Vehicle Yaw Stability Simulations

varying speed

varying road conditons: frictions of icy wet and dry

Brian Lesko, Graduate Researcher and Teaching Associate, Masters of mechanical engineering student

Robust Controls 8352 | Ohio State University | 12/15/2022

with use of Robust Controls of Mechatronic Systems by Dr. Levent Guvenc

Building off of previous work, titled: Lateral Vehicle Dynamics Modeling and Simulation

```
clc; clear; close all
```

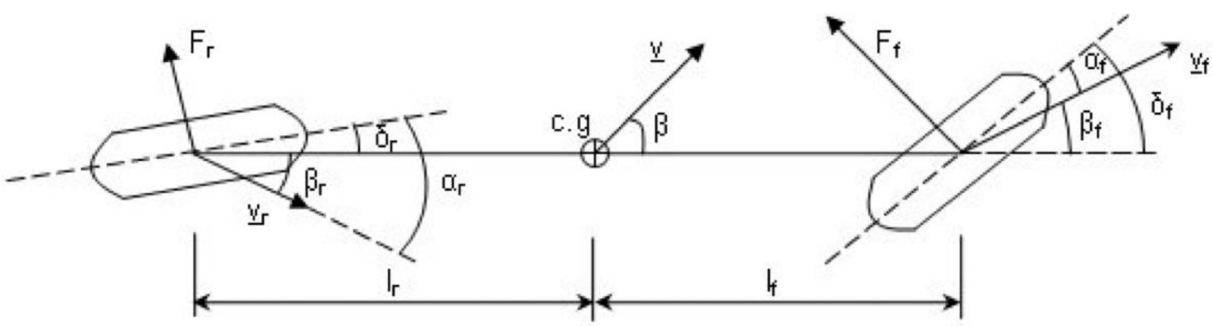
Background

Single Track Model (Bicycle) (with 3 degrees of freedom)

Valid for lateral accelerations smaller than 0.3-0.4 g - constant longitudinal vehicle speed and small side slip angle. This model has "lumped" axles and thus only models one front and one rear wheel.

for a model in the x y plane: roll is neglected (about x), pitch is neglected (about y)

longitudinal forces are neglected, but can be added.



with states described by: Vehicle Side Slip Angle, Beta, Vehicle center of gravity speed, V, and Yaw rate, r.

The model is used in practice as a linearized version which holds speed constant, and thus necessitates different models for different speeds.

Overall, the model yields the following transfer function of front wheel steering angle to yaw rate

Understeer gradient

Understeer is a vehicle's intentional tendency to turn less sharply than the change in steering wheel input from the driver when at higher steering angles - it is intuitively easy to control so vehicles are designed with an understeer. this understeer occurs as a gradient. During understeering, the side slip angle for the front tires are greater than the rear tires

Oversteer is the opposite and is more difficult to control, and hence is not a desirable characteristic. The side slip angle for the rear wheels during oversteering is greater than the rear wheel side slip angle.

Yaw Velocity Gain

Yaw velocity gain is used to compare the steering response of vehicles. It is **the ratio of the steady state yaw velocity to the steer angle**. where the steady state yaw velocity Ωz is the ratio of the longitudinal speed to the turning radius R

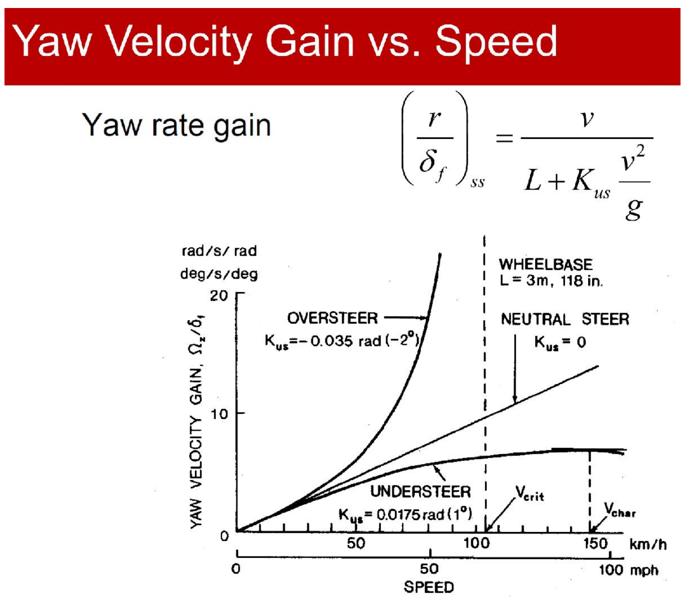
WHY ARE THE UNITS THIS WAY?

