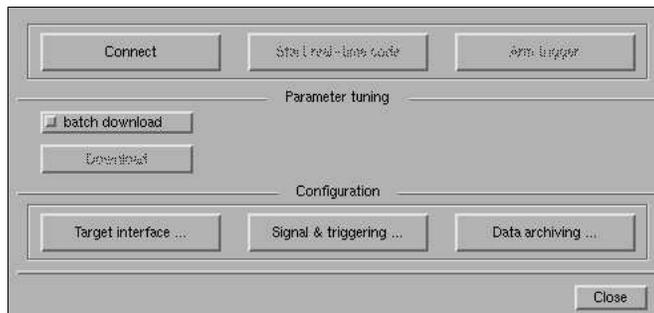
Building the Model: In order to run your Simulink model in real time, you must first *build* the code. Select **RTW Build** from the **Tools** menu. RTW/RTLTL will then perform the following steps:

1. RTLTL creates <model_name>.c and <model_name>.h source files, and the configuration file <model_name>_cfg.cfg.

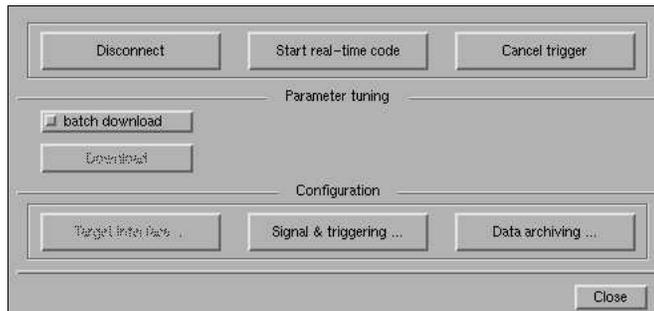
2. The build command for real-time applications (make-rtw) is issued which creates <model_name>.mk makefile from the template file (lrt.tmf).
3. The make utility (nmake) builds the <model_name>_rtl.o real time target using the makefile created in step 2.

If there are any errors in the Simulink model or an RTLT setting, a dialog box will appear with warnings and errors. If there are no errors, then you are ready to execute the real-time target. **Note that every time you change your model you will need to re-build its code** (If you just change a parameter value of a block, this *may* not be necessary. It is always safest to do so, however).

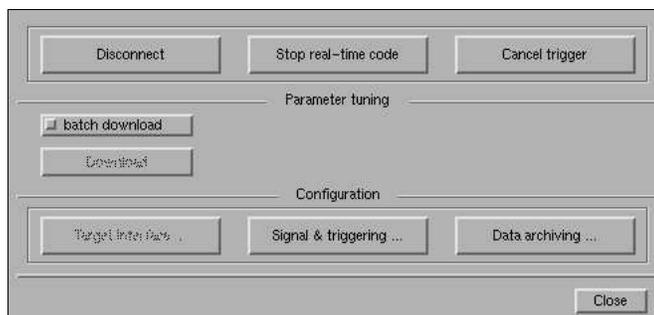
Executing the Model: To execute the real-time target, you connect to it by accessing the **Tools** menu from the Simulink model menubar and then selecting **External Mode Control Panel** from the pull-down menu. In the **External Mode Control Panel**, you click on the **Connect** button.



The window changes (button labels change):



To begin execution, you then click on the **Start real-time code** button. The window changes once again:



To stop execution of the real-time target, you can click either the **Stop real-time code** button or the **Disconnect** button.

B.3 Nonlinear Dynamic Model of MagLev System

B.3.1 Description of the system

Two views of the magnetic levitation system are depicted in Fig. B.9. The plant consists of upper and lower drive coils that produce a magnetic field in response to a dc current. One or two magnets travel along a precision ground Pyrex glass guide rod. By energizing the lower coil, a single magnet is levitated by a repulsive magnetic force. As current in the coil increases, the field strength increases and the levitated magnet height is increased. For the upper coil, the levitating force is attractive. Two magnets may be controlled simultaneously by stacking them on the glass rod. The magnets are of an ultra-high field strength rare earth (NeBFe) type and are designed to provide large levitated displacements to clearly demonstrate principles of levitation and motion control.

