

Balancing Robot

Izzy Mones and Heidi Dixon

April 15, 2025

Robot Design

Model

System Description (Lagrange's Method)

Derive the equations here

$$\dot{x} = v \tag{1}$$

$$\ddot{x} = \frac{-m^2 L^2 g \cos(\theta) \sin(\theta) + mL^2 (mL\omega^2 \sin(\theta) - \delta v) + mL^2 u}{mL^2 (M + m(1 - \cos(\theta)^2))} \tag{2}$$

$$\dot{\theta} = \omega \tag{3}$$

$$\dot{\omega} = \frac{(m + M)mgL \sin(\theta) - mL \cos(\theta)(mL\omega^2 \sin(\theta) - \delta v) + mL \cos(\theta)u}{mL^2 (M + m(1 - \cos(\theta)^2))} \tag{4}$$

where x is the cart position, v is the velocity, θ is the pendulum angle, ω is the angular velocity, m is the pendulum mass, M is the cart mass, L is the pendulum arm length, g is the gravitational acceleration, δ is a friction damping on the cart, and u is the control force applied to the cart.

Linearization

To build a control system for our model we will linearize our system of equations around a fixed point x_r where x_r is the position where the robot is vertical, unmoving and positioned at the origin.

The nonlinear system of differential equations

$$\frac{d}{dt} \mathbf{x} = \mathbf{f}(\mathbf{x}). \tag{5}$$

can be represented as a Taylor series expansion around the point \mathbf{x}_r .

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_r) + \left. \frac{d\mathbf{f}}{d\mathbf{x}} \right|_{\mathbf{x}_r} (\mathbf{x} - \mathbf{x}_r) + \left. \frac{d^2 \mathbf{f}}{d\mathbf{x}^2} \right|_{\mathbf{x}_r} (\mathbf{x} - \mathbf{x}_r)^2 + \dots \tag{6}$$

Because \mathbf{x}_r is a fixed point, we know that $\mathbf{f}(\mathbf{x}_r) = 0$. Additionally, we assume the value of $(\mathbf{x} - \mathbf{x}_r)$ is small, so higher order terms of this series will go to zero. So a fair estimate of our system is

$$\frac{d}{dt} \mathbf{x} \simeq \left. \frac{d\mathbf{f}}{d\mathbf{x}} \right|_{\mathbf{x}_r} (\mathbf{x} - \mathbf{x}_r) \tag{7}$$

where $\left. \frac{d\mathbf{f}}{d\mathbf{x}} \right|_{\mathbf{x}_r}$ is the Jacobian matrix for our system of equations $\mathbf{f}(\mathbf{x})$.

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{d}{M} & \frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{d}{ML} & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} \quad (8)$$

LQR

LQG

I'm just testing out how to include code in our document. This is how you do inline code text. `class MyClass`
And next is a full code snippet.

```
def loop_iteration():
    global u
    global x

    # estimate state
    dx = (A@(x - xr) + (B*u).transpose() + Kf@(y - C@(x-xr)))[0]
    x = x + dx*dT

    # compute the control value u, and update motor duty cycle
    u = -K@(x - xr)
    motors.run(u * duty_coeff)
```

Experiments

Conclusions

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (9)$$