	Kp	Kd	Velocity
Full Speed	100	5	630 degrees/sec
Lowest Speed	15	11	220 degrees/sec
Middle	32	4	560 degrees/sec

1) When running at full speed, the motor tended to oscillate severely as the positional error was reduced. Sometimes, it would oscillate back and forth almost a dozen times before settling on the final value, because the gain was high enough that the motor would overshoot the destination before the PID controller had a chance to run again. This was also a difficult test to run, since the motor would constantly draw too much current and cause the board to reset.

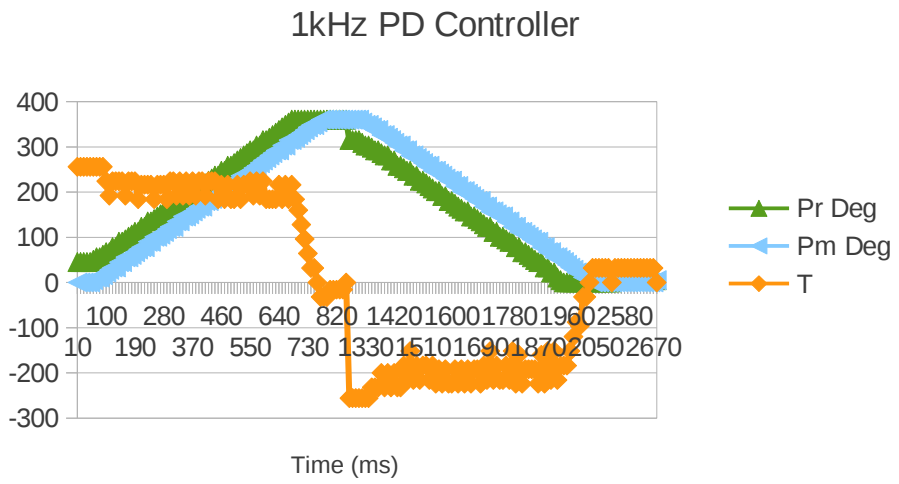
When running at the lowest speed, there was little to no oscillation, but it tended to take a lot longer to move larger distances. The middle speed offered the best compromise, where for large travels the speed ran towards its maximum, but once the error was reduced, the gains slowed the motor down so there wasn't any oscillation.

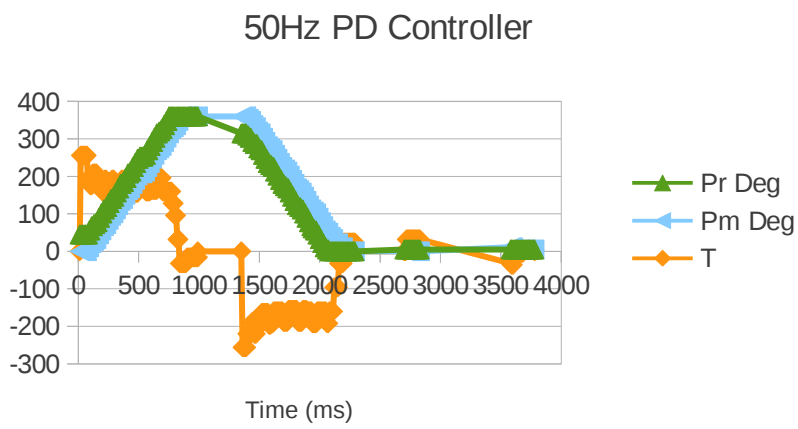
2) When using a larger step size, the torque ended up much higher (causing a really nasty noise from the motor). Other than the motor moving slightly faster, there wasn't much of a change compared to the normal step size; as the motor approached the destination position, the speed slowed down just as it did with the smaller step size, and there wasn't any oscillation.

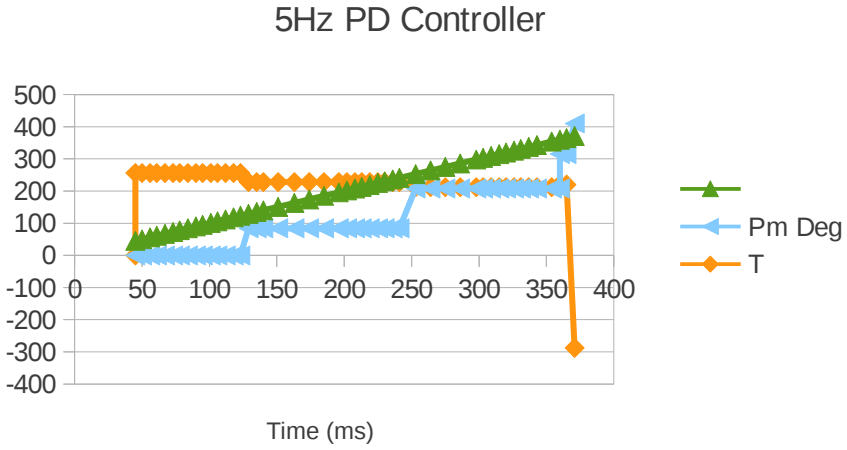
When tuning the controller for a large step size, it became impossible to move a small number of degrees. The Kp value was soo small to accommodate the large position error that the torque was too small to start moving the motor from a rest state.

3) When the PD controller was running at 50 Hz, it took twice as long to complete the cycle. At 50 Hz, the motor was able to travel much further during its period, usually about 25% of a step. This caused the PD controller to believe the current position was closer to the desired position, even though the interpolator is moving the finish line on the controller. Since the PD controller thins it's getting close to the destination, it calculates a lower torque and doesn't move the motor as fast. This causes it to approach the desired goal without overshooting, but it takes longer.

I was unable to complete the 5 Hz test due to the board resetting, however some information can be gleaned from the results I was able to measure. When the PID was running at 5 Hz, the motor would often move past an entire step before the PID controller would re-calculate torque values. Since during the last PID calculation the motor had to travel a full step, the motor was running at full speed forward. When it overshot the destination, it would often be by at least half a step. The PID controller calculated a negative torque, asking the motor to go in reverse. Since the motor was still going full speed forward, the current required to reverse the direction caused a drop in power and the board would always reset (hence why I only have data for one turn of the motor). However, if the board weren't resetting, evidence suggests that it would oscillate around the final point and would never (or rarely) actually settle.


1kHz PD Controller
400
300
200
100
0
-100
-200
-300
100 280 460 640 820 1420 1600 1780 1960 2580
10 190 370 550 730 1330 1510 1690 1870 2050 2670
Time (ms)
Pr Deg
Pm Deg
T


50Hz PD Controller
400
300
200
100
0
-100
-200
-300
0 500 1000 1500 2000 2500 3000 3500 4000
Time (ms)
Pr Deg
Pm Deg
T


5Hz PD Controller
500
400
300
200
100
0
-100
-200
-300
-400
0 50 100 150 200 250 300 350 400
Time (ms)
Pm Deg
T