IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 1, 2020, accepted January 3, 2021, date of publication January 19, 2021, date of current version February 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052720

MPC Based Vehicular Trajectory Planning in Structured Environment

QING SHI[1,2], JIN ZHAO[2], ABDELKADER EL KAMEL[1], (Senior Member, IEEE), AND ISMAEL LOPEZ-JUAREZ[3]

[1]CRIStAL, Centrale Lille Institut, 59650 Villeneuve-d'Ascq, France
[2]Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University, Guizhou 550025, China
[3]Center for Research and Advanced Studies (CINVESTAV), Ramos Arizpe 25900, Mexico

Corresponding author: Qing Shi (sqshiqing@gmail.com)

This work was supported in part by the National Science Foundation of China under Grant 51965008.

ABSTRACT In this paper, the hierarchical architecture of trajectory planning and control is set up for safe driving with multiple participants without collision, where both levels utilize a time-varying model predictive control methodology. Firstly, a high-level planner formulates an optimal control problem to obtain an optimal trajectory while satisfying different constraints. In particular, due to obstacles' occupation, several partition functions are generated as linear collision constraints through an optimization process in order to convexify the collision-free region into sub-regions. Secondly, the low-level controller receives the desired trajectory from the high-level planner, and then computes an appropriate steering angle to execute the planned maneuver. Both levels are formulated within the model predictive control(MPC) methodology. The strength of this framework is that it combines different constraints in each optimal control problem. Including a high-level planner ensures the feasibility of safe trajectory planning and the use of a low-level controller ensures tracking stability for safe driving, even under various collision constraints and model mismatch between system plant and predictive process model. Finally, several simulations verified the proposed framework, which was used to compute an optimal, safe trajectory over a set of static or moving obstacles and stabilize the vehicle around it.

INDEX TERMS Trajectory planning, collision avoidance, model predictive control.

I. INTRODUCTION

Numerous accidents have shown that steering control of vehicles is regarded as a solution to prevent collision with a static or moving obstacle. Generally, an active collision avoidance system includes two major objectives: i) generating a collision-free trajectory; ii) stabilizing the vehicle and keeping the desired trajectory.

Several trajectory planning algorithms have been commonly used for mobile robots in unstructured environments, such as the A^*, Rapidly-exploring Random Trees(RRT), polynomial functions or sampling based primitives [1]–[3]. Due to the driving scenario's highly structured lanes, even algorithms that incorporate the plant kinematics within the RRT searching algorithm show jerky resulting trajectories, which are not suitable for driving. Potential field-based techniques are also commonly used to generate collision-free trajectories in available, imaginary mountains of potential fields [4]. However, potential field-based methods do not incorporate vehicle dynamics and hence cannot ensure feasibility of the planned trajectory.

Geometric primitives for lane changing or simple collision avoidance situations are used in many fields of research to calculate trajectories with low computational effort [5]. These primitives defined a set of possible trajectories off-line within a short sampling time, hence trajectory planning is reduced to selecting one subset from all these feasible candidates. This method does not offer flexible actions in arbitrary, dynamic conditions.

Optimization-based methods formulate a generic framework in which vehicle dynamics, environment requirements, and additional constraints can be properly formulated as an optimal control problem [6]. The constraints can be adjustable for different requirements of the considered maneuver, i.e. safety driving and collision avoidance with each different participant. Optimal trajectory planning

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

algorithms have been proposed by researchers in different setups, varying in module complexity as well as in computational burden.

Especially, Model Predictive Control (MPC) has more advantages by solving an optimization problem online. At the same time, its internal predictive model takes vehicle kinematic characteristics into the path planning part, ensuring trajectory feasibility [7], [8]. A nonlinear bicycle model within a NMPC problem was used to generate an obstacle-free path in [9]. Inspired from above, a hierarchical optimization structure is adopted for an active collision avoidance system, including a higher-level safe trajectory planner and a lower-level stabilization controller. Both levels stem from its dependence on a number of factors, such as geometric relation to other objects, relative velocity, road setting, traffic condition, etc. [10]–[13]. Except when applied in planning layer, MPC is also popular in the path tracking control, the low-level tracking controller in this paper also utilizes a time-varying MPC to deal with stabilization. In [14], a tracking control problem was solved by combining MPC with PID feedback control of yaw rate. [15] used the outline of the preplanned road with the vehicle shape constraint in MPC process to perform tracking effectively. Liu provided symplectic pseudospectral with predictive time to realize tracking control with random disturbances [16]. Considering the computational effort and nonlinearity of the plant model, linear time-varying MPC is designed for both levels.

In complex scenarios with multiple participants in a highly structured roadway, essential challenges of the optimization-based planning methods lie on how to construct objective function and collision constraints. Their use requires not only the consideration of the nonlinearities in the vehicle dynamics, but also the nonconvexity caused by dynamic obstacles. Due to these factors, optimization approaches are limited in the simple design of objective functions or constraint design. The applicability of quadratic programs has been demonstrated in [4], [17]. Additionally, integrated trajectory planning with dynamic control by MPC scheme was solved by quadratic objective function and linear constraints. Being limited by dynamic objects, hence, their use is restricted to a trajectory re-planning method with low complexity or simple vehicle models.

Nonconvexity is another challenge appearing among trajectory optimization methods. Avoidance of one or more obstacles results in a nonconvex optimization problem, and the reason is that the collision-free area is always separated and occupied by other participants. To deal with it, different methods have been proposed by researchers such as convexification [18], change of reference frame [19], [20], creating a approximate collision avoidance constraint in [21], and a shared control in [22]. In [21], overtaking trajectories were generated by a point-mass model inside a virtual safe region located around the obstacles; moreover, it reformulated trajectory planning into a navigation problem. Although this decomposition enabled real-time implementation, the trajectories generated by the point-mass path planner were not always feasible. In [23], a space corridor was separated at each time step for safe driving while being a constraint in the proposed optimization problem.

In this paper, we present a control framework for safe, on-road driving with the presence of multi-obstacles, with the main component being a linear time-varying model predictive control (LTV-MPC) strategy for both trajectory planning in high-level and tracking control in low-level. The nonlinearity and varying velocity of the vehicle dynamic model are challenging parameters for model predictive control because they will cause model mismatch between the internal prediction model and the system plant model, which will result in unpleasant oscillation performance.

Thus, firstly, the high-level planner uses a kinematic model as the process model of LTV-MPC taking the kinematic limits and collision constraints into consideration, and, there, several linear functions are used to partition the nonconvex feasible space, so as to form the optimization problem as a quadratic program for computation reduction. Secondly, in the low-level, the proposed LTV-MPC tracking controller receives the desired profile and keeps the vehicle stabilized around this trajectory. While the high-level utilizes a kinematic vehicle model, the low-level controller is fed a desired path and velocity profile. Finally, these two levels will formulate a closed-loop control framework for the obstacle avoidance system.

The contributions of this paper lie on the following aspects: i) the nonconvex collision avoidance problem turns into a linear model-based, constrained optimization problem; ii) the nonlinearity and nonconvexity of the optimal control problem caused by obstacles can be solved by partitioning the nonconvex feasible region to be convex subregions at each predictive time step; iii) safe driving control is integrated in a hierarchical layer to guarantee position and yaw stabilization along the safe trajectory.

The remainder of this paper is as follows: In Section II, a general optimization problem for the collision avoidance system is formulated; in Section III, two process models for the model-based optimization process are introduced. One is a kinematic model for trajectory planner, the other one is a bicycle model for trajectory tracking controller; in Section IV, the optimal control problems of each layer are set up for these two different layers; in Section V, the details of the proposed space-partition functions are illustrated for the avoidance trajectory planning layer; in Section VI, several experimental tests are set up for this whole active collision avoidance system.

II. PROBLEM STATEMENT

A multi-lane, one-way scenario, which is a structured environment with easily maintained traffic rules, is set with one ego vehicle and other participants. As such, the driving task is quite straight forward: Maintaining a desired velocity, avoiding potential collision with surrounding participants and respecting the driving rules at the same time. The framework of the active collision avoidance is shown in Fig. 1.

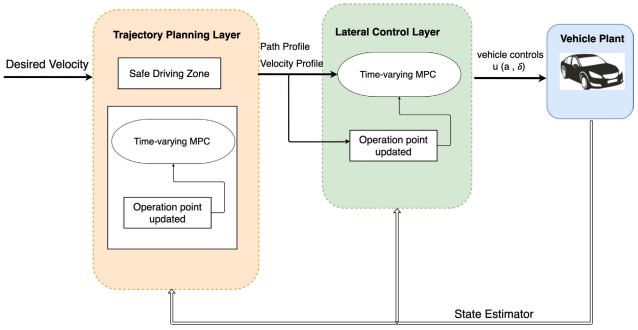


FIGURE 1. Control framework of the active collision avoidance system.

A general model-based optimization problem is:

$$\min \int_0^{T_{end}} g(x(t), u(t))dt \tag{1}$$
$$s.t.\ \dot{x}(t) = f(x(t), u(t)), x(0) = x_0 \tag{1a}$$
$$u(t) \in \mathcal{U} \tag{1b}$$
$$x(t) \in \mathcal{X} \tag{1c}$$

where $\dot{x}(t) = f(x(t), u(t)), x(0) = x_0$ is an approximation model for the real plant, which is a time variant vehicle model in normal driving conditions. Objective function $g(x(t), u(t))$ can reflect the different optimal objectives and $\mathcal{X}$ can be different feasible sets for states in planning and tracking layers.

A model-based predictive control is applied for the collision avoidance system in this paper including trajectory planning and tracking control. Generally, the main concept is using a *model* of the plant to *predict* the future evolution of the system. Based on this prediction, at each time t, a certain performance index $g(\cdot, \cdot)$ is optimized under operating constraints with respect to a sequence of future input moves. The first of such optimal moves is the control action applied to the plant at time t. At time $t+1$, a new optimization problem will be solved over a shifted prediction horizon.

III. MODEL DESIGN FOR OPTIMIZATION PROCESS

In this paper, the prediction models in the MPC scheme in both the high-level planner and the low-level tracking controller are stated in state-space function as provided by the equations (2) and (10).

A. PREDICTION PROCESS MODEL FOR HIGH-LEVEL PLANNING LAYER

To ensure real-time prediction accuracy, this process model should reflect good vehicle kinematics and dynamic constraints during the planning process. Assuming a good road surface condition in a low speed, Fig. 2 is chosen to be the process model in the planning layer, describing the kinematic characteristics of the vehicles with respect to the road coordinate frame, which is expressed as:

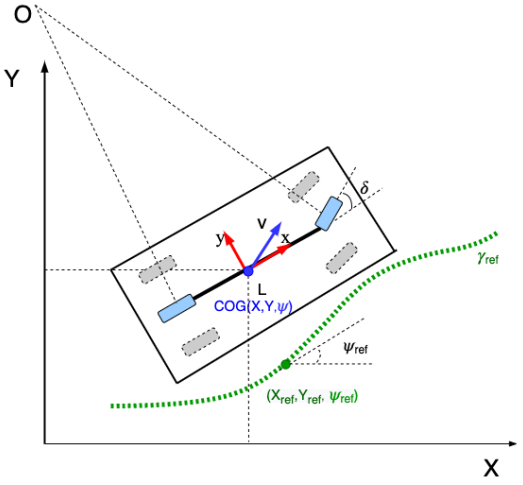


FIGURE 2. Kinematic model.

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \dfrac{v}{L}\tan(\delta) \\ \dot{v} = a \end{cases} \tag{2}$$

where L is the length of the wheelbase, state $\xi^{(1)} = [x, y, \theta, v]^T$ is the global X, Y position of the vehicle center of gravity in the road coordinate frame; and heading angle value and control input is $u^{(1)} = [a, \delta]^T$. In order to simplify the prediction model in the planning layer, it is assumed that only the front wheel can be steered. Moreover, in this layer, it is assumed that the ego car does not slip; zero slip angle means that the velocity is directed along the direction of the heading angle. Thus, any slip or road curvature is considered an external disturbance in the lower control layer.

