Vehicle Collision Avoidance and Cruise Control System Modelling

Mangirish Sanjeev Kulkarni, Onkar Digambar Chavan, and Vyankatesh Rajendra Chandge.

***Abstract*— The number of road accidents has increased as the country's road network, motorization, and urbanization have grown. While seatbelts and airbags help to reduce body movement after an accident, the force of the impact can induce abnormal jolting in the neck, upper back, arms, and legs. This work presents an MPC and PID-controlled vehicle collision avoidance system to avoid rear-end collisions with an active control system as a solution to such a problem. When designing the dynamic control system, safety-related aspects (relative speed and relative distance) were considered. The MPC control system is a potential way for minimizing the likelihood of an impact while maintaining the driver's comfort, according to the MATLAB/Simulink results.**

***Index Terms*— ADAS, CAS, MPC and PID**

I. INTRODUCTION

Most modern automobiles have steadily improved in terms of quality and functionality in recent years. Car manufacturers are concentrating on incorporating various types of ADAS systems into their vehicles. One of these systems is collision avoidance systems. Every year, over 1.3 million individuals are killed in automobile accidents. [1] Speeding, inattentive driving, dangerous vehicles, driving under the influence of alcohol, and a variety of other factors all contribute to road accidents. Different sorts of collision avoidance technologies are being developed to avoid these types of accidents and minimize the death rate. The primary goal of these technologies is to increase safety. Collision avoidance systems include forward collision warning, blind-spot warning, cross-traffic alert, pedestrian detection system, adaptive cruise control, parking assist, and electronic stability control.

The goal of this project is to prevent collisions by designing a system in which vehicles maintain a safe distance from the leading vehicle and continue to follow it. To avoid colliding with the leading car, the vehicle also implement emergency breaking in dangerous situations. If the leading car slows or stops, the following vehicle will take the appropriate steps to avoid a collision. MPC (Multi-Point Constraints) and PID (Proportional Integral Derivative) controllers were used to represent the vehicles.

A. Terminology

m	Mass of vehicle
v_x	Longitudinal velocity of vehicle
v_y	Lateral Velocity of Vehicle
d_r	Relative distance
d_s	Distance between vehicles
v_r	Relative speed
v_p	Preceding vehicle velocity
v_e	Ego vehicle velocity
v_i	Initial speed
t_h	Headway time
t_s	Sampling time
a_{ref}	Acceleration reference
a_{com}	Acceleration command given to vehicle
x_{outx}	Position output in x direction
x_{outz}	Position output in z direction
$\boldsymbol{\theta}$	Steering angle
$\boldsymbol{\Psi}$	Yaw angle
v_{ref}	Reference speed
x_{out}	Position output
v_{out}	Output speed
I_z	Polar moment of Inertia
C_f	Cornering stiffness of Front tire
C_r	Cornering stiffness of Rear tire

Table. 01 Terminology

B. Abbreviation

CAS	Collision Avoidance System
ADAS	Advanced Driver Assistance System
MPC	Model Predictive Control
PID	Proportional Integral Derivative Control
ABS	Anti-lock Braking System
DOF	Degree Of Freedom
OV	Output Variables
DMS	Drive Mode Selector

Table. 02 Abbreviation

II. PROPOSED MODEL

The first vehicle in the model was given a constant velocity to represent an automobile moving at a constant pace on a straight road, while the second vehicle were altered so that a rear-end accident would be possible, and the controller would have to intervene. For this either the MPC control logic or PID controller is used in the second vehicle.

The vehicle equipped with an MPC controller or PID controller concentrates on maintaining a constant speed and keeping a safe distance from the leading vehicle.

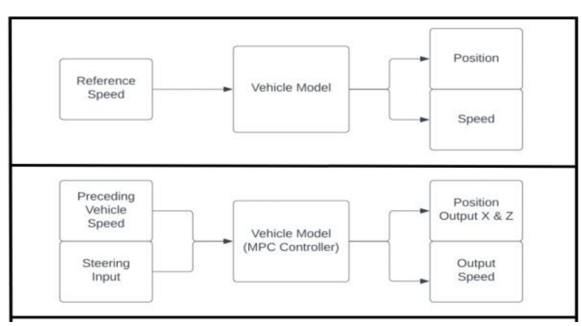


Fig. 01 Proposed Model

III. Vehicle Model

The vehicle dynamics are modeled using a 3-DOF rigid vehicle model (Single Track) imported from a Simulink dynamic block set.

Note- For full derivation of state-space model please refer appendix.

Therefore $\dot{x} = A_x + B_y$ is represented by –

State-Space Form

$$\begin{bmatrix}\dot{v}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\frac{-C_f-C_r}{mu} & \frac{-C_f l_a+C_r l_b}{mu}-u\\ \frac{-C_f l_a+C_r l_b}{Iu} & \frac{-C_f l_a^2-C_r l_b^2}{Iu}\end{bmatrix}\begin{bmatrix}v\\ r\end{bmatrix} + \begin{bmatrix}\frac{C_f}{m}\\ \frac{C_f l_a}{I}\end{bmatrix}\delta$$

[2]

A. *Vehicle Parameters (CarSIM)-*

CarSIM – E Class Sedan (Vehicle 01-02)

Mass	1650 kg
Longitudinal distance for CG to front axel	1.4 m
Longitudinal distance for CG to rear axel	1.650 m
Polar Moment of inertia	3234 kgm^2
Front tire corner stiffness	$12*10^3$ n/rad
Rear tire axle corner stiffness	$11*10^3$ n/rad

Table. 03 Vehicle 1 & 2 Parameters

Now for longitudinal dynamics, the plant model used for control design is transfer function between desired acceleration and actual vehicle speed

$$G(s) = \frac{7.273s - 98.24}{s^2 + 1.451s + 0.4302} \quad (49)$$

IV. ANALYSIS OF PROPERTIES

Controllability and Observability:

The controllability of a system is the ability of the system to reach any definite final state from a fixed initial state in a finite amount of time. A and B matrices evaluate a system's controllability using the state space equations.

$$P = [B\ AB\ A^2B\ \dots\ A^{n-1}B]$$

Similarly, a system's observability is a system's quality by which we can infer the previous states of the system from the final states and the inputs. It is also evaluated using C and A matrices from the state space equation.

$$Q = [C\ CA\ CA^2\ \dots\ CA^{n-1}]^T$$

The system is said to be controllable or observable if the P and Q matrices are full rank, this guarantees that we can transform the system from any one state to another without ever collapsing down to a lower unresolvable dimension. In our system, we evaluated the controllability and observability of the system and found out that both P and Q are full ranks, so the system is controllable and observable, so there are no states where the state space equations will fail to resolve the system. The P and Q for this system are:

P = [7.2727 -108.7918
5.1948 -4.1093]

Q = [1.0000 0
-0.6667 -20.0091]

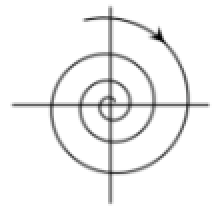
As the determinant of both P and Q is non-zero, they are full rank.

Stability and Dynamic Response:

The system is said to be stable if the output of the system is always within some finite range for a finite range of input, which means that the output and input of the systems are always finite and knowable. We determine the stability from the eigenvalues of the A matrix which are:

$$\lambda_1 = -0.4153\ ;\ \lambda_2 = -1.0359$$

The eigenvalues are generally represented as a complex number, so here we only have the real parts, we can see that the real parts of both the eigenvalues are negatives which implies that the system is asymptotically stable.



spiral sink

Similarly, to check the dynamic response of the system we calculated the damping ratio(ξ) from the transfer function and found that ξ > 1 which implies that the system is overdamped and we should expect the output to converge in a finite time without any overshoot.

From the transfer function, we can evaluate other parameters for a step response which are as follows:

```
        RiseTime: 5.93420900072095
   TransientTime: 10.7208685253556
    SettlingTime: 10.723333835756
     SettlingMin: -228.27573484182
     SettlingMax: -205.643945808944
       Overshoot: 0
      Undershoot: 0.102582776793623
            Peak: 228.27573484182
        PeakTime: 20.3670004394161
```

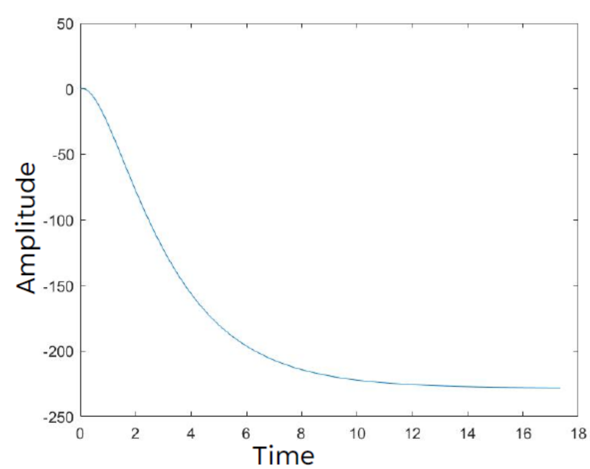


Fig. 02 Step Response

V. MODELLING OF VEHICLE ON SIMULINK

Simulink was used to model the Lead vehicle, which serves as the project's baseline model. The vehicle body transfer function given in equation (49) was utilized to depict the dynamics of acceleration. aids in approximating the dynamics of the throttle body and the inertia of the vehicle. [3]

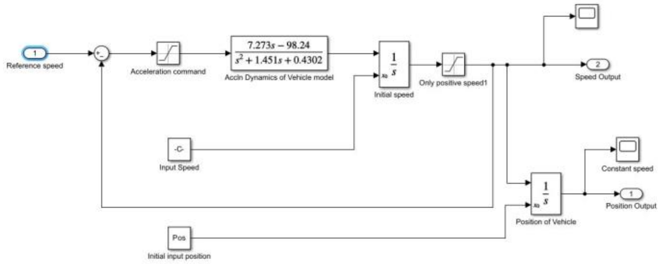


Fig. 03 Lead Vehicle Simulink Model

The transfer function block in this model corresponds to the transfer function G in the previous equation. The integrator blocks' initial position and input speed are used during setup. Since the vehicle has dynamic restrictions for acceleration and deceleration in real life, saturation blocks are utilized to replicate the output in a more realistic manner.

The model for Lead Vehicle assumes that it will go in a straight line. Additionally, it can be managed by inputting the initial speed and position. The vehicle one is set up for this project with an initial speed of 20 m/s and an initial position that is 100 m from the beginning position.

