Lab 2 Report

Dan Maas
SENG 5831
4/21/2013

Notes:

1. Velocity is measured in rpms in my system, NOT rotations per second.
2. Max torque is 100, min seems to be 6.
3. All KD values are divided by 12 internally, since I could not enter values < 1 via serial cable.

1. *Experiment with the speed of the motor: Run your motor at full speed. Modify your gains to achieve position control at that speed (as best you can). Slow the motor down as much as possible. Modify your gains to acheive position control. Repeat with one or two speeds in between. For each, record the approximate speed in rotations per second, record your equation (gains), and report on the behavior of the system and your ability to control the position.*

Max Speed (T=100)

Current parameters: kp=12 kd=4 vm=103. Control was very good, with slight jitter at the start and end of a rotation command, especially when changing motor direction.

Min Speed (T=6)

Current parameters: kp=12 kd=4 vm=9. Control was perfect due to extremely low velocity.

Mid Speed (T=50)

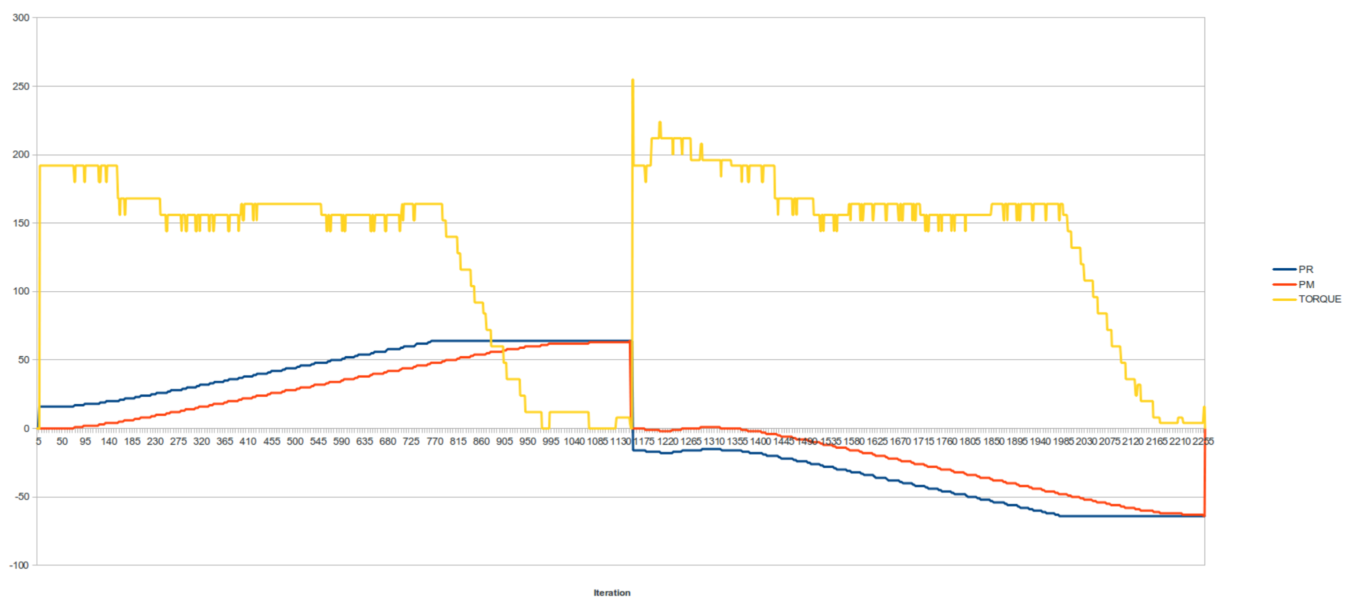
Current parameters: kp=12 kd=4 vm=60. Control was nearly perfect due to lower velocity, but slight, almost imperceptible jitter at completion of rotations.

