Lateral Dynamic Control of Vehicle

1. Introduction

With the improvement of electric vehicles (EVs) and artificial intelligence, auto-driving vehicles have become a hot topic and gradually become true. When more and more researchers lay emphasis on perception, decision, action, or even end-to-end big model. Controller always play an important role because it is the bridge from the software to the hardware. Thus, this project focuses on the control of the vehicle dynamics.

2. Description

The actions of vehicles can generally be separated into two parts, longitudinal dynamics, and lateral dynamics. While the former decides the velocity, the latter determines the stability. Compared to longitudinal dynamics, lateral dynamics is more complex in auto-driving including tracing ability, avoid ability, comfort, and stability. In this project, the stability of lateral vehicle dynamics will be discussed and be tried to become more stable.

Math Description

When it comes to vehicle dynamics, the bicycle dynamics is always introduced to reduce the complication of vehicles. Here are some basic assumptions:

- Longitudinal speed is fixed.
- Left and right wheels are simplified to a single wheel.
- Suspension motion, road inclination, and aerodynamics are ignored.

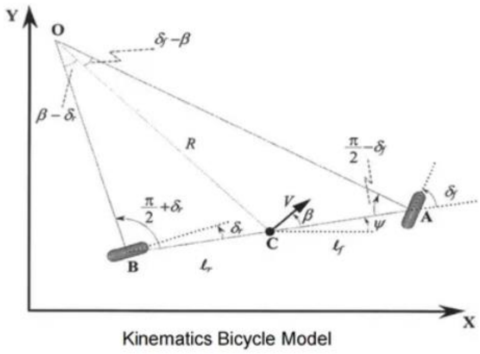

Figure 1 Kinematics Bicycle Model

In this figure, A represents the front wheels and B represents the rear wheels. δ_f, δ_r imply the steering angle of the front wheels and rear wheels. C is the center of gravity. $L = L_r + L_f$ is the wheelbase of vehicles. Ψ is the heading angle of vehicles. V is the velocity and β is the slip angle of vehicles. R is circular radius and O is instantaneous rolling center.

Based on Newton's Law and Kinematic relationships, a lateral equation of motion can be acquired:

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{2(C_{\alpha r}+C_{\alpha f})}{mV_x} & 0 & -V_x + \dfrac{2(C_{\alpha f}l_f - C_{\alpha r}l_r)}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{2(C_{\alpha f}l_f - C_{\alpha r}l_r)}{I_z V_x} & 0 & \dfrac{2(C_{\alpha f}l_f^2 + C_{\alpha r}l_r^2)}{I_z V_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{2C_{\alpha f}}{m} \\ 0 \\ \dfrac{2l_f C_{\alpha f}}{I_z} \end{bmatrix} \delta_f \tag{1}$$

, where δ_f is the steering input.

Considering a desire turning angle of the tyer, the error equation can be acquired:

$$\frac{d}{dt}\begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2(C_{\alpha f}+C_{\alpha r})}{mV_x} & -\frac{2(C_{\alpha f}+C_{\alpha r})}{m} & \frac{2(C_{\alpha f}l_f-C_{\alpha r}l_r)}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2(C_{\alpha f}l_f-C_{\alpha r}l_r)}{I_zV_x} & -\frac{2(C_{\alpha f}l_f-C_{\alpha r}l_r)}{I_z} & \frac{2(C_{\alpha f}l_f^2+C_{\alpha r}l_r^2)}{I_zV_x} \end{bmatrix}\begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{2C_{\alpha f}}{m} \\ 0 \\ -\frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix}\delta_f + \begin{bmatrix} 0 \\ -V_x+\frac{2(C_{\alpha f}l_f-C_{\alpha r}l_r)}{mV_x} \\ 0 \\ \frac{2(C_{\alpha f}l_f^2+C_{\alpha r}l_r^2)}{I_zV_x} \end{bmatrix}\dot{\varphi}_{des} \tag{2}$$

, where e_1 is the error of lateral velocity and e_2 is the error of heading angle.

