BMED436 Medical Robot
2018250064 Sibeen Kim

Manipulator Dynamics

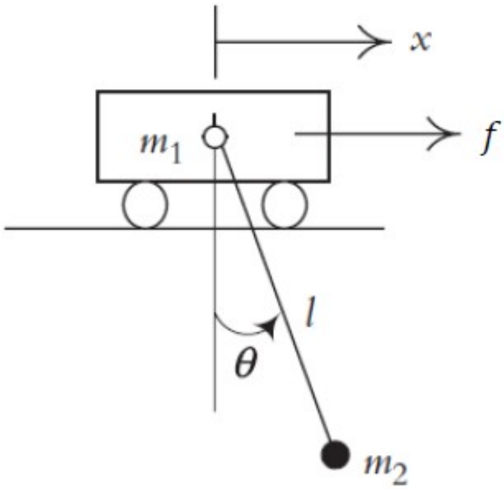

Figure 1: Cart-Pendulum System.

Figure 1 shows a cart-pendulum system. This setup consists of a cart that can move along a horizontal axis and a pendulum attached to it, which swings freely.

1 Dynamic Equations

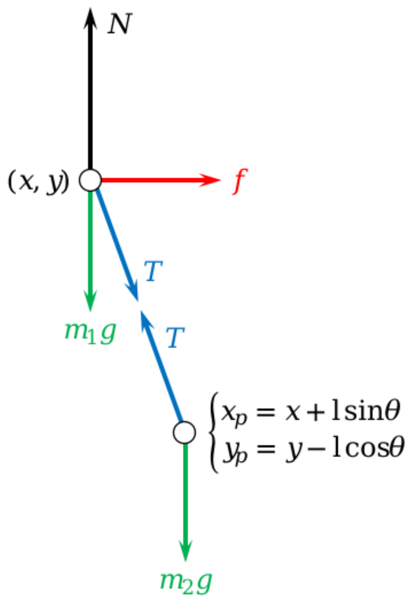

Figure 2: Free Body Diagram of the Cart-Pendulum System.

The relationship between the applied forces and the resulting motion of the cart and pendulum is described by dynamic equations. The dynamic equations will be derived using two approaches: the Newton method, which relies on force and acceleration relationships, and the Lagrangian method, which involves energy-based principles.

Figure 2 illustrates the forces acting on the cart-pendulum system, including gravitational forces, tension, and any applied forces. These forces will serve as the basis for deriving the equations of motion for the cart and pendulum.

1.1 Newton Method

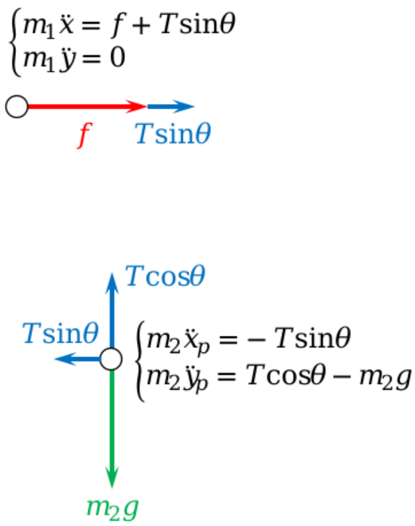

Figure 3: Free Body Diagram for Newton's Second Law Analysis.

$$m_1\ddot{x} = f + T\sin\theta \tag{1}$$
$$T = (m_1\ddot{x} - f)\csc\theta \tag{2}$$

$$m_2\ddot{x}_p = -T\sin\theta \qquad m_2\ddot{y}_p = T\cos\theta - m_2 g \tag{3}$$
$$m_2\ddot{x}_p = f - m_1\ddot{x} \qquad m_2\ddot{y}_p = (m_1\ddot{x} - f)\cot\theta - m_2 g \tag{4}$$

