$IA(MP)^2$: framework for online motion planning using interaction-aware motion predictions in complex driving situations

Juan Felipe Medina-Lee, Vinicius Trentin, Juan Luis Hortelano, Antonio Artuñedo, Jorge Godoy, and Jorge Villagra

***Abstract*—Motion planning is a process of constant negotiation with the rest of the traffic agents and is highly conditioned by their movement prediction. Indeed, an incorrect prediction could cause the motion planning algorithm to adopt overly conservative or reckless behaviors that can eventually become a dangerous driving situation. This paper presents a framework integrating motion planning and interaction-aware motion prediction algorithms, which interact with each other and are able to run in real-time on complex areas such as roundabouts or intersections. The proposed motion prediction strategy generates a multi-modal probabilistic estimation of the future positions and intentions of the surrounding vehicles by taking into account traffic rules, vehicle interaction, road geometry and the reference trajectory of the ego-vehicle; the resulting predictions are fed into a sampling-based maneuver and trajectory planning algorithm that identifies the possible collision points for every generated trajectory candidate and acts accordingly. This framework enables the automated driving system to have a more agile behavior than other strategies that use more simplistic motion prediction models and where the planning stage does not provide feedback. The approach has been successfully evaluated and compared with a state-of-art approach in highly-interactive scenarios generated from public datasets and real-world situations in a software-in-the-loop simulation system.**

***Index Terms*—autonomous vehicles, motion prediction, motion planning, interaction-aware.**

I. Introduction

Autonomous Vehicles (AV) are complex systems with software components that rely on each other to understand their surroundings and produce behaviors that allow them to drive in the real world. Decision-making is one of the critical tasks that AVs need to execute properly. It may become very challenging in crowded or highly dynamic environments, where it is often difficult to accurately predict the motion of traffic agents and generate an acceptable motion pattern accordingly.

To design an optimized decision-making system, the autonomous driving stack has traditionally been implemented as a function disaggregation: behavior planning, motion planning and motion prediction [1]. Under this paradigm, these are separate and sequential components, instantiating methods to predict driver trajectories first and using thereafter these predictions to plan collision-free maneuvers and trajectories in response [2].

Given the safety-critical nature of driving, many existing decision-making systems have primarily focused on safety [3], modeling the surrounding traffic agents as bounded disturbances [4] or evolving with oversimplified motion patterns (e.g. constant speed [5]). In addition to that, many prediction systems do not take into consideration (or do it too simplistically) the traffic agents' interactions, which are prevalent in traffic scenarios [6]. As a result, the planning components often ignore the mutual impact between the behaviors of AVs and the surrounding actors. This effect often leads to non-human-like behaviors, excessively over-cautious, potentially causing disruptions to the traffic flow and even originating dangerous situations [7].

In addition to the foregoing, decision-making in driving environments with multiple dynamic agents is challenging due to their associated uncertainty. Environment occlusions, sensor noise or data-tracking association errors are some of the measurement-related sources of uncertainty [8]. Also, the vehicle plan must consider the uncertainty regarding the state of nearby agents and, especially, their potential intentions (often conditioned by their interactions with other traffic participants). Both sensing and agents' evolution uncertainties appear in multiple forms in urban driving situations such as turning at an intersection or changing lanes.

Recent decision architectures try to cope with these demanding requirements using machine learning approaches [9]. However, the unpredictable nature of these techniques can sporadically lead to unreliable predictions, which in turn may have dangerous planning consequences. This can occur when the traffic situation is significantly different from the training dataset.

To overcome these limitations, this paper proposes a decision-making system that predicts a set of consistent probabilistic evolutions of the involved traffic agents and plans accordingly an action that efficiently deals with potential imminent hazards while not overreacting to low probability dangers further into the future. It will be shown how a multi-modal interaction-aware motion prediction system is seamlessly integrated with a sampling-based motion planning algorithm. The former takes into account the interaction between vehicles, road geometry and traffic rules, thus enabling the maneuver and trajectory planner to have a more agile behavior that would not be possible by using more simplistic motion

The authors are with the Centro de Automática y Robótica, CSIC-Universidad Politécnica de Madrid, 28500 Arganda del Rey, Madrid, Spain. (email: juan.medina@csic.es; vinicius.trentin@csic.es; juanluis.hortelano@csic.es; antonio.artunedo@csic.es; jorge.godoy@csic.es; jorge.villagra@csic.es). (Juan Felipe Medina-Lee and Vinicius Trentin are co-first authors.)

prediction models. The interactions between prediction and planning modules are shown in Fig. 1 where, in the forward direction, the prediction module indicates to the planner the probable occupied area along a given time horizon; and in the reverse direction, the reference trajectory of the ego-vehicle (EV) influences the predictions for the surrounding vehicles.

Although the prediction method has already been presented in previous publications, it has never been evaluated in real-time coupled with the motion planning algorithm. In this paper, the interactions between planning and predictions are tested in real-time simulations with scenarios obtained from publicly available datasets.

The main contributions of the proposed approach can be summarized as follows:

- A novel decision-making strategy that reproduces the interactions among traffic agents by explicitly considering the mutual interplay between planning and prediction components while keeping a traceable course of action.
- An approach that handles both perception and agents motion uncertainty combining in a real-time setting multi-modal probabilistic interaction-aware prediction with a lane-based maneuver planner and a sampling-based trajectory planner.
- A resulting decision-making framework that can be effectively used in highly dynamic urban scenarios. Several relevant urban use cases are exhaustively analyzed in real-time closed-loop experiments. They are extracted from open datasets in intersections and roundabouts where interacting vehicles are densely distributed.

The paper is structured as follows: Section II presents a review of the most relevant related work, Section III describes the complete framework for interaction between planning and prediction and the two cooperating algorithms in detail, Section IV gives an in-depth explanation of the implementation requirements and deployment, Section V describes the experimental setup and the results obtained from testing the presented hypothesis. Finally, Section VI gives a concluding review of the outcome of the project.

II. State of the Art

Predicting the behavior of traffic participants is a challenging task as human drivers do not always adhere to road rules or conventions producing ultimately complex behaviors. Furthermore, the interaction between actors may cause the driving scene to evolve in many different ways. Motion planning strategies need to transform this uncertainty into efficient, safe, comfortable and explainable motion patterns. To face these challenges, motion prediction and planning, which are two tightly-coupled components, have recently received attention as a joint task. These works can be categorized into two main groups: rule-based and learn-based.

A rule-based system consists of a set of predefined rules that mimic human behavior to model the interactions between agents and the environment via logical constraints, whereas a learn-based system makes use of machine learning approaches, such as neural networks, where the behaviors and interactions are self-learned and the outputs are obtained given the environment information.

Rule-based approaches are applied in [10] and [11] where a combination of an Interacting Multiple Model filter with a Kalman Filter is used to predict the future positions of the surrounding vehicles at intersections [10] and highways [11]. In both cases, the trajectory of the ego-vehicle is planned as a constrained optimization problem based on the Model Predictive Control (MPC) framework that takes into account traffic rules and the surrounding vehicles. Due to the uncertainty produced by the Kalman Filter, these methods should only be used to predict small prediction horizons. The computation of planning and control by the same algorithm demands low computation time, otherwise, the system might get locked which would not happen when planning and control are decoupled.

With the increased development of artificial intelligence, motion prediction and planning methods based on machine learning have been widely used. Zeng et al. [12] propose an interpretable neural motion planner that takes as input LiDAR data and an HD map and generates a space-time cost volume using a Convolutional Neural Network (CNN) and considering the detection of the obstacles and their future predictions. From this output, trajectories are sampled and the one with the minimum cost is selected. Another CNN is used in [13] where images are taken as input to generate a recurrent prediction of the future semantic map from which the ego-vehicle trajectory is obtained. Combining prediction and planning in end-to-end machine-learning models has shown interesting performances [14] but needs a large dataset to train the models and struggles to guarantee safety when their approach is not flexible enough [15]. In other recent approaches agents' interactions are characterized using either model-free [16][17] or model-based [18][19][20] prediction techniques, that provide a more consistent understanding in most of driving scenes. However, some of them still require to have large numbers of samples to cover the long tails of the occurrence probability distribution.

Some works are hybrid, using a mixture of rule-based and learn-based techniques. Given LiDAR data and an HD map, the authors in [7] use a Convolutional Neural Network and a spatially-aware Graph Neural Network to extract and predict multi-modal trajectories for the agents at the scene, considering interaction and traffic rules. An MPC controller is used to generate a single common immediate action that is safe to all possible developments of the scene. In [21], the indicators time-to-collision and time headway are computed to assess the risk caused by the surrounding vehicles regarding the ego-vehicle and are used to evaluate candidate sequences of acceleration and lateral position, computed with a radial basis function neural network. These indicators might not be able to fully represent the interaction between vehicles, since they are computed assuming a constant speed model. Huang et al. [22] recently proposed the use of a Dynamic Graph CNN with a nonlinear optimizer that explicitly considers dynamic constraints and traffic rules, enhancing performance and safety.

Most of the existing motion planners occasionally produce scene-unaware outputs, as they often ignore the fact that some agents' future motion cannot happen at the same time [23]. Some of these planners also assume that the ego-vehicle commits to a single long-term trajectory when in practice it can

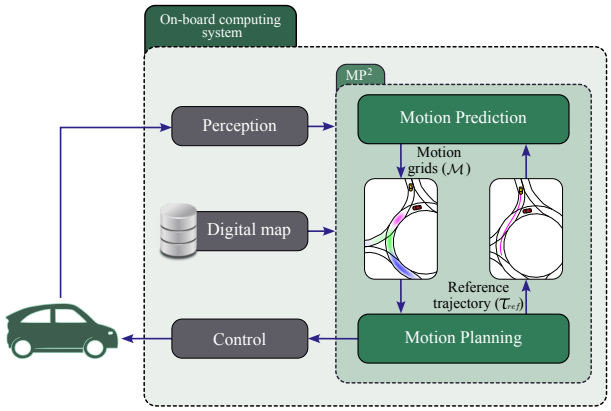


Fig. 1. (a) Architecture of the system. (b) Motion-prediction and motion-planning interaction scheme.

execute a short-term action and replan with a high frequency, which can lead to suboptimal and overly conservative trajectories. Motion planners relying on multi-modal probability predictions, such as [24] or [25], have proven to be a good remedy to palliate such undesired behaviors, but they are often built on data-driven frameworks, which may eventually lead to undesired behaviors.

Finally, note that a large number of works have been proposed for highway environments (e.g. [26]), but much less research activity that takes into account both motion planning and motion prediction simultaneously has been identified for other complex driving maneuvers where social interactions are even more relevant, such as roundabouts (e.g. [18]) or intersections (e.g. [10]).

