2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

A hierarchical control system for smart parking lots with automated vehicles: Improve efficiency by leveraging prediction of human drivers*

Yuchao Li, Karl H. Johansson and Jonas Mårtensson

Abstract— **In this work, we introduce a hierarchical architecture for management of multiple automated vehicles in a parking lot provided the existence of human-driven vehicles. The proposed architecture consists of three layers: behavior prediction, vehicle coordination and maneuver control, with the first two sitting in the infrastructure and the third one equipped on individual vehicles. We assume all three layers share a consistent view of the environment by considering it as a grid world. The grid occupancy is modeled by the prediction layer via collecting information from automated vehicles and predicting human-driven vehicles. The coordination layer assigns parking spots and grants permissions for vehicles to move. The vehicle control embraces the distributed model predictive control (MPC) technique to resolve local conflicts occurred due to the simplified vehicle models used in the design of the prediction and coordination layers. Numerical evaluation shows the effectiveness of the proposed control system.**

I. INTRODUCTION

Parking a car is no simple task. According to [1], British and German drivers wasted in average 41 and 44 hours in searching for parking spots in 2017, at an estimated cost of £23.3 billion and €40.4 billion, respectively. Since nearly 75% of parking spots in Europe are located in off-road parking lots [2], such as multilevel structures and underground car parks, improved management system for the parking lots would certainly contribute to overall efficiency of the parking procedure.

The good news is that the advances of sensor technologies have been exploited to facilitate the management system in those scenarios. For example, a combination of magnetic and ultrasonic sensors were used for detection of vehicles in a parking structure inside the campus of the University of Southern California [3]. In recent years, with improvement of image processing power, camera-based methods have gained great popularity [4]. Along with research publications, many relevant patents have been issued, [5], [6] to name a few. Most of aforementioned approaches have been applied to guide the drivers to park with the acquired information. In industry, a few companies have followed the path and taken one step further. Hikvision, a supplier of video surveillance products, has demonstrated their vision for improving the efficiency by introducing moving robots as carriers to move vehicles in a parking lot with a guidance system where the camera-based technique plays a key role [7].

On the other hand, the joint efforts from academia and industry in pursuing automated vehicles may provide new potential for parking lot management. Apart from general techniques for urban driving, some research has been specifically focusing on planning and control for parking procedures with emphasis on maneuver accuracy, such as [8], [9]. Meanwhile, many automobile manufacturers have started to provide automated parking systems in their commercially available products, such as the Remote Parking Assist from Mercedes [10] and Remote Control Parking from BMW [11], both enabling automated parking procedures even after drivers have exited the cars. Besides, emerging technologies related to distributed optimization have also been applied to automated vehicles to exploit the information shared between each other for better performance [12]. Combining all those technologies, fully automated vehicle for driving, maneuvering and interacting in the parking lot can be expected in the near future.

Despite the great success in those two research fields, little, if any, effort has been made to integrate the rich sensory information from the management level with the automated vehicles to explore the potential benefits brought to both the management system and the individual vehicles [13]. Besides, the estimation of human-driven vehicles, which traditionally has been designed for individual automated vehicles in the context of road networks [14], can now be performed by the management level with much more data and much higher accuracy, due to more available sensors and driver guidance systems.

In this paper, we provide a systematic approach to address the challenges for management of parking lots in the dawn of automated driving. The main contributions are:

1) A control architecture for handling mixed automated and human-driven vehicles in smart parking lots, consisting of: i) a prediction layer for predicting vehicle positions and grid occupancy, ii) a coordination layer for granting permission and initialization of cooperation, and iii) a vehicle layer with distributed planning and tracking functionality.
2) By means of numerical evaluation: i) the identification of potential problems faced by the management systems without human driver prediction, and ii) justification that the proposed control architecture would resolve potential issues (e.g., deadlocks) caused by human drivers.

An additional contribution is the design of the vehicle layer

This work was supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation and the Swedish Foundation for Strategic Research.

Y. Li, K. H. Johansson and J. Mårtensson are with Integrated Transport Research Lab, the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. {yuchao, kallej, jonas1}@kth.se.

as a cooperative MPC that accounts for predicted paths of any human driven vehicle.

The rest of this paper is organized as follows. Section II presents the overview of the proposed controller. The details of the functionality needed in each automated vehicle are introduced in Section III, whereas the coordination and prediction layers are introduced in Sections IV and V respectively. The numerical evaluation for the proposed control, in scenarios with and without human driver prediction, are shown in Section VI. Section VII concludes the paper.

II. CONTROLLER OVERVIEW

The control system discussed in this paper is designed to manage multiple automated vehicles in a parking lot provided the presence of human-driven vehicles. The components of the controller can be physically-located in the infrastructure or in each automated vehicle and those that reside in the infrastructure will be named *authority* herein. As suggested in Section I, with the guidance system for the drivers and sensors for position detection, it is reasonable to assume: 1) automated and human-driven vehicles can be distinguished by the authority; 2) all vehicles would move only after permission granted and would park to the assigned spots; 3) all vehicle positions can be detected by the authority and planning and control information of automated vehicles are broadcast by them; and 4) all automated vehicle share a consistent view with the authority as regards to viewing the enclosed environment as a grid world. For any vehicle in the parking lot, we also categorize them as: *active*, meaning moving; *parked*, as the name suggests, and *on-hold*, meaning waiting for the permission from the authority at the entrance, or at the parking spot, all applicable to both automated and human-driven vehicles. Based on the assumed scope, the proposed layered architecture is shown in Fig. 1, composed of vehicle layer, coordination layer, and prediction layer.

