PID Gain Estimation for a Self-Balancing Robot

1. Introduction

The self-balancing robot behaves like an inverted pendulum mounted on two wheels. The primary control objective is to stabilize the robot's tilt angle (θ) by applying corrective torque via motor-driven wheels. A Proportional–Integral–Derivative (PID) controller is employed to stabilize this unstable system using real-time feedback from sensors such as an IMU.

1. Physical Modeling of the Robot

Using classical mechanics, the robot is modeled as a rigid body rotating about the wheel axle. The linearized equation of motion for small angles θ is given by:

$$I\ddot{\theta} + Mgl\theta = \tau$$

Where:

- I: Moment of inertia about the wheel axle
- $M = 1.8\,\mathrm{kg}$: Mass of the robot
- $l = 0.1\,\mathrm{m}$: Distance from axle to center of mass
- $g = 9.81\,\mathrm{m/s}^2$: Gravitational acceleration
- τ: Torque applied by the motors

For a rectangular body pivoting about its base, we approximate the moment of inertia as:

$$I = \frac{1}{3}Ml^2$$

Substituting the known values:

$$I = \frac{1}{3} \times 1.8 \times (0.1)^2 = \frac{1.8 \times 0.01}{3} = 0.006\,\mathrm{kg} \cdot \mathrm{m}^2$$

Using this value, the system becomes:

$$\ddot{\theta} + \frac{Mgl}{I}\theta = \frac{\tau}{I} \Rightarrow \ddot{\theta} + 294.3\theta = \frac{\tau}{0.006}$$

2. Transfer Function

Taking the Laplace transform of the equation, we get the plant transfer function:

$$G(s) = \frac{\Theta(s)}{(s)} = \frac{1}{0.006s^2 + 1.7658}$$

This represents a second-order system with the tilt angle θ as output and the torque τ as input.

4. Simulink Integration

The system is simulated in Simulink using the model `Self_Balancing_Robot.slx`. The simulation setup includes:

- A plant model defined by the derived dynamics
- A PID controller block
- Feedback loop using simulated IMU data to measure θ

5. Estimating PID Gains

Step 1: Analytical Estimation

Assuming a desired closed-loop performance with:

- Natural frequency: $\omega_n = 15$
- Damping ratio: $\zeta = 0.7$

The desired characteristic polynomial is:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 21s + 225$$

Comparing this with the closed-loop system and matching coefficients, we estimate the PID gains:

$$K_p = 2\zeta\omega_n I = 2 \times 0.7 \times 15 \times 0.6 = 7.65$$
$$K_i = \omega_n^2 I = 225 \times 0.0006 = 0.337$$
$$K_d = 0.14$$

Step 2: Simulation-Based Refinement

Using the Simulink model:

1. Begin with estimated gains: $K_p = 7.65$, $K_i = 0.337$, $K_d = 0.14$
2. Run the simulation and observe:
   - Settling time
   - Overshoot
   - Steady-state error
3. Tune parameters iteratively:
   - Increase K_p to speed up response
   - Increase K_d to reduce overshoot
   - Increase K_i to eliminate steady-state error

Example trial:

$$K_p = 8.0, \quad K_i = 0.3, \quad K_d = 0.05$$

Final gains are chosen based on simulation results that achieve fast settling and minimal overshoot.

6. Conclusion

A combined approach using physics-based modeling and simulation allows for efficient and practical tuning of PID gains. Analytical estimates provide a reliable starting point, while simulation helps to fine-tune the controller for real-world behavior. This method ensures the self-balancing robot remains stable and responsive.