Model Predictive Control for Speed Regulation of Autonomous Vehicles at Road Intersections and Performance Evaluation in a V2X Communication Scenario

Angelo Fasciani[1], Yuriy Zacchia Lun[1], Francesco Smarra[2] and Alessandro D'Innocenzo[1]

Abstract— **The paper aims at evaluating the performance of a centralised control strategy, based on a scheduling procedure and MPC (model predictive control), that regulates the crossing of a four-way intersection by autonomous vehicles avoiding collisions in a V2X (vehicle-to-everything) communication scenario. In particular, we evaluate the performance considering different communication channel gain conditions, with packet loss process implemented through a Bernoullian probabilistic model based on the V2X communication protocol, as well as the impact of transmitting to the vehicles control aggregated commands of multiple future time horizons.**

I. INTRODUCTION

Road crashes are still among the leading causes of non-natural death [1], and more specifically, accidents happening at urban intersections are a problem of interest since they could be prevented by adequately controlling the crossing of cars. Recent advancements in telecommunications and smart vehicle technology enable the development of control systems to regulate traffic flows: modern vehicles integrate many electronic components such as sensors (GPS, lidar, radar, cameras, etc.) and actuators that allow the coordination of intersection passage to reduce accidents and fuel or battery consumption.

This paper focuses on implementing a centralised control strategy based on a sub-optimal sorting algorithm and on model predictive control (MPC) that regulates the flow of autonomous vehicles in an urban intersection, ensuring collision avoidance and a sub-optimal crossing time minimization. The non-idealities induced by the V2X (Vehicle-to-Everything) communication protocol are considered via an appropriate packet loss modeling framework.

Previous work. Other examples of centralised control for urban intersections that can be found in the literature are [2], [3] in which the problem of collision avoidance is treated as a verification problem and then translated into a scheduling one to be solved by a commercial tool, but the communication protocol is not taken into account. In [4], accidents are prevented by minimizing the overlapping of vehicles' trajectories within the intersection: also in this paper, the communication protocol is not taken into account. [5] provides another example of centralised MPC for collision avoidance (plus an optimal crossing sequence algorithm), shaping the problem as queue management: each lane approaching the intersection is treated as a queue, with arrivals considered as an external disturbance and departures as the regulated variable. [6] proposes a centralised control system where the crossing time is minimized by ordering vehicles in categories based on non-colliding trajectories at the intersection and scheduling groups of cars within their category. [7] implements a hybrid approach, meaning that a central scheduler managing vehicles' safe and optimized passage works alongside an algorithm running on each vehicle: the algorithm operates a collision warning that considers only the nearest cars.

Among the decentralised approaches, in [8], [9], [10] and [11] each vehicle solves a two-level optimal problem to pass the intersection minimizing the delays and the control effort. Cars exchange data to ensure safety, but no communication non-idealities are considered. [12] developed a distributed MPC and a priority-based strategy where the scheduler favours faster vehicles closer to the intersection. A similar solution was also proposed in [13]. [14] focuses on a multi-objective optimization problem for intersection coordination, solved in a receding horizon framework.

Communication non-idealities are explored in [15], where the medium access control (MAC) protocol resources shared by vehicles of a platoon are reserved with a semipermanent counting system. For each vehicle the reservation and communication process is modeled by a Markov chain. In [16], packet loss is described by finite-state Markov chains to cope with the fast, time-varying nature of vehicle-to-vehicle (V2V) communications: the authors chose the Nakagami fading model to derive the probability matrix and the steady state distribution. A slightly different technique is considered in [17], where pseudo-Markov chains (PMC) are used to model packet losses, with the PCM parameters derived from both the Rician fading representation and real-world measurement data. Finally, [25] and [26] used finite-state Markov channels to model packet loss processes on sensing and actuation links. They analytically solved the optimal output and state-feedback problems, the second involving the generalized

This work was partially supported by: the Italian government through CIPE Resolution 70/2017 (Centre EX-Emerge); the Italian National Recovery and Resilience Plan of NextGenerationEU through the MoVeOver/SCHEDULE project (CUP J33C22002880001); the EU through projects DigInTraCE (GA 101091801) and Resilient Trust (GA 101112282); the Italian Ministry of Enterprises and Made in Italy (MIMIT) through the Project "SICURA - Casa Intelligente Delle Tecnologie per la Sicurezza - Piano di Investimenti per la Diffusione Della Banda Ultra Larga FSC 2014-2020" under Grant CUP C19C20000520004.

