Article

Autonomous Collision Avoidance using MPC with LQR-based Weight Transformation

Shayan Taherian , Kaushik Halder, Shilp Dixit and Saber Fallah

Abstract: Model predictive control (MPC) is a multi-objective control technique that can handle system constraints. However, the performance of an MPC controller highly relies on a proper prioritization weight for each objective, which highlights the need for a precise weight tuning technique. In this paper, we propose an analytical tuning technique by matching the MPC controller performance with the performance of a linear quadratic regulator (LQR) controller. The proposed methodology derives the transformation of LQR weighting matrix with a fixed weighting factor using discrete algebraic Riccati equation (DARE) and designs MPC controller using the idea of discrete time linear quadratic tracking problem (LQT) in the presence of constraints. The proposed methodology ensures the optimal performance between unconstrained MPC and LQR controller and provides a sub-optimal solution while the constraints are active during transient operations. The resulting MPC behaves as the discrete time LQR by selecting appropriate weighting matrix in the MPC control problem and ensures the asymptotic stability of the system. In this paper, the effectiveness of the proposed technique is investigated in the application of a novel vehicle collision avoidance system which is designed in the form of linear inequality constraints within MPC. The simulation results confirm the potency of the proposed MPC control technique in performing a safe, feasible and collision free path while respecting the inputs, states and collision avoidance constraints.

Keywords: Trajectory planning, MPC, LQR, LQT, inverse optimal control, collision avoidance.

1. Introduction

MPC has been recognised as one of the most powerful multi-objective optimal control technique for a wide range of applications with its ability to formulate the system constraints as a finite-horizon constrained optimisation problem. MPC performance is highly relies on the weights on each objective but the challenge is that there is no systematic procedure to select the weights to assure the best performance of MPC. Currently, suitable weight selection is a time consuming procedure and requires large number of trial-and-error selections and many computer simulations [1–3]. The complexity increases with increase in the number of control objective [4]. Hence, the necessity of developing a precise tuning procedure for MPC weights has arisen. This has been investigated by previous research works using different techniques [5][6]. For example, [6] presented a solution for selecting weights in order to match the MPC as a LQR based linear controller. In this work LMI based LQR approach was chosen by recasting the DARE in the LMI form to calculate the weighting matrices of unconstrained MPC. Although this approach was found the solution of MPC control parameters, the cost function was limited to have zero cross state-space terms, such that only few controllers could be matched. In [7], the matching controller i.e, the task of matching the MPC performance with the performance of a favorite controller, has been addressed by presenting a systematic method to tune the parameters of generalized unconstrained MPC for SISO systems [8]. In this approach the tuning parameters were calculated based on the desired closed-loop transfer function in order to relate the tuning parameters to the desired specification of the system. However, in this algorithm, higher control horizon

leads to a set of inequality constraints bilinear in the unknown elements, which cannot be solved using convex optimisation for linear MPC. Other tuning strategies have been investigated by researchers using intelligent learning algorithms [9][10]. In [11][12] the authors implemented neural networks and fuzzy-based decision making to tune the MPC parameters, without using the reference system model. [1] proposed an automated strategy for MPC-based motion cueing algorithm optimized using a multi-objective genetic algorithm. This strategy found suitable MPC weights aiming to demonstrate the superiority of the proposed tuning strategy over empirical tuning methods without the prior knowledge of the plant model. The general idea behind auto-tuning using learning and neural network algorithm is that the tuning parameters are updated along with the optimisation algorithm, thus the solution always set to the optimal value. Moreover, it is not required to have a great amount of knowledge about the system model to initialize the tuning procedure. However, it is needed to operate two optimisation procedures per step time (optimisation required by intelligent algorithms and other optimisation required by MPC controller) that may require overhead computational time.

In this paper, inverse optimal control problem [13–15] of discrete time linear quadratic regulator (DLQR) has been used for designing MPC controller to perform collision avoidance manoeuvre by selecting appropriate weighting matrices. In general, the DLQR weights (Q, R) are used to solve the discrete algebraic Riccati equation (DARE) for discrete time systems in order to find a solution (P) which is a real symmetric positive definite matrix. Then using P, an optimal state feedback control gain can be obtained from the DLQR control law. However, these weighting matrices obtained from DLQR can also be used to compute the controller of unconstrained MPC [6]. Therefore, using the aforementioned concept, the objective of this paper is to design the weighting matrix Q of the DLQR in such a way that Q can be used within the MPC framework, while assuring asymptotical stability of the system [16]. DLQR and unconstrained MPC can be related through DARE but using the weighting matrix obtained from the inverse optimal control problem of DLQR in MPC design, cannot ensure reference tracking problem. To overcome this problem, we recast DLQR into discrete time linear quadratic tracking (DLQT) problem [17] to design the weighting matrix Q which can be used for MPC design to ensure reference tracking problem. In order to do this, the dynamics of the vehicle and Ackerman pole-placement criteria are considered in the formulation of Q selection through DLQT, which will be used in MPC design for trajectory planning task. In order to highlight practical use of this analytical tuning technique, the proposed technique is implemented to generate safe, admissible and collision free trajectories for autonomous vehicle collision avoidance systems. In this work, collision avoidance system is formulated as a finite horizon optimisation problem with vehicle dynamics while collision avoidance constraints are formulated within MPC algorithm. However, collision avoidance constraints are generally non-convex in the context of trajectory planning which limits the feasibility and uniqueness of the solution of the optimisation problem. To tackle this problem, researchers provide solutions such as convexification [18], changing the reference frame [19], [20] and affine linear collision avoidance constraints [21]. In this paper the collision avoidance constraints are formulated in the form of linear affine constraints to perform trajectory planning manoeuvre on a curve path. Moreover, in the formulation of the collision avoidance constraints, the longitudinal position of the vehicle is excluded, allowing the designers to reduce the state dimension of the system. This is advantageous as removing a state from the system model helps in lowering the computational overhead for solving the optimisation problem in MPC. The effectiveness of entire framework for autonomous collision avoidance system on a curved road is validated in a co-simulation platform where high-fidelity vehicle model is simulated in IPG-CarMaker while the proposed MPC design is implemented in MATLAB/Simulink.

The contribution of this paper are summarised as: *(i)* Design of an analytical tuning technique to perform a collision free trajectory for autonomous vehicle, and *(ii)* Development

of mathematical framework to design convex constraints in order to perform collision avoidance manoeuvre on a curved path.

2. Discrete time optimal controller design

In this section, the tuning strategy for designing the MPC controller with the appropriate selection of weighting matrix Q is described. To design the tuning strategy, a discrete time inverse optimal control problem is formulated using linear time invariant (LTI) system:

$$\bar{x}(k+1) = \bar{A}_d\bar{x}(k) + \bar{B}_d u(k) \tag{1}$$

The objective of this strategy is to design the Q matrix from discrete time algebraic Ricatti equation which will ensure almost equal optimality between DLQR and unconstrained MPC control problem. It is noted that, in this design the weighting factor R is chosen as a fixed value as in [16]. Moreover, the weighting matrix Q obtained from inverse DLQR problem should be designed in such a way that the closed loop poles are placed in the desired location within the unit circle by satisfying user defined specifications using Ackerman's pole placement formula. It is noted that this design, is a procedure of finding MPC weighting matrix Q by obtaining the sub-optimal solution from DLQR controller in the presence of the constraints.
In the following subsections a general discrete infinite linear quadratic regulator problem is briefly overviewed and then a recast of DLQR as a discrete linear quadratic tracking (DLQT) problem is described. Next with the concept of inverse DLQT control problem, a strategy for designing an analytical approach to select Q matrix for the MPC control design will be discussed.

