Washington University in St. Louis

ESE447 Robotics Laboratory

Hamiltonian Report

Chih Yun Pai
ID: SI452667

Instructor
Professor Dennis Mell

November 3, 2017

Objective:

There is difference between real-world situation and mathematical calculation when proceeding simulation. In order to make sure the parameters from mathematical derivation which is implemented in the simulation environment are acceptably close to the real-world, creating another reverse calculation process to examine the difference between theory and practices becomes necessary. In this experiment, we use the Lagrangian equation to build simulation model and create another Hamiltonian version to reverse the parameters though sampling real-world data and compare their difference. In short, to find more realistic for System Parameters $\Theta = [\theta_1; \theta_2; ...; \theta_6]$. θ's are defined in the following.

Procedure:

1. Build a SimuLink model simulate two-joint robot for collecting X, V and time data from simulation. System parameters and equations below are implemented in StateMachine function.

System parameters:

$J_1 = 0.0012, m_2 = 0.127, (l_1 + l_2\prime) = 0.2, l_2 = 0.3, l_{c2} = 0.15, \beta_1 = 0.015, \beta_2 = 0.002, R_a = 2.6, k_t = 0.00768, k_v = 0.00768, k_r = 70$

$$\theta_1 = \frac{J_1 + m_2(l_1 + l_2\prime)^2 R_a}{(k_r k_t)}$$

$$\theta_2 = \frac{(m_2 l_2^2) R_a}{3(k_r k_t)}$$

$$\theta_3 = \frac{(m_2(l_1 + l_2\prime) l_2) R_a}{2(k_r k_t)}$$

$$\theta_4 = \frac{m_2 l_{c2} R_a}{(k_r k_t)}$$

$$\theta_5 = \frac{\beta_1 R_a}{(k_r k_t)} + (k_r k_v)$$

$$\theta_6 = \frac{\beta_2 R_a}{(k_r k_t)}$$

$M = \begin{bmatrix} m_{11}, m_{12} \\ m_{21}, m_{22} \end{bmatrix}$, where $m_{11}(q) = \theta_1 + \theta_2 \sin(q_2)^2, m_{12}(q) = \theta_3 \cos(q_2), m_{21}(q) = \theta_3 \cos(q_2), m_{22}(q) = \theta_2$

$C = \begin{bmatrix} c_{11}, c_{12} \\ c_{21}, c_{22} \end{bmatrix}$, where $c_{11} = 2\theta_2 \sin(q_2)\cos(q_2)\dot{q_2}, c_{12} = -\theta_3 \sin(q_2)\dot{q_2}, c_{21} = -\theta_2 \sin(q_2)\cos(q_2)\dot{q_1}$

$F = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$, where $f_1 = \theta_5 \dot{q_1}, f_2 = \theta_6 \dot{q_2}$

$G = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$, where $g_1 = 0, g_2 = -\theta_6 \dot{q_2}$

$\dot{q} = \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \end{bmatrix}$

$\ddot{q} = M^{-1}(V - C\dot{q} - F - G)$

$\dot{X} = \begin{bmatrix} \dot{q_1} \\ \ddot{q_1} \\ \dot{q_2} \\ \ddot{q_2} \end{bmatrix}$

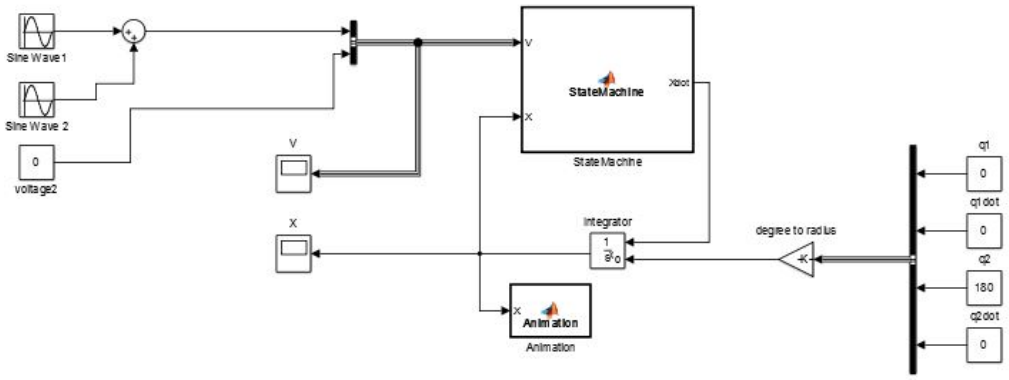


Figure 1: Simulation_Collect: Simulink model for simulation data collecting.

