Paper ID #20399

MAKER: A 3D Printed Balancing Robot for Teaching Dynamic Systems and Control

Dr. Ryan W. Krauss, Grand Valley State University

Dr. Krauss received his Ph.D. in mechanical engineering from Georgia Tech in 2006. His research interests include modeling and control design for flexible robots, feedback control, and microcontroller-based implementation of feedback control systems. In addition to the freshmen introduction to engineering design course, he has taught courses in mechatronics, controls, vibrations, dynamics and robotics as well as senior design.

Mr. Chad Thomas VanderRoest, Grand Valley State University

MAKER: A 3D Printed Balancing Robot for Teaching Dynamic Systems and Control

Abstract

This paper presents a student project focused on designing a low-cost robot that can be used to teach dynamic systems and control. The robot is a two-wheeled balancing robot that is essentially an inverted pendulum. The robot is designed to carry a glass of water while performing various maneuvers such as line following. Students learn the importance of using feedback to stabilize the unstable inverted-pendulum system.

The body of the robot was 3D printed. A low-level, real-time feedback control algorithm is implemented on an Arduino. The high-level control can be handled by a Raspberry Pi. The Raspberry Pi and Arduino can communicate over USB serial or i^2c. The Raspberry Pi can be used to provide a wifi connection for the robot. Students also learn various aspects of Linux and Raspberry Pi in order to do the high-level control.

The final cost for the robot is less than $250 and it makes a useful experimental system for dynamic systems and control courses. It could also be used to generate interest in dynamic systems and controls among prospective students.

Introduction and Background

Capturing students' interest and motivating them to learn about dynamic systems and control can be quite challenging. Dynamic systems and control can be abstract and mathematically intensive. Physical experiments have been shown to deepen understanding and motivate students in dynamic systems and control courses [1,2,3,4,5,6,7]. Stability is a very important concept in dynamic systems and control, but it can seem particularly abstract. Developing students' conceptual understanding of stability if very important [8]. Helping students understand that feedback can affect system stability is particularly challenging [9].

This paper presents a student project aimed at designing a two-wheeled balancing robot to cultivate student interest in dynamic systems and control and demonstrate how feedback can stabilize an unstable system.

Figure 1: A picture of the two-wheeled robot balancing itself.

System Description

The two-wheeled robot is shown balancing itself in Figure 1. The body is 3D printed in three sections. Starter CAD files can be downloaded from www.thingiverse.com/thing:810998. A summary of how to build the robot is presented here:

http://makezine.com/projects/arduroller-self-balancing-robot/.

Balancing control is provided by the combination of an Arduino Uno clone and an inertial measurement unit (IMU) known by the part number MPU-6050. The MPU-6050 is a combination accelerometer and gyroscope. It has the ability to do various calculations on board to convert the acceleration and gyroscope measurements into tilt angles that are particularly helpful in a feedback control algorithm to balance the robot. The calculations are done using the Digital Motion Processor (DMP). The DMP is not well documented. Presumably, the calculations involved some form of a Kalman filter to estimate the angular position variables. Using the MPU-6050 with an Arduino is discussed here:
http://playground.arduino.cc/Main/MPU-6050.

The robot is driven by two DC motors with metal gear heads from pololu.com. The Arduino drives the motors via a low-cost H-bridge board based on the L298N chip.

Arduino and Raspberry Pi can make a powerful combination for robotic systems. The Arduino is well-suited for low-level, real-time control. The Raspberry Pi works very well for high-level control and wireless communication. The Raspberry Pi and Arduino can communicate with one another using serial, i^2c, or spi. Alternatively, the robot can be controlled using only an Arduino with either a long USB cable or a some form of wireless serial connection. As of this writing, the Raspberry Pi has not yet been installed on the robot and a long USB cable is still being used.

