3. SYSTEM MODEL

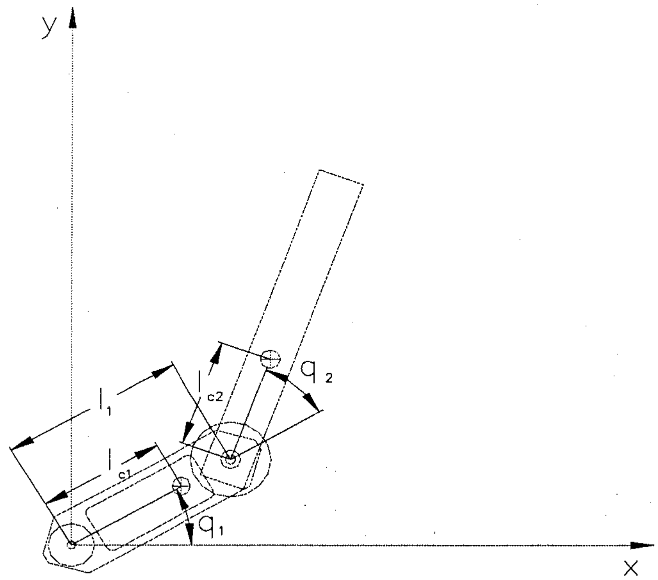

Figure 3.1 Coordinate Description of the Pendubot. l_1 is the length of link one, l_{c1} and l_{c2} are the distances to the center of mass of the respective links and q_1 and q_2 are the joint angles of the respective links.

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

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{3.1}$$

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

$$D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \qquad \begin{aligned} d_{11} &= m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos q_2) + I_1 + I_2 \\ d_{12} &= d_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2 \\ d_{22} &= m_2 l_{c2}^2 + I_2 \end{aligned} \tag{3.2}$$

and

$$\begin{aligned} C(q,\dot{q}) &= \begin{bmatrix} h\dot{q}_2 & h\dot{q}_2 + h\dot{q}_1 \\ -h\dot{q}_1 & 0 \end{bmatrix} \\ h &= -m_2 l_1 l_{c2}\sin q_2 \end{aligned} \tag{3.3}$$

and

$$g(q)=\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}$$
$$\phi_1=(m_1l_{c1}+m_2l_1)g\cos q_1+m_2l_{c2}g\cos(q_1+q_2)\ . \tag{3.4}$$
$$\phi_2=m_2gl_{c2}\cos(q_1+q_2)$$

m_1	:	the total mass of link one.
l_1	:	the length of link one (See Figure 3.1).
l_{c1}	:	the distance to the center of mass of link 1 (See Figure 3.1).
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 3.1).
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

$$\begin{aligned}\theta_1&=m_1l_{c1}^2+m_2l_1^2+I_1\\ \theta_2&=m_2l_{c2}^2+I_2\\ \theta_3&=m_2l_1l_{c2}\\ \theta_4&=m_1l_{c1}+m_2l_1\\ \theta_5&=m_2l_{c2}\end{aligned}\ . \tag{3.5}$$

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

$$D(q)=\begin{bmatrix}\theta_1+\theta_2+2\theta_3\cos q_2 & \theta_2+\theta_3\cos q_2\\ \theta_2+\theta_3\cos q_2 & \theta_2\end{bmatrix}, \tag{3.6}$$

$$C(q,\dot q)=\begin{bmatrix}-\theta_3\sin(q_2)\dot q_2 & -\theta_3\sin(q_2)\dot q_2-\theta_3\sin(q_2)\dot q_1\\ \theta_3\sin(q_2)\dot q_1 & 0\end{bmatrix}, \tag{3.7}$$

$$g(q)=\begin{bmatrix}\theta_4g\cos q_1+\theta_5g\cos(q_1+q_2)\\ \theta_5g\cos(q_1+q_2)\end{bmatrix}. \tag{3.8}$$

Finally, using the invertible property of the mass matrix D(q) [5], the state equations are given by

$$\begin{aligned}
&\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = D(q)^{-1}\tau - D(q)^{-1}C(q,\dot{q})\dot{q} - D(q)^{-1}g(q) \\
&x_1 = q_1, x_2 = \dot{q}_1, x_3 = q_2, x_4 = \dot{q}_2 \\
&\dot{x}_1 = x_2 \\
&\dot{x}_2 = \ddot{q}_1 \\
&\dot{x}_3 = x_4 \\
&\dot{x}_4 = \ddot{q}_2
\end{aligned} \tag{3.9}$$

4. SYSTEM IDENTIFICATION

4.1 CAD Solid Model

After formulating the mathematical model of the Pendubot, the next step was to identify the five parameters in equation (3.5). An AutoCAD solid model of the Pendubot was drawn to give approximate numbers for these parameters. As mentioned previously, these approximate parameters helped in the design of the Pendubot. Used in simulations, they allowed us to determine if the motor would be powerful enough to manipulate the two links. They also served as a guide to determine the accuracy of the on-line identification methods described in the next two sections. Taking into account the amplifier gain, Kamp =1.2A/V, and the torque constant of the motor, 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).