$$x_p = x + l\sin\theta \qquad y_p = y - l\cos\theta \tag{5}$$
$$\dot{x}_p = \dot{x} + l\cos\theta\dot{\theta} \qquad \dot{y}_p = l\sin\theta\dot{\theta} \tag{6}$$
$$\ddot{x}_p = \ddot{x} - l\sin\theta\dot{\theta}^2 + l\cos\theta\ddot{\theta} \qquad \ddot{y}_p = l\cos\theta\dot{\theta}^2 + l\sin\theta\ddot{\theta} \tag{7}$$

$$(m_1 + m_2)\ddot{x} + m_2 l\cos\theta\ddot{\theta} - m_2 l\sin\theta\dot{\theta}^2 = f \tag{8}$$
$$m_2 l\cos\theta\ddot{x} + m_2 l^2\ddot{\theta} + m_2 g l\sin\theta = 0 \tag{9}$$

1.2 Lagrangian Method

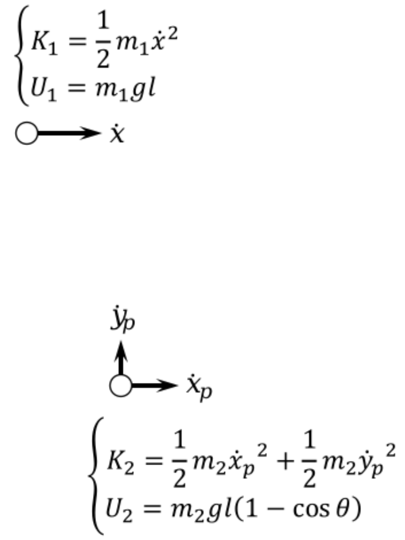

Figure 4: Free Body Diagram for Lagrangian Formulation.

$$
\begin{aligned}
K &= \frac{1}{2}(m_1 + m_2)\dot{x}^2 + m_2 l \cos\theta \dot{x}\dot{\theta} + \frac{1}{2} m_2 l^2 \dot{\theta}^2 && (10)\\
U &= m_1 g l + m_2 g l (1 - \cos\theta) && (11)\\
L &= K - U && (12)\\
&= \frac{1}{2}(m_1 + m_2)\dot{x}^2 + m_2 l \cos\theta \dot{x}\dot{\theta} + \frac{1}{2} m_2 l^2 \dot{\theta}^2 && (13)\\
&\quad - m_1 g l - m_2 g l (1 - \cos\theta) && (14)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial \dot{x}} &= (m_1 + m_2)\dot{x} + m_2 l \cos\theta \dot{\theta} && (15)\\
\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} &= (m_1 + m_2)\ddot{x} - m_2 l \sin\theta \dot{\theta}^2 + m_2 l \cos\theta \ddot{\theta} && (16)\\
\frac{\partial L}{\partial x} &= 0 && (17)\\
f &= \frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} && (18)\\
&= (m_1 + m_2)\ddot{x} - m_2 l \sin\theta \dot{\theta}^2 + m_2 l \cos\theta \ddot{\theta} && (19)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial \dot{\theta}} &= m_2 l \cos\theta \dot{x} + m_2 l^2 \dot{\theta} && (20)\\
\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} &= -m_2 l \sin\theta \dot{x}\dot{\theta} + m_2 l \cos\theta \ddot{x} + m_2 l^2 \ddot{\theta} && (21)\\
\frac{\partial L}{\partial \theta} &= -m_2 l \sin\theta \dot{x}\dot{\theta} - m_2 g l \sin\theta && (22)\\
0 &= \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} && (23)\\
&= m_2 l \cos\theta \ddot{x} + m_2 l^2 \ddot{\theta} + m_2 g l \sin\theta && (24)
\end{aligned}
$$

1.3 The Configuration-Space Equation

The derived dynamic equations can be consolidated into the configuration-space equation. This matrix form expresses the system's dynamics in terms of the mass matrix $\mathbf{M}$, the centrifugal term matrix $\mathbf{C}$, and the gravity term vector $\mathbf{G}$. Note that the Coriolis term matrix $\mathbf{B} = \mathbf{0}$ in this system, and nonrigid body effects are neglected.

