ENPM 667 Project 1 submission

Topic:

Predictive Active Steering Control for Autonomous Vehicle Systems

Group Members

Advait Patole

UID: 118130743

Gaurav Raut

UID: 118199306

Contents

ABSTRACT- ***The field of autonomous vehicles has been a great topic of research for many researchers. In this paper, we are presenting a model predictive control (MPC) approach for controlling an active front steering system in an autonomous vehicle. We have defined the trajectory for the vehicle using this data the MPC controller computes the front steering angle so that the vehicle follows the trajectory with minimum error with highest possible speed. We present two approaches with different computational complexities. In the first approach we are using a nonlinear vehicle model to formulate the MPC problem. The second approach is based on successive online linearization of the vehicle model wherein we discretize the time variant plant model. Discussions on computational complexity and performance of the two schemes are presented. For the demonstration of the effectiveness MPC formulation we have considered the speed of the vehicle up to 15 m/s on a double lane road and the results are presented in this paper as well as we have provided simulation for the same.***

INDEX TERMS AND ACRONYMS

AFS - Active Front Steering

MPC - Model Predictive Control

LTI - Linear Time Invariant

LTV - Linear Time Variant

AFS - Active Front Steering

TC - Traction Control

ABS - Active Braking System

VSR - Variable Steering Ratio

GNC - Guidance Navigation Control

Index Terms - Active steering, autonomous vehicles, model predictive control, nonlinear optimization, vehicle dynamics control, vehicle stability, linear time variant model predictive control, linear time invariant model predictive control, trajectory generation

INTRODUCTION

The vision for autonomous vehicles is ambitious and novel. It may sound like science fiction rather than a real development that could happen in our lifetimes. We are witnessing tremendous progress in this field and new technologies are being developed to improve the performance of the autonomous systems. There are lot of companies like Tesla, Ford who have their autonomous or semi-autonomous vehicles already in the market. Yet, the possibility exists that we will see fully autonomous vehicles on U.S. roads in the coming few years. The reduction in the number of fatalities, injuries, crashes and property damages was a huge motivating factor while developing autonomous vehicles [1]. The field of autonomous vehicles require combination of knowledge of various disciplines like mechanical engineering, computer science electronics engineering, control systems etc. The journey of research on autonomous vehicles is quite interesting and it began in 1926 with world's first radio-controlled car 'Linrrican Wonder'. A modified form of Linrrican Wonder was used by the name "Phantom Auto" and demonstrated in December 1926 in Milwaukee, by Achen Motors. GM (General Motors) sponsored Norman Bel Geddes's exhibit Futurama at the World's Fair, 1939[2]. With the advent of research in vision guided systems like LIDAR, radar, GPS and computer vision there has been tremendous growth in this field and people are actively

involved in developing newer technologies to enhance this field. This research led to development of autonomous technologies present in modern cars like lane parking, steering control, auto pilot, adaptive cruise control etc.

The research in the field of active safety systems dates to the 1980's and they were primarily involved with the development of longitudinal dynamics of the motion related to active braking systems (ABS) and traction control (TC) systems. This was followed by work on different vehicle stability control systems [11] (which are also known under different acronyms such as electronic stability program (ESP), vehicle stability control (VSC), interactive vehicle dynamics (IVD), and dynamic stability control (DSC)). These systems provide stability to the vehicle by minimizing the loss due to traction. Whenever they detect there is a loss in steering control, they apply brake to steer the vehicle where it is intended to go. The active braking system improves the braking performance while traction control helps prevent the slippage of the tires and improves the stability and steerability of the vehicle by maximizing the tractive and lateral forces between tires and the road[4].Active front steering system also makes use of active front steering (AFS) command along with the traction control and braking system for improving the lateral stability of vehicle[5].The steering command can also help to reject the disturbances that the system is facing due to external factors such as wind, asymmetric braking. In [9] an active front steering (AFS) controller with a new variable steering ratio (VSR) map is proposed that enhances the vehicle stability and improves driver's steering comfort. The paper [10] states how the decoupling of the lateral and yaw motions of a car and car's yaw damping are achieved simultaneously by feedback of both yaw rate and front steering angle with the scheduled gains. It has significant safety advantages in critical situations where the driver of the conventional car must control an unexpected yaw motion.

In this paper we have developed our own trajectory using the waypoints (lateral position, longitudinal position and yaw angle) on which the vehicle moves with a constant velocity. We have used the trajectory generation function presented in [4] to generate the trajectory using the longitudinal position of the vehicle. The input that is provided to the vehicle model is the steering angle. For simulation purpose we have considered double lane change scenario, here our major focus is on the yaw angle and the lateral position of the vehicle. The goal of the project is to travel the reference trajectory as close as possible. We are trying to minimize the error that between predicted and actual yaw angle, lateral position, lateral velocity and yaw rate. The future trajectory is generated over a finite horizon at each time step. In this paper we have worked on two different formulations. The first formulation deals with linear time invariant (LTI) MPC where we are tuning the controller with weights and using it on the vehicle model. The second formulation presents a suboptimal MPC controller based on successive online linearization of the vehicle model. This is linearized around the current operating point at each time step a linear MPC controller is designed for the resulting linear time-varying (LTV) system. The idea of using time varying models goes back to the early 1970s in the process control field although it has been properly formalized only recently. We have kept the speed of the car as 15m/s and the total simulation time is 10 seconds.

TIRE MODEL

Apart from aerodynamics force and gravity, tire ground reaction forces are the only external force acting on the vehicle. The tire forces have highly nonlinear behavior when slip ratio or slip angle is large. Thus, it is of extreme importance to have a realistic nonlinear tire force model for the vehicle dynamics when operating the vehicle in the tire nonlinear region. In such situations, large slip ratio and slip angle can happen simultaneously and the longitudinal and lateral dynamics of the vehicle is highly coupled and nonlinear due to the nature of the tire forces. Similar situations can occur even with small inputs when the surface friction coefficients small. When the slip ratio and slip angle are both small, both longitudinal and lateral tire forces show linear behavior and are less coupled. This situation holds true when the vehicle operates with moderate inputs on high μ surfaces. In such situations, linearized tire models might serve well in control design, with proper constraints on the slip ratio and slip angle. In this section, we will present two tire models. From a complex nonlinear semi-empirical model capturing the nonlinear and coupling behavior of the tire forces, to a simplest linear model with no coupling between longitudinal and lateral tire forces.

Pacejka Tire Model

It is a complex nonlinear semi-empirical model being able to describe the nonlinear behavior of tire forces under a wide range. It describes tire forces as a function of normal force, slip ratio, slip angle and surface friction coefficient.

$$Y(x) = Dsin(C\tan^{-1}(B\Theta(X))) + S_v$$

$$Y(x) : Longitudinal\ or\ Lateral\ force$$

$B, C, D\ and\ Sv\ are\ the\ parameters\ fit\ from\ the\ experimental\ data\ and\ \Theta(X)\ is\ defined\ as\ follows$

$$\Theta(X) = (1 - E)(X + S_h) + \left(\frac{E}{B}\right)\tan^{-1}(B(X + S_h))$$

$$E\ and\ S_h\ are\ parameters\ fit\ from\ experimental\ data$$

Linear Tire Model

This subsection presents a linear tire model, which can be used for modeling lateral tire force inside the linear region. This model is only valid when both slip angle and slip ratio are restricted to have small values. The linear tire model is the simplest tire model we can possibly get and should be implemented only with small slip angles, i.e., inside the tire linear region.