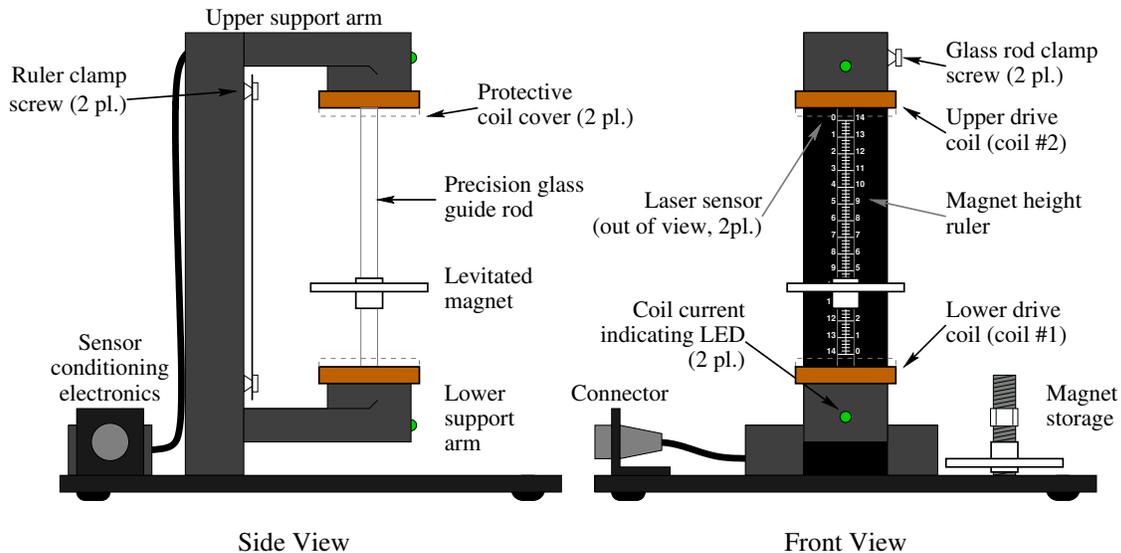


Figure B.9 Two views of the magnetic levitation system.

Two laser-based sensors measure the magnet positions. The lower sensor is typically used to measure a given magnet's position in proximity ($\approx 8\text{ cm}$ range) to the lower coil, and the upper one for proximity to the upper coil. The sensor design utilizes light amplitude measurement and includes special circuitry to desensitize the signal to stray ambient light and thermal fluctuations.

For many control scenarios, the system will be configured as shown in Fig. B.10 on page B-11. A controller is connected to the MagLev through a power amplifier. The amplifier has input voltage range -10 to 10 volts, and outputs current in the appropriate range to drive the electromagnetic coils. The amplifier also buffers the sensor signal from the plant. When a computer is used as controller, it is connected to the amplifier box via a "breakout box" which provides the appropriate input and output signals via banana plugs.

One or more magnetic disks may be placed on the Pyrex rod and levitated. Fig. B.11 on page B-11 shows a diagram of the disk. A dry-lubricated guide bushing at the center of the disk slides up and down the rod. A white reflective surface covers most of the disk. A "N" or "S" label identifies the polarity of a particular side of the disk.

The sensor makes use of the reflective properties of the disk surface. The laser sensor is depicted functionally in Fig. B.12 on page B-11. The monochromatic, coherent laser beam is spread via a single hemicylindrical optical element into a fan shape. This beam is projected on the diffuse white surface of the magnet. A photodetector views the beam and generates a voltage proportional to the amount of beam power incident on it. The white surface properties and laser/detector view factors and geometries are designed to maximize the sensor gain slope (change in output versus change in position) through the sensor operational range of about 8 cm .

