Double Inverted Pendulum Experiment

(Linear Motion Module using IP-02)

1.0 SYSTEM DESCRIPTION

The ***double inverted pendulum module*** attaches to the IP-02 Self Erecting linear motion inverted pendulum. It consists of two arms connected via an instrumented joint. The encoder on the joint measures the deflection of the second pendulum relative to the first. The angle of the lower pendulum is measured using the encoder already present on the IP-02 cart.

In order to assemble the system, remove the single pendulum from the IP-02. Remove the weight from the IP-02 cart. It is not needed for this experiment. Attach the double inverted pendulum module to the shaft of the IP-02 cart as shown in Figure 1. Make sure you push the pendulum attachment such that the stops are effective when the pendulum swings too far. It is strongly recommended you ensure that the stops are effective. Tighten up all the screws in the system.

It is strongly recommended that you perform the single inverted pendulum experiment before you perform this one.

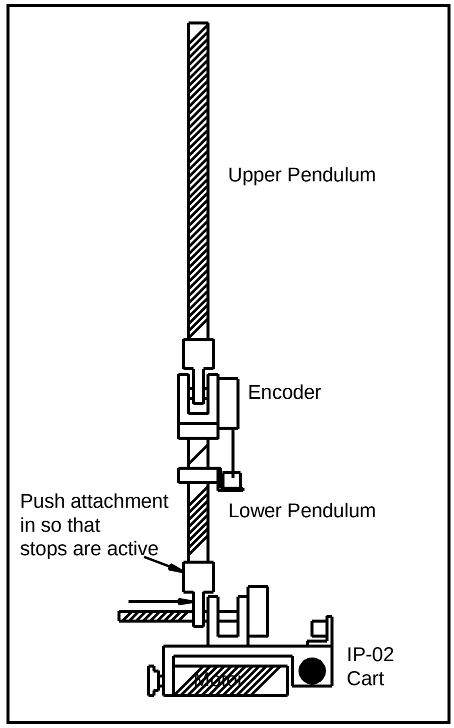


Figure 1: Assembly of Double inverted pendulum

2.0 MATHEMATICAL MODELLING

Consider the simplified model shown in Figure 2. We will derive the differential equations based on the Lagrange formulation. In order to derive the kinetic and potential enrgies of the moving elements, we need to obtain their positions and velocities realtive to a fixed reference frame.

The transformation are defined as shown in Figure 3. The transformation matrices between the frames are:

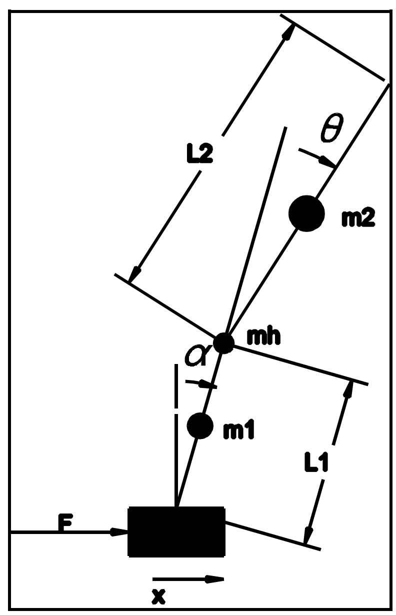


Figure 2: Simplified model

$$T_1^{hinge} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & L_1\ \sin(\alpha) \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & L_1\ \cos(\alpha) \\ 0 & 0 & 0 & 1 \end{bmatrix} (F_1 \rightarrow F_3)$$

$$T_1^{cg1} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & \frac{L_1}{2}\sin(\alpha) \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & \frac{L_1}{2}\cos(\alpha) \\ 0 & 0 & 0 & 1 \end{bmatrix} (F_1 \rightarrow F_2)$$

$$T_2^{cg2} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & \frac{L_2}{2}\sin(\theta) \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & \frac{L_2}{2}\cos(\theta) \\ 0 & 0 & 0 & 1 \end{bmatrix} (F_3 \rightarrow F_4)$$