This equation is the trace error state space dynamics equation, studied in this project. Some properties of the vehicle are shown below:

Mass	m (kg)	2650
Inertia	I ($kg \cdot m^2$)	3000
Distance from center of mass to front axle	a (m)	1.70
Distance from center of mass to front rear	b (m)	1.71
Lateral stiffness of the front wheels	$C_{\alpha f}$ (N/rad)	-82600
Lateral stiffness of the rear wheels	$C_{\alpha r}$ (N/rad)	-71900
The longitudinal velocity	V_x (m/s^2)	25
Rotation velocity	$\dot{\varphi}_{des}$ (rad/s)	2

Thus, the continuous state space function is:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -4.66 & 116.60 & -0.53 \\ 0 & 0 & 0 & 1 \\ 0 & -0.47 & 11.65 & -11.98 \end{pmatrix} x + \begin{pmatrix} 0 \\ 62.34 \\ 0 \\ 93.61 \end{pmatrix}\delta + \begin{pmatrix} 0 \\ -25.53 \\ 0 \\ -11.97 \end{pmatrix} \tag{3}$$

Suppose the sampling time $T_s = 0.25s$. Based on the function “c2d” in MATLAB, the discrete state space function is:

$$x(k+1) = \begin{pmatrix} 1 & 0.14 & 2.64 & 0.12 \\ 0 & 0.27 & 18.24 & 1.12 \\ 0 & -0.005 & 1.11 & 0.086 \\ 0 & -0.019 & 0.47 & 0.09 \end{pmatrix} x(k) + \begin{pmatrix} 2.17 \\ 20.31 \\ 1.35 \\ 7.73 \end{pmatrix}\delta(k) + \begin{pmatrix} 0 \\ -25.53 \\ 0 \\ -11.97 \end{pmatrix} \tag{4}$$

$$y = [0\ 0\ 1\ 0]x \tag{5}$$

An initial state is set to be: $x_0 = [0.8, 0.2, -0.6, 0]^T$. The reason for matrix C is that since the input is a steering angle, the output should also be an angle.

3. The Design of System and Results

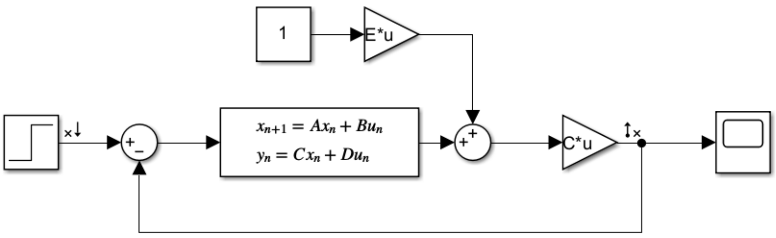

Figure 2 The System of Unit Feedback

Without any controller, the system is shown above, and the stability of the system can be shown in the Bode diagram.

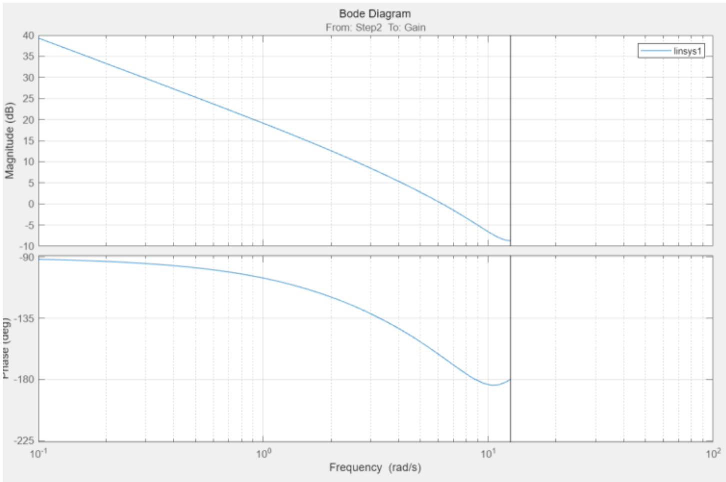

Figure 3 The Bode Diagram of Unit Feedback

Based on the bode diagram, the phase margin is about 20°. Based on the output of this error system, on one hand, the phase margin will be improved to make the system more stable. On the other hand, the overshoot and error are wished to be minimize because little error and fast response is anticipated when driving a vehicle.