2.0.1. Discrete time linear quadratic regulator (DLQR)

To design the discrete time optimal controller the standard infinite horizon quadratic cost needs to be minimized corresponding to the discrete time system model (1):

$$\mathcal{J} = \sum_{k=0}^{\infty} \bar{x}(k)Q^T\bar{x}(k) + u(k)R^T u(k) \tag{2}$$

where $Q = Q^T \geq 0$ and $R = R^T > 0$. The solution of the DARE can be obtained to minimize the performance index (2) and is given by:

$$P = Q + \bar{A}_d^T P\bar{A}_d - \bar{A}_d^T P\bar{B}_d(R + \bar{B}_d^T P\bar{B}_d)^{-1}\bar{B}_d^T P\bar{A}_d \tag{3}$$

The discrete time optimal controller gain $\bar{K}$ can be achieved using matrix $P = P^T = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} > 0$ calculated from (3) to minimizes the cost function (2). The controller gain $\bar{K}$ can be written as:

$$\bar{K} = (R + \bar{B}_d^T P\bar{B}_d)^{-1}\bar{B}_d^T P\bar{A}_d \tag{4}$$

Which gives the control law:

$$u(k) = -\bar{K}\bar{x}(k) \tag{5}$$

2.0.2. Discrete time linear quadratic tracking (DLQT)

In this paper, an augmented system is presented based on the system dynamic (1) and the reference trajectory. The objective for the DLQT problem is to find the optimal control law in such a way that the designed controller can ensure to track the reference

trajectory for the linear system (1) within the predefined finite horizon performance index. The augmented system becomes:

$$X_{k+1} = \begin{bmatrix} \bar{x}_{k+1} \\ r_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{A}_d & 0 \\ 0 & F \end{bmatrix}}_{\hat{A}} \underbrace{\begin{bmatrix} \bar{x}_k \\ r_k \end{bmatrix}}_{X_k} + \underbrace{\begin{bmatrix} \bar{B}_d \\ 0 \end{bmatrix}}_{\hat{B}} u(k) \tag{6}$$

where F is assumed to be Hurwitz. This strategy is adopted from [17], and for the brevity of the paper, an interested reader can refer to this technical paper for calculation of DLQT algebraic Riccati equation:

$$Q_1 - P_1 + \hat{A}^T P_1 \hat{A} - \hat{A}^T P_1 \hat{B}(R + \hat{B}^T P_1 \hat{B})^{-1} \hat{B}^T P_1 \hat{A} = 0 \tag{7}$$

Where Q_1 is the weighting matrix for the augmented system (1):

$$Q_1 = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \tag{8}$$

And P_1 is the solution of the algebraic riccati equation:

$$P_1 = P_1^T > 0 \tag{9}$$

Then the control law can be written as:

$$u(k) = -(R + \hat{B}^T P_1 \hat{B})^{-1} \hat{B}^T P_1 \hat{A} X(k) \tag{10}$$

Where the value of the controller gain is:

$$\hat{K} = (R + \hat{B}^T P_1 \hat{B})^{-1} \hat{B}^T P_1 \hat{A} \tag{11}$$

For the reference tracking problem using MPC for the system (6), the Ricatti equation (7) is hold. The Q matrix obtained from DLQR problem can be used for DLQT control problem as shown in (8). Therefore, Q can be obtained by solving the inverse DLQR optimal control problem to use it in the reference tracking problem from DLQT approach. Hence, we can use the Q matrix obtained from inverse DLQR approach in the MPC control problem, since MPC reference tracking problem is equivalent to DLQT control problem. The following section describes the system model that has been used for the MPC tuning strategy.

3. System Model

In this paper, a kinematic vehicle model is used to capture the vehicle model in MPC [22]. This model assumes no slip between tyre and the road, and is found to be suitable model for trajectory planning for the application of collision avoidance manoeuvre [23][24]. Under the assumption of small angles approximation [25] the vehicle bicycle model is:

$$\dot{\xi} = v_x \tag{12a}$$

$$\dot{y} = v_x \psi + \frac{l_r}{l_f + l_r} v_x \delta_f \tag{12b}$$

$$\dot{v}_x = a_x \tag{12c}$$

$$\dot{\psi} = \frac{v_x \delta_f}{l_f + l_r} \tag{12d}$$

Where ξ and y are the longitudinal and lateral displacement of the vehicle in inertial coordinates, ψ is the inertial heading angle of the vehicle, v_x is the longitudinal velocity of the vehicle, l_f is the distance of the front axle from centre of gravity (C.G), and l_r

is the distance of the rear axle from C.G. The control inputs are steering angle δ_f and longitudinal acceleration a_x. For a given nominal velocity $v_{x,nom}$, the system in (12) can be expressed using a LTI system using the compact notation given below:

$$\dot{x} = Ax + Bu \tag{13}$$

where $x \triangleq [\xi, y, v_x, \psi]^T \in \mathcal{X} \subseteq \mathbb{R}^4$ is the state vector and $u \triangleq [\delta_f, a_x]^T \in \mathcal{U} \subseteq \mathbb{R}^2$ is the input vector with $\mathcal{X}$ and $\mathcal{U}$ being polyhedron regions as state and input constraints respectively. The structure of the system matrix A and input matrix B are as follows:

$$\begin{bmatrix} \dot{\xi} \\ \dot{y} \\ \dot{v}_x \\ \dot{\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & v_{x,\text{nom}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \begin{bmatrix} \xi \\ y \\ v_x \\ \psi \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{v_{x,\text{nom}} l_r}{(l_r+l_f)} & 0 \\ 0 & 1 \\ \frac{v_{x,\text{nom}}}{(l_r+l_f)} & 0 \end{bmatrix}}_{B} \begin{bmatrix} \delta \\ a_x \end{bmatrix} \tag{14}$$

Where $v_{x,nom}$ is the nominal longitudinal velocity of the vehicle, which is chosen the same as the initial speed of the vehicle [3]. Then system (13) is descretised with a sampling time t_s to obtain linear time invariant discrete system shown below:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{15}$$

where $A_d = e^{At_s} = \begin{bmatrix} 1 & 0 & t_s & 0 \\ 0 & 1 & 0 & v_{x,\text{nom}} t_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $B_d = Bt_s = \begin{bmatrix} 0 & 0 \\ \frac{v_{x,\text{nom}} l_r t_s}{L} & 0 \\ 0 & t_s \\ \frac{v_{x,\text{nom}} t_s}{L} & 0 \end{bmatrix}$

As the dynamics of the state ξ in system (14) depends only on v, it is possible to further simplify the system for trajectory generation:

$$\bar{x}(k+1) = \bar{A}_d \bar{x}(k) + \bar{B}_d u(k) \tag{16}$$

where $\bar{x} = [y, v, \psi]^T$ is the system states, $u = [\delta_f, a_x]$ is the system inputs, and matrices $\bar{A}_d$ and $\bar{B}_d$ are obtained by extracting the appropriate rows and columns of A_d and B_d in (14).

4. Selection of weighting matrix Q for MPC design

In order to obtain the appropriate Q matrix, for designing MPC based reference tracking controller, we have chosen $Q = diag(Q_{11}, Q_{22}, Q_{33})$ for the system (16) and optimal control gain $\bar{K}$ with a fixed weighting factor R. For the reference tracking problem, MPC can be recast as DLQT problem as described in 2.0.2. Therefore, it can be inferred that the weighting matrix Q obtained from inverse DLQT problem will provide almost the same controller gain for MPC which is described in the following Theorem.

Theorem 1. *MPC will ensure the reference tracking for the system (16), guaranteeing the sub-optimality of DLQT control problem by satisfying (2)-(5) with $Q_1^T = Q_1 \geq 0$, $R^T = R > 0$ and $P_1^T = P_1 > 0$ where $Q_1 = diag(Q, Q)$, $R_1 = diag(R, R)$ and the controller gains $\hat{K} = diag(\bar{K}, \bar{K})$ with $Q = diag(Q_{11}, Q_{22}, Q_{33})$, and $P_1 = diag(P, P)$ where P will be as (3) and $\bar{K} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \end{bmatrix}$ respectively.*

Proof. The equation (3) can be recast as DLQR problem:

$$Q = P - \bar{A}_d^T P \bar{A}_d + \bar{A}_d^T P \bar{B}_d (R + \bar{B}_d^T P \bar{B}_d)^{-1} \bar{B}_d^T P \bar{A}_d \tag{17}$$

where $P = P^T = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} > 0$, $R = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} > 0$ and $Q = Q^T = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \geq 0$. The corresponding control law (4) can be written as:

$$\bar{K} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \end{bmatrix} = (R + \bar{B}_d^T P \bar{B}_d)^{-1} \bar{B}_d^T P \bar{A}_d \tag{18}$$

Since the input $\bar{B}_d$ matrix in (16) includes only three non-zero elements, the three components of controller gains (k_2, k_4, k_6) will be zero, and the rest of the gains will be non-zero elements. The following section describes the system model required for designing the weighting matrix within MPC controller. □

