Distributed Cooperative MPC for Autonomous Driving in Different Traffic Scenarios

Fatemeh Mohseni, Erik Frisk and Lars Nielsen

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-172226>

N.B.: When citing this work, cite the original publication.

Mohseni, F., Frisk, E., Nielsen, L., (2020), Distributed Cooperative MPC for Autonomous Driving in Different Traffic Scenarios, *IEEE Transactions on Intelligent Vehicles*, , 1-1.

<https://doi.org/10.1109/TIV.2020.3025484>

Original publication available at:

<https://doi.org/10.1109/TIV.2020.3025484>

Copyright: Institute of Electrical and Electronics Engineers

<http://www.ieee.org/index.html>

Distributed Cooperative MPC for Autonomous Driving in Different Traffic Scenarios

Fatemeh Mohseni, Erik Frisk, and Lars Nielsen

Abstract—A cooperative control approach for autonomous vehicles is developed in order to perform different complex traffic maneuvers, e.g., double lane-switching or intersection situations. The problem is formulated as a distributed optimal control problem for a system of multiple autonomous vehicles and then solved using a nonlinear Model Predictive Control (MPC) technique, where the distributed approach is used to make the problem computationally feasible in real-time. To provide safety, a collision avoidance constraint is introduced, also in a distributed way. In the proposed method, each vehicle computes its own control inputs using estimated states of neighboring vehicles. In addition, a compatibility constraint is defined that takes collision avoidance into account but also ensures that each vehicle does not deviate significantly from what is expected by neighboring vehicles. The method allows us to construct a cost function for several different traffic scenarios. The asymptotic convergence of the system to the desired destination is proven, in the absence of uncertainty and disturbances, for a sufficiently small MPC control horizon. Simulation results show that the distributed algorithm scales well with increasing number of vehicles.

Index Terms—Autonomous Vehicles, Collision Avoidance, Cooperative Vehicle Systems, Model Predictive Control.

I. INTRODUCTION

AUTONOMOUS vehicle technologies have the potential to fundamentally change the transportation system with a possibility to reduce energy consumption, pollution, accidents and congestion costs, see e.g., [1]. In addition, grouping vehicles leads to a reduction of space used by vehicles on the road, and can significantly address the above traffic issues. Consequently, the concept of cooperation and control algorithms that apply to multiple autonomous vehicle systems have become an important topic for researchers in the automation and control field.

Among the existing motion planning methods, see [2], path planning approaches that are based on optimization, more explicitly Model Predictive Control (MPC), have been successful in autonomous driving. Handling different limitations of both the state and input control signals while planning maneuvers based on the trade-off between different control objectives is the main advantages of MPC [3]. Therefore, it is an appropriate scheme for complex and constrained environments, and is well suited to the complex traffic situations considered in this paper.

Recently, formulating a cooperative control design among multiple agents assigned to a specific task that can navigate autonomously without collision has received significant attention, see e.g., [4]. Further, distributed control among multi-vehicle systems can improve tractability and provide better scalability than centralized approaches, see [5] and the recent survey [6]. These methods have less communication and computation cost compared to centralized methods, see e.g., [7], and therefore,

they are appealing to be used for control of a system that contains multiple autonomous vehicles.

When more than one autonomous vehicle work in the same area, the problem of vehicle collision has to be considered in control design in order to provide safety of passengers. Recently, many researchers have worked on improved safety in transportation systems, see e.g., [8], [9] and our conclusion is that it is important to consider the non-holonomic property of vehicles in the control design. In addition, it is also important to predict future behavior of other vehicles to be able to guarantee safety for the entire maneuver among autonomous vehicles. In a previous paper [10], a first attempt was made to handle this situation by a collision avoidance cost in an MPC criterion. In comparison, in our now proposed method, a collision avoidance constraint is introduced in a predictive way that takes the uncertain deviation between the actual predictive and estimated trajectories of vehicles into consideration. This constraint together with a compatibility constraint and the tuning parameters for the terminal region guarantee convergence and collision avoidance properties. The compatibility constraint motivated by [11]–[14] has an important role in guaranteeing both collision avoidance and the convergence of the system. The predictive collision avoidance feature of our proposed method is one of the contributions, and makes our method differ from some of the existing works that use MPC for multi-vehicle autonomous driving, e.g., [10], [13].

There exist different traffic scenarios that if they can be performed properly by autonomous vehicles, then different traffic issues, e.g., accidents and congestion costs can be reduced significantly. One of the challenging scenarios in autonomous vehicle technology that is also considered in this work is intersection maneuvers, see, e.g., [15]. In [16], by using Lagrangian formulation, an intersection maneuver for cooperative vehicles was defined in the form of a convex optimization problem. Their proposed method has been demonstrated to work well with minimizing the trip time. In [17], by using a new reservation-based approach, a mechanism has been proposed for autonomous multi-agent system coordination in intersection situations. The authors showed that their proposed method can perform well and improve the intersection control.

The main problem studied in this paper is how to control multiple vehicles in a safe and distributed computationally efficient way in complex traffic situations, accounting for capabilities of maneuvering and in the presence of limitations, e.g., road boundaries, speed limitations or other traffic rules and moving non-autonomous vehicles. Existing methods have been defined for one specific scenario, e.g., [13], [16], [17]; however, a key contribution of this work is defining an approach that is general enough to be applied for different traffic maneuvers.

Compared to [10], a new traffic scenario is included which includes an analysis of robustness towards non-autonomous vehicles in the traffic scenario, and a new formulation of the control objective is introduced. The control problem is reformulated with new and extended compatibility constraints that allows the vehicles to deviate from intended trajectories and still guarantee collision avoidance. The reformulation allows a treatment as trajectory stabilization instead of as a point stabilization problem, which is essential to complete the collision free convergence result. With this method it is possible to quantify the computational gains with the distributed approach in common traffic situations like double lane-switching and intersections maneuvers.

II. VEHICLE MODEL AND CONSTRAINTS

The system dynamics and kinematic constraints that are used in our proposed method are defined in this section. It is assumed that the model of each vehicle is described by an ODE and for the i^{th} vehicle, $i = 1, \dots, n$, the state and control vectors are $z_i(t) = (x_{ri}(t), y_{ri}(t), \theta_i(t), \psi_i(t))$ and $u_i(t) = (v_i(t), \omega_i(t))$ respectively, where $p_{ri}(t) = (x_{ri}(t), y_{ri}(t))$ and $\theta_i(t)$ are the mid-point of the rear wheel axle in the Cartesian coordinate frame and the orientation angle respectively; $v_i(t)$ is the speed; $\psi_i(t)$ is the steering angle of the front wheels with respect to the vehicle's body and $\omega_i(t)$ denotes the steering rate. In addition, $p_i(t) = (x_i(t), y_i(t))$ refers to the position of vehicle's center and the parameter L denotes the wheelbase length, see Fig. 1. Each vehicle is modeled by the kinematic equations

$$\dot{z}_i(t) = f(z_i(t), u_i(t)) = \begin{pmatrix} v_i(t) \cos(\theta_i(t)) \\ v_i(t) \sin(\theta_i(t)) \\ \frac{1}{L} v_i(t) \tan(\psi_i(t)) \\ \omega_i(t) \end{pmatrix}. \quad (1)$$

Throughout the rest of this paper, the nonholonomic vehicle model (1) is used. A more detailed model for each vehicle may include also load transfer, the rotation angles of each wheel as generalized coordinates and the body slip, as well as typical non-idealities such as tire deformation. However, in this paper regular traffic situations are considered with no aggressive maneuvers, and so, model (1) captures the essence of the vehicle motion and is well suited.

