Edinburgh Research Explorer

Autonomous Collision Avoidance Using MPC with LQR-Based Weight Transformation

Article

Autonomous Collision Avoidance Using MPC with LQR-Based Weight Transformation

Shayan Taherian, Kaushik Halder, Shilp Dixit and Saber Fallah *

Department of Mechanical Engineering Sciences, Connected Autonomous Vehicle Lab (CAV-Lab), University of Surrey, Guildford GU2 7XH, UK; s.taherian@surrey.ac.uk (S.T.); k.halder@surrey.ac.uk (K.H.); s.dixit@surrey.ac.uk (S.D.)
* Correspondence: s.fallah@surrey.ac.uk

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 a LQR weighting matrix with a fixed weighting factor using a discrete algebraic Riccati equation (DARE) and designs an MPC controller using the idea of a discrete time linear quadratic tracking problem (LQT) in the presence of constraints. The proposed methodology ensures optimal performance between unconstrained MPC and LQR controllers 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 an 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 that 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 techniques 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 reliant 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 that requires a large number of trial and error selections and many computer simulations [1–3]. The complexity increases with an increase in the number of control objectives [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 an LQR-based linear controller. In this work, an LMI-based LQR approach was chosen by recasting the DARE in the LMI form to calculate the weighting matrices of an unconstrained MPC. Although this approach was found as a solution for MPC control parameters, the cost function was limited to have zero cross state-space terms, such that only a 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, a higher control horizon leads to a set of inequality bilinear constraints in the unknown elements, which cannot be solved using convex optimisation for linear MPC. In [9], the authors proposed an analytical tuning method for the Dynamic Matrix Control (DMC) algorithm. DMC is one of the primary versions of MPC that utilizes the step response of the plant, which can be approximated by the first-order plus delay time (FOPDT) model. The proposed analytical method offers several advantages, such as low computational cost and the conservative behaviour of the control system in chemical plants. However, this approach fails to demonstrate the effectiveness of applying quadratic programming solvers or additional adaptive mechanisms within the controller architecture. Other tuning strategies have been investigated by researchers using intelligent learning algorithms [10,11]. In [12,13], the authors implemented neural networks and fuzzy-based decision making to tune the MPC parameters, without using the reference system model. The authors of [1] proposed an automated strategy for an 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 algorithms is that the tuning parameters are updated along with the optimisation algorithm, and thus the solution is 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 time step (optimisation required by intelligent algorithms and additional optimisation required by the MPC controller) that may require overhead computational time. In [14], the authors proposed a full review of tuning strategies for different MPC formulations, such as dynamic matrix control, generalized predictive control and state-space representation of an MPC controller. Moreover, a review on a few autotuning strategies is available in this work. The authors mentioned that these strategies benefit from not having full knowledge of the model of the system to initialize the tuning procedure. However, the major disadvantage of the autotuning strategy is the computational overhead, which has to perform two optimizations compared to offline tuning strategies. In [15], the authors proposed an evolutionary algorithm using a non-dominated sorting genetic algorithm for a multi-objective optimisation problem within an MPC framework. This method offers simple procedure to handle MPC constraints. Moreover, this algorithm suggests a flexible representation of the decision variables that expedite the controller performance. However, this algorithm suffers from computation overhead, as it requires two optimisations per time step.

In this paper, the inverse optimal control problem [16–18] of a discrete time linear quadratic regulator (DLQR) is used for designing an MPC controller to perform collision avoidance manoeuvres 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) that 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 [19]. 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 solving the reference tracking problem. To overcome this problem, we recast DLQR into a discrete time linear quadratic tracking (DLQT) problem [20] to design the weighting matrix Q, which can be used for MPC design to ensure solving the 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 a trajectory planning task. In order to highlight the 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, a collision avoidance system is formulated as a finite horizon optimisation problem with vehicle dynamics, while the collision avoidance constraints are formulated within the 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 [21], changing the reference frame [22,23] and affine linear collision avoidance constraints [24]. In this paper the collision avoidance constraints are formulated in the form of linear affine constraints to perform trajectory planning manoeuvres 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 dimensions 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 the entire framework for autonomous collision avoidance system on a curved road is validated in a co-simulation platform where a high-fidelity vehicle model is simulated in IPG-CarMaker while the proposed MPC design is implemented in MATLAB/Simulink.

The contributions of this paper are summarised as follows: *(i)* Design of an analytical tuning technique to perform a collision-free trajectory for autonomous vehicles, and *(ii)* development of a mathematical framework to design convex constraints in order to perform collision avoidance manoeuvres 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 the weighting matrix Q is described. To design the tuning strategy, a discrete time inverse optimal control problem is formulated using a 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 a discrete time algebraic Ricatti equation, which will ensure almost equal optimality between the DLQR and the unconstrained MPC control problem. It is noted that in this design the weighting factor R is chosen as a fixed value as in [19]. Moreover, the weighting matrix Q obtained from the 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 for finding the MPC weighting matrix Q by obtaining the sub-optimal solution from the 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 in DLQR as a discrete linear quadratic tracking (DLQT) problem. Next, with the concept of the inverse DLQT control problem, a strategy for designing an analytical approach to select the Q matrix for the MPC control design will be discussed.

2.1. 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 $R = R^T = \begin{bmatrix} 10 & 0 \\ 0 & 9 \times 10^{-2} \end{bmatrix} > 0$. 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 the 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:

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

2.2. 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 be ensured 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 [20], and for the brevity of the paper, an interested reader can refer to this technical paper for calculation of the 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) holds. The Q matrix obtained from the DLQR problem can be used for the 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 the DLQT approach. Hence, we can use the Q matrix obtained from the inverse DLQR approach in the MPC control problem, since the MPC reference tracking problem is equivalent to the 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 [25]. This model assumes no slip between the tyre and the road, and is found to be a suitable model for trajectory planning for the application of collision avoidance manoeuvres [26,27]. Under the assumption of small angles approximation [28] the vehicle bicycle model is:

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

$$\dot{y} = v_x \psi + \frac{l_r}{l_f + l_r} v_x \delta_f \tag{13}$$

$$\dot{v}_x = a_x \tag{14}$$

$$\dot{\psi} = \frac{v_x \delta_f}{l_f + l_r} \tag{15}$$

where ξ and y are the longitudinal and lateral displacement of the vehicle in inertial coordinates, respectively, 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 the centre of gravity (C.G), and l_r is the distance of the rear axle from C.G. The control inputs are the steering angle δ_f and longitudinal acceleration a_x. For a given nominal velocity $v_{x,nom}$, the system in (12), (13), (14), (15) can be expressed using an LTI system using the compact notation given below:

$$\dot{x} = Ax + Bu \tag{16}$$

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 $\mathcal{U}$ are polyhedron regions used as state and input constraints, respectively. The structure of the system matrix A and input matrix B are as follows:

$$\begin{bmatrix} \dot{\xi} \\ \dot{y} \\ \dot{v}_x \\ \dot{\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & v_{x,nom} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \begin{bmatrix} \xi \\ y \\ v_x \\ \psi \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{v_{x,nom} l_r}{(l_r + l_f)} & 0 \\ 0 & 1 \\ \frac{v_{x,nom}}{(l_r + l_f)} & 0 \end{bmatrix}}_{B} \begin{bmatrix} \delta \\ a_x \end{bmatrix} \tag{17}$$

where $v_{x,nom}$ is the nominal longitudinal velocity of the vehicle, which is chosen in the same way as the initial speed of the vehicle [3]. Then system (16) is descretised with a sampling time t_s to obtain the linear time invariant discrete system shown below:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{18}$$

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

As the dynamics of the state ξ in system (17) depends only on v, it is possible to further simplify the system for trajectory generation:

$$\bar{x}(k+1) = \bar{A}_d \bar{x}(k) + \bar{B}_d u(k) \tag{19}$$

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, respectively, in (17).

