Course Homework #5

Zhai Kexuan
Automotive Systems
11/21/19

Task 1: Transformation from local vehicle coordinates to global coordinates

1.1 Based on the lecture slides, implement the equations that transform local vehicle velocities into global inertial frame velocities, and then integrate these global velocities to obtain global positions.

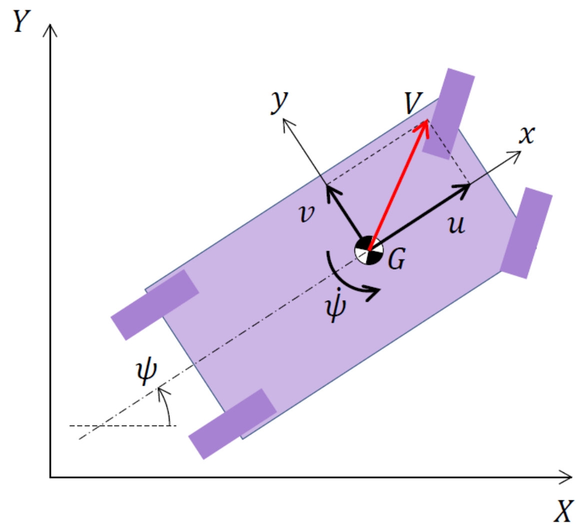

Fig 1.1 The coordinate of the vehicle

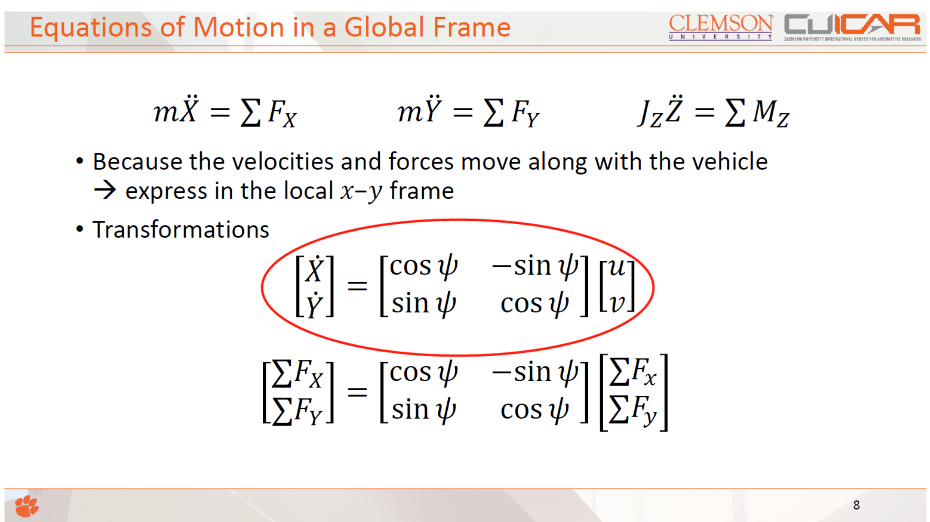

Fig 1.2 The slide from the lecture used in this specific task

Equations that transform local vehicle velocities into global inertial frame velocities
Base on the equation shown in fig 1.2, the matrix could be written as:

$$\dot{X} = u\cos\psi - v\sin\psi \tag{1}$$

$$\dot{Y} = u\sin\psi + v\cos\psi \tag{2}$$

$$\dot{\psi} = \dot{\psi} \tag{3}$$

Where u is the longitudinal velocity of the car, v is the lateral velocity, ψ is the orientation as it is shown in fig1.1.
Then, integrate global velocities in equation (1), (2) and (3) to obtain global positions.

$$X = \int (u\cos\psi - v\sin\psi)dt \tag{4}$$

$$Y = \int (u\sin\psi + v\cos\psi)dt \tag{5}$$

$$\psi = \int \dot{\psi}dt \tag{6}$$

1.2 To test whether this model component is working properly, set the inputs to constants. For instance, what do you think will happen if the local-frame forward vehicle velocity is set to a constant velocity of 10 m/s, the lateral velocity to 0 m/s and the rotational velocity to 0.1 rad/s? Perform a simulation to verify the answer to this question.

Base on the equation (4), (5) and (6), the expiration of global inertial frame velocities is shown as:

$$\psi = \int \dot{\psi}dt = \dot{\psi}t \tag{7}$$

$$X = \int u\cos\dot{\psi}dt = \frac{u}{\dot{\psi}}\sin\dot{\psi}t \tag{8}$$

$$Y = \int u\sin\dot{\psi}dt = -\frac{u}{\dot{\psi}}\cos\dot{\psi}t \tag{9}$$

Base on equation (7) and (8), the following equation is obtained:

$$X^2 + Y^2 = \left(\frac{u}{\dot{\psi}}\right)^2 \tag{10}$$

where u and $\dot{\psi}$ are constant.

Therefore, the trajectory of the car shall be a circle. The radius of the circle is:

$$R = \frac{u}{\dot{\psi}} = 100\,\mathrm{m} \tag{11}$$

1.3 Finally, set up your HW5_initialization script such that it automatically runs the Simulink model, and plots the results of the position of the vehicle in an X-Y plot.
Fig 1.3 shows the Local2Global in Simulink. Fig 1.4 shows the plot of the result. The script is shown at the end of the report.