In (2), two characters based on the plant dynamics appear to be valuable parameters: (i) nonlinearity in this system; (ii) close dependence of lateral and yaw dynamics of the vehicle on the velocity. To simplify the design of path planning, the prediction model expressed in (2) might be linearised around a nominal longitudinal speed. The resulting lateral and yaw predictions of such linear system are valid only when the longitudinal speed is constant.

$$\xi^{(1)}(t+1) = A^{(1)}\xi^{(1)}(t) + B^{(1)}u^{(1)}(t) + d_t \tag{3}$$

where d_t is the higher item in Taylor series expansion. $A^{(1)}, B^{(1)}$ are the state and control matrices for the continuous model, these being:

$$A^{(1)} = \begin{bmatrix} 0 & 0 & -vsin(\theta) & \cos(\theta) \\ 0 & 0 & vcos(\theta) & \sin(\theta) \\ 0 & 0 & 0 & tan(\delta)/L \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B^{(1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & v\tan^2\delta + 1/L \\ 1 & 0 \end{bmatrix}$$

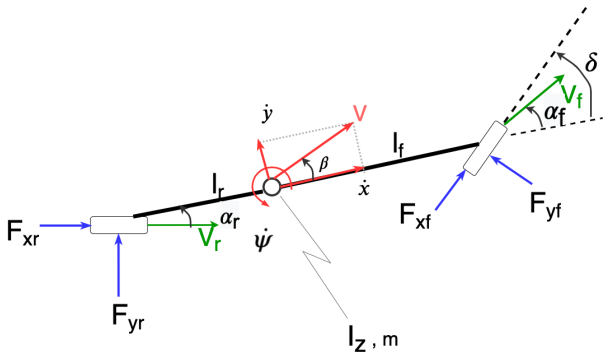


FIGURE 3. Dynamic model.

B. PREDICTION PROCESS MODEL FOR LOW-LEVEL TRACKING LAYER

The prediction model used in the tracking controller is a bicycle dynamic model combined with longitudinal speed. It assumes that the vehicle is symmetrical and that sideslip angles on the same axle are equal as depicted in Fig. 3.

The roll and pitch dynamics are neglected and slip angles are assumed to be small. Lateral dynamics are defined as indicated in equation (4):

$$\begin{cases} m\ddot{y} = -m\dot{x}\dot{\psi} + F_{fy} + F_{ry} \\ I_z\ddot{\psi} = l_f F_{fy} - l_r F_{ry} \end{cases} \tag{4}$$

where m is the vehicle mass; $\ddot{y}$ is the lateral acceleration and $\dot{x} = V_x$ is the longitudinal velocity of the vehicle; $\dot{\psi}$ and $\ddot{\psi}$ are the yaw velocity and acceleration around the z axis at the Center of Gravity (CoG) of the vehicle, respectively; $F_{fy}$and $F_{ry}$are the sum of the lateral tire forces of the front and rear axles' wheels, respectively; I_z is the inertial moment of the vehicle around its z axis; l_f and l_r are the distances from the CoG to the front or rear axles, respectively.

Considering small angle approximations($\sin\delta = 0$, $\cos\delta = 1$), lateral tire forces $F_{yf}$and F_{yr} are assumed to be proportional to the slip angle (as it is assumed that slip angles are small):

$$\begin{bmatrix} \ddot{y} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & -\dot{x} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\dot{y} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1/m & 1/m \\ l_f/I_z & -l_r/I_z \end{bmatrix} \begin{bmatrix} F_{yf} \\ F_{yr} \end{bmatrix} \tag{5}$$

Assumptions are made that lateral acceleration is less than $0.4g$, vertical tire load is constant and slip angles are no more than 4°. The lateral tire forces can be obtained:

$$\begin{aligned} F_{yf} &= 2\,C_{\alpha_f}\,\alpha_f \\ F_{yr} &= 2\,C_{\alpha_r}\,\alpha_r \end{aligned} \tag{6}$$

where $C_{\alpha_f}, C_{\alpha_r}$ are front and rear cornering stiffness constants and α_f, α_r the front and rear slip angles, respectively.

$$\begin{aligned} \alpha_f &= \delta - \frac{l_f\dot{\psi}}{\dot{x}} - \beta \\ \alpha_r &= -\beta + \frac{l_r\dot{\psi}}{\dot{x}} \\ \beta &= \tan^{-1}\frac{\dot{y}}{\dot{x}} = \frac{\dot{y}}{\dot{x}} \end{aligned} \tag{7}$$

where β is the side slip angle.

A linear vehicle state-space includes a linear tire model, small angle assumption and constant longitudinal velocity $\dot{x} = V_x$,

$$\begin{bmatrix} \ddot{y} \\ \dot{\psi} \end{bmatrix} = A_c \begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix} + \mathbf{B}_c\delta \tag{8}$$

where the state vector is $[\dot{y}, \dot{\psi}]^T$, the control action δ is the steering angle and the matrices A_c and B_c are expressed as:

$$A_c = \begin{bmatrix} -\frac{2\,C_{\alpha f} + 2\,C_{\alpha r}}{m\,\dot{x}} & -\frac{2\,l_f\,C_{\alpha f} - 2\,l_r\,C_{\alpha r}}{m\,\dot{x}} - \dot{x} \\ -\frac{2\,l_f\,C_{\alpha f} - 2\,l_r\,C_{\alpha r}}{I_z\,\dot{x}} & -\frac{2\,l_f^2\,C_{\alpha F} + 2\,l_r^2\,C_{\alpha r}}{I_z\,\dot{x}} \end{bmatrix},$$

$$B_c = \begin{bmatrix} \frac{2\,C_{\alpha f}}{m}, & \frac{2\,l_f\,C_{\alpha f}}{I_z} \end{bmatrix}^T$$

For the trajectory tracking problem, the error model with e_y, e_ψ should be integrated into vehicle dynamics. Consider a vehicle traveling with a constant longitudinal velocity V_x on a curve of constant radius R. Assume that the curve radius R is large enough so that a small steering angle assumption can be made. Then, the desired yaw rate of the vehicle is given by:

$$\dot{\psi}_d = \frac{V_x}{R} \tag{9}$$

The desired acceleration along the y axis of the vehicle is $a_{y,d} = \frac{{V_x}^2}{R} = V_x\dot{\psi}_d$, thus, e_y,we can get

$$\begin{aligned} \ddot{e}_y &= a_y - a_{yd} = (\ddot{y} + V_x\dot{\psi}) - \frac{{V_x}^2}{R} = \ddot{y} + V_x(\dot{\psi} - \dot{\psi}_d) \\ \dot{e}_y &= \dot{y} + V_x(\psi - \psi_d) \end{aligned} \tag{10}$$

From (8,9,10), the state space model for the prediction process model of tracking level can therefore be written as:

$$\xi^{(2)}(t+1) = A^{(2)}\xi^{(2)}(t) + B_1^{(2)}u^{(2)}(t) + B_2^{(2)}\rho + w \tag{11}$$

where $\xi^{(2)}$ is the state variable, $\xi^{(2)} = [y, \dot{y}, \psi, \dot{\psi}, e_y, e_\psi]^T$, and $u^{(2)} = \delta$; the curvature $\rho = 1/R$, w is a defined disturbance defined as Gaussian noise $w\,N(0, \sigma^2)$, $A^{(2)}$, $B_1^{(2)}$, and $B_2^{(2)}$, as shown at the bottom of the page.

In the next section, we make use of the model (11) to formulate a model predictive control problem.

IV. TIME-VARYING MODEL PREDICTIVE CONTROL DESIGN

The overall framework of a collision avoidance system is modelled through a two-level hierarchical architecture as shown in Fig. 1. In this section, two optimal control problems (OCPs) are set up based on time-varying model predictive control, among which, the first OCP is applied in the high-level planner for a feasible trajectory generation while the second one is responsible for the low-level tracking mission. The major differences lie in the prediction model, which means that the cost function and constraints are formulated differently. The core advantage of time-varying MPC is the ability of updating the prediction model as time evolves. Once updated, the model parameter and operating conditions

remain constant over the prediction horizon. An additional condition is for the sample time T_s to be constant and identical to the MPC control interval.

Optimization methods build on a generic framework based on criteria in which relevant information about the environment and vehicle dynamics, as well as desired driving properties, can be properly formulated. For each problem, different criteria in this framework can be featured with elements of different complexity or planning requirement, i.e. in (1), the cost function $g(\cdot,\cdot)$ can be different depending on the associated cost function and constraints (1a-1c). Lastly, the corresponding cost function for the high-level optimal control problem is analyzed.

A. COST FUNCTION FOR HIGH-LEVEL PLANNING

The high-level planner utilizes a kinematic model as (3) and certain collision constraint as (31) in order to compute a collision-free maneuver, where these constraints are some linear functions and explained afterwards. Through the following optimal control problem, the resulting maneuver $[x_{ref}(t), y_{ref}(t), \psi_{ref}(t), t]$ will be fed to the low-level controller.

1) OPTIMAL CONTROL PROBLEM 1

Safe trajectory planning problem, the objective is to minimize $J^{(1)}(\cdot,\cdot)$ to obtain an optimal state trajectory $\xi^{(1)}$.

$$
\begin{aligned}
OCP1: \min_{U_t,\xi_{t+1,t},\xi_{t+2,t},\ldots,\xi_{t+N,t}} & J^{(1)}(\cdot,\cdot) \\
&= P(\xi^{(1)}_{k+N,t}) + l(\xi^{(1)}_{k,t}, r^{(1)}) + l(\triangle u^{(1)}_{k,t}) \\
&= \| \xi^{(1)}_{k+N,t} - r^{(1)}_{N+t} \|_{P_1} \\
&+ \sum_{k=0}^{t+N-1} \| \xi^{(1)}_{k,t} - r^{(1)} \|_{Q_1} + \| \triangle u^{(1)}_{k,t} \|_{R_1} \qquad (12) \\
s.t.\ & \xi^{(1)}_{k+1,t} = A^{(1)}\xi^{(1)}_{k,t} + B^{(1)}u^{(1)}_{k,t} + d_t, \\
& k = t,\ldots,t+N-1, \qquad (12a) \\
& u^{(1)}_{k,t} \in \mathcal{U},\ k = t,\ldots,t+N-1, \qquad (12b) \\
& \xi^{(1)}_{k,t} \in \mathcal{X}_{free} = \mathcal{R}_{road} \cap \mathcal{R}^{(e)}(t), \\
& k = t,\ldots,t+N-1, \qquad (12c) \\
& \xi^{(1)}_{k+N,t} \in \mathcal{X}_f, \qquad (12d) \\
& \xi^{(1)}_{t,t} = \xi^{(1)}(t) \qquad (12e)
\end{aligned}
$$

where the reference for the process model is $r^{(1)} = [y_{target}, \psi_{target}, v_{target}]$; ego $V^{(e)}$ is expected to traverse at a constant desired longitudinal velocity v_{target} while maintaining its lane position; the terminal cost function $P(\xi^{(1)}_{k+N,t})$ is the penalty on the terminal state in the last predictive step $t+N$; $\mathcal{X}_{free}$ is the collision region for ego, especially in this paper, $\mathcal{X}_{free}$: $\mathcal{R}_{road} \cap \mathcal{R}^{(e)}(t)$, and it is defined by some time-varying functions and illustrated in next section. $\mathcal{X}_f$ denotes a terminal set for the prediction states of ego. N stands for the prediction time steps, and for simplicity in expression; here $N = N_p = N_c$. $d_{k,t}$ is the neglected higher item of Taylor Expansion.

Thus, by minimizing these quadratic cost functions, $P(\cdot) + l(\cdot,\cdot) + l(\cdot)$, we can get an optimal value $\triangle u^{(1)*}$ at every time interval, then applying it to the process model and "steer" the system model into the next time step so that state sequence $\xi^{(1)}_{k,t} = [x_{ref}(t), y_{ref}(t), \psi_{ref}(t), t]$ with time made up as a desired trajectory reference fed to low-level controller.