4. Selection of Weighting Matrix Q for MPC Design

In order to obtain the appropriate Q matrix, for designing an MPC-based reference tracking controller, we have chosen $Q = diag(Q_{11}, Q_{22}, Q_{33})$ for system (19) and the optimal control gain $\bar{K}$ with a fixed weighting factor R. For the reference tracking problem, MPC can be recast as a DLQT problem as described in Section 2.2. Therefore, it can be inferred that the weighting matrix Q obtained from the 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 system (19), guaranteeing the sub-optimality of the 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. Equation (3) can be recast as a DLQR problem:

$$Q = P - \bar{A}_d^T P \bar{A}_d + \bar{A}_d^T P \bar{B}_d (R + \bar{B}_d^T P \bar{B}_d)^{-1} \bar{B}_d^T P \bar{A}_d \tag{20}$$

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:

$$\bar{K} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \end{bmatrix} = (R + \bar{B}_d^T P \bar{B}_d)^{-1} \bar{B}_d^T P \bar{A}_d \tag{21}$$

Since the input $\bar{B}_d$ matrix in (19) 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 the MPC controller. □

Now, using (19), (21) and the matrices (P,Q,R) in (20) yields the diagonal and off-diagonal elements of the Q matrix as shown in (25)–(27) and

$$Q_{12} = -k_2 \left[P_{11} \frac{vl_r t_s}{L} + P_{13} \frac{vt_s}{L} \right] - P_{12} t_s k_5 \tag{22}$$

$$Q_{13} = vt_s P_{11} - \left[k_3 \left[P_{11} \frac{vl_r t_s}{L} + P_{13} \frac{vt_s}{L} \right] + P_{12} t_s k_6 \right] \tag{23}$$

$$Q_{32} = vt_s P_{12} - \left[\left[\frac{vl_r}{L} t_s (vt_s P_{11} + P_{13}) \right. \right. \\ \left. \left. + \frac{v}{L} t_s (vt_s P_{13} + P_{33}) \right] k_2 + (vt_s P_{12} + P_{23}) k_5 \right] \tag{24}$$

respectively. Since matrix Q is diagonal, all of 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 the P matrix can be obtained, i.e., (22) yields (32). Using (32) in (23) yields (29), and using (32) in (24) results in (33). Now, using the elements (32), (33), (29) and the R, B_d matrices in (21), Equations (34)–(36) will hold. Finally, by solving (34)–(36), Equations (28), (30) and (31) 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 (18) would be inside the unit circle. In this paper, Ackerman's pole placement formula is used to obtain the gains $(\bar{K})$ by specifying the closed-loop pole location inside the unit circle. This method solves the solution of the Riccati equation to generate optimal weights using an LQR controller. The

generated gains are then used in the MPC controller, which inherits all of the properties and optimal solutions from the LQR controller when MPC constraints are not active. However, when the constraints are active, the solution of the Riccati equation in LQR provides a sub-optimal solution for MPC, as it satisfies the constraints of the Quadratic Programming (QP) problem indirectly. In other words, the set of states $\bar{x}(k)$ where the matching occurs is the polyhedron, where the unconstrained optimizer (i.e., unconstrained MPC, therefore the DLQR controller) satisfies the constraints of the QP problem [6].

$$Q_{11} = (P_{11} \frac{vl_r}{l_r + l_f} t_s + P_{13} \frac{v}{l_r + l_f} t_s)k_1 \tag{25}$$

$$Q_{22} = P_{22} t_s k_5 \tag{26}$$

$$Q_{33} = -vt_s P_{13} + \left[\frac{vl_r}{l_r + l_f} t_s (vt_s P_{11} + P_{13}) + \frac{v}{l_r + l_f} t_s (vt_s P_{13} + P_{33}) \right] \tag{27}$$

$$P_{13} = \frac{-\frac{R_1 k_1 v t_s}{L} \left[\frac{k_3 v t_s}{L} + (1 - \frac{v t_s}{L}) \right]}{\frac{v^4 t_s^4 k_1}{L^3} \left[\frac{v l_r t_s k_3}{L} - l_r - v t_s \right] (1 - \frac{l_r k_3}{L}) - \frac{v^3 t_s^3}{L^2} \left[(1 - \frac{v l_r t_s k_1}{L})(1 - \frac{v t_s}{L})(1 - \frac{l_r k_3}{L}) \right] + \frac{v^2 l_r t_s^2}{L} (\frac{l_r k_3}{L} - 1) - \frac{v^3 t_s^3 k_1 k_3}{L^3}} \tag{28}$$

$$P_{11} = -\frac{P_{13} \frac{v}{L} t_s k_3}{\frac{vl_r}{L} t_s k_3 - v t_s} \tag{29}$$

$$P_{33} = \frac{P_{11} \left[\frac{v t_s l_r}{L} - \frac{k_1 v^2 l_r^2 t_s^2}{L^2} \right] - k_1 R_1 + P_{13} \left[\frac{v t_s}{L} - \frac{v^2 l_r t_s^2 k_1}{L^2} \right]}{\frac{k_1 v^2 t s^2}{L^2}} \tag{30}$$

$$P_{22} = \frac{R_2 k_5}{ts - ts^2 k_5} \tag{31}$$

$$P_{12} = 0 \tag{32}$$

$$P_{23} = 0 \tag{33}$$

$$k_1 = \frac{\frac{P_{11} v l_r t_s}{L} + \frac{P_{13} v t_s}{L}}{R_1 + \frac{v^2 l_r^2 t_s^2 P_{11}}{L^2} + \frac{\left[\frac{P_{13} v l_r t_s}{L} + \frac{P_{33} v t_s}{L} \right] v t_s}{L}} \tag{34}$$

$$k_3 = \frac{\frac{P_{11} v^2 l_r t_s}{L} + P_{13} \frac{[v t_s P_{13} + P_{33}] v t_s}{L}}{R_1 + \frac{P_{11} v^2 t_s^2}{L^2} + \frac{\left[\frac{P_{13} v t_s}{L} + \frac{P_3 3 v t_s}{L} \right] v t_s}{L}} \tag{35}$$

$$k_5 = \frac{P_{22} t_s}{R_2 + P_{22} t_s^2} \tag{36}$$

$$k_2 = k_4 = k_6 = 0 \tag{37}$$

5. Constraint Design

In this section, the mathematical framework for computing the collision avoidance constraints that 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 maintain a safe distance from the obstacle. A detailed explanation of the aforementioned constraints designs is presented 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 a Quadratic Programming (QP) framework [29]. 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 (Figure 1). The edges of the virtual straight road are obtained by calculating the points P_1 and P_2 in Figure 1. These virtual straight lines can be constructed using the generated points P_1 and P_2, followed by 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 the outer lane boundary and the intersected line are:

$$y_{outer} = f(x_{outer}) \tag{38}$$

$$a_{outer}\xi + b_{outer}y + c_{outer} = 0 \tag{39}$$

where (38) is the equation of the outer lane boundary, (39) 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 (39) 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 Figure 1) with the following equation:

$$m_{outer}\xi - y + P_{1y} - m_{outer}P_{1x} < 0 \tag{40}$$

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

$$\bar{\xi}(j) = \left[\xi_0 + \sum_{i=1}^{j}(\bar{v}(i) \cdot t_s)\right]; \quad j = 1, 2, ..., N_p \tag{41}$$