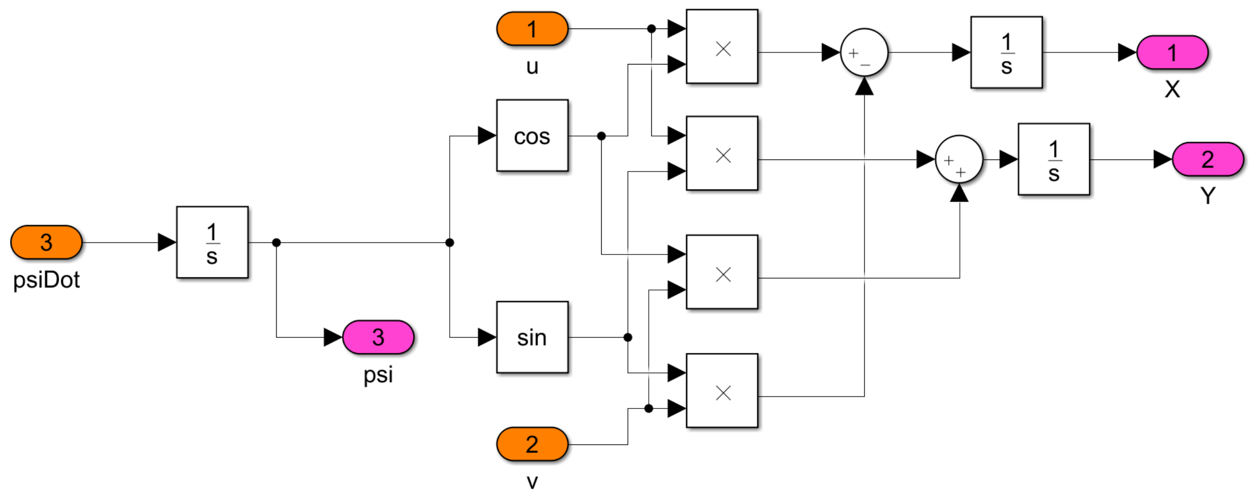

Fig 1.3 the Local2Global in Simulink

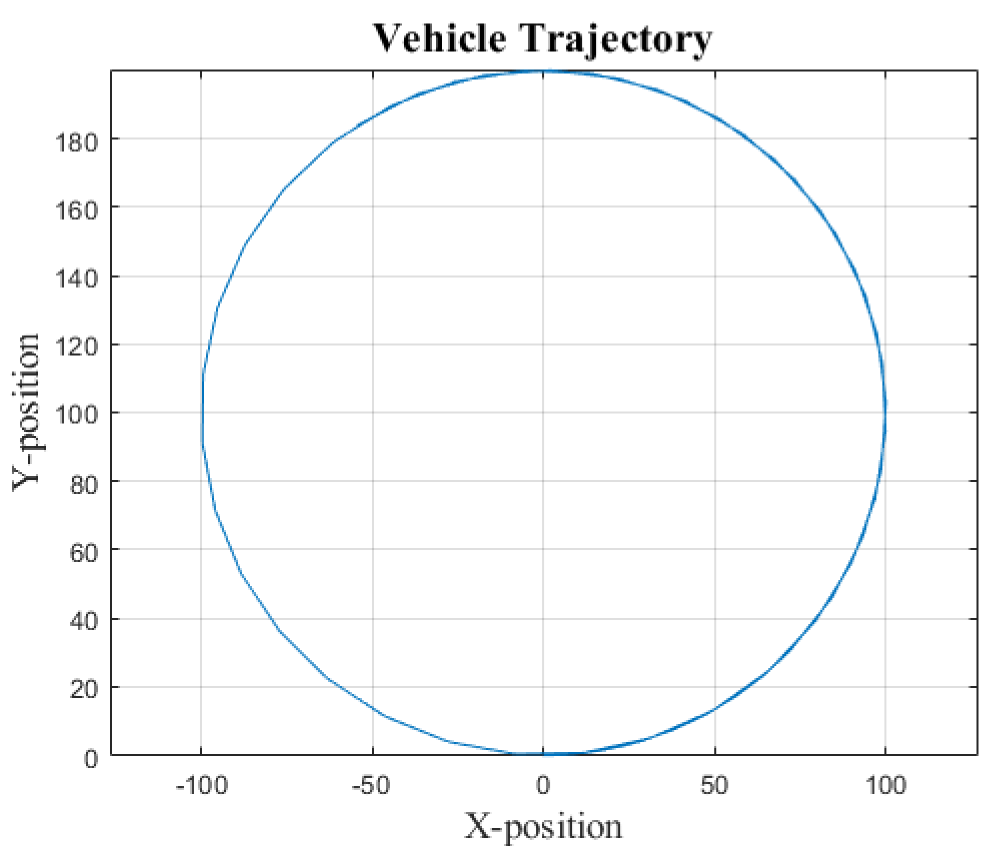

Fig 1.4 The plot of the result

1.4 Interpretation: What is the equation that expresses the (instantaneous) turning radius in terms of the (instantaneous) local forward and rotational velocities, assuming the lateral velocity to be zero? (Assume the turning radius to be measured from the

instantaneous center of rotation to the center of gravity). Does this correspond to your simulation results?

$$\text{Equation: } R = \frac{u}{\dot{\psi}}$$

this corresponds to the simulation results, as it is shown in Eq 10 and Fig 1.4. The calculation process is shown in task 1.2.

