

## Manual For Model 505

# Inverted Pendulum

(Instructor's Edition)

### Important Notice

Section 2.3 of this manual contains important safety information that must be read by all users prior to operating the ECP system.

This manual is for use in direct association with ECP experimental systems only and may be reproduced only for that purpose. All other use, dissemination, or reproduction in whole or in part without the written permission of ECP is strictly prohibited.

© COPYRIGHT 1994, by ECP Educational Control Products. All rights reserved.

This Edition is compatible with Executive Software Version 2.0

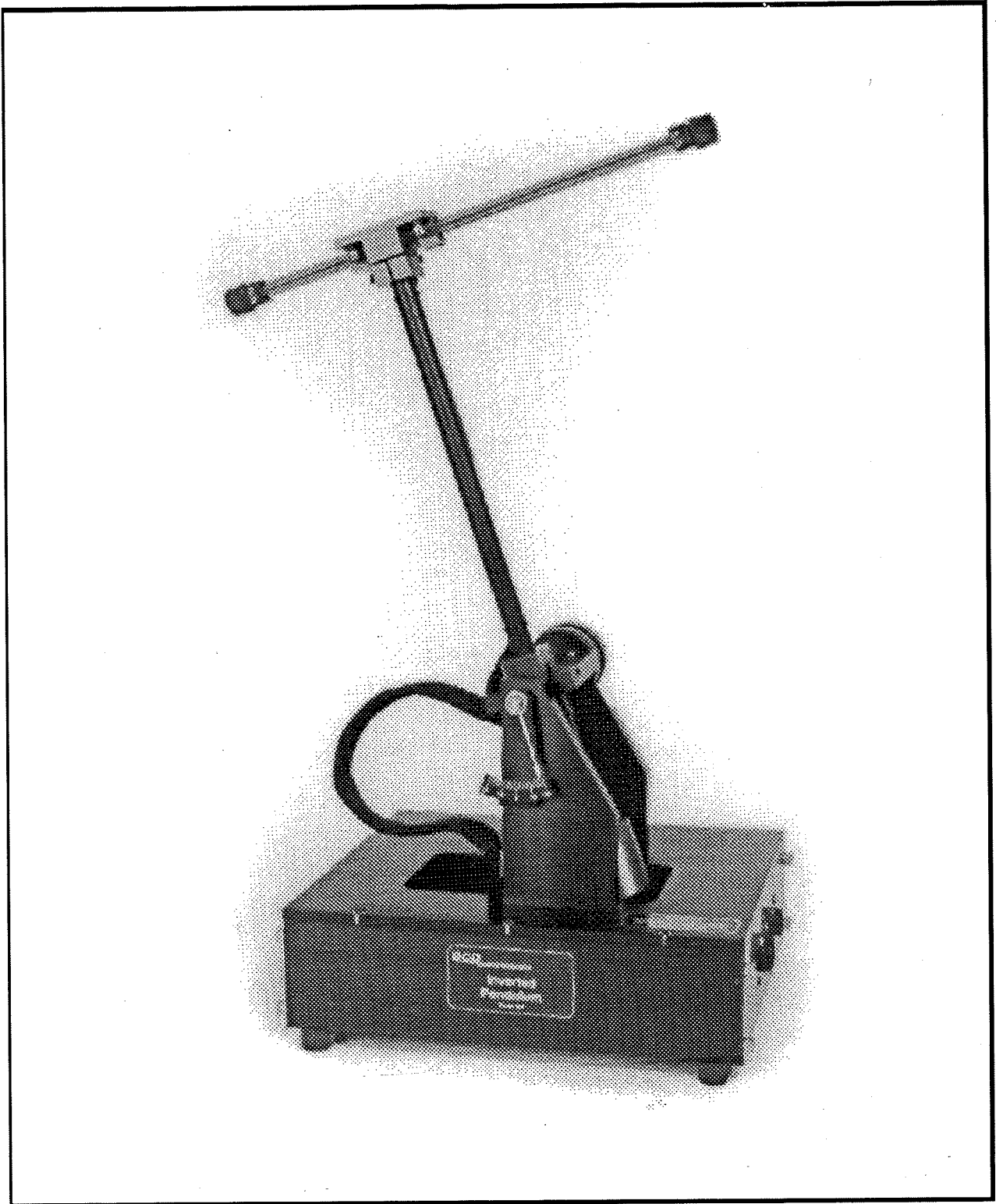
**eCP**

Educational Control Products  
1 Buckskin Court  
Bell Canyon, CA 91307

Educational Control Products

ph: (818) 703-0802

25



**The Inverted Pendulum**

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	System Overview	1
1.2	Manual Overview	2
<b>2</b>	<b>System Description &amp; Operating Instructions</b>	<b>4</b>
2.1	ECP Executive Software	4
2.2	Electromechanical Plant	25
2.3	Safety	28
<b>3</b>	<b>Start-up &amp; Self-Guided Demonstration</b>	<b>32</b>
3.1	Hardware Set-up Verification	32
3.2	Demonstration Of ECP Executive Program	34
<b>4</b>	<b>Real-time Control Implementation</b>	<b>40</b>
4.1	Servo Loop Closure	40
4.2	Command Generation	42
4.3	Servo Moter & Amplifier	46
4.4	Multi-tasking Environment	48
4.5	Sensors	49
4.6	Auxiliary Analog Output	50
<b>5</b>	<b>Plant Dynamic Models</b>	<b>51</b>
5.1	Equations Of Motion	51
5.2	Effect Of Mass Properties & Geometry On Plant Dynamics	55
<b>6</b>	<b>Experiments</b>	<b>60</b>
6.1	System Identification	60
6.2	Successive Loop Closure / Pole Placement Design	67
6.3	Low Pass Filter Augmentation To State Feedback Control	73
6.4	Pole Placement Controller For Unstable $\theta(s)$ / $x(s)$	74
6.5	LQR Control Design	75
6.6	Tracking Response	78

<b>A</b>	<b>Generic Control Issues: Nonminimum Phase &amp; Unstable Plants</b>	<b>79</b>
A.1	Generic Control Issues	79
A.2	Effect Of Right Half Plane Poles And Zeros	81
A.3	Numerical Values Of Bandwidth Limitations	83

*Instructor's Manual Only*

<b>6i</b>	<b>Instructor's Supplement To Experiments</b>	<b>85</b>
6.1i	System Identification	85
6.2i	Successive Loop Closure / Pole Placement Design	87
6.3i	Low Pass Filter Augmentation To State Feedback Control	91
6.4i	Pole Placement Controller For Unstable $\theta(s)$ / $x(s)$	92
6.5i	LQR Control Design	93
6.6i	Tracking Response	94
6.7i	Suggestions For Further Experiments	96
<b>Ai</b>	<b>Useful Matlab® Scripts</b>	<b>98</b>
A.1i	Plant Model Builder	98
A.2i	Successive Loop PD / Pole Placement	101
A.3i	LQR Synthesis	102
A.4i	Nonlinear Simulation	103

# 1 Introduction

Welcome to the ECP line of educational control systems. These systems are designed to provide insight to control system principles through hands-on demonstration and experimentation. Each consists of an electromechanical plant and a full complement of control hardware and software. The user interface to the system is via a friendly, versatile, PC window environment which supports a broad range of controller specification, trajectory generation, data acquisition, and plotting features. The systems are designed to accompany introductory through advanced level controls courses and support either high level usage (i.e. direct controller specification and execution) or machine level implementation study.

The Model 505 Inverted Pendulum apparatus is a unique mechanism which provides vivid demonstrations and challenging experiments for both undergraduate and graduate studies in controls. This novel design steers a horizontal sliding rod in the presence of gravity to balance and control the position of the vertical ("pendulum") rod. The mechanism is open loop unstable (right half plane pole) and non-minimum phase (right half plane zero). As a result feedback control is essential for stability and the structure of the controller must be selected carefully due the non-minimum phase characteristics.

## 1.1 System Overview

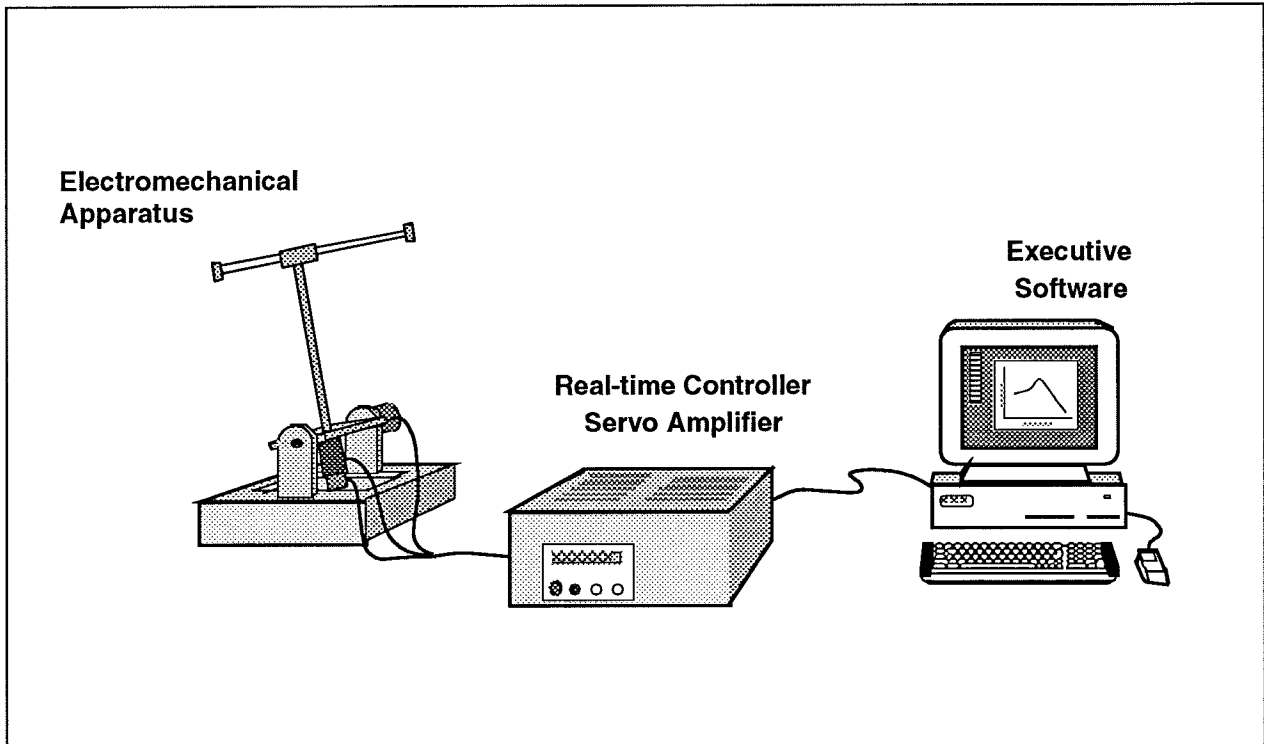
The experimental control system is comprised of the three subsystems shown in Figure 1.1-1. The first of these is the electromechanical plant which consists of the inverted pendulum mechanism, its actuator and sensors. The design features a DC servo motor, high resolution encoders, a low friction sliding balance rod, and adjustable balance weight.

Next is the real-time controller unit which contains the digital signal processor (DSP) based real-time controller<sup>1</sup>, servo/actuator interfaces, servo amplifier, and auxiliary power supplies. The DSP

---

<sup>1</sup>The system is also available in a PC bus installation form in which the DSP based real-time controller resides in the PC and all other control unit hardware remains in a separate box. This form has faster PC/controller communication rates. (Controller speed is unaffected.)

is capable of executing control laws at high sampling rates allowing the implementation to be modeled as continuous or discrete time. The controller also interprets trajectory commands and supports such functions as data acquisition, trajectory generation, and system health and safety checks. A logic gate array performs encoder pulse decoding. Two auxiliary digital-to-analog converters (DAC's) provide for real-time analog signal measurement. This controller is representative of modern industrial control implementation.



**Figure 1.1-1. The Experimental Control System**

The third subsystem is the executive program which runs on a PC in MS DOS. This menu-driven program is the user's interface to the system and supports controller specification, trajectory definition, data acquisition, plotting, system execution commands, and more. Controllers may assume a broad range of selectable block diagram topologies and dynamic order. The interface supports an assortment of features which provide a friendly yet powerful experimental environment.

## 1.2 Manual Overview

The next chapter, Chapter 2, describes the system and gives instructions for its operation. Section 2.3 contains important information regarding safety and is mandatory reading for all users prior to operating this equipment. Chapter 3 is a self-guided demonstration in which the user is easily walked through the salient system operations before reading all of the details in Chapter 2. A

description of the system's real-time control implementation as well as a discussion of generic implementation issues is given in Chapter 4. Chapter 5 presents dynamic equations useful for control modeling. Chapter 6 gives some example experiments including system identification, pole placement, and LQR control. Finally, Appendix A gives some details of the development of plant modeling equations.

## 2 System Description & Operating Instructions

This chapter contains descriptions and operating instructions for the executive software and the mechanism. The safety instructions given in Section 2.3 must be read and understood by any user prior to operating this equipment.

### 2.1 ECP Executive Software

The ECP Executive software is the user's interface to the system. It is a menu driven / window environment that the user will find is intuitively familiar and quickly learned. This software runs on an IBM PC or compatible computer and communicates with ECP's digital signal processor (DSP) based real-time controller. Its primary functions are supporting the downloading of various control algorithm parameters (gains), specifying command trajectories, selecting data to be acquired, and specifying how data should be plotted. In addition, various utility functions ranging from saving the current configuration of the Executive to specifying analog outputs on the auxiliary DAC's are included as menu items.

#### 2.1.1 PC System Requirements

The ECP Executive runs on any PC/AT or compatible computer. You will need at least 1 megabyte of RAM and a hard disk drive with at least 3 megabytes of space. All versions of the Executive program run under any DOS versions 3.x, 4.x, and 5.x. Be aware that DOS version 4.x has had reported problems working with various software packages. The Executive requires a VGA monitor with a VGA graphics card installed on the PC.

The Executive Program runs best on a 386 or 486 with 2 megabytes of memory (or more) under DOS 5.0 with HIGHMEM.SYS driver included in your CONFIG.SYS file.<sup>1</sup> Also, if the software does not "see" at least 2 megabytes of free RAM, you may find the program executing somewhat slowly since it will use the hard disk as virtual memory.

---

<sup>1</sup>A faster computer, such as 25 Mhz 486 with the real-time controller on the PC bus provides a much more expeditious working environment than a 386 or 286 and/or RS232 controller/PC communication. Real-time control speed, however, is unaffected.



### 2.1.2 Installation Procedure

The ECP Executive Program consists of several files on either a 5.25" 1.2 megabyte (high density) or 3.25" 1.44 megabyte distribution diskette. The files on the distribution diskette are:

ECP.EXE

ECP.DAT

\*.CFG

\*.PLT

\*.PMC

The "ECP.\*" files are needed to run the Executive Program. The "\*.CFG" and "\*.PLT" files are some controller configuration and plotting files that are included for the initial self-guided demonstration. The "\*.PMC" file is the controller Personality File and should only be used in the case of a non curable system failure (see "Utility Menu" below).

To install the Executive program, we recommend that you make a dedicated subdirectory on the hard disk and enter this subdirectory. For example type:

```
>MD ECP
```

```
>CD ECP
```

Next insert the distribution diskette in either "A:" or "B:" drive, as appropriate. Copy all files in the distribution diskette to the hard disk under the "ECP" sub directory. For example if the "B:" drive is used:

```
>COPY B:*. * C:
```

Next execute INSTALL.EXE by typing:

```
>INSTALL
```

You will notice some file decompression activities. This completes the installation procedure. You may run the ECP Executive by typing:

```
>ECP
```

The Executive program is window based with pull-down menus and dialog boxes. You may either use the cursor keys on the keyboard or a mouse to make selections from the pull-down menus. Vertical movement within these menus is accomplished by the up and down arrow keys, respectively. To make a selection with the keyboard, simply highlight the desired choice and press <ENTER>. Menu choices with highlighted letters may also be selected by pressing the corresponding function key. (The indicated key for menus; "alt" plus the indicated key within dialog boxes).

Within dialog boxes, movement from one object to the next is accomplished by using the <TAB> and the <SHIFT-TAB> keys. Here, "objects" includes input lines, check boxes, and "radio buttons". As you move from one object to the next, the focused object is highlighted, indicating that it is selected. Pressing <ENTER> will select the highlighted button (e.g. termination of the dialog box will result if the **OK** button is highlighted).

### 2.1.3 Background Screen

The *Background Screen*, shown in Figure 2.1-1, remains in the background during system operation including times when other menus and dialog boxes are active. It contains the main menu and a display of real-time data, system status, and an *Abort Control* button to immediately discontinue control effort in the case of an emergency.

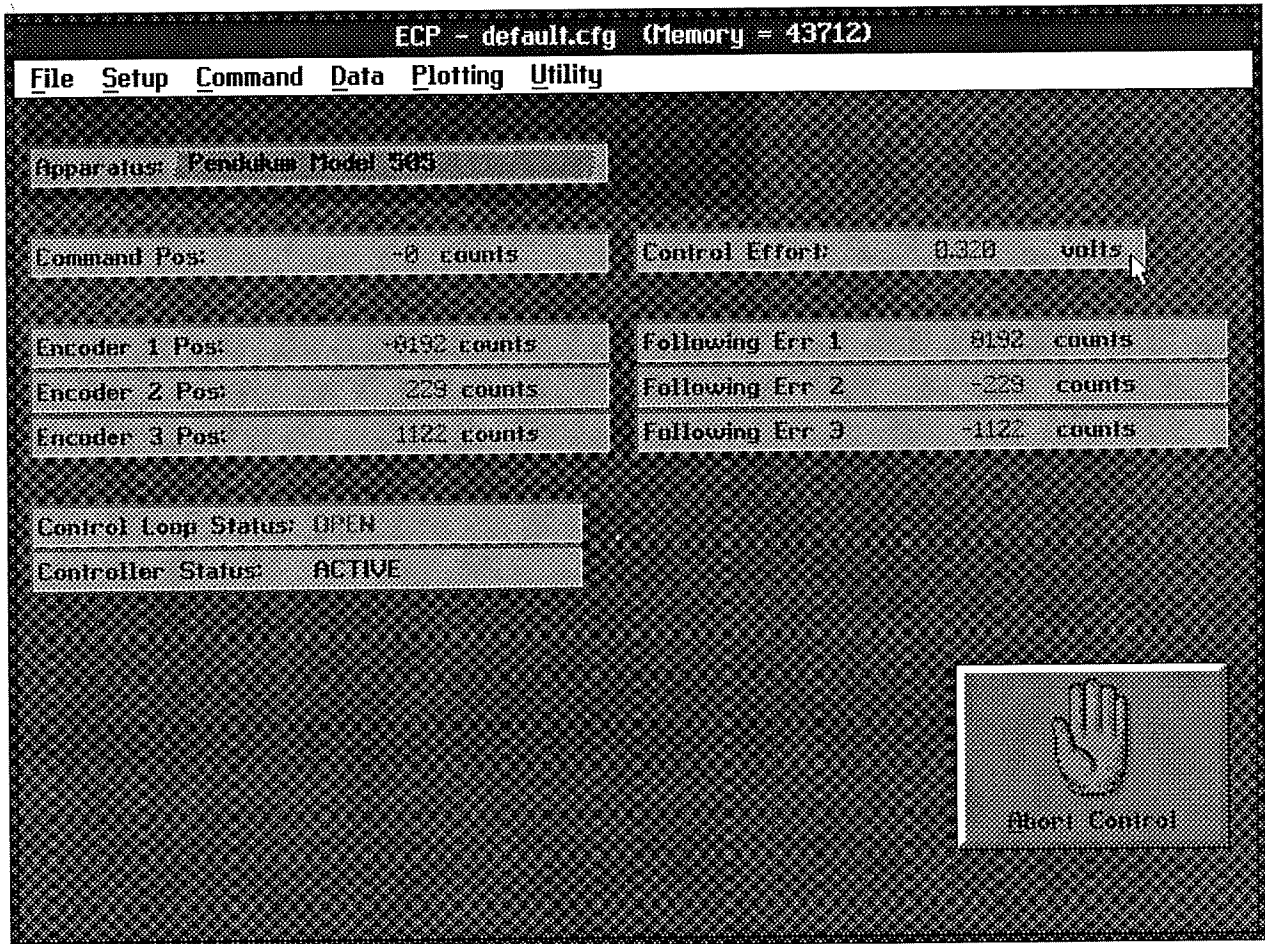


Figure 2.1-1. The Background Screen

### 2.1.3.1 Real-Time Data Display

In the *Data Display* fields, the instantaneous commanded position, the encoder positions, the following errors (instantaneous differences between the commanded position and the actual encoder positions), and the control effort in volts (on the DAC) are shown.

### 2.1.3.2 System Status Display

The **Control Loop Status** ("Open" or "Closed"), indicates "Closed" unless an open loop trajectory is being executed or a "Limit Exceeded" condition has occurred. In either of these cases the Control Loop Status will indicate "Open". The **Controller Status** will indicate "Active" unless a motor over-speed, an over-travel (limit switch), or motor/amplifier over-temperature condition has occurred (see Section 2.3 for more details). In either of these cases the Controller Status will indicate "Limit Exceeded". The Limit Exceeded indicator will reoccur unless the user takes one of the two following actions depending on the nature of the over-limit cause. Either a stable controller (one that does not cause limiting conditions) must be implemented via the **Control Algorithm** box under the **Setup** menu or an acceptable trajectory must be executed under the **Command** menu. An "acceptable" trajectory is one that does not over-speed the motor, cause excessive travel of the sliding rod or result in sustained high current to the motor. The controller must be "re implemented" in order to clear the Limit Exceeded condition – see Section 2.1.5.1.1.

### 2.1.3.3 Abort Control Button

Also included on the Background Screen is the **Abort Control** button. Clicking the mouse on this button simply opens the control loop. This is a very useful feature in various situations including one in which a marginally stable or a noisy closed loop system is detected by the user and he/she wishes to discontinue control action immediately. Note also that control action may always be discontinued immediately by pressing the red "OFF" button on the control box. The latter method should be used in case of an emergency.

### 2.1.3.4 Main Menu Options

The *Main menu* is displayed at the top of the screen and has the following choices:

- File**
- Setup**
- Command**
- Data**
- Plotting**
- Utility**
- Jog Motor**

### 2.1.4 File Menu

The File menu contains the following pull-down options:

- Load Settings**
- Save Settings**
- About**
- Exit**

2.1.4.1 The **Load Settings** dialog box allows the user to load a previously saved configuration file into the Executive. A configuration file is any file with a ".cfg" extension which has been previously saved by the user using **Save Settings**. Any "\*.cfg" file can be loaded at any time. The latest loaded "\*.cfg" file will overwrite the previous configuration settings in the ECP Executive *but* changes to an existing controller residing in the DSP real-time control card will not take place until the new controller is "implemented" – see Section 2.1.5.1. The configuration files include information on the control algorithm, trajectories, data gathering, and plotting items previously saved. To load a "\*.cfg" file simply select the **Load Settings** command and when the dialog box opens, select the appropriate file from the desired directory.<sup>1</sup> Note that every time the Executive program is entered, a particular configuration file called "default.cfg" is loaded.

2.1.4.2 The **Save Settings** option allows the user to save the current control algorithm, trajectory, data gathering and plotting parameters for future retrieval via the **Load Settings** option. To save a "\*.cfg" file, select the **Save Settings** option and save under an appropriately named file (e.g. "pid2.cfg"). By saving the configuration under a file named "default.cfg" the user creates a default configuration file which will be automatically loaded on reentry into the Executive program. You may tailor "default.cfg" to best fit your usage.

2.1.4.3 Selecting **About** brings up a dialog box with the current version number of the Executive program.

2.1.4.4 The **Exit** option brings up a verification message. Upon confirming the user's intention, the Executive is exited.

### 2.1.5 Setup Menu

The **Setup** menu contains the following pull-down options:

- Control Algorithm**
- User Units**
- Communications**

---

<sup>1</sup>Its fastest to simply double-click on the desired file.

2.1.5.1 **Setup Control Algorithm** allows the entry of various control structures and control parameter values to the real-time controller – see Figure 2.1-2. In addition to feedforward which will be described later, the currently available feedback options are:

- PID
- PI With Velocity Feedback
- PID+Notch
- Dynamic Forward Path
- Dynamic Prefilter/Return Path
- State Feedback
- General Form

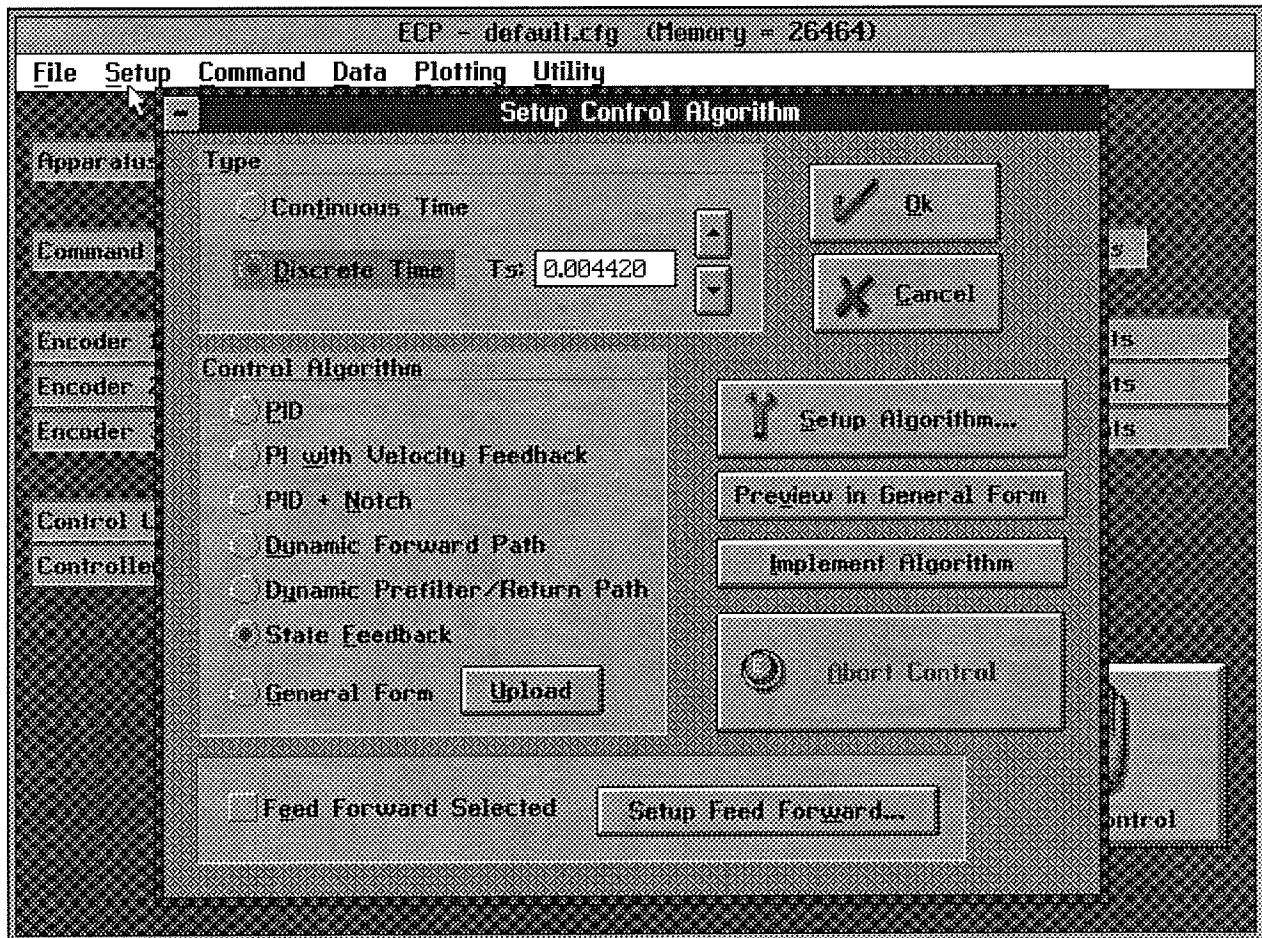


Figure 2.1-2. Setup Control Algorithm Dialog Box

2.1.5.1.1 Discrete Time Control Specification

The user chooses the desired option by selecting the appropriate "radio button" and then clicking on **Setup Algorithm**. The user must also select the sampling period which is always in multiples of 0.000884 seconds (1.1 KHz is the maximum sampling frequency).<sup>1</sup> To run the selected choice on the real-time controller click on the **Implement Algorithm** button. The control action will begin immediately. To stop control action and open the loop with zero control effort click on the **Abort** button. To upload the current controller select **General Form** then click on the **Upload Algorithm** button followed by **Setup Algorithm**. Here you will find the current controller in the form that is actually executed in real-time – see Figure 2.1-3.

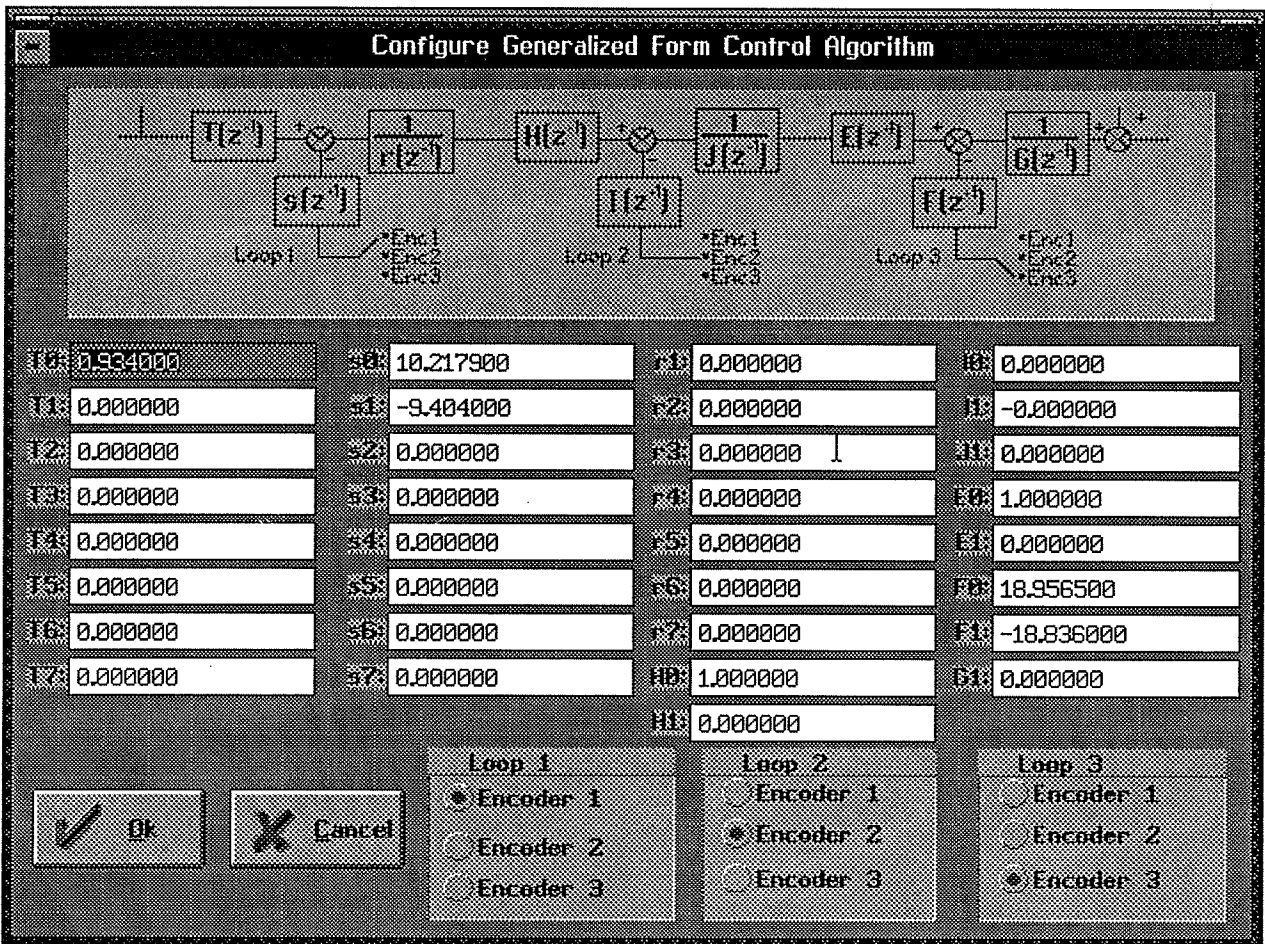


Figure 2.1-3. Dialog Box For Generalized Control Algorithm Input

<sup>1</sup> For many designs, the value  $T_s = .00442$  sec is a good midpoint between the spatial quantization induced noise associated with rate derivation over small sample periods and the phase lag associated with large ones.

A typical sequence of events is as follows: Select the desired servo loop closure sampling time  $T_s$  in multiples of 0.000884 seconds. Then select the control structure you wish to implement (e.g. radio buttons for PID, PID+Notch etc.). Select **Setup Algorithm** to input the gain parameters (coefficients). You must also select the desired feedback channel by choosing the correct encoder(s) used for your particular control design. Exit **Setup** by selecting **OK**. Now you should be back in the **Setup Control Algorithm** dialog box with a selected set of gains for a specified control structure. To download this set of control parameters to the real-time controller click on **Implement Algorithm**. This action results in an immediate running of your selected control structure on the real-time controller. If you notice unacceptable behavior (instability and/or excessive ringing or noise) simply click on **Abort Control** which opens up the control loop with zero control effort commanded to the actuator.

To inspect the form by which your particular control structure is actually implemented on the real-time controller, simply click on **Preview In General Form**. You may edit the algorithm in the **General Form** box, however when you exit, you must select **General Form** prior to "implementing" if you want the changes to become effective. (i.e. the radio button will still indicate the box you were in prior to previewing and this one will be downloaded unless **General Form** is selected).

The **Setup Feed Forward** option allows the user to add feedforward action to any of the above feedback structures. By clicking on this button a dialog box appears which allows the feedforward control parameters (coefficients) to be entered. To augment the feedforward action to the feedback algorithm the user must then check the **Feedforward Selected** check-box. Any subsequent downloading (via the **Implement Algorithm** button) combines the feedforward control algorithm with the selected feedback control algorithm The feedforward action remains in effect until all of its coefficients are cleared and downloaded again even when the check-box has been cleared. Thus to disable the feedforward action the user must go to its dialog box, set all of the coefficients to zero and download (via **Implement Algorithm**) the zero set of coefficients to the real-time controller with the check box checked. Alternatively, the user can reset the controller via **Reset** option in the **Utility** menu (see below). This action also clears out all of the coefficients of both the feedback and feedforward controllers.<sup>1</sup>

**Important Note:** Every time a set of control coefficients are downloaded via **Implement Algorithm** button, the commanded position as well as all of the encoder positions are reset to zero. This action is taken in order to prevent any instantaneous unwanted transient behavior from the controller. The control action then begins immediately.

**Important Note:** For high order control laws (those using more than 2 or 3 terms of either the R, S, T, K, or L polynomials), it is often important that the coefficients be entered with relatively high precision— say at least 5 to 6 points after the decimal. The real-time controller works with 96-bit

---

<sup>1</sup>The feedback gains will generally still reside in the appropriate feedback controller dialog box and may be re-downloaded via **Implement** once they have been verified.

real number arithmetic (48-bit integer plus 48-bit fraction). Although the Executive displays the coefficients with nine points after the decimal, it accepts higher precision numbers and downloads them correctly.

