
EIETL LAB, INGLIS BUILDING EXPERIMENT 3F2-A

PENDULUM CONTROLLER EXPERIMENT

Objectives

This experiment uses the control of a crane/inverted pendulum to illustrate

- state-space dynamic models
- state-feedback and pole-placement
- limit cycles and describing functions.

The experiment is primarily a demonstration but some questions are asked at various times in the text, which should be answered in your report, together with a description of your results and necessary graphs, calculations and discussion.

Appendix A contains relevant theory. It should be read and understood before writing your report — ideally before doing the experiment!

************ WARNING ************

THIS APPARATUS CONTAINS MECHANISMS OF REASONABLE MASS, MOVING AT POTENTIALLY VERY HIGH SPEED AND SOMETIMES QUITE UNPREDICTABLY. KEEP WELL CLEAR OF ALL MOVING PARTS AND IF IT IS NECESSARY TO TOUCH THE PENDULUM DO SO WITH DUE CARE. DO NOT MAKE ANY ADJUSTMENTS TO THE APPARATUS, INCLUDING POWERING UP, WHILE ANOTHER PERSON IS CLOSE TO THE PENDULUM.

****

Note that the experimental results should be collected first, and then analyzed on the teaching system (either from the EIETL or the DPO as directed by your demonstrator). The analysis uses MATLAB and you should have gone through the "Getting Started with MATLAB" exercise before starting this part, if you have not used MATLAB before.

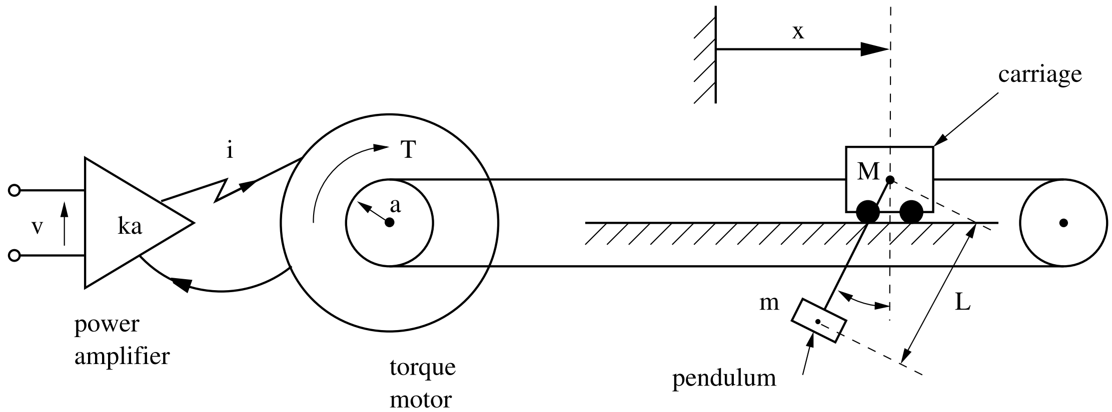

Figure 1: Apparatus

1 Apparatus

1.1 The System

The apparatus to be controlled is illustrated in Figure 1. It consists of a pendulum pivoted on a carriage which can be pulled up and down a track by a wire attached to a torque motor.

The system input is the voltage, v, to the power amplifier, which then produces a current, i, in the torque motor.

The observations are

- carriage position, CP, and carriage velocity, CV, from an optical encoder on the motor shaft.

- pendulum angular position, PP, and angular velocity, PV, from an optical encoder on the shaft on the carriage.

- motor current, MC.

Scale factors for these observations (all measured in volts) are as follows,

x	=	-0.08 × CP	m	
$\dot{x}$	=	-0.43 × CV	m/s	
θ	=	0.314 × PP	radians	$-\pi/2 < \theta < \pi/2$.
		π - 0.314 × PP	radians	$\pi/2 < \theta < 3\pi/2$.
$\dot{\theta}$	=	1.5 × PV	rads/s	
i	=	MC/0.146	A	

Other physical constants of the apparatus are:
Distance from the pendulum's centre of mass to the pivot, $L = 125$ mm
Radius of the pulley, $a = 16$ mm

Mass of the pendulum, $m = 0.320$ kg
Mass of the carriage, $M = 0.700$ kg
Moment of inertia on motor shaft, $I \cong 80 \times 10^{-6}$ kg-m^2
Torque-motor constant, $k_m = 0.080$ N-m/A
Amplifier constant, $k_a =$ -0.50 A/V
Frictional force on carriage $= F$ sgn $\dot{x}$ N
(F to be measured)

1.2 The Controller

The control box contains the power amplifier, signal conditioning and scaling for CP, CV, PP, PV and MC and various controller options. A circuit diagram of the controller is given in Figure 2.