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 Equation (40) to formulate the generalized constraint equation for N_p; different constraint equations given below:

$$m_{outer}\left[\xi_0 + \sum_{i=1}^{j}(\bar{v}(i) \cdot t_s)\right] - y(j) - m_{outer}P_{1x} + P_{1y} < 0 \tag{42}$$

where the predicted nominal velocity and predicted lateral position can be obtained from the prediction model in MPC. For different values of j in (41), Equation (42) represents the outer edge of the virtual straight road constraint, which solely depends on two states of the vehicle (longitudinal velocity and lateral position of the vehicle). 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:

$$y_{inner} = f(x_{inner}) \tag{43}$$

$$a_{inner}\xi + b_{inner}y + c_{inner} = 0 \tag{44}$$

where (43) is the equation of the inner lane boundary and (39) 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:

$$m_{inner}\left[\xi_0 + \sum_{i=1}^{j}(\bar{v}(i) \cdot t_s)\right] - y(j) - m_{inner}P_{4x} + P_{4y} > 0 \tag{45}$$
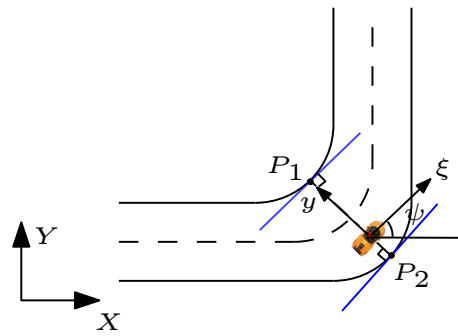
Figure 1. The hyperplanes (blue lines) indicate the constraints for optimization. The points on the borders are the projection of the centre line of the subject vehicle on the borders.

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 [30,31]. 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 Figure 2. This technique is inspired from [24], 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.
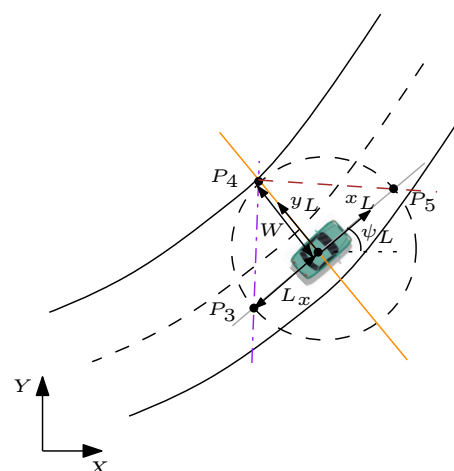

Figure 2. Schematic to construct forward and rear collision avoidance constraints line. Note: ψ_L is lead vehicle orientation, L_x and W indicate the safety distance from the lead vehicle.

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 Figure 3. Points (P_{3x}, P_{3y}) are obtained as a result of the intersection of the longitudinal axis of the lead vehicle (grey line in Figure 2):

$$a_{Lon}x_L + b_{Lon}y_L + c_{Lon} = 0 \tag{46}$$

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

$$(x - x_L)^2 + (y - y_L)^2 = (L_x + l_{rL})^2 \tag{47}$$

where L_x is the 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 the subject vehicle when approaching the lead vehicle, and finally L_c is the lead vehicle length. Moreover, the parameters $l_{rL}, a_{Lon}, b_{Lon}, c_{Lon}$ represent the distance of the centre of gravity of the vehicle to the rear wheels and coefficients of the intersected line, respectively. It is noted that at every time instant of MPC optimization, L_x is 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 the intersection of the lateral axis of the lead vehicle (yellow line in Figure 2) with a circle of radius W:

$$(x - x_L)^2 + (y - y_L)^2 = W^2 \tag{48}$$

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 distance from the subject vehicle to the centre line of the road [24]. After defining two sets of points, it would be trivial to find the equation of the line for forward collision avoidance constraints, where the final equation will be:

$$\underbrace{m_{FCC} \cdot t_s}_{a_{FCC}} \left[\sum_{i=1}^{j} (\bar{v}(i)) \right] + \underbrace{(-1)}_{b_{FCC}}y + \underbrace{P_{1y} - m_{FCC}P_{1x} + m_{FCC} \cdot \xi_0}_{c_{FCC}} > 0 \tag{49}$$

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, and Region 2: $a_{FCC}x + b_{FCC}y + c_{FCC} < 0$, which represents the unsafe region. FCC forces the vehicle to be in a safe region while performing a lane change on a curved/straight road. Moreover, (49) represents a linear affine constraint approximation, which can be formulated in QP format.
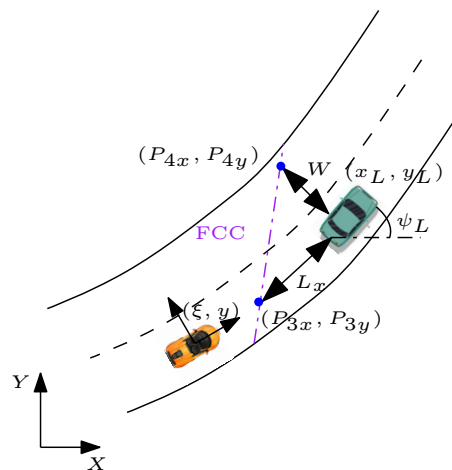

Figure 3. Schematic used to calculate the FCC. Note: orange car—subject vehicle; green car—lead vehicle; red line—RCC; L_x and W indicate the safety distance from the lead vehicle.

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 (Figure 4):

$$\underbrace{m_{RCC} \cdot t_s}_{a_{RCC}} \left[\sum_{i=1}^{j} (\bar{v}(i)) \right] + \underbrace{(-1)}_{b_{RCC}} y + \underbrace{P_{3y} - m_{RCC}P_{3x} + m_{RCC} \cdot \xi_0}_{c_{RCC}} < 0 \qquad (50)$$

It is noteworthy that the aforementioned collision avoidance constraints can be adopted for the scenarios where (i) the subject vehicle needs to change a lane while performing the collision avoidance manoeuvre, and (ii) where more traffic members are present on the road that 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 that forces MPC to plan a safe and collision-free trajectory on the road.
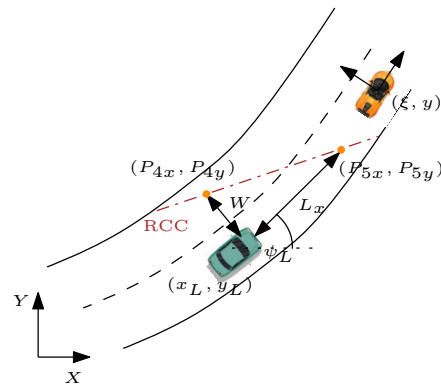

Figure 4. Schematic used to calculate RCC. Note: orange car—subject vehicle; green car—lead vehicle; L_x and W indicate the safety distance from the lead vehicle.

