Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Robust Hierarchical Model Predictive Control for Trajectory Tracking with Obstacle Avoidance ⋆

Yanchuan Xu*, Huarong Zheng, Weimin Wu*, Jun Wu***

*College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, (e-mail: 21832130@zju.edu.cn, {wmwu, jwu}@iipc.zju.edu.cn)
**Ocean College, Zhejiang University, Zhoushan 316021, China, (e-mail: hrzheng@zju.edu.cn)

Abstract: A robust hierarchical path planning and trajectory tracking framework is proposed to maintain a collision-free path for autonomous vehicles. For the path-planning, a constrained finite-time optimal problem is solved to generate a feasible and collision-free trajectory considering the vehicle kinematics. For the trajectory-tracking, a motion controller is proposed by solving a constrained model predictive control problem, obtaining the front wheel steering angles. Furthermore, to enhance the robustness of the motion controller against the uneven network-induced time delay and unmodelled lateral vehicle dynamics, an additional error feedback mechanism is introduced in the motion controller. Simulations are conducted when both moving obstacles and static obstacles exist. Simulation results show that the proposed hierarchical control framework can effectively guarantee safe and feasible driving maneuvers.

Keywords: Trajectory planning, vehicle motion control, autonomous vehicles, model predictive control, robustness

1. INTRODUCTION

Recently, autonomous driving has aroused great interest in both industry and academia. With the development of drive-by-wire, break-by-wire, and steer-by-wire technologies, the autonomous vehicle system usually adopts a distributed control architecture because of its modularity and high degree of flexibility (Li et al. (2010)). The controller signals and sensor signals are exchanged via a communication network, i.e., controller area network (CAN) (Caruntu et al. (2013)). Although the x-by-wire and in-vehicle network technology bring about actuation flexibility and easiness of system diagnosis, uneven network-induced delays are brought up in the sensor-to-controller (S-C) as well as the controller-to-actuator (C-A) communication channels (Zhu et al. (2014)).

To safely and fast navigate an autonomous vehicle to a target point, it is essential to design a robust motion plan method for generating feasible trajectories (Lim et al. (2018)) and tracking the generated trajectories. As reviewed in Paden et al. (2016), typical motion planners are search-based planners, sampling-based planners, and numerical optimization-based planners. Dijkstra and A* are well-known graph search-based planners and can find the shortest path by traversing the grid after discretizing the environment. However, it is difficult to choose a proper grid resolution as well as to track the path consisting of connected grids (Ma et al. (2015)). Different from these search-based approaches, sampling-based planners, such as RRT* (Karaman et al. (2011)) can produce valid paths by extending nodes in a continuous space until the target position is reached. However, the planned path is not optimal and could twist and turn, making it infeasible for vehicles to track directly. Numerical optimizations (Ziegler et al. (2014)) could generate optimal smooth trajectories by defining constraints and an objective function. Because of the capability to systematically handle system nonlinearities and constraints (Gao et al. (2014)), model predictive control (MPC) is usually applied in exploring the optimal trajectory in the space of feasible solutions. For the sake of computational efficiency, simple vehicle models, like the point-mass model, is often used (Gao et al. (2011)). However, oversimplified models could also lead to infeasibilities (Gao et al. (2014)). Therefore, kinematic vehicle models are considered in the motion planner to keep a balance between model accuracy and computational complexity in Li et al. (2017) and Polack et al. (2018).

Previous works mentioned above only focus on exploring a feasible trajectory and assume that the lower-level motion controller can perfectly track the planned trajectories without any uncertainty in the system. To maintain robustness under uneven network-induced delays, robust motion controllers are investigated. Shuai et al. (2013) and Zhu et al. (2014) design H_∞-based LQR active steering controllers for a four-wheel drive (4WD) vehicle, simulations verify the effectiveness of the proposed controllers. Liu et al. (2018) introduce a hierarchical steering control scheme to compensate visual sensor-induced uneven time delays and comparisons are conducted to show bet-

⋆ This research is supported by the National Natural Science Foundation of China under Grant 61903329 and Grant 61433013.
Corresponding author: hrzheng@zju.edu.cn

Copyright lies with the authors

ter tracking performance of the proposed control scheme. However, these robust controllers are only verified in static driving scenarios where the reference positions of the vehicle are predefined and no obstacles are considered.

In this paper, a robust hierarchical model predictive controller is designed so that the ego vehicle can maneuver safely and smoothly. Particularly, at the upper-level, local collision-free trajectories are generated by solving a constrained finite-time optimal problem. At the lower-level, a nominal nonlinear predictive controller, along with a linear error feedback control law, is designed to improve the tracking performance in the presence of uncertainties. In particular, the nominal MPC controller is designed based on the nominal parameters and the linear lateral tire force model, and the error feedback control law is designed using the predicted nominal states and the current measured state values. Simulations are carried out to demonstrate the effectiveness of the proposed algorithms.