The autonomous vehicle motion planning and stabilization problem contains limitations resulting from obstacles and input restrictions dealing with mechanical limits. These constraints must be satisfied in control design; therefore, the input limitations have to be considered simultaneously. So, in addition to (1), the constraints on the control variables are also considered to reflect physical and mechanical limitations

$$v_{min_i} \leq v_i(t) \leq v_{max_i}, \quad |\omega_i(t)| \leq \omega_{max_i}. \quad (2)$$

III. PROBLEM MOTIVATION

The main objective of this paper is to control multiple vehicles in an optimal and computationally efficient cooperative way in order to resolve different traffic situations, for example, double lane-switching and intersections maneuvers. Here, controlling multiple vehicles requires that vehicles in the traffic scenario communicate and send information about their intent,

i.e., collaborating to ensure safe driving for all included vehicles. To make the computational time scale well with increasing scenario complexity, and also to make each vehicle more autonomous and not dependent on any centralized infrastructure, the solution must be distributed.

When controlling an autonomous vehicle in general traffic, with significant situation uncertainties, it is essential that the proposed approach is robust against disturbances for example, changes in vehicles behavior and vehicles that do not collaborate and communicate their driving intent. In addition, the control design must deal with actuator limitations, vehicle motion constraints, and the other constraints described above. Based on this problem context, a model predictive control (MPC) approach is explored where a finite-horizon optimal control problem is solved repeatedly. A centralized approach is straight forward to formulate and provides good performance w.r.t the global cost function and constraints, but at the cost of computation effort. Therefore, first the problem is defined in a centralized way and then, since the aim is to solve the problem in a computationally more efficient way, the centralized problem is decomposed into a set of distributed optimal control problems each associated with one vehicle.

To limit the scope of the work and focus on the controls; this work does not consider the uncertainty in sensing and assumes that reliable positional information from cameras, lidars or radars are available. In addition, communication delays are not taken into account and only forward paths are of interest. Further, the mission is defined by a higher-level task planner.

IV. DISTRIBUTED COOPERATIVE MPC (DCMPC) FOR MULTIPLE VEHICLES

A. Optimization formulation

Assume n vehicles, and let z_i and u_i represent the state and control vectors from (1) and let $p_i^f(t) = (x_i^f(t), y_i^f(t))$ be the desired final destination of the i^{th} vehicle. With a position error $\|P(\cdot; t_k) - P^f\| = \sum_{i=1}^n \|p_i(\cdot; t_k) - p_i^f\|$ included in the objective function, the optimal control problem is formulated as a point stabilization problem. However, as shown in [18], for a nonholonomic dynamic model, the point stabilization problem is more difficult than path following and trajectory tracking. Therefore, the point stabilizing problem is transformed into a path tracking problem by defining a reference nominal path for each vehicle denoted by $p_i^d(t)$. Note that $p_i^d(t) = (x_i^d(t), y_i^d(t))$ is a position *trajectory* but as this position trajectory also defines a *path* we will refer to the position trajectory as a path.

A first step is to find an admissible desired nominal path for each vehicle. Here, the method defined in [19] that considers the continuous curvature reference path generation method is used in order to find an admissible nominal path and note that the collision is not considered at this stage. Collision avoidance will be added in the MPC problem that will result in the actual optimal path. After determining the desired nominal path $p_i^d(t) = (x_i^d(t), y_i^d(t))$ that satisfies the dynamical vehicle model, we can derive the remaining state and input trajectories.

The deviation from the desired nominal path for each vehicle is called adaption and is denoted by $p_i^a(t) = p_i(t) - p_i^d(t)$, and

the desired nominal state and control input vectors are denoted by z_i^d and u_i^d respectively. Furthermore, the corresponding adaption vectors are redefined as $z_i^a(t) = z_i(t) - z_i^d(t)$, and $u_i^a(t) = u_i(t) - u_i^d(t)$. By using the above adaption vectors, the Centralized Cooperative MPC (CCMPC) problem can be defined as follows. First, the path, state, control and corresponding adaption vectors of n vehicles are concatenated into vectors as $P(t)$, $Z(t)$, $U(t)$, $P^a(t)$, $Z^a(t)$ and $U^a(t)$. Then, for the weighting scalars γ_z , γ_u and γ_t , the centralized formulation of problem is formulated as follows.

Problem 1. With given prediction horizon h_p for a system of n vehicles and at any update time t_k , $k = 0, 1, \dots$, find $J^*(t_k, P^{a*}(t), U^{a*}(t)) = \min_{U(\cdot)} J(t_k, P^a(t), U^a(t))$, for

$$J = \int_{t_k}^{t_k+h_p} (\gamma_z \|P^a(\tau; t_k)\|^2 + \gamma_u \|U^a(\tau; t_k)\|^2) d\tau + \gamma_t g(P^a(t_k + h_p; t_k)), \quad (3)$$

subject to the constraints

$$\dot{Z}(t; t_k) = f(Z(t; t_k), U(t; t_k)), \quad Z(t; t_k) \in \mathcal{Z}, \quad U(t; t_k) \in \mathcal{U} \\ g_{col_{ij}}(p_i(t; t_k), p_j(t; t_k)) \leq 0, \quad \forall i, j \in n, \quad j \neq i \quad (4)$$

where $t \in [t_k, t_k + h_p]$. The first two terms in (3) are defined for performing the maneuver, and the last term is the terminal cost which is needed to guarantee the convergence of the system. This terminal cost is a continuous differentiable function, satisfying $g(0) = 0$ and $g(P^a(t)) > 0$ for any $P^a(t) \neq 0$, and is defined as $g = \sum_{i=1}^n g_i = \sum_{i=1}^n \|p_i^a(t_k + h_p; t_k)\|^2$. Furthermore, (4) represents the collision avoidance constraint and for all $i, j \in n$ where $j \neq i$ is defined as

$$g_{col_{ij}}(p_i(t; t_k), p_j(t; t_k)) := \alpha - \|p_i(t; t_k) - p_j(t; t_k)\| \quad (5)$$

where α is a safe distance between every two vehicles.

B. Decoupling of the system

A main issue in control of multiple vehicles is collision avoidance. To formulate a distributed solution, first note that in a system of multiple vehicles, it is not required to immediately act in response to the information from vehicles that are far away. In Problem 1, a collision constraint (5) is enforced between any two vehicles. Therefore, the communication and computational cost of implementing a centralized MPC grows with the number of vehicles. Thus, it is attractive to produce a distributed scheme of MPC that enables autonomy of the individual vehicles and improves the tractability of the MPC.

Let i, j be two arbitrary identical vehicles. Consider a circular *dominance zone* with radius $D_i = \sqrt{(L_f/2)^2 + (L/2 + L_r)^2}$, where L_f and L_r denote the vehicle frontal width and overhang length respectively, a *protection zone* with radius $D_i + \Phi_i$, where Φ_i is a safety threshold, and a *neighborhood zone* with radius r_i , see Fig. 1.