### 2.1.5.1.1 Continuous Time Control Specification

Depending on your course of study, It may be desirable to specify the control algorithm in continuous time form.<sup>1</sup> The method for inputting control parameters is identical to that described for the discrete time case. Again you may preview your controller in the continuous **General Form** prior to implementing. Upon selecting either **Implement Algorithm** or **Preview in General Form**, the algorithm also gets mapped into the discrete General Form where it may be viewed either before (following "Preview") or after (following "Implement") downloading to the real time controller.<sup>2</sup>

Again it is the discrete time general form that is actually executed in real time. The input coefficients are transformed to discrete time using one of the two following substitutions. For polynomials:  $n(s)$ ,  $d(s)$  in **PID + Notch**;  $s(s)$ ,  $t(s)$ , and  $r(s)$  in **Dynamic Forward Path, Dynamic Prefilter / Return Path**, and the **General Form**; and  $k(s)$ ,  $l(s)$  in **Feed Forward**, the Tustin (bilinear) transform

$$s = \frac{2}{T_s} \frac{1-z^{-1}}{1+z^{-1}}$$

is used. All other cases (first order) use the Backwards Difference method:

$$s = \frac{1-z^{-1}}{T_s}$$

Blocks using the Tustin transform must be proper in  $s$  while those using backwards difference may be improper – e.g. a differentiator.

2.1.5.2 The **User Units** dialog box provides the user with various choices of angular or linear units. For Model 505 the choices are *counts*, *centimeters*, *millimeters*, *inches*, *degrees*, and *radians*. There are 502 counts per centimeter travel of the sliding rod and 44.4 counts per degree

<sup>1</sup>A often used rule of thumb is that the continuous time approximation of sampled data systems is acceptably accurate if the sampling frequency is at least 10 times the system bandwidth. Since the attainable closed loop bandwidths for the mechanical system are in this case generally less than 5 Hz, sampling rates above 100 Hz generally provide results that are indistinguishable between equivalent continuous and discrete controller designs. I.e. for sampling rates above 50 Hz, the user may generally design and specify the controller in continuous time with no measurable difference in system behavior than if the controller were designed in discrete time.

<sup>2</sup>Note that in previewing the discrete generalized form of a continuous controller you should select **Discrete Time**, **General Form**, then **Setup Algorithm**. If instead the sequence **Discrete Time, Preview In General Form**, is used, then the selected discrete time algorithm (the one with the red dot next to it and which will not generally contain parameters that correspond to the continuous time design) will be previewed. Subsequent "Implementing" will then download the wrong design.



(16,000 counts per revolution) of the pendulum rod. By clicking on the desired radio button the units are changed automatically for trajectory inputs as well as the Background Screen displays, plotting and jogging activities. Units of *counts* are used exclusively for the examples in this manual.

2.1.5.3 The **Communications** dialog box is usually used only at the time of installation of the real-time controller. The choices are serial communication (RS232 mode) or PC-bus mode – see Figure 2.1-4. If your system was ordered for PC-bus mode of communication, you do not usually need to enter this dialog box unless the default address at 528 on the ISA bus is conflicting with your PC hardware. In such a case consult the factory for changing the appropriate jumpers on the controller. If your system was ordered for serial communication the default baud rate is set at 34800 bits/sec. To change the baud rate consult factory for changing the appropriate jumpers on the controller. You may use the **Test Communication** button to check data exchange between the PC and the real-time controller. This should be done after the correct choice of **Communication Port** has been made.

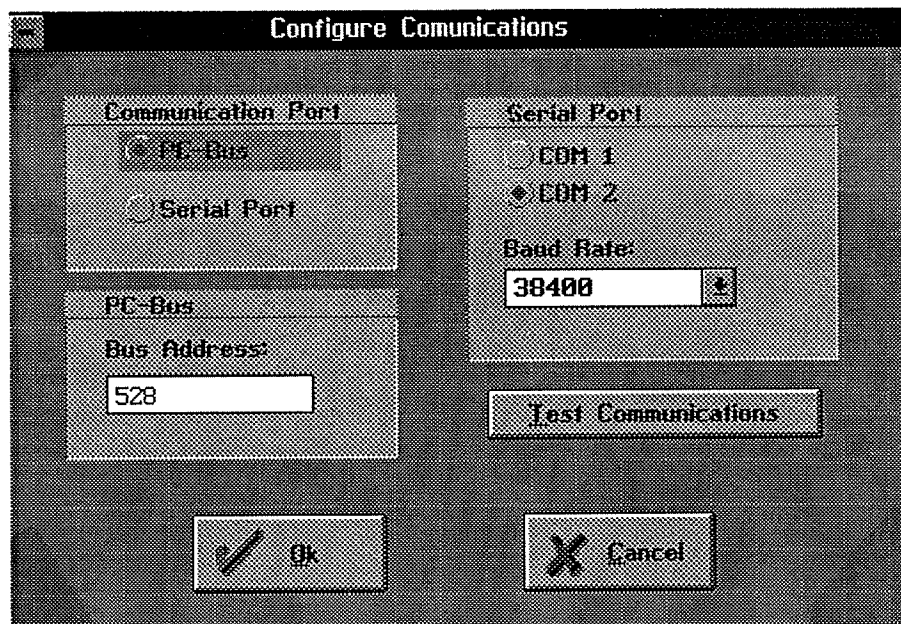


Figure 2.1-4. The Communications Dialog Box

### 2.1.6 Command Menu

The **Command** menu contains the following pull-down options

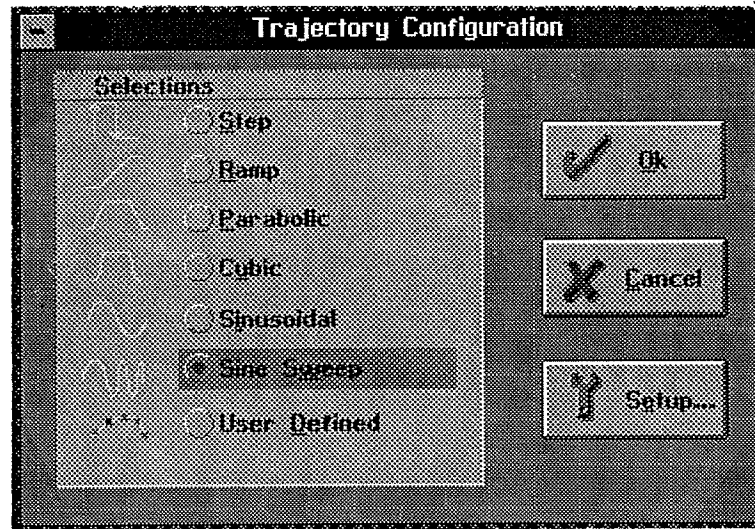
**Trajectory Configuration**

**Execute**

2.1.6.1 The **Trajectory Configuration** dialog box (see Figure 2.1.-5) provides a selection of trajectories through which the apparatus can be maneuvered. These are:

- Step**
- Ramp**
- Parabolic**
- Cubic**
- Sinusoidal**
- Sine Sweep**
- User Defined**

A mathematical description of these is given later in Section 4.1.



**Figure 2.1-5. The Trajectory Configuration Dialog Box**

By clicking the desired radio button followed by the Setup button one selects a specific dialog box for each trajectory.

The **Step** dialog box allows both **Closed loop** and **Open loop** step inputs. The **Closed loop** step subjects the closed loop system to a step command and is always in units of displacement (counts, inches, degrees etc.). The step size is incremental from the current commanded position and is always forward and backward with a specified dwell time and a number of repetitions. There are range limits for the maximum step size and dwell time which are apparatus-specific. Out-of-bounds inputs will cause an error message indicating the acceptable parameter range. The **Open**

**loop step**<sup>1</sup> subjects the plant to a step input and its units are always in volts. The maximum voltage is +/- 3 volts. Remember that a large open loop step size combined with a large open loop dwell time will result in an overtravel condition which is detected by the real-time controller. This condition will cause the open loop step test to be aborted and the **Controller Status** display in the Background screen to indicate **Limit Exceeded**. To run the test again you should reduce either or both the step size and the dwell time. Also note that for very large closed loop step sizes the **Limit Exceeded** condition may occur. This is generally true for all trajectories whose parameters have been selected such that they generate either too large a motion or a motor/amplifier over-temperature (stalled) condition (see Section 2.3 on safety).

The **Ramp** dialog box allows a constant speed closed loop input command. The displacement size is incremental from the current commanded position and is always forward and backward with a specified speed, dwell time, and number of repetitions.

The **Parabolic** trajectory allows a constant acceleration (quadratic in position) closed loop input command. The displacement size is again incremental from the current position and is always forward and backward with a specified acceleration time, speed, dwell time and number of repetitions. Note that the total displacement time may be longer than acceleration/deceleration time depending on the selected displacement size and the speed input. In this case the parabolic acceleration/deceleration curves are joined by a constant velocity ramp.

The **Cubic** option allows a constant jerk (cubic in position) closed loop input command. As before, displacement size is incremental from the current position and is always forward and backward with a specified acceleration time, speed, dwell time and number of repetitions. Again, the total displacement time may be longer than acceleration/deceleration time depending on the selected displacement size and the speed input.

Note that the only difference between a parabolic trajectory and a cubic trajectory is that, during the acceleration/deceleration times a constant acceleration is commanded in a parabolic input and a constant jerk (linearly changing acceleration) is commanded in the cubic input. Of course, in a ramp input the commanded acceleration/deceleration is infinite at the ends of a commanded displacement stroke and zero at all other times during the motion.

The **Sinusoidal** dialog box provides for both **Closed loop** and **Open loop** sine wave inputs. The **Closed loop** option subjects the closed loop system to a sine wave command with amplitude in units of displacement (counts, inches, degrees etc.). The amplitude is incremental from the current

---

<sup>1</sup>The open loop option here and for sinusoidal inputs may not be highly useful for many users of Model 505 as the plant  $\theta(s)/F(s)$  is unstable for all achievable mass property values. By clamping the pendulum rod, however it is possible to characterize the dynamics of the sliding rod in open loop tests, and by inverting the apparatus (hence making it a non-inverted pendulum) certain mass properties can be verified.

commanded position. The user also specifies the frequency in Hz and the number of cycles. The **open loop** option specifies a sine wave input to the plant with amplitude in volts (at the DAC). The maximum voltage is +/- 3 volts. Remember that a open loop input to an unstable plant will result in an overtravel condition. Also note that very high frequency large amplitude closed loop tests or smaller commands near a resonant frequency result in the **Limit Exceeded** condition. In general, all trajectories which generate either too large a travel, or excessive motor power will cause this condition – see the safety section 2.3. These conditions will cause the open loop test to be aborted and the **Controller Status** display in the Background Screen to indicate **Limit Exceeded**.

The **Sine Sweep** dialog box supports both **Closed Loop** and **Open Loop** sine sweep inputs. The **Closed Loop** option specifies a sine sweep in units of displacement (counts, inches, degrees etc.). The amplitude is incremental from the current commanded position. The user also specifies the starting and the ending frequencies in Hz and the sweep time. The frequency increase is linear in time. For example a sweep from 0 Hz to 10 Hz in 10 seconds results in a one Hertz per second frequency increase. There is an apparatus-specific amplitude limit beyond which the Executive will not accept the inputs. The **Open Loop** sine sweep subjects the plant to a sine sweep input whose units are always in volts. The maximum voltage is +/- 3 V. Remember again that any of the following may result in a **Limit Exceeded** condition: large open loop amplitude size combined with a low frequency; high frequency large amplitude closed loop tests and operation near or through resonances.<sup>1</sup>

The **User Defined** trajectory dialog box provides the interface for the input of any form of trajectory created by the user. In order to make use of this feature the user must first create an ASCII text file with an extension ".trj" (e.g. "random.trj"). The content of this file should be as follows:

The first line should provide the number of points in the trajectory. The maximum number of points is limited to 100. This line should not contain any other information. The subsequent lines (up to 100) should contain the consecutive set points. For example to input twenty points equally spaced in distance one can create a file called "example.trj" using any text editor as follows

```
20
5
10
15
20
25
30
```

---

<sup>1</sup>Sweeping through resonances is very useful in visualizing the frequency response dynamics, but must be done at a sufficiently low input amplitude. When viewing open loop sine sweep results, it is often best to view velocity rather than position data to reduce position drift effects. By selecting a relatively long sweep period, the transient effects of freq. change are minimized and the true freq. response is best approximated.

35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100

Now the **segment time** which is a time between each consecutive point can be changed in the dialog box. For example if a 100 milliseconds segment time is selected, the above trajectory would take 2 seconds to complete ( $100 \times 20 = 2000$  ms). The minimum **segment time** is restricted to five milliseconds by the real-time controller. The format of any ".trj" file is the same regardless of whether it was created for an open loop test or a closed loop test. When the points of a ".trj" file are selected for an open loop test their units are assumed to be in volts. For the closed loop tests the units are the current displacement units (counts, degrees, or radians). Obviously a user defined trajectory may also cause over-speed or over-deflection of the plant if the segment time is too short and the distance between the consecutive points is too long. Finally note that the closed loop user defined trajectories are cubic spline fitted in-between consecutive points by the real-time controller.

2.1.6.2 The **Execute** dialog box (see Figure 2.1-6) is normally entered after a trajectory is selected.<sup>1</sup> Here the user has a choice of sampling the data by clicking the **Sample Data** check box or not sampling data by clearing the check box (for the details of Data Gathering see "**Setup Data Acquisition**" below). To move the system through the currently specified trajectory, click on the **Run** button; the trajectory will be executed by the real-time controller. Once finished, and provided the **Sample Data** check box was checked, the data will be uploaded back into the Executive for plotting, saving and exporting. At any time during the execution of the trajectory or during the uploading of data the process may be terminated by clicking on the **Abort** button.

---

<sup>1</sup>Otherwise the last specified or otherwise potentially unknown trajectory will be executed.

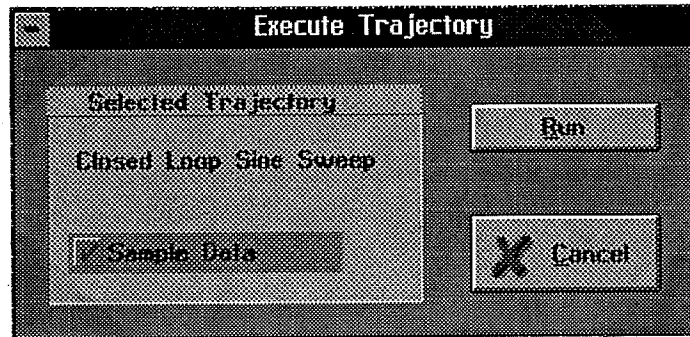


Figure 2.1-6. The Execute Dialog Box

### 2.1.7 Data Menu

The **Data** menu contains the following pull-down options

- Setup Data Acquisition**
- Upload Data**
- Export Raw Data**

2.1.7.1 **Setup Data Acquisition** allows the user to select one or more of the following data items to be collected at a chosen multiple of the servo loop closure sampling period while running any of the trajectories mentioned above – see Figures 2.1-7 and 4.1-1:

- Commanded Position**
- Encoder 1 Position**
- Encoder 2 Position**
- Encoder 3 Position** (not used for Model 505)
- Control Effort** (output to the servo loop or the open loop command)
- Node A** (input to the H polynomial in the Generalized Control Algorithm)
- Node B** (input to the E polynomial in the Generalized Control Algorithm)
- Node C** (output of the 1/G polynomial in the Generalized Control Algorithm)
- Node D** (output of the feedforward controller which is added to the node C value to form the combined regulatory and tracking controller).

In this dialog box the user adds or deletes any of the above items by first selecting the item, then clicking on the **Add Item** or **Delete Item** button. The user must also select the data gather sampling period (in multiples of the servo period). For example, if the sample time ( $T_s$  in the **Setup Control Algorithm**) is 0.00442 seconds and you choose 5 for your gather period here, then the selected data will be gathered once every fifth sample or once every 0.0221 seconds. Usually

for trajectories with fast frequency contents (e.g. Step, or high frequency Sine Sweep), one should choose a low data gather period. On the other hand, one should avoid gathering more often (or more data types) than needed since the upload and plotting routines become slower as the data size increase.

2.1.7.2 Selecting **Upload Data** allows any previously gathered data to be uploaded into the Executive. This feature is useful when one wishes to switch and compare between plotting previously saved raw data and the currently gathered data. Remember that the data is automatically uploaded into the executive whenever a trajectory is executed and data acquisition is enabled. However, once a previously saved raw data file is loaded into the Executive, the currently gathered data is overwritten. Now the **Upload Data** feature allows the user to bring the overwritten data back from the real-time controller into the Executive.

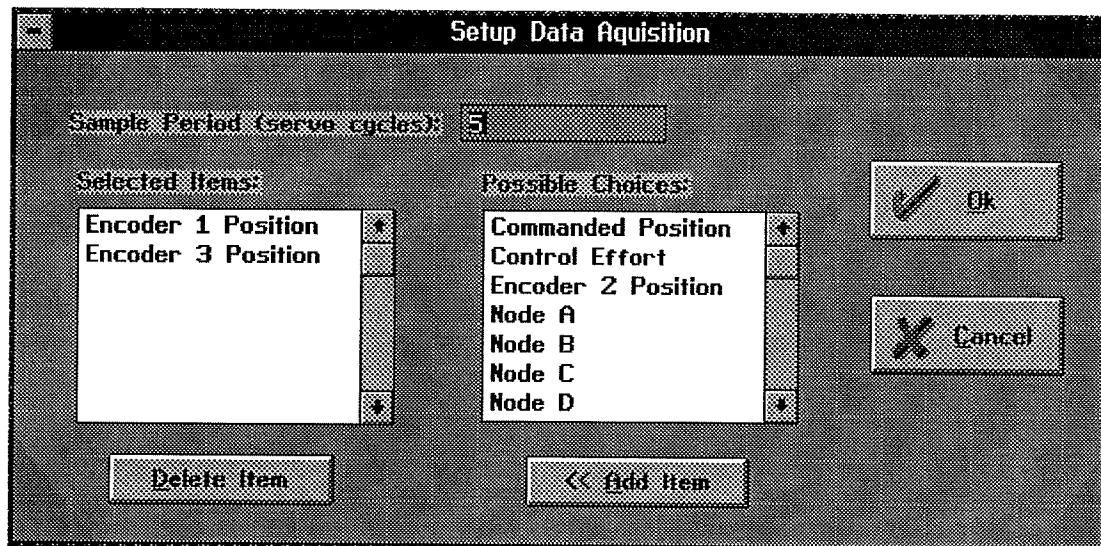


Figure 2.1-7. The Setup Data Acquisition Dialog Box

2.1.7.3 The **Export Raw Data** function allows the user to save the currently acquired data in a text file in a format suitable for reviewing, editing, or exporting to other engineering/scientific packages such as Matlab®.<sup>1</sup> The first line is a text header labeling the columns followed by bracketed rows of data items gathered. The user may choose the file name with a default extension

<sup>1</sup>The bracketed rows end in semicolons so that the entire file may be read as an array in Matlab by running it as a script once the header is stripped i.e. the script should be: <array name>= [exported data file]. Variable values over time are the columns of this array; the rows are the variable value set at successive sample numbers.

of ".txt" (e.g. lqrstep.txt). The first column in the file is sample number, the next is time, and the remaining ones are the acquired variable values. Any text editor may be used to view and/or edit this file.

### 2.1.8 Plotting Menu

The **Plotting** menu contains the following pull-down options

- Setup Plot**
- Plot Data**
- Print Data**
- Load Plot Data**
- Save Plot Data**
- Close Window**

2.1.8.1 The **Setup Plot** dialog box (see Figure 2.1-8) allows up to four acquired data items to be plotted simultaneously – two items using the left vertical axis units, and two using the right vertical axis units<sup>1</sup>. More than four items cannot appear on the same plot. Simply click on the item you wish to add to the left or the right axis and the click on the **Add to Left Axis** or **Add to Right Axis** buttons. You must select at least one item for the left axis before plotting is allowed – i.e. if only one item is plotted, it must be on the left axis. You may also change the plot title from the default one in this dialog box.

Items for comparison should appear on the same axis (e.g. commanded vs. encoder position) to ensure the same axis scaling and bias. Items of dissimilar scaling or bias (e.g. control effort in volts and position in counts) should be placed on different axes.

2.1.8.2 **Plot Data** generates a plot of the selected items. By clicking on the upper blue border of the plots, they may be dragged across the screen. The view size may be maximized by clicking on the up arrow of the upper right hand corner. It can also be shrunk to an icon by clicking on the down arrow of the upper left hand corner. This function is very useful for comparing several graphs. It can be expanded back to the full size at any time by double-clicking on the icon. Also more than one plot can be cascaded on the Background Screen<sup>2</sup>. By clicking on any point within the area of the desired plot it will appear over the others. Plots may be arbitrarily shaped by

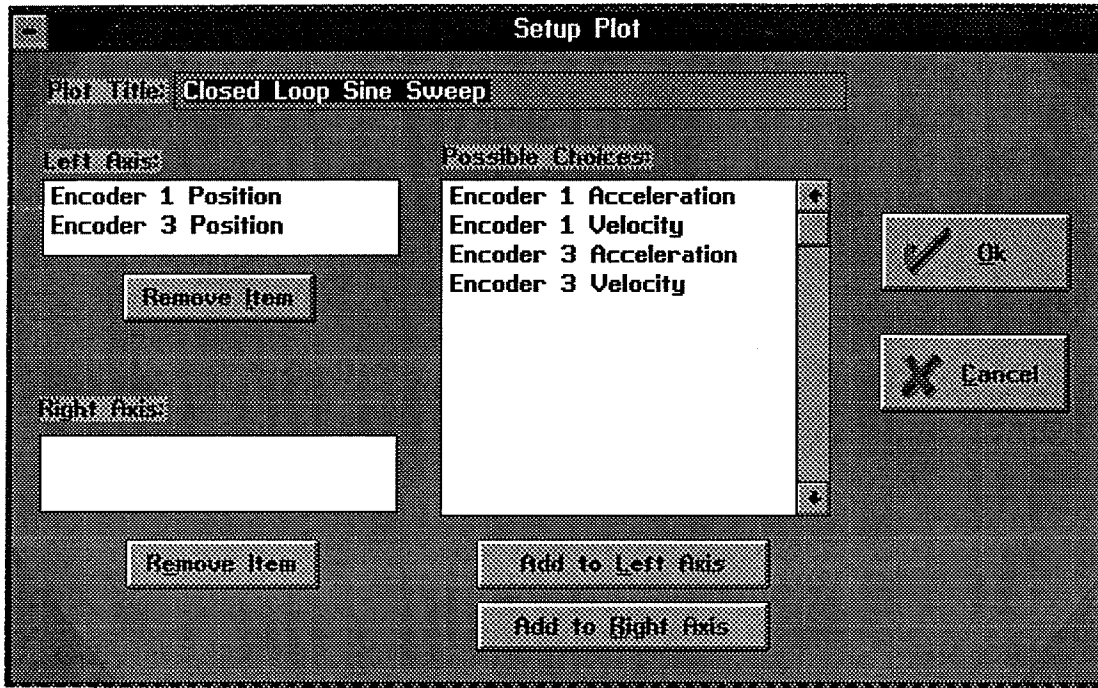
---

<sup>1</sup>You will see in the box plotting selections of velocity and acceleration for the position variables acquired. These are automatically generated during the uploading process.

<sup>2</sup>To avoid system slow-down, the number of open plots on the background screen should not become excessive.



moving the cursor to the lower right hand corner to the position where it becomes a double-arrow . The corner may then be "dragged" to reshape the plot. Finally by clicking on the top left hand corner of a plot screen one can close the plot window.



**Figure 2.1-8 The Setup Plot Dialog Box** (Shows case where data was gathered for encoders 1 and 3 only. Up to 20 variables may be made available for plotting)

2.1.8.3 The **Print Data** option allows the user to provide a hard copy of the selected plot on either an Epson compatible dot matrix printer or a HP Laserjet compatible printer.

2.18.4 The **Load Plot Data** dialog box enables the user to bring into the Executive previously saved ".plt" plot files. Note that such files are not stored in a format suitable for use by other programs. The ".plt" plot files contain the sampling period of the previously saved data. As a result, after plotting any previously saved plot files and before running a trajectory, you should check the servo loop sampling period  $T_s$  in the **Setup Control Algorithm** dialog box. If this number has been changed, then correct it. Also, check the data gathering sampling period in the **Data Acquisition** dialog box, this too may be different and need correction.

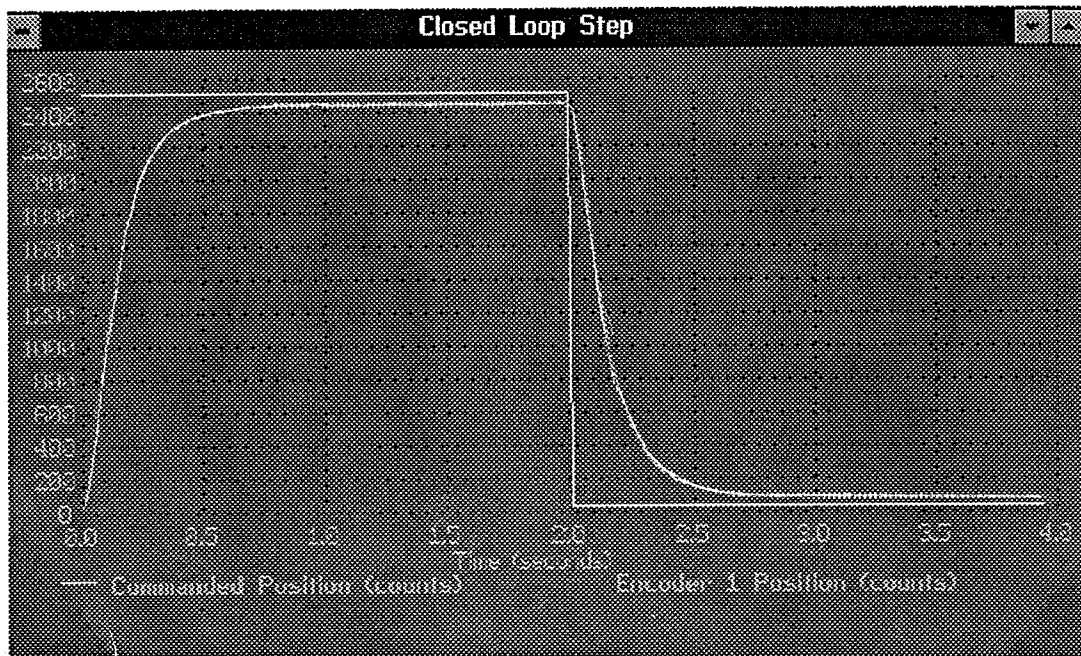


Figure 2.1-9. A Typical Plot Window

2.1.8.5 The **Save Plot Data** dialog box enables the user to save as a plot file of the data gathered by the controller. The default extension is ".plt" under the current directory. Note that ".plt" files are not saved in a format suitable for use by other programs. For this purpose the user must use the **Export Raw Data** option of the **Data** menu.

2.1.8.6 The **Close Window** option allows the currently marked plot window to close. This can also be done by clicking on the top left hand corner of the plot window.

### 2.1.9 Utility Menu

The **Utility** menu contains the following pull-down options:

- Configure Auxiliary DACs**
- Jog Position**
- Zero Position**
- Reset Controller**
- Rephase Motor**
- Down Load Controller Personality File**

2.1.9.1 The **Configure Auxiliary DACs** dialog box (see Figure 2.1-10) enables the user to select various items for analog output on the two analog channels in front of the ECP Control Box. Using equipment such as an oscilloscope, plotter, or spectrum analyzer the user may inspect the following items continuously in real time:

**Commanded Position**

**Encoder 1 Position**

**Encoder 2 Position**

**Encoder 3 Position** (Not used for Model 505)

**Control Effort**

**Node A**

**Node B**

**Node C**

**Node E**

The scale factor which divides the item can be less than 1 (one). The DACs analog output is in the range of +/- 10 volts corresponding to +32767 to -32768 counts. For example to output the commanded position for a sine sweep of amplitude 2000 counts you should choose the scale factor to be 0.061 ( $2000/32767=0.061$ ) This gives close to full +/- 10 volt reading on the analog outputs. In contrast, if the numerical value of an item is greater than +/- 32767 counts, for full scale reading, you must choose a scale factor of greater than one. Note that the above items are always in counts (not degrees or radians) within the real time controller and since the DAC's are 16-bit wide, + 32767 counts corresponds to +9.999 volts, and -32768 counts corresponds to -10 volts.

2.1.9.2 The **Jog Position** option enables the user to move the mechanism to a different commanded position. In contrast to displacements executed under the **Trajectory** dialog box, during a **Jog** command no data is acquired for plotting purposes. Since this motion is effected via the current controller, one can only jog under closed loop control with a stable controller. By selecting the appropriate radio button either incremental and absolute displacements may be carried out. The jogging feature allows the user to return to a known position after the execution of the various forms of open and closed loop trajectories.

2.1.9.3 The **Zero Position** option enables the user to reinitialize the current position as the zero position. Note that if following errors exists, then the actual positions may be other than zero even though the commanded position is at zero (since the action is similar to commanding an instantaneous zero set point, a sudden small jerk in position may occur).

2.1.9.4 The **Reset Controller** option allows the user to reset the real-time controller. Upon Power up and after a reset activity, the loop is closed with zero gains and there it behaves in the same way as in the open loop state with zero control effort. Thus the user should be aware that even though the **Control Loop Status** indicates "closed loop", all of the gains are zeroed after a

Reset. In order to implement (or re implement) a controller you must go to the **Setup Control Algorithm** box.

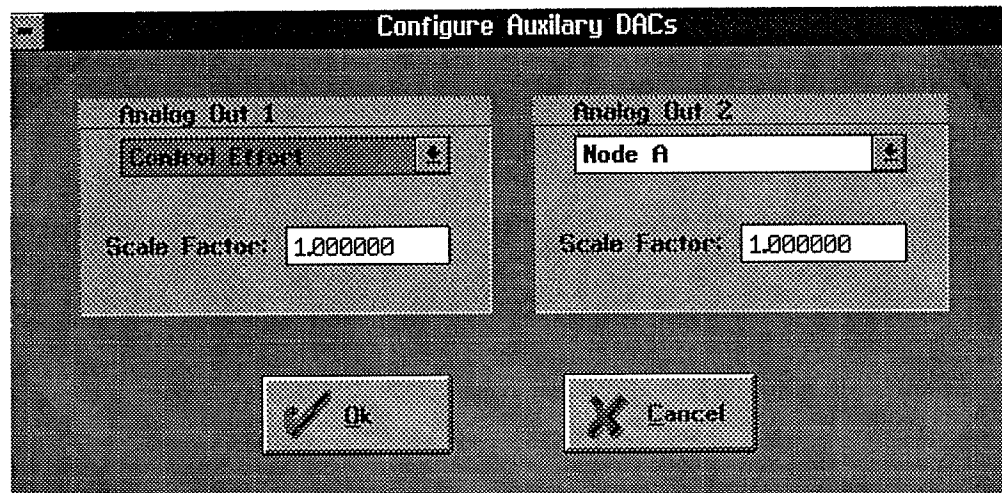


Figure 2.1-10. The Configure Auxiliary DACs Dialog Box

2.1.9.5 The **Rephase Motor** option enables the users of other ECP mechanisms to rephase their brushless motor commutation phase angle. This feature is not used in Model 505 with its DC brush motors.

2.1.9.6 The **Download Controller Personality File** is an option which should not be used by most users. In a case where the real-time controller irrecoverably malfunctions, and after consulting ECP, a user may download the personality file if a ".pmc" file exists. In the case of Model 505, this file is named "m505.pmc". Note that this downloading process takes a few seconds. If the Controller Box is powered during this down load process, this motor phasing will be effective.

## 2.2 Electromechanical Plant

### 2.2.1 Design Description

The plant, shown in Figure 2.2-1, consists of a pendulum rod which supports the sliding balance rod (the figure also serves to define the hardware terminology used throughout this manual). The balance rod is driven via a belt and pulley which in turn is driven by a drive shaft connected to a dc servo motor below the pendulum rod.

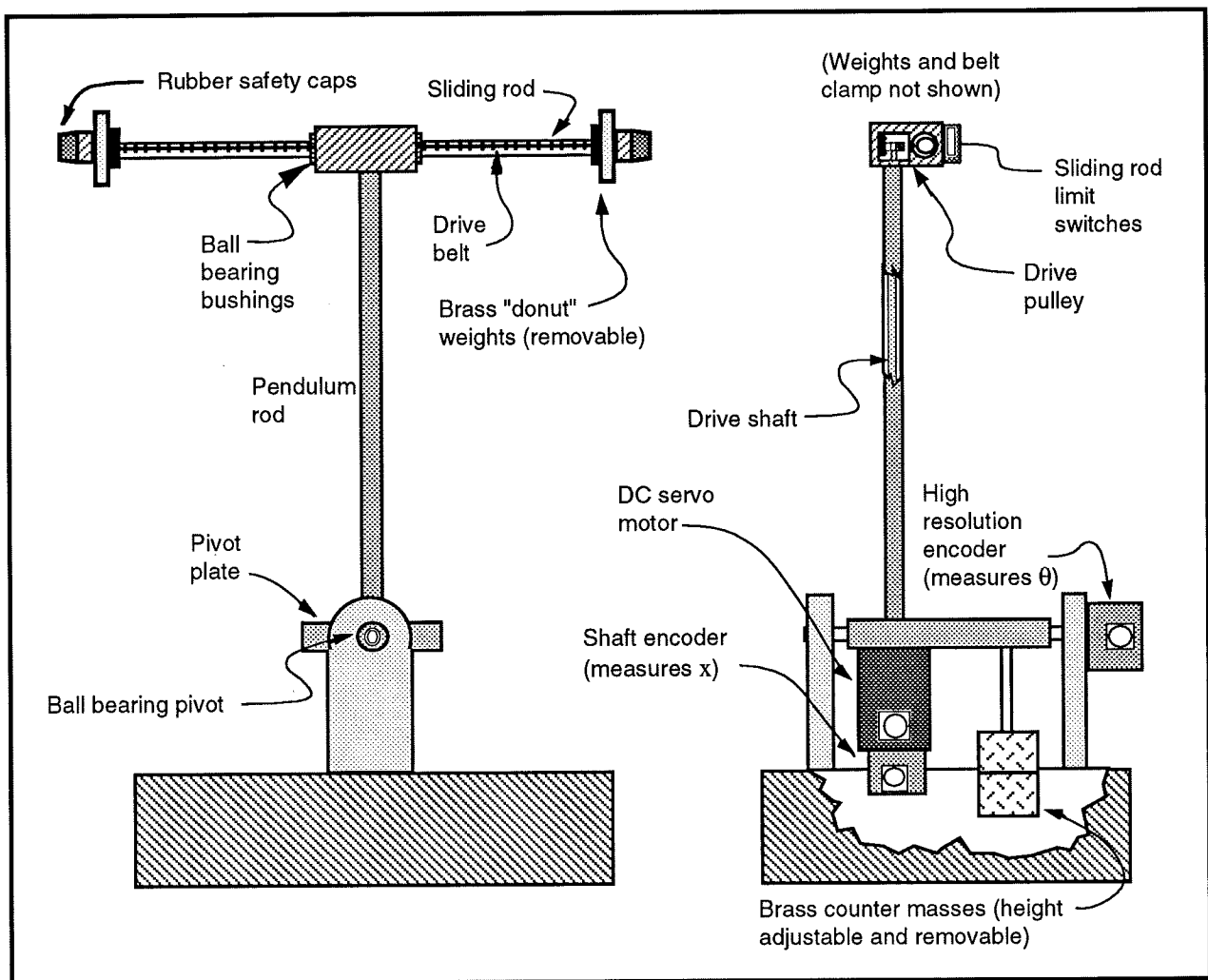


Figure 2.2-1. Inverted Pendulum Apparatus

Thus by steering the sliding rod in the presence of gravity the pendulum rod angle is controlled.<sup>1</sup> The weight at the bottom may be adjusted to alter the center of gravity of the pendulum rod and (as a result) alter the system dynamics. An encoder position at the back of the motor senses the position and (derived) velocity of the sliding rod. Another encoder connected to the pivoting base of the pendulum rod is used to sense its angular position and velocity.