where yaw velocity gain has units of deg/s/deg (or radians)

meaning degrees the vehicle turns per degree of steering angle change?

or

degrees of wheel angle (side slip angle?) per second per degree of steering angle?



Vehicle Parameters

Vehicle Model Parameters

Parameter Name	Symbol	Value [Units]	Description
Mass of the vehicle	M	2,000 [kg]	Vehicle mass with four passengers
Gravitational constant	g	9.8 [m/s^2]	Gravitational acceleration
Vehicle base	L	2.8 [m]	Distance between the axles
Front half length	l_f	1.3 [m]	Distance from the center of gravity of the vehicle (CG) to the front axle
Rear half length	l_r	1.5 [m]	Distance from the center of gravity of the vehicle (CG) to the rear axle
Cornering stiffness front	C_f	1.2e5 [N/rad]	Cornering stiffness of front tires
Cornering stiffness rear	C_r	1.9e5 [N/rad]	Cornering stiffness of rear Tires
Inertia moment around z axis	I_z	3,700 [kgm^2]	Yaw moment of inertia

```
L=2.85; % Distance between the axles [m]
g=9.81; %
Lr = 1.55;% Distance from the center of gravity of the vehicle (CG) to the rear axle
Lf = 1.3; % Distance from the center of gravity of the vehicle (CG) to the front axle
Cf = 3e5; % Cornering Stiffness of Front Tires x2
Cr = 3e5; % Cornering Stiffness of Rear Tires x2
Cs = 1.5e5; % Cornering Stiffness
m = 2000; %Mass of the vehicle [kg]
J = 3700; %Yaw moment of Inertia
mu = 0.7; %Dry coefficient of Friction
R = 0.3; % Wheel radius
Vref = 20; % Constant vehicle velocity.
alphaf = 0.1; % Steering wheel angle rad/sec

%Linear Parameters Calculation
a11 = -(Cr+Cf)/(m*Vref);
a12 = -1-((Cf*Lf-Cr*Lr)/(m*Vref^2));
a21 = (Lr*Cr-Lf*Cf)/J;
a22 = -((Cf*Lf^2)+(Cr*Lr^2))/(Vref*J);
b11 = Cf/(m*Vref);
b12 = Cr/(m*Vref); %delta_r parameter
b21 = Cf*Lf/J;
b22 = Cr*Lr/J; %delta_r parameter
e2 = 1/J; % For yaw moment term
```

The Understeer Gradient, Kus can be calculated from

```
Kus_vehicle = (Lr/Cf/L - Lf/Cr/L)*m*g
```

```
Kus_vehicle = 0.0057
```

Plotting the Vehicle's Yaw Velocity Gain vs speed from the gradient

```
%Oversteer, Yaw Velocity
Kus = -0.035;
```

