Numerically simulating the dynamics of the pendulum and cart

1 Kinematics

Consider a cart of mass M, motorized and free to move along the x-axis upon a frictionless track. A pendulum bob of mass m is attached to the cart by a rigid, massless, freely swinging rod of length l. Let's start by writing down the kinematics—that is, the position, velocity, and acceleration—for each of the cart and bob.

For the cart,

$$\begin{aligned}
\text{position:} &\quad x\hat{\mathbf{x}} \\
\text{velocity:} &\quad x'\hat{\mathbf{x}} \\
\text{acceleration:} &\quad x''\hat{\mathbf{x}}.
\end{aligned}$$

And for the bob,

$$\begin{aligned}
\text{position:} &\quad (x + l\sin\theta)\hat{\mathbf{x}} + l\cos\theta\hat{\mathbf{y}} \\
\text{velocity:} &\quad (x' + l\theta'\cos\theta)\hat{\mathbf{x}} - l\theta'\sin\theta\hat{\mathbf{y}} \\
\text{acceleration:} &\quad (x'' + l\theta''\cos\theta - l\theta'^2\sin\theta)\hat{\mathbf{x}} - (l\theta''\sin\theta + l\theta'^2\cos\theta)\hat{\mathbf{y}}.
\end{aligned}$$

2 Forces

We can list the forces acting on each of the cart and bob as well (see Fig. 1).

The cart experiences

$$\begin{aligned}
\text{force due to gravity:} &\quad -Mg\hat{\mathbf{y}} \\
\text{rod tension:} &\quad T\sin\theta\hat{\mathbf{x}} + T\cos\theta\hat{\mathbf{y}} \\
\text{normal force on track:} &\quad N\hat{\mathbf{y}} \\
\text{motor force:} &\quad f\hat{\mathbf{x}}.
\end{aligned}$$

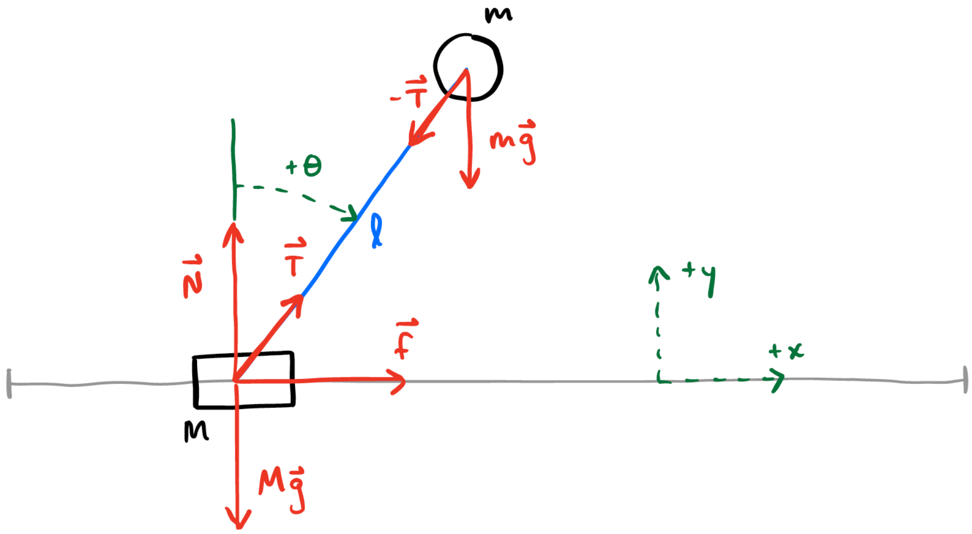

Figure 1: Free body diagram for our setting of the cart and pendulum.

And the bob experiences

$$\begin{aligned} \text{force due to gravity:} \quad & -mg\hat{\mathbf{y}} \\ \text{rod tension:} \quad & -T\sin\theta\hat{\mathbf{x}} - T\cos\theta\hat{\mathbf{y}}. \end{aligned}$$

3 Equations of motion

Using Newton's law $\mathbf{F}_{\mathbf{net}} = m\mathbf{x}''$ to relate forces with kinematics, we get the two expressions which follow.

For the cart,

$$T\sin\theta\hat{\mathbf{x}} + f\hat{\mathbf{x}} - Mg\hat{\mathbf{y}} + T\cos\theta\hat{\mathbf{y}} + N\hat{\mathbf{y}} = Mx''\hat{\mathbf{x}}. \tag{1}$$

And for the bob,

$$-T\sin\theta\hat{\mathbf{x}} - mg\hat{\mathbf{y}} - T\cos\theta\hat{\mathbf{y}} = mx''\hat{\mathbf{x}} + ml\theta''\cos\theta\hat{\mathbf{x}} - ml\theta'^2\sin\theta\hat{\mathbf{x}} - ml\theta''\sin\theta\hat{\mathbf{y}} - ml\theta'^2\cos\theta\hat{\mathbf{y}}. \tag{2}$$