Because of the fan beam shape and the fact that the reflected light is a combination of specular and diffuse, the power on the detector diminishes with the inverse and inverse square of the magnet distance. Additionally, the geometry

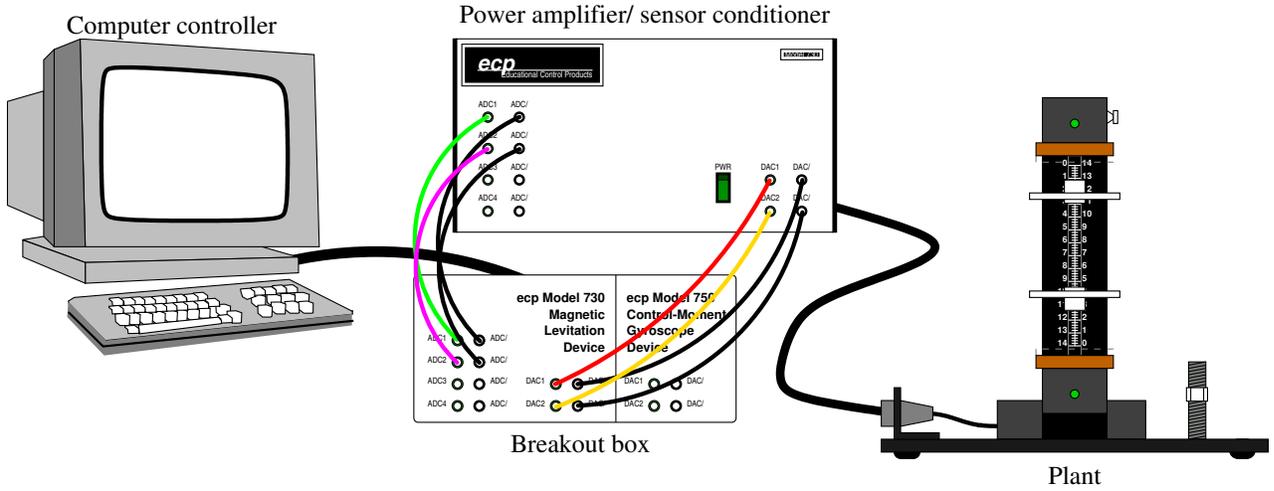


Figure B.10 System setup for control.

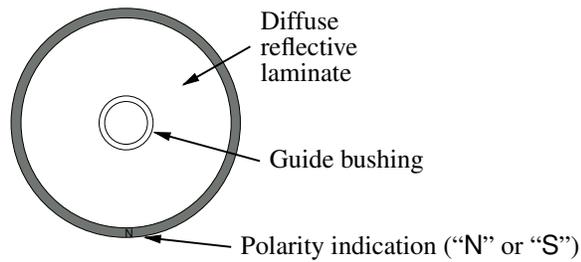


Figure B.11 Magnetic disk.

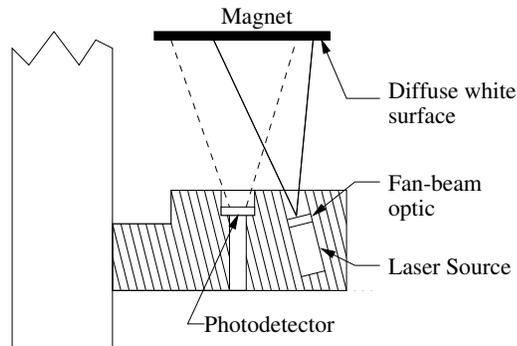


Figure B.12 Functional diagram of laser sensor.

of the laser, detector, and magnet and the resulting cutoff of the detector's view of the beam as a function of magnet height give rise to a linear dependency at small magnet distance. These relationships are used to motivate the general form of the sensor linearization function as discussed later.

Electronic circuitry makes the design immune to stray light noise, such as turning room lights on and off, and rejects most induced electronic disturbances. Thus a relatively low noise signal is output from the amplifier box.²

B.3.2 Making a Mathematical Model

We wish to determine a mathematical model of the MagLev system. To do so, free-body diagrams for the MagLev apparatus in two configurations are shown in Fig. B.13. We will consider the double-magnet configuration. Each magnet is acted on by forces from both drive coils, from the other magnet, from gravity, and from friction (modeled as viscous). In the diagram, y_1 is the position of the lower magnet, y_2 is the position of the upper magnet, y_c is the maximum magnet separation, i_1 is the current into the lower coil and i_2 is the current into the upper coil. All other labels refer to forces operating on the magnets. From the figure, we have an equation of motion for the first magnet

$$\begin{aligned} m\ddot{y}_1 &= \sum \text{Forces on disk 1} \\ m\ddot{y}_1 &= F_{u11} + F_{u21} - F_{m12} - c_1\dot{y}_1 - mg. \end{aligned}$$