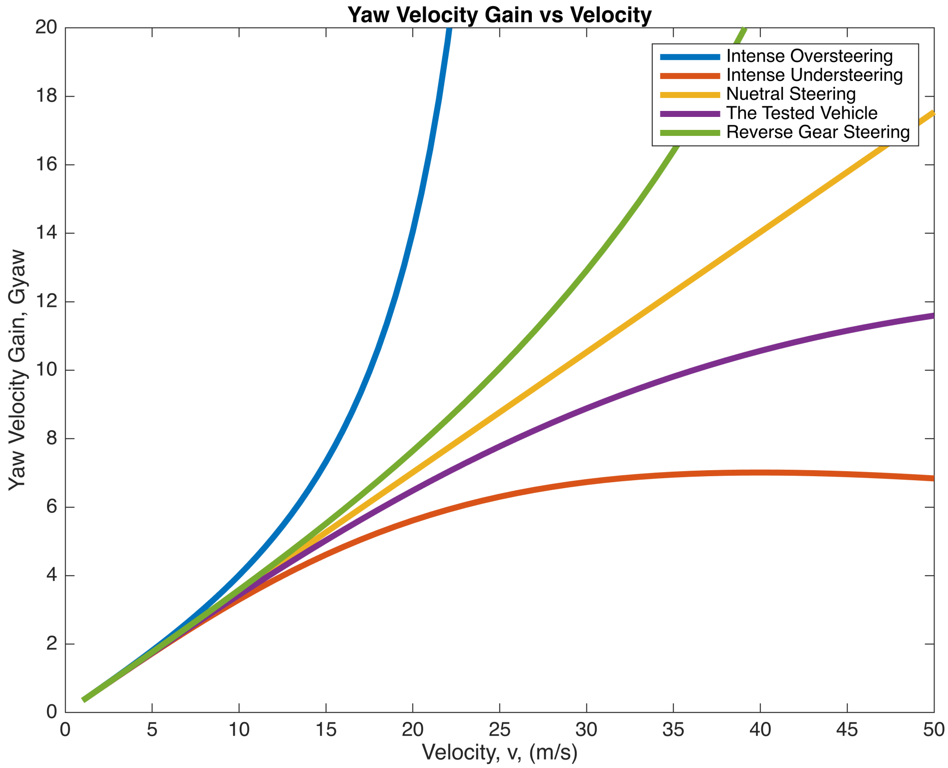
```
v_critical = sqrt(-L*g/Kus);
v = 1:0.5:25; %velocity
Gyaw_O = v./(L+Kus*v.^2/g);
plot(v,Gyaw_O,'LineWidth',3), hold on

%Understeer, Yaw Velocity
Kus = 0.0175; %SAME
v = 1:1:50; %velocity
Gyaw_U = v./(L+Kus*v.^2/g);
plot(v,Gyaw_U,'LineWidth',3)

%Nuetral Vehicle
Kus_nuetral = 0;
Gyaw_nuetral = v./(L+Kus_nuetral*v.^2/g);
plot(v,Gyaw_nuetral,'LineWidth',3)

%The given Vehicle
Gyaw = v./(L+Kus_vehicle*v.^2/g);
plot(v,Gyaw,'LineWidth',3)

%Reverse Gear Steering
% The vehicle has rear wheel steering. Understeering gradient for reverse gear
% steering can be calculated by switching Cf and Cr & switching Lf and Lr
Cf_r = Cr;
Cr_r = Cf;
Lf_r = Lr;
Lr_r = Lf;
Kus_reverse = (Lr_r/Cf_r/L - Lf_r/Cr_r/L)*m*g; %static yaw rate gain under steady stat
Gyaw = v./(L+Kus_reverse*v.^2/g);
plot(v,Gyaw,'LineWidth',3)
xlabel('Velocity, v, (m/s)'),ylabel('Yaw Velocity Gain, Gyaw'),title('Yaw Velocity Gai
legend('Intense Oversteering','Intense Understeering','Nuetral Steering','The Tested V
axis([0 50 0 20])
```



Note that most vehicles are designed to be understeering for ease of control. The test vehicle included. Notice how the reverse gear steering is slightly oversteered, which makes control of reverse gear less intuitive.

The Vehicle has an Understeering gradient of

```
disp(Kus_reverse)
```

```
   -0.0057
```

When in the forward gear and an Understeering Gradient of

```
disp(Kus_vehicle)
```

```
    0.0057
```

When in Reverse

The "Nominal" Linearized Single Track Vehicle Model, described by matricies for use with a State Space model block

```
% Building the appropriate matricies to describe the vehicle
ls = Kus*Vref;
An = [[a11 a12 0 0];[a21 a22 0 0];[0 1 0 0];[Vref ls Vref 0]]; % * dphi (or df, steeri
Bn = [b11 0 ; b21 0 ; 0 -Vref ; 0 -ls*Vref]; % * df pref
```

```
Cn = [[0 0 0 1];[0 1 0 0];[1 0 0 0]]; % * beta
Dn = [0 0 ; 0 0 ; 0 0]; % * df pref
```

