Robot Simulation using Lagrangian Dynamics

Task Objectives:
We already created code to animate the pendubot when provided pre-constructed or even measured values for the D.O.F. The final implementation fo these functions took the form of figure 1.

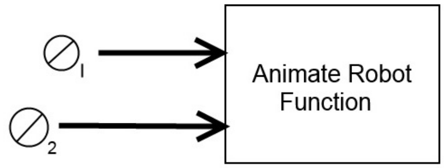

Figure 1: Animation Function Implementation from Previous Work.

However these functions serve only to animate or playback the movements of the robot, not calculate or simulate how it will move based off voltage input. To handle the task of true simulation, we turn to Lagrangian Dynamics. The objective here is to create a simulation environment that, when given an input voltage and a current position of the robot, can accurately predict/calculate the next time-step's position of the robot while considering the reality of physical forces acting on the limbs of the robot. Figure 2 demonstrates the general outline of this simulation. These calculated positional values can be inputs to our previous animation functions to yield a surprisingly effective simulation. This function combination is outlined in figure 2.

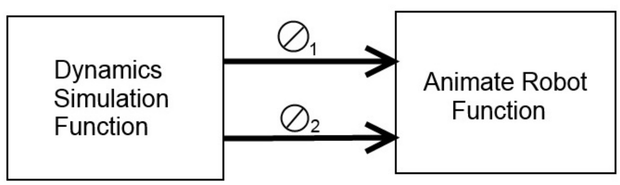

Figure 2: Simplistic overview of the desired model.

Background (Lagrangian Dynamics):
Opposing the use classical mechanics, Lagrangian mechanics can be used to solve complex problems regarding the trajectory of systems of particles. Lagrangian mechanics were touted by Lagrange himself as being simple Analytical Mechanics requiring little geometrical or mechanical reasoning, instead being derived only with algebraic means. This differentiates Lagrangian mechanics from the classical Newtonian mechanics which are based on vectors and on the concept of forces. Lagrangian forces do not present new laws of physics, but instead express Newton's laws of motion in an alternate form that permits easy algebraic manipulation when working with complex particle ensembles.

Eueler-Lagrange Method (energy based approach):
When approaching dynamics modeling for robots, a Newton-Euler method revolves around balancing forces and torques, whereas a Lagrange-Eueler method uses an energy-based approach. In robotics, each joint is generalized as a "q" (either an angle θ or some other transformable variable). The Lagrangian (as follows) can be thought of as a Kinetic Potential.

$$L(q, \dot{q}) = K(q, \dot{q}) - V(q, \dot{q}) \tag{Lagrangian}$$

From the Lagrangian we can generate the Euler-Lagrange Equation.

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} = \tau_i \tag{Euler Lagrange Eqn}$$

The Euler Lagrange Equation summarizes the involvement of forces generated because the limb $\dot{q}_i$ is moving, and forces generated because the limb $\dot{q}_i$ is somewhere in space. $\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i})$ brings into account the limb's velocity ($\dot{q}_i$) and therefore kinetic energy. $\frac{\partial L}{\partial q_i}$ brings into account potential energy by involving the limb's position (q_i). These latter involvements are considered the result of robot's "pose", whereas the prior were results of the robot's movement.

To simulate the dynamics for an entire robot, the Euler Lagrangian Equation can be applied to each link of the robot. The beauty of Lagrangian Dynamics is that they will intrinsically account for unintuitive concepts that would take a very astute classical mechanist to incorporate. For example, a joint that isn't changing place or speed can still require changes in torque (τ) because of alterations in the pose of the robot (in other joints or limbs). Think of supporting a heavy weight held in your hand by the shoulder vs at arm length. Despite the only difference in position being movement from the elbow joint, the application of torque to the shoulder joint can vary dramatically based off the pose.

Generalized Coordinates:
Generalized coordinates are a minimal set of variables q_1, q_2... that describe the position of the system completely. They are (x,y,z) for a linear system, (r,θ,z) for a system in polar coordinates or (θ, ϕ, r) for a system in spherical coordinates. **Generalized coordinates along with their rate of change gives the state of the system at any point.** For the single link rotary inverted pendulum, the generalized coordinates are the 2 joint angles q_1 and q_2.