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 keep the vehicle away from lane boundaries while avoiding possible collisions. The choice of manoeuvre to perform is the result of the decision-making process with the objective of following the desired trajectory x_{ref}, y_{ref} (Section 6.1) while respecting physical and design limitations of the subject vehicle. In this section, using the definition of the linear model in (16) and linear system constraints in Figure 5, a constrained linear quadratic optimal control problem is formulated over a prediction horizon N_p.
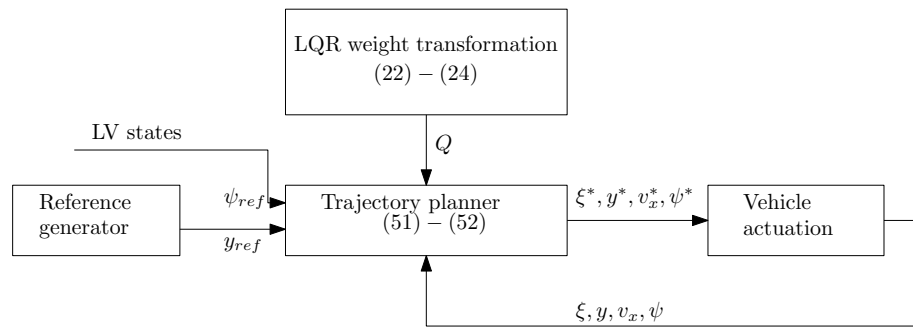
Figure 5. Closed-loop framework for collision avoidance Note: LV indicates lead vehicle.

6.1. Reference Trajectory Generator

The reference trajectory generator provides the longitudinal position $x_{ref}(s)$ and lateral positions $y_{ref}(s)$ as well as the 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:

$$\kappa = \frac{\dot{x}_{ref}\ddot{y}_{ref} - \dot{y}_{ref}\ddot{x}_{ref}}{(\dot{x}_{ref}^2 + \dot{y}_{ref}^2)^{\frac{3}{2}}} \tag{51}$$

The reference trajectories are formulated using the clothoids method, which is expressed by Fresnel integrals [32,33]. The trajectories are parametrized by curvature κ as a function of distance s along the path as follows:

$$x_{ref}(s) = \int_0^s \cos \psi_{ref}(x)dx \tag{52}$$

$$y_{ref}(s) = \int_0^s \sin \psi_{ref}(x)dx \tag{53}$$

where (52) and (53) are constructed using the polynomial integration method of the Fresnel function applied to clothoid [34], and ψ_{ref} is calculated as follows:

$$\psi_{ref}(s) = 2\tan^{-1}\left(\frac{y_{ref,n} - y_{ref,n-1}}{x_{ref,n} - x_{ref,n-1}}\right) \tag{54}$$

where $(x_{ref,n}), y_{ref,n}$, $n = 1...N$ are the sequence of N, the sampled points on the trajectory [35].

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:

$$\bar{x} \in \bar{\mathcal{X}}, u \in \mathcal{U} \tag{55}$$

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 of the polytope admissible regions (subscripts *min* and *max* are the minimum and maximum of the corresponding values). The following cost function is formulated:

$$\mathcal{J} = \sum_{k=0}^{N_p-1} [\| \bar{x}_{ref} - \bar{x}_{t+k|t} \|_Q^2 + \| u_{t+k|t} \|_R^2]$$
$$+ \| \bar{x}_{ref} - \bar{x}_{t+N|t} \|_P^2 \tag{56}$$

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 (18), starting from an initial state of $\xi_{t|t}$.

The parameter $N_p \in \mathbb{N}^+$ is the prediction horizon, whereas $Q \in \mathbb{R}^{3\times3}$, $P \in \mathbb{R}^{3\times3}$ and $R \in \mathbb{R}^{2\times2}$ are weighting matrices. The performance index in (56) consists of the stage cost, input cost and the terminal cost, respectively. The desired state $\bar{x}_{ref}$ represents 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:

$$\min_{U_t} \mathcal{J}(u_t; \bar{x}(t), \bar{x}_{ref}) \tag{57}$$

subject to (18), (42), (45), (49), (50), (55)

In this framework, at every time step, the problem (57), under the constraints (18), (42), (45), (49), (50), (55) are calculated based on the current state $x(t)$, over a shifted time horizon. The optimal control input is calculated as $u^*(\bar{x}(t)) = U_t^*(0)$, which is a solution to the problem (57). As the sets $\bar{\mathcal{X}}$ and $\mathcal{U}$ are convex, then the MPC problem (57) is solved as a standard QP optimisation problem. All of the steps required to design the proposed MPC controller can be seen in Algorithm 1, and the closed-loop form is illustrated in Figure 5. The following demonstrates the numerical results to show the efficacy of the proposed control strategy.

Algorithm 1 Trajectory planning

1: *initialize*:
2: $v_{des} \leftarrow$ desired longitudinal velocity from user
3: **procedure** GENERATE TRAJECTORY
4: *top*:
5:     Use Equation (3) to create matrix Q
6:     $\bar{K} \leftarrow$ Ackerman pole placement formula as (4) and calculate $\bar{K}$
7:     $P \leftarrow$ Obtained $\bar{K}$ in (28)–(37) and generate P
8:     $Q \leftarrow$ Obtained P and substitute (19) in (25)–(27) to calculate Q
9: *loop*:
10:     $\bar{x}_{ref} \leftarrow$ generate target states as
11:     $\bar{x} \leftarrow$ current state vector
12:     *getRoadBoundaryConstraints* as (38) (39)–(45)
13:     *getCollisionAvoidanceConstraints* as (46)–(50)
14:     $u^* \leftarrow$ solve MPC with generated Q from *top* with fixed R
15:     $x^* \leftarrow$ Apply optimal input $u(k)$ as (12)–(15)
16:     **if** user request to change v_{des} **then**
17:         **goto** *initialize*.
18:         **else**
19:     **goto** *loop*.

---

7. Numerical Results

In this section, a simulation environment is used to investigate the ability of the proposed trajectory planning architecture to perform a collision avoidance manoeuvre. The scenarios used to perform this are: (1) an obstacle avoidance when the subject vehicle (SV) is 300 m behind the stationary lead vehicle (LV), and travelling at $80 \, \text{km} \, \text{h}^{-1}$ on a two-lane one-way curved road with a radius of 750 m, (2) a collision avoidance manoeuvre with a moving obstacle under different road surface friction conditions. It is noted that the chosen curved road is approximately equal to the tightest corner allowed in highways [36]. The simulation results for the proposed closed-loop architecture were obtained using the MATLAB IPG CarMaker co-simulation environment.

