Distributed Minimum-Time Optimal Control for Lane-Free CAVs Intersection Crossing

Ebubekir Pinar[1], Nader Meskin[2] and Arash M. Dizqah[1]

Abstract—**Traffic intersections are major bottlenecks in urban environments, and studies have shown that connected and autonomous vehicles (CAVs) can improve their overall throughput. While a centralised control of CAVs is effective for this application, it suffers from communication overhead, computational bottlenecks, and single-point failure vulnerabilities. This paper proposes a fully distributed minimum-time optimisation framework where individual CAVs solve local optimal control problems using only local information with neighbours. The framework incorporates robust collision avoidance constraints with minimum-time objectives and fail-safe protocols, ensuring collision-free operations. Simulation validation in a ROS-based environment for six CAVs at a four-leg intersection demonstrates that the proposed distributed algorithm achieves trajectory smoothness closely matching centralised solutions with normalised root mean square error (NRMSE) values between 0.1% and 2.7%. Although crossing times are 18.8–69.5% longer than the centralised solution—which operates near the theoretical minimum—the distributed approach eliminates requirements for centralised infrastructure, coordination, and global information exchange while reducing computational complexity, confirming the feasibility and practical efficiency of the proposed framework for lane-free intersection management.**

I. Introduction

Traffic intersections are major bottlenecks where fixed-time traffic signals operate inefficiently due to inability to adapt dynamically. Connected and Autonomous Vehicles (CAVs) enable trajectory optimisation in lane-free environments, achieving 40–54% crossing time reductions through centralised solution. However, centralised solutions suffer from communication overhead, computational bottlenecks, and single-point failure vulnerabilities that limit practical deployment [1]. This paper addresses this gap by proposing a fully distributed optimal control framework where individual vehicles coordinate using only local neighbour information, enabling scalable distributed lane-free intersection management.

Based on previous research, several approaches have been proposed, each with distinct advantages and limitations. Fully distributed model predictive control (MPC) formulations have emerged as promising solutions, where agents solve nonconvex optimal control problem (OCP) locally and synchronously, exchanging optimised trajectories to avoid conflict set points via vehicle-to-vehicle communication [2]. Alternative approaches leverage semi-definite relaxation-based Alternating Direction Method of Multipliers (ADMM) to enable parallel computation frameworks for cooperative planning and control, addressing the nonlinearity and nonconvexity inherent in autonomous driving problems [3]. Recent advances demonstrate that improved consensus ADMM with locally connected topology networks can achieve linear time complexity $\mathcal{O}(N)$ by exploiting sparsity in dual update processes, enabling scalable cooperative motion planning for large-scale CAV systems involving up to 80 vehicles [4]. Furthermore, Event-triggered control strategies further reduce communication and computational overhead by determining control updates based on measurement error rather than fixed time intervals [5] [6].

In decentralised strategies, each CAV solves local OCP to achieve both local and global objectives. While decentralised frameworks with full information sharing are computationally efficient [7], [8], real-world connectivity constraints limit communication to subsets of vehicles. Recent methods address partial connectivity through distributed observation and state estimation, demonstrating robust cooperation with moderate performance trade-offs [9].

Despite these advances, several critical limitations remain. Many existing approaches focus on collision avoidance through feasibility checking rather than computing minimum-time optimal solutions, potentially leaving performance gains unexploited [10]. Global optimisation requires managing numerous constraints and high computational complexity, limiting real-time applicability. Formulating local trajectory optimisation problems for individual vehicles simplifies the optimisation landscape while maintaining adaptability [11]. Some methods focus on tracking predefined reference trajectories rather than computing optimal trajectories online [12].

In this work, we propose a distributed trajectory optimisation framework for CAV intersection management that achieves computational efficiency with near-optimal performance. The framework decomposes the global problem into coupled local subproblems solved by individual vehicles while maintaining collision-free guarantees. Through vehicle-to-vehicle (V2V) communication and predictive trajectory sharing, each CAV optimises its trajectory based on the anticipated behaviour of neighbours, enabling distributed decision-making without centralised solution overhead. Fallback protocols and conservative safety margins ensure safe operation under communication and computation failures.

Our key contributions include a distributed minimum-time optimisation framework enabling each CAV to solve local OCP using only neighbouring vehicle information, an integrated collision avoidance constraints embedding minimum-time objectives directly within the optimisation formulation,

This work was supported in part by the University of Sussex.

[1]Ebubekir Pinar and [1]Arash M. Dizqah are with Smart Vehicles Control Laboratory (SVeCLab), University of Sussex, Brighton BN1 9RH, UK e.pinar@sussex.ac.uk, a.m.dizqah@sussex.ac.uk

[2]Nader Meskin is with the Department of Electrical Engineering, College of Engineering, Qatar University, 2371, Doha, Qatar nader.meskin@qu.edu.qa

robust fail-safe protocols ensuring collision-free operations under communication delays and computation failures, and comprehensive simulation validation demonstrating near-optimal performance with significantly reduced computational complexity compared to centralised methods.

