Mixed-integer Programming for Centralized Coordination of Connected and Automated Vehicles in Dynamic Environment

Ravikumar, Shreejith; Quirynen, Rien; Bhagat, Akshay; Zeino, Eyad; Di Cairano, Stefano

TR2021-089 August 10, 2021

Abstract

Connected and automated vehicles (CAVs) have shown the potential to improve safety, increase throughput, and optimize energy efficiency and emissions in complicated traffic scenarios. This paper presents a mixed-integer linear programming (MILP) method for scheduling and coordination of CAVs in a highly dynamic environment that consists of multiple human-driven vehicles and multiple conflict zones, such as merging points and intersections. The proposed approach ensures safety, high throughput and energy efficiency by solving a centralized high-level decision making problem. The solution provides a feasible and optimal time schedule through road segments and conflict zones for the automated vehicles, by using information from the position, velocity, and destination of the manual vehicles, which cannot be directly controlled. The performance and computational load of the proposed method are assessed in closed-loop simulations on an illustrative scenario. Despite MILP having combinatorial complexity, the proposed formulation appears feasible for realtime implementation, e.g., in mobile edge computers (MECs).

IEEE Conference on Control Technology and Applications (CCTA)

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Mixed-integer Programming for Centralized Coordination of Connected and Automated Vehicles in Dynamic Environment

Shreejith Ravikumar[1], Rien Quirynen[1], Akshay Bhagat[2], Eyad Zeino[2] and Stefano Di Cairano[1]

Abstract— **Connected and automated vehicles (CAVs) have shown the potential to improve safety, increase throughput, and optimize energy efficiency and emissions in complicated traffic scenarios. This paper presents a mixed-integer linear programming (MILP) method for scheduling and coordination of CAVs in a highly dynamic environment that consists of multiple human-driven vehicles and multiple conflict zones, such as merging points and intersections. The proposed approach ensures safety, high throughput and energy efficiency by solving a centralized high-level decision making problem. The solution provides a feasible and optimal time schedule through road segments and conflict zones for the automated vehicles, by using information from the position, velocity, and destination of the manual vehicles, which cannot be directly controlled. The performance and computational load of the proposed method are assessed in closed-loop simulations on an illustrative scenario. Despite MILP having combinatorial complexity, the proposed formulation appears feasible for real-time implementation, e.g., in mobile edge computers (MECs).**

I. Introduction

Automated transportation systems, even in the case of partial automation, lead to reduced road accidents and more efficient usage of the road network [1]. Therefore, connected and automated vehicles (CAVs) show large promises for improving safety and traffic flow, and as a consequence for reducing congestion, travel time, emissions and energy consumption [2]. While this has been known for decades, most of the successful developments have been accomplished in recent years due to the technological advances in sensing, computing, control and connectivity. While the on-road scenarios often are highly dynamic, i.e, the vehicle participants and their behavior changes rapidly and significantly, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, also known as vehicle-to-everything (V2X) for short, enables advanced and efficient planning and decision making by providing access to real-time information on all the vehicles in a certain planning area [3], [4].

Significant progress has been made in planning and control for automated driving [5], which typically involves a multi-layer on-board guidance and control architecture [6]. At the highest level, an intelligent navigation system finds a route through the transportation network from the current vehicle position to the requested destination [7]. A decision maker selects the appropriate driving behavior at any point of time, given the route plan, the current environment condition, and the behavior of other traffic participants, e.g., using automata combined with set reachability [8] or formal languages and optimization [9]. Given the target behavior, a motion planning algorithm computes a dynamically feasible and safe trajectory that can be tracked in real-time by a low-level controller. A popular approach uses the combination of a sampling-based motion planner [10] and model predictive control (MPC) for reference tracking [11]. For CAVs, the guidance and control architecture may look similar, but some of the modules are implemented in the infrastructure, e.g., in mobile edge computers (MECs) and provide decisions to multiple vehicles in the area, while other modules are still implemented on board of each vehicle individually [3], [4].

This paper focuses on the centralized scheduling and coordination of CAVs [12], which provides targets and operational information to a motion planning and tracking system implemented separately in each automated vehicle, and in the presence of on-road vehicles that are human-driven, i.e., manual, referred to as non-controlled vehicles (NCVs). Recently, coordination strategies for intersection control have been proposed using nonlinear optimization in [13] or using mixed-integer linear programming (MILP) in [14]. The latter has been extended to a distributed MILP algorithm for scheduling a grid of interconnected intersections in [15]. An MILP-based approach for on-ramp merging of CAVs was proposed in [3]. Alternative techniques for coordination of CAVs can be found in [1], and references therein, where it can be also noted that the intersection and merging control problems are very similar in nature.

In this paper, we propose an MILP approach for centralized coordination of vehicles in an interconnected network of generalized conflict zones, including both intersections and merging points. Unlike the work in [13], [14] for all-autonomous vehicle coordination, we focus on the more realistic scenario that includes mixed traffic of both CAVs and human-driven NCVs. As discussed in [14], physical traffic lights and/or standard priority rules are needed for human-driven vehicles to cross intersections in the mixed traffic scenario. The proposed formulation directly incorporates the transportation of people and goods, given a real-time sequence of vehicle routing information.

The proposed MILP approach computes a schedule consisting of target velocities and times to enter and exit the road segments in a prediction window for each vehicle towards its desired destination, under safety constraints for conflict zones and occupancy constraints for road segments, and while optimizing the overall time and energy efficiency across all controlled vehicles. As opposed to some of the

[1] Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139, USA {ravikumar,quirynen,dicairano}@merl.com.

