Modelling the dynamics of a wheeled balancing robot

Peter Corke

September 2012
Revised 12 October 2012
Revised 23 October 2012
Revised Sept. 2014 & Sept. 2015 by Thierry Peynot

This document derives the equations of motion of a wheeled balancing robot — a Segway-like vehicle — using a Lagrangian, rather than Newton-Euler, approach.

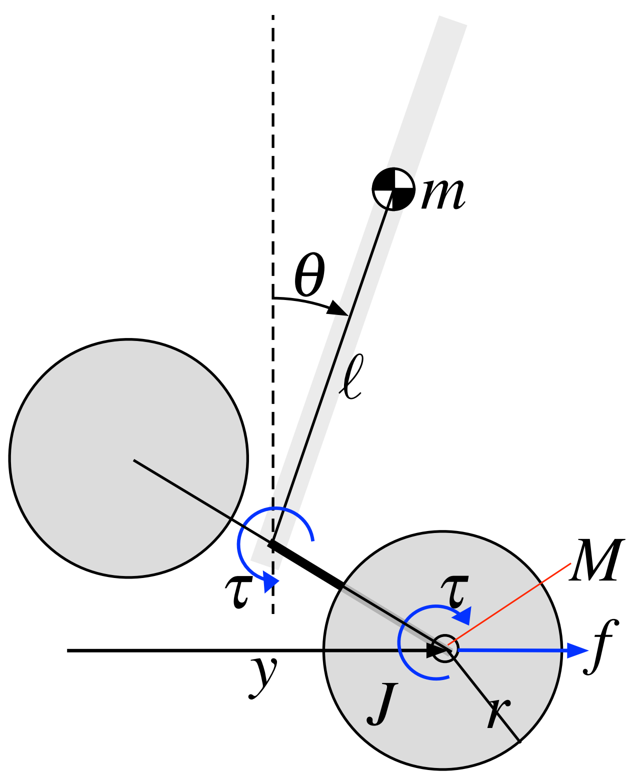

Figure 1: Notation and coordinate conventions for the wheeled balancing robot.

1 Introduction

The notation is shown in Figure 1. The body has a centre of mass m at a distance ℓ from the axle, and the body has an inclination of θ from vertical. The wheel has a mass of M and a rotational inertia of J and its centre is displaced a distance y from the origin. A motor exerts equal and opposite torques of τ on the wheel and the body. We assume that all motion is frictionless.

We choose the generalised coordinates to be $q_1 = \theta$ and $q_2 = y$. The corresponding generalised forces are the motor torque applied to the body $Q_1 = -\tau$ and $Q_2 = f$ which is the horizontal force applied to the centre of the wheel.

The first step in the Lagrangian approach is to write equations for the potential and kinetic energy of the system. The potential energy of the system is due only to the body

$$P = mg\ell\cos\theta \tag{1}$$

where g is the acceleration due to gravity, and as expected is zero when the body is horizontal $\theta = \pm\pi/2$.

The kinetic energy of the system is contributed by both the body and the wheel. Considering the body first, the centre of mass (COM), has two velocity components: $\dot{y}$ in the horizontal direction due to the motion of the axle, and $\ell\dot{\theta}$ normal to the body due to its rotation. The total horizontal velocity is

$$\ell\dot{\theta}\cos\theta + \dot{y} \tag{2}$$

and the total vertical velocity is

$$\ell\dot{\theta}\sin\theta \tag{3}$$

which yields a total squared velocity of

$$(\ell\dot{\theta}\cos\theta + \dot{y})^2 + (\ell\dot{\theta}\sin\theta)^2 \tag{4}$$

which we can simplify to

$$\dot{y}^2 + 2\ell\dot{y}\dot{\theta}\cos\theta + \ell^2\dot{\theta}^2 \tag{5}$$

Kinetic energy of a particle is $\frac{1}{2}mv^2$ so for the body component this is

$$\frac{1}{2}m\left(\dot{y}^2 + 2\ell\dot{y}\dot{\theta}\cos\theta + \ell^2\dot{\theta}^2\right) \tag{6}$$