Definition 1 (Neighbors). The set of neighbors of vehicle i is denoted by $\mathcal{N}_i$ and is defined as

$$\mathcal{N}_i := \{j : \|p_i - p_j\| \leq r_i, i \neq j\} \quad (6)$$

Fig. 1. Variables, and notations related to vehicles and zones.

where $r_i = 2(v_{max}h_p + D_i) + \Phi_i$, and v_{max} is the maximum speed limit. Note that the size of the neighborhood zone is defined as a function of maximum speed limit and prediction horizon in order to take into account the worst-case head-on collision over the MPC prediction horizon.

To avoid collision between vehicle i and neighbor j , the following inequality should hold:

$$\|p_i - p_j\| \geq 2D_i + \Phi_i, \quad \forall j \in \mathcal{N}_i. \quad (7)$$

Note that if two vehicles i and j are already apart by at least r_i at time t_k , it is not necessary to consider collision between them over the prediction horizon $[t_k, t_k + h_p]$.

C. Estimation of the neighbors behavior

Now suppose n DCMPC optimal control problems, one for each vehicle, are all solved at common time instants t_k . For each optimal control problem, the collision constraint contains the connection between the vehicle and its neighbors at time t_k . Therefore, when the local optimal control problems are solved, each vehicle must know the state trajectories of all its neighbors over the time interval $[t_k, t_k + h_p]$. However, this information is not available at instant t_k for the neighboring vehicles and each vehicle must estimate the state trajectories of its neighbors over $[t_k, t_k + h_p]$. With these estimates, the vehicle has all the information needed for solving its local optimal control problem. The trajectories that each vehicle predicts for its neighbors are called the *estimated trajectories*. Further, before time $t = t_k$ each vehicle has transmitted its control signal computed at t_{k-1} for the time interval $[t_k, t_{k-1} + h_p]$. This control signal will be used as the first part of the estimated controller when the local DCMPC problems are solved at time $t = t_k$. In addition, since the control signal for the time interval $[t_{k-1} + h_p, t_k + h_p]$ is not known, a stabilizing feedback controller is used for the second part of the estimated controller.

Definition 2 (Estimated control). At every time instant $\tau \in [t_k, t_k + h_p]$, the estimated control for each vehicle is defined by the optimal predictive input and a terminal controller, i.e.,

$$\hat{u}_i(\tau; t_k) = \begin{cases} u_i^*(\tau; t_{k-1}), & \tau \in [t_k, t_{k-1} + h_p] \\ u_i^K(\tau; t_k), & \tau \in [t_{k-1} + h_p, t_k + h_p] \end{cases} \quad (8)$$

In Fig. 2, the red curve is the optimal control signal computed at time $t = t_{k-1}$. Then, when computing the optimal

Fig. 2. An illustration of the optimal and estimated control signals.

control input at time $t = t_k$, each vehicle needs to know the trajectory of its neighboring vehicles also for the time interval $[t_{k-1} + h_p, t_k + h_p]$ which is unknown. Therefore, the terminal controller $u_i^K(\tau; t_k)$, a dynamic feedback controller here found via standard full-state linearization, see for example [20], [21], is used to compute an estimate of the optimal trajectories, the green curve in Fig 2. The dashed line indicates the estimated control signal and contains a part of the red curve and the terminal controller. Finally, the resulting optimal controller for the whole interval $[t_k, t_k + h_p]$ can be then computed which is the blue curve. According to the estimated control, the estimated predictive trajectory is given by

$$\hat{z}_i(\tau; t_k) = \begin{cases} z_i^*(\tau; t_{k-1}), & \tau \in [t_k, t_{k-1} + h_p] \\ z_i^K(\tau; t_k), & \tau \in [t_{k-1} + h_p, t_k + h_p] \end{cases} \quad (9)$$

where $z_i^K(\tau; t_k)$ is the estimated terminal trajectory.

Definition 3 (Terminal set). For every vehicle i , and for $\xi_i(t_k) \in (0, \infty)$, and $\tau = t_k + h_p$, the terminal set is

$$\Omega_i(t_k) := \{z_i : \|z_i^a(\tau; t_k)\| \leq \xi_i(t_k)\}. \quad (10)$$

Assumption 1. For every vehicle, the largest $\xi_i(t_k) > 0$ and the terminal feedback controller gains are chosen such that

$$\|p_i(\tau; z_i(t_k)) - p_j(\tau; z_j(t_k))\| \geq 2D_i + \Phi_i, \quad \forall j \in \mathcal{N}_i, \quad (11)$$

$$\gamma_{ti} \frac{d}{dt} g_i(p_i^a(\tau; z_i(t_k))) \leq -(\gamma_{zi} \|p_i^a(\tau; z_i(t_k))\|^2 + \gamma_{ui} \|u_i^a(\tau, z_i(t_k))\|^2) \quad (12)$$

where γ_{zi} , γ_{ui} and γ_{ti} are weighting scalars given a priori. Since γ_{ti} , γ_{zi} and γ_{ui} are positive values, (12) implies that Ω_i is invariant for the closed-loop nonlinear system $\dot{z}_i = f_i(z_i, u_i^k)$ controlled by the local terminal feedback controller. In addition, any trajectory of z_i^k starting in Ω_i stays in Ω_i and converges to the desired path. It is also assumed that for all $z_i \in \Omega_i$, u_i^k is the feasible stabilizing feedback, i.e., $u_i(\tau; z_i(t_k)) \in \mathcal{U}$. The condition (12) means that the terminal cost $\gamma_{ti} \|p_i^a(t_k + h_p; t_k)\|^2$ bounds the system cost for any initial state starting in the terminal set.

This condition is based on the definition of the terminal dynamic feedback controller and convergence property of that following a similar approach as in [22].

Remark 1. Note that finding the associated positively invariant terminal-state region is not an easy task and is still a challenging issue, even for control of one single nonholonomic vehicle, [23]. There exist some methods in the literature for finding this region, e.g., solving an optimization problem to determine the

borders of the region. However, following any of the existing methods does not result in a unique terminal state region for a given system. The size of the region depends on the selection of the terminal controller parameters and the non-linearity of the system to be controlled.

D. Compatibility and collision avoidance constraints

In the centralized method, the controller calculates the trajectory of all vehicles at the same time by considering the collision avoidance constraint (5). But, in the distributed method, each vehicle uses the estimated states of its neighbors and not all the vehicles, in the collision avoidance constraint. So, if any vehicle moves away too far from its estimated trajectory, it may collide with other vehicles. Therefore, inspired from [13] and [14], an additional constraint called the *compatibility constraint* is added to the DCMPC problem of each vehicle which ensures that each vehicle does not deviate too far from the trajectory expected by its neighbors.

In other words, when the predictive trajectory is planned over the prediction horizon $\tau \in [t_k, t_k + h_p]$, the collision avoidance must be guaranteed. Under ideal conditions, each vehicle i considers the collision avoidance constraint (7) over $[t_k, t_k + h_p]$. However, the actual predictive trajectory $p_j(\tau; t_k)$ is unavailable for vehicle i at t_k , and the corresponding estimated predictive trajectory $\hat{p}_j(\tau; t_k)$ is used instead. So, the influence of the uncertain deviation between the estimated and actual predictive trajectories of vehicle i , $\zeta_i(\tau; t_k) = p_i(\tau; t_k) - \hat{p}_i(\tau; t_k)$, on the collision avoidance cannot be ignored. Therefore, a compatibility constraint is proposed to handle the uncertain deviation. The distributed collision avoidance constraint is denoted by $g_{col_{ij}}(p_i, \hat{p}_j) \leq 0$, where $\hat{p}_j$ is the estimated path of the j^{th} neighbor. Constraints on the uncertain deviation $\zeta_i(\tau; t_k)$ is imposed in each individual optimization problem.

