Explicit Model Predictive Control for Lateral Control of Autonomous Vehicles

Saif Saiyed
Master of Engineering, Department of Mechanical and Mechatronics Engineering, University of Waterloo | UWaterloo ID: 20793112 | sa2saiye@uwaterloo.ca

Abstract
In this paper, autonomous lateral control of intelligent vehicles based on explicit model predictive control (E-MPC) is developed to implement optimal path tracking for lane changing scenario. For this purpose, a reference trajectory, which in practice is obtained by vehicle perception model and lane detection, was generated mathematically. Then, an implicit (online) model predictive control (I-MPC) with lateral dynamic vehicle model was proposed. I-MPC was first tuned until the satisfactory results are obtained. Then, the problem was solved as multi-parametric quadratic programs (mp-QP) to generate a set of explicit (offline) solutions. The polyhedral partition of the state space induced by the multi-parametric piecewise affine solution were generated to visualize the control scheme. Last, both control schemes were compared based on computation time and memory occupied.

Introduction
Application of microcontroller and computer technology with faster processing power and advancement in GPS and other mapping technologies over the past few decades has allowed the control engineers liberty to implement advanced control techniques in self-driving vehicles. The control of ground vehicles can be separated into lateral control and longitudinal control. The purpose of this research is to generate and implement optimal control law with minimal computation time for lane changing scenario for self-driving vehicles.

As shown in figure 1, reference trajectory was generated and plant parameters were defined first to develop I-MPC.

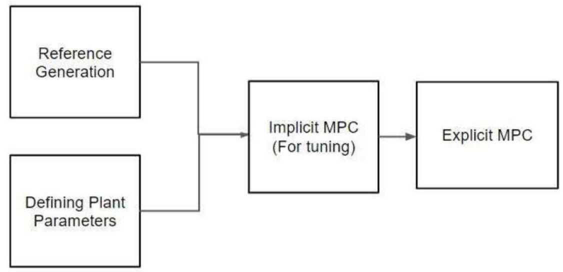

Figure 1 Walkthrough of EMPC control scheme

MPC controller is implemented due to its unique property to deal with constraints, both on the state variables and on the control input signals. This has a practical significance vehicle since a vehicle is constrained not only by the limitation of its mechanism (for instance, maximum amount torque an engine can produce or permissible range of steering) but also by the factors from the environment (for example, speed limit on a particular road). In MPC, the control problem is formulated as an optimization problem and solving it to generate an optimal control law. However, the difference is that the optimization is done on-line in contrast to determining an off-line feedback policy [1]. Linear MPC can be formulated as:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & z_{Hp}^T Q_f z_{Hp} + \sum_{i=1}^{H-1} z_i^T Q z_i + u_i^T R u_i \\ \text{subject to} \quad & z_{i+1} = Az_i + Bu_i \\ & Cz_i + Du_i \leq b \end{aligned}$$

where Q_f, Q and R are the terminal state weighting matrix, state weighting matrix (penalizing performance) and control input weighting matrix (penalizing cost), and $Q_f \geq 0$, $Q \geq 0$, $R > 0$. This optimization problem is subjected to the control oriented model (prediction model) given by $z_{i+1} = Az_i + Bu_i$ and a set of constraints in the form of $Cz_i + Du_i \leq b$.

I-MPC strategy described above requires running an optimization algorithm on-line at each time-step t by considering the value of the current state vector x(t) as initial state value x(0). For this reason, I-MPC demands a significant amount of processing power.

The idea of explicit MPC is to solve the optimization problem off-line for all x(t) within a given set X, that is assumed to be polytopic

$$X = \{x \in \mathbb{R}^n : S_1 x \leq S_2\} \subset \mathbb{R}^n$$

and to make the dependence of u(t) on x(t) explicit, rather than implicit as defined by traditional MPC. It appeared, as perceived in [2], this relationship is a piecewise affine. Hence, the MPC control scheme can also be represented in an equivalent way as

$$u(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq k_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if } H_M x \leq k_M . \end{cases}$$

where F and G are constants defining piecewise affine relationship between control input and parameterised state space. Consequently, on-line computation is reduced to simply evaluating the above set of constants which expands the application of MPC to fast processes. Explicit solutions of this form can be evaluated by solving multiparametric problems. MPC toolbox and Multi-Parametric Toolbox (MPT) of MATLAB were used to generate implicit and explicit MPC for this problem.