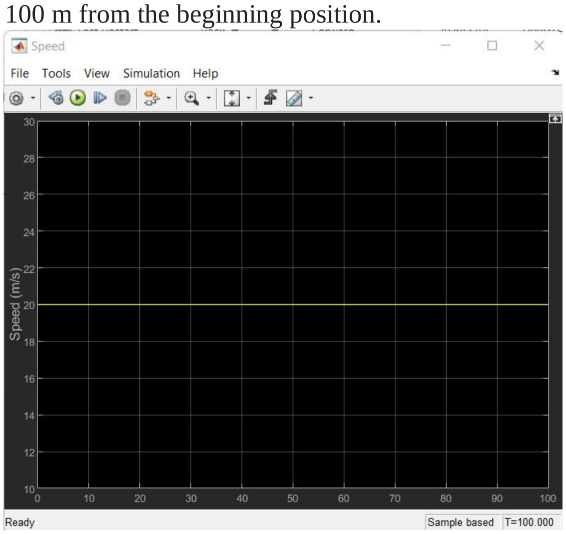


Fig. 04 Plot for lead vehicle speed

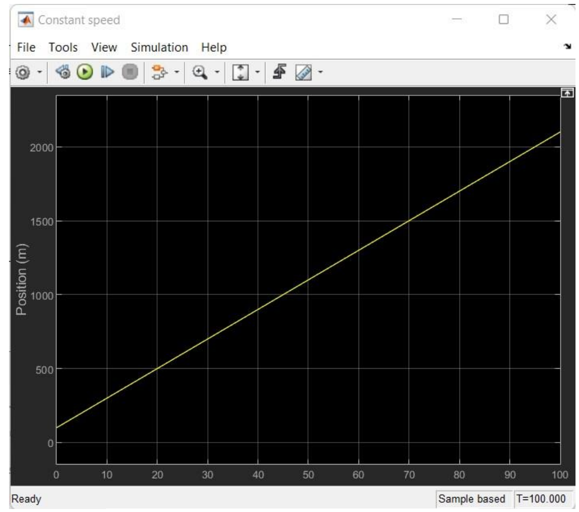


Fig. 05 Plot for lead vehicle position output

The graph in Fig depicts the trajectory of vehicle 1's constant 20 m/s speed over time. According to the plot in Fig. 1, the residence sends vehicle 01 over 100 meters in front of its initial position.

VI. DRIVE MODE SELECTOR

A. Drive Modes

1. Driving Mode 01: This is a free-drive mode in which the car would continue to travel in a straight line at a constant speed.

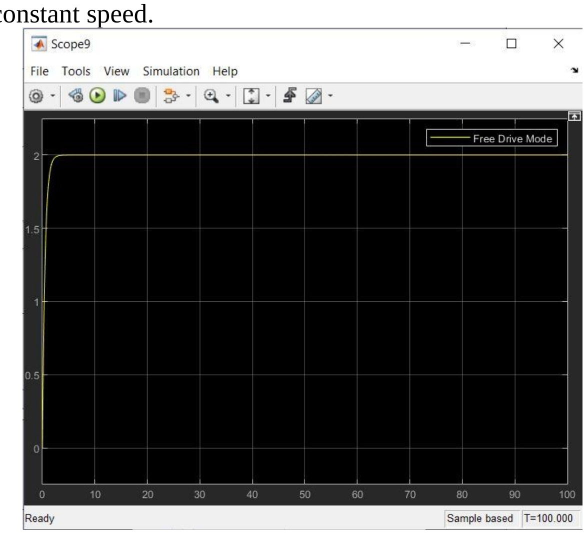


Fig. 06 Acceleration at Free Drive

2. Driving Mode 02: The vehicle eventually follows the leading vehicle in this mode. The vehicles that follow the leading vehicle maintain a safe following distance. Vehicles 02 are depicted below, and they are modelled using MPC and PID controllers, respectively. These controllers are in charge of keeping the cars' relative speeds and distances between them at a safe level.

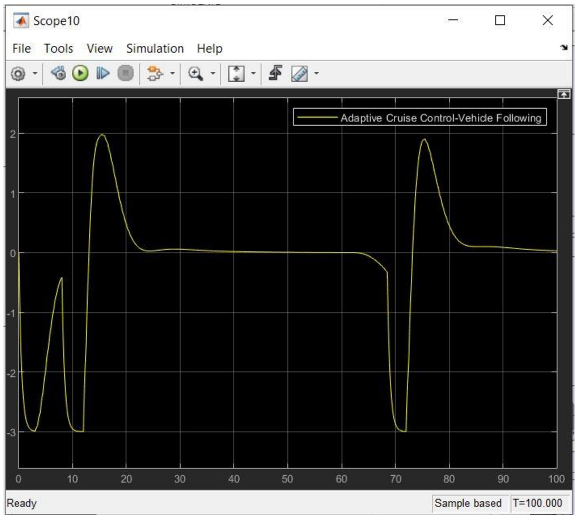


Fig. 07 Vehicle Acceleration with MPC Controller

3. Driving Mode 03: In this mode of the vehicle will end applying brakes on the following the vehicle. This mode would only be selected when the vehicle would sense any unnecessary obstacle, or the lead vehicle would stop suddenly.

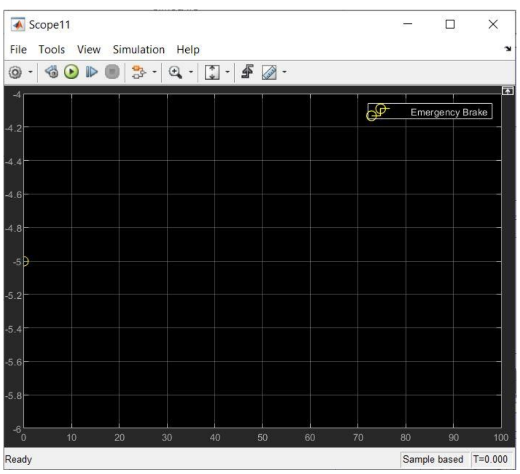


Fig. 08 Mode 03 Emergency Braking

B. Vehicle Modeled Using MPC Controller

MPC may control systems with several inputs and outputs that may interact with one another. Using the expected plant outputs, the MPC controller solves a quadratic programming optimization problem. At each time step, a controller solves the optimization problem by attempting to determine the best control signal at sampling time. The MPC controller attempts to find a trajectory that is as near as feasible to the reference trajectory.

MPC controller is modelled after the linear state space model after the linear system stated below:

$$x_1(k+1) = P\,x1(k) + Q_{u1}\,u_1(k) + Q_{d1}\,v_1(k)$$

$$y_1(k) = R\,x_1(k) \qquad [4]$$

P = state matrix

Q_{u1} and Q_{d1} are the input matrices corresponding to inputs u and v respectively.