Simulation with Simple Matrix Algebra:

To properly simulate and control the robot, we only have one selectable input: torque (τ) which we control by modulating motor input voltage (V). To determine what voltage or torque needs to be input, we first need to find angular acceleration (α). Recall that:

$$\alpha = \frac{\tau}{I} \tag{Rotational Motion from Newton's Second Law}$$

Assume u(V) to be the input voltage, and X to the state of the system defined by the position of the two limbs, along with their first derivatives (velocity).

$$u(V) \tag{Input f(voltage)}$$

$$X = \begin{bmatrix} q_1 \\ \dot{q_1} \\ q_2 \\ \dot{q_2} \end{bmatrix} \tag{Input States Vector}$$

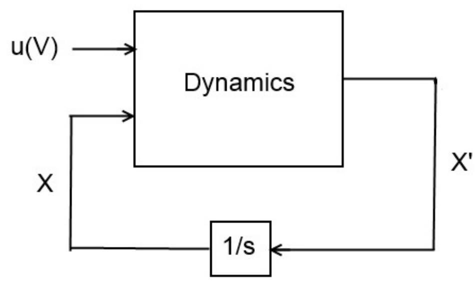

Figure 3: Basic Model Overview.

Using a similar model as shown in figure 3, where integration as a means of finding states leaves X' containing the second derivative of position (ie: angular acceleration α). With values for $q, \dot{q}, and \ddot{q}$ at each moment in time, and the derived equations for torque or voltage that came from Lagrangian Dynamics, it is possible to use basic matrix algebra to implement a simulation for the two limb robot.

$$\begin{bmatrix} m(q) \end{bmatrix} \begin{bmatrix} \ddot{q_1} \\ \ddot{q_2} \end{bmatrix} + \begin{bmatrix} c(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \end{bmatrix} + \begin{bmatrix} f(\dot{q_1}) \end{bmatrix} + \begin{bmatrix} g(q) \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix} \tag{1}$$

where... m(q) is 2x2 matrix describing the forces due to mass of each limb. The values depend only on the current pose of the robot, not its motion. For example, $m_{1,1}$ is the mass of limb 1 with respect to frame 1.

$$m(q) = \begin{bmatrix} m_{1,1}, m_{1,2} \\ m_{2,1}, m_{2,2} \end{bmatrix} \tag{Mass Matrix}$$

$c(q, \dot{q})$ is a 2x2 matrix describing the centrifugal and Coriolis forces.

$f(\dot{q_1})$ describes the frictional forces.

$g(q)$ describes the gravitational forces. This is not the force of gravity, it is similar to the resultant torque on a mass due to the force of gravity which depends on the objects position.

Derivation of Matrices (Summary):

$$[\theta_1 + \theta_2 \sin(q_2)^2]\ddot{q_1} + \theta_3 \cos(q_2)\ddot{q_2} + 2\theta_2 \sin(q_2)\cos(q_2)\dot{q_1}\dot{q_2} - \theta_3 \sin(q_2)\dot{q_2}^2 + \theta_5\dot{q_1} = v \tag{2}$$

$$\theta_3 \cos(q_2)\ddot{q_1} + \theta_2\ddot{q_2} - \theta_2 \sin(q_2)\cos(q_2)\dot{q_1}^2 - \theta_4 g \sin(q_2) + \theta_6\dot{q_2} = 0 \tag{3}$$

$$M(q) = \begin{bmatrix} \theta_1 + \theta_2 \sin(q_2)^2 & \theta_3 \cos(q_2) \\ \theta_3 \cos(q_2) & \theta_2 \end{bmatrix} \tag{4}$$

$$C(q, \dot{q}) = \begin{bmatrix} 2\theta_2 \sin(q_2)\cos(q_2)\dot{q_1}\dot{q_2} & -\theta_3 \sin(q_2)\dot{q_2} \\ -\theta_2 \sin(q_2)\cos(q_2)\dot{q_1} & \theta_6 \end{bmatrix} \tag{5}$$

$$f(\dot{q_1}) = \begin{bmatrix} \theta_5\dot{q_1} \\ 0 \end{bmatrix} \tag{6}$$

$$g(q) = \begin{bmatrix} 0 \\ \theta_4 g \sin(q_2) \end{bmatrix} \tag{7}$$

$\theta_1 = \theta'_1 \frac{R_a}{k_r k_t}$
$\theta_2 = \theta'_2 \frac{R_a}{k_r k_t}$
$\theta_3 = \theta'_3 \frac{R_a}{k_r k_t}$
$\theta_4 = \theta'_4 \frac{R_a}{k_r k_t}$
$\theta_5 = \beta_1 \frac{R_a}{k_r k_t} + k_r k_v$
$\theta_6 = \beta_2 \frac{R_a}{k_r k_v}$
$\theta'_1 = J_1 + m_2(l_1 + \prime(l_2))^2$
$\theta'_2 = \frac{1}{3} m_2 l_2^2$
$\theta'_3 = \frac{1}{2} m_2 (l_1 + l'_2) l_2$
$\theta'_4 = m_2 l_(c_2)$

