Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρ. Μηχ/κών και Μηχ/κών Υπολογιστών

Τομέας Σημάτων, Ελέγχου και Ρομποτικής

« ΕΡΓΑΣΤΗΡΙΟ ΡΟΜΠΟΤΙΚΗΣ »

Άσκηση 2. Έλεγχος Pendubot

Υπεύθυνος Εργαστηρίου: Κ. Τζαφέστας
Email: ktzaf@cs.ntua.gr
Web: robotics.ntua.gr/members/ktzaf/

Μεταπτυχιακοί Συνεργάτες:

Παρασκευάς Οικονόμου, Email: oikonpar@mail.ntua.gr

Σταματίνα Μπαράκου, Email: matbarakou@gmail.com

Παρατήρηση:

Η Άσκηση θα διεξαχθεί στο χώρο του Εργαστηρίου Ρομποτικής και Αυτοματισμού στο Κτήριο Β (Γενικές Έδρες), 2ος Όροφος, τηλ. 210-7721546.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ/ΚΩΝ & ΜΗΧ/ΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

«Εργαστήριο Ρομποτικής»

Άσκηση 2

Έλεγχος Pendubot

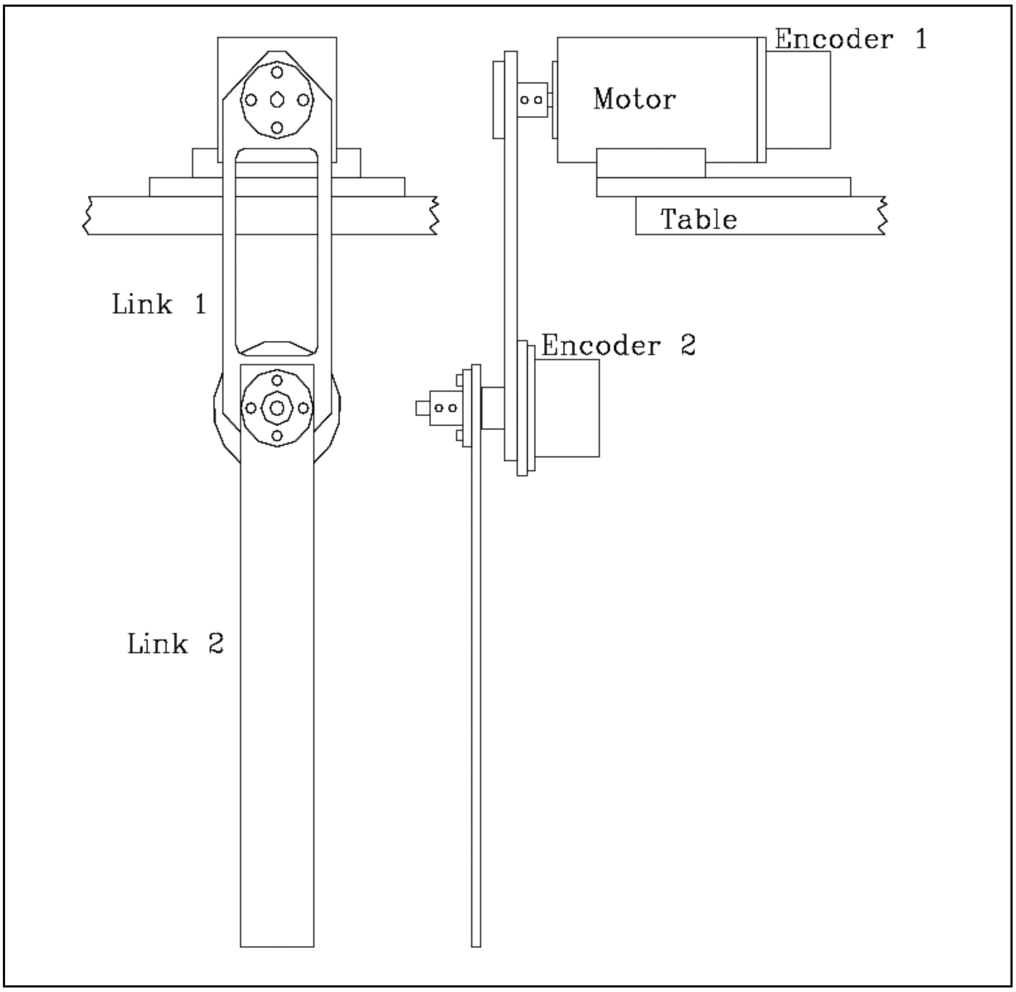
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ:
Έλεγχος Ρομποτικού Μηχανισμού Pendubot

1. Εισαγωγή

Σκοπός της εργαστηριακής αυτής άσκησης είναι να βοηθήσει τους φοιτητές στην κατανόηση των βασικών τεχνικών γραμμικού και μη γραμμικού ελέγχου ρομποτικών μηχανισμών. Στόχος είναι: (α) η εξοικείωση με την πειραματική ρύθμιση ενός ελεγκτή PID για τον έλεγχο θέσης μιας μεμονωμένης άρθρωσης του μηχανισμού αυτού, (β) η κατανόηση της επίδρασης συγκεκριμένων χαρακτηριστικών του δυναμικού ρομποτικού μοντέλου (όπως η επίδραση των όρων της βαρύτητας) στην επίδοση του γραμμικού τοπικού ελεγκτή PD (π.χ. μόνιμο σφάλμα θέσης και υπερύψωση, overshoot), (γ) η υλοποίηση ενός ελεγκτή ο οποίος θα συμπεριλαμβάνει και μη γραμμικούς όρους βασισμένους στο δυναμικό ρομποτικό μοντέλο, και (δ) ο πειραματισμός με τη λειτουργία ενός πλήρους μη-γραμμικού ελεγκτή (δύο φάσεων) για την εξισορρόπηση του αναστρόφου εκκρεμούς στην ενδιάμεση θέση ασταθούς ισορροπίας (middle balancing position, βλ. [1]).

2. Πειραματική Διάταξη

Για το σκοπό της εργαστηριακής αυτής άσκησης θα χρησιμοποιηθεί ένας ρομποτικός μηχανισμός τύπου "ανάστροφο εκκρεμές" (Pendubot, Mechatronics Systems Inc.). Ο μηχανισμός αυτός διαθέτει δύο βαθμούς ελευθερίας (δύο στροφικές αρθρώσεις, joint-1 και joint-2, και δύο κινούμενους συνδέσμους, link-1 και link-2), εκ των οποίων μόνο η $1^{η}$ άρθρωση της βάση (joint-1) ενεργοποιείται από έναν κινητήρα συνεχούς (DC motor), όπως φαίνεται στο Σχήμα 1 (βλ. [2]).