Plant Model and Plant Parameters

I. Formulating Plant Model

One of the most crucial characteristics in formulating MPC problem is defining the prediction model of the system, also known as control oriented model. This model predicts the future states of the system which would be used to generate optimal control law, hence it needs a certain level of precision to capture the dominating dynamics of the plant. On the contrary, the processing power requirement should not be too high. If the prediction model is too computational expensive, the time it takes to solve the optimization problem would be too long, while the time to perform the optimization should be less than the sample time of the system.

For this thesis, dynamic bicycle model is used which is one of the simplest models to represent lateral dynamics of ground vehicles. It is a simplified 3-DOF model which considers only two wheels, one front and one rear wheel at the center axis, thus the name 'bicycle model'. Since the dynamic bicycle model describes the system by the forces acting on it, the tire forces have to be considered. These forces are highly non-linear but we restrict ourselves by linearizing the model at a particular operating point. The effects of non-linearity can be better modelled by embedding the Pacejka Model for the tire forces [3]. However, the use of such non-linear models for the tire forces are computationally cumbersome.

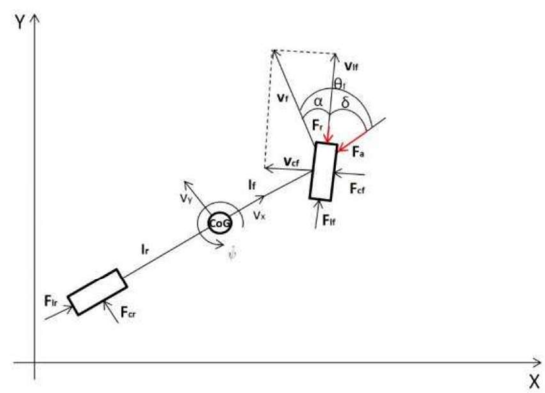

Figure 2 Vehicle represented by the bicycle model

Figure 2 shows bicycle model to formulate lateral control dynamics of a vehicle [4]. In order to fulfil the requirement of accuracy and at the same time not be too computational expensive, aerodynamic drag (F_a) and rolling resistance (F_r), shown with red arrows in Figure 2, are neglected. Differential equation based on the bicycle model can be written as:

$$
\begin{aligned}
m\dot{v}_x &= m v_y \dot{\psi} + 2F_{xf} + 2F_{xr} \\
m\dot{v}_y &= -m v_x \dot{\psi} + 2F_{yf} + 2F_{yr} \\
I\ddot{\psi} &= 2l_f F_{yf} - 2l_r F_{yr}
\end{aligned}
$$

The tire forces could be linearized and can be written as a linear function of tire slip angle, which is further computed in terms of steering geometry and longitudinal velocity. Finally, the state space model used as control oriented plant for MPC is as follows

$$
\frac{d}{dt}\begin{Bmatrix} Y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 0 & 1 & V_x & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x-\frac{2C_{\alpha f}\ell_f-2C_{\alpha r}\ell_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2\ell_f C_{\alpha f}-2\ell_r C_{\alpha r}}{I_z V_x} & 0 & -\frac{2\ell_f^2 C_{\alpha f}+2\ell_r^2 C_{\alpha r}}{I_z V_x} \end{bmatrix} + \begin{Bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2\ell_f C_{\alpha f}}{I_z} \end{Bmatrix}\delta
$$

The above state space is corrected by replacing first state from local y coordinate to global Y from [5], as the original state-space representation is not controllable.

II. Defining Plant Parameters

Plant parameters are defined by considering 1.6L Chery Tiggo 3 and the plant parameters are identified in the below table [6].