### 2.2.2 Changing The Drive Belt

In cases of heavy use or when the sliding rod contacts its travel limit under high torque, the teeth on the toothed belt that drives the sliding rod may become damaged. Often, the damage is of no consequence since it is beyond the normal operating region of the sliding rod. If the damage becomes extensive though (say more than two adjacent teeth, or extending into the more central operating region), the belt should be replaced.

Referring to Figure 2.2-2, the replacement procedure is as follows:

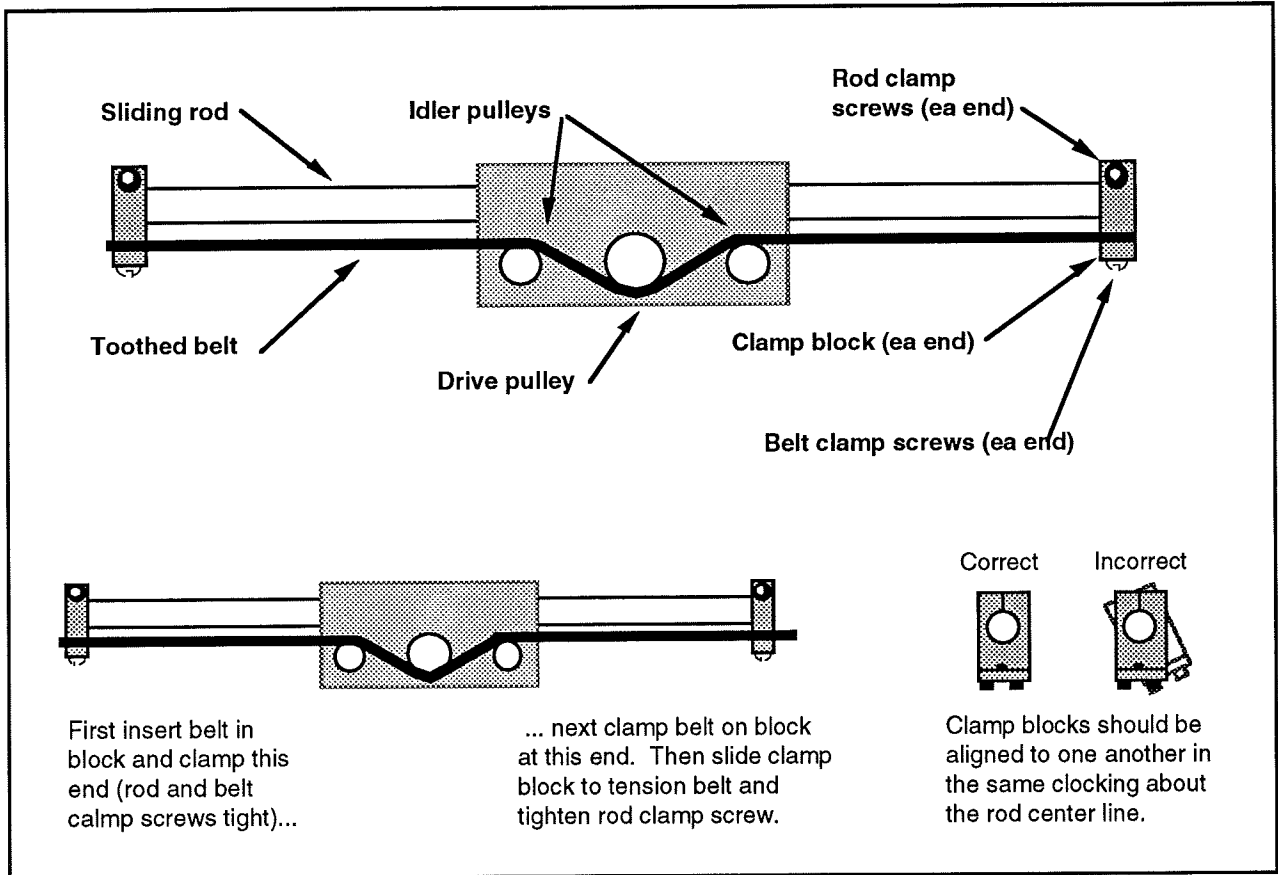
1. Disconnect the drive power cable from the servo amplifier box (this is both a safety precaution and removes the back EMF load from the motor)
2. With the sliding rod horizontal, use a gram force gauge to measure the friction in moving the sliding rod back and forth in its center ( $\pm 5$  cm) of travel. (This can be done by feel alone if a force gauge is not available) The reading will vary, but should generally fall in the range of 40 to 70 gr.
3. Before removing the worn belt, memorize or record how it is routed through the clamp blocks and idler and drive pulleys. Remove the worn belt by loosening the clamp screws on either end.
4. Feed the new belt through the pulley system and clamp one end via the clamping block as shown in the lower portion of Figure 2.2-2. Be sure to firmly torque the belt clamp screws.
5. Clamp the opposite end in its clamp block such that the belt is slightly tensioned through the pulley system and with 1-2 mm of clearance between the outer face of the clamp block and the end of the sliding rod. Firmly torque the belt clamp screws. Loosen the rod clamp screw on the same end and slide the block back and forth to adjust the belt tension as described in the following steps. Temporarily retighten the rod clamp screw when not adjusting tension.
6. Slide the balance rod back and forth about ten times to run it in. Measure running friction as in step 2. Readjust belt tension if necessary to achieve a similar friction force. The pendulum will generally run better with less tension (hence friction) than more; however there must be enough tension so that the belt does not sag, disengage the pulley system, or allow the sliding rod to rotate excessively. The two clamp blocks should have the same angular position about the rod center-line as described in the figure.

---

<sup>1</sup>In a static sense, when under closed loop control, the sliding rod is moved to maintain the pivoting system (or inverted pendulum) cg. directly above the pivot.

7. Once the tension is satisfactorily set, the any excess belt length beyond the clamps should be cut using wire cutters (note that the belt cable is hardened steel).
8. Check that the four belt clamp screws and two rod clamp screws are firmly torqued before operating system.

ECP supplies a spare belt with each inverted pendulum apparatus. Additional replacement belts are available by contacting ECP.



**Figure 2.2-2. Changing Sliding Rod Drive Belt**  
(see also instructions, this section)

### 2.2.3 Changing Or Moving Brass Weights

For safety, the following instructions must be followed faithfully whenever the weight position is changed or a weight is removed.

- a) Removing Or Replacing The Sliding Rod "Donut" Weights
  - 1) Removal or replacement is via the retaining screws (with rubber safety cap) at either end of the sliding rod. Make certain that the screw and safety cap are securely

torqued after making any changes and before operating the system. This applies whether the weights are attached or not.

- 2) Both donut weights must either be attached at opposite ends of the sliding rod or neither weight attached. Do not attach a single weight.
- b) Removing, Replacing or Changing Position Of Balance Weights
- 1) First unlock the weights by counter-rotating them relative to each other. Then move them to the new desired location and relock them by firmly counter-rotating in the opposite direction.
  - 2) The bottom weight must have at least three threads of full engagement.
  - 3) Two, one or zero weights may be used. If a single weight is used, lock it in place using a UNF 3/8" nut counter-rotated against the weight.

## 2.3 Safety

The following are safety features of the system and cautions regarding its operation. This section must be read and understood by all users prior to operating the system. If any material in this section is not clear to the reader, contact ECP for clarification before operating system.

**Important Notice: In the event of an emergency, control effort may be immediately discontinued by pressing the red "OFF" button on front of the control box.**

### 2.3.1 Hardware

A relay circuit is installed within the Control Box which automatically turns off power to the Box whenever the real-time Controller (within the PC) is turned on or off. Thus for the PC bus version of the real-time Controller the user should turn on the computer prior to pressing on the black ON switch. This feature prevents uncontrolled motor response during the transient power on/off periods. The power to the Control Box may be turned off at any time by pressing the red OFF switch.

Although not recommended, it will not damage the hardware to apply power to the Control Box even when the PC is turned off. However, doing so does not result in motor activation as the



motor's current amplifier will be disabled. The *amplifier enable* signal input to the Control Box is connected to the real-time Controller via the 60-pin flat ribbon cable. This input operates as a normally closed mode. When power to the real-time Controller is off, this input indicates open condition which in turn disables the motor amplifier.

The recommended **start up** sequence is as follows:

**First :** Turn on the PC with the real-time Controller installed in it.

**Second:** Turn on the power to Control Box (press on the black switch).

The recommended **shut down** is:

**First:** Turn off the power to the Control Box.

**Second:** Turn off the PC..

**FUSES:** There are two 3.0A 120V slow blow fuses within the Control Box. One of them is housed at the back of the Control Box next to the power cord plug. The second one is inside the box next to the large blue colored capacitor.

### 2.3.2 Software

The **Limit Exceeded** indicator of the **Controller Status** display indicates either one or both of the following conditions have occurred:

Over-travel of the sliding rod

Over-speed of the motor.

The real-time Controller continuously monitors the above limiting conditions in its background routine (intervals of time in-between higher priority tasks). In either limit case the real-time Controller opens up the control loop with a zero torque command sent to the actuator. The **Limit Exceeded** indicator stays on until a new set of (stabilizing) control gains are downloaded to the real-time Controller via the **Implement Algorithm** button of the **Setup Control Algorithm** dialog box, or a new trajectory is executed via the **Command** menu. Obviously the new trajectory must have parameters that do not cause the **Limit Exceeded** condition.

Also included is a *watch-dog timer*. This feature provides a fail-safe shutdown to guard against software malfunction and under-voltage condition. The use of the watch-dog timer is transparent

to the user. This shutdown condition turns on the red LED on the real-time Controller card. You may need to cycle the power to the PC in order to reinitialize the real-time Controller should a watch-dog timer shutdown occur.

### 2.3.3 Safety Checking The Controller

While it should generally be avoided, in some cases it is instructive or useful to manually contact the mechanism when a controller is active. This should always be done with caution and never in such a way that clothing or hair may be caught in the apparatus. By staying clear of the mechanism when it is moving or when a trajectory has been commanded, the risk of injury is greatly reduced. Being motionless, however, is not sufficient to assure the system is safe to contact. In some cases an unstable controller may have been implemented but the system may remain motionless until perturbed – then it could react violently.

In order to eliminate the risk of injury in such an event, you should always safety check the controller prior to physically contacting the system. This is done by lightly grasping a slender, light object with no sharp edges (e.g. a ruler without sharp edges or an unsharpened pencil) and using it to slowly move the pendulum rod (See Fig. 2.2-1) from side to side. Keep hands clear of the mechanism while doing this and apply only light force to the pendulum rod. If the rod does not react violently (a safe controller will cause the system to *regulate* sending the sliding rod in a direction to counteract the disturbance) then it may be manually contacted – but with caution. This procedure must be repeated whenever any user interaction with the system occurs (either via the Executive Program or the Controller Box) if the mechanism is to be physically contacted again.

### 2.3.4 Warnings

**WARNING #1: Stay clear of and do not touch any part of the mechanism while it is moving, while a trajectory has been commanded, or before the active controller has been safety checked – see Section 2.3.3.**

**WARNING #2: The following apply at all times except when motor drive power is disconnected (consult ECP if uncertain as to how to disconnect drive power):**

- a) Stay clear of the mechanism while wearing loose clothing (e.g. ties, scarves and loose sleeves) and when hair is not kept close to the head.
- b) Keep head and face – especially eyes – well clear of the mechanism.

**WARNING #3:** Verify that balance and sliding rod masses are secured per Section 2.2.3 of this manual prior to powering up the Control Box.

**WARNING #4:** Do not take the cover off or physically touch the interior of the Control Box unless its power cord is unplugged (first press the "Off" button on the front panel) and the PC is unpowered or disconnected.

**WARNING #5:** The power cord must be removed from the box prior to the replacement of any fuses.

### 3. Start-up & Self-guided Demonstration

This chapter provides an orientation "tour" of the system for the first time user. In Section 3.1 certain hardware verification steps are carried out. In Section 3.2 a self-guided demonstration is provided to quickly orient the user with key system operations and Executive program functions. Finally, in Section 3.3, certain system behaviors which may be nonintuitive to a first time user are pointed out .

#### 3.1 Hardware Setup Verification

At this stage it is assumed that

- a) The ECP Executive program has been successfully installed on the PC's hard disk (see Section 2.1.2).
- b) The actual DSP printed circuit board (the real-time Controller) has been correctly inserted into an empty slot of the PC's extension (ISA) bus (this applies to the PC bus version only).
- c) The supplied 60-pin flat cable is connected between the J11 connector (the 60-pin connector) of the real-time Controller and the JMACH connector of the Control Box<sup>1</sup>.
- d) The other two supplied cables are connected between the Control Box and the Inverted Pendulum apparatus;
- e) The Inverted Pendulum apparatus has the adjustable weight at the default height shipped from the factory. (i.e. Plant #2 from Section 6.1)
- f) You have read the safety section 2.3. All users must read and understand that section before proceeding.

Please check the cables again for proper connections.

#### 3.1 Hardware Verification (For PC-bus Installation)

**Step 1:** Switch off power to both the PC and the Control Box.

---

<sup>1</sup>This applies to the PC-bus installation only. For the controller in the Control Box, the RS-232 cable must be connected between the Control Box and the PC.

- Step 2:** With power still switched off to the Control Box, switch the PC power on. Enter the ECP program by typing ">ECP" in the appropriate directory. You should see the Background Screen (see Section 2.1.3) Gently rotate the pendulum rod and later the top sliding rod by hand. You should observe some following errors and changes in encoder counts.
- Step 3:** If the ECP program cannot find the real-time Controller, try the **Communication** dialog box under the **Setup** menu. Select PC-bus at address 528, and click on the test button. If the real-time Controller is still not found then switch off power to your PC and then take its cover off. With the cover removed check again for the proper insertion of the Controller card. Switch the power on again and observe the two LED lights on the Controller card. If the green LED comes on all is well, if the red LED comes on you should contact ECP for further instructions. If the green LED comes on, turn off power to your PC, replace the cover and turn the power back on again. Now go back to the ECP program and you should see the positions change as you gently move the pendulum rod.
- Step 4:** Make sure that you can rotate the pendulum rod and move the sliding rod freely. Now connect the power cord to the Control Box and press the black "ON" button to turn on the power to the Control Box. You should notice the green power indicator LED lit, but the motor should remain in a disabled state. Do not touch the apparatus any more, remember whenever power is applied to the Control Box there is a potential for uncontrolled motion of the pendulum (see the warnings in Section 2.3).
- Step 5:** Now turn off the power to your PC.

This completes the hardware verification procedure. Please refer again to Section 2.3 for future start up and shut down procedures.

### 3.2 Demonstration Of The Apparatus And ECP Executive Program

This section walks the user through the salient functions of the system. By following the instructions below you will actually implement a controller, maneuver the system through various trajectories, and acquire and plot data.

**Step 1:** With the power to the Control box turned off, enter the ECP Executive program. You should see the Background Display. Turn on power to the Control Box (press on the black button). Now enter the **File** menu, choose the **Load Setting** option and select the file `default.cfg`. This configuration file is supplied on the distribution diskette and should have been copied into the ECP directory by now. Note that this file would have been loaded into the Executive automatically (see Section 2.1.4.1). This particular `default.cfg` file contains the controller gain parameters and other trajectory, data gathering and plotting parameters specifically saved for the activities within this section.

This file has been created to operate with a plant which has the adjustable weight at the factory default setting. It could generate an unstable response if the position of the adjustable weight on the lead screw is significantly altered.

**Step 2:** Enter the **Utility** menu and **Reset** the controller. Place the sliding rod in its approximate center of travel. With aid of an object such as a ruler, balance the pendulum rod so that it remains vertical (see Fig. 3.2-1). If the pendulum remains balanced without support, you may proceed to the next step without holding it up. Otherwise you may need to manually keep it vertical with the ruler (so that you can stay clear of the mechanism). **Be sure to keep your head clearly away from the mechanism and out of the plane of action of the sliding rod.**

**Step 3:** Now enter the **Setup** menu and choose, **Setup Control Algorithm** option. You should see the sampling time  $T_s = 0.00884$  seconds, **Discrete time control**, and the **General Form** button selected. Now select **Setup Algorithm** and you should see the following gains:  $T_0=2.5$ ,  $S_0=24.61$ ,  $S_1=-24.00$ ,  $H_0=1.0$ ,  $I_0=1.8427$ ,  $I_1=-1.76$ ,  $E_0=0.35$ , and  $G_1=-0.5$ . This controller was designed using the linear quadratic regulator (LQR) technique with all error weighting on encoder #1 (the pendulum angle encoder). In addition, a low pass filter was included ( $G_1$  coefficient) to reduce quantization noise.

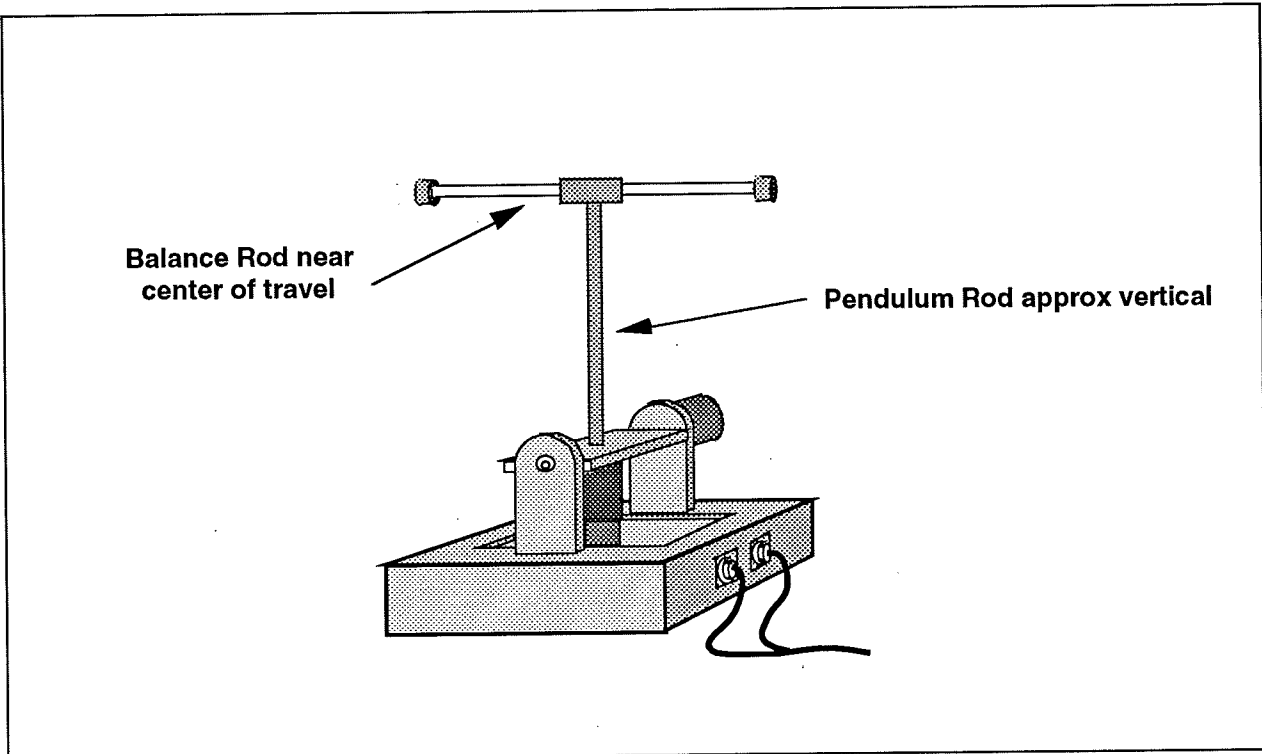


Figure 3.2-1 Equilibrium Position For Beginning Demo

Exit this dialog box and, with the pendulum in the vertical position and the sliding rod in its middle "T" position, select **Implement Algorithm**. The control law is now downloaded to the real-time Controller. Very gently push the pendulum rod with the ruler to verify that control action is in effect (see Sect. 2.3.3, "Safety Checking The Controller") If you do not notice any motor power, use the ruler to move it back to the vertical position and then, click on the **Implement Algorithm** button again until you notice the servo loop closed. Now select **Upload Algorithm**, then **General Form** and **Setup Algorithm**. You should see the same coefficients uploaded back from the real-time controller to the Executive program. Note that the general form algorithm is the only structure that actually runs in real-time. All other structures (e.g. PID etc.) are translated to the general form by the Executive prior to implementation.

**Step 4:** Enter the **Data** menu and select **Setup Data Acquisition**. In this box make sure that the following four items are selected: Commanded Position and Encoders 1,2, & Control effort. *Data sample period* should be 5 which means that data will be collected every fifth servo cycle (in this case every  $5 \times 0.00884$  seconds). Now exit this menu.

**Step 5:** Enter the **Command** menu and select **Trajectory**. In this box select **Step** and then **Setup**. You should see *step size* = 300 counts, *dwell time* = 4000 ms and *no. of*

*repetitions* = 1. If not, change the values to this parameter set. Exit this box and go to the **Command** menu. This time select **Execute** and with the **Sample Data** box checked run the trajectory. You should have noticed a step move of 300 counts, a dwell of 4 second and a return step move. Wait for the data to be uploaded from the real-time Controller to the Executive program running on the PC. Now enter the **Plotting** menu and choose **Setup Plot**. Select Encoder 1 and Commanded Position for plotting (left axis). Return to the **Plotting** menu and select **Plot Data**.

You should see a plot similar to the one shown in Figure 3.2-2. Replace Encoder 1 with 2 via the **Setup Plot** box and then plot data. A similar plot to the one shown in Figures 3.2-3 should be seen. Notice the initial reverse motion which is a characteristic of a non-minimum phase plant. This phenomenon is explained from basic laws of dynamics.

You may re-position the plots by dragging the upper plot frame bar. For the second plot try shrinking it by clicking on the down arrow on the top right hand corner. A minimized plot may be expanded at any time by double-clicking on its icon. Clicking on the top right hand upward arrow expands the plot to full size. (It must be brought back to normal size however prior to shrinking.) You can re-size the plots using the cursor to "drag" the lower right hand corner of the plot frame (position cursor at corner and begin "dragging" when cursor becomes double arrow).

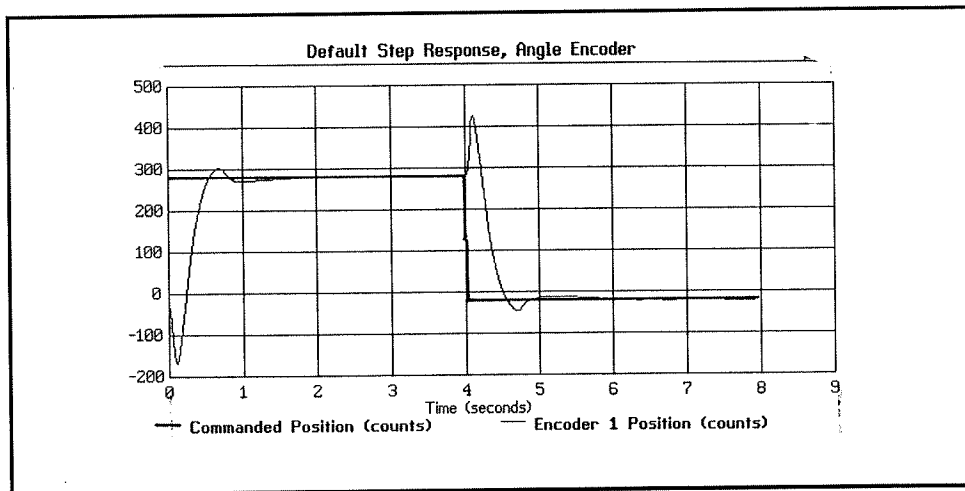


Figure 3.2-2 Step Response at Angle Encoder (#1)



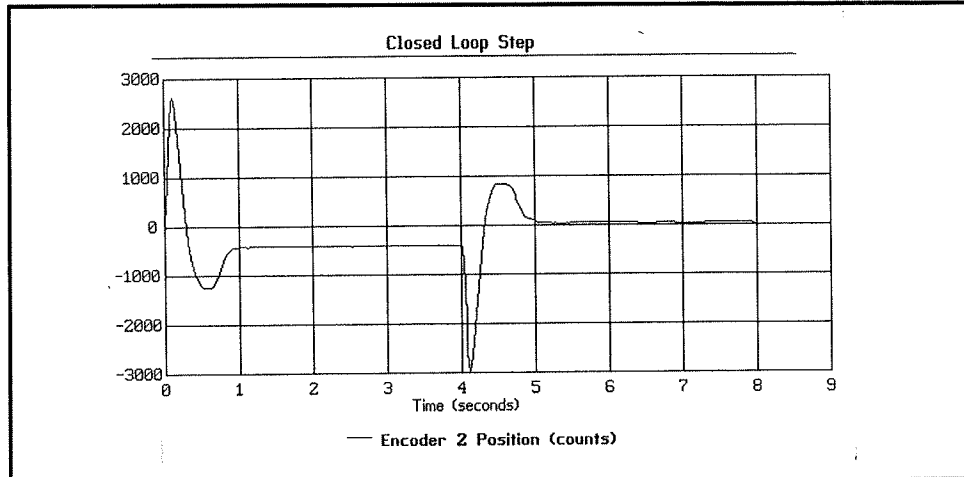


Figure 3.2-3 Step Response at Sliding Rod Encoder (#2)

**Step 6:** Enter the **Command** menu and select **Trajectory**. In this box select **Parabolic** and then **Setup** to enter the **Ramp** dialog box. You should see *distance* = 300 counts, *velocity* = 1000 counts/s, *dwell time* = 4000 ms and *no. of repetitions* = 1. If not, change the values to this set. Exit this box and go to the **Command** menu. Now select **Execute** and with **Sample Data** checked, run the trajectory and wait for the data to be uploaded from the real-time Controller. You should have noticed the ramp move (constant acceleration to top speed) of 300 counts followed by a dwell of 4 seconds and then the reverse motion.. Now enter the **Plotting** menu and choose **Setup Plot**. Select **Encoder 1** and **Commanded Position** for plotting and then plot the data. You should see a plot similar to the one shown in Figure 3.2-4. Note that the initial reverse motion is smaller in amplitude. You may save the ramp or the step response under *anyname.plt* using the **Save Plot Data** option. Any plot data thus saved may be reloaded from the disk using the **Load Plot Data** option for future inspection, plotting or printing. To print simply choose the **Print Data** menu option and select the appropriate printer before the printing command. Alternatively, any set of collected data may be exported as an ASCII text file by way of the **Export Data** option of the **Data** menu.

Now try **Parabolic** and **Cubic** trajectories all default amplitudes should be 300 counts. Compare commanded and actual velocities of encoder 1 under these trajectories. Many conclusions can be drawn regarding the nature of the actual motion based upon the commanded trajectory profiles. Now close all plot windows.

If the *sinusoidal trajectory* is selected, the default values of amplitude and frequency should be 200 counts and 0.5 Hz respectively with 10 repetitions selected. Executing this and plotting the response should yield a graph similar to the one shown in Figure 3.2-5. Note the amplitude attenuation and the phase lag.

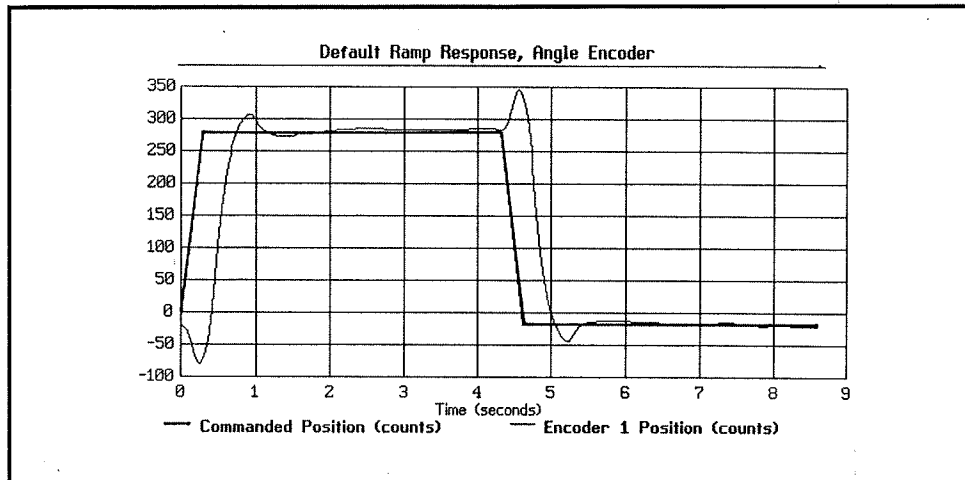


Figure 3.2-4 Ramp Response at Encoder 1

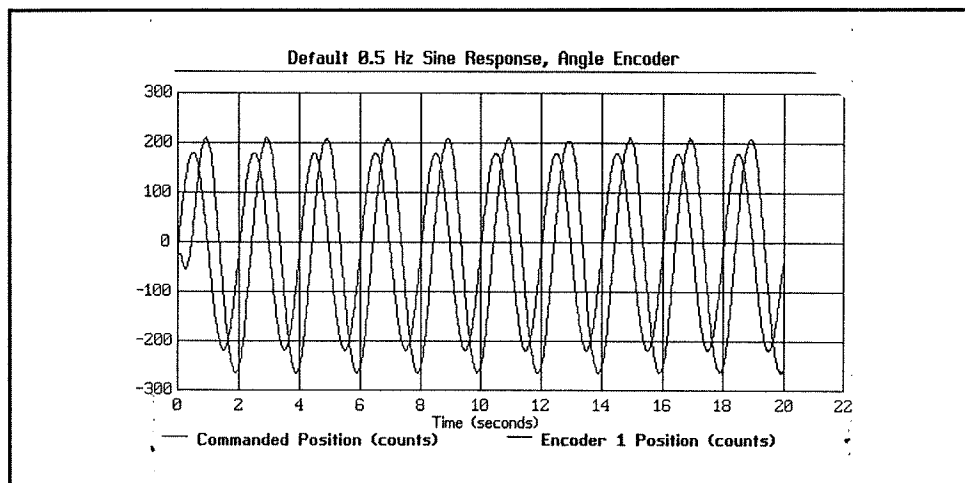


Figure 3.2-5 Sinusoidal Response at Encoder 1 (0.5 Hz)

**Step 6:** Again enter **Trajectory** and select **Sine Sweep** then **Setup**. You should see the *amplitude = 175 counts, max. freq.= 6 Hz, min. freq.= 0.1 Hz* and the *sweep time = 29.5 sec*. Again, if different, change the values to correspond this set. While running this trajectory, you should notice sinusoidal motion with increasing frequency for one minute. Now enter the **Plotting** menu and choose **Setup Plot**. This time select only Encoder 1 for plotting and then plot the data. You should see a plot similar to the one shown in Figure 3.2-6.

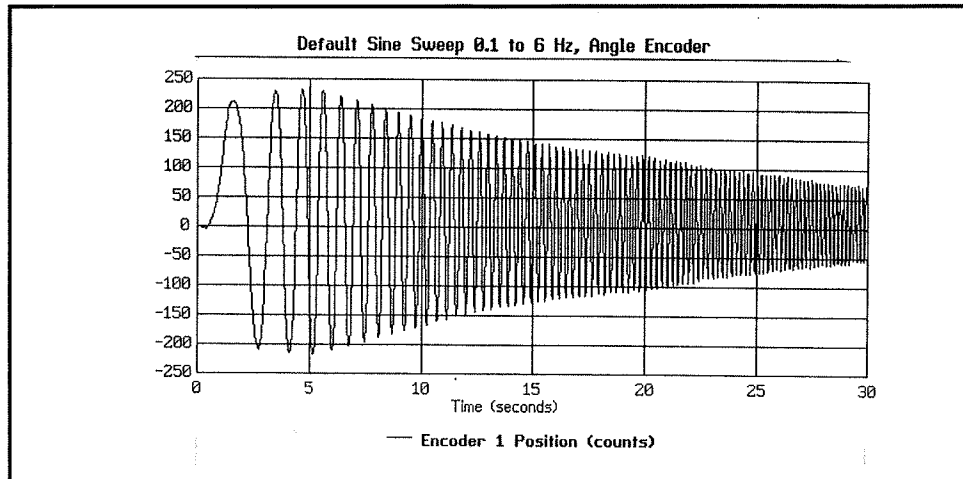


Figure 3.2-6 Sine Sweep Response at Encoder 1 (Pendulum Angle Encoder)

### 3.3 Nonintuitive System Behavior

An explanation of system functionality which may be nonintuitive to first time (or first few time) users is given below.

- Point 1:** Upon PC power up, the control loop is closed but all gains are cleared to zero. As a result, even though the status display indicates **CLOSED LOOP**, the control effort is zero indicating that there is no force demand from the actuator. With zero gains, the closed loop system behaves like an open loop one. This initialization of the gains to zero also takes place every time the real-time controller is **Reset** under the **Utility** menu.
- Point 2** Even though the parameters for a particular control algorithm may exist within the Executive, the real-time Controller does not carry them until the **Implement Algorithm** button of the **Setup Control Algorithm** dialog box is activated. Then the control is implemented immediately
- Point 3** The **Limit Exceeded** message comes on as a result of either motor over-speed or sliding rod over travel. In either case the message does not clear until a stable set of control gains are downloaded or a less abrupt trajectory is requested (see Section 2.1.3 ). Also if the control box is turned off, the **Limit Exceeded** message may be shown even when the sliding rod is not in contact with the limits.

## 4. Real-Time Control Implementation

This chapter is a description of the real-time controller functionality and a tutorial on related implementation issues.

The ECP real-time Controller is a Digital Signal Processor (DSP) based single board computer which executes the following real-time tasks:

- Servo loop closure
- Command generation
- Brush motor commutation (not applicable to Model 505)
- Sensor Interface

In addition, in the background (while not executing the above tasks) it carries out the following:

- User Interface via PC and/or RS232 Bus
- Limits and Safety Activities

### 4.1 Servo Loop Closure

Servo loop closure involves computing the control algorithm at the sampling time. The real-time Controller executes the General Form equation of the control law at each sample period  $T_s$ . This period can be as short as 0.000884 seconds (approx. 1.1 KHz) or any multiple of this number. The Executive program's **Setup Control Algorithm** dialog box allows the user to alter the sampling period. All forms of control laws are automatically translated by the Executive program to the General Form prior to downloading ("implementing") to the Controller. The General Form uses 96-bit real number (48-bit integer and 48-bit fractional) arithmetic for the computation of the control effort. The control effort is saturated in software at +/- 16384 to represent +/- 5 volts on the 16-bit DACs whose range is +/- 10 volts. The +/-5 volt limitation is due to the actuator's amplifier input voltage scaling.