The compatibility constraint is important for guaranteeing the trajectory to converge to the desired target neighborhood and avoiding collision, and is defined as

$$g_{com_i}(\zeta_i(\tau; t_k)) = \|\zeta_i(\tau; t_k)\| - \eta_i(t_k) \leq 0 \quad (13)$$

where $\eta_i(t_k) = \min_{j \in \mathcal{N}_i} \{\eta_{ij}(t_k)\}$ and

$$\eta_{ij}(t_k) = \frac{\min_{\tau} \{\|\hat{p}_i(\tau; t_k) - \hat{p}_j(\tau; t_k)\|\}}{2} - D_i \quad (14)$$

for $\tau \in [t_k, t_k + h_p]$ and $j \in \mathcal{N}_i$. In (13), $\eta_i(t_k)$ is introduced to guarantee collision avoidance under uncertainty ζ_i . By using the triangle inequality property for (13), we have

$$\begin{aligned} \|p_i(\tau; t_k) - p_j(\tau; t_k)\| &= \\ \|p_i(\tau; t_k) - \hat{p}_j(\tau; t_k) - \zeta_j(\tau; t_k)\| &\geq \\ \|p_i(\tau; t_k) - \hat{p}_j(\tau; t_k)\| - \eta_i(t_k). \end{aligned} \quad (15)$$

The constraint (7) can then be guaranteed by

$$\|p_i(\tau; t_k) - \hat{p}_j(\tau; t_k)\| \geq 2D_i + \Phi_i + \eta_i(t_k), \quad \forall j \in \mathcal{N}_i \quad (16)$$

Thus the collision avoidance constraint is

$$\begin{aligned} g_{col_{ij}}(p_i(\tau; t_k), \hat{p}_j(\tau; t_k)) &= \\ 2D_i + \Phi_i + \eta_i(t_k) - \|p_i(\tau; t_k) - \hat{p}_j(\tau; t_k)\| &\leq 0. \end{aligned} \quad (17)$$

Therefore, if for all $\tau \in [t_k, t_k + h_p]$ the compatibility constraint (13) and the collision constraint (17) are satisfied, then by implementing $u_i^*(\tau; t_k)$, the collision avoidance (7) will be guaranteed over the time interval $\tau \in [t_k, t_k + 1]$. Compared with the compatibility constraints proposed in [13] and [14], the constraint defined here also considers collision avoidance.

E. DCMPC control design

To reformulate the centralized optimization problem into a distributed one including n different problems and to ensure that each vehicle does not move away too far from the trajectory expected by its neighbors, a compatibility constraint is added to each problem. Here, it is assumed that prediction horizon (h_p) and control horizon (h_c) are identical for all vehicles and are chosen such that h_p covers the slowest system mode; and h_c , to be shorter than the fastest system's mode. In that way, the controller can react timely to disturbances.

Let (10) define the terminal set at instant t_k , (13) the compatibility constraint, and (17) the collision avoidance constraint. The DCMPC problem can now be stated as follows.

Problem 2. For each vehicle $i \in \{1, \dots, n\}$ at any MPC update time t_k , $k \in \mathbb{N}$, and with given $\hat{z}_j(t; t_k)$, $z_i(t_k)$, $\hat{z}_i(t; t_k)$ for all $\tau \in [t_k, t_k + h_p]$, where $j \in \mathcal{N}_i$, for given weighting scalars γ_{zi} , γ_{ui} and γ_{ti} find $J_i^*(t_k, p_i^a, \hat{p}_i, u_i^a) = \min_{u_i(\cdot)} J_i(t_k, p_i^a, \hat{p}_i, u_i^a)$, where

$$J_i(t_k, p_i^a, \hat{p}_i, u_i^a) = \int_{t_k}^{t_k + h_p} \gamma_{zi} \|p_i^a(\tau; t_k)\|^2 + \gamma_{ui} \|u_i^a(\tau; t_k)\|^2 d\tau + \gamma_{ti} g_i(p_i^a(t_k + h_p; t_k)) \quad (18)$$

subject to

$$\dot{z}_i(t; z_i(t_k)) = f(z_i(t; z_i(t_k)), u_i(t; z_i(t_k)))$$

$$u_i(\tau; z_i(t_k)) \in \mathcal{U}, \quad z_i(\tau; z_i(t_k)) \in \mathcal{Z}$$

$$\dot{\hat{z}}_i(t; z_i(t_k)) = f(\hat{z}_i(t; z_i(t_k)), \hat{u}_i(t; z_i(t_k))) \quad (19)$$

$$\dot{\hat{z}}_j(t; z_j(t_k)) = f(\hat{z}_j(t; z_j(t_k)), \hat{u}_j(t; z_j(t_k))) \quad (20)$$

$$g_{com_i}(\zeta_i(\tau; t_k) \leq 0) \quad (21)$$

$$g_{col_{ij}}(p_i(\tau; t_k), \hat{p}_j(\tau; t_k)) \leq 0 \quad (22)$$

$$z_i(t_k + h_p; t_k) \in \Omega_i(t_k). \quad (23)$$

The constraint (19) is added because estimated states are needed in the compatibility constraint (21). In addition, (20) is the kinematic model for the j^{th} neighbor and can be removed if the cars communicate the estimated states. The main difference between the centralized and distributed problems defined in Problem 1 and 2 is that the complexity inherent with the centralized approach is reduced by defining n smaller problems, one for each vehicle, where only the connections between any given vehicle and its neighbors are included, and these n problems can be solved in parallel. The optimal solution for each DCMPC problem is denoted by $u_i^*(\tau; t_k)$ and the optimal state trajectory for the i^{th} vehicle is denoted by $z_i^*(\tau; t_k)$, $\tau \in [t_k, t_k + h_p]$.

Assumption 2. For each vehicle i , it is assumed that the optimal solution to Problem 2 exists and is numerically obtainable, and the position of each vehicle i is bounded.

F. Convergence results

Now, convergence is investigated and it is shown that the closed-loop position of all vehicles converges to a neighborhood of the desired destination P^f . Note that here convergence does not imply optimality. First, it is proved that initial feasibility of the implementation of the DCMPC algorithm provides subsequent feasibility. Then, by considering the sum of all the individual optimal cost functions, used as a Lyapunov function, the convergence of the path of the vehicles to the destinations will be proved similar to [11], [13], [24], [25].

The set of initial states for which Problem 2 is feasible at initialization is denoted by Z_σ . Feasibility of the optimization problem means that for every vehicle, there exists at least one control input such that the objective function J_i is bounded and all the constraints are satisfied. In addition, subsequent feasibility means that $z_i(\cdot; t_k) \in Z_\sigma$ implies $z_i(\cdot; t_{k+1}) \in Z_\sigma$.

Lemma 1. Under Assumptions 1–2 for a system of n autonomous vehicles with model (1), by assuming that the prediction horizon has been tuned such that the Problem 2 is feasible at initial time t_0 , and assuming that there exist at least one feasible solution for the system, then by applying the input $u_i(\tau) = u_i^*(\tau; t_k)$, $\tau \in [t_k, t_{k+1}]$, for a sufficiently small control horizon h_c , the optimization Problem 2 has a feasible solution at any t_k , $k > 0$.