III. Architecture

The autonomous driving system proposed in this work is divided into 5 different modules with specific tasks in the driving process: Perception, Digital map, Motion prediction, Motion planning and Control. This section describes the motion prediction and motion planning modules and how they interact to improve the performance of the system. The architecture onboard the autonomous vehicle and the detailed interaction scheme between the motion prediction and the motion planning modules can be seen in Fig. 1.

First, the perception module processes data from the sensors to estimate the surrounding vehicles' position, shape and speed. The digital map stores information about road elements like lanes or regulatory signals; this map uses the *lanelet2* [27] format, which represents drivable areas by modeling its left and right bounds in objects named lanelets; it also creates an adjacency graph using the relationship between adjacent lanelets. The motion planning module receives the lanelet map and creates a global route from the current position of the EV to a goal position. Note that in order to achieve an appropriate performance of the proposed architecture the digital map is expected to be accurate in the lane definition and to have the regulatory elements well defined.

The motion prediction module takes the information about the surrounding vehicles generated by the perception module, the map and the trajectory of the EV to produce a 3D

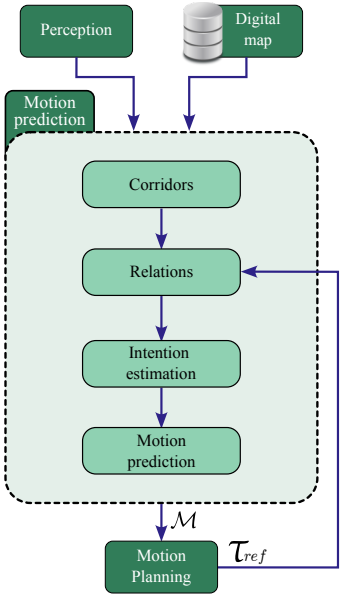


Fig. 2. Intention inference and motion prediction flowchart [28].

motion grid $\mathcal{M} = \left\{M^1, ..., M^{N_P}\right\}$, $N_P \in \mathbb{N}$, where M^i represents the prediction for the i^{th} timestep and N_P is the size of the prediction horizon vector, containing the informed motion predictions of the other vehicles (OV). The motion planning module then uses $\mathcal{M}$, along with other information regarding the current driving scene, to update the reference trajectory (τ_{ref}), which complies with safety, comfort and time constraints. This trajectory is fed-back to the motion prediction module, as depicted in Fig. 1, so that the upcoming set of predictions is aware of the ego-vehicle's intentions. Finally, the control component transforms τ_{ref} into the necessary steering wheel, throttle and brake commands for the vehicle to follow the reference trajectory.

The following sections describe in detail the motion prediction and motion planning blocks, which are the main components of the $IA(MP)^2$ architecture proposed in this work. Their interaction and mutual influence are also presented.

A. Interaction Awareness Motion Prediction

The Interaction-Aware Motion Prediction (IAMP) is a probabilistic method that infers the intentions of the surrounding vehicles, using Dynamic Bayesian Network (DBN), and predicts their subsequent via Markov Chains. It explicitly considers interactions, the road layout, and traffic rules, and can be used in any driving context as it does not rely on training data.

At each time step, the vehicle-to-vehicle and vehicle-to-layout interactions are taken into account to infer the probability of stopping or crossing the intersections, the probability of changing lanes, and the probability of being in each of the possible navigable corridors. These intentions are updated every time new sensor data arrives and are fused with multi-modal motion predictions computed with a kinematic model to result in a motion grid used by the EV to navigate through the scene.

The grid-based representation of the predictions takes into account the uncertainties both in the motion model and in the input data, resulting in a more reliable and robust prediction when compared with a point-based trajectory prediction. Besides, it offers a probability distribution over possible future trajectories of the surrounding vehicles, allowing the motion planner to understand the likelihood of different outcomes in a better way. This helps the EV to make more cautious decisions when facing uncertain situations where the interactions between vehicles are high.

The algorithm is summarized in the flowchart from Fig. 2. Each state is extensively described in [28] and will be briefly reviewed here.

1) Corridors: A navigation corridor $\zeta^i, \ \ i = \ 1, 2, ..., N_C$ is a lanelet sequence that represents a possible route a vehicle might take. Firstly, the lanelets the vehicle is at are found using its pose. Secondly, a graph search for adjacent lanelets is done beginning with the vehicle lanelets, until the maximum reachable distance within a time horizon is reached. For every corridor, a road-shaped grid is created, over which the motion prediction is projected. This grid is obtained from the centerline of the corridor, which is adapted every iteration to be as close as possible to the vehicle's center.

2) Relations: The interactions obtained from the set of corridors ζ and the map are threefold: lateral relation, corridor-to-intersection (distance to intersections) and corridor-to-corridor (corridors dependencies);

- lateral relation: for each target vehicle, a search of the surrounding vehicles is performed, and the resulting bumper-to-bumper distances and speeds are stored.
- corridor-to-intersection: the intersections each corridor goes through are determined by intersecting the lanelet's identifiers of the corridor with the ones from the intersections. For all intersections a corridor goes through, the distance to the intersection is computed in the Frenet frame. The entrance through which the corridor passes is also determined.
- corridor-to-corridor: determines which corridor will influence the predictions of the other, acting as an obstacle ahead. The centrelines of all corridors are pairwise intersected, generating a list of possible collisions. Based on a set of rules, one (if any) corridor is selected as the corridor influencing the motion of a vehicle in a given corridor.

3) Intention's inference: The Dynamic Bayesian Network (DBN) proposed in [29] inspired the one implemented here to compute the intention of traffic participants. The network represented in Fig. 3 is instantiated for each of the vehicles present in the scene, where bold arrows represent the influences of the other vehicles on vehicle n through some key variables (E_t^n, I_t^n, R_t^n Φ_t^n, Z_t^n).

- Expected maneuver E_t^n: reflects the expected behavior of the vehicle n at moment t in accordance with traffic laws.
- Intended maneuver I_t^n: reflects the intention of the vehicle.
- Route R_t^n: contains the desired route followed by the vehicle.

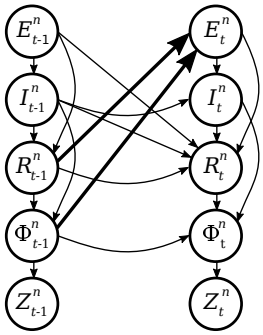


Fig. 3. Bayesian network for two consecutive time steps.

- Physical vehicle state Φ_t^n: represents the pose, curvature, and speed of the vehicle. They are determined at each instant based on the vehicles' intentions.
- Measurements Z_t^n: represents the real measurements of the physical state of the vehicle, obtained directly from exteroceptive sensors of the EV or via V2X communications [30].

The relationships between the variables in Fig. 3 allow the following generalized distribution to model the driving scene:

$$\begin{aligned} P(\boldsymbol{E_{0:T}}, \boldsymbol{I_{0:T}}, \boldsymbol{R_{0:T}}, \boldsymbol{\Phi_{0:T}}, \boldsymbol{Z_{0:T}}) = \\ P(\boldsymbol{E_0}, \boldsymbol{I_0}, \boldsymbol{R_0}, \boldsymbol{\Phi_0}, \boldsymbol{Z_0}) \\ \times \prod_{t=1}^{T} \times \prod_{n=1}^{N} [P(E_t^n | \boldsymbol{R_{t-1}} \boldsymbol{\Phi_{t-1}}) \times P(I_t^n | I_{t-1}^n E_t^n) \\ \times P(\Phi_t^n | \Phi_{t-1}^n R_t^n I_t^n) \times P(Z_t^n | \Phi_t^n)] \end{aligned} \quad (1)$$

Since an exact inference of (1) is generally impractical, a particle filter is employed to estimate the hidden states $\mathbf{E_t}$, $\mathbf{I_t}$, $\mathbf{R_t}$ and $\mathbf{\Phi_t}$, given the observed variables $\mathbf{Z_t}$. This process is detailed in [31].

4) Motion prediction: The computation of the predictions of the surrounding vehicles is inspired by the method proposed by [32].

The system dynamics are abstracted into Markov chains, where the state space X and input space U are discretized into intervals. The state space consists of longitudinal position s and speed v, and the input space represents the potential acceleration a vehicle might use.

The transition probability matrices of the Markov chains for a time step $\Upsilon(\Delta t)$, and for a time interval $\Upsilon([0, \Delta t])$, where Δt is the time increment, are computed offline with Monte Carlo simulation [32], using the following differential equation as the vehicle's longitudinal dynamics:

$$\begin{aligned} \dot{s} &= v \\ \dot{v} &= \begin{cases} a^{\max} u, & 0 < v < v^{\max} \\ 0, & v \leq 0 \vee v \geq v^{\max} \end{cases} \end{aligned} \quad (2)$$

where $a^{\max}$ and $v^{\max}$ are the maximum allowed acceleration and speed, respectively, and u is sampled for the discretized input space U.

