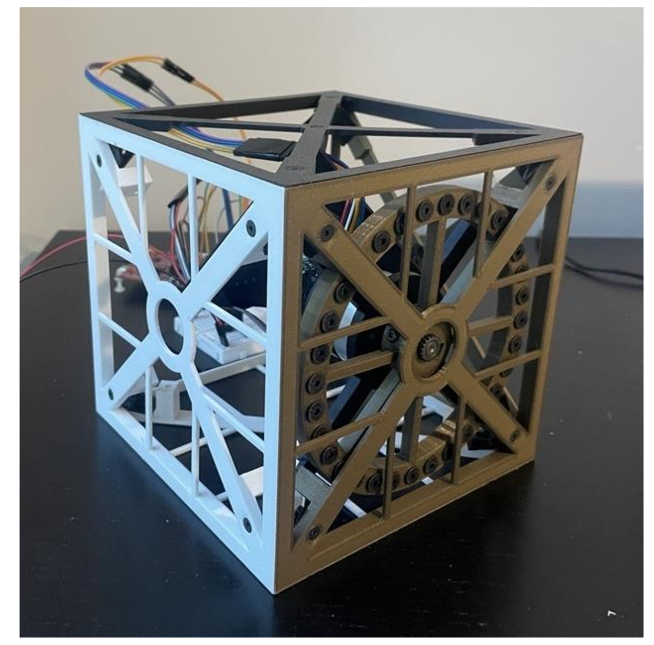
Reaction Wheel Balanced Cube

AE 6705 Final Project
Yunqing (JJ) Jia

Background

Inspired by Cubli, a 15x15x15 cm 3D cube that can jump up and balance on a corner [1], I decided to build a reaction wheel balanced cube. This is similar to the classic inverted pendulum problem, where active controls are required to maintain balance, but utilizes conservation of angular momentum.

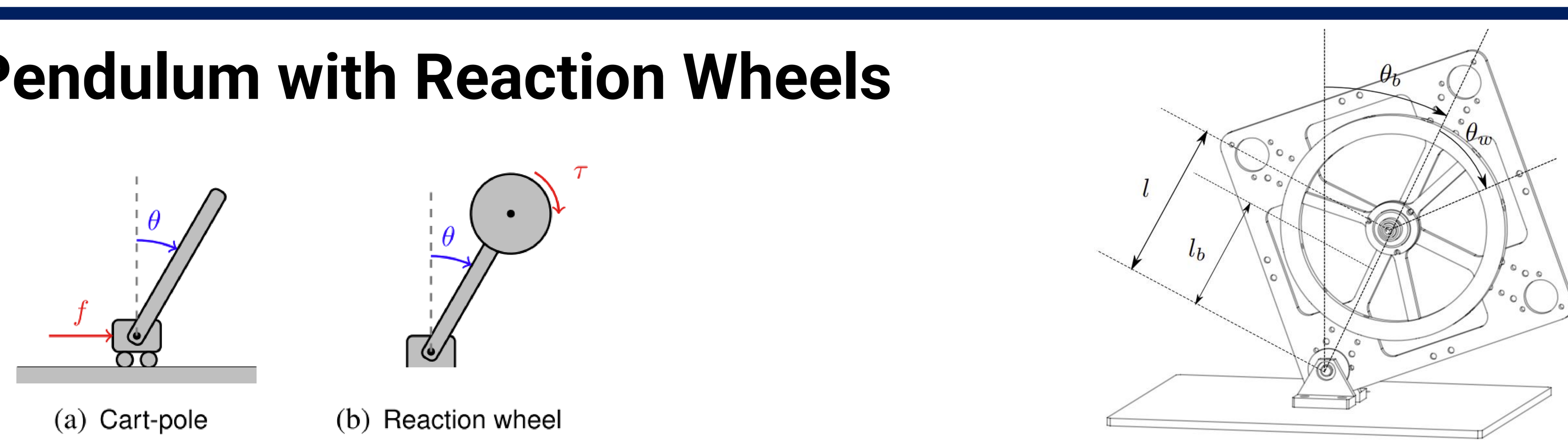
[1] M. Gajamohan, M. Merz, I. Thommen and R. D'Andrea, "The Cubli: A cube that can jump up and balance," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, 2012, pp. 3722-3727, doi: 10.1109/IROS.2012.6385896.



Performance Objectives

The goal is to balance the cube on one of its edges for **at least 10 seconds** before reaching the reaction wheel's saturation limit.

Inverted Pendulum with Reaction Wheels



(a) Cart-pole (b) Reaction wheel

There are multiple papers on how to solve this problem [1] [2]. Reaction wheels are commonly used to stabilize satellites. The motor attached to the center of the reaction wheel generates angular momentum and a reacting force on the rest of the system in the opposite direction ($\boldsymbol{I_c\, \omega_c = I_w\, \omega_w}$). By speeding up, slowing down, or reversing the direction of the wheel, the system can adjust its orientation and maintain balance.