The rest of the paper is organized as follows. In section II, kinematic and dynamic vehicle models are developed for motion planning and vehicle control purposes. The upper-level planner for trajectory generation is proposed in Section III. The low-level trajectory tracking controller is designed in Section IV. Simulation results in the MATLAB/Simulink environment are presented in Section V. Conclusions are given in Section VI.

2. VEHICLE MODEL DEVELOPMENT

In this section, a kinematic bicycle vehicle model and a dynamic vehicle model are developed. As Fig. 1 shows, a global cartesian coordinate system (x-y coordinate) is used for the inertial frame of reference.

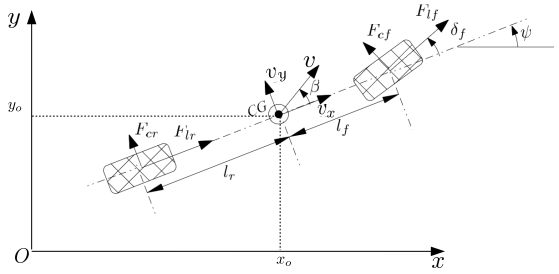


Fig. 1. Vehicle dynamics in the global coordinate system.

2.1 Kinematic Bicycle Model for Motion Planning

The kinematic bicycle vehicle model typically assumes that the vehicle moves without skidding. Therefore, the motion of the vehicle can be modelled in the x-y coordinate as:

$$\begin{aligned}&\dot{x}_o = v\cos(\psi+\beta(\delta_f)), \dot{y}_o = v\sin(\psi+\beta(\delta_f))\\&\dot{\psi} = \frac{v\cos(\beta)}{l_r+l_f}\tan(\delta_f), \beta(\delta_f) = \tan^{-1}(\frac{l_r}{l_f+l_r}\tan(\delta_f))\end{aligned} \tag{1}$$

where x_o and y_o are the global coordinates of the vehicle's center of gravity (CG), and ψ, $\dot{\psi}$ are the yaw angle and the yaw rate, respectively. v is the velocity of the vehicle, and l_f, l_r are the distances of the front tire and the rear tire to the vehicle's CG, respectively. β is the slip angle, denoted as a function of the front steering angle δ_f. Then, the kinematic model dynamics are compactly written as:

$$\dot{X}(t) = f_{kin}(X(t), u(t)) \tag{2}$$

where the kinematic dynamic system state is $X \triangleq [x_o, y_o, \psi]^T$ and the control input is defined as $u \triangleq \delta_f$. To formulate the MPC controller as a constrained finite-dimensional optimization problem, we discretize the kinematic model $\dot{X} = f_{kin}(X, u)$ with sampling time T_s^{kin}:

$$\begin{aligned}X(k+1) &= f_{kin}^d(X(k), u(k)),\\ u(k) &= u(k-1) + \Delta u(k)\end{aligned} \tag{3}$$

where f_{kin}^d is the discrete vehicle kinematic function.

2.2 Vehicle Lateral Dynamics

The lateral dynamics of the vehicle can be written as:

$$\begin{aligned}m\dot{v}_y &= -mv_x\dot{\psi} + 2F_{yf} + 2F_{yr},\\ I_z\ddot{\psi} &= 2l_fF_{yf} - 2l_rFyr\end{aligned} \tag{4}$$

where m, I_z denote the vehicle mass, yaw inertia, respectively. v_x and v_y denote the vehicle longitudinal and lateral velocities in the body-fixed coordinate oxy, respectively. F_{yf} and F_{yr} are the front and rear tire forces in the vehicle lateral direction, respectively. The lateral tire forces in the vehicle body frame are calculated by:

$$F_{yf} = F_{lf}\sin(\delta_f) + F_{cf}\cos(\delta_f), \quad F_{yr} = F_{cr} \tag{5}$$

where $F_{l\star}$ and $F_{c\star}$ are the longitudinal and lateral tire forces, $\star \in f, r$, respectively. Considering the transformation between the global coordinate and the body-fixed coordinate, the following equations are given:

$$\begin{aligned}\dot{y}_o &= v_x\sin(\psi) + v_y\cos(\psi),\\ \dot{x}_o &= v_x\cos(\psi) - v_y\sin(\psi).\end{aligned} \tag{6}$$