[2] Mitsubishi Electric Automotive America, Northville, MI, 48168, USA {ABhagat,EZeino}@meaa.mea.com.

previously cited works, the proposed approach supports

- a transportation network of multiple conflict zones, i.e., merging points and/or intersections,
- mixed traffic including both autonomous and human-driven vehicles, where position, speed and heading for each vehicle is available through V2X communication, but only some of the vehicles are controlled,
- minimization of multiple objectives, e.g., travel time, waiting time and energy efficiency,
- routing information for transport of people and goods.

The operation, performance and real-time feasibility of the proposed MILP-based implementation is validated based on closed-loop simulation results on an illustrative scenario.

The rest of the paper is organized as follows. Section II introduces the problem setup. Section III presents the proposed MILP for coordination of connected vehicles. The performance and real-time feasibility are illustrated on closed-loop simulations in Section IV, followed by our conclusions and future outlook in Section V.

II. Problem Setup for Centralized Coordination

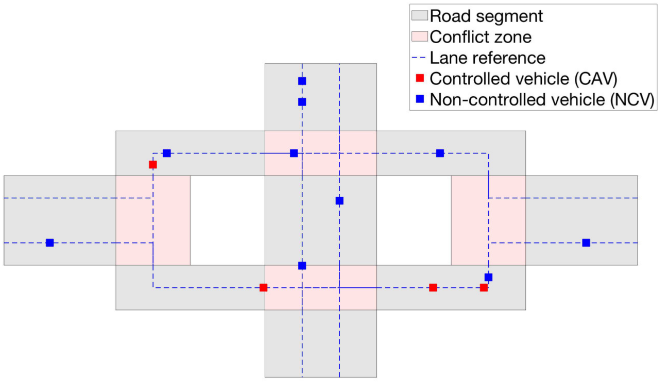

Fig. 1. Illustration of problem setup for centralized coordination, including both controlled (CAVs) and non-controlled vehicles (NCVs) in a local area of multiple interconnected conflict zones and road segments.

The road network is composed of n_s road segments, with indices in the set $\mathcal{J}_\text{s} = \{1, \ldots, n_\text{s}\}$. The set $\mathcal{C} \subset \mathcal{J}_\text{s}$ denotes the index set of road segments that are conflict zones, such as intersections and merge points, that can be occupied by a single vehicle at all times. The set $\mathcal{F} \subset \mathcal{J}_\text{s}$ denotes the index set of road segments that are conflict-free zones, e.g., standard driving lanes that may be occupied by multiple vehicles at the same time. Hence, $\mathcal{F} = \mathcal{J}_\text{s} \backslash \mathcal{C}$ and $\mathcal{J}_\text{s} = \mathcal{F} \cup \mathcal{C}$.

Differently from the fully autonomous case, for the CAV case, the target behavior for the motion planner is not computed separately for each vehicle by an on-board decision making algorithm, e.g., in [8], [9], but rather for all vehicles at the same time in a centralized coordination and scheduling module. Specifically, given real-time information on the environment from the mapping and navigation module, on each CAV's state, and on the routing of each CAV, the centralized vehicle coordination and scheduling module determines the target behaviors for all CAVs in the area, at the same time.

As illustrated in Figure 1, we consider a scenario where both CAVs (in red) and NCVs (in blue) are present. Let n_c be the number of CAVs, with index set $\mathcal{I}_\text{c} = \{1, \ldots, n_\text{c}\}$, and n_nc be the number of NCVs, with index set $\mathcal{I}_\text{nc} = \{n_\text{c}+1, \ldots, n_\text{c}+n_\text{nc}\}$. Hence, in total we consider $n_\text{c}+n_\text{nc}$ vehicles, with index set $\mathcal{I}_\text{v} = \mathcal{I}_\text{c} \cup \mathcal{I}_\text{nc}$.

The information for the centralized vehicle coordination and scheduling module is assumed to be available through communication between vehicles and infrastructure, i.e., V2X, of data acquired by sensors in both CAVs and NCVs, and possibly in the infrastructure too, e.g., see [16]. Specifically, the inputs to the vehicle coordination system are

- map information $\mathcal{M}$ for the transportation network that is considered by the coordination module, including lane information for each road segment and conflict zone, i.e., merging points and/or intersections.
- for each vehicle α^i, $i \in \mathcal{I}_\text{v}$, i.e., both CAVs and NCVs, the current position p^i and velocity v^i.
- for each NCV α^i, $i \in \mathcal{I}_\text{nc}$, a possibly short-term route prediction defined as a sequence $\phi^i = \left[\phi^i(1), \ldots, \phi^i(n_\phi^i)\right]$ of road segment indices, $\phi^i(j) \in \mathcal{J}_\text{s}$, where n_ϕ^i is the route length in terms of segments.
- for each CAV α^i, $i \in \mathcal{I}_\text{c}$, a relatively long-term route plan determined as sequence $\phi^i = \left[\phi^i(1), \ldots, \phi^i(n_\phi^i)\right]$ of road segment indices, $\phi^i(j) \in \mathcal{J}_\text{s}$, where n_ϕ^i is the route length in terms of segments.
- for each CAV α^i, $i \in \mathcal{I}_\text{c}$, and for each segment in its planned route $j \in \phi^i$, a set of planned stop durations for each segment along the route $\bar{t}^i_\text{wait} = \left[\bar{t}^i_\text{wait}(1), \ldots, \bar{t}^i_\text{wait}(n_\phi^i)\right]$, e.g., for transportation.