Pedagogical Value

This paper is not primarily about how to build the self-balancing robot, but how to use it in dynamic systems and control education. A two-wheeled robot that balances itself and carries a glass of water should at least pique students' interest, but this robot serves several other pedagogical purposes. Feedback control experiments can often digress into PID tuning, allowing students to succeed without going very deep in control theory or modeling. It is very difficult to simply tune a system that is open-loop unstable - most gain combinations that students will try will result in the robot simply falling over without giving them any real data. The difficulty of tuning a PID controller for this system should motivate students to learn how to generate a reasonable model for the system and how to use the model to actually design a controller rather than just tune one. Motivating students to learn about modeling of dynamic systems and then giving them a chance to apply the root locus technique to an experimental system should deepen student learning.

Additionally, the balancing robot can be valuable in teaching students that feedback can alter the stability of a system, which is often a very difficult concept for students to grasp. The controller balances the robot by altering the closed-loop pole locations for the system.

Modeling

Models for inverted pendulum systems can be quite complicated. Complicated models can be perfectly appropriate for advanced classes. However, if modeling is not a major focus for a particular courses, students could be provided with a mathematical model of the system. It is easier to model this system using Lagrange's equations rather than Newton's second law. So, this system could be used to bring Lagrange's method into the curriculum earlier than students might otherwise see it[10]. A reasonable model can be found using Lagrange's equations for the system of Figure 2.

The kinetic energy of the system is

$$\mathrm{KE} = \frac{1}{2}m_c\dot{x}^2 + \frac{1}{2}m_p(\dot{x}+r\dot{\theta})^2 + \frac{1}{2}I\dot{\theta}^2$$

where r is the distance from the pivot point to the center of gravity of the pendulum.

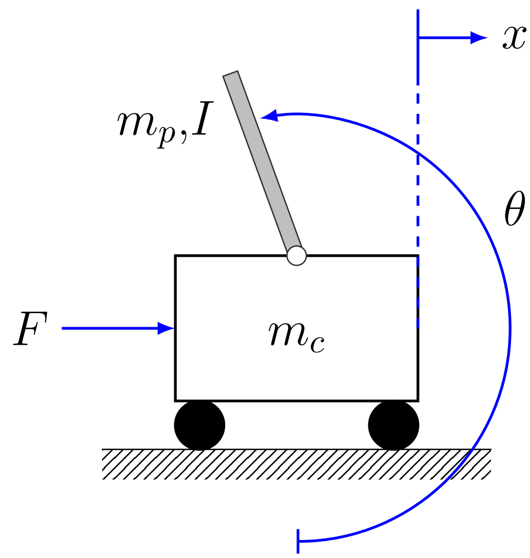

Figure 2: Sketch of a simple model of an inverted pendulum attached to a cart.

If θ is defined to be zero in the vertically downward position (i.e. the stable position), then the potential energy can be written as

$$\mathrm{PE} = m_p r g(1 - cos\theta)$$

Using Lagrange's equations, a model with the pendulum in the down position will take the form

$$\hat{C}\ddot{\theta} + \hat{A}\sin\theta = \hat{B}u$$

Dividing by $\hat{C}$ gives

$$\ddot{\theta} + A\sin\theta = Bu$$

Linearizing about $\theta = 0$ produces

$$\ddot{\theta} + A\theta = Bu$$

Linearizing about the vertically up position ($\theta = \pi$) leads to

$$\ddot{\hat{\theta}} - A\hat{\theta} + Bu$$

where $\hat{\theta} = \theta - \pi$. It is important to note that the A and B coefficients in the up and down positions are the same. This means that system identification in the stable, downward position can be used to experimentally determine the values of A and B. The natural frequency of the pendulum in the downward position will be $\omega_n = \sqrt{A}$.

Control Design

Beyond capturing students' interest in dynamic systems and control, the primary pedagogical value of this robot is motivating students to learn about control design. Unless students are