With small α and s assumption, the lateral tire force is modeled as

$$F_c = C_\propto(\mu, F_z)\alpha$$

$$where\ C_\propto\ is\ called\ tire's\ cornering\ stiffness\ coefficient\ and\ is\ a\ function\ of\ friction\ coefficient\ \mu$$

$$and\ normal\ force\ F_z$$

FRAMEWORK FOR AUTONOMOUS GUIDANCE

The main components of an autonomous vehicle guidance system are trajectory generator, trajectory replanning, low level control system and a vehicle(plant) model. The trajectory generation generates offline the trajectory with the given set of conditions that are instrumental in generating the trajectory. The offline generated trajectory can be regenerated online with the help of trajectory replanning module based on the measurements or due to the occurrence of some error. The low-level control system actuates the different parts of vehicle model based on inputs from sensors, states, parameter estimation and reference command(trajectory) from trajectory replanning module. The purpose of the low-level controller is to reduce the difference between predicted and actual signal despite disturbances, noise or uncertainties in vehicle conditions. The combination of these three modules is referred to as guidance and navigation control (GNC). The information that reaches to trajectory replanning module and low-level control may not be same. The trajectory replanning module is based on receding horizontal control design. The planning problem is formulated as constrained optimization problem that minimizes the weighted sum of time, steering and acceleration control efforts. To generate the trajectory, we have used MATLAB Driving Scenario Designer App wherein we have developed the trajectory scenario based on the longitudinal position. We have added actors of the trajectory scenario like vehicle, and we have specified the speed of 15m/s to the vehicle.

VEHICLE MODELING

When looking from the control's perspective, the vehicle dynamics can be represented efficiently using the simple bicycle model which aims to reduce the complexity while achieving a similar performance [1]. The literature in [8]. The dynamics of the model can be phrased from the following equations:

$$m\ddot{x} = m\dot{y}\dot{\psi} + 2F_{x_f} + 2F_{x_r} \tag{1a}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{y_f} + 2F_{y_r} \tag{1b}$$

$$I\ddot{\psi} = 2aF_{y_f} - 2bF_{y_r} \tag{1c}$$

In the car's inertial frame, the vehicle's equations can be given as,

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi, \qquad \dot{Y} = \dot{y}\cos\psi + \dot{x}\sin\psi \tag{2}$$

The latera and longitudinal velocities with reference to the vehicle's front and rear wheels are as mentioned,

$$v_{l_f} = v_{y_f}\sin\delta_f + v_{x_f}\cos\delta_f \tag{3a}$$

$$v_{l_r} = v_{y_r}\sin\delta_r + v_{x_r}\cos\delta_r \tag{3b}$$

$$v_{c_f} = v_{y_f}\cos\delta_f - v_{x_f}\sin\delta_f \tag{3c}$$

$$v_{c_f} = v_{y_r}\cos\delta_r - v_{x_r}\sin\delta_r \tag{3d}$$

Here,

$$v_{y_f} = \dot{y} + a\dot{\psi} \qquad v_{y_r} = \dot{y} - b\dot{\psi} \tag{4a}$$

$$v_{x_f} = \dot{x} \qquad v_{x_r} = \dot{x} \tag{4b}$$

In the car's inertial frame, the forces acting on the C.G. are:

$$F_y = F_l\sin\delta + F_c\cos\delta \qquad F_x = F_l\cos\delta - F_c\sin\delta \tag{5}$$

The longitudinal and lateral forces are calculated below. Here, both the forces are a function of the following parameters,

$$F_l = f_l(\alpha, s, \mu, F_z) \qquad F_c = f_c(\alpha, s, \mu, F_z) \tag{6}$$

The vehicle model in the paper is further linearized for calculations. For this, we further derive the linearized lateral and longitudinal forces as mentioned in the equation (6).

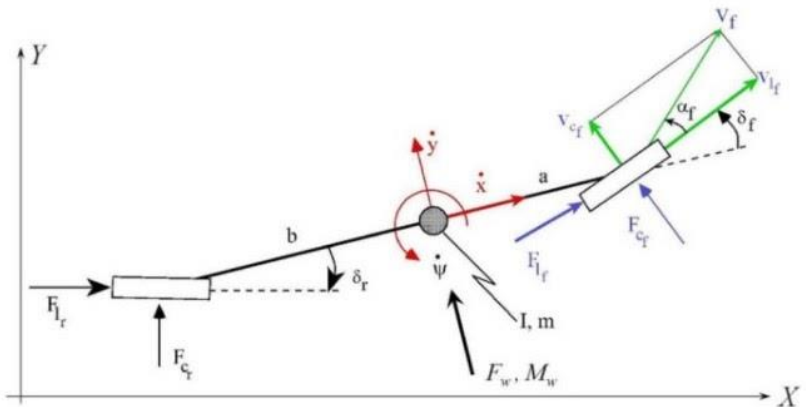


Figure 1: Bicycle Model Dynamics

Front Tire

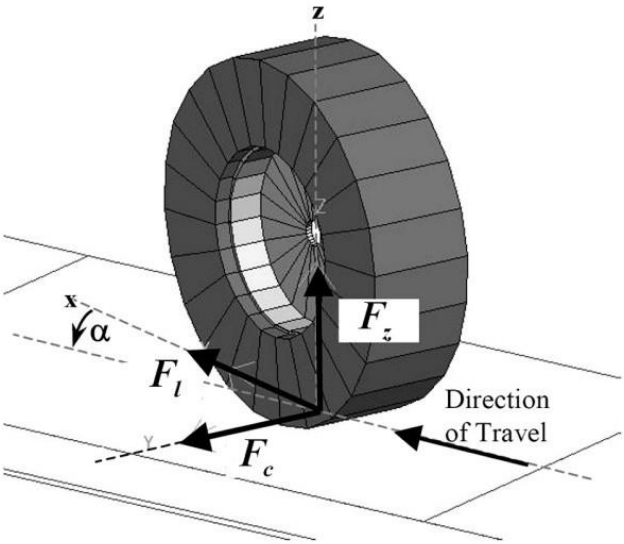


Figure 2: Tire Force Model

For F_{c_f},

The front tire's heading angle is given as:

$$\psi_f \approx \tan\psi_f = \frac{v_{c_f}+ar}{v_{l_f}} \tag{7a}$$

To obtain the slip angle α, we will subtract the heading angle with the front steering angle δ_f,

$$\alpha_f = \delta_f - \frac{v_{c_f}+ar}{v_{l_f}} \tag{7b}$$

Now, the lateral force on the front tire for a linearized tire model can be given as,

$$F_{c_f} = C_f \alpha_f = C_f \left(\delta_f - \frac{v_{c_f} + ar}{v_{l_f}} \right) \tag{7c}$$

Rear Tire

Since most cars don't have steering in the rear, the steering angle can be taken as zero (δ_r=0).

Now, similar to the front tire's model, the rear heading angle can be approximated as,

$$\psi_r \approx \tan \psi_r = \frac{v_{c_r} - br}{v_{l_r}} \tag{8a}$$

To obtain the slip angle α, we will subtract the heading angle with the front steering angle $\delta_r = 0$,

$$F_{c_r} = C_r \alpha_r = -C_r \frac{v_{c_r} + ar}{v_{l_r}} \tag{8b}$$

The slip ratio s is defined as,

$$s = \begin{cases} \frac{r\omega}{v_l} - 1 \text{ , if } v_l > r\omega, v_l \neq 0 \; for \; braking \\ 1 - \frac{v_l}{r\omega} \text{, if } v_l < r\omega, \omega \neq 0 \; for \; driving \end{cases} \tag{9}$$

The z-axis weight distribution among the front and rear wheels are given as:

$$F_{z_f} = \frac{bmg}{2(a+b)} \qquad F_{z_r} = \frac{amg}{2(a+b)} \tag{10}$$

Finally, the nonlinear vehicle model is defined as,

$$\frac{d\xi}{dt} = f_{s(t),\mu(t)}(\xi(t), u(t)) \tag{11a}$$

$$\xi = [\dot{Y}, \dot{\psi}, Y, \psi]' \tag{11b}$$

For the aforementioned equations, the parameter notations are as follows:

$(\cdot)_{f,r,l,c}$: The *f, r, l, c* subscripts represent the front, rear, longitudinal and lateral relations respectively

$v_{(\cdot)}$: Represents the corresponding velocity as defined in the subscript

m: Mass of the vehicle

$\ddot{x}$: Longitudinal acceleration

$\ddot{y}$: Lateral acceleration

$\dot{y}$: Lateral velocity

$\dot{x}$: Longitudinal velocity

ψ: Heading angle

$\dot{\psi}$: Yaw rate

$\ddot{\psi}$: Yaw acceleration

a, *b*: Distance of C.G. from the front and rear axle respectively

$F_{(\cdot)}$: Forces acting in the respective direction and reference

ω: Angular velocity of the vehicle

$\delta_{f,r}$: Steering angle for front and rear wheels

$C_{f,r}$: Tire's cornering stiffness for front and rear respectively

r: The radius of curvature of the vehicle

I_z: Vehicle's moment of inertia along the C.G.