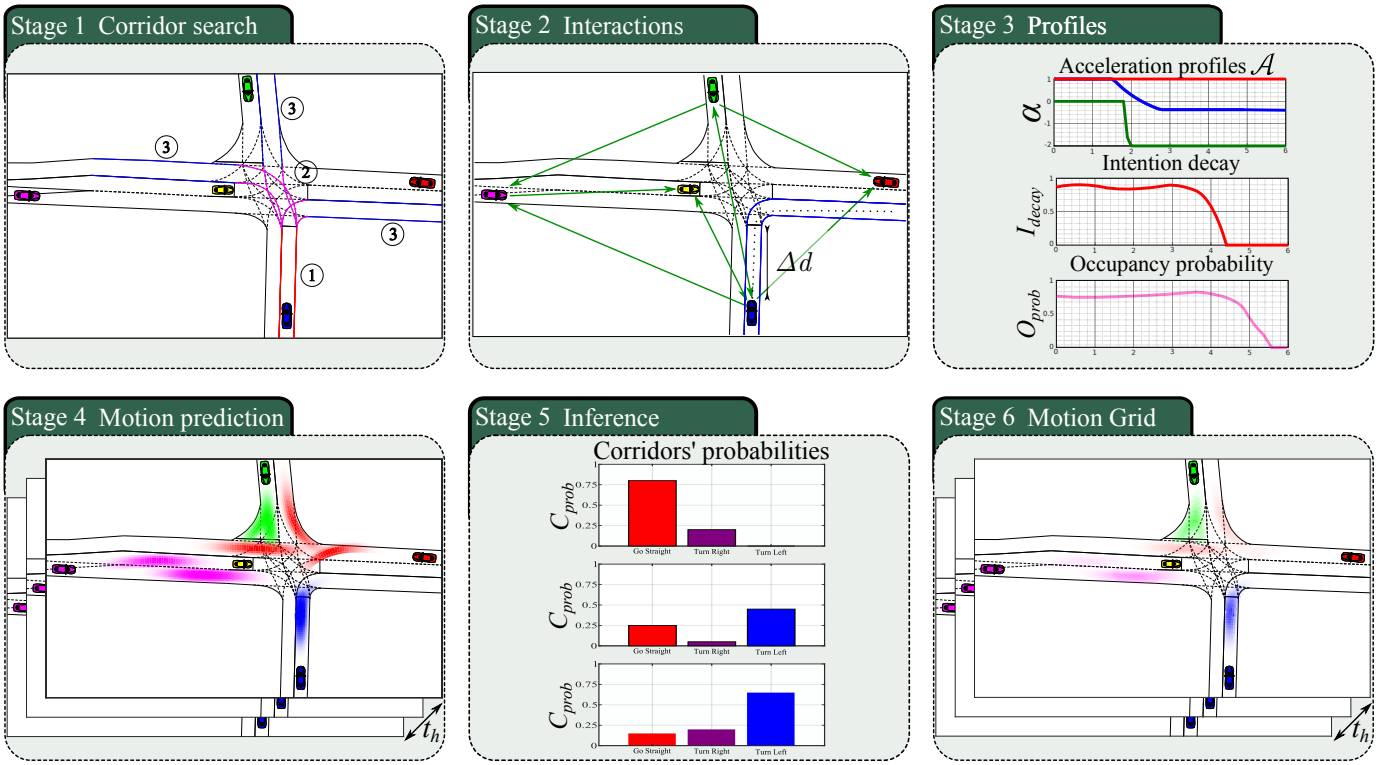


Fig. 4. Integration of DBN + Markov chains: In *Stage 1* all possible corridors are found and in *Stage 2* the interactions are found. In *Stage 3* the acceleration profiles are computed taking into account the longitudinal intention decay I_{decay} and the occupancy probability O_{prob}. In *Stage 4* the predictions of each corridor are computed with equation (3). In *Stage 5* the probabilities of being in each of the possible corridors are inferred, and in *Stage 6* the corridors' predictions are joined into the motion grid considering these probabilities.

The states probability distributions for future time steps $p(t_{k+1})$ and time intervals $p(t_k, t_{k+1})$ are computed as follows:

$$\begin{aligned} p(t_{k+1}) &= \Gamma(t_k)\Upsilon(\Delta t)p(t_k) \\ p(t_k, t_{k+1}) &= \Upsilon([0, \Delta t])p(t_k) \end{aligned} \tag{3}$$

where $\Gamma(t_k)$ is the input transition matrix that represents the transition probabilities between the input states.

The complete diagram that summarizes the process can be seen in Fig. 4. Below, the system is described briefly but for a more in-depth account, the reader is referred to [28]. In order to compute $\Gamma(t_k)$ it is necessary to process the scene and find out all the corridors (Stage 1) and the dependencies between vehicles and with the layout, which happens in Stage 2. Once the interactions are found, they are used to produce acceleration profiles $\mathcal{A}$ for each vehicle in Stage 3, which take into account the longitudinal intention decay I_{decay} (obtained from I_t^n and the relations at the intersections) and the occupancy probability O_{prob} of the corridor causing the dependency (if any).

In Stage 4, the acceleration profiles $\mathcal{A}$ are used to generate individual equiprobable predictions for each car and corridor. Then, the DBN is used to infer each corridor's probability C_{prob} in Stage 5. Finally, the predictions of each corridor are fused together with their probabilities to generate a three-dimensional motion grid $\mathcal{M}$ in Stage 6. In Stages 4 and 6 from Fig. 4, the color of the predictions (for the last time interval) matches the color of the vehicles. At the end of each iteration, the motion grid is compressed and sent to the motion planner which evaluates the possible threats that the EV might face when pursuing its current path.

Since the trajectory τ_{ref} of EV is known, it is expected that the OVs will act accordingly and the predictions of the surrounding OVs will be influenced by τ_{ref}. An example of the effects of knowing the trajectory is presented in Fig. 5, where for the same initial conditions, the EV has two different τ_{ref}; one is to go straight (Fig. 5a) and the other one is to do a right turn at the intersection (Fig. 5b). As can be seen, the predictions for the green OV take the EV's motion plan into account and reacts differently to the different τ_{ref}. The difference in the predictions is due to the different speed profiles caused by the curvature of the EV's path (note the different accelerations at each trajectory). It is important to notice that the EV does not always have priority over the other vehicles. Its trajectory takes into account traffic signs and the interaction with the other agents through the motion grid, as explained in the next section.

B. Motion Planning

The motion planning module used in this work is divided into two main components, a maneuver planner and a trajectory generator. This hierarchical architecture has been proven to be efficient on autonomous driving systems [33] since it allows to solve complex driving urban scenarios such as changing to a more convenient lane, waiting on a pedestrian

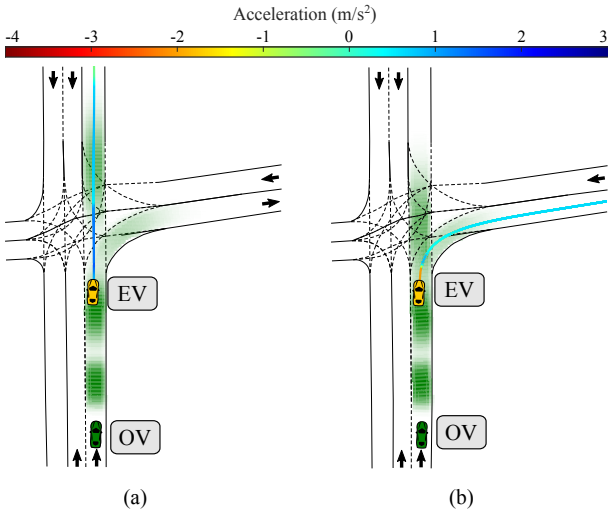


Fig. 5. Motion grid showing 3 prediction instants inside the prediction horizon, considering two different trajectories τ_{ref} (in blue) for the EV: (a) going straight. (b) turning right.

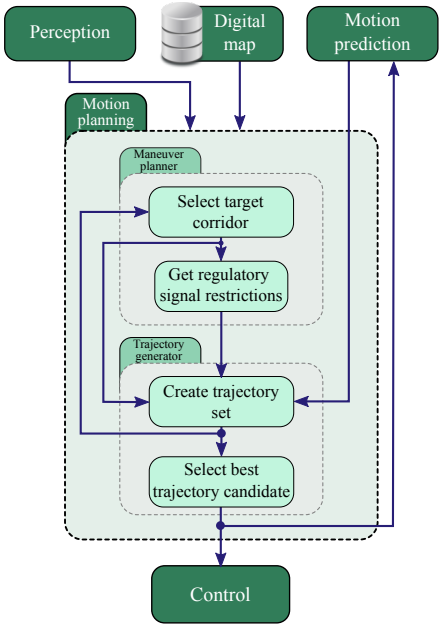


Fig. 6. Motion planning architecture.

crossing, merging efficiently on a roundabout, or handling a crossroad intersection properly. The maneuver planner selects the most appropriate navigation corridor for the EV and also establishes a behavior that adapts according to the traffic regulatory signals. The trajectory generator creates a set of trajectory candidates $\mathcal{T} = \{\tau^1, ..., \tau^{N_T}\}$, each of which consists of a path and a speed profile, that take into account the conditions set by the maneuver planner and the motion grid $\mathcal{M}$. The best candidate is selected as a reference trajectory τ_{ref} using a multi-objective merit function. The architecture of the motion planning module is depicted in Fig. 6.

The maneuver planner and the trajectory generator are briefly described below.

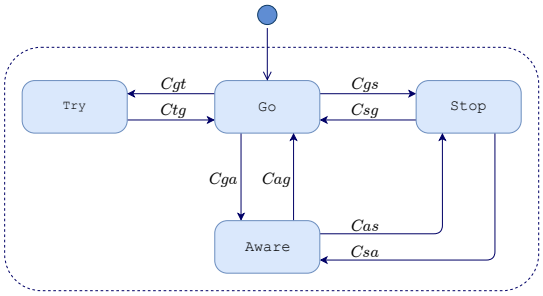


Fig. 7. FSM for traffic regulation state selection.

1) Maneuver planner: The maneuver planner is in charge of two main tasks: selecting the navigation corridor for the EV and obtaining a set of restrictions to handle the signal regulations present on the scene.

The target corridor ζ_t is selected by evaluating a utility function based on the lane-selection model proposed in [34]. This approach considers different explanatory variables associated to each navigation corridor of the EV, such as the speed of the OVs on the corridor, the convenience with respect to the global route, the location of the corridor on the road, etc.

The other main task of the maneuver planner is to establish a set of restrictions for the trajectory generator to properly handle the regulatory signals. This is done by using a Finite State Machine (FSM) that selects between four possible traffic regulation states $S = \{\texttt{Stop}, \texttt{Try}, \texttt{Aware}, \texttt{Go}\}$ (see Fig. 7) [5]. The value of the current state S_k is set according to the type of the closer regulatory signal on ζ_t and to the traffic state, as described in Table I. Depending on the current value of S_k, the trajectory generator will create different types of speed profiles for the trajectory candidates.

TABLE I
STATE TRANSITION CONDITIONS FOR THE TRAFFIC REGULATION STATE FSM.

Condition	Definition
c_{gt}	A right-of-way signal is detected in the traffic scene.
c_{tg}	The right-of-way signal is near the EV and there is not a close vehicle in the intersecting lanes.
c_{ga}	A pedestrian crossing signal is detected in the traffic scene.
c_{ag}	The detected pedestrian crossing signal has been crossed by the EV.
c_{as}	A pedestrian is detected in the crossing zone.
c_{sa}	No pedestrian is detected in the crossing zone.
c_{gs}	A must-stop signal is detected in the traffic scene.
c_{sg}	The EV is stopped at a must-stop signal and there is no near obstacle in the intersection. Or the EV is stopped at a pedestrian crossing signal and the crossing zone has been free over a period of time.

2) Trajectory generator: The trajectory generator creates N_T trajectory candidates and then, using a merit function selects the best of them, which is set as a reference trajectory τ_{ref} for the EV to follow. This process is repeated several times, as the EV drives in the scenario, in order to maintain an appropriate trajectory accordingly to the traffic dynamic

conditions. Each candidate $\tau^i \in \mathcal{T}$ is formed by a path (ρ^i) and a speed profile (V^i). The paths are created using 5th order Bézier curves starting from a future point of the current reference trajectory and ending on different way-points located along ζ_t. This process is depicted in Fig. 8, which shows two time-steps of a roundabout-crossing maneuver with two OVs. The first time-step (Fig. 8a) shows the EV merging into the roundabout while there is a green vehicle OV_1 and a red vehicle OV_2 inside the roundabout. The figure shows the path candidates in orange, as well as the motion grids corresponding to the predictions in 2 s, 4 s and 6 s. The colors of the motion grids match the color of the vehicles. Finally, the trajectory selected as τ_{ref} is plotted using a green dotted line. Fig. 8b shows the path candidates when the EV has already merged into the roundabout. The blue stars depict the way-points used as ending points for the candidates. It can be observed that the motion prediction algorithm identified a possibility for OV_2 to leave the roundabout from the inner lane –even though it is a bold maneuver that should be avoided in a normal situation– requiring a speed reduction from the EV to maintain a safe distance. The speed-profile generation process for this use case is discussed later in this section.