II. System Description

A. Vehicle Kinematics

This study represents the lateral dynamics of CAVs using the advanced (nonlinear) bicycle model [13]. The bicycle model consists the vehicle's essential planar motion through two primary degrees of freedom (DoF): sideslip angle β_i and yaw rate r_i. An additional DoF accounts for the longitudinal velocity V_i. Together with three kinematic states that describe the CAV's ground-fixed reference frame, these variables form a set of non-linear differential equations representing the motion of CAV$_i$ as follows:

$$\begin{bmatrix} r_i(k+1) \\ \beta_i(k+1) \\ V_i(k+1) \\ x_i(k+1) \\ y_i(k+1) \\ \theta_i(k+1) \end{bmatrix} = \begin{bmatrix} r_i(k) \\ \beta_i(k) \\ V_i(k) \\ x_i(k) \\ y_i(k) \\ \theta_i(k) \end{bmatrix} + T_{s,i} \begin{bmatrix} \frac{N_r(k)\cdot r_i(k)}{I_z} + \frac{N_\beta \cdot \beta_i(k)}{I_z} \\ (\frac{Y_r(k)\cdot r_i(k)}{m \cdot V_i(k)} - r_i(k)) + \frac{Y_\beta \cdot \beta_i(k)}{m\cdot V_i(k)} \\ 0 \\ V_i(k)\cdot \cos\theta_i(k) \\ V_i(k)\cdot\sin\theta_i(k) \\ r_i(k) \end{bmatrix}$$

$$+ T_{s,i} \begin{bmatrix} 0 & \frac{N_\delta}{I_z} \\ 0 & \frac{Y_\delta}{m\cdot V_i(k)} \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_i(k) \\ \delta_i(k) \end{bmatrix} \tag{1}$$

where $\mathbf{x}_i(k) = [r_i(k), \beta_i(k), V_i(k), x_i(k), y_i(k), \theta_i(k)]^T$ and $\mathbf{u}_i(k) = [a_i(k), \delta_i(k)]^T$ are, respectively, the system states and control inputs of CAV$_i$ at time step k. The pose vector $\mathbf{z}_i(k) = [x_i(k), y_i(k), \theta_i(k)]^T$ represents the global position and orientation of CAV$_i$ in non-inertial reference system. The control inputs $a_i(k)$ and $\delta_i(k)$ denote the vehicle's longitudinal acceleration (m/s^2) and steering angle (rad), respectively. Vehicle parameters include the mass m (kg) and moment of inertia I_z (kg.m^2). The sampling time T_s (s) represents the discrete time step interval. The calculation of vehicle parameters N_r, N_β, N_δ, Y_r, Y_β and Y_δ is shown as follows [13]:

$$\begin{aligned} N_r(k) &= \frac{1}{V_i(k)+\varepsilon}\cdot(l_f^2\cdot C_F + l_r^2 \cdot C_R) \\ N_\beta &= l_f\cdot C_F - l_r\cdot C_R) \\ N_\delta &= -l_f \cdot C_F \\ Y_r(k) &= \frac{1}{V_i(k)+\varepsilon}\cdot(l_f\cdot C_F - l_r\cdot C_R) \\ Y_\beta &= C_F + C_R \\ Y_\delta &= -C_F \end{aligned}$$

where C_F and C_R are the cornering stiffness coefficients of the front and rear tyres, respectively, and l_f and l_r represent the distances from the vehicle's center of gravity to the front and rear axles. A regularisation term $\varepsilon > 0$ is added to ensure numerical stability.

To ensure that each CAV$_i$ drives within its dynamic limitations, the following physical constraints are enforced for each CAV$_i$:

$$\underline{V} \le V_i(k) \le \overline{V} \tag{2a}$$
$$\underline{a} \le |a_i(k)| \le \overline{a} \tag{2b}$$
$$\underline{\delta} \le |\delta_i(k)| \le \overline{\delta} \tag{2c}$$
$$\underline{r} \le |r_i(k)| \le \overline{r} \tag{2d}$$
$$\underline{\beta} \le |\beta_i(k)| \le \overline{\beta} \tag{2e}$$

where $\bar{\cdot}$ and $\underline{\cdot}$ denote the upper and lower boundaries, respectively.

B. Modelling of CAV and intersection

The intersection environment and road geometry are modelled using polytopic representations following the formulation in [1].

Each CAV$_i$, $i \in \{1,\ldots,N\}$ with N as the total number of CAVs, is modelled as a rectangular polytope $\tilde{P}_i$ centred at its geometric centre, expressed as $\tilde{A}_i X \le \tilde{b}_i$ in the vehicle's local coordinate frame, where $X = [x,y]^T$ is a Cartesian point. $\tilde{A}_i$ defines the orientations of the vehicle's sides, and $\tilde{b}_i$ contains the corresponding offset distances from the vehicle's geometric centre to each side.

When transformed to the global coordinate frame at time step k, the time-varying polytope becomes $P_i(k) = \{X \in \mathbb{R}^2 | A_i(\mathbf{z}_i(k))X \le b_i(\mathbf{z}_i(k))\}$, where $A_i(\mathbf{z}_i(k))$ and $b_i(\mathbf{z}_i(k))$ are functions of the vehicle's pose $\mathbf{z}_i(k)$ obtained through rotation and translation of the body-fixed polytope [10]. This representation enables compact convex modelling for collision avoidance constraints. Road boundaries are described by convex polytopic sets O_r, represented as $\tilde{A}_r X \le \tilde{b}_r$, where $r \in \{1,\ldots,N_r\}$ and N_r for a typical four-leg intersection, defining the feasible spatial region bounded by lane edges and intersection limits.

C. Constraints to Avoid Collisions

To prevent collision between any two CAVs at each discrete time step, their respective polytopic sets must not intersect, i.e., $P_i(k) \cap P_j(k) = \emptyset$, where $P_i(k) = \{\mathrm{X} \in \mathbb{R}^2 | A_i(\mathbf{z}_i(k))\mathrm{X} \le b_i(\mathbf{z}_i(k))\}$ and $P_j(k) = \{\mathrm{Y} \in \mathbb{R}^2 | A_j(\mathbf{z}_j(k))\mathrm{Y} \le b_j(\mathbf{z}_j(k))\}$. Directly enforcing this condition leads to a non-convex and non-differentiable constraint.

