Learning-Based Stochastic Model Predictive Control for Autonomous Driving at Uncontrolled Intersections

Surya Soman, Mario Zanon, *Senior Member, IEEE*, and Alberto Bemporad, *Fellow, IEEE*

***Abstract*— Autonomous driving in urban environments requires safe control policies that account for the non-determinism of moving obstacles, such as the position other vehicles will take while crossing an uncontrolled intersection. We address this problem by proposing a stochastic model predictive control (MPC) approach with robust collision avoidance constraints to guarantee safety. By adopting a stochastic formulation, the quality of closed-loop tracking is increased by avoiding giving excessive importance to future obstacle configurations that are unlikely to occur. We compute the probabilities associated with different obstacle trajectories by learning a classifier on a realistic dataset generated by the microscopic traffic simulator SUMO and show the benefits of the proposed stochastic MPC formulation on a simulated realistic intersection.**

***Index Terms*— Autonomous vehicles, model predictive control, scenario trees, stochastic model predictive control, supervised learning, classification methods, decision trees.**

I. Introduction

AUTONOMOUS driving has attracted massive interest both from industry and academia in the last years due to the enormous potential benefits it could introduce in terms of safety, energy optimization, and increased infrastructure efficiency. Unfortunately, achieving fully automated reliable driving requires significant further research efforts [1]. One of the main challenges consists of dealing with uncertain moving obstacles, such as other vehicles, pedestrians, etc. Various types of sensors are nowadays able to provide reliable information about the *current* obstacle positions, see, e.g., [2]. However, in order to drive safely and effectively, the control algorithm needs to take *future* obstacle positions into account. This poses two challenges: how to obtain such information; and how to exploit it in order to issue safe control commands.

Several approaches have been proposed for modeling pedestrians and surrounding vehicles; see, e.g., the overview in [3]. In particular, interacting multiple-model (IMM) filters that predict the intention and future states of surrounding vehicles were suggested in [4] and [5]. A similar approach was taken in [6], where, in addition, hidden Markov models (HMMs) were used to recognize vehicle maneuvers and probabilistic trajectories generated with the help of variational Gaussian mixture models. A multimodal hierarchical Inverse Reinforcement Learning (IRL) approach was proposed instead in [7] to learn joint driving pattern-intention-motion models and use them to probabilistically predict continuous motions. A simpler but effective and computationally inexpensive approach that retains the ability to predict multimodal distributions was proposed in [8]. In [9], the authors proposed a general semantic-based intention and motion prediction based on deep neural networks; similar approaches were also proposed in [10] and [11] to predict lane change maneuvers. A human-like decision model for unsignalized intersections was suggested in [12] by an intention-aware prediction of other vehicles via convolutional neural networks with multiple object tracking combined with a Kalman filter. In [13], recurrent neural networks were used in long short-term memory (LSTM) form to predict the future driving lane of the vehicle. In [14] LSTMs were also used to jointly predict turning intentions and the corresponding obstacle trajectories. A hybrid approach using a neural classifier for maneuver classification and an LTSM memory for trajectory prediction was analyzed in [15]. For lane-change maneuver prediction, a combination of support vector machines and neural networks was considered in [16], while random forests and conditional random fields method were suggested in [17] for a T-intersection. To address uncontrolled intersections, a learning-based approach has been applied to both intention prediction and control in [18].

Assuming that a model estimating the future positions of the surrounding obstacles is available, this can be naturally exploited for planning the motion of the ego vehicle by using model predictive control (MPC) techniques [19], [20], as collision avoidance can then be formulated as an explicit constraint. Indeed, MPC has gained considerable attention in the last years in automotive control, especially for its ability to handle constraints on system variables, not only in academic research but also in industrial practice; see, e.g., [21] for a documented use of MPC in high-volume production. Regarding motion planning, MPC enables tracking reference paths at desired speeds while ensuring collision avoidance, thanks to the introduction of explicit constraints that keep the predicted distance between the ego vehicle and obstacles above a prescribed safety margin; see, e.g., [22], in which an MPC formulation was used for path planning in a dynamic environment by modeling the surrounding obstacle vehicles as polygons, and [23] for an MPC formulation using

Received 13 July 2023; revised 24 May 2024; accepted 16 November 2024. The Associate Editor for this article was K. Gao. *(Corresponding author: Mario Zanon.)*

The authors are with the IMT School for Advanced Studies Lucca, 55100 Lucca, Italy (e-mail: mario.zanon@imtlucca.it).

