Control of an Unstable, Inverted Pendulum

Angela Almquist & Ethan Spaid
Methods of Experimental Physics II

Objective

The purpose of this project is to program the movements of a robotic control arm to maintain control of an unstable, inverted pendulum using a PID algorithm.

Theory

Control

With any perturbation from the pendulum's upright position, gravity produces a torque at the pendulum's center of mass. The control arm can counteract this motion by an acceleration at the pendulum's pivot. Summing the torques gives

$$\sum \mathcal{T} = \ddot{\phi}_{control} mR\cos\theta - mg\sin\theta = I\ddot{\theta}.$$

Critical Angle Determination

If this mechanism were made mobile, it would likely encounter an inclined plane in its path. There exists a critical angle where the arm cannot supply enough torque to maintain control. This angle is given by

$$\theta_{max} = \tan^{-1}\left(\frac{\ddot{\phi}_{max}R}{g}\right),$$

which has no dependence on the length of the pendulum. This is an interesting result which we attempted to verify.

The Algorithm

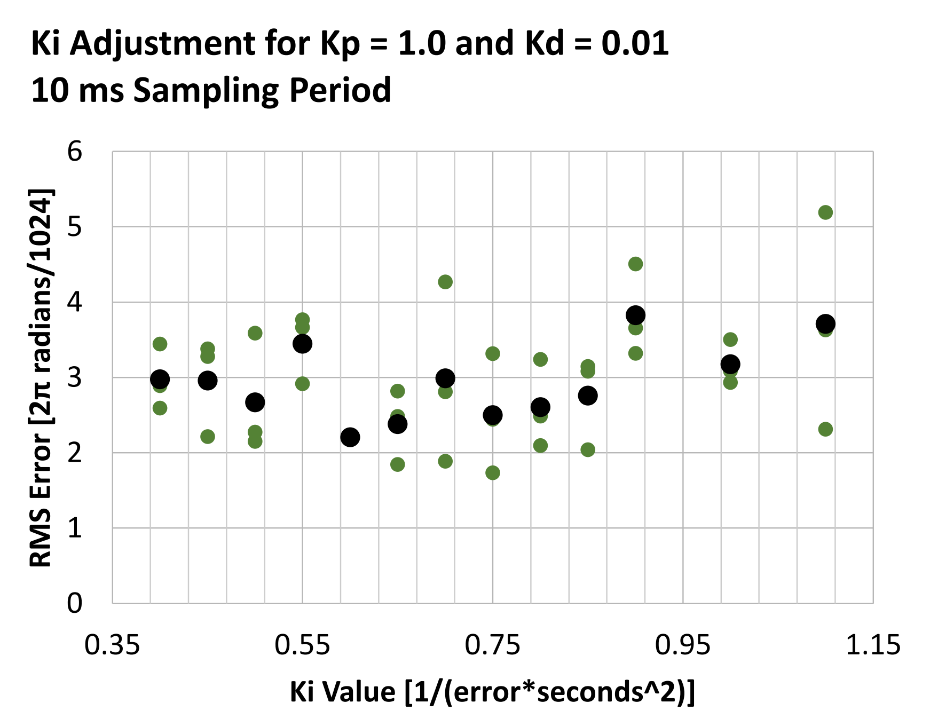
The PID algorithm is a control theory that consists of three parts: Proportional, Integral and Derivative. The sum of its parts outputs the corrective action needed to bring an unstable system into equilibrium. The corrective output is given by

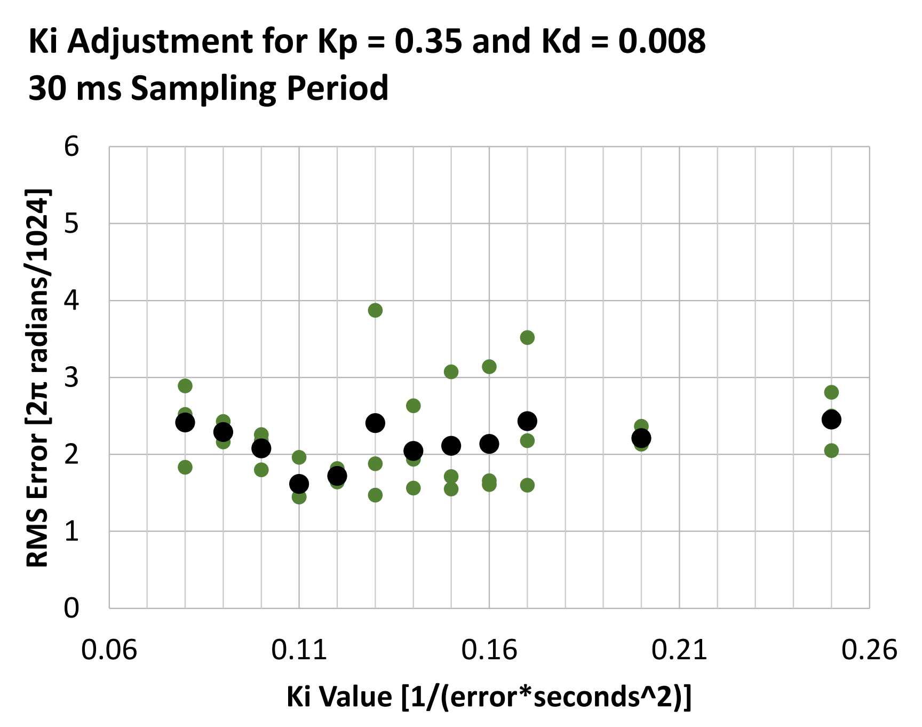
$$U(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{d\,e(t)}{dt},$$