To maintain convexity and differentiability, the intersection condition is replaced with a distance constraint $\mathrm{dist}(P_i(k), P_j(k)) \ge d_{\min}$, where $d_{\min}$ is the minimum safe distance between CAVs and $\forall j \ne i \in \{1,\ldots,N\}$ represents the neighbouring vehicles within the communication zone of CAV$_i$. Since the computation of the distance between convex polytopes is itself a convex problem [14], the above condition can be expressed equivalently in its dual form [15]:

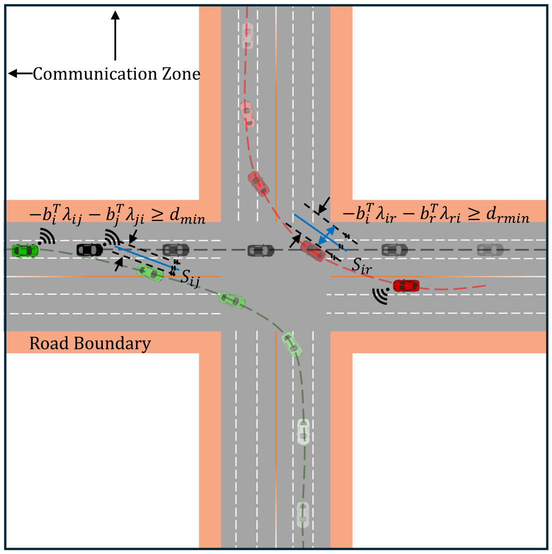

Fig. 1: Schematic representation of the lane-free intersection showing geometric configuration and sufficient conditions for collision avoidance.

$$
\begin{aligned}
&\operatorname{dist}(P_i(k), P_j(k)) = \\
&\max_{\lambda_{ij}(k),\, \lambda_{ji}(k),\, s_{ij}(k)} \big(-b_i(\mathbf{z}_i(k))^\top \lambda_{ij}(k) \\
&\qquad -b_j(\mathbf{z}_j(k))^\top \lambda_{ji}(k)\big) \qquad (3)\\
&\text{s.t.}\quad A_i(\mathbf{z}_i(k))^\top \lambda_{ij}(k) + s_{ij}(k) = 0 \\
&\qquad A_j(\mathbf{z}_j(k))^\top \lambda_{ji}(k) - s_{ij}(k) = 0 \\
&\qquad \|s_{ij}(k)\|_2 \leq 1 \; \lambda_{ij}(k) \geq 0 \; \lambda_{ji}(k) \geq 0 \\
&\qquad \forall\, i \neq j \in \{1, \dots, N\},\; k \in \{0, \dots, N_p - 1\}
\end{aligned}
$$

The dual variables $\lambda_{ij}(k)$, $\lambda_{ji}(k)$, and $s_{ij}(k)$ at each time step define a separating hyperplane between the two CAVs. Feasible dual variables satisfying $-b_i(\mathbf{z}_i(k))^\top \lambda_{ij}(k) - b_j(\mathbf{z}_j(k))^\top \lambda_{ji}(k) \geq d_{\text{min}}$ guarantee $\operatorname{dist}(P_i(k), P_j(k)) \geq d_{\text{min}}$, ensuring collision-free trajectory [15].

To ensure that each CAV$_i$ remains within the intersection limits at all time steps, $P_i(k)$ must not intersect road boundary polytopes O_r, where $r \in \{1, \dots, N_r\}$, expressed as $(P_i(k) \cap O_r) = \emptyset, \quad \forall i \in \{1, \dots, N\}, r \in \{1, \dots, N_r\}$. This non-convex condition is replaced by a convex minimum distance constraint $\operatorname{dist}(P_i(k), O_r) \geq d_{\text{rmin}}$ at each time step, as depicted in Fig. 1, where d_{rmin} is the minimum safe distance between CAVs and road boundaries. Feasible dual variables at each discrete time step guarantees that CAV$_i$ stays within the safe region, ensuring collision-free and dynamically feasible trajectories [1].

III. Distributed Intersection Control

A. Optimal Control Problem Formulation

The distributed lane-free intersection crossing is formulated as a discrete-time local OCP for each CAV$_i$. Following Section II-C, non-differentiable and non-convex collision avoidance constraints are replaced with smooth dual formulations, ensuring convexity, differentiability, and compatibility with gradient-based optimisation algorithms. The discrete-time OCP is formulated as follows:

$$
\begin{aligned}
&\{T_{s,i}(\cdot), \mathbf{z}_i(\cdot), \mathbf{u}_i(\cdot)\}^* = \\
&\arg \min_{\substack{T_{s,i}, \mathbf{z}_i, \mathbf{u}_i, \\ \lambda_{ij}, s_{ij}, \\ \lambda_{ri}, \lambda_{ir}, s_{ir}}} J(\mathbf{z}_i(\cdot)) := \sum_{k=0}^{N_p-1} \Big[\alpha T_{s,i}(k)^2 + \\
&(\mathbf{z}_i(k) - \mathbf{z}_{f,i})^\top \mathbf{Q} (\mathbf{z}_i(k) - \mathbf{z}_{f,i})) + \gamma a_i(k)^2 + \eta (V_i(k) - \overline{V})^2\Big] \qquad (4a)
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.}\quad & (1), (2), && (4b)\\
& -b_i(\mathbf{z}_i(k))^\top \lambda_{ij}(k) - b_j(\mathbf{z}_j^{(j\to i)}(k))^\top \lambda_{ji}(k) \geq d_{\text{min}} && (4c)\\
& A_i(\mathbf{z}_i(k))^\top \lambda_{ij}(k) + s_{ij}(k) = 0 && (4d)\\
& A_j(\mathbf{z}_j^{(j\to i)}(k))^\top \lambda_{ji}(k) - s_{ij}(k) = 0 && (4e)\\
& -b_i(\mathbf{z}_i(k))^\top \lambda_{ir}(k) - b_r^\top \lambda_{ri}(k) \geq d_{\text{rmin}} && (4f)\\
& A_i(\mathbf{z}_i(k))^\top \lambda_{ir}(k) + s_{ir}(k) = 0 && (4g)\\
& A_r^\top \lambda_{ri}(k) - s_{ir}(k) = 0 && (4h)\\
& \lambda_{ij}(k),\ \lambda_{ir}(k),\ \lambda_{ri}(k) \geq 0 && (4i)\\
& \|s_{ij}(k)\|_2 \leq 1,\ \|s_{ir}(k)\|_2 \leq 1 && (4j)\\
& \mathbf{z}_i(0) = z_{0,i},\ V_i(0) = V_{0,i} && (4k)\\
& \forall i \neq j \in \{1, \dots, N\},\ \forall r \in \{1, \dots, N_r\} \\
& \forall k \in \{0, \dots, N_p - 1\}
\end{aligned}
$$