Digital Object Identifier 10.1109/TITS.2024.3510041

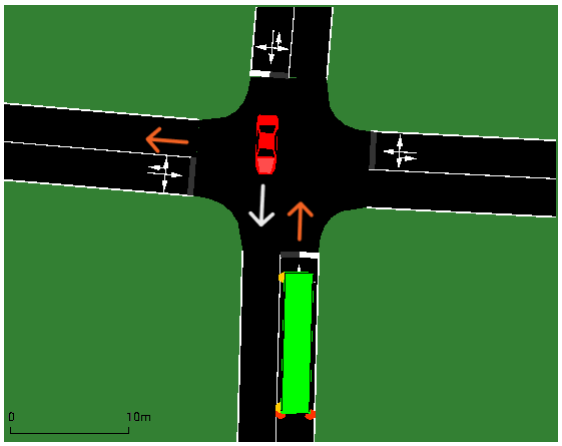


Fig. 1. Uncontrolled intersection example: the ego vehicle (red) drives straight and the obstacle vehicle (green) takes a left turn.

spatial-based models. More recently, a distributed MPC scheme was used in [24] to control vehicle platoons in urban road networks.

To handle uncertainty in the prediction, *stochastic* MPC formulations [25] can be introduced, in particular, scenario-based approaches [26]; see, e.g., [27], [28], and [29] for applications of stochastic MPC in automotive control. SMPC controllers based on imposing chance constraints for collision avoidance autonomous driving can be found in [30] and [31]. In [32], the authors proposed the use of stochastic MPC based on Gaussian mixture models to get a multimodal prediction of the trajectories of the surrounding vehicles. MPC approaches tailored to collision avoidance with pedestrians were proposed in [33] and [34], while a generic MPC framework providing rigorous safety guarantees in uncertain environments was analyzed in [35] and [36].

Alternative approaches for safe path planning were proposed in [37] and [38]. A stochastic scenario-based MPC approach was adopted in [39] by relying on partially observable Markov decision process models. More recently, [40] employed what the authors called *branch MPC*, which uses ideas similar to [34] and explicitly models human decision-making to predict the obstacle vehicle intentions.

A. Contribution

This paper proposes a scenario-based stochastic MPC formulation with robust collision-avoidance constraints for commanding the longitudinal acceleration and steering rate of the ego vehicle, focused on the case of uncontrolled intersections. We estimate the probabilities associated with different future scenarios, which are crucial for closed-loop performance, by means of a classifier that predicts the intention of incoming vehicles to drive straight ahead, turn left, or turn right. The classifier is trained on a realistic dataset generated by the microscopic traffic simulator SUMO (Simulation of Urban Mobility) [41]. We adopt bagged decision trees for classification due to their ability to associate a probability to each target class (straight/left/right) and their superior performance with respect to other supervised learning approaches that we have considered. The effectiveness of the overall approach is analyzed in realistic simulation settings.

The paper is organized as follows. In Section II, we formulate the proposed scenario-based stochastic MPC problem. Section III presents the classifier based on bagged decision trees used to predict the other vehicles' behavior, how it is trained, how it is exploited to formulate the MPC problem, and compares it to alternative classification methods. Closed-loop simulation results are presented in Section IV, and conclusions are drawn in Section V.

II. Problem Formulation

In order to define a prediction model for MPC, we consider for simplicity the following classical kinematic bicycle model [42] of the ego vehicle

$$\dot{x} = v\cos(\theta+\delta) \tag{1a}$$
$$\dot{y} = v\sin(\theta+\delta) \tag{1b}$$
$$\dot{\theta} = \frac{v}{l}\sin(\delta) \tag{1c}$$
$$\dot{v} = a \tag{1d}$$
$$\dot{\delta} = \omega \tag{1e}$$

in which (x, y) are the fixed absolute frame coordinates of the center of the front axle of the ego vehicle; θ denotes the vehicle orientation with respect to the direction of the x axis; v is the longitudinal speed; and δ is the steering angle. In the sequel, we denote by $X = [x\ y\ \theta\ v\ \delta]'$ the state vector of the model and by $U = [a\ \omega]'$ the input vector, where a is the longitudinal acceleration and ω the steering rate. Model (1) is discretized in time by employing an explicit Runge-Kutta 4 method with sampling interval T_s, obtaining the following discrete-time nonlinear model

$$X_{t+1} = f(X_t, U_t) \tag{2}$$

where t denotes the sample step.