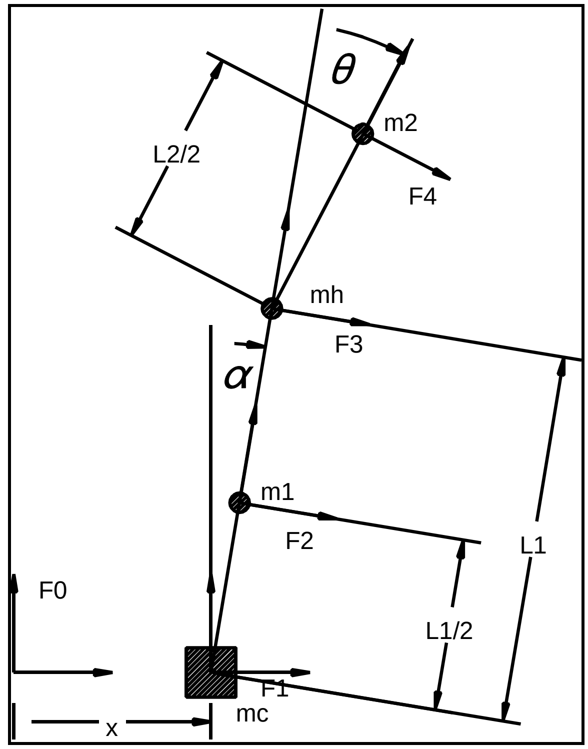


Figure 3: Transformation frames

The following are the definition of the parameters in the model:

L_1	*Length of lower pendulum(m)*	m_1	*Mass of lower pendulum(Kg)*
L_2	*Length of upper pendulum(m)*	m_2	*Mass of upper pendulum(Kg)*
m_c	*Mass of cart (Kg.)*	m_h	*Mass of hinge & encoder(Kg)*
F	*force applied to cart (N)*	x	*Cart displacement (m)*
α	*Angle of lower pendulum (rad) relative to vertical*	θ	*Angle of upper pendulum relative to first pendulum (rad)*

From the transformation matrices we derive the positions of the relevant moving parts relative to the fixed reference frame F0.

$$\begin{bmatrix} p_x^{m1} \\ p_y^{m1} \\ p_z^{m1} \end{bmatrix} = \begin{bmatrix} T_1^{cg1}(1,4) + x \\ T_1^{cg1}(2,4) \\ T_1^{cg1}(3,4) \end{bmatrix}$$

$$\begin{bmatrix} p_x^{mh} \\ p_y^{mh} \\ p_z^{mh} \end{bmatrix} = \begin{bmatrix} T_1^{hinge}(1,4) + x \\ T_1^{hinge}(2,4) \\ T_1^{hinge}(3,4) \end{bmatrix}$$

$$\begin{bmatrix} p_x^{m2} \\ p_y^{m2} \\ p_z^{m2} \end{bmatrix} = \begin{bmatrix} (T_1^{hinge} \times T_2^{cg2})(1,4) \;+\; x \\ (T_1^{hinge} \times T_2^{cg2})(2,4) \\ (T_1^{hinge} \times T_2^{cg2})(3,4) \end{bmatrix}$$

(note that the x positions are simply augmented by x to account for the translation of the cart along the x axis)

Then the potential energies of the elements are:

$$PE_{m1} = m_1 \; g \; p_z^{m1}$$
$$PE_{mh} = m_h \; g \; p_z^{mh}$$
$$PE_{m2} = m_2 \; g \; p_z^{m2}$$

The kinetic energies of the moving element are:

$$KE_{m1} = \frac{1}{2} \; m_1 \; (\; (\frac{dp_x^{m1}}{dt})^2 + (\frac{dp_z^{m1}}{dt})^2 \;)$$
$$KE_{mh} = \frac{1}{2} \; m_h \; (\; (\frac{dp_x^{mh}}{dt})^2 + (\frac{dp_z^{mh}}{dt})^2 \;)$$
$$KE_{m2} = \frac{1}{2} \; m_2 \; (\; (\frac{dp_x^{m2}}{dt})^2 + (\frac{dp_z^{m2}}{dt})^2 \;)$$
$$KE_{mc} = \frac{1}{2} \; m_c \; \dot{x}^2$$

Note that there is no “y” component in the movements and thus the terms of the form $\frac{d \; p_y}{dt}$ are zero.

These are then used in the Lagrange formulation to obtain the nonlinear differential equations (see ***dbl_pend.map***) which are linearized to obtain a linear model (see ***the file dbl_p_m.m***).