B. COST FUNCTION FOR LOW-LEVEL OPTIMAL CONTROL PROBLEM

In this subsection, we consider the stabilization problem of the lateral and yaw angle of a nonlinear vehicle along a desired path. In each optimization process, we seek to minimize position deviation and yaw angle error $[e_y, e_\psi]$ with respect to the reference of $[y_{ref}, \psi_{ref}]$ in a road-aligned coordinate. Applying successive linearization is a way of dealing with the nonlinearity of the vehicle plant. This process control method can date back to the early 17*th* century. Recent applications of time-varying or parameter-varying model predictive control schemes are formed in [24], [25].

Research on trajectory tracking control for vehicle nonlinear discrete systems was shown in [26], and, there, an additional boundary as constraint in the tracking control process was used, a quadratic function including control value and predicted states in prediction steps. Taking the idea from these papers, we adopt some certain convex constraints to

$$
A^{(2)} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & -\frac{2C_{af}+2C_{ar}}{mV_x} & 0 & -V_x - \frac{2C_{af}l_f - 2C_{ar}l_r}{mV_x} & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & -\frac{2l_fC_{af} - 2l_rC_{ar}}{I_zV_x} & 0 & -\frac{2{l_f}^2C_{af} + 2{l_r}^2C_{ar}}{I_zV_x} & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & V_x \\
0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix},
$$

$$
B_1^{(2)} = \begin{bmatrix} 0 & \frac{2C_{af}}{m} & 0 & \frac{2C_{af}l_f}{I_z} & 0 & 0 \end{bmatrix}^T,
$$

$$
B_2^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -Vx \end{bmatrix}^T.
$$

guarantee the lateral stabilization for the collision avoidance system.

1) OPTIMAL CONTROL PROBLEM 2

The objective is to control the state of a discrete-time nonlinear system $\xi^{(2)}(t)$ to the origin $\xi_e = 0, u_e = 0$. The requirements are the following: i) the nonlinear dynamics model (10) is to be used as process model; ii) a desired trajectory from planning layer.

$$
\begin{aligned}
OCP2: \min_{U_t,\xi_{t+1,t},\xi_{t+2,t},\ldots,\xi_{t+N,t}} & J^{(2)}(\cdot) \\
&= P(\xi^{(2)}_{k+N,t}) + l(\xi^{(2)}_{k,t}, r^{(2)}) + l(\triangle u^{(2)}_{k-1,t}) \\
&= \sum_{i=1}^{t+N} \|\eta^{(2)}_{k,t} - \eta^{(1)}_{k,t}\|_{Q_2} + \|\triangle u^{(2)}_{k,t}\|_{R_2} \qquad (13) \\
& s.t.\ \xi^{(2)}_{k+1,t} = A^{(2)}_{k,t}\xi_{k,t} + B^{(2)}_{k,t}u_{k,t} + \triangle d_{k,t}, \\
& \quad k = t, \ldots, t+N-1 \qquad (13a) \\
& \quad \xi^{(2)}_{k,t} \in \mathcal{X}, k = t, \ldots, t+N \qquad (13b) \\
& \quad u^{(2)}_{k,t} \in \mathcal{U}, k = t, \ldots, t+N-1 \qquad (13c) \\
& \quad \xi^{(2)}_{k+N,t} \in \mathcal{X}_f \qquad (13d) \\
& \quad \xi^{(2)}_{t,t} = \xi^{(2)}(t) \qquad (13e) \\
& \quad l(\xi_{t+N_p-1,t}, u_{t+N-1,t}) \\
& \quad < l(\xi_{t-1,t-1}, u_{t-1,t-1}) + \epsilon) \qquad (13f)
\end{aligned}
$$

where, $U_t = [u_{t,t}, u_{t+1,t}, \ldots, u_{t+N_p-1,t}]$ is the optimization sequence calculated at time t; trajectory $\xi_{k,t}$ is the predictive state of the process model, assumed to be measurable and obtainable by applying the optimal sequence $U_t = [u_{t,t}, u_{t+1,t}, \ldots, u_{t+N-1,t}]$ to system 13(d). N_p and N_c denote the output prediction horizon and the control horizon, respectively; the reference signal $\eta^{(1)}_{k,t}$ represents the state outputs of high-level process model, representing the desired trajectory from the high-level planner.

In the above cost function $J^{(2)}(\cdot) = J^{(2)}(\xi^{(2)}, u^{(2)}) : \mathcal{R}^{Nm} \times \mathcal{R}^{Nn} \to \mathcal{R}^{+}$, Q_2, R_2 are weighting matrices of appropriate dimensions. The first summand with Q_2 reflects the desired performance on reference tracking, the second summand with R_2 is a measure of the steering effort. If constraints (13b) -(13e) are linear, this whole optimal control problem could be solved with Linear Programming (LP) or Quadrating Programming (QP) solvers. Additionally, the nonlinear system plant is a 7 degree-of-freedom vehicle dynamic model, which will be introduced in section 5. In (13g), by looking at the tire characteristics, shown in [25], [27], [28], it is clearly stated that the front slip angle should be constrained in a range to ensure stability. Because state transition matrices are space-varying for simplicity, linear time-varying MPC (LTV-MPC) is used to handle nonlinearity with linearization and discretization around reference trajectories; it formulates a trajectory-tracking problem in the form of a quadratic program (QP) to stabilize the vehicle along the trajectory.

Differing from the high-level optimization process, this level has another formulation in one OCP considering the penalty on lateral position deviation and relative yaw angle within prediction steps N. The controller input includes a desired path point and velocity value from the planning layer; the control objective is then to get an optimal command: Steering command to system plant.

C. FEASIBILITY AND STABILITY ANALYSIS

For the avoidance trajectory planning method, the feasibility would be guaranteed in the MPC scheme. Firstly, the controlled system is a discrete nonlinear system:

$$\xi(t+1) = f(\xi(t), u(t)) \qquad (14)$$

where state $\xi(t) \in \mathcal{R}^n$, input $u(t) \in \mathcal{R}^m$. Then, the nominal prediction model is the same with the additional state and input constraint:

$$\xi(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U}$$

Assumption 1: There exists a function Φ that bounds all the open-loop states in a set $\mathcal{X}$ with a given input u_t such that for all $\xi_t \in \mathcal{X}_t$, $u_t \in \mathcal{U}_t$, it has a terminal cost:

$$f(\xi_t, u_t) \in \mathcal{X}_{t+1} := \Phi(\mathcal{X}_t, \mathcal{U}_t) \qquad (15)$$

Using the terminal cost V_f and stage cost l to define an open-loop cost function:

$$J_{N_p}(\xi, u) = \sum_{k=0}^{N_p-1} l(\xi_{k,t}, u_{k,t}) + V_f(\xi_{N_p,t}) \qquad (16)$$

Assumption 2: There exists a terminal set $\mathcal{X}_f \subset \mathcal{R}^n$, with a terminal cost and a terminal control $\kappa_f(\cdot) : 2^{\mathcal{R}^n} \to \mathcal{R}^m$, such that:

$$
\begin{aligned}
&\Phi(\xi, \kappa_f(\xi), u) \subset \mathcal{X}_f, \\
&V_f(\Phi(\xi, \kappa_f(\xi)) \le V_f(\xi) - l(\xi, \kappa_f(\xi)), \qquad (17)
\end{aligned}
$$

Thus, the resulted problem OCP1 turns out to be:

$$
\begin{aligned}
OCP1: \min_{U_t,\xi_{t+1,t},\xi_{t+2,t},\ldots,\xi_{t+N,t}} & J^{(1)}(\cdot,\cdot) \\
& s.t.\ \xi^{(1)}_{k+1,t} = \Phi(\xi^{(1)}_{k,t}, u^{(1)}_{k,t}, \\
& \quad k = t, \ldots, t+N-1, \\
& \quad u^{(1)}_{k,t} \in \mathcal{U}, \quad k = t, \ldots, t+N-1, \\
& \quad \xi^{(1)}_{k,t} \in \mathcal{X}_{free}, \quad k = t, \ldots, t+N-1, \\
& \quad \xi^{(1)}_{k+N,t} \in \mathcal{X}_f, \\
& \quad \xi^{(1)}_{t,t} = \xi^{(1)}(t) \qquad (18)
\end{aligned}
$$

The solution to (18) is the optimal trajectory $u^*_{\cdot,t}$, and then the closed-loop system is given by:

$$\xi(t+1) = f_{cl}(\xi(t), u(t)), u(t) = u^*_{0,t} \qquad (19)$$

Theorem 1: Let Assumption 1 and 2 hold, supposing that the problem (18) is feasible at initial time $t = 0$, then the problem (18) is recursive feasible and the constraints are satisfied.

Proof: It is a standard result in constrained MPC, as shown in [29]. □

For this hierarchical collision avoidance system, stabilization analysis is mainly focused on the lateral position and relative yaw angle.

Definition 2: $V(t,\xi(t))$ denotes the value of the cost function $J_{N_p}(\xi_t, U_t)$ in (13) at time t with initial state $\xi(t)$, $U(t)$. OCP 2 with constraints are defined as $\mathcal{P}_t(\xi(t))$.

Assumption 3: In $\mathcal{P}_t(\xi(t))$, matrices $A_{k,t}, B_{k,t}$ are the time-varying matrices, but it will keep being invariant during each prediction horizon in the optimization process, considering reduced computation. $A_{k,t} = A_t, B_{k,t} = B_t$, $k = t, t+1, \ldots, t+N_p-1$

Assumption 4: Consider the control laws (13a)-(13f) with Assumption 1, and assuming the feasibility of problem $\mathcal{P}_t(\xi(t))$ at $t = 0$, then it would be feasible for all $t > 0$.

Definition 3: A function $\gamma(\cdot) : \mathbb{R}^+ \to \mathbb{R}^+$ can be defined as class $\mathcal{K}$ if $\gamma(\cdot)$ is continuous and strictly increasing with $\gamma(0) = 0$. [29]

Definition 4: Value function $V(t,\xi) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^+$ is to be a locally positive definite function, if it has a relation with $\gamma(\cdot)$ [29]:

$$V(t,\xi) \geq \gamma(|\ \xi\ |), \quad \text{with } V(t,0) = 0, \ \ \forall \xi \in B_h, t \geq 0 \tag{20}$$

where $B_h \subset \mathbb{R}^n$ is a ball centered in the origin with radius h.

Definition 5: A scalar function $V(t,\xi)$ is decrescent in the set $\mathcal{X} \in \mathbb{R}^n$ if there is a class $\mathcal{K}$ function $\beta(\cdot)$ that satisfies:

$$V(t,\xi) \leq \beta(|\ \xi\ |) \tag{21}$$

Now, using the standard Lyapunov stability theorem in [30]:

Lemma 6: Considering the following nonlinear discrete time system, with $\xi \in \mathbb{R}^n, f(t,0) = 0, f(\cdot,\cdot) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$,

$$\xi(t+1) = f(t,\xi(t)), with \xi(0) \tag{22}$$

If a continuous function $V(t,\xi)$ is a positive, definite and decrescent function and the difference $V(t-1,\xi(t-1)) - V(t,\xi(t)$ is a locally positive definite function, then the origin of the system can be asymptotically stable.