Similarly, the output information from the centralized MILP-based coordination module is

- for each CAV α^i, $i \in \mathcal{I}_\text{c}$, the sequence of average velocities $v^i = \left[v^i(1), \ldots, v^i(n_\phi^i)\right]$ for the road segments along its planned route ϕ^i.
- for each CAV α^i, $i \in \mathcal{I}_\text{c}$, the sequence of expected entering times $t^i_\text{in} = \left[t^i_\text{in}(1), \ldots, t^i_\text{in}(n_\phi^i)\right]$ for the road segments along its planned route ϕ^i.
- for each CAV α^i, $i \in \mathcal{I}_\text{c}$, the sequence of expected exit times $t^i_\text{out} = \left[t^i_\text{out}(1), \ldots, t^i_\text{out}(n_\phi^i)\right]$ for the road segments along its planned route ϕ^i.

Depending on the system, obtaining a precise prediction for the NCV route may prove challenging. Because of this, the vehicle coordination and scheduling module is implemented in a receding horizon fashion based on the most recent information. An approximate short-term route prediction, $n_\phi^i \ll n_\phi^j$, for all $i \in \mathcal{I}_\text{nc}$, $j \in \mathcal{I}_\text{c}$, for NCVs is sufficient, e.g., until the next conflict zone, and any discrepancies in the predictions are adjusted by the intrinsic feedback mechanism of the receding horizon strategy. An update period of $1-5$ seconds allows for real-time computation of the scheduling, while providing sufficiently fast updates to account for erroneous prediction of NCV behaviors.

III. Mixed-integer Programming Formulation

Next, we describe the MILP formulation that needs to be solved to implement the centralized vehicle coordination and scheduling module.

A. Parameters and Decision Variables

We make a few simplifications for ease of implementation and for clarity. First, we consider routes that do not involve loops, so that each segment $j \in \mathcal{J}_\mathrm{s}$ is included in ϕ^i at most once, for all $i \in \mathcal{I}_\mathrm{v}$. This condition could be removed simply by adding more index values to the same road segment. Second, while we can directly control only CAVs, for obtaining a simple structure in the formulation of the optimization problem we define variables for all the vehicles, i.e., CAVs and NCVs, and then formulate the problem so that the NCVs behave according to their prediction. As discussed further, the latter additionally ensures that a feasible solution always exists for the proposed MILP. Finally, by considering a simple kinematic model with constant segment velocity v, the distance $\overline{d}$ travelled from time t_in to time t_out with an intermediate stopping time $\overline{t}_\mathrm{wait}$ is

$$\overline{d} = v \cdot (t_\mathrm{out} - t_\mathrm{in} - \overline{t}_\mathrm{wait}). \tag{1}$$

Thus, the decision variables are:

- $t_\mathrm{out}^{i,j}$, $i \in \mathcal{I}_\mathrm{v}$, $j \in \mathcal{J}_\mathrm{s}$, the predicted time at which vehicle α^i, $i \in \mathcal{I}_\mathrm{v}$ *exits* segment $j \in \mathcal{J}_\mathrm{s}$,
- $t_\mathrm{in}^{i,j}$, $i \in \mathcal{I}_\mathrm{v}$, $j \in \mathcal{J}_\mathrm{s}$, the predicted time at which vehicle α^i, $i \in \mathcal{I}_\mathrm{v}$ *enters* segment $j \in \mathcal{J}_\mathrm{s}$,
- $v_\mathrm{inv}^{i,j}$, $i \in \mathcal{I}_\mathrm{v}$, $j \in \mathcal{J}_\mathrm{s}$, the inverse of the velocity, i.e., $1/v$, of vehicle $i \in \mathcal{I}_\mathrm{v}$ on segment $j \in \mathcal{J}_\mathrm{s}$,
- $b^{i,j,k} \in \{0,1\}$, $i \in \mathcal{I}_\mathrm{c}$, $k \in \{\ell \in \mathcal{I}_\mathrm{c},\ \ell \leq i-1\} \cup \mathcal{I}_\mathrm{nc}$, $j \in \mathcal{J}_\mathrm{s}$ is the binary variable that determines whether vehicle i crosses segment j ahead of vehicle k.

The input data for the optimization problem is obtained from the available road and vehicle data, see Section II, and it includes the following:

- the routes ϕ^i, $i \in \mathcal{I}_\mathrm{v}$, i.e., each with length n_ϕ^i, which is long for CAVs and usually short for NCVs, respectively.
- the maximum velocity $\overline{v}_\mathrm{max}^j$, for each segment $j \in \mathcal{J}_\mathrm{s}$.
- the wait times $\overline{t}_\mathrm{wait}^{i,j}$, $i \in \mathcal{I}_\mathrm{v}$, $j \in \mathcal{J}_\mathrm{s}$ for any planned stop of vehicle $i \in \mathcal{I}_\mathrm{v}$ in segment $j \in \mathcal{J}_\mathrm{s}$, e.g., for transportation, where $\overline{t}_\mathrm{wait}^{i,j} = 0$ for all $i \in \mathcal{I}_\mathrm{nc}$.
- the distance $\overline{d}^{i,j}$, $i \in \mathcal{I}_\mathrm{v}$, $j \in \mathcal{J}_\mathrm{s}$ that vehicle $i \in \mathcal{I}_\mathrm{v}$ must travel in segment $j \in \mathcal{J}_\mathrm{s}$ to reach its end, where

$$\overline{d}^{i,j} = \begin{cases} 0 & \text{if} \quad j \notin \phi^i \\ \overline{d}^j - \mathcal{D}^j(p^i) & \text{if} \quad j \in \phi^i \end{cases}, \tag{2}$$

