Vehicle Dynamics and intro to Controls

Rohan Sinha, rohan.sinha@berkeley.edu, Goldeneye.

October 2017

1 Introduction

For the initial simulation for Goldeneye's automated driving project, we will be using a simplified model of a car to keep things as simple as possible. Here the dynamics of a car are represented by a bicycle model, which can be thought of as compressing a car laterally until the left and right of the vehicle coincide. Our approach to designing control systems for this highly non-linear model will be to simplify and or linearize this model based on what we are trying to accomplish, and then to tune the gains to our controller based on the simplification.

We can then verify how well our system works by applying it to the actual model, and eventually, our goal is to model the vehicle on the Ubuntu virtual machine. This will give us a good "Car on a USB-stick" setup with many uses. To illustrate this point for those who have little control theory exposure, this introduction will provide the general idea. A linear dynamical system in canonical form is governed by a key (matrix) ODE:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

Here x is the vector with all state parameters, A and B are matrices of coefficients, and u is a vector with the control inputs (the things we can change). Disregarding noise and disturbances, we want to drive some of these parameters to a certain value, like a cruise controller making a car go 60 mph on the highway. To do this, we come up with a "Control Rule" which sets u to a certain value, based upon the desired value (r) of our state and the current actual/measured state. To keep things simple, we will use Proportional Integral Derivative Control, or PID control to do this. The equation for u is here:

$$u = K_P \cdot e + K_d \cdot \dot{e} + K_I \cdot \int_0^t e dt \tag{2}$$

$$e = r - x \tag{3}$$

The hard part here is to find values for the K's that accomplish this in the way we desire. Given a dynamical system governed by a diff eq of the form in equation 1, we can derive a great deal about the behavior of the system and the effects that the control inputs will have on the state of the system. The properties we can quantify can then help us to define the Ks in terms of the coefficients in the matrices A and B.

Sadly, real life is not as easy, and our vehicle's behavior cannot be described by an equation of the form of equation 1. In general, our non-linear system is described by:

$$\dot{x}(t) = f(x(t), u(t)) \tag{4}$$

Where f can be any random equation. To deal with this we simplify our function f with a specific goal in mind, and then we use the simplification to define our Ks. To find out how well this worked, we first compare our simplified behavior with the real behaviour (in this case this is a simplified bicycle model), and we apply our control law to the real system to see how well it works.

The rest of this document outlines the dynamics of our full system, think of it as our f(x,u). Which parameters are x and which are u, really depend on what you're trying to accomplish. So as a specific introduction to vehicle controls, you will all individually make a Simulink model of this dynamical system. This will make you understand the dynamics better which will help us all moving forward into our specific tasks. First we will work on lane keepers and cruise controllers (longitudinal velocity), then on controllers that drive the vehicle in a specific direction and speed, and finally on predictive controllers planning motion between points in real-time. We will get the BARC car running in the next week or so, so we will then move from our bicycle model to the real thing.

