Identification of Simulation Parameters using Hamiltonian

Task Objectives:
We want to develop a model to find system parameters ($\theta_1 - \theta_6$) experimentally. Using this model we can coax out the parameter values for any system. This increases portability of the final controller we create. This won't be entirely experimental, as we need the strong starting point of knowing what the transfer function looks like.

Background:

Hamiltonian:
In Hamiltonian Mechanics, the Hamiltonian ($\mathcal{H}$) corresponds to the total energy of a system. For a closed system this means the sum of kinetic ($\mathcal{K}$) and potential($\mathcal{U}$) energies.

$$\mathcal{H} = \mathcal{K} + \mathcal{U} \tag{Hamiltonian}$$

Think of the Hamiltonian as a snapshot of the total energy in the system at any given instant.

$$\mathcal{H}(q, \dot{q}) = \sum_{i=1}^{4} h_i(q, \dot{q})\theta_i' + m_1 g z_{c1} + m_2 g z_{c2} \tag{Hamiltonian}$$

For our purposes, we will start with this conservation of energy principle and use it to generate an enormous quantity of equations from the system experimentally. From these equations we will try to define parameters. We already have the equation for $\mathcal{K}$ and $\mathcal{U}$ from the previous Lagrangian Dynamics section. First we can plug those equations for $\mathcal{K}$ and $\mathcal{U}$ into the Hamiltonian equation $\mathcal{H} = \mathcal{K} + \mathcal{U}$. Then we can isolate terms by θ_i and the resultant product in each term yields h_i by symmetry the Hamiltonian summation equation.

$$h_1 = \frac{1}{2}\dot{q_1^2} \tag{derived h_1}$$

$$h_2 = \frac{1}{2}(\dot{q_2}^2 + \dot{q_1}^2 \sin^2(q_2)) \tag{derived h_2}$$

$$h_3 = \cos(q_2)\dot{q_1}\dot{q_2} \tag{derived h_3}$$

$$h_4 = g\cos(q_2) \tag{derived h_4}$$

XXXXXXXXX
With the knowledge that

$$torque \times speed = power \tag{1}$$

and that

$$\int power = energy \tag{2}$$

we can derive another equation for energy which we can then compare to the Hamiltonian. The energy currently in the system can be found by taking the total energy in the system and subtracting off the initial energy in the system.

$$\int_{t_0}^{t} [\tau(s)\dot{q_1}(s) - \beta_1 \dot{q_1}^2 ds] = \mathcal{H}(q(t), \dot{q}(t)) - \mathcal{H}(q(t_0), \dot{q}(t_0)) \tag{3}$$

Because the result is a difference of the two summations, the identical mgz terms will cancel out and the equation only depends on h_i.

$$\int_{t_0}^{t} [\tau(s)\dot{q_1}(s) - \beta_1 \dot{q_1}^2 ds] = \sum_{i=1}^{4} [h_i(q(t), \dot{q}(t)) - h_i(q(t_0), \dot{q}(t_0))]\theta_i \tag{4}$$

To get the equations into a form with θ instead of θ prime...

$$\int_{t_0}^{t} V(s)\dot{q_1}(s) ds = \bar{\mathcal{H}}(t, t_0) \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} + \bar{F}(t, t_0) \begin{bmatrix} \theta_5 \\ \theta_6 \end{bmatrix} \tag{5}$$

$\bar{\mathcal{H}}(t, t_0)$ is a [1x4] matrix where each of the 4 elements is defined as follows:

$$\bar{\mathcal{H}}(t, t_0)_i \triangleq h_i(q(t), \dot{q}(t)) - h_i(q(t_0) \tag{6}$$

$\bar{F}(t, t_0)$ is a [1x2] matrix defined as follows:

$$\bar{F}(t, t_0) \triangleq \left[\int_{t_0}^{t} \dot{q_1}^2(s)ds, \int_{t_0}^{t} \dot{q_2}^2(s)ds\right] \tag{7}$$

If you recall we were planning to generate an "enormous" quantity of equations. Say we generate "n" equations, one equation for each of "n" instants in time that describes the energy from time t_0 up until t_n.