R is the output matrix.

$$\min_{u_1} J = \sum_{j=1}^{N} \left|\left| y_{p1}(k+j|k) - y_{ref}(k+j|k) \right|\right|_{A_y} + \sum_{j=0}^{M-1} \left|\left| u1(k+j|k) \right|\right|_{B_u} \qquad [4]$$

$$\textit{subject to} \quad x_1(k+j+1|k) = P x_1(k+j|k) + Q_{u_1} u_1(k+j|k) + Q_d v_1(k+j|k)$$

$$x_1(k|k) = x_1(k)$$

$$y_1(k+j|k) = R x_1(k+j|k)$$

$$|u_1(k+j|k)| \leq u_{1_{limit}} \qquad [4]$$

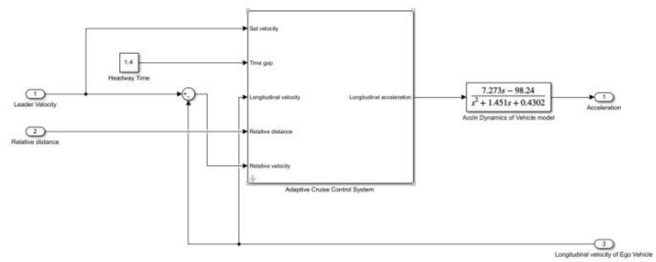


Fig. 09 Simulink Model for MPC Controller.

Where u_1 = Manipulated Variable

A_y and B_u = Weights for output and manipulated variables respectively.

The optimization problem searches for optimal value of input u where it reaches a minimal point. It can be observed that when the relative distance drops below 30m, vehicle reduces its speed before crossing a threshold value of relative distance.

Between 13th – 14th time step the MPC controller is activated, and the relative speed is brought to 0 and maintained for some time.

The output acceleration is a function of relative speed and is affected by it; as relative velocity rises (in the positive y-direction), acceleration rises, and as relative velocity falls, acceleration falls. As a result, the output acceleration plot matches the relative velocity plot, while the MPC controller's response (output) is visible a few time steps later. You can see the longitudinal velocity of ego vehicle changing as the MPC controller steps in to maintain a safe distance between the lead vehicle and it.

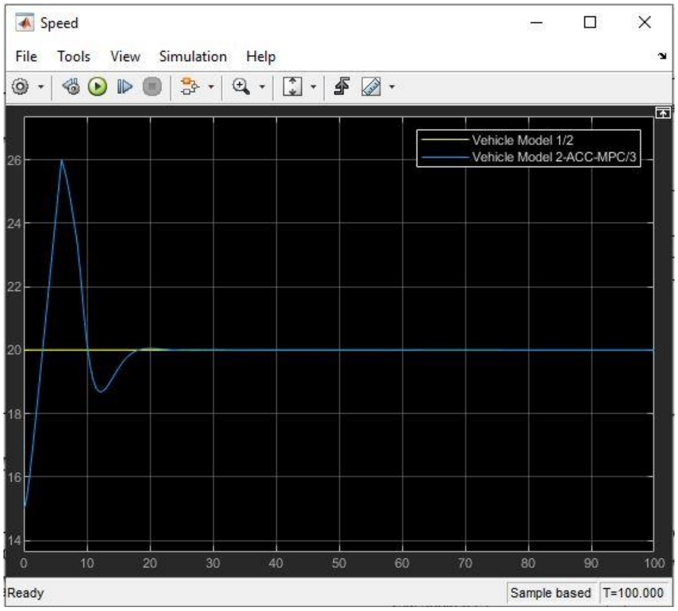


Fig. 10 Stabilized System

The sample time was kept at 0.5 sec, while the prediction horizon was kept at 200 steps. The controller behavior was adjusted so that the ego vehicle cruises at the same speed (i.e. converges) as the lead vehicle without compromising computation time.

C. Vehicle Modeled Using PID Controller

Proportional-Derivative-Integral (PID) Controller.

PID is simple and practical compared to MPC control method. Here we use PID to control the distance error and make it to zero. This will allow Vehicle 2 to follow a desired yaw rate and the side slip is reduced as much as possible.

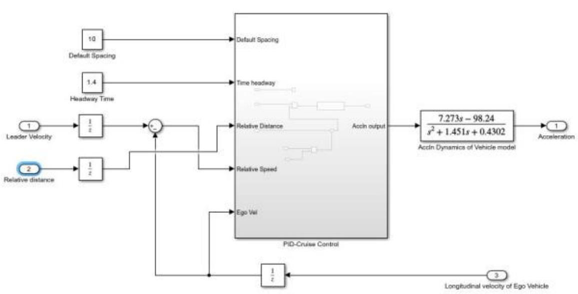


Fig.11 Mode 2: PID Control Dynamics Model

First, the input Relative speed which is difference between the Leader velocity and Longitudinal velocity of ego vehicle is terminated to keep the system simplified. This will help to obtain a system consisting of Single input- Relative Distance and Single output- Acceleration Dynamics of Vehicle.

Thus, the Controller Design to obtain Acceleration output is the difference between Relative distance and Reference distance. The inputs used in PID control are single input-single output, thus they have both pros and cons. As it is easy to control, the distortion analysis needs to be performed to further tune the output parameters.

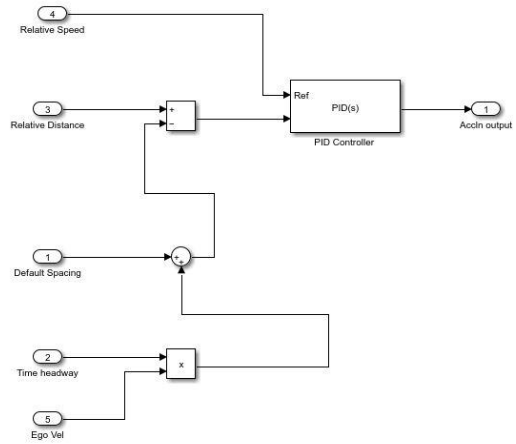


Fig.12 PID Controller.