[1]Department of Information Engineering, Computer Science and Mathematics. University of L'Aquila, Italy. angelo.fasciani@student.univaq.it, yuriy.zacchialun@univaq.it, alessandro.dinnocenzo@univaq.it

[2]Department of Civil, Construction-Architectural and Environmental Engineering. University of L'Aquila, Italy. francesco.smarra@univaq.it

packet dropout compensation strategy at the actuators' end.

Paper contribution. In this paper, we develop a centralised MPC solution to guarantee the safe and efficient flow of fully autonomous vehicles at an intersection: this type of control provides robustness with respect to non-idealities such as emergency brakes, drivers not following the suggested speeds, etc. We include a sub-optimal sorting algorithm to minimize the crossing time while keeping the problem quadratic and thus solvable via standard tools with small sampling times. Also, we thoroughly evaluate the system's performance in the event of packet losses by modelling the communication channel with a Bernoulli model congruent with the 5G V2X protocol to offer a detailed insight into this crucial phenomenon. We also investigate the impact of the simultaneous transmission of actuation commands with different future time horizons on the system's performance and safety, taking into account that sending larger packets is associated with larger packet loss probability and communication delays. In this sense, we advance the state of the art by developing a comprehensive methodology that (1) implements a control technique that can be applied in real-time, is robust to the most common disturbances for this class of problems and whose safeness and liveness have been validated via extensive Monte Carlo simulation, and (2) also considers a packet loss model, referring to the most recent and used version of the 5G V2X communication protocol.

Paper organization. The paper is organised as follows: in Section II, we present the intersection model and the dynamic model of the vehicles. The design of the control algorithm is reported in Section III, while Section IV is dedicated to the description of the communication scenario. The paper ends with simulation results in Section V, where we tested the safeness of the control algorithm, and with Section VI, which is dedicated to conclusions and future work.

II. Problem Statement

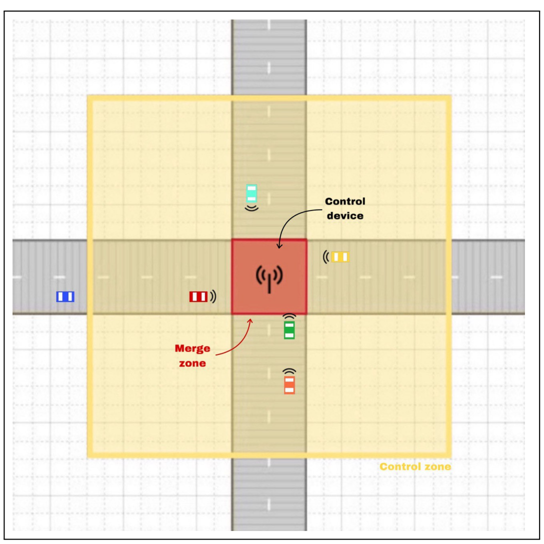

Fig. 1. Illustration of the intersection scheme: a 4-way scenario is shown with the two main operative zones (merge in red, control in yellow). Vehicles inside the control zone communicate with the controller situated in the exact centre of the intersection.

We consider a urban intersection where 4 roads approach orthogonally, as depicted in Fig. 1. Each road consists of two lanes associated to opposite travel directions: lanes are numbered from 1 to 8 clockwise, starting from the top-left. The main areas to deal with the crossing problem are the following:

- the control zone is the yellow square in Fig. 1 with side L centered in the middle of the intersection: it represents the space where the vehicles can communicate with the controller;
- the merge zone is the red square in Fig. 1 with side l: it consists of the surface where roads overlap and lateral collisions may occur.

We assume that the centralised controller is located in the middle of the intersection. The goal is to track the speed trajectories of the vehicles from the moment they enter the control zone until they exit the merge zone, to fulfil safe and efficient intersection flow. We refer to vehicles in this state as *active*. After the completion of the crossing, they can return to their previous driving strategy.