[2] F. Bobrow, B. A. Angelico, F. P. R. Martins and P. S. P. da Silva, "The Cubli: Modeling and Nonlinear Attitude Control Utilizing Quaternions," in IEEE Access, vol. 9, pp. 122425-122442, 2021, doi: 10.1109/ACCESS.2021.3108426.

Inertial Measurement Unit (IMU) for Attitude Sensing

A **MPU6050** IMU is used to estimate the cube's attitude. The IMU communicates with the microcontroller via I^2C and features 16-bit registers for acceleration and gyro readings along three axes. The digital low pass filter was enabled on the sensor hardware so the sensor has a bandwidth of **256 Hz** according to the spec sheet.

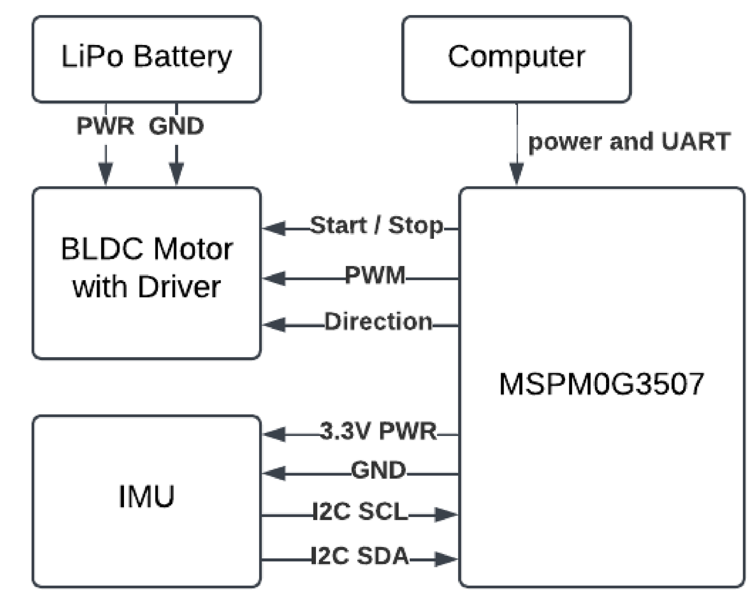
Alpha smoothing is applied to reduce high-frequency noise in accelerometer data:
$accel_k = \alpha \cdot accel_{k-1} + (1-\alpha) \cdot accel_k$

The gravitational force (1g) is projected onto multiple axes depending on the IMU orientation, and the tilt angle (θ) can be calculated using trigonometry.

A **complementary filter** is used to fuse together gyroscope and accelerometer data. The gyroscope provides short-term accuracy in angular velocity measurement but drifts over time due to random walk, while accelerometer is stable in the long-term:
$angle = \beta \cdot (angle + \{gyro\ reading\} \cdot dt) + (1-\beta) \cdot \{accel\ angle\}$

Wiring Diagram and Parts List

1 pin linked to TimerA0 was used to generate PWM signal, and 2 GPIO pins are used to start/stop and control the direction of the motor.

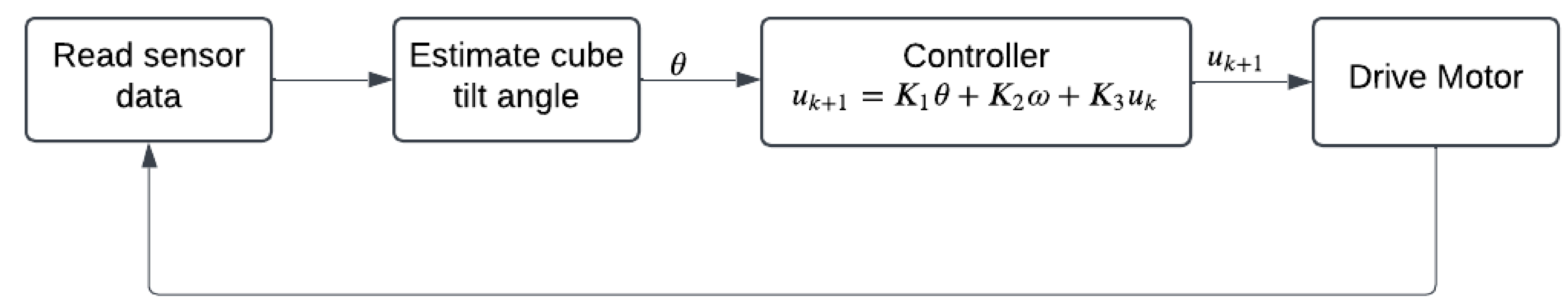


- 1x Nidec BLDC Motor (12V)
- 1x MPU6050 IMU
- 1x 11.1V LiPo battery
- 1x MSPM0G3507