Moreover, it was assumed that the dimension of the road, lane limits and LVs states are available to the SV, and the initialisation parameters are tabulated in Table 1. Additionally, to evaluate the efficacy of the proposed algorithm, MPC was initialized with a prediction

horizon equal to 14 and low time gap ($t = 0.8$) when designing the constraints for imitating more extreme manoeuvres to perform a 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 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 was compared with a generic MPC controller where the values of the Q matrix have been generated through a trial and error procedure (which will be referred to 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} was based on trial and error and the 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 was equally chosen be to $R = diag(10, 9 \times 10^{-2})$.

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	
$\mathcal{X}$	$-[500, -11.1, 1.5] \leq x \leq [500, 27.7778, 1.5]$	-
$\mathcal{U}$	$-[0.0698, 3] \leq u \leq [0.0698, 3]$	-

7.1. Collision Avoidance with Static Obstacle

The result of the obstacle avoidance scenario with the assumption of a high friction surface, using the proposed MPC design, is demonstrated in Figure 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 Figure 6 indicates that the proposed structure generated a consistent trajectory with no overshoot while maintaining the safety margins to the lead vehicle during both lane changes.

For further analysis of the performance of the proposed collision avoidance system, the evolution of the states and control inputs of the SV over time is shown in Figure 7. A particular feature of 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 reverse behaviour, where the SV attempts returning to the original lane. Another key point is the two states of the SV, heading-angle and 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 behaviour (i.e., increasing the velocity while performing 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 accelerates after evading the obstacle, which shows the efficacy of the proposed control structure.
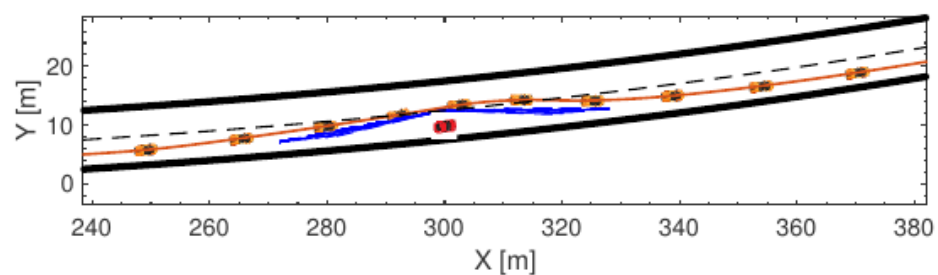

Figure 6. Simulation result: SV and LV (red vehicle) trajectories. Note: FCC and RCC collision avoidance constraints represented as blue dashed lines to provide a safe region between SV and LV.
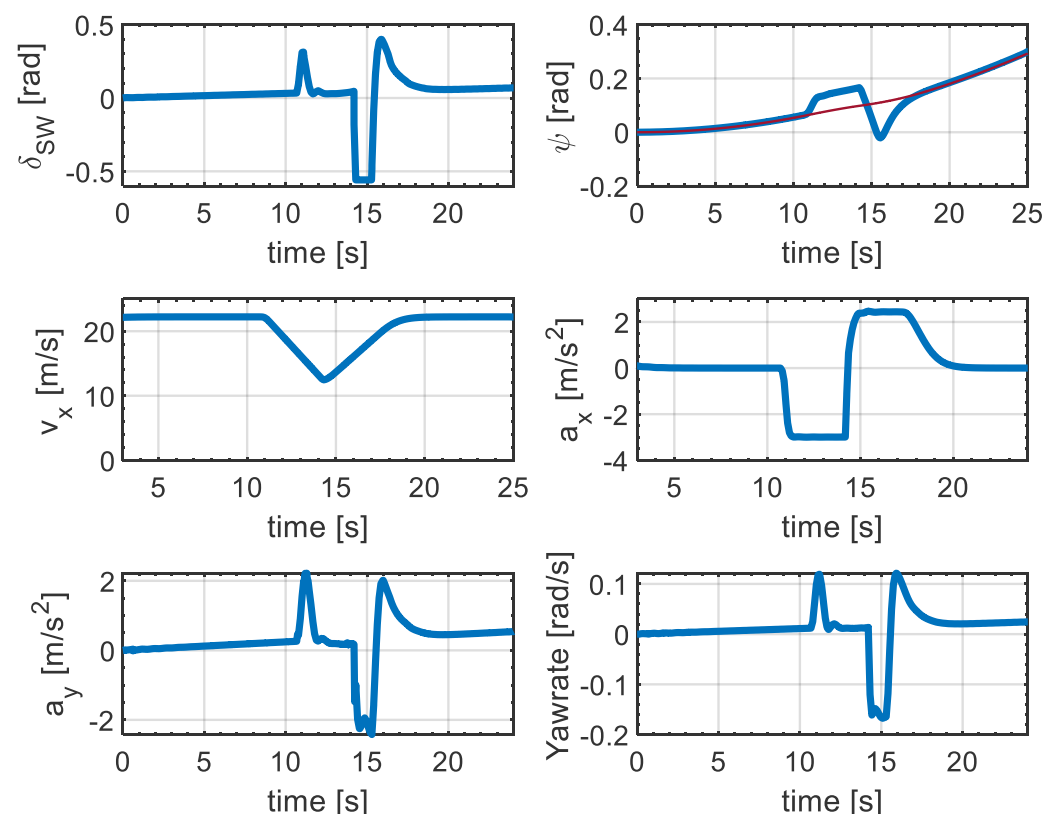

Figure 7. Simulation results: Steering wheel, heading angle (red line indicates the reference heading angle), longitudinal velocity, longitudinal acceleration, lateral acceleration, and yaw-rate for a collision avoidance maneuver.

Moreover, the longitudinal acceleration also shows a smooth profile without any oscillation during deceleration and acceleration for both of the lane changes while respecting the system constraints. Similarly, the bottom subplot in Figure 7 demonstrates the lateral

acceleration and yaw-rate of the SV, which shows a smooth profile without having high oscillation for both 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 its limits of handling. This can be observed by investigating the yaw-rate profile, which the maximum absolute value reaches at about $0.18\,\mathrm{rad\,s^{-1}}$, indicating that the vehicle operates in the linear region of the tire.

To show the need for the proposed MPC design to perform the collision avoidance manoeuvre, we compared the performance of the proposed framework with the MPC design using manual tuning. Figure 8 demonstrates the lateral and heading angle errors for both the analytical tuning and manual tuning of the MPC. In this analysis, the lateral error e_y is defined as the 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 the desired heading angle, which are defined as [37]:

$$e_y = (y_{CG} - y_{ref}) \cos \psi_{ref} - (x_{CG} - x_{ref}) \sin \psi_{ref} \tag{58a}$$
$$e_\psi = \psi - \psi_{ref} \tag{58b}$$

The top figure shows the heading angle error, which shows similar behaviour for both the 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 $0.15\,\mathrm{rad}$. 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 the analytical tuning strategy almost converges to zero, whereas the conventional tuning reaches about $30\,\mathrm{cm}$. This figure confirms that the proposed strategy is able to successfully maintain the directionality of the vehicle during the collision avoidance manoeuvre while performing better reference tracking with minimum tracking error compared to conventional trial and error tuning.