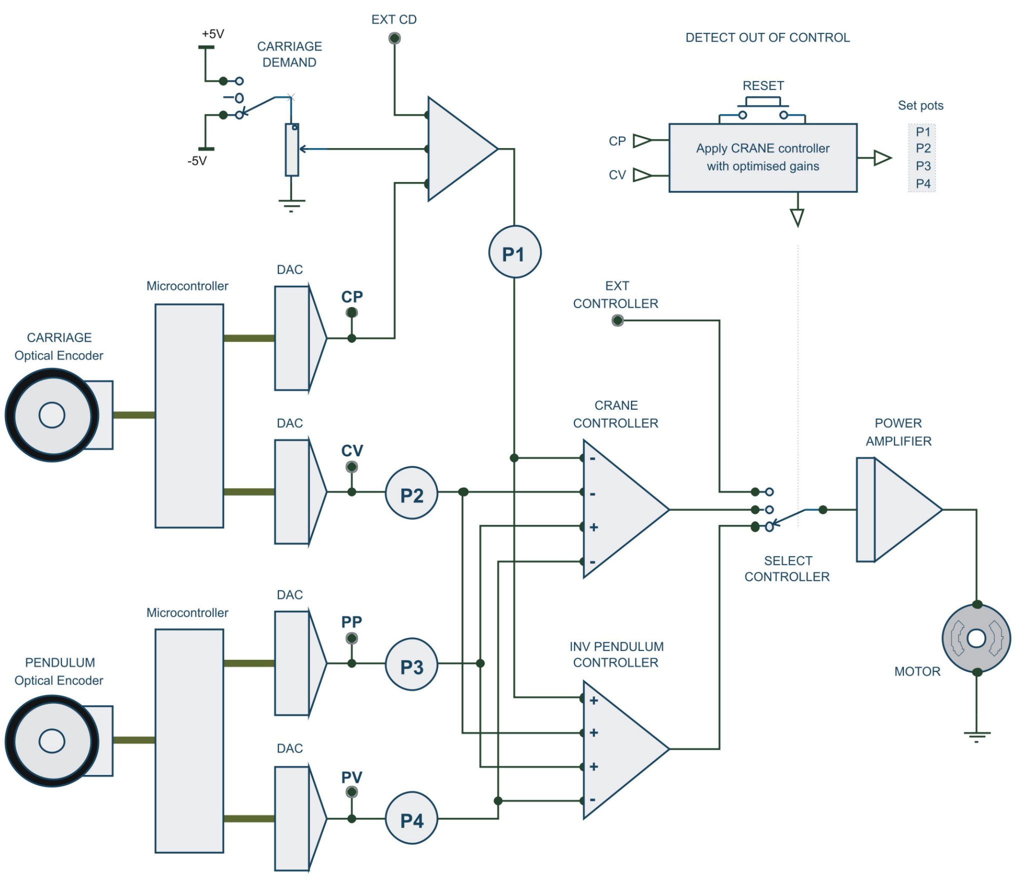

Figure 2: Controller Circuit

A circuit detects when the system is out of control and then switches in a controller that controls the carriage back to the centre with the pendulum vertically down. This latch remains set (with the red RESET light lit), until the RESET button is pressed. For this experiment external controllers will *not* be used, so the EXT CONT light should be off and either the CRANE (for section 4) or INV PEND (for section 5) control mode

selected.
The CRANE and INV PEND controllers differ in the sign and magnitudes of the gains associated with the variables.

For the CRANE controller,

$$v = -20 \times p_1 \times (CP - CD) - 30 \times p_2 \times CV + 20 \times p_3 \times PP - 10 \times p_4 \times PV$$

For the INV PEND controller,

$$v = 10 \times p_1 \times (CP - CD) + 20 \times p_2 \times CV + 30 \times p_3 \times PP - 20 \times p_4 \times PV,$$

where CD = "EXT + OFFSET" is the carriage position demand signal. By adjusting the OFFSET potentiometer, different sizes of step inputs can be produced. The feedback gains p_1, p_2, p_3 and p_4 are the settings **(in the range 0.000 to 1.000)** of the potentiometers labelled CP, CV, PP and PV respectively on this front panel **(do not confuse these feedback gains with the state measurements CP, CV, PP, and PV)**.

2 Using the computer

2.1 Data Logging

To start the data-logging program, double-click on the shortcut to *Pendulum Datalogger* on the windows desktop.

The Pendulum Datalogger will log and display data from the apparatus, sampled at 400 Hz. Data may be logged by clicking on the *Acquire Data* button. Spacebar or enter may be used to start and stop data acquisition. The plot of the logged data is immediately displayed in the main window but the axes are unchanged. If the axes do not cover the most suitable ranges, press the *Auto* button.

The plot defaults to displaying all five signals; however each signal may be hidden by removing the *check* in the *show* box besides its name. Signals may also be *inverted*.

The *Save* and *Open* buttons may be used to store and retrieve plots. The plots are automatically named `Plot` n`.txt`, where the number n is incremented with each plot, and are stored in the `C:\Data loggers/Inverted Pendulum` sub-directory.

2.2 Pole Calculations in MATLAB

To start MATLAB, double-click on the MATLAB shortcut on the windows desktop. MATLAB will be used for some calculations in order to set gains during the experiment and more extensively for analysis of the results. The relevant MATLAB .m files are listed in the appendix. To calculate the closed loop poles of the linear model type:

```
>> penddata
```

to load the physical constants, and to input the potentiometer settings, e.g.

```
>> p = [0.1 0.2 0.3 0.4]
```

and to obtain the poles for the crane position type

```
>> eig(Ac - b*p*Cc)
```

or for the inverted pendulum position type

```
>> eig(Ap - b*p*Cp)
```

3 Crane Control: Experimental Procedure

Throughout the experiment a large number of step responses are requested; however it is only necessary to plot them when the instructions explicitly say so, and not when verbs such as note and observe are used. Be sure to note the section number, gains and time duration of every logged data set.

3.1

Note the apparatus number. Measure the moving frictional force, F, on the carriage with a spring balance as well as the static friction force. Select the CRANE controller.

3.2 Carriage Controller

Design a carriage position servo by setting $p_1 = 0.350, p_3 = p_4 = 0$ and varying p_2 to give a critically-damped response in the carriage (ignoring the pendulum), say $p_2 \simeq 0.150$. Note the step response.

3.3

With p_1 and p_2 as in 3.2, increase p_3 from zero and note the step responses. Now spend 5 minutes varying p_3 and p_4 to minimize the oscillations in the pendulum. Log your best effort and obtain the theoretical closed-loop poles using MATLAB (see section 2.2 above). Record your optimal values of p_3 and p_4, and the theoretical closed-loop pole positions.

Comment on the transient response and pole positions.

3.4 Pole-placement

(a) Use example 1 of section A.5 together with the scale factors of section A.7 to place all the closed-loop poles at $-\omega_1 = -\sqrt{78.5}$. Record your calculated values of p_1–p_4, and note the step response. Check the theoretical poles using MATLAB, and comment on the consistency with the target pole position of $-\omega_1 = -\sqrt{78.5}$.

(b) Now increase the speed of response by placing the closed-loop poles at $-\alpha, -\beta, -\omega \pm j\omega$ for suitable values of α, β and ω. Type

>> alpha=α ; beta=β ; omega=ω

```
>> pole2p
```

in MATLAB to obtain the corresponding potentiometer settings. Record your choice (of α, β and ω) and the corresponding potentiometer settings. Log the step response and comment.
[It is not expected that you will choose the same values as any other students!].

3.5 Variation of p_2

With the design 3.4(b) vary p_2 until instability just occurs. Log the step response just prior to the onset of oscillations, and record the value of p_2. Use the linear model in appendix A to predict the gain k_2 at which oscillation will occur, and predict the resonant frequency $\hat{\omega}$ (see section A.5). Compare these with your experimental results (note: $k_2 = 165p_2$ — see A.7(a))

4 Inverted Pendulum: Experimental Procedure

4.1

Stand well clear of the pendulum and select the INV. PENDULUM controller. The carriage will probably be driven to the end of the track and the cut-out operated. To start controllers in future, hold the pendulum upright in the centre of the track and press RESET at the same time as releasing the pendulum.

4.2 No carriage feedback

Set $p_1 = p_2 = 0$ and p_3 and p_4 to stabilize the pendulum dynamics, say $p_3 = 0.500$ and $p_4 = 0.110$. Now hold the pendulum upright and press RESET. Manually note the force, on the pendulum, required to move the carriage - TAKE DUE CARE. Let go of the pendulum and explain the subsequent behaviour.