To this end, the n_k vehicles inside the control zone at time k exchange data via the V2X protocol with the control unit. Specifically, they send a packet of metadata at their entrance containing their ID, the lane where they are coming from, the one they want to reach, and the timestamp of their arrival into the control zone. Moreover, they send information about their state (position and velocity), which is assumed fully observable, at each time step until the crossing is completed. Each vehicle is modelled as a discrete-time double integrator, with sampling time τ_s and considered as a point mass moving along a straight line, flawlessly following its trajectory: this is due to the fact that each vehicles is responsible for driving, while the centralised controller only regulates the modulo of the speed, and not its direction. The associated dynamics for each vehicle i are:

$$x_i(k+1) = Ax_i(k) + Bu_i(k),\ \ k \in \mathbb{N}, \tag{1}$$

with $A = \begin{bmatrix} 1 & \tau_s \\ 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} \frac{\tau_s^2}{2} \\ \tau_s \end{bmatrix}$.

The state $x_i(k) = [p_i(k)\ v_i(k)]' \in \mathbb{R}^2$ consists of the car position and speed, while the control input $u_i(k) \in \mathbb{R}$ is the recommended acceleration. The overall dynamics for the n_k vehicles inside the control zone at time k are given by the aggregation of the individual cars' dynamics, resulting in:

$$x(k+1) = A_k x(k) + B_k u(k), \tag{2}$$

where A_k and B_k are diagonal block matrices whose n_k blocks are given by matrices A and B, respectively. Thus, the state vector $x(k) = [x_1(k)\ x_2(k)\ \cdots\ x_{n_k}(k)]'$ is composed by all the states of the n_k vehicles. Similarly, the control vector $u(k) = [u_1(k)\ u_2(k)\ \cdots\ u_{n_k}(k)]'$ is the aggregation of all the control inputs of the active vehicles.

Remark 1: Note that the dimensions of $x(k)$ and $u(k)$ change over time according to n_k. In this paper, as we do not suppose to know in advance when and how many vehicles

will enter the control zone: as a consequence, when solving MPC, we assume that matrices A_k and B_k do not change over the horizon, i.e., $A_{k+j} = A_k$ and $B_{k+j} = B_k$, $j = 0, \dots, N-1$. When a new car enters the intersection, a new MPC problem is formalised with matrices A_k and B_k of appropriate dimensions.
We will assume that the vehicles' speed is constrained in the compact set $\mathcal{V} = [0, v_{max}]$ to avoid reverse gear (lower bound) and guarantee driving comfort/speed limits (upper bound). We will also assume that the vehicles' acceleration is constrained in the compact set $\mathcal{U} = [u_{min}, u_{max}]$, where u_{max} represents the largest tolerated acceleration and u_{min} the most intense admitted braking.

III. Controller Design

To solve the problem of safe and efficient intersection crossing, we developed a two-stage algorithm, composed of a (sub-optimal) scheduler and an MPC solvable via Quadratic programming (QP). The first stage, the scheduler, aims to reduce the time required to empty the merge zone by prioritizing the vehicles that, on the basis of current position and velocity, need less time to exit the merge zone. Then, the order for the vehicles to leave the control zone is translated into linear constraints for the MPC algorithm to keep the computation times acceptable using a standard QP solver.

A. Scheduler

The scheduler has the objective to sort the n_k active vehicles, and consists of two steps. First, cars are ordered by considering the estimated time required to complete the intersection crossing, assuming that each vehicle is taken to maximum velocity v_{max} with maximum acceleration starting from its current state values. Then, a second sorting step is applied that considers the relative positions of the vehicles within the same lane to avoid overpositioning. The scheduling strategy is described in detail in Algorithm 1.