$$\begin{bmatrix} m(q) \end{bmatrix} \begin{bmatrix} \ddot{q_1} \\ \ddot{q_2} \end{bmatrix} + \begin{bmatrix} c(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \end{bmatrix} + \begin{bmatrix} f(\dot{q_1}) \end{bmatrix} + \begin{bmatrix} g(q) \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix} \tag{8}$$

Derivation of Matrices (Long Math):

$$K_1 = \frac{1}{2} J \dot{q}_1^2$$

1

$$V_1 = m_1 g z_{c_1}$$

$$p(x_2) = \begin{bmatrix} (l_1 + l_2')\cos(q_1) - x_2 \sin(q_2)\sin(q_1) \\ (l_1 + l_2')\sin(q_1) + x_2 \sin(q_2)\cos(q_1) \\ x_2 \cos(q_2) \end{bmatrix}$$

$q_1 = f(t)$

$l_1, l_2', x_2 = \text{constant}$

$(x+y+z)^2$
$= x^2 + y^2 + z^2 + 2xy + 2yz + 2xz$

$$v(x_2) = \dot{p}(x_2) = \begin{bmatrix} -(l_1 + l_2')\sin(q_1)\dot{q}_1 - x_2\left[\sin(q_2)\cos(q_1)\dot{q}_1 + \cos(q_2)\dot{q}_2\sin(q_1)\right] \\ (l_1 + l_2')\cos(q_1)\dot{q}_1 + x_2\left[-\sin(q_2)\sin(q_1)\dot{q}_1 + \cos(q_2)\dot{q}_2\cos(q_1)\right] \\ -x_2\sin(q_2)\dot{q}_2 \end{bmatrix}$$

$$|v(x_2)|^2 = v_1^2 + v_2^2 + v_3^2$$

$$v_3^2 = x_2^2 \sin^2(q_2)\dot{q}_2^2$$

$$v_1 = -(l_1 + l_2')\sin(q_1)\dot{q}_1 - x_2\sin(q_2)\cos(q_1)\dot{q}_1 - x_2\cos(q_2)\dot{q}_2\sin(q_1)$$

$$\begin{aligned} v_1^2 = {} & (l_1 + l_2')^2 \sin^2(q_1)\dot{q}_1^2 + x_2^2\sin^2(q_2)\cos^2(q_1)\dot{q}_1^2 + x_2^2\cos^2(q_2)\dot{q}_2^2\sin^2(q_1) \\ & + 2(l_1 + l_2')\sin(q_1)\dot{q}_1^2 x_2\sin(q_2)\cos(q_1) + 2(l_1 + l_2')\sin(q_1)\dot{q}_1 x_2\cos(q_2)\dot{q}_2\sin(q_1) \\ & + 2x_2^2\sin(q_2)\cos(q_1)\dot{q}_1\dot{q}_2\cos(q_2)\sin(q_1) \end{aligned}$$

$$\begin{aligned} v_1^2 = {} & \left[(l_1 + l_2')^2 \cancel{\sin^2(q_1)} + x_2^2\sin^2(q_2)\cos^2\cancel{(q_1)} + \cancel{2(l_1 + l_2')\sin(q_1)x_2\sin(q_2)\cos(q_1)}\right]\dot{q}_1^2 \\ & + \left[2(l_1 + l_2')\cancel{\sin^2(q_1)}x_2\cos(q_2) + \cancel{2x_2^2\sin(q_2)\cos(q_1)\cos(q_2)\sin(q_1)}\right]\dot{q}_1\dot{q}_2 \\ & + \left[x_2^2\sin^2\cancel{(q_1)}\cos^2(q_2)\right]\dot{q}_2^2 \end{aligned}$$