The "Actual model", with 20% increased mass

```
m = 2000*1.2;

%Linear Parameters Calculation
a11 = -(Cr+Cf)/(m*Vref);
a12 = -1-((Cf*Lf-Cr*Lr)/(m*Vref^2));
a21 = (Lr*Cr-Lf*Cf)/J;
a22 = -((Cf*Lf^2)+(Cr*Lr^2))/(Vref*J);
b11 = Cf/(m*Vref);
b12 = Cr/(m*Vref); %delta_r parameter
b21 = Cf*Lf/J;
b22 = Cr*Lr/J; %delta_r parameter
e2 = 1/J; % For yaw moment term

% Building the appropriate matricies to describe the vehicle
ls = Kus*Vref;
A = [[a11 a12 0 0];[a21 a22 0 0];[0 1 0 0];[Vref ls Vref 0]]; % * dphi (or df, steering
B = [b11 0 ; b21 0 ; 0 -Vref ; 0 -ls*Vref]; % * df pref
C = [[0 0 0 1];[0 1 0 0];[1 0 0 0]]; % * beta
D = [0 0 ; 0 0 ; 0 0]; % * df pref
```

Disturbance Observer Parameters

The nominal plant is also used in the DOB

```
s = tf('s');
Qnum = 0.03 * s + 1; Qden = (0.01)^3*s^3 + 3*(0.01)^2*s^2 + 0.03*s + 1;
Q = Qnum / Qden;
Gn = Q;
```