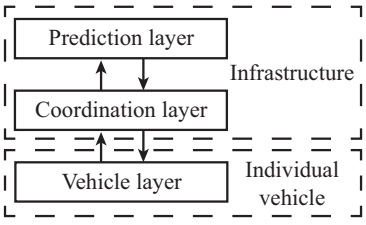


Fig. 1: Layered control architecture for management of vehicles in an enclosed environment. Those layers and their communication interfaces are detailed in Section III, IV and V respectively.

Specifically, the vehicle layer is composed of planning and control functionality for individual automated vehicles, which construct their own feasible paths over the access-granted area from the coordinator, and track the paths in a distributed manner. The control layer also contains a distributed MPC which is applied to resolve local conflicts when the risk of collision arises.

The aim of the coordination layer is twofold. First, it grants permission to enter and to leave, assigns parking spots to vehicles, and grants access to certain areas for automated vehicles to travel over at the same time, based upon current conditions. Second, it closely monitors the situation and initiates cooperation between vehicles if needed.

The prediction layer also has two functions. First, it models a human-driven vehicle by a path planner with a tracking MPC to predict its trajectory to assist cooperation. Second, it updates the grid occupancy conditions based on information from the coordination layer and the vehicle layer, both of which are viewed by the prediction layer as inputs for updating the grid condition. For automated vehicles, its predicted path can be directly obtained from the relevant individual, whereas for human-driven vehicles, it includes all the surrounding grids as positions that a driver may intend to drive to.

III. VEHICLE LAYER

A. Vehicle dynamics

The vehicle dynamics is described by a kinematic bicycle model. Denote $(x,\, y)$ as the vehicle coordinate, v as the vehicle speed, ψ as the heading angle, all measured in the same Cartesian global coordinate system, β as the angle of the vehicle velocity with respect to the longitudinal axis of the car, a as the acceleration along the same direction, and l_r as the distance from the center of mass of the vehicle to the rear axle. The model is given by

$$\dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)), \tag{1a}$$

$$\dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)), \tag{1b}$$

$$\dot{v}(t) = a(t), \tag{1c}$$

$$\dot{\psi}(t) = \frac{v(t)}{l_r}\sin(\beta(t)), \tag{1d}$$

and we apply a and β as control inputs. The constraints imposed on the system are

$$|a(t)| \leq a_{max},\ |\beta(t)| \leq \beta_{max},\ |\dot{\beta}(t)| \leq \dot{\beta}_{max}.$$

B. Communication interface with coordination layer

Consider enclosed environment composed of grids which form a set $\mathcal{G}$. At a certain instance t a set $\mathcal{N} = \{1,\, 2, ..., n\}$ of automated vehicles are active inside. Due to the assumptions introduced in Section II, all those vehicles broadcast their current states, planned paths, predicted paths, and all of them have a consistent view of the environment and therefore their dynamics can be described by the coordinates of the same coordinate frame. Denote $\mathbf{z} = [x\ \ y\ \ \psi]^T$ as the state vector, $\mathbf{Z}_r = [\mathbf{z}_{r,0}\ \ \mathbf{z}_{r,1}\ \ ...\ \ \mathbf{z}_{r,N_z}]$ and $\mathbf{v}_r(t) = [v_{r,0}\ \ v_{r,1}\ \ ...\ \ v_{r,N_v}]^T$ as the reference from the planner at time t with length N_z and N_v respectively, $\mathbf{u}_0^* = [a_0^*\ \ \beta_0^*]^T$ as the control law calculated by longitudinal and lateral controllers. Denote the predicted trajectory from the MPC controller as $\tilde{\mathbf{Z}}$. For an automated vehicle k, denote its access-granted area as $G \subseteq \mathcal{G}$, the suggested velocity is v_r and γ is cooperation initialization signal, which will be introduced in Section IV. We use superscript k to indicate which vehicle the variable is related to. Note that G^k is only sent to k when the permission is granted in the first time. While a vehicle is active, coordinator does not repeat G^k

to the same vehicle. For human-driven vehicles, the only information available is its current location, which is detected by the authority. Then the assumed communication interface between active vehicles and the coordination layer is given in Fig. 2. The dashed line from the human-driven vehicle indicates its position $\mathbf{z}^{hm}(t)$ is detected by, not reported to, the authority and we use superscript hm to indicate human-driven vehicles. The information flow between the prediction layer and the coordination layer is explained in Section IV.

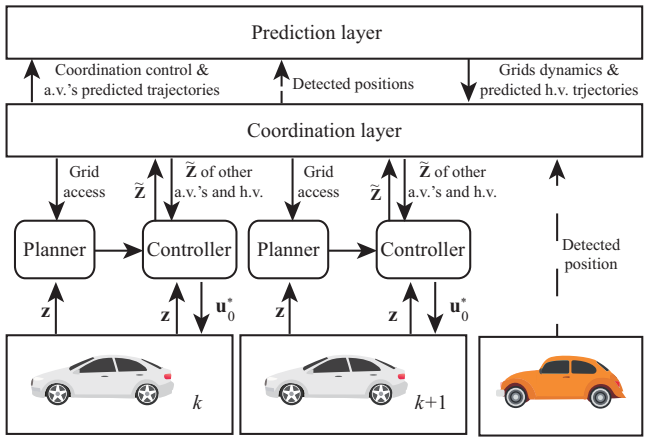


Fig. 2: Communication interface between different layers. The automated vehicles (a.v.'s) are depicted as white cars while the human-driven vehicle (h.v.) is the orange one. Note that all the signals appeared are function of time and notation (t) has been dropped.