To derive the stability condition, some assumptions are proposed to restrict the system and optimization problem as follows:

Assumption 5: For system (22), we assume that $\forall t \in \mathbb{R}^+, \forall \xi \in \mathbb{R}^n$. The state matrix and input matrix are bounded when being linearized at the nominal system point$(\xi(t), u(t))$:

$$\frac{\partial f(\xi,u)}{\partial \xi}\ |_{\xi(t),u(t)}, \frac{\partial f(\xi,u)}{\partial u}\ |_{\xi(t),u(t)} \tag{23}$$

The stage cost function $l(\xi,u)$ and the terminal cost function $P(\xi)$ have the formation as:

$$\begin{aligned} l(\xi_{k,t}, u_{k,t}) &= \xi_{k,t}^T Q \xi_{k,t} + u_{k,t}^T R u_{k,t} = \| \xi_{k,t} \|_Q^2 + \| u_{k,t} \|_R^2, \\ P(\xi_{k+N_p,t}) &= \xi_{k+N_p,t}^T Q \xi_{k+N_p,t} = \| \xi_{k+N_p,t} \|_Q^2 \end{aligned} \tag{24}$$

where $Q \in \mathbb{R}^{n\times n}$, $P \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$ are definite positive matrices.

Lemma 7: Within Assumption 1, 2, 3, the value function $V(t,\xi(t))$ defined in Definition IV.1 has an upper bound, and it is possible for it to have a relation with a positive definite function $\beta(\cdot)$: $V(t,\xi(t)) \leq \beta(\cdot)$ for all $\xi \in \mathcal{X}$. Moreover, value function is a decrescent function in set $\mathcal{X}$.

To continue, we follow the constraints definition in [31].

Lemma 8: Considering problem $\mathcal{P}_t(\xi(t))$ and control laws (13a)-(13f), with Assumption 1, 2, 3, a difference of the value function is $\triangle V(\cdot,\xi(\cdot)) = V(t-1,\xi(t-1)) - V(t,\xi(t))$. If the following function is satisfied, then $\triangle V(\cdot,\xi(\cdot))$ can be defined as a locally positive definite function, with $t > 0, \xi \in \mathcal{X}$:

$$\begin{aligned} &l(\xi_{t-1,t-1}, u_{t-1,t-1}) - l(\xi_{t+N_p-1,t}, u_{t+N_p-1,t}) \\ &\qquad > \sum_{i=0}^{N_p-2} \| \xi_{t+i,t} - \xi_{t+i,t-1} \|_Q \end{aligned} \tag{25}$$

where $[\xi_{t,t}, \xi_{t+1,t}, \ldots, \xi_{t+N_p-1,t}]$ is the predicted state obtained by feeding the input sequence $[u_{t,t-1}, \ldots, u_{t+N_p-2,t-1}]$ to process model (8).

Proof: For optimization problem $\mathcal{P}_t(\xi(t))$ at time t, recall the value function is:

$$V(t,\xi) = \sum_{i=0}^{N_p-1} (\| \xi_{t+i,t} \|_Q + \| u_{t+i,t} \|_R) + \| \xi_{t+N_p,t} \|_P \tag{26}$$

where at time t, the optimal solution to problem $\mathcal{P}_t(\xi(t))$ is

$$U_t^* = [u_{t,t}, u_{t+1,t} \ldots, u_{t+N_p-2,t}, u_{t+N_p-1,t}],$$

with the corresponding state being $[\xi_{t+1,t}, \xi_{t+2,t} \ldots, \xi_{t+N_p,t}]$, and the time at $t-1$, the solution of problem $\mathcal{P}_{t-1}(\xi(t-1))$ is

$$U_{t-1}^* = [u_{t-1,t-1}, u_{t,t-1}, \ldots, u_{t+N_p-1,t-1}, u_{t+N_p-2,t-1}],$$

due to rolling horizon, the state of the system plant is $\xi(t) = f(\xi(t-1), u_{t-1,t-1}))$.

Comparing two sequences U_{t-1}^* and U_t^*, a new sequence $\tilde{U}_t$ can be made by discarding the first component $u_{t-1,t-1}$ of U_{t-1}^*, then adding a new $\triangle u_{t,t}$ to it when time step is pushed forward from $t-1$ to t.

$$\tilde{U}_t = [u_{t-1,t-1}, \ldots, u_{t+N_p-2,t-1}, u_{t+N_p-1,t}]$$

from Assumption 2, we know that it is a feasible sequence for problem $\mathcal{P}_t(\xi(t))$. From above, $V(t,\xi(t))$ can be revised as:

$$\begin{aligned} V(t,\xi) = \sum_{i=0}^{N_p-1} \| \xi_{t+i,t} \|_Q + \sum_{i=0}^{N_p-1} \| u_{t-1+i,t-1} \|_R \\ + \| u_{t-1+N_p,t} \|_R + \| \xi_{t+N_p,t} \|_P \end{aligned} \tag{27}$$

Thus, using (27), we can obtain:

$$\begin{aligned} &\triangle V(t,\xi(t)) \\ &= V(t-1,\xi(t-1)) - V(t,\xi(t)) \\ &= \sum_{i=0}^{N_p-1} \| \xi_{t-1+i,t-1} \|_Q + \sum_{i=0}^{N_p-1} \| u_{t-1+i,t-1} \|_R \end{aligned}$$

$$
\begin{aligned}
&+ \| \xi_{t+N_p-1,t-1} \|_P - \Big(\sum_{i=0}^{N_p-1} \| \xi_{t+i,t} \|_Q \\
&+ \sum_{i=1}^{N_p-1} \| u_{t-1+i,t-1} \|_R + \| u_{t-1+N_p,t} \|_R + \| \xi_{t+N_p,t} \|_P \Big) \\
&= \sum_{i=0}^{N_p-2} \Big(\| \xi_{t+i,t-1} \|_Q - \| \xi_{t+i,t} \|_Q \Big) + \| \xi_{t-1,t-1} \|_Q \\
&- \| \xi_{t+N_p-1,t} \|_Q + \| u_{t-1,t-1} \|_R - \| u_{t+N_p-1,t} \|_R
\end{aligned} \tag{28}
$$

at time t,

$$
\begin{aligned}
&\because \| x_1 \| - \| x_2 \| \leq \| x_1 - x_2 \|, \\
&\sum_{i=0}^{N_p-2} \Big(\| \xi_{t+i,t-1} \|_Q - \| \xi_{t+i,t} \|_Q \Big) \\
&\leq \sum_{i=0}^{N_p-2} \| \xi_{t+i,t-1} - \xi_{t+i,t} \|_Q
\end{aligned}
$$

Thus, (28) turns into:

$$
\begin{aligned}
\Delta V(t, \xi(t)) \geq\ & l(\xi_{t-1,t-1}, u_{t-1,t-1}) - l(\xi_{t+N_p-1,t}, u_{t+N_p-1,t}) \\
& - \sum_{i=0}^{N_p-2} \| \xi_{t+i,t-1} - \xi_{t+i,t} \|_Q
\end{aligned} \tag{29}
$$

In the above inequality (29), if Lemma IV.3 (25) were satisfied, the right hand side would be bounded as: $l(\xi_{t-1,t-1}, u_{t-1,t-1}) - l(\xi_{t+N_p-1,t}, u_{t+N_p-1,t}) - \sum_{i=0}^{N_p-2} \| \xi_{t+i,t-1} - \xi_{t+i,t} \|_Q > 0$, thus $\Delta V(t, \xi(t)) \geq 0$ is proved to be a locally positive definite function. □

Theorem 9: In system (22)*, problem* $\mathcal{P}_t(\xi(t))$ *with control laws (13a)-(13f) and constraints* (29) *are satisfied and terminal set* $\mathcal{X}_f = 0$*: Holding with Assumptions 1,2,3, results in the origin of the closed-loop system are uniformly, locally and asymptotically stable.*

Assumption 6: In a closed-loop system (22) *and problem* $\mathcal{P}_t(\xi(t))$*, if problem* $\mathcal{P}(0, \xi)$ *with terminal constraint* $\mathcal{X}_f = 0$ *is feasible, then* $\forall t > 0$*, the problem* $\mathcal{P}(t, \xi(t))$ *with the same constraint would always be feasible.*

Proof: Recall **Assumptions 3,4,5,6**, a closed-loop system (22) with constraint in (29) and $\mathcal{X}_f = 0$, thus the associated Lyapunov function $V(\cdot, \xi(\cdot))$ would be decrescent, and it would be a positive function from Lemma IV.2. Furthermore, difference function $\Delta V(\cdot, \xi(\cdot))$ has been proved to be a positive definite function with a lower bound 0 from Lemma IV.3 (25). Using Lemma IV.1, we can conclude that the origin of the system (22) is asymptotically stable. □

V. CONSTRAINTS FOR COLLISION AVOIDANCE

The basic objective of an active avoidance system is to ensure that an ego car travels inside the lane boundaries and avoiding all obstacles. The most challenging point is solving collision constraints in this optimization-based trajectory planning problem 1 (12), that is, how to *convexify* the feasible region outside the obstacles. Furthermore, it is much more complicated because of a time-varying environment (with dynamically moving objects) requiring frequent replanning. Next, we propose several time-varying functions as a solution.

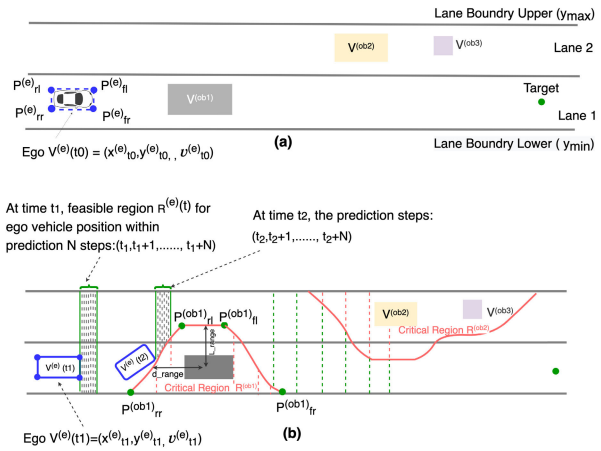


FIGURE 4. Geometric illustration with some objects $V^{(ob)}$. The feasible region for ego vehicle $V^{(e)}$ in the structured environment (a) can be represented as green lines in (b) at each time step t_i.

By restricting the ego vehicle to stay outside the critical regions $R^{(ob)}$, illustrated in Fig. 4(b), that are separated by red lines around each obstacle $V^{(ob)}$. Intuitively, the area outside these critical regions is non-convex and shown as vertical green lines making up a feasible "driving corridor". In order to turn the feasible region to be convex sets, we use the relation between $V^{(e)}$ and $V^{(ob)}$ to define some certain linear inequality constraints for ego vehicle's states, namely $A(t)\xi \leq B(t)$ and $[y_{min}, y_{max}]$ to "cut" and re-define some subsets at each time step. Therefore, the resulting $A(t)\xi \leq B(t)$ will be integrated in $\mathcal{X}_{free}$ and be used in the optimization process of the high-level MPC planner, finally, details will be the following.

A. CONSTRAINTS FOR ACTUATOR LIMITS

The input constraints usually come from actuator limitations, defined by variable increment:

$$
\Delta\mathbb{U} := \{(\Delta a, \Delta\delta) \in \mathbb{R}^2\} = \begin{cases} -0.10 \leq \Delta\delta \leq 0.10(rad) \\ -0.2 \leq \Delta a \leq 0.2(m/s^2) \end{cases} \tag{30}
$$

B. CONSTRAINTS FOR VEHICLE COLLISION

Firstly, considering a scenario in Fig. 4, there are two straight lanes with a static obstacle on the right lane with width W_{ob} and length L_{ob}; the relative position of the critical region for each obstacle is represented by two points: front right point and front left point P_{fl}^{obj}, rear left point P_{rl}^{obj}, rear right point. A critical area is defined around each obstacle with red lines, and the feasible region is outside these critical areas, which is, namely, a safe driving permission for ego vehicle. All road participants like pedestrians, cyclists or vehicles can be enlarged by a rectangle. Thus, the feasible

region is irregular. The proposed planning method considers potential collision obstacles and lane boundary information to plan a collision-free trajectory during the maneuver without exceeding the road boundaries.