2 Vehicle Dynamics

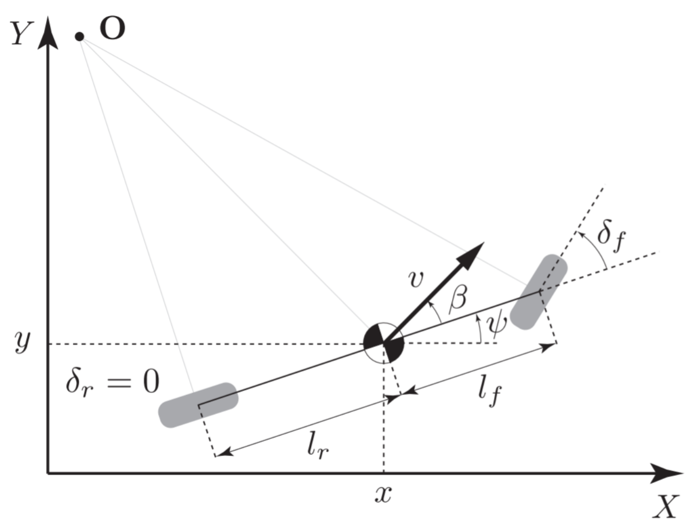

Figure 1: Dynamic Bicycle Model

We will formulate the dynamic equations of the vehicle in components with the local x (longitudinal) axis along the body of the vehicle (positive direction is forward), and the local y (lateral) axis (positive direction is to the left). Because this local reference frame is moving, their rates of change will impact each other as a function of the yaw rate ($\dot{\psi}$). The relationship between the global coordinates X,Y and the local coordinates x,y is given by the following inverse rotation:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) \\ sin(\psi) & cos(\psi) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{5}$$

Writing down Newton's second law in the global reference frame:

$$a_X^{Global} = \ddot{x} - \dot{\psi}\dot{y} = \frac{1}{m}\Sigma F_x \tag{6}$$

$$a_Y^{Global} = \ddot{y} + \dot{\psi}\dot{x} = \frac{1}{m}\Sigma F_y \tag{7}$$

A neat little trick that lets us decouple the longitudinal drive train and friction dynamics from the lateral behaviour of the vehicle, is to split up the sum of forces in the longitudinal direction into 2 terms:

$$\ddot{x} = \dot{\psi}\dot{y} + \frac{1}{m}\Sigma F_{long} + \frac{1}{m}\Sigma F_{cornering} \tag{8}$$

The longitudinal term will be denoted with a_{long} moving forwards. This makes it possible for us to define a_{long} separately from the lateral dynamics by analyzing free body diagrams of the vehicle moving in a straight line, so we won't need to deal with lateral dynamics for cruise control, or deal with longitudinal dynamics for lane keeping.

The cornering force is now the only thing that still needs to be dealt with to describe the dynamics of the vehicle motion. The cornering force is a result of the difference between the orientation of the wheel and its direction of travel. The most simplified tire behavior is modelled by the angle between the velocity vector and the direction vector of the wheel.

$$F_c = -C_c(\beta - \delta) \tag{9}$$

Where β is the angle of the velocity vector with the x axis and δ is the angle of the wheel with the x axis. C is the cornering coefficient and is constant.

$$\beta_{front} = arctan(\frac{\dot{y} + l_f\dot{\psi}}{\dot{x}}) \tag{10}$$

$$\beta_{rear} = arctan(\frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}) \tag{11}$$

An important thing to note is that the direction of the cornering force is perpendicular to the tire it is acting on. This means that the rear tire will not affect the forces in the x direction, but the front tire will. Together with Newton's second law for rotational motion, the final dynamic equations of the bicycle become:

$$\ddot{x} = \dot{\psi}\dot{y} + a_{long} - \frac{1}{m}F_{cf}sin(\delta) \tag{12}$$

$$\ddot{y} = -\dot{\psi}\dot{x} + \frac{1}{m}(F_{cf}cos(\delta) + F_{cr}) \tag{13}$$

$$\ddot{\psi} = \frac{1}{I_z}(l_f \cdot F_{cf}cos(\delta) - l_r \cdot F_{cr}) \tag{14}$$

Where the subscripts cr and cf denote cornering force on the rear and front tire respectively.

3 Conclusion

Your goal for this week is to take this model, and implement it in Simulink. Go look into the textbooks that I previously recommended for more detailed information or try to repeat the derivations to come to the same equations as I did. You might find that many people multiply each set of forces by 2 to account for the two missing wheels, but I think it is easier to just multiply the cornering coefficients by 2. For the longitudinal dynamics we must sum the four friction forces on each wheel and the aerodynamic drag force, which can be done in a separate analysis. I will not provide you guys with knowledge on how that works, as it is very simple when you don't have any complicated powertrain dynamics. The BARC obviously uses electric motors, which are simple to model, instead of internal combustion with transmissions etc.. Steps we are going to take will be for example to linearize our bicycle model around constant velocity rectilinear motion for lane keeping, among other things. I am really looking for you guys to come up with a lot of stuff on your own, but I wanted to give some grounding for the vehicle dynamics to start with.