24-774: Special Topics in ACSI
Laboratory 2: Balancing Robot Control

Background: In this lab we will use the full functionality of the Tumbller system, replacing the stock controller with an LQR based design. The controller design will be broken into two parts: state estimation, where we combine the sensors on the device to estimate the robot state; and LQR design, where we design the state feedback portion of the controller

Figure 1: Tumbbler hardware. The hardware has various sensors (wheel encoders, IMU, ultrasonic) that can be used to control the 4 degree of freedom robot.

Background

We need a dynamics model of the Tumbller system. For this lab we will ignore the yaw angle of the robot, i.e. we will look only at straight line translation of the system.

The system dynamics are exactly the same of those of a Segway or hover board type system, and there are many sources online that derive these dynamics. A good resource for the overall approach is given by https://kth.diva-portal.org/smash/get/diva2:916184/FULLTEXT01.pdf. The linearized equations about the upright position are given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I_{pend} + m_{pend} l^2) f}{I_{pend}(m_{cart} + m_{pend}) + m_{cart} m_{pend} l^2} & \frac{m_{pend}^2 g l^2}{I_{pend}(m_{cart} + m_{pend}) + m_{cart} m_{pend} l^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-m_{pend} l f}{I_{pend}(m_{cart} + m_{pend}) + m_{cart} m_{pend} l^2} & \frac{m_{pend} g l (m_{cart} + m_{pend})}{I_{pend}(m_{cart} + m_{pend}) + m_{cart} m_{pend} l^2} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \dfrac{I_{pend} + m_{pend} l^2}{I_{pend}(m_{cart} + m_{pend}) + m_{cart} m_{pend} l^2} \\ 0 \\ \dfrac{m_{pend} l}{I_{pend}(m_{cart} + m_{pend}) + m_{cart} m_{pend} l^2} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D = 0$$

for the state vector $x = \begin{bmatrix} x \\ \acute{x} \\ \theta \\ \acute{\theta} \end{bmatrix}$ and input of force. To convert between motor voltage and force we can use a simple linear approximation as shown in Section 3 in the reference.

$$u = \frac{2k_T}{Rr} V,$$

where u is the force, k_T is the motor torque constant, R is the motor resistance, r is the wheel radius, and V is the motor voltage.

As seen in the reference, the parameters are typically estimated using CAD models / experimentation; for this lab we will provide you with values that are close enough to perform the design.

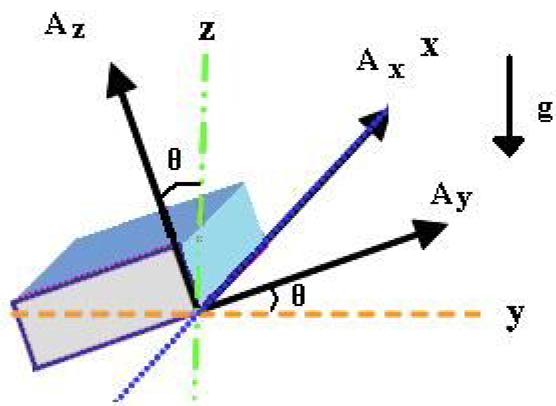
Parameter	Value
m_{cart}	0.493 kg
m_{pend}	0.312 kg
I_{pend}	0.00024 kg.m^2
L	0.04 m
f	0.01 N.s/m
k_T	0.11 N.m/A
R	10 ohm
r	0.0335 m

State Estimator Design

A state estimator is used to fuse the sensors in the IMU (gyros and accelerometers) and to add information from the wheel encoders to estimate the full robot state.

1. Let's begin by getting a strong estimate of the angle using the two primary IMU sensors – the accelerometer and gyroscope. A *complementary filter* can be used to fuse the accelerometer data and gyro data into a good estimate of the angle. A 3-axis accelerometer

can be used to determine the angle with respect to the gravity vector as shown below; it is accurate in steady state, but disturbance forces lead to transient errors. On the contrary, gyros provide accurate *angular velocity* estimates, but the integration to absolute angle leads to low frequency drift errors. The complementary filter fuses the two sensors by high pass filtering the gyros and low pass filtering the accelerometers.



$$\theta = arctan\left(\frac{A_y}{\sqrt{A_x{}^2 + A_z{}^2}}\right)$$

Figure 2: Accelerometer principle of angle measurement. The inverse tangent operation can be used to determine the angle of an accelerometer axis relative to the gravity vector.

We can write this filtering mathematically via

$$\theta(k) = \alpha\big(\theta(k-1) + \omega_{gyro}(k)T\big) + (1-\alpha)\theta_{acc}(k),$$

where α is a parameter that weights our relative trust in the gyro and accelerometer data. If $\alpha = 1$, we simply integrate the gyro data to get to angle; if $\alpha = 0$, we simply use the accelerometer data.

For this portion of the lab, construct a complementary filter that estimates the angle. Change the robot angle manually and observe your filter output (pay attention to how quickly it adapts AND how the angle drifts when you stop). Choose a value of α that gives performance you are happy with (NOTE: α is typically close to 1 to give a fast response speed). Record a video of you moving the robot along with your real-time angle estimates and provide a youtube link.

2. Now, using your complementary filter as your angle estimate, consider two approaches to creating your overall state estimator:
   a. Use the gyro measurements directly for the angular velocity state. Here we are effectively changing the C matrix to $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.
   b. Apply a low pass filtered version of the angle estimate derivative for the angular velocity state. Choose the bandwidth of your low pass filter to provide a fast response.

Move the robot by hand and plot the states for both state estimators on the same graph. What are the pros and cons to using approach a or b?

LQR Control

For this 4-state system LQR is a very effective control design method. We will start with a PD balancing controller and then move on to LQR. HINT: It's probably easiest to get the robot to balance on a carpet.

1. Using the continuous time state space model, find the transfer function from motor voltage to angle output. Design a PD controller using the method of your choice to balance the pendulum in the upright position. Implement the controller, and record a video of the robot balancing upright.

2. Design an LQR controller based on the continuous time model above. Tune Q and R to highly penalize errors in angle and angular velocity relative to position and control usage. Record a video that shows your robot standing upright.

   Adjust the sampling rate of your implementation. What is the slowest sampling rate for which your controller works?

3. Repeat the design for a discrete time model using the lowest sampling rate from above. Again record a video of your implementation – how does the performance compare to the CT design at this sampling rate? Can you get it to stabilize at a lower sampling rate using this approach?

Reporting

Compile a single PDF lab report following the guidelines below.

Your report should be organized in terms of numbered items in the lab procedure. For each numbered item in the lab procedure you must address the following items at a minimum:

1.) The details of all calculations involved in generating your results. Be sure to highlight the main results.
2.) Presentation of your results in the form of plots and tables. This should include all relevant plots and Simulink models. Do not present plots that use the black background that is the Simulink scope default.
3.) General discussion. What sense do you make of the results? What can you conclude?
4.) Answers to all the discussion questions in the lab procedure.

After completing these tasks for all numbered items in the lab procedure, complete the following sections to finish your report:

- Conclusions: What were the main results? What did you learn (if anything) by completing the lab? What suggestions do you have to make the lab better or more interesting?
- References: Compile all of your references into a single section at the end of the document. I highly recommend the use of a reference manager, e.g. Bibtex, EndNote, etc.