# Project:P1

Ding Zhao
24-677 Special Topics: Linear Control Systems

**Due: Nov 12, 2019, 5:00 pm. Submit BOTH in class and online.**

- You need to upload your solution to Gradescope ( https://www.gradescope.com/) to be graded. The link is on the panel of CANVAS. If you are not familiar about the tool, post your questions on Piazza or ask during the office hours. We will use the online submission time as the timestamp.
- Submit **controller.py** and **BuggyStates.npz** to Gradescope under **Programming_P1** and your solutions in **.pdf** format to **Project-P1**. Insert the Buggy Simulator performance plot image in the .pdf. We will test your controller.py and manually check all answers.
- You are recommended to test your codes in Google Colab before submission, to ensure it executes with standard python compilers. Please refer to (https://drive.google.com/open?id =1g12AAAATJUSY0gRrZ0C2GTM6Y0MGndTkjNOavd41ExY) for documentation on how to use Colab.
- Right after each week's recitation, we have a half-hour Q&A session. You are requested to work on the assignment early and bring your questions to this session to take advantage of this support.
- You can also post your questions on the Piazza. We will try our best to give feedback within 24 hours during the workday and within 48 hours during the weekend. We will keep answering questions until 8:00 pm, Monday.

# 1 Introduction

This project is an individual project in which you are asked to design a controller for your Buggy (autonomous) team (go to https://www.springcarnival.org/buggy.shtml, if you would like to find out more about Buggy).
A control engineer would take the following steps for designing a controller for this system:

1. Examine control model (nonlinear) [already provided to you].
2. Linearize the state space system equations. [P1]
3. Develop a PID controller for the system. [P1]
4. Check the controllability and stabilizability of the system. [P2]
5. Design feedback controller using Pole Placement. [P2]
6. Design optimal controller. [P3]
7. Design stochastic controller. [P4]

This project is designed to get you well acquainted with the controller designing process for a 4 wheeler vehicle. The project is sub-divided into 4 assignments as:

- P1 (a) Linearize the state space model.
  (b) Design a PID lateral and PID longitudinal controller.
- P2 (a) Check the controllability and stabilizability of the linearized system.
  (b) Pole placement.
- P3 (a) Design an optimal controller for the vehicle.
- P4 (a) Design a Kalman Filter to filter disturbances in the states.
  (b) Race with other Buggy teams in the class

The deadlines of the 4 sub-parts will be :

- P1 is due at 5:00 PM on $Nov.12^{th}$
- P2 is due at 5:00 PM on $Nov.19^{th}$
- P3 is due at 5:00 PM on $Nov.26^{th}$
- P4 is due at 5:00 PM on $Dec.10^{th}$

# 2 Model

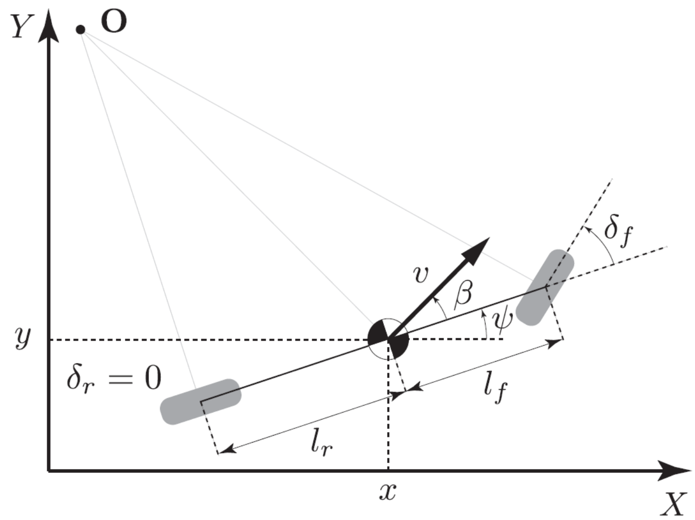


Figure 1: Bicycle model[2]

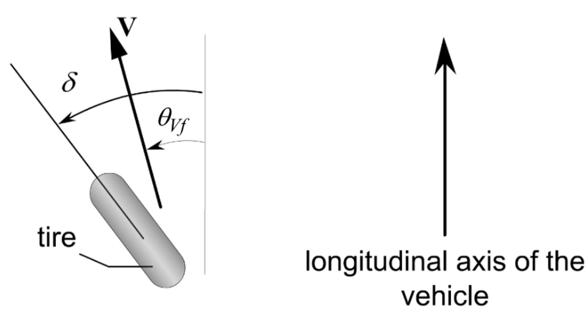


Figure 2: Tire slip-angle[2]

Here you will use the bicycle model for the vehicle, which is a popular model in the study of vehicle dynamics. Shown in Figure 1, the car is modeled as a two-wheel vehicle in two degree of freedom, described in longitudinal and lateral dynamics separately. The model parameters are defined in Table 1.