4.2 Energy Equation Method

This on-line identification scheme 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

$$K = \frac{1}{2}\dot{q}D(q)\dot{q} \tag{4.1}$$

where D(q) is defined by equation (3.6). Performing the matrix multiplication produces the following equation for the kinetic energy

$$K = \tfrac{1}{2}\dot{q}_1^2\theta_1 + (\tfrac{1}{2}\dot{q}_1^2 + \dot{q}_1\dot{q}_2 + \tfrac{1}{2}\dot{q}_2^2)\theta_2 + (\cos q_2\dot{q}_1^2 + \cos q_2\dot{q}_1\dot{q}_2)\theta_3 . \tag{4.2}$$

The potential energy of the Pendubot is written

$$V = (m_1 l_{c1} + m_2 l_1) g \sin q_1 + m_2 l_{c2} g \sin(q_1 + q_2) . \tag{4.3}$$

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

$$V = \theta_4 g \sin q_1 + \theta_5 g \sin(q_1 + q_2) . \tag{4.4}$$

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

$$K = \sum_{i=1}^{5} \frac{\partial K}{\partial \theta_i}\theta_i = \sum_{i=1}^{5} DK_i\theta_i$$
$$V = \sum_{i=1}^{5} \frac{\partial V}{\partial \theta_i}\theta_i = \sum_{i=1}^{5} DV_i\theta_i. \tag{4.5}$$

For the Pendubot the new terms DK and DV are

$$\begin{aligned} DK_1 &= \tfrac{1}{2}\dot{q}_1^2 \\ DK_2 &= \tfrac{1}{2}\dot{q}_1^2 + \dot{q}_1\dot{q}_2 + \tfrac{1}{2}\dot{q}_2^2 \\ DK_3 &= \cos q_2 \dot{q}_1^2 + \cos q_2 \dot{q}_1\dot{q}_2, \\ DK_4 &= 0 \\ DK_5 &= 0 \end{aligned} \tag{4.6}$$

$$\begin{aligned} DV_1 &= 0 \\ DV_2 &= 0 \\ DV_3 &= 0 \\ DV_4 &= g\sin q_1 \\ DV_5 &= g\sin(q_1 + q_2) \end{aligned} \tag{4.7}$$

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

$$\int_{t_1}^{t_2} \mathbf{T}^T \dot{\mathbf{q}}\,dt = (K(t_2) + V(t_2)) - (K(t_1) + V(t_1)) = L(t_2) - L(t_1) \tag{4.8}$$

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

$$\mathbf{T} = \tau + \Gamma_f. \tag{4.9}$$

For this study we neglected friction setting Γ_f to zero. See Appendix A for the addition of friction terms.

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

$$\mathbf{DL}^T = [DL_1(t_2)-DL_1(t_1) \;\ldots\; DL_5(t_2)-DL_5(t_1)] \tag{4.10}$$

and

$$DL_i(t_k) = DK_i(t_k)+DV_i(t_k). \tag{4.11}$$

This leaves the energy equation in the form

$$\int_{t_1}^{t_2} \mathbf{T}^T \dot{\mathbf{q}} dt = \mathbf{DL}^T \theta . \tag{4.12}$$

Defining a new vector d, $d^T = \mathbf{DL}^T$, the K^{th} equation related to the time interval (t_{K-1},t_K) is written

$$(\int_{t_{K-1}}^{t_K} \mathbf{T}^T \dot{\mathbf{q}} dt)_K = d_K^T \theta . \tag{4.13}$$

The K equations can be combined into a standard over determined matrix equation, Ax=b, and solved by least squares techniques. Also since $DL_i(0) = 0$ for (i=1,...,5), we can write this equation as

$$(\int_{t_0}^{t_K} \mathbf{T}^T \dot{\mathbf{q}} dt)_K = d_K^T \theta \tag{4.14}$$

where the K^{th} equation is now for the time interval (t_0,t_K).