The kinetic energy of the wheel components is due to both translational and rotational motion

$$\frac{1}{2}M\dot{y}^2 + \frac{1}{2}J\dot{\phi}^2 \tag{7}$$

where ϕ is the rotational velocity of the wheel. However we can relate wheel angle to displacement y since

$$y = r\phi \tag{8}$$

so we can substitute $\dot{\phi} = \dot{y}/r$ giving total kinetic energy

$$K = \frac{1}{2}m\left(\dot{y}^2 + 2\ell\dot{y}\dot{\theta}\cos\theta + \ell^2\dot{\theta}^2\right) + \frac{1}{2}M\dot{y}^2 + \frac{1}{2r^2}J\dot{y}^2 \tag{9}$$

$$= \frac{1}{2}m'\dot{y}^2 + \frac{1}{2}m\ell^2\dot{\theta}^2 + m\ell\dot{y}\dot{\theta}\cos\theta \tag{10}$$

where $m' = m + M + \frac{J}{r^2}$.
The next step is to form the Lagrangian

$$L = K - P \tag{11}$$

$$= \frac{1}{2}m'\dot{y}^2 + \frac{1}{2}m\ell^2\dot{\theta}^2 + m\ell\dot{y}\dot{\theta}\cos\theta - mg\ell\cos\theta \tag{12}$$

and to compute the equations of motion

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{13}$$

Considering first $q_1 = \theta$ we evaluate

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = -\tau \tag{14}$$

from which we can write

$$m\ell^2\ddot{\theta} + m\ell\ddot{y}\cos\theta - m\ell g\sin\theta = -\tau \tag{15}$$

Next we consider $q_2 = y$ and evaluate

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \frac{\partial L}{\partial y} = f \tag{16}$$

where the force $f = \tau/r$ is due to the wheel torque and an assumption of no wheel slip so we can write

$$m'\ddot{y} + m\ell\ddot{\theta}\cos\theta - m\ell\dot{\theta}^2\sin\theta = \tau/r \tag{17}$$

We now have a set of coupled second-order differential equations (15) and (17) that describe the motion of this mechanism. In a more compact matrix form this can be written

$$\begin{pmatrix} m\ell^2 & m\ell\cos\theta \\ m\ell\cos\theta & m' \end{pmatrix}\begin{pmatrix} \ddot{\theta} \\ \ddot{y} \end{pmatrix} = ml\sin\theta\begin{pmatrix} g \\ \dot{\theta}^2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1/r \end{pmatrix}\tau \tag{18}$$

or more compactly again as

$$\mathbf{M}(\mathbf{x})\dot{\mathbf{x}} = \mathbf{b}(\mathbf{x}, \tau) \tag{19}$$

where $\mathbf{M}$ is a 2×2 matrix, often referred to as the mass matrix, and is a function of the state of the system $\mathbf{x}$. $\mathbf{b}(\cdot)$ is the generalised force which is a function of state and the input τ to the plant. We can solve for the motion of the system by integrating

$$\dot{\mathbf{x}} = \mathbf{M}(\mathbf{x})^{-1}\mathbf{b}(\mathbf{x}, \tau) \tag{20}$$

2 Linearization

We linearize the mass matrix by assuming $\cos\theta \approx 1$

$$\bar{\mathbf{M}} = \begin{pmatrix} m\ell^2 & m\ell \\ m\ell & m' \end{pmatrix} \tag{21}$$

The function $\mathbf{b}(\cdot)$ is linearized by assuming the small angle approximation $\sin\theta \approx \theta$ and that $\dot{\theta}^2 \approx 0$ giving

$$ml\theta \begin{pmatrix} g \\ 0 \end{pmatrix} = \begin{pmatrix} ml\theta g \\ 0 \end{pmatrix} \tag{22}$$

Then the linear form is

$$\begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \alpha_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \alpha_2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \\ y \\ \dot{y} \end{pmatrix} + \begin{pmatrix} 0 \\ \beta_1 \\ 0 \\ \beta_2 \end{pmatrix} \tau \tag{23}$$

$$\tag{24}$$

where the coefficients α_i and β_i are given by

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \bar{\mathbf{M}}^{-1} \begin{pmatrix} m\ell g \\ 0 \end{pmatrix} \tag{25}$$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \bar{\mathbf{M}}^{-1} \begin{pmatrix} -1 \\ 1/r \end{pmatrix} \tag{26}$$

3 Linear control design

Consider the parameter values: $M = 0$, $m = 1$, $L = 1$, $r = 0.1$, $J = 2$, $g = 9.81$. These are defined by the script `segway.m` which uses a structure to hold the parameters.

```
>> segway

params =

       M: 0
       m: 1
       L: 1
       r: 0.100000000000000
       J: 2
       g: 9.810000000000000
```

The linearization can be achieved by the script `linearize.m` which takes a parameter structure

```
>> [A,B] = linearize(params)

alpha =

    9.8590
   -0.0491

A =
         0    1.0000         0         0
    9.8590         0         0         0
         0         0         0    1.0000
   -0.0491         0         0         0


B =
         0
   -1.0055
         0
    0.0550
```

and returns the A and B matrix.