Figure 3 1.6L Chery Tiggo 3 to define plant parameters

Vehicle Parameter	Value
vehicle mass/kg	2325
vehicle yaw moment inertia $I / kg \cdot m^2$	4132
wheelbase/m	3.025
longitudinal position of front wheel from vehicle centre of gravity a/m	1.430
height of vehicle centre of gravity h_g /m	0.5
cornering stiffness of front tyre $C_f / N \cdot rad^{-1}$	80000
cornering stiffness of rear tyre $C_r / N \cdot rad^{-1}$	96000

Figure 4 Identifying plant parameters

Assuming the longitudinal velocity of the vehicle is 30 ms^{-1} (V_x = 30 ms^{-1}) and referring to figure 4 pertinent to the vehicle parameters, substituting these values in state space model, we have following state space vectors for LTI system $\dot{x}(t) = Ax(t) + Bu(t)$ and $y(t) = Cx(t) + Du(t)$ with $y(t)$ as global lateral position of vehicle (Y) and vehicle yaw angle (Ψ). This continuous plant was further discretized at 0.1 s sample time to generate discrete prediction model.

```
A =
         0    1.0000   30.0000         0
         0   -5.0466         0  -28.8897
         0         0         0    1.0000
         0    0.6247         0   -6.5798

B =
         0
   68.8172
         0
   55.3727

C =
     1     0     0     0
     0     0     1     0

D =
     0
     0
```

Figure 5 State space matrices for the assumed plant

Generating Reference Trajectory

In order to generate MPC control scheme, the references for all the outputs, $\underline{y}$ = [Y, ψ], need to be generated. For this purpose, trajectory tracking is used in which time and space references are used as opposed to the path following where only space references are used. This would automatically incorporate effect of longitudinal velocity of vehicle in the waypoints generation. The X and Y references represent the vehicle position in the global frame. The goal of the control is to reach a specific position (X, Y) at a specific time, depending on the velocity of the vehicle. After generating required path, it could be used to generate reference for yaw angle.

If the vehicle is supposed to follow an existing road on the map we can receive the coordinates for the road by looking at the map. Nevertheless, for this thesis we also studied the scenarios when the vehicle is making a lane change from right to left. We assume that the vehicle is traversing 4 meters in the left at the speed of 30 ms^{-1} while it covers longitudinal distance of 120 meters on two lane 8 meter wide road. To generate it we have used quadratic Bezier curves. These curves are constructed by three control points, P_0, P_1, and P_2 according to [7]. Parametric equation for Bezier curves is:

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

where t ϵ [0, 1]

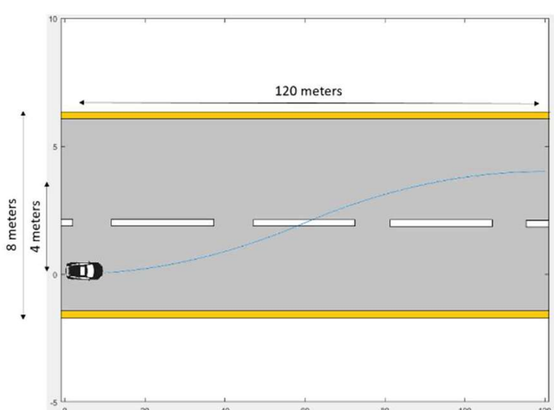

Figure 6 Waypoints for lane changing scenario

A useful property of the Bezier Curve is that it always passes through the point defined as the start point, P_0 and the point defined as the end point, P_2. [8] This property gives a better hold on generating the waypoints and is more intuitive than other methods.

Implicit MPC Control Strategy

In order to generate E-MPC, first I-MPC was generated to study the performance of control scheme for lane changing scenario under imposed constraints. This controller was tuned until the satisfactory control action was achieved. The controller parameter such as weighting matrices, prediction horizon and control horizon were further used to design E-MPC through mp-QP.

As constraints in formulating optimization problem. It was assumed that the steering could be turned for maximum of 35 degrees (0.6109 rad) in both the directions and rate of steering was 10 degrees per second (0.1745 rad/s). To restrict vehicle moving out of the road while changing lane, the maximum global position to which vehicle could move in lateral direction is taken as 5 meters (edge of road is 6 meters from initial position, shown with yellow solid line in Figure 6). Also, for a comfortable ride, the yaw angle was restricted to less than ±12 degrees. Below table summarises the constraints imposed in optimization problem. It should be noted that the constraints on output variables (Y and ψ) is hard constraint and that on input variable (u) is soft constraint in formulating optimization problem.

Variable	Min	Max
Steering angle, u (deg)	-35	+35
Steering rate, $\dot{u}$ (deg/s)	-10	+10
Global Lateral Position, Y (m)	0	5
Yaw, ψ (deg)	-12	+12

Figure 7 Constraints in designing MPC controller

A satisfactory performance of I-MPC was achieved for the prediction horizon of 3 and control horizon of 1. The rate change of manipulated variable was weighted 1.