Under the assumption of small slip angle $\alpha_{f,r}$ and small front steering angles δ_f, the following equations are obtained (Ma et al. (2019))) :

$$\begin{aligned}&F_{yf} = F_{cf} = C_f\alpha_f, \qquad F_{yr} = F_{cr} = C_r\alpha_r,\\ &\alpha_f = -\frac{v_y + l_f\dot{\psi}}{\dot{x}} + \delta_f, \qquad \alpha_r = -\frac{v_y - l_r\dot{\psi}}{\dot{x}},\end{aligned} \tag{7}$$

where C_f is the front tire cornering stiffness and C_r is the rear tire cornering stiffness. Combining (4)-(7), the nonlinear vehicle dynamics can be compactly written as:

$$\begin{aligned}&\dot{\xi}(t) = f_{dyn}(\xi(t), u(t)),\\ &\eta(t) = C_c\xi(t), \quad C_c = \begin{bmatrix}0&0&0&1\\0&1&0&0\end{bmatrix}\end{aligned} \tag{8}$$

where the state variables of the lateral dynamic model are defined as $\xi \triangleq [v_y\ \psi\ r\ y_o]^T$, with $r \triangleq \dot{\psi}$ denoting the yaw rate. The control input is defined as $u \triangleq \delta_f$ and outputs are defined as $\eta \triangleq [y_o\ \psi]^T$. The lateral vehicle dynamics (8) can be rewritten as:

$$\dot{\xi}(t) = A_c(t)\xi(t) + B_c(t)u(t) + d(t) \tag{9}$$

where the state matrix $A_c(t)$, the input matrix $B_c(t)$ and the deviation of the linear system from the nonlinear system, $d(t)$, are defined as:

$$A_c(t) = \left.\frac{\partial f_{dyn}}{\partial \xi}\right|_{\xi(t),u(t)}, \quad B_c(t) = \left.\frac{\partial f_{dyn}}{\partial u}\right|_{\xi(t),u(t)} \tag{10}$$
$$d(t) = f_{dyn}(\xi(t), u(t)) - A_c(t)\xi(t) - B_c(t)u(t).$$

In digital computations, the continuous systems (9) is transformed in the discretized form with a constant sampling time T_s. Both the C-A delay τ_k^{ca} and the S-C delay τ_k^{sc} exist in the controlled system. Lumping the two transmission delays, we have $\tau_k \triangleq \tau_k^{ca} + \tau_k^{sc}$. Assuming that the total transmission delay τ_k is smaller than the sampling time T_s, the discrete system model at time k can be revised as (Shuai et al. (2013)):

$$\begin{aligned}\xi(k+1) = A_d(k)\xi(k) + B_d(k)u(k) \\ + B_{d,1}(k)\Delta u(k) + D_d(k)\end{aligned} \tag{11}$$

where

$$\begin{aligned} &A_d(k) = e^{A_c(k)T_s},\ \Delta u(k) = u(k) - u(k-1), \\ &B_{d,1}(k) = \int_{\tau_k}^{T_s} e^{A_c(k)s} ds \cdot B_c(k), \\ &B_d(k) = \int_0^{T_s} e^{A_c(k)s} ds \cdot B_c(k), \\ &D_d(k) = \int_0^{T_s} e^{A_c(k)s} ds \cdot d(k) \end{aligned} \tag{12}$$

Remark 1. $A_c(k)$, $B_c(k)$ and $d(k)$ are considered to be time-invariant when calculating the discrete state space equations at time k.

To formulate the MPC problem, the state variables are augmented as $\xi_a(k+1) \triangleq [\xi(k+1), u(k)]^T$. The discrete state-space model can be rewritten as:

$$\begin{aligned} &\xi_a(k+1) = A(k)\xi_a(k) + B(k)\Delta u(k) + D(k), \\ &\eta(k) = C\xi_a(k) \end{aligned} \tag{13}$$

where

$$\begin{aligned} &A(k) = \begin{bmatrix} A_d(k) & B_d(k) \\ 0 & I \end{bmatrix}, \quad B(k) = \begin{bmatrix} B_{d,1}(k) \\ I \end{bmatrix}, \\ &C = \begin{bmatrix} C_c & 0 \\ 0 & 0 \end{bmatrix}, \quad D(k) = \begin{bmatrix} D_d(k) \\ 0 \end{bmatrix}. \end{aligned} \tag{14}$$

3. UPPER-LEVEL PLANNER FOR TRAJECTORY GENERATION

A hierarchical controller is proposed to track the reference trajectory, while guaranteeing safe maneuvers in the existence of multiple obstacles. As depicted in Fig. 2, the control architecture is composed of a high-level motion planner and a lower-level trajectory tracking controller. The upper-level motion planner is mainly designed to find an optimal collision-free path in the configuration space. The kinematic model is considered to make sure the trajectory is feasible for the vehicle to maneuver.