We define the output of the system to be the states x_1 and x_3 which are the angle and displacement respectively, and create a linear time-invariant state-space model

```
>> C = [1 0 0 0; 0 0 1 0];
>> s=ss(A,B, C, 0);
>> eig(s)
ans =
         0
         0
    3.1399
   -3.1399
```

which has three problematic poles, two at the origin and one in the right-half plane.

We design an LQR regular with weighting matrices with a high penalty on errors in θ and y. However we choose a lower penalty on $\dot{\theta}$ and $\dot{y}$ since velocity is required in order to minimise the position errors, penalising both is contradictory.

```
>> K=lqr(s, diag([10 1 100 1]), 1)
K =
  -26.1440    -8.4435   -10.0000   -26.9007
```

We create a closed loop system

```
>> sc=ss(A-B*K, B, [1 0 0 0; 0 0 1 0], 0)
>> sc.OutputName = {'\theta', 'y'}
>> eig(sc)
ans =
  -3.2857
```

```
  -3.1668
  -0.4879 + 0.4831i
  -0.4879 - 0.4831i
```

which has stable poles. We simulate the process from an initial state which has $\theta_0 = 0.1$ but stationary

```
>> x0 = [0.1 0 0 0]';
>> t=[0:0.005:20]';  % simulation time vector
>> lsim(sc, 0*t, t, x0);
```

and this demonstrates a satisfactory response as shown in Figure 2. Note that for the simulation we set the input to the system to zero `0*t` since the effect of the control is already included in the closed-loop system `sc`. The body rapidly moves to the upright configuration but it has to move forward in order to achieve this, and it then moves slowly back to its initial position (this motion is described by the conjugate pair of very slow poles close to the origin).

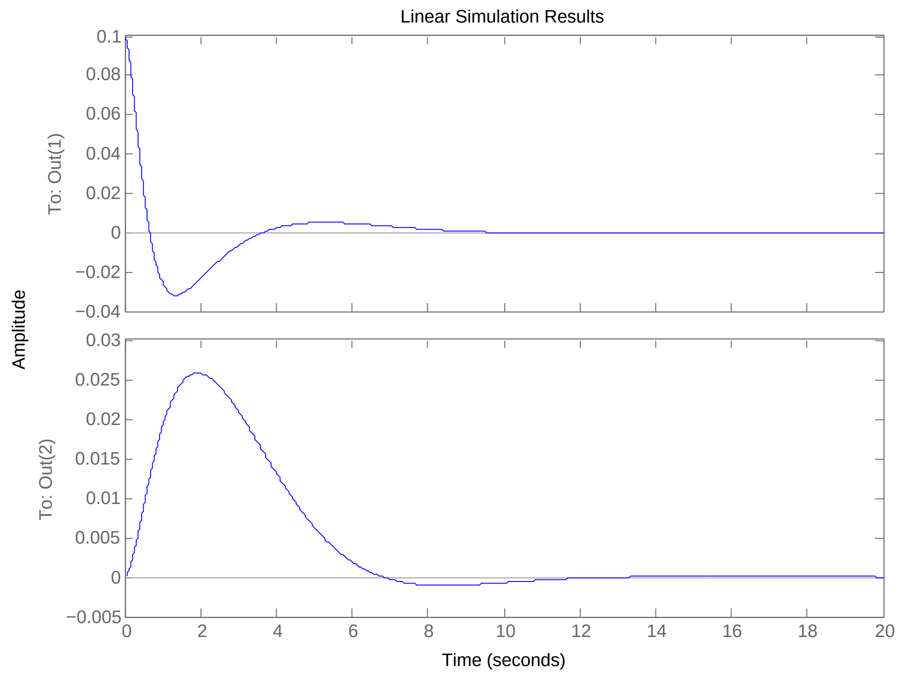

Figure 2: Closed-loop step response.

4 Implementation on the NXT

This system is torque actuated, that is, it responds to motor acceleration not velocity. The NXT motor is quite highly geared and thus has significant static and viscous friction and is voltage, or speed controlled. An optional speed-control loop can be enabled via RobotC. To eliminate the effect of motor back EMF an appropriate approach to control may be the following.

1. Compute τ using the state feedback controller above. This requires knowledge of the state which can be estimated from gyro and encoder sensors.

2. Compute the required acceleration τ/J_b where J_b is the total inertia of the system, body and wheels.

3. Integrate the required acceleration at each time step to obtain the required velocity and send this to the NXT motor in closed-loop speed control mode.