  $\overline{d}^j$ is the total length of segment $j \in \mathcal{J}_\mathrm{s}$, and $\mathcal{D}^j(p)$ is the progress already accomplished in segment $j \in \mathcal{J}_\mathrm{s}$ according to the position p, and $\mathcal{D}^j(p) = 0$ if the position p is not in segment $j \in \mathcal{J}_\mathrm{s}$.
- the current velocity $\overline{v}^i$, $i \in \mathcal{I}_\mathrm{v}$ for each vehicle.

As previously mentioned, the problem includes time and velocity variables for both controlled and non-controlled vehicles. The formulation is such that the time schedule and velocity for NCVs is equal to their current prediction, unless a particular vehicle is required to slow down due to traffic conditions. For example, the proposed MILP would predict that an NCV following a slower vehicle, will reduce its velocity if necessary to avoid collision. On the other hand, the MILP can choose any feasible time schedule and corresponding average velocities for each of the CAVs in order to minimize the global objective function.

B. Constraints and Objective Function

We introduce the constraints for the MILP problem that enforce each of the requirements for the vehicle coordination and scheduling module.

1) Vehicle motion model: The kinematic one-dimensional motion model (1), where we take as variables the entering and exiting time t_in, t_out and the inverse of the velocity v_inv, results in the linear equality constraints

$$t_\mathrm{out}^{i,j} - t_\mathrm{in}^{i,j} = \overline{d}^{i,j} v_\mathrm{inv}^{i,j} + \overline{t}_\mathrm{wait}^{i,j}, \quad i \in \mathcal{I}_\mathrm{v},\ j \in \mathcal{J}_\mathrm{s}. \tag{3}$$

The maximum velocity constraints for each controlled vehicle in each segment also result in

$$0 \leq \frac{1}{\overline{v}_\mathrm{max}^j} \leq v_\mathrm{inv}^{i,j}, \quad i \in \mathcal{I}_\mathrm{c}, j \in \mathcal{J}_\mathrm{s}, \tag{4a}$$

$$0 \leq \frac{1}{\overline{v}^i} \leq v_\mathrm{inv}^{i,j}, \qquad i \in \mathcal{I}_\mathrm{nc}, j \in \mathcal{J}_\mathrm{s}, \tag{4b}$$

where $\overline{v}^i$ is the current velocity for vehicle α^i, $i \in \mathcal{I}_\mathrm{nc}$, because the MILP is only allowed to predict a slowdown for non-controlled vehicles to ensure feasibility.

2) Timing constraints of route plan: The route plan $\phi^i = [\phi^i(1), \ldots, \phi^i(n_\phi^i)]$ for each vehicle $i \in \mathcal{I}_\mathrm{v}$ enforces the order between exiting and entering subsequent road segments $\phi^i(k-1)$ and $\phi^i(k)$ as follows

$$(t_\mathrm{out}^{i,\phi^i(k-1)} - t_\mathrm{in}^{i,\phi^i(k)}) \leq 0, \quad i \in \mathcal{I}_\mathrm{v},\ k \in \mathbb{Z}_2^{n_\phi^i}. \tag{5}$$

We additionally introduce the following compact notation $\psi^i(j)$ to denote the *next* segment, after segment j, in the route plan for vehicle i. Therefore, in case of two subsequent road segments j and $l = \psi^i(j)$ for vehicle i, there exists an index k such that $j = \phi^i(k)$ and $l = \phi^i(k+1)$.

3) Safety constraints for conflict zones: Each CAV cannot enter a conflict zone while another vehicle is present in that same zone. This results in the following constraints

$$\left(t_\mathrm{out}^{i,j} \leq t_\mathrm{in}^{k,j} - \epsilon\right) \quad \vee \quad \left(t_\mathrm{out}^{k,j} \leq t_\mathrm{in}^{i,j} - \epsilon\right), \tag{6}$$

for scheduling the order between vehicle i and k within the conflict zone $j \in \mathcal{C}$, where $i \in \mathcal{I}_\mathrm{c}$, i.e., α^i is a CAV, and $k \in \{\ell \in \mathcal{I}_\mathrm{c},\ \ell \leq i-1\} \cup \mathcal{I}_\mathrm{nc}$ is either a CAV or a NCV, where we just need to enforce the constraints for the vehicles with a higher index, and where ϵ denotes a tolerance on timing. The constraints in (6) are implemented using auxiliary binary variables $b^{i,j,k}$ and the big-M formulation

$$\forall i \in \mathcal{I}_\mathrm{c},\ j \in \mathcal{C},\ k \in \{\ell \in \mathcal{I}_\mathrm{c},\ \ell \leq i-1\} \cup \mathcal{I}_\mathrm{nc} :$$

$$(t_\mathrm{out}^{i,j} - t_\mathrm{in}^{k,j} - M\, b^{i,j,k}) \leq -\epsilon, \tag{7a}$$

$$(t_\mathrm{out}^{k,j} - t_\mathrm{in}^{i,j} + M\, b^{i,j,k}) \leq M - \epsilon, \tag{7b}$$

where M is a larger number compared to the other terms in the constraints. Note that the uniqueness constraint in (6) can be conservative in certain situations, e.g., if the vehicles motions allow them to share the intersection without causing conflicts. Such situations can be accounted for by enforcing constraint (6) only when two vehicles motions through the conflict zone do not allow for them to share it.