Now using (16), (18) and the matrices (P,Q,R) in (17) yields the diagonal and off-diagonal elements of Q matrix as shown in (22)-(24) and

$$Q_{12} = -k_2 \Big[P_{11} \frac{v l_r t_s}{L} + P_{13} \frac{v t_s}{L} \Big] - P_{12} t_s k_5 \tag{19}$$

$$Q_{13} = v t_s P_{11} - \Big[k_3 \Big[P_{11} \frac{v l_r t_s}{L} + P_{13} \frac{v t_s}{L} \Big] + P_{12} t_s k_6 \Big] \tag{20}$$

$$\begin{aligned} Q_{32} = v t_s P_{12} - \Big[[\frac{v l_r}{L} t_s (v t_s P_{11} + P_{13}) \\ + \frac{v}{L} t_s (v t_s P_{13} + P_{33})] k_2 + (v t_s P_{12} + P_{23}) k_5 \Big] \end{aligned} \tag{21}$$

respectively. Since matrix Q is diagonal, all the off-diagonal elements are equal to zero i.e. $Q_{12} = Q_{13} = Q_{32} = 0$. Therefore, using these conditions and gains $k_2 = k_4 = k_6 = 0$ the elements of P matrix can be obtained i.e. from (19) yields (29). Using (29) in (20) yields (26), and using (29) in (21) results in (30). Now using the elements (29), (30), (26) and R, B_d matrices in (18), equations (31)-(33) will hold. And finally by solving (31)-(33) equations (25),(27) and (28) can be obtained. It is noted that, the gain matrix $\bar{K}$ needs to be specified such that the closed-loop poles of the system (15) would be inside the unit circle. In this paper, Ackerman's pole placement formula has been used to obtain the gains ($\bar{K}$) by specifying the closed-loop pole location inside the unit circle. This method solves the solution of Riccati equation to generate optimal weights using LQR controller. The generated gains then used in MPC controller which inherit all the properties and optimal solutions from LQR controller when MPC constraints are not active. However, when the constriants are active, the solution of Riccati equation in LQR provides a sub-optimal solution for MPC as it satisfies the constraints of Quadratic Programming (QP) problem indirectly. In other word, the set of states $\bar{x}(k)$ where the matching occurs

is the polyhedron, where the unconstrained optimizer (i.e unconstrained MPC, therefore DLQR controller) satisfies the constraints of the QP problem [6].

$$Q_{11} = (P_{11}\frac{vl_r}{l_r+l_f}t_s + P_{13}\frac{v}{l_r+l_f}t_s)k_1 \tag{22}$$

$$Q_{22} = P_{22}t_s k_5 \tag{23}$$

$$Q_{33} = -vt_s P_{13} + [\frac{vl_r}{l_r+l_f}t_s(vt_s P_{11} + P_{13}) + \frac{v}{l_r+l_f}t_s(vt_s P_{13} + P_{33})] \tag{24}$$

$$P_{13} = \frac{-\frac{R_1 k_1 vt_s}{L}[\frac{k_3 vt_s}{L} + (1-\frac{vt_s}{L})]}{\frac{v^4 t_s^4 k_1}{L^3}[\frac{vl_r t_s k_3}{L} - l_r - vt_s](1-\frac{l_r k_3}{L}) - \frac{v^3 t_s^3}{L^2}[(1-\frac{vl_r t_s k_1}{L})(1-\frac{vt_s}{L})(1-\frac{l_r k_3}{L})] + \frac{v^2 l_r t_s^2}{L}(\frac{l_r k_3}{L}-1)\frac{v^3 t_s^3 k_1 k_3}{L^3}} \tag{25}$$

$$P_{11} = -\frac{P_{13}\frac{v}{L}t_s k_3}{\frac{vl_r}{L}t_s k_3 - vt_s} \tag{26}$$

$$P_{33} = \frac{P_{11}[\frac{vt_s l_r}{L} - \frac{k_1 v^2 l_r^2 t_s^2}{L^2}] - k_1 R_1 + P_{13}[\frac{vt_s}{L} - \frac{v^2 l_r t_s^2 k_1}{L^2}]}{\frac{k_1 v^2 ts^2}{L^2}} \tag{27}$$

$$P_{22} = \frac{R_2 k_5}{ts - ts^2 k_5} \tag{28}$$

$$P_{12} = 0 \tag{29}$$

$$P_{23} = 0 \tag{30}$$

$$k_1 = \frac{\frac{P_{11} vl_r t_s}{L} + \frac{P_{13} vt_s}{L}}{R_1 + \frac{v^2 l_r^2 t_s^2 P_{11}}{L^2} + \frac{[\frac{P_{13} vl_r t_s}{L} + \frac{P_{33} vt_s}{L}]vt_s}{L}} \tag{31}$$

$$k_3 = \frac{\frac{P_{11} v^2 l_r t_s}{L} + P_{13}\frac{[vt_s P_{13} + P_{33}]vt_s}{L}}{R_1 + \frac{P_{11} v^2 t_s^2}{L^2} + \frac{[\frac{P_{13} vt_s}{L} + \frac{P_3 3 vt_s}{L}]vt_s}{L}} \tag{32}$$

$$k_5 = \frac{P_{22} t_s}{R_2 + P_{22} t_s^2} \tag{33}$$

$$k_2 = k_4 = k_6 = 0 \tag{34}$$

5. Constraint design

In this section, the mathematical framework for computing the collision avoidance constraints which are used to plan trajectories on curved roads is presented. In the absence of traffic vehicles, a convex region is designed to ensure that the planned vehicle trajectories remain within the boundaries of the road. On the other hand, in the presence of obstacles, a convex region is designed to ensure that the planned trajectories remain within the road boundaries and maintains a safe distance from the obstacle. The detailed explanation of the aforementioned constraints designs are presenting in the following subsections.

5.1. Road Boundary Constraints

Using the edges of a curved road as constraints within the MPC framework results in non-convex constraints which is not suitable for Quadratic Programming (QP) framework [26]. Consequently, boundary constraints are designed by assuming that the road segment in the immediate vicinity of the subject vehicle is part of a straight road (1). The edges of the virtual straight road are obtained by calculating the points P_1 and P_2 in 1. These virtual straight lines can be constructed using the generated points P_1 and P_2, following with the slope equal to the heading angle of the upper and lower road boundaries. Moreover, the equation of these lines can be formulated in the form of linear inequality constraints which can ensure that the planned trajectories lie within the edges of this virtual road. The point P_1 is located at the intersection of the left (outer) road edge with an imaginary line passing through the subject vehicle's centre of gravity (C.G)

and perpendicular to the subject vehicle's longitudinal axis. The equations of outer lane boundary and intersected line is:

$$y_{outer} = f(x_{outer}) \tag{35a}$$
$$a_{outer}\xi + b_{outer}y + c_{outer} = 0 \tag{35b}$$

where (35a) is the equation of the outer lane boundary, and (35b) is the equation of the intersected line perpendicular to the outer lane boundary, and the parameters a_{outer}, b_{outer} and c_{outer} in equation (35b) are the coefficients of the intersected line. The line passing through P_1 with a slope equal to the reference path heading angle ψ_{ref} forms the outer edge (left line) of the virtual straight road (blue line in 1) with equation of:

$$m_{outer}\xi - y + P_{1y} - m_{outer}P_{1x} < 0 \tag{36}$$

where m_{outer} is the slope of the equation of the outer edge road and (P_{1x}, P_{1y}) are coordinates position of point P_1. From equations (12a) it is evident that the evolution of ξ depends on longitudinal velocity of the vehicle. Therefore, the predicted longitudinal position $\bar{\xi}$ can be estimated using the initial position ξ_0 and predicted velocity for the entire prediction horizon N_p using the equation bellow

$$\bar{\xi}(j) = \left[\xi_0 + \sum_{i=1}^{j}(\bar{v}(i)\cdot t_s)\right]; \quad j = 1, 2, ..., N_p \tag{37}$$

ξ_0 is the current longitudinal position of the subject vehicle and $\bar{v}$ is the predicted nominal velocity. The equation above is substituted into the equation (36) to formulate the generalized constraint equation for N_p different constraint equations given below:

$$m_{outer}\left[\xi_0 + \sum_{i=1}^{j}(\bar{v}(i)\cdot t_s)\right] - y(j) - m_{outer}P_{1x} + P_{1y} < 0 \tag{38}$$

where the predicted nominal velocity and predicted lateral position can be obtained from the prediction model in MPC. For different value of j in (37), equation (38) represents the outer edge of the virtual straight road constraint which is solely depends on two states of the vehicle (longitudinal velocity and lateral position of the vheicle). Similarly, the line passing through P_2 with a slope ψ_{ref} forms the right (inner) edge of the virtual straight road. The inner edge of the virtual straight road can be expressed as:

$$y_{inner} = f(x_{inner}) \tag{39a}$$
$$a_{inner}\xi + b_{inner}y + c_{inner} = 0 \tag{39b}$$

where (39a) is the equation of the inner lane boundary and (35b) is the intersected line perpendicular to the inner lane boundary. Consequently in the same way as the upper bound, the lower hyperplane can be defined as:

$$m_{inner}\left[\xi_0 + \sum_{i=1}^{j}(\bar{v}(i)\cdot t_s)\right] - y(j) - m_{inner}P_{4x} + P_{4y} > 0 \tag{40}$$

To decouple the orientation of the virtual straight road with the subject vehicle's orientation, the edges are parameterised using ψ_{ref}. It is noted that, this technique is suitable to be implemented for different range of curvature of the road. Moreover, it can be ensured that the optimization problem in MPC will always be feasible, and guarantee the convexity on the curve road.