Referring to Figure 4.1-1, the control equations are as follows:

$$R(q^{-1})\text{nodeA}(k) = T(q^{-1}) * c_p(k) - S(q^{-1}) * f_{b1}(k)$$

Here T, S, and R are seventh order polynomials of the unit time shift operator q, and the letter k represents the k<sup>th</sup> sampling period for k=0, 1,2, ... . The variable nodeA is an intermediate value

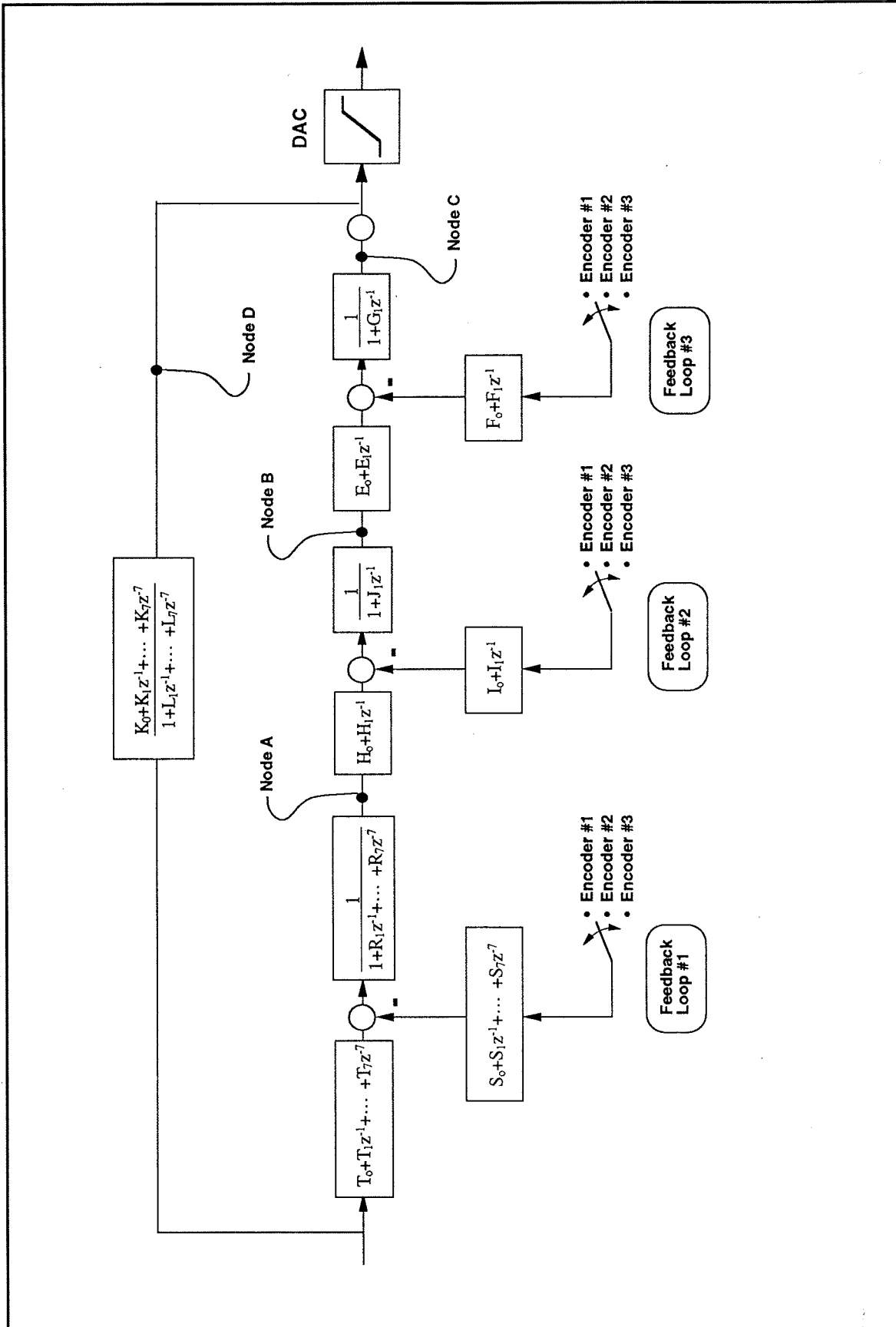


Figure 4.1-1. General Form Of The ECP Control Algorithm

of the overall control law equation which is stored in the memory and may be acquired by the user through the Data Acquisition feature of the Executive program. The variable  $c_p(k)$  is the commanded position which is generated by the real-time Controller as described in the next section. The variable  $f_{b1}(k)$  may be from any of the three possible feedback sensors depending on the state of software loop switch 1.

Next the intermediate loop is computed as follows

$$J(q^{-1}) * \text{nodeB}(k) = H(q^{-1}) * \text{nodeA}(k) - I(q^{-1}) * f_{b2}(k)$$

Here again nodeB is an intermediate value stored in the memory and  $f_{b2}$  is the sensor feedback selected via loop switch 2. J, H, and I are second order polynomials. For the inner loop we have

$$G(q^{-1}) * \text{nodeC}(k) = E(q^{-1}) * \text{nodeB}(k) - F(q^{-1}) * f_{b3}(k)$$

NodeC is the contribution to the value of the control effort generated by the overall regulator and  $f_{b3}$  is the sensor feedback selected via loop switch 3. G, E, and F are second order polynomials. For the feedforward loop we have

$$L(q^{-1}) * \text{nodeD}(k) = K(q^{-1}) * c_p(k)$$

In this case nodeD is the contribution to the value of the control effort generated by the feedforward terms, L and K are sixth order polynomials.

The combined regulatory and tracking controller generates the control effort as:

$$\text{control effort}(k) = \text{nodeC}(k) + \text{nodeD}(k)$$

The general control structure described above supports the implementation of a broad range of specific control forms.

## 4.2 Command Generation

*Command generation* is the real-time generation of motion trajectories specified by the user. The parameters of these trajectories are downloaded to the real-time Controller through the Executive program via the **Trajectory Configuration** dialog box. This section describes the trajectories generated in the current control version.

4.2.1 Step Move

Figure 4.2-1a shows a step move demand. The desired trajectory for such a move can be described by

$$\begin{aligned}
 c_p(t) &= c_p(0)+C && \text{for } t > 0 \\
 c_v(t) &= 0 && \text{for } t > 0 \\
 c_v(0) &= \infty
 \end{aligned}$$

Where  $c_p(t)$  and  $c_v(t)$  represent commanded position and velocity at time  $t$  respectively and  $C$  is the constant step amplitude. Such a move demand generates a strong impulsive torque from the control actuator. The response of a mechanical system connected to the actuator would depend on the dynamic characteristics of the controller and the system itself. However, in a step move, the instantaneous velocity and its derivatives are not directly controllable. Usually step moves are used only for test purposes; more gentle trajectories are nearly always used for practical maneuvers.

4.2.2 Ramp Move

A ramp demand is seen in Figure 4.2-1b. The trajectory can be described by

$$\begin{aligned}
 c_p(t) &= c_p(0)+V*t && \text{for } t > 0 \\
 c_v(t) &= V && \text{for } t > 0 \\
 c_a(0) &= \infty
 \end{aligned}$$

where  $c_a(0)$  represents commanded acceleration at time zero and  $V$  is a constant velocity. Relative to a step demand, a ramp demand is more gentle, however the acceleration is still impulsive. The commanded velocity is a known constant during the maneuver.

4.2.3 Parabolic Move

Figure 4.2-1c shows a parabolic move demand. Its trajectory can be expressed as:

$$\begin{aligned}
 c_p(t) &= c_p(0)+c_v(0)*t+1/2 A*t^2 && \text{for } t > 0 < 1/2 t_f \\
 c_v(t) &= c_v(0)+A*t && \text{for } t > 0 < 1/2 t_f \\
 c_a(t) &= A && \text{for } t > 0 < 1/2 t_f \\
 c_j(0) &= \infty
 \end{aligned}$$

where  $c_j(t)$  represents commanded jerk at time  $t$  and  $A$  is a constant acceleration, and  $t_f$  is the final destination time. Relative to a ramp demand, a parabolic demand is more gentle, however the rate of change of acceleration (jerk) is still impulsive. Note that the commanded acceleration is a known constant during the maneuver. The second half of a parabolic demand uses  $-A$  for deceleration.

4.2.4 Cubic Move

Figure 4.2-1d shows a cubic demand which can be described by

$$\begin{aligned}
 c_p(t) &= c_p(0) + c_v(0) * t + 1/2 c_a(0) * t^2 + 1/6 J * t^3 && \text{for } t > 0 < 1/4 t_f \\
 c_v(t) &= c_v(0) + c_a(0) * t + 1/2 J * t^2 && \text{for } t > 0 < 1/4 t_f \\
 c_a(t) &= c_a(0) + J * t && \text{for } t > 0 < 1/4 t_f \\
 c_j(0) &= J
 \end{aligned}$$

where  $J$  represents a constant jerk. Relative to all the above demands, a cubic demand is more gentle. The commanded acceleration is linearly changing during the three sections of the maneuver. The second half of a cubic demand uses  $-J$  and the third part uses  $J$  again for the jerk input.

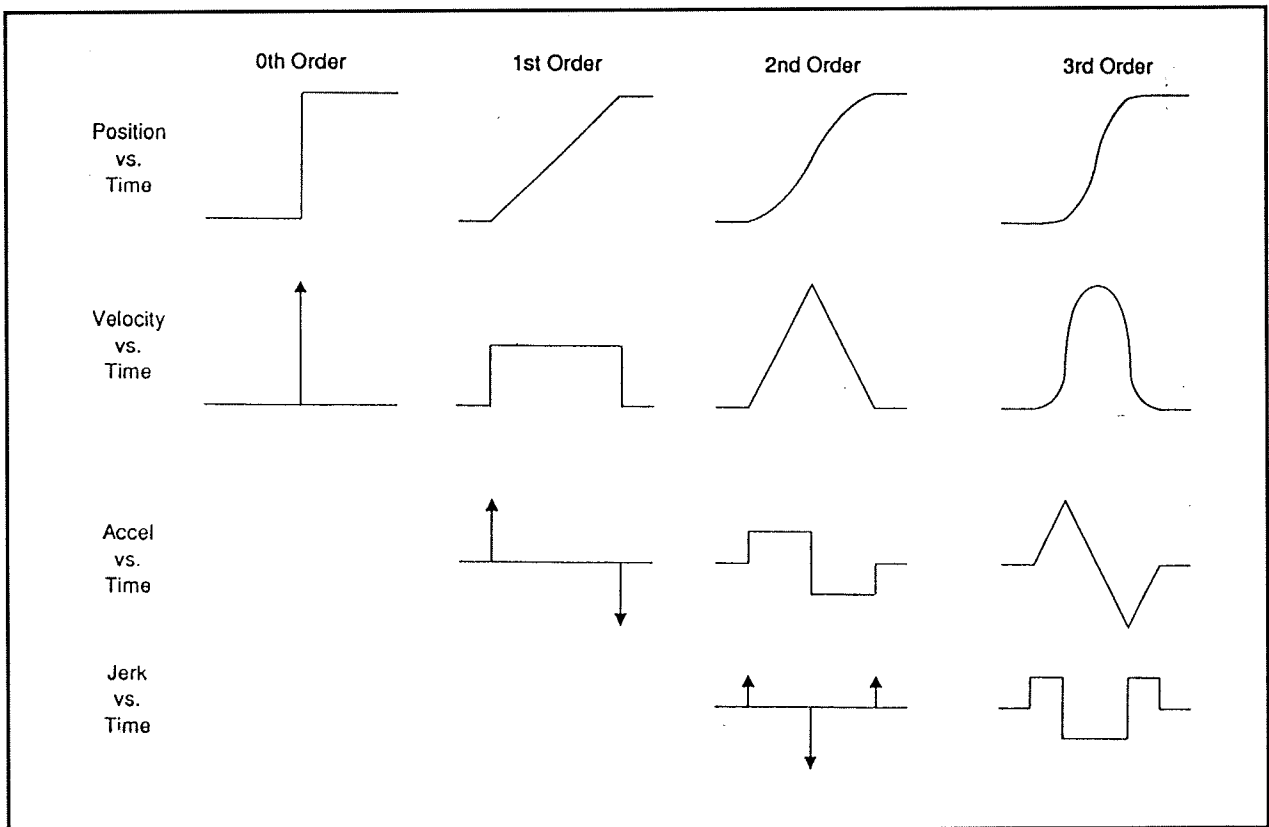


Figure 4.2-1. Geometric Command Trajectories Of Increasing Order



4.2.5 The Blended Move

Any time a ramp, a parabolic or a cubic trajectory move is demanded the real-time Controller executes a general blended move to produce the desired reference input to the control algorithm. The move is broken into five segments as shown in the velocity profile of Figure 4.2-2. For each section a cubic (in position) trajectory is planned. Five distinct cubic equations can describe the forward motion . After the dwell time, the reverse motion can be described by five more cubic trajectories. Each cubic has the form:

$$c_{p_i}(t) = c_{p_i}(0) + V_i * t + 1/2 A_i * t^2 + 1/6 J_i * t^3 \quad i = 1, \dots, 5$$

Using a known set of trajectory data (i.e. the requested total travel distance, acceleration time  $t_{acc}$ , and the maximum speed  $v_{max}$ , for each move), the constant coefficients  $V_i$ ,  $A_i$ , and  $J_i$  are determined for each segment of the move by the real-time Controller. This function is known as the "motion planning" task. Note that for a parabolic profile  $J_i=0$ , and for a ramp profile  $A_i$  is also zero which further simplifies the task. Having determined the coefficients for each section, the real-time Controller uses these values at the servo loop sampling periods to update the commanded position (reference input). For example if the segment is a cubic ( $J \neq 0$ ):

$$\begin{aligned} c_a(k) &= c_a(k-1) + J * T_s \\ c_v(k) &= c_v(k-1) + c_a(k) * T_s \\ c_p(k) &= c_p(k-1) + c_v(k) * T_s \end{aligned}$$

where  $T_s$  is the sampling period and  $c_a(k)$ ,  $c_v(k)$ ,  $c_p(k)$  represent commanded acceleration, velocity and position at the  $k^{th}$  sampling period.

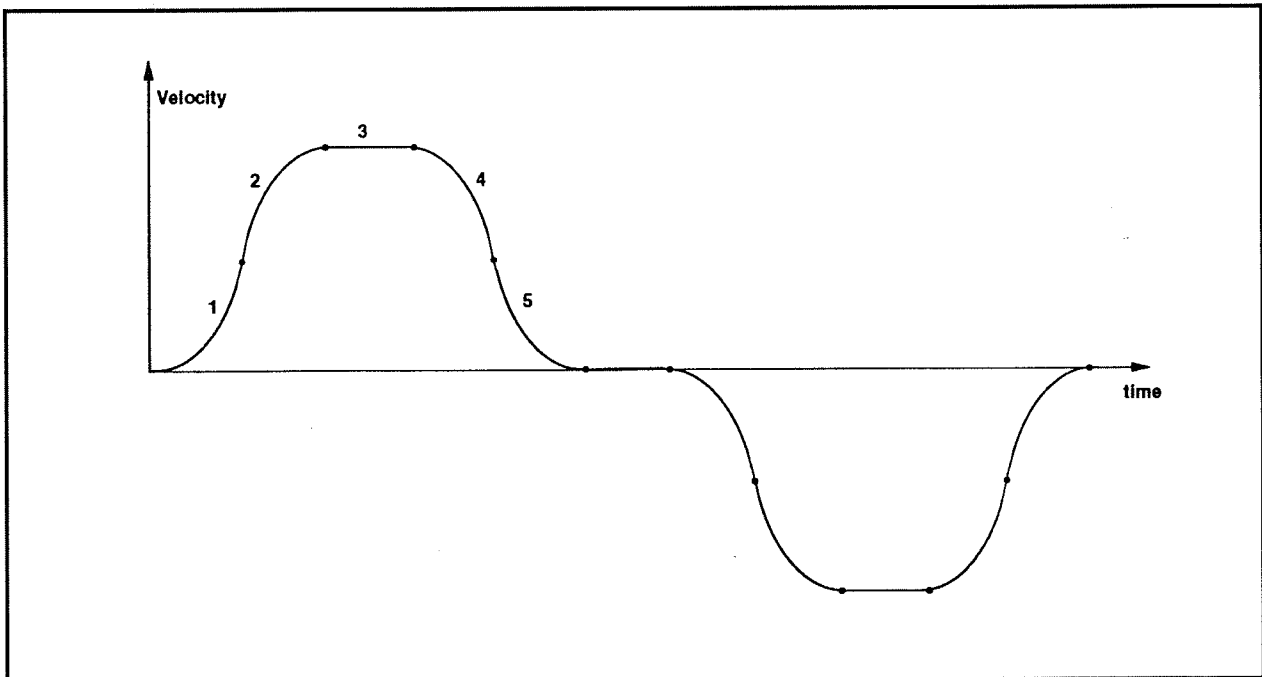


Figure 4.2-2 Velocity Profile for General Blended Move

**4.2.6 Sinusoidal Move**

The sinusoidal move is generated using the following equation:

$$c_p(k) = R * \sin(\theta(k))$$

where R is the amplitude,  $\theta(k) = \omega * k * T_p$  for  $k=0,1,\dots$ , and  $\omega$  is the commanded frequency in Hz.  $T_p$  is set to five milliseconds. To further smooth out the trajectory, a cubic spline is fitted between the points as follows:

$$c_p'(k) = (c_p(k-1) + 4 * c_p(k) + c_p(k+1)) / 6$$

For the linear sine sweep,  $\omega(k) = \alpha * T_p$ , where  $\alpha$  is a constant determined by the difference between the maximum and the minimum frequency divided by the sweep time

$$\alpha = (\omega_{max} - \omega_{min}) / \text{sweep time}$$

**4.3 Brush Type DC Servo Motor and Drive Amplifier**

The control effort at the  $k^{th}$  sampling period is the input to a 16-bit DAC which provides an analog signal for the motor amplifier. The amplifier operates in a transconductance mode providing a current (as opposed to voltage) demand to the motor which in turn represents a torque demand from the motor. To provide the current source capability an analog proportional plus integral (PI) controller is implemented within the amplifier for tracking the demanded current. Referring to the block diagram of Fig. 4.3-1, the transfer function between the motor current and the DAC output (control effort) is given by:

$$\frac{i(s)}{u(s)} = \frac{k_A k_c (k_p s + k_i)}{s(Ls + R + k_t k_b G(s)) + k_p s + k_i} \tag{4.3-1}$$

Here  $k_c$  is the DAC gain in volts/count (10 volts per 32767 counts),  $k_A$  is the amplifier forward gain which is dimensionless (V/V), R is the motor armature and brush resistance in ohms, L is the motor armature inductance in henrys,  $k_b$  is the motor back emf constant in v/(rad/s),  $k_t$  is the motor torque constant in Nm/ampere,  $k_m$  is the mechanical advantage constant which is the ratio of the force generated by the belt driving the sliding rod to the torque generated by the motor (N/Nm),

and  $G(s)$  is the transfer function between current and the motor velocity. In current mode drive amplifier, both the proportional gain  $k_p$  and the integral gain  $k_i$  of the amplifier are chosen to be very high relative the inner back emf loop within the practical range of motor operation. As a result, the effect of the inner loop may be ignored and the transfer function may be simplified to:

$$\frac{i(s)}{u(s)} = \frac{k_A k_c (k_p s + k_i)}{s(Ls + R) + k_p s + k_i} \quad (4.3-2)$$

At steady state this transfer function becomes:

$$\frac{i(s)}{u(s)} = k_A k_c \quad (4.3-3)$$

The force delivered to the sliding rod at steady state then becomes:

$$\frac{F(s)}{u(s)} = k_A k_c k_T k_m \quad (4.3-4)$$

In general, the analog PI controller gains of the amplifier are such that the dynamics of the current loop are much faster relative to the dynamics of the motor and mechanical plant. As a result the steady state value of current is achieved virtually instantaneously relative to changes in velocities and positions. Thus in this transconductance (current feedback) mode, the combined amplifier/motor combination can be thought of as a force generator with the characteristic of gain value  $k_f$ . The specific value of  $k_f$  for the Model 505 apparatus is given in Chapter 6 Experiment #1.

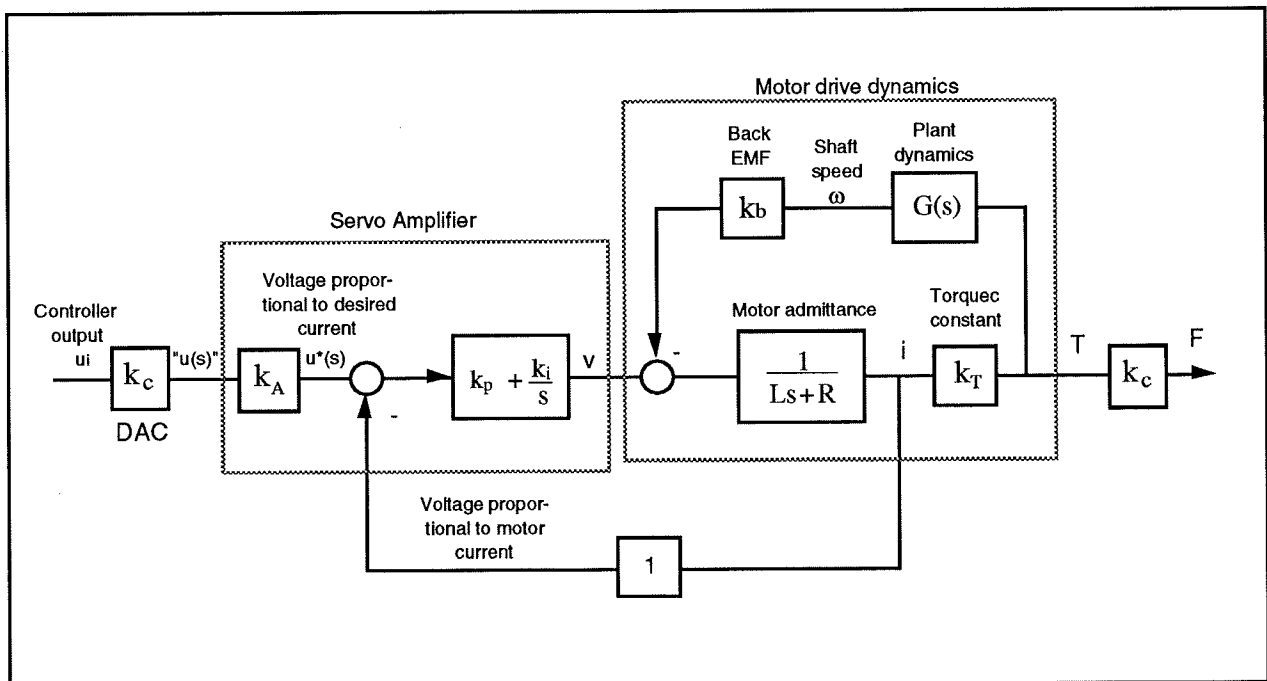


Figure 4.3-1. Mechanism Drive Block Diagram

**4.4 Multi-Tasking Environment**

Digital control implementation is intimately coupled with the hardware and software that supports it. Nowhere is this more apparent than in the architecture and timing used to support the various data processing routines. A well prioritized time multi-tasking scheme is essential to maximizing the performance attainable from the processing resources.

The priority scheme for the ECP real-time Controller's multi-tasking environment is tabulated in Table 4.4-1. The task with the highest priority is the commutation algorithm which takes place at the rate of 9.049 kHz (since the motor is not brushless, this algorithm is bypassed in the case of Model 505 system). The next highest priority task is the trajectory update and servo loop closure computation which takes place at the maximum rate of 1.131 KHz (minimum sampling period is 0.000884 seconds). In this case, the user may reduce the sampling rate through the Executive Program via changes to  $T_s$  in the **Setup Control Algorithm** dialog box.

The trajectory planning task has the third highest priority and is serviced at a maximum rate of 377 Hz. Here the parameters for a new trajectory need not be calculated every time this task is serviced by the real-time Controller. Whenever a new trajectory is required (i.e. the current trajectory is near its completion) this task is executed. The lower priority tasks are system house keeping routines including safety checks, interface and auxiliary analog output.

**Table 4.4-1 The Multi-Tasking Scheme of the Real-Time Controller**

<b>Priority</b>	<b>Task Description</b>	<b>Service Frequency</b>
1	Commutation (not Executed for Model 505)	9.049 KHz
2	Servo Loop Closure & Command Update	1.31 KHz
3	Trajectory Planning	377 Hz
4	Background Tasks including User Interface, Auxiliary DAC Update, Limit checks etc.	Background (In time between other tasks)

The higher priority tasks always prevail over lower ones in obtaining the computational power of the DSP. This multi-tasking scheme is realized by a real-time clock which generates processor interrupts.

## 4.5 Sensors

There are two incremental rotary shaft encoders used in the inverted pendulum. These are used to sense the position and velocity of the pendulum rod and the sliding rod. The pendulum angle encoder has a resolution of 4000 counts per revolution. The sliding rod encoder (connected to the motor) has a resolution of 500 counts per revolution which is equivalent to approximately 12500 counts per meter of the sliding rod travel. (the actual sliding rod travels approximately 0.24 meter).

The encoders are all optical type whose principle of operation is depicted in Figure 4.5-1. A low power light source is used to generate two 90 degrees out of phase sinusoidal signals on the detectors as the moving plate rotates with respect to the stationary plate. These signals are then squared up and amplified in order to generate quadrate logic level signals suitable for input to the programmable gate array on the real-time Controller. The gate array uses the A and B channel phasing to decode direction and detects the rising and falling edge of each to generate 4x resolution – see Figure 4.5-2. (Hence the effective encoder resolution is 16,000 counts/rev. and 50200 counts/m for the respective pendulum and balance rod measurements.) The pulses are accumulated continuously within 24-bit counters (hardware registers). The contents of the counters are read by the DSP once every servo (or commutation) cycle time and extended to 48-bit word length for high precision numerical processing.

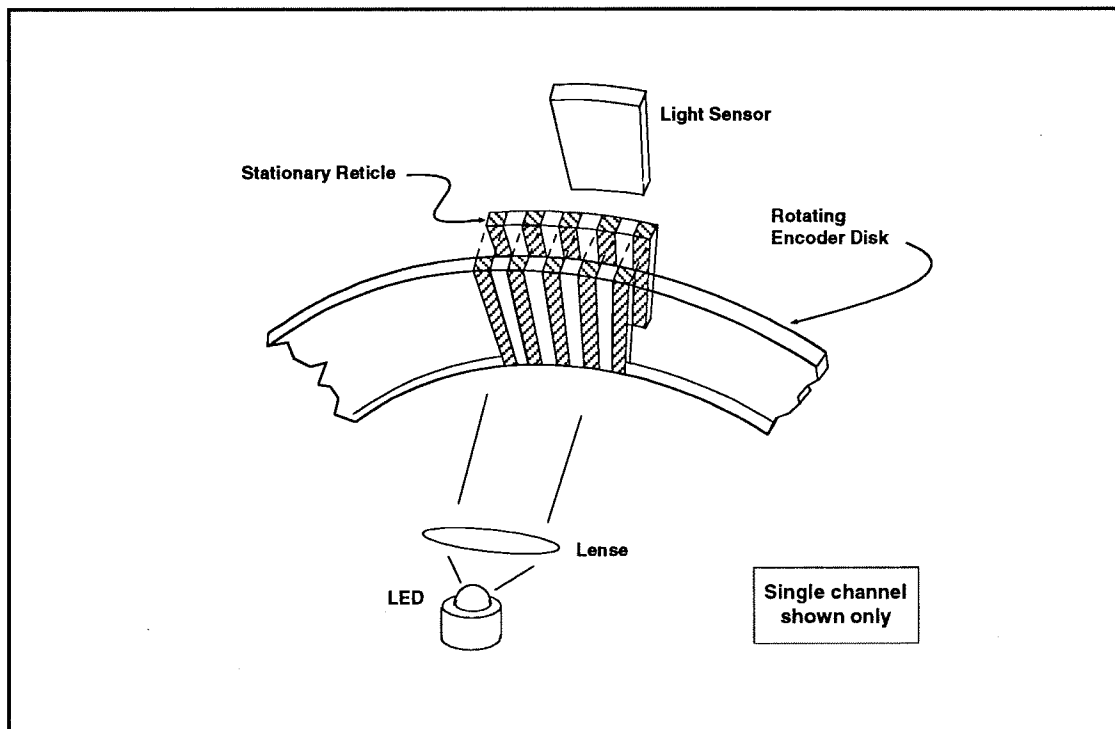


Figure 4.5-1 The Operating Principle of Optical Incremental Encoders

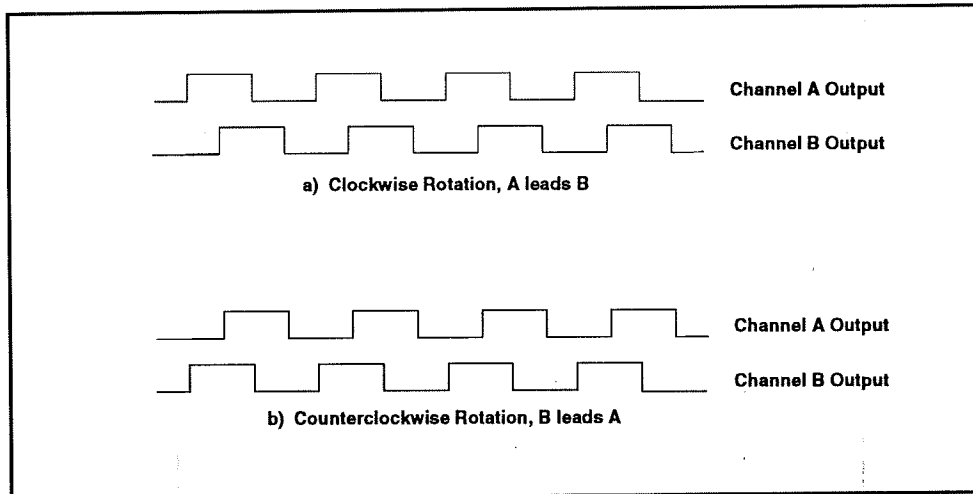


Figure 4.5-2. Optical Encoder Output

#### 4.6 Auxiliary Analog Output

There are two analog output channels in the Control Box which are connected to two 16-bit DACs which physically reside on the real-time Controller. Each analog output has the range of +/- 10 volts (-32768 to +32767 counts) with respect to the analog ground. The outputs on these DACs are updated by the real-time Controller as a low priority task. However, for most trajectories (e.g. for sine sweep up to approx. 25 Hz ), the update rate is sufficiently fast for an oscilloscope inspection of the Controller's internal signals. See the section on the Executive Program's **Utility** menu for the available signals to output on these DACs.

## 5. Plant Dynamic Models

This Chapter provides time domain expressions which are useful for nonlinear plant modeling and Laplace domain expressions useful for linear control implementation. These are used in the experiments described later in this manual. A discussion of generic control issues relating to this plant type is provided in Appendix B.

### 5.1 Equations of Motion

#### 5.1.1 Nonlinear Expressions

Neglecting friction effects, the plant may be modeled as shown in Fig. 5.1-1<sup>1</sup>. From the figure we have that the system kinetic energy,  $T(s)$  is:

$$T = \frac{1}{2}m_1v_{cg1}^2 + \frac{1}{2}J_1\dot{\theta}^2 + \frac{1}{2}m_2v_{cg2}^2 + \frac{1}{2}J_2\dot{\alpha}^2 \quad (5.1-1)$$

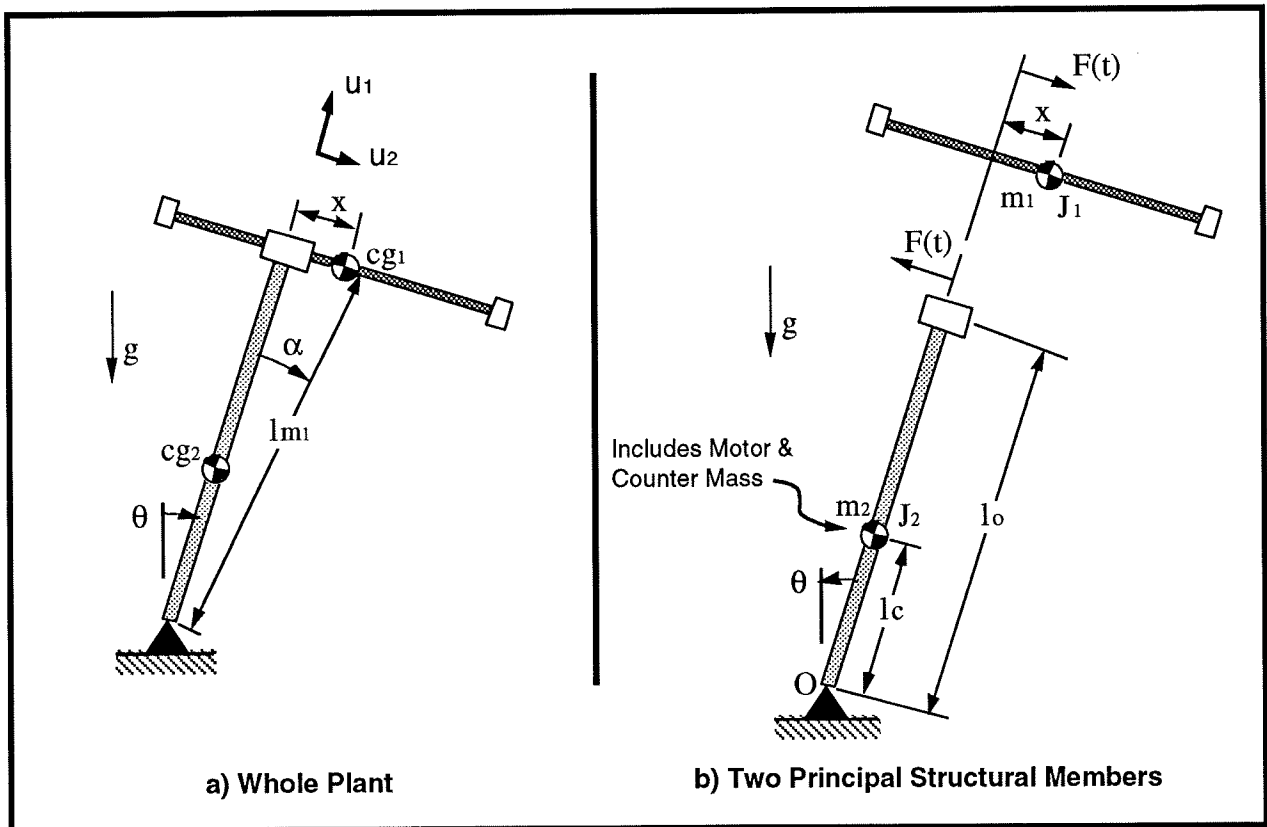


Figure 5.1-1 Plant Model Descriptions

<sup>1</sup>The force  $F(t)$  shown in the figure is equivalent to the motor torque divided by the drive pulley / belt contact radius.

where the  $v_{cg}$ 's are the inertial velocities of the respective members' centers of gravity (c.g.'s) (see Fig. 5.1-1b) and the  $J_i$ 's are the polar moments of inertia about the respective c.g.'s. The members are kinematically constrained so that:

$$v_{cg1} = \dot{x} + \dot{\theta} \times l_{m1} \quad (5.1-2)$$

$$v_{cg2} = l_c \dot{\theta} \quad (5.1-3)$$

where the underscored quantities are treated as vectors and "x" is the vector cross product. By resolving unto the  $\underline{u1}$  and  $\underline{u2}$  components and squaring, it is readily shown that:

$$v_{cg1}^2 = \dot{x}^2 + (l_{m1} \dot{\theta})^2 + 2(l_{m1} \dot{x} \dot{\theta} \cos \alpha) \quad (5.1-4)$$

Substituting Eq's 5.1-3,-4 into 5.1-1 and using:  $l_{m1}^2 = l_o^2 + x^2$  and  $\cos \alpha = l_o/l_{m1}$  yields:

$$\begin{aligned} T &= \frac{1}{2}(J_1 + J_2 + m_1(l_o^2 + x^2) + m_2 l_c^2) \dot{\theta}^2 + \frac{1}{2} m_1 \dot{x}^2 + m_1 l_o \dot{x} \dot{\theta} \\ &= \frac{1}{2} J_o(x) \dot{\theta}^2 + \frac{1}{2} m_1 \dot{x}^2 + m_1 l_o \dot{x} \dot{\theta} \end{aligned} \quad (5.1-5)$$

where

$$J_o(x) = J_1 + J_2 + m_1(l_o^2 + x^2) + m_2 l_c^2 \quad (5.1-6)$$

is the system moment of inertia about O.