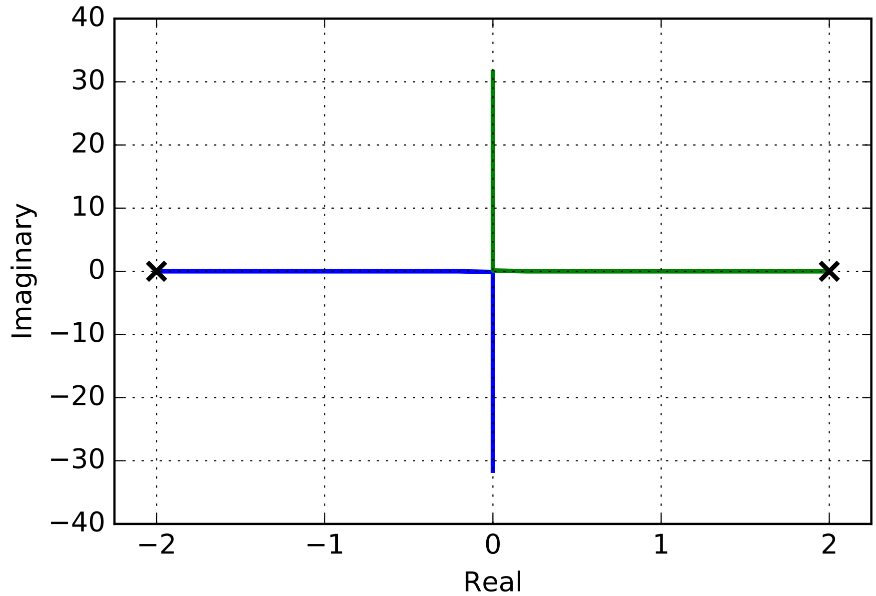

Figure 3: Root locus for proportional control of the balancing robot.

exceptionally lucky, it will be very difficult to tune a controller to stabilize the balancing robot in the vertically upward position. It would probably be a good learning activity to let them try. Assuming they are fairly quickly frustrated by trying to guess PID gains that work, they should be motivated to learn how the root locus design technique applies to this problem.

The model of the robot in the vertically upward position should lead to a transfer function of the form

$$G(s) = \frac{N}{(s+p)(s-p)}$$

where $p = \sqrt{A}$. It should be relatively easy to determine A from a free response test in the stable position by simply releasing the robot from some initial angle and finding the frequency of the pendulum.

Once the form of the transfer function is known, it is easy to construct a proportional control root locus, as shown in Figure 3. The system is unstable for small values of the the proportional gain K_p and then becomes marginally stable for larger values. A marginally stable balancing robot would move back and forth continually, never settling down but also not falling over. Students could deepen their understanding of the root locus technique by investigating what happens as K_p is increased. However, ultimately what is needed is PD control.

The cleanest way to design a PD controller for the balancing robot is to cancel the stable pole at $s = -p$ by using the form

$$K_d s + K_p = K_d(s+a)$$

where $a = K_p/K_d$. Placing a zero on the root locus corresponds to specifying the ratio of K_p/K_d. Once this is done, the students only need to adjust one gain to stabilize the robot and larger gains should ensure stability. Again, it would be valuable for students to carefully observe the system response as the gain K_d is increased and compare the response to the root locus pole locations.

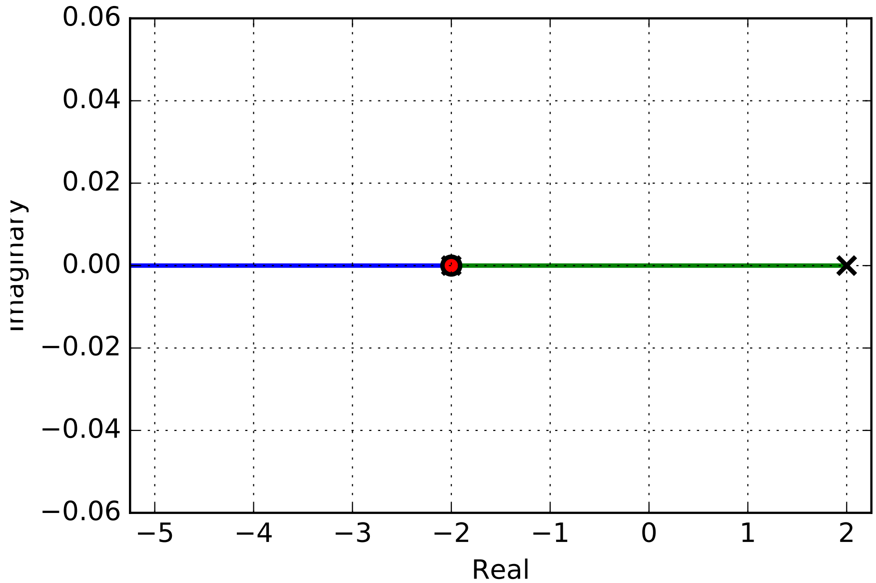

Figure 4: PD control root locus for the balancing robot.

