Laboratory Experiment I : Modelling of a Monorail Crane

MTRN3020 - Modelling and Control of Mechatronic Systems

I verify that the contents of this report are my own work

Zachary Hamid
z5059915
15/09/2017

Contents

1 Introduction

The purpose of this experiment was to correctly model a cart and pendulum system as seen in the following figure:

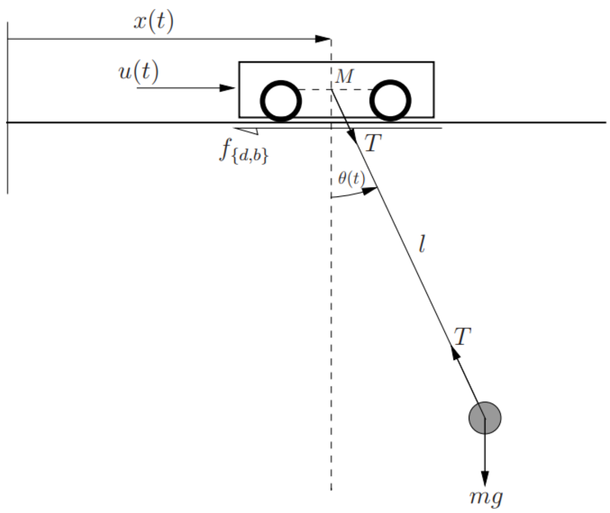

Figure 1: Schematic diagram of cart and pendulum system used in experiment[1]

The modelling procedure involved both mathematical derivations for the behaviour of the system and some trial-and-error and curve-fitting to fit simulated data to collected experimental data in order to obtain important constants that describe the driving and braking systems. In doing so, control could be executed over the systems output and, for a given T_1 (time after which system starts braking for the first time), times T_2 and T_3 could be found such that the cart and pendulum come to rest at the same time after two drive-brake cycles, with T_2 describing the time between the end of the first braking cycle and start of second driving cycle and T_3 describing the between the end of the second driving cycle and start of the second braking cycle.

2 Determining Gains and Time Constants

In order to model the system, we must first find the time constants for the driving and braking motion of the system, τ_d and τ_b, respectively. Additionally, the gains for the driving and braking motion must be found as well, which are A_d and A_b, respectively. In order to find the constants A_d and τ_d, the transfer function for the system must be used, which is as follows:

$$G(s) = \frac{X(s)}{V(s)} = \frac{A}{s(1+\tau s)} \tag{1}$$

Rearranging the above we arrive at:

$$sX(s) = \frac{A}{s(1+\tau s)}V(s) \tag{2}$$

Applying the Inverse Laplace Transform to Eq. (2), we get the following:

$$\dot{x}(t) = A(1 - e^{-\frac{t}{\tau}})v(t) \tag{3}$$

The Eq. (3) above denotes the first-order equation that will be fitted to experimental data obtained from the MATLAB file *STATPEND.m*. It should be noted that Eq. (3) can only directly be used for the driving system, for the braking system, another method will need to be used.

For the driving system, Eq. (3) was used with MATLAB's curve-fitting tool, *cftool*, and the following fit was obtained by isolating the data related to driving the system:

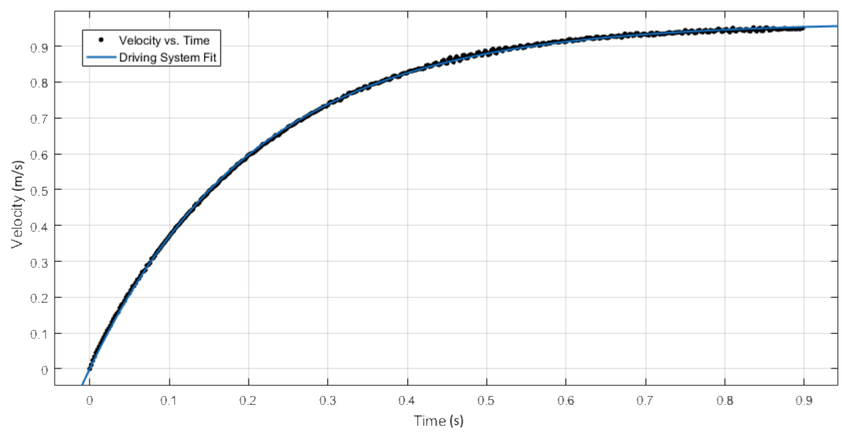

Figure 2: MATLAB *cftool* fit for driving system data from STATPEND.m

From the fit in Fig. 2, the constants were found to be:

$$A_d = 0.0805 \tag{4}$$
$$\tau_d = 0.2079 \text{ s} \tag{5}$$

In order to find the braking system constants, A_b and τ_b, the system had to first be modelled in MATLAB (See Appendix A) using Eq. (3) for the cart motion and a derived Newton-Euler equation for the pendulum motion, this derivation will be covered in a later section. Once a model was created, it had to be simulated (See Appendix B) and plotted. Using a plot of velocity against time as a visual tool, the braking constants, A_b and τ_b, were found by trial-and-error. It should be noted that the braking voltage was assumed to be -1V. The best fitting constants were found to be:

$$A_b = 0.0835588 \tag{6}$$
$$\tau_b = 0.1452650 \text{ s} \tag{7}$$

Which, when simulated against the real data, yields the following plot:

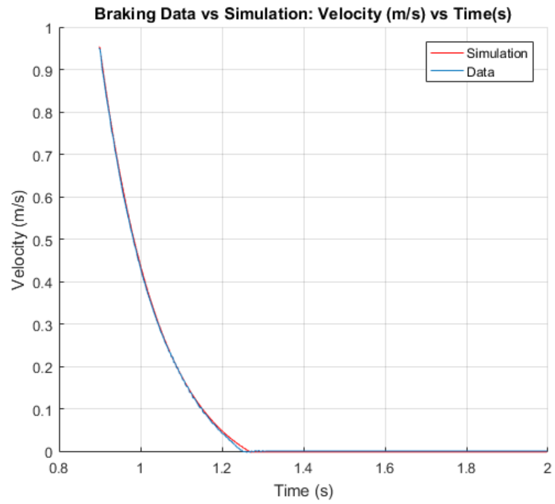

Figure 3: MATLAB simulation using results (6) and (7) against actual data

While not a perfect fit, a compromise had to be made for how well it fit the velocity data and how well it fit the position data, as some values resulted in the position plot shape being incorrect due to integration inaccuracies.

3 Pendulum Equation Derivation

A free-body diagram of the pendulum can be obtained by simplifying Fig. 1:

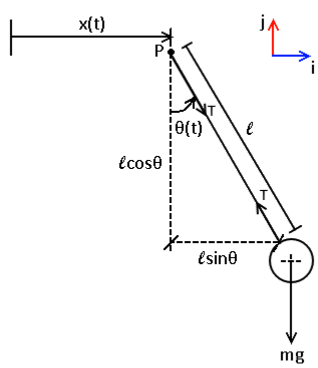

Figure 4: Free-body diagram of pendulum

From Fig. 4 it should be noted that $x(t)$ is a displacement from an origin to the left of $x(t)$. Now separating the acceleration of the pendulum into the horizontal and vertical components $a_{p,i}$ and $a_{p,j}$, respectively, and then further separating these into their tangential ($\ddot{\theta}r$) and radial ($\omega^2 r$), components

$a_{t,i/j}$ and $a_{r,i/j}$, respectively, the following results were obtained:

$$a_{p,i} = a_{r,i} + a_{t,i} + \tfrac{d^2}{dt^2}(x) = -l\sin\theta * \omega^2 + \tfrac{d^2}{dt^2}(\theta) * l\cos\theta + \tfrac{d^2}{dt^2}(x) \tag{8}$$

$$a_{p,j} = a_{r,j} + a_{t,j} = l\cos\theta * \omega^2 + l\sin\theta * \tfrac{d^2}{dt^2}(\theta) \tag{9}$$

Finding the forces in the horizontal (i) and vertical (j) directions by Newton's Second Law of Motion and using the results (8) and (9) above:

$$\sum F_i = ma_{p,i} = -ml\sin\theta * \omega^2 + m\frac{d^2}{dt^2}(\theta) * l\cos\theta + m\frac{d^2}{dt^2}(x) = -T\sin\theta \tag{10}$$

$$\sum F_j = ma_{p,j} = ml\cos\theta * \omega^2 + ml\sin\theta * \frac{d^2}{dt^2}(\theta) = -mg + T\cos\theta \tag{11}$$