Σχήμα 1: Κινηματική δομή ρομποτικού μηχανισμού Pendubot

Και οι δύο αρθρώσεις είναι εφοδιασμένες με οπτικό κωδικοποιητή (encoder) για τη μέτρηση -κατά τα γνωστά- της γωνίας στροφής (q_1 και q_2) σε κάθε άρθρωση.
Ο προγραμματισμός του συστήματος γίνεται σε προγραμματιστικό περιβάλλον C/C++, με κλήση καταλλήλων συναρτήσεων: (α) για την "εγγραφή" σε κάρτα ελέγχου (D/A, CIO-DAC-02 της Computer Boards Inc.) των εντολών πρός την $1^η$ οδηγούμενη ρομποτική άρθρωση (έξοδος: αναλογική τάση μετατρεπόμενη, μέσω ενός PWM servoamplifier, σε σταθμιζόμενο ρεύμα, δηλαδή τελικά σε ελεγχόμενη ροπή κινητήρα), καθώς και (β) για την "ανάγνωση" από κάρτα διεπαφής με τους οπτικούς κωδικοποιητές (encoder interface card της Dynamics Research Corporation) της τρέχουσας "θέσης" κάθε άρθρωσης (βλ. [3]).

3. Λήψη Πειραματικών Μετρήσεων – Σχεδίαση Ελεγκτών

Η εκτέλεση της εργαστηριακής αυτής άσκησης πρέπει να ακολουθήσει τα εξής βήματα:

3.1. Πειραματική Ρύθμιση Ελεγκτή PD μεμονωμένης Ρομποτικής Άρθρωσης

Με είσοδο μια βηματική συνάρτηση (π.χ. $q_{1d}(t)=30^o$, για $t \geq 0$), και εκτελώντας έλεγχο θέσης μεμονωμένα της $1^{ης}$ άρθρωσης με έναν "τοπικό" ελεγκτή PD, ζητείται:

(α) Να γίνει διαδοχική ρύθμιση του κέρδους Kp και Kv του ελεγκτή PD για την $1^{η}$ άρθρωση του μηχανισμού. Μεταβάλλοντας το αναλογικό κέρδος Kp μέσα σε ένα κατάλληλο εύρος τιμών (ξεκινώντας από μια μικρή τιμή, μέχρι την εμφάνιση μονίμων ταλαντώσεων στο σύστημα), ζητείται να ευρεθεί το "βέλτιστο" Kv (για κάθε τιμή του κέρδους Kp) που ελαχιστοποιεί την υπερύψωση (overshoot).

(β) Να ληφθούν μετρήσεις για το μόνιμο σφάλμα θέσης και το overshoot για κάθε ζεύγος τιμών (Kp, Kv), και να παρασταθεί γραφικά η "πορεία" του βελτίστου (Kp,Kv) για αυξανόμενο αναλογικό κέρδος.

3.2. Επίδραση μη-γραμμικών όρων του ρομποτικού δυναμικού μοντέλου

(γ) Να μελετηθεί πειραματικά η επίδραση των όρων της βαρύτητας στον έλεγχο θέσης της $1^{ης}$ άρθρωσης (βάρος πάνω στους κινούμενους συνδέσμους), χρησιμοποιώντας τις βέλτιστες τιμές του ανωτέρω τοπικού ελεγκτή ρομποτικής άρθρωσης PD. Να καταγραφεί το μόνιμο σφάλμα θέσης, με αύξηση του βάρους των συνδέσμων, και να μελετηθεί η πιθανή μείωση στο περιθώριο ευσταθείας του συστήματος.

(δ) Να τροποποιηθεί κατάλληλα ο ανωτέρω ελεγκτής (της παραγράφου 3.1) με την προσθήκη κατάλληλου μη-γραμμικού όρου για την "ενεργό αντιστάθμιση" της βαρύτητας. Να καταγραφούν εκ νέου τα χαρακτηριστικά επίδοσης του συστήματος για τον έλεγχο θέσης της $1^{ης}$ άρθρωσης.

3.3. Πειραματική Μελέτη Μη-γραμμικού Ρομποτικού Ελεγκτή

(ε) Να πειραματισθείτε με τη λειτουργία του μη γραμμικού ελεγκτή εξισορρόπησης του αναστρόφου εκκρεμούς στην ενδιάμεση θέση ασταθούς ισορροπίας (middle balancing position), ο οποίος περιγράφεται στα [1] και [2]. Ο ελεγκτής αυτός αποτελείται από δύο φάσεις ελέγχου: τη φάση ταλάντωσης (swing up control phase) και τη φάση εξισορρόπησης (balancing control phase). Να παρατηρήσετε ιδιαίτερα την ιδιαιτερότητα του συστήματος αυτού, στο οποίο μόνο η $1^η$ άρθρωση οδηγείται άμεσα με έναν κινητήρα συνεχούς, ενώ η $2^η$ άρθρωση δεν οδηγείται (είναι δηλαδή ελεύθερη, unactuated), και κινείται έμμεσα, μέσω των επιταχύνσεων του $1^{ου}$ συνδέσμου λόγω της σύζευξης των δύο αρθρώσεων στο ρομποτικό δυναμικό μοντέλο. Τέτοια συστήματα ονομάζονται "υπο-οδηγούμενα" (underactuated), διότι διαθέτουν λιγότερα κινητήρια στοιχεία από αρθρώσεις (δηλαδή διαθέτουν ελεύθερους βαθμούς ελευθερίας).

4. Παραδοτέο

Να παραδοθεί γραπτή αναφορά, η οποία να περιγράφει τα θεωρητικά και πειραματικά βήματα, καθώς και τα συμπεράσματά σας κατά την εκτέλεση της εργαστηριακής αυτής άσκησης.

Βιβλιογραφία

[1] Block, D.J., and Spong, M.W., "Mechanical Design & Control of the Pendubot," SAE Earthmoving Industry Conference, IL, April 4-5, 1995.

[2] Spong, M.W., and Block, D.J., "The Pendubot: A Mechatronic System for Control Research and Education," 34th IEEE Conf. on Decision and Control (CDC'95), pp. 555-556, New Orleans, Dec., 1995.

[3] "The Pendubot: User's Manual," Mechatronic Systems Inc.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ/ΚΩΝ & ΜΗΧ/ΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

«Εργαστήριο Ρομποτικής»

ΠΑΡΑΡΤΗΜΑ

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ – ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

SAE TECHNICAL PAPER SERIES

951199

Mechanical Design and Control of the Pendubot

Daniel J. Block and Mark W. Spong
University of Illinois

The Engineering Society For Advancing Mobility Land Sea Air and Space®
INTERNATIONAL

46th Annual Earthmoving Industry Conference
Peoria, Illinois
April 4 & 5, 1995

400 Commonwealth Drive, Warrendale, PA 15096-0001 U.S.A. Tel: (412)776-4841 Fax:(412)776-5760

Mechanical Design and Control of the Pendubot

Daniel J. Block and Mark W. Spong
University of Illinois

ABSTRACT

In this paper we demonstrate our work to date on our underactuated two link robot called the Pendubot. First we will overview the Pendubot's design, discussing the components of the linkage and the interface to the PC making up the controller. Parameter identification of the Pendubot is accomplished both by solid modeling methods and energy equation least squares techniques. With the identified parameters, mathematical models are developed to facilitate controller design. The goal of the control is to swing the Pendubot up and balance it about various equilibrium configurations. Two control algorithms are used for this task. Partial feedback linearization techniques are used to design the swing up control. The balancing control is then designed by linearizing the dynamic equations about the desired equilibrium point and using LQR or pole placement techniques to design a stabilizing controller.