Proof. By considering that Def. 1, and Assumptions 1–2 are satisfied, in a similar way as [11], the recursive feasibility is proven by induction.

Let $u_i(\cdot; t_k)$ and $z_i(\cdot; t_k)$ denote the feasible control and state solution at time t_k . The estimated control $\hat{u}_i(\cdot; t_{k+1})$, defined in (8), is a candidate control that can steer $z_i(t_{k+1}; t_k)$ to the terminal region Ω_i in time $t_{k+1} + h_p$. Under Assumption 2, since there are no disturbances and full state measurement is available, then for each vehicle $z_i(t_{k+1}) = z_i^*(t_{k+1}; t_k)$. Therefore, the estimated control $\hat{u}_i(\cdot; t_{k+1})$ defined in (8) can be a candidate control as the solution of the optimal control problem at time t_{k+1} with initial condition $z_i(t_{k+1}; t_{k+1})$ that can steer $z_i(t_{k+1}; t_k)$ to the terminal region in time $t_{k+1} + h_p$.

In addition, from Assumption 1, the terminal region is invariant for the system model controlled with the terminal controller, and the control and terminal constraint remain feasible from the property of the terminal controller. Furthermore, the compatibility constraint is satisfied since $z_i(\cdot; t_{k+1}) = \hat{z}_i(\cdot; t_{k+1})$ for all i and $\zeta_i(\tau; t_{k+1}) = \hat{p}_i(\tau; t_{k+1}) - p_i(\tau; t_{k+1}) = 0$, and according to Assumption 1, (13), and (16), it holds that for $[t_k, t_{k+1}]$, $\|p_i^*(\tau; t_k) - p_j^*(\tau; t_k)\| \geq 2D_i + \Phi_i$. Consequently, according to Definition 1 and Assumption 1, it holds that

$$\begin{aligned} 2D_i + \Phi_i &\leq \|\hat{p}_i(\tau; t_{k+1}) - \hat{p}_j(\tau; t_{k+1})\| \\ &= \begin{cases} \|p_i^*(\tau; t_k) - p_j^*(\tau; t_k)\|, & \tau \in [t_{k+1}, t_k + h_p], \\ \|p_i^{K^*}(\tau; t_k) - p_j^{K^*}(\tau; t_k)\|, & \tau \in [t_k + h_p, t_{k+1} + h_p], \end{cases} \end{aligned}$$

which guarantees the collision avoidance constraint for all $t \in [t_{k+1}, t_{k+1} + h_p]$. Therefore, the predictive input $u_i(\tau; t_{k+1}) = \hat{u}_i(\tau; t_{k+1})$ is a feasible solution at time t_{k+1} . $\square$

Theorem 1. Suppose that for a system of n autonomous vehicle with dynamics (1), Assumptions 1–2 are satisfied and

the Problem 2 has a feasible solution for each vehicle at the initial instant t_0 . By solving the optimization Problem 2 at each update time $t_k, k \geq 0$, by applying the input $u_i(\tau) = u_i^*(\tau; t_k), \tau \in [t_k, t_{k+1})$ of each vehicle, the position tracking adaption vector of the system converges to zero.

Proof. The proof follows from Lemma 1, Definition 1, Assumptions 1- 2, and using the sum of all n cost functions as a Lyapunov criterion, i.e., $V(t_k) = \sum_{i \in n} J_i(t_k, p_i^{a*}, \hat{p}_i, u_i^{a*})$, of the closed loop system $\dot{Z}(t) = F(Z(t), U^*(t))$. Following the arguments in [11], the function $V(t_k)$ satisfies $V(0) = 0$ and $V(Z^a(t_k)) > 0$ for $Z^a \neq 0$. The difference between the Lyapunov function for two consecutive update instants is

$$V(t_{k+1}) - V(t_k) = J^*(z(:, t_{k+1}), u(:, t_{k+1})) - J^*(z(:, t_k), u(:, t_k))$$

Since u^* and $\hat{u}$ both are feasible controls and since u^* is the optimal control input, the cost of applying u^* to the system is less than the total cost of applying $\hat{u}$. So,

$$\begin{aligned} V(t_{k+1}) - V(t_k) &\leq \int_{t_{k+1}}^{t_{k+1}+h_p} \gamma_z \|\hat{P}^a(\tau; t_{k+1})\|^2 + \gamma_u \|\hat{u}^a(\tau; t_{k+1})\|^2 d\tau + \\ &\gamma_t g(\hat{P}^a(t_{k+1} + h_p; t_{k+1})) - \int_{t_k}^{t_k+h_p} \gamma_z \|P^{a*}(\tau; t_k)\|^2 + \\ &\gamma_u \|u^{a*}(\tau; t_k)\|^2 d\tau - \gamma_t g(P^{a*}(t_k + h_p; t_k)) = \\ &\int_{t_{k+1}}^{t_k+h_p} \gamma_z \|\hat{P}^a(\tau; t_{k+1})\|^2 + \gamma_u \|\hat{u}^a(\tau; t_{k+1})\|^2 d\tau + \\ &\int_{t_k+h_p}^{t_{k+1}+h_p} \gamma_z \|\hat{P}^a(\tau; t_{k+1})\|^2 + \gamma_u \|\hat{u}^a(\tau; t_{k+1})\|^2 d\tau + \\ &\gamma_t g(\hat{P}^a(t_{k+1} + h_p; t_{k+1})) - \int_{t_k}^{t_k+h_p} \gamma_z \|P^{a*}(\tau; t_k)\|^2 + \\ &\gamma_u \|u^{a*}(\tau; t_k)\|^2 d\tau - \gamma_t g(P^{a*}(t_k + h_p; t_k)). \end{aligned}$$

Denote, $P^{a*}(t_k + h_p; t_k) = \hat{P}^a(t_k + h_p; t_{k+1})$. Then

$$\int_{t_{k+1}}^{t_k+h_p} \gamma_z \|\hat{P}^a(\tau; t_{k+1})\|^2 = \int_{t_{k+1}}^{t_k+h_p} \gamma_z \|P^{a*}(\tau; t_k)\|^2;$$

by using Definition 2. Therefore,

$$\begin{aligned} V(t_{k+1}) - V(t_k) &\leq \int_{t_{k+1}}^{t_k+h_p} \gamma_z \|P^{a*}(\tau; t_k)\|^2 + \gamma_u \|u^{a*}(\tau; t_k)\|^2 d\tau + \\ &\int_{t_k+h_p}^{t_{k+1}+h_p} \gamma_z \|\hat{P}^a(\tau; t_{k+1})\|^2 + \gamma_u \|\hat{u}^a(\tau; t_{k+1})\|^2 d\tau - \\ &\int_{t_k}^{t_{k+1}} \gamma_z \|P^{a*}(\tau; t_k)\|^2 + \gamma_u \|u^{a*}(\tau; t_k)\|^2 d\tau - \\ &\int_{t_{k+1}}^{t_k+h_p} \gamma_z \|P^{a*}(\tau; t_k)\|^2 + \gamma_u \|u^{a*}(\tau; t_k)\|^2 d\tau + \\ &\gamma_t g(\hat{P}^a(t_{k+1} + h_p; t_{k+1})) - \gamma_t g(\hat{P}^a(t_k + h_p; t_{k+1})). \end{aligned}$$