STATE-SPACE REPRESENTATION

Final dynamics state equations

Now that we have obtained the equations for the lateral forces, we culminate the results in (1), we get,

$$\ddot{y} = -2\left(\frac{C_r+C_f}{m\dot{x}}\right)\dot{y} - \left(\frac{2C_fa-2C_rb}{m\dot{x}} + \dot{x}\right)r + \frac{2C_f}{m}\delta_f \tag{12a}$$

$$\ddot{\psi} = -2\left(\frac{-C_rb+C_fa}{I_z\dot{x}}\right)\dot{y} - \left(\frac{2C_fa^2-2C_rb^2}{I_z\dot{x}}\right)r + \frac{C_fa}{I_z}\delta_f \tag{12b}$$

State Equations

The following state equations represent a linearized time varying model which is used in our LTV based MPC. The computational load of a nonlinear MPC is troublesome to implement in an online mobile environment, in this case, a car. Hence, this literature suggests linearizing the vehicle model in order to reduce this computational load in expense of controller accuracy. Although, experimental results suggest that the difference between a NLMPC and LTV-MPC fall between acceptable limits of errors and hence, the latter is preferred in many literatures [] [] []. In the later part of the report, the below states have been described with reference to the specific controller and simulations are conducted on the MATLAB/Simulink interface.

$$\xi(t) = A(t)X + B(t)\delta_f \tag{13a}$$

$$\alpha(t) = C(t)\xi(t) + D(t)\delta_f \tag{13b}$$

Here,

$$A = \begin{bmatrix} -2\left(\frac{C_r+C_f}{m\dot{x}(t)}\right) & 0 & -\left(\frac{2(C_fa-C_rb)}{m\dot{x}(t)} + \dot{x}(t)\right) & 0 \\ 0 & 0 & 1 & 0 \\ -\left(\frac{-2(C_rb+C_fa)}{I_z\dot{x}(t)}\right) & 0 & -\left(\frac{2(C_fa^2-C_rb^2)}{I_z\dot{x}(t)}\right) & 0 \\ 1 & \dot{x}(t) & 0 & 0 \end{bmatrix} \tag{13c}$$

$$B = \begin{bmatrix} \frac{2C_f}{m} \\ 0 \\ \frac{C_fa}{I_z} \\ 0 \end{bmatrix} \tag{13d}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{13e}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{13f}$$

TRAJECTORY GENERATION (DOUBLE CHANGE SCENARIO)

In this paper, a custom path has been generated as a reference for simulation inputs. This desired path is generated based on the lateral position Y_{ref} and yaw angle ψ_{ref} mapped as a function of *X*. The following are the equations used for trajectory generation,

$$\psi_{ref}(X) = \frac{d_{y_1}}{2}(1 + \tanh(z_1)) - \frac{d_{y_2}}{2}(1 + \tanh(z_2))$$

$$Y_{ref}(X) = \arctan\left(d_{y_1}\left(\frac{1}{\cosh(z_1)}\right)^2\left(\frac{1.2}{d_{x_1}}\right) - d_{y_2}\left(\frac{1}{\cosh(z_2)}\right)^2\left(\frac{1.2}{d_{x_2}}\right)\right)$$

Here, $z_1 = \left(\frac{2.4}{25}\right)(X - 27.19) - 1.2$, $z_2 = \left(\frac{2.4}{21.95}\right)(X - 56.46) - 1.2$, $d_{x_1} = 25$, $d_{x_2} = 21.95$, $d_{y_1} = 4.05$ and $d_{y_2} = 5.7$.

X (meters)	$\psi_{ref}(X)$ (deg)	$Y_{ref}(X)$ (meters)
5	0.1344	0.0099
10	0.2744	0.0257
15	0.7276	0.0666
20	1.7719	0.1691
25	4.0099	0.4112
30	7.2189	0.9001
35	6.8963	1.5741
40	-0.9370	1.8887
45	-8.5861	1.4145
50	-9.9232	0.5650
55	-11.8339	-0.3520
60	-14.2681	-1.5485
67.9	-1.1234	-2.4
76.3	1.7619	-2.2
83.5	0.2126	-2.1

Procedure to generate reference trajectory:

1. Use the above $\psi_{ref}(X)$ and $Y_{ref}(X)$ to generate waypoints for the car's trajectory. The MATLAB code used to generate trajectory waypoints is as follows:

```
X = [5,10,15,20,25,30,35,40,45,50,55,60,67.9,76.3,83.5];
dy1 = 4.05;
dy2 = 5.7;
dx2 = 21.95;
dx1 = 25;
z1 = ((2.4/25)*(X-27.19))-1.2;
z2 = ((2.4/21.95)*(X-56.46))-1.2;

psi = ((dy1)/2).*(1+tanh(z1))-((dy2)/2).*(1+tanh(z2));
```

```
Y = atan((dy1*((1./cosh(z1)).^2)*(1.2/dx1))-(dy2*((1./cosh(z2)).^2)*(1.2/dx2)));

p = psi' .* (180/pi);
y = 10.* Y';
```

2. Use the MATLAB's Driving Scenario Generator to add a road and an actor (car). Add the above generated waypoints in the actor's trajectory.

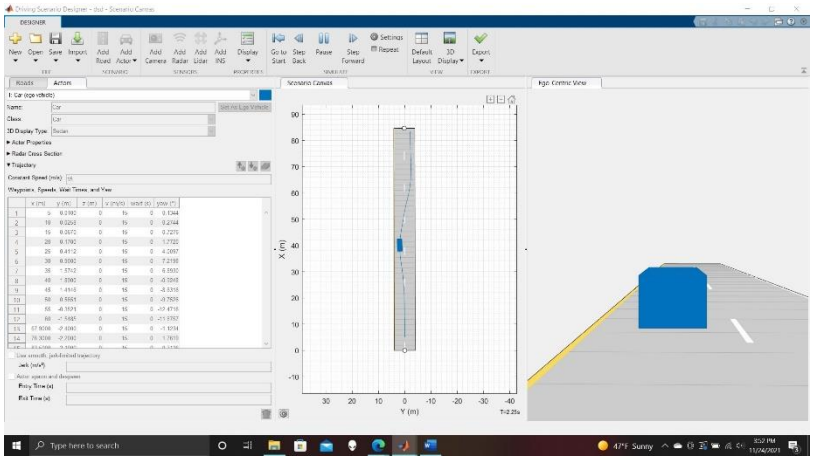


Figure 3: MATLAB Driving Scenario Designer

3. Save and export as a MATLAB function. The generated code should look similar to as shown below:

```
function [scenario, egoVehicle] = dsd_new()

scenario = drivingScenario;

% Add all road segments
roadCenters = [0 0 0;
    84.7 0.2 0];
laneSpecification = lanespec(2, 'Width', 3.925);
road(scenario, roadCenters, 'Lanes', laneSpecification, 'Name', 'Road');

% Add the ego vehicle
egoVehicle = vehicle(scenario, ...
    'ClassID', 1, ...
    'Position', [5 0.01 0], ...
    'Mesh', driving.scenario.carMesh, ...
    'Name', 'Car');
waypoints = [5 0.01 0;
    10 0.0258 0;
    15 0.067 0;
    20 0.17 0;
    25 0.4112 0;
    30 0.9 0;
    35 1.5742 0;
    40 1.89 0;
    45 1.4146 0;
    50 0.5651 0;
    55 -0.3521 0;
    60 -1.5485 0;
    67.9 -2.4 0;
    76.3 -2.2 0;
    83.5 -2.1 0];
speed = [15;15;15;15;15;15;15;15;15;15;15;15;15;15;15];
waittime = [0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
trajectory(egoVehicle, waypoints, speed, waittime);
```

4. Open the reference generator file and make sure that it calls the generated function. Run the 'Reference_generator.m' file. This will generate a '.mat' with all the generated waypoints. Link this file to

the trajectory's Simulink block. This block with give the Lateral position and Yaw angle as reference to the MPC controller.