INTRODUCTION

At the University of Illinois, extensive research and development has gone into the concept and design of the two link underactuated planar robot called the Acrobot [1]. To extend this research in underactuated planar linkages we came up with the concept of the Pendubot. It is a counter part of the Acrobot having two links, but, instead of having the actuation at its elbow, the Pendubot is driven at its shoulder joint. This makes for a slightly simpler plant and control design when compared with the Acrobot, but all similar control topics can be study and implemented. The goal of the Pendubot controller is to swing the linkages from their stable downward position to unstable equilibrium points and then catch the unactuated link and balance it there.

In the first section of this paper we explain the design of the Pendubot. In the second section we study parameter identification methods to identify the Pendubot's dynamic parameters. This method uses the energy equations of the Pendubot to form a least squares problem that can be solved for the unknown parameters. The advantage of this method is that is does not require the realization of acceleration.

The next three sections discuss the control algorithms used to swing up and balance the links at unstable equilibrium points. For the swing up control we use partial feedback linearization methods discussed in [2] and [3]. The balancing control was found by linearizing the system and designing a full state feedback controller for the linearized model. Finally in the last section we show the results of actual runs of the Pendubot system.

MECHANICAL DESIGN AND CONTROLLER INTERFACE

When designing the Pendubot our goal was to stay as simple as possible. For this reason we made the actuated joint directly driven by a high torque 90VDC permanent magnet motor. See Figure 1 for 2D perspectives of the Pendubot and Figure 2 for a photograph of the Pendubot in its upright, "top", balancing position.

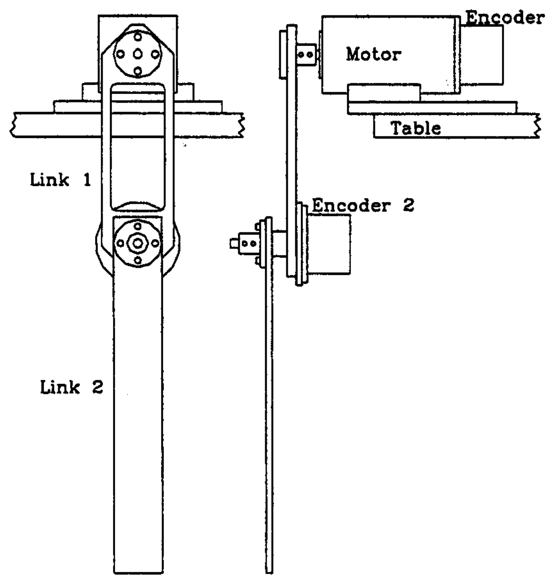


Figure 1: Front and side perspective drawings of the Pendubot.

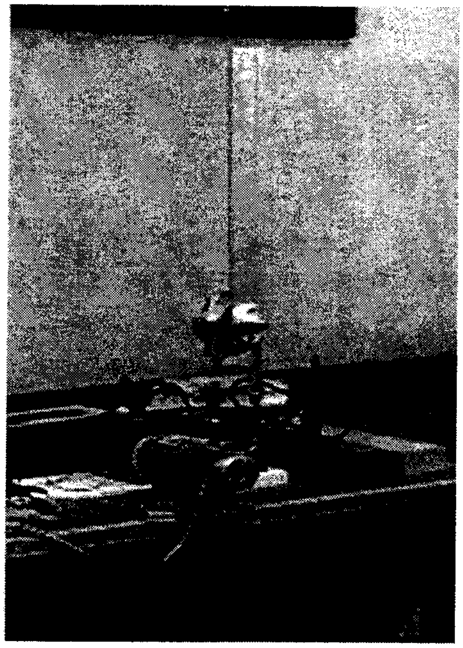


Figure 2: Photograph of the Pendubot in its top balancing position.

To give joint one direct drive control, we designed the Pendubot to hang off the side of a table, coupling link one directly to the motor's shaft. The motor's mount and bearings are then the support for the entire system. Link one also includes the bearing housing for joint two's motion. Here needle roller bearings riding on a ground shaft were used to construct joint two's revolute joint. The shaft extends out both directions of the housing allowing coupling to link two and to an optical encoder mounted on link one producing link two's position feedback. The design gives both links full 360° of motion. Link one cannot continuously rotate due to link two's encoder cable. Link two, however, has no constraint on continuous revolutions.

Link two is simply a quarter inch (0.635 cm) thick length of aluminum with a coupling that attaches to joint two's shaft.

The lengths of the links were arrived at initially by intuition and earlier work on the Acrobot and then confirmed with simulation. The intuition comes from thinking about balancing a broom or a similar object in the palm of your hand. The longer the broom the easier it is too balance. Of course if it gets too long it is too heavy to hold and in the case of the Pendubot even harder to swing up from the hanging position. A length of 14 inches (35.56 cm) was chosen for link two. This gave it a good center of mass location without creating too heavy of a link.

Designing link one's length is a little different. It needs to have some length and good stiffness so it can quickly get under link two when balancing, but the heavier it is the more torque the motor must produce. A length of 8 inches (20.32cm) was chosen for link one and the center material of the link was removed (See Figure 1).

To test our intuition on the length and weights of the links, we ran simulations of the Pendubot and its controller. AutoCAD's solid modeling extension was used to get an approximation of the dynamic parameters of the system and Simnon [4] was used to simulate the Pendubot's dynamic equations and controller. The design was confirmed by observing that the control effort remained less than the maximum torque of the motor when swinging the links to their upright position and balancing them there.

The final component of the Pendubot's hardware is its controller. BEI 1024 counts/rev resolution optical encoders, one attached at the elbow joint and one attached to the motor, were used as the feedback mechanism for the joint angles. Advanced Motion Control's 25A8 PWM servo amplifier was used to drive the motor. In the control algorithm this amplifier can be thought of as just a gain. In the case of the Pendubot we setup the amplifier in torque mode and adjusted it for a gain of 1V=1.2Amps.

In an attempt to simplify the Pendubot's controller and minimize its cost we ran our control algorithm using only the microprocessor on our PC instead of purchasing an expensive DSP card. We used a 486DX2/50 IBM compatible PC with a D/A card and an encoder interface card. Keithley Metrabyte's DAC-02 card was used for the digital to analog converter and Technology 80's 5312B was used to interface with the optical encoders. Then with the software library routines accompanying the cards we were able to write a C program to form the controller's algorithm.

The only difficulty using a PC as a controller is finding a way to reliably get a fast sampling period. DOS does produce a clock pulse, but it only occurs every 55ms, making it useless for this system which needs at least a 10 ms sampling period. To get around this, the PC's timer chip needed to be directly programmed to give a higher resolution. Ryle Design's "PC Timer Tools" software includes an alarm algorithm that can be used to produce an appropriate sampling period (i.e. 5ms). The format of the control algorithm then is as follows:

```
/* Perform all needed initializations */
/* start 5ms alarm */
while (Continue_Control==TRUE) {
   /* sample encoder positions */
   /*  use finite differences to calculate velocity */
   /* calculate needed control effort */
   /* output control value to motor */
   while (Alarm_expired == FALSE) {
     /* continue to loop until alarm expires */
   } /***** end of second while *****/
} /***** end of first while ****/
```

This control design worked very well. We were able to reliably achieve a 1ms sampling period even when computing the inverse dynamic equations for the partial feedback linearization control. A 5 ms sampling period was used by most of our controllers in order to allow us to save response data while the controller was operating. Using a 5 ms sampling period will also leave us room to update our controller with a Windows GUI interface, which requires more overhead running in the Windows operating system.