Algorithm 1

1: **Input:** $p(k)$, v_{max}, a_{max}
2: **Output:** idx^s
3: $t_{arr} = []$
4: **for all** $i \in \{1, \dots, n_k\}$ **do**
5: $\quad t_{arr,i} = proj(p_i(k), v_{max}, a_{max})$
6: **end for**
7: $[t^s_{arr}, idx^s] = sort(t_{arr})$
8: **for all** $i \in \{1, \dots, n_k\}$ **do**
9: $\quad j = pred(i)$
10: $\quad$ **if** $t^s_{arr,i} \le t^s_{arr,j}$ **then**
11: $\quad\quad idx^s = [idx^s_{1:i-1}, idx^s_{i+1:j}, idx^s_i, idx^s_{j+1:end}]$
12: $\quad$ **end if**
13: **end for**
14: return idx^s, and use it in constraint (6)

The first sorting step is illustrated in lines 3 to 7, where the vector t_{arr} (i.e., the arrival time estimated for each vehicle) is filled with the projections $proj()$ of each vehicle arrival time. The function $proj()$ simulates the evolution of each active car state by using the dynamic model to compute how many time steps are required to complete the crossing of the merge zone, assuming that from its current state the vehicle is taken to v_{max} accelerating at a_{max}. The maximum speed is then kept constant until exiting the merge zone. Then t_{arr} is sorted in ascending order generating t^s_{arr} (i.e., the ordered vector of times needed by each vehicle to exit the merge zone) and idx^s (i.e., the ordered vector of the corresponding indeces). The second sorting step, illustrated in lines 8 to 13, reorders idx^s to avoid possible conflicts due to scheduled vehicles on the same lane: e.g., if vehicle i is behind vehicle j but has been scheduled before because of a shorter arrival time, then vehicle i will collide with vehicle j: for this reason vehicle i has to be rescheduled to arrive after vehicle j. In the algorithm, this is considered using function $pred()$, where $j = pred(i)$ means that vehicle j precedes vehicle i.

B. Model Predictive Control

Given the crossing order provided by the scheduler via Algorithm 1 the MPC problem (3)–(12) can be formulated to solve an optimal crossing problem of the merge zone as follows:

$$\min_{\mathbf{u}} \sum_{j=0}^{N-1} \left(\|v_{k+j+1} - v_r\|^2_Q + \|u_{k+j}\|^2_R \right) + \lambda \|\varepsilon\|^2 \tag{3}$$

$$\text{s.t.:} \quad x_{k+j+1} = A_k x_{k+j} + B_k u_{k+j} \tag{4}$$

$$p_{i,k+j+1} \ge p_{\nu,k+j+1} + d_{safe}, \quad \nu = pred(i) \tag{5}$$

$$p_{i,k+N} = p_{lim} + \rho_i + \varepsilon_i, \ \ i \in idx^s_{2:n_k} \tag{6}$$

$$v_{k+j+1} \in \mathcal{V}, \tag{7}$$

$$u_{k+j} \in \mathcal{U}, \tag{8}$$

$$\varepsilon_i \ge 0 \quad i = 1, \dots, n_k \tag{9}$$

$$x_k = x(k) \tag{10}$$

$$j = 0, \dots, N-1 \tag{11}$$

$$\mathbf{u} = u_k, u_{k+1}, \dots, u_{k+N-1} \tag{12}$$

The cost function consists of a weighted sum of the vehicles' deviation from a speed reference $v_r \in \mathbb{R}^{n_x}$ and of the control effort: the speed reference for each vehicle is equal to v_{max} (i.e., $v_{r,i} = v_{max}, \ i = 1, \dots, n_x$) to maximise the intersection throughput, while the norm of the control input is considered to penalise intense accelerations/decelerations (i.e., improving driving comfort and reducing energy consumption). The slack variables vector $\varepsilon \in \mathbb{R}^{n_x}$ is used to ensure recursive feasibility despite the constraint (6). Note that, to avoid collisions during the crossing of the merge zone, we imposed constraint (6) on the final positions of the cars to allow only one car at a time to be in the merge zone. The prediction horizon N is imposed equal to the time needed by the first scheduled car to leave the merge zone. All the other vehicles, from the second to the last, must keep a certain distance $\rho_i = t_{arr,1} v_i(k+N) = \frac{l}{v_{max}} v_i(k+N)$ from the beginning of the merge zone, indicated by p_{lim}. The value is cumulative so that a cascade effect originates, allowing all vehicles to pass without collisions and respecting the sub-optimal order. The algorithm can be configured to allow cars scheduled consecutively to cross the intersection simultaneously if their trajectories are not colliding (e.g., both vehicles turn right), thus leading to the definition of two different approaches: single and coupled. For the first one, we suppose to ignore the trajectory of the vehicles and ensure safety by allowing the passage of only one of them at a time through the merge zone. In the second version, the driver's intentions are assumed known, resulting in the possibility of simultaneous passages and a performance improvement in terms of total time needed to empty the intersection.

