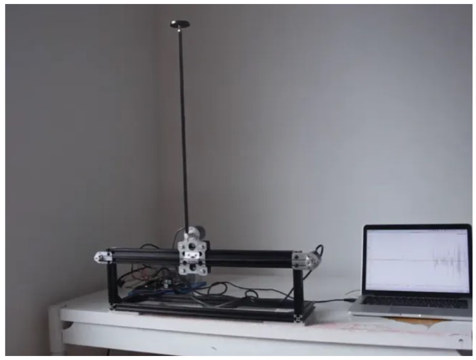
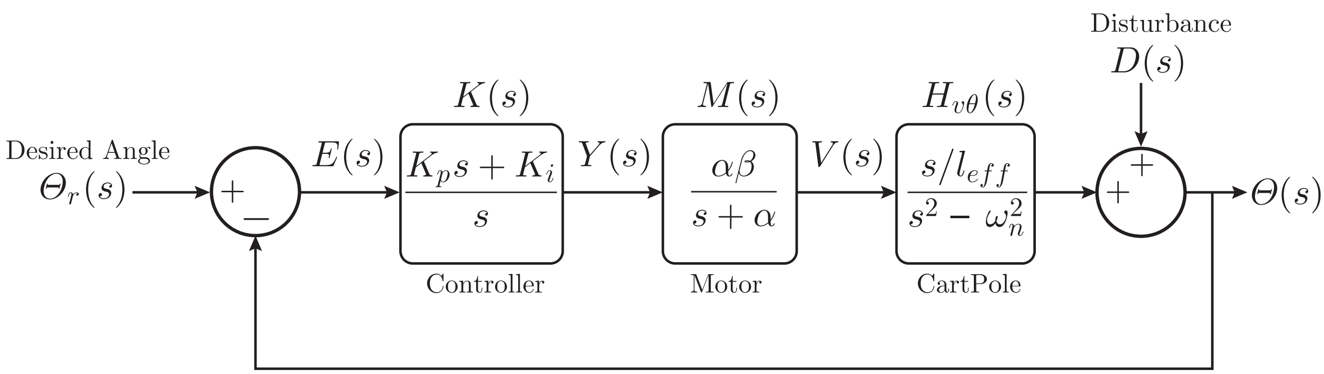
Designing a Controller for a Balancing Robot



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.



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 $H(s) = \frac{\Theta(s)}{V(s)}$ 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

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

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

Calibration and Initial Controller

Do the following with your Rocky.

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

(b) Determine the natural frequency of the system. 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 $\omega_n^2 = \frac{g}{l_{eff}}$.

(c) 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. (Hint: try moving the dominant pole as far left as possible!).

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

(e) Determine the controller parameter values, K_p and K_i that result in an output response that meets your specifications. *Note* that you can't place the closed-loop poles in any desired location *exact* location: the average pole value is fixed at $-\frac{\alpha}{3}$ (see the resource document). Some 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.

(f) 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.

(g) 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.*

Stationary Balancing System

Design a control system to that returns that regulates the Rocky's current position (driving it to zero) while also 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 angular position of the body. A suggested controller topology has been provided in the resource document. The matlab script 'Rocky_5_closed_loop_poles.m' can aid in the design of such a controller.

(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?

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.

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

Deliverables

1. **Written Report**: It's not necessary to include lots of text but the report should be clear and easy to read. A student familiar with the material in ESA should be able to follow and reproduce what you've done. Present evidence to support your design decisions by means of labeled plots, figures, tables, equations, calculations, etc. At a minimum, your report should describe/include the following:

   - Parameter identification procedure to determine the motor constants. Show calculations (e.g., a plot from the MATLAB curve fit tool) and values.
   - Parameter identification procedure to determine the (first-swinging) natural frequency of the robot and the effective length of the pendulum body. Show calculations with plots and values.
   - Initial system's response-to-disturbance behavior (including the angle of body, and velocity and position of the cart), pole values and rationale for their placement, calculation of control constants, and Simulink model with response plots.
   - Stationary balancing system's response-to-disturbance behavior (including the angle of body, and velocity and position of the cart), pole values and rationale for their placement, calculation of control constants, and Simulink model with response plots.
   - Modified system (for stretch goal). Description of the purpose of the control system. Details of how the control system was designed with accompanying calculations of system and control parameters. Simulink model if appropriate. Description and evaluation of performance.
   - Include block diagrams that show your system implementations. Define all blocks, specify their respective transfer functions, and identify parameters describing how they were chosen or determined.
   - Include copies of all MATLAB code that you wrote/used/modified and the Arduino IDE sketch that you created/modified (no need to submit calibration sketches).
   - Note: the documentation for your project **cannot** be a livescript. However, it's fine if you used livescripts to perform your analysis.

2. **Video**: Short video demonstrating controlled motion of your stationary balancing system and modified system, if done. You may combine various clips to make up the video. Important: Please submit your video by uploading it YouTube, and by providing a link to it as part of your project report. Do not upload your video file directly to Canvas as it cannot be viewed without downloading.

3. **Individual Engagement**: There is an expectation that each team member will fully contribute to project. This includes working in-class. Please notify the instructors if you'll be absent. Individual grades may be adjusted for lack of engagement.