The potential energy,  $V$ , taken with the datum at  $\{\theta=90^\circ, x=0\}$  is:

$$\begin{aligned} V &= m_1 g l_{m1} \cos(\theta + \alpha) + m_2 g l_c \cos \theta \\ &= m_1 g l_{m1} (\cos \theta \cos \alpha - \sin \theta \sin \alpha) + m_2 g l_c \cos \theta \\ &= m_1 g (l_o \cos \theta - x \sin \theta) + m_2 g l_c \cos \theta \end{aligned} \quad (5.1-7)$$

To obtain the equations of motion, we use Lagrange's equation in the form:

$$\frac{\partial}{\partial t} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \quad (5.1-8)$$



where  $q_i$  is the  $i^{\text{th}}$  generalized coordinate, and  $Q_i$  is the associated generalized force<sup>1</sup>. Choosing  $q_1=x$ ,  $q_2=\theta$ , we have:

$$m_1\ddot{x} + m_1l_0\ddot{\theta} - m_1x\dot{\theta}^2 - m_1g\sin\theta = F(t) \quad (5.1-9)$$

$$m_1l_0\ddot{x} + J_0\ddot{\theta} + 2m_1x\dot{x}\dot{\theta} - (m_1l_0 + m_2l_c)g\sin\theta - m_1gx\cos\theta = 0 \quad (5.1-10)$$

These equations are useful for nonlinear system design and analysis. For example, when linear control is implemented based on a linearized plant model, the system time response with the full-term (nonlinear) and linearized plants may be simulated to evaluate the range of validity of the linear approximation. The simulation with nonlinear plant model may then be compared with actual test results on the ECP system to see the effects of further unmodeled and non-ideal dynamic behavior.<sup>2</sup>

### 5.1.2 Linearization About Equilibrium

The equilibrium points  $x=x_e$ ,  $\theta=\theta_e$ , may be found from Eq's 5.1-8 & -9 by solving for the motionless system ( $\dot{x}=\ddot{x}=\dot{\theta}=\ddot{\theta}=0$ ) when  $F(t) = 0$ . These are readily obtained as:<sup>3</sup>

$$\theta_e = 0, x_e = 0 \quad (5.1-11)$$

A linearized approximation of the system may be found via the first two (zeroeth and first order) terms of the Taylor's series expansion about the equilibrium points<sup>4</sup>. This results in:

<sup>1</sup>The  $Q_i$ 's may be found using the definition of virtual work,  $\delta W_i$ ; i.e.  $\delta W_i = Q_i \delta q_i$ . In this case, the only external force acting on the system is  $F(t)$  which acts linearly in the positive  $x$  direction and is independent of  $x$  and  $\theta$ , i.e.  $\delta W_1 = F(t)\delta x$ ,  $\delta W_2 = 0$ .

<sup>2</sup>While ECP systems follow their dynamic models quite closely, virtually any physical systems' behavior is more complex than our mathematical models of it. Characteristics not accounted for in this analysis include Coulomb friction at the belt drive, bearings, and motor brushes; motor torque ripple; quantization of the encoder signals and signal processing; and saturation of the digital-to-analog converter when the control effort signal becomes excessive. Useful models may be built however without accounting for all such effects. It is the essence of control modeling to establish a plant model of minimal order (complexity) to describe the salient dynamics through the intended control bandwidth.

<sup>3</sup>An additional solution exists at  $\theta_e = 180^\circ$ , but as the physical system is constrained to preclude this position, it shall not be considered here.

<sup>4</sup>I.e. for our variable set:  $(\bullet) \Big|_{\substack{x=x_e \\ \theta=\theta_e}} + \left( x \left( \frac{\partial}{\partial x} (\bullet) \right) \Big|_{x=x_e} \right) + \theta \left( \frac{\partial}{\partial \theta} (\bullet) \right) \Big|_{\theta=\theta_e} \Bigg)$  where  $(\bullet)$  is Eq. 5.1-9 or -10. In this case, the result is the same as setting all squared and cross terms equal to zero,  $\sin\theta$  equal to  $\theta$ , and  $\cos\theta$  equal to 1.

$$m_1\ddot{x} + m_1l_0\ddot{\theta} - m_1g\theta = F(t) \quad (5.1-12)$$

$$m_1l_0\ddot{x} + J_{o_e}\ddot{\theta} - (m_1l_0 + m_2l_c)g\theta - m_1gx = 0 \quad (5.1-13)$$

where:

$$J_{o_e} = J_o \Big|_{\substack{x=x_e \\ \theta=\theta_e}}$$

### 5.1.3 State Space Realization

Isolating the second derivative terms in equations 5.1-12,-13 we have:

$$(J_{o_e} - m_1l_0^2)\ddot{x} + m_1l_0gx - (J_{o_e} - m_1l_0^2 - m_2l_0l_c)g\theta = \frac{F(t)J_{o_e}}{m_1} \quad (5.1-14)$$

$$(J_{o_e} - m_1l_0^2)\ddot{\theta} - m_1gx - m_2l_cg\theta = -F(t)l_0 \quad (5.1-15)$$

By inspection, a state space realization of the linearized plant is:

$$\begin{aligned} \dot{x} &= Ax + BF(t) \\ Y &= Cx \end{aligned} \quad (5.1-16)$$

where:

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ m_2l_cg/J^* & 0 & m_1g/J^* & 0 \\ 0 & 0 & 0 & 1 \\ (J^* - m_2l_0l_c)g/J^* & 0 & -m_1l_0g/J^* & 0 \end{bmatrix},$$

$$B = \frac{1}{J^*} \begin{bmatrix} 0 \\ -l_0 \\ 0 \\ J_{o_e}/m_1 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{bmatrix}$$

and

$$J^* = [J_{o_e} - m_1l_0^2] \quad (5.1-17)$$

$C_i = 1$  ( $i=1,2,3,4$ ) when  $x_i$  is an output and equals 0 otherwise.

### 5.1.4 Transfer Function

By Laplace transform of Eq. (5.1-13) and assuming zero valued initial conditions it is straightforward to express

$$\frac{\theta(s)}{x(s)} = \frac{m_1 l_o}{J_{\alpha_e}} \frac{-s^2 + g/l_o}{s^2 - (m_1 l_o + m_2 l_c)g/J_{\alpha_e}} \quad (5.1-18)$$

which when substituted into Eq. (5.1-12) leads to:

$$\frac{\theta(s)}{F(s)} = \frac{l_o}{J^*} \frac{-s^2 + g/l_o}{s^4 + ((m_1 l_o - m_2 l_c)g/J^*)s^2 - m_1 g^2/J^*} \quad (5.1-19)$$

The first of the above transfer functions relates the motion of the nominally vertical (or pendulum) rod to the motion of the sliding rod and the second relates pendulum rod motion to the force acting on the sliding rod via the drive belt. It is the motion of the pendulum rod (i.e.  $\theta$ ) that is to be controlled in the experiments that follow.

The linearized relationship between the applied force and the sliding rod position follows from the product of Eq.(5.1-19) and the inverse of Eq. (5.1-18):

$$\frac{x(s)}{F(s)} = \frac{J_{\alpha_e}}{m_1 J^*} \frac{s^2 - (m_1 l_o + m_2 l_c)g/J_{\alpha_e}}{s^4 + ((m_1 l_o - m_2 l_c)g/J^*)s^2 - m_1 g^2/J^*} \quad (5.1-20)$$

## 5.2 Effect Of Mass Properties & Geometry On Plant Dynamics

From inspection of Eq. 5.1-19, it is readily seen that the transfer function of the linearized plant has one right half plane (RHP) zero and at least one RHP pole. Thus the open loop system is both nonminimum phase and unstable. As will be discussed later, these characteristics have strong implications to the attainable performance and stability of any subsequent closed loop design. In particular, the nonminimum phase zero sets an upper limit to the achievable system bandwidth, and the unstable pole sets a lower limit. These properties are discussed in more detail in Appendix A. The effects of various dynamic parameters on the values of the plant transfer function roots are described in this section.

5.2.1 Distribution And Effect Of Zeros

From Eq 5.1-19, the zeros lie at:

$$z = \pm\sqrt{g/l_o} \tag{5.2-1}$$

Thus for the experimental plant the zeros are fixed since the pendulum rod length,  $l_o$ , is not changeable. From the discussions of Appendix A, the lower the "frequency" (magnitude) of the RHP zero, the lower the maximum attainable closed loop bandwidth, and hence performance. From the Appendix, for practical control implementation, we would not expect to attain closed loop bandwidths greater than about  $(g/l_o)^{1/2}$  or 1.2 Hz for our particular geometry.

5.2.2 Distribution And Effect Of Poles

In order to gain insight as to the effect of mass properties on the distribution of the poles, consider the following notation:

$$a \triangleq (m_1l_o - m_2l_c) / J^* \tag{5.2-2}$$

$$b \triangleq m_1 / J^* \tag{5.2-3}$$

$$k \triangleq l_o / J^* \tag{5.2-4}$$

Note that since:

$$J^* = J_o - m_1l_o^2 = J_1 + J_2 + m_2l_c^2 (>0) \tag{5.2-5}$$

$b>0$  and  $k>0$ . Equation 5.1-19 may then be expressed as:

$$\frac{\theta(s)}{F(s)} = \frac{-k(s^2 + g/l_o)}{s^4 + ags^2 - bg^2} \tag{5.2-6}$$

or:

$$\frac{\theta(s)}{F(s)} = \frac{-k(s^2 + g/l_o)}{(s^2 - p_1g)(s^2 - p_2g)} \tag{5.2-7}$$

where:

$$p_1 = \frac{-a + \sqrt{a^2 + 4b}}{2} (>0) \tag{5.2-8}$$

and

$$p_2 = \frac{-a - \sqrt{a^2 + 4b}}{2} (<0) \tag{5.2-9}$$

Equations 5.2-7,8,&9 show that there are two real poles at  $\pm \sqrt{p_1 g}$  and two imaginary poles at  $\pm i\sqrt{|p_2| g}$ . Two possible pole-zero distributions are shown in Figure 5.2-1.

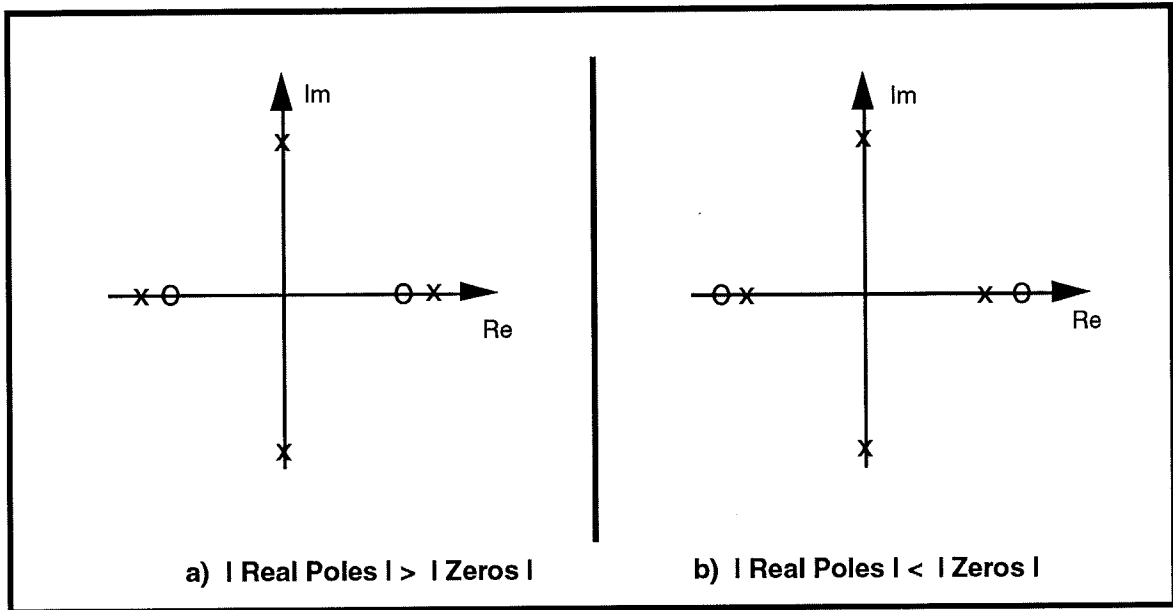


Figure 5.2-1. Possible Pole-Zero Distributions

According to Eq's 5.2-1,-7,&-8, the distribution of Figure 5.2-1a implies that:

$$\frac{-a + \sqrt{a^2 + 4b}}{2} > \frac{1}{l_0} \tag{5.2-10}$$

It is relatively straightforward to show that this holds if and only if:

$$J^* < m_2 l_c l_0 \tag{5.2-11}$$

Similarly, the distribution of Fig. 5.2-1b implies that:

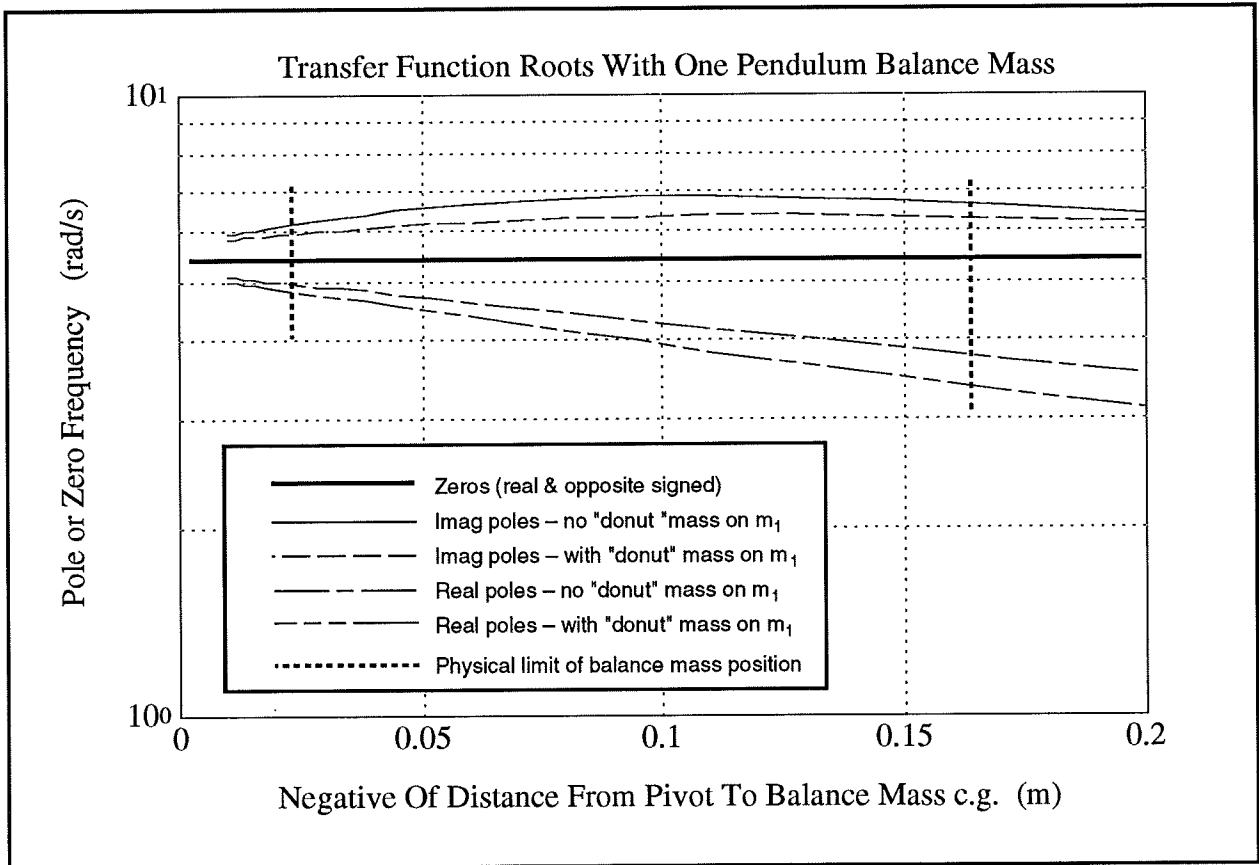
$$J^* > m_2 l_c l_0 \tag{5.2-12}$$

Eq. 5.2-11 is equivalent to the condition that  $J_1 + J_2 + m_2(l_c)^2 < m_2 l_c l_0$ . This in turn requires that  $J_1$  and  $J_2$  be small and  $l_c$  be small relative to  $l_0$ . It can be shown that for the range of available parameters, this condition is never realized in the experimental system. Therefore, all pole-zero distributions are as shown in Figure 5.2-1b. Note also from Eq's 5.2-8,-9 that the magnitude (frequency) of the imaginary poles is always greater than the real poles.

For the experimental system, the masses  $m_1$  and  $m_2$  may be changed discretely by adding or removing the appropriate brass weights and by changing the distance of the balance weight along

the lead screw from the pivot. The effect of these changes is shown in Figure 5.2-2. From the discussions of Appendix A, the lower the frequency of the unstable pole, the less difficult it is to stabilize the system under closed loop control. As seen in the Figure, stabilizing the system becomes less difficult when the balance masses are added, when they are moved as far downward as possible,<sup>1</sup> and when the "donut" weights on the sliding rod are removed.<sup>2</sup>

The case where both balance masses are removed altogether is nearly identical to the left most points on the curves of Figure 5.2-2a. From the discussion of Appendix A this would be a difficult plant to stabilize indeed.



**Figure 5.2-2a. Effect of Changing Plant Mass Properties - Single Balance Mass**

<sup>1</sup>This result is rather intuitive – as the system c.g. moves downward and below the pivot, the system would appear to become similar to an ordinary (non inverted) pendulum. Note from the transfer function denominator that the system *never* becomes stable even for large negative values of the product  $m_2 l_c$ . This somewhat non intuitive result may be visualized by considering small static perturbations about the equilibrium. If for example the pendulum is moved slightly in the clockwise direction, the sliding rod will tend to move toward the right thereby causing further clockwise pendulum displacement. Thus the equilibrium is unstable. It may be demonstrated on the experimental system however that in such cases, relatively small amounts of friction result in a stable equilibrium.

Note that in the  $\theta(s)/x(s)$  (Eq. 5.1-18) case however the transfer function becomes stable for sufficiently large negative values of  $m_2 l_c$ .

<sup>2</sup>Removing the "donut" weights however reduces the available control authority (e.g. smaller attainable range in  $\theta$ ).

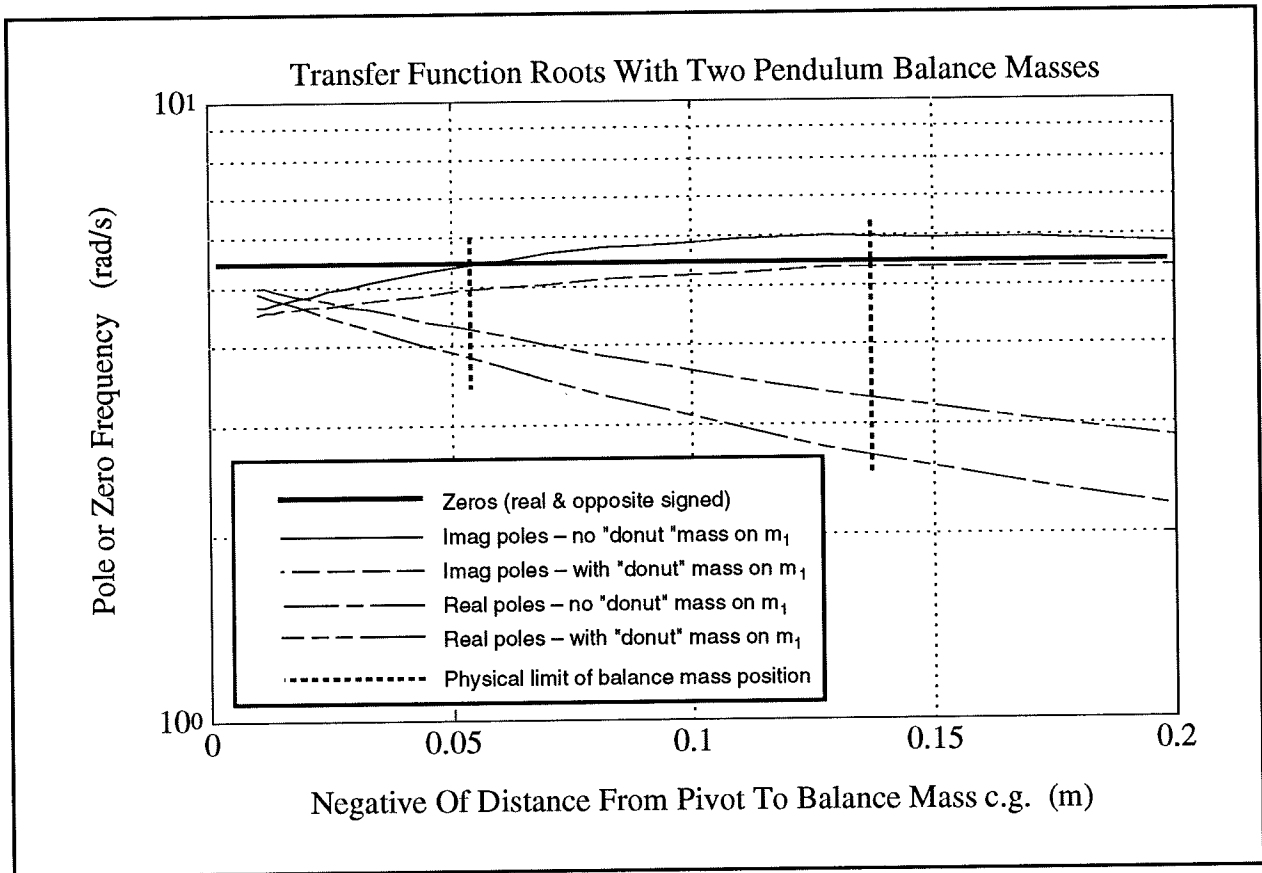


Figure 5.2-2a. Effect of Changing Plant Mass Properties - Two Balance Masses

5.2.3 Mass Property Determination.

Both the nonlinear (Eq's 5.1-9,-10) and linearized (Eq's 5.1-12) time domain equations of motion and the transfer function (Eq. 5.1-19 or Eq's 5.2-6,-7) are fully specified by the parameters  $m_1$ ,  $m_2$ ,  $l_o$ ,  $l_c$ , and  $J^*$ . For the experimental pendulum (and for a given set of attached brass weights), the parameters  $m_1$ ,  $m_2$  and  $l_o$  are fixed and known. The parameters  $l_c$ , and  $J^*$  however depend on the location of the movable brass counter-mass. Define  $l_{c_0}$  and  $m_{2_0}$  to be the value of  $l_c$  and  $m_2$  respectively when the counter-mass,  $m_{w2}$ , (both brass weights) is removed. Then the parameter  $l_c$  is found from a simple center of mass calculation, i.e.:

$$l_c = \frac{m_{w2}l_{w2} + m_{2_0}l_{c_0}}{m_2} \tag{5.2-13}$$

where  $l_{w2}$  is the distance (signed) from the pivot to the center of  $m_{w2}$ . In our case, the parameters  $l_{c_0}$  and  $m_{2_0}$  are also fixed and known so that determination of  $l_c$  is straightforward. The remaining parameter  $J^*$  is found experimentally<sup>1</sup> as described in Chapter 6.

<sup>1</sup>More precisely, we will find  $J_{o_c}$  and then obtain  $J_o^*$  via Eq. 5.1-17.

## 6. Experiments

This chapter outlines experiments which identify the plant parameters, implement several control schemes, and demonstrate important control principles. The versatility of this software / hardware system allows for a broader range of experimental uses than will be described here and the user is encouraged to explore whatever topics and methodologies may be of interest – subject of course to your school and laboratory guidelines and the safety notations of this manual. The safety portion of this manual, Section 2.4, must be read and understood by any user prior to operating this equipment.

The instructions in this chapter begin at a high level of detail so that they may be followed without a great deal of familiarity with the PC system interface and become more abbreviated in details of system operation as the chapter progresses. To become more familiar with these operations, it is strongly recommended that the user read Chapter 2 prior to undertaking the operations described here. Remember here, as always, it is recommended to save data and control configuration files regularly to avoid undue work loss should a system fault occur.

### 6.1 System Identification

It is necessary to identify pertinent dynamic and scaling parameters in order to design and implement controllers which stabilize the inverted pendulum plant and allow it to track set-points in angular displacement. In Chapter 5 the nonlinear dynamical equations of the plant were derived. A linearized model of the plant dynamics was also presented in both state-space form and transfer function form. The linearized model can be shown to be an adequate representation in many cases for the purpose of linear control design and implementation (e.g. the controller used as the default controller in Section 3.2 uses a linear controller designed based upon this linearized model). The purpose of the proceeding identification procedures will be to provide numerical values for the symbolic equations derived in Chapter 5. It is recognized that not all parameters may be measured with a fully assembled pendulum. As result, Table 6.1-1 below shows the parameters which are fixed. Their nominal values have been measured prior to assembly:



**Table 6.1-1. Mass Property Values**

Parameter	Value	Description
$l_o$	0.330 (m)	Length of pendulum rod from pivot to the sliding rod T section
$m_1$	TBD <sup>†</sup> (kg)	Mass of the complete sliding rod including all attached elements.
$m_{1o}$	0.103 (kg)	Mass of the sliding rod with belt, belt clamps, and rubber end guards (but without sliding rod brass "donut" weights)
$m_{w1}$	0.110 (kg)	Combined mass of both of the sliding rod brass "donut" weights (=0 if not used)
$m_2$	TBD (kg)	Mass of the complete assembly minus $m_1$
$m_{w2}$	1.000 (kg) (÷2 if only one weight used)	Mass of brass balance weight
$m_{2o}$	0.785 (kg)	Mass of the complete moving assembly <u>minus <math>m_1</math> and <math>m_{w2}</math></u>
$l_{co}$	0.071 (m)	Position of c.g. of the complete pendulum assembly with <u>the sliding rod and balance weight removed</u>
$J_o^*$	0.0246 (kg-m <sup>2</sup> )	$[J_{oe} - m_1 l_o^2]$ evaluated when $m_{w2}=0$ .

<sup>†</sup>To be determined

From the definitions in the table we have:

$$m_1 = m_{1o} + m_{w1} \tag{6.1-1}$$

$$m_2 = m_{2o} + m_{w2} \tag{6.1-2}$$

The parameter

$$l_{w2} \triangleq \text{signed distance from pivot to c.g. of balance mass (} m_{w2} \text{)}$$

is changeable by the user and is readily measured. That is, referring to Figure 6.1-1,

$$l_{w2} = - \frac{(t+l_t+l_b)}{2} \tag{6.1-3}$$

The three remaining parameters –  $J^*$ ,  $J_{oe}$ , and  $l_c$  – are derived from the above as follows:

$$l_c = \frac{m_{w2}l_{w2} + m_{2o}l_{co}}{m_2} \tag{5.2-13}$$

$$J_{oe} = J_o^* + m_1 l_o^2 + m_{w2}(l_{w2})^2 \tag{6.1-4}$$

$$J^* = [J_{oe} - m_1 l_o^2] \tag{5.1-17}$$

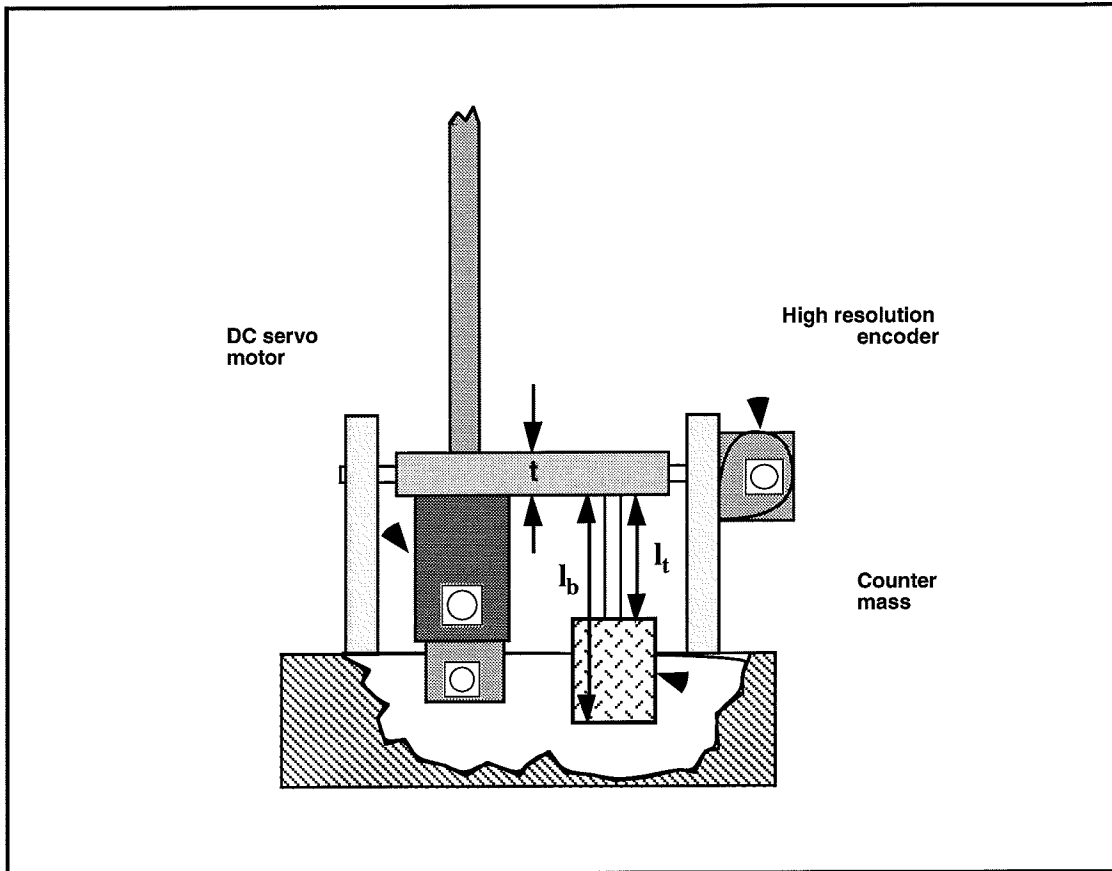


Figure 6.1-1 Measurement of  $I_w 2$

**Plant Configurations For Control Experimentation.**

The experiments that follow in this manual use two distinct physical setups and corresponding parameter sets for their models. In each case both "donut" weights are used for the sliding rod and both balance masses are used. For the first case, "Plant #1",  $l_t = 10.0$  cm, and for the second ("Plant #2")  $l_t = 7.0$  cm, .

Based upon the above information you should now be able to calculate the numerical values of the linearized plant parameters as given by Eq's 5.1-18, -19, and -20 in Chapter 5. Calculate these values for each plant case. Calculate the roots of the transfer function Eq 5.1-19 and show that the real poles and zeros are mirror images about the imaginary axis on the s-plane and that there is also an imaginary pole pair. Show that the poles of Eq. 5.1-18 are imaginary for plant #1 and are real and opposite for plant #2.<sup>1</sup>

Now we can use the numerical value of plant transfer function or state space model to design controllers. However to actually implement them we need four scaling parameters which are associated with the amplifier/actuator, the two sensors (encoders) and the controller firmware.

<sup>1</sup>In Plant #1, the system c.g. – given by  $(m_1 l_o + m_2 l_c) / (m_1 + m_2)$  – is below the pivot and according to Eq. 5.1-17 results in imaginary poles. Hence in the presence of friction it is stable (for  $\theta(s)/x(s)$ , not for  $\theta(s)/F(s)$ ). For plant #2 the system c.g. is above the pivot and hence  $\theta(s)/x(s)$  is stable.

With reference to the block diagram of Figure 6.1-2, these four scale factors (gains) are as shown in Table 6.1-2 where  $\theta_c$  and  $x_c$  are the respective values of  $\theta$  and  $x$  in units of encoder counts.<sup>1</sup>

Now, all of the above parameters are fixed. The value of  $k_f$  is factory set and will not change unless the amplifier analog gains are changed. The value of  $k_x$  comes from the fact that encoder 2 generates 2000 counts per revolution which translates to approximately 50,000 counts per one meter travel of the sliding rod (after 4x interpolation, see Section 4.5). Similarly,  $k_a$  reflects the 16,000 counts per rev. resolution of encoder 1. The value of  $k_s$  is fixed by the firmware within the real-time control card.

Construct the properly scaled plant models (with variables  $\theta_c$  and  $x_c$  and control force  $F_c$ ) in both transfer function and state space form (i.e. Eq's 5.1-16, -18, -19, and -20) by using the following substitutions.<sup>2</sup>

$$\theta = \theta_c / (k_s k_a) \tag{6.1-5}$$

$$x = x_c / (k_s k_x) \tag{6.1-6}$$

$$F = F_c k_f \tag{6.1-7}$$

**Table 6.1-2. Scaling Constants**

Scale Factor	Value	Description
$k_f$	0.0013 (N / DAC increment)	Combined DAC-amplifier-motor-pulley gain (see Sect. 4.3)
$k_x$	50200 (increment / m)	Sliding rod encoder (encoder 2) scale factor
$k_a$	2546 (increment / rad)	Pendulum angle encoder (encoder 1) scale factor
$k_s$	32 (controller or DAC incr./ encoder or demand incr.)	DSP controller firmware scale factor

<sup>1</sup>The control effort gain,  $k_f$ , is the product of the DAC gain (V/increment), the amplifier gain (amps/V), the motor torque constant (N-m/amp) and the inverse of the effective belt pulley radius ( $m^{-1}$ ). The "x" variable feedback gain,  $k_x$ , is actually the encoder resolution in counts/radian times the effective belt drive pulley radius in meters. The "θ" variable feedback gain is the corresponding encoder resolution.

The controller gain,  $k_s$ , is part of the controller firmware and multiplies all feedback and command input signals prior to their entering the control block (i.e. all inputs to the control law are scaled by 32x). This is for improved numerical resolution in pulse period measurement (to obtain derived rate) which occurs within the controller and is transparent to the user.. Note that for display and plotting purposes, the Executive program divides the corresponding controller-internal values by 32 so that they appear in the original scaling.

<sup>2</sup> For the transfer function models this scaling process is straightforward. For the state space plant model however the scaling affects both the A and B matrices. It may best to first express Eq's 5.1-14 and -15 in terms of the controller scaled variables and then rewrite the expression Eq. 5.1-16.