4.3 Pole Placement

Using example 2 in section A.5 and the data in section A.7 calculate p_i to place the closed-loop poles at $-\omega_1 = -\sqrt{78.5}$, and record your calculated values. Log the response.

4.4 Limit Cycles

Set the potentiometers to

$$p_1 = 0.23 \quad p_2 = 0.50 \quad p_3 = 0.63 \quad p_4 = 0.40$$

which should give a reasonably stable response. Now reduce p_2 until large oscillations occur (i.e. the carriage nearly hits the end stops). Record your value of p_2, and log the response. Now increase p_2 until the system is almost unstable. Record your value of p_2, and log the response to a small step.

4.5 No pendulum feedback

If the design of 4.3 is implemented except with $p_3 = p_4 = 0$ what would happen, and why?

5 Analysis of results

When you have finished the experimental work ftp your logged data to your account on the teaching system. Double-click on the *WinSCP* shortcut on the Windows desktop, and log into host *gate* using your DPO username and password. Using `Files/New/Directory`..., create a directory in your account called `3F2exptA` and transfer your logged data files from the `C:\Data loggers/Inverted Pendulum` subdirectory into this directory.

Using MATLAB on the teaching system you can perform a variety of comparisons between the linear theory, the nonlinear model, and these experimental results. After you log into the teaching system from the EIETL or DPO terminals, type:

```
start 3F2exptA
```

which will copy the .m files into the `3F2exptA` directory.

> On each of your six plots from sections 3 and 4, comment on the consistency between pole positions of the linear model and the experiment response.

A nonlinear simulation model is available and can be run as in the following example. Note that you need not type the comments.

We begin by setting various constants required for all simulations (using your own values for dynamic and static friction):

```
>> penddata          % get physical constant
>> h = 0.0025;       % sampling period
>> h_sim = 0.002;    % select simulation time step
>> F = 0.2*g         % measured dynamic friction
>> Fstatic =0.3*g;   % measured static friction
```

Then for each set of data to be analysed we load the appropriate file, set the potentiometer values (corresponding to the data file loaded) and choose appropriate simulation start and finish times. Finally, the nonlinear simulation is run using the command `nlsimc` (crane model) or `nlsimp` (inverted pendulum model).

```
>> n=1                          % select file number
>> logdata = loadlogdata(n);    % load into the logdata format
>> plot(0:h:2000*h,logdata)     % plot the data first
>> t0 = 0.4; tf = 3.4;          % select time interval to simulate
>> p = [0.1 0.2 0.3 0.4];       % input potentiometer settings used
>> nlsimc                       % to simulate the crane model
                                % or nlsimp to simulate the
                                % inverted pendulum model
```

The simulation is done using a simple Euler method which for the equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ successively calculates

$$\mathbf{x}(t+h) \simeq \mathbf{x}(t) + h.\mathbf{f}(\mathbf{x}(t), t)$$

A particular version has been written to enable the static and dynamic function to be simulated, with two different equation sets dependent on whether the carriage is stopped or moving. The program concludes by plotting the experimental results and the simulated results for comparison.

Plot a comparison of the experimental and simulation results for each of the six cases. On each plot, comment on the consistency between your experimental and simulation results.

* Compare the describing function predictions of the limit cycle amplitude and frequency with those observed in the results of section 4.4. You will need to work through the theory of section A.6 using your measured dynamic friction term.

6 Writing up

6.1 Laboratory Report

You are *not* required to write a full report (with introduction, aims and objectives etc). *All* that is required is:

1. The 12 plots requested in section 5 (with comments). These should be ordered in accordance with the sequence of experiments in sections 3 and 4
2. Answers to the underlined questions in sections 3 and 4
3. Answers to the (non starred) questions in the theory appendix A

Your answers for 2. and 3. should be provided on the worksheet at the end of this handout.

6.2 Full Technical Report

Guidance on the preparation of Full Technical Reports is provided both in Appendix I of the General Instructions document and in the CUED booklet A Guide to Report Writing, with which you were issued in the first year.

If you are offering a Full Technical Report on this experiment, you should include a discussion section addressing the starred questions contained in the analysis section 5, and in the theory appendix A. Include your Laboratory Report as an appendix to your write up, and refer to it where appropriate.

In preparing your report, you may find [1] useful on matters relating to linear state space systems, and [2] useful on nonlinear systems and the describing function method.

K. Glover,	October 2000
Revised: J. Paxman and J. Maciejowski,	March 2003
Revised: P. Goulart,	January 2007
Corrected: J. Maciejowski,	January 2011
Revised: R. Pates,	January 2015.
Revised: T. Hughes,	January 2017.

References

[1] Franklin, G. F., Powell, J. D., and Emami-Naeni, A., *Feedback Control of Dynamic Systems*, Addison-Wesley, 2nd edition, 1991.

[2] Khalil, H. C., *Nonlinear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 1996.

A Theory

The following sections describe some theoretical results which will aid the analysis and understanding of the experiment. You should ensure you have read and understood this material before completing your laboratory report.

Questions to be answered are boxed. The non-starred questions must be answered for the laboratory report, and the starred questions must also be anwered for the full technical report.

A.1 State-space model

Analysis of the dynamics of Figure 1, (assuming that the moments of inertia of the pendulum about its centre of mass is negligible) gives the equations

$$L\ddot{\theta} = \cos\theta.\ddot{x} - g\sin\theta \tag{1}$$

$$\left(1 + \frac{M}{m} + \frac{I}{ma^2}\right)\ddot{x} = \frac{T}{ma} - \left(\frac{F}{m}\right)\ \text{sgn}\ (\dot{x}) + L\cos\theta.\ddot{\theta} - L\sin\theta.\dot{\theta}^2 \tag{2}$$

where

$$\text{sgn}\ (\dot{x}) = \begin{cases} 1 & \dot{x} > 0 \\ -1 & \dot{x} < 0 \\ \text{undefined for } \dot{x} = 0. & \end{cases}$$

* <u>Derive</u> (1) and (2) using figure 1, and appropriate free body diagrams.