With the demand, a PID controller is designed to control the system. After adjusting the parameters of P, I and D, $K_p = 0.4, K_I = 0.06, K_D = 0.1$.

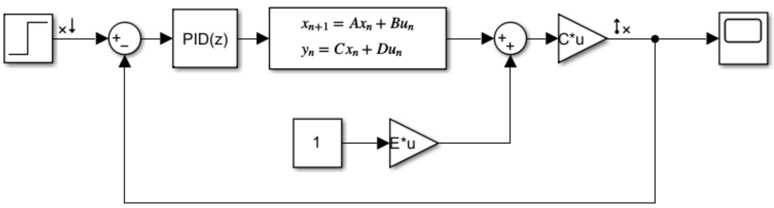

Figure 4 The System of PID Control

Compared to figure 3, the phase margin has been enhanced to about 60°. The stability has been improved and the overshoot has been decreased significantly and there is almost no fluctuation at the first time when the response reaches the input, which is a step function. This output proves that PID has great power to control and stabilize the system. However, when it comes to the physical meaning, the input is an ideal steering angle and the output is the error of steering angle, which we want the error to be zero, but PID controller is output feedback. The aim of output feedback is to enable the output to follow the input. Thus, a state feedback controller is necessary to make the output be steady at zero.

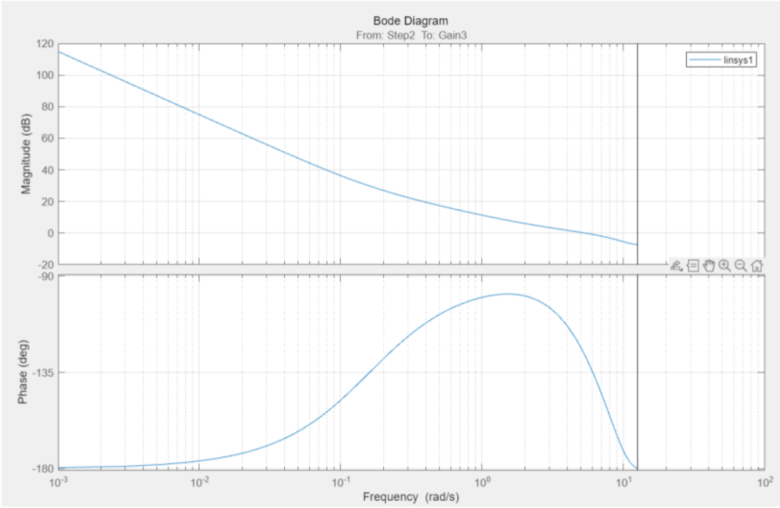

Figure 5 The Bode Diagram of Unit Feedback

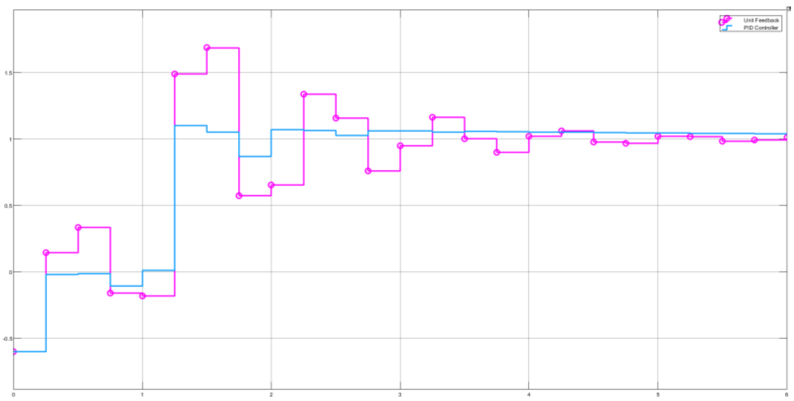

Figure 6 The Response of Unit Feedback Control and PID Control