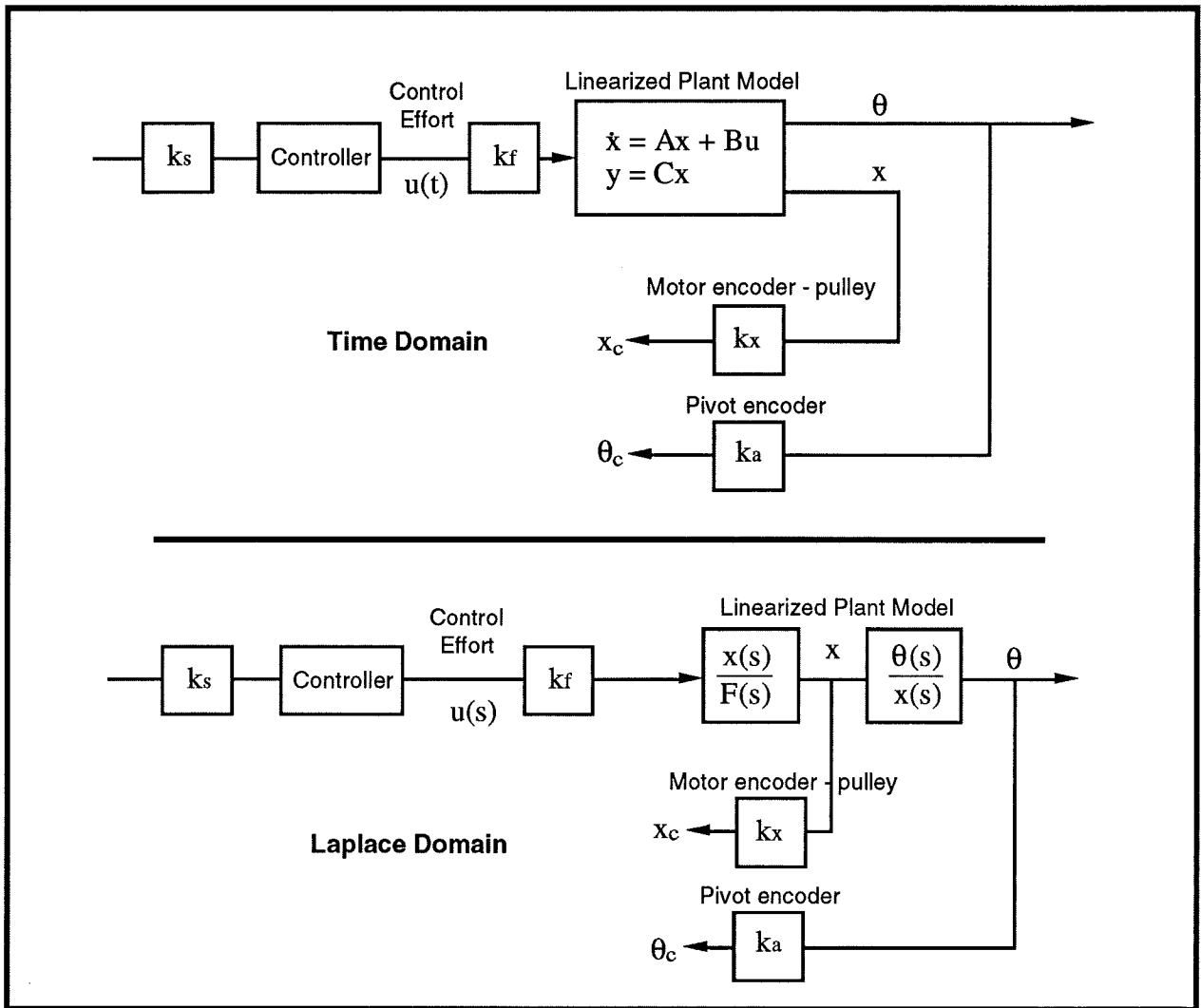


Figure 6.1-2 Plant Block Diagram With Scale Factors

6.1.1 Confirmation of  $k_a$  and  $k_x$

These two scale factors can be easily confirmed via a simple procedure using the Executive program

Procedure

1. Remove the power cord from the control box but have all other cables connected. Position the pendulum rod to the right and the sliding rod to the far right at the limit of travel. Enter the Executive program.
2. Select **Zero Position** in the **Utility** menu. You should see the position of Encoder 1 and Encoder 2 to be approximately zero.
3. Hold on to the pendulum rod and push the sliding rod from one end until it hits the opposite end of its travel (hits the right limit switch). Now record the number of counts moved by

encoder 2 on the **Background Screen**. With a ruler, measure the distance in meters that the sliding rod has traveled from one limit switch to the other. The ratio of the two is the value of  $k_x$ .

- Now move the pendulum rod in the anti-clockwise direction all the way to the left. Record the number of counts moved by encoder 1 on the **Background Screen**. Measure the angle of rotation in radians, the ratio of the two is the value of  $k_a$ .

### 6.1.2 Confirmation of $J_o^*$

The following procedure may be used to confirm the value of  $J_o^*$  from which  $J^*$  and  $J_{oe}$  are obtained via Eq's 6.1-4 and 5.1-17

#### Procedure:

- Remove the brass balance weights from the apparatus and install both "donut" weights on the sliding rod. Use a rubber band to restrain the sliding rod in its center of travel as shown in Figure 6.1-2.
- Disconnect the power cord from the control box but leave the other cables connected. (This allows the encoder signals to pass to the control card but precludes accidental driving of the motor.) Very carefully position the entire pendulum mechanism up-side down on two coplanar flat surfaces such that the pendulum rod is free to rotate as a non-inverted (regular) pendulum. (E.g. two tables side-by-side with approx. 8 in. gap between. It is recommended that a piece of cloth or other soft material be placed between the mechanism and the table on each side to avoid scratching.)

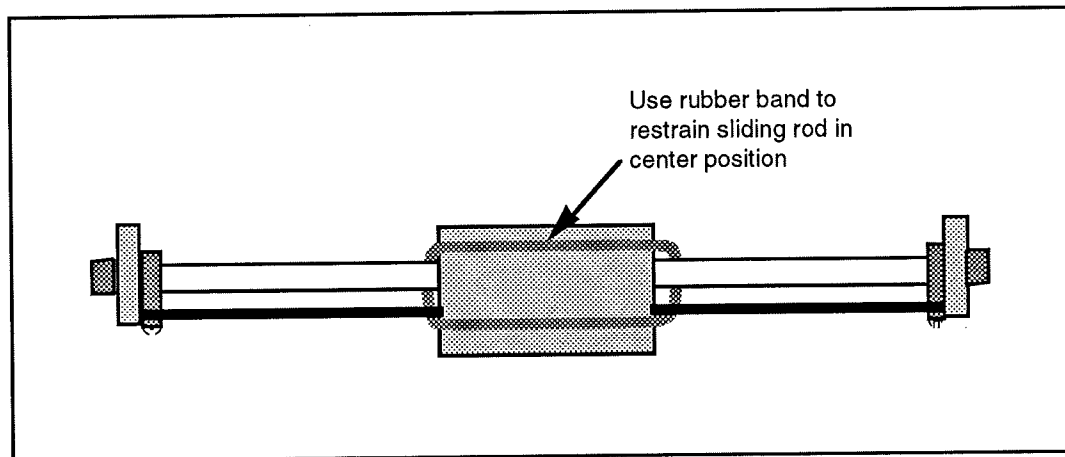
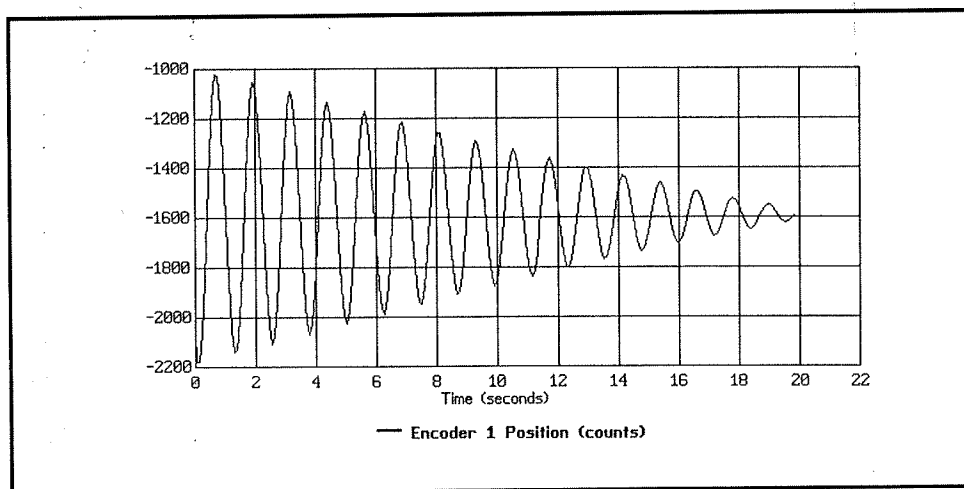


Figure 6.1-2. Securing Sliding rod For  $J_o^*$  Measurement via Pendulum Frequency

- With the controller powered up, enter the **Control Algorithm** box via the **Set-up** menu and set  $T_s = 0.00442$ . Go to **Set up Data Acquisition** in the **Data** menu and select **encoder #1** as data to acquire and specify data sampling every 5 servo cycles (i.e. every 5  $T_s$ 's). Select **OK** to exit. With the pendulum hanging freely under gravity, select **Zero Position** from the **Utility** menu to zero the encoder positions.

4. Check again that the pendulum rod can freely rotate in this position. Now select **Trajectory** in the **Command** menu. Enter the **Step** dialog box and click on **step-up**. Choosing **Open loop Step**, input a step size of **0 (zero)**, a duration of **10000 ms** and **repetition** of **1**. Exit to the Background Screen by consecutively selecting **OK**. This puts the controller in a mode for acquiring 20 sec of data on command but without driving the actuator. This procedure may be repeated and the duration adjusted to vary the data acquisition period.
5. Select **Execute** from the **Command** menu. Manually displace the pendulum rod approximately 20 deg. from vertical and let go. You should notice that the "non-inverted" pendulum rod will oscillate and slowly attenuate while the encoder data is collected to record this response. Select **OK** after data is uploaded.
6. Select **Set-up Plot** from the Plotting menu and choose **encoder #1 position**; then select **Plot Data** from the **Plotting** menu. You should see the pendulum rod angle encoder response similar to one shown in Figure 6.1-2.



**Figure 6.1-2 Determination of  $J_o^*$  via Pendulum Period Measurement**

7. Measure the period of oscillation,  $T$ , in seconds by taking the time for completion of several cycles divided by the number of cycles. Confirm that its value for the factory default setting of the balance weight (all the way down the lead screw but just clearing the pocket hole) is approximately 1.25 seconds<sup>1</sup>. Use the following classic linearized "non-inverted" equation of motion to derive an expression for the inertia  $J$  in terms of the measured  $T$ .

$$J\ddot{\theta} + ml_{cg}g\theta = 0 \quad (6.1-8)$$

where  $m$ ,  $J$  and  $l_{cg}$  are the respective mass, inertia, and length to center of gravity (from the pivot) of the combined pendulum elements during the test.

8. Calculate the values of  $m$  and  $l_{cg}$  for the test case and hence obtain  $J$  from your derived expression. (Note that  $J$  here is the same as  $J_{oe}$  of this manual's notation) Now determine  $J_o^*$  according to  $J_o^* = J - m_1 l_o^2$  and hence verify the value shown in Table 6.1-1.

<sup>1</sup>For more accurate measurement, you may use tabular data via **Export Raw Data** in the **Data** menu to export data to a text editor where precise numerical values may be read.

9. Carefully re-orient the mechanism to the regular "inverted pendulum" position (up-side up) and reconnect the power cord to the control box. (Make certain that the control box is turned off however.) This concludes the confirmation procedure for  $J^*$ .

From the above exercises all plant parameters and scaling constants have been found or verified.

## 6.2 Successive Loop Closure / Pole Placement Design

In this experiment we first close the loop about the "x" position with a relatively high bandwidth (close tracking) PD control. We then make the assumption that the sliding rod closely follows its internal demand  $u_x$  so that for designing a controller for  $\theta$ , the "plant" is approximated by the transfer function  $\theta_c(s)/x_c(s)$  (i.e. the scaled version of Eq. 5.1-18). The block diagram for this approach is given in Figure 6.2-3.

### 6.2.1 PD Control Of The Sliding Rod

A simplified dynamical model of the relationship between the sliding rod and the applied force,  $F$  when the plant is at the equilibrium position is shown in Figure 6.2-1. Here the inertia of the pendulum assembly is shown as an equivalent mass  $m_2^*$  at the sliding rod interface according to:

$$m_2^* = J^*/l_0^2 \tag{6.2-1}$$

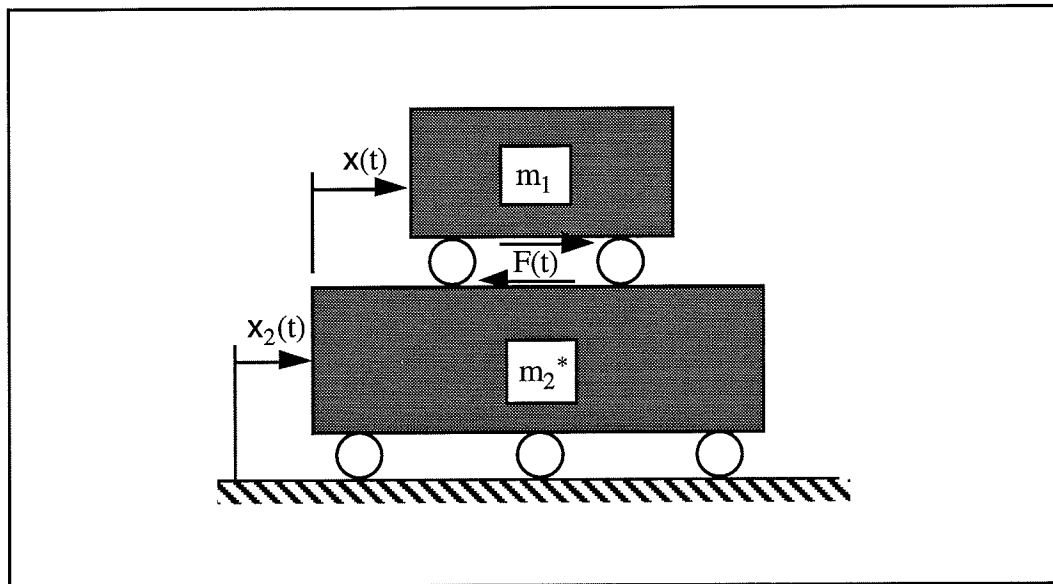


Figure 6.2-1 Simplified Model Of Sliding Rod Dynamics When Plant Is At Equilibrium

The reader should verify that the following relationship holds for the motion  $x(t)$ :

$$F(t) = m^* \ddot{x} \tag{6.2-2}$$

where

$$m^* = \frac{m_1 m_2^*}{m_1 + m_2^*} \tag{6.2-3}$$

Thus the reduced dynamics are that of a simple rigid body. For our particular plant, the corresponding transfer function in controller units is:

$$\frac{x_c(s)}{F_c(s)} = \frac{k_{hw}}{m_1^* s^2} \tag{6.2-4}$$

where

$$k_{hw} \triangleq k_x k_s k_f \tag{6.2-5}$$

The block diagram of the proportional plus derivative (PD) control of a rigid body is shown in Figure 6.2-2 and has the closed loop transfer function:<sup>1</sup>

$$c(s) = \frac{x(s)}{r(s)} = \frac{(k_{hw}/m_1^*)(k_d s + k_p)}{s^2 + (k_{hw}/m_1^*)(k_d s + k_p)} \tag{6.2-6}$$

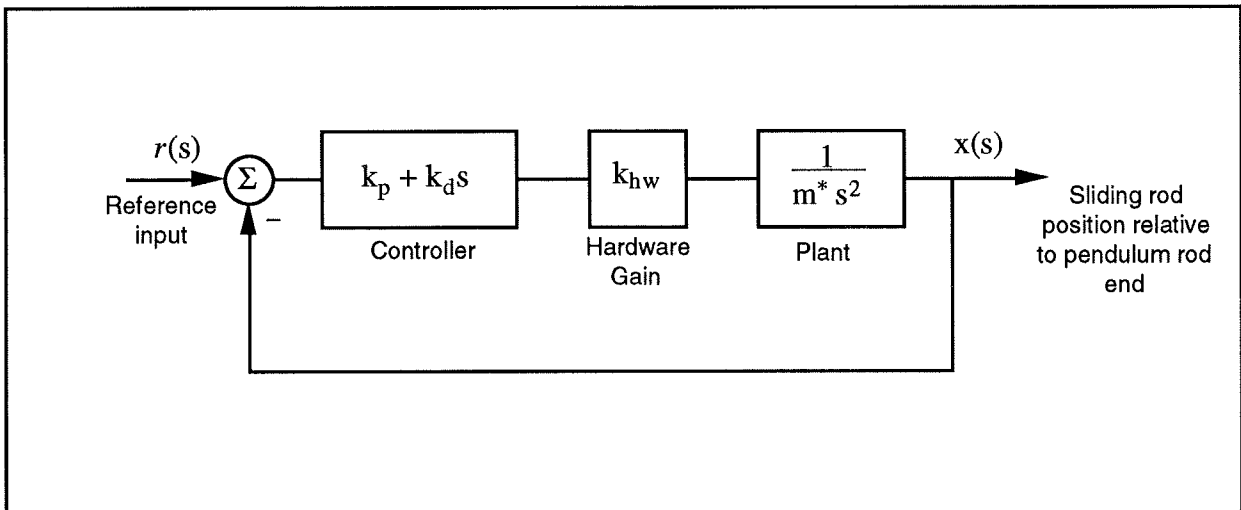


Figure 6.2-2. Rigid Body PID Control – Control Block Diagram

<sup>1</sup>Here we omit the subscript "c" since  $x_c(s)/r_c(s) = x(s)/r(s)$ .



By defining:

$$\omega_n \triangleq \sqrt{\frac{k_p k_{hw}}{m^*}} \quad (6.2-7)$$

$$\zeta \triangleq \frac{k_d k_{hw}}{2m^* \omega_n} = \frac{k_d k_{hw}}{2\sqrt{m^* k_p k_{hw}}} \quad (6.2-8)$$

we may express:

$$c(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (6.2-9)$$

Here  $\omega_n$  is the *natural frequency* and  $\zeta$  is the *damping ratio* of the closed loop system  $x(s)/F(s)$ .

Procedure :

PD Control

1. Using the results of Section 6.1 for Plant #1 determine the values of the parameters  $k_{hw}$  and  $m_1^*$ .
2. Using Equations 6.2-7 and -8, determine values of the control parameters  $k_p$  and  $k_d$  to provide a closed loop natural frequency of 10 Hz (63 rad/s) and damping ratio of 1 (*critically damped*). Note: if you obtain  $0.15 < k_p < 0.35$  or  $0.004 < k_d < 0.012$  you should check and correct your calculations.
3. Adjust the position of the balance masses to  $l_t = 10.0$  cm being sure to secure them on the threaded rod by counter rotating them. Verify that the donut weights are in place and secure on the sliding rod. Hence place the apparatus in the Plant #1 configuration.
4. In the Executive program, set-up to collect **Encoder #2** and **Commanded Position** information via the **Set-up Data Acquisition** box in the **Data** menu with data sampling every two sample periods. Enter the **Command** menu, go to **Trajectory** and select **Step, Set-up**. Select **Closed Loop Step** and input a step size of **1000** counts, a duration of **1000 ms** and **1 repetition**. Exit to the Background Screen by consecutively selecting **OK**. This puts the controller board firmware in a mode to command a 1000 count step move (about 2 cm.) forward and back with a one second dwell.
5. Enter the **Control Algorithm** box under **Set-up** and set **Ts=0.00442 s** and select **Continuous Time Control**. Select **PID** and **Set-up Algorithm**. Enter the  $k_p$  and  $k_d$  values ( $k_i = 0$ ) determined in step 2 (again, do not enter values outside of the ranges  $0.15 < k_p < 0.35$  or  $0.004 < k_d < 0.012$ ) and select **Encoder #2** for feedback, then **OK**.

Orient the mechanism with the sliding rod in its approximate middle of travel so that the pendulum rod is approximately vertical. In this and all future work, be sure to stay clear of the mechanism before doing the next step. Selecting **Implement Algorithm** immediately implements the specified controller; if there is an instability or large control signal<sup>1</sup>, the plant may react violently. If the system

<sup>1</sup>E.g. a large error at the time of implementation.

appears stable after implementing the controller, first displace it with a light, non sharp object (e.g. a plastic ruler) to verify stability prior to touching plant. AT ALL TIMES KEEP HEAD AND ESPECIALLY EYES WELL CLEAR OF THE APPARATUS

Select **Implement Algorithm**, then **OK**.

6. Enter **Execute** under **Command** and – again staying clear of the apparatus –select **Run**. You should see the sliding rod move rapidly about 2 cm back and forth while the pendulum rod swings due to the reactive force.
7. Plot encoder #2 output and commanded position, both on the left axis (see step 6, Section 6.1.2). You should see a critically damped response with 90% rise time of approximately 60 ms. Print your plot via **Print Data** under the **Plotting** menu.

### 6.2.2 Pole Placement Control of $\theta(s)/x(s)$

Having closed a relatively high bandwidth ( $\approx 10$  Hz) loop about the sliding rod position we utilize the fact that the transfer function of Eq. 6.2-9 has near unity input/output gain (and near zero phase) through the bandwidth ( $\approx 0.75$  Hz) that we will attempt to attain in the overall control of  $\theta$ . Thus for the control of  $\theta$  we consider the block diagram of Figure 6.2-3.<sup>1</sup>

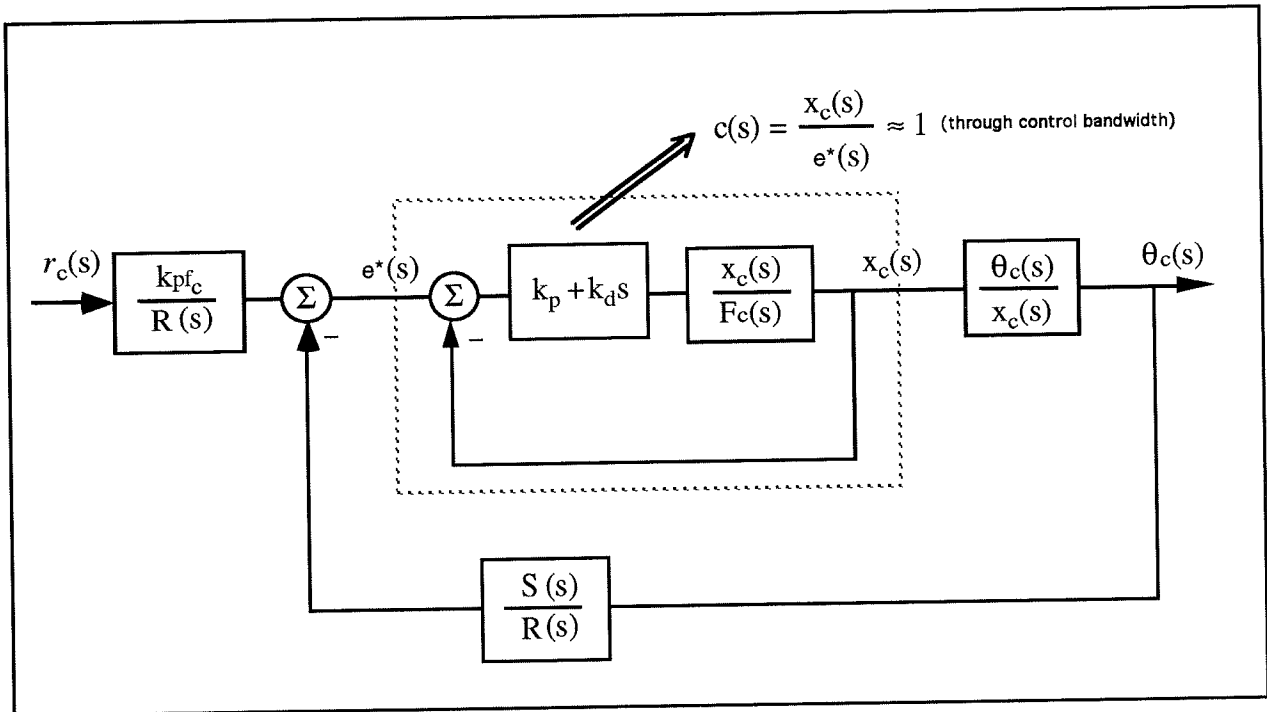


Figure 6.2-3. Control Structure For Outer Loop Closure

<sup>1</sup>Note again that  $\theta/r = q_c/r_c$ .

Now the plant to be controlled is:

$$\frac{\theta_c(s)}{x_c(s)} = \frac{k_a m_1 l_o}{k_x J_{o_e}} \frac{-s^2 + g/l_o}{s^2 - (m_1 l_o + m_2 l_c)g/J_{o_e}} \triangleq k^* \frac{N_{ax}(s)}{D_{ax}(s)} \quad (6.2-10)$$

where the obvious substitutions are made in obtaining the right hand expression. The numerical values of the parameters in this expression were determined in Experiment #1.

We now seek to find a controller  $S(s)/R(s)$  which will result in a prescribed set of closed loop poles. The closed loop denominator will have the form:

$$D_{cl}(s) = D_{ax}(s)R(s) + k^* N_{ax}(s)S(s) \quad (6.2-11)$$

which may be expressed as<sup>1</sup>

$$D_{cl}(s) = (d_2 s^2 + d_1 s + d_0)(r_1 s + r_0) + (n_2 s^2 + n_1 s + n_0)(s_1 s + s_0) \quad (6.2-12)$$

where the  $d_i$ 's and  $n_i$ 's are known from the plant model Eq. 6.2-10

By linear system theory, for coprime  $N_{ax}(s), D_{ax}(s)$  with  $N_{ax}(s)/D_{ax}(s)$  proper, there exists an  $(n-1)^{th}$  order  $S(s), R(s)$  which when convolved as per Eq. (6.2-11) form an arbitrary  $(2n-1)^{th}$  order  $D_{cl}(s)$  where  $n$  is the order of  $D_{ax}(s)$ .

Here we shall solve for the desired denominator:

$$D_{cl}(s) = (s + \pi + j\pi)(s + \pi - j\pi)(s + 3\pi) \quad (6.2-13)$$

I.e. closed loop poles at  $-0.5 \pm j0.5$  and  $-1.5$  Hz.

Procedure :

Pole Placement Control

1. Determine the coefficients of the controller polynomials  $S(s)$  and  $R(s)$  by equating coefficients in the expanded forms of Eq's 6.2-12 and 6.2-13.
2. Calculate the scalar prefilter gain  $k_{pfc}$  by referring to Figure 6.2-3 and Eq. 5.1-18 using the scaled variable set  $\theta_c$ , and  $x_c$ . The goal is to have the output  $\theta_c(s)$  (or

---

<sup>1</sup>The notation here is the obvious one. The constant  $k^*$  has been incorporated in the  $n_i$ 's.

$\theta(s)$  scaled equal to the input  $r_c(s)$  (or  $r(s)$ ) Hint: Consider the system in static equilibrium. Set  $\theta_c = 1$  and  $r_c = 1$ ; determine the required input  $x_c(s)$  to the plant  $\theta_c(s) / x_c(s)$  and hence the necessary signal at the left hand input to the left most summing junction in Figure 6.2-3.

4. In the Executive program, set-up to collect **Encoder #1, Encoder #2** and **Commanded Position** information<sup>1</sup> via **Set-up Data Acquisition** with data sampling every two sample periods. Enter the **Command** menu, go to **Trajectory** and select **Step, Set-up**. Select **Closed Loop Step** and input a step size of **500** counts, a duration of **2500 ms** and **1 repetition**. Exit to the Background Screen by consecutively selecting **OK**. This puts the controller board firmware in a mode to command a 500 count step move (about 11 deg.) forward and back with a 2.5 second dwell.
5. Enter the **Control Algorithm** box under **Set-up** and set<sup>2</sup> **T<sub>s</sub>=0.00884 s** and select **Continuous Time Control**. Select **General Form** and **Set-up Algorithm** and enter the  $k_p$  and  $k_d$  values ( $k_i = 0$ ) from the previous section (Section 6.2.1) as the coefficients  $e_0$  and  $e_1$  respectively. Enter the coefficients for  $S(s)$  and  $R(s)$  determined in step 1. Enter the value  $k_{pf}$  calculated in step 2 as **t<sub>o</sub>** in the General Form. Make sure that Encoder #1 is selected for Loop #1 and Encoder #2 for Loop #2. Select **OK** to exit to the controller selection dialog box.
6. Always do the following to avoid transients when implementing your controller. Exit to the background screen. Select **Abort Control** to make certain that the pendulum is safe to touch. Set the sliding rod at its approximate center of travel (pendulum rod approximately vertical). Check that the controller box is powered on and turn on if it is not. Select **Reset Controller** from the **Utility** menu. Do not disturb the pendulum.
7. Reenter the **Control Algorithm** box and while staying clear of the mechanism select **Implement Algorithm**. If the pendulum reacts violently you have implemented an unstable controller or otherwise improperly entered the control coefficients and you will need to repeat the above steps as appropriate. If the pendulum is well behaved you may (again, while staying clear of the pendulum) perturb the pendulum rod gently with a long blunt object such as a ruler. You should see the sliding rod move in an attempt to regulate against the disturbance. You may notice some "twitching" or buzzing due to noise in the system. This is to be expected. The time under closed loop control should be minimized however to reduce the stress on the drive components (In the next experiment, a filter is designed to reduce the noise.)
8. Enter **Jog Position** under **Utility** and input an absolute jog of -250 counts. You should see the pendulum move clockwise about 5.5 degrees.

<sup>1</sup>You may also select **Control Effort** if you wish to later observe this value when plotting system responses.

<sup>2</sup>The sample period is set higher here than for the PD inner loop to reduce noise propagation in the closed loop system. A filter shall be used for this purpose in later experiments.

9. Enter **Execute** under **Command** and – again staying clear of the apparatus –select **Run**. You should see the pendulum – actuated by the sliding rod – move rapidly about 11 deg back and forth. Open the loop by selecting "Abort Control".
10. Plot encoder #1 output and commanded position, both on the left axis. Note the initial negative going motion of the pendulum. This is a time response characteristic of nonminimum phase systems. Print your plot via **Print Data** under the **Plotting** menu. Save your control configuration via **Save Settings** in the **File** menu.

### 6.3 Low Pass Filter Augmentation Of Controller

It may have been noticed in the previous experiment that the pendulum appeared to "jitter" or "twitch" as a result of quantization noise which is always amplified by numerical differentiation. To overcome this problem a low pass filter may be used which should have a break frequency much higher than the closed loop bandwidth (so that system stability is not significantly affected) and yet low enough to effectively attenuate the noise. In this section we shall investigate the effect of a simple first order lag with unity dc gain.

#### Procedure

1. Verify that a first order filter represented by the transfer function  $G(s) = 1/(\tau s + 1)$  has a low-pass frequency response with bandwidth  $1/\tau$  (rad/s).
2. Consider the filter given by:

$$G(s) = 1/(0.008s + 1)$$

What is the cut-off frequency of this filter and how does its value compare to the magnitude of the closed loop poles of the design in the previous section?

3. To implement this filter as a cascade to the output of the control blocks used in Section 6.2, we must use the **General Form** structure within the **Setup Control Algorithm** Dialog Box. Enter the ECP Executive program and in **Setup Control Algorithm**, enter the sampling period  $T_s = 0.001768$ <sup>1</sup>. Input control coefficients in the **General Form** box from the design of Section 6.2 (you may do this quickly by uploading your saved configuration from the last experiment via **Load Settings** in the **File** menu – make sure however that you choose  $T_s = 0.001768$ ) Now input  $g_1 = 0.008$ ,  $g_0 = 1$ . This corresponds to the implementation of the above first order low pass filter to the output of the control action from the PD inner loop (i.e. immediately prior to the control signal output from the controller

<sup>1</sup>This relatively fast sampling rate is used to minimize sampled data effects.

and being put on the DAC). Now exit the **General form** box and after following the instructions of step 6 of Section 6.2.2 and staying clear of the mechanism. **Implement** the algorithm. You should see a considerable reduction in jittering (i.e. numerical noise) from your previous controller's operation. Why should this filter be implemented to the output as opposed to the input of the controller? Remember it is the numerical noise that we are trying to attenuate not sensor noise. Is sensor noise a source of concern with optical shaft encoders as used in the pendulum mechanism? Execute and plot a step response as per steps 4, 6,7,8, 9, and 10 of Section 6.2.2.

4. Repeat the procedures in steps 2 & 3 reducing the time constant of the filter progressively (e.g. to 75% of its existing value each time) from 0.008 seconds keeping sampling time the same. You should note that as the time constant gets smaller the effect of filter fades away. Why?
5. Repeat the procedure in steps 2 & 3 increasing the time constant of the filter (e.g. by 150%) progressively from 0.008 seconds keeping the sampling time the same. You should notice that as the time constant of the filter gets larger it starts to affect the underlying dynamic response by increasing the response time to a step input, overshoots and finally instability. Why?

#### 6.4 Pole Placement Controller For Unstable $\theta(s)/x(s)$

For Plant #1, the denominator of  $\theta(s)/x(s)$  has left half plane roots and is stable<sup>1</sup>. The physical interpretation is that the system c.g. is below the pivot so that if the sliding rod were fixed, the mechanism would behave as an ordinary (noninverted) pendulum. In Plant #2, however, the balance masses are raised bringing the c.g. above the pivot and  $\theta(s)/x(s)$  is unstable. Stabilizing and controlling such a system graphically demonstrates the effectiveness of closed loop control.

##### Procedure:

1. Adjust the position of the balance masses to  $l_t = 7.0$  cm being sure to secure them on the threaded rod by counter rotating them. Verify that the donut weights are in place and secure on the sliding rod. Hence place the apparatus in the Plant #2 configuration.
2. Repeat the design and implementation process of Section 6.2, noting the following:
  - a) You may assume that the PD controller designed previously (Section 6.2.1) has similar natural frequency and damping (you may verify this by recalculating  $m^*$  and then  $\omega_n$  and  $\zeta$  for the new mass properties) when applied here and hence you may use the same  $k_p$  &  $k_d$ .

<sup>1</sup>Recall however that  $\theta(s)/F(s)$  is unstable.

- b) Use the same desired closed loop denominator given by Eq. (6.2-13).
  - c) In initializing the system per step 6 in Section 6.2.2 it will be more difficult to manually balance the pendulum such that you **Reset Controller** then **Implement Algorithm** without the pendulum moving. With a little practice however you should be able to do it.
3. You should add a noise filter to your design per Section 6.3 before implementing.
  4. Compare your step response with that obtained in Section 6.2. How do the closed loop transfer functions compare? (Assume that  $c(s)=1$  throughout the control bandwidth.)
  5. Save your controller configuration via **Save Settings** in the **File** menu.

## 6.5 LQR Control Design

Since the open loop pendulum mechanism (Plant #2 configuration) is both naturally unstable and non-minimum phase it would be difficult (if not impossible) to implement controllers which only use the angle sensor as feedback (e.g. PD control of pendulum angle). In this section, full state feedback control of the pendulum mechanism based upon the linearized model is addressed. LQR synthesis via the Riccati equation solution or numerical synthesis algorithms is used to find the state feedback controller gains which minimize the cost function (scalar control effort):

$$J = \int (x'Qx + Ru^2) dt \quad (6.4-1)$$

Carry out the LQR design as follows:

### Procedure

1. Construct the numerical value of the state space model of Eq. 5.1-16 using the Plant #2 model from Section 6.1 (with variables  $\theta_c$  and  $x_c$  and control force  $F_c$ ).
2. Choose the output vector  $C$  to be a row vector equal to  $[1 \ 0 \ 0 \ 0]$  which defines the pendulum angle to be the single output. Choose  $Q=C'*C$  which means that pendulum angle error as the error to be minimized subject to the control effort cost.
3. Choose the control weighting function  $R$  to range between  $R = 5.0$  to  $R = 50$ . Remember the smaller the value of  $R$  (control effort weight) the higher the value of gains and potentially the higher the bandwidth. However too small a value of  $R$  can cause unacceptable demand on the actuator and almost always causes excessive noise in the system. Can you explain why?
4. Perform LQR synthesis via the Riccati equation or otherwise to obtain a state feedback gain vector  $K$ , given by  $[k_1 \ k_2 \ k_3 \ k_4]$ , such that the feedback law is:

$$u(t) = -K \mathbf{x}(t)$$

For each value of R determine the gain vector K. Here, since the state space plant has already been scaled in controller units, the values for K are themselves scaled for direct input in the real time controller.