1) B1. ROAD BOUNDARY CONSTRAINTS

Basically, planning methods aim to guarantee a regular driving, i.e. not driving out of the road as shown in Fig. 4(a); a structured environment has the maximum and minimum lane boundary $y_{max} = lane_{up}, y_{min} = lane_{low}$. To keep all corners of $V^{(e)}$ within these boundaries, i.e. at time t_i, $V^{(e)}$ configuration of C.O.G. in road frame has $(x_{t_i}^{(e)}, y_{t_i}^{(e)}, v_{t_i}^{(e)})$ and it can be seen as a blue rectangle with one front right point $P_{fr}^{e} = (x_{fr}^{(e)}, y_{fr}^{(e)})$, one front left point $P_{fl}^{(e)} = (x_{fl}^{(e)}, y_{fl}^{(e)})$, one rear right point $P_{rr}^{e} = (x_{rr}^{(e)}, y_{rr}^{(e)})$, and one rear left point $P_{rl}^{e} = (x_{rl}^{(e)}, y_{rl}^{(e)})$. The following inequality functions is to ensure every:

$$\xi(t) \in \mathcal{R}_{road} \Rightarrow \begin{cases} \min(y_{rr}^{(e)}, y_{fr}^{(e)}) & \geq y_{min} \\ \min(y_{rl}^{(e)}, y_{rl}^{(e)}) & \leq y_{max} \end{cases} \tag{31}$$

where

$$\begin{aligned} y_{rr}^{(e)} &= y^{(e)} - l_f \sin(\theta) - W_e/2\cos(\theta) \\ y_{fr}^{(e)} &= y^{(e)} + l_f \sin(\theta) - W_e/2\cos(\theta) \\ y_{rl}^{(e)} &= y^{(e)} + l_f \sin(\theta) + W_e/2\cos(\theta) \\ y_{fl}^{(e)} &= y^{(e)} - l_f \sin(\theta) + W_e/2\cos(\theta) \end{aligned}$$

where $l_f + l_r = L_e$, W_e are the car length and width for ego vehicle $V^{(e)}$.

2) B2. SPACE-PARTITION FUNCTIONS FOR COLLISION CONSTRAINTS

The way to define a *safe distance* between vehicles and obstacles differs between researchers, but circular and polygonal polytopes are widely used to describe the critical regions occupied by obstacles. Outside these regions, we can find that the feasible region for ego vehicle is non-convex, i.e., some certain pairs of position points in the feasible regions cannot be connected by a straight line segment. This non-convexity raises a big challenge for the optimization process as (12).

To deal with it, a linear function $A(t)\xi = B(t)$ is set up as constraints for optimization process so as to "convexify" this irregular feasible region $\mathcal{R}^{(e)}$ as shown in Fig. 4(b). $\mathcal{R}^{(e)}$ is a safe region for ego vehicle's lateral position $y^{(e)}(t)$ at time t_1 and is marked out with a vertical green line combined with several N dashed gray lines (as "predicted future positions").

For example, with time evolved from t_1 to t_3, $\mathcal{R}^{(e)}$ keeps changing as $\mathcal{R}^{(e)}(t_1)$, $\mathcal{R}^{(e)}(t_2)$ and $\mathcal{R}^{(e)}(t_3)$ as shown in Fig. 5-Fig. 7, respectively; the details are explained in Algorithm (1). $\mathcal{R}^{(e)}$ is calculated by the function $A(t)\xi = B(t)$ in Algorithm(1) mainly using the geometric relations between $V^{(e)}$ and $V^{(ob1)}$. $R^{(ob)}$ is a critical region comprising four points: the front right point $P_{fr}^{(ob1)} = (x_{fr}^{(ob1)}, y_{fr}^{(ob1)})$, front left point $P_{fl}^{(ob1)} = (x_{fl}^{(ob1)}, y_{fl}^{(ob1)})$, rear right $P_{rr}^{(ob1)} = (x_{rr}^{(ob1)}, y_{rr}^{(ob1)})$, rear left point $P_{rl}^{ob1} = (x_{rl}^{(ob1)}, y_{rl}^{ob1} + W_{lane})$:

$$\xi(t) \in \mathcal{R}^{(e)}(t) \Rightarrow A(t)\xi(t) \leq B(t) \tag{32}$$

where $A(t)$ and $B(t)$ are the time-varying matrices for region $\mathcal{R}^{(e)}(t)$, which keeps updating as the ego vehicle drives.

$$A(t) = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad B(t) = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

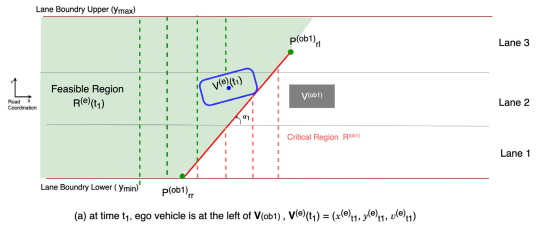
Algorithm 1 Updating Constraints

Input:
$V^{(e)}(t) = (x^{(e)}(t), y^{(e)}(t), v^{(e)}(t))$,
$V^{(ob1)} = (x^{(ob1)}, y^{(ob1)}, v^{(ob1)})$, d_{range};
Output:
$A(t) = (a_1^T, a_2^T, a_3^T)$, $B(t) = (b_1, b_2, b_3)^T$;
while $((x^{(e)} - x^{(ob1)})^2 + (y^{(e)} - y^{(ob1)})^2)^{1/2} \leq d_{range}$ **do**
  initialization $\tan\alpha = (y^{(e)} - y_{rleft})/(x^{(e)} - x_{rleft})$
  if $x^{(e)} \leq x_{rleft}$ **then**
    if $y^{(e)} > y_{rleft}$ **then**
      ego $V^{(e)}$ is not within the same lane as $V^{(ob1)}$:
      $\tan\alpha = 0 = \tan\alpha_1$,intercept=$y_{min}$
    else
      ego enters in $\mathcal{R}^{(e)}(t_1)$ as shown in Fig.5,
      $\tan\alpha_1 = \tan\alpha$,intercept $= \tan\alpha_1 * x_{rleft} - y_{rleft}$
    end if
  else if $x_{ego,t} > x_{rleft}, x_{ego,t} < x_{fleft}$ **then**
    ego stays in $\mathcal{R}^{(e)}(t_2)$ as shown in Fig.6,
    $\tan\alpha_1 = 0$,intercept $= y_{rleft}$
  else
    ego in $\mathcal{R}^{(e)}(t_3)$ as Fig.7,
    $\tan\alpha_1 = 0$,intercept $= -y_{min}$
  end if
  Generate matrices
  $A(t) = [0, 1, 0, 0; 0, -1, 0, 0; \tan\alpha_1, -1, 0, 0]$,
  $B(t) = [y_{max}; y_{min}; -intercept]$
end while
return $A(t), B(t)$

In Fig. 5, once the ego vehicle $V^{(e)}$ detects an obstacle within d_{range}, $V^{(e)}$ checks the drivable area from time t_1 to t_N. At t_1, obstacle $V^{(ob1)}$ separates the whole feasible area with red line into two parts, and it becomes the critical area $R^(ob1)$, thus, the left side of the obstacle has left more free space for the ego one to go through, as the green area $\mathcal{R}^{(e)}(t_1)$ shown in Fig. 5.

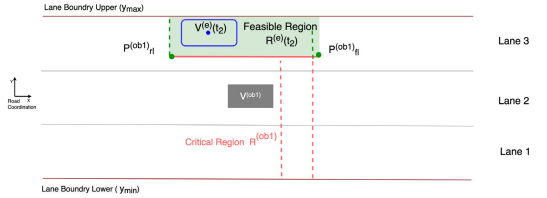
Integrating (31, 32), the collision-free constraint could be obtained eventually:

$$\begin{aligned} \mathcal{R}^{(e)}(t) &= \mathcal{R}^{(e)}(t_1) \cup \mathcal{R}^{(e)}(t_2) \cup \mathcal{R}^{(e)}(t_3), \\ \mathcal{X}_{free} &:= \mathcal{R}_{lane} \cap \mathcal{R}^{(e)}(t) \end{aligned} \tag{33}$$



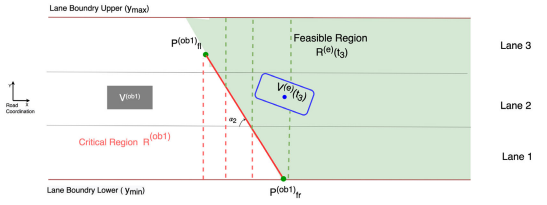
(a) at time t_1, ego vehicle is at the left of $V_{(ob1)}$, $V^{(e)}(t_1) = (x^{(e)}_{t1}, y^{(e)}_{t1}, v^{(e)}_{t1})$

FIGURE 5. Ego is driving before entering Region $\mathcal{R}^{(e)}(t_1)$, and one static obstacle $V^{(ob1)}$ is in front of it.



(b) at time t_2, ego vehicle is in parallel with $V_{(ob1)}$, $V^{(e)}(t_2) = (x^{(e)}_{t2}, y^{(e)}_{t2}, v^{(e)}_{t2})$

FIGURE 6. At time t_2, ego is driving within region $\mathcal{R}^{(e)}(t_2)$, and one static obstacle $V^{(ob1)}$ is parallel to it.



(c) at time t_3, ego vehicle is driving over $V_{(ob1)}$, $V^{(e)}(t_3) = (x^{(e)}_{t3}, y^{(e)}_{t3}, v^{(e)}_{t3})$

FIGURE 7. At time t_3, ego enters Feasible Region $\mathcal{R}^{(e)}(t_3)$ with $V^{(ob1)}$ behind it.

VI. SIMULATION RESULTS

In this section, several scenarios are set up to verify the performance of the proposed optimization problems. Firstly, Case 1 is to verify the constrained optimal control problem (**OCP 1**) in (12) for trajectory generation of high-level control. The Parameter Values in Trajectory Planning for the OCP 1 are indicated in Table 1.

Case 2 deals with the optimal control problem (**OCP 2**) in (13) while evaluating a proposed lateral position deviation and relative yaw angle by steering control of a 7 DOF vehicle plant; Case 3 tests the overall collision avoidance system with the presence of multiple participants in roadway driving.

Firstly, in section IV-A, we have established an optimal control problem 1 for collision-free trajectory generation with time-varying MPC(LTV-MPC) which uses a successive linearization process model and its ability for predicting future behaviors by evaluating the objective function in (12), thus, a good or poor performance mainly relates to how close those forecasting states are to system plant evolution.

A. CASE 1: TRAJECTORY PLANNING TEST OF HIGH-LEVEL PLANNER

Case 1 is designed to test two objectives; the first one is to compare the proposed time-varying MPC in section (IV-A) with another conventional MPC to show which planner could have a corresponding trajectory with less position deviation; finally, the second objective is to show the proposed space-partition function how to *define* collision constraints in different time steps, and, to do so, we set up 3 scenarios with different MPC planners and velocities (from $30km/h$ to $80km/h$), which is a general scenario setting as shown in Fig. 8.

TABLE 1. Parameter values in trajectory planning OCP 1.

Symbols	Value	Descriptions
L_{lane}, W_{lane}	$+\infty, 4m$	longitudinal& lateral tire forces
L_{ob}, W_{ob}	$5m, 2m$	length and width of obstacle vehicle $V^{(ob1)}$
L_e, W_e	$5m, 2m$	length and width of ego vehicle $V^{(e)}$
N_p, N_c	$30Ts, 5Ts$	prediction time steps, control steps, Ts is time interval set in simulation (sometimes using abbreviation N replacing $N = N_p = N_c$)
d_{range}	$30m$	detective range for ego $V^{(e)}$, $d_{range} = v^{(e)}_{goal} * t_{gap}, t_{gap} = 1.5s$
Ts	$0.05s$	sampling time
$[\delta_{min}, \delta_{max}]$	$[-1, 1](rad)$	constraint range for input value in planning OCP

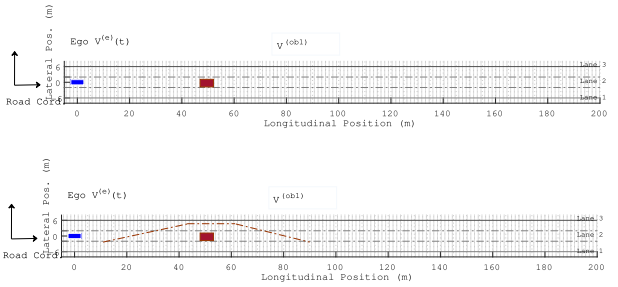


FIGURE 8. Scenario illustration of Case 1 where ego vehicle $V^{(e)}$ is driving with different entry velocity $v^{(e)}(t0) = 30km/h, 60km/h, 80km/h$ respectively.