4) Occupancy constraints for free road segments: In this work, we consider roads with a single lane per direction, and hence overtaking maneuvers are not allowed, as common in city driving. Thus, the order between vehicles needs to be preserved across two subsequent road segments resulting in the following vehicle order constraint

$$\begin{aligned}&\left(t_{\text{in}}^{i,j} \le t_{\text{in}}^{k,j} - \epsilon \,\wedge\, t_{\text{out}}^{i,j} \le t_{\text{out}}^{k,j} - \epsilon \,\wedge\, t_{\text{in}}^{i,l} \le t_{\text{in}}^{k,l} - \epsilon\right)\\ \vee\, &\left(t_{\text{in}}^{k,j} \le t_{\text{in}}^{i,j} - \epsilon \,\wedge\, t_{\text{out}}^{k,j} \le t_{\text{out}}^{i,j} - \epsilon \,\wedge\, t_{\text{in}}^{k,l} \le t_{\text{in}}^{i,l} - \epsilon\right),\end{aligned} \tag{8}$$

within each conflict-free segment $j \in \mathcal{F}$ for two vehicles α^i, α^k, $i \in \mathcal{I}_{\text{c}}$, $k \in \{\ell \in \mathcal{I}_{\text{c}},\ \ell \le i-1\} \cup \mathcal{I}_{\text{nc}}$, if a subsequent segment l exists for which $l = \psi^i(j) = \psi^k(j)$. The constraint in (8) can be implemented again using the binary variables $b^{i,j,k}$ and the big-M formulation

$$\begin{aligned}&\forall i \in \mathcal{I}_{\text{c}},\ j \in \mathcal{F},\ k \in \{\ell \in \mathcal{I}_{\text{c}},\ \ell \le i-1\} \cup \mathcal{I}_{\text{nc}},\\ &\quad l = \psi^i(j) = \psi^k(j):\\ &\quad (t_{\text{in}}^{i,j} - t_{\text{in}}^{k,j} - M\, b^{i,j,k}) \le -\epsilon, && \text{(9a)}\\ &\quad (t_{\text{in}}^{k,j} - t_{\text{in}}^{i,j} + M\, b^{i,j,k}) \le M - \epsilon, && \text{(9b)}\\ &\quad (t_{\text{out}}^{i,j} - t_{\text{out}}^{k,j} - M\, b^{i,j,k}) \le -\epsilon, && \text{(9c)}\\ &\quad (t_{\text{out}}^{k,j} - t_{\text{out}}^{i,j} + M\, b^{i,j,k}) \le M - \epsilon, && \text{(9d)}\\ &\quad (t_{\text{in}}^{i,l} - t_{\text{in}}^{k,l} - M\, b^{i,j,k}) \le -\epsilon, && \text{(9e)}\\ &\quad (t_{\text{in}}^{k,l} - t_{\text{in}}^{i,l} + M\, b^{i,j,k}) \le M - \epsilon. && \text{(9f)}\end{aligned}$$

5) Objective function terms: We consider an objective function with several terms modeling different goals

$$J(\mathcal{T},\mathcal{V}) = J_t(\mathcal{T},\mathcal{V}) + J_w(\mathcal{T},\mathcal{V}) + J_p(\mathcal{T},\mathcal{V}) + J_a(\mathcal{T},\mathcal{V}), \tag{10}$$

where $\mathcal{T} = \{t_{\text{in}}^{i,j}, t_{\text{out}}^{i,j}\}_{i \in \mathcal{I}_{\text{v}}, j \in \mathcal{J}_{\text{s}}}$ and $\mathcal{V} = \{v_{\text{inv}}^{i,j}\}_{i \in \mathcal{I}_{\text{v}}, j \in \mathcal{J}_{\text{s}}}$ are defined. The first objective term

$$J_t(\mathcal{T},\mathcal{V}) = \omega_1 \sum_{i \in \mathcal{I}_{\text{c}}} \sum_{k=1}^{n_\phi^i} \frac{t_{\text{out}}^{i,\phi^i(k)}}{\overline{d}_{\text{tot}}^{i,k}}, \tag{11}$$

where $\overline{d}_{\text{tot}}^{i,k} = \sum_{j=1}^{k} \overline{d}^{i,\phi^i(j)}$ denotes the total distance traveled until the end of the k^{th} segment in the route plan for CAV α^i, $i \in \mathcal{I}_{\text{c}}$, and it aims at minimizing the travel time to reach the end of each segment in the route plan with corresponding weight value $\omega_1 \ge 0$ and normalized by the traveled distance for each CAV. The term

$$\begin{aligned}J_w(\mathcal{T},\mathcal{V}) = \omega_2 &\sum_{i \in \mathcal{I}_{\text{c}}} \sum_{k=2}^{n_\phi^i} \left(t_{\text{in}}^{i,\phi^i(k)} - t_{\text{out}}^{i,\phi^i(k-1)}\right)\\ + \omega_4 &\sum_{i \in \mathcal{I}_{\text{nc}}} \sum_{k=2}^{n_\phi^i} \left(t_{\text{in}}^{i,\phi^i(k)} - t_{\text{out}}^{i,\phi^i(k-1)}\right)\end{aligned} \tag{12}$$