2. Build a SimuLink model for collecting variables X given the same initial Parameters and voltages V as in simulation.

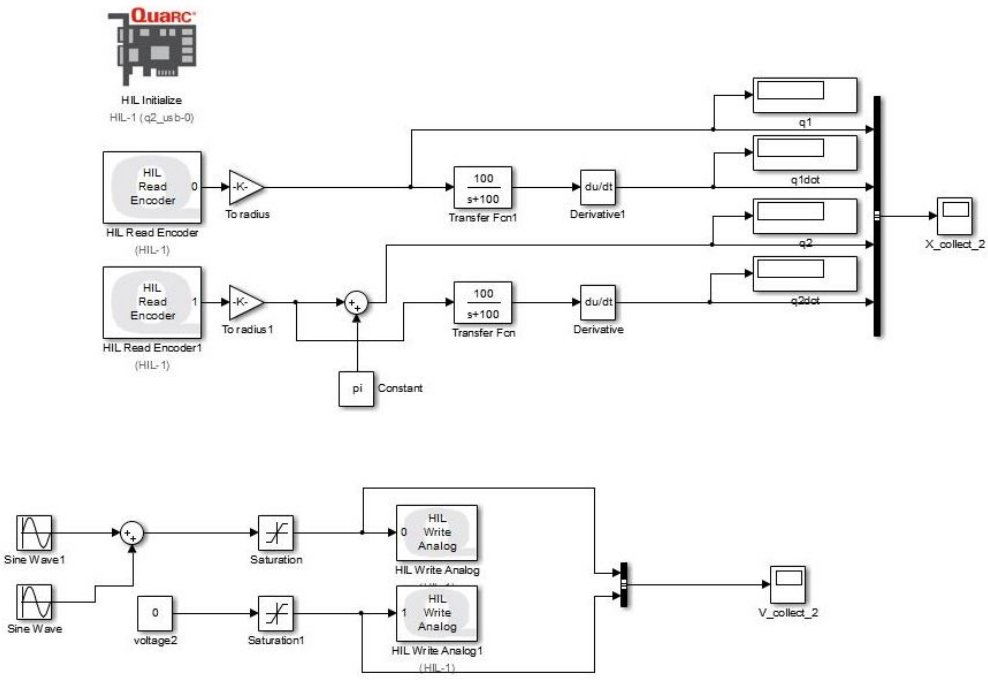


Figure 2: Real_Collect: Simulink model for real-world data collecting.

3.Analyze data to find best fit for System Parameters.

- Hamiltonian derivation:

$\Theta = [\theta_1; \theta_2; ...; \theta_6], X = [q_1; \dot{q_1}; q_2; \dot{q_2}], V = [v_1; v_2 = 0]$

n: number of sampling point, greater than 10,000.

Hamiltonian (Total Mechanical energy (instantaneous)) $\Rightarrow H = K + V$

$h_1(q, \dot{q}) = \frac{1}{2}\dot{q_1}^2$

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

$h_3(q, \dot{q}) = \cos(q_2)\dot{q_1}\dot{q_2}$

$h_4(q, \dot{q}) = g\cos(q_2)$

$\int_{t_0}^{t}[\tau(s)\dot{q_1}(s)ds - \beta_1\dot{q_1}^2(s) - \beta_2\dot{q_2}^2(s)]ds \Rightarrow \tau(s)\dot{q_1}(s)$: much power for motor s

$\int_{t_0}^{t}[V(s)\dot{q}(s)]ds - \int_{t_0}^{t}[\dot{q_1}^2(s)]ds - \int_{t_0}^{t}[\dot{q_2}^2(s)]ds \Rightarrow \int_{t_0}^{t}[V(s)\dot{q}(s)]ds : electrical$

$\bar{H}(t, t_0) = [h_1(q(t), \dot{q}(t)) - h_1(q(t_0), \dot{q}(t_0)), ..., h_4(q(t), \dot{q}(t)) - h_4(q(t_0), \dot{q}(t_0))]$

