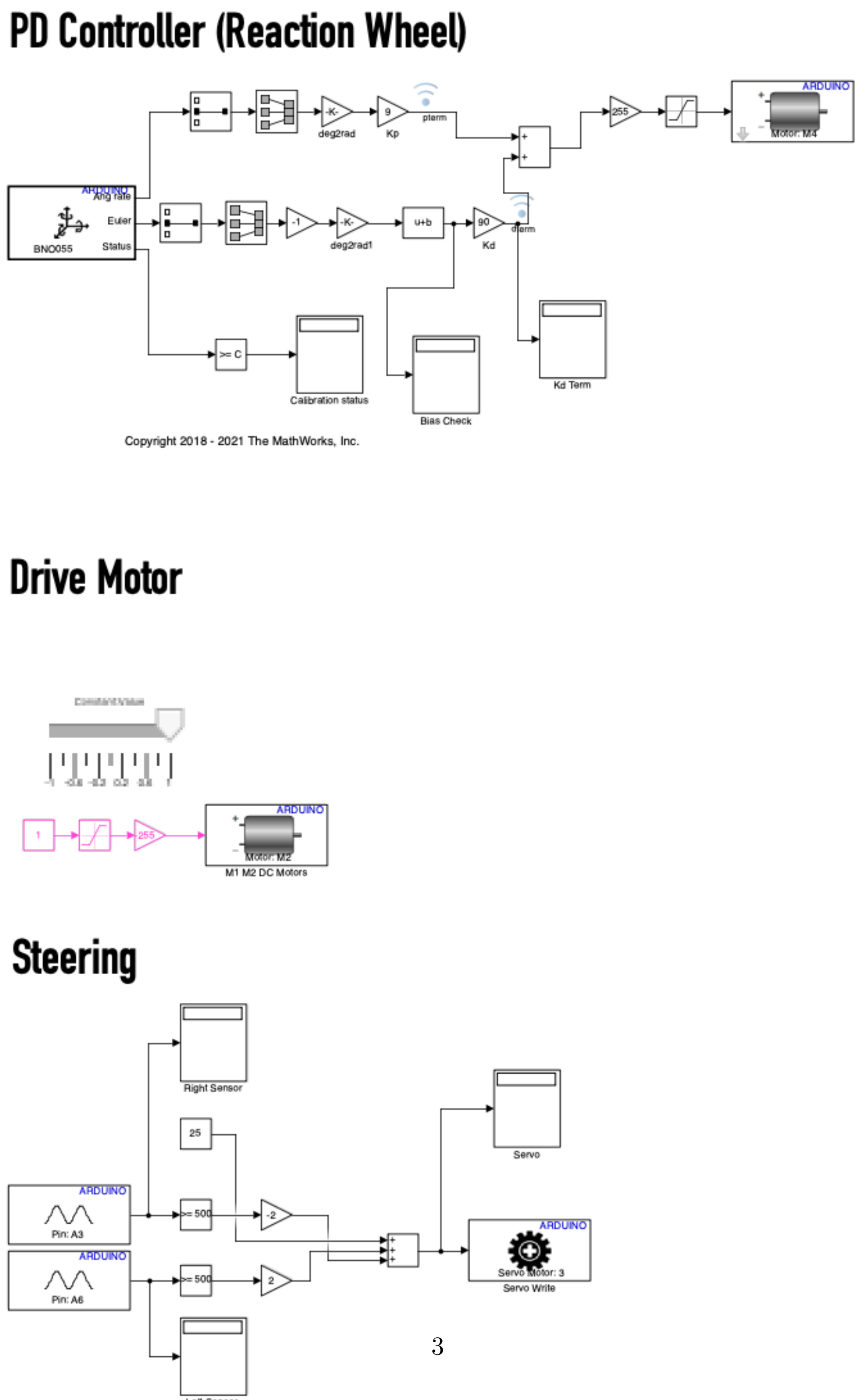
MENG 390 Final Report

Esteban Figueroa (Lab Partner: Maria Korolik)