We also imposed with (5) that all cars must keep a certain distance d_{safe} from the one ahead of them in the same lane to avoid rear-end collisions. The remaining constraints consist of the system dynamics, the update of the initial conditions, speed and acceleration constraints (without slack variables) and the positivity of the slack variables.

Remark 2: In our approach, we did not consider the crossing time as the variable to be minimized, thus making it sub-optimal, allowing us to keep the problem solvable via QP. This is essential in all those applications where real-time computation of the solution is crucial to guarantee safety.

IV. V2X COMMUNICATION SCENARIO

The vehicles and the control infrastructure communicate via independent full duplex channels. It is assumed that transmissions happen exactly at the beginning of each time step, and no propagation nor processing delays are taken into consideration.

The packet loss phenomena are modeled on each channel i by a Bernoullian process with a distance-dependent loss probability $pr_{loss}(d_i(k))$, where $d_i(k)$ indicates the distance of i-th car from the control infrastructure at time k. As the car comes closer to the controller, the probability decreases following the real-world measurements derived from Fig. 9 in [18], where authors reported the packet delivery rate (PDR) for 5G V2X application. Bernoulli's model is justified because the sampling period considered allows us to assume that the packet loss probability at each step is independent of previous ones. In our case, packet loss is only intended on the receiver's end when signal corruption occurs due to interference and disturbances. This phenomenon implies a classification among vehicles as visible or invisible. The first ones manage to communicate correctly with the central unit from the moment they enter the control zone and are the ones we refer to as active, while the latter, failing to send the first packet, remain undetected until their first correct transmission, hence, they cannot be controlled up to that moment.

The loss of information due to packet corruption is managed by exploiting the inherent property of the implemented model predictive control. By computing the control inputs for the active cars over a future horizon, we can provide them with enough information to cope with future non-idealities, like packet losses. Hence, we distinguish three different types of messages used in the system:

- the packet of metadata that vehicles send to the controller as first communication;
- the packet of data containing the current state of a vehicle, sent to the controller at each time step until the exiting;
- the packet of control inputs sent by the controller to the active cars.

The latter can have a variable length len_m depending on how much of the computed horizon we want to send to the cars. By setting $len_m = h$, $h \leq N$, then the message sent by the control unit to car i at time k will contain the input $u_{m,k,i} = [u_{k,i}\ u_{k+1,i}\ \dots\ u_{k+h-1,i}]'$.

The mechanism that regulates the application of the correct control input for vehicle i input at time k is managed by an internal counter $\Delta_{k,i}$, which considers the number of consecutively discarded packets by the car, and it is recursively defined by the following formula:

$$\Delta_{k,i} = (1 - \delta_{k,i})(\Delta_{k-1,i} + 1). \tag{13}$$

where $\delta_{k,i}$ is the stochastic variable which models the single loss event:

$$\delta_{k,i} = \begin{cases} 1 & \text{if there is no packet loss on } i\text{th channel at time } k \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

At time k, the counter is reset to zero if the packet is received correctly ($\delta_{k,i} = 1$) or incremented, if not. When the number of consecutive losses overcomes the length of the horizon that has been sent, the cars keep applying the last control input component, in a HOLD fashion, until they correctly receive a new message from the central unit.

V. SIMULATION RESULTS

We conducted several Monte Carlo simulations on the system to prove its safety and evaluate its performance. All tests were run on Matlab 2024a with the support of the Automated Driving Toolbox for 2D visualisation. For the solution of the optimisation problem, we used the Yalmip framework [19] with Gurobi [20] as quadratic solver. This setup on a centralised workstation ensures control over a reasonable number of cars at a single intersection within the considered sampling period. The chosen values for the simulation parameters are reported in Table 1, in SI unit measurements.