where K_p is the proportional constant, K_i is the integral constant and K_d is the derivative constant. For our algorithm, the proportional term adjusts the frequency of the control arm proportional to the current error of the pendulum. The derivative responds to the rate change in the error compared to the expected rate of change, and the integral term adjusts the frequency if there is an accumulation of error over time.

Data Analysis

The algorithm was tested at two sampling periods, 10ms and 30ms. Every time the position was sampled, the frequency of the control arm was adjusted. The 30 ms algorithm worked best for small perturbations from the vertical. The 10 ms algorithm could respond faster to an applied force on the pendulum. The PID constants were finely tuned by recording the root-mean-square of the error over time as the value of one constant was varied while the other two were held constant.

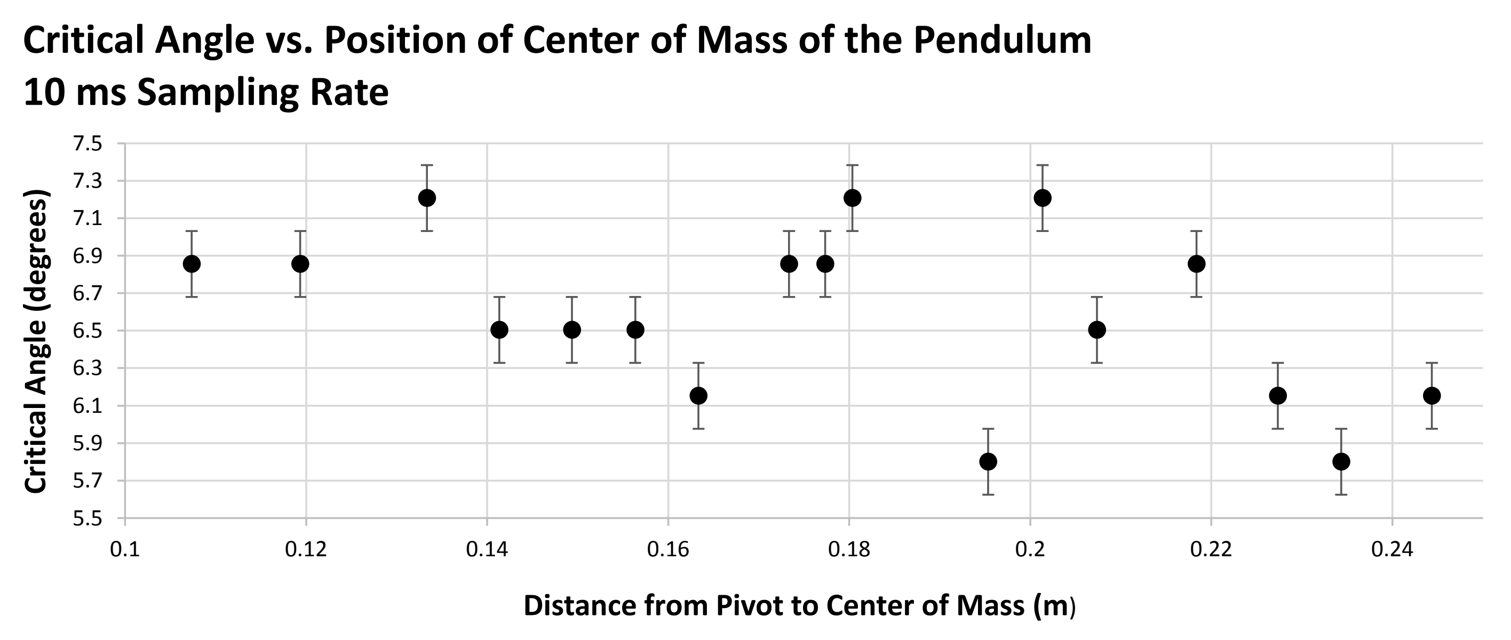




In the above plots, the average (in black) of the error from three trials (in green) is shown. The optimal PID constants for 10 ms and 30 ms were measured to be