Rearranging Eq. (10) and (11) for T:

$$T = -\frac{1}{\sin\theta}(-ml\sin\theta * \omega^2 + m\frac{d^2}{dt^2}(\theta) * l\cos\theta + m\frac{d^2}{dt^2}(x)) \tag{12}$$

$$T = \frac{1}{\cos\theta}(ml\cos\theta * \omega^2 + ml\sin\theta * \frac{d^2}{dt^2}(\theta) + mg) \tag{13}$$

Setting (12) = (13) and multipling both sides by $\cos\theta\sin\theta$:

$$ml\sin\theta\cos\theta\omega^2 - ml\cos^2\theta\frac{d^2}{dt^2}(\theta) - m\cos\theta\frac{d^2}{dt^2}(x) = ml\sin\theta\cos\theta\omega^2 + ml\sin^2\theta\frac{d^2}{dt^2}(\theta) + mg\sin\theta \tag{14}$$

Simplifying Eq. (14) we get:

$$l\frac{d^2}{dt^2}(\theta)(\sin^2\theta + \cos^2\theta) + \cos\theta\frac{d^2}{dt^2}(x) + g\sin\theta = 0 \tag{15}$$

Which finally simplifies down to the required equation:

$$\boxed{l\ddot{\theta} + \cos\theta\ddot{x} + g\sin\theta = 0} \tag{16}$$

4 Validating System Model

Using a value of $T_1 = 266$ ms (from student data), the following simulation was made:

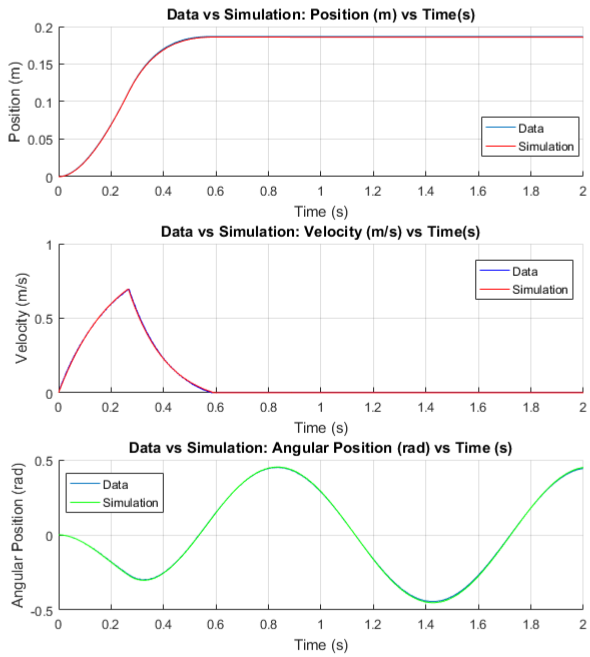

Figure 5: MATLAB simulation against actual data of $x(t)$, $\dot{x}(t)$, and $\theta(t)$

It can be seen directly from the figure above that the simulation fits the data well and thus can be considered to be valid using the results (6) and (7). However, analytical validity is a better metric for determining the accuracy of the simulation. Therefore, three analytical methods will be used which are the mean-square error (MSE) and the maximum absolute error (MAE) and the maximum percentage error (MPE). The mean-square error is denoted by the equation:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{17}$$

As the name suggests, it is simply the difference between the theoretical value and experimental value (the error) squared (to remove sign and provide more weight to larger values), and then the mean of the set of errors is taken (N is the number of data pairs).

The maximum absolute error is denoted by:

$$MAE = \max_{i=1}^{N} |y_i - \hat{y}_i| \tag{18}$$

And the maximum percentage error is given by:

$$MPE = \max_{i=1}^{N}\left(\frac{|y_i - \hat{y}_i|}{y_i}\right) \text{ x } 100\% \tag{19}$$

Calculating the mean-square error, maximum absolute error and maximum percentage error for $x(t)$, $\dot{x}(t)$, and $\theta(t)$, taking y_i as the actual data value and $\hat{y}_i$ as the simulated value yields the following table:

Table 1: MSE, MAE and MPE for $x(t)$, $\dot{x}(t)$ and $\theta(t)$

	MSE	MAE	MPE (%)
$x(t)$	6.8231 x 10^{-7} m^2	0.0013 m	0.696
$\dot{x}(t)$	8.5868 x 10^{-6} m^2/s^2	0.0141 m/s	2.029
$\theta(t)$	1.6234 x 10^{-5} rad^2	0.0086 rad	1.915

It can be seen from Table 1 that the errors are overall relatively small, which is confirmed by the maximum percentage errors being $\leq$ 2.03% and so the model can be considered valid for the purpose of this experiment.