aims at minimizing the sum of waiting times between two subsequent road segments $\phi^i(k-1)$ and $\phi^i(k)$ for each vehicle i with weights $\omega_2 \ge 0$ and $\omega_4 \ge 0$ for CAVs and NCVs, respectively. To prevent predicting unrealistic changes to the speed of the NCVs, the term

$$J_p(\mathcal{T},\mathcal{V}) = \omega_3 \sum_{i \in \mathcal{I}_{\text{nc}}} \sum_{j \in \mathcal{J}_{\text{s}}} v_{\text{inv}}^{i,j} \tag{13}$$

aims at maximizing the velocity of NCVs. By setting $\omega_3 \gg \omega_p$, $p = 1, 2, 4, 5$, together with constraint (4b) that sets the NCVs velocity upper bounds at their predicted velocity, this forces the MILP solution to plan NCVs velocities at their predicted values, unless the NCVs must necessarily slow down for feasibility, e.g., in case of slow traffic ahead.

A final term of the cost function can penalize any changes in the velocity from one segment to the next,

$$J_a(\mathcal{T},\mathcal{V}) = \omega_5 \sum_{i \in \mathcal{I}_{\text{v}}} \sum_{k=1}^{n_\phi^i} \left| v_{\text{inv}}^{i,\phi^i(k)} - v_{\text{inv}}^{i,\phi^i(k-1)} \right|, \tag{14}$$

where $v_{\text{inv}}^{i,\phi^i(0)} = \frac{1}{\overline{v}^i}$, i.e., it denotes the inverse of the current velocity for vehicle α^i, $i \in \mathcal{I}_{\text{v}}$, in an effort to reduce accelerations/decelerations and increase energy efficiency, with corresponding weight value $\omega_5 \ge 0$.

Remark 1: The summation over vehicles in the objective terms of eqs. (11)-(14) could be replaced by the maximum operator. The latter corresponds to using the ℓ_∞ instead of the ℓ_1 norm and still results in an MILP. For simplicity, we further restrict to the summation-based ℓ_1 formulation.

C. Problem Dimensions and Redundant Variables

The complete MILP has following dimensions:

- # of continuous optimization variables: $3\, n_{\text{s}}\, (n_{\text{c}} + n_{\text{nc}})$, including time and inverse velocity variables for each vehicle $i \in \mathcal{I}_{\text{v}}$ on each road segment $j \in \mathcal{J}_{\text{s}}$,
- # of binary optimization variables: $n_{\text{s}}\, n_{\text{c}} \left(n_{\text{nc}} + \frac{n_{\text{c}}-1}{2}\right)$, including one binary variable for each unique pair of vehicles on each road segment,
- # of equality constraints: $n_{\text{s}}\, (n_{\text{c}} + n_{\text{nc}})$,
- # of inequality constraints (excluding simple bounds): $\sum_{i=1}^{n_{\text{c}}+n_{\text{nc}}} \left(n_\phi^i - 1\right) + n_{\text{c}}\, \left(2\,|\mathcal{C}| + 6\,|\mathcal{F}|\right) \left(n_{\text{nc}} + \frac{n_{\text{c}}-1}{2}\right)$,

where $|\cdot|$ denotes the cardinality of a set, i.e., $|\mathcal{C}|$ and $|\mathcal{F}|$ are the total number of conflict and conflict-free segments, respectively. For practical applications of vehicle coordination in a receding horizon implementation, it is desirable for the MILP to have fixed dimensions for a certain upper bound of vehicles and road segments. Redundant optimization variables are removed automatically by the pre-solve routine in MILP solvers as discussed in [17], or they can be fixed explicitly by adjusting the corresponding simple bounds for each redundant optimization variable in the MILP.

Note that the inverse velocity optimization variables are removed by the pre-solve routine in state of the art solvers such as `Gurobi` [18], based on the equality constraints in (3). Therefore, these variables do not affect the computational complexity of solving the MILP in practice.

IV. Closed-loop Simulation Results

In this section, we present closed-loop simulation results to illustrate the performance and real-time feasibility of the proposed approach for centralized vehicle coordination, based on a Matlab implementation that uses `Gurobi` [18] to solve the MILP at each sampling time $T_s = 1$ s.

A. Map and Scenario Generation

The same map information as shown in Figure 1 is used for the closed-loop simulations, consisting of 13 road sections (4 conflict zones and 9 conflict-free segments). Vehicles can move into and out of the considered area through 4 of the conflict-free *exit segments* in the north, east, south and west side of the map as illustrated in Figure 1. Each segment contains one or multiple lanes in which the vehicles can move and each lane connects two adjacent segments, except for the exit segments. Each CAV α^i, $i \in \mathcal{I}_c$ is provided with a route plan that is defined as an ordered sequence ϕ^i of segments. To validate the performance of the proposed vehicle coordination module, we use closed-loop simulations with randomly generated scenarios, including different starting positions and routes for each vehicle.