For further illustration of the effectiveness of the proposed MPC design, we compared the performance index value of the proposed technique with the MPC design using manual tuning to investigate the level of optimality between the two procedures. This can be seen in Figure 9, where the performance index ($\mathcal{J}$) indicates the minimized value of the cost function in (56). It can be inferred that the proposed MPC design gives an improvement in $\mathcal{J}$ of more than 70% over the manual tuning between $11\,\mathrm{s}$ and $15\,\mathrm{s}$, where the performance index is at its maximum, and for the rest of the manoeuvre the performance index for the analytical tuning is still significantly improved compared to the trial and error tuning. For the simulation purposes, the performance index for both methods was analysed for the straight driving scenario. This analysis can be seen in Figure 10, where the performance index is significantly enhanced by the proposed MPC design, which resulted in an improvement close to 80% compared 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 examined under different values of surface friction. Moreover, the collision avoidance manoeuvre is performed with the presence of a moving vehicle, which resembles a real world driving scenario in the simulation environment. The SV is traveling at $80\,\mathrm{km\,h^{-1}}$ located $300\,\mathrm{m}$ behind the LV, and the LV is traveling at $40\,\mathrm{km\,h^{-1}}$ under different values of surface friction conditions 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 the subject vehicle and the moving vehicle. Figure 11 illustrates the collision avoidance simulation with an assumption of the lowest friction surface condition ($\mu = 0.3$).
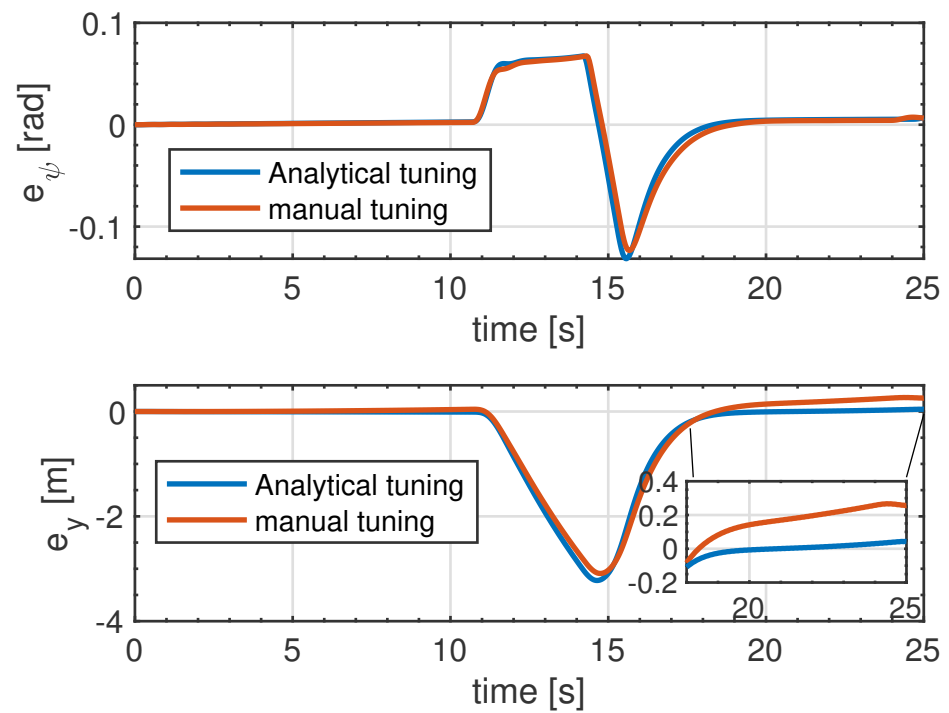
Figure 8. Lateral error e_y and heading error e_ψ for analytical and manual tuning approach.
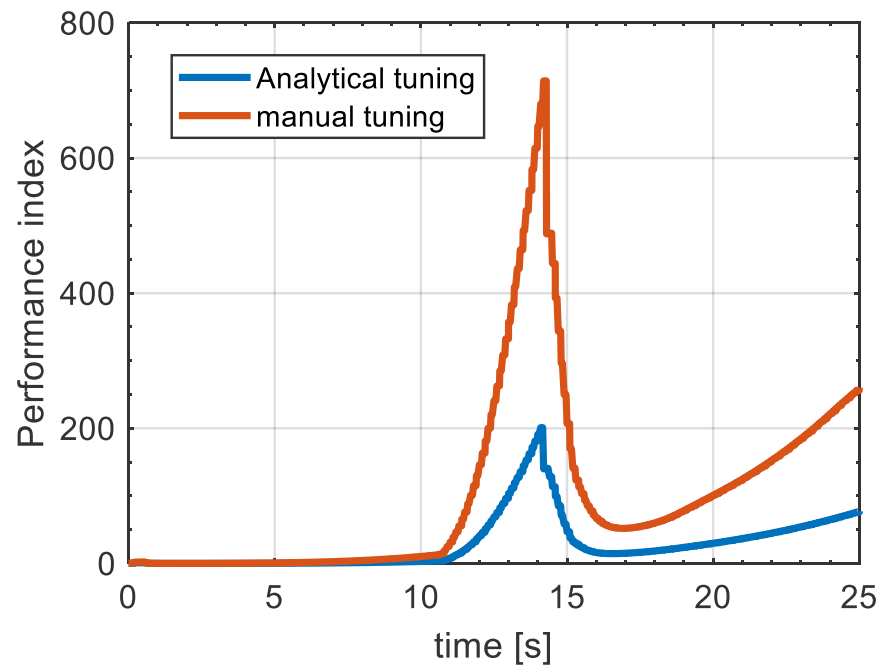

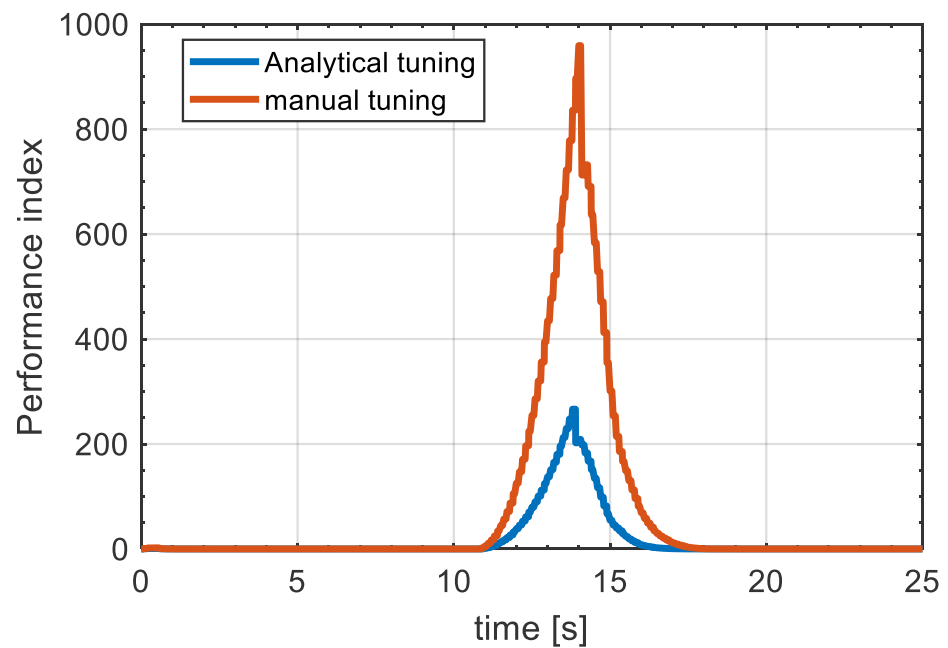**Figure 9.** Performance index for both automatic tuning and manual tuning on curved driving condition.