Furthermore, by using (12),

$$V(t_{k+1}) - V(t_k) \leq - \int_{t_k}^{t_k+h_c} \gamma_z \|P^{a*}(\tau; t_k)\|^2 + \gamma_u \|u^{a*}(\tau; t_k)\|^2 d\tau. \quad (24)$$

Since γ_z and γ_u are positive scalars, (24) becomes

$$V(t_{k+1}) - V(t_k) \leq - \int_{t_k}^{t_k+h_c} \gamma_z \|P^{a*}(\tau; t_k)\|^2 d\tau \quad (25)$$

for any $h_c \in (0, h_p]$. Therefore,

$$V(t) - V(t_k) \leq - \int_{t_k}^t \gamma_z \|P^{a*}(\tau)\|^2 d\tau \quad (26)$$

for any $t \in [t_k, t_{k+1}]$. By applying this recursively, it is implied that for any times t_1, t_2 , where $t_0 \leq t_1 < t_2 \leq \infty$, along the trajectory of the closed-loop system starting from any $Z_0 = Z_\sigma$, the optimal cost function satisfies

$$V(t_2) - V(t_1) \leq - \int_{t_1}^{t_2} \gamma_z \|P^a(\tau)\|^2 d\tau. \quad (27)$$

By induction, (26) states that the optimal cost function is non-increasing for any $h_c \in (0, h_p]$. Since by assumption J^* is bounded at time t_0 , then $V(t_0) < \mu$, and as a result $V(t) < \mu, \forall t \geq t_0$, where μ is positive. Following the argument in Theorem 1 in [11], there exists a constant $\rho > 0$ such that $\|P^a(t_0)\| \leq \rho$ implies $V(t_0) < \mu$ which results in $V(t) < \mu \Rightarrow \|P^a(t)\| < \varrho$ where $\varrho > 0$. By using (27) and induction,

$$V(\infty) - V(t_0) \leq - \int_{t_0}^{\infty} \gamma_z \|P^a(\tau)\|^2 d\tau \quad (28)$$

and since $V(t_0) < \mu$ and $\gamma_z > 0$, by using Barbalat's lemma it is guaranteed that $\|P^a(t)\| \rightarrow 0$ as $t \rightarrow \infty$. $\square$

Note that all the conditions that have been used to prove the convergence are sufficient but not necessary conditions. It would be useful to have an analytic expression for validating convergence of the method, but since the state and control vectors that have been used are implicitly defined in (18), it is difficult to analyze the method in other ways. However, in the next section, the method has been tested and the problem has been solved in a numerical way to show how the DCMPC method achieves the control objective. In addition, there are many other factors, e.g., disturbances, communication delays that could be considered for practical application of the proposed method. These factors are considered as future work.

V. SIMULATION RESULTS

The objective of this section is to assess the properties of the distributed algorithm and illustrate performance in three different traffic scenarios, “Cooperative lane-merging maneuver”, “Double lane-switching”, and “Cooperative intersection”. In addition, the effect of the optimization horizon h_p on the performance of the system is investigated. Also, the effect of different values of the control horizon h_c on the robustness of the method is investigated in a situation where a non-autonomous vehicle is included, and then the difference between the complexity and computational time

of the centralized (CCMPC) and the distributed (DCMPC) methods are illustrated. In basic discrete time receding horizon control, the first control is applied and the remaining controls are discarded and then optimization is redone at each sampling instant. The DCMPC optimization problem is formulated in continuous time and therefore, the problem is discretized before given to a numerical solver. So, the number of optimization variables increases linearly with the optimization horizon h_p and to reduce controller computational complexity a control horizon $h_c \in (0, h_p]$ has been introduced. Controls in the time interval $[t_k, t_k + h_c]$ where t_k is the MPC update time are used before re-optimizing. Thus, a smaller h_c means solving more optimization problems and thereby h_c can be used to tune computational effort. A smaller h_c results in a more agile controller being able to react faster to disturbances and changes in the environment. The number of optimization variables grows linearly with the prediction horizon h_p . Thus, h_c and h_p controls computational effort and disturbance rejection and this is shown with numerical results in this section.

For simplicity, for all scenarios, it is assumed that all vehicles have the same dynamics (1). In order to solve the optimization problem, the Casadi framework [26] in MATLAB has been used. Two of the scenarios, “Cooperative lane-merging maneuver” and the “Cooperative intersection maneuver” are inspired from GDCD [27].

A. Cooperative lane-merging maneuver

In this highway scenario, multiple vehicles in different lanes receive a message from a construction site ahead that the left lane is blocked up the road. The vehicles communicate to merge into a single lane before the site, see Fig. 3-(a). Let D_r , D_s and D_f be the horizontal distance between the two lanes in the road, the initial and the final desired distance between every two consecutive vehicles respectively. Assume that vehicles with odd index numbers are initially moving in the construction site lane (group 1) and the ones with even index numbers are in the other lane. The location of the construction site is $P_{cons} = (0, 180)^T$.

The initial state vector of each vehicle in group 1 is set as

$$z_{(2k_1-1)} = z_{s,g_1} + (k_1 - 1)z_{d,g_1}, \quad k_1 = 1, \dots, n_{g_1}$$

where $z_{s,g_1} = (0, 0, \pi/2, 0)^T$, $z_{d,g_1} = (0, D_s, 0, 0)^T$ and n_{g_1} is the number of vehicles in group 1. The initial state vector for vehicles that are moving in the other lane (group 2), is

$$z_{(2k_2)} = z_{s,g_2} + (k_2 - 1)z_{d,g_2}, \quad k_2 = 1, \dots, n_{g_2}$$

where $z_{s,g_2} = (D_r, 0, \pi/2, 0)^T$ and $z_{d,g_2} = (0, D_s, 0, 0)^T$.

In order to merge the vehicles before the construction site, the final desired position of vehicles is formulated as a function of time in order to let vehicles keep moving on the middle of the lane, continuing regular driving, after finishing the merging maneuver. The desired destination for each vehicle is

$$P_i^f(t) = \begin{cases} P_{cons} + (D_r, (i-1)D_f)^T, & \text{if } |x_i(t_k) - D_r| \geq D_r/2 \\ P_{cons} + (D_r, (i-1)D_f)^T + tv_i(t_k), & \text{otherwise.} \end{cases}$$

The results for this scenario are shown in Fig. 3. Sub-figure 3-(a) shows the three phases related to the scenario, starting,

Fig. 3. The cooperative lane-merging maneuver on highway scenario. Fig. (b) illustrates the trajectories, (c) shows the distances between representative sample pairs of vehicles and (d) is the heading angle during the maneuver.

merging and the end of the scenario. In sub-figures (b) and (d), the blue and red curves correspond to group 1 and 2 respectively. Figures 3-(b) and 3-(c) illustrate the trajectories and distances between pairs of vehicles. Thus, during the maneuver the distance between vehicles is more than the $6^{[m]}$ predefined safe distance, which means that there is no collisions during the maneuver where the vehicles merge successfully. Sub-figure 3-(d) illustrates the heading angle of the vehicles during the maneuver.

B. Double lane-switching Scenario

In this scenario, vehicles are moving in different lanes and aim to switch lanes, see Fig. 4. This is a complex and difficult scenario since many paths are intersecting. In regular traffic, when it comes to busy areas like highways where the incidents may happen while changing from one lane to the other. In the US, more than 200,000 reported lane-change crashes occur annually in recent years; resulting in at least 60,000 injuries [28]. Therefore, it is interesting to handle lane changes using autonomous vehicles. Thus, it is fruitful to have a method for multiple vehicles, e.g., platoons, that controls the vehicles in a cooperative way and let them switch lanes in a safe way.