## 2.1 Lateral dynamics

Ignoring road bank angle and applying Newton's second law of motion along the y axis

$$ma_y = F_{yf} \cos \delta_f + F_{yr}$$

where $a_y = \left(\frac{d^2y}{dt^2}\right)_{inertial}$ is the inertial acceleration of the vehicle at the center of geometry in the direction of the y axis, $F_{yf}$ and $F_{yr}$ are the lateral tire forces of the front and rear wheels respectively and $\delta_f$ is the front wheel angle which will be denoted as $\delta$ later. Two terms contribute to $a_y$: the acceleration $\ddot{y}$ which is due to motion along the y axis and the centripetal acceleration . Hence

$$a_y = \ddot{y} + \dot{\psi}\dot{x}$$

Combining the two equations, the equation for the lateral translational motion of the vehicle is obtained as

$$\ddot{y} = -\dot{\psi}\dot{x} + \frac{1}{m}(F_{yf}\cos\delta + F_{yr})$$

Moment balance about the axis yields the equation for the yaw dynamics as

$$\ddot{\psi} I_z = l_f F_{yf} - l_r F_{yr}$$

The next step is to model the lateral tire forces $F_{yf}$ and $F_{yr}$. Experimental results show that the lateral tire force of a tire is proportional to the slip-angle for small slip-angles when vehicle's speed is large enough, let's say when $\dot{x} \geq 0.5$ m/s. The slip angle of a tire is defined as the angle between the orientation of the tire and the orientation of the velocity vector of the vehicle. the slip angle of the front and rear wheel is

$$\alpha_f = \delta - \theta_{Vf}$$
$$\alpha_r = -\theta_{Vr}$$

where $\theta_{Vp}$ is the angle between the velocity vector and the longitudinal axis of the vehicle, for $p \in \{f, r\}$. A linear approximation of the tire forces are given by

$$F_{yf} = 2C_\alpha \left(\delta - \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}}\right)$$
$$F_{yr} = 2C_\alpha \left(-\frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}\right)$$

where $C_\alpha$ is called the cornering stiffness of tires. If $\dot{x} < 0.5$ m/s, we just set $F_{yf}$ and $F_{yr}$ both to zeros.

## 2.2 Longitudinal dynamics

Similarly, a force balance along the vehicle longitudinal axis yields

$$\ddot{x} = \dot{\psi}\dot{y} + a_x$$
$$ma_x = F - sign(\dot{x})F_f$$
$$F_f = fmg$$

where $F$ is the total tire force along x axis, $F_f$ is the force due to rolling resistance at the tires, and $f$ is the friction coefficient. *sign* function returns +1 when $\dot{x} \geq 1$ otherwise -1.

### 2.3 Global coordinates

In the global frame we have

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi$$
$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi$$

### 2.4 System equation

Gathering all the equations, if $\dot{x} \geq 0.5$ m/s we have:

$$\begin{aligned}
\ddot{y} &= -\dot{\psi}\dot{x} + \frac{2C_\alpha}{m}(\cos\delta\left(\delta - \frac{\dot{y}+l_f\dot{\psi}}{\dot{x}}\right) - \frac{\dot{y}-l_r\dot{\psi}}{\dot{x}}) \\
\ddot{x} &= \dot{\psi}\dot{y} + \frac{1}{m}(F - fmg) \\
\ddot{\psi} &= \frac{2l_fC_\alpha}{I_z}\left(\delta - \frac{\dot{y}+l_f\dot{\psi}}{\dot{x}}\right) - \frac{2l_rC_\alpha}{I_z}\left(-\frac{\dot{y}-l_r\dot{\psi}}{\dot{x}}\right) \\
\dot{X} &= \dot{x}\cos\psi - \dot{y}\sin\psi \\
\dot{Y} &= \dot{x}\sin\psi + \dot{y}\cos\psi
\end{aligned}$$

otherwise since the lateral tire forces are zeros we only consider the longitudinal model.

### 2.5 Measurements

The observable states are with some Gaussian noise $\epsilon = N(0,\sigma)$, where

$$y = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ X \\ Y \\ \psi \end{bmatrix} + \epsilon,\ \sigma = \begin{bmatrix} 0.5 & & & \cdots & & 0 \\ & 0.5 & & & & \\ & & 0.05 & & & \\ \vdots & & & 0.05 & & \\ & & & & 1 & \\ 0 & & & & & 0.5 \end{bmatrix}$$

.

### 2.6 Physical constraints

The system satisfies the constraints that:

$$\begin{aligned}
&|\delta| \leqslant \tfrac{\pi}{6}\ rad/s \\
&|\dot{\delta}| \leqslant \tfrac{\pi}{6}\ rad/s \\
&|F| \leqslant 10000\ N \\
&0 \text{ m/s} \leqslant \dot{x} \leqslant 100 \text{ m/s} \\
&|\dot{y}| \leqslant 10 m/s
\end{aligned}$$

Table 1: Model parameters.