$$
\begin{bmatrix} f \\ 0 \end{bmatrix} = \begin{bmatrix} m_1 + m_2 & m_2 l c_\theta \\ m_2 l c_\theta & m_2 l^2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -m_2 l s_\theta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}^2 \\ \dot{\theta}^2 \end{bmatrix} + \begin{bmatrix} 0 \\ m_2 g l s_\theta \end{bmatrix} \tag{25}
$$

$$
\tau = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\left(\dot{\mathbf{q}} \odot \dot{\mathbf{q}}\right) + \mathbf{G} \text{ where } \mathbf{q} = \begin{bmatrix} x \\ \theta \end{bmatrix} \tag{26}
$$

2 Dynamic Simulation

To simulate the system's motion, we begin by calculating the acceleration vector $\ddot{\mathbf{q}}$ using the configuration-space equation:

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1} \left(\tau - \mathbf{C} \left(\dot{\mathbf{q}} \odot \dot{\mathbf{q}} \right) - \mathbf{G} \right) \tag{27}$$

Euler integration is then applied iteratively to update the velocity and position at each timestep. Given $\ddot{\mathbf{q}}$ at the current timestep, we can calculate the necessary increments for $\dot{\mathbf{q}}$ and $\mathbf{q}$ at the next timestep as follows:

$$\Delta\dot{\mathbf{q}} = \ddot{\mathbf{q}}\Delta t \tag{28}$$

$$\Delta\mathbf{q} = \dot{\mathbf{q}}\Delta t + \frac{1}{2}\ddot{\mathbf{q}}(\Delta t)^2 \tag{29}$$

Listing 1 demonstrates the iterative calculation of the dynamic states using Euler integration. The entire source code for this simulation is available in `Supplementary Material`.

```
q(:,1) = q0;
q_dot(:,1) = q_dot0;

for i = 1:T-1
    M = [m1+m2, m2*l*cos(q(2,i)); m2*l*cos(q(2,i)), m2*l^2];
    C = [0, -m2*l*sin(q(2,i)); 0, 0] * q_dot(:,i).^2;
    G = [0; m2*g*l*sin(q(2,i))];
    q_ddot(:,i) = M \ ([f(t(i)); 0]-C-G);

    dt = t(i+1)-t(i);
    q_dot(:,i+1) = q_dot(:,i) + q_ddot(:,i)*dt;
    q(:,i+1) = q(:,i) + q_dot(:,i)*dt + 0.5*q_ddot(:,i)*dt^2;
end
```

Listing 1: Dynamic Simulation of Cart-Pendulum System

2.1 $\mathbf{q}(0) = \begin{bmatrix} 0 \\ -\frac{\pi}{3} \end{bmatrix}, \ \dot{\mathbf{q}}(0) = \mathbf{0}, \ f = 0$