CONTROLLER DESIGN

Non-Linear MPC

$\xi = \{\dot{Y}, \dot{\psi}, Y, \psi\}$

The state comprises of lateral velocity, yaw rate, lateral position and yaw angle

The system dynamics that are obtained earlier are discretized with fixed sampling time T_s

$$\xi(k+1) = f^{dt}_{a(k),\mu(k)}(\xi(k), \Delta u(k)) \tag{14a}$$

$$u(k) = u(k-1) + \Delta u(k) \tag{14b}$$

Here,

$\Delta u(k)$ is the change in the inputs at each time step

$\xi(k)$ are the state vectors

$u(k)$ is the input to the system which is the steering angle

$$\eta(k) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \xi(k) = [\psi, Y]' \tag{15}$$

Here, ψ is the yaw angle

Y is the lateral position

$\eta(k)$ is the output map that comprises of yaw angle and the lateral position states

$$J(\xi(t), \Delta U_t) = \sum_{i=1}^{H_p} \left\| \eta_{t+i,t} - \eta_{(ref)t+i,t} \right\|_Q^2 - \sum_{i=0}^{H_C-1} \left\| \Delta u_{t+i,t} \right\|_R^2 \tag{16}$$

Here, $J(\xi(t), \Delta u_t)$ is the cost function for NLMPC

The first summand reflects the penalty on trajectory tracking error while the second summand is a measure of the steering effort.

$\Delta U_t = [\Delta u_{t,t} \ldots\ldots \Delta u_{t+H_c-1,t}]$ is the optimization vector at time t

$\eta_{t+i,t}$ denotes the predicted state at time t + i obtained by applying the state $\xi_{t,t}$ to equation (14a) and (14b)

$\bar{u} = \{u_{t,t} \ldots\ldots u_{t+k,t}\}$ input to the system for entire time duration

H_p and H_C denote the prediction horizon and control horizon, respectively.

Q, R are weights of appropriate dimension penalizing state tracking error, control action respectively.

To reduce the error between the reference signal and predicted output at each time step we formulate the nominal optimization problem with tightened constraints as

$$\min J(\xi_t, \Delta U_t)$$

subject to,

$$\varepsilon_{k+1,t} = f^{dt}_{s_{k,t},\mu_{k,t}}(\xi_{k,t}\, \Delta u_{k,t})$$

$$\eta_{k,t} = h(\xi_{k,t})$$

In the nonlinear MPC the coefficient of friction ($\mu_{t,t}$) between the tires and ground and the slip ratio ($s_{t,t}$) varies with time which makes tire forces variable. Between two time steps the coefficient of friction and slip ratio is nonlinear which makes the system nonlinear.

$$\mu_{k,t} = \mu_{t,t}$$

$$s_{k,t} = s_{t,t}$$

Where k = t …. t + H_p

The constraints that are applied while solving the optimal control problem.

$$\delta_{f,min} \leq u_{k,t} \leq \delta_{f,max}$$

$$\Delta\delta_{f,min} \leq \Delta u_{k,t} \leq \Delta\delta_{f,max}$$

$$u_{k,t} = u_{k-1,t} + \Delta u_{k,t}$$

Here,

$$\text{k} = \text{t}\ldots\text{t} + H_c - 1$$

The input signal is assumed to be constant between $H_c \leq t \leq H_p$

$$\text{Therefore}\quad \Delta u_{k,t} = 0, \text{ where k} = \text{t} + H_c \ldots\ldots \text{t} + H_p$$

$$\xi_{t,t} = \xi(t)$$

$$\Delta U_t^* = [\Delta u_{t,t}^* \ldots\ldots \Delta u_{t+H_c-1,t}^*]'$$

Here ΔU_t^* represents sequence of optimal input increments by solving the cost function for current states $\xi(t)$

The first sample of ΔU_t^* ($\Delta u_{t,t}^*$) is used to compute the optimal control action and the resulting state feedback control law is

$$u(t, \xi(t)) = u(t-1) + \Delta u^*{}_{t,t}(t, \xi(t)) \tag{17}$$

Linear-Time-Variant MPC

The system dynamics that are obtained earlier are discretized with fixed sampling time T_s

$$\xi(k+1) = f^{dt}_{a(k),\mu(k)}(\xi(k), \Delta u(k)) \tag{18a}$$

$$u(k) = u(k-1) + \Delta u(k) \tag{18b}$$

Here,

$\Delta u(k)$ is the change in the inputs

$\xi(k)$ are the state vectors

$u(k)$ are the inputs to the system

The cost function is found for discretized states for each time step

$$J(\delta\xi(t), \delta U_t) = \sum_{i=1}^{H_p} \left\| \delta\eta_{t+i,t} - \delta\eta_{(ref)t+i,t} \right\|_Q^2 - \sum_{i=0}^{H_C-1} \left\| \delta u_{t+i,t} \right\|_R^2 + \rho\epsilon \quad (20)$$

Here, $J(\varepsilon(t), \Delta u_t)$ is the cost function for Linear Time Variant MPC

ϵ is a slack variable. The term $\rho\epsilon$ penalizes the violation of the constraint on the slip angle and ρ is a weight coefficient

By linearizing the above model (14a) and (14b) at each time step around the point $\xi(t)$ and u(t-1) we get,

Subject to,

$$\delta\xi_{k+1,t} = A_t\delta\xi_{k,t} + B_t\delta u_{k,t} \quad (21a)$$

$$\delta\alpha_{k,t} = C_t\delta\xi_{k,t} + D_t\delta u_{k,t} \quad (21b)$$

$$\delta\eta_{k,t} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \delta\xi_{k,t} = [\delta\psi, \delta Y]' \quad (21c)$$

Here, k = t …... t + H_p

Here,

$$u_{k,t} = u_{t-1,t} + \delta u_{k,t} \quad (21d)$$

$$\Delta u_{k,t} = u_{k,t} - u_{k-1,t} \quad (21e)$$

Constraints Applied:

$$\delta_{f,min} \leq u_{k,t} \leq \delta_{f,max} \quad (21f)$$

$$\Delta\delta_{f,min} \leq \Delta u_{k,t} \leq \Delta\delta_{f,max}$$

Here, k = t …... t + $H_c - 1$

$$\delta\alpha_{min} - \varepsilon \leq \delta\alpha_{k,t} \leq \delta\alpha_{max} + \varepsilon \quad (21g)$$

Here, k=t……...t + H_p

$$\Delta u_{k,t} = 0$$

Here, k =t+H_c…t + H_p

$$\varepsilon \geq 0$$

$$\delta\xi_{t,t} = 0$$

$$\Delta U_t = [\delta u_{t,t} \dots\dots \delta u_{t+H_c-1,t}]'$$

The optimization problem can be recast as quadratic program

$$\Delta U_t^* = [\delta u_{t,t}^* \dots\dots \delta u_{t+H_c-1,t}^*]'$$

Where ΔU_t^* is the sequence of optimal inputs deviations and $\delta u_{t,t}^*, \dots \dots, \delta u_{t+H_c-1,t}^*$ are all optimal inputs deviations that are computed by solving the equation (20) for current state $\xi(t)$.