For each path, a set of speed profiles is generated taking into account comfort and safety constraints, as well as the potential traffic regulations that may be imposed by S_k. In case the current traffic-regulation state is $S_k = \texttt{Stop}$, the generated speed profile stops on the yielding line of the traffic signal; note that this state is only reachable when the EV is facing a must-stop traffic signal. If the current state is $S_k = \texttt{Try}$ and there is a lag vehicle (OV_{lag}) in the scenario, the speed profile tries to merge before that vehicle. If $S_k \neq \texttt{Try}$ or is not possible to merge on the signal before the OV_{lag}, a follow-leader speed profile is generated. In case $S_k \neq \texttt{Go}$, the regulatory signal is omitted in the speed generation process. This process is depicted in Fig. 9.

The future positions of the traffic agents are crucial in the speed generation process since the speed profiles for the trajectory candidates take them into account to maintain safe distances at all times with other OVs. The predicted positions of each OV are projected into each path candidate ρ^i using a 1D spatio-temporal representation, known as Possible Collision Points (PCP). Let $\mathcal{P}^i$ be the occupancy polygon of the EV when tracking ρ^i; the PCPs are found by intersecting $\mathcal{P}^i$ and the motion grids $M^h \in \mathcal{M}$. A single PCP is created from the center of mass of the intersection between $\mathcal{P}^i$ and the prediction area of an OV at M^h. The accumulated probability of that prediction-area has to be larger than a predefined threshold $\epsilon_{\text{PCP,min}}$ in order to be considered. This process is repeated for every $M^h \in \mathcal{M}$. The PCPs also store data about the ID and the speed of the OVs.

Fig. 10 shows an example of the speed-profile generation process for the green dotted path of Fig 8b. Note that this procedure is repeated for every single candidate path. The resulting set of PCPs for this driving scene is plotted in Fig. 10a. Green and red circles represent the PCPs after intersecting the path of interest with predicted positions of OV_1 and OV_2, respectively. Also, the blue dots are time-propagated positions of the OVs along ρ^i, computed from the

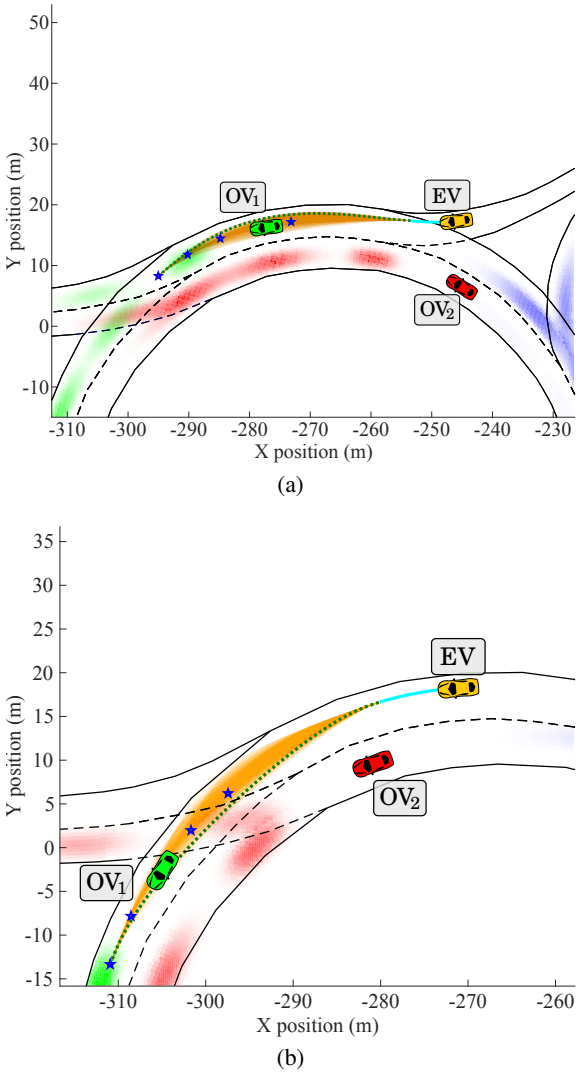


Fig. 8. Trajectory generation process in a roundabout-merging scenario. (a) Merging in the roundabout. (b) Driving on the roundabout.

green and red PCPs. This data represent how the positions on path of OVs evolve over time according to their predicted positions, and they are used to compute a speed profile that maintains a safe distance from them. Finally, the yellow circles show the position-in-path evolution of the EV after computing the speed profile for the driving scene.

The resulting speed profile for the path of interest is shown in Fig. 10b. The figure shows the current speed of the OVs (green and red dotted lines), a limit-speed profile computed using comfort restrictions (dotted yellow line), and the dynamic speed profile (thick yellow line) that uses the limit speed profile as the maximum limit, but also takes into account the PCPs in order to maintain a safe gap with other vehicles (see [35]). The figure shows that the EV reduces the speed at the beginning of the trajectory, and then it stabilizes around the speed of OV_2. This speed reduction indicates that the EV will maintain a safe gap with respect to OV_1 in case the latter decides to leave the roundabout from the inner lane.

Once the trajectory set $\mathcal{T}$ is computed, each candidate is evaluated using a set of Trajectory Performance Indicators

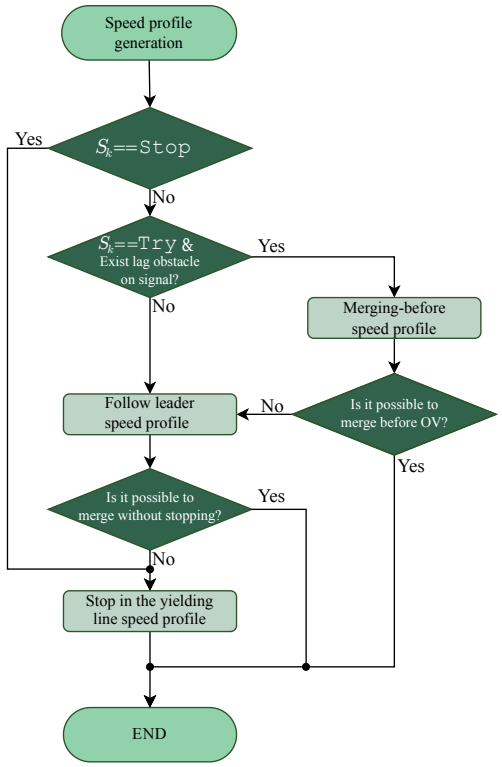


Fig. 9. Speed profile generation flowchart.

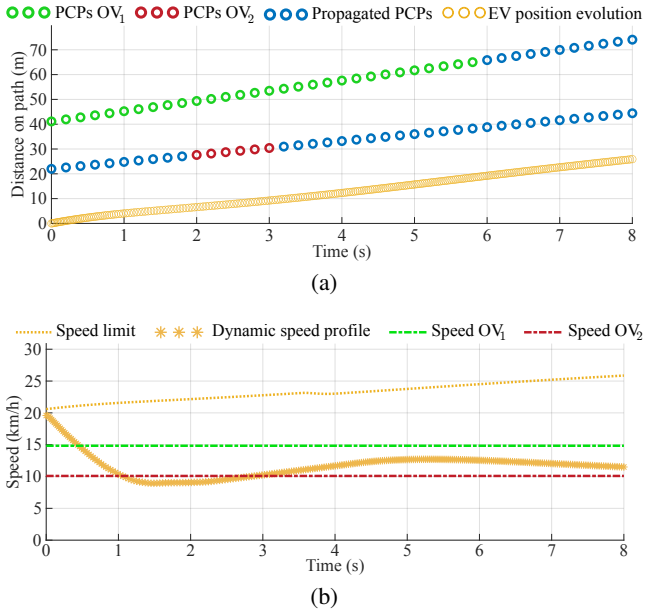


Fig. 10. Speed profile generation process on a roundabout-merging scenario. (a) PCPs for the selected candidate (b) Speed profile for the selected candidate.

(TPI), such as longitudinal and lateral accelerations, jerks, lane invasion, and distance to obstacles. These TPIs are combined into four Decision Variables (DV): longitudinal comfort, lateral comfort, safety and utility. The merit score (m) of each candidate is calculated using a modified weighted product to combine the four DV, which are evaluated on a non-linear weighting function in order to vary its influence on the final merit of the candidate. The merit function is defined as follows:

$$m^i = \sqrt[4]{\prod_{k=1}^{4} \mathscr{F}\left(DV_k^i\,,\,\omega_k\right)}, \quad i = 1, 2, ..., N_T. \tag{4}$$

where DV_k^i represents one of the four decision variables of a candidate τ^i, and $\omega_k \in [0, 1]$ is the weight of that DV. More details about the TPIs used to calculate each DV, or the definition of the non-linear function $\mathscr{F}\left(DV_k^j\,,\,\omega_k\right)$ can be found in [36].

Finally, the candidate with the highest merit is selected as the reference trajectory τ_{ref} for the EV to follow. This trajectory is sent to the trajectory tracking control module, which obtains the proper throttle, brake and steering wheel commands to track it. A study for different lateral control strategies to perform this task is presented in [37]. The selected trajectory is also used by the motion prediction module to establish the relations of the EV with the other agents present on the scene, as mentioned in Section III-A.

IV. Deployment

Deploying such a complex system in real-time requires the design of a custom simulation architecture and the development of specific parallelization techniques. The following subsections describe the overall software-in-the-loop architecture needed to run the dataset-based simulations in real-time and the GPU acceleration techniques developed to optimize IAMP.

A. Simulation Architecture

The system operates on a software-in-the-loop configuration connected through an LCM API [38]. The simulation environment is SCANeR Studio 1.9 [39], which offers a framework to model complete virtual scenarios including buildings, roads, signals, traffic and vehicle dynamic models. The simulation is orchestrated from a mission handler module that reads the information from the chosen public datasets and generates the virtual scene in the simulator. Then, the motion planning algorithm is able to interact with the simulator by reading the state of the vehicles and sending commands to the EV. The motion prediction module also receives this information and makes its computed predictions available through the network.