Task 2: Populating the rest of the skeleton model

2.1 Simulink Models

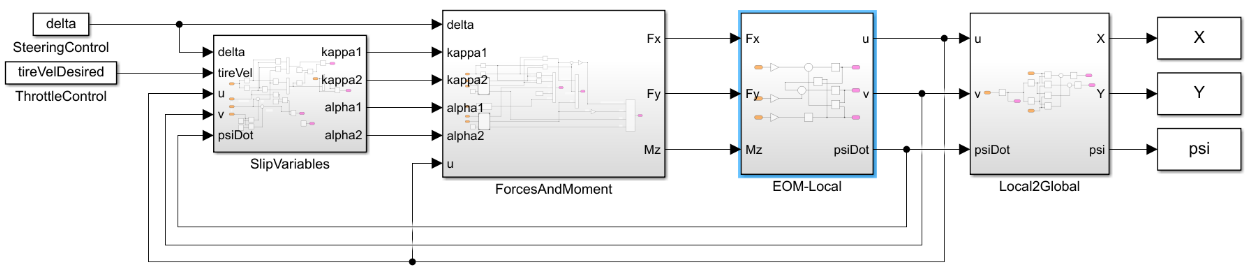

Fig 2.1 The overall system in Simulink

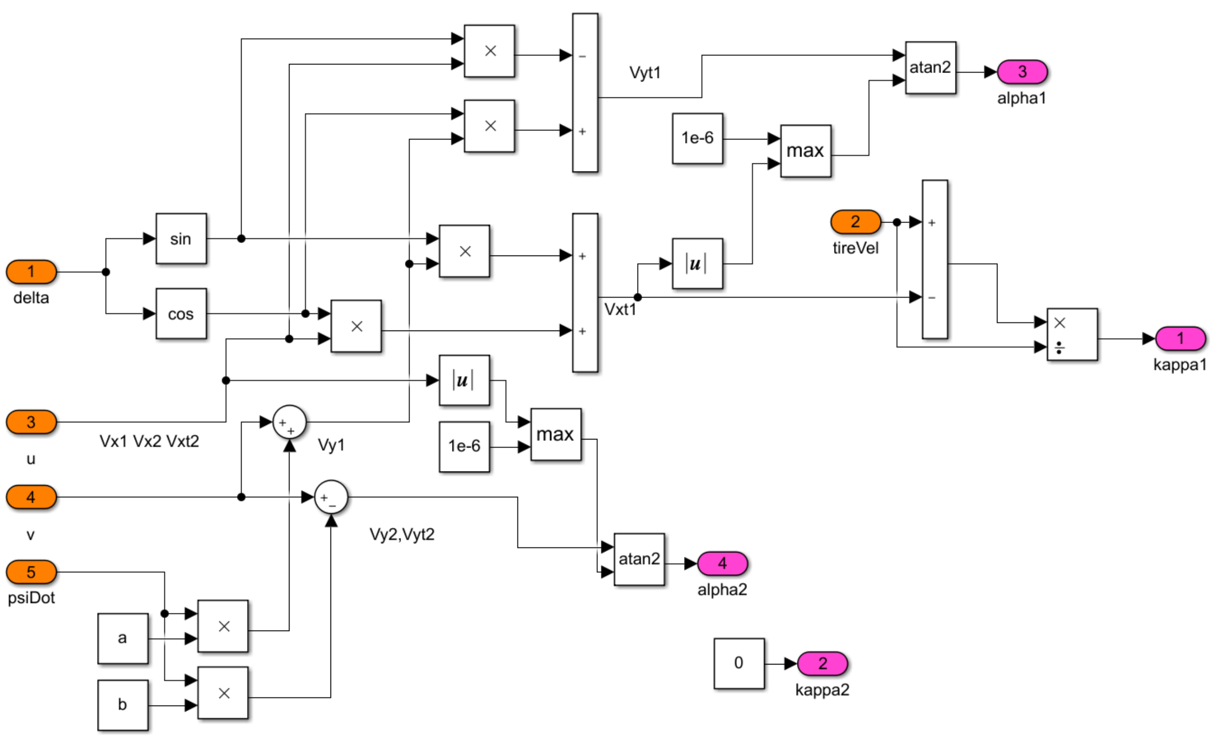

Fig 2.2 SlipVariables model in Simulink

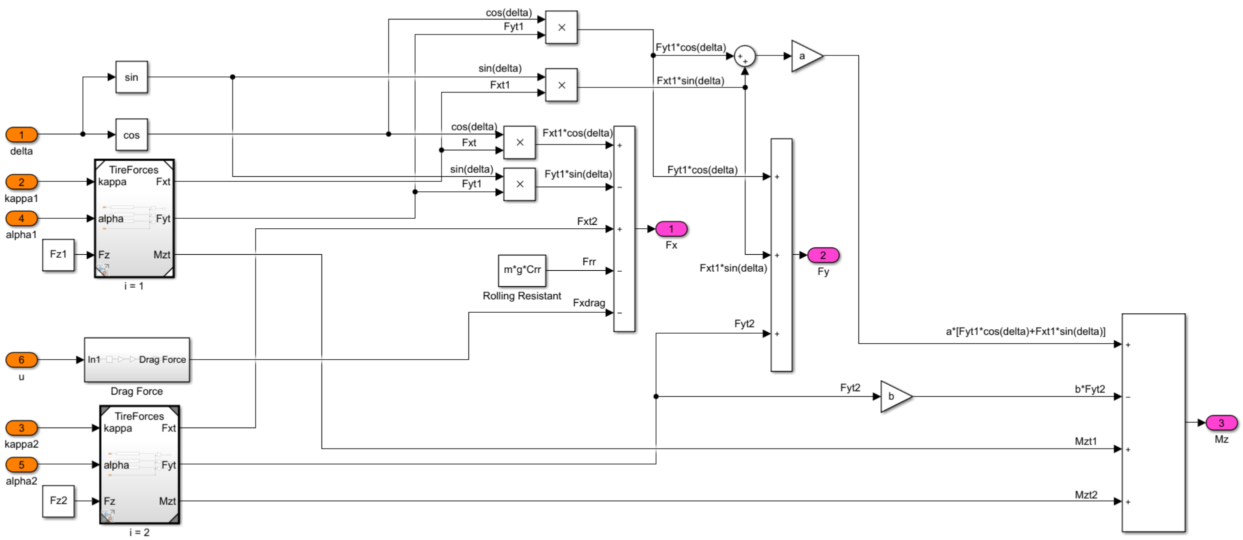

Fig 2.3 Forces and Moment model in Simulink

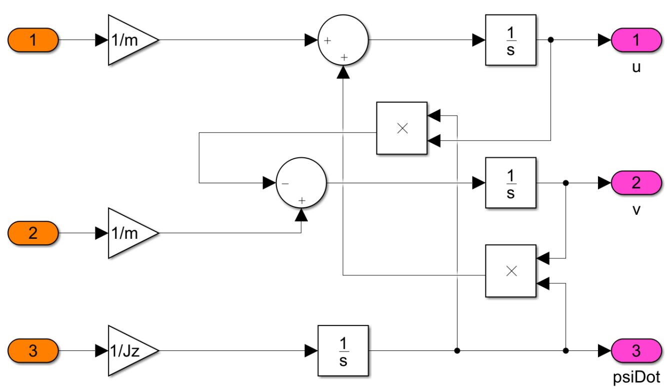

Fig 2.4 EOM-Local model in Simulink

2.2 The mathematical derivation for the initial velocities

Two circumstances are considered in this case:

Assume the initial longitudinal and lateral speeds to be zero. Therefore, the initial status of the vehicle is static, where:

$$u=0,\ v=0,\ \psi=0$$

Base on the description of the question in HW5, assume the initial lateral velocity to be zero, and the steering angle is given. The consistent set of velocities for a given steering angle is shown as:

$$u = desiredVelocity * \cos\delta$$

$$\dot{\psi} = u\frac{\tan\delta}{a+b}$$

$$v = u\dot{\psi}$$

where the desired tire speed is assumed as the front tire speed

2.3 The assumptions and corresponding explanations

- No aerodynamic downforce

According to the aerodynamic principle, resistance is generated during the driving of the car. Some of this resistance will translate into downforces. The downforce of the air on the body translates into the downforce of the tire against the ground. Therefore, the static friction limit of the tire is increased, and the cornering limit of the automobile is also increased. In this model, air resistance is considered, but the effect of the downforce is not considered.

- No longitudinal weight transfer

During acceleration or deceleration steering, the center of gravity of the car will move forward or backward. This will cause the front and rear wheels to change in grip, resulting in oversteer or understeer. In this model, the car's center of gravity is fixed by default, without taking into account the impact of the center of gravity shift on the car.