For the next time step t+1 the optimization problem in equation (20) is solved over the next predicted horizon for new measurements with states $\xi(t+1)$.The first sample of ΔU_t^* is used to calculate optimal control action and the resulting control feedback law is given by

$$u\big(t,\xi(t)\big) = u(t-1) + \delta u_{t,t}^*(t,\xi(t)) \tag{22}$$

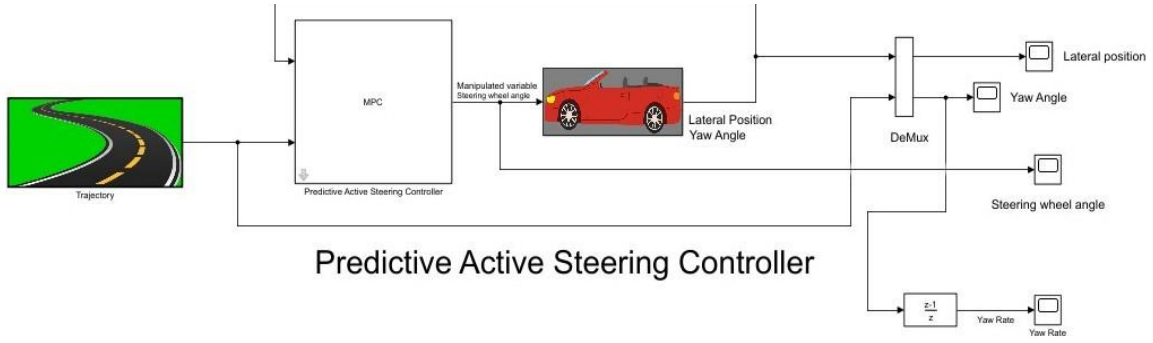
The model (18) is linearized around an operating point therefore the equations (13c), (13d) become linear time invariant and they are used to find the output and the states instead of an LTI model $\hat{\xi}(k)$, $\hat{\eta}(k)$, $\hat{\alpha}(k)$ over the prediction horizon for k = t, ……. t + H_p.

The variation in the input from its previous input is given by optimization variables $[\delta u_{t,t}, \dots\dots, \delta u_{t+H_c-1,t}]$ as stated in equation (21d)

SIMULATION

Controller A

Figure 4: Controller A block diagram



The following controller is a LTI MPC controller. Here, we input the controller with a reference trajectory which references the yaw angle and the lateral position as per the planned trajectory. We consider a constant longitudinal velocity of $\dot{x} = 15\ m/s$. The controller further outputs the manipulated variable (steering angle δ_f) and feeds it to the plant. Since the input velocity is constant, the state matrices *A, B, C, D* remain constant for the whole maneuver. The plant outputs the measured lateral position *Y* and the yaw angle ψ. The following are the parameters considered for this simulation,

T = 0.05s, H_p = 7, H_c = 2, -30° < δ_f < 30°, -15° < $\Delta\delta$ < 15°, $\mu = 0.3$.

Here,

T – Sampling time

H_p – Prediction Horizon

H_c – Control Horizon

μ – Coefficient of friction between the road and tire

The weighing matrix for the minimization of the cost function is take as,

$Q = \begin{bmatrix} 500 & 0 \\ 0 & 75 \end{bmatrix}, R = 150$ and the slack variable as 10

Apart from the above-mentioned constraints, the following values have been feed into the state's *A, B, C, D* matrices:

C_f = 19000 N/m, C_r = 33000 N/m, I_z = 3344 kg/m^2, $\dot{x} = 15$ m/s, a = 1.2 m, b = 1.6 m, m = 2050 kgs

Here

C_f – Front Tire Cornering Stiffness Coefficient

C_r – Rear Tire Cornering Stiffness Coefficient

I_z – Moment of Inertia of vehicle

$\dot{x}$ – Velocity of vehicle

a – Distance of front tire from center of mass of vehicle

b – Distance of rear tire from center of mass of vehicle

m – Mass of vehicle

Results of Controller A

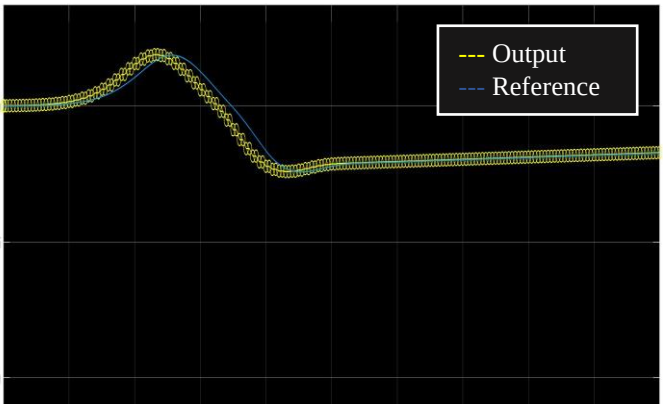


Figure 5: Lateral Position

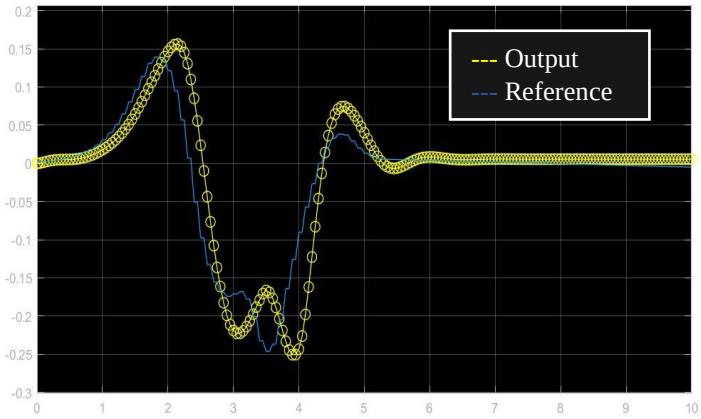


Figure 6: Yaw Angle

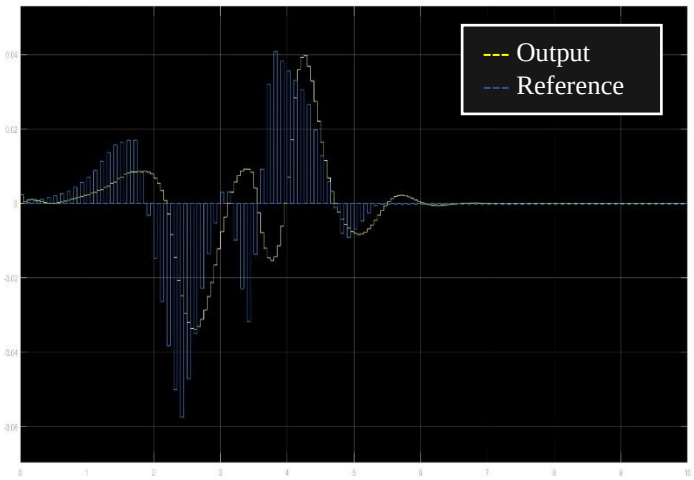


Figure 7: Yaw rate

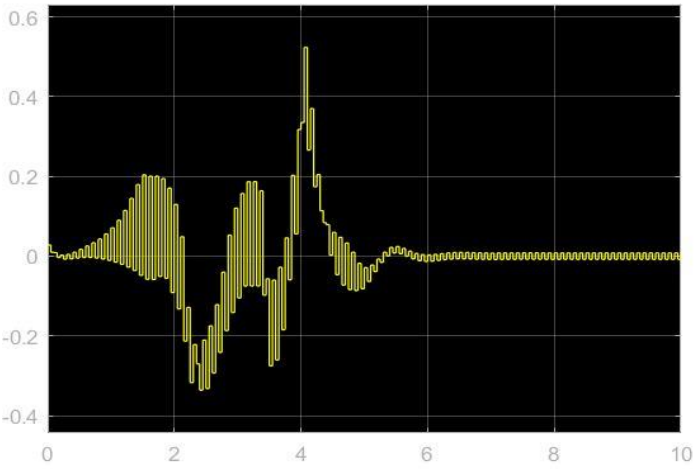


Figure 8: Front Steering angle

Lateral Position: Figure 5 provides the graph for the reference lateral position in comparison with the output position.

Yaw Angle: Figure 6 provides the graph for the reference yaw angle in comparison with the actual yaw angle.

Yaw rate: Figure 7 provides the graph for the reference yaw rate in comparison with the actual yaw rate.

Steering angle: Figure 8 provides how the manipulated variable keeps changing with time to keep the system stable.