Substituting system parameters we obtain

$$\begin{bmatrix} \dot{x} \\ \dot{\alpha} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -5.96 & 0.136 & 0 & 0 & 0 \\ 0 & 109.19 & -50.95 & 0 & 0 & 0 \\ 0 & -113.25 & 131.68 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \alpha \\ \theta \\ \dot{x} \\ \dot{\alpha} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.7875 \\ -12.3805 \\ 12.8405 \end{bmatrix} F$$

The above system is modelled with a Force F as input. In order to design a feedback system, we need to control the ***voltage*** to the motor. The force F is generated via the motor voltage through the equation (See self-erecting section):

$$F = K_m K_g \frac{I_m}{r} = \frac{K_m K_g}{R\ r} V - \frac{K_m^2 K_g^2}{R\ r^2} \dot{x}$$

Substituting system parameters we obtain:

$$\begin{bmatrix} \dot{x} \\ \dot{\alpha} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -5.96 & 0.136 & -13.7315 & 0 & 0 \\ 0 & 109.19 & -50.95 & 95.1 & 0 & 0 \\ 0 & -113.25 & 131.68 & -98.64 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \alpha \\ \theta \\ \dot{x} \\ \dot{\alpha} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3.07 \\ -21.28 \\ 22.07 \end{bmatrix} V$$

Which is the linear model used to design the control system.

3.0 CONTROL SYSTEM DESIGN

The design is performed using the Linear Quadratic Regulator.

The system matrix is augmented with the state

$$\zeta = \int x$$

in order to compensate for steady state errors that may arise from measurement bias which are inevitable when starting the controller.

This results in the final state space equations:

$$\begin{bmatrix} \dot{x} \\ \dot{\alpha} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\alpha} \\ \ddot{\theta} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -5.96 & 0.136 & -13.7315 & 0 & 0 & 0 \\ 0 & 109.19 & -50.95 & 95.1 & 0 & 0 & 0 \\ 0 & -113.25 & 131.68 & -98.64 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \alpha \\ \theta \\ \dot{x} \\ \dot{\alpha} \\ \dot{\theta} \\ \zeta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3.07 \\ -21.28 \\ 22.07 \\ 0 \end{bmatrix} V$$

Defining

$$Q = diag([0.1 \;\; 10000 \;\; 10000 \;\; 0 \;\; 0 \;\; 0 \;\; 2000]) \\ r = 4$$

results in the optimal feedback gain:

$$k_{rad} = [35.1475 \;\; 117.5454 \;\; 242.3782 \;\; 23.1535 \;\; 30.0031 \;\; 28.6220 \;\; 22.3607]$$

whose units are V/m and V/rad.

The controller is implemented using the gains converted to V/cm and V/deg which are:

$$k_{met} = [0.3515 \;\; 2.0544 \;\; 4.2361 \;\; 0.2315 \;\; 0.5244 \;\; 0.5002 \;\; 0.2236]$$

4.0 CONTROL SYSTEM IMPLEMENTATION

The above controller is implemented using SIMULINK and run in realtime using WinCon. One important factor to consider is the method we differentiate the signals.

The angles are measured via encoders but the derivatives of the angles are not available. We have to obtain the derivatives numerically. The simplest method to perform this is to pass the signals through a high-pass filter of the form:

$$H(s) = [s]\left[\frac{w_c^2}{s^2 + 2\,\zeta\, w_c + w_c^2}\right] = \frac{w_c^2\, s}{s^2 + 2\,\zeta\, w_c + w_c^2}$$

which essentially differentiates the signal (via the transform **s**) and low pass filters it to eliminate high frequency noise. The important parameter to select here is the bandwidth of the low pass filter w_c. Care must be taken in selecting w_c since a low bandwidth filter will introduce unnecessary delays which will cause instability while a high bandwidth filter will be noisy.

Consider the closed loop system:

$\dot{x} = (A - bk)\,x$

The closed loop eigenvalues for the gain selected above are at :

-30.5118 +25.1919i
-30.5118 -25.1919i
  -6.8298 + 2.8625i
  -6.8298 - 2.8625i
  -0.8783 + 1.3399i
  -0.8783 - 1.3399i
  -1.6683

which indicates that the highest mode of the system is at around 25 rad/sec. So to effectively differentiate the measured signals we require that w_c > 25 rad/sec.

The system is implemented with wc = 40 rad /sec and critical damping ($\zeta = 1$) which differentiates the signals effectively and is not noisy.

5.0 Results

If you have a MultiQ board, you can simply load ***q_db_p.wcl*** into WinCon and start the controller.

Always start the controllers with both pendula held as close as possible to the vertical and the cart in the middle of the track. The encoder counters are initialized to zero thus defining the initial angle as “zero” degrees. If you start with angles that are a little off from the vertical, this results in steady state errors which the system may not be able to compensate for.

Figure 4 shows the initial response of the controller. As soon as you start the controller, the cart will travel in one direction to stabilize the pendula and then travel back to the starting point slowly while keeping both pendula upright. The initial response is due to constant disturbances resulting from innacurate positioning along the vertical axis. The integrator in the loop compensates for these constant disturbances and forces the cart back to the zero position. The cart never stops moving as it constantly tries to compensate for the falling of the pendula.