where $\mathbf{z}_j^{(j\to i)}(k)$ represents CAV$_j$'s predicted trajectory, computed in the previous optimisation step, is communicated to CAV$_i$. Each CAV$_i$ solves problem (4) at each sampling time k over the prediction horizon N_p, yielding the optimal trajectory, sampling time and control $\{\mathbf{z}_i^*(k+l \mid k),\, T_{s,i}^*(k),\, \mathbf{u}_i^*(k+l \mid k)\}$, $\forall l \in \{0, \dots, N_p - 1\}$. It is worse noting that (4) is a minimum-time OCP and hence the sampling time $T_{s,i}(k)$ is treated as a decision variable rather than a fixed parameter.

The objective function (4a) minimises a weighted combination of the sampling time $T_{s,i}^*(k)^2$ with weight $\alpha > 0$, tracking error to the final pose $\mathbf{z}_{f,i}$ weighted by positive definite matrix $\mathbf{Q}$, and control effort $a_i^2(k)$ with weight $\gamma > 0$, and the speed tracking term $\eta(V_i(k) - \overline{V})^2$ with weight $\eta > 0$, which explicitly penalises deviation from maximum speed $\overline{V}$ throughout the horizon to help optimiser to solve minimum time problem. This structure encourages efficient trajectory planning and smooth, feasible control actions. Constraints (4c)-(4e) enforce minimum separation distance $d_{\text{min}} > 0$ between vehicles at each time step through dual formulation, converting non-convex separation $P_i(k) \cap P_j(k) = \emptyset$ into smooth constraints tractable for gradient-based solvers. Road boundary constraints (4f)-(4h) maintain minimum distance $d_{\text{rmin}} > 0$ from time-invariant boundaries $N_r = 4$ for a four-leg intersection; these constraints are equivalent to barrier conditions, ensuring each CAV remains within the safe drivable region at all times. It is assumed that all N CAVs share a common communication

Algorithm 1 Pseudo-code of the proposed Distributed Lane-Free Intersection Optimal Control Algorithm for each CAV$_i$

1: **Initialisation:**
2: Initialise $\mathbf{z}_{i,0}, V_{0,i}, \mathbf{Z}_j^{(j\to i)}(k),\ \forall i \neq j \in \{1,\ldots,N\}$
3: **for** $k = 0, 1, 2, \ldots$ until all CAVs reach destinations **do**
4: $\quad T^*_{s,i}(k),\ \mathbf{U}_i(k),\ \mathbf{Z}_i(k),\ \forall l \in \{0,\ldots,N_p-1\} \leftarrow$ Solve Problem (4)
5: $\quad$ Communicate $\mathbf{Z}_i(k),\ T^*_{s,i}(k)$ to CAV$_{j\neq i}$
6: $\quad$ Receive $\mathbf{Z}_j^{(j\to i)}(k),\ T_{s,j}^{(j\to i)}(k)$ from CAV$_{j\neq i}$
7: $\quad$ **if** communication failed with CAV$_j$ **then**
8: $\qquad$ Use $\mathbf{Z}_j^{(j\to i)}(k-1)$ from previously received trajectory of CAV$_j$
9: $\quad$ **end if**
10: $\quad$ Compute average sampling time $\overline{T}_s(k) = \frac{1}{N}\left(T^*_{s,i}(k) + \sum_{j\neq i} T_{s,j}^{(j\to i)}(k)\right)$
11: $\quad$ **for** $l = 0$ **to** N_c **do**
12: $\qquad$ Apply $\mathbf{u}_i^*(k+l \mid k)$ for duration $\overline{T}_s(k)$
13: $\quad$ **end for**
14: $\quad k \leftarrow k+1$
15: **end for**

zone, so every CAV$_i$ exchanges information with all remaining CAVs. Initial conditions $(\mathbf{z}_{0,i}, V_{0,i})$ are predetermined inputs that define the starting state of each CAV$_i$. The final pose $\mathbf{z}_{f,i}$ is the desired exit point of CAV$_i$, provided as a fixed reference input to Problem (4) by a separate algorithm.

B. Distributed Control Architecture

Extending the centralised framework in [1], this work implements a fully distributed architecture where each CAV$_i$ solves Problem (4) using only local variables and communicated neighbour information.