The sgn $(\dot{x})$ term cannot be sensibly linearized but the others can to give approximately,

$$L\ddot{\theta} = \ddot{x} - g\theta \tag{3}$$

$$\left(1 + \frac{M}{m} + \frac{I}{ma^2}\right)\ddot{x} = \frac{T}{ma} - \left(\frac{F}{m}\right)\ \text{sgn}\ (\dot{x}) + L\ddot{\theta} \tag{4}$$

These two simultaneous 2nd-order o.d.e.'s are equivalent to a 4-th order o.d.e. in either x or θ, but can also be written as a *first-order vector o.d.e., as follows. Let the* state-vector be the 4×1 vector,

$$\underline{x} = \begin{bmatrix} x \\ \dot{x} \\ L\theta \\ L\dot{\theta} \end{bmatrix} \tag{5}$$

We want to write $\dot{\underline{x}}$ in terms of $\underline{x}$ and T. Equations (3) and (4) can be solved for $\ddot{x}$ and $L\ddot{\theta}$ to give

$$\ddot{x} = -(\omega_0^2 - \omega_1^2)L\theta + u - f \tag{6}$$

$$L\ddot{\theta} = -\omega_0^2 L\theta + u - f \tag{7}$$

where

$$\omega_1^2 = \frac{g}{L} \tag{8}$$

$$\omega_0^2 = \omega_1^2 \left(1 + \frac{m}{(M + I/a^2)}\right) \tag{9}$$

$$u = \frac{T}{a\,(M + I/a^2)} \tag{10}$$

$$f = \left(\frac{F}{M + I/a^2}\right)\ \mathrm{sgn}\ (\dot{x}) \tag{11}$$

Note that ω_1 = natural frequency of the pendulum with the carriage fixed and ω_0 = natural frequency of the pendulum with the carriage free to move (assuming no friction).

Equations (6) and (7) can be written as a first order vector o.d.e. as follows

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ L\theta \\ L\dot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \omega_1^2 - \omega_0^2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_0^2 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x \\ \dot{x} \\ L\theta \\ L\dot{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}}_{\underline{B}} (u - f) \tag{12}$$

Ignoring friction, this is a particular case of the standard form,

$$\underline{\dot{x}} = A\underline{x} + B\underline{u} \tag{13}$$

which will be studied extensively in the Module 3F2 lectures, and some basic results will now be summarized.

A.2 Inverted pendulum model

Equations (1) and (2) still hold for the inverted case but the linearization is now about $\theta = \pi$. Let $\phi = \pi - \theta$ gives

$$-L\ddot{\phi} = -\cos(\phi)\ddot{x} - g\sin(\phi) \tag{14}$$

$$\left(1 + \frac{M}{m} + \frac{I}{ma^2}\right)\ddot{x} = \frac{T}{ma} - \left(\frac{F}{m}\right)\ \mathrm{sgn}\ (\dot{x}) + L\cos(\phi)\ddot{\phi} - L\sin(\phi)\dot{\phi}^2 \tag{15}$$

which linearizes to,

$$L\ddot{\phi} = \ddot{x} + g\phi \tag{16}$$

$$\left(1+\frac{M}{m}+\frac{I}{ma^2}\right)\ddot{x}=\frac{T}{ma}-\left(\frac{F}{m}\right)\ \text{sgn}\ (\dot{x})+L\ddot{\phi} \tag{17}$$

which are identical to (3) and (4) replacing θ by ϕ and changing the sign of the $g\phi$ term. Hence (12) becomes

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ L\phi \\ L\dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \omega_0^2-\omega_1^2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \omega_0^2 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ L\phi \\ L\dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}(u-f) \tag{18}$$

Verify equation (18), assuming that equations (14), and (15) are correct.

A.3 System Poles

Consider equation (13) in the general case when dimension of $\underline{x}$ is n. Assume $\underline{x}(0)=\underline{0}$ and take Laplace transforms to give,

$$s\underline{X}(s)=A\underline{X}(s)+\underline{B}U(s) \tag{19}$$

$\Rightarrow$

$$(sI-A)\underline{X}(s)=\underline{B}U(s) \qquad (I = n\times n \text{ identity matrix}) \tag{20}$$

$\Rightarrow$

$$\underline{X}(s)=(sI-A)^{-1}\underline{B}U(s) \tag{21}$$

Now $(sI-A)^{-1}$ can be written as the matrix of cofactors of $(sI-A)$ divided by the determinant of $(sI-A)$, say

$$(sI-A)^{-1}=N(s)/d(s) \tag{22}$$

where

$$d(s)=\det(sI-A) \tag{23}$$

and $N(s)$ is an $n\times n$ matrix of polynomials in s. $d(s)$ is called the characteristic polynomial of A. Hence

$$\underline{X}(s)=\frac{N(s)\underline{B}}{d(s)}\cdot U(s) \tag{24}$$

Now if (λ_i) is such that $d(\lambda_i)=0$ then λ_i is an *eigenvalue* of A by (23) and a system pole by (24). Hence,

the system poles = eigenvalues of A

Recall from your Part IB Linear Systems and Control lectures that the pole positions determine the stability and speed of response of a system. Briefly, if a system has a pole at the real value λ, then the transient response includes a term $e^{\lambda t}$. Similarly if a system has a pair of complex poles at $\sigma \underline{+} j\omega$ then the transient response includes terms like

$$Ae^{j\phi}e^{(\sigma+j\omega)t} + Ae^{-j\phi}e^{(\sigma-j\omega)t} = 2Ae^{\sigma t}\cos(\omega t + \phi) \tag{25}$$

which is an exponentially increasing ($\sigma > 0$) or decreasing ($\sigma < 0$) sinusoid of frequency ω rad/s.

> Just to refresh your memories repeat the examples paper question from last year which asked you to associate the pole positions with the transient responses given in figure 3. (Note the zero positions are not given).

In the present experiment the system is fourth order with its response made up from combinations of responses from the four poles.

A.4 State-feedback

Suppose that in (13) all the states are measured and can be used for control, what sort of performance can be achieved? Let

$$\begin{aligned} u &= w - k_1 x_1 - k_2 x_2 \cdots, -k_n x_n \\ &= w - \underline{K}\ \underline{x} \end{aligned} \tag{26}$$

where $\underline{K}$ is a $1 \times n$ matrix of feedback gains. (13) now becomes

$$\begin{aligned} \underline{\dot{x}} &= A\underline{x} + \underline{B}(w - \underline{Kx}) \\ &= (A - \underline{BK})\underline{x} + \underline{B}w \end{aligned} \tag{27}$$

This means that the “A-matrix” is now $(A - \underline{BK})$ and its new eigenvalues give the new closed-loop poles.

The *pole-placement theorem* states that if a system is “controllable” then its poles can be arbitrarily assigned using state-feedback.

> * Define controllability. Determine the values of ω_0 and ω_1 for which the linear crane equation (12) describes a controllable system.

Hence we can theoretically obtain any set of closed-loop poles by suitable choice of the state-feedback gains, $\underline{K}$. Fast and stable poles are generally desired but it is sometimes difficult to know which combinations of poles are easy to achieve and which are hard.