5 Derivation of State Equations

Using the MATLAB script mentioned earlier that models the system (See Appendix A), a simulation can be developed to show the effect of two drive-brake sequences on the cart and pendulum. The script uses state equations to model the system with an input vector of t and x, which contains the state variables, where:

$$y = \begin{pmatrix} x(t) \\ \dot{x}(t) \\ \theta(t) \\ \dot{\theta}(t) \end{pmatrix} \tag{20}$$

Where $x(t)$ is the cart position, and $\theta(t)$ is the pendulum angular position. The model outputs a vector of the same size which represents the state equations of the system:

$$\dot{y} = \begin{pmatrix} \dot{x}(t) \\ \ddot{x}(t) \\ \dot{\theta}(t) \\ \ddot{\theta}(t) \end{pmatrix} \tag{21}$$

The above state equation vector can be integrated across a fixed-time interval to give the system behaviour for that time period. It can be seen from the above equations that:

$$\dot{y}_1 = y_2 \tag{22}$$

$$\dot{y}_3 = y_4 \tag{23}$$

To find $\dot{y}_2$, the transfer function from Eq. (1) can be rearranged:

$$X(s) * [s(1 + \tau s)] = A * V(s)$$
$$\rightarrow \tau s^2 X(s) + sX(s) = A * V(s)$$

Taking the Inverse Laplace Transform of the above, the following expression is obtained:

$$\rightarrow \tau \ddot{x} + \dot{x} = Av \tag{24}$$

From this we can rearrange to obtain $\ddot{x}$:

$$\ddot{x} = \frac{1}{\tau}[Av - \dot{x}] = \frac{1}{\tau}[Av - y_2] \tag{25}$$

Where τ, A, and v are determined by whether the system is braking or driving; if the system is driving, $v = 12$V, $A = A_d$ and $\tau = \tau_d$, if the system is braking then $v = -1$V, $A = A_b$ and $\tau = \tau_b$. Lines 27-50 of the MATLAB script (See Appendix A) handle setting the appropriate constants based on whether the system is driving or braking (which is subsequently based on the time, t).

To find the last required variable, $\ddot{\theta}$, the derivation made in Section 3 is required. Rearranging that equation:

$$l\ddot{\theta} + g\sin\theta = -\cos\theta\ddot{x} \tag{26}$$

Rearranging the above for $\ddot{\theta}$:

$$\ddot{\theta} = -\frac{g}{l}\sin\theta - \frac{1}{l}\cos\theta\ddot{x} = -\frac{g}{l}\sin(y_3) - \frac{1}{l}\cos(y_3)\dot{y}_2 \tag{27}$$

6 Simulation

With this, the derivation of the required state equations is complete, and so a simulation of two drive-brake cycles can be developed, choosing $T_1 = 266$ ms from the student data and a random T_2 and T_3, the following is plotted:

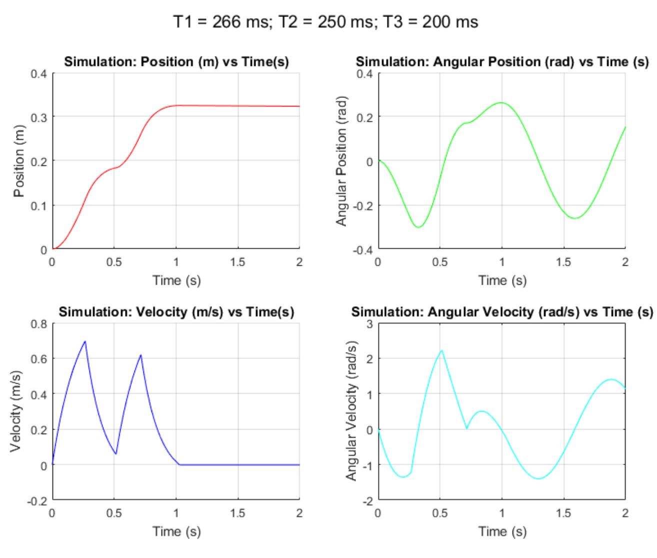

Figure 6: MATLAB simulation of two drive-brake cycles

It can be seen from the velocity plot in the above figure that at $t = T_1 + T_2 + T_3 = 716$ ms, the system starts braking for the second time and the velocity settles to 0 m/s and the cart position settles to a constant value. However, the pendulum velocity is still quite large and so the pendulum position is still changing rapidly, the plots above reflect this. In order to force the pendulum to settle at the same time as the cart, a different T_2 and T_3 must be chosen. By trial and error, it was found that an optimal T_3 value was such that $T_3 = T_1 = 266$ ms, and $T_2 = 328$ ms.

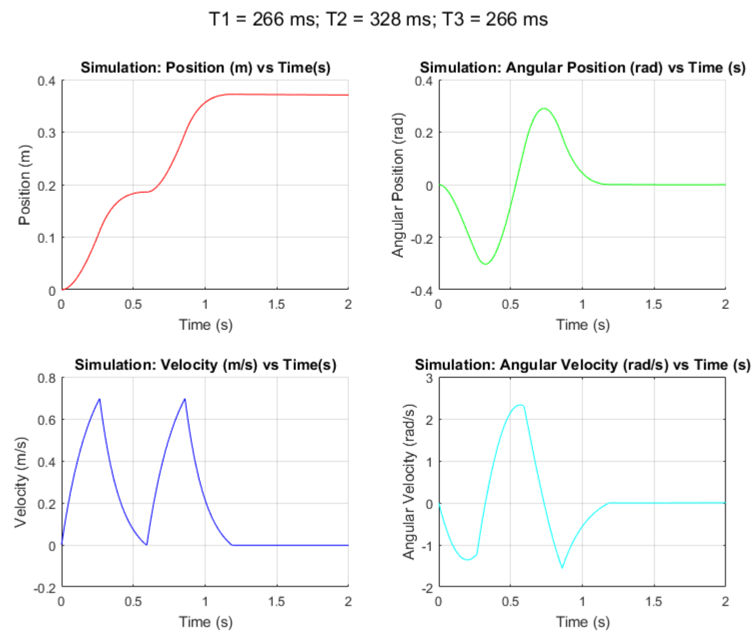

Figure 7: MATLAB simulation of two drive-brake cycles

The shape of the linear velocity and position plots of the above figure are similar to the first one, with the velocity decreasing to 0 m/s and the position settling to a constant value for $t > T_1 + T_2 + T_3$. However, the angular position and velocity plots are very different, with both now settling to approximately 0 rad and 0 rad/s, respectively, which was the initial aim of this experiment.

7 Discussion

There were many approximations that needed to be made in order to not over-complicate the system modelling process. With regards to physical approximations related to the system itself, the angular position of the pendulum was always assumed to start at exactly 0 rad from the vertical and from rest, but this may not always be the case as the pendulum may be already swinging with the cart starts moving. The pendulum rod that the mass is held by was assumed to have negligible mass and width and so its effects on the motion of the pendulum were ignored. For the motion equation of the cart, the cross-coupling effect of the pendulum was ignored, and although the effect was small, it may still contribute to the overall inaccuracy of the system. Also, friction was ignored in the pendulum pivot, and the friction force between the cart and the track was over-simplified and modelled as a Coulomb frictional force, which does not model dynamic systems effectively[2]. By ignoring the pendulum pivot friction, the simulated angular position, θ, did not decay over time as it would realistically, and this meant that there were inaccuracies in the pendulum position modelling. The implications of this was that it was impossible to choose a T_2 and T_3 such that the angular position and velocity went to exactly 0, and so a tolerance of approximately $\tau_\theta \approx \pm 0.002$ rad had to be settled on for the angular position and $\tau_{\dot{\theta}} \approx \pm 0.011$ rad/s for the angular velocity. In addition to the physical approximations mentioned, there were some theoretical approximations made for the T_2 and T_3 values required for the pendulum to come to rest at the same time as the cart and the braking system constants, A_b and τ_b, which were all found by trial-and-error and so are inherently inaccurate. The inaccuracy of all of these values was amplified as they were not only found through trial-and-error but the most suitable values were chosen by visual inspection of a plot, adding human error to the process.