C. Path planning and nominal tracking

Upon receiving the access-granted grid numbers G^k at the entrance, or at its parking spot, automated vehicle k constructs its own feasible path. One example is shown in Fig. 3, where on the left the access-granted area is marked as green, and the red line on the right plot is one path $\mathbf{Z}_r^k$ from the planner.

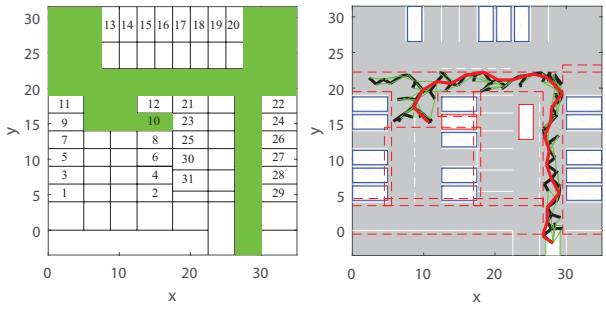


Fig. 3: Access granted grids (marked green on the left) with grid code names in G^k and path planning by vehicle k. The names of parking spots are shown in the left plot.

For this study, each vehicle needs to have two different steering controllers: one for the case when the cooperative vehicles are far away, which we call nominal MPC; the other one for the case when those vehicles are close and information from control layer is shared to perform collision avoidance, which we call cooperative MPC. We restrict our attention to the lateral control for collision avoidance, and the longitudinal control therefore is not presented here.

When vehicles are far away from each other, the primary target of the controller is to track the reference signals $\mathbf{Z}_r$ and $\mathbf{v}_r$ given by the planners. Therefore, at each sampling time t_s, the nominal MPC is a constrained quadratic programming problem given by

$$\min_{\mathbf{b}} \quad \sum_{i=0}^{N}(\|\mathbf{z}_i - \mathbf{z}_{r,i}\|_Q + \|\beta_i\|_R) + \|\mathbf{z}_{N+1} - \mathbf{z}_{r,N+1}\|_{Q_f} \tag{2a}$$

$$\text{s.t.} \quad \mathbf{z}_0 = \mathbf{z}(t),\ \beta_{-1} = \beta(t - t_s), \tag{2b}$$

$$\mathbf{z}_{i+1} = A_i\mathbf{z}_i + B_i\beta_i + \mathbf{f}_i,\ \forall i, \tag{2c}$$

$$|\beta_i| \le \beta_{max},\ \forall i, \tag{2d}$$

$$|\frac{\beta_i - \beta_{i-1}}{t_s}| \le \dot{\beta}_{max},\ \forall i, \tag{2e}$$

where $Q \in \mathbb{R}_+^{3\times3}$ and is diagonal, $R \in \mathbb{R}_+$, $Q_f \in \mathbb{R}_+^{3\times3}$, N is the prediction horizon, $\mathbf{b} = [\beta_0\ \beta_1\ ...\ \beta_N]^T$. $A_i \in \mathbb{R}^{3\times3}$, $B_i \in \mathbb{R}^{3\times1}$ and $\mathbf{f}_i \in \mathbb{R}^{3\times1}$ are obtained by linearizing (1) along $\mathbf{v}_r$, $[\psi_{r,0}\ \psi_{r,1}\ ...\ \psi_{r,N}]$ and $\beta_i = 0$ and then discretizing it through forward Euler. The presence of $\mathbf{f}_i$ is due to that the reference control signal is not given for tracking the reference position and the states here are not tracking errors. Once the optimization problem is solved and $\mathbf{b}^*$ is obtained, the control law implemented is β_0^*. Here shared information between cooperative vehicles would be disregard as tracking the reference is the primary goal.

D. Cooperative MPC

Once some individual vehicle k gets a cooperation signal, $\gamma^k = 1$, the primary goal is to avoid collision. The shared information between cooperative vehicles can be used to form additional constraints or to create a collective cost function. Here we present an approach where the vehicles are coupled by cost function. By doing so, the feasibility can be easily ensured compared to using dynamic constraints. Note that the dynamics of the vehicles are fully decoupled. Denote the set of cars engaging cooperation as $\mathcal{C} \subseteq \mathcal{N}$ and suppose $k \in \mathcal{C}$. Denote $\mathcal{C}^k = \mathcal{C} \setminus \{k\}$. Suppose $\mathcal{C} = \{c_1, c_2, ..., c_{|\mathcal{C}|}\}$ where $|\cdot|$ denotes set cardinality, and $k = c_q$. Then we have $\mathcal{C}^k = \{c_1, c_2, ..., c_{q-1}, c_{q+1}, ..., c_{|\mathcal{C}|}\}$. Suppose the sampling time instance when the vehicle starts cooperation is $t = jt_s$. The object of the centralized MPC for the overall system containing those $|\mathcal{C}|$ vehicles is formulated as

$$\min_{\mathbf{U}} \quad \sum_{k\in\mathcal{C}}\Big(\sum_{i=0}^{N_c}\big(\|\mathbf{z}_i^k - \mathbf{z}_{r,i}^k\|_{Q^k} + \|\beta_i^k\|_{R^k} + \sum_{p\in\mathcal{C}^k}\phi_{k,p}(\mathbf{z}_i^k, \mathbf{z}_i^p)\big) + \|\mathbf{z}_{N_c+1}^k - \mathbf{z}_{r,N_c+1}^k\|_{Q_f^k}\Big) \tag{3}$$

where $\mathbf{U} = [\mathbf{b}^{c_1}\ \mathbf{b}^{c_2}\ ...\ \mathbf{b}^{c_{|\mathcal{C}|}}]$, k indexes the vehicles, N_c is the prediction horizon for cooperation, $\phi_{k,p} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}_+$ is some quadratic or linear function indicating the vehicle

distances between each other. The optimal control law can be obtained by solving such a problem once every sampling time.