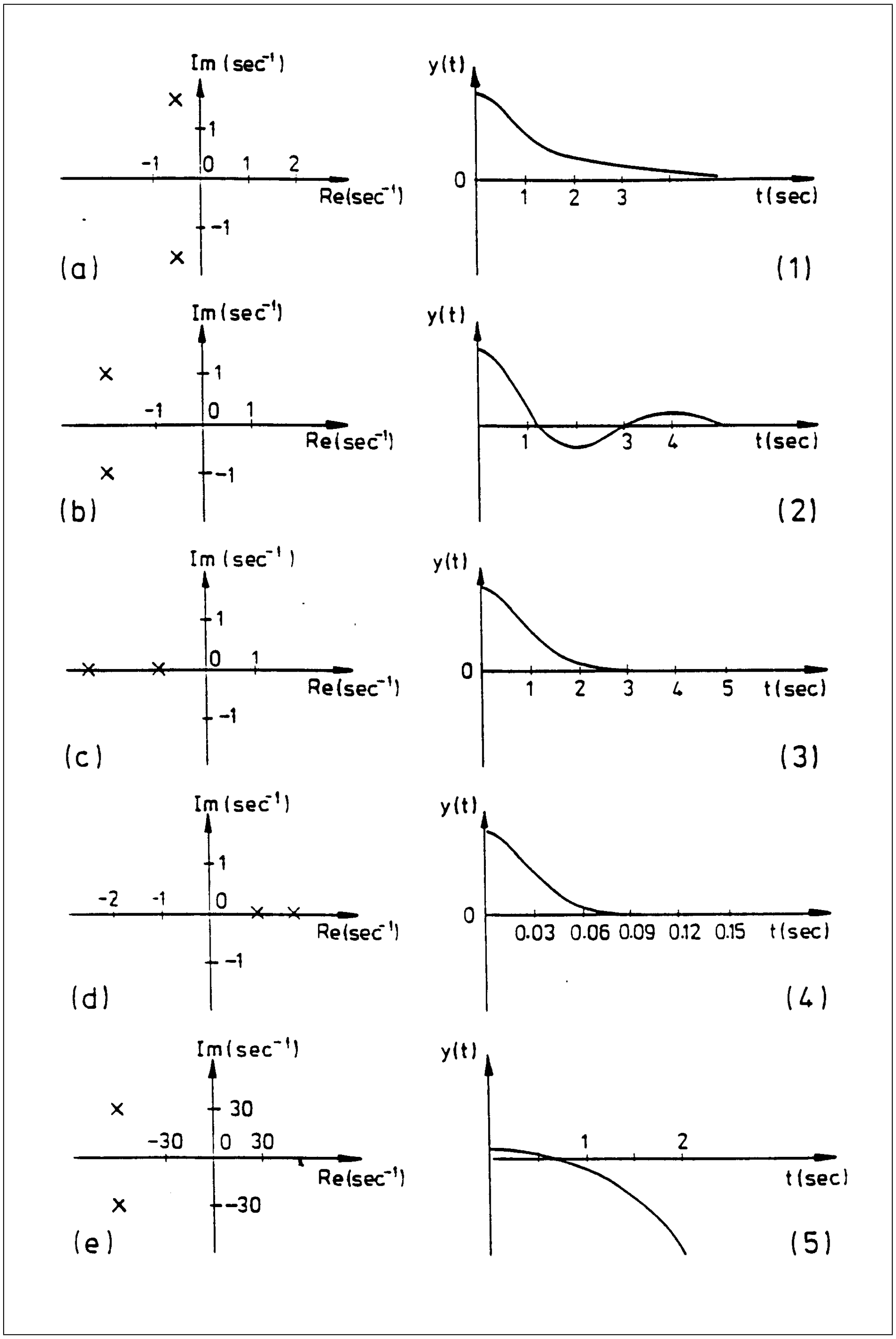

Figure 3: System poles exercise

A.5 Closed-loop characteristic equation

For the crane and inverted pendulum systems (still assuming $f = 0$), state feedback will give the closed-loop characteristic polynomials

$$\begin{aligned} d_c(s) &= \det(sI - A + \underline{BK}) \\ &= s^4 + (k_2 + k_4)s^3 + (k_1 + k_3 + \omega_0^2)s^2 + k_2\omega_1^2 s + k_1\omega_1^2 \end{aligned} \tag{28}$$

for the crane model, and

$$d_p(s) = s^4 + (k_2 + k_4)s^3 + (k_1 + k_3 - \omega_0^2)s^2 - k_2\omega_1^2 s - k_1\omega_1^2, \tag{29}$$

for the inverted pendulum model.

Verify equation (28).

It is clear that K can be chosen to make $d_c(s)$ or $d_p(s)$ arbitrary polynomials with any set of prescribed poles.

*

Use the Routh Hurwitz test (see Electrical and Information Data Book) to verify that the crane linear model will be stable if $k_1 > 0$, $k_2 > 0$, $k_2 + k_4 > 0$, $k_1 + k_3 + w_0^2 > 0$ and

$$k_2^2(k_3 + \omega_0^2 - \omega_1^2) + k_2k_4(k_3 + \omega_0^2 - k_1) > k_1k_4^2, \tag{30}$$

Furthermore if the last inequality becomes an equality show that there will be an oscillation of frequency

$$\hat{\omega} = \sqrt{\frac{k_2}{k_2 + k_4}}\omega_1 \tag{31}$$

Derive similar conditions for the inverted pendulum.

Example 1

For the crane set all the closed-loop poles to $-\omega_1 = -\sqrt{g/L}$. Then

$$\begin{aligned} d_c(s) &= (s + \omega_1)^4 = s^4 + 4\omega_1 s^3 + 6\omega_1^2 s^2 + 4\omega_1^3 s + \omega_1^4 \\ \Rightarrow k_1 &= \omega_1^2,\ k_2 = 4\omega_1,\ k_3 = 5\omega_1^2 - \omega_0^2,\ k_4 = 0 \end{aligned} \tag{32}$$

Example 2

For the inverted pendulum set all the closed-loop poles to $-\omega_1$. Then from (29)

$$k_1 = -\omega_1^2,\ k_2 = -4\omega_1,\ k_3 = \omega_0^2 + 7\omega_1^2,\ k_4 = 8\omega_1. \tag{33}$$

Eigenvalue Sensitivity

The roots of a polynomial can be very sensitive to the coefficients, particularly when the roots are nearly repeated. For example suppose $d(s) = (s + k)^4 + \varepsilon k^4$ where ε is the proportional error in the constant term.