For weighting of output variables, $\underline{y}$ = [Y, ψ], the inverse of RMS value from the reference set of each variable was used initially to normalize the weights. This normalized weights were further fine tuned. Final weight assigned to for global lateral position Y was 1.768 and that of yaw ψ was 14.184 as shown in Figure 8.

Output variable	Reference RMS value	Normalized weight	Final tuned weight
Global Lateral Position, Y	2.828	0.353	1.768
Yaw, ψ	0.07049	14.184	14.184

Figure 8 Weighting scheme for manipulated variables

Generating Explicit MPC

Once I-MPC was generated, the same plant model can be used to develop E-MPC. As discussed, E-MPC can be solved using mp-QP where each parameter can treated as a parameter whose value varies in the defined range. In our case, four state variables $\underline{x}$ = [Y, *dy*, ψ, dψ], two output variables $\underline{y}$ = [Y, ψ], one manipulated variable u were defined over a range to parametrize as discussed further.

To make the controller more general to use, it is assumed that the global Y can vary up to ±6 m for both left and right lane change and range of yaw is governed by limiting yaw rate as ±30 deg. For the assumed longitudinal speed of 30 ms^{-1}, the lateral velocity is calculated to be in range of ±10 ms^{-1} and yaw rate to be ±10 deg/s. As the manipulated variable u is soft constraint in optimization problem, its range is increased to ±50 deg although it is constrained to be ±35 deg in I-MPC. The output variables, Y and ψ, are ranged ±4.5 and ±30 deg respectively.

After parameterizing all the variables by defining a suitable range, the optimization problem was converted from QP to mp-QP. This problem was solved in MPT to generate polyhedral partitions. For the range defined above, 45 polyhedral partitions were generated. Further, in order to reduce the memory consumption by the controller, the E-MPC was simplified by retaining the polyhedral partitions whose Chebyshev radius was more that 0.04 [9]. After simplification, the number of polyhedral regions were reduced to 19. Figure 9 shows the 2D polyhedral partitions in subspace of ψ v/s Y.

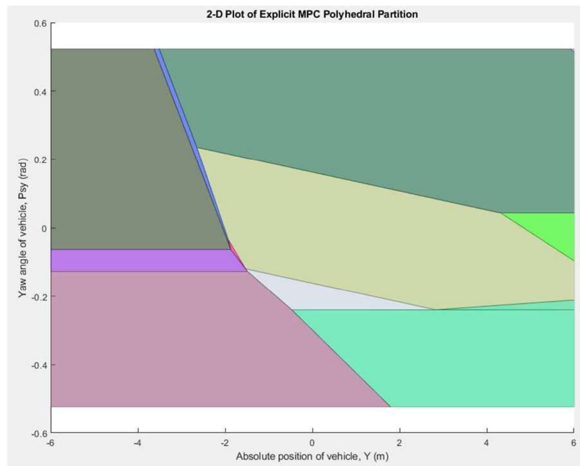

Figure 9 2D plot of E-MPC polyhedral partitions

Simulation Results

In this section, simulation results for both, implicit and explicit controllers, are shown and are discussed.

Figure 10 shows the performance of I-MPC with tuning scheme from Figure 8 and performance of E-MPC is shown in Figure 11. It can be seen that both the controllers performs stable operation and follow the references to a satisfactorily. However, in case of E-MPC, there is an abrupt steering input near the end of tracking, shown with red circle.

2D polyhedral partitions in subspace of ψ v/s Y are also shown. In the case of left lane change, the plant operates only in one region (Figure 12) while if the vehicle wants to do right lane change under similar circumstances, the plant would pass through 3 partitions (Figure 13).