B. Closed-loop Simulations

The results of a closed-loop simulation for a randomly generated scenario are illustrated in Figure 2 based on four time frames of interest. Similar to Figure 1, CAVs are depicted in red and NCVs in blue, while road segments in grey are conflict-free and the ones in red denote conflict zones. The number above each vehicle shows its current velocity as a normalized value. The first snapshot in Figure 2 shows the vehicle positions after 2.4 seconds in the closed-loop simulation. One of the CAVs is moving behind a NCV and the CAV has slowed down to a normalized speed value of 0.3, in order to reduce the waiting time for the CAV as the vehicle is predicted to stop at the conflict zone until the NCV moves out of the zone. Then, the CAV reaches the conflict zone and waits at the edge of the conflict zone until the conflict zone has been cleared by the NCV. In the third snapshot of Figure 2, it can be observed that the CAV enters the conflict zone and moves at a normalized speed of 0.5 as there are no obstacles. In the fourth snapshot of Figure 2, we can see an example of more complicated queuing behavior of both CAVs and NCVs at an intersection.

In addition, Figure 3 presents time trajectories of a closed-loop simulation for a scenario of 10 NCVs and 4 CAVs with a relatively shortened route plan for illustration purposes. It shows the percentage of task completion, i.e., of reaching the target destination, and whether the CAV is waiting or not at each time step in the closed-loop simulation.

C. Vehicle Coordination Performance

Next, the performance of the proposed vehicle coordination module is illustrated in Table I for closed-loop simulations of 120 s using 100 randomly generated scenarios with 8 NCVs and 3 CAVs. In addition, the simulation for each scenario was executed multiple times with different values for the weights ω_1 and ω_2 in the terms (11) and (12) of the MILP objective (10), respectively. More specifically, Table I shows the average and maximum values of the total waiting time and travel time over all of the CAVs and over all of the randomly generated scenarios. Table I additionally shows values for the normalized travel time and for the average inverse velocity, i.e., the total travel time divided by the total traveled distanced and the total distance divided by the total moving time for each CAV, respectively. Note that the total travel time is equal to the sum of the total waiting time and the total moving time. The weight values (ω_1, ω_2) are chosen to be equal to either $(100, 1)$, $(1, 1)$ or $(1, 100)$ to illustrate the effect of both objective terms in (10). The remaining weight values are chosen to be equal to $\omega_3 = 1000$, $\omega_4 = 1000$ and $\omega_5 = 0$.

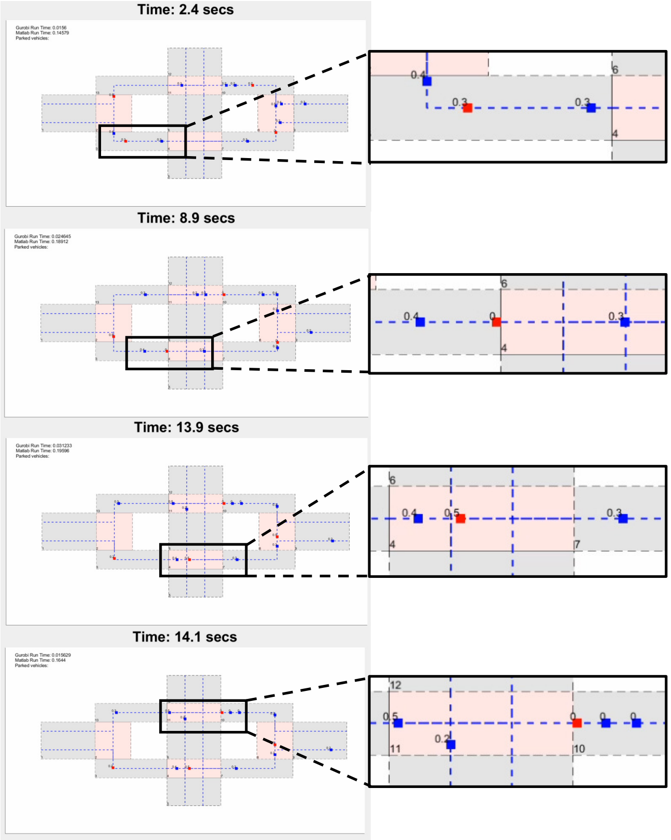

Fig. 2. Four time frames of interest in closed-loop simulation results of the centralized MILP-based vehicle coordination module for a particular scenario of 10 NCVs and 4 CAVs on the map in Figure 1.

It can be observed from Table I that the total wait time is reduced when increasing the weight value ω_2 in (12). With larger values for ω_2, the vehicle coordination module may instruct a CAV to slow down when it approaches a conflict zone, in order to reduce its overall waiting time when the conflict zone is predicted to be occupied. However, it can also be observed that the travel time may slightly increase due to the latter behavior, e.g., a CAV may miss its chance to pass through the conflict zone at the time when a NCV is detected to approach the same conflict zone. Finally, as expected, the average inverse velocity values in Table I can

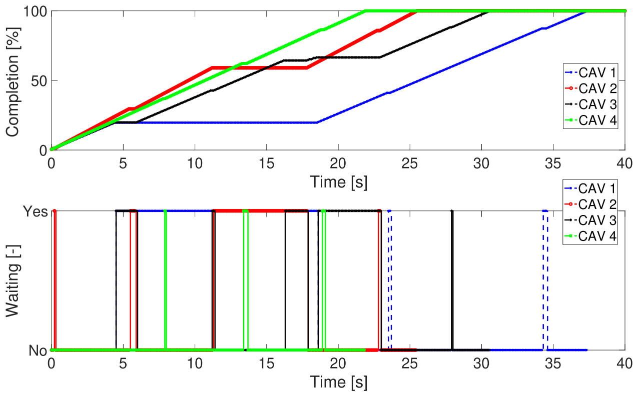

Fig. 3. Time trajectories for a closed-loop simulation of 10 NCVs and 4 CAVs with a shortened route plan for illustration purposes: percentage of task completion and waiting times for each CAV.

be seen to decrease when increasing the weight value ω_1 in the first objective term (11).

