Observation Report Balance bot

Introduction

Model-

We model the balanced bot based on Inverted Cart pendulum model. used Euler–Lagrange equation to derive the dynamical equations.

Control-

We used -

- PID control
- LQR control

1 Modelling

We started the modelling with a less accurate rather very simple approach. neglected the moment of inertia of wheels and the coefficient of viscous friction (c_α) on the wheel axis.

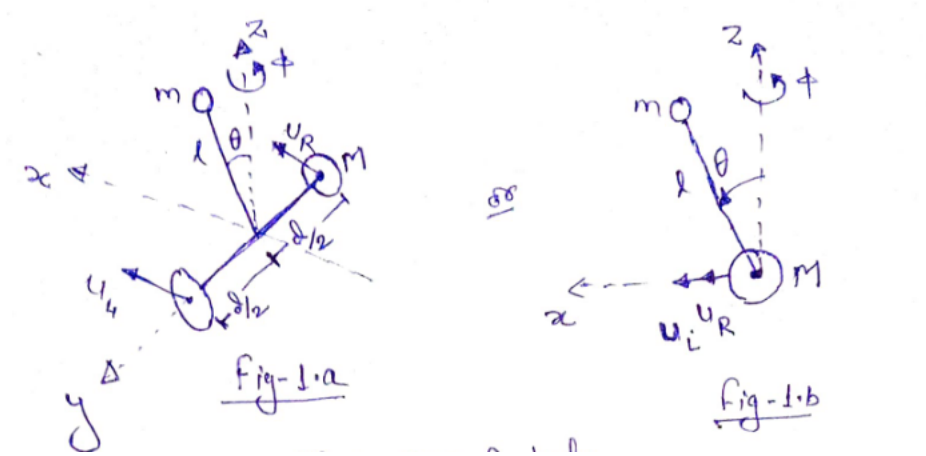

Fig 1 - 2-wheel-inverted-cart-pendulum

Here -
m = 10Kg (mass of body)

l = 0.4m (height of the body from axis of wheel)
M = 2Kg (mass of both the wheels ($\frac{M}{2}$ for each wheel))
d = 0.3m (width between the wheels)
U_l, U_r - torques on the respective wheels

State variables -

$$Y = \begin{bmatrix} X \\ \theta \\ \phi \\ \dot{X} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}$$

Here -
X - X position
θ - Pitch angle
ϕ - Yaw angle

Let (x_m, y_m) be the position of body (mass m) -

$$y_m = lcos(\theta), \dot{y}_m = -lsin(\theta)\dot{\theta}$$

$$x_m = x + lsin(\theta), \dot{x}_m = \dot{x} + lcos(\theta)\dot{\theta}$$

Kinetic energy -

$$K = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}_m^2 + \dot{x}_m^2) + \frac{1}{2}I\dot{\phi}^2$$

$$K = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}m(l^2\dot{\theta}^2 + 2\dot{x}lcos(\theta)\dot{\theta}) + \frac{1}{2}\left(\frac{Md^2}{4} + ml^2sin^2(\theta)\right)\dot{\phi}^2$$

Potential energy -

$$V = mglcos(\theta)$$

Using Euler–Lagrange equation -

$$L = K - V$$

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}_i}\right) - \frac{\partial L}{\partial x_i} = U_i$$

We got -

$$(M+m)\ddot{x} + mlcos(\theta)\ddot{\theta} - mlsin(\theta)\dot{\theta}^2 = U_r + U_l \tag{1}$$

$$cos(\theta)\ddot{x} + l\ddot{\theta} - lcos(\theta)sin(\theta)\dot{\phi}^2 - gsin(\theta) = 0 \tag{2}$$

$$\left(\frac{Md^2}{4} + ml^2sin^2(\theta)\right)\ddot{\phi} = \frac{d}{2}U_r - \frac{d}{2}U_l \tag{3}$$

Detailed derivation can found - Here
Now, we need to solve these equations in order to find out the values of $\ddot{x}, \ddot{\theta}, \ddot{\phi}$. we can see it is a very tedious to calculate it manually. so to solve it, se represent all the equations into a single matrix form equation. and then we can easily solve for $\ddot{x}, \ddot{\theta}, \ddot{\phi}$
Note that, this method will have more importance when there are more variables and equation, as solving them manually we be much more difficult.

So, we use to represent equation (1) (2) and (3) as -

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + G(q) = B\tau \tag{4}$$

Here -
M - denotes Mass matrix
C - denotes Coriolis matrix
D - denotes damping matrix
G - denotes Gravity vector
B - denotes Input matrix
τ - denotes Input vector
q - Coordinates

While transforming equations (1), (2, (3) into equation (4), we find the values of these matrices.

$$M(q) = \begin{bmatrix} (M+m) & mlcos(\theta) & 0 \\ cos(\theta) & l & 0 \\ 0 & 0 & \left(\frac{Md^2}{4} + ml^2sin^2(\theta)\right) \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & -mlsin(\theta)\dot{\theta} & 0 \\ 0 & 0 & -lcos(\theta)sin(\theta)\dot{\phi} \\ 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ -gsin(\theta) \\ 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ \frac{d}{2} & -\frac{d}{2} \end{bmatrix}, q = \begin{bmatrix} x \\ \theta \\ \phi \end{bmatrix}, \tau = \begin{bmatrix} U_r \\ U_l \end{bmatrix}$$

We can find $\ddot{q}$ by-

$$\ddot{q} = M(q)^{-1}(-C(q,\dot{q})\dot{q} - D\dot{q} - G(q) + B\tau)$$

$$Y = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T => Y = \begin{bmatrix} X \\ \theta \\ \phi \\ \dot{X} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}$$

$$\dot{Y} = \begin{bmatrix} \dot{q}^T & \ddot{q}^T \end{bmatrix}^T => \dot{Y} = \begin{bmatrix} \dot{X} \\ \dot{\theta} \\ \dot{\phi} \\ \ddot{X} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix}$$