The distributed control strategy follows a receding horizon approach (Algorithm 1). To simplify notation, let $\mathbf{Z}_i(k) \triangleq \{\mathbf{z}_i^*(k+l \mid k)\}$, $\mathbf{U}_i(k) \triangleq \{\mathbf{u}_i^*(k+l \mid k)\}$, and $\mathbf{Z}_j^{(j\to i)}(k) \triangleq \{\mathbf{z}_j^{(j\to i)}(k+l \mid k)\}$, $\forall l \in \{0,\ldots,N_p-1\}$, denote the optimal predicted trajectory and the control trajectory of CAV$_i$, and the received predicted trajectory of neighbouring CAV$_j$, respectively. At each sampling time k, each CAV$_i$ independently executes the following steps: (i) solves problem (4) to obtain the optimal predicted trajectory $\mathbf{Z}_i(k)$ and the corresponding sampling time $T_{s,i}(k)$; (ii) communicates $\mathbf{Z}_i(k)$ and $T_{s,i}(k)$ to all neighbours; (iii) receives $\mathbf{Z}_j^{(j\to i)}(k)$ and $T_{s,j}^{(j\to i)}(k)$ from each neighbour, or falls back to the shifted trajectory $\mathbf{Z}_j^{(j\to i)}(k-1) \triangleq \{\mathbf{z}_j^{(j\to i)}(k+l \mid k-1)\} \quad \forall l \in \{1,\ldots,N_p-1\}$ if communication fails; and (iv) applies the first N_c control inputs $\mathbf{u}_i^*(\cdot)$, where $N_c \ll N_p$, before re-solving Problem (4) with the updated neighbouring trajectories.

The architecture operates fully distributed without a central coordinator. Each CAV$_i$ initialise the initial conditions $(\mathbf{z}_{i,0}, V_{0,i})$ and calculate the simplify trajectory $\mathbf{Z}_j^{(j\to i)}(k)$ for CAV$_j$ (line 2) before going to solving problem (4). Before advancing, each CAV$_i$ waits to receive information from all neighbours before proceeding, thereby establishing implicit synchronisation among vehicles. For robustness, if CAV$_i$ fails to receive information from CAV$_j$ within timeout, it uses a shifted trajectory previously communicated (line 8), assuming the neighbour continues along its last communicated trajectory. This eliminates single points of failure and enables true scalability.

TABLE I
MAIN PARAMETERS OF THE MODEL

Parameter	Value	Parameter	Value
m (kg)	2.2	$\overline{V}$ (m/s)	1.0
d_{min} (m)	0.1	$\underline{V}$ (m/s)	0.0
d_{rmin} (m)	0.1	$\overline{a} = -\underline{a}$ (m/s^2)	0.4
N_p (-)	15	$\overline{\delta} = -\underline{\delta}$ (rad)	0.5
d (-)	5	$\overline{r} = -\underline{r}$ (rad/s)	1.0
$V_{0,i}$ (m/s)	0.01	$\overline{\beta} = -\underline{\beta}$ (rad)	0.4

Following the information exchange phase, each CAV$_i$ computes the average sampling time $\overline{T}_s(k)$ by averaging its own optimal sampling time $T^*_{s,i}(k)$ with those received from all neighbours $T_{s,j}^{(j\to i)}(k)$, $\forall j \neq i \in \{1,\ldots,N\}$ (line 10). Since each vehicle's local OCP independently minimises its own crossing time, the resulting optimal sampling times differ across vehicles. This arithmetic averaging, therefore, enforces a common effective time step across all CAVs without requiring a dedicated synchronisation protocol or event-triggered mechanism [5], [6]. The optimal control input $\mathbf{u}_i^*(k+l \mid k)$ is then applied for the duration $\overline{T}_s(k)$, $\forall l \in \{0,\ldots,N_c-1\}$, ensuring implicit synchronisation and coordinated operation even under asynchronous local timings. After executing the N_c control inputs, each CAV$_i$ updates its local state and resolves Problem (4) using the updated local and neighbours information. This iterative process repeats until all CAVs reach their destinations. To further enhance computational efficiency, any CAV$_i$ that reaches $\mathbf{z}_{f,i}$ ahead of others is excluded from the communication network, reducing the number of collision avoidance constraints in Problem (4) for the remaining active CAVs.

IV. RESULTS AND DISCUSSION

A. Real-time Simulation Environment

The proposed distributed minimum-time lane-free control algorithm was evaluated in a realistic simulation environment using RobiL (Robot-in-the-Loop), a comprehensive ROS-based platform developed at SVeCLab, University of Sussex. In this framework, the optimisation problem is implemented in Python using CasADi [16].

RobiL is specifically designed for testing and validating distributed and collaborative multi-agent algorithms in settings that closely approximate real-world scenarios, while maintaining full experimental control and cost-effectiveness. The platform comprises three essential components working in concert: a user server for experimental configuration and monitoring, a core server hosting the computational algorithms, and a visualiser server for real-time trajectory and system