The considered scenario in all trials is the one illustrated in Fig. 1, designed in the Automated Driving Toolbox; the spawning events are regulated by a Poisson process of parameter λ_{poiss}, with the assumption that they happen exactly at the beginning of the designated time step. The value of λ_{poiss}, indicating the average rate of spawning, is referred to the sampling time. In each simulation, we tested a number M of vehicles of the same type (regular car of dimensions: len_{car}, wid_{car}); for all cars the initial speed is randomly selected in the range $[v_{0,min}, v_{0,max}]$. For each sample i, all the associated random values (initial speeds, coming lanes, arrival lanes and spawning time steps) are generated before the simulations by setting the integer seed of the random generator functions equal to i.

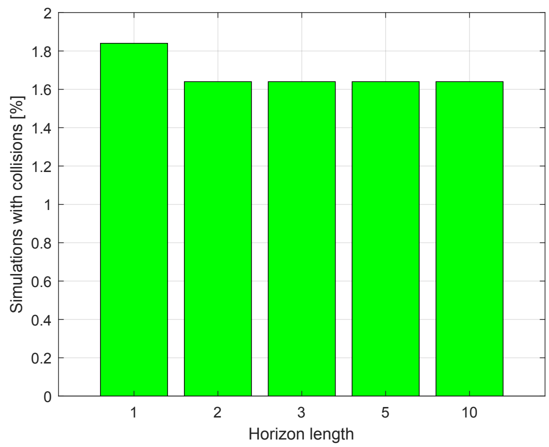

Fig. 2. Percentages of simulations with a detected collision for different lengths of the control input sent to the vehicles.

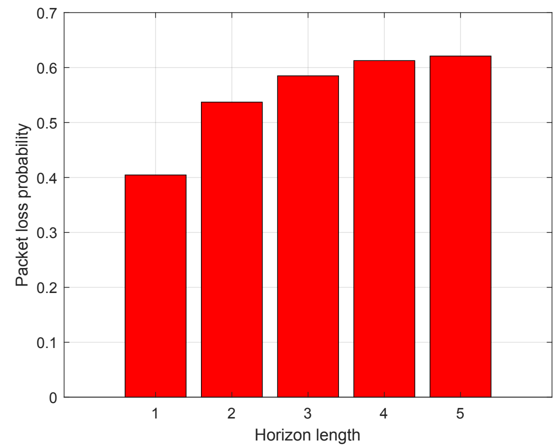

Fig. 3. Maximal packet loss probability that the system can tolerate before a collision occurs for different lengths of the control input sent to the vehicles.

For the first experiment, we evaluated the robustness of the control algorithm by considering real-world measurements for the loss probability by retrieving the values from plots of Fig. 9 in [18], where they tested 5G NR V2X mode 2 technology for various traffic scenarios. In our test, we refer to the data obtained for aperiodic traffic in the case of 25 veh/km, in accordance with the spawning process described at the beginning of this section. Tests were done over $n_{samples} = 10000$ samples by classifying a simulation as failed when at least one collision occurred due to communication issues. In Fig. 2, we show the test results conducted for different lengths of the prediction horizon, finding that from an initial value of 1.84%, the failure percentage settles at 1.64%. This implies that the control system ensured safety in about 98.1% of the times (for the worst case) and that a limit can be found to the robustness achievable by increasing the number of future control inputs sent to the cars. Moreover, it opens a new experimental scenario related to the influence of the intersection structural features (e.g., control zone dimensions, maximum speed value v_{max}, etc.) on the failure percentage. Future work will concern the impact of these parameters' variations on the system with the number of future control inputs sent to the vehicles set to the best trade-off found here.