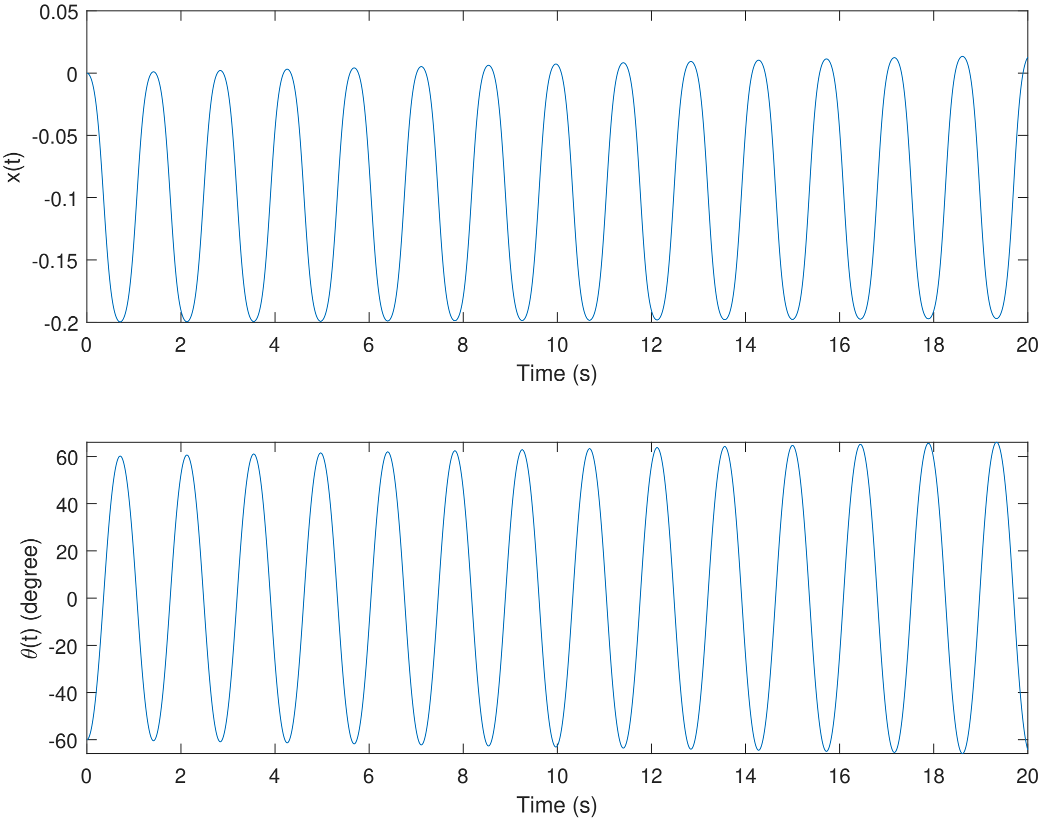

Figure 5: Motion of Cart-Pendulum System.

Figure 5 shows the simulated motion of the cart-pendulum system over a period of 20 seconds. The upper plot represents the displacement of the cart $x(t)$ in meters, while the lower plot shows the pendulum's angle $\theta(t)$ in degrees. Initial conditions were set as $\mathbf{q}(0) = \begin{bmatrix} 0 & -\frac{\pi}{3} \end{bmatrix}^T$, with zero initial velocity $\dot{\mathbf{q}}(0) = \mathbf{0}$ and no external force applied ($f = 0$). The results illustrate the oscillatory behavior of both the cart and the pendulum under these initial conditions. Animation is provided in `[link]`.

2.2 $\mathbf{q}(0) = \mathbf{0},\ \dot{\mathbf{q}}(0) = \mathbf{0},\ f = 2\sin\pi t$