3.1 The local MPC planner formulation

Considering the discrete kinematic vehicle model (3), the optimization problem for the motion planner is formulated as follows:

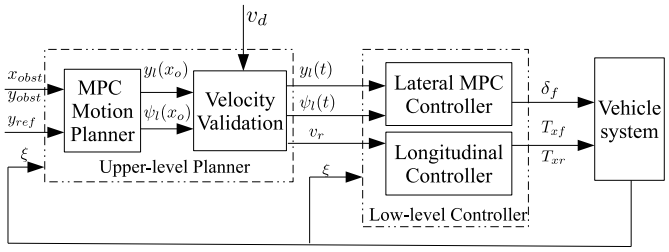


Fig. 2. Architecture of the hierarchical design.

$$\begin{aligned}\min_{\Delta \mathbf{u}} \quad & \sum_{i=1}^{H_p} \|y_o(k+i|k) - y_r\|_Q^2 \\ & + \sum_{j=1}^{H_c} \|\Delta u(k+j|k)\|_R^2 + \sum_{l=1}^{N_{obs}} J_{obst,l}\end{aligned} \tag{15a}$$

$$\begin{aligned} s.t. \quad & X(k+1) = f_{kin}^d(X(k), u(k)),\ X(k|k) = X(k) \\ & \delta_{f,min} \le u(k+i|k) \le \delta_{f,max} \\ & \Delta\delta_{f,min} \le \Delta u(k+i|k) \le \Delta\delta_{f,max}, i = 1, ..., H_c \end{aligned} \tag{15b}$$

where $X(k+i|k)$ denotes the ith predicted kinematic dynamic state at time step k. The prediction and control horizon are denoted as H_p and H_c, respectively. N_{obs} is the number of obstacles within the perception range. $\delta_{f,j}$ and $\Delta\delta_{f,j}$, $j \in \{min, max\}$ are the boundaries on the amplitude and rate of the input, respectively.

$J_{obst,l}$ is defined as the cost of obstacle avoidance and it is assumed that positions of the surrounding obstacles can be precisely detected and be predicted during the prediction horizon H_p. The rectangular shape of the ego vehicle can be approximated by three circle disks D_1, D_2, D_3 (Gao et al. (2011)). $O_{l,n}, n = 1, .., N$ are the sampling points on the contour of the obstacle l and $d_{l,n}$ are the distances between the front circle disk center D_1 and sampling points $O_{l,n}$. $p_{l,n}$ are defined as:

$$p_{l,n} = \begin{cases} 0 & if \quad d_{l,n} \le a + R_{disk} \\ d_{l,n} - R_{disk} & if \quad d_{l,n} > a + R_{disk} \end{cases} \tag{16}$$

where R_{disk} is the radius of each circle disk and a is the mimimun safety distance. $p_{l,min}(k+i|i) = \min_n p_{l,n}(k+i|k)$ denotes the ith predicted minimum distance between the front of the front disk D_1 and sampling points $O_{l,n}$ at time step k. $J_{obst,l}$ is determined by the distance between the front of the vehicle and the obstacle:

$$J_{obst,l} = \sum_{i=1}^{H_p} \frac{S_{obs} v}{p_{l,min}(k+i|k) + \iota} \tag{17}$$

where v is the velocity of the vehicle, ι is a small positive number and S_{obs} is the collision avoidance weight.

By solving the optimization problem (15), the optimal control sequence $\Delta\mathbf{u}$ as well as the predicted pose $X(k+i|k)$ are obtained. Because of different sampling time values between the upper planner and lower controller, the predicted pose $X(k+i|k)$ cannot be used in the lower controller directly. Quartic polynomials are used to fit the discrete predicted pose $X(k+i|k)$:

$$y_l(x_o) = a_0 x_o^4 + a_1 x_o^3 + a_2 x_o^2 + a_3 x_o + a_4 \quad (18)$$

$$\psi_l(x_o) = b_0 x_o^4 + b_1 x_o^3 + b_2 x_o^2 + b_3 x_o + b_4 \quad (19)$$

where $a_{0,\ ...,4}$ and $b_{0,\ ...,4}$ are the corresponding coefficients. Therefore, the local reference path $\eta_r(x_o)$ is obtained.

Remark 2. During the prediction horizon, the speed of the ego car is considered the same as that measured at time k in the planner design.

3.2 Velocity validation

The local reference path $\eta_r(x_o)$ only contains the future target waypoints without speed regulation. A velocity validation module is introduced to guarantee safe maneuver. Taking account of the current speed v and the desired speed v_d, the maximum speed variation Δv_{max} in the prediction horizon as well as the maximum curvature of the local path κ_{max}, the reference speed v_r is given as follows (Polack et al. (2018)):

$$v_r = \min(\sqrt{\frac{0.5\mu g}{\kappa_{max}}}, v_d, v + \Delta v_{max}) \quad (20)$$

Combining the planned local reference path $\eta_r(x_o)$ and reference speed v_r, the local reference trajectory $\eta_r(t)$ is planned.