Controller B

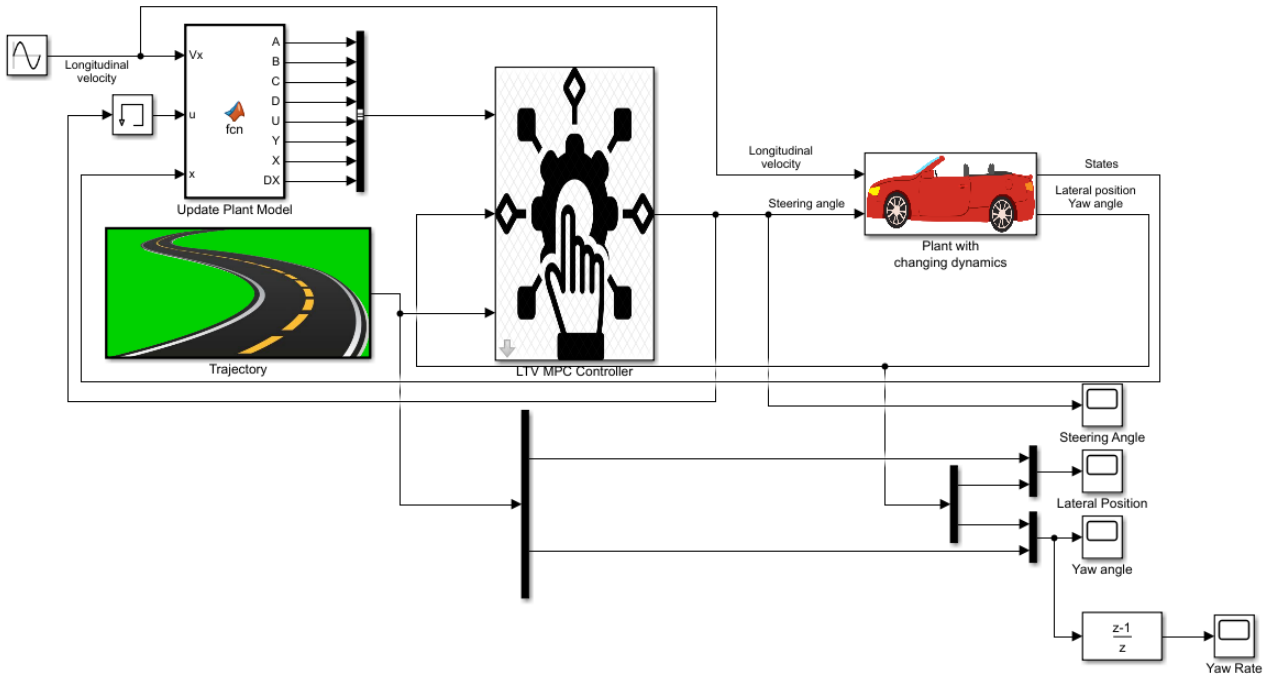


Figure 9 Controller B Simulink Model

The input signal that is provided to the LTV Model must be a discrete signal hence we converted the continuous input signal to discrete signal. Following is the MATLAB code to convert continuous signal to discrete signal:

```
ct_plant = mpc1.Model.Plant;

dt_plant = c2d(ct_plant,Ts);

mpc1.Model.Plant = dt_plant;
```

We have generated a trajectory using MATLAB Driving Scenario Designer App as shown in the trajectory planning section, this app generates the trajectory object that stores the lateral position and yaw angle of the vehicle at each time step over the entire duration of the trajectory. We have stored all the lateral position and the yaw angle of the vehicle at each instance in the posRef and yawRef array.

To load the waypoints data from workspace (posRef, yawRef) into the simulink we used from workspace block parameters. We have used the mux to combine the yaw angle and lateral position data into a single output. We have created an mpc object with all the a parameters and it is fed to the model predictive controller block. The mpc object stores the information about control horizon, predictive horizon, sample time, weights.

The following are the parameters we have set that is stored in the mpc object

$T_s = 0.05$

Prediction Horizon (H_p) $= 25$

Control Horizon (H_c) = 10

Weight Matrices Q = $\begin{bmatrix} 200 & 0 \\ 0 & 100 \end{bmatrix}$ and R = 50000

Weight Coefficient ρ = 1000

Constraints:

$\delta_{f,min} = -30^o$, $\delta_{f,max} = 30^o$

$\Delta\delta_{f,min} = -15^o$, $\Delta\delta_{f,max} = 15^o$

Coefficient of friction (μ) = 0.3

Apart from the above-mentioned constraints, the following values have been feed into the state's *A, B, C, D* matrices:

C_f = 19000 N/m, C_r = 33000 N/m, I_z = 3344 kg/m^2, $\dot{x} = 15$ m/s, a = 1.2 m, b = 1.6 m, m = 2050 kgs

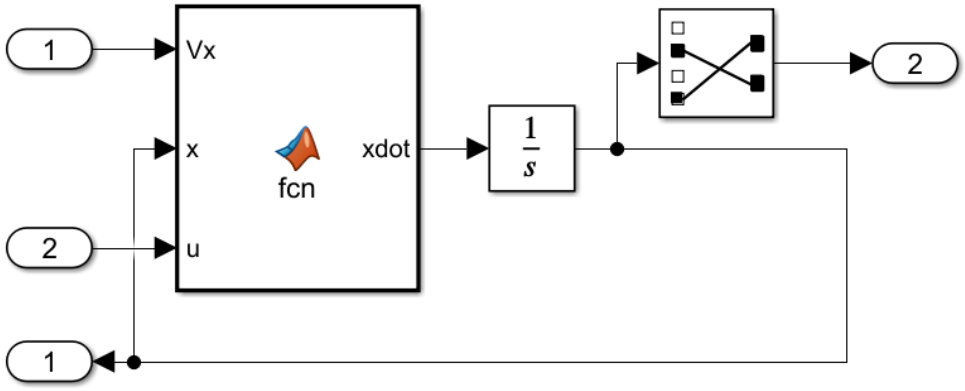


Figure 10 Vehicle Model

The input of the vehicle model is the steering angle (δ_f) from the controller. In LTV MPC the longitudinal velocity is varying with time hence we have taken a sinusoidal signal as our longitudinal velocity input. In plant model we created a function block to compute the change in state and we have provided the block with the input to the plant model i.e., steering angle and the longitudinal velocity.

The A, B, C, D matrix are calculated from equations (13c) (13d), (13e), (13f). As the longitudinal velocity is varying with time the A and B matrix are also time variant. We have used an integrator block to get the states. From the updated states we have selected the yaw angle (ψ) and lateral position(Y) that is passed to the controller to create a new input to the plant. Following code shows how the states are calculated that are updated in the update plant model.

```
function xdot = fcn(Vx,x,u)
% Model parameters
m = 2050;
Iz = 3344;
lf = 1.2;
lr = 1.6;
Cf = 19000;
Cr = 33000;
% Continuous-time model
A = [-(2*Cf+2*Cr)/m/Vx, 0, -Vx-(2*Cf*lf-2*Cr*lr)/m/Vx, 0;
     0, 0, 1, 0;
     -(2*Cf*lf-2*Cr*lr)/Iz/Vx, 0, -(2*Cf*lf^2+2*Cr*lr^2)/Iz/Vx, 0;
     1, Vx, 0, 0];
B = [2*Cf/m 0 2*Cf*lf/Iz 0]';
C = [0 0 0 1; 0 1 0 0];
D = zeros(2,1);
xdot = A*x + B*u;
```

The LTV MPC provides a new plant model at each time step as the operating conditions change. Therefore, it makes

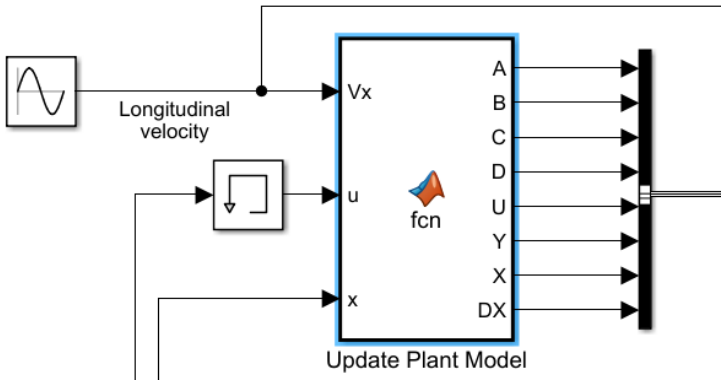


Figure 11 Updating Vehicle Model

more accurate predictions for new operating conditions. We have created a new internal plant model which keeps on updating with the operating conditions. For the internal plant model, we have developed function block that takes input parameters as longitudinal velocity (Vx), states(x) and steering angle(x) as inputs. This block provides an internal plant model to the controller which is updated at each time step.