However, solving the optimization problem for multiple vehicles in real time is difficult. Luckily, the optimal control law can still be obtained by solving the optimization problem (3) in a distributed and iterative manner. Assume the dynamics of all vehicles can be captured by (1) and each vehicle communicates its local optimal control law $\mathbf{b}^{k*}$ and its current state. For vehicle k, its local optimization objective is given by

$$\min_{\mathbf{b}^k} \; V^k := \sum_{i=0}^{N_c} \big(\|\mathbf{z}_i^k - \mathbf{z}_{r,i}^k\|_{Q^k} + \|\beta_i^k\|_{R^k} + \sum_{h\in\mathcal{C}} \sum_{p\in\mathcal{C}^h} \phi_{h,p}(\mathbf{z}_i^h, \mathbf{z}_i^p) \big) + \|\mathbf{z}_{N_c+1}^k - \mathbf{z}_{r,N_c+1}^k\|_{Q_f^k}, \tag{4}$$

Denote the computation and communication time to solve (4) for one iteration as Δt_c. Then it is guaranteed to recover the global optimal solution of (3) by solving (4) when $\Delta t_c \ll t_s$ under some other conditions if the constraints are properly included [15]. Those constraints are introduced later. The optimal solution is recovered in an iterative manner, that is the same optimization objective (4) is solved several times during two sampling instants based on updated information from other vehicles.

We use subscript l with brackets to indicate which iteration the variables are associated to. Define the optimal solution of (4) at iteration l as

$$\mathbf{b}_{(l)}^{k*} := \arg\min_{\mathbf{b}^k} V_{(l)}^k. \tag{5}$$

Here we introduce two approaches to recover the optimal solution via distributed MPC depending on whether the models or the predictions are shared.

1) Model shared approach: To begin with, at the instance of any sampling time $t = mt_s$ where $m \in \mathbb{N}_+$ during the course of cooperation, vehicle k broadcasts its current state $\mathbf{z}^k(t)$. In addition, at the instance of the sampling time $t = jt_s$ when the cooperation starts, each vehicle also broadcasts its $v^k(jt_s)$, $\psi^k(jt_s)$ and $\beta^k(jt_s)$ to others such that the dynamics of vehicle $p \in \mathcal{C}^k$, namely A_j^p, B_j^p and $\mathbf{f}_j^p$, are also available locally for vehicle k. At each sampling instant, $l = 0$, $\mathbf{U}_{(0)}$ is the same for all the vehicles and can be available locally. Then the local optimization objects Eq. (4) can be minimized to get $\mathbf{b}_{(0)}^{k*}$ subject to certain constraints. After $\mathbf{b}_{(l)}^{k*}$ is obtained, the optimal control law can be updated locally by

$$\mathbf{U}_{(l+1)} = \sum_{k\in\mathcal{C}} \omega^k \tilde{\mathbf{U}}_{(l)}^k \tag{6}$$

where $\sum_{k\in\mathcal{C}} \omega^k = 1$ are predefined weights and

$$\tilde{\mathbf{U}}_{(l)}^k = \Big[\mathbf{U}_{(l)}[1] \; ... \; \mathbf{U}_{(l)}[q-1] \; \mathbf{b}_{(l)}^{k*} \; \mathbf{U}_{(l)}[q+1] \; ... \; \mathbf{U}_{(l)}[|\mathcal{C}|] \Big]$$

where $\mathbf{U}_{(l)}[q]$ denotes the q-th column vector of matrix $\mathbf{U}_{(l)}$. Denote the maximum number of iteration allowed as l_{max}, then the local optimization problem at time $t + l\Delta t_c$ where $0 \le l \le l_{max}$ is

$$\min_{\mathbf{b}_{(l)}^k} \; V_{(l)}^k \tag{7a}$$

$$\text{s.t. } \mathbf{z}_{0(l)}^k = \mathbf{z}^k(t),\; \beta_{-1(l)}^k = \beta^k(t - t_s), \tag{7b}$$

$$\mathbf{z}_{0(l)}^p = \mathbf{z}^p(t),\; \forall p \in \mathcal{C}^k, \tag{7c}$$

$$\mathbf{z}_{i+1(l)}^p = A_j^p \mathbf{z}_{i(l)}^p + B_j^p \beta_{i(l)} + \mathbf{f}_j^p,\; \forall i,\; \forall p \in \mathcal{C}, \tag{7d}$$

$$|\beta_i^k| \le \beta_{max}^k,\; \forall i, \tag{7e}$$

$$|\frac{\beta_i^k - \beta_{i-1}^k}{t_s}| \le \dot{\beta}_{max}^k,\; \forall i, \tag{7f}$$

$$\mathbf{b}_{(l)}^p = \mathbf{U}_{(l)}[d],\; \forall p = c_d \in \mathcal{C}^k, \tag{7g}$$