Our goal is to design a controller that can make the state X and input U of the ego-vehicle track given reference trajectories X^r, U^r, respectively, while avoiding colliding with obstacles. In this paper, we will only consider the case of a single obstacle characterized by its coordinates (x^o, y^o), although the methodology presented here can straightforwardly be extended to the presence of multiple obstacles. In the following paragraphs, we introduce three different MPC formulations of increasing computational complexity that we will compare in the numerical simulations reported in Section IV.

A. Prescient Model Predictive Control

Consider the following (ideal) **Prescient MPC** formulation

$$\min_{\substack{\{X_{k|t},\,U_{k|t}\}\\ k=0,\ldots,N}} \sum_{k=0}^{N} \|X_{k|t} - X^r_{k|t}\|^2_Q + \|U_{k|t} - U^r_{k|t}\|^2_R \tag{3a}$$
$$\text{s.t.} \quad X_{k+1|t} = f(X_{k|t}, U_{k|t}) \tag{3b}$$
$$X_{0|t} = X_t \tag{3c}$$

$$X_{k|t} \in \mathcal{X} \tag{3d}$$
$$U_{k|t} \in \mathcal{U} \tag{3e}$$
$$A_{k|t}(X_{k|t} - X^{\mathrm{r}}_{k|t}) \leq B_{k|t} \tag{3f}$$
$$(x_{k|t} - x^{\mathrm{o}}_{k|t})^2 + (y_{k|t} - y^{\mathrm{o}}_{k|t})^2 \geq d^2_{\min}. \tag{3g}$$

In (3), we assume that at the current time t a measurement or an estimate of the state X_t is available, $(x^{\mathrm{o}}_{k|t}, y^{\mathrm{o}}_{k|t})$ denotes the known obstacle trajectory (we will relax this assumption later). The given reference samples are $(X^{\mathrm{r}}_{k|t}, U^{\mathrm{r}}_{k|t})$, $k = t, t+1, \ldots, t+N$, and $\|x\|^2_Q = x'Qx$, where matrix $Q \in \mathbb{R}^{5\times 5}$ is symmetric and positive semi-definite, while matrix $R \in \mathbb{R}^{2\times 2}$ is symmetric and positive definite. The sets $\mathcal{X}$, $\mathcal{U}$ denote, respectively, the sets of feasible state and input vectors. The linear constraints in (3f) are used to model constraints such as road bounds, where

$$A_{k|t} = [I_2 \ \ -I_2]' \begin{bmatrix} \cos(\theta^{\mathrm{r}}_{k|t}) & \sin(\theta^{\mathrm{r}}_{k|t}) \\ -\sin(\theta^{\mathrm{r}}_{k|t}) & \cos(\theta^{\mathrm{r}}_{k|t}) \end{bmatrix}$$

with $\theta^{\mathrm{r}}_{k|t}$ is the given reference orientation of ego vehicle with respect to the global axis for $k = t, t+1, \ldots, t+N$ and I_2 is the identity matrix of dimension 2. The right hand side is given by $B_{k|t} = \left[\frac{L_{\mathrm{r}}}{2} \ \frac{W_{\mathrm{r}}}{2} \ \frac{L_{\mathrm{r}}}{2} \ \frac{W_{\mathrm{r}}}{2}\right]'$, where L_{r} and W_{r} are lengths that depend on the road profile and are chosen small enough to guarantee that the ego vehicle remains within the road. The constraints in (3g) enforce collision avoidance in that the Euclidean distance between the position of the ego vehicle and that of the obstacle is greater than a given safety distance $d_{\min}$. Note that $U_{N|t}$ is a redundant variable that could be eliminated from (3).

Problem (3) is solved at each time step t. The first component $U^{\mathrm{PMPC}}_{0|t}$ of the optimal solution is commanded as the input U_t to the vehicle. According to a receding horizon strategy, the remaining components $U^{\mathrm{PMPC}}_{1|t}, \ldots, U^{\mathrm{PMPC}}_{N|t}$ are discarded, and the problem is solved again at time $t+1$, and so on.