(2)

$$v_2 = (l_1+l_2')\cos(q_1)\dot{q}_1 - x_2\sin(q_2)\sin(q_1)\dot{q}_1 + x_2\cos(q_2)\dot{q}_2\cos(q_1)$$

$$v_2^2 = (l_1+l_2')^2\cos^2(q_1)\dot{q}_1^2 + x_2^2\sin^2(q_2)\sin^2(q_1)\dot{q}_1^2 + x_2^2\cos^2(q_2)\dot{q}_2^2\cos^2(q_1)$$
$$- 2(l_1+l_2')\cos(q_1)\dot{q}_1^2 x_2\sin(q_2)\sin(q_1) + 2(l_1+l_2')\cos^2(q_1)\dot{q}_1\cos(q_2)\dot{q}_2 x_2$$
$$- 2x_2^2\sin(q_2)\sin(q_1)\dot{q}_1\cos(q_2)\cos(q_1)\dot{q}_2$$

$$= \left[(l_1+l_2')^2\cancel{\cos^2(q_1)} + x_2^2\sin^2(q_2)\cancel{\sin^2(q_1)} - \cancel{2(l_1+l_2')\cos(q_1)x_2\sin(q_1)\sin(q_2)}\right]\dot{q}_1^2$$
$$+ \left[2(l_1+l_2')x_2\cancel{\cos^2}(q_1)\cos(q_2) - \cancel{2x_2^2\sin(q_2)\sin(q_1)\cos(q_2)\cos(q_1)}\right]\dot{q}_1\dot{q}_2$$
$$+ \left[x_2^2\cos^2(q_2)\cancel{\cos^2(q_1)}\right]\dot{q}_2^2$$

$$v_1^2+v_2^2 = \left[(l_1+l_2')^2 + x_2^2\sin^2(q_2)\right]\dot{q}_1^2 + \left[2(l_1+l_2')x_2\cos(q_2)\right]\dot{q}_1\dot{q}_2 + \left[x_2^2\cos^2(q_2)\right]\dot{q}_2^2$$
$$+ v_3^2 \Rightarrow$$

$$\boxed{\left[(l_1+l_2')^2 + x_2^2\sin^2(q_2)\right]\dot{q}_1^2 + \left[2(l_1+l_2')x_2\cos(q_2)\right]\dot{q}_1\dot{q}_2 + x_2^2\dot{q}_2^2}$$

q_1, q_2 indep of x_2

$$K_2 = \frac{1}{2}\int_0^{l_2}\rho_2 A_2|v(x_2)|^2 dx_2 = \frac{\rho_2 A_2}{2}\left[\int_0^{l_2}(l_1+l_2')^2\dot{q}_1^2\,dx + \int_0^{l_2}\left(x_2^2\sin^2(q_2)\dot{q}_1^2 + 2(l_1+l_2')\cos(q_2)x_2\dot{q}_1\dot{q}_2 + x_2^2\dot{q}_2^2\right)dx\right]$$

$$\boxed{\frac{\overbrace{l_2\rho_2 A_2}^{m_2}}{2}\left[l_2(l_1+l_2')^2\dot{q}_1^2 + \frac{l_2^3}{3}\sin^2(q_2)\dot{q}_1^2 + l_2(l_1+l_2')\cos(q_2)\dot{q}_1\dot{q}_2 + \frac{l_2^3}{3}\dot{q}_2^2\right]}$$

$$K_2 = \boxed{\frac{1}{2}m_2\left[(l_1+l_2')^2\dot{q}_1^2 + (l_1+l_2')l_2\cos(q_2)\dot{q}_1\dot{q}_2 + \frac{1}{3}l_2^2\left(\sin^2(q_2)\dot{q}_1^2 + \dot{q}_2^2\right)\right]}$$

$J_1 = 0.0044$ $\quad\beta_1 = 0.015$ $\quad$ 3

$m_2 = 0.14\,kg$ $\quad\beta_2 = 0.002$

$(l_1 + l_2') = 0.2\,m$ $\quad K_t = K_v = K_n = 0.00767$

$l_2 = 0.3\,m$ $\quad K_r = K_g = 70$

$l_{2c} = 0.15\,m$ $\quad R_e = R_m = 2.6$

$K_1 = \frac{1}{2} J_1 \dot{q}_1^2$

$K_2 =$ see prev. pg $\qquad$ separates the terms $\dot{q}_1^2$, $\dot{q}_1\dot{q}_2$ & $\dot{q}_2^2$