> * Find the roots of $d(s)$ for $\varepsilon = 10^{-3}, 10^{-4}, 10^{-5}$ and plot on the Argand diagram (Hint: consider the 4th roots of $-\varepsilon$). Comment on this sensitivity in the light of your results on pole placement.

A.6 Friction and Limit Cycles

So far we have only considered the linear theory. Unfortunately in this experiment there is a significant frictional force on the carriage. One approximate method of analysis for sinusoidal responses or stability determination is the *describing function method*. Suppose that $\dot{x} = E \ \sin \ \omega t$ then $\text{sgn}(\dot{x})$ will be a square wave and the Fourier series expansion of $\text{sgn}(\dot{x})$ gives

$$\text{sgn }(\dot{x}) = \frac{4}{\pi} \sin \omega t + \frac{4}{3\pi} \sin 3\omega t + \ldots \tag{34}$$

The describing function method "describes" the nonlinearity by the first term only, assuming that the higher harmonics are filtered out by the system. The gain of the $\text{sgn }(\dot{x})$ nonlinearity is hence

$$N(E) = \frac{4}{\pi E} \tag{35}$$

and increases quickly as the amplitude of $\dot{x}, E,$decreases. Therefore if an oscillation is present the friction term looks like an additional feedback gain from $\dot{x}$, with equivalent gain (see (11))

$$\frac{4F}{\pi E(M + I/a^2)} \tag{36}$$

For the crane problem this implies that for small amplitudes of oscillation the system has a large additional damping term from the friction and will be stable; whereas for large amplitudes of oscillation the additional damping will be smaller and hence a large initial disturbance may excite large oscillations in certain circumstances.

For the inverted pendulum the friction is the cause of limit cycling (i.e. oscillations of well-defined amplitude) about the equilibrium. Notice that in this case the k_2 term must be negative for closed-loop stability (see Section A.5), and this term will be opposed by the effective damping term due to friction. The amplitude of the oscillation, E, will then adjust so that the additional damping term $N(E)$ would be just sufficient to cause instability in the linear model. The amplitude of the resulting limit cycle at a particular value of k_2 can be estimated as follows.

(i) From the linear model and section A.5 determine the reduction in gain k_2 that will just induce instability, call this Δk_2, and note the frequency, $\widehat{\omega}$.

(ii) The amplitude of the limit cycle for $\dot{x}, E$, then satisfies

$$E = \frac{4F}{\pi(M + I/a^2)\Delta k_2} \tag{37}$$

The corresponding limit cycle in x will be of amplitude $E/\widehat{\omega}$.

Therefore if the linear design has Δk_2 sufficiently large the amplitude of the limit cycle in $\dot{x}$ can be made acceptably small.

A.7 Physical Constants in the Models

The data of section 1.1 and 1.2 can now be incorporated into the state-space model and controller. Firstly (8) and (9) imply

$$\begin{aligned} \omega_1^2 &= 78.5 \quad (\text{rad/s})^2 \\ \omega_0^2 &= 103.3 \quad (\text{rad/s})^2 \end{aligned}$$

and hence

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \pm 24.8 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \pm 103.3 & 0 \end{bmatrix}$$

(the data of section 1.1 and 1.2 is accurate to only 10% in some cases so the values given in this section will be rounded to 2 or 3 significant digits).

From (10) we obtain

$$\begin{aligned} u &= \frac{T}{a\,(M + I/a^2)} \\ &= \frac{k_a k_m v}{a\,(M + I/a^2)} \end{aligned} \tag{38}$$

and substituting the physical values of section 1.1 gives

$$u = -2.47v \tag{39}$$

(a) <u>Crane controller</u>

The equation for v in section 1.2 together with the scale factors on the measurements give

$$\begin{aligned} u &= -2.47v \\ &= -617p_1(x - x_d) - 165p_2\dot{x} - 1256p_3L\theta + 126p_4L\dot{\theta} \end{aligned} \tag{40}$$

Hence

$$\begin{aligned} k_1 &= 617p_1 \\ k_2 &= 165p_2 \\ k_3 &= 1256p_3 \\ k_4 &= -126p_4 \end{aligned} \tag{41}$$

(b) Inverted pendulum controller

Similarly in this case, noting that $\phi = 0.314$ PP and $\dot{\phi} = -1.5$ PV, we obtain

$$\begin{aligned} k_1 &= -309p_1 \\ k_2 &= -110p_2 \\ k_3 &= 1884p_3 \\ k_4 &= 253p_4 \end{aligned} \tag{42}$$

B Matlab files for the analysis and simulation

B.1 penddata.m

```
% Matlab script file penddata.m  to calculate
% the  physical constants for the pendulum experiment

g=9.81;  % m/s^2
L=0.125; % distance from pendulum's centre of mass t pivot in m
a=0.016; % radius of pulley in m
m=0.32;  % mass of pendulum in kg
M=0.7;  % mass of carriage in kg
I=8e-5;  % moment of inertia on motor shaft in kg m^2
km = 0.08;  % torque motor constant in Nm/A
ka = - 0.50; % amplifier constant in A/V

gamma =  M/m + I/(m*a^2);

%scale factors to get physical units metres radians and seconds.
%in the crane/down position
% [x xdot Ltheta Lthetadot]=[CP CV PP PV]*Sc where,

Sc=diag([-(1/12.5) -(1/2.23) (L/3.18) (L/0.64)]);

% in the inverted position
% [x xdot phi phidot]=[CP CV PP PV]*Sp where,

Sp=diag([-(1/12.5) -(1/2.23) (L/3.18) -(L/0.64)]);
```

```
%controller amplifier gains on each measurement

opamp_c=diag([-20 -30 20 -10]); % for crane controller
opamp_p=diag([10 20 30 -20]); % for inverted pendulum controller

%maximum torque from the motor in Nm
Tmax=0.4;


% squares of the natural frequencies
om12=g/L; om02=om12*(1+1/gamma);

%linearized crane  model

Ac=[0 1 0 0 ; 0 0 om12-om02 0; 0 0 0 1; 0 0 -om02 0];
Cc=-(ka*km/(m*a*gamma))*opamp_c/Sc; b=[0;1;0;1];

%linearized inverted pendulum model

Ap=[0 1 0 0 ; 0 0 om02-om12 0; 0 0 0 1; 0 0 om02 0];
Cp=-(ka*km/(m*a*gamma))*opamp_p/Sp;
```