The control law (3) cannot be applied in practice, as it requires knowing the future obstacle positions $(x^{\mathrm{o}}_{k|t}, y^{\mathrm{o}}_{k|t})$, which is an unrealistic assumption in most cases of interest. Nonetheless, we will consider (3) as a baseline policy for comparison with more realistic control strategies which we detail next.

B. Robust Model Predictive Control

To take into account the uncertainty associated with future obstacle positions, we consider next the following **Robust MPC** formulation

$$\min_{\substack{\{X_{k|t},\, U_{k|t}\} \\ k=0,\ldots,N}} \ \sum_{k=0}^{N} \|X_{k|t} - X^r_{k|t}\|^2_Q + \|U_{k|t} - U^r_{k|t}\|^2_R \tag{4a}$$
$$\text{s.t.} \quad X_{k+1|t} = f(X_{k|t}, U_{k|t}) \tag{4b}$$
$$X_{0|t} = X_t \tag{4c}$$
$$X_{k|t} \in \mathcal{X} \tag{4d}$$
$$U_{k|t} \in \mathcal{U} \tag{4e}$$
$$A_{k|t}(X_{k|t} - X^{\mathrm{r}}_{k|t}) \leq B_{k|t} \tag{4f}$$

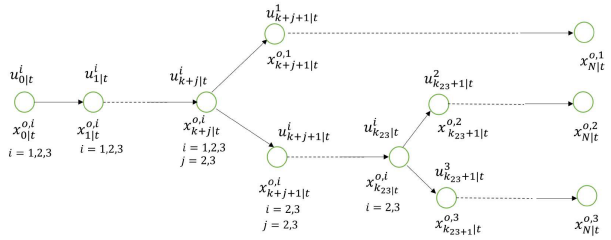


Fig. 2. Scenario tree used in the SMPC formulation (5).

$$(x_{k|t} - x^{o_i}_{k|t})^2 + (y_{k|t} - y^{o_i}_{k|t})^2 \geq d^2_{\min}$$
$$\forall i = 1, \ldots, m_t \tag{4g}$$

where $(x^{o_i}_{k|t}, y^{o_i}_{k|t})$, $i = 1, \ldots, m_t$, are meaningful corner-case scenarios used to account for the set of all possible trajectories over the future horizon taken by the obstacle. Formulation (4) is robust in that collisions are avoided in all possible scenarios; in fact, the only difference between (3) and (4) is in the obstacle avoidance constraint (4g). In (4), m_t denotes the number of corner cases considered, which in general depends on t.

C. Stochastic Model Predictive Control

The main drawback of (4) is its possible conservativeness, due to the fact the same predicted input sequence $\{U_{k|t}\}$ is used to counteract all possible future realizations $\{(x^{o_i}_{k|t}, y^{o_i}_{k|t})\}$, and all such corner cases are considered equally likely. To attenuate such a potentially conservative behavior without sacrificing robustness with respect to obstacle avoidance constraints, let us associate a probability ω^i_t to each scenario i, $\sum_{i=0}^{m_t} \omega^i_t = 1$, $\omega^i_t \geq 0$, $\forall t \geq 0$. We assume that the scenarios are organized as a scenario tree; see the example depicted in Figure 2. Each different branch of the tree is parameterized with a set of corresponding command inputs $\{u^i_{k+j|t}\}$ to be optimized and the number of leaves of the tree is equal to the number m_t of considered corner cases. The time instants over the prediction horizon at which a split in sub-scenarios occurs are those at which one will be able to recognize different maneuvers taken by the obstacle, e.g., drive straight or drive left.

As multiple control moves can occur at the same prediction instant, the state prediction is no longer unique as in the prescient and robust MPC cases. In fact, we associate the state trajectory $\{X^i_{k|t}\}$ to scenario i and let $\{U^i_{k|t}\}$ denote the corresponding sequence of free inputs. To take into account the tree structure, i.e., that scenarios i and j have a common subpath

$$(x^{o_i}_{k|t}, y^{o_i}_{k|t}) = (x^{o_j}_{k|t}, y^{o_j}_{k|t}), \ \ k = 0, \ldots, k_{ij}$$

we impose the equality constraints (a.k.a. "causality constraints") $U^i_{k|t} = U^j_{k|t}$ for all $k = 0, \ldots, k_{ij}$, where k_{ij} denotes the number of future prediction steps that scenarios i and j have in common. Note that all scenarios originate from the current state, i.e., $X^i_{0|t} \equiv X_t$, and have the first move $u^i_{0|t}$ in common, that will be used as the actuation values commanded to the vehicle at time t.

