

FEEDBACK CONTROL SYSTEMS FINAL REPORT

CARAVAN CONTROL

TUCKER HAYDON AND CONNOR BRASHAR

ABSTRACT. This is the project abstract.

1. SCENARIO

You are an engineer in charge of designing and implementing an estimation and control system for a fleet of self-driving semi-trucks. Due to government regulation, the truck fleet can only operate in the self-driving mode when they are on long, straight stretches of highway between cities. In order to reduce costs and conserve fuel, the trucks drive very closely together at a pre-determined speed in ‘caravan formation’. By driving very closely together, the trucks can draft off of one another, reduce drag, and save fuel by upwards of 21% [1].

For the system you are designing, only three trucks will be in the caravan. The caravan is equipped with the following sensor suite:

- (1) The lead truck is equipped with a GPS receiver that measures its position at 1 Hz.
- (2) The two following trucks are equipped with range sensors that measure the relative position between themselves and the truck in front of them at 10 Hz.
- (3) All three trucks are equipped with an IMU that measures respective acceleration at 100 Hz.

Each truck may be independently controlled and may be instructed to either accelerate or decelerate.

2. NOMINAL SYSTEM DESCRIPTION

Define the system state.

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (1)$$

Define the system input.

$$\vec{u} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (2)$$

Define the system dynamics.

$$\dot{\vec{x}} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \vec{x} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_B \vec{u} \quad (3)$$

Define the system observer equation.

$$\vec{y} = \begin{bmatrix} x_1 \\ \Delta x_{12} \\ \Delta x_{23} \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_C \vec{x} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_D \vec{u} \quad (4)$$

2.1. Observability. Using the nominal system, determine whether or not the system is observable. The nominal system is linear and time-invariant, so the observability Grammian is simplified:

$$W_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (5)$$

If the rank of $W_o = n$ then the system is observable. **The nominal system is observable.**

2.2. Controllability. Using the nominal system, determine whether or not the system is controllable. The nominal system is linear and time-invariant, so the controllability Grammian is simplified:

$$W_c = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \quad (6)$$

If the rank of $W_c = n$ then the system is controllable. **The nominal system is controllable.**

2.3. Conclusion. The fact that the nominal system is both observable and controllable indicates that it is well-designed and amenable to both estimation and control.

3. ESTIMATOR

TODO Connor

4. CONTROL SYSTEM DESCRIPTION

4.1. Error State System. Controllers drive systems to the origin. The nominal system, however, should not be driven to the origin. Instead an error-state system should be specified where the error is the deviation of the nominal system from a reference signal.

First, define the reference signal. From the problem statement, the goal is to maintain a specified distance between the trucks and to keep that at a constant velocity.

$$\vec{x}_r = \begin{bmatrix} x_{1_r} \\ \Delta x_{12_r} \\ \Delta x_{23_r} \\ v_{1_r} \\ v_{2_r} \\ v_{3_r} \end{bmatrix} \quad (7)$$

The error signal is the difference between the nominal and the reference states. Note that not all states have a corresponding reference signal.

$$\vec{x}_e = \begin{bmatrix} x_1 - x_2 - \Delta x_{12_r} \\ x_2 - x_3 - \Delta x_{23_r} \\ v_1 - v_{1_r} \end{bmatrix} \quad (8)$$

The error signal dynamics don't conform to the standard $\dot{x} = Ax + Bu$ form — there is an affine transform between the error dynamics and the nominal and reference signals. However, the standard form can be composed via the following trick: append the reference signal to the nominal state and define zero dynamics and full observability for the reference substate.

Stack the nominal and reference states into the *stacked state*.

$$\vec{x}_s = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ v_1 \\ v_2 \\ v_3 \\ \Delta x_{12_r} \\ \Delta x_{23_r} \\ v_{1_r} \end{bmatrix} \quad (9)$$

Define the stacked state dynamics.

$$\dot{\vec{x}}_s = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ a_1 \\ a_2 \\ a_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_s} \vec{x}_s + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{B_s} \vec{u} \quad (10)$$

Define the stacked state observer equation. Note that the reference states are specified and fully known so there is full state feedback for these states.

$$\vec{y} = \begin{bmatrix} x_1 \\ \Delta x_{12} \\ \Delta x_{23} \\ a_1 \\ a_2 \\ a_3 \\ \Delta x_{12_r} \\ \Delta x_{23_r} \\ v_{1_r} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{C_s} \vec{x}_s + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{D_s} \vec{u} \quad (11)$$

Now define the error state by linearly transforming the stacked state via a *similarity transform*.

$$\vec{x}_e = \begin{bmatrix} x_1 \\ \Delta x_{12_e} \\ \Delta x_{23_e} \\ v_{1_e} \\ v_2 \\ v_3 \\ \Delta x_{12_r} \\ \Delta x_{23_r} \\ v_{1_r} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_T \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ v_1 \\ v_2 \\ v_3 \\ \Delta x_{12_r} \\ \Delta x_{23_r} \\ v_{1_r} \end{bmatrix}$$