Separating these two equations into vector components gives four equations.

$$\begin{aligned} \text{cart x:} \quad & T\sin\theta + f = Mx'' && (3) \\ \text{cart y:} \quad & -Mg + T\cos\theta + N = 0 && (4) \\ \text{bob x:} \quad & -T\sin\theta = mx'' + ml\theta''\cos\theta - ml\theta'^2\sin\theta && (5) \\ \text{box y:} \quad & -mg - T\cos\theta = -ml\theta''\sin\theta - ml\theta'^2\cos\theta && (6) \end{aligned}$$

Adding together equations (3) and (5) to eliminate T gives

$$f = (M + m)x'' + ml\theta'' \cos\theta - ml\theta'^2 \sin\theta. \tag{7}$$

Multiplying (5) through by $\cos\theta$ and multiplying (6) through by $-\sin\theta$ and then adding these together gives

$$mg \sin\theta = mx'' \cos\theta + ml\theta''. \tag{8}$$

Our two equations of motion are (7) and (8). However, these must be rewritten to isolate x'' and θ'' in each equation. This can be done algebraically or using *Mathematica*, yielding

$$x'' = \frac{f + ml\theta'^2 \sin\theta - mg \sin\theta \cos\theta}{M - m - m\cos^2\theta} \tag{9}$$

$$\theta'' = \frac{-f \cos\theta + (M + m)g \sin\theta - ml\theta'^2 \sin\theta \cos\theta}{l(M + m - m\cos^2\theta)}. \tag{10}$$

We can turn these two second-order ODEs into four first-order ODEs, yielding a form which will be suitable for numerical simulation via the Runge-Kutta method:

$$x' = v \tag{11}$$

$$\theta' = \omega \tag{12}$$

$$v' = \frac{f + ml\omega^2 \sin\theta - mg \sin\theta \cos\theta}{M + m - m\cos^2\theta} \tag{13}$$

$$\omega' = \frac{-f \cos\theta + (M + m)g \sin\theta - ml\omega^2 \sin\theta \cos\theta}{l(M + m - m\cos^2\theta)}. \tag{14}$$

These are the equations of motion we will use for numerical simulation.

4 Runge-Kutta update rule

Now we will work out the update rules for simulating via the Runge-Kutta method. We can represent the system with a 5-dimensional state vector

$$\mathbf{x} = (t, x, \theta, v, \omega) \in \mathbb{R}^5, \quad \text{where, equivalently,} \tag{15}$$

$$\begin{aligned} x_1 &= t \\ x_2 &= x \\ x_3 &= \theta \\ x_4 &= v \\ x_5 &= \omega \end{aligned} \tag{16}$$

So, our equations of motion become

$$\frac{d}{dt}x_1 = f_1(\mathbf{x}) = 1 \qquad \text{(passage of time; } dt = 1\text{)}$$

$$\frac{d}{dt}x_2 = f_2(\mathbf{x}) = x_4 \qquad \text{(change in cart position; } \tfrac{dx}{dt} = v\text{)}$$

$$\frac{d}{dt}x_3 = f_3(\mathbf{x}) = x_5 \qquad \text{(change in pendulum angle; } \tfrac{d\theta}{dt} = \omega = 1\text{)}$$

$$\frac{d}{dt}x_4 = f_4(\mathbf{x}) = \frac{f + mlx_5{}^2 \sin x_3 - mg \sin x_3 \cos x_3}{M + m - m\cos^2 x_3} \qquad \text{(change in cart velocity; } \tfrac{dv}{dt} = a\text{)}$$

$$\frac{d}{dt}x_5 = f_5(\mathbf{x}) = \frac{-f\cos x_3 + (M+m)g\sin x_3 - mlx_5{}^2 \sin x_3 cos x_3}{l(M + m - m\cos^2 x_3)}.$$

$$\text{(change in pendulum angular velocity; } \tfrac{d\omega}{dt} = \alpha\text{)}$$

This can be expressed even more generally as

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}) \tag{17}$$

where $\mathbf{f}$ is a loosely defined "vector of functions" $\mathbf{f} = (f_1, f_2, f_3, f_4, f_5)$, each accepting the system state $\mathbf{x}$ as an argument and returning the rate of change of a particular state variable.

The Runge-Kutta method, using a step of size h, numerically approximates four intermediate values of the state as it advances from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}) \tag{18}$$

$$\mathbf{k}_2 = \mathbf{f}(\mathbf{x} + \frac{h}{2}\mathbf{k}_1) \tag{19}$$

$$\mathbf{k}_3 = \mathbf{f}(\mathbf{x} + \frac{h}{2}\mathbf{k}_2) \tag{20}$$

$$\mathbf{k}_4 = \mathbf{f}(\mathbf{x} + h\mathbf{k}_3). \tag{21}$$

and then returns a weighted average as the result:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4). \tag{22}$$

This is the update rule for advancing the simulation by a step h.