2) Path shared approach: Note that the model shared approach can be modified to be extended for the cases where the vehicles are heterogeneous, by which we mean that not all of their dynamics can be captured by (1) and one simple data package containing $v^k(jt_s)$, $\psi^k(jt_s)$, and $\beta^k(jt_s)$. Recall that we assume $k = c_q$, that is $k \in \mathcal{N}$ is q-th element of $\mathcal{C}$. Then $\mathbf{U}[q]$ is the control corresponding to vehicle k. The solution comes from the observation that

$$\mathbf{U}_{(l+1)}[q] = \omega^k \mathbf{b}_{(l)}^{k*} + (1 - \omega^k)\mathbf{U}_{(l)}[q] \tag{8}$$

due to that the weight ω^k, $k \in \mathcal{C}$ is predefined and remains unchanged in (6). That is, when $\mathbf{U}_{(l)}[q]$ is available and $\mathbf{b}_{(l)}^{k*}$ can be computed, $\mathbf{U}_{(l+1)}[q]$ can be updated locally in vehicle k. To compute $\mathbf{b}_{(l)}^{k*}$, vehicle k directly requires the predicted paths of vehicle p. At $t = mt_s + l\Delta t_c$ during the course of cooperation, assume $\mathbf{U}_{(l)}[q]$ has been obtained. Then vehicle k solves $\mathbf{b}_{(l)}^{k*}$ and computes $\mathbf{U}_{(l+1)}[q]$ by (8). Then it applies $\mathbf{U}_{(l+1)}[q]$ to (7d) to compute its predicted trajectory $\tilde{\mathbf{Z}}_{(l)}^k = [\tilde{\mathbf{z}}_{0(l)}^k \; \tilde{\mathbf{z}}_{1(l)}^k \; ... \; \tilde{\mathbf{z}}_{N_c(l)}^k]$ under such control and communicate $\tilde{\mathbf{Z}}_{(l)}^k$ to vehicle p $\forall p \in \mathcal{C}^k$ for iteration $l + 1$. By doing so, each iteration would require data packages of other vehicles' predicted path instead of their dynamics and the local optimization problem Eq. (7) for vehicle k is slightly modified as given in Eq. (9). However, the overall optimization problem remains the same and the properties of distributed MPC also hold for this approach. The communication data package shown in Fig. 2 is for the path-shared approach.

$$\min_{\mathbf{b}_{(l)}^k} \; V_{(l)}^k \tag{9a}$$

$$\text{s.t. } \mathbf{z}_{0(l)}^k = \mathbf{z}^k(t),\; \beta_{-1(l)}^k = \beta^k(t - t_s), \tag{9b}$$

$$\mathbf{z}_{i+1(l)}^k = A_j^k \mathbf{z}_{i(l)}^k + B_j^k \beta_{i(l)} + \mathbf{f}_j^k,\; \forall i, \tag{9c}$$

$$\mathbf{z}_{i(l)}^p = \tilde{\mathbf{z}}_{i(l)}^p,\; \forall p \in \mathcal{C}^k, \tag{9d}$$

$$|\beta_i^k| \le \beta_{max}^k,\; \forall i, \tag{9e}$$

$$|\frac{\beta_i^k - \beta_{i-1}^k}{t_s}| \le \dot{\beta}_{max}^k,\; \forall i, \tag{9f}$$

In case of cooperation with a human-driven vehicle, the predicted path $\tilde{\mathbf{Z}}^{hm}$ is directly given by the prediction layer. The details are given in Section V-A

IV. COORDINATION LAYER

The coordination layer has two main responsibilities: managing grid occupancy via assigning parking spots and granting access to automated vehicles, which we call resource allocation, and initiating cooperation among some vehicles if necessary, which we name as cooperation initiation.

A. Communication interface with the prediction layer

Denote $\mathbf{l} \in \{0, 1\}^{|\mathcal{G}|}$ to indicate grid occupancy and $\mathbf{l}[g]$ as the state of grid g. $\mathbf{l}[g] = 0$ means g is not assigned to any active vehicles and $\mathbf{l}[g] = 1$ otherwise. Denote the sampling time of the coordination and prediction layer as T_s and denote $\mathbf{n} \in \mathbb{Z}_+^{|\mathcal{G}|}$ as the number of cars planned to reach each grid at $t + T_s$ predicted at time t. Besides, we also denote $\Xi(t) \in 2^{\mathcal{G}}$ as the set of grids the coordinator grants permission at instance t, where $2^{\mathcal{G}}$ denotes the power set of $\mathcal{G}$. The predicted human-driven trajectory by the prediction layer is denoted as $\tilde{\mathbf{Z}}^{hm}$. The communication interface is also shown in Fig. 2.

B. Resource allocation

Denote $\mathcal{P} \subset \mathcal{G}$ as the set of all parking spots and $\mathcal{F}(t) \subseteq \mathcal{P}$ as the set of free parking spots at time t. Since the grid structure remains unchanged, $\forall g \in \mathcal{P}$, we can find $\tilde{G}_e(g)$ and $\tilde{G}_l(g)$ offline, which are the set of grids that a vehicle is allowed to travel to in order to reach the parking spot g and grids to leave from g respectively. Besides, $\forall g \in \mathcal{G}$, we can also find $O(g) \subseteq \mathcal{G}$ which is its neighbor grids. Define a label function $\lambda : \mathcal{G} \to \mathcal{N} \cup \{0\}$ as

$$\lambda(g) = \begin{cases} 0, & \text{no car active in grid } g, \\ k, & k \in \mathcal{N} \text{ active in grid } g. \end{cases}$$

Note that the label function bridges the grid name and car name at the current moment. Then the central task is to decide whether a vehicle is allowed to leave if it wants to, and which spot one should be assigned to if there is any free. We introduce the following rules based on heuristics but note that those rules can be formally verified.

For a vehicle parked at g to leave, we require $\forall g' \in O(g)$, $\mathbf{l}[g'] = 0$. Once the permission is granted, the active vehicle set $\mathcal{N}$ changed accordingly by the prediction layer and if we assume $\mathcal{N}(t) = \mathcal{N}(t - T_s) \cup \{|\mathcal{N}(t - T_s)| + 1\}$ with car $|\mathcal{N}(t)| = |\mathcal{N}(t - T_s)| + 1$ departing from g, then $\Xi(t) = \tilde{G}_l(g)$, $\lambda(g) := |\mathcal{N}(t)|$ and $\mathcal{F}(t) = \mathcal{F}(t - T_s) \cup \{g\}$. When $\mathcal{F}(t) \neq \emptyset$, namely there are parking spots available, the rule of assigning spot is $\Xi(t) = \tilde{G}_e\big(\arg\min \mathcal{F}(t)\big)$.