4. ROBUST LOW-LEVEL CONTROLLER FOR TRAJECTORY TRACKING

The low-level controller is designed to track the reference trajectory $\eta_r(t)$ computed by the high-level motion planner. Longitudinal and lateral controller are designed separately. For the longitudinal velocity tracking, a simple PID controller is determined by the velocity error $e(t) \triangleq v(t) - v_r(t)$:

$$T_{xf} = T_{xr} = -K_p e(t) - K_d \dot{e}(t) - K_i \int_0^t e(\tau) d\tau \quad (21)$$

where K_p, K_d, K_i are the constant gains of the PID controller, and T_{xf}, T_{xr} are the torque demands on the front and rear wheels. For the lateral position tracking, a robust MPC controller design is proposed in the following

4.1 Disturbance Analysis and Controller Architecture

The vehicle lateral dynamic model (13) is established under the assumptions of small slip angles, no external disturbances, and the deterministic transmission delay τ_k. However, both unmodelled vehicle dynamics as well as unknown time delays exist in the actual vehicle system. The nominal value of a variable is denoted in its overline format and the deviation from the actual value is expressed in its tilde format as follows:

$$\xi_a(k) = \tilde{\xi}_a(k) + \bar{\xi}_a(k), \quad (22)$$

$$\Delta u(k) = \Delta\tilde{u}(k) + \Delta\bar{u}(k), \quad (23)$$

$$\tau_k = \tilde{\tau}_k + \bar{\tau}_k \quad (24)$$

The dynamics of the nominal, error, and actual systems can then be expressed respectively as follows:

$$\bar{\xi}_a(k+1) = A(k)\bar{\xi}_a(k) + B(k)\Delta\bar{u}(k) + D(k) \quad (25)$$

$$\tilde{\xi}_a(k+1) = A(k)\tilde{\xi}_a(k) + B(k)\Delta\tilde{u}(k) + D(k) + \omega(k) \quad (26)$$

$$\xi_a(k+1) = A(k)\xi_a(k) + B(k)\Delta u(k) + D(k) + \omega(k) \quad (27)$$

where $\omega(k) \in \mathbb{W} = \left\{\omega(k) \in \mathbb{R}^5 |\ \|\omega(k)\|_\infty \le c_\omega\right\}$ denotes the bounded disturbance.

A robust nonlinear predictive controller is designed to handle the impact of the disturbance $\omega(k)$, caused by unmodeled dynamics as well as the time delay error $\tilde{\tau}_k$. The controller is composed of a nominal MPC controller along with linear error feedback law. The nominal MPC controller input $\Delta\bar{u}(k)$ and error feedback input $\Delta\tilde{u}(k)$ are designed based on nominal system (25) and error system (26), respectively, as to be detailed in the next.

4.2 Nominal Model Predictive Control

In this part, we adopt the MPC method to track the planned trajectory computed by the planning module. We obtain an optimal input sequence at each sampling instant by solving a constrained finite-time optimal control problem. The computed optimal input sequence as well as the predicted nominal vehicle states at time step k are stored as input and state trajectories, $\Delta\bar{u}_k(k+i)$ and $\bar{\xi}_{a,k}(k+i)$. Only the first control input is applied to the vehicle system and at the next time step, this optimization routine is repeated with new state measurements.

Constraints: The main objective of the controller is to track the desired lateral position y_r as well as the desired yaw angle ψ_r. We define the safe constraints as follows:

$$\begin{aligned} y_{o,min} \le Y_o(k+i) \le y_{o,max} \\ \psi_{min} \le \psi(k+i) \le \psi_{max}, \quad i = 1...H_p \end{aligned} \quad (28)$$

where $y_{o,min}$ and $y_{o,max}$ denote the upper and lower bounds on the lateral positions. The safety constraints can be rewritten in the compact form as:

$$h^\xi(\xi(k+i+1), u(k+i)) \le \mathbf{0}, \quad i = 1...H_p \quad (29)$$

Apart from safety constraints, there exist amplitude and rate limits on the actuator input $u = \delta_f$:

$$\begin{aligned} \delta_{f,min} \le u(k+i) \le \delta_{f,max} \\ \Delta\delta_{f,min} \le \Delta u(k+i) \le \Delta\delta_{f,max}, \quad i = 1...H_c \end{aligned} \quad (30)$$