Initial and desired final positions for each vehicle is defined and the DCMPC algorithm is applied in the double lane-switching maneuver. Assume that vehicles with index numbers $i = 1, \dots, 4$ are initially moving on the left side of the road (group 1) and $i = 5, \dots, 7$ are moving on the right side. The initial state vector of each vehicle in group 1 is set as

$$z_{k_1} = z_{s,g_1} + (k_1 - 1)z_{d,g_1}, \quad k_1 = 1, \dots, n_{g_1}$$

where $z_{s,g_1} = (0, 0, \pi/2, 0)$, $z_{d,g_1} = (0, D_s, 0, 0)$, and n_{g_1} and D_s are defined as before. The initial state vector for vehicles

Fig. 4. The double lane-switching scenario. Fig. (b) illustrates the trajectories, (c) shows the distances between sample representative pairs of vehicles and (d) is the heading angle during the maneuver.

that are moving in the other lane (group 2), is defined as

$$z_{k_2} = z_{s,g_2} + (k_2 - (n_{g_1} + 1))z_{d,g_2}$$

where $k_2 = n_{g_1} + 1, \dots, n_{g_1} + n_{g_2}$, $z_{s,g_2} = (D_r, 0, \pi/2, 0)$, $z_{d,g_2} = (0, D_s, 0, 0)$ and $n_{g_2} = 3$.

The desired destination for the vehicles are given by

$$p_{k_1}^f = \begin{cases} p_{f,g_1} + (k_1 - 1)p_{d,g_1}^f, & \text{if } |x_{k_1}(t_k) - D_r| \geq D_r/2 \\ p_{f,g_1} + (k_1 - 1)p_{d,g_1}^f + tv_{k_1}(t_k), & \text{otherwise} \end{cases}$$

$$p_{k_2}^f = \begin{cases} p_{f,g_2} + (k_2 - (n_{g_1} + 1))p_{d,g_2}^f, & \text{if } |x_{k_2}(t_k) - D_r| \geq D_r/2 \\ p_{f,g_2} + (k_2 - (n_{g_1} + 1))p_{d,g_2}^f + tv_{k_2}(t_k), & \text{otherwise} \end{cases}$$

where, $k_1 = 1, \dots, n_{g_1}$, $p_{f,g_1} = (D_r, 200)$, $p_{d,g_1}^f = (0, D_f)$, $k_2 = n_{g_1} + 1, \dots, n_{g_1} + n_{g_2}$, $p_{f,g_2} = (0, 200)$, $p_{d,g_2}^f = (0, D_f)$ and $n_{g_2} = 3$. The results are shown in Fig. 4 where sub-figure 4-(a) illustrates the three phases related to the scenario, starting, lane change and the end of the scenario. In sub-figures (b) and (d), the blue curves correspond to group 1 and the red curves to group 2. Figures 4-(b) and 4-(c) illustrate the trajectories and distance between some sample pairs of vehicles and show that the vehicles change their lanes successfully. Note that all cars, both in front and at the back, can adjust their velocity depending on the vehicle model, distance of each car to the destination, and the objective function weights. As shown in the figure, there is no collision between vehicles. Sub-figure 4-(d) illustrates the heading angle of the vehicles. Furthermore, Fig. 5 illustrates the compatibility constraint, g_{com_i} , for one vehicle from each group. Note that at some points $g_{com_i} = 0$ and for these points (14) for at least one neighbor of vehicle i is zero and as a result $\eta_i(t_k)$ is equal to zero. Therefore, the uncertain deviation between the estimated and actual predictive trajectories of vehicle i is zero. These points correspond to time

Fig. 5. The compatibility constraint in the double lane-switching maneuver.

instances at which the vehicles were close to their neighbors and therefore, the deviation of p_i from its estimated value must be small in order to avoid collision.

C. Intersection maneuver including non-autonomous vehicle

In MPC, as the control horizon h_c decreases, usually the rejection of unknown disturbances improves. In order to investigate the sensitivity of the DCMPC method to changes of h_c , the cooperative intersection scenario is investigated in this subsection by adding a non-autonomous vehicle to the system.

This scenario demonstrates an intersection coordination activity where one non-autonomous vehicle is approaching a busy road. It intends to do a left turn onto the road and approaching vehicles collaborate to allow it a comfortable and safe passage, see Fig. 6(a). The available information for autonomous vehicles at update time t_k about the non-autonomous vehicle is its position and the directed velocity at t_k . In addition, it is assumed that the autonomous vehicles know the intention of the car and that it is turning on to the right lane. Then the autonomous vehicles assume that this vehicle will continue moving with the constant speed of over the prediction horizon.

Fig. 6 and 7 show the results of using DCMPC method for this scenario. The results show that the system performance depends on the selection of h_c that, in turn, depends on the characteristic of the system and dynamic model. But, since our system is a nonlinear complex system, the optimal value for h_c is not known. Therefore, the results of this subsection and the following one are experimental results.

In MPC, as h_c becomes small, the computational effort increases dramatically. Thus, the optimal choice is a balance between performance and computational effort, which is illustrated in Fig. 7. For a small value of h_c , there is no collision between vehicles and the reason is the fast update which does not allow the speed to be too different from what other vehicles expect. However, as h_c increases, the collision avoidance depends on the speed deviation of the non-autonomous vehicle from the speed at t_k . When the deviation is not large, there are no collisions, but, if the speed deviation is large and h_c is large, the probability of collision is increased because the speed of the non-autonomous car is becoming far from the expectation of the autonomous vehicles. This situation usually does not happen if the moving obstacle is a pedestrian because the acceleration of pedestrian usually cannot be too high unless someone accidentally jumps into the road. For this kind of unexpected situations, a backup controller can be used in order to perform emergency braking maneuvers.

Fig. 6. Results for the cooperative intersection maneuver. (a) shows the three phases related to the scenario. (b) is the trajectory of the vehicles. Sub-figure (c) illustrates the distance between sample representative pairs of vehicles. (d) illustrates the heading angle of the vehicles during the maneuver. In sub-figures (b) and (d), blue curves correspond to one group of vehicles, the red ones are for the other group and the green ones are corresponding to the oncoming non-autonomous vehicle.

Fig. 7. (a) Speed profile, (b) min distance between non-autonomous vehicle and autonomous vehicles for $h_p = 3$ sec.

D. Analysis of the Prediction Horizon

The control horizon h_c controls how often the optimization is performed, and prediction horizon h_p controls how large the optimization problem is at each update. Now, the computation time for different prediction horizon h_p values is compared for the intersection maneuver when all 7 vehicles are autonomous. Fig. 8-(a) shows a comparison of the average computation time as time proceeds. For each second after the initialization, the data corresponds to the average computational time. It was observed that as the prediction horizon h_p is becoming shorter, the average computation time decreases. For the 3 seconds prediction horizon, the average computation time was relatively small and the trajectories of the vehicles were smooth.

Fig. 8-(b) shows the optimal heading angle, θ^* , of one of the vehicles for different prediction times. A prediction horizon closer to the rise time of the system results in a feasible optimization. As noted in [5], [11], a larger prediction horizon leads to better tracking performance, see Fig. 8.