In Fig. 8, assuming that an ego vehicle $V^{(e)}$ initially travels at its entry velocity $(v^{(e)}(t_0))$ in lane 2 with initial state $(x^{(e)}(t_0), y^{(e)}(t_0), \theta^{(e)}(t_0), v^{(e)}(t_0)) = [0, 0, 0, v^{(e)}(t_0)]$ in the current road coordinate frame; one static obstacle $V^{(ob1)}$ is located in the same lane $[x^{(ob1)}(t), y^{(ob1)}(t), \theta^{(ob1)}(t), v^{(ob1)}(t)] = [50, 0, 0, 0]$. If $V^{(e)}$ approaches the goal point $(x^{(e)}(t_{goal}), y^{(e)}(t_{goal}), \theta^{(e)}(t_{goal}), v^{(e)}(t_{goal})) = [+\infty, 0, 0, v^{(e)}(t_{goal})]$, the aim becomes how to replan a safe path and follow it to avoid collision with other objects along the road.

To test generation performance with varying parameters, 3 different entry velocities are applied in this scenario, $v^{(e)}(t_0) = 30km/h, 60km/h, 80km/h$, respectively, and the corresponding parameters are shown in Table 2.

In Fig. 9, all lateral position curves are kept within the upper boundary and lower boundary $[y_{min}, y_{max}] = [-6, 6](m)$ indicating that the lane boundary constraint defined in section V-B is active when ego was driving.

TABLE 2. Parameter settings.

Scenarios	Initial $V^{(e)}(t0)$	Target $V^{(e)}(t_{goal})$	Initial $V^{(ob1)}$
Scenario 1	(50m,0,30km/h)	(+∞,0,0,60km/h)	(50m,0,0,0)
Scenario 2	(50m,0,60km/h)	(+∞,0,0,80km/h)	(50m,0,0,0)
Scenario 3	(50m,0,80km/h)	(+∞,0,0,80km/h)	(50m,0,0,0)

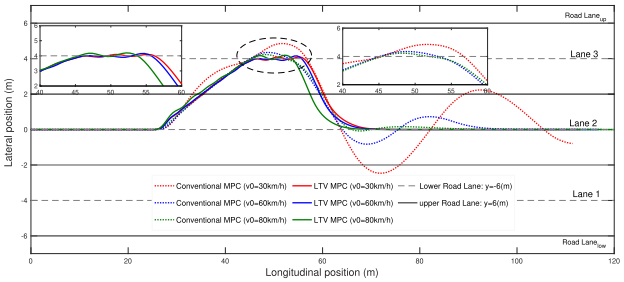


FIGURE 9. Avoidance trajectory with different entry velocity $v^{(e)}(t_0) = 30km/h, 60km/h, 80km/h$, respectively.

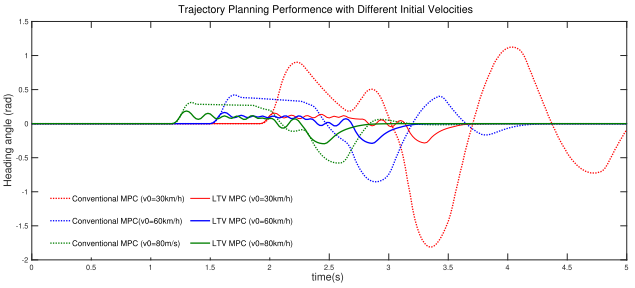


FIGURE 10. Heading angle comparison of avoidance path with different entry velocity $v^{(e)}(t_0) = 30km/h, 60km/h, 80km/h$, respectively.

TABLE 3. Computational time.

OCP in Control Strategy	Computational Time(s)
Linear time-varying MPC	0.003317
Conventional MPC	0.003296

it showed the planned trajectory output variables of ego vehicle when performing avoidance maneuver. Ego $V^{(e)}$ started at $(x^{(e)}(t_0), y^{(e)}(t_0), \theta^{(e)}(t_0), v^{(e)}(t_0)) = [0, 0, 0, v^{(e)}(t0)]$ and was able to avoid the obstacles $50m$ in front for both two controllers. However, the blue and red dashed lines, representing conventional MPC controllers in speed 30 km/h and 60 km/h respectively, had oscillation when the ego vehicle passed over the obstacle $v^{(ob1)}$. After approaching over $V^{(ob1)}$ and especially for conventional MPC with entry speed 30 km/h, the ego vehicle had much large deviation on lateral position and could not keep stable.

In Fig.10, the heading angle of blue and red dashed lines had much more oscillation before converging to zero, especially the red one, as the heading angle of ego cannot come to zero when approaching over $V^{(ob1)}$. In the two above figures, and when under the same scenario and entry velocity of 80 km/h, the time-varying MPC controller had better performance in the trajectory planning problem.

The cause of these oscillations in lateral position and heading angles is the same as that of the mismatch between the prediction model and the controlled vehicle model. When ego vehicle was driving, the velocity augmented from $v^{(e)}(t_0)$ to $v^{(e)}(t_{goal})$; this difference of velocity caused changes in state-space parameters of the real vehicle model, however, the prediction model of the conventional MPC got linearized at entry speed and didn't keep updated with time. This difference between two models resulted in an inaccurate prediction value, namely *the model mismatch*. This inaccurate prediction state would generate a future prediction state profile with large deviation from the desired reference $(y^{(e)}(t_{goal}), \theta^{(e)}(t_{goal}), v^{(e)}(t_{goal})) = (0, 0, 80km/h)$.

From the above results, we can conclude that, using time-varying MPC-based planner, we can observe that, when the entry speed varies from $v_0 = 30\ km/h - 80km/h$, the lateral position and heading angle of the ego vehicle can steer back to the zero when passing through the obstacle position. It can be verified that the online linearization helps the time-varying prediction model to be consistent with the changing parameter of the real vehicle plant.

The OCP 1 was implemented in Matlab and solved with QP solver quadprog; the average computation time was around 3ms. The detailed average computation effort between the time-varying MPC and the conventional MPC in the above scenarios is shown in Table 3:

B. CASE 2: COMPARISON WITH SEQUENCE CONVEX PROGRAM

Some works consider that the avoidance constraint is set to be the inter-distance between an enlarged obstacle and the ego vehicle, i.e. a circular polytope represents this enlarged area in [32], [33] or a rectangular polytope. The core difference for those different shapes of obstacles is how to treat and solve the non-convexity raised by those shapes. Thus, in this subsection, we compare the circular polytope constraint with the partition function constraint proposed in this paper. Generally, the distance between an obstacle with one circular polytope and the ego would be written as:

$$d^{(i,ob)} = \| \xi^{(i)}(t) - x^{(ob)}(t) \| \geq d_{safe} \tag{34}$$

where the position in this equation will be used during prediction steps; intuitively, this constraint would result in a quadratic form; $\xi^{(i)}(t)$ is the ego vehicle state (including position value).

The main idea of Sequential Convex Programming (SCP) is to convexify the non-convex parts of the objective function and the constraints, and then preserve their convex parts. Convex optimization methods always involve a global solution with numerical accuracy. Their computation time grows up with the size of the optimization problem. Contrarily, non-convex optimization methods either compute a local solution with less computational efficiency than those convex methods, or compute a global solution with computational inefficiency, i.e., their computation time grows exponentially

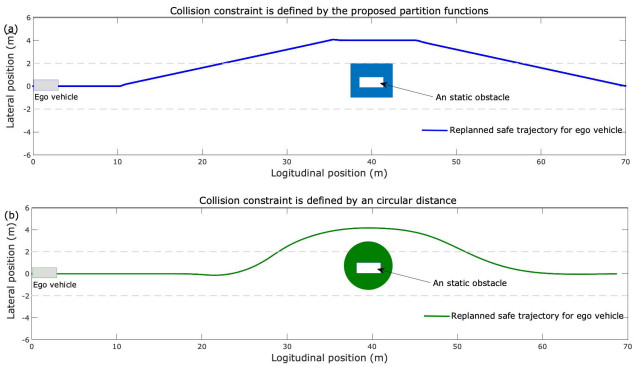


FIGURE 11. Comparison test between the proposed planning method and the SCP method in trajectory planning problem.

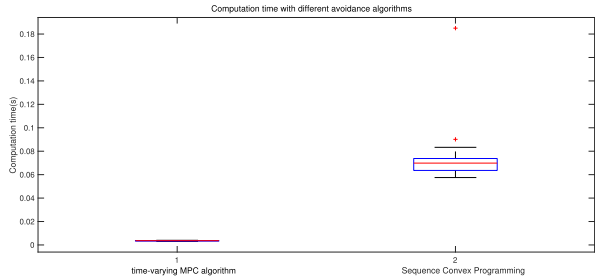


FIGURE 12. Computation time comparison between the proposed planning method and the SCP method in trajectory planning problem.

with the size of the optimization problem such as mixed integer programming. SCP is a local optimization method to solve the non-convex optimization problems. Thus, to deal with the non-convexity raised by obstacles in **OCP1** (12). Case 3 in this subsection is proposed to have some comparison tests with our proposed methods, and these two methods are about to solve the same planning problem in (12).

In Fig. 11, the obstacle is located in a fixed position $(x^{(ob)}, y^{(ob)}) = (40, 0)$; there are two different shapes, one blue rectangle and a green circle. As defined in (34), the safe distance in SCP is set with radius $d_{safe} = W_{lane}^2 = 4^2(m)$. The blue one represents the obstacle shape used in the proposed planning algorithm, while the green circle is the shape used in SCP algorithm. With respect to these two algorithms, the difference is that the green trajectory is curved with adaptation to the circular outline of the green obstacle; the outline of the blue trajectory is shaped by proposed partition constraint, using Algorithm 1.

Another difference is the computation time shown in Fig. 12. The proposed planning method uses less computational efforts than the use of circular area for obstacles.

C. CASE 3: DOUBLE-SHIFT-LINE TEST OF LOW-LEVEL TRACKING CONTROLLER

In section IV-B, a different process model than that of the high-level LTV-MPC planner is used; the system plant for the whole system is a 7 degrees-of-freedom pre-assembled vehicle dynamics model in *Vehicle Dynamics Blockset*^{TM} in MATLAB to analyze the dynamic system response in

TABLE 4. Nomenclature.

Symbols	Value	Descriptions
m	$1575kg$	total mass
l_f, l_r	$1.2, 1.6m$	longitudinal distance from front axle to center of mass, longitudinal distance from rear axle to center of mass
I_z	$2875\ kg * m^2$	yaw polar inertial
h	$0.35m$	vertical distance from center of mass to axle plane
β	-	body slip angle of C.O.G
$\alpha_{i,j}$	-	slip angle of steering wheel, $i \in rear, front, j \in left, right$
g	$9.81\ m * s^{-2}$	gravitational acceleration
μ	0.91	nominal friction scaling factor
w	$w(0, 0.1^2)$	Gaussian noise

common ride and handling maneuvers including three DOF for vehicle body and one DOF for each wheel. The corresponding process model in this level is a dynamic bicycle model presented in section (III-B), the parameters are set as Table 4.

When the desired trajectory $\gamma_{des}^{(2)} = \xi^{(1)}(t)$ is fed from the high-level planner, dynamic parameters of vehicles such as the lateral stiffness and slip rate of tires will result in some deviation in position or yaw angle error. The control input for the plant vehicle is the steering angle of the front wheel and the goal is to follow the trajectory as closely as possible by minimizing the vehicle deviation from the desired trajectory with corresponding speed.