C. Cooperation initiation

Define the translation function $\tau : \mathbb{R}^2 \to \mathcal{G}$, which calculates its corresponding grid name for a coordinate (x, y). Then with a slight abuse of notation, we also write $\tau(\mathbf{z}) = \tau(x, y)$ where $\mathbf{z} = [x\, y\, \phi]$ and denote $\tilde{\mathbf{s}} = \tau(\tilde{\mathbf{Z}})$ as the vector of grid names of some predicted trajectory. Recall the definition of label function $\lambda(\cdot)$, then at current time t, if $\lambda(g) \neq 0$, then $g = \tau(\mathbf{z}^{\lambda(g)}(t))$ shall hold. Recall that $\mathbf{n} \in \mathbb{Z}_+^{|\mathcal{G}|}$ predicts in short time the number of vehicles planned to reach each grid. Then for any $g \in \mathcal{G}$, if $\mathbf{p}(g) > 1$, namely within short time $[t, t + T_s]$, at least two vehicles plan to reach g, then vehicle $\lambda(g') \in \mathcal{N}$ where $g' \in O(g)$ are notified by signal $\gamma^{\lambda(g')} = 1$ to perform cooperation.

V. PREDICTION LAYER

The prediction layer has two functionalities. First, it predicts the human-driven vehicle trajectory by modeling it as a vehicle with nominal tracking functionality. Second, it processes the information from the vehicle layer and the control signals from the coordination layer to update the grid occupancy by modeling it as a hybrid automaton.

A. Human-driven vehicle dynamics

Although the position of a human-driven vehicle $\mathbf{z}^{hm}(t)$ can be detected, its predicted trajectory can not be obtained from the human driver. The prediction layer models the dynamics of human-driven vehicle as (1) and predicts its trajectory within short time span. Due to the assumption in Section II that drivers would park to the assigned spot, the destination of human-driven vehicle is always known. Then the behavior of the driver is modeled by the path planning and nominal tracking functionality introduced in Section III-C. By doing so, we can obtain a predicted path $\tilde{\mathbf{Z}}^{hm}$, which can be used for individual automated vehicles form their collective cost functions in the cooperative MPC as if the human-driven vehicles are in cooperation.

B. Grid dynamics

To model the grid dynamics, we introduce a function $\epsilon : \mathcal{G} \times \mathcal{G} \times \mathcal{G} \to \{-1, 0\}$ as

$$\epsilon(g_1, g_2, g_3) = \begin{cases} -1, & \text{if } g_2 = g_3 \text{ and } g_1 \neq g_2, \\ 0, & \text{else.} \end{cases} \tag{10}$$

This function is used for calculating the grid occupancy. For example, for some $g_1 \in O(g_2)$, at time t, $\epsilon(g_1, \tau(\mathbf{z}^{\lambda(g_1)}(t - T_s)), g_2)$ is to check whether some car $\lambda(g_1) \in \mathcal{N}$ travels from grid g_2 to g_1 during time $[t - t_s, t]$. Due to such definition, once a vehicle is parked and is removed from set $\mathcal{N}$, the corresponding parking spot would remain assigned. Denote the indicator function for set $\mathcal{N}$ as $\chi_{\mathcal{N}}$. Recall that $\Xi(t)$ is the set of grids the coordinator grants permission to at instance t. Then $\chi_{\Xi(t)}$ can be applied to update the number of vehicles due to the coordinator command. Besides, if $g = \tau(\mathbf{z}^{hm})$, we assume that human-driven vehicle has been granted permission to $\forall g' \in O(g) \setminus \mathcal{P}$. Denote $\mathbf{n} \in \mathbb{Z}_+^{|\mathcal{G}|}$ as number of vehicles assigned permission to, but not yet to travel over the grid and denote the g-th element as $\mathbf{n}[g]$, then we have grid dynamics as

$$\begin{aligned} \mathbf{n}[g](t) =& \mathbf{n}[g](t - T_s) + \chi_{O(g)\setminus\mathcal{P}}\Big(\tau\big(\mathbf{z}^{hm}(t)\big)\Big) - \\ & \chi_{O(g)\setminus\mathcal{P}}\Big(\tau\big(\mathbf{z}^{hm}(t - T_s)\big)\Big) + \chi_{\Xi(t)}(g) + \\ & \sum_{g' \in O(g)} \epsilon(g', \tau(\mathbf{z}^{\lambda(g')}(t - T_s)), g). \end{aligned} \tag{11}$$

Note that we use $\chi_{O(g)\setminus\mathcal{P}}(\tau(\mathbf{z}^{hm}))$, rather than $\tilde{\mathbf{z}}^{hm}$ predicted by the prediction layer, which would be relatively

conservative since $\tau(\tilde{\mathbf{Z}}^{hm}) \subset O(g)$ if $g = \tau(\tilde{\mathbf{Z}}^{hm})[1]$. However, by doing so, we do not solely rely on the predicted $\tau(\tilde{\mathbf{Z}}^{hm})$ to make decision. In addition to the grid dynamics described by $\mathbf{n}(t)$, the set of active vehicles $\mathcal{N} \subset \mathbb{Z}_+$ is also time dependent, therefore we write it as $\mathcal{N}(t)$. Recall that $\mathbf{l}(t) \in \{0, 1\}^{|\mathcal{G}|}$ denotes whether each grid has been assigned to any vehicle or not until time t, then we have

$$\mathbf{l}[g](t) = \begin{cases} 1, & \text{if } \mathbf{n}[g](t) > 0, \\ 0, & \text{else.} \end{cases} \tag{12}$$