Task 3: Simulate the vehicle trajectory given different inputs

3.1 Plots of the vehicle trajectory

Fig 3.1 Vehicle trajectory for different steering angles

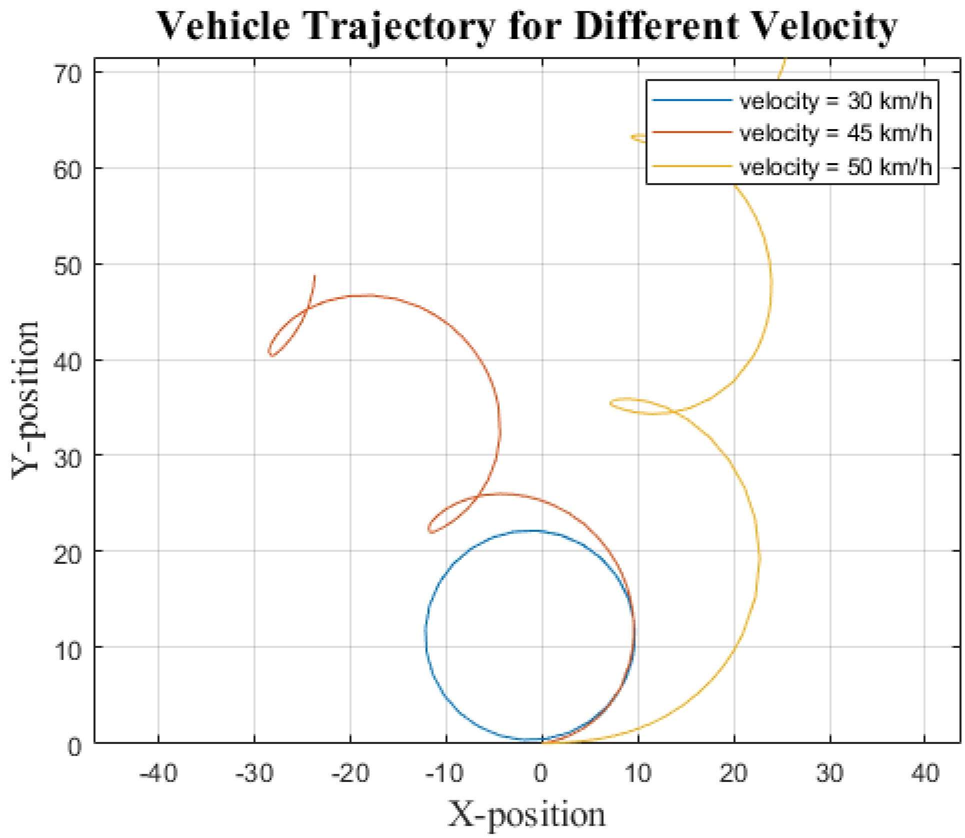

Fig 3.2 Vehicle trajectory for different velocities

3.2 Explain the behavior of the car at a constant velocity for different steering angle inputs

As is shown in fig 3.1, the radius of the trajectory decreases as the steering angle increase. In this case, the vehicle is under control. There is no slip happened.

3.3 As the velocity increases, the vehicle exhibits unusual behavior. What is this unusual behavior and why might the vehicle exhibit this behavior?

As is shown in figure 3.2, the vehicle loses control at the speed of 45 km/h and 50 km/h while it remains under control at the speed of 30 km/h. When the vehicle is at a higher speed during cornering, a higher force for lateral acceleration is required. The force is provided by the tires. When it's maximum static friction can't match the lateral acceleration, the vehicle goes understeering at first, then loses control. This is the unusual behavior that happened at speeds of 45 km/h and 50 km/h.

Task 4: Effect of changing the CG on the predicted vehicle trajectory.

4.1 plot of the vehicle trajectory for all given CG locations

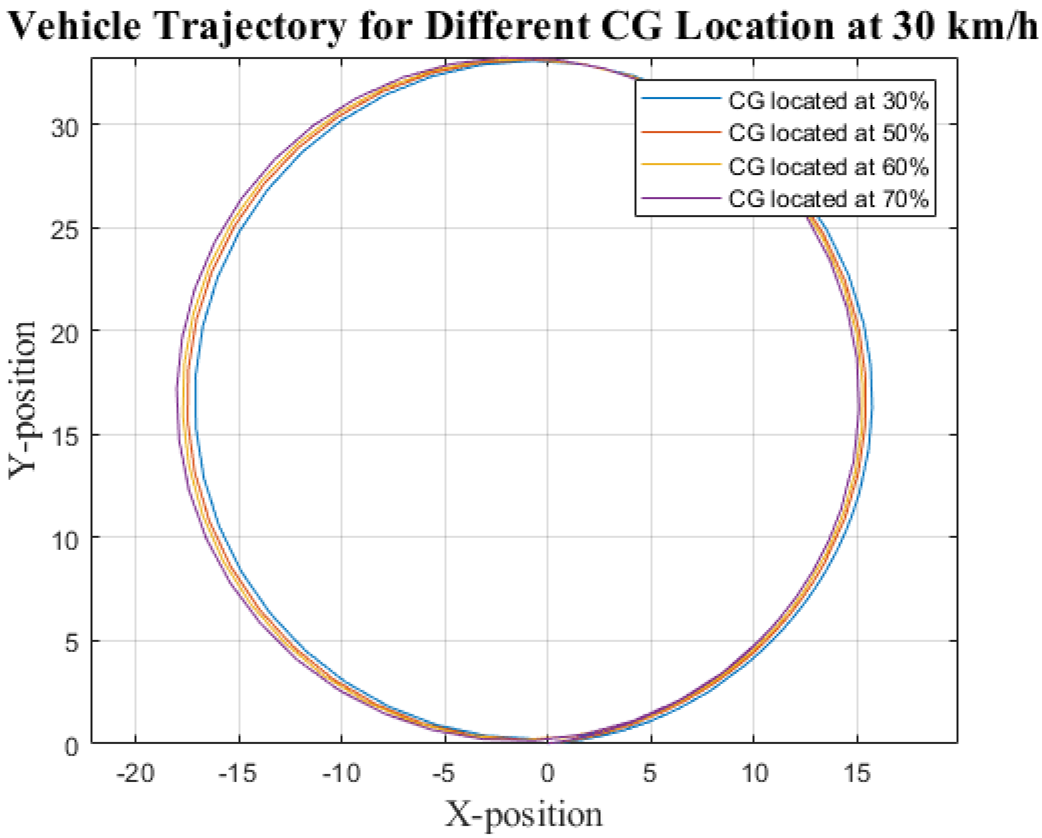

Fig 4.1 Vehicle trajectory for different CG locations at 30 km/h

4.2 How does the vehicle behavior change for the different CG locations? Provide an explanation

According to fig 4.1, as the CG moves, the trajectory changes are not obvious. There are two major reasons:

- As the CG changes, the only change is the pressure ratio of the front and rear wheels. Other factors that affect vehicle trajectory remain the same in these scenarios.
- In this case, the tires can provide enough lateral force to remain the vehicle under control during steering.

4.3 Change the vehicle velocity to 60 mph and observe the vehicle behavior for different CG locations. Which of them exhibits unusual behavior and why?

- According to fig 4.2, the vehicle can't maintain the circular trajectory of any CG point at the velocity of 60km/h. All the configurations exhibit unusual behaviors. As mentioned in task 4.2, the tires can't provide enough lateral force to remain the vehicle under control during steering in this case.
- As the curve of CG located at 30% shown, the vehicle is oversteering and loses control due to the heavy loads on the rear wheels. In contrast, the vehicle shows understeering as the CG point moves away from the rear.