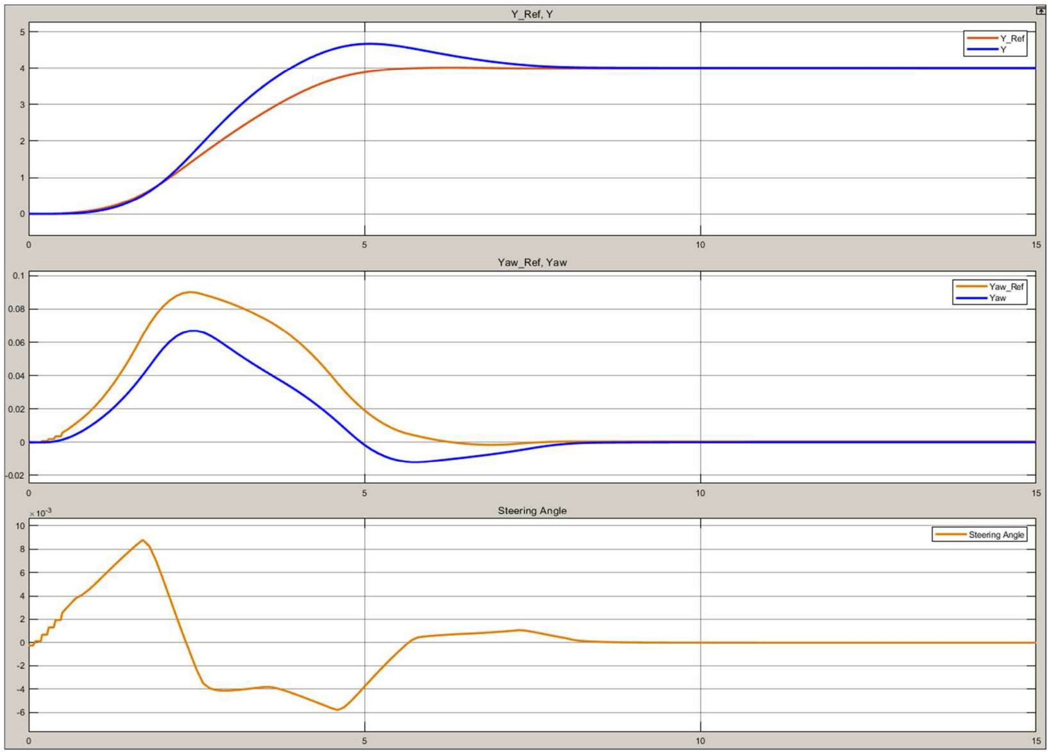

Figure 10 Performance of I-MPC

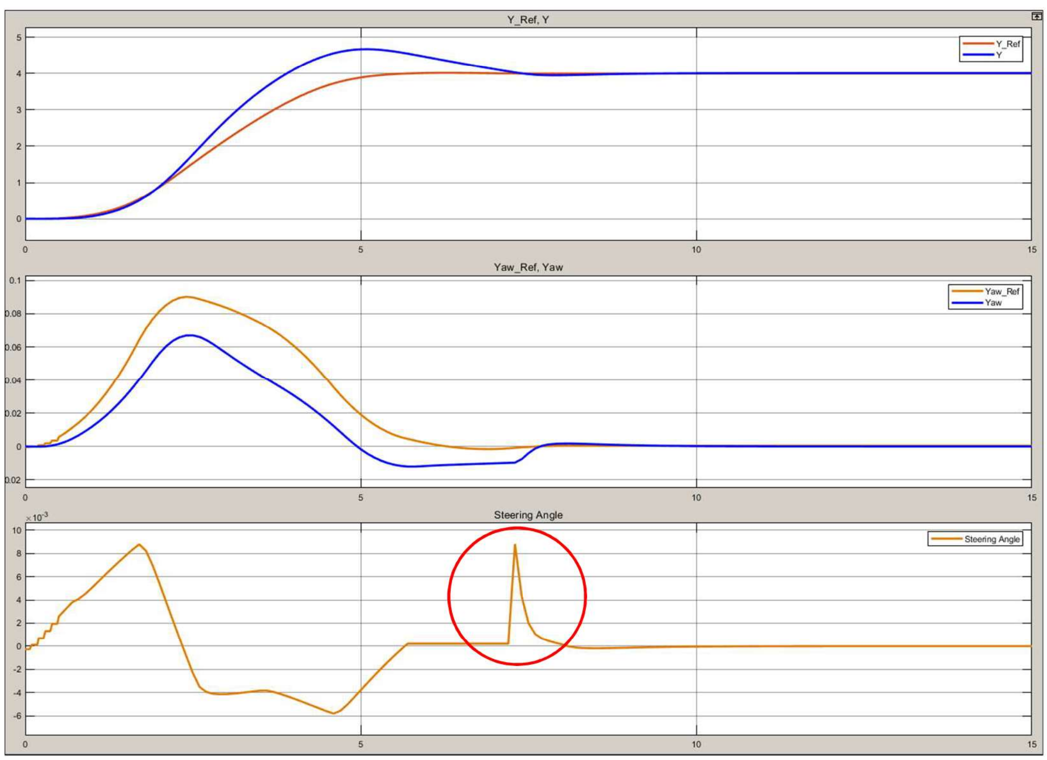

Figure 11 Performance of E-MPC

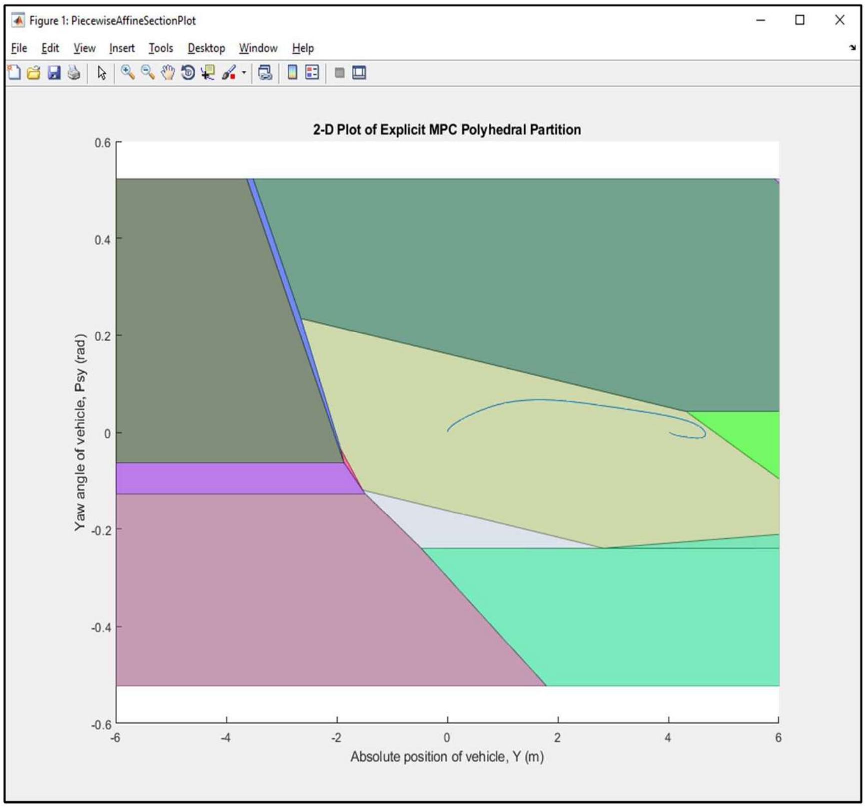

Figure 12 Visualizing control lad in left lane changing

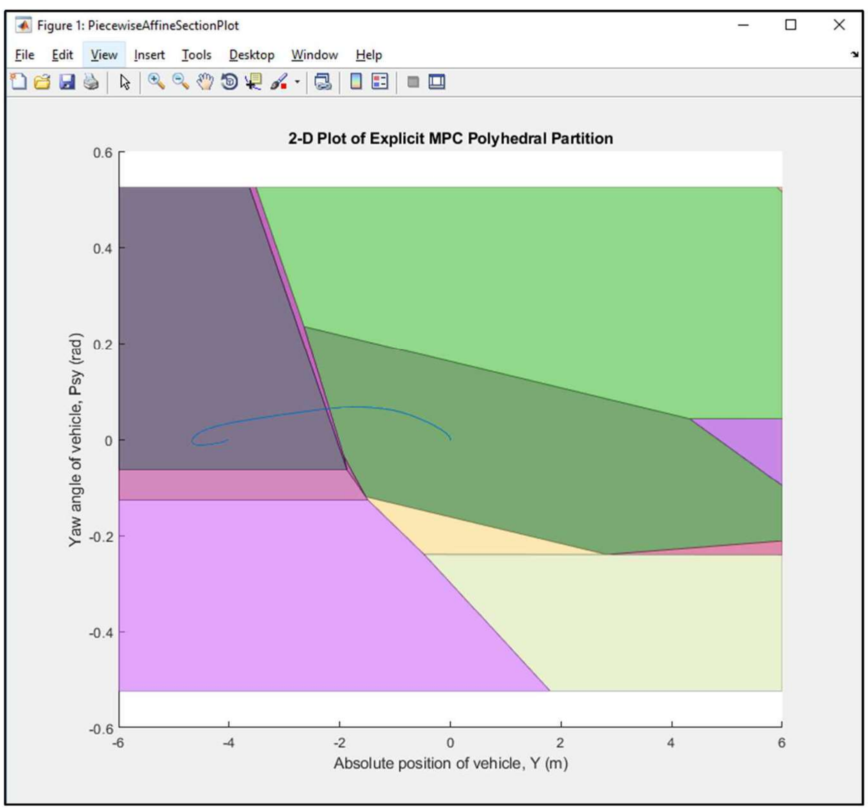

Figure 13 Visualizing control lad in right lane changing

Conclusion

In this paper, explicit model predictive control scheme was proposed for lateral control of ground vehicles in lane changing scenario. For this purpose, first implicit MPC control was developed, which was further used to generate explicit control scheme.

For tuning Implicit MPC, the controller was first designed for prediction horizon of 10, control horizon of 3 with arbitrary weightings. As prediction and control horizon was reduced, the controller became less sluggish. Finally, the prediction horizon was chosen to be 4 and control horizon to be 1. The weights were then changes until desired performance was achieved. Next, parametric range for each variable was define and QP was converted to mp-QP which was solved in MPT. In the end, 19 piecewise affine solutions were generated.