Now, to further understand how the system behaves, below are represented inputs results for Relative distance and Reference distance-

The Ref. Distance is obtained using following formula

Reference distance = Ego Vehicle Speed + Time Headway + Default Spacing. [6]

Now the Acceleration Output = Relative Distance – Reference Distance. [6]

PID Tuning is done using PID control law in which proportional gain K_p, Integral gain K_i and Derivative gain K_d is used to form controller transfer block G_c.

$$G_c = K_p + K_i/s + K_d s \qquad [5]$$

The output values of K_p, K_d and K_i are tuned using Zieglar-Nichols Method. In this method initially K_i and K_d are set to zero and K_p is increased until loop begins to oscillate. After tuning we obtained satisfactory results using K_p= 2.4, K_i= 2.7, K_d= 0.08 and Filter co-efficient N = 500 which can be represented in below figure.

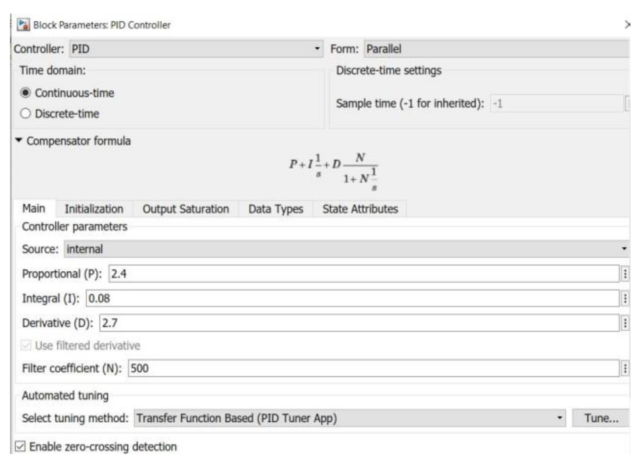


Fig.13 Tuning PID Controller

After initial tuning using PID, the output signal for Acceleration output is further tuned using Acceleration

Dynamics Plant model (G(s) Transfer Function). The resultant acceleration is shown below-

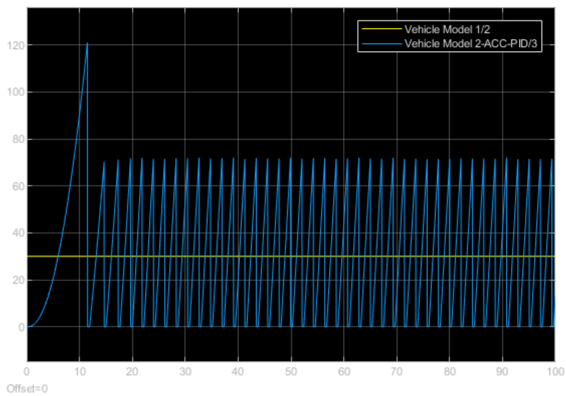


Fig.14 Mode 02- Acceleration Dynamics of Vehicle Model using PID controller.

The limitations of single input and single output can be represented in the above results. As relative speed is not taken into consideration the Vehicle behaves in fixed accelerated and decelerated pattern with respect to relative distance between Vehicle 1(Lead) and Vehicle 2(Ego). Unlike MPC controller where the vehicle converges after maintaining constant speed and fixed relative distance, The PID controller has limited command over the system. This will be further analyzed in the results section.

D. State Flow Model Used for Selecting the Drive Modes:

The state flow model is used for selecting the drive modes. The selection of drive modes is completely dependent on the relative distance and relative speed between the two cars.

- Consider a vehicle travelling on the road in straight direction. This is like the drive mode 01. Now this vehicle which is modeled either using MPC or PID controller, will calculate the relative distance between him and the car in front of it. And if the relative distance is more than 50m the vehicle would shift to the driving mode 02 where it will follow the lead vehicle and maintain a constant relative distance.
- Now if the relative distance goes below 60m the vehicle would shift to drive mode 01.
- Consider the vehicle in driving mode 02 and if in this case the relative distance goes below 30m the vehicle would enter the driving mode 03 were the vehicle would end up applying the brakes.
- And as the relative distance increases and goes above 35m the vehicle would shift back to driving mode 02.
- In another situation, the vehicle would switch from mode 01 to mode 02 as soon as the relative distance fell below 30 meters.

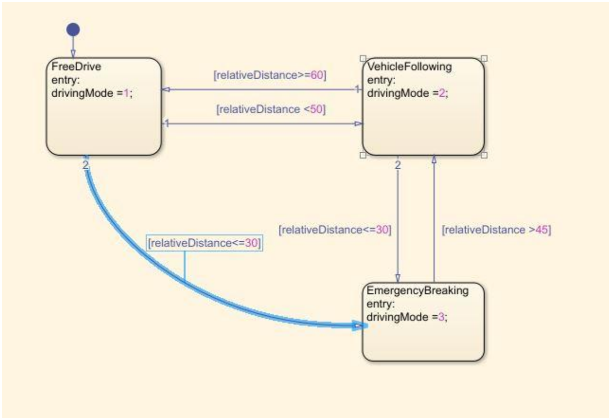


Fig.15 State Flow Model

VII. RESULTS

<u>Results for All Vehicles Simulation (Without Lane Change Trajectory):</u>

The vehicle 2 (MPC controlled) converges after maintaining relative distance that is provided in the State Flow Diagram which accesses the Mode selector in case to avoid collision and maintain smooth control of system. The system detects obstacle i.e., Vehicle A at 0.5 seconds and then applies breaks to gradually decrease its acceleration with respect to relative distance between them. As we ran the simulation keeping in mind of high risk of collision, The vehicle 2 came right to stop at 12 seconds and automatically picked up pace the next second thus maintaining smooth transition even under panic situation. Vehicle 2 then picks up pace with maintaining the relative speed and distance in mind and converges at 20 seconds. Also, Vehicle 2 picking up pace after decelerating is kept avoiding impact from rear vehicle. Thus, driver and passenger comfort are maintained throughout the scenario. In case of Vehicle 2 (PID controlled), as it is a single input-single output system the system although it avoids collision, The downside is its sudden acceleration and deceleration. This creates uncomfortable driving scenarios. In case of Vehicle 2 it detects the collision at 6th second and applies panic emergency breaking. This results to vehicle gradual stop at 17^{th} second, but due to aberrant breaking the system needs time to initiate its longitudinal acceleration. This results in decrease in response of 2-3 seconds as compared to MPC controller as we can see in the above figure which could be deadly in real life scenario.