Then the overall system can be viewed as a trivial hybrid system and modeled as (L, D, E) where $\mathbf{l} \in L = \{0, 1\}^{|\mathcal{G}|}$ is the control location, $D : L \to D(\mathbf{l})$ is the dynamics of system state, which is a duple $(\mathbf{n}, \mathcal{N})$ and the dynamics of $\mathbf{n}$ is given by Eq. (11), and the edge $E \subset L \times Guard \times Jump \times L$ where $Guard$ and $Jump$ is specified by (12). The model can be interpreted as processing n grid maps, each of which belongs to one active vehicle in $\mathcal{N}$, and extract information from them. One illustration of such procedure is shown in Fig. 4, where the last layer is due to those parked vehicles.

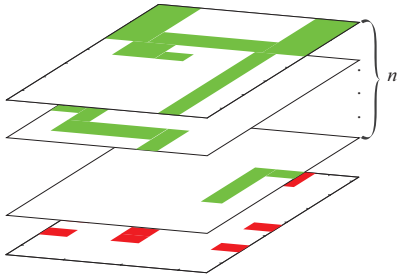


Fig. 4: Processing information from vehicles and coordinator.

Recall that $\mathbf{p} \in \mathbb{Z}_+^{|\mathcal{G}|}$ is defined as number of vehicles planned to travel to each grid at $t + T_s$. The approach for modeling $\mathbf{p}$ is similar to that of $\mathbf{n}$ and therefore is neglected.

VI. NUMERICAL EVALUATION

This section provides numerical evaluation of the proposed control architecture. The results are demonstrated in two different scales, namely in the infrastructure level and in individual vehicle level. We first remove the human-driver prediction to demonstrate that a deadlock maybe caused by human driving behavior and then show that such deadlock can be resolved by adding the predicting functionality to assist individual automated vehicle involved in the situation.

A. Setup and parameters

We consider one particular layout of a parking lot, as shown on the left in Fig. 3, with 31 parking spots, two of which (grid 30 and 31) are reserved spots, and in total 82 grids. Therefore, we have $\mathcal{G} = \{1, 2, ..., 82\}$ and $\mathcal{P} = \{1, 2, ..., 29\}$ (30 and 31 excluded). At time $t = 0$, we have $|\mathcal{F}(0)| \sim U(|\mathcal{P}|)$, that is the number of free spots in the beginning is a random variable with uniform distribution. For vehicle in $\mathcal{P} \setminus \mathcal{F}(t)$, its intention to leave is a Bernoulli distribution $Be(p_l)$, that is the probability it may leave at any sampling instance is p_l. Similarly, at any sampling instance, the event that a vehicle would like to enter the parking lot is following $Be(p_e)$. When a vehicle shows up at the entrance, the event that the vehicle is human-driven is also following a distribution $Be(p_h)$. The parameters used here are $p_l = p_e = 0.5$ and $p_h = 0.05$. $T_s = 3t_s$ where T_s and t_s are the sampling time in the infrastructure and the vehicle respectively.

For vehicle cooperation, the parameters used are listed in Table I and the distance function $\phi_{h,p}$ appearing in the cost function (4) is defined as

$$\phi_{k,p}(\mathbf{z}_i^k, \mathbf{z}_i^p) := \|\text{sgn}(y_j^p - y_j^k)(\mathbf{z}_i^k - \mathbf{z}_1^p) + \mathbf{d}\|_{Q_c} \tag{13}$$

where $\text{sgn}(\cdot)$ is a sign function and we require $\text{sgn}(0) = 1$, $\mathbf{d} = [0\ y_{min}\ 0]^T$ and $Q_c = \text{diag}(0, \Delta, 0)$. Here y_{min} defines the minimum distance in y direction between two vehicles.

TABLE I: Simulation parameters (SI units)

l_w	l_r	l_f	β_{max}	$\dot{\beta}_{max}$	a_{max}
2	2	2	$\pi/4$	$\pi/1.8$	1
t_s	t_p	N	N_c	N_z	N_v
0.1	1	15	15	100	100

B. Performance without human driver prediction

Although performing accordingly most of time, the designed control may indeed cause a deadlock. Fig. 5 shows one such instance. The plot on the left illustrates $\mathbf{n}$, namely the number of vehicles planning to reach each grid at $t + T_s$ estimated at time t. Empty grid g means $\mathbf{n}[g] = 0$, one with green circle means $\mathbf{n}[g] = 1$ and red if $\mathbf{n}[g] > 1$. The figure on the right shows vehicle position with blue for parked, green for entering, red for leaving, black for on-hold (not showing in this figure), and solid for human-driven. In this case, two automated vehicles plan to take spot 10 and 12 respectively, and the human-driven vehicle is assigned to spot 11 since the vehicle there just departures.

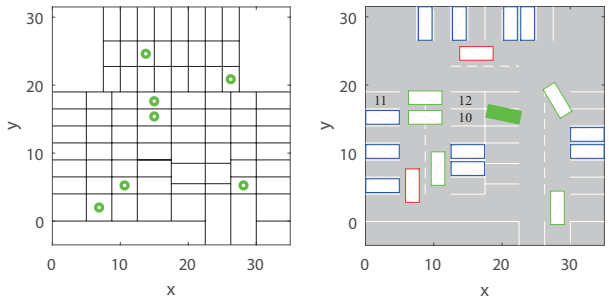


Fig. 5: Grid dynamics $\mathbf{n}$ and vehicle positions at certain instance without human driver prediction. Deadlock is formed around parking spot 12.