In accordance with the probabilistic scenario tree defined above, we formulate the following **Stochastic MPC** problem

$$\min_{\substack{\{X^i_{k|t},\,U^i_{k|t}\}\\ k=0,\dots,N\\ i=1,\dots,m_t}} \sum_{i=1}^{m_t}\omega_t^i \sum_{k=0}^{N} \|X^i_{k|t}-X^r_{k|t}\|^2_Q + \|U^i_{k|t}-U^r_{k|t}\|^2_R \tag{5a}$$

$$\text{s.t.} \quad X^i_{k+1|t} = f(X^i_{k|t}, U^i_{k|t}) \tag{5b}$$

$$X^i_{0|t} = X_t \tag{5c}$$

$$X^i_{k|t} \in \mathcal{X} \tag{5d}$$

$$U^i_{k|t} \in \mathcal{U} \tag{5e}$$

$$A^i_{k|t}(X^i_{k|t} - X^{\mathrm{r}}_{k|t}) \le B^i_{k|t} \tag{5f}$$

$$(x^i_{k|t} - x^{o_i}_{k|t})^2 + (y^i_{k|t} - y^{o_i}_{k|t})^2 \ge d^2_{\min}$$

$$\forall i = 1,\dots,m_t \tag{5g}$$

$$U^i_{k|t} = U^j_{k|t},\ k=0,\dots,k_{ij}. \tag{5h}$$

An obvious reason why the SMPC formulation (5) is less conservative than the RMPC formulation (4) is that it has more degrees of freedom since it can employ different input values in different scenarios. This implicitly defines a closed-loop optimal policy over the prediction horizon, rather than a single open-loop trajectory as in (4) and (3). Note that constraints (5g) are enforced for all possible scenarios $i = 1,\dots,m_t$, that can occur at time t. Consequently, obstacle avoidance is ensured independently of the probabilities ω_i.

III. Learning Obstacle Scenario Probabilities

In order to use the SMPC formulation described in the previous section to drive the ego vehicle, we need to define the corner-case uncertainty scenarios and the corresponding probabilities ω_t^i, $i = 1,\dots,m_t$.

For simplicity, as depicted in Figure 1, we consider the simplest case of a four-way intersection in which collisions with a single obstacle vehicle must be avoided. The adaptation of our framework to more realistic scenarios is straightforward and we selected simple scenarios also for the ease of discussing and interpreting the results. Hence, we have $m_t \le 3$ possible obstacle maneuvers

$$M \in \{\text{straight, left, right}\}$$

i.e., the obstacle can only pass the intersection by driving straight, turning left, or right.

In addition to each corner case, we need to also define the following probabilities: ω_t^1 associated with $M =$ straight, ω_t^2 with $M =$ left, and ω_t^3 with $M =$ right. To this end, we collect trajectories (see Section III-A) of multiple obstacles at intersections and use them to train a classifier (described in Section III-B), with categorical output M and a suitably defined input feature vector F that is sufficiently informative about the current state of the obstacle (see also Section III-B). As our ultimate goal is to get the time-varying discrete probability distribution $\{\omega_t^1, \omega_t^2, \omega_t^3\}$, we will employ classification methods that also return probabilities.

Each of the three possible corner-case scenarios is associated with an obstacle trajectory

$$(x^{o_i}_{k|t}, y^{o_i}_{k|t}),\ k=0,\dots,N,\ i=1,2,3 \tag{6}$$

that will be used in (4g) and (5g). The obstacle coordinates in (6) are $(x^{o_i}_{k|t}, y^{o_i}_{k|t}) = f^{o_i}(d^{\mathrm{o}}_{k|t})$. In order to obtain a continuous function, we use B-spline interpolation, as described in the next section, and parameterize it by the distance d^{o} traveled by the obstacle vehicle along the center of its lane.

A. Data Collection

We use the microscopic traffic simulator SUMO (Simulation of Urban Mobility) [41] to generate realistic data of obstacle trajectories. We consider a real uncontrolled intersection extracted from Open Street Map and use different types of vehicles with a variety of driving styles.