May 2024

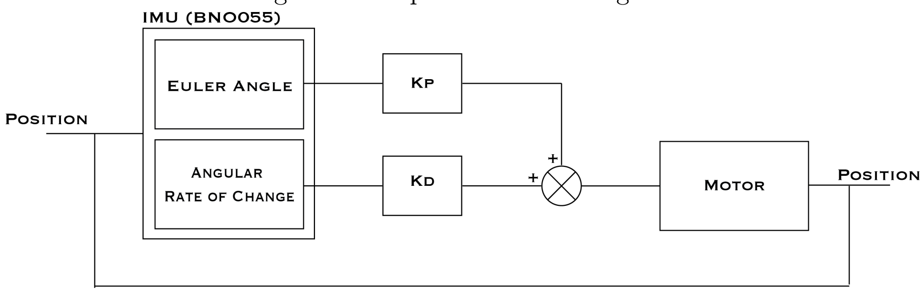
1 Block Diagram

Figure 1: Simulink Block Diagram



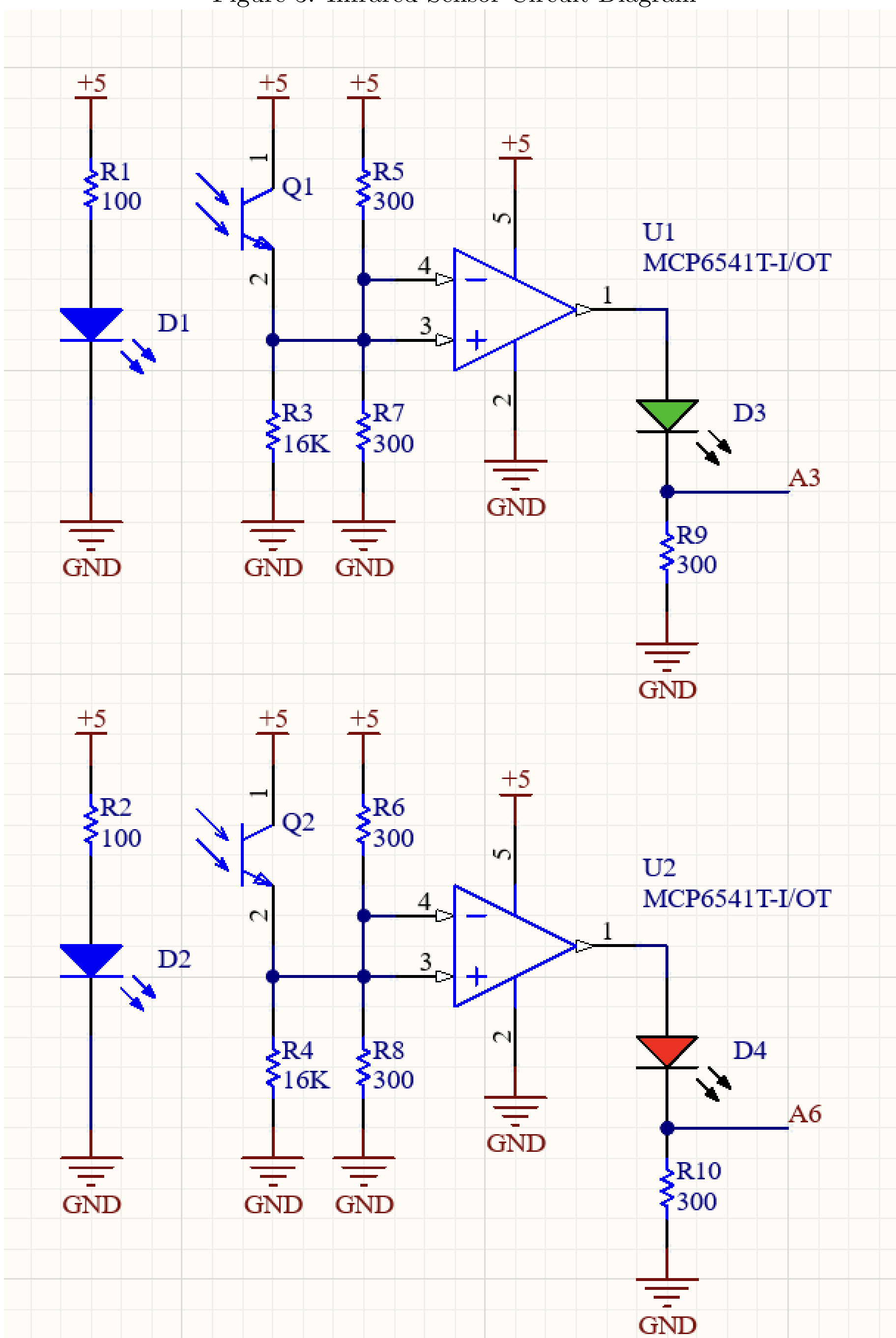
Please note: Kp and Kd labels are flipped in Figure 1. Euler angles should correspond to the proportional gain and angular rate should correspond to the derivate gain.