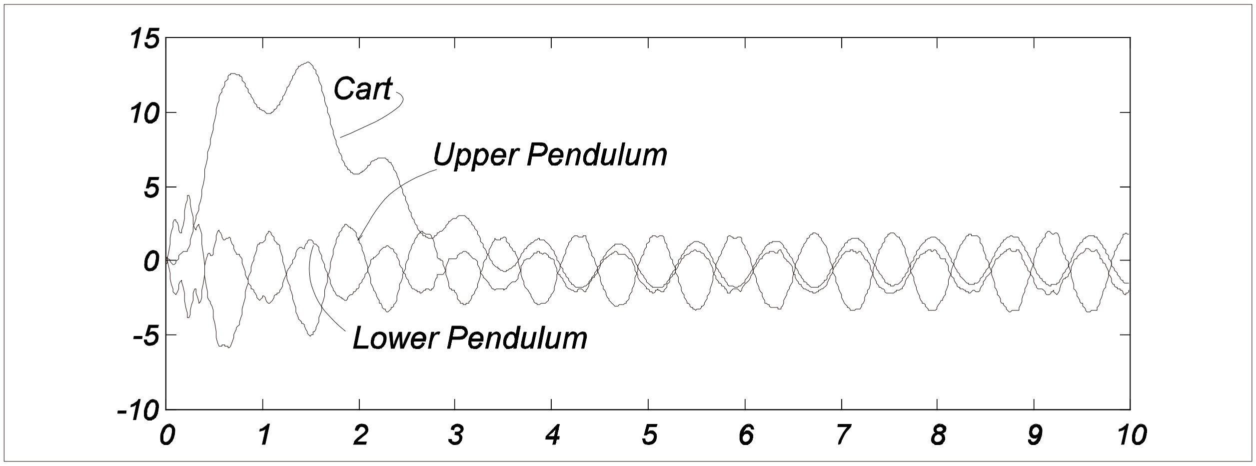


Figure 4: Initial response of stabilized system

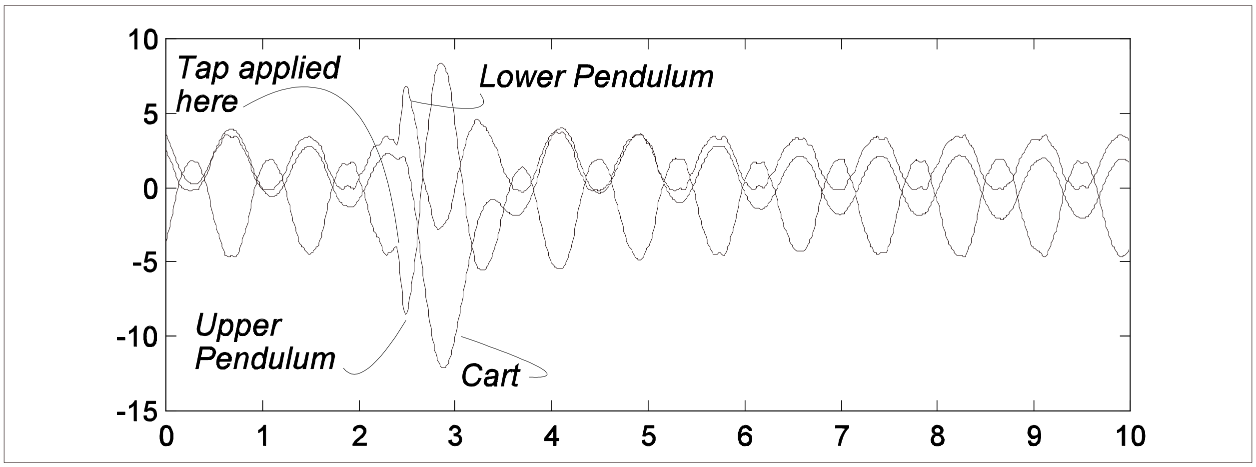


Figure 5: Response to a gentle tap to the upper pendulum

Figure 5 shows the response of a system to a *gentle* tap to the upper pendulum Note how the system recovers well. Do not tap the pendulum too hard as there is not enough energy in the sytsem to recover from large disturbances!