5. Verify that two of the closed loop poles determined from the eigenvalues of [A-BK] approach the negative real zero of the open loop plant as R reduces. Also, the other two poles increase in magnitude and tend toward 135 and 225 degrees in the complex plane as R is reduced. Is this result consistent with the theory of the asymptotic behavior of LQR controllers? Explain.
8. Now we determine the prefilter gain working on commanded position such that the steady state value of the actual angle corresponds to the commanded angle. Using closed loop transfer function with full state feedback, show that the value of commanded angle position  $\theta_d$  must be pre-scaled by the following expression for zero steady state error:

$$r_c = [-m_1 * g / (k_a * k_s * k_f) - k_x * ((m_1 l_o + m_2 l_c) / (m_1 * k_a)) * k_3 + k_1] * \theta_d \quad (6.4-2)$$

Here the signal  $r_c$  is the output of the prefilter gain in the state feedback control scheme. Remember that  $k_1$  is the gain feedback from the pendulum rod angle position sensor and  $k_3$  is the gain feedback from the sliding rod position sensor.

Determine the value of the quantity  $[-m_1 * g / (k_a * k_s * k_f) - k_x * ((m_1 l_o + m_2 l_c) / (m_1 * k_a)) * k_1 + k_3]$ . Carry out this task for each value of the gain set determined for each value of R. i.e. for each value of R, determine:

$$k_{pf} = [-m_1 * g / (k_a * k_s * k_f) - k_x * ((m_1 l_o + m_2 l_c) / (m_1 * k_a)) * k_3 + k_1] \quad (6.2-8)$$

9. You are should now be ready to implement these state feedback controllers and observe these respective behaviors. Turn off power to the Control Box. Enter the ECP Executive program and select the **Setup** menu. Click on **Control Algorithm** and then choose **Continuous time**, then **State Feedback** and **Setup Algorithm**. Using the state feedback structure enter the scaled prefilter  $k_{pf}$  gain determined in step 9. Enter the scaled values of  $k_1$  (for encoder 1 position),  $k_2$  (for encoder 1 velocity),  $k_3$  (for encoder 2 position) and  $k_4$  (for encoder 2 velocity) for the value of gains determined for R=50. Choose the sampling period  $T_s$  to be 0.00442 seconds. Now manually move the pendulum rod to the vertical position and the sliding rod to the middle position to balance the apparatus (this may take a little practice). Complete the initialization and implementation procedure described in Steps 6 and 7 of Section 6.2.2 keeping your hands and head well away from the mechanism. You should see a stable response. Gently disturb the pendulum rod (not the sliding rod) with a slender blunt object (such as a ruler) and note that as for the other controllers, it attempts to servo back to its original position.
10. If a stable response results, go to the **Data** menu and select *commanded position, encoder 1 & 2 positions* and *control effort* with a sample period of 5 servo cycles. Go to **Jog Position**



under the **Utility** menu and "jog" to the -200 count position. Exit this menu and enter the **Command** menu. Select **Step** and then **Setup**. Choose Closed Loop with step size of 400 counts and a dwell time of 4000 ms with one repetition. Exit this menu and choose **Command** followed by (with **Sample Data** box checked) **Execute**. Plot and print your response data.

11. Repeat steps 9 & 10 for each gain set corresponding to the value of  $R=50$  to 5.0. Note that as  $R$  gets smaller, gains get larger and the response becomes faster. However at higher gains the system appears to make a buzzing sound associated with sensor quantization noise. The system should not be ran for longer than a few seconds in this condition as it is stressful on the drive components. When this noisy condition occurs implement a suitable low pass filter from Section 6.3 so that the "buzzing" is eliminated. Plot your step response for each gain case and note the increase in peak control effort with increased gain.

**NOTE:** When implementing a low pass filter under state feedback control, it is numerically superior to enter the filter coefficients in discrete time form. The Executive software must transform the improper derivative terms (e.g.  $k_2\dot{\theta}(t)$  becomes  $k_2s\theta(s)$ ) to near-equivalent proper ones in the continuous time general form dialog box. (Using the general form is a necessary intermediate step for the user in appending the low pass filter to the state feedback control design). The proper form is necessary for the subsequent bilinear transformation from continuous to discrete time carried out internally by the executive software prior to downloading for real-time execution.

The "near equivalent" forms described above and their bilinear transformation result in discrepancies from the as-designed control characteristics that are not always negligible. (The discrepancies do not occur if a continuous time state feedback control is implemented without going through the general form dialog box). Thus one should always use the discrete time forms when augmenting state feedback control with the low pass filter. The discrete time state feedback control gains are identical to their continuous time counter parts except that the derivative gains  $k_2$  and  $k_4$  are divided by the sample rate prior to their entry (in the discrete time state feedback dialog box).

## 6.6 Tracking Response

In this experiment the tracking response of the pole placement designed controller of Section 6.4 (Plant #2) will be investigated for the various command trajectory options provided on the ECP Executive program.

### Procedure

1. Verify that the pendulum is in the configuration of Plant #2. Enter the Executive program and input the controller gains from Section 6.4 (you may do this quickly by uploading your saved configuration from that experiment) which should include a low-pass noise filter. Follow the procedure of steps 6 and 7 from Section 6.2.2 to initialize and implement your controller. Go to **Jog Position** under the **Utility** menu and "jog" to the -200 count position.
2. Set up the system to execute a *step size* of 400 counts with 2500 ms *dwell* and 1 *repetition*. Execute this trajectory and plot the response. Observe carefully the size of the initial reverse motion, rise and settling times. Save and print this and all plots in this exercise.
3. Now under **Trajectory Configuration** select **Ramp**. Enter *Ramp size* of 400 counts, *velocity* of 500 counts/s, *dwell time* of 1000 ms, and 2 *repetitions*. Exit and then execute this trajectory. Plot the response as in step 2 and compare the size of the initial reverse motion, rise and settling time with respect to the step input. Is it qualitatively as expected? Now add the plots for *Commanded velocity* and *Encoder 1 velocity* on the left vertical axis and plot all four quantities. Note the rise (almost instantaneous) in commanded velocity for a ramp input.
4. This time select a **Parabolic** trajectory. Enter *Parabolic size* of 400 counts, *velocity* of 2000 counts/s<sup>1</sup>, *acceleration time* of 500 ms, *dwell time* of 1000 ms, *repetition* 1. Exit and then execute this trajectory. Plot the response as in step 3 and compare the response with that of the ramp input. What is the significant difference?
5. Now select a **Sinusoidal** trajectory. Enter *amplitude* of 200 counts, *frequency* of 0.5 Hz, and repetition of 10. Exit and execute this trajectory. Plot the response of *Commanded position* versus *Encoder 1 position*. Note the amplitude ratio and the phase lag at steady state. Repeat for frequencies lower and higher than 0.5 Hz (do not go beyond 6 Hz). Do the responses agree with your expectations?
6. To experimentally evaluate this frequency response go to the **Trajectory Configuration** and this time select **Sine Sweep** followed by **Setup**. Enter *amplitude* of 175 counts *starting frequency* of 0.1 Hz, *ending frequency* of 3 Hz, and *sweep time* of 30 seconds. Execute this trajectory and plot the response of the *Commanded position* versus *Encoder 1 position*. Does the response correspond with your expectations and those noted from the step and the sinusoidal responses?

---

<sup>1</sup>Note that the trajectory generation algorithm is constrained such that in this case, the specified velocity is not reached in the acceleration time, hence the trajectory's shape on the non constant slope is that of connected parabolas. (For a lower velocity value the two parabolas would be connected by a straight line.)

## Appendix A. Generic Control Issues: Nonminimum Phase & Unstable Plants

This appendix discusses some generic control issues and the implications of right half plane poles and zeros to attainable performance and stabilizability.

### A.1 Generic Control Issues

Define the *open loop transfer function*, and the sensitivity and *complimentary sensitivity* functions respectively as [Fre88]:

$$L(s) \triangleq G(s)K(s) \quad (\text{A.1-1})$$

$$S(s) \triangleq \frac{1}{1+L(s)} \quad (\text{A.1-2})$$

$$T(s) \triangleq \frac{L(s)}{1+L(s)} \quad (\text{A.1-3})$$

Note that for the single degree of freedom controller of Figure A.1-1b,  $T(s)$  is identical to the closed loop transfer function  $y_t(s)/r(s)$ .

Consider the scalar system<sup>1</sup> of Figure A.1-2 for which the following are readily obtained:<sup>2</sup>

$$\frac{y_t(s)}{d(s)} = \frac{e(s)}{r(s)} = \frac{\partial G_{cl}(s)}{\partial G(s)} \frac{G(s)}{G_{cl}(s)} = S(s) \quad (\text{A.1-4})$$

$$\frac{y_t(s)}{n(s)} = T(s) \quad (\text{A.1-5})$$

where  $G_{cl}(s) = y_t(s)/r(s)$ .

<sup>1</sup> Most of the scalar properties discussed here have multivariable analogs. As is discussed in Ref. [Fre88], scalar feedback properties are approached in the SIMO case when a given output has a relatively large weighting in the frequency region of interest. This is generally the case, for a design whose objective is close regulation or tracking of the reference trajectory.

<sup>2</sup> The properties discussed in this section and in Sect. A.2 generally hold for other linear controller forms such as state-feedback and return path compensation.

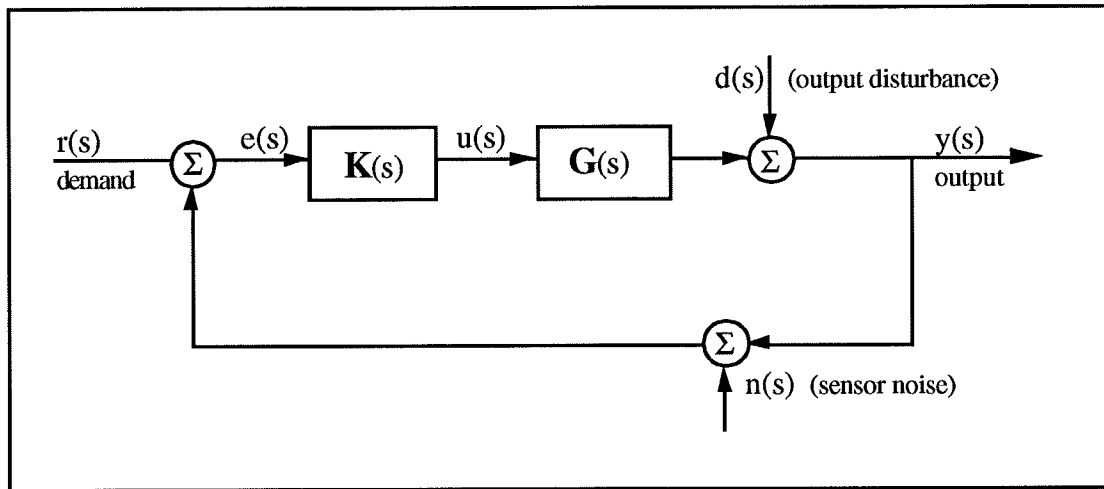


FIGURE A.1-1 SCALAR FEEDBACK SYSTEM

Note that for the open loop system,  $L(s)$ , all quantities equating to  $S$  in Eq.(A.1-4) equal 1. It is therefore necessary that  $S < 1$  in order for feedback to provide improvement in the transmission of output disturbances, in error reduction, and in closed loop transfer function sensitivity<sup>1</sup> to plant changes. Equation (A.1-5) states that  $T$  is the transmittance of sensor noise to the output. It is also well known that  $|1/T|$  evaluated in the open loop crossover frequency neighborhood is a measure of stability. It is therefore desirable to have both  $S$  and  $T$  as small as possible. We have, however that:

$$S + T = 1 \tag{A.1-6}$$

In addition, a relationship due to Bode states that if  $L$  has a pole excess of at least two and is stable, then:

$$\int_0^\infty \ln|S(j\omega)| d\omega = 0 \tag{A.1-7}$$

These equations express fundamental limitations present in scalar feedback systems. Equation (A.1-7) shows that  $|S|$  cannot be made less than 1 over an arbitrarily large frequency range, and that the area of sensitivity reduction ( $|S| < 1$ ) is equal to the area of sensitivity increase ( $|S| > 1$ ) through the semi-infinite frequency range.<sup>2</sup> Eq.(A.1-6) shows that  $|S|$  and  $|T|$  cannot both be made arbitrarily small at the same frequency, and that one being large implies the other is large. Given these constraints, a typical design tradeoff is to make  $|S|$  small at lower frequencies to provide good

<sup>1</sup> A similar expression exists for *robustness* of the closed loop to changes in  $G(s)$ .

<sup>2</sup> The sensitivity reduction/increase tradeoff is more acute - i.e it happens primarily within the bandwidth of  $L$ - than may first appear due to the pole excess in  $L$ .

disturbance rejection and insensitivity to plant parameter changes, and to make  $|T|$  small near and beyond the cutoff frequency to provide stability and insensitivity to sensor noise.

## A.2 Effects of Right Half Plane Poles and Zeroes

The open loop transfer function may be factored as:

$$L(s) = L^*(s)A_z(s)A_p(s), \quad (\text{A.2-1})$$

where:

$$A_z(s) \triangleq \prod_{i=1}^{N_z} A_{z_i}(s) \triangleq \prod_{i=1}^{N_z} \frac{z_i - s}{z_i + s} \quad (\text{RHP zeroes}) \quad (\text{A.2-2})$$

$$A_p(s) \triangleq \prod_{i=1}^{N_p} A_{p_i}(s) \triangleq \prod_{i=1}^{N_p} \frac{s + p_i}{s - p_i} \quad (\text{RHP poles}) \quad (\text{A.1-2-3})$$

Here  $N_z$  and  $N_p$  are the number of right half plane zeroes and poles respectively in  $L(s)$ , and the  $z_i$ 's and  $p_i$ 's have positive real parts. The factor  $A_z$  in  $L(s) = L^*(s)A_z(s)$ , causes phase loss without effecting gain, hence the term "nonminimum phase". Note that  $A_z$  and  $A_p$  may occur due to RHP roots in either  $G$  or  $K$ . For notational purposes, let

$$\Theta_Q(\omega) \triangleq \arg(Q(j\omega)) \quad (\text{A.2-4})$$

for some complex function  $Q$ . The Bode phase characteristics of single terms (as is the case in the experimental pendulum) in  $A_z$  and  $A_p$  are shown in Figure A.2-1

Many effects (generally deleterious) of these RHP roots on the system feedback properties are well known [Hor63, Fre85, Fre88]. Several of these may be interpreted via the Bode phase-gain relationships [Bod45] which imply that associated with any positively (negatively) sloped segment of the gain of stable minimum phase  $L^*$  there is a positive (negative) shift in its phase. These lead to a rule of thumb that a  $20N$  dB/decade negative gain slope in the neighborhood of some frequency  $\omega_0$  results in  $\Theta_L(\omega_0) \approx -90N^\circ$ .

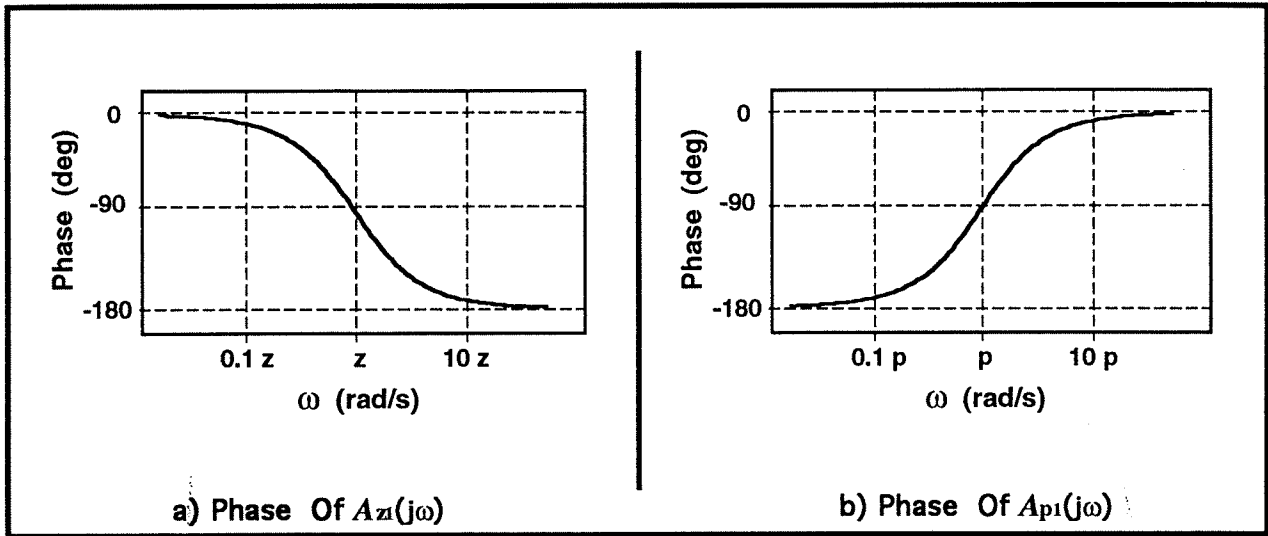


FIGURE A.2-1 Phase Characteristics Of Single-term All-Pass Factors  $A_z(j\omega)$  &  $A_p(j\omega)$

Using this relationship, Eq.(A.2-2), and Nyquist arguments, Horowitz [Hor63] showed<sup>1</sup> that when  $L$  has RHP zeroes and no RHP poles, the maximum attainable crossover frequency<sup>2</sup> for  $L$  lies below the frequency at which  $\Theta_{A_z}(\omega) = -180^\circ$ . We shall call this  $\omega_H$ . This corresponds to a case where  $L$  is only infinitesimally greater than one below the crossover frequency, and where roll-off is at an infinitesimal rate. A more realistic bandwidth limitation is the frequency at which  $\Theta_{A_z} \approx -90^\circ$  which allows for about -15 dB/decade avg. gain slope in  $L$  in the crossover neighborhood (resulting in about  $60^\circ$  phase lag) and  $30^\circ$  phase margin.<sup>3</sup> While lesser gain slopes and stability margins are possible, the resulting systems will have poor stability and general feedback properties.

<sup>1</sup>The argument showing the bandwidth limitation is outlined as follows:

- a) Any RHP zero in  $L(s)$  may be factored as in Eq. A.1-8 thereby isolating the nonminimum phase loss to the all-pass factor  $A_z$ .
- b) Now there are no restrictions on  $L^*(s)$  other than it have the factor  $(z_1 + s)$  in its numerator and be stable. (This preclude among other things an unstable pole-zero cancellation. It can be shown that the bandwidth can be extended by the inclusion of unstable poles but only with very poor resulting feedback properties) Note that  $L^*(s)$  may also have the factor  $(z_1 + s)$  in its denominator and hence the only true restriction is that of stability.
- c) By the Bode phase-gain relationships, in order for  $L^*(s)$  to have negative slope in its gain plot in the neighborhood of some frequency, it must also have negative phase. Applying this relationship to the crossover frequency in  $L^*(s)$ , the negative phase in  $L^*(s)$  adds to that of  $A_z$  (since  $A_z$  has unity gain at all frequencies,  $L(s)$  and  $L^*(s)$  have the same crossover frequency).
- d) Then by Nyquist stability the total phase cannot be less than  $-180^\circ$  for closed loop stability. The limiting case occurs when there is infinitesimally small gain slope in the crossover frequency neighborhood allowing  $180^\circ$  of phase lag from  $A_z$ .

<sup>2</sup> In systems with multiple crossovers, this applies to the initial crossover of  $L(s)$ .

<sup>3</sup>The bandwidth limitations are on the open loop system  $L(s)$ . For reasonable gain slopes (say  $>10$  dB/decade), the closed loop bandwidths will also be close to this range.

It may similarly be shown that a practical lower bandwidth limit (i.e. in the presence of RHP poles) is  $\Theta_{Ap} \approx -90^\circ$ .

Related quantitative results are given in [Fre85, Fre88] where the limitation given by Eq.(A.1-7) is extended to

$$\int_0^\infty \ln|S(i\omega)|d\omega = \pi \sum_{i=1}^{N_p} \text{Re}(p_i) \quad (\text{A.1-12})$$

showing that the addition of RHP poles in  $L$  further exasperate attempts at making  $|S|$  small broadly through frequencies. It is also shown that such poles impose additional limitations on maintaining small  $|T|$ .

Further, as is shown in [Fre85, Fre88], attempts at reducing  $|S|$  at frequencies in which the nonminimum phase lag becomes large, result in a disproportionate increase in  $|S|$  at other frequencies. Thus as a design approaches its NMP bandwidth limit, it pays a high price in terms of available sensitivity reduction.

### A.3 Numerical Values Of Bandwidth Limitations

Based on the discussion of the pervious section, we would expect the practical range of achievable bandwidths,  $\omega_b$ , of  $L(s)$  to fall between the frequencies at which  $\Theta_{Ap} \approx -90^\circ$  and  $\Theta_{Az} \approx -90^\circ$ .<sup>1</sup> From Section 5.2.1, for the experimental pendulum, the RHP zero lies at

$$z = \sqrt{g/l_0} = \sqrt{9.81/0.330} = 5.45 \text{ rad/s} = 0.87 \text{ Hz} \quad (\text{A.3-1})$$

From Figure A.1-2a,  $\Theta_{Az} = -90^\circ$  at  $\omega=z$ . Thus we expect

$$\omega_b \leq 6 \text{ rad/s} = 0.9 \text{ Hz} \quad (\text{A.3-2})$$

The RHP poles depend on the locations of the counter-mass and hence fall within some range. Here we will consider the cases of the two "donut" weights in place on the sliding rod and the balance masses either removed or fully extended (See Section 6.1 for definition of terms). With the masses removed we have:

$$\begin{aligned} J^* &= 0.025 \text{ kg-m}^2 \\ m_1 &= 0.213 \text{ kg} \\ l_0 &= 0.330 \text{ m} \\ m_2 &= 0.785 \text{ kg} \end{aligned}$$

<sup>1</sup>Here we are assuming the unstable pole frequency lies below the zero as described in Section 5.2.2. By the discussion above, no practical controller may be found if this order were reversed.

$$l_c = 0.071 \text{ m}$$

so that by Eq's 5.2-2,-3,-7& -8, the RHP pole lies at  $p=5.1 \text{ rad/s}$ . By the above discussion we would expect

$$\omega_b \geq 5.1 \text{ rad/s} = 0.8 \text{ Hz} \quad (\text{A.3-3})$$

Hence it appears that any stabilizing controller (if one can be found), for the pendulum with counter-balance removed, would have poor feedback properties.

For the case of the counter-mass fully extended at  $l_w = -15 \text{ cm}$ , ("donuts" removed) we have:

$$J^* = 0.047 \text{ kg-m}^2$$

$$m_1 = 0.103 \text{ kg}$$

$$l_o = 0.330 \text{ m}$$

$$m_2 = 1.785 \text{ kg}$$

$$l_c = -0.053 \text{ m}$$

so that again by Eq's 5.2-2,-3,-7& -8, the RHP pole lies at  $p=2.5 \text{ rad/s}$ . Here we would expect

$$\omega_b \geq 2.5 \text{ rad/s} = 0.4 \text{ Hz} \quad (\text{A.3-4})$$

In this case we would expect resulting open loop bandwidths to fall in the range  $0.4 \leq \omega_b \leq 0.9 \text{ Hz}$ .

#### Bibliography for Appendix A:

- [Bod45] Bode, H. W., *Network Analysis and Feedback Amplifier Design*, Van Nostrand, Princeton New Jersey, 1945
- [Fre85] Freudenberg, J. S. and Looze, D. P., "Right Half Plane Poles and Zeros And Design Tradeoffs In Feedback Systems", *IEEE Transactions on Automatic Control*, June 1985
- [Fre88] Freudenberg, J., S. and Looze, D., P., *Frequency Domain Properties of Scalar and Multivariable Feedback Systems*, Springer-Verlag, Heidelberg, 1988
- [Hor63] Horowitz, I. M., *Synthesis of Feedback Systems*, Academic Press, New York, 1963





## 6i. Instructor's Supplement To Experiments

This is the instructor's supplement to the experiments described in Chapter 6 of the main manual. It contains some of the expected experimental results as well as suggestions for further experiments and details on certain analytical developments. There are many dynamics and controls topics that relate to these experiments which are not explored in the manual, but which the instructor may wish to bring out. The methodologies presented here demonstrate some fundamental principles and behaviors, but are not represented to be necessarily superlative practical control solutions. They do in the least however, give the undergraduate student a hands-on introduction to control implementation, and the more advanced user a working orientation to the system from which much more general topics may be explored.

This supplement is organized with section numbers consistent with those of the six experiments in the main manual body. In addition to describing experimental results, Section 6.1i contains numerical models of two plant configurations. Suggestions for further experiments are given in Section 6.6i. Appendix Ai contains some MATLAB® scripts useful in building numerical plant models, performing certain steps in the experiments and simulating the nonlinear dynamics.

### Note to instructors and advanced users:

If you or your students synthesize and implement a control structure that is of particular academic or instructional interest, please share it with us! Pending ECP review it may be included in a future revision of this manual with credits of course to the originator.

### 6.1i System Identification

The MATLAB® script *penplnt.m* may be used to provide the A,B & C matrices for the transfer functions and state space representation of the two plants. Using units scaled to controller counts the following are obtained (see *penplnt.m* for expressions incorporating the scaling constants)

#### Plant #1:

$$m_1 = 0.213, m_2 = 1.785, l_w = -0.1384$$

$$A = \begin{bmatrix} 0 & 1.0 & 0 & 0 \\ -18.52 & 0 & 2.422 & 0 \\ 0 & 0 & 0 & 1.0 \\ 313.9 & 0 & -0.15.76 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ -798.8 \\ 0 \\ 15000 \end{bmatrix}$$

$$C = [1 \quad 0 \quad 0 \quad 0]$$

$$\theta_c(s) / F_c(s) = -798.8(s^2 - 5.451^2) / (s^2 + 6.689^2)(s^2 - 3.236^2)$$

$$x_c(s) / F_c(s) = 15000(s^2 + 1.342^2) / (s^2 + 6.689^2)(s^2 - 3.236^2)$$

$$\theta_c(s) / x_c(s) = -0.05325(s^2 - 5.451^2) / (s^2 + 1.342^2)$$

**Plant #2:**

$$m_1 = 0.213, m_2 = 1.785, l_w = -0.1084$$

$$A = \begin{bmatrix} 0 & 1.0 & 0 & 0 \\ -0.14.19 & 0 & 2.914 & 0 \\ 0 & 0 & 0 & 1.0 \\ 285.7 & 0 & -18.96 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ -961.5 \\ 0 \\ 16060 \end{bmatrix}$$

$$C = [1 \quad 0 \quad 0 \quad 0]$$

$$\theta_c(s) / F_c(s) = -961.5(s^2 - 5.451^2) / (s^2 + 6.748^2)(s^2 - 3.518^2)$$

$$x_c(s) / F_c(s) = 16060(s^2 - 1.708^2) / (s^2 + 6.748^2)(s^2 - 3.518^2)$$

$$\theta_c(s) / x_c(s) = -0.05987(s^2 - 5.451^2) / (s^2 - 1.708^2)$$

Which has real zeros at  $\pm 5.5$  rad/s ( $\theta_c(s) / F_c(s)$ ), imaginary poles at  $\pm 6.7i$  & real poles at  $\pm 3.5$ .

**Confirmation of  $k_x$  &  $k_a$** 

This procedure is straight forward. However, it is important that the students are aware of sensor (and other) scaling factors which exists in all practical control systems. The values determined for both  $k_x$  and  $k_a$  should agree closely with those provided in the Table 6.1-2.

### Confirmation of $J^*$

Provided  $l_c$  is known reasonably accurately, the simple "non-inverted" undamped pendulum equation of motion leads the expression for  $J_o^*$  (actually  $J_o^* + m_1 l_o^2$ ) in terms of the period of oscillations  $T$ . It is possible to measure  $J_{oe}$  more directly using the specific  $m_{w1}$ ,  $m_{w2}$  and  $l_w$  for a particular configuration. In cases where the system c.g. is close to the pivot (i.e.  $(m_1 l_o + m_2 l_c)$  is small) however, this method loses accuracy due to measurement uncertainty and friction. When  $(m_1 l_o + m_2 l_c) = 0$  in fact the inertia is indeterminate by this method. By using the configuration described for this experiment relatively accurate inertia measurement should result.

*Ans:*

$$J = m l_{cg} g (T / (2\pi))^2 (= J_{oe} \text{ by defined notation})$$

with:  $m = 0.998 \text{ kg}$

$$l_{cg} = 0.126 \text{ m}$$

$$T = 1.235 \text{ s (measured)}$$

so that:

$$J = 0.0478 \text{ kg-m}^2$$

now:

$$J_o^* = J - m_1 l_o^2 = 0.0246 \text{ kg-m}^2$$

## 6.2 Successive Loop Closure / Pole Placement Design

The Matlab® script *polplc.m* may be used to generate the control parameters discussed in this section.

### 6.2.1 PD Control Of The Sliding Rod

Step 1)  $k_{hw} = 2088 \text{ N/m}$ ,  $m^* = 0.139 \text{ kg}$ .

Step 2)  $k_p = 0.26$ ,  $k_d = 0.084$

Step 7) The plots should appear as a classic critically damped form with rise time of about 50 ms – see Figure 6.2i-1. Note that due to drive saturation effects the system will largely not respond to the derivative action by the numerator of the closed loop transfer function in attempting to follow the discontinuous step input. Thus the response will appear more like one of the system:

$$c(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

For inputs without discontinuity in rate however, the response will appear as one for the system of Eq. 6.2-9.

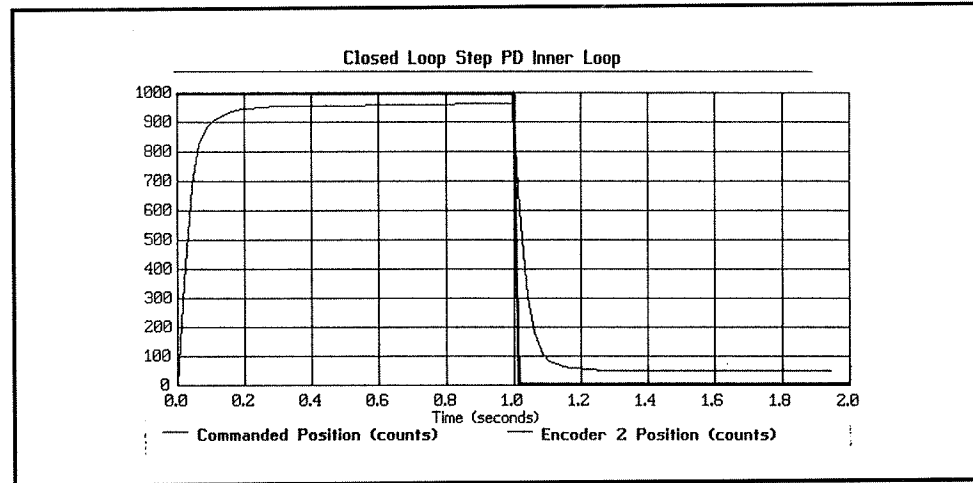


Figure 6.2i-1 Step Response of PD Controlled Sliding Rod

The simplification of Figure 6.2-1 provides an intuitive basis for PD design and allows classical techniques to be employed. In applying the resulting PD control to the higher fidelity linear model of Eq. 5.1-20 two of the closed loop poles are very near the open loop zeros resulting in a "net" closed loop transfer function very much like Eq 6.2-9. The difference between simulated step responses of both models under the designed PD controller is indistinguishable.

6.2.2 Pole Placement Control of  $\theta(s)/x(s)$

The pole placement controller for  $\theta_c(s) / x_c(s)$  (Plant #1) is

$$S = 45.97s + 93.99$$

$$R = 3.448s + 20.71$$

with prefilter gain:

$$t_o = k_{pfc} = s_o - (m_1 * l_o + m_2 * l_c) * k_x * r_o / m_1 / k_a = 117.6$$

where  $s_o$  and  $r_o$  are as implied in Eq. (6.2-12).

The experimental step response is shown in Figure 6.2i-2<sup>1</sup>. Note the close agreement with the simulation of the same controller  $S(s)/R(s)$  applied to the plant  $\theta(s)/x(s)$  shown in Figure 6.2i-3.

<sup>1</sup>The offset at the beginning and dwell of the step will vary depending on how well the system is initialized. By closely following the initialization procedure given in step 6 of Section 6.2.2 small initialization errors as shown in Figure 6.2i-2 are obtainable.

Note that by putting  $S(s)$  in the prefilter numerator (which is readily supported by the General Form controller) we could have effectively put the controller  $S(s)/R(s)$  in the forward path and would have obtained the same closed loop poles. This results in considerably more lead in the tracking response with faster rise time and more overshoot.