In order to define the driving style, we rely on the Intelligent Driver Model (IDM) [43], included in SUMO and describes the position and velocity of a vehicle as follows

$$\dot{x}_\delta = v_\delta \tag{7a}$$

$$\dot{v}_\delta = a_{\max}\left(1 - \left(\frac{v_\delta(t)}{v_{\mathrm{des}}(t)}\right)^{\epsilon} - \left(\frac{\sigma^*(v_\delta(t), \Delta v_\delta(t))}{\sigma_\delta(t)}\right)^2\right) \tag{7b}$$

where $a_{\max}$ and $d_{\max}$ are the vehicle's maximum desired acceleration and deceleration, respectively; v_δ, v_{des}, and ϵ are the current velocity, the desired velocity, and a tuning parameter; Δv_δ is the relative velocity with respect to the vehicle in front. Moreover, in (7)

$$\sigma^*(v_\delta(t), \Delta v_\delta(t)) = \sigma_0 + v_\delta(t)T + \frac{v_\delta(t)\Delta v_\delta(t)}{2\sqrt{a_{\max} d_{\max}}} \tag{8}$$

where σ_0 is the minimum desired distance from the vehicle in front, and T the desired time headway, while the net distance between the two vehicles is given by

$$\sigma_\delta(t) = d_{\mathrm{f}}(t) - d_\delta(t) - l_{\mathrm{f}} = \Delta d_\delta(t) - l_{\mathrm{f}}. \tag{9}$$

In (9), label f indicates that the quantity refers to the front vehicle, and d is the position of a vehicle along the centerline of the road, while l_{f} is the length of the vehicle in front.

In this paper, as we consider free road behaviors, Equation (7) simplifies to $\dot{v}_\delta = a_{\max}\left(1 - \left(\frac{v_\delta(t)}{v_{\mathrm{des}}(t)}\right)^{\epsilon}\right)$.

In SUMO, in a free-driving scenario the term $v_{\mathrm{des}}(t)$ is given by

$$v_{\mathrm{des}}(t) = \min(v_{\max}, s\ v^{\mathrm{des}}_{\max}, s\ v_{\mathrm{lim}})$$

where $v_{\max}$, $v^{\mathrm{des}}_{\max}$, and v_{lim} are the maximum speed, desired maximum speed, and speed limit respectively, and s is a parameter called speed factor.

To collect obstacle vehicle data, we consider 3 types of vehicles, namely a passenger car, a motorcycle, and a bus, each with 5 different speed factor values ranging from 0.6 to 1.4. Moreover, for each type of vehicle, we select 6 different vehicle speeds within the range of 40 to 60 km/h.

B. Classifier

To classify the intent of the obstacle vehicle to go straight, turn left, or turn right we use a bagged decision tree [44], due to its ability to return the discrete probability distribution $\{\omega_t^1, \omega_t^2, \omega_t^3\}$ associated with the predicted categorical target M. The motivation to choose this specific classifier is that, as we will discuss next, we observed experimentally that it produces better probability estimates compared to other methods, such as, e.g., Naïve-Bayes classifiers [45] and Support Vector Machines (SVM) [46], without using additional calibration methods [47].

The feature vector consumed by the classifier is defined as

$$F_t = [v^{\mathrm{o}} \; a^{\mathrm{o}} \; \theta^{\mathrm{o}}_{diff} \; d^{\mathrm{o}}_{ln} \; d^{\mathrm{o}}_{lt} \; d^{\mathrm{o}}_{t}]'$$

where, assuming the center of the lane for a straight scenario as the reference lane, v^{o} and a^{o} are, respectively, the speed and acceleration of the obstacle, $\theta^{\mathrm{o}}_{\mathrm{diff}}$ denotes the obstacle's orientation with respect to the reference, d^{o}_{ln}, d^{o}_{lt} are, respectively, its longitudinal and lateral distances from the reference, and d^{o}_t is the distance traveled by the obstacle.