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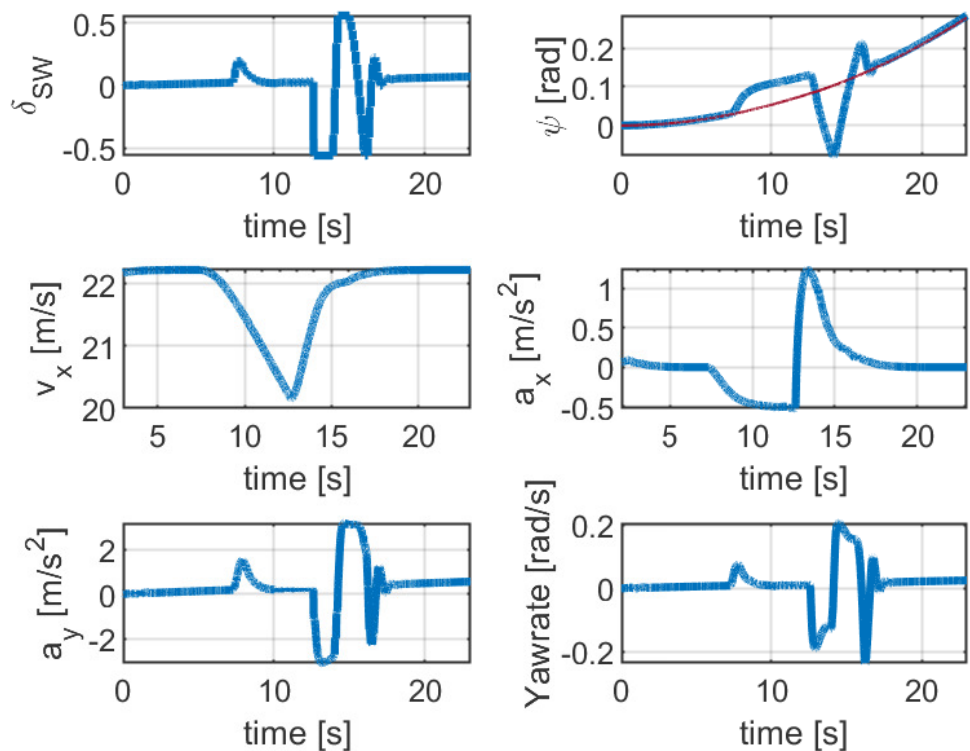

Figure 11. Simulation results: Steering wheel, heading angle (red line indicates the reference heading angle), longitudinal velocity, longitudinal acceleration, lateral acceleration, and yaw-rate for a collision avoidance manoeuvre.

It is shown that while the SV vehicle was able to smoothly overtake the moving LV vehicle, it experienced some overshooting when merging back to the original lane. This is due to the sudden and harsh steering actuation of the SV vehicle after performing the first lane change from the LV vehicle. Nevertheless, this amount of actuation is essential in

order to maintain the directionality of the vehicle under low surface friction conditions. The results shown in Figure 11 also illustrate that the maximum lateral acceleration reached about $3.5\,\mathrm{m\,s^{-2}}$, where the value indicates that the vehicle operates in the linear region of the tire and does not reach its limit of handling. However, this simulation demonstrated some oscillations as the SV vehicle steered back to the original lane, which can be seen in the yaw-rate profile, where its value reached about $0.26\,\mathrm{rad\,s^{-1}}$. Overall, the simulation results indicate that the vehicle respected the collision avoidance constraints and was able to safely converge back to the original lane and follow the desired trajectory on a curved path.

Finally, a parametric analysis was carried out to verify the effectiveness of the proposed framework after performing the collision avoidance manoeuvre. In this analysis, the maximum error between lateral displacement (Figure 12) from the trajectory planner and actual vehicle response was 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 increases gradually when lowering the surface friction. This figure indicates a peak deviation of 0.05 m at a 0.3 surface friction, 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.
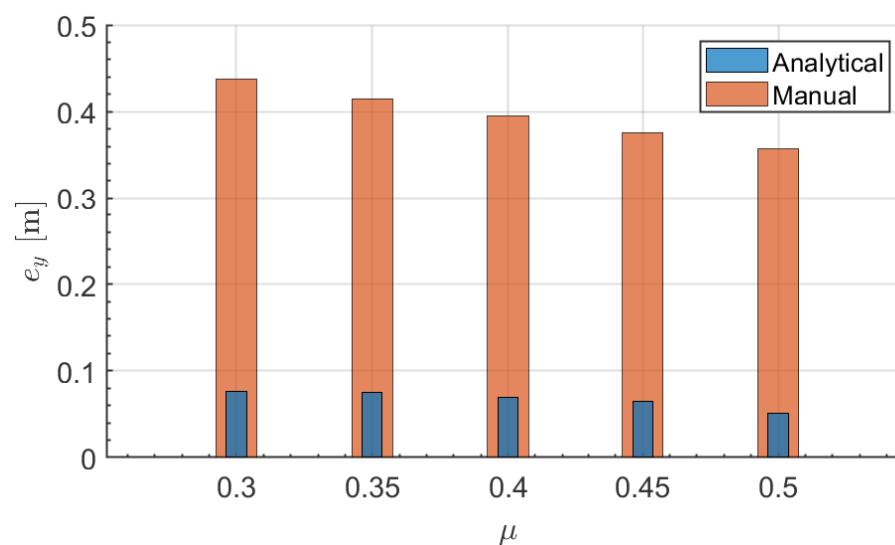

Figure 12. Lateral displacement error over different friction surfaces.

8. Conclusions

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

(1) The weighting matrix calculation was based on the gain matrix $\bar{K}$ in (4), which needed to be specified such that the closed-loop poles remain inside of the unit circle. However, the selection of the poles was chosen using Ackerman's pole placement method. This approach can be extended by means of reinforcement learning in future research. In this way, the position of the poles inside of the unit circle can be adjusted depending on different operating conditions, such as variations in surface friction. (2) A path tracking controller can be integrated as a lower-level controller to capture the vehicle dynamics to improve the stability and handling of the vehicle in extreme driving conditions.