This can be further verified by analyzed by visualizing change in relative distance of both vehicles.

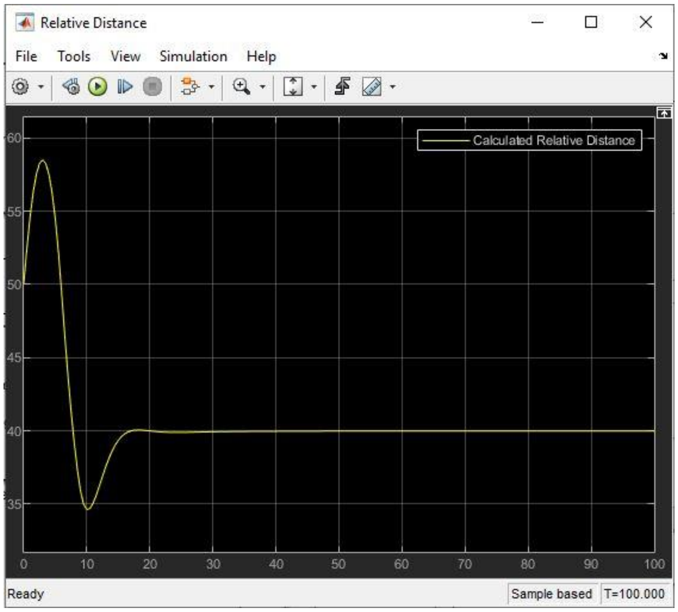


Fig.16 Relative Distance between Vehicle 1 and Vehicle 2(MPC Controlled).

As results shows smooth transition is maintained throughout the scenario.

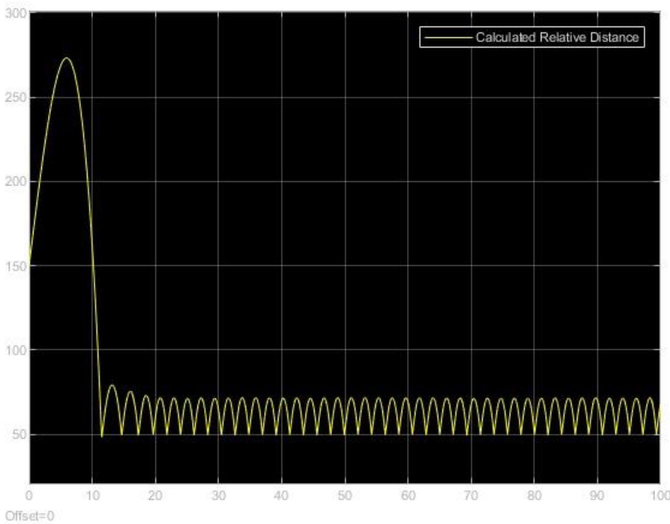


Fig.17 Relative Distance between Vehicle 1 and Vehicle 2(PID Controlled).

Due to sudden acceleration and deceleration the transition is not smooth thus lead to harsh driving scenarios and uncomfortable system for driver and passengers.
Thus, we can conclude that MPC is superior system then PID as it shows better performance when tested in same scenarios.

VIII. CONCLUSION

After evaluating the results from the Full system simulation, we can confirm that Car 2 (E class Sedan) handled by MPC is more stabilized and converges after 25 seconds of sampling time. Although relative speed is taken into account, PID is more challenging to converge. The emergency brake is engaged to prevent a collision, and after a brief period of stabilization, the system resumes normal operation.

IX. FUTURE WORK

- The project focuses on a lane changing maneuver for straight future it can be extended for a curve road.
- The project works on collision avoidance system for one way street in future collision avoidance systems can be developed of two-way streets as well.
- In two-way street the vehicle would detect the vehicle coming in other direction and abort the lane changing maneuver
- This project was based obstacle detection and collision avoidance, where the obstacle was another car. In future we can also consider human or any other obstacle detection.
- Try implementing this system on intersection with unpredictable obstacles.

X. REFERENCE

1. “Road traffic injuries,” *World Health Organization*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries
2. R. Rajamani, Vehicle Dynamics and control. New York: Springer, 2012.
3. “Adaptive Cruise Control System,” *MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/mpc/ug/adaptive-cruise-control-using-model-predictive-controller.html. [Accessed: 06-May-2022].
4. “Adaptive MPC controller,” *MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/mpc/ug/obstacle-avoidance-using-adaptive-model-predictive-control.html. [Accessed: 06-May-2022].
5. “PID theory explained,” *NI*. [Online]. Available: https://www.ni.com/en-us/innovations/white-papers/06/pid-theory-explained.html. [Accessed: 06-May-2022].
6. “Understanding model predictive control,” *MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/videos/series/understanding-model-predictive-control.html. [Accessed: 06-May-2022].
7. Shuping Chen and Huiyan Chen 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **892** 012034

XI. APPENDIX.

Vehicle Kinematic Model

Vehicle Motion has three degrees of freedom and is on a horizontal plane (DOF). Axis of rotation and displacement are on a lateral and longitudinal plane that is parallel to that plane (Yaw Rotation). You can describe the trajectory and control the course by manipulating these three degrees of freedom. The following kinematic model can be used to generate the equations for the bicycle model. [2]

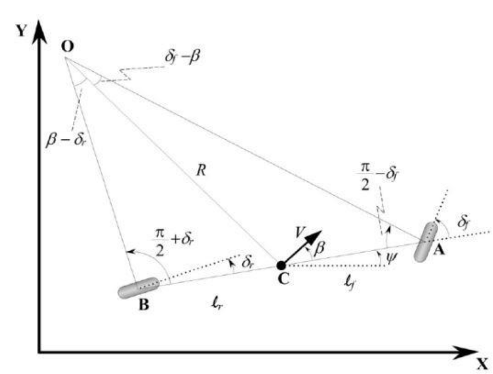


Fig. 18 Vehicle Kinematic Model [2]

As shown in Fig. 01 above, points A and B are represented by a single central tire instead of the two front and rear wheels.