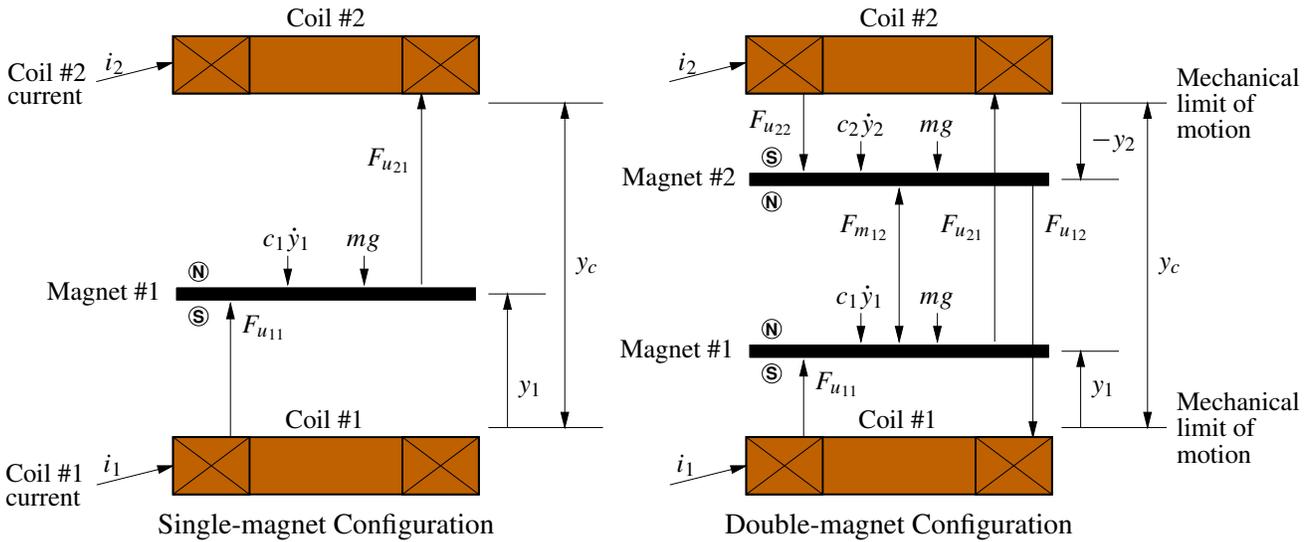


Figure B.13 Force diagrams for the MagLev device in two configurations.

Similarly for the second magnet

$$\begin{aligned} m\ddot{y}_2 &= \sum \text{Forces on disk 2} \\ m\ddot{y}_2 &= F_{m12} - F_{u22} - F_{u12} - c_2\dot{y}_2 - mg. \end{aligned}$$

The magnetic force terms are modeled as having the following forms (see Appendix A for details)

$$\begin{aligned} F_{u11} &= \frac{i_1}{a(y_1 + b)^N} & F_{u12} &= \frac{i_1}{a(y_c + y_2 + b)^N} & F_{m12} &= \frac{c}{(y_{12} + d)^N} \\ F_{u22} &= \frac{i_2}{a(-y_2 + b)^N} & F_{u21} &= \frac{i_2}{a(y_c - y_1 + b)^N} \end{aligned}$$

²A laser temperature signal is also output. This may be used to compensate for the laser's inherent reduction in emitted power as a function of temperature (approx. 15% maximum for operational temperatures of the MagLev device).

where the actual magnet separation y_{12} is

$$y_{12} = y_c + y_2 - y_1, \quad y_c \approx 13.0 \text{ cm}$$

and a , b , c , d , and N are constants which may be determined by numerical modeling of the magnetic configuration or by empirical methods. Typically $3 < N < 4.5$. The maximum magnet separation may be found by measurement to be $y_c \approx 13.0 \text{ cm}$.