Specs

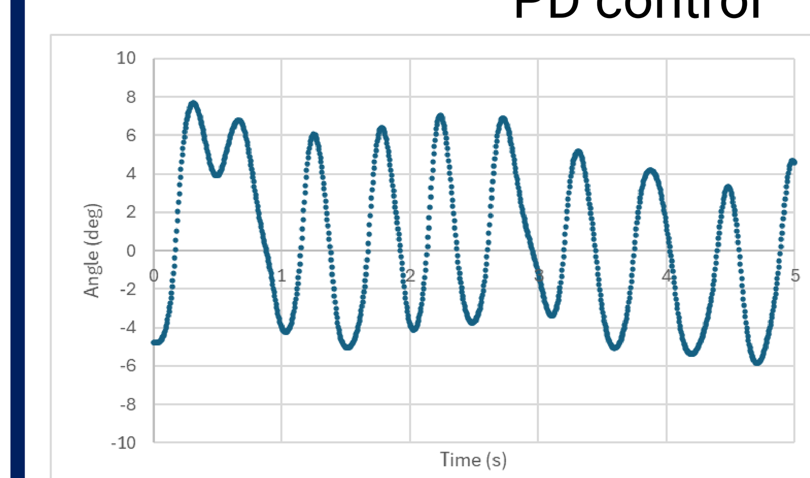
- The IMU has **16-bit measurements**. With smallest full scale range: **16384 LSB/g for accelerometer and 131 LSB/deg/s for gyro.**
- For this application, since we are controlling the system through current / torque control, the motor needs to have a high stall torque and a high maximum RPM. This motor has a stall torque of **70 mN-m** and a max of **5800 RPM.**

Feedback Control

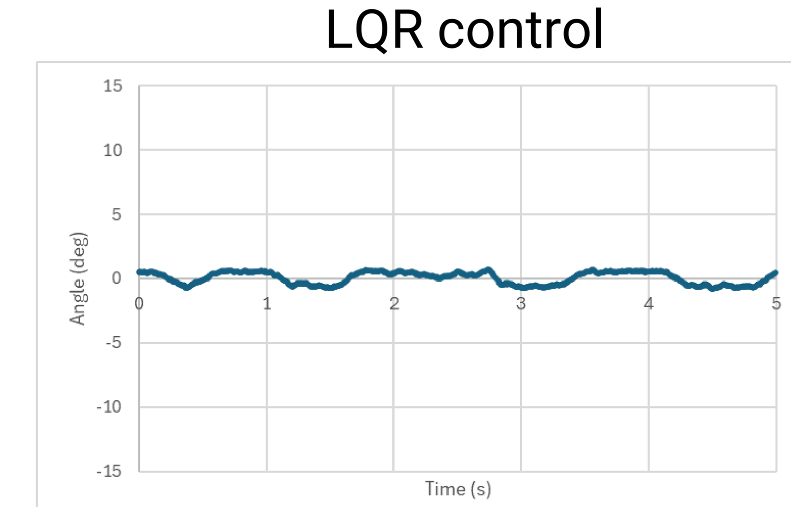


Results

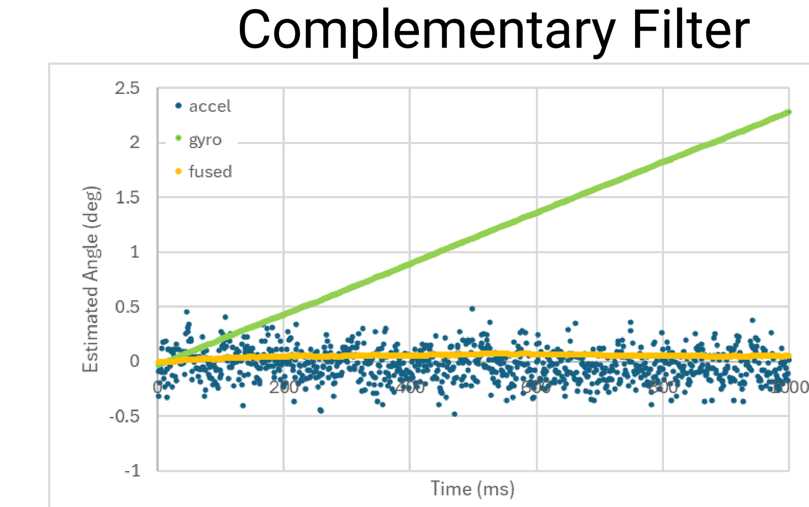
PD control



LQR control



Complementary Filter



There was a lot of oscillation when I tried to control the motor with just a PD controller. When the angular error is close to zero, the motor velocity is also reduced to close to zero. Since I want to control angular acceleration, I added in the current speed (in terms of duty cycle) to the control calculation and used LQR control instead and was able to achieve stability.

Lessons Learned and Potential Enhancements

My initial goal was to be able to balance the cube on a corner as a 3D inverted pendulum. I eventually realized that it was a bigger undertaking than I had anticipated, so I pivoted to doing just 1D balancing. In the future, I hope to implement the 3D version.

It took a long time to get the correct angle estimation because I was calculating the complementary filter wrong. I learned to print out intermediate calculations and that was helpful in subsequent code development and gain tuning as well. I also tried to estimate the angle through quaternions at one point but was unsuccessful. Might be worth going back and revisiting that for trying 3D.

The motor loses charge over time so the RPM is not always the same for the same PWM duty cycle. An improvement could be either using a voltage regular or an encoder or hall effect sensor to either make sure that the voltage source is constant and / or the RPM estimation is accurate.