5.2. Collision Avoidance Constraints

As mentioned above the collision avoidance constraints are designed to ensure the subject vehicle evades the obstacle vehicle without any collisions. In literature it is common to design ellipsoids/rectangles around the obstacle to obtain the collision avoidance constraints [27,28]. However, these techniques result in non-convex optimisation problem which limits the uniqueness and feasibility of the solution of the optimisation. In this paper, the collision avoidance constraints are generated using two lines entitled as forward collision avoidance constraints (FCC) and rear collision avoidance constraints (RCC) as shown in 2. This technique is inspired from [21], where the collision avoidance constraints are used for only straight driving conditions. The purpose of generating the FCC is to prevent front end collision while performing a collision free lane change and the purpose of the RCC is to prevent rear end collision while returning to the original lane. The activation of FCC is when the subject vehicle is behind the lead vehicle while RCC is activated when the subject vehicle crosses the lead vehicle. In this framework, three points (P_3, P_4, P_5) are required to generate collision avoidance constraints. These points can be calculated by intersecting virtual longitudinal and lateral axis of the lead vehicle (grey and yellow lines respectively) with a circle centred at the lead vehicle's geometrical center, which provides the location of the points P_3, P_4, and P_5. The following subsections explain the procedure for generating collision avoidance constraints.

5.2.1. Forward Collision Avoidance Constraints

The virtual line on the road segment representing FCC is the line that passes through the points (P_{3x}, P_{3y}) and (P_{4x}, P_{4y}) in 3. Points (P_{3x}, P_{3y}) are obtained as a results of the intersection of longitudinal axis of the lead vehicle (grey line in 2):

$$a_{Lon}x_L + b_{Lon}y_L + c_{Lon} = 0 \tag{41}$$

with a circle centred at the the lead vehicle geometrical centre and radius of $L_x + l_{rL}$:

$$(x - x_L)^2 + (y - y_L)^2 = (L_x + l_{rL})^2 \tag{42}$$

Where L_x is safety distance which can be defined as $L_x = v_x t + L_c$. The longitudinal velocity of the subject vehicle expressed as v_x, t is the desired time gap of subject vehicle when approaching lead vehicle and finally L_c is the lead vehicle length. Moreover, parameters $l_{rL}, a_{Lon}, b_{Lon}, c_{Lon}$ represent the distance of center of gravity of the vehicle to rear wheels and coefficients of the intersected line respectively. It is noted that at every time instant of MPC optimization, L_x updated according to the current value of the subject vehicle velocity (v_x). For generating FCC, another set of points is required (P_{4x}, P_{4y}). These points are generated by means of intersection of lateral axis of the lead vehicle (yellow line in 2) with a circle of radius W:

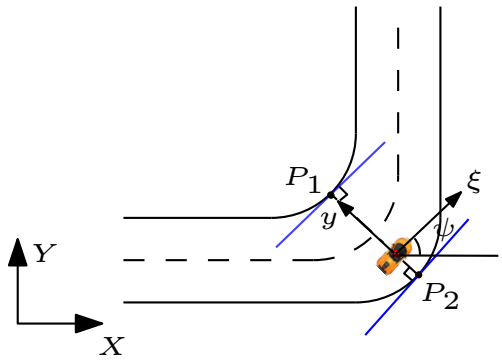

Figure 1. The hyperplanes (blue lines) indicate the constraints for optimization. The points on the borders are the projection of the center line of the subject vehicle on the borders

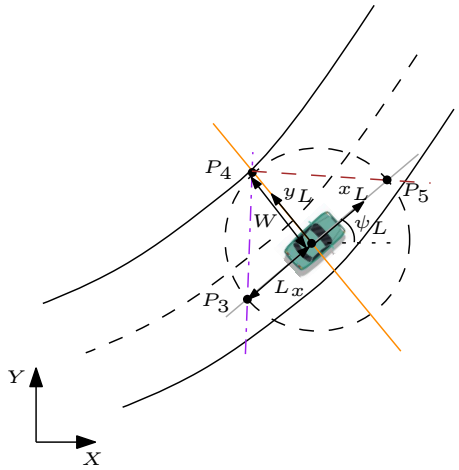

Figure 2. Schematic to construct forward and rear collision avoidance constraints line. **Note**: ψ_L is lead vehicle orientation, L_x and W indicate the safety distance from the lead vehicle

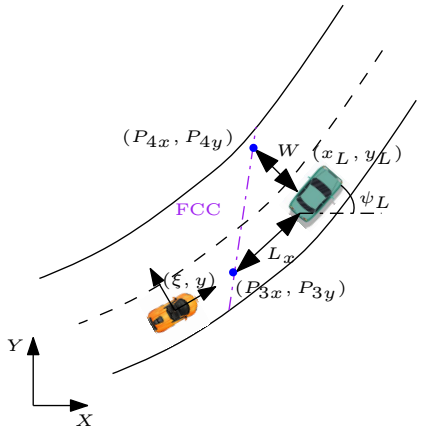

Figure 3. Schematic to calculate the FCC. **Note**: orange car-subject vehicle, green car-lead vehicle, red line RCC, L_x and W indicate the safety distance from the lead vehicle

$$(x - x_L)^2 + (y - y_L)^2 = W^2 \tag{43}$$

the parameter $W = \frac{1}{2}W_c + W_L$, where W_c is the width of the subject vehicle and W_L is the lane width plus the distant from the subject vehicle to the center line of the road [21]. After defining two set of points, it would be trivial to find the equation of the line for forward collision avoidance constraints, where the final equation will be:

$$\underbrace{m_{FCC} \cdot t_s}_{a_{FCC}} \left[\sum_{i=1}^{j} (\bar{v}(i)) \right] + \underbrace{(-1)}_{b_{FCC}} y + \underbrace{P_{1y} - m_{FCC} P_{1x} + m_{FCC} \cdot \xi_0}_{c_{FCC}} > 0 \tag{44}$$

Where m_{FCC} is the slope of the forward collision avoidance constraint. The generated FCC divides the (x, y) plane into two regions. Region 1: $a_{FCC}x + b_{FCC}y + c_{FCC} > 0$ which is the safe region, Region 2: $a_{FCC}x + b_{FCC}y + c_{FCC} < 0$ which represent the unsafe region. FCC forces the vehicle to be in a safe region while performing a lane change on a curve/straight road. Moreover, (44) represents a linear affine constraint approximation which can be formulated in QP format.

5.2.2. Rear Collision Avoidance Constraints

Similar to FCC, RCC can be generated by following similar steps. The procedure for calculating the intersection point P_3, is identical to FCC, thus resulting in generating the equation of the RCC line (4):

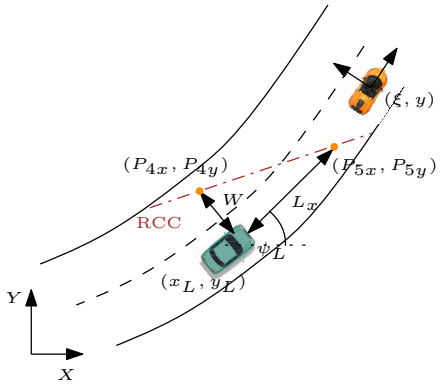

Figure 4. Schematic to calculate RCC. **Note**: orange car is subject vehicle, green car is lead vehicle, L_x and W indicate the safety distance from the lead vehicle

$$\underbrace{m_{RCC} \cdot t_s}_{a_{RCC}} \left[\sum_{i=1}^{j} (\bar{v}(i)) \right] + \underbrace{(-1)}_{b_{RCC}} y + \underbrace{P_{3y} - m_{RCC} P_{3x} + m_{RCC} \cdot \xi_0}_{c_{RCC}} < 0 \tag{45}$$

It is noteworthy that the aforementioned collision avoidance constraints can be adopted for the scenarios where (i) the subject vehicle needs to change the lane while performing the collision avoidance manoeuvre and (ii) where more traffic member on the road are present which require multiple hyperplanes.

Remarks

- One of the benefits of the proposed approach lies in the fact that there is only one parameter to be tuned (desired time gap) and the rest of the parameters are based on the geometry of the lead vehicle.
- The design of the collision avoidance constrains is based on basic mathematical operation, therefore the aforementioned constraints for each traffic vehicle can be generated without any major computation overhead, thus it is suitable for real time implementation.
- Reducing a state dimension in the system model helps in bringing down the computation requirement for solving the optimisation problem.

It is noted that, by using boundary constraints and the appropriate FCC/RCC based constraints, a convex region representing safe zones on a curved road is generated which enforce MPC to plan a safe and collision free trajectories on the road.

6. Trajectory planning controller

In general, autonomous highway driving involves motion planning and control of a subject vehicle in order to maintain the velocity and keeping the vehicle away from lane boundaries while avoiding possible collisions. The choice of manoeuvre to perform is the results of the decision making process with the objective of following the desired trajectory x_{ref}, y_{ref} (6.1) while respecting physical and design limitations of the subject vehicle. In this section, using the definition of the linear model in (13) and linear system constraints in 5, a constrained linear quadratic optimal control problem is formulated over a prediction horizon N_p.