This closed-loop architecture runs with a real-time factor of 1 and uses the same software as in the real car, making the whole system transparent to the motion planning and motion prediction algorithms, which can then be directly applied in real-world scenarios.

B. GPU Acceleration

IAMP relies on the combination of a DBN to estimate intention with a Markov chains model to predict the motion of the surrounding vehicles. The performance of particle filters, used to infer the DBN, is linked to the number of particles and the accuracy of Markov chains increases with the dimension of their matrices. This makes both systems computationally sensitive to complexity scaling. As a result, in order to execute

IAMP in real-time it is necessary to employ acceleration techniques.

GPUs were the chosen hardware to accelerate the code due to their ability to execute instructions in parallel, which would be fitting to perform matrix multiplications and particle computations. Specifically, Nvidia GPUs and their programming language CUDA [40].

In IAMP this technology is applied in the computation of the particle filter and the Markov chains. For the particle filter, the acceleration consists in creating an aligned data structure that contains all the information from the scene in each particle, which enables the system to compute every particle in parallel while maintaining memory alignment. To compute the output of the filter a parallel reduction algorithm [41] is applied. As for the Markov chains motion model, matrix multiplication is accelerated by applying an informed combination of the matrices that reduces memory accesses by identifying the smaller relevant sections of the matrix, storing them in local memory and then performing the necessary subset of arithmetic operations. A more comprehensive exploration of the acceleration techniques implemented can be found in [42].

V. Experimental Results

This section presents a collection of behavioral results of the system acting together in three urban scenarios with high traffic: a crossroads, a T-junction and a double-lane roundabout. The first two are recreations of real driving scenes extracted from the *inD* dataset [43]. The third one was created ad-hoc due to the lack of data from the public datasets in multi-lane roundabouts scenarios. Subsection V-A presents the evaluation of the motion planning algorithm when handling the proposed scenarios. Subsection V-B depicts the motion prediction metrics for the same scenarios. Finally, a study of the computation times for the complete system is presented in subsection V-C.

The results are obtained by performing a comparison of IAMP with a baseline strategy in order to test how the motion prediction and the motion planning modules are interconnected. The baseline is a state of art learning-based approach called Enhanced Graph-based Interaction-aware Trajectory Prediction (GRIP++) [44], from now on referred to as Baseline Prediction Strategy (BPS). This model has been trained with the datasets used in this work. In this method, the interaction between close agents is represented with a graph, and it applies graph convolutional blocks to extract the features from the scene and subsequently uses an encoder-decoder long short-term memory (LSTM) model to make predictions. The output of the model is a point-based prediction that is augmented to consider the vehicle's size and to be used by the motion planner. This model is also used as a baseline in the recent IAMP publication [28] and has now been tested in a closed-loop approach.

Table II shows the configuration parameters for the acceleration and speed limits of the motion planner. It includes the comfort accelerations thresholds $\gamma_{\text{x,max,com}}$, $\gamma_{\text{x,min,com}}$ and $\gamma_{\text{y,lim,com}}$, the minimum acceleration threshold to avoid collisions $\gamma_{\text{x,min,saf}}$, the maximum driving speed v_{max}, the number of trajectory candidates N_T and the probability threshold for the PCP generation $\epsilon_{\text{PCP,min}}$. It is worth mentioning that all experiments were executed using the same configuration of the motion planner, which shows its low dependency on the traffic pattern.

TABLE II
Configuration Parameters for the motion planner.

Parameter	Value	Unit
$\gamma_{x,\text{max,com}}$	2.3	m/s^2
$\gamma_{x,\text{min,com}}$	-2.5	m/s^2
$\gamma_{x,\text{min,saf}}$	-7	m/s^2
$\gamma_{y,\text{lim,com}}$	± 2.4	m/s^2
v_{max}	55	km/h
N_T	600	-
$\epsilon_{\text{PCP,min}}$	0.25	-

TABLE III
Configuration Parameters for IAMP.

Parameter	Value	Unit
a_{max}	2	m/s^2
a_{min}	-3	m/s^2
a_{int}	1	m/s^2
v_{max}	72	km/h
Δt	0.2	s
N_p	30	-
P_n	20000	-
$t_f = \Delta t * N_p$	6	s

Table III shows the execution parameters for IAMP in the conducted experiments. Notice that a_{max} and a_{min} are the maximum and minimum possible accelerations for the prediction model, a_{int} is the interval between possible accelerations, v_{max} is the maximum allowed speed, Δt is the time step increment between predictions in the motion grid (can also be referred as sampling time), N_p is the number of predictions produced, P_n is the number of particles and t_f is the prediction horizon.

A. Motion planning with different motion predictions

In order to evaluate the performance of the motion planner, a total of four experiments were executed in each scenario, two of them using the BPS and two using IAMP. The experiments were compared using six Key Performance Indicators (KPI) computed from the resulting trajectory of the EV after completing the scenario. These KPIs' definitions are presented in Table IV.

The longitudinal and lateral acceleration KPIs evaluate the maximum value of these variables during the experiment; the longitudinal and lateral jerk KPIs are computed from the average value of the jerk. These four KPIs are used to measure the comfort of the experiment. The risk-to-collision indicator is computed using the Risk from Time to Closest Encounter (RTTCE) metric proposed in [45], by comparing the position of the EV with respect to the OVs all along the experiment. This metric was selected over more commonly used risk evaluation indexes (such as TTC) since it takes into account both spatial and temporal relations between vehicles, as well as driving uncertainties, which allows to consider potential

TABLE IV
PERFORMANCE INDICATORS.

Parameter	Formula
Longitudinal Acceleration	$\max(\lvert\gamma_x(s)\rvert)$
Longitudinal Jerk	$\overline{\lvert j_x(s)\rvert}$
Lateral Acceleration	$\max(\lvert\gamma_y(s)\rvert)$
Lateral Jerk	$\overline{\lvert j_y(s)\rvert}$
Risk-to-collision	$\overline{RTTCE} + \max(RTTCE)$
Travel time	t_N

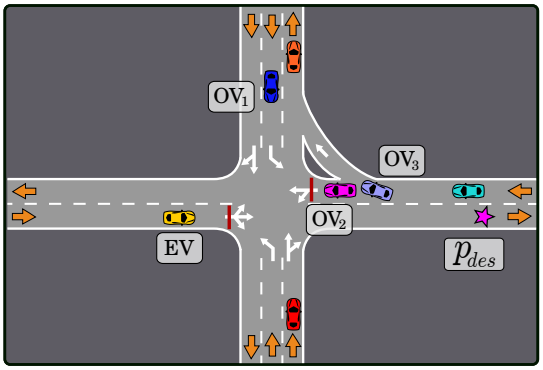


Fig. 11. Experimental setup for the crossroads scenario.

near-to-crash situations using a single coefficient. This KPI is the result of adding the maximum risk value and the average risk value after completing the experiment. Finally, the travel time quantifies the time it took for the EV to reach the goal position.

1) Crossroads scenario: This scenario consists of a four-armed intersection with a priority road located in Aachen, Germany. There are two center left-turn lanes and no regulated pedestrian crossings. Fig. 11 shows the driving scene selected to evaluate the motion planning in this scenario. In this scene, the EV starts at the left arm of the intersection and has to cross the priority lanes to get to the destination point (p_{des}), which is on the other side of the intersection, 52 m away from the initial position of the EV.

The resulting speed and acceleration profiles along the path for the experiments carried out in this scenario are shown in Fig. 12. It can be seen in Fig. 12a that both BPS experiments stopped in the intersection (at $s = 13$ m), while the IAMP experiments did not stop. The reason for this is the difference in the predictions of OV_3, which is going to perform a right turn without stopping. The predictions for the BPS occupy the ego lane, while the IAMP does not allow this, due to the presence of OV_2. Using IAMP, the EV keeps safely approaching and, once it infers that OV_2 will yield, it crosses the intersection. This divergence in the predictions is appreciated in Fig. 13, where the predictions of both strategies are shown at $t = 6.2$ s, when the EV is approaching the intersection. In the image, the resulting planned trajectory of the EV is illustrated by the continuous line, where its color represents the acceleration associated with the speed profile.

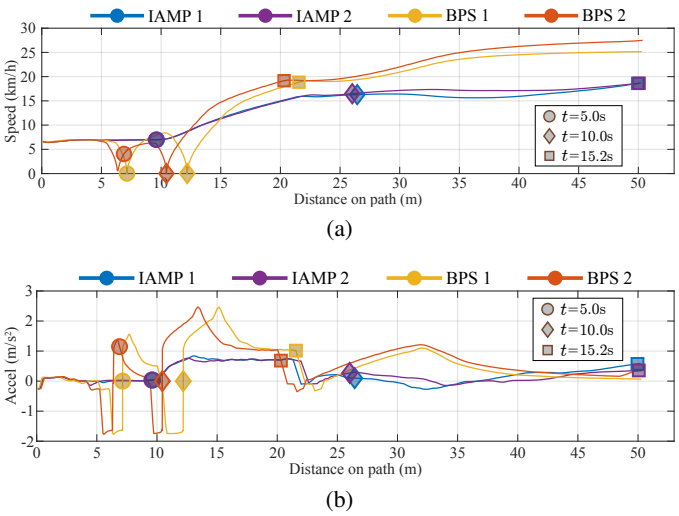


Fig. 12. Speed and acceleration profiles for the different experiments on the crossroads scenario. (a) Speed profile. (b) Longitudinal acceleration profile.

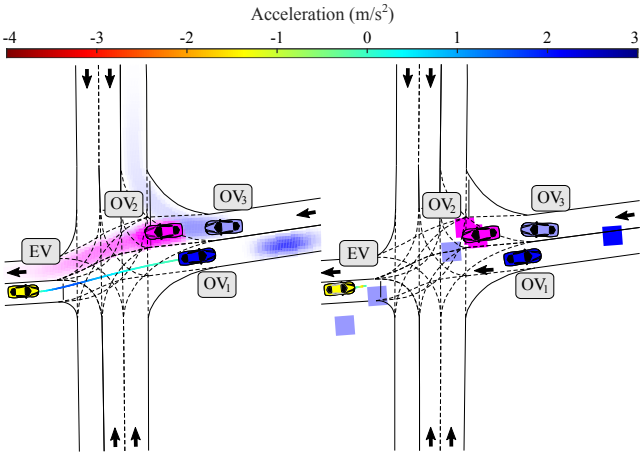


Fig. 13. Predictions for 2, 4 and 6 s for the crossroad scenario at $t = 6.1$ s. The color of the predictions corresponds to the vehicle to which they belong. (a) IAMP. (b) BPS.

As can be observed, the planned trajectory is shorter for the BPS experiment because it ends at the yielding line. This happens due to the already mentioned invasion of lane caused by OV_3.