Figure 2: Simpliefied Block Diagram



2 IR Sensor Circuit Diagram

Figure 3: Infrared Sensor Circuit Diagram



For specificity, the top diagram corresponds to the right IR sensor and the bottom circuit corresponds to the left. Only the LED is different.

3 Table of Gains and Biases

Property	Value
Proportional Gain (Kp)	90
Derivative Gain (Kd)	9
Bias (b)	0.8*pi/180
Right Sensor	-2
Left Sensor	2
Normal	25

Please note: The bias value would often have to be changed depending on the surface which was being tested due to the sensitivity of the IMU to imperfections of the ground floor. The left and right sensor gains varied based on how far the servo needed to be turned and which direction. It also depended on how tight of turns were needed for the track. The Normal value is the constant between the right and left sensor which allow for the servo to return to center when sensors are not detecting anything.

4 Debugging

The most challenging part of the project and the largest energy sink was balancing the bike in place. We initially started with the code and gain values given in Lab 5 for baseline balancing, and quickly realized it would not balance. We then started playing around with the gain values for Kp and Kd. We mainly kept the same ratio and made sure Kp was higher than Kd. We realized that was not working so we started from scratch and from the fundamentals and looked at hardware itself. We studied the physics of the flywheel to determine which way it should spin to counteract the downward force of the bike falling. We decided that the wheel should spin in the direction of the angle at which the bike is falling, and made sure that gain had the correct sign. We verified this qualitatively verified this by feeling the bike as it's falling, if it span the incorrect direction, it would "feel" like it pushes against our hand as we feel it fall. Despite verification, it still was not balancing, so I decided to add signal loggers to the Kp and Kd signals to take a more quantitative approach debugging. This would allow for analyzing the oscillations through the data inspector. At the same time, I had asked my roommate, who has some controls experience with robotics, and he had described me a process to fine tune the gain, and this is what I learned: Start with Kd and Kp at 0, increase the proportional gain until oscillations start to happen, once you see that and it overcorrects, increase the derivative gain until the oscillations dampen to convergence. This process was not working. One final strategy we did was double checking our knowledge on Kp and Kd and one thing we realized, was that from the Lab 5 model, Kp was tied to the angular rate, while the Kd was tied to the euler angle. From my understanding, the rate is the change in euler angle, which should be derivative term, and the euler angle was the proportional value, meaning there was an error in labelling. We then tried swtiching the intial values, putting 4 where Kp was and 60 where Kd was, and it started getting close to balancing. After some fine tuning, we reached 9 and 90 for Kd and Kp respectively, (Kp and Kd respectively looking at the original code from the PSET5 spec.) and it was balancing. It was even better after eliminating weight shifting (i.e. from the battery holder), and determining the bias value. We finally had arrived at a very stable motorcycle.