$$K_{tot} = K_1 + K_2 = \frac{1}{2}\left[\overbrace{\boxed{J_1 + \tfrac{1}{2} m_2 (l_1 + l_2')^2}}^{\theta_1'} + \overbrace{\boxed{\tfrac{1}{3} m_2 l_2^2}}^{\theta_2'} \sin^2(q_2)\right] \dot{q}_1^2$$

$$+ \boxed{\tfrac{1}{2}}\left[\overbrace{\boxed{m_2 (l_1 + l_2') l_2}}^{\theta_3'} \cos(q_2)\right] \dot{q}_1 \dot{q}_2$$

$$+ \frac{1}{2}\left[\overbrace{\boxed{\tfrac{1}{3} m_2 l_2^2}}^{\theta_2'}\right] \dot{q}_2^2$$

---

c1 = ctr of mass of 1

$V_1 = m_1 g z_{c1}$ $\qquad V_2 = m g l_{c2} \cos(q_2)$

$$V_{Tot} = m_1 g z_{c1} + \underbrace{m_2 g l_{c2}}_{\theta_4'} \cos(q_2)$$

$$\mathcal{L} = K_{tot} - V_{tot} = \frac{1}{2}\left(\theta_1' + \theta_2' \sin^2(q_2)\right)\dot{q}_1^2 + \left(\theta_3' \cos(q_2)\right)\dot{q}_1\dot{q}_2 + \frac{1}{2}\theta_2' \dot{q}_2^2 - m_1 g z_{c1} - \theta_4' g \cos(q_2)$$

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{q}_1}\right] - \frac{\partial \mathcal{L}}{\partial q_1} = \tau$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_1} = \left(\theta_1' + \theta_2' \sin^2(q_2)\right)\dot{q}_1 + \left(\theta_3' \cos(q_2)\right)\dot{q}_2 \Rightarrow$$

$$\frac{d}{dt}(\cdot) = \theta_1' \ddot{q}_1 + \theta_2'\left(2\sin(q_2)\cos(q_2)\dot{q}_1\dot{q}_2 + \sin^2(q_2)\ddot{q}_1\right) + \theta_3'\left(-\sin(q_2)\dot{q}_2^2 + \cos(q_2)\ddot{q}_2\right)$$

$$\frac{\partial \mathcal{L}}{\partial q_1} = 0 \qquad \therefore \quad \tau = \left[\theta_1' + \theta_2' \sin^2(q_2)\right]\ddot{q}_1 + \left[\theta_3' \cos(q_2)\right]\ddot{q}_2 + \left[2\theta_2' \sin(q_2)\cos(q_2)\dot{q}_1\dot{q}_2\right] + \left[-\theta_3' \sin(q_2)\right]\dot{q}_2^2$$

4

$$\tau_2 = 0 = \frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{q}_2}\right] - \frac{\partial \mathcal{L}}{\partial q_2}$$

4

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_2} = \theta_3' \cos(q_2)\dot{q}_1 + \theta_2' \dot{q}_2$$

$$\frac{d}{dt}(\cdot) = \theta_3'\left(\cancel{-\sin(q_2)\dot{q}_1\dot{q}_2} + \cos(q_2)\ddot{q}_1\right) + \theta_2' \ddot{q}_2$$

$$\frac{\partial \mathcal{L}}{\partial q_2} = \frac{1}{2}\left(\theta_2' \cdot 2\sin(q_2)\cos(q_2)\right)\dot{q}_1^2 - \cancel{\left(\theta_3' \sin(q_2)\dot{q}_1\dot{q}_2\right)} + \theta_4' g\sin(q_2)$$

$$\boxed{0 = \theta_3' \cos(q_2)\ddot{q}_1 + \theta_2' \ddot{q}_2 - \theta_2' \sin(q_2)\cos(q_2)\dot{q}_1^2 - \theta_4' g\sin(q_2)}$$

$$k_r q_1 = q_m \Rightarrow k_r \dot{q}_1 = \dot{q}_m$$

$$\tau = K_t K_r i_a$$

$$\tau \dot{q}_1 = \tau_m \dot{q}_m = \tau_m k_r \dot{q}_1 = k_t k_r i_a \dot{q}_1$$

$$i_a = \frac{\tau}{k_t k_r}$$

$$\boxed{V = R_a i_a + k_v \dot{q}_m = \frac{R_a}{k_t k_r}\tau + k_v k_r \dot{q}_1}$$

$$V = \frac{R_a}{k_t k_r}\big(\underbrace{\tau + \beta_1 \dot{q}_1}_{\text{actual } \tau \text{ when considering friction}}\big) + k_v k_r \dot{q}_1$$

$$v = \frac{R_a}{k_t k_r}\tau + \underline{\frac{R_a}{k_t k_r}}(\beta_1 \dot{q}_1) + k_v k_r \dot{q}_1$$