It can be observed in Fig. 12b that the BPS-based longitudinal acceleration values were comprised between $\gamma_x(s) = [-1.8,\ 2.3\ m/s^2]$ during the experiments. In the case of the IAMP experiments, the acceleration variations were lower and smoother. Fig. 12 also highlights three relevant time instants $t = [5.0, 10.0, 15.2]$ s that depict how differently the experiments develop over time. In the case of IAMP-based experiments, at $t = 10$ s the EV has traveled more than 25 m and it is driving at 17 km/h, while in the BPS-based experiments, the EV is stopped in the yielding line at $t = 10$ s and has traveled less than 13 m. After crossing the intersection, the EV drives behind OV_1, which limits the speed for the IAMP experiments. This limitation does not occur in the BPS experiments because, when the EV crosses the intersection, the OV_1 is further ahead so the traveling speed after crossing the intersection is higher for the latter experiments.

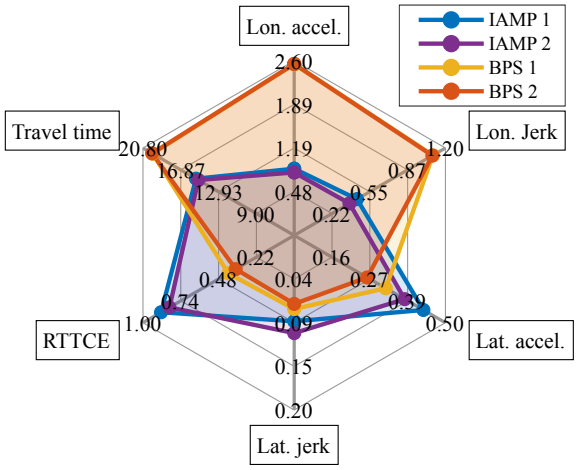


Fig. 14. KPIs for the different experiments on the crossroads scenario.

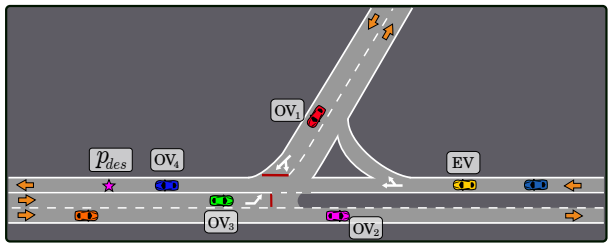


Fig. 15. Experimental setup for T-junction scenario.

Fig. 14 shows the KPIs of the four experiments on a radar plot. The definition of the KPIs requests a smaller-is-better (SiB) format. Hence, the closer to the center of the figure, the better the performance of the KPI. It can be seen that the longitudinal comfort-related performance of the IAMP experiments is notably better than the performance of the BPS experiments, mainly due to the braking/accelerating maneuver in the yielding line. In the case of the lateral-comfort KPIs, both configurations show similar performance, keeping bounded the maximum lateral acceleration to $\max(|\gamma_y(s)|) \leq 0.2\ m/s^2$. Regarding the RTTCE KPI, the IAMP experiments presented a higher risk value due to the crossing of the intersection without stopping, but always within the safety threshold. Finally, the IAMP experiments reached the destination point $\Delta t = 4.5\ s$ faster than the BPS experiments.

2) T-junction scenario: The second experiment was recorded in a T-junction located in Aachen, Germany. The main road has the right of way and there is a left turn lane into the side road. The EV is located on the main road and has to cross the intersection completely, as depicted in Fig. 15. The distance between the EV and p_{des} is $s = 150\ m$. There are two OVs that have a larger interaction with the EV during this experiment: OV_3 and OV_1. In the case of OV_3, it has to make a left turn to incorporate to the side road, crossing the ego-lane; as for OV_1, it incorporates from the side road to the main road, on the ego-lane.

Fig. 16 shows the speed and acceleration profiles for the T-junction scenario. The difference between the motion of the EV using IAMP and BPS was greater for this scenario

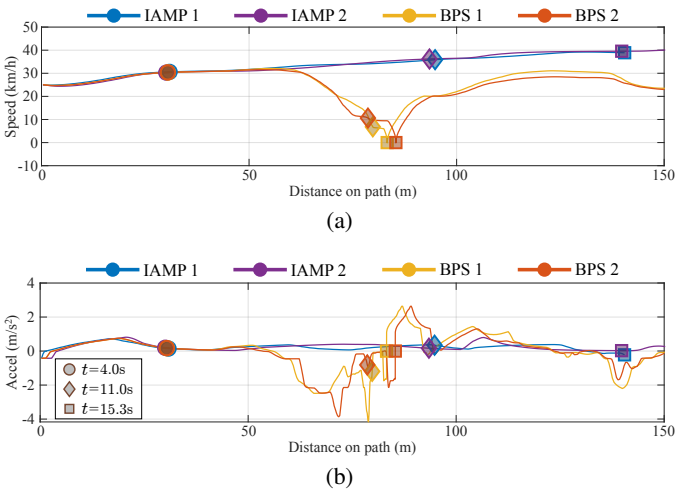


Fig. 16. Speed and acceleration profiles for the different experiments on the T-junction scenario. (a) Speed profile. (b) Longitudinal acceleration profile.

than for the crossroads scenario because, for the BPS-based experiments, the EV slowed down and stopped on the main road during $t \approx 2\ s$. Notice that the EV should not stop at all, since it is driving on the priority road. Note also that this kind of maneuver (stopping on a priority road) is not only inefficient but also dangerous as it may cause accidents with OVs driving behind the EV that does not expect this type of behavior. All experiments have very similar speed profiles for $s \leq 60\ m$, (which is the first $t \approx 7.2\ s$ of travel). At that moment, the predictions for OV_3 using BPS go through the ego-lane without taking into account the roads' priorities (see Fig. 17b), which causes a speed reduction of the EV (red portion of the EV's trajectory in the figure). On the other hand, the predictions of the IAMP strategy take into account the traffic rules and the predictions of OV_1 end at the stop line, as can be seen in Fig. 17a. As the EV keeps traveling, it keeps smoothly accelerating along the road for the IAMP-based experiments at $s = 70\ m$, while it reduces the speed at a rate of $\gamma_x \approx -4\ m/s^2$ for the BPS-based experiments. Finally, the EV reaches the destination point at $t \approx 15.3\ s$ for the fastest IAMP-based experiment, whereas for the BPS-based experiment, the EV is completely stopped in the middle of the road at $t \approx 15.3\ s$. Due to this speed reduction in the EV, both OV_3 and OV_1 merge into the intersection before the EV, which affects its speed during the last section of the travel, as OV_1 travels at a lower speed after it merges into the main road.

The performance indicators of the experiments for the T-junction scenario are shown in Fig. 18. Similarly to the crossroads scenario, the longitudinal comfort-related indicators are better for the IAMP-based experiments than for the BPS-based experiments; this is due to the accelerations involved in the stopping maneuver on the T-junction. The BPS-based experiments showed better performance on the lateral comfort-related indicators. This can be explained by the lower speeds of these experiments, which lead to low values of lateral acceleration. Nevertheless, the difference between the maximum lateral acceleration between the BPS and the IAMP-based experiments is only $\Delta\gamma_y \approx 0.29\ m/s^2$. Regarding the RTTCE,

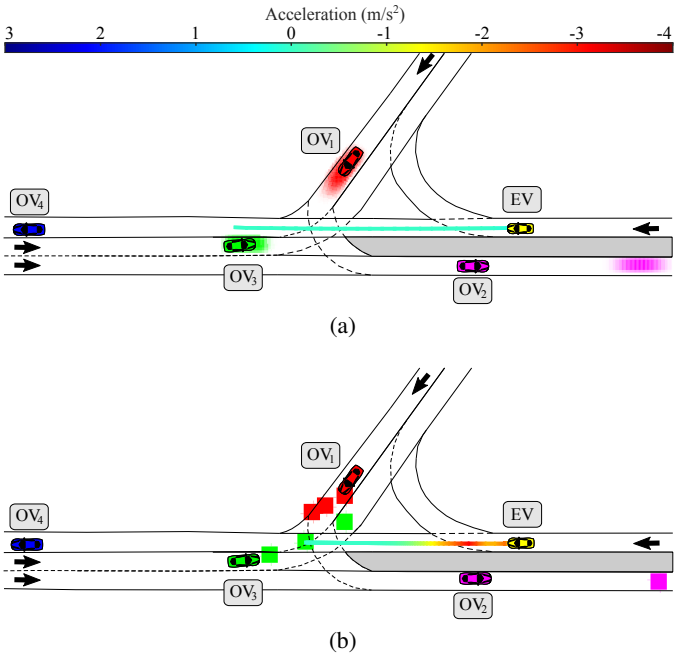


Fig. 17. Predictions for 2, 4 and 6 s for the T-junction scenario at $t = 7.8$ s. The colors of the predictions match the corresponding OV. (a) IAMP. (b) BPS.

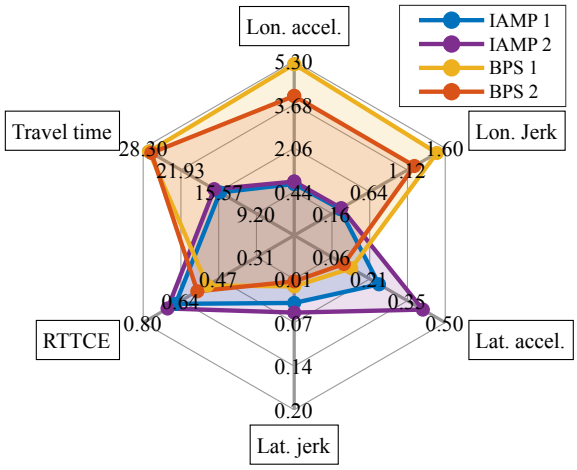


Fig. 18. KPIs for the different experiments on the T-junction scenario.

both strategies presented a similar performance, none of which indicates any dangerous situation (although in the BPS the EV comes to a full stop in a priority lane). The main difference in the performance of the experiments is presented in the travel time, which was $\bar{t} = 15.8$ s for IAMP-based experiments, and $\bar{t} = 27.1$ s for BPS-based experiments.

3) Roundabout scenario: The last scenario to test the motion planner is a double-lane roundabout located in Arganda del Rey, Spain. In this case, the EV must merge into a roundabout with three vehicles, and keep driving on the outer lane of the roundabout while another vehicle is driving on the inner lane. The layout of this scenario is depicted in Fig. 19.