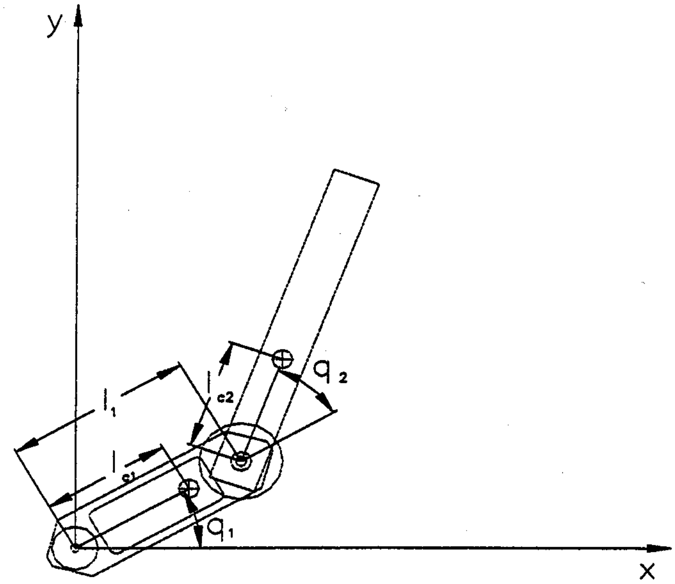


Figure 3: Pictorial of the Pendubot. l_1 is the length of link one, l_{c1} and l_{c2} are the distances to the center of mass of the respective links and q_1 and q_2 are the joint angles of the respective links.

SYSTEM MODEL

The equations of motion for the Pendubot can be found using Lagrangian dynamics [5]. In matrix form the equations are

$$\tau = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) \quad (1)$$

where $\boldsymbol{\tau}$ is the vector of torque applied to the links and $\mathbf{q}$ is the vector of joint angle positions with

$$D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

$$d_{11} = m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos q_2) + I_1 + I_2$$

$$d_{12} = d_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2$$

$$d_{22} = m_2 l_{c2}^2 + I_2$$

and

$$C(q,\dot{q}) = \begin{bmatrix} h\dot{q}_2 & h\dot{q}_2 + h\dot{q}_1 \\ -h\dot{q}_1 & 0 \end{bmatrix}$$

$$h = -m_2 l_1 l_{c2}\sin q_2$$

and

$$g(q) = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

$$\phi_1 = (m_2 l_{c1} + m_2 l_1) g\cos q_1 + m_2 l_{c2} g\cos(q_1 + q_2)$$

$$\phi_2 = m_2 g l_{c2}\cos(q_1 + q_2)$$

m_1, the total mass of link one,
l_1, the length of link one (See Figure 3),
l_{c1}, the distance to the center of mass of link 1 (See Figure 3),
I_1 ,the moment of inertia of link one about its centroid,
m_2, the total mass of link two,
l_{c2}, the distance to the center of mass of link 2 (See Figure 3),
I_2, the moment of inertia of link two about its centroid,
g, the acceleration of gravity.

From the above equations it is observed that the seven dynamic parameters can be grouped into the following five parameter equations

$$\begin{aligned} \theta_1 &= m_1 l_{c1}^2 + m_2 l_1^2 + I_1 \\ \theta_2 &= m_2 l_{c2}^2 + I_2 \\ \theta_3 &= m_2 l_1 l_{c2} \\ \theta_4 &= m_1 l_{c1} + m_2 l_1 \\ \theta_5 &= m_2 l_{c2} \end{aligned} \quad . \quad (2)$$

For a control design that neglects friction, these five parameters are all that are needed. There is no reason to go a step further and find the individual parameters since the control equations can be written with only the five parameters. Substituting these parameters into the above equations leaves the following matrices

$$D(q) = \begin{bmatrix} \theta_1 + \theta_2 + 2\theta_3\cos q_2 & \theta_2 + \theta_3\cos q_2 \\ \theta_2 + \theta_3\cos q_2 & \theta_2 \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} -\theta_3\sin(q_2)\dot{q}_2 & -\theta_3\sin(q_2)\dot{q}_2 - \theta_3\sin(q_2)\dot{q}_1 \\ \theta_3\sin(q_2)\dot{q}_1 & 0 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} \theta_4 g\cos q_1 + \theta_5 g\cos(q_1 + q_2) \\ \theta_5 g\cos(q_1 + q_2) \end{bmatrix}. \quad (3)$$

Finally, using the invertible property of the mass matrix, D(q), the state equations are given by

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = D(q)^{-1}\tau - D(q)^{-1}C(q,\dot{q})\dot{q} - D(q)^{-1}g(q)$$

$$\begin{aligned} & x_1 = q_1, x_2 = \dot{q}_1, x_3 = q_2, x_4 = \dot{q}_2 \\ & \dot{x}_1 = x_2 \\ & \dot{x}_2 = \ddot{q}_1 \\ & \dot{x}_3 = x_4 \\ & \dot{x}_4 = \ddot{q}_2 \end{aligned} \quad .(4)$$

IDENTIFICATION USING THE ENERGY THEOREM

The next step is to identify the parameters listed above, EQ (2). The identification scheme we chose uses the energy theorem to form equations that can be solved for the unknown parameters by a least squares problem [6].

The kinetic energy of the Pendubot is written as

$$K = \frac{1}{2}\dot{q}D(q)\dot{q}$$

where D(q) is defined by EQ (3). Performing the matrix multiplication produces the following equation for the kinetic energy

$$K = \tfrac{1}{2}\dot{q}_1^2\theta_1 + (\tfrac{1}{2}\dot{q}_1^2 + \dot{q}_1\dot{q}_2 + \tfrac{1}{2}\dot{q}_2^2)\theta_2 + (\cos q_2\dot{q}_1^2 + \cos q_2\dot{q}_1\dot{q}_2)\theta_3. \quad (5)$$

The potential energy of the Pendubot is written

$$V=(m_1 l_{c1}+m_2 l_1)g\sin q_1+m_2 l_{c2} g\sin(q_1+q_2).$$

In terms of the parameters to be identified it is simplified to

$$V=\theta_4 g\sin q_1+\theta_5 g\sin(q_1+q_2). \quad (6)$$

Looking at the above equations it is observed that the kinetic and potential energy equations are both linear in the inertial parameters. A simple way to write these equations then is

$$K=\sum_{i=1}^{5}\frac{\partial K}{\partial\theta_i}\theta_i=\sum_{i=1}^{5}DK_i\theta_i$$

$$V=\sum_{i=1}^{5}\frac{\partial V}{\partial\theta_i}\theta_i=\sum_{i=1}^{5}DV_i\theta_i.$$

For the Pendubot the new terms DK and DV are

$$DK_1=\tfrac{1}{2}\dot q_1^2$$
$$DK_2=\tfrac{1}{2}\dot q_1^2+\dot q_1\dot q_2+\tfrac{1}{2}\dot q_2^2$$
$$DK_3=\cos q_2\dot q_1^2+\cos q_2\dot q_1\dot q_2$$
$$DK_4=0$$
$$DK_5=0$$
$$DV_1=0$$
$$DV_2=0$$
$$DV_3=0$$
$$DV_4=g\sin q_1$$
$$DV_5=g\sin(q_1+q_2)$$

The energy theorem which states that the work of forces applied to a system is equal to the change of the total energy of the system can be written as

$$\int_{t_1}^{t_2}\mathbf{T}^T\dot{\mathbf{q}}dt=(K(t_2)+V(t_2))-(K(t_1)+V(t_1))=L(t_2)-L(t_1) \quad (7)$$

where $L(t_i)$ is the total energy at time t_i, $L(t_i)=K(t_i)+V(t_i)$, and **T** is the vector of torque applied at the joints. **T** includes both the motor torque and the friction forces and can be written

$$\mathbf{T}=\tau+\Gamma_f.$$

For this study we neglected friction setting Γ_f to zero.