Code for updating the internal plant model at each time step:

```
function [A,B,C,D,U,Y,X,DX] = fcn(Vx,u,x)
% Sample time
Ts = 0.05;
% Model parameters
m = 2050;
Iz = 3344;
```

```
lf = 1.2;

lr = 1.6;

Cf = 19000;

Cr = 33000;

% Continuous-time model

Ac = [-(2*Cf+2*Cr)/m/Vx, 0, -Vx-(2*Cf*lf-2*Cr*lr)/m/Vx, 0;

     0, 0, 1, 0;

     -(2*Cf*lf-2*Cr*lr)/Iz/Vx, 0, -(2*Cf*lf^2+2*Cr*lr^2)/Iz/Vx, 0;

     1, Vx, 0, 0];

Bc = [2*Cf/m 0 2*Cf*lf/Iz 0]';

Cc = [0 0 0 1; 0 1 0 0];

Dc = zeros(2,1);

% Generate discrete-time model

nx = size(Ac,1);

nu = size(Bc,2);

M = expm([[Ac Bc]*Ts; zeros(nu,nx+nu)]);

A = M(1:nx,1:nx);

B = M(1:nx,nx+1:nx+nu);

C = Cc;

D = Dc;

% Nominal conditions for discrete-time plant

X = x;

U = u;

Y = C*x + D*u;

DX = A*x+B*u-x;
```

Results of Controller B

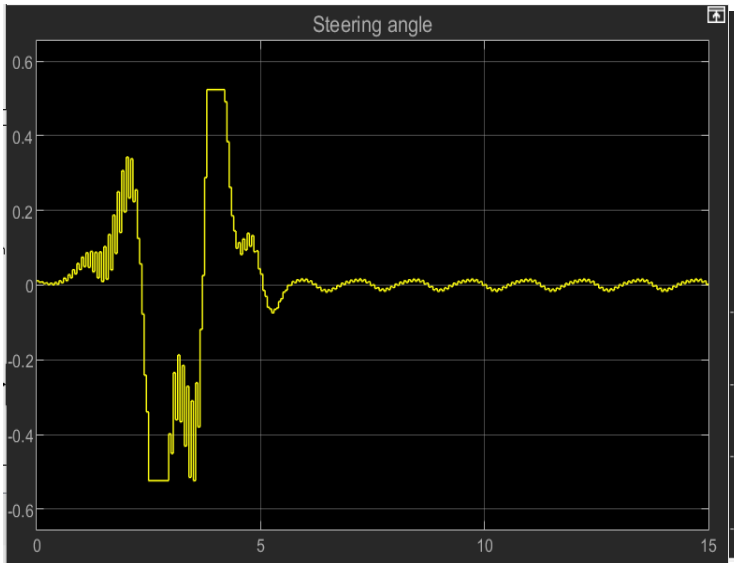


Figure 12 Steering Angle

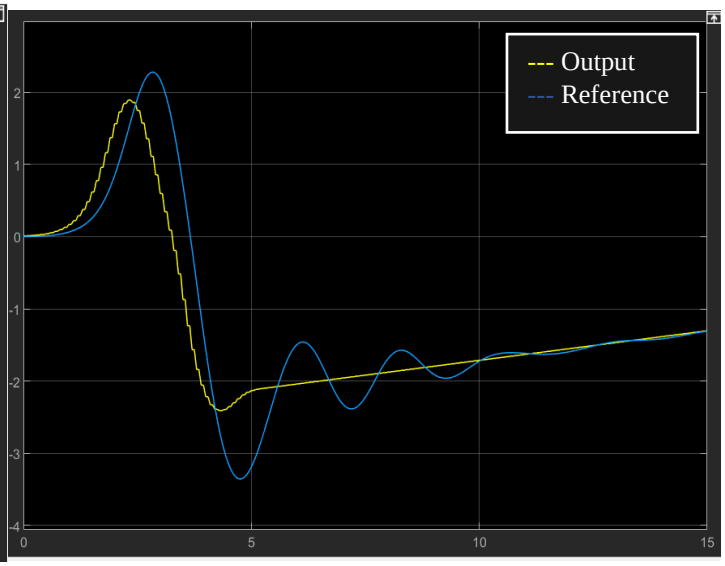


Figure 13 Lateral Position

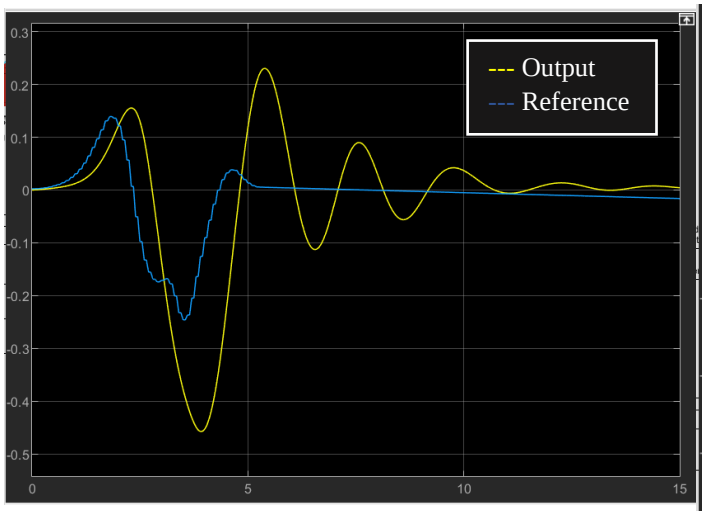


Figure 14 Yaw Angle

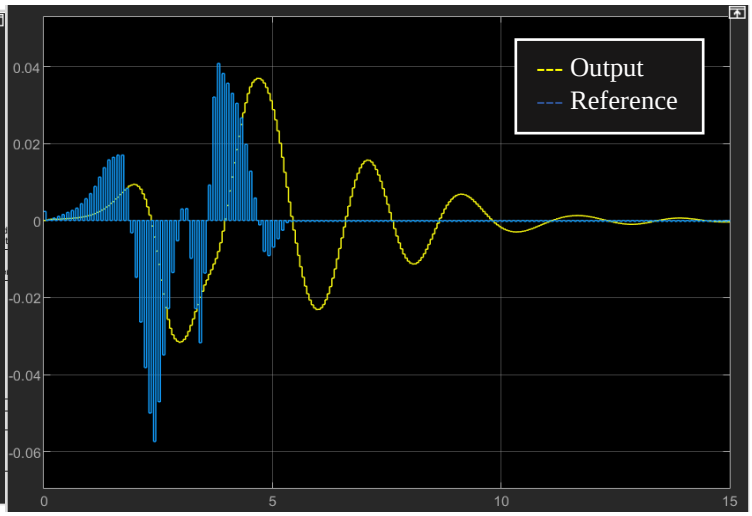


Figure 15 Yaw Rate

Steering angle: Figure 12 provides how the manipulated variable keeps changing with time to keep the system stable.

Lateral Position: Figure 13 provides the graph for the reference lateral position in comparison with the output position.

Yaw Angle: Figure 14 provides the graph for the reference yaw angle in comparison with the actual yaw angle.

Yaw rate: Figure 15 provides the graph for the reference yaw rate in comparison with the actual yaw rate.