6.1. Reference Trajectory Generator

The reference trajectory generator provides longitudinal position $x_{ref}(s)$ and lateral positions $y_{ref}(s)$ as well as orientation $(\psi_{ref}(s))$ of the road for the vehicle to follow. The

generated paths are parametrized by curvature as a function of the distance s along the path. The equation of curvature can be formulated as follows:

$$\kappa = \frac{\dot{x}_{ref}\ddot{y}_{ref} - \dot{y}_{ref}\ddot{x}_{ref}}{(\dot{x}_{ref}^2 + \dot{y}_{ref}^2)^{\frac{3}{2}}} \tag{46}$$

The reference trajectories are formulated using clothoids method which expressed by Fresnel integrals [29,30]. The trajectories are parametrized by curvature κ as a function of distance s along the path as follows:

$$x_{ref}(s) = \int_0^s \cos\psi_{ref}(x)dx \tag{47}$$

$$y_{ref}(s) = \int_0^s \sin\psi_{ref}(x)dx \tag{48}$$

where (47) and (48) are constructed using polynomial integration method of Fresnel function applied to clothoid [31], and ψ_{ref} is calculated as follows:

$$\psi_{ref}(s) = 2\tan^{-1}\left(\frac{y_{ref,n} - y_{ref,n-1}}{x_{ref,n} - x_{ref,n-1}}\right) \tag{49}$$

where $(x_{ref,n}), y_{ref,n}, n = 1...N$ are the sequence of N the sampled points on the trajectory [32].

6.2. Trajectory Planning: MPC

For the trajectory planning using linear MPC, the system states and inputs are subjected to the following state and input constraints:

$$\bar{x} \in \bar{\mathcal{X}}, u \in \mathcal{U} \tag{50}$$

where $\bar{\mathcal{X}} = \{\bar{x} \in \mathbb{R}^3 : \bar{x}_{min} \leq x \leq \bar{x}_{max}\} \subset \mathbb{R}^3$ and $\mathcal{U} = \{u \in \mathbb{R}^2 : u_{min} \leq u \leq u_{max}\} \subset \mathbb{R}^2$ are states and inputs polytope admissible regions (subscripts *min* and *max* are the minimum and maximum of the corresponding values). The following cost function is formulated:

$$\begin{aligned}\mathcal{J} = \sum_{k=0}^{N_p-1} [\parallel \bar{x}_{ref} - \bar{x}_{t+k|t} \parallel_Q^2 + \parallel u_{t+k|t} \parallel_R^2] \\ + \parallel \bar{x}_{ref} - \bar{x}_{t+N|t} \parallel_P^2\end{aligned} \tag{51}$$

where $\bar{x}_{t+k|t}$ is the predicted state trajectory at time $t+k$ obtained by applying the control sequence $U_t^* = [u_t^T, \ldots, u_{t+N-1}^T]^T$ to the system (15), starting from initial state of $\xi_{t|t}$. The parameter $N \in \mathbb{N}^+$is the prediction horizon while $Q \in \mathbb{R}^{3\times3}$, $P \in \mathbb{R}^{3\times3}$ and $R \in \mathbb{R}^{2\times2}$ are weighting matrices. The performance index in (51) consists of the stage cost, input cost and the terminal cost, respectively. The desired state x_{ref} representing the reference state for the subject vehicle and is defined as $\bar{x}_{ref} = [y_{ref}, v_{xref}, \psi_{ref}]^T$. It is noted that in this paper v_{xref} is taken as the initial value of the subject vehicle's velocity. The following constrained optimization problem, for each sampling time, is formulated as:

$$\min_{U_t} \mathcal{J}(u_t; \bar{x}(t), \bar{x}_{ref}) \tag{52a}$$

subject to

$$(15), (38), (40), (44), (45), (50) \tag{52b}$$

In this framework, at every time step, the problem (52a), under the constraints (52b) are calculated based on the current state $x(t)$, over a shifted time horizon. The optimal

Algorithm 1 Trajectory planning

initialize:
$v_{des} \leftarrow$ desired longitudinal velocity from user
procedure GENERATE TRAJECTORY
top:
    Use equation (3) to create matrix Q
    $\bar{K} \leftarrow$ Ackerman pole placement formula as (4) and calculate $\bar{K}$
    $P \leftarrow$ Obtained $\bar{K}$ in (25)-(34) and generate P
    $Q \leftarrow$ Obtained P and substitute (16) in (22)-(24) to calculate Q
loop:
    $\bar{x}_{ref} \leftarrow$generate target states as
    $\bar{x} \leftarrow$current state vector
    getRoadBoundaryConstraints as (35)-(40)
    getCollisionAvoidanceConstraints as (41)-(45)
    $u^* \leftarrow$ solve MPC with generated Q from *top* with fixed R
    $x^* \leftarrow$ Apply optimal input $u(k)$ as (12)
    if user request to change v_{des} **then**
        goto *initialize*.
        else
    goto *loop*.

control input is calculated as $u^*(\bar{x}(t)) = U_t^*(0)$, which is a solution to the problem (52a). As the sets $\bar{\mathcal{X}}$ and $\mathcal{U}$ are convex, then the MPC problem (52a) is solved as a standard QP optimisation problem. All the steps required to design the proposed MPC controller can be seen in 1, and the closed-loop form is illustrated in 5. The following demonstrates the numerical results to show the efficacy of the proposed control strategy.

7. Numerical results

In this section, a simulation environment is used to investigate the ability of the proposed trajectory planning architecture to perform collision avoidance manoeuvre. The scenarios used to perform this are as: 1) an obstacle avoidance when the subject vehicle (SV) is 300 m behind the stationary lead vehicle (LV), and travelling at 80 km h^{-1} on a two-lane one-way curve road with radius of 750 m, 2) collision avoidance manoeuvre with moving obstacle under different road surface friction conditions. It is noted that, the chosen curve road is approximately equal to the tightest corner allowed in highways [33]. The simulation results for the proposed closed-loop architecture are obtained using MATLAB IPG CarMaker co-simulation environment. Moreover, it is assumed that the dimension of the road, lane-limits and LVs states are available to the SV, and the initialisation parameters are tabulated in 1. Additionally, to evaluate the efficacy of the proposed algorithm, MPC is initialized with prediction horizon equal to 14 and low time gap ($t = 0.8$) when designing the constraints for imitating more extreme manoeuvre to perform collision avoidance scenario.
The discrete linear quadratic tracker (6) is used to obtain the proper weighting matrix Q in MPC to perform the collision avoidance manoeuvre. The value of the generated

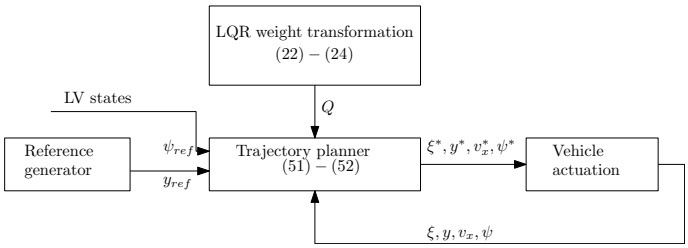

Figure 5. Closed-loop framework for collision avoidance **Note**: LV indicates lead vehicle

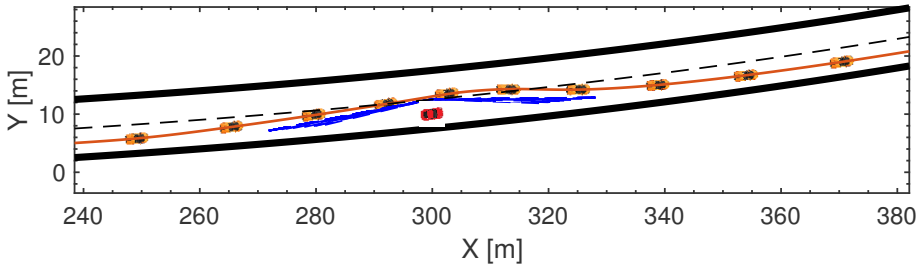

Figure 6. Simulation result: SV and LV (red vehicle) trajectories **Note:** FCC and RCC collision avoidance constraints represented as blue dash lines to provide a safe region between SV and LV.