Again using the property that K and V are linear in the inertial parameters, the difference in the total energy is defined $L(t_1)-L(t_2)=\mathbf{DL}^T\theta$

where:

$$\mathbf{DL}^T=[DL_1(t_2)-DL_1(t_1)\ \ \dots\ \ DL_m(t_2)-DL_m(t_1)]$$

and

$$DL_i(t_k)=DK_i(t_k)+DV_i(t_k).$$

This leaves the energy equation in the form

$$\int_{t_1}^{t_2}\mathbf{T}^T\dot{\mathbf{q}}dt=\mathbf{DL}^T\theta. \quad (8)$$

Defining a new matrix d, $d^T=DL^T$, the energy equation can be written in the standard least squares form

$$\left(\int_{t_{K-1}}^{t_K}\mathbf{T}^T\dot{\mathbf{q}}dt\right)_k=d_k^T\theta$$

where the subscript k denotes the number of time intervals or equations used in the least squares problem. Also since $DL_i(0)=0$ for (i=1,...,5), we then can write this equation as

$$\left(\int_{t_0}^{t_K}\mathbf{T}^T\dot{\mathbf{q}}dt\right)_k=d_k^T\theta$$

To implement this identification scheme we wrote a simple program that drove the Pendubot with an open loop signal and at the same time recorded the response of the system. This response data was then loaded into Matlab where the identification algorithm could be performed. To approximate the integral on the left hand side of the least squares problem the backwards trapezoidal rule was used. The resulting Matlab M-file was as follows:

```
%q1 and q2 are collected joint positions
%dq1 and dq2 are collected joint velocities
%tau is the open loop control effort
g=386; (SI Units = 9.8)
dL1 = (.5*dq1.^2);
dL2 = (.5*dq1.^2 + dq1.*dq2 + .5*dq2.^2);
dL3 = (cos(q2).*(dq1.^2 + dq1.*dq2));
dL4 = (g*sin(q1));
dL5 = (g*sin(q1+q2));
taudq1 = tau.*dq1;
for i = 1:(length(dL1)-10),
DL(i,1) = dL1(i+10)-dL1(1);
DL(i,2) = dL2(i+10)-dL2(1);
DL(i,3) = dL3(i+10)-dL3(1);
DL(i,4) = dL4(i+10)-dL4(1);
DL(i,5) = dL5(i+10)-dL5(1);
Itq(i,1) = trapz(t(1:i+10,1),taudq1(1:i+10,1));
end
theta = nnls(DL,Itq).
```

Matlab's nnls(A,b) function returns the non-negative least squares solution to Ax=b.

Different open-loop inputs were tried in an attempt to see which best identified the system. A simple step input was found to work well giving the most consistent results. The parameters found for the Pendubot were

$\theta_1=0.0799$ V*s^2
$\theta_2=0.0244$ V*s^2
$\theta_3=0.0205$ V*s^2
$\theta_4=0.0107$ V*s^2/in (0.42126 V*s^2/m)
$\theta_5=0.0027$ V*s^2/in (0.10630 V*s^2/m).

To check the accuracy of these results they were compared with an AutoCAD Solid Model of the Pendubot. After taking into account the amplifier gain, Kamp =1.2A/V, and the motor's torque constant, K_T =3.546 lbin/A (0.4006 Nm/A), the solid model parameters were

$\theta_1=0.089252$ V*s^2
$\theta_2=0.027630$ V*s^2
$\theta_3=0.023502$ V*s^2
$\theta_4=0.011204$ V*s^2/in (0.44110 V*s^2/m)
$\theta_5=0.002938$ V*s^2/in (0.11567 V*s^2/m).

These parameters compare with the actual parameters found. The solid model produced slightly higher numbers but this is probably due to some over sizing when making assumptions in the model of the Pendubot or static friction which we neglected.

We did attempt to add the friction components to the identification algorithm. Unfortunately we were unable to find conclusive results for the friction terms. The friction in the Pendubot system is very low which may be the reason the energy based identification algorithm does not identify it well. As pointed out in Prüfer, Schmidt and Wahl [7] the energy based algorithm generally has difficulty identifying friction terms. The parameters found ignoring friction, as we will demonstrate, work very well in controlling the system. For this reason we did not dwell on the identification and continued on with the control design.

SWING UP CONTROL

As stated earlier the goal of the Pendubot control is to swing the links from their stable hanging position to unstable equilibrium positions and then balance the links about that equilibrium. This control is broken into to two parts, swing up control and balancing control. For the swing up control we chose to demonstrate the use of partial feedback linearization. Many different control algorithm's could have been used to perform the swing up. In fact initially we used a PID controller servoing around only link one's position to swing up the Pendubot. This worked fine but amplified numerical noise. Partial Feedback Linearization needs both link one and link two's position feedback but takes into account the nonlinear effects of link one and two. This creates a much cleaner control compared to a PID control that must reject the effects of the first and second link.

We will now derive the partial feedback control for the Pendubot. To see a general derivation of partial feedback linearization please refer to [2].

The equations of motion of the Pendubot are given by EQ (1). Performing the matrix and vector multiplication the equations of motion are written

$$\tau_1 = d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{11}\dot{q}_1 + c_{12}\dot{q}_2 + \phi_1 \quad (9)$$
$$0 = d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + \phi_2. \quad (10)$$

Due to the underactuation of link two we can not linearize the dynamics about both degrees of freedom. We can however linearize one of the degrees of freedom. This will allow us to design an outer loop control that will track a given trajectory for the linearized degree of freedom. In the case of the Pendubot we chose to linearize about the collocated degree of freedom q_1. EQ (10) was solved for link two's angular acceleration

$$\ddot{q}_2 = \frac{-d_{21}\ddot{q}_1 - c_{21}\dot{q}_1 - \phi_2}{d_{22}}.$$

This was then substituted into EQ (9) and written as

$$\tau_1 = \bar{d}_{11}\ddot{q}_1 + \bar{c}_{11}\dot{q}_1 + \bar{c}_{12}\dot{q}_2 + \bar{\phi}_1$$

with

$$\bar{d}_{11} = d_{11} - \frac{d_{12}d_{21}}{d_{22}}$$
$$\bar{c}_{11} = c_{11} - \frac{d_{12}c_{21}}{d_{22}}$$
$$\bar{c}_{12} = c_{12}$$
$$\bar{\phi}_1 = \phi_1 - \frac{d_{12}\phi_2}{d_{22}}.$$