To evaluate the computational advantage and gauge increase in memory consumption of E-MPC, run time for both the controllers were recorded using SIMULINK profiler. It was observed that the computation speed was increased by around 58% for E-MPC.

Simulink Profile Report: Summary

Report generated 03-Apr-2019 11:48:56

Total recorded time:	1.17 s
Number of Block Methods:	71
Number of Internal Methods:	8
Number of Model Methods:	11
Number of Nonvirtual Subsystem Methods:	12
Clock precision:	0.00000003 s
Clock Speed:	3501 MHz

To write this data as empc1ProfileData in the base workspace click here

Figure 14 Runtime for I-MPC

Simulink Profile Report: Summary

Report generated 03-Apr-2019 11:50:15

Total recorded time:	2.81 s
Number of Block Methods:	147
Number of Internal Methods:	8
Number of Model Methods:	11
Number of Nonvirtual Subsystem Methods:	12
Clock precision:	0.00000003 s
Clock Speed:	3501 MHz

To write this data as mpc1ProfileData in the base workspace click here

Figure 15 Runtime for E-MPC

On the contrary, the downside of E-MPC is that it consumed approximately 21 times the memory of I-MPC.

```
>> whos mpc1 empc1
  Name        Size             Bytes  Class

  empc1       1x2             231574  explicitMPC
  mpc1        1x2              10958  mpc
```

Figure 16 Memory occupied by the controllers

Reference