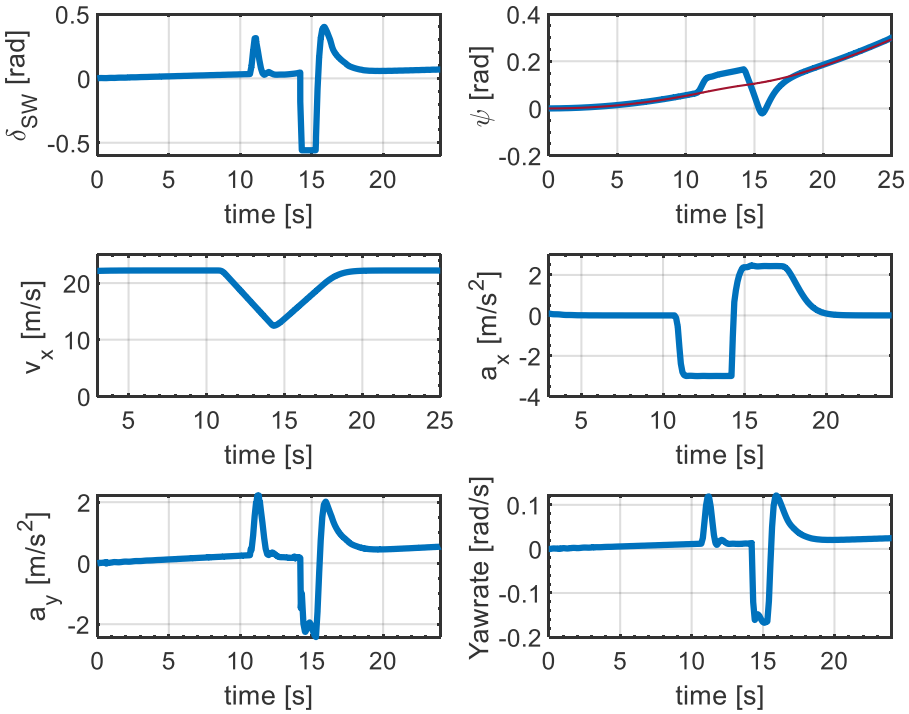

Figure 7. Simulation results: Steering wheel, heading angle (red line indicates the reference heading angle), longitudinal velocity, longitudinal acceleration, lateral acceleration, and yawrate for a collision avoidance maneuver

matrix Q corresponding to the proposed MPC design is $Q = diag(1.53, 0.023, 34.06)$. It is noted that the simulation results relating to the proposed MPC design is compared with a generic MPC controller where the values of the Q matrix have been generated through trial-and-error procedure (which will be referred as manual tuning for the rest of the paper) to demonstrate the superiority of the proposed control algorithm. It is noted that, the choice of manual tuning of Q_{manual} is based on trial-and-error and result of many simulations in order to select a set of suitable values to perform trajectory planning with $Q_{manual} = diag(5, 1e^{-1}, 230)$. Moreover, in both cases the value of the weighting matrix

R is equally chosen to $R = \text{diag}(10, 9 \times 10^{-2})$.

7.1. Collision avoidance with static obstacle

The result of the obstacle avoidance scenario with the assumption of high friction surface, using the proposed MPC design, is demonstrated in 6. This figure shows the actual trajectory of the vehicle where the SV starts to change the lane 30 m from the lead vehicle and converges to the original lane shortly after a collision free path is detected, while staying within the road boundaries. Moreover, it can be seen that the vehicle respects the collision avoidance constraints by maintaining the safety margins from the lead vehicle during the collision avoidance manoeuvre for all sub-manoeuvres. The result in 6 indicates that the proposed structure generates consistent trajectory with no overshoot while maintaining the safety margins to the lead vehicle during both lane changes.

For further analysis the performance of the proposed collision avoidance system, the evolution of the states and control inputs of the SV over time are shown in 7. A particular feature about the collision avoidance manoeuvre is that the vehicle starts the manoeuvre close to 11 s where the steering wheel angle for the lateral motion shows a smooth response without having high-frequency oscillation. However, steering action after returning to the original lane is more aggressive and the activation of the input constraints can be observed in this figure. This action suggests that for a successful collision avoidance and accurate reference tracking, large steering action is required on the reveres behavior where the SV attempts returning to the original lane. Another key point is the two states of the SV, *(i)* heading-angle and *(ii)* longitudinal velocity which demonstrate smooth evolution without having any oscillation during either of the lane changes while maintaining the SV's pose directionality. It can be seen that the velocity of the vehicle decreases rapidly while the first lane change manoeuvre is being performed. The reverse behavior (i.e increasing the velocity while perform the lane change) is visible right after evading the lead vehicle. This is reminiscent of a real-world collision avoidance manoeuvre where the vehicle decelerate when an obstacle is detected to prevent a possible collision, and accelerate after evading the obstacle, which shows the efficacy of the proposed control structure. Moreover, the longitudinal acceleration also shows a

Table 1: Design Parameters

Symbols	**Value**	**Value**
	Road Geometry	
N_{lanes}	2	-
w_{lane}	5	m
	Subject vehicle	
l_f	1.144	m
l_r	1.2060	m
	Lead vehicle	
$l_{Lv,l}$	4.1	m
$l_{Lv,w}$	1.7	m
	Controller Parameters	
t_s	0.1	s
N_p	14	-
closed-loop poles	$(5e^{-1}, 0.6, 0.95)$	-
	System	
$\mathcal{X}$	-[500, -11.1,1.5]$\leq x \leq$[500,27.7778,1.5]	-
$\mathcal{U}$	-[0.0698, 3]$\leq u \leq$[0.0698,3]	-

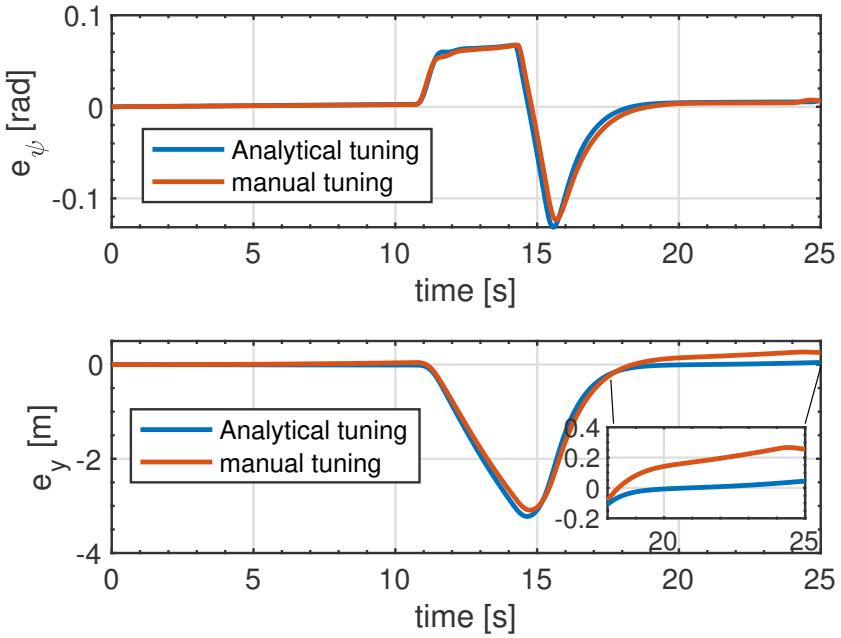

Figure 8. lateral error e_y and heading error e_ψ for analytical and manual tuning approach

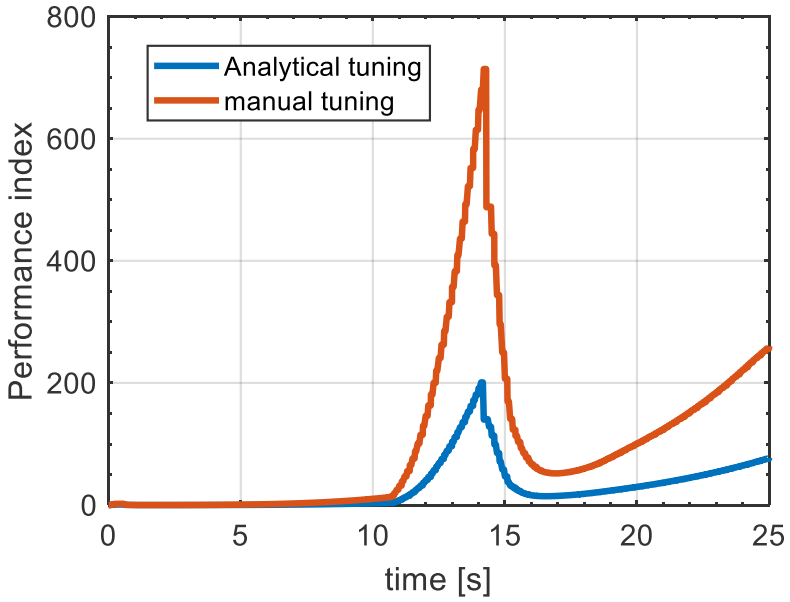

Figure 9. Performance index for both automatic tuning and manual tuning on curve driving condition

smooth profile without any oscillation during deceleration and acceleration for either of the lane changes while respecting the system constraints. Similarly the bottom subplot in 7 demonstrates the lateral acceleration and yaw-rate of the SV which shows smooth without having high oscillation profile for either of the lane changes. It is noted that, the absolute maximum value of lateral acceleration is 2.3 $\mathrm{m\,s^{-2}}$ which demonstrates that the vehicle does not reach to its limits of handling. This can be observed by investigating the yawrate profile, which the maximum absolute value reaches about 0.18 $\mathrm{rad\,s^{-1}}$, indicates that the vehicle operates in the linear region of the tire.