Plug in τ_1 for τ as calculated on bottom page ③

$$v = \left[\theta_1 + \theta_2 \sin^2(q_2)\right]\ddot{q}_1 + \left[\theta_3 \cos(q_2)\right]\ddot{q}_2 + \left[2\theta_2 \sin(q_2)\cos(q_2)\right]\dot{q}_1\dot{q}_2 - \left[\theta_3 \sin(q_2)\right]\dot{q}_2^2$$

$$+ \underbrace{\boxed{\beta_1 \frac{R_a}{k_t k_r} + k_v k_r}}_{\theta_5}\dot{q}_1$$

$$\boxed{v = \left[\theta_1 + \theta_2 \sin^2(q_2)\right]\ddot{q}_1 + \left[\theta_3 \cos(q_2)\right]\ddot{q}_2 + \left[2\theta_2 \sin(q_2)\cos(q_2)\right]\dot{q}_1\dot{q}_2 - \left[\theta_3 \sin(q_2)\right]\dot{q}_2^2 + \theta_5 \dot{q}_1}$$

Similarly, $\tau_2 = \tau_2 - \beta_2 \dot{q}_2$ when we consider friction

$$0 = 0 - \beta_2 \dot{q}_2$$

$\therefore \beta_2 \dot{q}_2 =$ "answer we had for τ_2 from top of page ④" $\theta_6 = \beta_2 \frac{R_a}{k_t k_r}$

Multiplying both sides by $\frac{R_a}{k_t k_r}$ for convenience...

$$\boxed{\left[\theta_3 \cos(q_2)\right]\ddot{q}_1 + \theta_2 \ddot{q}_2 - \theta_2 \sin(q_2)\cos(q_2)\dot{q}_1^2 - \theta_4 g \sin(q_2) + \theta_6 \dot{q}_2 = 0}$$

From the final equation here we can produce the matrices needed for simulation. To see this math (which picks up right where this ends), one can jump right to the summary section that came previously titled **Derivation of Matrices (Summary)**.

Implementation:

Simulink Model:

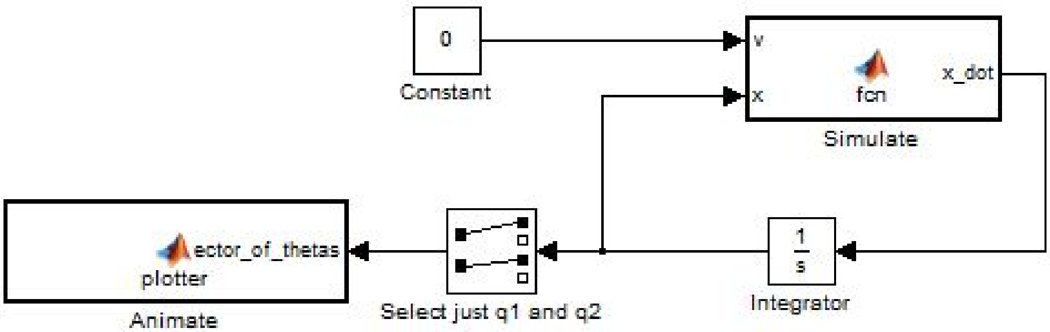

Figure 4: Implemented Simulink Model.