Controller C

Controller C is designed in the same manner as controller B but the control horizon of Controller C is kept as 1, i.e.,

$H_c = 1$

Results of Controller C

Figure 16: Steering Angle

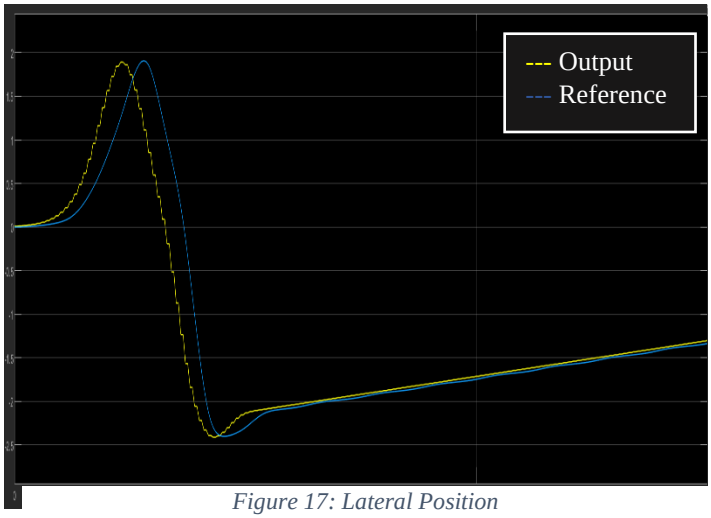


Figure 17: Lateral Position

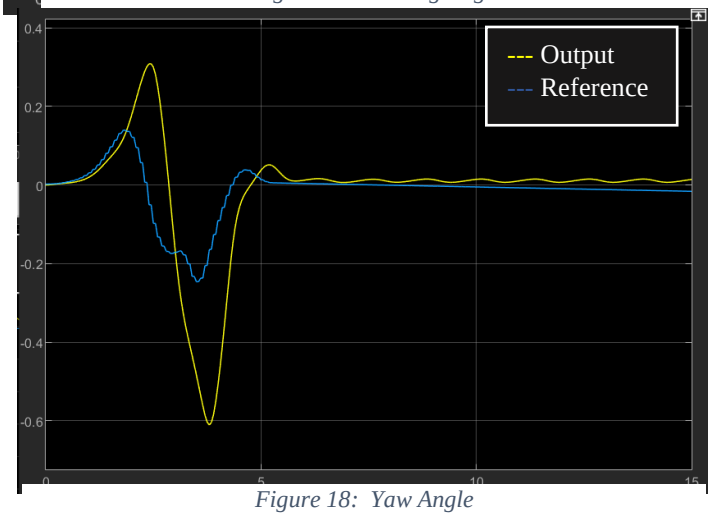


Figure 18: Yaw Angle

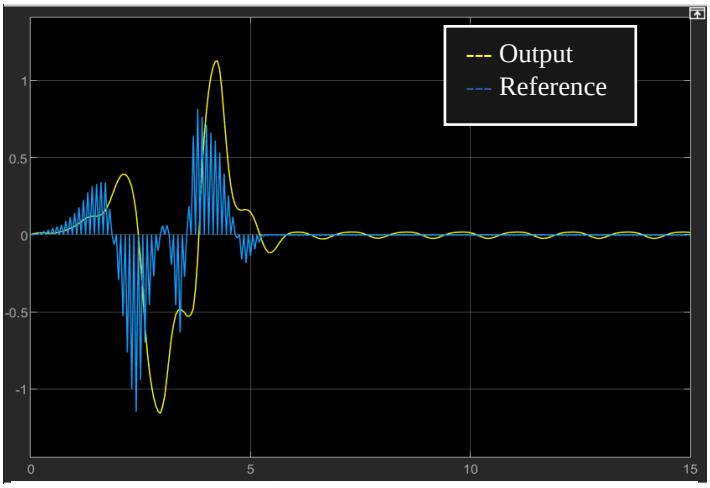


Figure 19: Yaw Rate

Steering angle: Figure 16 provides how the manipulated variable keeps changing with time to keep the system stable.

Lateral Position: Figure 17 provides the graph for the reference lateral position in comparison with the output position.

Yaw Angle: Figure 18 provides the graph for the reference yaw angle in comparison with the actual yaw angle.

Yaw rate: Figure 19 provides the graph for the reference yaw rate in comparison with the actual yaw rate.

DISCUSSIONS AND CONCLUSIONS

This report has represented the implementation of a Predictive Active Steering Controller which makes use of the bicycle model to simulate a double lane changing scenario on highly slippery roads, such as the snow road. The framework uses a Model Predictive Control on a nonlinear vehicle model. This vehicle model is further linearized to perform simulations on an LTV MPC controller and an LTI MPC controller.

Simulations results suggest that the LTV MPC system provides a satisfactory control result with the added advantage of faster computations as compared to the NLMPC. We further simplify the model into an LTI MPC and conclude that for low-speed maneuver, the LTI MPC controller provides an adequate performance and consumes the least computational resources. This means that for a mobile system such as a car with limited computational power and energy, the LTI and LTV systems work seamlessly while providing acceptable results. The LTV MPC can show accurate results with varying velocity which is not possible with LTI MPC.

The literature makes use of a rapid prototyping system dSPACE Autobox for real-world simulation of the NLMPC. This uses real time data to capture the changing slip ratios and coefficient of friction; the parameters which make the system nonlinear. The system makes use of various sensors and technologies such as the IMU to calculate nonlinear and predictable parameters in real time, such as the yaw rate. Such a system includes a trajectory replanning module where the MPC computes anew trajectory based on the online sensors' information using the receding horizontal control design. In such a system, the trajectory planning problem is formulated as constrained optimization problem where it minimizes the controller efforts using a weighted sum.

We conclude that the effectiveness of a MPC framework to control an AFS for a lane changing maneuver is as like a skilled driver and such a system can be implemented in an autonomous/semi-autonomous car either as a support system or an independent main system.

BIBLIOGRAPHY

[1] Vehicle Dynamics: The Dynamic Bicycle Model (thef1clan.com)

[2] Bimbraw, Keshav. (2015). Autonomous Cars: Past, Present and Future - A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology. ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings. 1. 191-198. 10.5220/0005540501910198.

[3] Feng, Jiwei & Bao, Chunjiang & Wu, Jian & Cheng, Shuo & Xu, Guangfei & Liu, Shifu. (2018). Research on Methods of Active Steering Control Based on Receding Horizon Control. Energies. 11. 2243. 10.3390/en11092243.

[4] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng and D. Hrovat, "Predictive Active Steering Control for Autonomous Vehicle Systems," in IEEE Transactions on Control Systems Technology, vol. 15, no. 3, pp. 566-580, May 2007, doi: 10.1109/TCST.2007.894653.

[5] J. Ackermann and W. Sienel, "Robust yaw damping of cars with front and rear wheel steering," in IEEE Transactions on Control Systems Technology, vol. 1, no. 1, pp. 15-20, March 1993, doi: 10.1109/87.221348.
[6] J. Ackermann, "Robust control prevents car skidding," in IEEE Control Systems Magazine, vol. 17, no. 3, pp. 23-31, June 1997, doi: 10.1109/37.588073.
[7] Zhang, Wei-Bin & Parsons, Robert & West, Thomas. (1990). An Intelligent Roadway Reference System for Vehicle Lateral Guidance/Control. 281 - 286. 10.23919/ACC.1990.4790741.
[8] P. Polack, F. Altché, B. d'Andréa-Novel and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?," 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 812-818, doi: 10.1109/IVS.2017.7995816.
[9] Cho, J., Huh, K. Active Front Steering for Driver's Steering Comfort and Vehicle Driving Stability. Int.J Automot. Technol. 20, 589–596 (2019). https://doi.org/10.1007/s12239-019-0056-1
[10] Zheng, B. et al. "Active steering control with front wheel steering." *Proceedings of the 2004 American Control Conference* 2 (2004): 1475-1480 vol.2.
[11] Jianhua Guo, Liang Chu, Hongwei Liu, Mingli Shang and Yong Fang, "Integrated control of Active Front Steering and Electronic Stability Program," 2010 2nd International Conference on Advanced Computer Control, 2010, pp. 449-453, doi: 10.1109/ICACC.2010.5486880.
[12] Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPCbased approach to active steering for autonomous vehicle systems," Int. J. Veh. Autonomous System., vol. 3, no. 2/3/4, pp. 265–291, 2005.
[13] Lee J, Chang H-J (2018) Explicit model predictive control for linear time-variant systems with application to double-lane-change maneuver. PLoS ONE 13(12): e0208071. https://doi.org/10.1371/journal.pone.0208071
[14] Gao, Y. (2014). Model Predictive Control for Autonomous and Semi Autonomous Vehicles. UC Berkeley. ProQuest ID: Gao_berkeley_0028E_14270. Merritt ID: ark:/13030/m5mw5mns. Retrieved from https://escholarship.org/uc/item/8xd0b56h