As can be seen in Fig. 20, all experiments had a similar speed profile during the first $s \approx 55$ m of travel ($t \approx 13$ s). At that moment, the BPS predictions for OV_2 cross to the outer lane of the roundabout from the inner lane, which

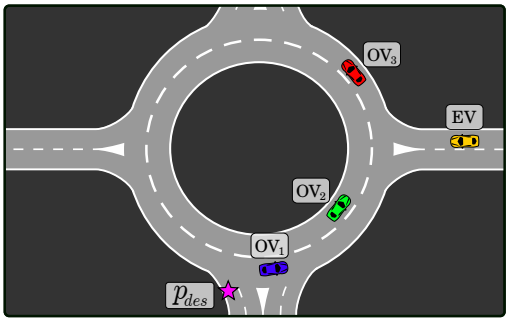
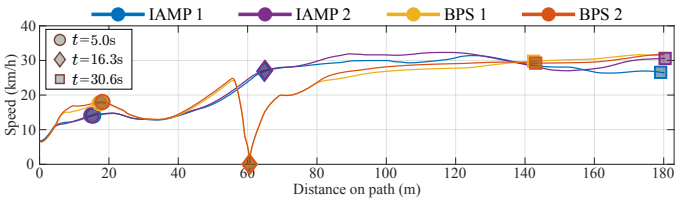
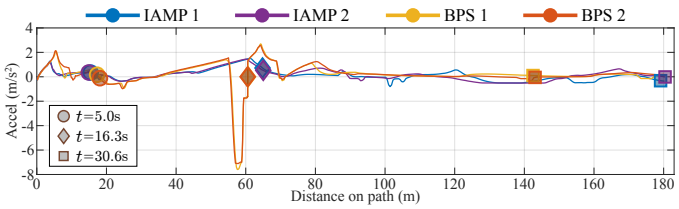


Fig. 19. Experimental setup for a double lane roundabout scenario.



(a)



(b)

Fig. 20. Speed and acceleration profiles for the different experiments on the roundabout scenario. (a) Speed profile. (b) Longitudinal acceleration profile.

causes the motion planner to brake at $\gamma_x \approx -7$ m/s^2 (maximum deceleration) to prevent the collision. This frame of the simulation is shown in Fig. 21b (notice the red portion on the EV's trajectory). Contrarily, the IAMP strategy takes into account the influence of the EV on the speed profile of OV_2 and, even though this OV can leave the roundabout from the inner lane, IAMP predicts that it has to brake to do so.

The predictions of the IAMP strategy for this moment are shown in Fig. 21a, where it can be seen that the predictions of the OV_2 do not occupy the outer lane of the roundabout, allowing the EV to keep its current speed. Notice that this traffic scene is similar to the one presented in Fig. 8b, but in this case the EV is ahead of the OV in the inner lane of the roundabout, which causes the predictions to be different, prioritizing the EV. The EV speed on the roundabout is limited by the motion planner to 30 km/h approximately so that the lateral acceleration meets the comfort requirements.

B. Motion prediction

In order to evaluate the precision of the predictions from both approaches, IAMP and the BPS, two metrics are used:

- ADE (Average Displacement Error): average L_2 distance between the ground truth positions and the weighted average position of the predictions.

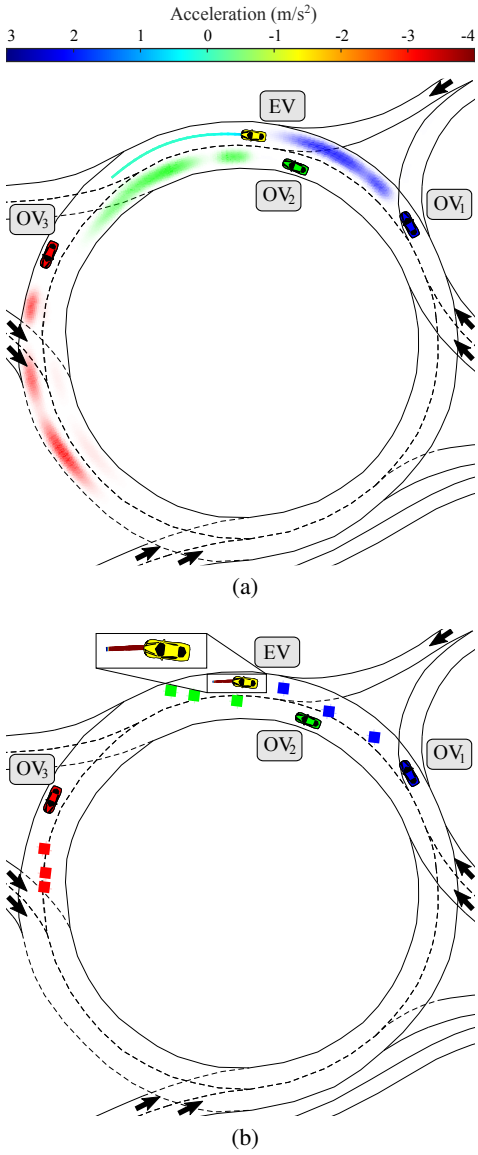


Fig. 21. Predictions for 2, 4 and 6 s for the roundabout scenario at $t = 15.5$ s. The colors of the predictions match the corresponding OV. (a) IAMP. (b) BPS.

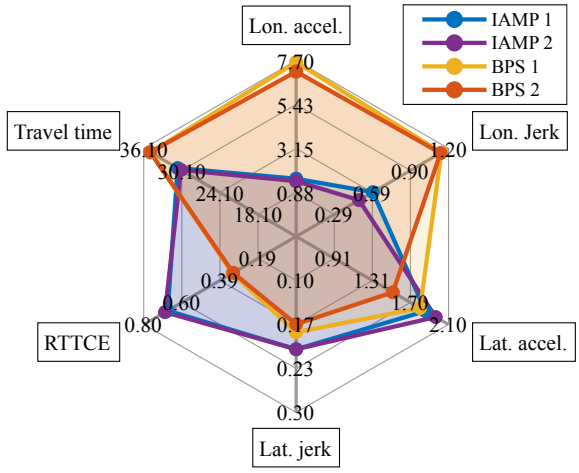


Fig. 22. KPIs for the different experiments on the roundabout scenario.

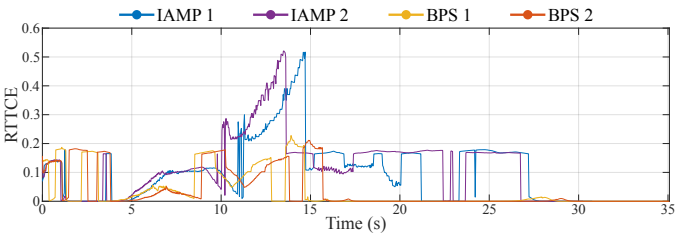


Fig. 23. RTTCE for the experiments on the roundabout scenario.

TABLE V
COMPARISON METRICS FOR IAMP AND THE BPS

Scenario	Approach	ADE 3s (m)	FDE 3s (m)	ADE 6s (m)	FDE 6s (m)
Crossroad	IAMP	0.86	1.77	2.90	8.63
	BPS	1.40	2.86	3.62	8.87
T-junction	IAMP	1.17	1.64	1.93	3.85
	BPS	4.50	6.10	6.50	10.87
Roundabout	IAMP	0.79	1.29	2.20	5.81
	BPS	1.90	3.87	5.26	13.69

TABLE VI
EXECUTION FREQUENCY OF THE SYSTEM ON THE DIFFERENT SCENARIOS

	Crossroads	T-junction	Roundabout
Motion Prediction	7.93 Hz	9.92 Hz	4.28 Hz
Motion Planning	5.00 Hz	5.00 Hz	5.00 Hz

- FDE (Final Displacement Error): L_2 distance between the last ground truth position and the weighted average position of the last prediction.

These metrics are evaluated considering two time-horizons: 3 s and 6 s. Given that the output from IAMP is multi-modal, the minimal value from the metrics is utilized for each vehicle. For the BPS, no selection is necessary, since it produces only one output per vehicle.

The results of the evaluation in the three use cases are presented in Table V. From the two experiments of each strategy, only the best one is shown. Notice that IAMP outperforms its rival in all metrics since it can better handle the interaction between vehicles and can better take into consideration the driving scenario.

C. Computation Time

In order to show the real-world viability of the proposed work, the computation time of the different parts of the algorithm is studied here. The system run on three different computers that hosted SCANeR, motion planning and motion prediction respectively. The motion planning system was run on a PC with an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz with 8 GB RAM DDR3 @ 1600MHz. The motion prediction system run on a laptop with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 16 GB RAM DDR4 @2667MHz and a Nvidia Quadro P1000 Mobile with 4 GB VRAM.

The time requirements for the motion planning and motion prediction modules are imposed by their relation to each other and by the demands of the other system modules. Their internal time requirements are to produce data at similar rates due to the fact that they are both producers and consumers

of each other's information. The external requirements for motion planning come from the control module which requires updated trajectories at a steady rate that are also reactive to environment and obstacle changes. The motion prediction module has no external requirements outside of $IA(MP)^2$, as motion planning is the sole consumer of its predictions. Motion predictions are consumed by the motion planning system using a Last In First Out (LIFO) strategy. The motion prediction message includes the position where the ego-vehicle was in the moment of the prediction, which is used by the motion planning algorithm to translate and rotate the predictions to the current moment of planning.

Table VI shows the average execution frequency of the two systems. Motion planning keeps a steady rate of trajectories to feed the control system on time. Motion prediction, using a prediction horizon of 6s as can be seen in Table III, maintains a minimum rate of 4Hz, which is considered a good tradeoff between update rate and horizon length to meet the motion planning requirements.

VI. Conclusion

Motion prediction and path planning are two core tasks of autonomous driving systems. In this paper, the integration of interaction-aware motion predictions with a state-of-the-art motion planner is proven to achieve more human-like levels of planning while dealing with complex scenarios. The proposed framework was evaluated in three urban scenarios with high levels of traffic, a crossroad and a T-junction from the public dataset *inD* and a multi-lane roundabout from a proprietary dataset. The results showed that the use of informed predictions avoids overly cautious behaviors that incur sudden breaking and lower average speeds which impact travel time and overall comfort for the occupants. The system was able to maintain minimum frequencies of 4Hz in prediction and 2Hz in planning, which validates the real-time capabilities of the algorithms. The prediction algorithm was compared to a state-of-the-art neural network, and it demonstrated superior performance in the prediction metrics. Future work will focus on integrating motion predictions that are able to deal with sensor occlusions and traffic violations into motion planning.

Acknowledgments

This work has been partially funded by MCIN/AEI/10.13039/501100011033 and by ERDF A way of making Europe with the projects DISCERN (PID2021-125850OB-I00) and NEWCONTROL (PCI2019-103791), and by the European Commission and ECSEL Joint Undertaking through the Project NEWCONTROL (826653)

References

[1] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.