δ_f = steering angle at front wheel,

δ_r = steering angle at rear wheel.

Assumption: Vehicle model assumes front wheel steering angle only, therefore δ_r set to zero.
C – center of gravity,
l_f and l_r – distances from point A and B to Centre of gravity.
Sum of these 2 corresponds to L (wheelbase) i.e.,

$$l_f + l_r. \tag{1}$$

Assumption – Vehicle is in planar motion, therefore 3 co-ordinates required to describe it. (X, Y and Ψ).

V – velocity of vehicle at CG,
β – slip angle with longitudinal axis.

Applying sine rule ΔOCA and ΔOCB,

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin\left(\frac{\pi}{2} - \delta_f\right)}{R} \tag{2}$$

$$\frac{\sin(\beta)}{l_r} = \frac{1}{R} \tag{3}$$

Now, multiplying eq (2) by $l_f / \cos(\delta_f)$,

$$\tan(\delta_f)\cos(\beta) - \sin(\beta) = \frac{l_f}{R} \tag{4}$$

Similarly, multiplying eq (3) by l_r,

$$\sin(\beta) = \frac{l_f}{R} \tag{5}$$

Now, adding (4) and (5),

$$\tan(\delta_f)\cos(\beta) = \frac{(l_f + l_r)}{R} \tag{6}$$

.

If radius (R) of vehicle trajectory changes slowly to low velocity, the yaw rate (Ψ') = angular velocity (ω)

$$\omega = \frac{V}{R} \tag{7}$$

Thus, Yaw rate can be described as,

$$\varphi' = \frac{V}{R} \tag{8}$$

Now, substituting R from eq (8) in eq (6), we get,

$$\varphi' = \frac{V\cos(\beta)}{l_f + l_r}\tan(\delta_f) \tag{9}$$

Now, the overall equation of kinematic model can be defined as,

$$X' = V\cos(\varphi' + \beta) \tag{10}$$

$$Y' = V\sin(\varphi' + \beta) \tag{11}$$

$$\varphi' = \frac{V\cos(\beta)}{l_f + l_r}\tan(\delta_f) \tag{12}$$

[2]

B. *Vehicle Dynamic Model*

The kinematic model's drawback is that results must be obtained at low enough vehicle speeds and steering angles. Dynamics must be included in our system because under high-speed conditions, the tire begins to drift away from the road. [2]

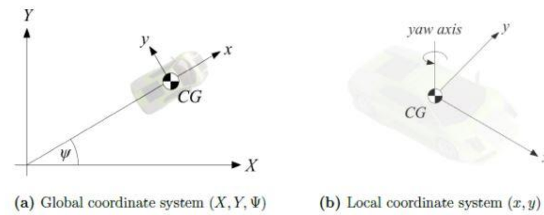


Fig. 19 Model Coordinate System [2]

We used a Simulink block set with a 3 DOF rigid vehicle model (single track) for our project. Our test car can be turned with the front wheels.
Consider,
x and y - Longitudinal and Lateral direction in vehicle frame,
X and Y – Longitudinal and Lateral direction in global frame,
Yaw angle – x, y vehicle frame (ϕ)
Heading angle – X, Y frame,
F_x and F_y – Longitudinal and Lateral tire forces on C.G. of vehicle,
F_z – Normal tire load,
F_l and F_c – Longitudinal and Lateral stiffness force,
ϕ – heading angle
δ – wheel steering
α – slip angle
I_z – Polar moment of inertia

Assumption: No front and rear tire forces in x-axis.

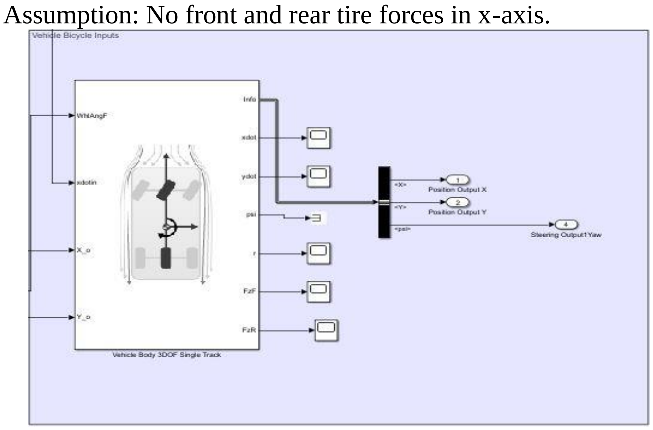


Fig. 20 Simulink Bicycle Model

Only the lateral tire forces of the dynamical model are independently applied to each wheel in coordinate frames, as indicated in the image. In this case, the vehicle C.G. will serve as a benchmark for computing Newton's law.