A discrete control system based on pole placement is shown in figure 7. After adjusting the poles in z plane to make the overshoot and response time more ideal, the poles are selected as follows: $p = [0.6, 0.5, -0.12, -0.1]$.

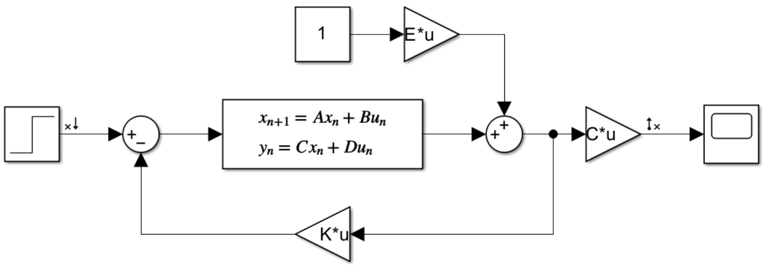

Figure 7 The System of Pole Placement

The bode diagram and the output response is shown in figure 8 and figure 9. From figure 8, a very high phase margin is shown. The phase margin is about 135°. The output keeps steady at zero. However, the overshoot and stable time is bigger than PID.

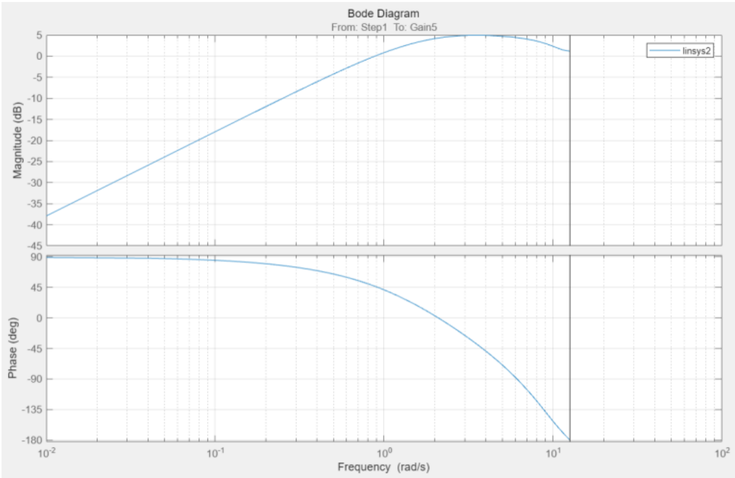

Figure 8 The Bode Diagram of Unit Feedback

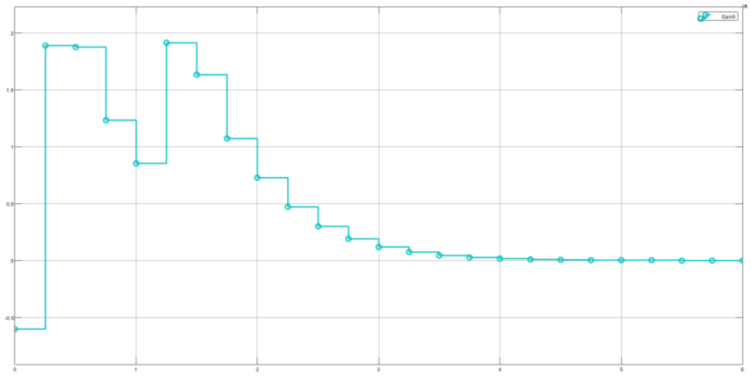

Figure 9 The Response of Pole Placement

4. Conclusion

When it comes to stability, the pole placement can greatly change the phase margin. The closer to zero the poles are in z plane; the bigger phase margin can be acquired. However, the PID controller is needed to adjust the parameter of P, I and D properly to get a bigger phase margin, because D has a positive effect on stability, but I is opposite. A balance is essential for PID.