To verify the reference trajectory tracking performance of this low-level LTV-MPC controller, the test of a double-shift-lane maneuver will be taken with different given initial velocities; these velocities range from $30km/h$ to $90km/h$ as shown in (35) and Fig. 13. So, the parameters of a double-shift reference are set as:

$$\begin{aligned} Y_{ref} &= \frac{d_{y1}}{2}(1 + \tan(z_1)) - \frac{d_{y2}}{2}(1 + \tan(z_2)) \\ \psi_{ref} &= \arctan(\frac{1.2d_{y1}}{d_{x1}}(\frac{1}{\cosh(z_1)})^2 - \frac{1.2d_{y2}}{d_{x2}}(\frac{1}{\cosh(z_2)})^2) \end{aligned} \tag{35}$$

where X, Y indicate the longitudinal/lateral position in global coordination, $z_1 = 2.4(X - 27.19)/25 - 1.2$, $z_2 = 2.4(X - 56.46)/25 - 1.2, d_{x1} = 25, d_{x2} = 21.95$, $d_{y1} = 4.05, d_{y2} = 5.7$.

Fig.14 depicts the simulation results when the double-lane change maneuver is performed by the low-level LTV-MPC controller. Fig.14(a) is the lateral deviation from the desired trajectory, the maximum value happens when velocity is $90km/h$; e_y has a range $[-1.8 \times 10^{-4}, 2.7 \times 10^{-4}]$ with some oscillation between $t = 0.1s - 2s, t = 4.5s$; an exception to this would happen when the position deviation is kept within $[-0.02, 0.015](m)$ when entry velocity is $30km/h$ or $60km/h$; the corresponding relative yaw angle is in 14(b), the error is within $[-0.25, 0.25](rad)$ for initial velocity $30km/h$ and

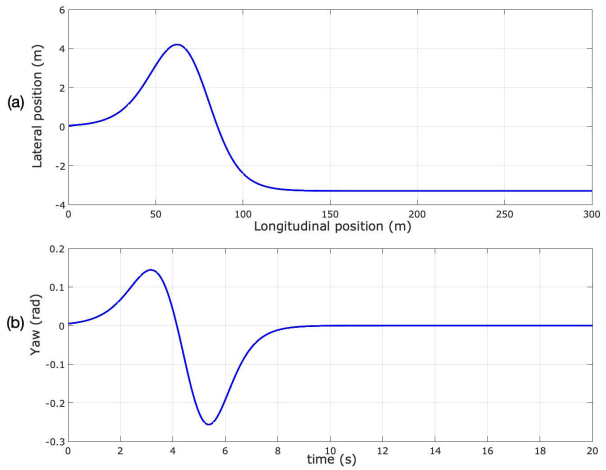


FIGURE 13. Double-shift-line reference.

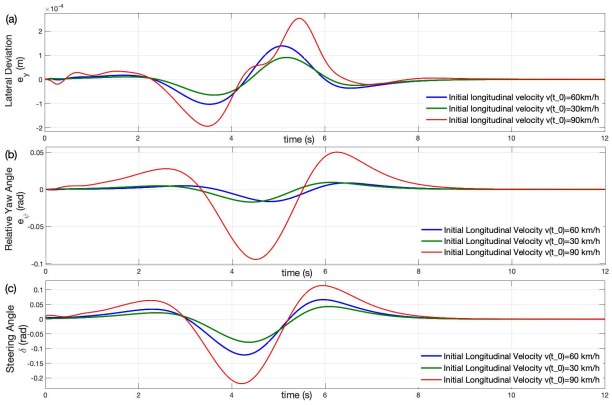


FIGURE 14. Double-shift-line test of the low-level time-varying MPC controller under different entry velocities $v^{(e)}(t_0) = 30km/h$, $60km/h$, $90km/h$, respectively.

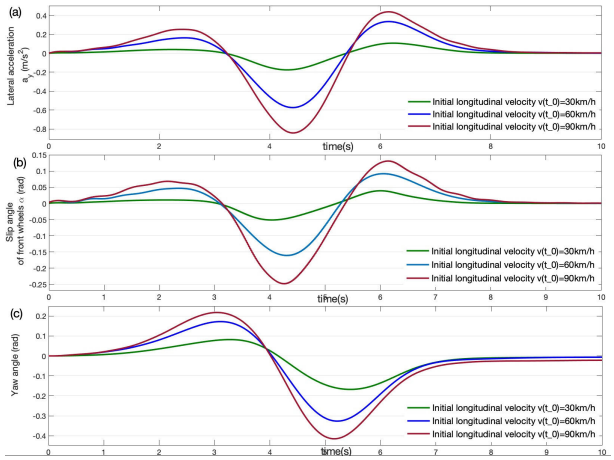


FIGURE 15. Double-shift-line test of the low-level time-varying MPC controller under different entry velocity $v^{(e)}(t_0) = 30km/h$, $60km/h$, $90km/h$, respectively.

$60km/h$ which shows a good tracking ability of proposed controller.

The slip angle of the two front wheels is kept the same as Fig. 15(b) within the range $[-0.3, 0.2](rad)$. This is to guarantee that it stays asymptotically stable in the tracking control problem (13).

A sufficient condition for uniform asymptotic stability of the origin of this lateral dynamic control problem has been validated in this subsection. In [26], [27], the vehicle dynamic control problem has been implemented with a time-varying MPC controller and validated by some ground tests with speeds up to $17m/s = 61.2km/h$ and by considering the lateral and yaw stabilization along the path by seeting additional constraint on the vehicle process model.

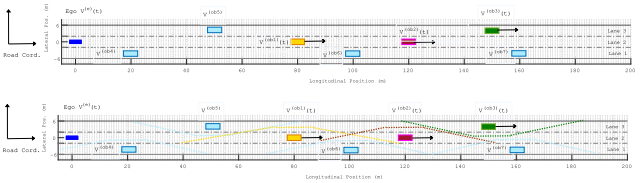


FIGURE 16. The divided space for trajectory planning among multiple static or slow-moving objects, ego one in dark blue $V^{(e)}(t_0) = (0, 0, 0, v^{(e)}(t_0))$, $V^{(e)}(t_0) = V^{(e)}(t_{goal}) = 30km/h$, one in yellow $V^{(ob1)} = (80, 0, v^{(ob1)})$, one in red $V^{(ob2)}(120, 0, v^{(ob2)})$, one in green $V^{(ob3)} = (150, 0, v^{(ob3)})$ and 4 static ones in light blue $V^{(ob4)}$, $V^{(ob5)}$, $V^{(ob6)}$, $V^{(ob7)}$.

D. CASE 4: AVOIDANCE PERFORMANCE FOR HIERARCHICAL COLLISION AVOIDANCE SYSTEM WITH MULTIPLE PARTICIPANTS

The last two tests have dealt with high-level planning and low-level tracking performance with corresponding constraints, respectively. In this subsection, on-line collision avoidance performance is determined with multiple participants; this would show the integrated performance of the proposed two-layer control scheme, i.e. time-varying MPC for both trajectory planning and tracking control is applied in the scenario shown in Fig. 16.

These initial settings of the ego vehicle parameters and road structure stay identical to Case 1 the obstacles are slow-moving vehicles or static objects, especially, slow-moving objects refer to $V^{(ob1)}$, $V^{(ob2)}$, $V^{(ob3)}$; initially, we have $V^{(ob1)} = (80, 0, v^{(ob1)})$, $V^{(ob2)} = (120, 0, v^{(ob2)})$, $V^{(ob3)} = (150, 0, v^{(ob3)})$ with a random velocity in range $v = [1, 10](km/h)$; others are static objects, $V^{(ob4)} = (20, 0, 0)$, $V^{(ob5)} = (45, 4, 0)$, $V^{(ob6)} = (100, 0, 0)$, $V^{(ob7)} = (160, 0, 0)$. For experimental validation of the proposed algorithm, all required routines were developed in MATLAB/Simulink environment.

In Fig. 16, the partition functions are represented as dashed lines for each object. Accordingly, at each time step, the ego vehicle just does maneuvers inside the safe area where it is refined based on these lines and one line is worked at one time step, that is, $V^{(e)}$ just stays at one region, $\mathcal{R}^{(ob)}$ at time t_1 is shown as $\mathcal{R}^{(ob1)}$ in Fig. 5; at time t_2, $\mathcal{R}^{(ob)}$ changes to $\mathcal{R}^{(ob2)}$ in Fig. 6; at time t_3, $\mathcal{R}^{(ob)}$ becomes $\mathcal{R}^{(ob3)}$ in Fig. 7.

In this case, we could see that the proposed state partition functions also work for the situation with multiple obstacles and are able to generate a safe trajectory across them; the

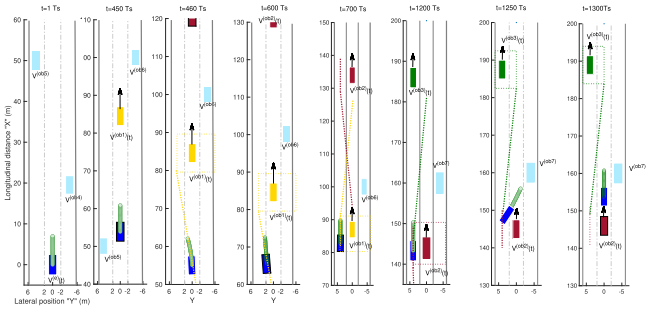


FIGURE 17. Obstacle avoidance planned trajectory with 6 obstacles, including 3 moving participants with random velocity $V^{(ob1)}$, $V^{(ob2)}$, $V^{(ob3)}$, 4 static ones $V^{(ob4)}$, $V^{(ob5)}$, $V^{(ob6)}$, $V^{(ob7)}$.

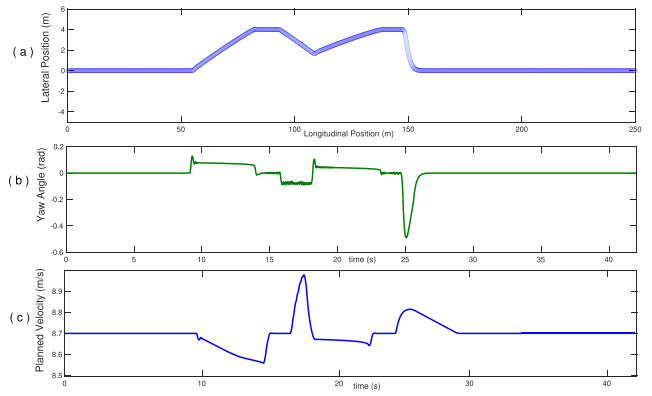


FIGURE 18. Safe trajectory solved by optimal control problem 1, with 6 participants along the road.

details were shown in Fig. 17. T_s is the sampling time, $0.02s$. At time $t = 1\ T_s$, two static objects $V^{(ob4)}$ and $V^{(ob5)}$ in front of ego $V^{(e)}$ occupy a critical region in their corresponding lane, thus they do not influence the lane of ego; the predicted trajectory is a set of small green circles attached to the ego, thus the predicted trajectory can stay in its original lane until passing them; at time $t = 450T_s$, it has passed the $V^{(ob4)}$ and $V^{(ob5)}$, which are located at $(20, -4)$ the right-side lane of ego and $(50, 4)$ the left-side lane of ego, respectively. Ego approaches the yellow object $V^{(ob1)}$; at time $460T_s$, ego is within the d_{range} of $V^{(ob1)}$, and the yellow dashed line is the partition line, which means ego could drive on the "outer side" of this yellow line and it cannot be crossed. Therefore, the predicted trajectory of ego at this moment is twisted to keep ego vehicle on the left side of this yellow line; the same occurs for time at $600T_s$; when driving parallel to yellow $V^{(ob1)}$ at time $700T_s$, the partition yellow line becomes a parallel line; it is also forbidden for the ego to go across it to right side.