[2] S. Arbabi, D. Tavernini, S. Fallah, and R. Bowden, "Planning for autonomous driving via interaction-aware probabilistic action policies," *IEEE Access*, vol. 10, pp. 81 699–81 712, 2022.

[3] 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," *Proceedings IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[4] A. Gray, Y. Gao, J. K. Hedrick, and F. Borrelli, "Robust predictive control for semi-autonomous vehicles with an uncertain driver model," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 208–213.

[5] J. F. Medina-Lee, V. Jiménez, J. Godoy, and J. Villagra, "Maneuver planner for automated vehicles on urban scenarios," in *Proceedings IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2022.

[6] K. Brown, K. Driggs-Campbell, and M. J. Kochenderfer, "A taxonomy and review of algorithms for modeling and predicting human driver behavior," *arXiv preprint arXiv:2006.08832*, 2020.

[7] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, "Lookout: Diverse multi-future prediction and planning for self-driving," in *Proceedings IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 16 107–16 116.

[8] J. Villagra and A. Artuñedo, "Behavior planning," in *Decision-Making Techniques for Autonomous Vehicles*. Elsevier, 2023, pp. 39–59.

[9] J. Villagra, M. Clavijo, A. Díaz-Álvarez, and V. Trentin, "Motion prediction and risk assessment," in *Decision-Making Techniques for Autonomous Vehicles*. Elsevier, 2023, pp. 61–101.

[10] Y. Jeong and K. Yi, "Target vehicle motion prediction-based motion planning framework for autonomous driving in uncontrolled intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 168–177, 2021.

[11] L. Zhang, W. Xiao, Z. Zhang, and D. Meng, "Surrounding vehicles motion prediction for risk assessment and motion planning of autonomous vehicle in highway scenarios," *IEEE Access*, vol. 8, pp. 209 356–209 376, 2020.

[12] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proceedings IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8652–8661.

[13] H. Wang, P. Cai, Y. Sun, L. Wang, and M. Liu, "Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation," in *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 731–13 737.

[14] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene transformer: A unified architecture for predicting future trajectories of multiple agents," in *Proceedings International Conference on Learning Representations*, 2021.

[15] T. Zhang, M. Fu, T. Liu, and W. Song, "Spatio-temporal decision-making and trajectory planning framework with flexible constraints in closed-loop dynamic traffic," *IET Intelligent Transport Systems*, 2022.

[16] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *Proceedings European Conference on Computer Vision*, 2020, pp. 624–641.

[17] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[18] E. Debada, A. Ung, and D. Gillet, "Occlusion-aware motion planning at roundabouts," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 276–287, 2020.

[19] V. Trentin, C. Ma, J. Villagra, and Z. Al-Ars, "Learning-enabled multi-modal motion prediction in urban environments," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–7.

[20] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Interaction-aware probabilistic behavior prediction in urban environments," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3999–4006.

[21] Y. Wang, C. Wang, W. Zhao, and C. Xu, "Decision-making and planning method for autonomous vehicles based on motivation and risk assessment," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 107–120, 2021.

[22] Z. Huang, H. Liu, J. Wu, and C. Lv, "Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving," *arXiv preprint arXiv:2207.10422*, 2022.

[23] Y. Chen, B. Ivanovic, and M. Pavone, "Scept: Scene-consistent, policy-based trajectory predictions for planning," in *Proceedings IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 103–17 112.

[24] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[25] J. Sun, C. Yuan, S. Sun, Z. Liu, T. Goh, A. Wong, K. P. Tee, and M. H. Ang Jr, "Get-dipp: Graph-embedded transformer for differentiable

integrated prediction and planning," *arXiv preprint arXiv:2211.06031*, 2022.
[26] X. Tang, K. Yang, H. Wang, J. Wu, Y. Qin, W. Yu, and D. Cao, "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 4, pp. 849–862, 2022.
[27] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proceedings IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1672–1679.
[28] V. Trentin, A. Artuñedo, J. Godoy, and J. Villagra, "Multi-modal interaction-aware motion prediction at unsignalized intersections," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 5, pp. 3349–3365, 2023.
[29] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Intention-Aware Risk Estimation for General Traffic Situations, and Application to Intersection Safety," INRIA, Research Report RR-8379, Oct. 2013.
[30] J. Godoy, V. Jiménez, A. Artuñedo, and J. Villagra, "A grid-based framework for collective perception in autonomous vehicles," *Sensors*, vol. 21, no. 3, 2021.
[31] V. Trentin, A. Artuñedo, J. Godoy, and J. Villagra, "Interaction-aware intention estimation at roundabouts," *IEEE Access*, vol. 9, pp. 123 088–123 102, 2021.
[32] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.
[33] H. Li, G. Yu, B. Zhou, P. Chen, Y. Liao, and D. Li, "Semantic-level maneuver sampling and trajectory planning for on-road autonomous driving in dynamic scenarios," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1122–1134, 2021.
[34] T. Toledo, C. F. Choudhury, and M. E. Ben-Akiva, "Lane-changing model with explicit target lane choice," *Transportation Research Record*, no. 1934, pp. 157–165, 2005.
[35] J. F. Medina-Lee, J. Godoy, A. Artuñedo, and J. Villagra, "Speed profile generation strategy for efficient merging of automated vehicles on roundabouts with realistic traffic," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2448–2462, 2023.
[36] J. Medina-Lee, A. Artuñedo, J. Godoy, and J. Villagra, "Merit-based motion planning for autonomous vehicles in urban scenarios," *Sensors*, vol. 21, p. 3755, May 2021.
[37] M. Moreno-Gonzalez, A. Artuñedo, J. Villagra, C. Join, and M. Fliess, "Speed-adaptive model-free lateral control for automated cars," in *Proceedings IFAC Symposium on System Structure and Control SSSC*, 2022.
[38] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4057–4062.
[39] AVSimulation, "Scaner studio 1.9," https://www.avsimulation.com/scaner-studio/, Last accessed on 2022-09-21, 2022.
[40] NVIDIA, "Cuda api documentation," https://docs.nvidia.com/cuda/cuda-runtime-api/index.html, Last accessed on 2022-06-30, 2022.
[41] A. Cano, A. Zafra, and S. Ventura, "Parallel evaluation of pittsburgh rule-based classifiers on gpus," *Neurocomputing*, vol. 126, pp. 45–57, 2014.
[42] J. L. Hortelano, V. Trentin, A. Artuñedo, and J. Villagra, "Gpu-accelerated interaction-aware motion prediction," *Electronics*, 2023.
[43] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934.
[44] X. Li, X. Ying, and M. C. Chuah, "Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving," *arXiv preprint arXiv:1907.07792*, 2020.
[45] J. Eggert and T. Puphal, "Continuous risk measures for adas and ad," in *Proceedings Future Active Safety Technology Symposium*, 2017, pp. 1–8.

VII. Biography Section



Juan Medina-Lee graduated as an Electronic Engineer from the Universidad del Quindío, Colombia in 2012. He received his Master's degree in Engineering from the Universidad del Valle in 2016, and his Ph.D. degree in automation and robotics from the Universidad Politécnica de Madrid, Spain, in 2023. His PhD thesis was awarded the distinction "Cum Laude". His research areas of interest are decision-making of autonomous vehicles, embedded systems, and robotics. He worked as a professor at the University of Quindío, teaching subjects in digital systems area and robotics from 2012 until 2018. He is currently working as a researcher in the Centre for Automation and Robotics (CSIC-UPM) as a member of AUTOPIA research group.



Vinicius Trentin was born in São Carlos - SC, Brazil, in 1993. He received the bachelor's degree in control and automation engineering from the Federal University of Santa Maria, Brazil in 2018. During his graduation, he was part of an exchange program in Germany, where he studied at the TU Dresden and, later, made an internship at the Fraunhofer IIS EAS, also in Dresden. Since 2019 he joined AUTOPIA research group as a Pre-Doctoral researcher for the Centre for Automation and Robotics (CSIC-UPM) and studying a PhD degree on robotics at the Universidad Politécnica de Madrid. His research interests include autonomous driving, interaction awareness, and motion prediction.



Juan Luis Hortelano graduated as a Telecommunications Engineer at Universidad Politécnica de Madrid in 2015. In 2017 he received his Masters Degree in Advanced Robotic Systems from University of Southern Denmark with a specialization in mobile robotics. From 2015 to 2020 he lived and worked in Denmark where he was technical lead in mobile robotics at Blue Ocean Robotics and developed an antibodies testing robot for the Danish government while working at Novo Nordisk. Currently, he is a CUDA Researcher and a PhD candidate with research activities related to perception, computer vision and drivable area estimation.



Antonio Artuñedo received the B.Sc. degree in electrical engineering from the University of Castilla–La Mancha, Spain, in 2011, the MSc degree in industrial engineering from the University Charles III of Madrid in 2014, and the PhD degree in automation and robotics from the Technical University of Madrid (UPM), Spain, in 2019. His PhD thesis, awarded with the distinction "Cum Laude" and the International Mention, won multiples prizes including the Best PhD Thesis on Intelligent Transportation Systems 2020 by the Spanish Chapter of the IEEE-ITS Society. He is currently a Post-Doctoral Researcher in the Centre for Automation and Robotics (CSIC-UPM), AUTOPIA research group, Madrid, Spain, which he joined in 2013. He has been working on both national and European research projects in the scope of autonomous vehicles. He has published and peer-reviewed multiple journal and conference papers focused on this field. His research interests include system modeling and simulation, intelligent control, motion planning, and decision-making systems.



Jorge Godoy received the degree in electronics engineering from Universidad Simón Bolívar in 2008, and the M.E. and Ph.D. degrees in automation and robotics from the Universidad Politécnica de Madrid in 2011 and 2013, respectively. From 2013 to 2017 he was the Technical Coordinator of the AUTOPIA research group, funded by research contracts from National and European research projects. In 2017 he joined the Intelligent Control research group at the Universidad Politécnica de Madrid under the program Juan de la Cierva for Post-Doctoral Research. In 2022, he rejoins AUTOPIA as permanent staff of the Centre for Automation and Robotics (CSIC-UPM). He has participated in more than 15 projects in the scope of connected and automated vehicles, having published more than 60 articles in international journals and conferences. His research interest include decision-making, path planning, control and interaction of connected and automated vehicles.



Jorge Villagrá holds a degree in Industrial Engineering from the Polytechnic University of Madrid and a PhD in Real-Time Computing, Robotics and Automatic Control from the École des Mines de Paris (France) since 2006. He leads the AUTOPIA research group at CSIC since October 2016. He has participated in more than 40 R&D projects of which he is or has been principal investigator in half of them. He is a regular speaker in various specialized forums and has published more than 100 articles in international journals and conferences on decision-making, world modeling, prediction, planning and control of connected and automated vehicles.