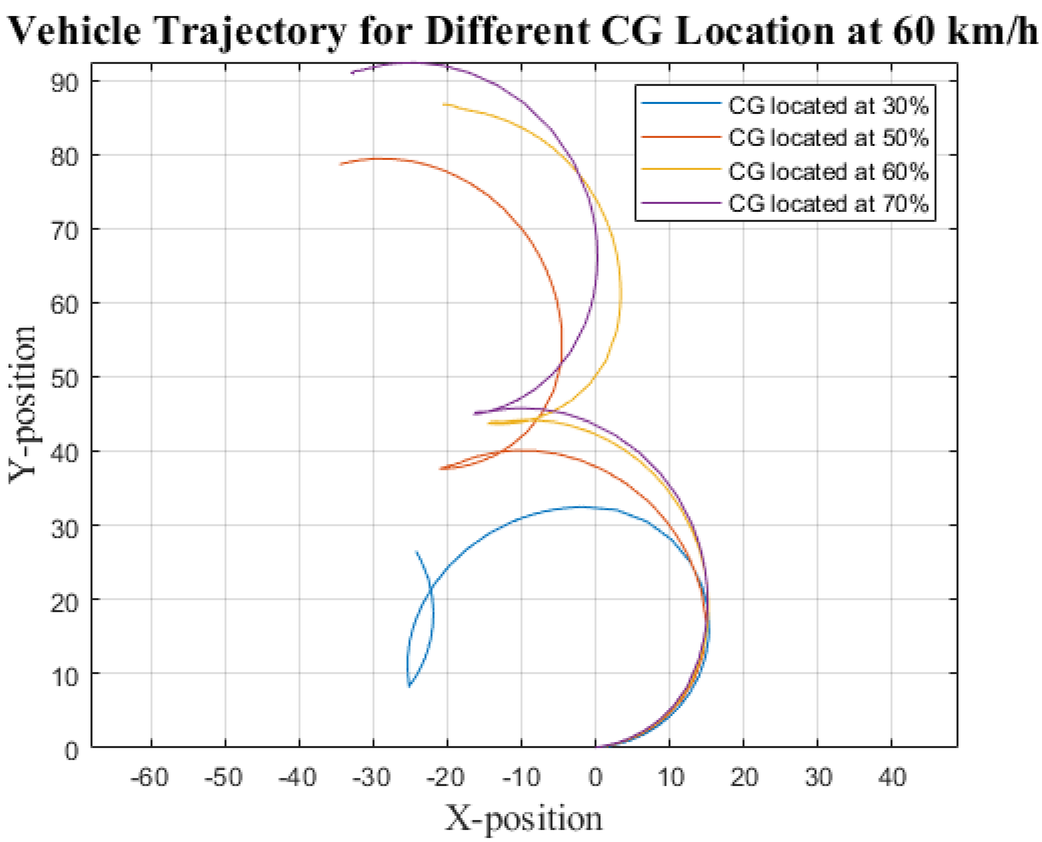

Fig 4.2 Vehicle trajectory for different CG locations at 60 km/h

Task 5: Test Result

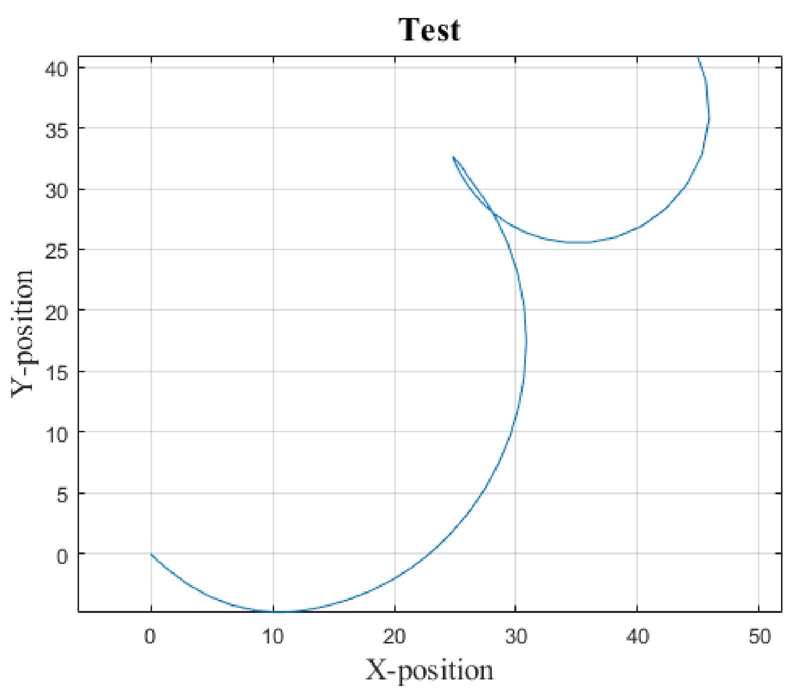

Fig 5.1 Vehicle trajectory for the test

Attachment: MatLab scripts

```
clear all;
clc;

% Drive configuration
RWD = 0;  % 1 for RWD, 0 for FWD

% vehicle kinematic and dynamic parameters.
g = 9.81;                 % graviational acceleration in
[m/s2]
m = 1800;                 % vehicle mass in [kg]
rho_air = 1.26;           % the density of air in [kg/m3]
Cd= 0.23;                 % the drag coefficient in []
Af = 2.34;                % the cross-sectional area in [m2]
Crr = 0.009;              % rolling resistance coefficient in
[]
length = 4.694;           % length of vehi in [m]
Jz = 0.9*m*(length/2)^2; % polar moment of inertia
(approximation)
wb = 2.875;               % wheel base in [m]
b = wb*0.5;               % distance to CG from rear axle in
[m]
a = wb-b;                 % distance to CG from front axle in
[m]
Fz = m*g;                 % total gravitational force in [N]
Fz1 = Fz*b/wb;            % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;           % gravitational force on rear tires
in [N]

% Tire parameters
Bx = 10;       % shape factor []
Cx = 1.9;      % stiffness factor []
Dx = 1;        % peak friction coefficient []
Ex = 0.8;      % curvature factor []

By = 12.5;     % shape factor []
Cy = 1.2;      % stiffness factor []
Dy = -1;       % peak friction coefficient []
Ey = -1;       % curvature factor []

Bz = 20;       % shape factor []
Cz = 2.4;      % stiffness factor []
Dz = 0.011;    % peak friction coefficient []
```

```
Ez = -1;      % curvature factor []

% Operation point
delta = 5*pi/180;         % steering angle in [rad]
tireVelDesired = 30/3.6; % velocity in [m/s]

% figure counting
n = 1;

%% Task 1.3

% Automatically runs the Simulink model
tireVelDesired = 30/3.6; % Keep the desired velocity at 30
km/h
iniU = 10;
iniV = 0;
iniPsiDot = 0.1;
task1_result = sim("Task1model", 100); % Export result from
Simulink

%Plots the results of the position of the vehicle in and X-
Y plot
figure(n);
n = n + 1;
plot(task1_result.X.Data , task1_result.Y.Data);
title('Vehicle Trajectory','FontName','Times New
Roman','FontWeight','Bold','FontSize',16);
xlabel('X-position','FontName','Times New
Roman','FontSize',14);
ylabel('Y-position','FontName','Times New
Roman','FontSize',14);
axis equal;
grid on;

%% Task 3 Case 1
tireVelDesired = 30/3.6; % Keep the desired velocity at 30
km/h
iniU = 0;
iniV = 0;
iniPsiDot = 0;
% Steering angle = 5°
delta = 5*pi/180;
Angle_5_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
15); % Export result from Simulink
```

```
%Plots
figure(n);
n = n + 1;
plot(Angle_5_result.X.Data , Angle_5_result.Y.Data);
axis equal;
hold on;

% Steering angle = 10°
delta = 10*pi/180;
Angle_10_result =
sim("AuE881_HW5_Zhai_Kexuan_BicycleModel", 15); % Export
result from Simulink

%Plots
plot(Angle_10_result.X.Data , Angle_10_result.Y.Data);
axis equal;
hold on;

% Steering angle = 20°
delta = 20*pi/180;
Angle_20_result =
sim("AuE881_HW5_Zhai_Kexuan_BicycleModel", 15); % Export
result from Simulink

%Plots
plot(Angle_20_result.X.Data , Angle_20_result.Y.Data);
axis equal;
hold off;

title('Vehicle Trajectory for Different Steering
Angle','FontName','Times New
Roman','FontWeight','Bold','FontSize',16);
xlabel('X-position','FontName','Times New
Roman','FontSize',14);