Here longitudinal velocity is constant V_x = 0, as a result force F_{xf} = 0

$$mv_x' = mv_y\varphi' + F_{xf}\cos(\delta) + F_{xr} - F_{yf}\sin(\delta) \tag{13}$$

$$mv_y' = mv_x\varphi' + F_{xf}\sin(\delta) + F_{yr} + F_{yf}\cos(\delta) \quad (14)$$
$$I_z\varphi' = L_fF_{yf}\cos(\delta) + L_fF_{xf}\sin(\delta) - L_rF_{yr} \quad (15)$$

The eq of motion of the vehicle in an absolute inertial frame

$$F_x = F_l\cos(\delta) - F_c\sin(\delta) \quad (18)$$
$$F_y = F_l\sin(\delta) + F_c\cos(\delta) \quad (19)$$

Currently, the front and rear axles' steering angles can be characterized as,

Assuming front wheel drive,

$$J_{wf}\,(\omega_f') = (T_w - T_b - r_wF_{l,f}) \quad (20)$$
$$J_{wr}\,(\omega_r') = (-T_b - r_wF_{l,r}) \quad (21)$$

Where,
ω' = angular acceleration of wheel
I_w = Moment of Inertia of wheels
T_b = brake torque
T_w = wheel torque

Now, the front & rear speeds along x-axis and y-axis are calculated as [2] –

$$V_{x,i} = V_i\cos(\propto_i) \quad (22)$$
$$V_{y,i} = V_i\sin(\propto_i) \quad (23)$$

Where,

α_f and α_r - Front and Rear tire slip angle
V_f and V_r – Front and Rear tire speed

The relationship between host vehicle speed and tire speeds[2] -

$$V_{x,f} = V_{h,f} \quad (24)$$
$$V_{x,r} = V_{h,x} \quad (25)$$
$$V_{y,f} = V_{h,y} + l_f\,\varphi' \quad (26)$$
$$V_{y,r} = V_{h,y} - l_r\,\varphi' \quad (27)$$

The slip angle is the angle between the speed of the wheel and its direction.

Then tire side-slip angles are,

$$\alpha_f = \delta_f - \frac{V_{y,f}}{V_{x,f}} \quad (28)$$
$$\alpha_r = \delta_r - \frac{V_{y,r}}{V_{x,r}} \quad (29)$$

Now, longitudinal forces are assumed to depend on normal force, surface friction and longitudinal slip ratio,

$$F_{x,i} = F_x(\sigma_{i,u}, F_{z,i}) \quad (30)$$

Pacejka model utilized to calculate the friction co-efficient as function of longitudinal slip ratio,

$$\mu(\sigma) = D\sin(C\tan^{-1}(B\sigma - E\,(B\sigma - \tan^{-1}(B\sigma)))) \quad (31)$$

The value of B, C, D and E are different road types,
The longitudinal traction force,
F_x is calculated as

$$F_x = \mu\,(\sigma)F_z \quad (32)$$

Where, Fz – normal force exerting on wheel,
μ(σ) = Friction Coefficient

The lateral tire forces, which rely on normal force, surface friction, and slip angle, are essential for maneuvering range.

$$F_{y,i} = F_y(\alpha, \mu, F_{z,i}) \quad (33)$$

Tire force F_y now varies in direct proportion to slip angle.

The ratio of F_y to is known as the cornering stiffness C_f. commuter with slip angle in a linear zone. Consequently, slip angles are a linear function of lateral tire forces.

$$F_{y,i} = 2\,C_{y,i}\,\alpha_i \quad (34)$$

Based on the static weight distribution between the axles, the cornering stiffness of the front and rear tires is determined.

$$C_{y,i} = C_s\frac{F_{z,i,static}}{2} \quad (35)$$

where C_s is the coefficient of cornering stiffness.

The front wheel slip angle, α_f, is the difference between the front wheel's steering angle and the tire velocity's orientation angle to the longitudinal axis.

$$\alpha_f = \delta - \theta_{vf} \quad (36)$$

Similarly, the angle of rear wheel slip,

$$\alpha_f = -\theta_{vr} \quad (37)$$

Therefore, the lateral tire forces for front and rear wheels of vehicle are –

$$F_{yf} = 2\,C_{\alpha f}(\delta - \theta_{vf}) \quad (38)$$
$$F_{yr} = 2\,C_{\alpha r}(\delta - \theta_{vr}) \quad (39)$$

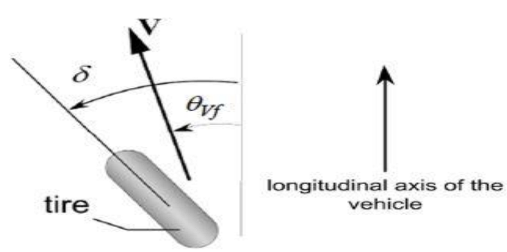


Fig. 21 Tire Slip Angle [2]

Now, to calculate velocity angle of the front wheel θ_{Vf}out and rear wheel θ_{Vr}, we use

$$\tan(\theta_{vf}) = \frac{V_y + l_f\dot{\varphi}}{V_x} \quad (40)$$

$$\tan(\theta_{vr}) = \frac{V_y - l_f\dot{\varphi}}{V_x} \quad (41)$$

Equations (40) and (41) can be rewritten as follows because we are assuming small angle approximations in this case:

$$\theta_{Vf} = \frac{V_y + l_f\dot{\varphi}}{V_x} \quad (42)$$

$$\theta_{Vr} = \frac{V_y - l_f\dot{\varphi}}{V_x} \quad (43)$$

Consequently, the relationship between tire longitudinal forces can be expressed as,

$$F_{yf} = 2C_{\alpha f}(\delta - \frac{V_y + l_f\dot{\varphi}}{V_x}) \quad (44)$$

$$F_{yr} = 2C_{\alpha r}(- \frac{V_y - l_f\dot{\varphi}}{V_x}) \quad (45)$$

Here is the internal controller plant inaccuracy in relation to the reference trajectory. The error is a lateral displacement error, which is modeled by the following equations

$$\dot{e_1} = V_x e_{2+} V_y \quad (46)$$

$$\dot{e_2} = \varphi - \varphi_{der} \quad (47)$$

The desired yaw angle rate is given by

$$\dot{\varphi_{der}} = V_x \mathrm{k} \quad (48)$$

where k is a road curvature indicator (if applicable)
By linearizing the bicycle model, the state-space model for lateral dynamics