The input constraints can be rewritten in the compact form as:

$$h^u(u(k+i), \Delta u(k+i)) \le \mathbf{0}, \quad i = 1...H_c \quad (31)$$

Optimal Control Problem: Consider system (13), we formulate the CFTOC problem as:

$$\min_{\Delta\bar{U},\varepsilon} \quad \sum_{i=1}^{H_p} \|\bar{\eta}(k+i|k) - \eta_r\|_Q^2 + \sum_{j=1}^{H_c} \|\Delta\bar{u}(k+j|k)\|_R^2 + \rho\varepsilon^2 \quad (32a)$$

$$\begin{aligned} s.t.\ \bar{\xi}_a(i+1|k) =& A(k+i|k)\bar{\xi}_a(i|k) \\ &+ B(k+i|k)\Delta\bar{u}(i|k) + D(k) \end{aligned} \quad (32b)$$

$$\bar{\eta}(i+1|k) = C\bar{\xi}_a(i+1|k) \quad (32c)$$

$$h^{\bar{\xi}}(\bar{\xi}(i+1|k), \bar{u}(i|k)) \le \varepsilon\mathbf{1}, \quad \varepsilon \ge 0 \quad (32d)$$

$$h^{\bar{u}}(\bar{u}(i|k), \Delta\bar{u}(i|k)) \le \mathbf{0}, i = k, ..., k+H_c \quad (32e)$$

$$\bar{\xi}_a(k|k) = \bar{\xi}_a(k) = \xi_a(k) \quad (32f)$$

where $\bar{\xi}_a(k+i|k)$ and $\bar{\eta}(k+i|k)$ denote the predicted nominal states and outputs at the $k+i$th step at

time step k by applying the control sequence $\Delta\bar{\mathbf{u}} \triangleq \{\Delta\bar{u}(k|k), ..., \Delta\bar{u}(k+i|k)\}$ to the discrete-time dynamics (32b). The safety constraints (32d) are imposed as soft constraints, by introducing the slack variable ε in (32a). Q, R and ρ are weights of proper dimension for penalizing tracking error, control action and violation of the soft constraints, respectively. The initial nominal state value $\bar{\xi}_a(k)$ is set to be the same as the measured current state value $\xi_a(k)$ at each time instant k.

4.3 Feedback Control Design

Consider error system (26), we choose the infinite horizon linear quadratic regulator gain K^{∞}_{LQR} for system (A, B) as the stabilizing state feedback gain K. Therefore, the error and actual control inputs are denoted as follows:

$$\Delta\tilde{u}(k) = K^{\infty}_{LQR}(\xi_a(k) - \bar{\xi}_a(k|k-1)) \tag{33}$$

$$\Delta u(k) = \begin{cases} \Delta\tilde{\delta}_{f,min} + \Delta\bar{u}(k), \ if \ \Delta\tilde{u}(k) < \Delta\tilde{\delta}_{f,min} \\ \Delta\tilde{u}(k) + \Delta\bar{u}(k), \ if |\Delta\tilde{u}(k)| \leq \tilde{\delta}_{f,max} \\ \Delta\tilde{\delta}_{f,max} + \Delta\bar{u}(k), \ if \ \Delta\tilde{u}(k) > \Delta\tilde{\delta}_{f,max} \end{cases} \tag{34}$$

where $\Delta\tilde{\delta}_{f,min}$ and $\Delta\tilde{\delta}_{f,max}$ denote the upper and lower bounds on the error feedback input $\Delta\tilde{u}(k)$. Considering the limit of computing resources, the feedback control is only activated when the following event condition is satisfied:

$$W \cdot \bar{\xi}_a(k) \geq \lambda > 0 \tag{35}$$

where $W \triangleq [w_1 \ w_2 \ w_3 \ w_4 \ w_5]$ is the parameter vector, whose elements are all constant and positive, and $\lambda > 0$ is the threshold parameter. The scale of $w_i, \ i = 1, 2, 3, 4, 5$ is determined by the corresponding prediction error scale.

5. SIMULATION RESULTS

In this section, simulation results are provided to demonstrate the effectiveness of the proposed algorithm. The control algorithm development, the vehicle dynamic simulation and driving scenario description are implemented at MATLAB R2018b with Intel i7-7700HQ and 8GB RAM. Utilizing Vehicle Dynamics Blockset in MATLAB, a 7 Degrees of Freedom (DOF) passenger vehicle dynamics model is established to simulate the dynamics of the vehicle (Pacejka (2005)). In the 7 DOF, 3 DOF corresponds to the vehicle body (x, y, ψ) and 4 DOF corresponds to the wheels $(\omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr})$.

Parameters of the vehicle in the simulation are listed in Tables 1. Parameters of upper and lower controllers are mainly referred from Gao et al. (2011). In the simulation, the sampling time and the average computation time of the lower motion controller are 50ms and 8ms, respectively. And those of the upper path planner are 100ms and 12ms, respectively. The upper bound of the uncertain network-induced delay is 50 ms and the nominal value $\bar{\tau}$ is 30 ms. The Pacejka Magic Formula is used for high-fidelity vehicle simulation models (Pacejka (2005)).