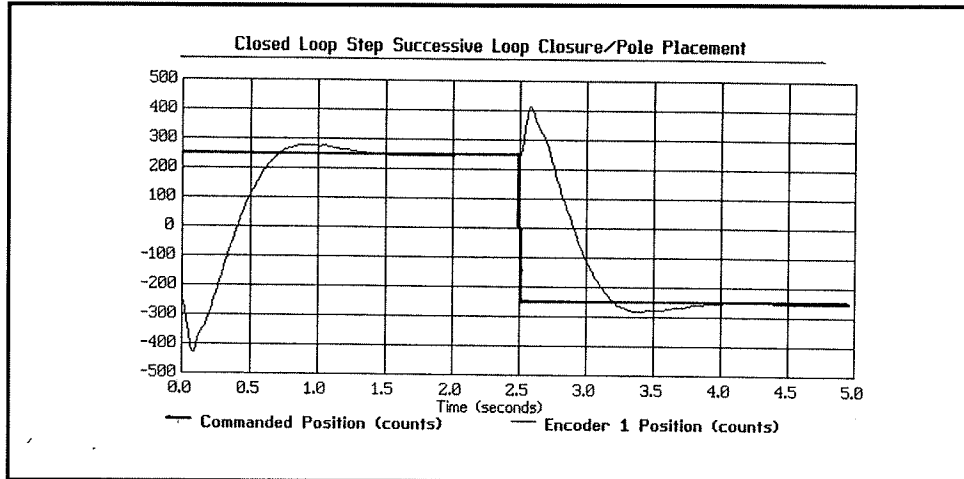


Figure 6.2i-2 Experimental Step Response of Pole Placement Controlled System

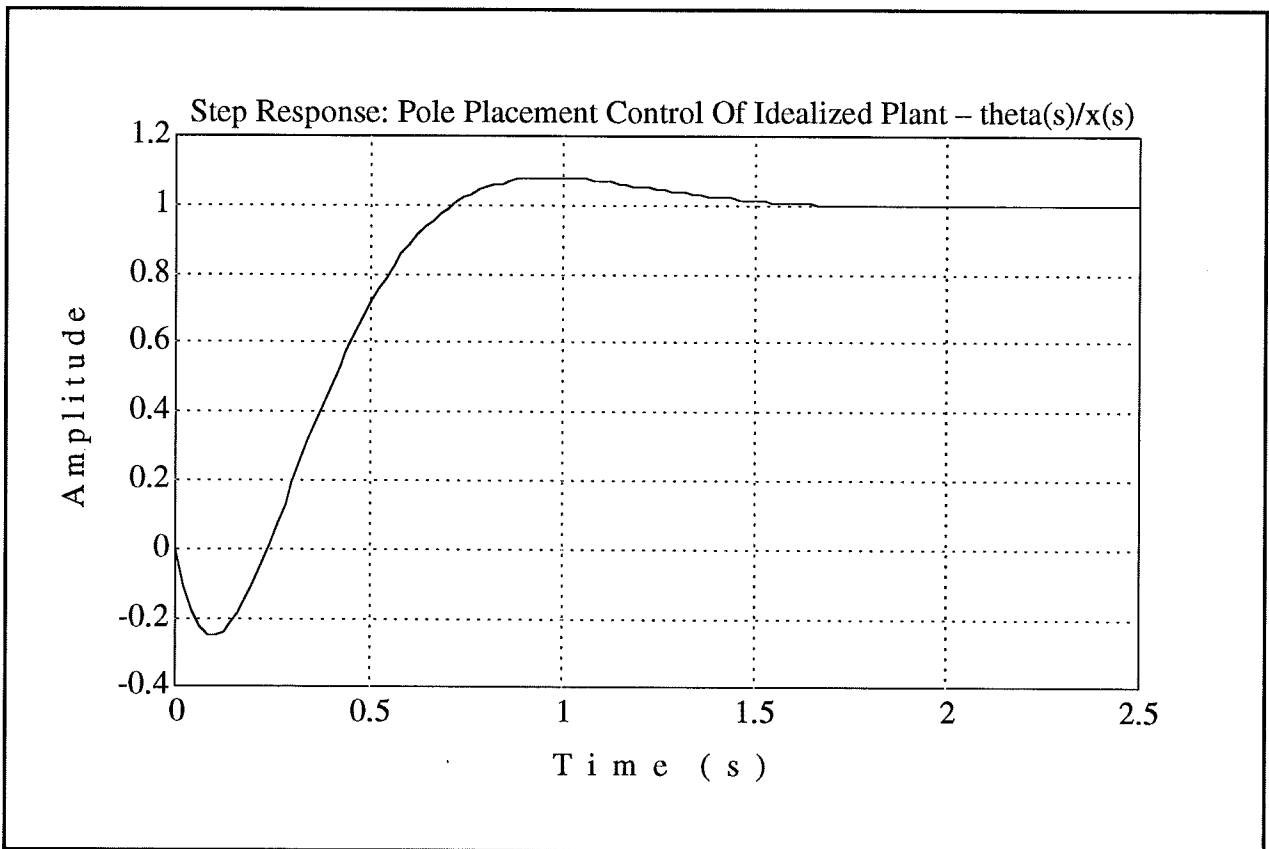


Figure 6.2i-3 Simulated Step Response of Pole Placement Controlled System

### Additional Topics

1. Plot other trajectories (e.g., 800 step ramp, 500 counts/sec starting @ -400 counts via **Jog Position**) to characterize tracking performance. Such a plot is shown in Figure 6.2i-4. Note that any prolonged testing should be with the low pass filter of the next section in place. This is to preclude excessive stress on the drive components due to noise on the control effort.
2. Establish gain margin by slowly increasing  $h_o$  and perturbing system with a safe non sharp object. The system becomes unstable for gains above about  $h_o = 1.25$  (i.e. 25% GM – a relatively low value). The analytical gain margin of the idealized controller/plant is about 40%. The difference in these values is attributable to phase loss due to sampling effects and finite bandwidth of the inner loop.
3. This experiment can be performed in the context of discrete time control using a suitable (e.g. Tustin (bilinear)) transformation for  $S(s)/R(s)$ , and backwards difference for  $k_p+k_d s$  ( for the early PD design, in the discrete time PD control dialog box simply enter  $k_{d(\text{discrete})} = k_{d(\text{continuous})}/T_s$ ). The experimental procedure follows as described in the student's section except control parameters are entered with **discrete time** checked in the **Setup Control Algorithm** dialog box. Note that due to relatively low stability margins in this design, the sampling rate should be kept relatively high, say  $T_s \leq 0.00884$ .
4. The method of applying a high bandwidth inner loop to effect the pseudo plant  $\theta(s)/x(s)$  may be used to study a variety of control design methodologies for closure of the outer loop (e.g. Bode design, linear quadratic,  $H_\infty$ ,  $\mu$ -synthesis, or QFT). The outer loop controller may be either a forward path cascade or return path/prefilter type. State space controllers are easily expressed in one of these forms.

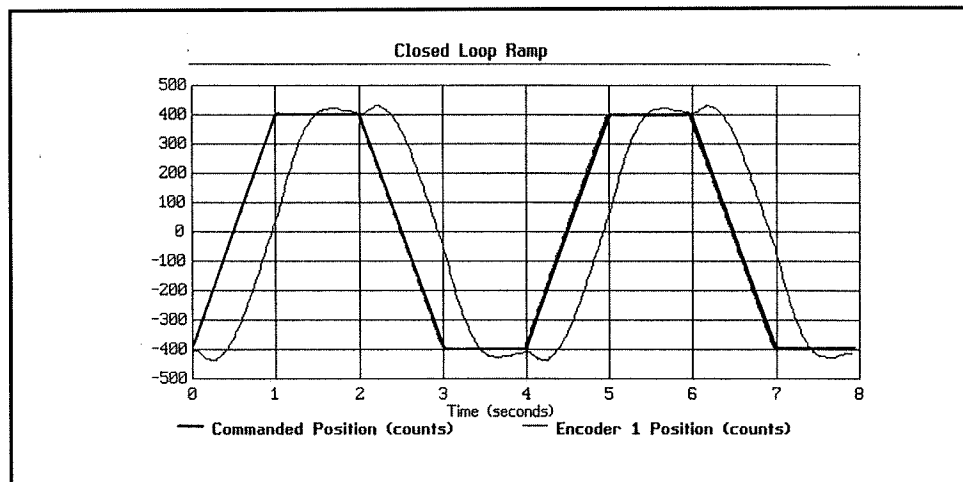


Figure 6.2i-4 Experimental Ramp Tracking Response

### 6.3i Low Pass Filter Augmentation Of Controller

The answers to the questions in steps 4 and 5 are straightforward – i.e. reducing the time constant tends to put the filter bandwidth into and beyond the noise spectrum and therefore becomes ineffective in noise reduction. Increasing the time constant imparts greater phase loss to the system, eventually resulting in system instability. Noise attenuation without noticeable destabilizing effects are achieved for  $\tau$  in the range of .002 to .010 sec.

The following procedure is similar to Section 6.3 but poses the low pass filter exercise in the context of discrete time control. This text may be given to students in lieu of that section if it is more suitable to the particular course of study.

#### Procedure For Digital Implementation

1. Verify that a first order low pass filter represented by the following transfer function  $G(s) = 1/(\tau s + 1)$  may be implemented in discrete time (via the *backward difference* mapping) as:

$$G(z) = (T_s / (T_s + \tau)) / (1 - \tau / (T_s + \tau) * z^{-1})$$

where  $\tau$  is the filter time constant in sec.,  $T_s$  is the sampling period in sec. and  $z^{-1}$  is the unit delay operator in frequency domain. Confirm that the dc gain of this discrete time filter is also unity.

2. Choose  $T_s = 0.00442$ , and determine the cutoff frequency of the filter given by:

$$G(z) = 0.333 / (1 - 0.667 * z^{-1})$$

How does this frequency compare to the magnitude of the closed loop poles of the design in the previous section?

3. To implement this filter as a cascade to the output of the control blocks used in Section 6.2, we must use the **General Form** structure within the **Setup Control Algorithm** Dialog Box. Enter the ECP Executive program and in **Setup Control Algorithm**, enter the sampling period  $T_s = 0.00442$  and make sure that **Discrete Time** is selected. Enter the ECP Executive program and input control coefficients in the **General Form** box from the design of Section 6.2 (you may do this quickly by uploading your saved configuration from the last experiment via **Load Settings** in the **File** menu – make sure however that you choose  $T_s = 0.00442$ ) Now in the discrete time **General Form** box, multiply the existing values of  $E_0$  and  $E_1$  by 0.333 and input  $G_1 = -0.667$ . This corresponds to the implementation of the above first order low pass filter to the output of the control action from the PD inner loop (i.e. immediately prior to the control



signal output from the controller and being put on the DAC). Now exit the **General form** box and after following the instructions of step 6 of Section 6.2.2 and staying clear of the mechanism, Implement the algorithm. You should see a considerable reduction in jittering (i.e. numerical noise) from your previous controller's operation. Why should this filter be implemented to the output as opposed to the input of the controller? Remember it is the numerical noise that we are trying to attenuate not sensor noise. Is sensor noise a source of concern with optical shaft encoders as used in the pendulum mechanism? Execute and plot a step response as per steps 4, 8, 9, and 10 of Section 6.2.2.

4. Repeat the procedures in steps 2 & 3 reducing the time constant of the filter progressively (e.g. to 75% of its existing value each time) from 0.00884 seconds keeping sampling time the same. You should note that as the time constant gets smaller the effect of filter fades away. Why?
5. Repeat the procedure in steps 2 & 3 increasing the time constant of the filter progressively (e.g. by 150%) from 0.00884 seconds keeping the sampling time the same. You should notice that as the time constant of the filter gets larger it starts to affect the underlying dynamic response by increasing the response time to a step input, overshoots and finally instability. Why?

End of Procedure

#### **6.4i Pole Placement Controller For Unstable $\theta(s)/x(s)$**

The pole placement controller for the plant  $\theta_c(s) / x_c(s)$  (Plant #2) is:

$$S = 51.02s + 144.48$$

$$R = 4.057s + 24.36$$

with prefilter gain:

$$t_o = k_{pfc} = 104.6$$

The step response here should appear nearly identical to that from Section 6.2. The ideal closed loop transfer functions are identical.

#### Additional Topics

The four additional topics described in Section 6.2i of course apply here as well. The experimentally measured gain margin should be about 15%. The analytical gain margin for the idealized system is about 30%. This reduced margin is implied by the phase/gain relationships involving nonminimum phase and unstable plants as discussed in Appendix A.

### 6.5i LQR Control Design

The Matlab® script *lq.m* may be used to generate the state feedback gains.

For the range of R values suggested, the table below shows the state feedback gains computed via *lq.m*. These gains are scaled for direct input to the real-time controller via the Executive program.

R	k <sub>1</sub> (pendulum angle position gain)	k <sub>2</sub> (pendulum angle velocity gain)	k <sub>3</sub> (sliding rod position gain)	k <sub>4</sub> (sliding rod velocity gain)
5	0.6007	0.2187	0.0814	0.0229
10	0.4355	0.1594	0.0603	0.01547
20	0.3179	0.1167	0.0450	0.00879
50	0.2128	0.0780	0.0307	0.00616

Using the above value of feedback gains, the prefilter gains are:

R	$k_{pf} = [-m_1 * g / (k_a * k_s * k_f) - k_x * ((m_1 l_o + m_2 l_c) / (m_1 * k_a)) * k_3 + k_1]$
5	0.4476
10	0.3168
20	0.2245
50	0.1428

Also the poles of the closed loop system which are the eigenvalues of  $[A-BK]$  may be obtained to investigate the "asymptotic properties of LQR controllers as the value of R reduces (e.g. see Kwakernaak and Sivan, "Linear Optimal Control Systems", Wiley and Sons, 1972). The matrices  $w$  and  $z$  in *lq.m* provide the natural frequency and damping ratio of these closed loop state feedback poles which can then be plotted and compared with the location of the open loop zeros.

The experimental step response is shown in Figure 6.5i-1 [R=20].

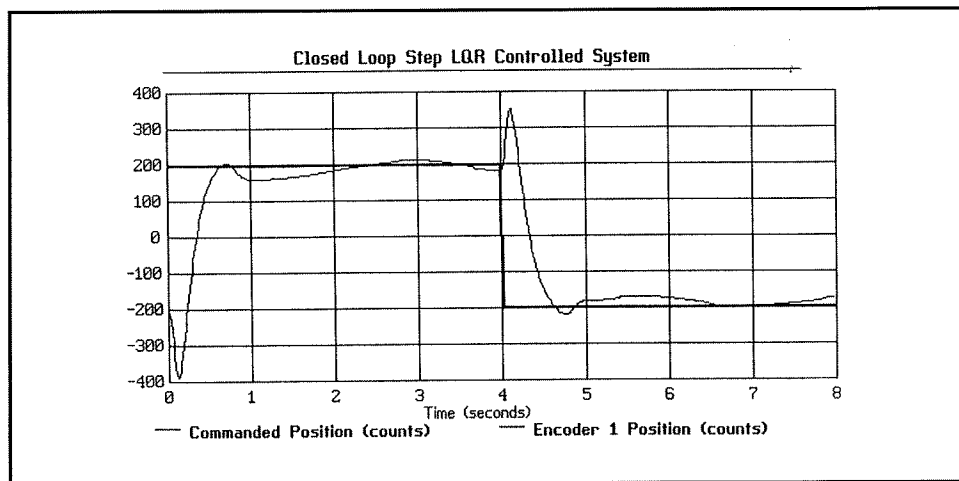


Figure 6.5i-1 Step Response of LQR Controlled System

The gain margin may also be investigated here experimentally. Theoretically the LQR controllers have infinite gain margin as is indicated by their asymptotic properties. It may be demonstrated experimentally that a gain margin of 2x to 5x exists for the range of controllers above. Instability eventually occurs as a result of unmodeled phase lags due of data sampling and the noise filter. If such experiments are carried out it should be with strict supervision as the resulting high gain behavior may cause rapid and potentially violent motion of the plant and could constitute a human safety and equipment hazard. In all such cases the low pass filter must be included to prevent damage to the drive components due to actuator noise. Here and in all operational cases, all people must stay well clear and out of the plane of motion of the pendulum and sliding rod. It is recommended that any such experiments be performed as a demonstration only by an operator that is very familiar with the system and that the system be shut down immediately (red "OFF" button at the front of the controller box) at the onset of instability. ECP shall not be responsible for human safety or the damage to equipment due to running the system with excessively high gains.

The script *nonlsim.m* (which calls *v.m*) may be used for the nonlinear simulation of the plant response to a step input under linear state feedback control. Each time any plant parameters or the state feedback gains are changed, the new gain matrix  $G$  (generated by *lq.m*) and plant parameters should be edited into both *nonlsim.m* and *v.m*.

The procedure for implementing discrete time full state feedback control is nearly identical to that described for continuous time. When inputting the rate feedback gains  $k_2$  and  $k_4$  in the *Discrete Time State Feedback* window you must first divide them by  $T_s$ .

## 6.6i Tracking Control

The step response for the pole placement controlled system is essentially identical to that shown previously in Figure 6.2i-2. The transient response for this system is always constrained by the non-minimum phase behavior (i.e. faster step response also means a larger initial reverse motion). Note that Fourier spectrum of a step decays rather gradually – proportional to  $\omega^{-1}$  – which means that higher frequency poles of the system can be significantly excited by the step (e.g. overshoots).

The ramp, shown in Figure 6.2i-4 is a less abrupt demand relative to the step input. Its spectrum decays as  $\omega^{-2}$ , which means the higher frequency poles of the system are not excited with the same energy as with the step. The parabolic spectrum is proportional to  $\omega^{-3}$ , and for the cubic in time, the proportionality is to  $\omega^{-4}$ . As a result, the higher order polynomials of time are much more suitable if large under and overshoots are to be avoided. Figure 6.6i-1 shows the closed loop cubic response for the 800 increments in position with control effort also plotted. Relative to the step response, the students should note the size of the initial undershoot and the subsequent overshoot. The control effort may also be seen to be sharply reduced from the step response case.

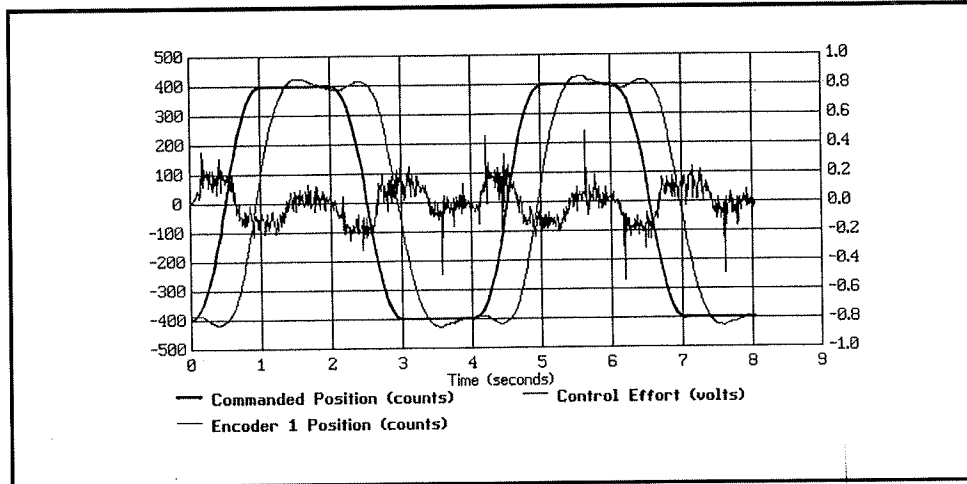


Figure 6.6i-1 Closed Loop Cubic Response

Figure 6.6i-2 shows the frequency response at 0.5 Hz . It is instructive to have the students measure the phase lag from this graph and compare it to that of a theoretical linearized model under closed loop control with the same set of gains. They should note the extra phase lag relative to the more common minimum phase systems of the same order and pole excess.

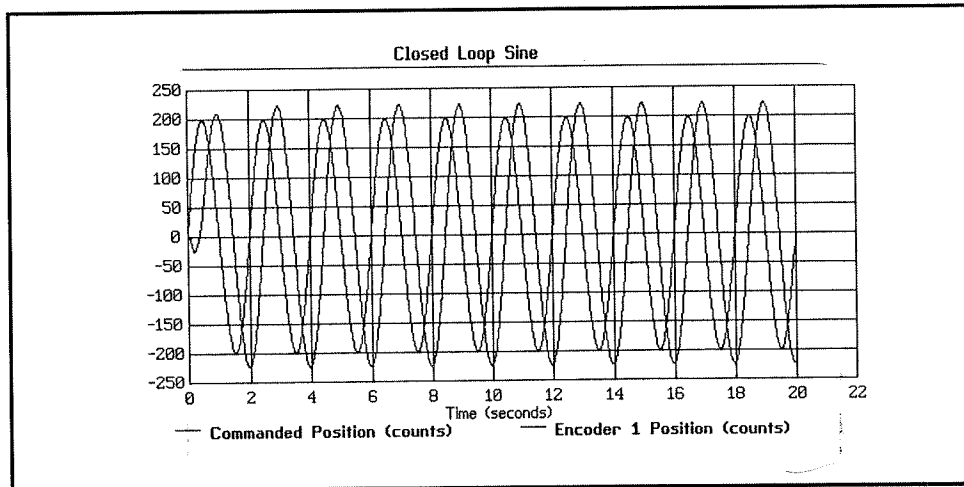


Figure 6.6i-2 Closed Loop Sinusoidal Response 0.5 Hz

The sine sweep is shown in Figure 6.6i-3 . Note that the break frequency is approximately 0.75 Hz. The students should be able to confirm that this value matches the predicted theoretical value by plotting the Bode diagram for the linearized version of the plant under the corresponding set of control gains.

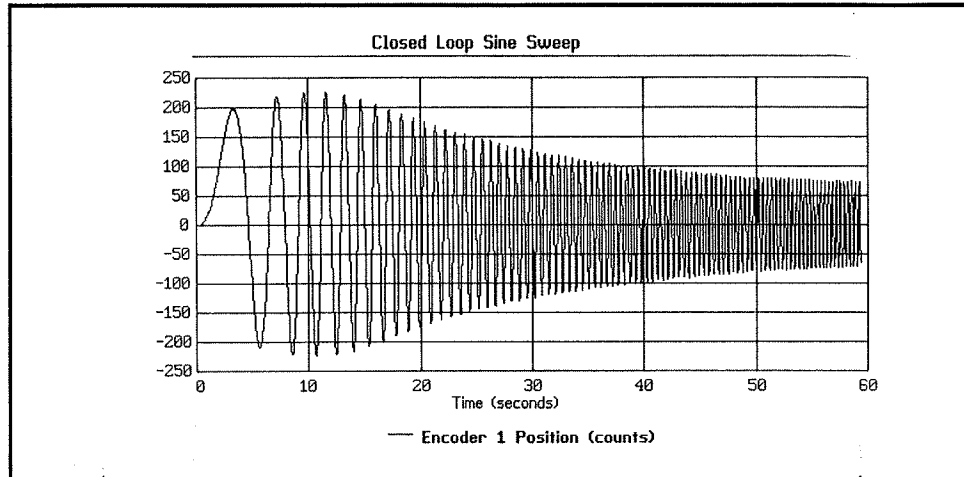


Figure 6.4i-3 Closed Loop Sine Sweep 0.1 to 3 Hz

### 6.7i Suggestions For Other Experiments

The following experiments are readily supported by the ECP control system and inverted pendulum apparatus. Still others are possible.

#### 1) Digital Control:

- a) Demonstrate limitations of continuous time analysis & design assumptions by increasing  $T_s$  on a continuous time designed controller until performance is visibly altered and/or instability is imminent. (usually seen as output oscillation).
- b) Demonstrate that a significantly lower  $T_s$  can be accommodated by designing in discrete rather than time.
- c) Demonstrate digital controller implementation by simulating a controller response to an input then implementing the controller and seeing the same. This may be accomplished via the general controller form by setting feedback or "downstream" gains to zero. Then perform a closed loop move (e.g. step response) while collecting data at the appropriate node (See Figure 4.1-1).<sup>1</sup>

#### 2) System Robustness:

- a) Characterize a controller designed for the nominal plant, then vary the plant mass properties (changes both plant gain & frequencies) and test system performance.

<sup>1</sup>It is a good idea to do this check on any complicated controller before implementing it.

- b) Continue changing mass properties until instability is approached.
- c) Design robust controllers and test comparatively with nonrobust designs.

3) Disturbance Rejection:

Controllers may be tested for their disturbance rejection capability with the disturbance being introduced manually or by apparatus. One such possible device might be a small motor (e.g. battery operated) temporarily attached to the (nominally) horizontal plate at the pivot and revolving an eccentric inertia. (Note that ECP is not responsible for the equipment or human safety of any such setup. The guidelines of Section 2.3 must be followed at all times).

4) Advanced Control:

The general control form supports virtually any linear controller up to and including seventh order (e.g. observer based, LQG/LTR,  $H_\infty$ , QFT). One approach is to close a high bandwidth inner loop about  $x(s)/F(s)$  as done in Experiment #2, then design some advanced controller for control of  $\theta(s)/x(s)$ . The data acquisition, trajectory generation, and file management features of the system allow rapid implementation and characterization.

5) Noninverted Pendulum:

By placing the apparatus upside down, (See e.g. Section 6.1 – pendulum frequency measurement) the dynamics become that of a noninverted pendulum. The derivation of the plant model and controls experiments given here may be followed to study the control of this distinct system. All users should take extra care to assure that the mechanism is properly supported and not scratched during such tests.

Closing Note:

The control designs given here are in no sense optimized, but rather are designed to demonstrate several fundamental principles of control. The user is invited to implement other designs or invent other approaches and share them with us at ECP.

## Appendix Ai. Useful Scripts

Listed in this appendix are Matlab® scripts and functions for building the numerical plant models; designing the PD / pole placement and LQR controllers; and nonlinear simulation from Chapter 6. These are not represented as being numerically or methodologically optimal, but may be useful to some users.

### A.1i Plant Model Builder

```
%Penplant.m
% Initialization file for numerical value of plant & control hardware
parameters
% Also includes a set of vectors and matrices for the lqr design of state
feedback controller
% This .m file must be executed prior to running the function lq(...).

%% INITIALIZE
% The following user-specified values vary with plant configuration
mw1=.110; % Mass of Donut weights, set = zero if weights not used (kg)
mw2=1.00; % Balance mass (kg)
lw=-0.1084; % Signed distance from pivot to c.g. of balance mass (m)

% Misc data required for other scripts - Change As Appropriate
ts=0.00442; % Sampling time (s)
d01=[0;0;0;0]; %Initial disturbance used in nonlsim function for
simulation only
% Vectors and matrices used for lqr design
R=[2 5 10 20 50]; % control weighting for lqr design used in lq function
C=[1,0,0,0]; % C vector in state-space model (output is angle)
Q=C'*C; % Q matrix for lqr design used in lq function

% Fixed values:
g=9.807; % (m/s^2)
lo=0.330; %pendulum rod length (m)
lco= 0.0711; %lc when mw2=0 (m)
m1o=.103; % Mass of sliding rod without donut weights (kg)
m2o=0.785; % Mass of pendulum assembly minus m1 and minus balance mass (kg)
Jos=.0246; %Polar moment of inertia about the pivot of pend ass'y minus
sliding rod and balance weight (kg-m^2)
% Scale factors of hardware (sensors and actuator and digital controller)
kx=50200; % Sliding rod scale factor (encoder counts/m)
ka=2546; % Angle encoder scale factor (encoder counts/radian)
kf=0.0013; % Combined DAC amplifier scale factor (N/increment of DAC)
ks=32; % Controller card firmware scale factor (increments/increments)

% Derived values
m1=m1o+mw1;
m2=m2o+mw2;
```

```

lc=(m2o*lc0+m2*lw)/m2;          %cg of the pendulum assembly minus the sliding
rod(m)
Joe=Jos+m1*lo^2+m2*lw^2;        %Polar moment of inertia of complete pend ass'y
about the pivot (kg-m^2)
Js=Jos+m2*lw^2;                %This is J*(=Joe-m1*lo^2) in the theory section (kg
m^2)

%% BUILD PLANTS
% Use The Following For Plant in The State Space Form:
% x = [angle; angular velocity; position; velocity]
%          xdot = Ax + Bu;
%          y = Cx
% For system in SI units use the following:
a1=[0,1,0,0];
a2=[m2*lc*g/Js,0,m1*g/Js,0];
a3=[0,0,0,1];
a4=[(Js-m2*lc*lo)*g/Js,0,-m1*lo*g/Js,0];
B=[0;-lo;0;(Js+m1*lo^2)/m1]/Js;
%For prefilter gain, kpf, in lqr synthesis
kpf1=- (m1*g);
kpf2=- (m1*lo+m2*lc)/m1;

% For system in units of controller counts use...
a1=[0,1,0,0];
a2=[m2*lc*g/Js,0,m1*g/Js*ka/kx,0];
a3=[0,0,0,1];
a4=[(Js-m2*lc*lo)*g/Js*kx/ka,0,-m1*lo*g/Js,0];
B=[0;-lo*ka;0;(Js+m1*lo^2)/m1*kx]*ks*kf/Js;
%For kpf in lqr synthesis
kpf1=- (m1*g)/ka/kf/ks;
kpf2=-kx*(m1*lo+m2*lc)/m1/ka;

% The following are common to both forms
A=[a1;a2;a3;a4];
C=[1,0,0,0];          % Output vector (output is angle), use C=[0 0 1 0] for x(t)
output
%Use the following for linear quadratic synthesis
Q=C'*C;

% For The Transfer Function theta(s)/F(s) = Na(s)/D(s), Use:
% For Na in SI units use...
%Na = (lo/Js)*[-1 0 g/lo];
% For Na in units of controller counts use...
Na = ka*ks*kf*(lo/Js)*[-1 0 g/lo];
D = [1 0 (m1*lo-m2*lc)*g/Js 0 -m1*g^2/Js];

% For The Transfer Function x(s)/F(s) = Nx(s)/D(s), Use:
% For Nx in SI units use...
%Nx = (1/m1/Js)*[Joe 0 -(m1*lo+m2*lc)*g];
% For Nx in units of controller counts use...
Nx = kx*ks*kf*(1/m1/Js)*[Joe 0 -(m1*lo+m2*lc)*g];
D = [1 0 (m1*lo-m2*lc)*g/Js 0 -m1*g^2/Js];

```



---

```
% For The Transfer Function theta(s)/x(s) = Nax(s)/Dax(s), Use:  
% For Nax in SI units use...  
% Nax=m1*lo/Joe*[-1 0 g/lo];  
% For Nax in units of controller counts use...  
Nax=ka/kx*m1*lo/Joe*[-1 0 g/lo];  
Dax=[1 0 -(m1*lo+m2*lc)*g/Joe];
```

## A.2i Successive Loop PD / Pole Placement

```

PDFOLPLC.M
% This script first solves for the PD gains kp & kd for the "x" control loop,
then S(s)/R(s) for pole placement control of theta
% You must first run PENPLNT.M
% For discrete implementation, input sampling time, ts, if different than
value from PENPLNT.M
%ts=.00442

% INNER LOOP PD CONTROL
% Input desired natural frequency, wn (rad/s), and damping ratio, z
wn=10*2*pi;
z=1;

% Calculate equivalent second mass, m2s, and effective rigid body mass, ms
m2s=Js/lo^2;
ms=m1*m2s/(m1+m2s);

% Calculate kp & kd.
khw=kx*ks*kf;
kp=wn^2*ms/khw
kd=2*z*sqrt(ms*kp*khw)/khw
% For discrete implementation:
kdd=kd/ts

% OUTER LOOP POLE PLACEMENT CONTROL
% Input desired CL poles, pi to construct desired closed loop denom., Dcl:
p1=-pi-pi*i;
p2=-pi+pi*i;
p3=-3*pi;
Dcl=poly([p1;p2;p3]);

% Solve Diophantine Equation for S(s) & R(s) via Sylvester matrix
SYLVa=toeplitz([Dax zeros(1,1)],zeros(1,2));
SYLVb=toeplitz([Nax zeros(1,1)],zeros(1,2));
SYLV=[SYLVa SYLVb];
SR=SYLV\Dcl';
S=SR(3:4)';
R=SR(1:2)';

% Calculate prefilter gain, kpf
kpf=S(2)-(m1*lo+m2*lc)*kx*R(2)/m1/ka

% For discrete implementation use Sd(z) & Rd(z)
[Sd,Rd]=c2dm(S,R,ts,'tustin')

%Plot simulation of idealized (c(s)=1) closed loop system
t=0:.02:2.5;
Ncli=kpf*Nax;
Dcli=conv(Nax,S)+conv(Dax,R);
stepi=step(Ncli,Dcli,t);
plot(t,stepi),grid,pause

```

### A.3i LQR Synthesis

```

function [k,kd,G,kpf,w,z,Pcl]=lq(A,B,Q,R,kpf1,kpf2,ts)
% k: state feedback controller gains;
% w: natural frequency of closed-loop poles;
% z: damping of closed-loop poles;
% Open loop system:
%
%
%
% Control law:
%
%
% Cost term:
%
% Row i of k,w, and z corresponds with element i of R;
% k is a matrix whose rows are lqr design gains scaled suitable for direct
% insertion in continuous time ECP state space control dialog box
% G is the same as k with columns swapped suitable for the nonlsim function
% kd is scaled for insertion in discrete time ECP state space dialog box
% w & z are the nat. freq. and damping ratio of respective closed loop poles

n=length(R);
t=0:.01:2;
for i=1:n,
    [k(i,:),s]=lqr(A,B,Q,R(i));
    Ac=A-B*k(i,:);
    [waux,zaux]=damp(Ac);
    w(i,:)=waux';
    z(i,:)=zaux';
    Pcl(i,:)=eig(Ac)'; %closed loop poles
    kpf(i)=kpf1+k(i,1)+kpf2*k(i,3);
    stepcl(:,i)=kpf(i)*step(Ac,B,[1 0 0 0],0,1,t); %step response
end;
kd= k;
kd(:, [2,4])=kd(:, [2,4])/ts;
G=zeros(k);
G=[k(:,3) k(:,1) k(:,4) k(:,2)];
plot(t,stepcl)

```

#### A.4i Nonlinear Simulation

```

% NONLSIM.M file:
% Command file for the nonlinear simulation of the inverted pendulum.
% v = name of the file where the nonlinear equations are defined.
% y = matrix of state trajectories; each column one state:
%   y(:,1)=position, y(:,2)=angle, y(:,3)=velocity, y(:,4)=angular veloc.
% u = input of DAC ( control force).
% G(i,:) = (row vector) controller gains for a specific closed-loop
%          characteristic polynomial. the state gain must be [x theta xdot
%          thetadot]
%
% [t,y]=ode23('v',t0,tf,d01): t0=initial time,
%                               tf=final time,
%                               d01=initial conditions,
%
%   Before editing this file ,file v.m must be edited with the proper
% gains k(i,:) and parameters M,m,lo,lc,J.
%
% CHANGES: t0, tf, d01, g(i,:), text
%
% If the mass properties are changed from the default setting then the
following parameters must be %changed accordingly.

lo=0.330;           %pendulum rod length (m)
lc=-0.0281; %cg of the pendulum assembly minus the sliding rod in default bal.
wt. condition (m)
m1=0.213;          % mass of the sliding rod (kg)
m2=1.755; % mass of the pendulum assembly minus the sliding rod in default
bal. wt. condition %(kg)
Js=0.036; %this is really J*=J-m1*lo^2 in the theory section (kg m^2), J is
moment of inertia %about pivot
kx=50200; % sliding rod scale factor (encoder counts/m)
ka=2546; % angle encoder scale factor (encoder counts/radian)
kf=0.0013; %combined DAC amplifier scale factor (N/increment of DAC)

[t,y]=ode23('v',0,4.0,d01);
%R=0.0001
G=[381.3273  170.8638  69.5313  52.7100];
%step size of angle from vertical in radians (can be changed but not more than
0.5 rad).
step=0.25;

```

```

%
%feedforward signal where u is the input to the full state feedback closed
loop system
ff=step*(-((m1*lo+m2*lc)/m1)*G(1)+G(2));
u=(ff-G*y')/kf; % in units of DAC increment
tx1='';
tx2='';
tx3='';
tx4='';
tx5='';
pprint(t,y,u,tx1,tx2,tx3,tx4,tx5)

%*****
function yprim=v(t,y)
%called by nonlsim for the nonlinear simulation of the pendulum via a linear
state feedback controller
% with gain vector g. This vector may be updated from rows of g matrix after
running the lq m-function.
% If the mass properties are changed from the default setting then the
following parameters must be
%changed accordingly.
lo=0.330; %pendulum rod length (m)
lc=-0.028; %cg of the pendulum assembly minus the sliding rod in default bal.
wt. condition (m)
m1=0.213; % mass of the sliding rod (kg)
m2=1.755; % mass of the pendulum assembly minus the sliding rod in default
bal. wt. condition (kg)
Js=0.036; %this is really J*(=J-m1*lo^2) in the theory section (kg m^2), J
is moment of inertia about pivot
g=9.8174;

% y(1) = x
% y(2) = theta
% y(3) = d(x)/dt
% y(4) = d(theta)/dt
%
A=inv([ m1, m1*lo ; m1*lo, Js+m1*lo^2+m1*y(1)^2 ]);
B=m1*y(1)*[ y(4)^2 ; -2*y(3)*y(4) ];
C=g*[ m1*sin(y(2)) ; (m2*lc+m1*lo)*sin(y(2))+m1*y(1)*cos(y(2)) ];
%change state feedback vector if the response under a new controller is to be
simulated
% R=0.0001

```

```

G=[381.3273  170.8638  69.5313  52.7100];
%
%step size of angle from vertical in radians (can be changed but not more than
0.5 rad).
step=0.25;
%
%feedforward signal where ud is the input to the full state feedback closed
loop system
uud=step*(-((m1*l0+m2*l0)/m1)*G(1)+G(2));
%
D=[(uud-G*y);0];
yprim=[ y(3) ; y(4) ; A*(B+C+D)];

%*****
function []=pprint(t,y,u,tx1,tx2,tx3,tx4,tx5)
n=nargin;
clg,subplot(211),plot(t,y(:,1)),title('Position x(t)'),
    subplot(212),plot(t,y(:,2)),title('Angle a(t)'),
pause;
%print
meta plot
clg,subplot(211),plot(t,u),title('Control force UVAL'),
    subplot(212),plot([0,6,0,6],[0,0,3,3],'.');
if n>=4,
text(0.5,2.5,tx1);
end,
if n>=5,
text(0.5,2,tx2);
end,
if n>=6,
text(0.5,1.5,tx3);
end,
if n>=7,
text(0.5,1,tx4);
end,
if n==8,
text(0.5,0.5,tx5);
end,
%print
meta plot;

```