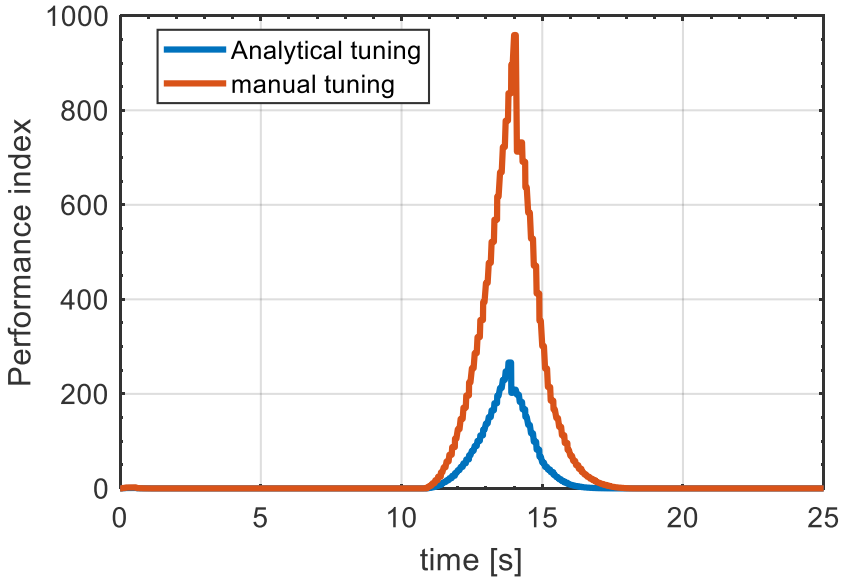

Figure 10. Performance index for both automatic tuning and manual tuning on straight driving condition

To show the need of the proposed MPC design to perform the collision avoidance manoeuvre, we compare the performance of the proposed framework with MPC design using manual tuning. 8 demonstrates the lateral and heading angle error for both analytical tuning and manual tuning of MPC. In this analysis lateral error e_y is defined as lateral distance of the centre of gravity of the vehicle (subject vehicle) from the desired path and e_ψ is defined as the difference between the actual orientation of the vehicle and desired heading angle which are defined as [34]:

$$e_y = (y_{CG} - y_{ref}) \cos \psi_{ref} - (x_{CG} - x_{ref}) \sin \psi_{ref} \tag{53a}$$

$$e_\psi = \psi - \psi_{ref} \tag{53b}$$

The top figure shows the heading angle error which shows the similar behaviour for both proposed MPC design and manual tuning. It can be seen that the directionality of that vehicle is maintained during the collision avoidance manoeuvre for both cases with minimum discrepancy with the maximum error of 0.15 rad. The bottom figure, represents the lateral displacement error for the collision avoidance manoeuvre. The main difference can be observed after performing the collision avoidance manoeuvre, where the error in analytical tuning strategy almost converges to zero, whereas the conventional tuning reaches to about 30 cm. This figure confirms that the proposed strategy is able to successfully maintaining the directionality of the vehicle during the collision avoidance manoeuvre while performing better reference tracking with minimum tracking error compare to conventional trial and error tuning.
For further illustration of the effectiveness of the proposed MPC design, we compare the performance index value of the proposed technique with MPC design using manual tuning to investigate the level of optimality between the two procedures. This can be seen in 9 where the performance index ($\mathcal{J}$) indicates the minimized value of the cost function in (51). It can be inferred that, the proposed MPC design gives an improvement in $\mathcal{J}$ of more than 70% over the manual tuning between 11 s and 15 s where the performance index is at its maximum, and for the rest of the manoeuvre still performance index for the analytical tuning is significantly improved compares to the trial and error tuning. For the simulation purposes, the performance index for both methods will be analysed for the

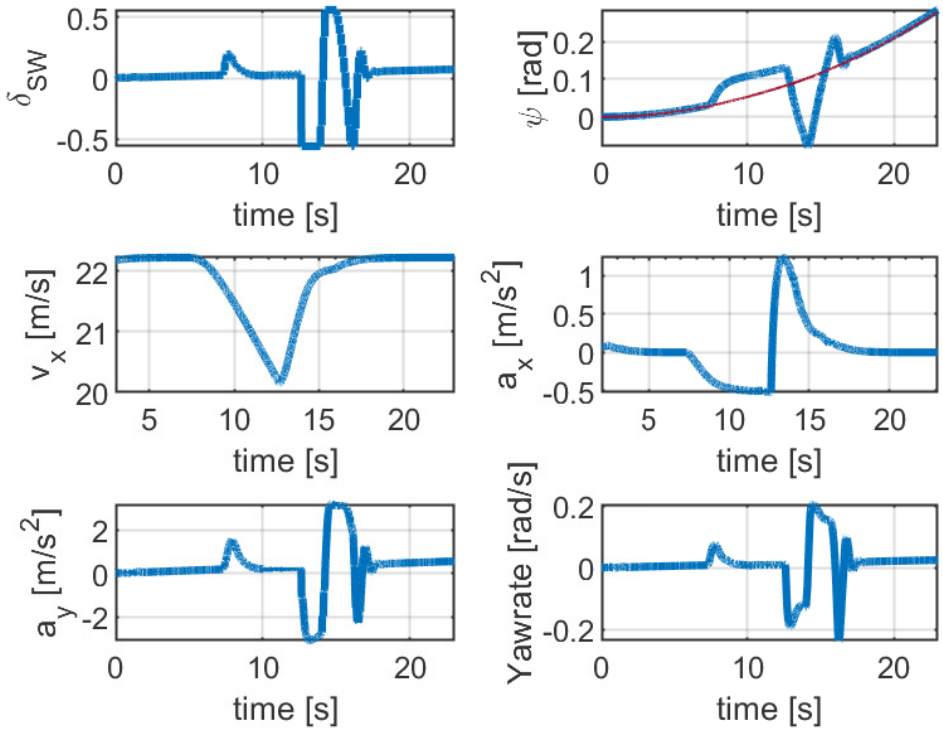

Figure 11. Simulation results: Steering wheel, heading angle (red line indicates the reference heading angle), longitudinal velocity, longitudinal acceleration, lateral acceleration, and yawrate for a collision avoidance maneuver

straight driving scenario. This analysis can be seen in 10 where the performance index is significantly enhanced by the proposed MPC design which gives an improvement close to 80% compare to manual tuning.

7.2. Collision avoidance with moving obstacle and different surface friction conditions

In this section, the efficacy of the proposed algorithm is carried out under different values of surface friction. Moreover, the collision avoidance manoeuvre is performed with the presence of moving vehicle which resembles a real world driving scenario in the simulation environment. The SV is traveling at 80 km h^{-1} located 300 m behind the LV, and LV is traveling at 40 km h^{-1} under different values of friction surface condition ranging from 0.3 to 0.5. Then the system plans the trajectory at each sampling time while applying FCC and RCC constraints based on the position of subject vehicle and moving vehicle. 11, illustrates the collision avoidance simulation with assumption of lowest friction surface condition ($\mu = 0.3$). It is shown that while SV vehicle able to smoothly overtake from moving LV vehicle, it experiences some overshooting when merging back to the original lane. This is due to the sudden and harsh steering actuation of SV vehicle after performing the first lane change from LV vehicle. Nevertheless this amount of actuation is essential in order to maintain the directionality of the vehicle under low surface friction condition. The results in 11 also illustrates that the maximum lateral acceleration reaches to about 3.5 m s^{-2} where the value indicates that the vehicle operates in the linear region of the tire and does not reach to its limit of handling. However, this simulation demonstrates some oscillations as the SV vehicle steer back to the original lane which can be seen in the yawrate profile where it value reaches about 0.26 rad s^{-1}. Overall the simulation results indicate that the vehicle respects the collision avoidance constraints and able to safely converges back to the original lane and follow the desired trajectory on a curve path.