In the second test, we studied a trade-off between the maximal packet loss probability that the system can bear before collisions become unavoidable and the length of the prediction horizon that the control infrastructure can send to the vehicles. By sending longer messages, cars have more information to cope with future packet loss events, but if consecutive losses continue occurring for enough time due to bad channel conditions, then the control cannot ensure safety anymore. Hence, the scope of the experiment is to understand how low probability can be in this scenario, considering both the robustness of the controller and that a larger message dimension implies the introduction of larger delays over transmissions. In this case, we characterized the Bernoullian model with a constant loss probability, independent from distances. Tests were conducted over $n_{samples} = 1000$ samples, first estimating a critical failure probability pr_{crit} by means of a bisection method and then refining the value by testing all the probabilities in the range $0 : 0.1 : pr_{crit}$. As shown in Fig. 3, simulation data indicate the correlation between the two metrics, with the maximal loss probability accepted by the system that grows as the horizon length increases. In particular, for message length from one to two, we have a relevant 30% improvement in loss probability, while going from length two to five, there is a less steep increase, with the highest achievement at the largest dimension considered, of a 17%. Hence, we claim that two is the best trade-off in this case, meaning that the controller sends input $u_{m,k,i} = [u_{k,i} \quad u_{k+1,i}]'$ to active car i at time k, to achieve the highest loss probability sustainable while occupying the channel for the least time possible.

In the last test, we investigated the difference in performance between the two developed scheduling approaches: single and coupled. Specifically, we confronted the two algorithms' total crossing times, i.e., the time steps required to make all cars exit the intersection. The concept behind it is that in the coupled case, pairs of vehicles can cross the intersection together if their trajectories are not colliding, leading to a faster emptying of the merge zone since cars do not always have to wait for the passage of all the ones scheduled before, as in the single case. The experiment was conducted over $n_{samples} = 100$ samples in ideal communication conditions by setting the packet loss probability for each channel to zero for simplicity. We discovered that there is an enhancement in system performance passing from the single version to the coupled one, of about 1%, which is not much (0.4s less per car, on average), but it's acceptable by considering that the coupled algorithm is derived from the single one, hence it is not natively designed to prefer the passage of paired cars, like the one developed in [6], for example. Though having proved that a greater amount of information leads to better performance, in this case, the single algorithm is preferred due to its relative simplicity.

TABLE I

SIMULATION PARAMETERS

Parameters	Value
Control zone side L	300 m
Merge zone side l	20 m
Number of vehicles M	10
Poisson spawning process constant λ_{poiss}	0.1
Vehicle length len_{car}	4.7 m
Vehicle width wid_{car}	1.8 m
Minimum initial speed $v_{0,min}$	8.3 m/s
Maximum initial speed $v_{0,max}$	12.3 m/s
Maximum speed v_{max}	17 m/s
Maximum acceleration a_{max}	6 m/s^2
Sampling period τ_s	0.1 s

VI. CONCLUSIONS

In this work, we presented a centralised control architecture to manage the crossing of an urban intersection by autonomous vehicles. The control algorithm consisted of a scheduler and an MPC to obtain a sub-optimal minimisation of the time required by the cars to empty the intersection while keeping the problem quadratic and hence solvable in real-time, which is strictly necessary for this kind of applications. We considered the case of packet loss over communication channels by modeling the process with Bernoullian variables and considering both a position-dependent probability and the relation between acceptable loss probability and message length. Finally, we proved the safeness of the algorithm by Monte Carlo simulations.

Future work will focus on testing the algorithm on larger distances to study the impact of the intersection structure on collision avoidance and the implementation of an analytical formula for the derivation of packet loss probability and more complex models as Markov chains to deal with burst losses. It is also planned to set up a real implementation of this scenario, with real cars and a V2X communication system, in collaboration with the partners of the MoVeOver/SCHEDULE project indicated in the acknowledgments. Sensitivity analysis on key parameters will be considered as well to understand how variations in the controller tunings affect performance. Also, given the complexity of the problem, more advanced models can be considered to take into account more elaborate scenarios. This can be achieved through machine learning algorithms, such as regression trees, random forests, and neural networks, combined with control algorithms, building on methods developed in [21], [22], while also incorporating stochastic elements [23] and ensuring stability guarantees [24].

REFERENCES

[1] World Health Organization, "Road traffic injuries." [Online]. Available: www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[2] H. Ahn and D. Del Vecchio, "Semi-autonomous intersection collision avoidance through job-shop scheduling," in *Proc. 19th Int. Conf. Hybrid Syst. Comput. Control*, 2016, pp. 185–194.

[3] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1515–1527, 2014.

[4] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 81–90, 2012.