Then just as with the full linearization method (also called the computed torque method [5]) the inner loop control that linearizes q_1's degree of freedom can be defined as

$$\tau_1 = \bar{d}_{11}v_1 + \bar{c}_{11}\dot{q}_1 + \bar{c}_{12}\dot{q}_2 + \bar{\phi}_1.$$

This results in the system

$$\ddot{q}_1 = v_1 \quad (11)$$
$$-d_{21}v_1 = d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + c_{22}\dot{q}_2 + \phi_2. \quad (12)$$

Since EQ (11) is now linear, an outer loop control can be designed to track a given trajectory for link one. Link two's response then is given by the resulting nonlinear equation, EQ (12). EQ (12) represents internal dynamics with respect to an output $y = q_1$. The goal of the outer loop control then is to track a given trajectory for link one and at the same time excite the internal dynamics to swing link two to a balancing position. For the Pendubot we chose to use a PD with feedforward acceleration

$$v_1 = \ddot{q}_1^d + K_d(\dot{q}_1^d - \dot{q}_1) + K_p(q_1^d - q_1).$$

Now given this controller a trajectory needs to be determined to swing the links to their unstable equilibrium position. To swing the links to the upright position ($q_1 = \pi/2, q_2 = 0$) we simply used a step trajectory $q_1^d = \pi/2$. This worked very well in simulation but when trying it out on the actual system the motor's instantaneous torque was not strong enough to swing the second link all the way to its upright position. To get around this we added to the swing up control a small open loop step that sent the link in the negative direction for a short period of time adding potential energy into the system. This added energy allowed the motor to excite the internal dynamics more and swing both links to the upright position.

The swing up trajectory to swing the links to the middle position ($q_1 = -\pi/2, q_2 = \pi$) was a little more difficult (See Figure 4).

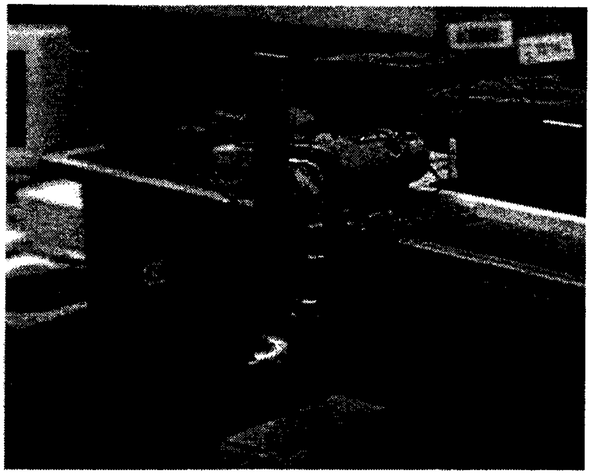


Figure 4: Photograph of the Pendubot in its middle balancing position.

Simulations were run to find a trajectory that would work well. The trajectory found for the swing up can be written

$q_1^d = 1.4\sin(5t) - \pi/2 : t < 2\pi/5$

$q_1^d = -\pi/2 \quad : t > 2\pi/5.$

This trajectory pumps energy into the system by causing the second link to swing back and forth and finally up to its middle equilibrium point.

Finally to fine tune the swing up control the K_p and K_d gains are adjusted. The idea is to find the correct gains that swing the second link slowly into its equilibrium position so that the balancing controller can catch and balance the link.

BALANCING CONTROL

The control for balancing the Pendubot is very similar to the classical cart and broom balancing problem. To design the controller we linearized the Pendubot's nonlinear equations of motion, EQ (4), and designed a full state feedback controller with the linear model. The Taylor series

$$f_a(x,u) = f_a(x_r, u_r) + \frac{\partial f}{\partial x}\bigg|_{x_r,u_r}(x - x_r) + \frac{\partial f}{\partial u}\bigg|_{x_r,u_r}(u - u_r)$$

was used to linearize the plant. x is the vector of states given in EQ (4). u is the single control input for the Pendubot. x_r and u_r are the equilibrium values of the states and control respectively. Since we are only interested in controlling the Pendubot at equilibrium positions $f_a(x_r,u_r)$ will always be zero. All that is needed then is to find the partial derivative matrices and evaluate them at the equilibrium points. Studying EQ (3) and (4) it is observed that the Pendubot's equilibrium points can be defined by

$$u_r = \theta_4 g \cos(x_{r1})$$
$$x_{r1} + x_{r3} = \pi/2.$$

Differentiating EQ (4) leaves

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & \frac{\partial f_2}{\partial x_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & 0 & \frac{\partial f_4}{\partial x_3} & 0 \end{bmatrix} \tag{13}$$

and

$$\frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix} \tag{14}$$

Each equilibrium point defines a different linearized system. This means that different control gains will be needed for each equilibrium point for best results in the balancing of the Pendubot. Most of our work with the Pendubot dealt with two equilibrium points. We define the "top" balancing position as the up right position with $x_{r1}=\pi/2, x_{r3}=0$ and ur = 0 (See Figure 2). The "mid" balancing position is defined as $x_{r1}=-\pi/2, x_{r3}=\pi$ and ur = 0 (See Figure 4).

Using these equilibrium values and the parameters identified in the system identification section, the linear models for the top and mid equilibrium positions are as follows

<u>Top</u>

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 51.9265 & 0 & -13.9704 & 0 \\ 0 & 0 & 0 & 1 \\ -52.8402 & 0 & 68.4210 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 15.9549 \\ 0 \\ -29.3596 \end{bmatrix}$$

Mid

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -51.9265 & 0 & 13.9704 & 0 \\ 0 & 0 & 0 & 1 \\ 51.0128 & 0 & 40.4801 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 15.9549 \\ 0 \\ -2.5502 \end{bmatrix}$$

Now given these linear models we can use LQR or pole placement techniques to design full state feedback controllers u=-Kx. For example with

$$\mathbf{R} = [1]$$

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

Matlab's "LQRD" function can be used to derive optimal control gains for a discrete controller. For example using a sampling period of 5ms the optimal gains are

$$\mathbf{K}_{Top} = \begin{bmatrix} -32.68 & -7.14 & -32.76 & -4.88 \end{bmatrix}$$

$$\mathbf{K}_{Mid} = \begin{bmatrix} 10.96 & 1.44 & 19.28 & 2.81 \end{bmatrix}.$$

COMBINING AND IMPLEMENTING THE CONTROLLERS

With both the swing up control and the balancing control complete an algorithm was needed to connect the two. Initially when working with only computer simulations of the Pendubot, the controllers were switched at a determined time. This switch time was determined by observing when the swing up control had brought the links almost to rest at the desired equilibrium position. This worked very well for the simulation but behaved poorly when realized on the actual Pendubot. The reason being that the simulations are exactly repeatable but the actual runs are susceptible to different initial conditions and computational noise making them unable to repeat exactly. The following algorithm was used to give the Pendubot more intelligence and switch the control by watching the states of the system.

```
if |x1- xr1| < .10 {
  if |x3- xr3| < .20 {
    u = -Kx;
    if |u| < 9 /* Volts */ {
      /* Output balancing control */
    } else /* Output swing up control */
  } else  /* Output swing up control */
} else /* Output swing up control */
```

This algorithm waits for link one to arrive within 0.1 radians of its equilibrium position and then checks link two. If link two is also within 0.20 radians of its equilibrium position, the balancing control is calculated. If the control output is less then 9, 10 being the maximum DAC output for the Pendubot, the control is switched to the balancing mode. Otherwise the links are passing too quickly though the equilibrium point and the swing up control remains in tack.