Author Contributions: Conceptualization, S.T., K.H., S.D. and S.F.; methodology, K.H., S.T.; software, S.T.; validation, S.T., K.H., S.D.; formal analysis, S.T., S.F., K.H.; investigation, S.T.; resources, S.F.; original draft preparation, S.T.; review and editing, S.F., K.H., S.D.; project supervision and management, S.F. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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 Trans. Cybern.* **2019**, *49*, 3471–3481. [CrossRef]
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. In Proceedings of the 2012 American Control Conference, Montreal, QC, Canada, 27–29 June 2012.
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. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems, Maui, HI, USA, 4–7 November 2018.
4. Fontes, M.; Martins, A.F.; Odloak, D. An automatic tuning method for model predictive control strategies. *Proc. Am. Control Conf.* **2019**, *49*, 21602–21613. [CrossRef]
5. Julian, B.-G. Tuning for MPC Controllers. In *The Role of Population Games in the Design of Optimization-Based Controllers*; Springer: New York, NY, USA, 2017; pp. 11–16.
6. Cairano, S.; Bemporad, A. Model Predictive Control Tuning by Controller Matching. *IEEE Trans. Autom. Control* **2010**, *55*, 185–190. [CrossRef]
7. Shah, G.; Engell, S. Tuning MPC for desired closed-loop performance for SISO systems. In Proceedings of the 18th Mediterranean Conference on Control and Automation, MED'10-Conference Proceedings, Marrakech, Morocco, 23–25 June 2010; pp. 628–633.
8. Clarke, D.W.; Mohtadi, C.; Tuffs, P.S. Generalized predictive control-Part I. The basic algorithm. *Automatica* **1987**, *23*, 137–148. [CrossRef]
9. Klopot, T.; Skupin, P.; Metzger, M.; Grelewicz, P. Tuning strategy for dynamic matrix control with reduced horizons. *ISA Trans.* **2018**, *76*, 145e154. [CrossRef]
10. Shen, C.; Gonzalez, Y.; Chen, L.; Jiang, S.; Jia, X. Intelligent Parameter Tuning in Optimization-Based Iterative CT Reconstruction via Deep Reinforcement Learning. *IEEE Trans. Med. Imaging* **2018**, *37*, 1430–1439. [CrossRef] [PubMed]
11. 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 Trans. Ind. Electron.* **2019**, *67*, 5637–5646. [CrossRef]
12. Grosso, J.M.; Ocampo-Martínez, C.; Puig, V. Learning-based tuning of supervisory model predictive control for drinking water networks. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1741–1750. [CrossRef]
13. Lee, J.H.; van der Svrcek, W.Y.; Young, B.R. A tuning algorithm for model predictive controllers based on geneticalgorithms and fuzzy decision makingtle. *ISA Trans.* **2007**, *47*, 53–59.
14. Jorge, L. Garriga and Masoud Soroush. Model Predictive Control Tuning Methods: A Review. *Ind. Eng. Chem.* **2010**, *49*, 3505–3515.
15. Gutiérrez-Urquídez, R.C.; Valencia-Palomoa, G.; Rodríguez-Elias, O.M.; Trujillob, L. Systematic selection of tuning parameters for efficient predictive controllers using a multiobjective evolutionary algorithm. *Appl. Soft Comput.* **2015**, *31*, 326–338. [CrossRef]
16. Qi, Z.; Shi, Q.; Zhang, H. Inverse optimal control for discrete-time finite-horizon Linear Quadratic Regulators. *Automatica* **2019**, *110*, 6663–6668.
17. Willems, J.; Van De Voorde, H. Inverse optimal control problem for linear discrete-time systems. *Electron. Lett.* **1977**, *13*, 493. [CrossRef]

18. Luo, W.; Chu, Y.C.; Ling, K.V. Inverse optimal adaptive control for attitude tracking of spacecraft. *IEEE Trans. Autom. Control* **2005**, *50*, 1639–1654.

19. Halder, K.; Das, S.; Gupta, A. Transformation of LQR Weights for Discretization Invariant Performance of PI/PID Dominant Pole Placement Controllers. *Robotica* **2019**, *38*, 271–298. [CrossRef]

20. Kiumarsi-Khomartash, B.; Frank, L.; Naghibi-Sistani, L.; Mohammad, B.; Karimpour, A. Optimal tracking control for linear discrete-time systems using reinforcement learning. In Proceedings of the 52nd IEEE Conference on Decision and Control, Firenze, Italy, 10–13 December 2013.

21. Franze, G.; Lucia, W. A Receding Horizon Control Strategy for Autonomous Vehicles in Dynamic Environments. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 695–702. [CrossRef]

22. Gao, Y.; Gray, A.; Frasch, J.; Lin, T.; Tseng, E.; Hedrick, J.K.; Borrelli, F. Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles. In Proceedings of the 11th International Symposium on Advanced Vehicle Control, Seoul, Korea, 9–12 September 2012; pp. 1–6.

23. Karlsson, J.; Murgovski, N.; Sjöberg, J. Temporal vs. Spatial formulation of autonomous overtaking algorithms. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems, Rio de Janeiro, Brazil, 1–4 November 2016.

24. Nilsson, J.; Falcone, P.; Ali, M.; Sjöberg, J. Receding horizon maneuver generation for automated highway driving. *Control Eng. Pract.* **2015**, *41*, 124–133. [CrossRef]

25. Kong, J.; Pfeiffer, M.; Schildbach, G.; Borrelli, F. Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium, Seoul, Korea, 28 June–1 July 2015.

26. Liu, C.; Lee, S.; Varnhagen, S.; Tseng, H.E. Path planning for autonomous vehicles using model predictive control. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA, 11–14 June 2017.

27. Gutjahr, B.; Gröll, L.; Werling, M. Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1586–1595. [CrossRef]

28. Mad, D.; Philippsen, R.; Eidehall, A.; Dahl, K. On Path Planning Methods for Automotive Collision Avoidance. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, Gold Coast, QLD, Australia, 23–26 June 2013.

29. Nilsson, J.; Gao, Y.; Carvalho, A.; Borrelli, F. Manoeuvre generation and control for automated highway driving. *IIFAC Proc. Vol.* **2014**, *47*, 6301–6306. [CrossRef]

30. Gao, Y.; Gray, A.; Carvalho, A.; Tseng, H.E.; Borrelli, F. Robust nonlinear predictive control for semiautonomous ground vehicles. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014.

31. Plessen, M.; Lima, P.; Martensson, J.; Bemporad, A.; Wahlberg, B. Trajectory planning under vehicle dimension constraints using sequential linear programming. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems, Yokohama, Japan, 16–19 October 2017.

32. Scheuer, A.; Fraichard, T. Collision-free and continuous-curvature path planning for car-like robots. *Proc. IEEE Int. Conf. Robot. Autom.* **1997**, *1*, 867–873.

33. Shin, D.H.; Singh, S. Path Generation for a Robot Vehicle Using Composite Clothoid Segments. *IFAC Proc. Vol.* **1992**, *25*, 443–448. [CrossRef]

34. The Practice of Clothoid Connections on Road Interchanges. Available online: http://precismultipla.altervista.org/ESU2/chap08.htm (accessed on 20 June 2021).

35. Funke, J. Collision Avoidance Up to the Handling Limits for Autonomous Vehicles. Ph.D. Thesis, Standford University, Stanford, CA, USA, 2015.

36. Schmeitz, A.; Zegers, J.; Ploeg, J.; Alirezaei, M. Towards a generic lateral control concept for cooperative automated driving theoretical and experimental evaluation. In Proceedings of the 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017-Proceedings, Naples, Italy, 26–28 June 2017; pp. 134–139.

37. Chatzikomis, C.; Sorniotti, A.; Gruber, P.; Zanchetta, M.; Willans, D. Comparison of Path Tracking and Torque-Vectoring Controllers for Autonomous Electric Vehicles. *IEEE* **2018**, *3*, 559–570. [CrossRef]