[1] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. Automatica, 36(6):789 – 814 (2000)

[2] Zafiriou, E. Robust model predictive control of processes with hard constraints. Computers & Chemical Engineering, 14(4/5):359–371 (1990)

[3] Simone Garatti, Sergio Bittanti. Parameter estimation in the Pacejka's tyre model through the TS method. IFAC Proceedings: Volume 42, Issue 10, 1304-1309 (2009)

[4] Mia Isaksson Palmqvist. Model Predictive Control for Autonomous Driving of a Truck. Degree Project in Electrical Engineering, KTH Royal Institute of Technology, School of Electrical Engineering, Chapter 2, 7-19 (2016)

[5] R. Rajamani. Vehicle Dynamics and Control, Springer Science + Business Media Inc. Chapter 2, 30 (2005)

[6] Pan Zhao et al. Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID. International Journal of Advanced Robotic Systems, Vol. 9, 44:2012 (2012)

[7] Jiajia Chen, Pan Zhao, Tao Mei, and Huawei Liang. Lane change path planning based on piecewise bezier curve for autonomous vehicle. In Vehicular Electronics and Safety (ICVES), 2013

IEEE International Conference on, pages 17–22, (2013)

[8] http://web.mit.edu/hyperbook/Patrikalakis-Maekawa-Cho/node12.html

[9] https://link.springer.com/content/pdf/10.107%2F978-3-0348-5487-0_16.pd