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, dq1, q2 and dq2 are vectors of joint positions and velocities
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;      %tau is the open loop control effort
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)  %non-negative least squares solution to Ax=b.
```

Different open-loop inputs (i.e. sine wave, square wave, single steps) 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 input units were volts (V) applied to the amplifier in order that the parameters identified also contained the amplifier gain. The mean of the parameters found by this method with the step input, v = 2.5, were

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

identified parameters

We did attempt to add the friction components to the identification algorithm. Unfortunately, we were unable to find conclusive results for the friction terms. Please see Appendix A for the addition of the friction to the Pendubot model and this identification method. The friction in the Pendubot system is 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. Fortunately the parameters found ignoring friction, as we will demonstrate, work very well in controlling the system.

4.3 Optimization Method

The second on-line method implemented to identify the unknown parameters of the Pendubot uses a constrained minimization algorithm. The error between actual data collected from the Pendubot and data created by simulations is minimized by varying the unknown parameters until a best fit is found. The minimization problem can be written as follows

$$\begin{aligned} &Min_\theta\left\{\sum_{i=1}^{4}\int_0^t (y_{ri}-y_{si})^2\right\} \\ &\text{s.t.} \\ &\theta_{lb} < \theta < \theta_{ub} \end{aligned} \tag{4.15}$$

where y_{ri} is the output position and velocity data of the actual Pendubot and y_{si} is the output position and velocity data found by simulation runs. θ is the vector of inertial parameters to be found and θ_{lb} and θ_{ub} are the lower and upper bounds on these parameters.

The Matlab function "constr" [8] is used to perform this minimization algorithm. "Constr" uses a sequential quadratic programming [9] method to solve the constrained minimization problem.

To implement this identification scheme in Matlab, two function m-files were needed. An objective function to be called and minimized by "constr" and a simulation function modeling the dynamics of the Pendubot. Please refer to Appendix C for the listing of these m-files and the other initialization files used. The simulation function, "pend.m" in Appendix C, defines the nonlinear O.D.E.s of the Pendubot just as seen in equation (3.9) or equation (A.3) if friction is added. To drive the simulated linkage, the same open loop torque equation applied to the Pendubot when collecting the real time data is used to calculate the torque applied to the simulated system. With this function Matlab can simulate the dynamics of the Pendubot with its O.D.E. solver "ode45".

The objective function, "pend_obj.m" in Appendix C, takes as input the unknown parameters and outputs the integral of the error squared, equation (4.15). To accomplish this the objective function first calls "ode45" with the simulation function and the given inertial parameters. This in turn returns simulated response data. The simulated response data and the actual response data are then compared at each time interval of the actual response data, 0.005 seconds. "Ode45" does not perform equally spaced time steps when evaluating the O.D.E. so the simulated response data is interpolated to match up with the time intervals of the actual response data. The Matlab function "interp1" with the spline option is used for this purpose. Finally, using the "trapz" function to evaluate the integrals, the objective function is calculated and returned to "constr".

As with the energy equation method, section 4.2, depending on the open-loop torque trajectory applied, the minimization method was able to do a good job of identify the parameters of the Pendubot. In this case sinusoidal inputs were found to give the best results. For example an open loop input, $v = 0.5\sin 7.7t$ (volts), excited the system enough to allow for an adequate identification. Note, as in the energy equation method, the units of the input signal are taken to be volts (V) applied to the amplifier so the identified parameters will contain the amplifier gain. The parameters found with the above input were

$$\theta_1 = 0.09242 \text{ V*s}^2$$
$$\theta_2 = 0.0247 \text{ V*s}^2$$
$$\theta_3 = 0.0214 \text{ V*s}^2$$
$$\theta_4 = 0.01184 \text{ V*s}^2\text{/in} \quad (0.46614 \text{ V*s}^2\text{/m})$$
$$\theta_5 = 0.00254 \text{ V*s}^2\text{/in} \quad (0.1000 \text{ V*s}^2\text{/m}).$$

As with the energy equation method, this method was unable to identify the friction parameters of the Pendubot. Matlab had a problem solving the minimization problem when the friction terms were added. Stack faults occurred in the middle of each attempted minimization run which indicated some type of numerical problem. Possibly since the friction is low in the linkages, the "constr" function had problems finding adequate friction parameters. It is noted though that the friction is obviously present in the actual Pendubot and will add to the five parameters found.

4.4 Comparison of the Results

Table 4.1 shows a list of each set of parameters identified by the different methods. Similar results were produced by each method, which indicated success in identifying the parameters of the Pendubot. There are some variations in the parameters between the

	Solid Model	**Energy Equation**	**Minimization**
θ_1 (V*s^2)	0.08925	0.0799	0.09242
θ_2 (V*s^2)	0.02763	0.0244	0.0247
θ_3 (V*s^2)	0.0235	0.0205	0.0214
θ_4 (V*s^2/in)	0.0112	0.0107	0.01184
θ_5 (V*s^2/in)	0.00294	0.0027	0.00254

Table 4.1 Comprehensive List of Identified Parameters by Method.

methods, but nothing to conclude that one of the methods were in error. Since friction is ignored in both the energy equation method and the minimization method, its effect on the system are added to the parameters identified. Ignoring the friction possibly has different effects on the identification schemes, therefore creating the differences seen in the parameters. Friction does not enter into the solid model method, therefore also creating possible causes for the differences seen. Of course there are also numerical and modeling errors that are different between the methods. The solid model method is relying on the assumptions of the drafter for accuracy in the model. Both the energy equation method and the minimization method use the data collected from the Pendubot which is inherently noisy, especially the velocity data, which is derived by a finite difference equation (See Chapter 7). Since only these small variations were found, we were able to conclude that the actual parameters of the Pendubot were in close proximity to the parameters listed in Table 4.1.

Due to the small variation of the results found, the parameters identified by the energy equation method were chosen as the parameters to be used for the control experiments. Initially attempts were made to determine if one set of parameters out performed the others. Only small performance variations, if any, were seen when comparing the different parameter sets. For this reason only one set was chosen for consistency.