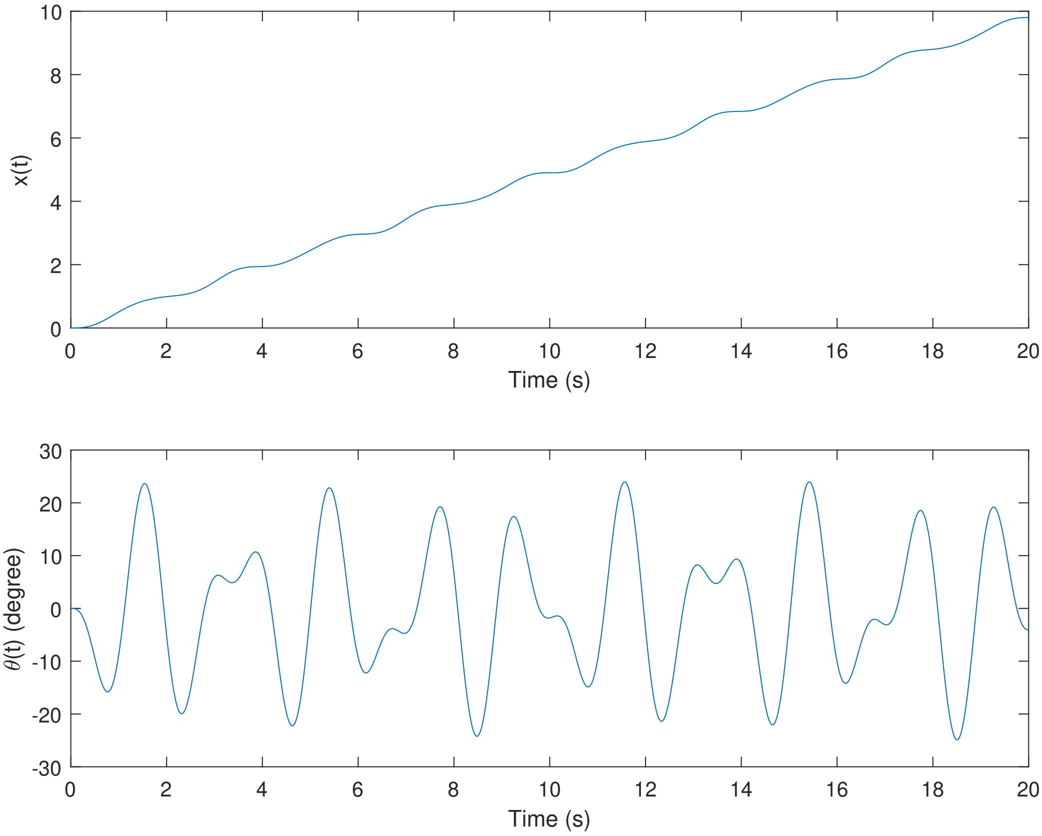

Figure 6: Motion of Cart-Pendulum System.

Figure 6 shows the simulated motion of the cart-pendulum system over a period of 20 seconds. The upper plot represents the displacement of the cart $x(t)$ in meters, while the lower plot shows the pendulum's angle $\theta(t)$ in degrees. Initial conditions were set as $\mathbf{q}(0) = \mathbf{0}$, with zero initial velocity $\dot{\mathbf{q}}(0) = \mathbf{0}$ and an external force $f = 2\sin\pi t$ applied.

Notably, the cart continuously moves to the right over the entire 20-second period. This consistent rightward motion results from two factors: the phase of the sine function and the relatively small influence of the pendulum. The sine function's phase ensures that after each deceleration, the cart regains rightward momentum, preventing any lasting reversal in direction. Additionally, the pendulum's influence is minimal due to its small mass compared to the cart and a zero initial angle, so its oscillations do not significantly affect the cart's dominant rightward movement. Animation is provided in `[link]`.