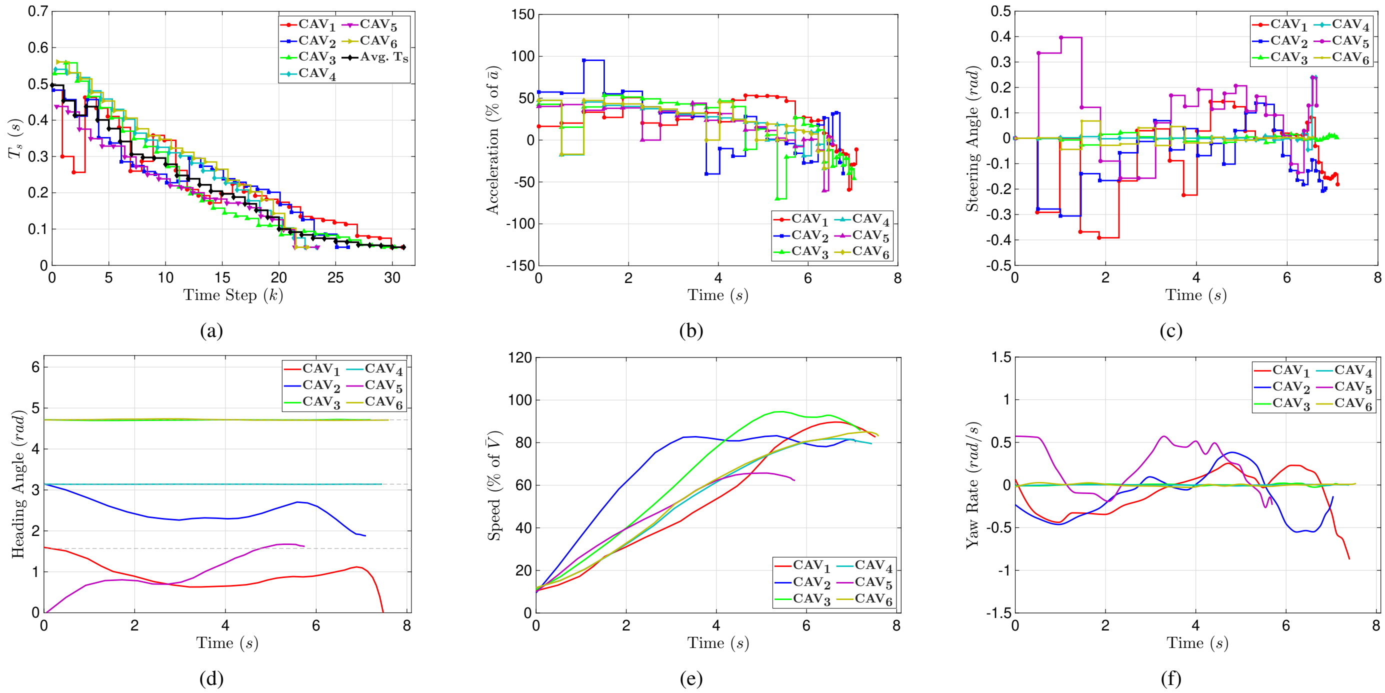

Fig. 2: Evolution of key control inputs and state variables during distributed intersection crossing for six CAVs. (a)-(c) Optimal control outputs from the distributed algorithm: (a) Variable sampling time $T_{s,i}(k)$; (b) Acceleration trajectories; (c) Steering angle trajectories. (d)-(f) Measured state variables from RobiL-Gazebo simulation: (d) Heading angle trajectories; (e) Longitudinal speed trajectories; (f) Yaw rate trajectories.

state visualisation. The core server operates ROS Noetic on Ubuntu 20.04, enabling efficient simulation and low-latency communication with CAVs.

The CAVs simulated in this study are based on the JetRacer Pro AI[1] platform, a small-scale autonomous robot vehicle designed for research applications. The simulation incorporates the platform's physical parameters and sensor suite, ensuring high-fidelity representation of vehicle dynamics and sensor characteristics.

B. Results

The proposed distributed minimum-time lane-free optimal control algorithm is tested in a four-leg intersection scenario with six CAVs arriving simultaneously from different lanes. Table I lists the parameter values used in the proposed and benchmark algorithms; the vehicle parameters and dynamic bounds correspond to the small-scale robot platform, all vehicles are assumed to move forward only, and the prediction horizon N_p and number of collocation points d are tuned to achieve the best performance with minimum computational time. Fig. 2 presents the time evolution of key control inputs and state variables throughout the intersection crossing.

Fig. 2a shows the sampling time $T_{s,i}$ computed from each CAV$_i$ from its local optimisation problem, as well as the average sampling time $\overline{T}_s(k)$ throughout the simulation. Occasional peaks appear in $T_{s,i}$ for all vehicles, primarily when the shifted mechanism is triggered due to either infeasibility or the solver reaching its maximum iteration limit. Despite these occasional spikes, the overall trend of the average sampling time $\overline{T}_s(k)$ begins around 0.5 s and progressively decreases as the vehicles approach their destinations, demonstrating the computational efficiency of the distributed framework; as each CAV nears its destination, the complexity of the optimisation problem is reduced.

Fig. 2b and Fig. 2c present the optimal longitudinal acceleration and steering angle trajectories, respectively, plotted against the average sampling time $\overline{T}_s(k)$ throughout the simulation, as computed by the distributed control algorithm. The acceleration trajectories in Fig. 2b are expressed as a percentage of the maximum acceleration $\overline{a}$, yielding the corresponding vehicle speed trajectories shown in Fig. 2e, with CAV$_3$ achieving the highest sustained velocity of approximately $0.95\,\overline{V}$ and CAV$_5$ the lowest sustained velocity of approximately $0.65\,\overline{V}$, where $\overline{V}$ denotes the maximum speed considered in the simulation. The steering angle trajectories in Fig. 2c remain within the allowable range $\underline{\delta}$ to $\overline{\delta}$ for all vehicles, indicating that the control inputs are bounded while ensuring collision-free intersection crossing. Notably, CAV$_1$ and CAV$_2$ demonstrate greater steering angle variations than CAV$_3$, as they execute turning manoeuvres within the intersection.