B.3.3 Equations of Motion for the Double-magnet Configuration

For the MagLev apparatus, the coefficients of friction affecting both disks are the same, and may be written as $c_1 = c_2 = m\beta$. In addition, a value of $N = 4$ has been shown empirically to yield a close approximation of the force/distance relationships. Finally, since current in a coil is proportional to the voltage applied to the power amplifier for that coil, we find that the following equations are useful for design purposes:

$$m\ddot{y}_1 = F_{u_{11}} + F_{u_{21}} - F_{m_{12}} - m\beta\dot{y}_1 - mg \quad (\text{B.1})$$

$$m\ddot{y}_2 = F_{m_{12}} - F_{u_{22}} - F_{u_{12}} - m\beta\dot{y}_2 - mg \quad (\text{B.2})$$

$$F_{u_{11}} = \frac{u_1}{a(y_1 + b)^4} \quad (\text{B.3})$$

$$F_{u_{12}} = \frac{u_1}{a(13 + y_2 + b)^4} \quad (\text{B.4})$$

$$F_{u_{22}} = \frac{u_2}{a(-y_2 + b)^4} \quad (\text{B.5})$$

$$F_{u_{21}} = \frac{u_2}{a(13 - y_1 + b)^4} \quad (\text{B.6})$$

$$F_{m_{12}} = \frac{c}{(y_{12} + d)^4}. \quad (\text{B.7})$$

B.3.4 Nonlinear Sensor

As mentioned previously, the sensors on the MagLev measure the magnet position indirectly, through a nonlinear function. The raw sensor voltage read $y_{k_{raw}}$ is related to the true magnet position y_k through the approximation:

$$y_k \approx \frac{e_k}{y_{k_{raw}}} + \frac{f_k}{\sqrt{y_{k_{raw}}}} + g_k + h_k \cdot y_{k_{raw}}. \quad (\text{B.8})$$

The constants $\{e_1, f_1, g_1, h_1\}$ and $\{e_2, f_2, g_2, h_2\}$ may be determined using empirical methods to model the bottom and top sensors, respectively. For the purposes of control design, we “calibrate” the sensor by calculating an estimate $y_{k_{cal}}$ of the magnet position

$$y_k \approx y_{k_{cal}} = \frac{e_k}{y_{k_{raw}}} + \frac{f_k}{\sqrt{y_{k_{raw}}}} + g_k + h_k \cdot y_{k_{raw}}.$$

B.4 Linearized Dynamic Model

The dynamics of Eqs. (B.1) to (B.7) are very nonlinear. Furthermore, the sensor measures a nonlinear function of the true magnet position. We need to *linearize* these equations around some desired operating point. That is, we make an approximate linear model of the system in the region of operation space we plan to use.

For the purpose of the present discussion we will consider the case where there is one magnet on the device. The equation of motion for that magnet is then

$$m\ddot{y}_1 = \frac{u_1}{a(y_1 + b)^4} + \frac{u_2}{a(13 - y_1 + b)^4} - m\beta\dot{y}_1 - mg. \quad (\text{B.9})$$

These dynamics are represented schematically as in Fig. B.14. In the diagram, y_1 is the true position of the magnet and $y_{1_{raw}}$ is the measured voltage corresponding to that position. $G(s) = 1/(ms^2 + m\beta s)$.

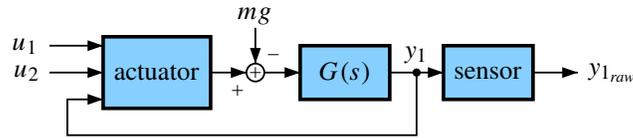


Figure B.14 Block diagram of MagLev dynamics, single-magnet configuration.

Even in Eq. (B.9) with one magnet on the device we see that the dynamics are nonlinear. (It is not a linear constant coefficient ordinary differential equation). There are three sources of nonlinearity: the actuator contributes a nonlinearity via Eq. (B.9), the sensor contributes another nonlinearity via (the inverse of) Eq. (B.8), and gravity contributes a nonlinearity via the constant added force mg (LCCODEs do not have constant terms).

B.4.1 Small-Signal Linearization \Rightarrow Model predicts height of magnet y_k .

The linearization method is shown in Fig. B.15. The constant force due to gravity is *canceled* by applying a constant steady-state voltage u_{1o} to the controller output.³ This alone has the effect of levitating the magnet at a constant height, in the absence of disturbance.