B.2 nlsimc.m

```
% Matlab script file nlsimc.m to simulate the nonlinear crane model
% and compare with the experimental logged data.

% the .m file penddata needs to be run before this one.

% The following variables need to be set before running this file.
% logdata: array of expt results.
% h: the sampling period of the logged data (i.e. total time/1000)
% h_sim: the simulation sampling period
% t0 and tf : the start and end time of the simulation.
% F and Fstatic: the moving and static friction terms.
% p : a 1x4 vector of potentiometer gains.


% scale the logged data to states in phyisical units: m m/s m m/s

xdata=[logdata(:,1:4)*Sc, logdata(:,5)*Sc(1,1)];
```

```
%set up time axes for the logged data, noting the cycling of the ADC
t1=0:h:(length(xdata)-1)*h; t2=t1+h/5; t3=t2+h/5; t4=t3+h/5; tdem=t4+h/5;

%note that xdem is the external input into the system.
xdem=xdata(:,5);


% set up the initial conditions and time index for the simulation

x0=[interp1(t1,xdata(:,1),t0);
    interp1(t2,xdata(:,2),t0);
    interp1(t3,xdata(:,3),t0);
    interp1(t4,xdata(:,4),t0)];


t_sim=(t0:h_sim:tf);


% determine whether carriage is initially effectively stationary

if abs(x0(2))<3*abs(Sc(2)),
    stopped = 1;
  else
    stopped = 0;
 end

x_sim=[]; newx=x0;
T_sim=[];


for t=t_sim,
  x=newx;
  x_sim=[x_sim; x'];
  xd=interp1(tdem,xdem,t);
  theta=x(3)/L; sinth=sin(theta);costh=cos(theta);
  T = ka*km*p*opamp_c*(Sc\(x-xd*eye(4,1)));
  if abs(T)>Tmax,
      T=Tmax*sign(T);
    end
  T_sim=[T_sim;T];

  if stopped,
    newx=[x(1); 0; x(3)+h_sim*x(4); x(4)+h_sim*(-g*sinth)];
    % check whether it will start moving i.e. whether the nett force on
```

```
      % on carriage exceed the maximum frictional force.
    if abs(T/a - m*g*sinth*costh - m*sinth*x(4)^2/L)>Fstatic,
       stopped = 0; t,
      end

  else   % not stopped
    x24dot=[1+gamma, -costh; -costh, 1]\ ...
       [ (-sinth*x(4)^2/L + T/(m*a) - (F/m)*sign(x(2)));-g*sinth];
    newx=x+h_sim*[x(2) ; x24dot(1) ; x(4) ; x24dot(2)];
    % check whether carriage will stop in this step
    if (x(2)~=0) & (x(2)*newx(2)<=0),
      stopped = 1; newx(2)=0;t,
     end
  end

end % end of for loop

% get the indices of the logged data in the simulation period
ind1=find(t1>=t0 & t1 <=tf);
ind2=find(t2>=t0 & t2 <=tf);
ind3=find(t3>=t0 & t3 <=tf);
ind4=find(t4>=t0 & t4 <=tf);
inddem=find(tdem>=t0 & tdem <=tf);


% plot the simulated and logged data one variable at a time.
subplot(1,1,1);
plot(t1(ind1),xdata(ind1,1),t_sim,x_sim(:,1),'--',tdem(inddem),xdata(inddem,5),':')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('x (m)');
disp('press any key to continue'); pause

plot(t2(ind2),-xdata(ind2,2),t_sim,x_sim(:,2),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('v (m/s)');
disp('press any key to continue');
pause

plot(t3(ind3),-xdata(ind3,3),t_sim,x_sim(:,3),'--')
ax = axis;
```

```
ax(2) = tf;
axis(ax);
ylabel('L\theta (m)');
disp('press any key to continue');pause

plot(t4(ind4),-xdata(ind4,4),t_sim,x_sim(:,4),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('L\omega (m/s)');
disp('press any key to continue');pause


% plot the results all on one page
subplot(4,1,1),
plot(t1(ind1),xdata(ind1,1),t_sim,x_sim(:,1),'--',tdem(inddem),xdata(inddem,5),':')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('x (m)');
figtitle='Crane position with p=[ ';
for i=1:length(p), figtitle=[figtitle,num2str(p(i),3),'  '];end
figtitle=[figtitle,']'];
title(figtitle);
subplot(4,1,2),
plot(t2(ind2),-xdata(ind2,2),t_sim,x_sim(:,2),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('v (m/s)');
subplot(4,1,3),
plot(t3(ind3),-xdata(ind3,3),t_sim,x_sim(:,3),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('L\theta (m)');
subplot(4,1,4),
plot(t4(ind4),-xdata(ind4,4),t_sim,x_sim(:,4),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('L\omega (m/s)');
xlabel('time (s)');
```