Figures 2d–2f are plotted as functions of simulation time (s) and describe the evolution of key state variables throughout the intersection-crossing manoeuvre. Specifically, Fig. 2d depicts the temporal evolution of the heading angle, Fig. 2e provides the longitudinal speed trajectories for each CAV, and

[1]https://www.waveshare.com/jetracer-pro-ai-kit.htm

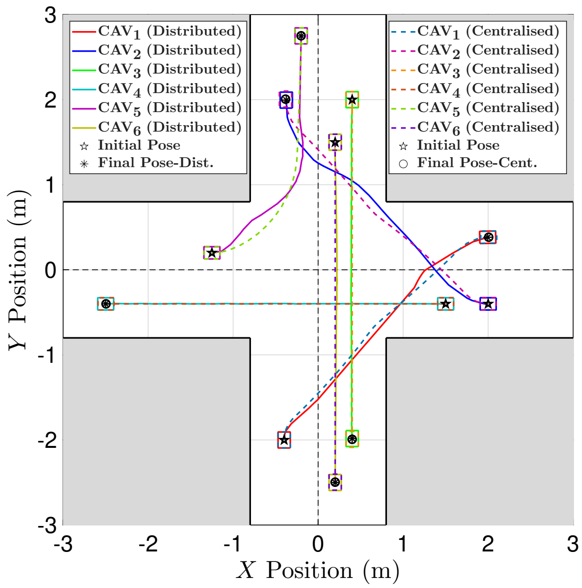

Fig. 3: Collision-free trajectories of six CAVs at a four-leg intersection: distributed (solid) vs. centralised (dashed) solutions. CAV_1, CAV_2 and CAV_5 turn; CAV_3, CAV_4 and CAV_6 go straight. An animated simulation is available at (https://youtu.be/JT0y9dnnumE).

Fig. 2f presents the corresponding yaw-rate responses. Table II shows the resulting performance of the proposed algorithm as compared to the centralised solution [1]. For each vehicle, the table reports the normalised root mean square error (NRMSE), the average speed, standard deviation of speed, standard deviation of acceleration, and crossing time difference. The NRMSE is computed from the positional RMSE between the distributed and centralised trajectories. The results indicate that the distributed approach achieves low NRMSE values, demonstrating close agreement with the centralised method. Average vehicle speeds range from approximately 54.2% to 79.0% of maximum speed. The standard deviation of speed varies between 21.4% and 38.5%, while the standard deviation of acceleration ranges from 22.3% to 35.5%. Notably, although the proposed distributed algorithm results in crossing times that are 18.8–69.5% longer than those of the centralised solution (where the crossing time difference is defined as $\Delta T_{\text{cross,i}} = \frac{T_{\text{cross,i}} - T_{\text{cross,cent}}}{T_{\text{cross,cent}}} \times 100\%$), it achieves this performance without requiring centralised infrastructure, coordination, or global information exchange.

The crossing time overhead directly reflects the fundamental limitations of distributed architecture, where each vehicle acts on local information only. Each CAV optimises its trajectory using only the previously communicated neighbour trajectories rather than solving a coupled global problem, introducing sequential sub-optimality at each receding horizon step. Additionally, the conservative safety margins and shifted-trajectory fallback protocol constrain the feasible solution space, and the centralised benchmark [1] operates near the theoretical optimum by jointly minimising all crossing times simultaneously — a target unreachable by any fully distributed method. Despite this trade-off, the distributed framework eliminates centralised infrastructure and single-point failure, reduces communication load to local neighbour exchange only, and lowers computational complexity by decomposing the global problem into local sub-problems — directly addressing the scalability and reliability limitations of centralised solutions [1].

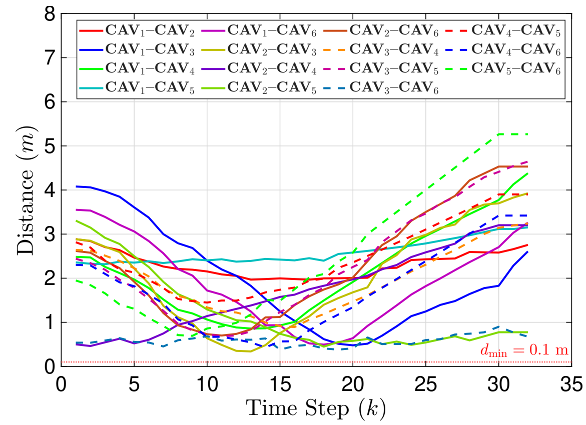

Fig. 4: Distributed solution inter-vehicle Euclidean distances over the simulation horizon for six CAVs at the intersection.

Fig. 3 demonstrates that both centralised and distributed approaches successfully generated collision-free trajectories, maintaining minimum separation distance $d_{\min}$ between CAVs throughout the manoeuvre. Table II presents a quantitative comparison of the distributed algorithm's performance against the centralised solution from [1]. The distributed algorithm achieves trajectory smoothness comparable to the centralised solution with NRMSE values between 0.1–2.7% across all CAVs, whilst eliminating centralised infrastructure requirements.

Fig. 4 presents the pairwise minimum inter-vehicle distances $\text{dist}(P_i(k), P_j(k))$ between all six CAVs throughout the simulation for the distributed approaches. Denoting by $\text{dist}(\text{CAV}_i, \text{CAV}_j)$ the minimum of $\text{dist}(P_i(k), P_j(k))$ over the simulation the smallest separation observed is $\text{dist}(\text{CAV}_2, \text{CAV}_3) = 0.34\,\text{m}$, followed closely by $\text{dist}(\text{CAV}_3, \text{CAV}_6) = 0.38\,\text{m}$ and $\text{dist}(\text{CAV}_1, \text{CAV}_3) = 0.48\,\text{m}$. All pairwise distances remain strictly above the enforced minimum safety distance $d_{\min} = 0.1\,\text{m}$ (Table I) throughout the manoeuvre, confirming that collision avoidance constraint (4d) is satisfied. The CAV_1–CAV_5 pair exhibits the largest minimum separation of 2.32 m, consistent with these vehicles arriving from different entry lanes and executing turning manoeuvres toward separate exit lanes. Conversely, CAV_2–CAV_3 and CAV_3–CAV_6 reach their closest proximity mid-manoeuvre as CAV_3 traverses the intersection straight through while the turning vehicles cross its trajectory, yet the minimum separation distance $d_{\min}$ is maintained throughout.