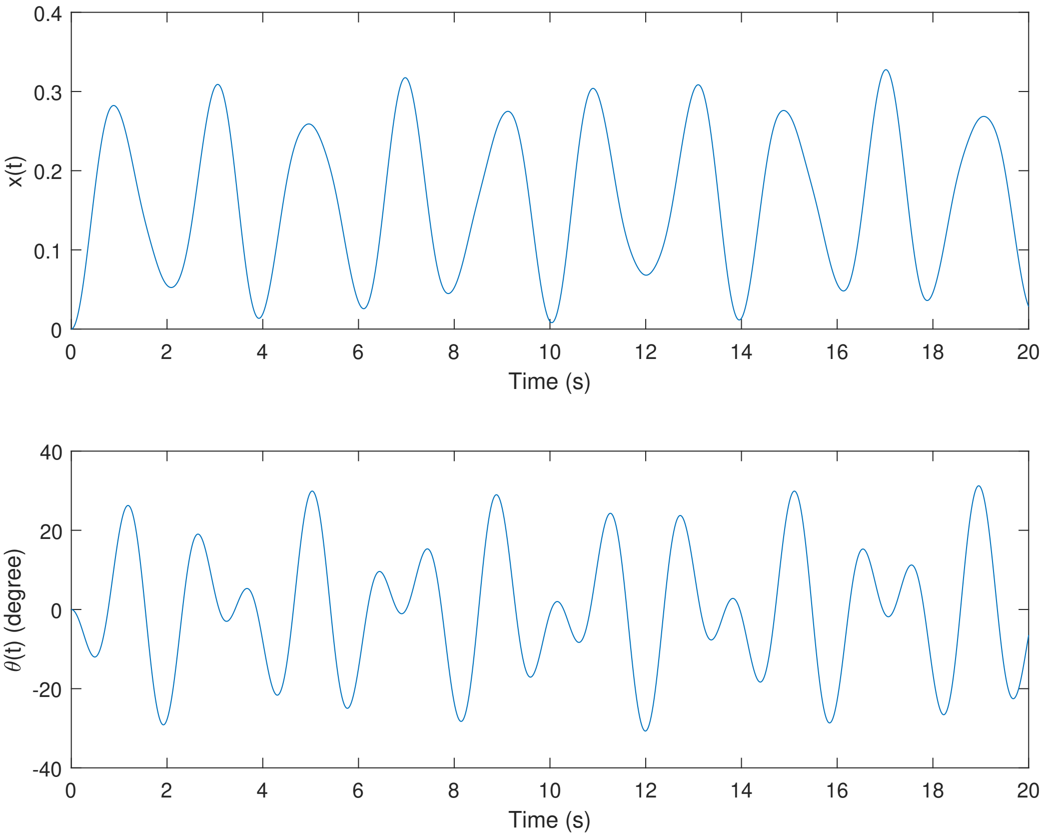

Figure 7: Motion of Cart-Pendulum System.

Figure 7 shows the results when the applied force is changed to $f = 2\cos \pi t$, shifting the phase by 90 degrees. With this change, the cart's motion includes direction reversal, moving both left and right. This phase shift reveals that the constant rightward movement seen with $f = 2\sin \pi t$ is due to the specific phase of the sine function, and shifting to cosine introduces the reversal. Animation is provided in `[link]`.

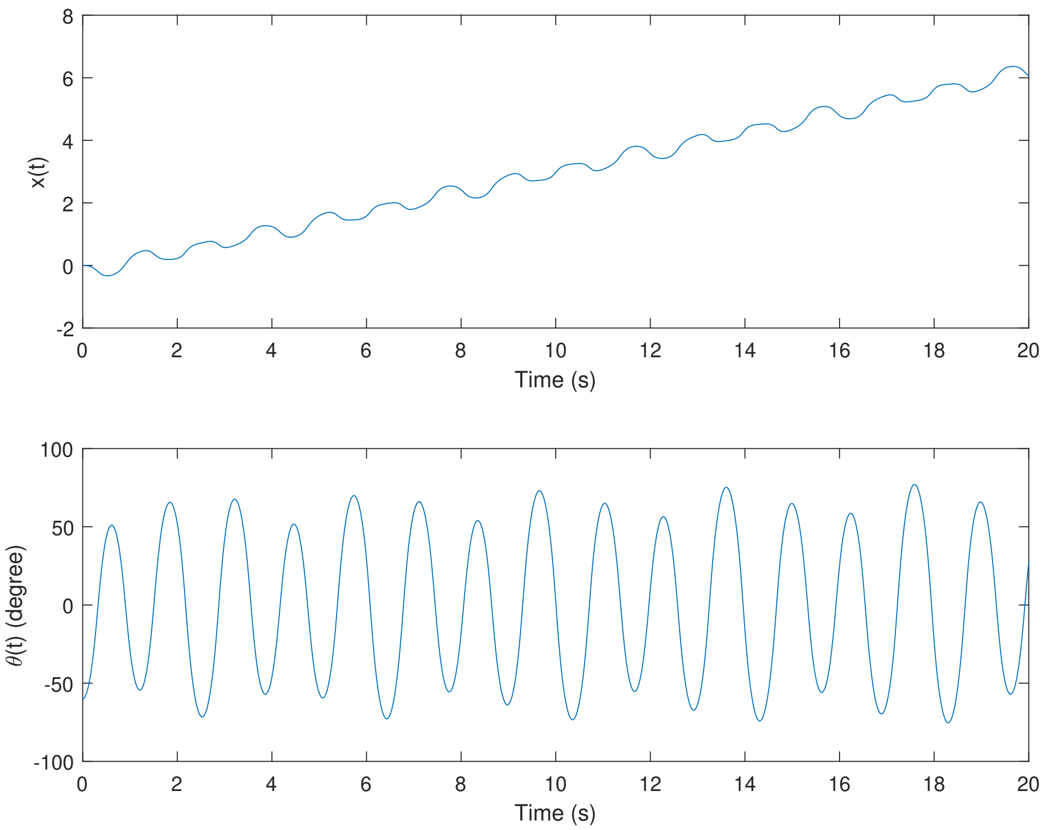

Figure 8: Motion of Cart-Pendulum System.