Yaw Rate: with Steering Input step at t = 1, Disturbance step at t = 5

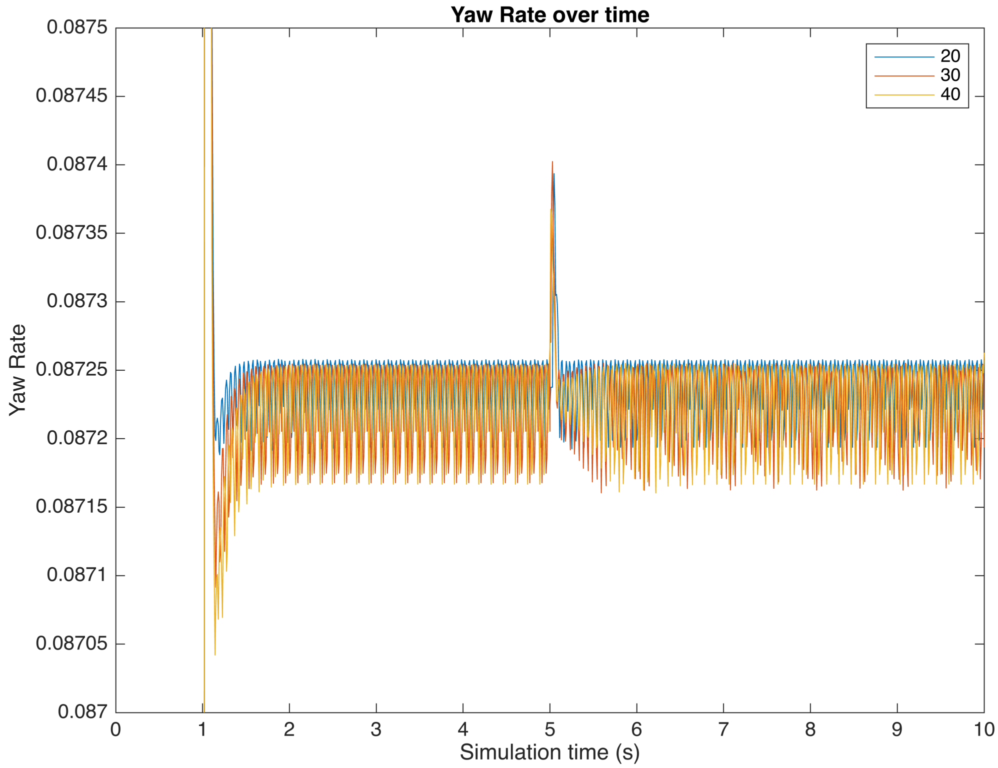
```
sim('DOB_control.slx')
yaw20 = yaw_rate;
beta20 = Beta;

% Higher Speed
Vref = 30;
% Vehicle Parameters
a11 = -(Cr+Cf)/(m*Vref);
a12 = -1-((Cf*Lf-Cr*Lr)/(m*Vref^2));
a21 = (Lr*Cr-Lf*Cf)/J;
a22 = -((Cf*Lf^2)+(Cr*Lr^2))/(Vref*J);
b11 = Cf/(m*Vref);
b12 = Cr/(m*Vref); %delta_r parameter
```

```
% Building the appropriate matricies to describe the vehicle
ls = Kus*Vref;
A = [[a11 a12 0 0];[a21 a22 0 0];[0 1 0 0];[Vref ls Vref 0]]; % * dphi (or df, steering
B = [b11 0 ; b21 0 ; 0 -Vref ; 0 -ls*Vref]; % * df pref
C = [[0 0 0 1];[0 1 0 0];[1 0 0 0]]; % * beta
D = [0 0 ; 0 0 ; 0 0]; % * df pref
sim('DOB_control.slx')
yaw30 = yaw_rate;
beta30 = Beta;

% Higher Speed
Vref = 40;
% Vehicle Parameters
a11 = -(Cr+Cf)/(m*Vref);
a12 = -1-((Cf*Lf-Cr*Lr)/(m*Vref^2));
a21 = (Lr*Cr-Lf*Cf)/J;
a22 = -((Cf*Lf^2)+(Cr*Lr^2))/(Vref*J);
b11 = Cf/(m*Vref);
b12 = Cr/(m*Vref); %delta_r parameter
% Building the appropriate matricies to describe the vehicle
ls = Kus*Vref;
A = [[a11 a12 0 0];[a21 a22 0 0];[0 1 0 0];[Vref ls Vref 0]]; % * dphi (or df, steering
B = [b11 0 ; b21 0 ; 0 -Vref ; 0 -ls*Vref]; % * df pref
C = [[0 0 0 1];[0 1 0 0];[1 0 0 0]]; % * beta
D = [0 0 ; 0 0 ; 0 0]; % * df pref
sim('DOB_control.slx')
yaw40 = yaw_rate;
beta40 = Beta;


%Yaw Rate Comparison between models
figure()
len = length(time);
plot(time,yaw20(1:len)), hold on, plot(time,yaw30(1:len)), plot(time,yaw40)
title('Yaw Rate over time'), xlabel('Simulation time (s)'),ylabel('Yaw Rate')
legend('20','30','40'), axis([0 10 .087 .0875])
```