[5] F. Creemers, A.I.M Medina, E. Lefeber, and N. van de Wouw, "Design of a supervisory controller for cooperative intersection control using model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 74–79, 2018.

[6] F. Yan, M. Dridi, and A. El Moudni. "Autonomous vehicle sequencing algorithm at isolated intersections," *12th Int. IEEE Conf. Intell. Transp. Syst.*, 2009, pp. 1–6.

[7] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 804–818, 2011.

[8] A.A. Malikopoulos, L. Beaver, and I.V. Chremos, "Optimal time trajectory and coordination for connected and automated vehicles," *Automatica*, vol. 125, 2021, Art. no. 109469.

[9] S.D. Kumaravel, A.A. Malikopoulos, and R. Ayyagari, "Optimal coordination of platoons of connected and automated vehicles at signal-free intersections," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 186–197, 2022.

[10] A.A. Malikopoulos and L. Zhao, "Optimal path planning for connected and automated vehicles at urban intersections," *IEEE 58th Conf. Decis. Control (CDC)*, 2019, pp. 1261–1266.

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

[12] M. Kloock, P. Scheffe, S. Marquardt, J. Maczijewski, B. Alrifaee, and S. Kowalewski, "Distributed model predictive intersection control of multiple vehicles," *IEEE Intell. Trans. Syst. Conf.*, pp. 1735–1740, 2019.

[13] K. Kim and P.R. Kumar, "An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3341–3356, 2014.

[14] G.R. Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "Cooperative receding horizon conflict resolution at traffic intersections," *53rd IEEE Conf. Decis. Control (CDC)*, 2014, pp. 2932–2937.

[15] X. Gu, J.Peng, L. Cai, X. Zhang, and Z. Huang, "Markov analysis of C-V2X resource reservation for vehicle platooning," *IEEE 95th Veh. Technol. Conf.*, June 2022, pp. 1–5.

[16] S. Lin, Y. Li, Yuanxuan Li, B. Ai, and Z. Zhong, "Finite-state Markov channel modeling for vehicle-to-infrastructure communications," *IEEE 6th Int. Symp. Wireless Veh. Commun.*, 2014, pp. 1–5.

[17] I. Bocharova *et al.*, "Characterizing packet losses in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8347–8358, 2019.

[18] A. Molina-Galan, L. Lusvarghi, B. Coll-Perales, J. Gozalvez, and M. L. Merani, "On the impact of re-evaluation in 5G NR V2X Mode 2," *IEEE Trans. Veh. Technol.*, vol.73, no. 2, pp. 2669–2683, Feb. 2024.

[19] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in Proc. Int. Conf. Robot. Autom., Taipei, Taiwan, 2004.

[20] Gurobi Optimization, LLC, *Gurobi optimizer reference manual*, 2024.

[21] F. Smarra, G.D. Di Girolamo, V. De Iuliis, A. Jain, R. Mangharam, and A. D'Innocenzo, "Data-driven switching modeling for MPC using regression trees and random forests," *Nonlinear Analysis: Hybrid Systems*, vol. 36, Art. no. 100882, 2020.

[22] A. Jain, F. Smarra, E. Reticcioli, A. D'Innocenzo, and M. Morari "NeurOpt: Neural network based optimization for building energy management and climate control," *Learning for Dynamics and Control*, pp. 445–454, 2020.

[23] F. Smarra, and A. D'Innocenzo "Learning Markov jump affine systems via regression trees for MPC," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5252–5257, 2020.

[24] V. De Iuliis, F. Smarra, C. Manes, and A. D'Innocenzo, "Stability analysis of switched ARX models and application to learning with guarantees," *Nonlinear Analysis: Hybrid Systems*, vol. 46, Art. no. 101250, 2022.

[25] A. Impicciatore, Y. Zacchia Lun, P. Pepe, and A. D'Innocenzo, "Optimal output-feedback control over Markov wireless communication channels," *IEEE Trans. Autom. Control*, vol. 69, no. 3, pp. 1643–1658, 2024.

[26] Y. Zacchia Lun, F. Smarra, and A. D'Innocenzo, "Optimal control over Markovian wireless communication channels under generalized packet dropout compensation," *Automatica*, vol. 176, Art. no. 112240, 2025.