2. *Change the step size to something very large (more than 2pi), and try a reference position of 4pi+current_position. How does system behavior differ from your tuned step size? Try tuning your controller for that very large step size. What happens if you then set the reference position to be very close to the current position (within a few degrees)?*

I saw a little bit more overshoot. This is expected, because the first term of the PID equation will be much larger due to the larger reference position relative to current position. This will make the torque be larger when the motor is close to its final position, relative to when it had a smaller step size.

I decreased kp to 8, and saw the elimination of the overshoot.

3. *Using your optimally tuned values for the PD controller running at 1kHz, graph Pm, Pr and T while executing the trajectory: rotate the motor forward 360 degrees, hold for .5 seconds, then rotate backwards for 360 degrees, hold for .5 seconds, rotate forwards for 5 degrees. Be sure to graph the entire trajectory.*



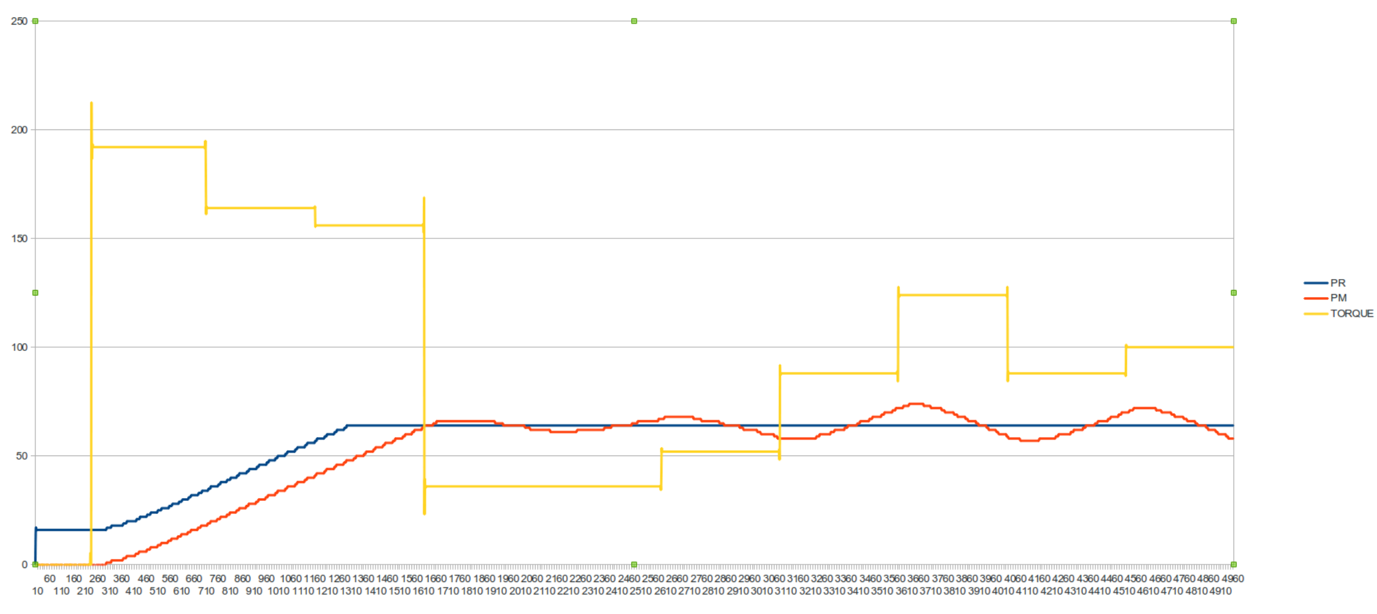
Note: my program converted degrees to ticks by truncation, so 5 degrees did not register. Rotations of > 10 degrees worked without overshoot or oscillations.

4. *Run your PD controller at 50Hz and 5Hz while graphing the same variables. Discuss the results.*

50Hz

At this frequency, you can see that the torque does not change values as often, but the overall shape is the same. Again, my program could not detect 5 degrees, as it converted to 0 ticks.

5Hz



At this frequency, you can see that the torque only changes values a few times. This causes overshoot and an infinite oscillation at the destination. The program was unable to proceed past the initial 360 degree rotation, but subsequent tests showed that all rotations cause oscillations at the end position.