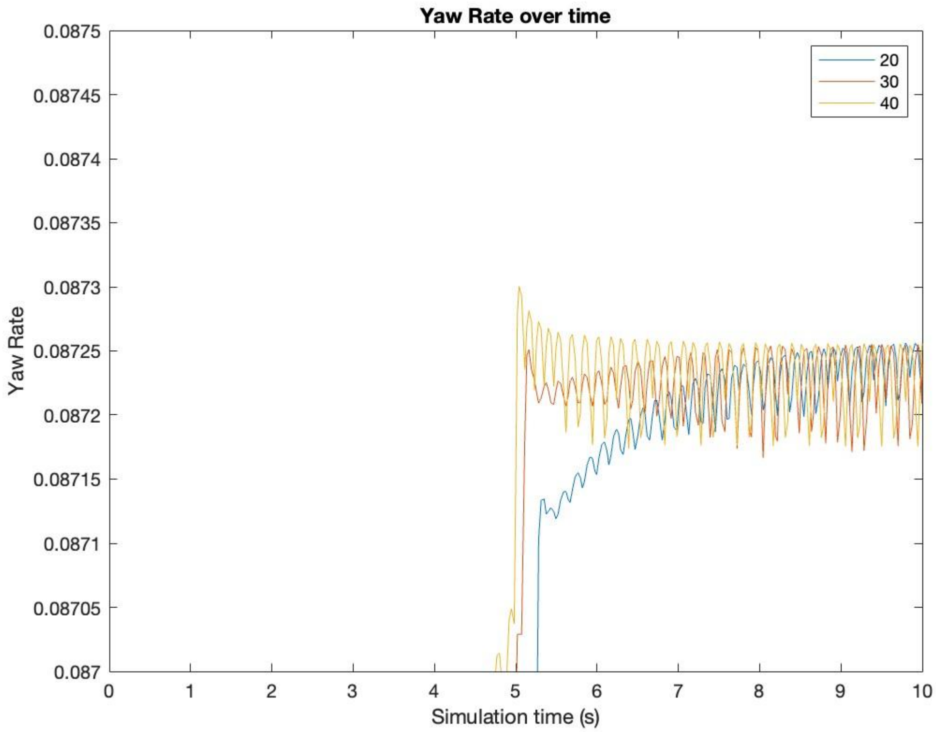
Side Slip Angle: with Steering Input step at t = 1, Disturbance step at t = 5

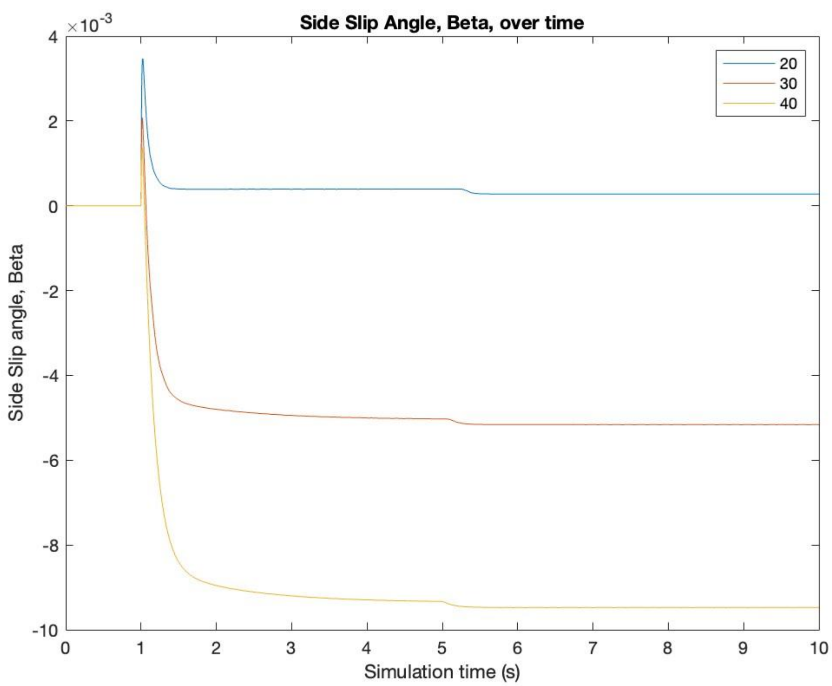
```
%Side Slip angle, Beta, Comparison between models
figure()
plot(time,beta20(1:321)), hold on, plot(time,beta30(1:321)), plot(time,beta40)
```

```
Error using plot
Vectors must be the same length.
```

```
title('Side Slip Angle, Beta, over time'), xlabel('Simulation time (s)'),ylabel('Side S
legend('Linear Model'),legend('20','30','40')
```

The Simulations with Mu = 0.15; %ice coefficient of Friction


Yaw Rate over time
Yaw Rate
Simulation time (s)
20
30
40
0.0875
0.08745
0.0874
0.08735
0.0873
0.08725
0.0872
0.08715
0.0871
0.08705
0.087
0
1
2
3
4
5
6
7
8
9
10


Side Slip Angle, Beta, over time
×10-3
Side Slip angle, Beta
Simulation time (s)
20
30
40
4
2
0
-2
-4
-6
-8
-10
0
1
2
3
4
5
6
7
8
9
10