Define D as a [nx1] matrix as follows:

$$d = \begin{bmatrix} \int_{t_0}^{t_1} V(s)\dot{q}_1(s)ds \\ \int_{t_0}^{t_2} V(s)\dot{q}_1(s)ds \\ ... \\ \int_{t_0}^{t_n} V(s)\dot{q}_1(s)ds \end{bmatrix} \tag{8}$$

Define A as an [nx6] matrix where the 6 columns come from the [1x4] and [1x2] sized $\mathcal{H}$ and F matrices respectively.

$$A = \begin{bmatrix} \bar{\mathcal{H}}(t_1, t_0), \bar{F}(t_1, t_0) \\ \bar{\mathcal{H}}(t_2, t_0), \bar{F}(t_2, t_0) \\ ... \\ \bar{\mathcal{H}}(t_n, t_0), \bar{F}(t_n, t_0) \end{bmatrix} \tag{9}$$

Knowing that...

$$d = A \begin{bmatrix} \theta_1 \\ \theta_2 \\ ... \\ \theta_6 \end{bmatrix} \tag{10}$$

we can backsolve to find θs. Using MATLAB's "pseudo-inverse" fxn...

$$\bar{\theta} = [(A^T A)^{-1} A^T]d \tag{11}$$

Generating Data for Measured Voltage and States

The process of experimentally determining more accurate and individualized parameters $\theta_1 - \theta_6$ requires a set of dynamic measured data. The better the data (more physical conditions tested) the more accurate the parameters. In other words, a lot of variation of motion is desired in the measured data. By using the simple controllers created in previous labs a combination of sinusoids was input to the pendubot while the voltage and encoder readings for position were noted in a stored array. Please note that the picture of the simulink schematic below was taken on a system not setup with Quarc and is throwing a few errors. It will soon be replaced with a snapshot from the system on which it was run.

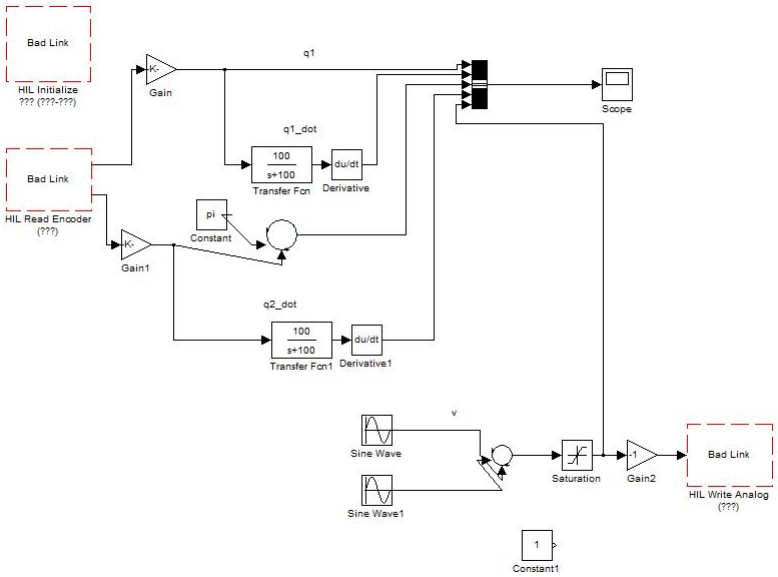

Figure 1: Simulink Model to Generate Data for Experimentally Determining Parameters Values.

Implementation (Finding Experimental Parameters)
Using the array of measured voltage and position a set of theoretical values for $\theta_1 - \theta_6$ were found using the following script.

```
%%%%%%%%%%%%%   Experimentally determine theta values %%%%%%%%%%%%%%%%%%%%
clear all; close all; clc;
load all_data;
x1 = all_data;

x = x1(:,2:5)';
v = x1(:,6)';
dt = x1(:,1)';

q1 = x(1,:);
q1_dot = x(2,:);
q2 = x(3,:);
q2_dot = x(4,:);

[rows,cols] = size(x);
num_pts = cols;
grav = 9.81;

h1 = 0.5*q1_dot.^2;
h2 = 0.5.*(q2_dot.^2 + (q1_dot.*sin(q2)).^2);
h3 = cos(q2).*q1_dot.*q2_dot;
h4 = grav.*cos(q2);

h1_to = h1(1);
h2_to = h2(1);
h3_to = h3(1);
h4_to = h4(1);

H = zeros(num_pts,4);
F = zeros(num_pts,2);
d = zeros(num_pts,1);
A = zeros(num_pts,6);

H(1,1:4) = [h1_to h2_to h3_to h4_to];
F(1,1:2) = [0,0];
d(1,1) = [0];
for i=2:num_pts
    H(i,1:4) = [h1(i)-h1_to,h2(i)-h2_to,h3(i)-h3_to,h4(i)-h4_to];
    F(i,1:2) = [trapz(dt(1:i), q1_dot(1:i).^2),trapz(dt(1:i), q2_dot(1:i).^2)];
    d(i,1) = trapz(dt(1:i), v(1:i).*q1_dot(1:i));
end

A(1:num_pts,1:4) = H;
A(1:num_pts,5:6) = F;

theta_vec = pinv(A)*d;
```

Testing Experimental Parameters:
Once the experimental parameters $\theta_1 - \theta_6$ were obtained, they were substituted in for the theoretical parameters used in previous simulations. The resultant animations from the simulations using experimental parameters were closer to the behavior of the real pendubot.