ylabel('Y-position','FontName','Times New
Roman','FontSize',14);
legend('Steering angle = 5°','Steering angle =
10°','Steering angle = 20°');
grid on;

%% Task 3 Case 2'
delta = 15*pi/180; % Keep the steering angle at 15°
```

```
% Desired velocity = 30 km/h
tireVelDesired = 30/3.6; % velocity in [m/s]
iniU = 0;
iniV = 0;
iniPsiDot = 0;
Velocity_30_result =
sim("AuE881_HW5_Zhai_Kexuan_BicycleModel", 15); % Export
result from Simulink

% Plot
figure(n);
n = n + 1;
plot(Velocity_30_result.X.Data ,
Velocity_30_result.Y.Data);
axis equal;
hold on;

% Desired velocity = 45 km/h
tireVelDesired = 45/3.6; % velocity in [m/s]
iniU = 0;
iniV = 0;
iniPsiDot = 0;
Velocity_45_result =
sim("AuE881_HW5_Zhai_Kexuan_BicycleModel", 15); % Export
result from Simulink

% Plot
plot(Velocity_45_result.X.Data ,
Velocity_45_result.Y.Data);
axis equal;
hold on;

% Desired velocity = 50 km/h
tireVelDesired = 50/3.6; % velocity in [m/s]
iniU = tireVelDesired;
iniV = 0;
iniPsiDot = 0;
Velocity_50_result =
sim("AuE881_HW5_Zhai_Kexuan_BicycleModel", 15); % Export
result from Simulink

% Plot
plot(Velocity_50_result.X.Data ,
Velocity_50_result.Y.Data);
```

```
axis equal;
hold off;

title('Vehicle Trajectory for Different
Velocity','FontName','Times New
Roman','FontWeight','Bold','FontSize',16);
xlabel('X-position','FontName','Times New
Roman','FontSize',14);
ylabel('Y-position','FontName','Times New
Roman','FontSize',14);
legend('velocity = 30 km/h','velocity = 45 km/h','velocity
= 50 km/h');
grid on;

%% Task 4.1
% Intialization
delta = 10*pi/180; % Keep the steering angle at 10°
tireVelDesired = 30/3.6; % Keep the desired velocity at 30
km/h
iniU = 0;
iniV = 0;
iniPsiDot = 0;
t = 15; % Set time

% CG located at 30% of the wheelbase from the rear axle.
b = wb*0.3; % distance to CG from rear axle in [m]
a = wb-b;                 % distance to CG from front axle in
[m]
Fz1 = Fz*b/wb;            % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;           % gravitational force on rear tires
in [N]
CG_30_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink


% Plot
figure(n);
n = n + 1;
plot(CG_30_result.X.Data , CG_30_result.Y.Data);
axis equal;
hold on;

% CG located at 50% of the wheelbase from the rear axle.
```

```
b = wb*0.5; % distance to CG from rear axle in [m]
a = wb-b;              % distance to CG from front axle in
[m]
Fz1 = Fz*b/wb;         % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;        % gravitational force on rear tires
in [N]
CG_50_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink

% Plot
plot(CG_50_result.X.Data , CG_50_result.Y.Data);
axis equal;
hold on;

% CG located at 60% of the wheelbase from the rear axle.
b = wb*0.6; % distance to CG from rear axle in [m]
a = wb-b;   % distance to CG from front axle in [m]
Fz1 = Fz*b/wb;         % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;        % gravitational force on rear tires
in [N]
CG_60_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink

% Plot
plot(CG_60_result.X.Data , CG_60_result.Y.Data);
axis equal;
hold on;

% CG located at 70% of the wheelbase from the rear axle.
b = wb*0.7; % distance to CG from rear axle in [m]
a = wb-b;              % distance to CG from front axle in
[m]
Fz1 = Fz*b/wb;         % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;        % gravitational force on rear tires
in [N]
CG_70_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink

% Plot