Figure 8 shows that direction reversal can also be achieved with the original sine force $f = 2\sin\pi t$ by increasing the pendulum mass m_2 from 0.3 kg to 1 kg and setting a larger initial angle $\theta(0) = -60°$. In the initial setup, the pendulum's effect on the cart was minimal due to its small mass and zero initial angle. However, by increasing m_2 and setting a significant non-zero initial angle, the pendulum exerts a stronger influence on the system, resulting in observable reversals in the cart's direction. This demonstrates that modifying the pendulum's mass and initial angle can also induce direction-reversing behavior. Animation is provided in `[link]`.

Supplementary Material

```matlab
%% f = @(t) 0; q0 = [0; deg2rad(-60)]; m_2 = 0.3;

f = @(t) 0;
q0 = [0; deg2rad(-60)];
q_dot0 = [0; 0];
t = 0:0.001:20;
q = simulate_cart_pendulum_dynamics(f, q0, q_dot0, t);

plot_cart_pendulum_dynamics('cart_pendulum_1', q, t);
animate_cart_pendulum_dynamics('cart_pendulum_1.gif', q, t);

%% f = @(t) 2*sin(pi*t); q0 = [0; 0]; m_2 = 0.3;

f = @(t) 2*sin(pi*t);
q0 = [0; 0];
q_dot0 = [0; 0];
t = 0:0.001:20;
q = simulate_cart_pendulum_dynamics(f, q0, q_dot0, t);

plot_cart_pendulum_dynamics('cart_pendulum_2', q, t);
animate_cart_pendulum_dynamics('cart_pendulum_2.gif', q, t);

%% f = @(t) 2*cos(pi*t); q0 = [0; 0]; m_2 = 0.3;

f = @(t) 2*cos(pi*t);
q0 = [0; 0];
q_dot0 = [0; 0];
t = 0:0.001:20;
q = simulate_cart_pendulum_dynamics(f, q0, q_dot0, t);

plot_cart_pendulum_dynamics('cart_pendulum_3', q, t);
animate_cart_pendulum_dynamics('cart_pendulum_3.gif', q, t);

%% f = @(t) 2*sin(pi*t); q0 = [0; deg2rad(-60)]; m_2 = 1;

f = @(t) 2*sin(pi*t);
q0 = [0; deg2rad(-60)];
q_dot0 = [0; 0];
t = 0:0.001:20;
q = simulate_cart_pendulum_dynamics(f, q0, q_dot0, t, 1, 1);

plot_cart_pendulum_dynamics('cart_pendulum_4', q, t);
animate_cart_pendulum_dynamics('cart_pendulum_4.gif', q, t);
```

Listing 2: Dynamic Simulation

```
function q = simulate_cart_pendulum_dynamics(f, q0, q_dot0, t,
    m1, m2, l, g)
    arguments
        f function_handle        % External force (N)
        q0 (2,1) double          % Initial position
        q_dot0 (2,1) double      % Initial velocity
        t (1,:) double           % Timesteps (s)
        m1 (1,1) double = 1      % Cart mass (kg)
        m2 (1,1) double = 0.3    % Pendulum mass (kg)
        l (1,1) double = 0.5     % Pendulum length (m)
        g (1,1) double = 9.81
    end

    T = length(t);
    q = zeros(2, T);
    q_dot = zeros(2, T);
    q_ddot = zeros(2, T);

    q(:,1) = q0;
    q_dot(:,1) = q_dot0;

    for i = 1:T-1
        M = [m1+m2, m2*l*cos(q(2,i)); m2*l*cos(q(2,i)), m2*l^2];
        C = [0, -m2*l*sin(q(2,i)); 0, 0] * q_dot(:,i).^2;
        G = [0; m2*g*l*sin(q(2,i))];
        q_ddot(:,i) = M \ ([f(t(i)); 0]-C-G);

        dt = t(i+1)-t(i);
        q_dot(:,i+1) = q_dot(:,i) + q_ddot(:,i)*dt;
        q(:,i+1) = q(:,i) + q_dot(:,i)*dt + 0.5*q_ddot(:,i)*dt
            ^2;
    end
end
```