Scenario with static and dynamic obstacles: A dynamic driving scenario is designed to validate the effectiveness of the proposed algorithm.

Table 1. Vehicle and road parameters.

Parameters (units)	Value	Parameters (units)	Value
$m \ (kg)$	1723	$I_z \ (kg \cdot m^2)$	4175
$l_f \ (m)$	1.232	$l_r \ (m)$	1.468
$C_f \ (N \cdot rad^{-1})$	66900	$C_r \ (N \cdot rad^{-1})$	62700
μ	0.8	$R_{disk} \ (m)$	1.2

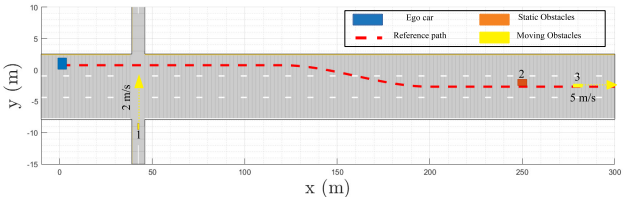
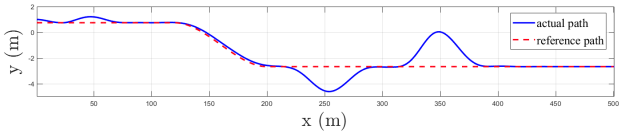
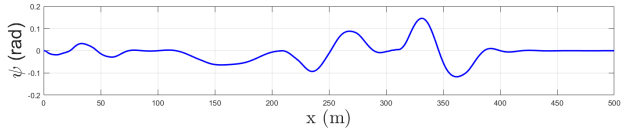


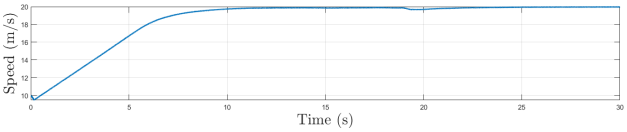
Fig. 3. A scenario with static and dynamic obstacles.



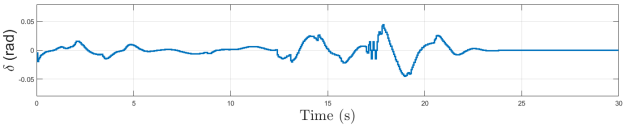
(a) lateral position



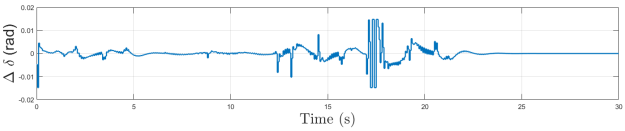
(b) yaw angle



(c) velocity



(d) front steering angle



(e) front steering angle increment

Fig. 4. Simulation results in the driving environment.

As shown in Fig. 3, the ego car travels on a three-lane road in the dynamic environment with both static and dynamic obstacles. The ego car is expected to maneuver from the left lane to the middle lane at a speed of 20 m/s. Obstacle 1 is moving along the y direction at a speed of 2 m/s, obstacle 2 is a static barrier in the middle lane and obstacle 3 is moving along the x direction at a speed of 5 m/s. Fig. 4 (a)(b)(c) show the lateral position, the yaw angle and the speed of the ego vehicle, respectively. Fig. 4(d)(e) show the front steering angle input and its increment trajectories. Fig. 5 shows the snapshots of the simulation results at critical time steps. At 3.2s, the ego car turns left to avoid collision with obstacle 1 and the local planned path is

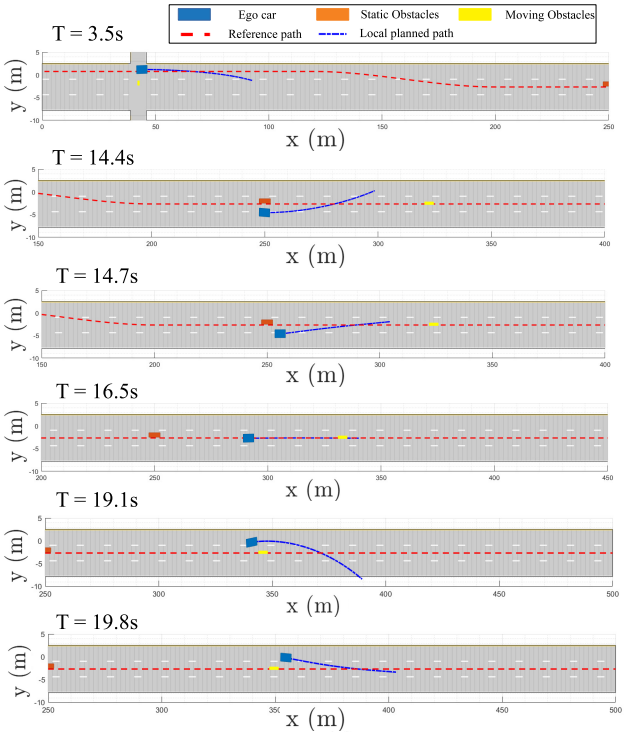


Fig. 5. Snapshots of the simulation with ego car and other obstacles in a three-lane road.