| Name | Description | Unit | Value |
|---|---|---|---|
| $(\dot{x}, \dot{y})$ | Vehicle's velocity along the direction of vehicle frame | m/s | State |
| $(X, Y)$ | Vehicle's coordinates in the world frame | m | State |
| $\psi$, $\dot{\psi}$ | Body yaw angle, angular speed | rad | State |
| $\delta$ or $\delta_f$ | Front wheel angle | rad | State |
| $\dot{\delta}$ | Steering Rate | rad | Input |
| $F$ | Total input force | N | Input |
| $m$ | Vehicle mass | kg | 1000 |
| $l_r$ | Length from front tire to the center of mass | m | 1.7 |
| $l_f$ | Length from front tire to the center of mass | m | 1.1 |
| $C_\alpha$ | Cornering stiffness of each tire | N | 15000 |
| $I_z$ | Yaw intertia | kg m^2 | 3344 |
| $F_{pq}$ | Tire force, $p \in \{x, y\}$,$q \in \{f, r\}$ | N | Depend on input force |
| $m$ | vehicle mass | Kg | 2000 |
| $f$ | Friction coefficient | 1 | 0.01 |

# 3 Resources

## 3.1 Buggy Simulator

A Buggy Simulator designed in python has been provided along with the assignment. The simulator takes the control command [steering, longitudinal Force] and then generates the buggy state after the given fixed time step (fixed fps) as output. Additional script `util.py` contains functions to help you design and execute the controller. Please design your controller in `controller.py`. After the complete run, a response plot is generated by the simulator. This plot contains visualization of the buggy trajectory and variation of states with respect to time.

## 3.2 Trajectory Data

The trajectory is given in `buggyTrace.csv`. It contains the coordinates of the trajectory: $(x, y)$. The satellite map of the track is shown in Figure 3.

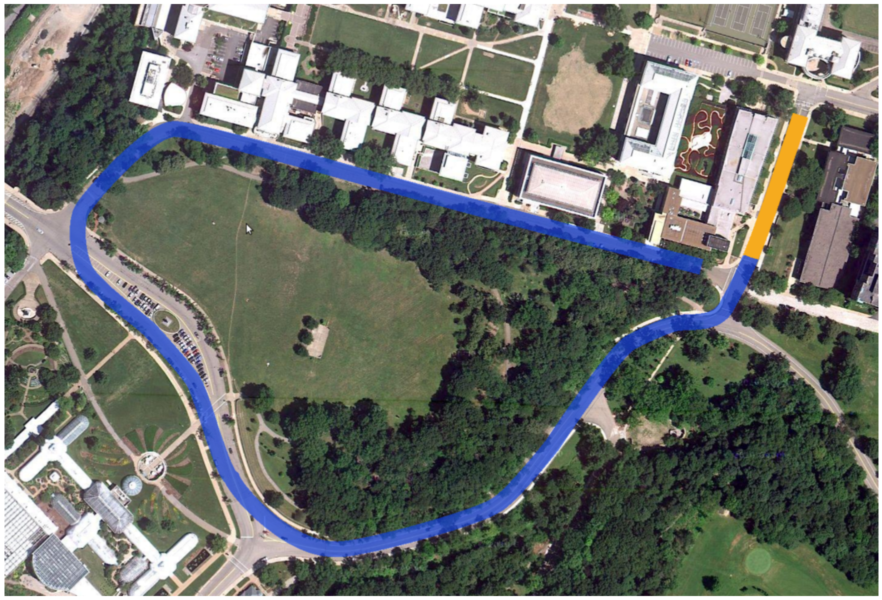

Figure 3: Buggy track[3]

# 4 P1:Problems [Due 5:00 PM, November 12]

**Exercise 1.** Model Linearization
As mentioned in class, model linearization is always the first step for non-linear control. During this assignment, you will approximate the given model with a linear model.
Since the longitudinal term $\dot{x}$ is non-linear in the lateral dynamics, we can simplify the controller by controlling the lateral and longitudinal states separately. You are required to write the system dynamics in linear forms as $\dot{s_1} = A_1 s_1 + B_1 u$ and $\dot{s_2} = A_2 s_2 + B_2 u$ in terms of the following given input and states:

$$u = \begin{bmatrix} \delta \\ F \end{bmatrix}, s_1 = \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix}, s_2 = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

**Exercise 2. Introduction to the Simulator.** The Buggy Simulator has been implemented as a python code in the `BuggySimulator.py`, takes the steering, brake and longitudinal force as input.
For this question, you have to design a PID longitudinal controller and a PID lateral controller for the vehicle. PID is an error based controller, that requires tuning of the proportional, derivative and integral gains.
Design the controllers in `controller.py`.
[You have to edit only the controller.py python script]
Execute the `main.py` python script to check your controller. It generates a performance plot and saves the vehicle states in a .npz file. Submit the Buggy-states and the response plot with the code.
Your controller is required to achieve the following performance criteria:

1. Time to complete the loop = 450 s
2. Maximum deviation from the reference trajectory = 8.0 m
3. Average deviation from the reference trajectory = 4.5 m

# 5 Reference


1. Rajamani Rajesh. Vehicle dynamics and control. Springer Science & Business Media, 2011.

2. Kong Jason, et al. "Kinematic and dynamic vehicle models for autonomous driving control design." Intelligent Vehicles Symposium, 2015.

3. cmubuggy.org, `https://cmubuggy.org/reference/File:Course_hill1.png`