Another implementation issue that arises in the controller design of the Pendubot is the approximation of the joint's velocities. There is only position feedback in the system so a finite approximation is found. To find the velocity we simply used the finite difference method (x-xold)/sample period. This creates numerical error or noise in the calculation of the control effort, though, due to the finite resolution of the optical encoders. We found that simply taking the average of the last three velocities helped to filter and decrease this noise.

EXPERIMENTAL RESULTS

This final section shows actual responses of the Pendubot system. The balancing controller gains for these runs were found by pole placement methods and are listed in the figure's description. Figures 5 and 6 show the Pendubot swinging and balancing at the top and mid positions respectively. Figure 7 demonstrates the Pendubot's ability to balance at its many equilibrium points. The control used for Figure 7 swings the links to the mid position and then steps the links at 5° increments away from the mid position every two seconds. Each new equilibrium point has its own balancing control gains and an equilibrium control ur = $\Theta_4 g\cos(x_{r1})$ which is no longer zero.

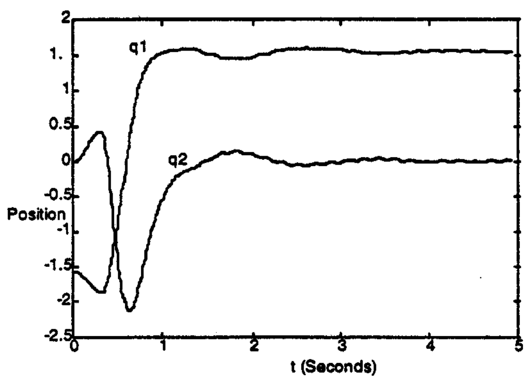


Figure 5: Swing-Up and Balance Control at the Top Position. Outer-Loop control gains used for the swing-up control: Kp=150.0,Kd=21.4. Full state feedback gains used for the balancing control: K=[-27.48 -6.07 -28.58 -4.24].

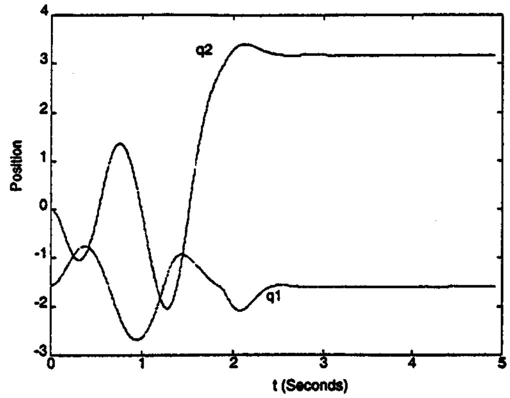


Figure 6: Swing-Up and Balance Control at the Mid Position. Outer-Loop control gains used for the swing-up control: Kp=32.0,Kd=4.85. Full state feedback gains used for the balancing control: K=[15.31 1.76 22.86 3.38]. Swing-up trajectory: $1.4\sin(5t) - \pi/2$.

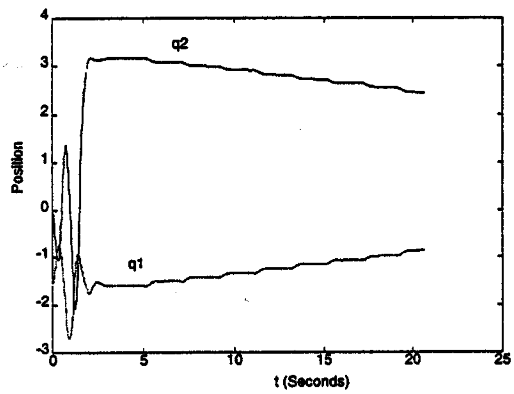


Figure 7: This plot first shows the swing-up and balance control at the mid position identical to Figure 6. This plot also goes further and demonstrates the Pendubot's balancing capabilities at other equilibrium points. Full state feedback gains change for each equilibrium point.

CONCLUSION

This paper presented our new design of a two link underactuated planar revolute robot, named the Pendubot. Its actuated joint is located at the shoulder and the elbow joint is unactuated and allowed to swing free. The controller for the Pendubot was implemented using data acquisition cards and an IBM compatible 486DX2 PC. Parameter identification was performed using a method that takes advantage of the energy equations of the linkage which are linear in terms of the unknown dynamic parameters. A simple least squares problem was then derived to solve for the parameters. Two controllers were designed for the Pendubot. Partial feedback linearization techniques were used to design the control that swung the links from their hanging stable position to unstable equilibrium positions. Then to catch and balance the second link at the unstable equilibrium, full state feedback controllers were designed using the linearized model of the links at the desired position. Results were presented demonstrating the performance of the Pendubot with these controllers.

REFERENCES

[1] Bortoff, S.A., Pseudolinearization using Spline Functions with Application to the Acrobot, Ph. D. Thesis, Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1992.

[2] Spong, M.W., "The Control of Underactuated Mechanical Systems", Plenary lecture at the First International Conference on Mechatronics, Mexico City, January 26-29, 1994.

[3] Spong, M.W., "Swing Up Control of the Acrobot using Partial Feedback Linearization," SY -ROCO'94, Capri, Italy, pp. 833-838, September, 1994.

[4] Elmquist, H., *Simnon-User's Guide,* Dept. of Automatic Control, Lund Inst. of Tech., CODEN:LUTFD@/(TFRT-3091), 1975.

[5] Spong, M.W., and Vidyasagar, M., *Robot Dynamics and Control,* John Wiley & Sons, Inc., New York, 1989.

[6] Gautier, M., and Khalil, W., "On the Identification of the Inertial Parameters of Robots. In *Proceedings of 27th CDC,* pages 2264-2269, 1988.

[7] Prüfer, M., Schmidt, C., Wahl, F., "Identification of Robot Dynamics with Differential and Integral Models: A Comparison", Proc. IEEE Int. Conf. on Robotics and Automation, San Diego, CA, pp. 340-345, May, 1994.

The Pendubot: A Mechatronic System for Control Research and Education

Mark W. Spong Daniel J. Block

Coordinated Science Laboratory
University of Illinois at Urbana–Champaign
m-spong@uiuc.edu

Abstract[1]

In this paper we describe the *Pendubot*, a mechatronic device for use in control engineering education and for research in nonlinear control and robotics. This device is a two–link planar robot with an actuator at the shoulder but no actuator at the elbow. With this system, a number of fundamental concepts in nonlinear dynamics and control theory may be illustrated. The pendubot complements previous mechatronic systems, such as the Acrobot [3] and the inverted pendulum of Furuta [4].