expected to follow the reference path afterwards. At time 14.4s and 14.7s, the ego car takes an obstacle-avoidance maneuver and tracks the desired path at time 16.5s. At 19.1s and 19.8s, the ego car overtakes obstacle 3 and then gets back to the middle lane. In general, the ego car travels smoothly to follow the reference trajectory, while avoiding collisions with other obstacles.

6. CONCLUSION

This paper proposes a robust control framework for trajectory tracking and obstacle avoidance. The framework formulates the problem as two nonlinear MPC problems. A kinematic bicycle vehicle model and an augmented dynamic vehicle model are developed and used in the controller design. At the upper-level, local trajectories are generated by solving a constrained finite-time optimal problem. At the lower-level, a nominal nonlinear predictive controller along with a linear error feedback law is introduced to improve the tracking performance in the presence of uncertainties. Simulation results in different driving scenarios verify that the hierarchical controller can guarantee safe and smooth vehicle maneuvers in the presence of uncertainties.

REFERENCES

Caruntu, C.F., Lazar, M., Gielen, R.H., Van den Bosch, P., and Di Cairano, S. (2013). Lyapunov based predictive control of vehicle drivetrains over can. *Control Engineering Practice*, 21(12), 1884–1898.

Gao, Y., Gray, A., Tseng, H.E., and Borrelli, F. (2014). A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6), 802–823.

Gao, Y., Lin, T., Borrelli, F., Tseng, E., and Hrovat, D. (2011). Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In *Proceedings of the ASME 2010 dynamic systems and control conference*, 265–272. Cambridge, Massachusetts, United states.

Karaman, S., Walter, M.R., Perez, A., Frazzoli, E., and Teller, S. (2011). Anytime motion planning using the rrt*. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, 1478–1483. Shanghai, China.

Li, S., Li, K., Rajamani, R., and Wang, J. (2010). Model predictive multi-objective vehicular adaptive cruise control. *IEEE Transactions on Control Systems Technology*, 19(3), 556–566.

Li, X., Sun, Z., Cao, D., Liu, D., and He, H. (2017). Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mechanical Systems and Signal Processing*, 87, 118–137.

Lim, W., Lee, S., Sunwoo, M., and Jo, K. (2018). Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. *IEEE Transactions on Intelligent Transportation Systems*, 19(2), 613–626.

Liu, Q., Liu, Y., Liu, C., Chen, B., Zhang, W., Li, L., and Ji, X. (2018). Hierarchical lateral control scheme for autonomous vehicle with uneven time delays induced by vision sensors. *Sensors*, 18(8), 2544.

Ma, L., Xue, J., Kawabata, K., Zhu, J., Ma, C., and Zheng, N. (2015). Efficient sampling-based motion planning for on-road autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 1961–1976.

Ma, Y., Chen, J., Zhu, X., and Xu, Y. (2019). Lateral stability integrated with energy efficiency control for electric vehicles. *Mechanical Systems and Signal Processing*, 127, 1–15.

Pacejka, H. (2005). *Tire and Vehicle Dynamics*. Elsevier, Oxford, United Kingdom.

Paden, B., Čáp, M., Yong, S.Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Iintelligent Vehicles*, 1(1), 33–55.

Polack, P., Altché, F., d'Andréa Novel, B., and de La Fortelle, A. (2018). Guaranteeing consistency in a motion planning and control architecture using a kinematic bicycle model. In *Proceedings of the American Control Conference*, 3981–3987. Milwauke, Wisconsin, United states.

Shuai, Z., Zhang, H., Wang, J., Li, J., and Ouyang, M. (2013). Combined afs and dyc control of four-wheel-independent-drive electric vehicles over can network with time-varying delays. *IEEE Transactions on Vehicular Technology*, 63(2), 591–602.

Zhu, X., Zhang, H., Wang, J., and Fang, Z. (2014). Robust lateral motion control of electric ground vehicles with random network-induced delays. *IEEE Transactions on Vehicular Technology*, 64(11), 4985–4995.

Ziegler, J., Bender, P., Dang, T., and Stiller, C. (2014). Trajectory planning for berthaa local, continuous method. In *Proceedings of the 2014 IEEE intelligent vehicles symposium proceedings*, 450–457. Dearborn, Michigan, United states.