Now, leverage the similarity transform to define the error state dynamics.

$$\begin{aligned} \dot{\vec{x}}_e &= T A_s T^{-1} \vec{x}_e + T B_s \vec{u} \\ y &= C T^{-1} \vec{x}_e + D \vec{u} \end{aligned}$$

Finally, clean up the equation by redefining the system.

$$\begin{aligned} \vec{x} &:= \vec{x}_e \\ A &:= T A_s T^{-1} \\ B &:= T B_s \\ C &:= C_s T^{-1} \\ D &:= D \end{aligned}$$

Now the system is in the standard form.

$$\begin{aligned} \dot{\vec{x}} &= A \vec{x} + B \vec{u} \\ y &= C \vec{x} + D \vec{u} \end{aligned}$$

4.2. Discrete-Time Dynamics. Software implementations of continuous kinematic systems must first discretize the system dynamics. Many sensors and computers cannot produce nor consume continuous signals. As a result, system dynamics or measurements are represented as discrete systems.

As the system is LTI,

4.3. Discrete-Time LQR Controller. Together, a Kalman Filter and LQR controller form a Linear-Quadratic-Gaussian (LQG) controller.

5. SIMULATION

The LQG controller was implemented in Matlab. The goal of the controller is to bring the trucks into the 'caravan' formation where the trucks follow each other at 30 meters/second with a separation of only 5 meters.

$$\vec{x}_r = \begin{bmatrix} \Delta x_{12} \\ \Delta x_{23} \\ v_1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ 30 \end{bmatrix}$$

6. DISCUSSION

The LQG

7. MUSEUM EXHIBIT

The following section describes a potential museum exhibit aimed at high-school and middle-school children.

Note that this section is an abridged version of the original project proposal. In discussion with Dr. Tanaka, we agreed that the primary portion of the project would be the design, implementation, and simulation of an LQG controller and that, to make grading easier, we would try to conform to the original project proposal.

7.1. Description. We propose a museum exhibit composed of toy trains on tracks where visitors attempt to tune an LQR controller to bring three independent locomotives into the caravan formation. On a 3 meter by 3 meter tables, toy train tracks are laid out in a wide circle ¹. Three battery-powered locomotives may be placed anywhere on the tracks. Visitors may turn 6 knobs corresponding to the 6 diagonal entries of the LQR weighting matrices to tune the system.

7.2. Implementation. Visitors may walk around all sides of the table. On one side of the table are six knobs that visitors may twist to tune the LQR controller. On the far end of the table opposite the knobs is a TV. The TV first shows a short 30-second video describing the goal of the exhibit, what a 'good' controller looks like, and what each of the 6 knobs do. Visitors are then free to twist the knobs (a digital value is displayed on the screen), and then press the 'go' button.

Behind the scenes, a Raspberry Pi B+ (RPB) controls the three locomotives. The locomotives have been 'hacked' and carry a Raspberry Pi Zero (RPZ) that modulates the supplied battery

¹The hope is that a wide circle will approximately represent a long, straight, endless track

voltage to each of the locomotives. By modulating the battery voltage, the RPZ can directly control the speed of each of the locomotives. Each of the RPZs carry an inertial measurement unit that measures the acceleration of each of the locomotives. The RPZs transmit the acceleration data to the RPB via wifi. A camera is suspended above the table and connected to the RPB. The RPB uses an image processing algorithm on the incoming camera data to track the location of each of the locomotives and generate the position and position difference measurements. The RPB fuses the camera measurements and the accelerometer measurements together with the kalman filter described in section 3.

After the 'go' button is pressed, the RPB reads in the current analog knob values and solves the discrete-time algebraic riccati equation. Once solved, the RPB streams control input commands to the RPZs via wifi. The RPZs convert the control input into motor voltages and drive the locomotives.

If the camera detects that two locomotives have touched (i.e. a crash has occurred), the trains are stopped, and the visitor is notified of the crash and given another chance.

7.3. Price Breakdown.

Item	Total Price
2 Wooden Train Track	\$100
3 Battery-Powered Locomotives	\$150
1 Raspberry Pi B+	\$40
1 Wifi Dongle	\$15
1 Raspberry Pi Camera Module	\$25
3 Raspberry Pi Zero W	\$30
3 IMU Sensors	\$45
3 PWM Motor Shields	\$120
1 LCD TV	\$300
6 Knobs and 1 button	\$50
Total Price	\$875

REFERENCES

- [1] C. Bonnet and H. Fritz, "Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing," tech. rep., SAE Technical Paper, 2000.

THE UNIVERSITY OF TEXAS AT AUSTIN

E-mail address: thaydon@utexas.edu, connor.brashar@utexas.edu