1 Introduction

In this paper we discuss the design, and control of the *Pendubot*, a two–link, underactuated robotic mechanism that we are using for research in nonlinear control and to educate students in various concepts in nonlinear dynamics, robotics, and control system design. The novelty of our system lies in the ease with which we are able to demonstrate advanced concepts such as partial feedback linearization and zero dynamics, both as a vehicle for research, and as an instructional device.

2 Description of the Hardware

Figure 1 shows a drawing of the Pendubot.

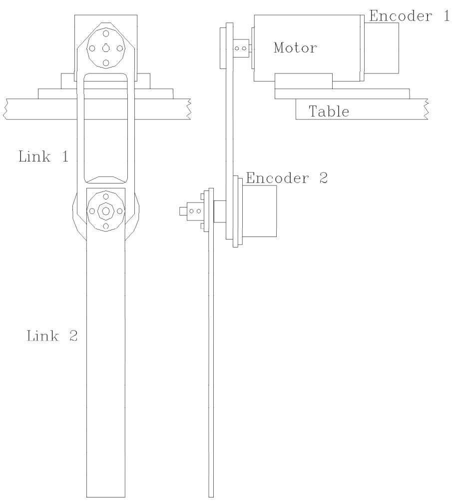


Figure 1: a. Front and Side Perspective Drawings of The Pendubot

[1]Research partially supported by the National Science Foundation under grants CMS-9402229 and IRI-9216428

The Pendubot consists of two rigid aluminum links of lengths 14in and 8in, respectively. Link 1 is directly coupled to the shaft of a 90V permanent magnet DC motor mounted to the end of a table. The motor mount and bearings are then the support for the entire system. Link 1 also includes the bearing housing for joint two. Needle roller bearings riding on a ground shaft were used to construct the revolute joint for link 2. The shaft extends out both directions of the housing allowing coupling to the second link and to an optical encoder mounted on link one. The design gives both links full 360^o of rotational motion. Link 2 is constructed of a $\frac{1}{4}$–inch (0.635 cm) thick length of aluminum with a coupling that attaches to the shaft of joint two.

All of our control computations are performed on a Dell 486DX2/50 PC workstation with a D/A card and an encoder interface card. Using the standard software library routines supplied with these interface cards we are able to program control algorithms directly in C.

3 Dynamics

Since our device is a two link robot (with only one actuator) its dynamic equations can be found in numerous robotics textbooks as

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + h_1 + \phi_1 = \tau \quad (1)$$
$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + h_2 + \phi_2 = 0 \quad (2)$$

where q_1, q_2 are the joint angles and τ is the input torque. The important distinction then between the system (1)–(2) and a standard two–link robot is, of course, the absence of a control input torque to the second equation (2). Underactuated mechanical systems generally have equilibria which depend on both their kinematic and dynamic parameters. If the Pendubot is mounted so that the joint axes are perpendicular to gravity, then there will be a continuum of equilibrium configurations, each corresponding to a constant value, $\bar{\tau}$, of the input torque τ. These equilibria are characterized by the second link vertical for any position of the first link.

4 Identification

The dynamic parameters of the Pendubot were identified two ways: using the AutoCAD drawings, and on–line using the Hamiltonian based approach of Gautier and Khalil [2]. Because of the low friction in our device the Hamilton based approach to identification works remarkably well.

4 The Pendubot as an Inverted Pendulum

In this section we discuss the application of the Pendubot as an inverted pendulum in which the motion of the actuated first link is used to balance the second link. This system is distinct from the more classical cart–pole system in which the linear motion of the cart is used to balance a pendulum. Our system may also be contrasted with the recent and very elegant variation due to Furuta [4] which mounts the pendulum at the end of a horizontally rotating first link. One may think of Furuta's device as a two link underactuated robot arm with the two joint axes orthogonal, while our Pendubot has the joint axes parallel.

4.1 Controllability and Balancing

The balancing problem for the Pendubot may be solved by linearizing the equations of motion about an operating point and designing a linear state feedback controller, very similar to the classical cart–pole problem, so we will only give a few remarks here. One very interesting distinction of the Pendubot over both the classical cart–pole system and Furuta's system is the continuum of balancing positions. This feature of the Pendubot is pedagogically useful in several ways, to show students how the Taylor series linearization is operating point dependent and for teaching controller switching and gain scheduling. Students can also easily understand physically how the linearized system becomes uncontrollable at $q_1 = 0, \pm\pi$. As the Pendubot approaches this uncontrollable configuration, the controllability matrix becomes increasingly ill–conditioned.

4.2 Swing Up Control

The problem of swinging the Pendubot up from the downward configuration to the inverted configuration is an interesting and challenging nonlinear control problem. With this problem one may illustrate the nonlinear control ideas of nonlinear relative degree, partial feedback linearization and zero dynamics.

We have used the method of partial feedback linearization to swing up the Pendubot, i.e., to move from the stable equilibrium $q_1 = -\pi/2$, $q_2 = 0$, to the inverted position $q_1 = \pi/2$, $q_2 = 0$. The same approach can easily be modified to swing the system to any point on its equilibrium manifold. The control is switched to a second control to balance the Pendubot about the equilibrium whenever the swing up controller moves it into the basin of attraction of the balancing controller. The actual control design and analysis is similar to our previous work with the Acrobot [3].

The experimental results have been very good. As a balancing controller we have used linear quadratic

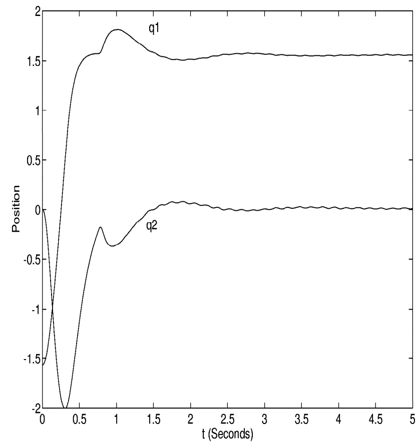


Figure 2: Swing-Up and Balance Control at the Top Position

methods. Figure 2 shows the Pendubot swinging and balancing at its top position using a partial feedback linearization swingup strategy.

5 Conclusions

This paper presents our concept of a two link underactuated planar revolute robot, named the Pendubot. This system is useful both in research and for instruction in controls. Students at all levels may benefit from this system. Various aspects of control theory from local linear state feedback to balance the Pendubot in one position to more complex global nonlinear controllers to swing up the Pendubot are easily illustrated. The reader is referred to [1] for additional details. In addition the full version of this paper is available upon request.

References

[1] Block, D.J., "Mechanical Design & Control of the Pendubot," *SAE Earthmoving Industry Conference*, Peoria, IL, April 4-5, 1995.

[2] Gautier, M., and Khalil, W., "On the Identification of the Inertial Parameters of Robots," *Proceedings of 27th IEEE CDC*, pages 2264-2269, 1988.

[3] Spong, M.W., "Swing Up Control of the Acrobot," *IEEE Control Systems Magazine*, Feb. 1995.

[4] Furuta, K., and Yamakita, M., "Swing up Control of Inverted Pendulum," In *IECON'91*, pp. 2193-2198, 1991.