The root locus for PD control is shown in Figure 4. Students could also investigate what would happen if a different location for the zero a is chosen. If a is placed to the left of the stable pole at $-p$, the root locus will be more complicated and would lead to an under-damped response for small to moderate valued of K_d.

Current State of the Design

The robot is able to balance itself and reject small disturbances that attempt to knock it over. The balancing is done using PD control and the combination of an Arduino Uno clone and the MPU-6050. The Raspberry Pi has not yet been attached and the robot cannot yet follow a path.

There are two issues related to friction that must be overcome before the robot will fully function. One is deadband in the motors most likely caused by friction in the gear head. The result is that the motors do not turn until the input voltages are large enough to over come friction. The other issue is that friction seems to be more severe in one motor, causing it to be slightly slower than the other motor over most of the operating range. Both of these issues could be solved by low-level motor control using encoder feedback to ensure that the motors are spinning at the same speed and that they spin even for small inputs. The primary challenge in implementing this low-level motor control is that an Arduino Uno only has two pin interrupts. One of those is currently being used by the MPU-6050. Encoder feedback for both wheels would require at least two additional interrupts. One solution is to replace the Uno with an Arduino Mega, which has 6 pin interrupts. Another option would be to combined two Unos and have them communicate over i^2c.

Concussions and Future Work

The balancing robot can be constructed for less than $250 using 3D printed parts and open-source electronics such as Arduino and Raspberry Pi. It will be used in the future to capture students interest in dynamic systems and control and to motivate students to learn about modeling and root locus control design.

Future work will include implementing low-level motor control that over comes deadband and ensures that the wheels turn at the same speed when the robot is supposed to be traveling in a straight line. The Raspberry Pi will be attached soon and higher-level path following control will be added.

Assessment of the effect of the robot on student learning and motivation in dynamic systems and control courses will also be conducted in future courses.

References

[1] Bernstein, D., "Enhancing undergraduate control education," *Control Systems Magazine, IEEE*, Vol. 19, No. 5, oct 1999, pp. 40 –43.

[2] Bernstein, D., "Control experiments and what I learned from them: a personal journey," *Control Systems Magazine, IEEE*, Vol. 18, No. 2, apr 1998, pp. 81–88.

[3] Shiakolas, P. and Piyabongkarn, D., "Development of a Real-Time Digital Control System with a Hardware-in-the-Loop Magnetic Levitation Device for Reinforcement of Controls Education," *IEEE Transactions on Education*, Vol. 46, No. 1, 2003, pp. 79–87.

[4] Kamis, Z., Topcu, E., and Yuksel, I., "Computer-Aided Automatic Control Education With a Real-Time Development System," *Computer Applications in Engineering Education*, Vol. 13, No. 3, 2005, pp. 181–191.

[5] Salzmann, C., Gillet, D., and Huguenin, P., "Introduction to Real-time Control using LabVIEW with an Application to Distance Learning," *Int. J. of Engineering Education*, Vol. 16, No. 5, 2000, pp. 372–384.

[6] Schinstock, D., McGahee, K., and Smith, S., "Engaging students in control systems using a balancing robot in a mechatronics course," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 6658–6663.

[7] Bay, C. J. and Rasmussen, B. P., "Exploring controls education: A re-configurable ball and plate platform kit," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 6652–6657.

[8] Raviv, D. and Roskovich, G. J., "An intuitive approach to teaching key concepts in Control Systems," .

[9] Krauss, R. W. and Croxell, J. R., "A Low-Cost Microcontroller-in-the-Loop Platform for Controls Education," *American Control Conference, 2012. Proceedings of the 2012*, IEEE, 2012.

[10] Bedillion, M. D., Raisanen, R., and Nizar, M., "Improving Transitions Between Sophomore Dynamics and Junior Dynamic Systems Courses," *Proceedings of the ASEE Annual Conference*, 2014.