The voltage $y_{1_{raw}}$ corresponding to the magnet's height is measured. We compute a calibrated estimate $y_{1_{cal}}$ of the actual magnet position y_1 using the measured sensor voltage $y_{1_{raw}}$ via the relation

$$y_{1_{cal}} = \frac{e_1}{y_{1_{raw}}} + \frac{f_1}{\sqrt{y_{1_{raw}}}} + g_1 + h_1 \cdot y_{1_{raw}}. \quad (\text{B.10})$$

In the diagram, this operation is performed by the “sensor inverse” block. It effectively inverts or cancels the sensor nonlinearity.

The actuator nonlinearity is not explicitly canceled. Rather, the nonlinear portion of the block diagram inside the thick gray rectangle is “linearized” using methods described below. A linear approximation of the nonlinear function inside the gray box is used as the plant model.

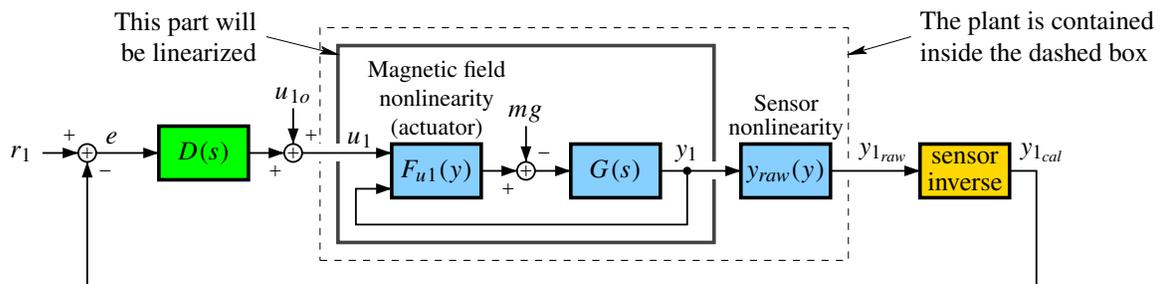


Figure B.15 Linearization: The linear model predicts the actual magnet height. Your controller must specifically invert the sensor nonlinearity.

Taylor-Series Function Expansion

Nonlinear equations may be linearized using Taylor-series expansion. You should recall from your calculus courses that a function may be expanded via a Taylor-series expansion to be represented as a polynomial. For example, the

³This value is a function of the desired steady-state output y_{1o} .

$\sin(x)$ function may be expanded around $x = 0$ and represented as

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \cdots$$

The expansion is valid for all x . (x in radians).

For small values of x , the sinusoidal function may be approximated as linear. We can say

$$\sin(x) \approx x$$

for $|x| \ll 1$. In general, linearization of a function is done by keeping the zeroth- and first-order terms of the Taylor-series expansion. For some functions, linearization gives a good approximation to its form in a small operating region. Other times, linearization does not work well at all. We are fortunate with the MagLev device to discover that linearization works well.

A nonlinear system is linearized around an “operating point”. For example, suppose we apply a constant control effort u_{1o} to the bottom coil to keep the bottom magnet levitated to a height of 2.5 cm in the absence of disturbances. Then, small changes in the control effort u_1 around the constant value of u_{1o} will lead to linear behavior in the magnet height, plus the constant offset of 2.5 cm. In this example, the operating “point” consists of the constant u_{1o} and the constant $y_{1o} = 2.5$ cm.

The linearized equations of motion are found by solving for the zeroth- and first-order terms of the Taylor-series expansion of the respective equations about the operating point. For example, the full set of equations of motion for the MagLev system are listed in Eqs. (B.1) to (B.7).

We customarily re-arrange the equations of motion so that one side of the equation contains the linear terms, and the other side of the equation contains the nonlinear terms. So, Eqs. (B.1) and (B.2) become:

$$\begin{aligned} m\ddot{y}_1 + m\beta\dot{y}_1 &= F_{u_{11}} + F_{u_{21}} - F_{m_{12}} - mg \\ m\ddot{y}_2 + m\beta\dot{y}_2 &= F_{m_{12}} - F_{u_{22}} - F_{u_{12}} - mg. \end{aligned}$$