6.0 System Parameters

The same system parameters apply as in the IP-02 . Over and above those parameters we have:

Parameter	Symbol	Value	Units
Lower pendulum length	l_1	0.1524	m
Lower pendulum mass	m1	0.04	Kg
Mass of hinge	m_h	0.17	Kg
Upper pendulum length	l_2	0.4318	m
Upper pendulum mass	m_2	0.14	Kg
Upper pendulum encoder resolution	N/A	4095	count/rev
Upper pendulum encoder calibration constant	N/A	.08789	deg/count

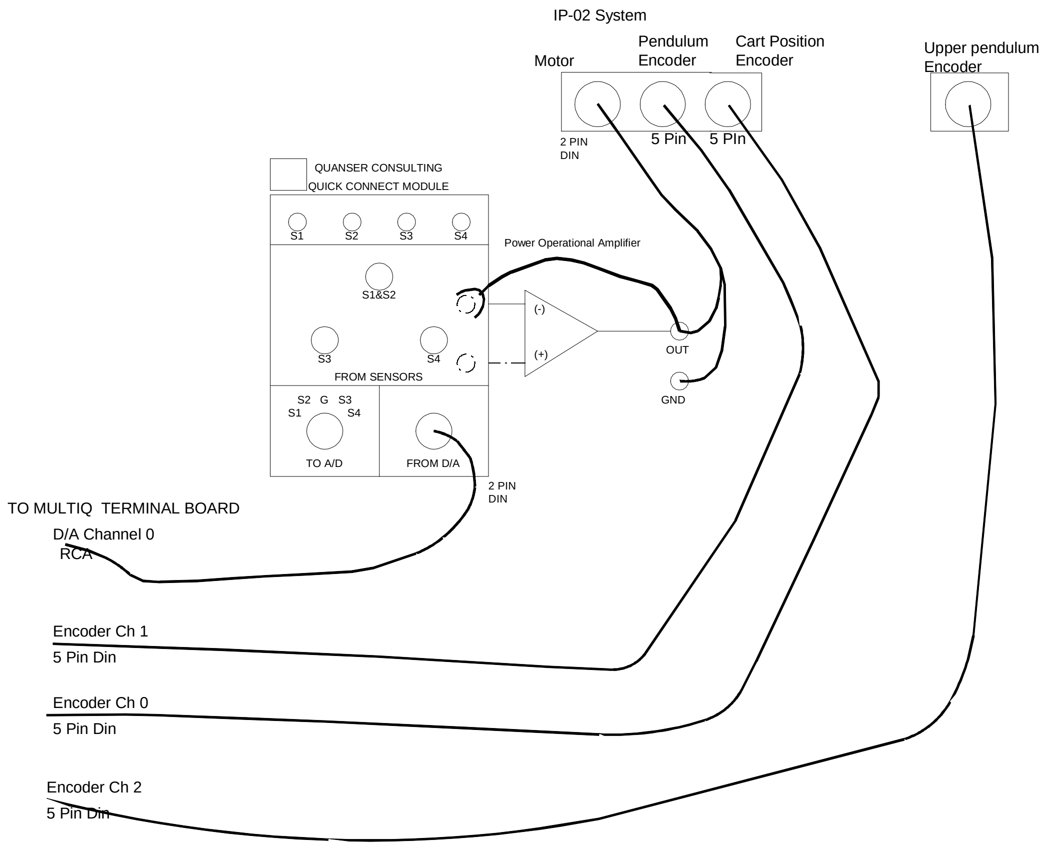
7.0 WIRING & SOFTWARE

Wire the system as shown in Figure 7. Note that the wiring is the same as in the self-erecting pendulum except that you now also measure the upper pendulum angle via encoder channel 2.

The following software is supplied with the system

Filename	Function	Required Windows application
dbl_pend.map	Derive differential equations	MAPLE V
dbl_p_m.m	Linear model obtained from MAPLE	MATLAB
db_pend	Design LQR controller	MATLAB
q_dp_p.m	SIMULINK diagram used for realtime code generation	SIMULINK, Watcom, RTW
q_db_p.wcl	Executable WinCon Library of SIMULINK controller. Realtime controller.	WinCon

In order to run the SIMULINK program in realtime under Windows you need REALTIME WORKSHOP, WATCOM C compiler and WinCon. See WinCon manual for details.


IP-02 System
Motor
Pendulum Encoder
Cart Position Encoder
Upper pendulum Encoder
2 PIN DIN
5 Pin
5 PIn
QUANSER CONSULTING
QUICK CONNECT MODULE
S1
S2
S3
S4
Power Operational Amplifier
S1&S2
(-)
(+)
OUT
GND
S3
S4
FROM SENSORS
S2 G S3
S1 S4
TO A/D
FROM D/A
2 PIN DIN
TO MULTIQ TERMINAL BOARD
D/A Channel 0
RCA
Encoder Ch 1
5 Pin Din
Encoder Ch 0
5 Pin Din
Encoder Ch 2
5 Pin Din