The replanned safe path is shown in Fig. 18(a); ego $V^{(e)}$ drives with 6 participants along a 3-lane-one-way scenario, and is located in lane 2; this replanned path was done by a time-varying MPC controller. This controller also validated that the collision constraint of space partition in (12d) function worked for multiple obstacles.

In Fig. 19, the tracking error in lateral position and relative yaw angle are within acceptance. Compared to Case 2, slip

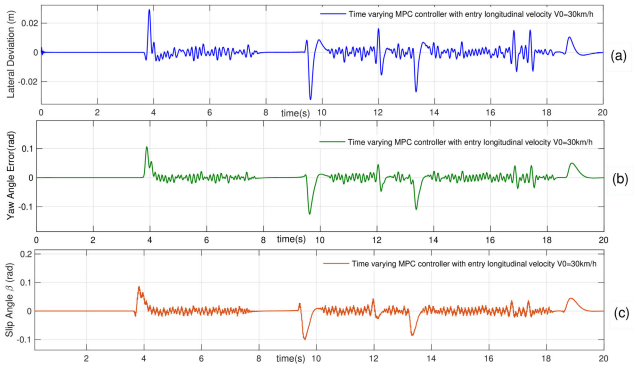


FIGURE 19. Low-level tracking controller performance found by optimal control problem 2. The corresponding results are lateral position error and relative yaw angle, which are the errors with respect to the desired path in the tracking control problem. Additionally, slip angle of vehicle C.O.G.

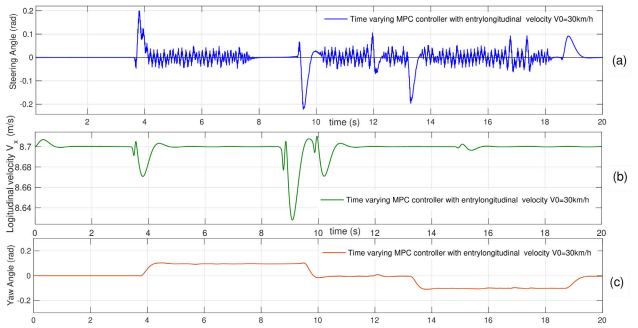


FIGURE 20. Low-level tracking controller performance found by optimal control problem 2. The corresponding steering effort of the front wheels is kept within constraints, and the corresponding velocity is kept around the desired value of $v^{(e)}_{t_{goal}} = 8.7m/s = 30km/h$.

angle is less than that in double-shift-lane, which ensures the tracking stability around the planned path.

VII. CONCLUSION

In this paper, a two-level planning architecture for collision avoidance systems has been presented including avoidance maneuver generation and tracking control for automated vehicle dynamics. The high-level path planner computes a feasible trajectory through constrained optimization problems, then the low-level controller computes the required vehicle steering angle value to stabilize a 7 DOF nonlinear vehicle plant around this trajectory. The core of this two-layer model based optimal control scheme is the successive online linearization for the prediction model to stay consistent with the nonlinear system plant.

First, in the simulation section, several test scenarios showed that our planning algorithm is capable of controlling the vehicle to avoid collision with other road participants; additionally, when compared with another conventional MPC (using different entry speeds), the ego vehicle performed better and had greater control over the steering angle of the front wheels back to the original lane, mostly because of less oscillation.

Secondly, the optimization problem for trajectory tracking was evaluated using a double shift-line where the lateral position deviation and relative yaw angle were constrained within an acceptable range. What's more, this hierarchical system was evaluated in a structured roadway scenario with multiple participants, integrating the optimization-based replanning problem with the stabilization control problem.

Finally, this paper has illustrated the ability of appropriate avoidance maneuver decision and trajectory planning for automated driving with the guarantee of lateral and yaw stabilization of a vehicle. The limitation of this safe trajectory planning method is that it does not take the obstacles' or other vehicles' movement into consideration, i.e., not considering the coupled objective function when solving its own objective function. Potential future research lies on the cooperative optimal control for collision avoidance and on-road driving.

REFERENCES

[1] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 589–606, Sep. 2010.

[2] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "A fast RRT algorithm for motion planning of autonomous road vehicles," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1033–1038.

[3] W. Khaksar, K. S. M. Sahari, and T. S. Hong, "Application of sampling-based motion planning algorithms in autonomous vehicle navigation," *Auton. Vehicle*, vol. 735, pp. 22–38, Sep. 2016.

[4] U. Z. A. Hamid, H. Zamzuri, T. Yamada, M. A. A. Rahman, Y. Saito, and P. Raksincharoensak, "Modular design of artificial potential field and nonlinear model predictive control for a vehicle collision avoidance system with move blocking strategy," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 232, no. 10, pp. 1353–1373, Sep. 2018.

[5] R. Hayashi, J. Isogai, P. Raksincharoensak, and M. Nagai, "Autonomous collision avoidance system by combined control of steering and braking using geometrically optimised vehicular trajectory," *Vehicle Syst. Dyn.*, vol. 50, no. sup1, pp. 151–168, Jan. 2012.

[6] S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, and A. Mouzakitis, "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annu. Rev. Control*, vol. 45, pp. 76–86, Jan. 2018.

[7] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 4239–4244.

[8] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1204–1216, Jul. 2017.

[9] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J. K. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," in *Proc. 11th Int. Symp. Adv. Vehicle Control*, no. 2, 2012, pp. 1–6.

[10] T. Shamir, "How should an autonomous vehicle overtake a slower moving vehicle: Design and analysis of an optimal trajectory," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 607–610, Apr. 2004.

[11] J. Baber, J. Kolodko, T. Noel, M. Parent, and L. Vlacic, "Cooperative autonomous driving: Intelligent vehicles sharing city roads," *IEEE Robot. Autom. Mag.*, vol. 12, no. 1, pp. 44–49, Mar. 2005.

[12] G. Hegeman, K. Brookhuis, and S. Hoogendoorn, "Opportunities of advanced driver assistance systems towards overtaking," *Eur. J. Transp. Infrastruct. Res.*, vol. 5, no. 4, pp. 281–296, 2020.

[13] E. I. Vlahogianni, "Modeling duration of overtaking in two lane highways," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 20, pp. 135–146, Sep. 2013.

[14] L. Tang, F. Yan, B. Zou, K. Wang, and C. Lv, "An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles," *IEEE Access*, vol. 8, pp. 51400–51413, 2020.

[15] H. Guo, J. Liu, D. Cao, H. Chen, R. Yu, and C. Lv, "Dual-envelop-oriented moving horizon path tracking control for fully automated vehicles," *Mechatronics*, vol. 50, pp. 422–433, Apr. 2018.

[16] J. Liu, W. Han, Y. Zhang, Z. Chen, and H. Peng, "Design of an online nonlinear optimal tracking control method for unmanned ground systems," *IEEE Access*, vol. 6, pp. 65429–65438, 2018.

[17] S. Li, Z. Li, Z. Yu, B. Zhang, and N. Zhang, "Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control," *IEEE Access*, vol. 7, pp. 132074–132086, 2019.

[18] G. Franze and W. Lucia, "A receding horizon control strategy for autonomous vehicles in dynamic environments," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 695–702, Mar. 2016.

[19] N. Murgovski and J. Sjoberg, "Predictive cruise control with autonomous overtaking," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 644–649.

[20] J. Karlsson, N. Murgovski, and J. Sjoberg, "Temporal vs. Spatial formulation of autonomous overtaking algorithms," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1029–1034.

[21] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," *Control Eng. Pract.*, vol. 41, pp. 124–133, Aug. 2015.

[22] J. Jiang and A. Astolfi, "Shared-control for a rear-wheel drive car: Dynamic environments and disturbance rejection," *IEEE Trans. Human-Machine Syst.*, vol. 47, no. 5, pp. 723–734, Oct. 2017.

[23] S. M. Erlien, S. Fujita, and J. C. Gerdes, "Shared steering control using safe envelopes for obstacle avoidance and vehicle stability," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 441–451, Feb. 2016.

[24] Z. Wan and M. V. Kothare, "Efficient scheduled stabilizing model predictive control for constrained nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 13, nos. 3–4, pp. 331–346, 2003.

[25] P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "MPC-based yaw and lateral stabilisation via active front steering and braking," *Vehicle Syst. Dyn.*, vol. 46, pp. 611–628, Sep. 2008.

[26] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 862–875, 2008.

[27] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Proc. 46th IEEE Conf. Decis. Control*, 2007, pp. 2980–2985.

[28] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *Int. J. Veh. Auto. Syst.*, vol. 3, nos. 2–4, pp. 265–291, 2005.

[29] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[30] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*, vol. 10. Cham, Switzerland: Springer, 2013.

[31] C. C. Chen and L. Shaw, "On receding horizon feedback control," *Automatica*, vol. 18, no. 3, pp. 349–352, May 1982.

[32] J. Liu, W. Han, C. Liu, and H. Peng, "A new method for the optimal control problem of path planning for unmanned ground systems," *IEEE Access*, vol. 6, pp. 33251–33260, 2018.

[33] M. Kloock, P. Scheffe, L. Botz, J. Maczijewski, B. Alrifaee, and S. Kowalewski, "Networked model predictive vehicle race control," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1552–1557.

QING SHI received the B.S. degree from Beijing Jiaotong University (BJTU), in 2014, and the M.S. degree from Guizhou University, in 2017. She is currently pursuing the Ph.D. degree with the CRIStAL, Centrale Lille Institut, with the research interests of trajectory optimization and control for automated vehicles and driving safety in intelligent transportation systems, under the supervision of Prof. A. E. Kamel.

JIN ZHAO received the Ph.D. degree in 2010. He is currently a Full Professor with Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University. His research interests include intelligent transportation systems and design of heat exchange systems. He is also a Senior Member of the Chinese Mechanical Engineering Society, and is on the editorial board of *Modern Machinery Science Journal*; additionally, he is also the Certified Expert of the China Engineering Education Certification Association.

ABDELKADER EL KAMEL (Senior Member, IEEE) received the engineering diploma, master diploma from the Ecole Centrale de Lille, Villeneuve-d'Ascq, France, in 1990, and the Ph.D. degree in 1994. He was in charge, from 2008 to 2013, of the Computer-Telecommunication-Robotics Department, Ecole Centrale de Beijing. He is currently a Full Professor with the Ecole Centrale de Lille. He is appointed at the Permanent Visiting Professor in major Engineering/Business Schools in Tunisia, and the Visiting Professor in Chili, Canada, India, and Japan. He has been invited speaker for more than 25 plenary lectures or tutorials in international conferences and is on the Program Committee for about 150 IEEE SMC and CSS, IFAC, and several other conferences. His research interests include intelligent systems monitoring, complex systems analysis and control, and computational intelligence and optimization. Applying intelligent technologies, virtual reality and optimization techniques in transportation systems, and mobile cooperative robots are among the current focus of work. He has published more than 100 technical articles in international journals and conferences; has participated to five books in his field of interest and has edited more than 15 proceedings of various conferences and workshops.

ISMAEL LOPEZ-JUAREZ received the B.Eng. degree from the National Autonomous University of Mexico, the M.Sc. degree from UMIST, in 1996, and the Ph.D. degree in intelligent robotics from the Nottingham Trent University, U.K., in 2000. He is currently the Leader with the Intelligent Manufacturing Laboratory, CINVESTAV. He has published over 200 articles, has supervised eight Ph.D., 17 M.Sc., and seven B.Eng. students. He has one patent and has been responsible for several technological projects and technology transfer within the industry. His research interests include instrumentation, self-adaptive industrial robots, neural networks, and machine vision. He has acted as a regular reviewer for major journals in his field. He is also a member of the National Researchers Systems in Mexico (SNI), level II. He is also a Guest Editor of the *Transactions on Intelligent Welding Manufacturing* (Springer) and also a member of the Editorial Committee of the IEEE LATIN AMERICA TRANSACTIONS.