C. Performance with human driver prediction

When the human driving prediction is installed, the deadlock can be predicted, as shown in Fig. 6 and the cooperation of automated vehicles can be initiated. Note that due to the

human-driven vehicle at g, all $g' \in O(g) \setminus \mathcal{P}$ are set to more than 0 to raise the caution. Fig. 7 shows the vehicle positions before and after the cooperation where the cooperative MPC is initiated by the coordinator and operated by the automated vehicles to avoid human-driven vehicles. Blue dots are planned path, purple ones are predicted and blue solid lines are traveled trajectories. The convergence of the distributed MPC and the control with and without distributed MPC engaged are shown in Figs. 8 and 9 respectively.

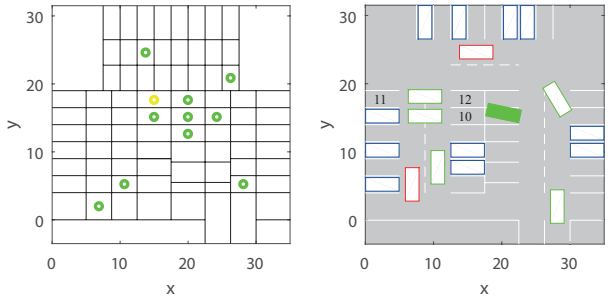
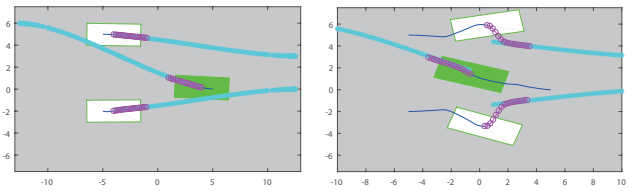


Fig. 6: Grid dynamics $\mathbf{n}$ and vehicle positions at certain instance with prediction layer. Deadlock around spot 12 is predicted.



(a) Before cooperation. (b) After cooperation.

Fig. 7: Cooperative MPC for avoiding human-driven vehicle.

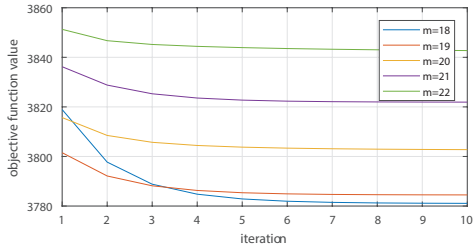


Fig. 8: Convergence results during 10 iteration starting from different sampling instances m.

VII. CONCLUSION

In this paper, a hierarchical control system is introduced to manage mixed automated and human-driven vehicles in smart parking lots. The layered system includes distributed planning and tracking for individual vehicles and centralized prediction and coordination in the management level. Due to the unique characters of parking lots, the prediction of human drivers is addressed by the management level. The predicted information is applied by the automated vehicles to resolve deadlock via cooperative distributed MPC.

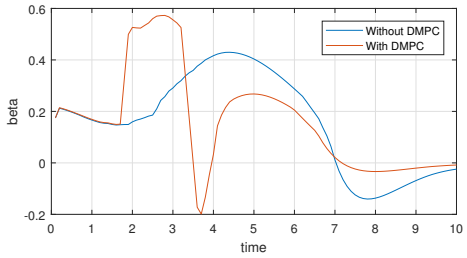


Fig. 9: Control comparison without and with the cooperative MPC.

Numerical evaluation shows the effectiveness of the proposed architecture and indicates its potential for any enclosed environment. However, in the current setup the distributed control in the vehicle layer only handles the steering control while the desired speeds are set by the management level. In the future, the longitudinal controller will also be designed as part of cooperative controller in the vehicle layer.

REFERENCES

[1] INRIX, "The impact of parking pain in the US, UK and Germany," 2017. [Online]. Available: http://www2.inrix.com/research-parking-2017

[2] European Parking Association, "Scope of parking in Europe," 2013. [Online]. Available: http://www.europeanparking.eu/en/documents/data-collection/

[3] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*. IEEE, 2008, pp. 48–57.

[4] S. Valipour, M. Siam, E. Stroulia, and M. Jagersand, "Parking-stall vacancy indicator system, based on deep convolutional neural networks," in *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*. IEEE, 2016, pp. 655–660.

[5] J. B. Slemmer and N. F. Rivenburgh, "Automated parking director systems and related methods," Nov. 14 2006, US Patent 7,135,991.

[6] D. K. Chew, "Apparatus and method for locating, identifying and tracking vehicles in a parking area," Feb. 22 2011, US Patent 7,893,848.

[7] Hikvision, "Mobile robot solutions: Smart parking industry solutions," 2018. [Online]. Available: http://en.hikrobotics.com/robot/robotplaninfo.htm?type=546&oid=1598

[8] X. Du and K. K. Tan, "Autonomous reverse parking system based on robust path generation and improved sliding mode control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1225–1237, 2015.

[9] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *2018 IEEE 57th Annual Conference on Decision and Control (CDC)*. IEEE, 2018.

[10] Daimler AG, "Finally: Parking made easy," July 2018. [Online]. Available: http://www.daimler.com/innovation/next/parking-pain-2.html

[11] BMW AG, "Intelligent parking," 2018. [Online]. Available: https://www.bmw.co.uk/bmw-ownership/connecteddrive/driver-assistance/intelligent-parking#gref

[12] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2017.

[13] T. Lin, H. Rivano, and F. L. Moul, "A survey of smart parking solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229–3253, Dec 2017.

[14] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 66–73.

[15] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Systems & Control Letters*, vol. 59, no. 8, pp. 460–469, 2010.