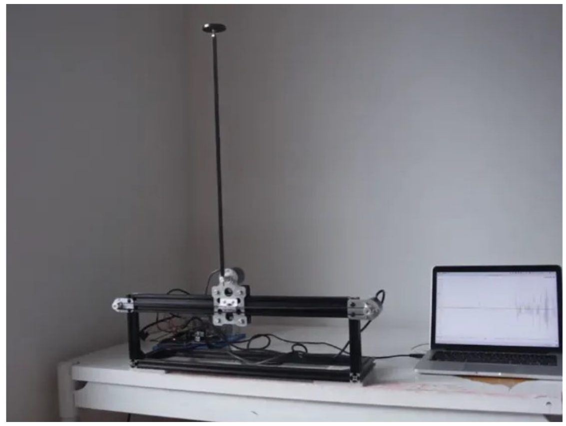
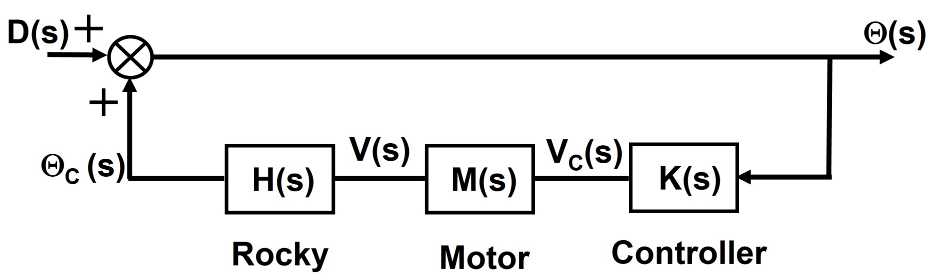


Now it's time to put together everything you've learned this semester. The project is a classic problem from control theory-to balance an inverted pendulum on a horizontally-translating cart. This is one of the most commonly used representations for an unstable system. We will be using a modified Balboa (hence the double-pun nickname, 'Rocky') 32U4 Balancing Robot Kit from Pololu Robotics and Electronics. Your job is to design control systems that keep the robot upright when it's subject to an impact disturbance. For this project, you will apply quantitative analysis using analytical, symbolic, and numerical tools, not to an idealization, but to a real system in a real way.

1 Initial System



The project starts with the simulation-based design of an *Initial System* in which the control system maintains balance by keeping its body (pendulum) in the vertical position and rejects any angular disturbance. You'll add a motor model to the closed-loop system that you created in the Day 10 in-class assignment. Recall that you found the transfer function for Rocky to be $H(s) = \frac{\Theta_c(s)}{V(s)}$ and used a PI controller, $K(s) = K_p + \frac{K_i}{s}$. Assume that the motor model can be represented as a first order system where its transfer function

$$M(s) = \frac{V(s)}{V_c(s)} = \frac{\frac{K}{\tau}}{(s + \frac{1}{\tau})} \tag{1}$$

relates the velocity of the cart (output) to the motor (velocity) control signal, $V_c(s)$ (input).

Do the following with your Rocky.

(a) Determine values for the parameters τ and K of the motor model from input-output measurements recorded from each (left and right) motor.

(b) Calibrate the gyroscope of the IMU (inertial measurement unit) onboard Rocky. Perform the calibration procedure described in the comments at the top of the sketch, Rocky_Gyro_Calibration.

(c) Determine the natural frequency of the system. For simplicity, you can consider damping to be negligible. Based on output measurements of angular position (from the gyro), determine the frequency of oscillations of the robot as it swings back-and-forth like a pendulum. Then determine its effective length, l_{eff}, using $\omega_n = \sqrt{\frac{g}{l_{eff}}}$. Recall that the equations of motion were derived with the assumption that the pendulum was a massless-rigid rod with an end mass. Directions for making the measurements are also in the header comments of the gyro calibration sketch.

(d) Decide upon performance specifications for balancing, *e.g.*, frequency of oscillations after a disturbance, decay rate of oscillations for disturbance rejection, *i.e.*, $\theta \to 0$. Explain how you chose your specifications and why they are reasonable.

(e) Identify locations for the poles of your system that correspond to your specifications.

(f) Determine the controller parameter values, K_p and K_i that result in an output response that meets your specifications. *Note* that you may not be able to place the closed-loop poles at *exact* locations so some trial-and-error iteration will be required. Use/modify the MATLAB analysis tool (script) that has been provided to determine the control parameters corresponding to your poles.

(g) Make a Simulink model of the closed-loop system. You'll have to define the form of the disturbance, *e.g.*, a small bump as a narrow square pulse. Make plots (output from a Scope block) that show the response of the angular position of body, $\theta(t)$, and the position, $x(t)$ and velocity, $v(t)$ of the cart to the disturbance. Describe and explain what the body and cart are doing.

2 Stationary Balancing System

Design a control system to keep Rocky as stationary as possible while balancing upright. There is no single, unique way to do this; although, it's likely that most of the ways involve using nested feedback loops to control the robot's position and velocity, in addition to the anglular position of the body. Before you start, read through the Guide for Stationary Balancing of an Inverted Pendulum on a Cart.

(h) Implement and test your control system on your Rocky. Modify the Rocky_Balance_Starter_Code sketch by adding your balancing algorithm. Follow the instructions in the header comments. You will need to add values for the control parameters and equations for the *motor control input signals*.

(i) Make a Simulink model of your closed-loop, stationary balancing system and run your control system with it. What are similarities and differences between the Simulink model's and Rocky's response behavior?

3 Modified System (Optional)

(j) If you're looking for an additional challenge, modify your control system to make Rocky do something *interesting*. Here's where you can be creative. For example, control the velocity of the cart to go to a specified position and come to a stop or drive around an obstacle course or "dance" to music. You could interact with the buzzer. Or ...

For the modfoied system, you do not have to make a Simulink model of it but may find it helpful to do so.