State space model -

$$\dot{Y} = AY + BU$$

For state space modelling, we linearize the system at its, equilibrium

$$Y := (0, 0, 0, 0, 0, 0).$$

Using Jacobian, we find-

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{gm}{M} & 0 & 0 & 0 & 0 \\ 0 & \frac{g}{l}\left(1 + \frac{m}{M}\right) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{M} & \frac{1}{M} \\ -\frac{1}{Ml} & -\frac{1}{Ml} \\ \frac{2}{Md} & -\frac{2}{Md} \end{bmatrix}$$

Although this model is not very much accurate, but it gives satisfying result. how ever we also used an much accurate model referred to -
Dynamic Modeling of a Two-wheeled Inverted Pendulum Balancing Mobile Robot. by - Sangtae Kim and SangJoo Kwon

2 Control

We use PID as well as LQR controller to control the robot in Gazebo simulator environment.

2.1 PID controller

We tuned the robot to get PID values as -

$$Kpid = \begin{bmatrix} 25 & 1 & 500 \end{bmatrix}$$

The Output equation - (error is pitch error. i.e., θ)

$$output = Kpid[0] * error + Kpid[1] * sum_error + Kpid[2] * diff_error$$

And the output is feed into the wheels to perform the motion.

2.2 LQR controller

We tuned the robot to get Q, R values as -

$$Q = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Based on these values we calculated the gain matrix K and then feed the value of output to the wheels

$$output = K.errorMatrix$$

Here errorMatrix is the matrix of difference of the state variables form their setpoints.

3 Result

3.1 PID controller

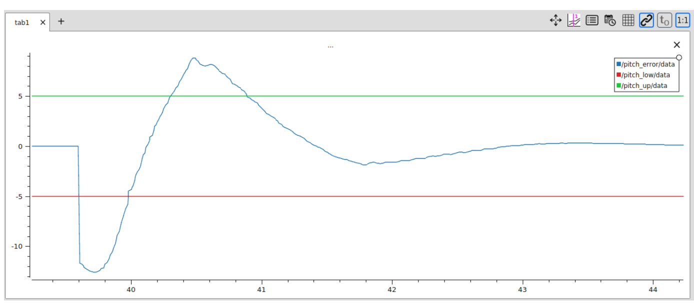

Fig 2 - PID response when about perturbation 15° is given

3.2 LQR controller

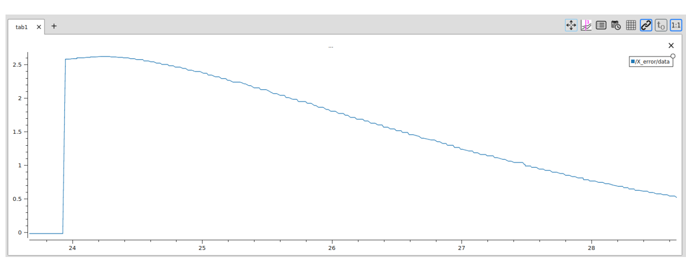

Fig 3 - LQR response when about perturbation 2.5m is given in x

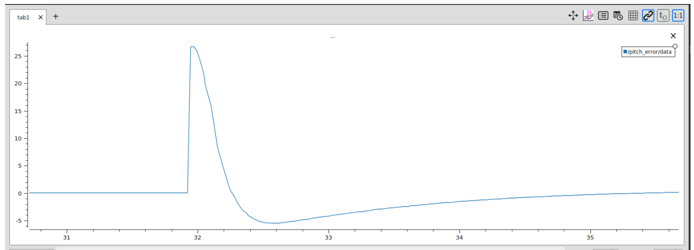

Fig 4 - LQR response when about perturbation 30° is given in Pitch

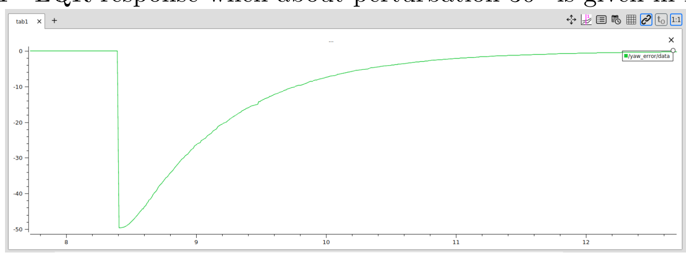

Fig 5 - PID response when about perturbation 50° is given in yaw

Simulation - click here

Source code - click here

References

[1] Modeling and control of two-legged wheeled robot, *by Adam Kollarčík January 2021*

[2] Dynamic Modeling of a Two-wheeled Inverted Pendulum Balancing Mobile Robot, *International Journal of Control, Automation, and Systems (2015)*

[3] Ascento: A Two-Wheeled Jumping Robot, *IEEE, ICRA 2019*

[4] LQR-Assisted Whole-Body Control of a Wheeled Bipedal Robot with Kinematic Loops, *IEEE ROBOTICS AND AUTOMATION LETTERS. PREPRINT VERSION. ACCEPTED JANUARY, 2020*

[5] Modelling and Control of a Hybrid Wheeled Jumping Robot, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*

[6] Hardware and control design of a ball balancing robot, *2019 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems DDECS*

[7] Feedback Control of Dynamic Systems, *by Gene F. Franklin — J. David Powell — Abbas Emami-Naeini*

[8] MODERN ROBOTICS MECHANICS, PLANNING, AND CONTROL, *by Kevin M. Lynch and Frank C. Park1*