Simulation Block Contents:
This block takes the current state of the system, as defined by the position rate of change of the two generalized coordinates and spits out the derivative of the state of the system: velocity and angular acceleration. It uses theoretical parameters $\theta_1 - \theta_6$.

```
function x_dot = fcn(v,x)

q1 = x(1);
q2 = x(3);
q1_dot = x(2);
q2_dot = x(4);
%%%%%%%%%%%%%%%%%%%%%%%%  Parameters  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
J1 = 0.0044;
m2 = 0.14;
L1_L2_prime = 0.2;
L2 = 0.3;
L2c = 0.15;
b1 = 0.015;
b2 = 0.002;
kt = 0.00767;
kv = 0.00767;
kn = 0.00767;
kr = 70;
kg = 70;
Ra = 2.6;
Ru = 2.6;
grav = 9.81;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

theta1_prime = J1 + m2*(L1_L2_prime^2);
theta2_prime = 1/3*m2*L2^2;
theta3_prime = 1/2*m2*L1_L2_prime*L2;
theta4_prime = m2*L2c;
theta1 = theta1_prime*Ra/(kr*kt);
theta2 = theta2_prime*Ra/(kr*kt);
theta3 = theta3_prime*Ra/(kr*kt);
theta4 = theta4_prime*Ra/(kr*kt);
theta5 = b1*Ra/(kr*kt)+kr*kv;
theta6 = b2*Ra/(kr*kt);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Forces due to mass, depends on position of robot
M = [theta1+theta2*(sin(q2))^2 theta3*cos(q2);
    theta3*cos(q2) theta2];
% Centrifugal and coriolis forces
C = [2*theta2*sin(q2)*cos(q2)*q2_dot -theta3*sin(q2)*q2_dot;
    -theta2*sin(q2)*cos(q2)*q1_dot theta6];
% Frictional forces
F = [theta5*q1_dot;
    0];
% Gravitational forces
G = [0;
    theta4*grav*sin(q2)];
% acceleration
% q_dd = [q1_dd;
%     q2_dd];
% velocity
q_d = [q1_dot;
    q2_dot];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% M*q_dd+C*q_d+F+G = [v; 0] invalid line of code, just for reference

q_dd = inv(M)*(-C*q_d-F-G+[v;0]);

x_dot = [q_d(1),q_dd(1),q_d(2),q_dd(2)];

end
```

Animation Block Contents:

This block takes the two positional values q_1 and q_2 and plots them.

```
function plotter(vector_of_thetas)
vector_of_thetas = vector_of_thetas';
%Frame #0 x, y and z coordinates for the limb in frame #0.  (robot base)
%Ex: the 2nd point of the base is (x0(1),y0(1),z0(1)) = (0,0,-10)
x0 = [0 0];
y0 = [0 0];
z0 = [0 -10];

%Frame #1 x, and z coordinates for the limb in frame #1. (middle arm)
x1 = [0 0 0 0 0];
y1 = [0 0 6 6 8];
z1 = [0 -1 -1 0 0];

%Frame #2 x, y and z coordinates for the limb in frame #2.  (swing arm)
x2 = [0 0];
y2 = [0 12];
z2 = [0 0];

    theta1 = vector_of_thetas(1);
    theta2 = vector_of_thetas(2);

    axis([-10 10 -10 10 -10 10]);

    %Transform points of the middle arm from Frame 1 to Frame 0
    middle_arm_pts_in_F0 = zeros(3,length(x1));
    for i=1:length(x1)
       middle_arm_pts_in_F0(:,i) = trans0_1_day6(x1(i), y1(i), z1(i),theta1);
    end

    %Transform points of swing arm from Frame 2 to Frame 0
    swing_arm_pts_in_F1 = zeros(3,length(x2));
    swing_arm_pts_in_F0 = zeros(3,length(x2));
    for j=1:length(x2)
       %transform pts from Frame 2 to Frame 1
       swing_arm_pts_in_F1(:,j) = trans1_2_day6(x2(j), y2(j), z2(j),theta2);
       %transform pts from Frame 1 to Frame 0
       swing_arm_pts_in_F0(:,j) = trans0_1_day6(swing_arm_pts_in_F1(1,j), swing_arm_pts_in_F1(2,j), swing_arm_pts_in_
    end

    %plot all limb pts (base, middle & swing arms) transformed in Frame 0
    hold off;
    % plot base
    plot3([0 0], [0 0], [0 -10]);

    %reset plot parameters
    title('Animated Robot in Frame 0');
    xlabel('x');ylabel('y');zlabel('z');
    axis([-10 10 -10 10 -10 10]);
    grid on;
    hold on;

    %plot middle arm
    plot3(middle_arm_pts_in_F0(1,:)',middle_arm_pts_in_F0(2,:)',middle_arm_pts_in_F0(3,:)','r');
    %plot swing arm
    plot3(swing_arm_pts_in_F0(1,:)',swing_arm_pts_in_F0(2,:)',swing_arm_pts_in_F0(3,:)','k');
    %plot End effector
    plot3(swing_arm_pts_in_F0(1,length(x2))',swing_arm_pts_in_F0(2,length(x2))',swing_arm_pts_in_F0(3,length(x2))','g
end
```