Calling right-hand-side of the first equation $\alpha(y_1, y_2, u_1, u_2, t)$, and calling the right-hand-side of the second equation $\beta(y_1, y_2, u_1, u_2, t)$, we have

$$\begin{aligned} \alpha(y_1, y_2, u_1, u_2, t) \approx & \alpha(y_{1o}, y_{2o}, u_{1o}, u_{2o}, t) + \left. \frac{\partial \alpha}{\partial y_1} \right|_{y_{1o}, y_{2o}, u_{1o}, u_{2o}} \cdot (y_1 - y_{1o}) + \left. \frac{\partial \alpha}{\partial y_2} \right|_{y_{1o}, y_{2o}, u_{1o}, u_{2o}} \cdot (y_2 - y_{2o}) + \\ & \left. \frac{\partial \alpha}{\partial u_1} \right|_{y_{1o}, y_{2o}, u_{1o}, u_{2o}} \cdot (u_1 - u_{1o}) + \left. \frac{\partial \alpha}{\partial u_2} \right|_{y_{1o}, y_{2o}, u_{1o}, u_{2o}} \cdot (u_2 - u_{2o}), \end{aligned} \quad (\text{B.11})$$

where y_{1o} , y_{2o} , u_{1o} and u_{2o} are the respective magnet positions and control efforts that define the operating point. For the purposes of control design we choose the operating point to be at an equilibrium (the system is at rest and not accelerating) so that

$$\alpha(y_{1o}, y_{2o}, u_{1o}, u_{2o}, t) = (F_{u_{11}} + F_{u_{21}} - F_{m_{12}} - mg) \Big|_{y_{1o}, y_{2o}, u_{1o}, u_{2o}} = 0; \quad (\text{B.12})$$

$$\text{also, } \beta(y_{1o}, y_{2o}, u_{1o}, u_{2o}, t) = (F_{m_{12}} - F_{u_{22}} - F_{u_{12}} - mg) \Big|_{y_{1o}, y_{2o}, u_{1o}, u_{2o}} = 0. \quad (\text{B.13})$$

Evaluating Eq. (B.11) and using Eq. (B.12) we have

$$\begin{aligned} m\ddot{y}_1 + m\beta\dot{y}_1 + \left(\frac{4u_{1o}}{a(y_{1o} + b)^5} - \frac{4u_{2o}}{a(13 - y_{1o} + b)^5} + \frac{4c}{(y_{12o} + d)^5} \right) (y_1 - y_{1o}) - \frac{4c}{(y_{12o} + d)^5} (y_2 - y_{2o}) = \\ \frac{1}{a(y_{1o} + b)^4} (u_1 - u_{1o}) + \frac{1}{a(13 - y_{1o} + b)^4} (u_2 - u_{2o}), \end{aligned}$$

which may be rewritten as (where $y_1^* = y_1 - y_{1o}$, $u_1^* = u_1 - u_{1o}$ and so forth; note also that $\ddot{y}_1 = \ddot{y}_1^*$ and $\dot{y}_1 = \dot{y}_1^*$)

$$m\ddot{y}_1^* + m\beta\dot{y}_1^* + (k_{11} - k_{21} + k_{m12})y_1^* - k_{m12}y_2^* = k_{u11}u_1^* + k_{u12}u_2^*. \quad (\text{B.14})$$

Similarly for the second differential equation

$$m\ddot{y}_2^* + m\beta\dot{y}_2^* + (k_{22} - k_{12} + k_{m12})y_2^* - k_{m12}y_1^* = k_{u22}u_2^* + k_{u21}u_1^*, \quad (\text{B.15})$$

where

$$\begin{aligned}
 k_{11} &= \frac{4u_{1o}}{a(y_{1o} + b)^5}, & k_{12} &= \frac{4u_{1o}}{a(13 + y_{2o} + b)^5}, & k_{21} &= \frac{4u_{2o}}{a(13 - y_{1o} + b)^5}, & k_{22} &= \frac{4u_{2o}}{a(-y_{2o} + b)^5}, \\
 k_{u11} &= \frac{1}{a(y_{1o} + b)^4}, & k_{u12} &= \frac{1}{a(13 - y_{1o} + b)^4}, & k_{u21} &= \frac{-1}{a(13 + y_{2o} + b)^4}, & k_{u22} &= \frac{-1}{a(-y_{2o} + b)^4}, \\
 k_{m12} &= \frac{4c}{(y_{12o} + d)^5}.
 \end{aligned}$$

For small motions, and if cross terms are negligible, the dynamics are identical in form to those of the linear discrete system shown in Fig. B.16.