plot(CG_70_result.X.Data , CG_70_result.Y.Data);
axis equal;
```

```
hold off;

title('Vehicle Trajectory for Different CG Location at 30
km/h','FontName','Times New
Roman','FontWeight','Bold','FontSize',16);
xlabel('X-position','FontName','Times New
Roman','FontSize',14);
ylabel('Y-position','FontName','Times New
Roman','FontSize',14);
legend('CG located at 30%','CG located at 50%','CG located
at 60%','CG located at 70%');
grid on;

%% Task 4.2
% Intialization
delta = 10*pi/180; % Keep the steering angle at 10°
tireVelDesired = 60/3.6; % Keep the desired velocity at 60
km/h
iniU = 0;
iniV = 0;
iniPsiDot = 0;
t = 15; % Set time

% CG located at 30% of the wheelbase from the rear axle.
b = wb*0.3; % distance to CG from rear axle in [m]
a = wb-b;   % distance to CG from front axle in [m]
Fz1 = Fz*b/wb;         % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;        % gravitational force on rear tires
in [N]
CG_30_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink

% Plot
figure(n);
n = n + 1;
plot(CG_30_result.X.Data , CG_30_result.Y.Data);
axis equal;
hold on;

% CG located at 50% of the wheelbase from the rear axle.
b = wb*0.5; % distance to CG from rear axle in [m]
a = wb-b; % distance to CG from front axle in [m]
```

```
Fz1 = Fz*b/wb;          % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;         % gravitational force on rear tires
in [N]
CG_50_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink

% Plot
plot(CG_50_result.X.Data , CG_50_result.Y.Data);
axis equal;
hold on;

% CG located at 60% of the wheelbase from the rear axle.
b = wb*0.6; % distance to CG from rear axle in [m]
a = wb-b; % distance to CG from front axle in [m]
Fz1 = Fz*b/wb;          % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;         % gravitational force on rear tires
in [N]
CG_60_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink

% Plot
plot(CG_60_result.X.Data , CG_60_result.Y.Data);
axis equal;
hold on;

% CG located at 70% of the wheelbase from the rear axle.
b = wb*0.7; % distance to CG from rear axle in [m]
a = wb-b; % distance to CG from front axle in [m]
Fz1 = Fz*b/wb;          % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;         % gravitational force on rear tires
in [N]
CG_70_result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",
t); % Export result from Simulink

% Plot
plot(CG_70_result.X.Data , CG_70_result.Y.Data);
axis equal;
hold off;
```

```
title('Vehicle Trajectory for Different CG Location at 60
km/h','FontName','Times New
Roman','FontWeight','Bold','FontSize',16);
xlabel('X-position','FontName','Times New
Roman','FontSize',14);
ylabel('Y-position','FontName','Times New
Roman','FontSize',14);
legend('CG located at 30%','CG located at 50%','CG located
at 60%','CG located at 70%');
grid on;

%% Task Test

figure(n);
n = n + 1;
iniU = 10;
iniV = -10;
iniPsiDot = 0;

delta = 15*pi/180; % steering angle in [rad]
tireVelDesired = 20; % velocity in [m/s]
b = wb*0.5;
a = wb-b;
Fz1 = Fz*b/wb;          % gravitational force on front tires
in [N]
Fz2 = Fz - Fz1;         % gravitational force on rear tires
in [N]
result = sim("AuE881_HW5_Zhai_Kexuan_BicycleModel",10);
plot(result.X.Data,result.Y.Data);
title('Test','FontName','Times New
Roman','FontWeight','Bold','FontSize',16);
xlabel('X-position','FontName','Times New
Roman','FontSize',14);
ylabel('Y-position','FontName','Times New
Roman','FontSize',14);
grid on;
axis equal;
```