E. Centralized vs. distributed MPC

One factor, which is important to take into account in multi-vehicle systems control, is computation time. The average

Fig. 8. (a) The average of computational time of DCMPC with different prediction horizon for each vehicle and for each iteration over the entire intersection maneuver and for $h_c = 1$ sec, and (b) the heading angle.

n	Normalized Computational time	
	CCMPC	DCMPC
2	0.0286	0.0279
3	0.1052	0.0368
4	0.2237	0.0498
5	0.4038	0.0608
6	0.6547	0.0682
7	1	0.0768

Fig. 9. Normalized Computational time versus number of vehicles.

TABLE I
NORMALIZED COMPUTATIONAL TIME.

normalized computational time for each controller for the double lane-switching scenario is provided in Table I. As mentioned in Section III, a centralized MPC scheme to control multiple vehicles provides a better performance, according to the global cost function for the system. However, the communication and computational cost of implementing a centralized MPC may be too expensive in a multiagent application. The simulations show that a balanced trade-off between optimality performance and computation time is provided by using distributed MPC.

For comparison, the DCMPC and CCMPC controller were both applied to scenarios with 2–7 vehicles. All the simulations have been done using Matlab R2018b on an Intel Core i7-3770, 3.40GHz processor, and CentOS Linux 64-bit machine. The results showed that when the system contains more than 2 vehicles, the DCMPC provides a considerably lower computational time compared to the CCMPC. In Fig. 9, the average normalized controller computation time of both controllers is illustrated. The computational time for the centralized controller is divided with the number of vehicles to be able to compare the centralized with the distributed controller. One time unit in the figure is 2.63 seconds and the corresponding computational time in the distributed case is 0.2 seconds. As a real-time indicator, the the control horizon, h_c , for the MPC controller is 1 second. Computational times are presented in normalized form mainly for two reasons: the implementation is not optimized for computational performance and to keep the presentation as implementation independent as possible. The computational time for CCMPC in our problem is approximately a quadratic function of the number of vehicles, while for the distributed is approximately a linear function. Since the size of the distributed optimal problem for each vehicle does not increase too much by increasing the number of vehicles, the computational time for DCMPC is a linear function of the number of vehicles and clearly, the DCMPC algorithm is computationally advantageous for the larger number of vehicles.

VI. CONCLUSION

In this paper, a cooperative control approach for autonomous vehicles was proposed to address traffic scenarios involving several autonomous vehicles. The problem was formulated as a distributed optimal control problem and the proposed formulation and constraints provide a collision-free trajectory for each vehicle. This formulation has a generality property in the sense that it can be extended to be applied to vehicles with other kinematic models, and as it was shown in the simulation results, that it can be used for different traffic scenarios. In addition, it was proven that by using the proposed method, each vehicle has a feasible solution and will converge to the desired target. Furthermore, not only do the trajectories for the vehicles converge to desired targets, simulation in a number of scenarios show smooth and well-behaved vehicle behavior. Further, due to the distributed property of the proposed method, the method scales efficiently and can be used to control larger groups of vehicles.

REFERENCES

- [1] A. Vahidi and A. Sciarretta, "Energy saving potentials of connected and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 822 – 843, 2018.
- [2] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical Systems and Signal Processing*, vol. 100, pp. 482 – 500, 2018.
- [3] A. Veksler, T. A. Johansen, F. Borrelli, and B. Realfsen, "Dynamic positioning with model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1340–1353, July 2016.
- [4] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems Measurement and Control*, vol. 129, no. 5, pp. 571–583, 2007.
- [5] C. Wang and C. J. Ong, "Distributed model predictive control of dynamically decoupled systems with coupled cost," *Automatica*, vol. 46, no. 12, pp. 2053 – 2058, 2010.
- [6] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2483–2499, 2019.
- [7] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 7, pp. 1616–1629, Aug 2008.
- [8] R. Zheng, K. Nakano, S. Yamabe, M. Aki, H. Nakamura, and Y. Suda, "Study on emergency-avoidance braking for the automatic platooning of trucks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, pp. 1748–1757, 2014.
- [9] L. Xu, L. Wang, G. Yin, and H. Zhang, "Communication information structures and contents for enhanced safety of highway vehicle platoons," *IEEE Transactions on Vehicular Technology*, vol. 63, pp. 4206–4220, 2014.
- [10] F. Mohseni, E. Frisk, J. Åslund, and L. Nielsen, "Distributed model predictive control for highway maneuvers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8531–8536, 2017.
- [11] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205 – 1217, 1998.
- [12] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems*, vol. 22, no. 1, pp. 44–52, Feb 2002.
- [13] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, Apr. 2006.
- [14] B. Ding, L. Xie, and W. Cai, "Distributed model predictive control for constrained linear systems," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 11, pp. 1285–1298, 2010.
- [15] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 287–299, Sep. 2018.
- [16] Y. Bichiou and H. A. Rakha, "Real-time optimal intersection control system for automated/cooperative vehicles," *International Journal of Transportation Science and Technology*, vol. 8, no. 1, pp. 1 – 12, 2019.
- [17] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.
- [18] C. Samson, "Control of chained systems application to path following and time-varying point-stabilization of mobile robots," *IEEEA. Isidori, Nonlinear Control Systems, 3rd ed. Springer-Verlag, (1995). Transactions on Automatic Control*, vol. 40, no. 1, pp. 64–77, Jan 1995.
- [19] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, Dec 2004.
- [20] A. De Luca, G. Oriolo, and C. Samson, *Feedback control of a nonholonomic car-like robot*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 171–253.
- [21] A. Isidori, *Nonlinear control systems*. Springer Science & Business Media, 2013.
- [22] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.
- [23] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 743–749, July 2006.
- [24] H. Genceli and M. Nikolaou, "Robust stability analysis of constrained 11-norm model predictive control," *AIChE Journal*, vol. 39, no. 12, pp. 1954–1965, 1993.
- [25] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1623–1633, Nov 1993.
- [26] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [27] J. Ploeg, E. Semsar-Kazerooni, A. I. Morales Medina, J. F. C. M. de Jongh, J. van de Sluis, A. Voronov, C. Englund, R. J. Bril, H. Salunkhe, A. Arrúe, A. Ruano, L. García-Sol, E. van Nunen, and N. van de Wouw, "Cooperative automated maneuvering at the 2016 grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1213–1226, April 2018.
- [28] G. Fitch, S. Lee, S. Klauer, J. Hankey, J. Sudweeks, and T. Dingus, "Analysis of lane-change crashes and near-crashes," *US Department of Transportation, National Highway Traffic Safety Administration*, 2009.

Fatemeh Mohseni was born in Tehran, Iran in 1984. She received the M.Sc. degree in electrical engineering from Amirkabir University of Technology (formerly Tehran Polytechnic), Tehran, Iran, in 2012. Since 2015 she is PhD student at the Department of Electrical Engineering, Linköping University, Sweden.

Erik Frisk was born in Stockholm, Sweden in 1971 and is a professor at the Department of Electrical Engineering, Linköping University, Sweden. He finished his PhD in 2001 and a main research interest has been model-based fault diagnostics and prognostics. Another main area of research is optimization techniques for autonomous vehicles in complex traffic scenarios.

Lars Nielsen was born in Sweden in 1955. He received the Ph.D. degree in automatic control from Lund University, Sweden, in 1985. Since 1992, he is the Sten Gustafsson Professor of Vehicular Systems at Linköping University, Sweden. His research interests include automotive control, diagnosis, and autonomy.