$\bar{F}(t, t_0) = [\int_{t_0}^{t} \dot{q_1}^2(s)ds, \int_{t_0}^{t} \dot{q_2}^2(s)ds]$

Total energy in system at t $\Rightarrow \bar{H}(t, t_0)[\theta_1; \theta_2; \theta_3; \theta_4] + \bar{F}(t, t_0)[\theta_5; \theta_6] = \int_{t_0}^{t} V(s)\dot{q_1}(s)ds$

$d = [\text{Total Energy at t}]_{n\times 1}$

$$A = \begin{bmatrix} \bar{H}(t_1, t_0), \bar{F}(t_1, t_0) \\ \bar{H}(t_2, t_0), \bar{F}(t_2, t_0) \\ \vdots \\ \bar{H}(t_n, t_0), \bar{F}(t_n, t_0) \end{bmatrix}_{n\times 6}, \quad d = A \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_6 \end{bmatrix} \Rightarrow d = A\Theta \Rightarrow \Theta = A^{-1}d$$

- Create *calc_theta.m* function, which implements Hamiltonian equation derived above for calculating Θ.

Process documentation:

- Code:

```
function Xdot  =StateMachine(V, X)
% J1=0.0012;m2=0.127;l1_l2=0.2;l2=0.3;lc2=0.15;
% beta1=0.015;beta2=0.002;Ra=2.6;kt=0.00768;kv=0.00768;kr=70;
%
% theta1=(J1+m2*l1_l2^2)*Ra/(kr*kt);
% theta2=(m2*l2^2/3)*Ra/(kr*kt);
% theta3=(m2*l1_l2*l2/2)*Ra/(kr*kt);
% theta4=(m2*lc2)*Ra/(kr*kt);
% theta5=beta1*Ra/(kr*kt) + kr*kv;
% theta6=beta2*Ra/(kr*kt);
q1=X(1);
q1dot=X(2);
q2=X(3);
q2dot=X(4);

theta1=0.030372023809524;
theta2=0.018426339285714;
theta3=0.018426339285714;
theta4=0.092131696428571;
theta5=0.610144642857143;
theta6=0.009672619047619;

m11=theta1+theta2*sin(q2)^2;
m12=theta3*cos(q2);
m21=theta3*cos(q2);
m22=theta2;
c11=2*theta2*sin(q2)*cos(q2)*q2dot;
c12=-theta3*sin(q2)*q2dot;
c21=-theta2*sin(q2)*cos(q2)*q1dot;
c22=0;

f1=theta5*q1dot;
f2=theta6*q2dot;
g=9.8;
g1=0;
g2=-theta4*g*sin(q2);
M=[m11,m12;m21,m22];
C=[c11,c12;c21,c22];
F=[f1;f2];
G=[g1;g2];
qdot=[q1dot;q2dot];
qdotdot=M\(V-C*qdot-F-G);
Xdot=[q1dot,qdotdot(1),q2dot,qdotdot(2)];
end
```