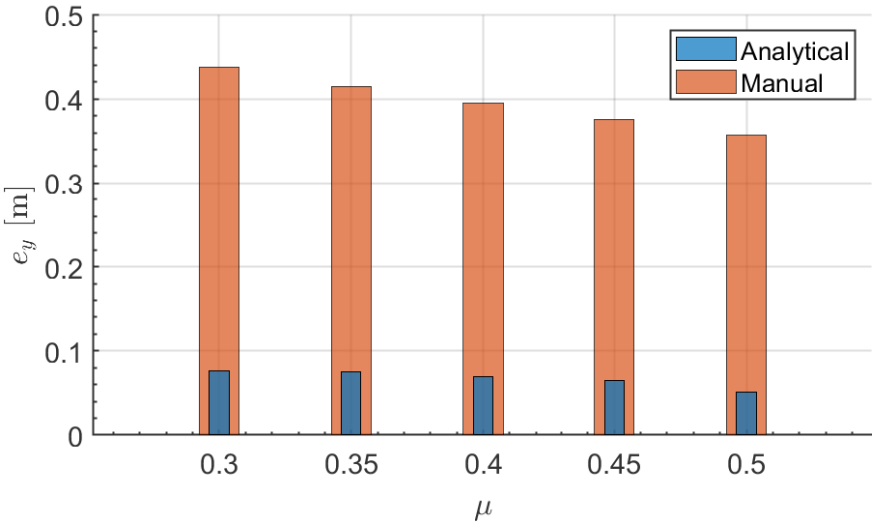

Figure 12. Lateral displacement error over different friction surfaces

Finally, a parametric analysis carried out to verify the effectiveness of the proposed framework after performing collision avoidance manoeuvre. In this analysis, the maximum error between lateral displacement (12) from trajectory planner and actual vehicle response has been tested in order to investigate the tracking capability of the algorithm under different road surface conditions. It is noted that, in this analysis the comparison is made between the proposed analytical MPC and generic MPC controller. The lateral displacement error (e_y) for analytical tuning is increasing gradually with lowering the surface friction. This figure indicates a peak deviation of 0.05 m at 0.3 friction surface which shows major superiority over manual tuning which reaches about 0.44 m with the same friction value. This trend is similar for the rest of the friction values which demonstrates the effectiveness of the proposed control algorithm to improve vehicle handling while performing the collision avoidance manoeuvre.

8. Conclusion

In this paper, we have presented a MPC-based collision avoidance framework by means of an analytical tuning methodology. The main contribution of this paper lies into the derivation of transformation of discrete time LQR weighting matrix Q to generate the MPC weighting matrix. The importance of analytical tuning is investigated next, with results showing that it is not only improve the tracking error, but it can also minimize the performance index compares to manual tuning of MPC. Moreover, the analytical tuning design is then tested under various friction conditions to investigate the efficacy of the proposed algorithm. Results show that while the manual tuning of MPC is effective, the analytical tuning can achieve better performance close to the optimal solution while perform the collision avoidance manoeuvre. The numerical results in Simulink/IPG Carmaker co-simulation environment indicated that, the proposed technique fulfil the safety considerations and generates feasible and collision free trajectory on the curve road.

References

1. Mohammadi, A;Asadi, H; Mohamed S; Nelson, K;Nahavandi, S. Multiobjective and Interactive Genetic Algorithms for Weight Tuning of a Model Predictive Control-Based Motion Cueing Algorithm. *IEEE Transactions on Cybernetics* **2019**, pp. 3471–3481.
2. Gray, A; Gao, Y; Lin, T; Hedrick, J.K; Tseng, H.E; Borrelli, F. Predictive control for agile semi-autonomous ground vehicles using motion primitives. *Proceedings of the American Control Conference* **2012**, pp. 4239–4244.
3. Dixit, S; Montanaro, U; Fallah, S; Dianati, M; Oxtoby, D; Mizutani, T; Mouzakitis, A. Trajectory Planning for Autonomous High-Speed Overtaking using MPC with Terminal Set Constraints. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* **2018**, pp. 1061–1068.

4. Fontes, M; Martins, A.F; Odloak, D. An automatic tuning method for model predictive control strategies. *Proceedings of the American Control Conference* **2019**.
5. Julian B-g. Tuning for MPC Controllers. In *The Role of Population Games in the Design of Optimization-Based Controllers*, 2017; pp. 11-16.
6. Cairano, S; Bemporad, A. Model Predictive Control Tuning by Controller Matching. *IEEE Transactions on Automatic Control* **2010**, 185-190.
7. Shah, G; Engell, S. Tuning MPC for desired closed-loop performance for SISO systems. *18th Mediterranean Conference on Control and Automation, MED'10 - Conference Proceedings* **2010**; pp. 628-633.
8. Clarke, D W; Mohtadi, C; Tuffs, P S. Generalized predictive control-Part I. The basic algorithm. *Automatica* **1987**, pp. 137-148.
9. Shen, C; Gonzalez, Y; Chen, L; Jiang, S. ;Jia, X. Intelligent Parameter Tuning in Optimization-Based Iterative CT Reconstruction via Deep Reinforcement Learning. *IEEE Transactions on Medical Imaging* **2018**, pp. 1430-1439.
10. Qi, Z; Shi, Q; Zhang, H. Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays. *IEEE Transactions on Industrial Electronics* **2019**.
11. Grosso, J. M; Ocampo-Martínez, C; Puig, V. Learning-based tuning of supervisory model predictive control for drinking water networks. *Engineering Applications of Artificial Intelligence* **2013**, pp. 1741-1750.
12. Lee, J.H. van der, W.Y. Svrcek, B.R. Young. A tuning algorithm for model predictive controllers based on geneticalgorithms and fuzzy decision makingtle. *ELSEVIER* **2007**, pp. 53-59.
13. Qi, Z; Shi, Q; Zhang, H. Inverse optimal control for discrete-time finite-horizon Linear Quadratic Regulators. *Automatica* **2019**, pp. 6663-6668.
14. Willems, J; Van De Voorde, H. Inverse optimal control problem for linear discrete-time systems. *Electronics Letters* **1977**, pp. 493.
15. Luo, Wencheng; Chu, Yun Chung; Ling, Keck Voon. Inverse optimal adaptive control for attitude tracking of spacecraft. *IEEE Transactions on Automatic Control* **2005**, pp. 1639-1654.
16. Halder, K; Das, S; Gupta, A. Transformation of LQR Weights for Discretization Invariant Performance of PI/PID Dominant Pole Placement Controllers. *Robotica* **2019**.
17. Kiumarsi-Khomartash, B; Lewis, Frank L; Naghibi-Sistani, Mohammad B; Karimpour, A. Optimal tracking control for linear discrete-time systems using reinforcement learning. *Proceedings of the IEEE Conference on Decision and Control* **2013**, pp. 3845-3850.
18. Franze, G; Lucia, W. A Receding Horizon Control Strategy for Autonomous Vehicles in Dynamic Environments. *IEEE Transactions on Control Systems Technology* **2016**, pp. 695-702.
19. Gao, Y; Gray, A; Frasch, J; Lin, T; Tseng, E; Hedrick, J.K; Borrelli, F. Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles. *Proceedings of the 11th International Symposium on Advanced Vehicle Control* **2012**, pp. 1-6.
20. Karlsson, J; Murgovski, N; Sjöberg, J. Temporal vs. Spatial formulation of autonomous overtaking algorithms. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* **2016**, pp. 1029-1034.
21. Nilsson, J; Falcone, P; Ali, M; Sjöberg, J. Receding horizon maneuver generation for automated highway driving. *Control Engineering Practice* **2015**, pp. 124-133.
22. Kong, J; Pfeiffer, M; Schildbach, G; Borrelli, F. Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. **2015**.
23. Liu, C; Lee, S; Varnhagen, S; Tseng, H.E. Path planning for autonomous vehicles using model predictive control. *IEEE Intelligent Vehicles Symposium, Proceedings* **2017**, pp. 174-179.
24. Gutjahr, B; Gröll, L; Werling, M. Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC. *IEEE Transactions on Intelligent Transportation Systems* **2017**, pp. 1586-1595.
25. Mad, D; Philippsen, R; Eidehall, A; Dahl, K. On Path Planning Methods for Automotive Collision Avoidance. **2013**.
26. Nilsson, J; Gao, Y; Carvalho, A; Borrelli, F. Manoeuvre generation and control for automated highway driving. *IIFAC Proceedings Volumes* **2014**, pp. 6301-6306.
27. Gao, Y; Gray, A; Carvalho, A; Tseng, H.E; Borrelli, F. Robust nonlinear predictive control for semiautonomous ground vehicles. *Proceedings of the American Control Conference* **2014**, pp. 4913-4918.
28. Plessen, M; Lima, P; Martensson, J; Bemporad, A; Wahlberg, B. Trajectory planning under vehicle dimension constraints using sequential linear programming. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* **2018**, pp. 1-6.
29. Scheuer, A; Fraichard, T. Collision-free and continuous-curvature path planning for car-like robots. *Proceedings - IEEE International Conference on Robotics and Automation* **1997**, pp. 867-873.
30. Shin, D.H; Singh, S. Path Generation for a Robot Vehicle Using Composite Clothoid Segments. *IFAC Proceedings Volumes* **1990**.
31. The practice of Clothoid Connections on Road Interchanges. Available online: http://precismultipla.altervista.org/ESU2/chap08.htm.
32. Funke, J. Collision Avoidance Up To The Handling Limits For Autonomous Vehicles. PhD, Standford University, 2015.
33. Schmeitz, A; Zegers, J; Ploeg, J; Alirezaei, M. Towards a generic lateral control concept for cooperative automated driving theoretical and experimental evaluation. *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017 - Proceedings* **2017**, pp. 134-139.
34. Chatzikomis, C; Sorniotti, A; Gruber, P; Zanchetta, M; Willans, D. Comparison of Path Tracking and Torque- Vectoring Controllers for Autonomous Electric Vehicles. *IEEE* **2018**, pp. 559-570.