Listing 3: function simulate_cart_pendulum_dynamics

```matlab
function plot_cart_pendulum_dynamics(filepath, q, t, width,
    height)
     arguments
         filepath string
         q (2,:) double              % Position ([(m); (rad)])
         t (1,:) double              % Timesteps (s)
         width (1,1) double = 800    % Figure width
         height (1,1) double = 600 % Figure height
     end

     fig = figure;
     fig.Position(3) = width;
     fig.Position(4) = height;

     subplot(2,1,1);
     plot(t, q(1,:));
     xlabel('Time (s)');
     ylabel('x(t)');

     subplot(2,1,2);
     plot(t, rad2deg(q(2,:)));
     xlabel('Time (s)');
     ylabel('\theta(t) (degree)');

     print(filepath, '-depsc');
end
```

Listing 4: function `plot_cart_pendulum_dynamics`

```matlab
function animate_cart_pendulum_dynamics(filepath, q, t, l,
    skip_every)
    arguments
        filepath string
        q (2,:) double           % Position ([(m); (rad)])
        t (1,:) double           % Timesteps (s)
        l (1,1) double = 0.5     % Pendulum length (m)
        skip_every (1,1) double = 100
    end

    H = (1.00 / 4.0) * l; % Cart height (m)
    W = (2.50 / 4.0) * l; % Cart width (m)
    r = (0.15 / 4.0) * l; % Pendulum mass radius (m)
    d = (0.50 / 4.0) * l; % Offset (m)

    figure('Position', [100, 100, 1500, 800]); axis equal;
    xlim([min(q(1,:))-l-d, max(q(1,:))+l+d]);
    ylim([d+H/2-l-d,         d+H/2+l+d]);
    yline(0);
    hold on; grid on;

    r0 = [q(1,1),               d+H/2];
    r1 = [r0(1)+l*sin(q(2,1)), r0(2)-l*cos(q(2,1))];

    cart = rectangle('Position', [r0(1)-W/2, r0(2)-H/2, W, H]);
    wheel_L = rectangle('Position', [r0(1)-2*d, 0, d, d], ...
                        'Curvature', [1, 1]);
    wheel_R = rectangle('Position', [r0(1)+d, 0, d, d], ...
                        'Curvature', [1, 1]);

    pendulum = plot([r0(1), r1(1)], [r0(2), r1(2)], Color='k');
    pendulum_mass = rectangle('Position', [r1(1)-r, r1(2)-r, 2*r
        , 2*r], ...
                              'Curvature', [1, 1], ...
                              'FaceColor', 'k');

    for i = 1:skip_every:length(t)
        r0 = [q(1,i),               d+H/2];
        r1 = [r0(1)+l*sin(q(2,i)), r0(2)-l*cos(q(2,i))];

        cart.Position(1) = r0(1)-W/2;
        wheel_L.Position(1) = r0(1)-2*d;
        wheel_R.Position(1) = r0(1)+d;

        pendulum.XData = [r0(1), r1(1)];
        pendulum.YData = [r0(2), r1(2)];

        pendulum_mass.Position(1) = r1(1)-r;
```

```
        pendulum_mass.Position(2) = r1(2)-r;

        frame = getframe(gcf);
        im = frame2im(frame);
        [imind, cm] = rgb2ind(im, 256);

        if i == 1
            imwrite(imind, cm, filepath, 'gif', ...
                    'Loopcount', inf, 'DelayTime', 0.05);
        else
            imwrite(imind, cm, filepath, 'gif', ...
                    'WriteMode', 'append', 'DelayTime', 0.05);
        end
    end
end
```

Listing 5: function `animate_cart_pendulum_dynamics`