Figure 3: StateMachine.m

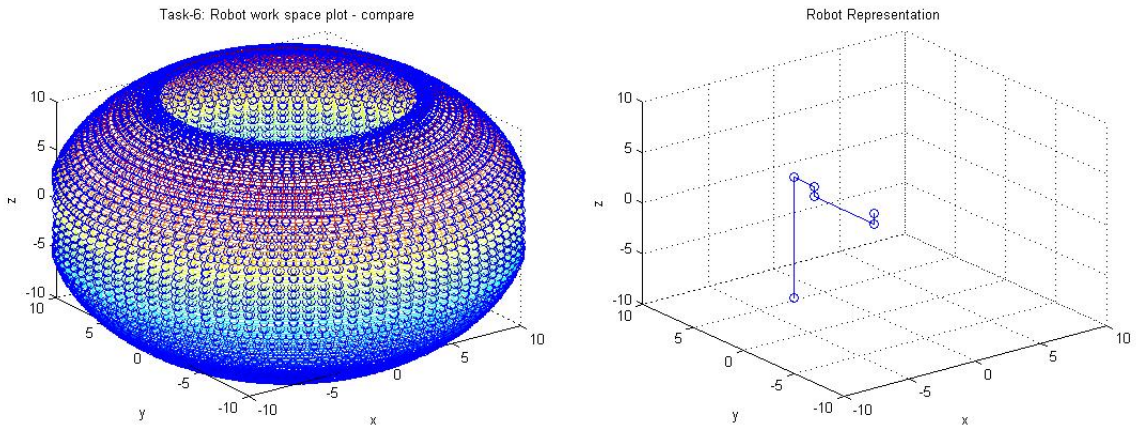
```
function Theta = calc_theta(X_collect, V_collect);
time = X_collect.time;
n = min(size(time,1), 10800);
X=zeros(n,4);
for i=1:n
    X(i,:)=X_collect.signals.values(:,:,i);
end
V = V_collect.signals.values;
g = 9.81;
h1 = @(q1dot)0.5*q1dot^2;
h2 = @(q1dot, q2, q2dot) 0.5*((q1dot^2)*(sin(q2)^2) + (q2dot^2));
h3 = @(q1dot, q2, q2dot) cos(q2)*q1dot*q2dot;
h4 = @(q2) g*cos(q2);
H_bar = zeros(n-1, 4);
F_bar = zeros(n-1, 2);
q1 = X(1,1);
q1dot = X(1,2);
q2 = X(1,3);
q2dot = X(1,4);
h1_0 = 0.5*q1dot^2;
h2_0 = 0.5*q1dot^2*(sin(q2)^2) + 0.5*(q2dot^2);
h3_0 = cos(q2)*q1dot*q2dot;
h4_0 = g*cos(q2);
d = zeros(n-1,1);
for i = 1:(n-1)
%     q1 = X(i+1,1);
    q1dot = X(i+1,2);
    q2 = X(i+1,3);
    q2dot = X(i+1,4);
    H_bar(i,:) = [h1(q1dot)-h1_0,...
                  h2(q1dot, q2, q2dot)-h2_0,...
                  h3(q1dot, q2, q2dot)-h3_0,...
                  h4(q2)-h4_0];
    F_bar(i,:) = trapz(time(1:i+1), [X(1:i+1, 2).^2, X(1:i+1, 4).^2]);
    d(i,1) = trapz(time(1:i+1), V(1:i+1,1).*X(1:i+1, 2));
end
A = [H_bar, F_bar]; % A: size of (n-1)x6
Theta = A\d;
end
```

Figure 4: *calc_theta.m*

- Data:

$X\_simulation$: 10,800 by 4 matrix, $V\_simulation$: 10,800 by 2 matrix
X_STA7: 10,800 by 4 matrix, $V\_STA7 = V\_simulation$
X_STA8: 10,800 by 4 matrix, $V\_STA8 = V\_simulation$
X_STA9: 10,800 by 4 matrix, $V\_STA9 = V\_simulation$

- Graph:



(a) Workspace.m (b) Starting position: $\theta_1 = 0, \theta_2 = \pi$

Figure 5: Pictures of working space.

Results:

	Simulation	STA7	**STA8**	STA9	AVG(7~9)
Theta1	0.030372	0.078661	0.081489	0.077162	0.079104
Theta2	0.018426	0.030567	0.031237	0.025134	0.028979
Theta3	0.018426	0.025982	0.02843	0.023294	0.025902
Theta4	0.092038	0.148116	0.150364	0.127642	0.14204
Theta5	0.610145	0.544905	0.553647	0.549331	0.549294
Theta6	0.009672	0.011675	0.008797	0.022395	0.014289

Figure 6: Thetas are calculated by *cal_theta.m* using sampling data created from simulation and 3 different robot. AVG(7 - 9) column is the avergae from STA7 to STA9.

Observation and Summary:

The average of STA7-STA9 Thetas is slightly different from the simulation Thetas pre-calculated at the beginning of simulation StateMachine. The max absolute difference between average value and the theoretical value is about 0.06, which is reasonably close, while it still can help the system Thetas to be more fitting into the real-world parameters.

Furthermore, there is some more interference from the real-world, such as the force generated from the motor cable, the experimental error between theoretical and practical of gear friction and the infulence of discreteness on integration in sampling against continuity. Although the difference between two is expectable, the adequate amount of sampling data points generates feasible parameters in engineering perspective.