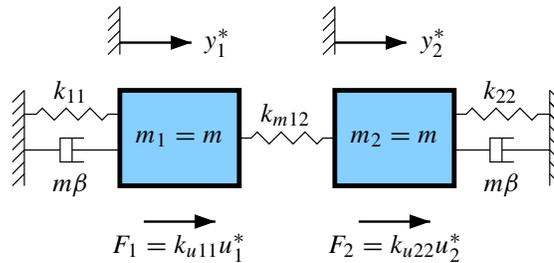


Figure B.16 Approximate linear block diagram of MIMO MagLev system.

B.4.2 Calculating u_{1o} and u_{2o}

We may calculate the steady-state control effort u_{1o} and u_{2o} for a specific operating point $\{y_{1o}, y_{2o}\}$ by evaluating Eqs. (B.12) and (B.13). These give the two equations and two unknowns which may be combined into a matrix equation:

$$\begin{bmatrix} \frac{1}{a(y_{1o}+b)^4} & \frac{1}{a(13-y_{1o}+b)^4} \\ \frac{-1}{a(13+y_{2o}+b)^4} & \frac{-1}{a(-y_{2o}+b)^4} \end{bmatrix} \begin{bmatrix} u_{1o} \\ u_{2o} \end{bmatrix} = \begin{bmatrix} mg + \frac{c}{(y_{12o}+d)^4} \\ mg - \frac{c}{(y_{12o}+d)^4} \end{bmatrix}.$$

B.5 Using the Model in Simulink

Figure B.17 shows how to incorporate the MagLev into a Simulink diagram. The “Fix Sensor” Simulink subsystems will be provided to you. These counteract the nonlinearities in the sensor.

The following chart is a summary of fitting parameters for the actuators and sensors of the four systems:

Sys	$a \left[\frac{V}{N \text{ cm}^3 \text{ m}} \right]$	$b \text{ [cm]}$
1	0.00000416	5.575067
2	0.00000384	5.743740
3	0.00000407	5.700130
4	0.00000576	4.383893

$c = 4408140 \text{ [N cm}^5/\text{m]}$, $d = 5.193394 \text{ [cm]}$. The mass of each magnetic disk is $m = 0.12 \text{ kg}$. The friction coefficient $\beta \approx 5 \text{ [units]}$.

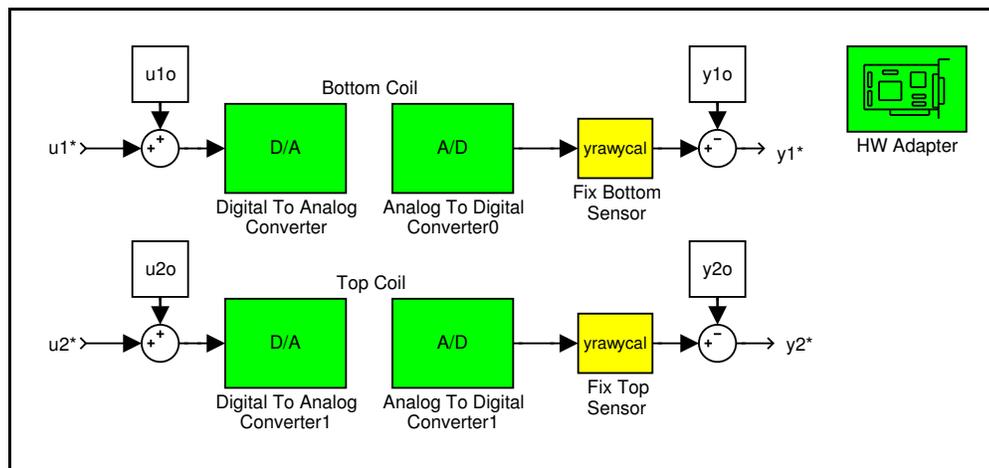


Figure B.17 Diagram showing u_k^* input to plant, and y_k^* computed from plant output. You will use this as a portion of your control design.

(mostly blank)