When using PID to control a system, the overshoot and the response time is always a reference. P, I and D serve their parts separately and there are only three parameters to be changed. In pole placement, there remains many poles to be adjusted and poles are not separated from each other. It is hard to find which is the optimal poles and which poles to change in order to achieve the final goal.

A steady error has been considered in the equation of the system. In the Simulink model, it is shown as E, which is the last part of equation 4. When considering small random errors, the stability will not be changed. Only a slight difference can be observed in the figure. When it comes to some certain error, especially signal or device error, these kinds of errors can be measured. These errors can be added into the model just like the steady error in the equation.

Compared the output feedback like PID with the state feedback like pole placement. Each method has its cons and pros. Specific issues are analyzed on a case-by-case basis. In this case, if state feedback is not used, it is difficult to make the steady output become 0. Moreover, other can be more effective than pole placement. Methods like LQR and MPC are more commonly used in lateral vehicle control.

Reference

1. users.sussex.ac.uk/~tafb8/mas/MAS_03a_LateralVehicleDynamics.pdf
2. The Lateral Control of Vehicles with LQR (zhihu.com)

Appendix

```
%%%% Lateral Vehicle Dynamics
clc
clear all
%the property of vehicle
m = 2650; % mass(kg)
I = 3000; %Inertia(kg*m^2)
a = 1.70; % Distance from center of mass to front axle (m)
b = 1.71; % Distance from center of mass to  rear axle (m)
Caf = -82600; %Lateral stiffness of the front wheels (N/rad)
Car = -71900; %Lateral stiffness of the rear wheels (N/rad)
Vx = 25; %Vx is the longitudinal velocity (m/s)
Wr = 2; %rotation velocity (rad/s)

%% Initial Element
% y is the vehicle lateral postion, measured along the vehicle
% lateral axis to the point O which is the center of rotation
% Vx is the longitudinal velocity;
% yaw is the yaw angle;
% define x = [y; dot(y); yaw; dot(yaw)];  change to error state
As = [ 0, 1, 0, 0;
      0, 2*(Caf+Car)/(m*Vx), -2*(Caf+Car)/m, 2*(Caf*a-Car*b)/(m*Vx);
      0, 0, 0, 1;
      0, 2*(Caf*a-Car*b)/(I*Vx) ,-2*(Caf*a-Car*b)/I,
2*(Caf*b^2+Car*a^2)/(I*Vx)];
Bs = [0; -2*Caf/m; 0; -2*a*Caf/I];
Es = [0; 2*(Caf*a-Car*b)/(m*Vx)-Vx;0;2*(Caf*a^2+Car*b^2)/(I*Vx)];
E = Es * Wr;
Cs = [0 0 1 0];
Ds = zeros(size(Cs,1), size(Bs,2));
% initial state
x0 = [0.8;0.2;-0.6;0];
%% turn the continuous part into discrete and judge the stability of the
initial state
Ts = 0.25; %sample frequency(s)
sys = ss(As,Bs,Cs,Ds);
G = tf(sys);
p = pole(G);
[GM, PM, Wcg, Wcp] = margin(G);
```

```
sys_d  = c2d(sys,Ts,'zoh');
% get discrete A,B,C
A = sys_d.A;
B = sys_d.B;
C = sys_d.C;
D = sys_d.D;
[numerator, denominator] = tfdata(sys, 'v');
G_d = tf(sys_d)
p_d = pole(G_d)
[GM, PM, Wcg, Wcp] = margin(G_d)
%% Use PID to make the system more stable
Kp = 0.4;
Ki = 0.06;
Kd = 0.1;
%% pole placement
p0 = [0.7,0.4,-0.7,-0.3];
K = place(A,B,p0);
```