TABLE I

CLOSED-LOOP SIMULATION RESULTS WITH 8 NCVS AND 3 CAVS: WAITING AND TRAVEL TIME FOR DIFFERENT VALUES OF ω_1 AND ω_2 IN (10). TUNING (ω_1, ω_2) – $1 : (100, 1)$; $2 : (1, 1)$; $3 : (1, 100)$

Performance metric	Tuning 1		Tuning 2		Tuning 3	
	mean	max	mean	max	mean	max
WAITING TIME (S)	10.58	42.00	6.52	36.30	0.75	10.50
TRAVEL TIME (S)	54.11	99.90	54.93	103.40	54.92	103.30
NORMALIZED TRAVEL TIME	2.52	5.39	2.56	5.39	2.56	5.39
INVERSE VELOCITY	2.00	2.17	2.21	3.23	3.04	27.09

D. Real-time Feasibility

Finally, we investigate the computation times for the MILP solver on a modern computer, based on running 100 randomly generated closed-loop simulations of 120 s and with different numbers of CAVs $(2-4)$ and NCVs $(5-20)$. Specifically, we are interested in the worst-case computation times that are needed for `Gurobi` to solve the MILP at each sampling time that the vehicle coordination module is called. The overall worst-case computation time is below 0.08 s and therefore well below the sampling time of $T_\text{s} = 1$ s that is used in the closed-loop simulations. This indicates that the proposed implementation can be real-time feasible when the vehicle coordination module is executed in the infrastructure, such as in mobile edge computers (MECs).

V. CONCLUSIONS

This paper proposed an MILP approach for centralized coordination of CAVs in a highly dynamic environment that consists of multiple human-driven, i.e., manual vehicles in an interconnected network of conflict zones, such as merging points and intersections. The high-level decision making method provides direct support for transportation routing, while ensuring safety and optimizing time and energy efficiency for each of the CAVs. The performance and real-time feasibility of the MILP-based implementation is validated based on closed-loop simulation results on an illustrative scenario. High-fidelity simulations and real-world experiments, based on the multi-layer guidance and control architecture, are part of ongoing work.

REFERENCES

[1] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. Intell. Transport. Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[2] J. Guanetti, Y. Kim, and F. Borrelli, "Control of connected and automated vehicles: State of the art and future challenges," *Annual Reviews in Control*, vol. 45, pp. 18 – 40, 2018.

[3] F. Ye, J. Guo, K. J. Kim, P. V. Orlik, H. Ahn, S. Di Cairano, and M. J. Barth, "Bi-level optimal edge computing model for on-ramp merging in connected vehicle environment," in *2019 IEEE Intell. Veh. Symp.*, 2019, pp. 2005–2011.

[4] G. Lee, J. Guo, K. J. Kim, P. Orlik, H. Ahn, S. Di Cairano, and W. Saad, "Edge computing for interconnected intersections in internet of vehicles," in *IEEE Intell. Veh. Symp.*, 2020, pp. 480–486.

[5] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, pp. 33–55, 2016.

[6] K. Berntorp, T. Hoang, R. Quirynen, and S. Di Cairano, "Control architecture design for autonomous vehicles," in *IEEE Conf. Control Technology and Applications*, 2018, pp. 404–411.

[7] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," *Algorithm engineering*, pp. 19–80, 2016.

[8] H. Ahn, K. Berntorp, P. Inani, A. J. Ram, and S. Di Cairano, "Reachability-based decision-making for autonomous driving: Theory and experiments," *IEEE Trans. Control Syst. Technol.*, pp. 1–15, 2020.

[9] Y. E. Sahin, R. Quirynen, and S. Di Cairano, "Autonomous vehicle decision-making and monitoring based on signal temporal logic and mixed-integer programming," in *Am. Contr. Conf.*, 2020, pp. 454–459.

[10] K. Berntorp, T. Hoang, and S. Di Cairano, "Motion planning of autonomous road vehicles by particle filtering," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 197–210, 2019.

[11] T. Uno, R. Quirynen, S. Di Cairano, and K. Berntorp, "Vehicle integrated control by nonlinear model predictive control in GNSS-based autonomous driving system," *Transactions of Society of Automotive Engineers of Japan*, vol. 50, no. 4, pp. 1176–1181, 2019.

[12] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, 2016.

[13] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2510–2525, 2019.

[14] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Trans. Intell. Veh.*, vol. 3, no. 3, pp. 287–299, 2018.

[15] F. Ashtiani, S. A. Fayazi, and A. Vahidi, "Multi-intersection traffic management for autonomous vehicles via distributed mixed integer linear programming," in *Am. Contr. Conf.*, 2018, pp. 6341–6346.

[16] A. Festag, A. Hessler, R. Baldessari, L. Le, W. Zhang, and D. Westhoff, "Vehicle-to-vehicle and road-side sensor communication for enhanced road safety," in *World Congr. Intell. Transport Systems*, 2008.

[17] P. Hespanhol, R. Quirynen, and S. Di Cairano, "A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control," in *Europ. Contr. Conf.*, 2019, pp. 2763–2768.

[18] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021.

[1] Computation times were obtained on a modern computer that is equipped with a 2.3 GHz 8-Core Intel Core i7 processor.