For a completely theoretically correct model of the system, the friction would need to be modelled more accurately (and in the case of the pendulum, actually considered), as well as taking the mass of the cart, pendulum and pendulum rod into account. The cross-coupling effect of the pendulum on the cart would need to be considered. Additionally, the braking system would need to be mathematically defined rather than having it's characteristics found through trial-and-error. In order to achieve an exact 0 rad and 0 rad/s for the angular position and velocity of the pendulum, respectively, a more accurate simulation is required with T_2 and T_3 being derived by mathematical methods.

8 Conclusion

The cart and pendulum system was modelled with acceptable validity with the largest percentage error between simulated and experimental data being $\approx 2.03\%$. Times $T_2 = 328$ ms and $T_3 = 266$ ms were found such that the angular position and velocity of the pendulum came to rest within a reasonable tolerance band. The simulation of the model matched collected experimental data acceptably, however for an even more accurate fit there are more factors that could have been considered in the modelling process.

References

[1] Katupitiya, J. (2017). Laboratory Experiment I : Modelling of a Monorail Crane. [pdf] p.2. Available at: https://moodle.telt.unsw.edu.au/pluginfile.php/2728235/mod_folder/content/0/MonorailCraneExperiment.pdf
[Accessed Sep. 2017].

[2] Mogi.bme.hu. (2017). Chapter 8. Models of Friction: 8.3 - Coulomb Friction [online] Available at: http://www.mogi.bme.hu/TAMOP/robot_applications/ch07.html#ch-8.3
[Accessed Sep. 2017].

9 Appendix

9.1 Appendix A

```matlab
function dy = monorail_model(t,y)
   % y(1) - Cart position [x]
   % y(2) - Cart velocity [x_dot]
   % y(3) - Pendulum position [theta]
   % y(4) - Pendulum velocity [theta_dot]
   v_drive = 12;          % driving voltage
   v_brake = -1;          % braking voltage (taken as -1V)
   l = 340.6/1000;        % pendulum length (m)
   g = 9.81;              % gravity
   is_braking = false;    % true if the system is braking

   T1 = 266/1000;         % T1 from my data is 266 ms
   T2 = 328/1000;         % 328 ms was found to be optimal T2
   T3 = 266/1000;         % T3 = T1 was found to be optimal

   % braking constants
   Tau_b = 0.145265;
   A_b = 0.0835588;

   % driving constants
   Tau_d = 0.2079;
   A_d = 0.0805;

   dy = zeros(4,1);

   % drive the system in interval [0, T1]
   if (t <= T1)
       Tau = Tau_d;
       A = A_d;
       v = v_drive;
       is_braking = false;
   % stop driving and apply brakes in interval (T1, T2]
   elseif ((t > T1) && (t <= T1+T2))
       Tau = Tau_b;
       A = A_b;
       v = v_brake;
       is_braking = true;
   % drive the system in interval (T2, T3]
   elseif ((t > T1+T2) && (t <= T1+T2+T3))
       Tau = Tau_d;
       A = A_d;
       v = v_drive;
       is_braking = false;
   % stop driving and apply brakes in interval (T3, Inf}
   else
       Tau = Tau_b;
       A = A_b;
       v = v_brake;
       is_braking = true;
    end
   % velocity
   dy(1) = y(2);
   if ((y(2) < 0) && is_braking)
       dy(2) = 0;
   else
       % acceleration
       dy(2) = (1/Tau) * (A *v - y(2));
   end
   % angular velocity
   dy(3) = y(4);
   % angular acceleration
   % equation obtained from Newton-Euler formula for the pendulum
   dy(4) = -(g/l) * sin(y(3)) - (1/l) * cos(y(3)) * dy(2);
end
```

MATLAB code for modelling the cart and pendulum system

9.2 Appendix B

```matlab
options = odeset('RelTol', 1e-4, 'AbsTol', [1e-4 1e-4 1e-4 1e-4]);
% x(1) - Cart Position at time t (m)
% x(2) - Cart Velocity at time t (m/s)
% x(3) - Pendulum Angular Position at time t (rad)
% x(4) - Pendulum Angular Velocity at time t (rad/s)
[t, x] = ode45(@monorail_model, [0:0.002:2], [0 0 0 0], options);
```

MATLAB code for simulating the cart and pendulum system