It is worth noting that the results above arise from a minimum-time optimal control formulation in which the intersection crossing manoeuvre is explicitly optimised with respect to time, rather than evaluated under a fixed-duration

TABLE II

PERFORMANCE COMPARISON WITH THE CENTRALISED SOLUTION IN [1] FOR SIX CAVS CROSSING THE INTERSECTION

$\mathbf{CAV}_i$ $\rightarrow$	$\mathbf{CAV}_1$	$\mathbf{CAV}_2$	$\mathbf{CAV}_3$	$\mathbf{CAV}_4$	$\mathbf{CAV}_5$	$\mathbf{CAV}_6$
NRMSE (%) w.r.t. centralised solution	1.1	2.7	0.2	0.1	1.7	0.3
Average speed (% of max speed)	65.8	74.2	79.0	64.5	54.2	65.6
Standard deviation of speed (%)	38.5	21.4	28.3	33.5	26.5	34.4
Standard deviation of acceleration (%)	28.2	35.5	35.0	22.3	24.0	22.8
Crossing time difference (% of centralised solution)	69.5	59.9	62.0	55.7	18.8	59.0

scenario. Furthermore, all experiments are conducted on a small-scale autonomous robot vehicle platform (JetRacer Pro AI), and as listed in Table I, the vehicle parameters, operating speeds, and enforced minimum distance d_{min} are all consistent with the physical scale and safety requirements of this experimental setup.

V. CONCLUSION

This paper presented a distributed minimum-time optimisation framework for lane-free intersection management that addresses scalability and robustness limitations inherent in centralised approaches. The proposed framework decomposes the global intersection OCP into coupled local optimal control subproblems, enabling each CAV to compute trajectories independently using only information from neighbours. Key contributions include a fully distributed formulation requiring no centralised infrastructure, integrated collision avoidance with minimum-time objectives, and validation demonstrating close trajectory agreement with reduced computational complexity.

Simulation results in a ROS-based environment for six CAVs demonstrated collision-free trajectories maintaining minimum separation distance throughout the manoeuvre. The distributed algorithm closely matched the centralised solution with NRMSE values between 0.1% and 2.7%. While crossing times are 18.8–69.5% longer than the centralised solution—which operates near the theoretical minimum—this trade-off eliminates requirements for centralised infrastructure, coordination, or global information exchange. This fully distributed architecture, leveraging V2V communication, enhances robustness against individual vehicle malfunctions and improves scalability, thereby confirming the feasibility and practical efficiency of the proposed framework for lane-free intersection management. Future work will extend the framework to continuous traffic scenarios with a larger number of CAVs to further validate scalability. Formal investigations of convergence guarantees for the receding horizon approach, the impact of communication imperfections such as packet loss and transmission delays, and the integration of safety constraints within a unified barrier framework are also planned. Finally, experimental validation on physical autonomous vehicle platforms at a real intersection will be pursued.

REFERENCES

[1] M. Amouzadi, M. O. Orisatoki, and A. M. Dizqah, "Optimal lane-free crossing of cavs through intersections," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1488–1500, 2023.

[2] A. Katriniok, B. Rosarius, and P. Mähönen, "Fully distributed model predictive control of connected automated vehicles in intersections: Theory and vehicle experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 288–18 300, 2022.

[3] X. Zhang, Z. Cheng, J. Ma, S. Huang, F. L. Lewis, and T. H. Lee, "Semi-definite relaxation-based admm for cooperative planning and control of connected autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9240–9251, 2022.

[4] H. Liu, Z. Huang, Z. Zhu, Y. Li, S. Shen, and J. Ma, "Improved consensus admm for cooperative motion planning of large-scale connected autonomous vehicles with limited communication," *IEEE Transactions on Intelligent Vehicles*, 2024.

[5] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on automatic control*, vol. 57, no. 5, pp. 1291–1297, 2011.

[6] D. Liuzza, D. V. Dimarogonas, M. Di Bernardo, and K. H. Johansson, "Distributed model based event-triggered control for synchronization of multi-agent systems," *Automatica*, vol. 73, pp. 1–7, 2016.

[7] M. Kloock, P. Scheffe, S. Marquardt, J. Maczijewski, B. Alrifaee, and S. Kowalewski, "Distributed model predictive intersection control of multiple vehicles," in *2019 IEEE intelligent transportation systems conference (ITSC)*. IEEE, 2019, pp. 1735–1740.

[8] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang, "A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections," *Automatica*, vol. 93, pp. 244–256, 2018.

[9] Y. Bian, S. E. Li, W. Ren, J. Wang, K. Li, and H. X. Liu, "Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10 744–10 754, 2019.

[10] R. Firoozi, L. Ferranti, X. Zhang, S. Nejadnik, and F. Borrelli, "A distributed multi-vehicle coordination algorithm for navigation in tight environments," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 10, pp. 14 499–14 509, 2024.

[11] J. Wu, L. Dai, and Y. Xia, "Iterative distributed model predictive control for heterogeneous systems with non-convex coupled constraints," *Automatica*, vol. 166, p. 111700, 2024.

[12] L. Ferranti, L. Lyons, R. R. Negenborn, T. Keviczky, and J. Alonso-Mora, "Distributed nonlinear trajectory optimization for multi-robot motion planning," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 2, pp. 809–824, 2023.

[13] W. F. Milliken, D. L. Milliken, and L. D. Metz, *Race car vehicle dynamics*. SAE international Warrendale, 1995, vol. 400.

[14] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[15] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2021.

[16] J. A. E. 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.