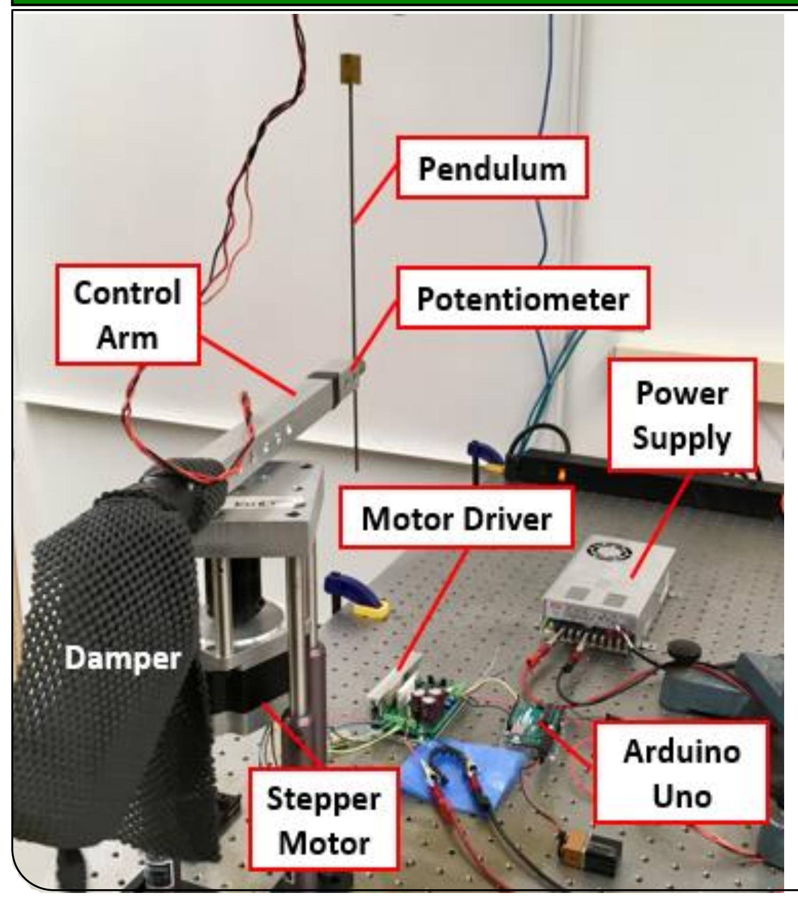
Sampling Period	K_p	K_i	K_d
10 ms	0.35	0.11	0.008
30 ms	1.00	0.60	0.007

The critical angle at which our mechanism lost control of the pendulum was $\theta_c = (6.59 \pm 0.18)^\circ$. The critical angle was also compared for various lengths of pendulum to prove no dependence on pendulum length.



The reduced χ^2 for a constant critical angle was 6.76. This is not an outrageous fit considering the limited amount of data collected.

Experimental Setup



The Arduino Uno is able to detect the pendulum's position by measuring a voltage difference across a potentiometer attached at the pendulum's pivot. The Arduino sends pulses out to the motor driver by the PID algorithm. For each pulse emitted, the stepper motor turns the control arm 0.9˚.

Conclusion

- Our algorithm was a success in that our robotic arm can correct for a force applied to the pendulum, and it can keep the pendulum upright for twenty minutes before human intervention is needed.
- Our theory suggests the critical angle should be independent of the length of the pendulum. We found $\theta_c = (6.59 \pm 0.18)^\circ$. Our data is insufficient to draw a conclusion on whether θ_c is truly independent of the length of the pendulum.
- There were a few potential sources of error we encountered. First, the computed frequency of our algorithm didn't precisely match the pulse output of the Arduino. Another source of uncertainty is that the Arduino's analog-to-digital converter has 10 bits so our resolution was limited.
- Future groups could improve on this project by investigating the effect of a known impulse force on the system.

References

1. X. Zhu, "Practical PID controller implementation and the theory behind," in2009 Second International Conference on Intelligent Networks and Intelligent Systems, pp. 58–61, Nov 2009.
2. J. Milton, J. L. Cabrera, T. Ohira, S. Tajima, Y. Tonosaki, C. W. Eurich, and S. A. Campbell, "The time-delayed inverted pendulum: Implications for human balance control," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 19, no. 2, p. 026110, 2009.
3. C. B. Prakash and R. S. Naik, "Tuning of PID controller by Ziegler-Nichols algorithm for position control of DC motor," International Journal of Innovative Science, Engineering & Technology, vol. 1, 2014