B.3 nlsimp.m

```
% Matlab script file nlsimp.m to simulate the nonlinear inverted pendulum
%  model and compare with the experimental logged data.

% the .m file penddata needs to be run before this one.

% The following variables need to be set before running this file.
% logdata:  a 1000x5 array of expt results.
% h: the sampling period of the logged data (i.e. total time/1000)
% h_sim: the simulation sampling period
% t0 and tf : the start and end time of the simulation.
% F and Fstatic: the moving and static friction terms.
% p : a 1x4 vector of potentiometer gains.


% scale the logged data to states in phyisical units: m m/s m m/s

xdata=[logdata(:,1:4)*Sp, logdata(:,5)*Sp(1,1)];

%set up time axes for the logged data, noting the cycling of the ADC
t1=0:h:(length(xdata)-1)*h; t2=t1+h/5; t3=t2+h/5; t4=t3+h/5; tdem=t4+h/5;

%note that xdem is the external input into the system.
xdem=xdata(:,5);


% set up the initial conditions and time index for the simulation

x0=[interp1(t1,xdata(:,1),t0);
    interp1(t2,xdata(:,2),t0);
    interp1(t3,xdata(:,3),t0);
    interp1(t4,xdata(:,4),t0)];


t_sim=(t0:h_sim:tf);


% determine whether carriage is initially effectively stationary

if abs(x0(2))<3*abs(Sc(2)),
    stopped = 1;
  else
    stopped = 0;
```

```
 end

x_sim=[]; newx=x0;
T_sim=[];


for t=t_sim,
  x=newx;
  x_sim=[x_sim; x'];
  xd=interp1(tdem,xdem,t);
  phi=x(3)/L; sinphi=sin(phi);cosphi=cos(phi);
  T = ka*km*p*opamp_p*(Sp\(x-xd*eye(4,1))); %calculate torque
  if abs(T)>Tmax,  %saturate torque
      T=Tmax*sign(T);
    end
  T_sim=[T_sim;T];

  if stopped,  % perform one step of Euler's method when stopped
    newx=[x(1); 0; x(3)+h_sim*x(4); x(4)+h_sim*(g*sinphi)];
    % check whether it will start moving i.e. whether the nett force on
      % on carriage exceed the maximum frictional force.
    if abs(T/a + m*g*sinphi*cosphi - m*sinphi*x(4)^2/L)>Fstatic,
       stopped = 0;t,
      end

  else   % perform one step of Euler's method when not stopped
    x24dot=[1+gamma, -cosphi; cosphi, -1]\ ...
        [ (-sinphi*x(4)^2/L + T/(m*a) - (F/m)*sign(x(2)));-g*sinphi];
    newx=x+h_sim*[x(2) ; x24dot(1) ; x(4) ; x24dot(2)];
    % check whether carriage will stop in this step
    if (x(2)~=0) & (x(2)*newx(2)<=0),
      stopped = 1; newx(2)=0; t,
     end
  end

end % end of for loop

% get the indices of the logged data in the simulation period
ind1=find(t1>=t0 & t1 <=tf);
ind2=find(t2>=t0 & t2 <=tf);
ind3=find(t3>=t0 & t3 <=tf);
ind4=find(t4>=t0 & t4 <=tf);
inddem=find(tdem>=t0 & tdem <=tf);
```

```
% plot the simulated and logged data one variable at a time.
subplot(1,1,1);
plot(t1(ind1),xdata(ind1,1),t_sim,x_sim(:,1),'--',tdem(inddem),xdata(inddem,5),':')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('x (m)');disp('press a key to continue');pause

plot(t2(ind2),-xdata(ind2,2),t_sim,x_sim(:,2),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('v (m/s)');
disp('press any key to continue'); pause

plot(t3(ind3),-xdata(ind3,3),t_sim,x_sim(:,3),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('L\theta (m)');
disp('press any key to continue'); pause

plot(t4(ind4),-xdata(ind4,4),t_sim,x_sim(:,4),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('L\omega (m/s)');
disp('press any key to continue'); pause

% plot the results all on one page
subplot(4,1,1),
plot(t1(ind1),xdata(ind1,1),t_sim,x_sim(:,1),'--',tdem(inddem),xdata(inddem,5),':')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('x (m)');
figtitle='Inverted position with p=[ ';
for i=1:length(p), figtitle=[figtitle,num2str(p(i),3),'  '];end
figtitle=[figtitle,']'];
title(figtitle);
subplot(4,1,2),
plot(t2(ind2),-xdata(ind2,2),t_sim,x_sim(:,2),'--')
ax = axis;
ax(2) = tf;
```

```
axis(ax);
ylabel('v (m/s)');
subplot(4,1,3),
plot(t3(ind3),-xdata(ind3,3),t_sim,x_sim(:,3),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('L\theta (m)');
subplot(4,1,4),
plot(t4(ind4),-xdata(ind4,4),t_sim,x_sim(:,4),'--')
ax = axis;
ax(2) = tf;
axis(ax);
ylabel('L\omega (m/s)');
xlabel('time (s)');
orient tall
```

B.4 pole2p.m

```
% Matlab script file pole2p.m to calculate
% the potentiometer settings p which place the closed-loop poles
% of the crane at -alpha, -beta and -omega +/- j omega

penddata k=[]; k(1)=2*omega^2*alpha*beta/om12;
k(2)=(2*omega^2*(alpha+beta)+2*omega*alpha*beta)/om12;
k(3)=2*omega^2+2*omega*(alpha+beta)+alpha*beta-om02-k(1);
k(4)=2*omega+alpha+beta-k(2); p=k/Cc
```

B.5 loadlogdata.m

```
%specify a file number n
%the function will load the corresponding file Plot 00n.txt, and convert it
%into the appropriate form for nlsimc.m and nlsimp.m

% this file required due to hardware update of computers and dataloggers in
% 2014-15

% experimental data recorded as Plot 00n.txt, where n is the file number
% file has header, followed by data in fields [t,CD,CP,CV,PP,PV]
```

```
% t recorded in seconds, all other data recorded in binary

% nlsimc.m and nlsimp.m accept an array with columns [CP,CV,PP,PV,CD] in
% Volts

% the txt files recording the data should be converted to csv format - this
% would make the data much easier to read and avoid the need for such a
% complex file...

%%
function data=loadlogdata(n)


if n<10
    fid=fopen(['Plot 00',num2str(n),'.txt']);
else
    fid=fopen(['Plot 0',num2str(n),'.txt']);
end

%load data
dt=fread(fid);dt=char(dt)';

%skip to end of headers
n=strfind(dt,'0.0');dt=dt(n(1):end);

%find ‘enters’ in file (these go between readings)
n1=(double(dt)==13);

%each line of data is then 32 characters long
data=zeros(sum(n1),5);
prv=1;
for i = 1:size(data,1)

    %lift out reading
    [~,nxt]=max(n1);n1(nxt)=0;
    rdng=dt(prv:nxt);prv=nxt;

    %load reading into the appropriate fields
    %fields separated by spaces
    n3=(double(rdng)==9);idx=1:length(rdng);idx=idx(n3);

    data(i,1)=str2double(rdng(idx(2):idx(3)));%CP
    data(i,2)=str2double(rdng(idx(3):idx(4)));%CV
    data(i,3)=str2double(rdng(idx(4):idx(5)));%PP
```

```
    data(i,4)=str2double(rdng(idx(5):idx(6)));%PV
    data(i,5)=str2double(rdng(idx(1):idx(2)));%CD

end

%convert readings to volts
data=data*20/4095-10;

end
```

Worksheet

3.1 Apparatus number:

Moving frictional force: Static frictional force:

3.3 Optimal values: p_3 p_4

Theoretical closed-loop pole positions:

3.4 Potentiometer settings to place poles closed-loop poles at $-\omega_1 = -\sqrt{78.5}$:
p_1 p_2 p_3 p_4

Calculated closed-loop pole positions:

Comment on consistency:

Chosen pole locations: α β ω

Corresponding potentiometer settings:
p_1 p_2 p_3 p_4

3.5 Value of p_2 at onset of oscillations:

Predicted gain k_2: Predicted resonant frequency $\hat{\omega}$:

Comparison with actual gain and resonant frequency:

4.2 Explain the behaviour when you let go of the pendulum:

4.3 Potentiometer settings to place poles closed-loop poles at $-\omega_1 = -\sqrt{78.5}$:

p_1 p_2 p_3 p_4

4.4 Value of p_2 for large oscillations:

Value of p_2 at onset of instability:

Describe and explain behaviour when $p_3 = p_4 = 0$:

A.2 Verify equation (18):

A.3 Transient response corresponding to pole positions:

(a) (b) (c) (d) (e)

A.5 Verify equation (28):