We collected 270 scenarios, out of which 216 trajectories are used for training and the remaining 54 for testing the classifier. The obstacle starts 250 m away from the beginning of the intersection and travels a total distance of 280 m. As a result, by using a resolution of 0.1 m, we collected 605,016 training data and 151,254 test data. The bagged-tree classifier, based on 25 learners, is trained in approximately 160 s on an Intel(R) Corei7-8550U CPU @ 1.80GHz machine in MATLAB R2019a using the fitcensemble and templateTree functions available in the Statistics and Machine Learning Toolbox for MATLAB. The resulting number of splits in the tree is 90,323 (14.93% of the training data set). The predicted probability ω_t^i associated with the actual outcome M_i, $i = 1, 2, 3$, averaged on all trajectories, is displayed in Figure 3 (training data) and Figure 4 (test data): it is evident that when the obstacle is almost 150 m away from the intersection, all the scenarios are equally probable ($\omega_t^i \approx \frac{1}{3}$, $\forall i = 1, 2, 3$); they become more distinguishable as the obstacle moves towards the intersection. In particular, the obstacle is correctly predicted with probability 1 to drive straight about 21 m before the intersection, and about 5.5 m for left and right turns.

Figure 5 compares the Receiver Operating Characteristic (ROC) curve, relating the True Positive Rate (TPR) and False Positive Rate (FPR), obtained by the trained bagged decision tree with those obtained by a Naïve-Bayes (NB) and Support Vector Machine (SVM) classifier for the three different scenarios, i.e., {Straight, Left, Right} on test data. More specifically, Figure 5a shows the ROC curve by considering data points whose distance from the intersection is in the interval [100, 25] m (i.e., far away from the intersection), while Figure 5b for data in the interval [25, 5] m (i.e., close to the intersection). It can be observed that, close to the intersection, the TPR is much better for all the classifiers and the bagged decision tree has a significantly higher TPR compared to NB and SVM. As expected, far away from the intersection it is nearly impossible to distinguish the three scenarios. Finally, when the obstacle is closer than 5 m to the intersection the TPR for the bagged decision tree is 1 for all scenarios, indicating that the classifier correctly identifies the obstacle's intention.

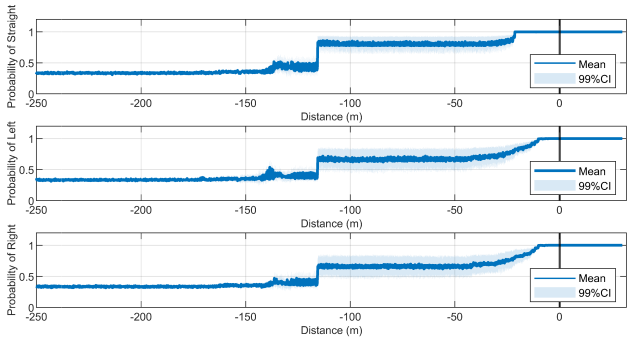


Fig. 3. Predicted probability ω_t^i associated with the outcome M_i of each training trajectory as a function of the distance traveled by obstacle d_t^{o} with respect to the intersection, averaged on all training trajectories with outcomes M_j, $j = 1, 2, 3$. The black vertical line represents the starting point of the intersection.

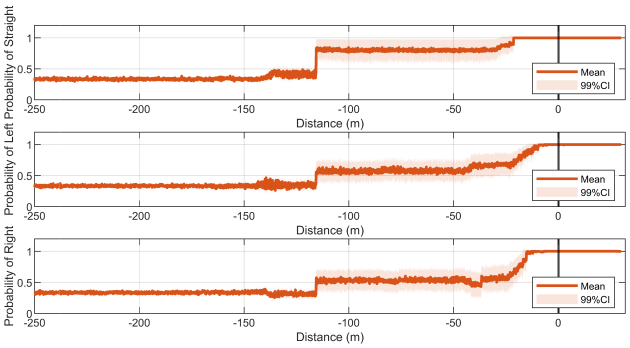


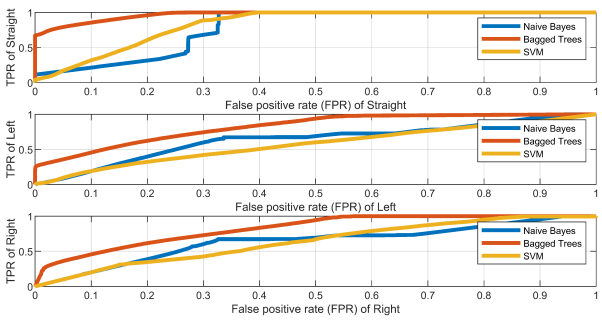
Fig. 4. Predicted probability ω_t^i associated with the outcome M_i of each test trajectory as a function of the distance traveled by obstacle d_t^{o} with respect to the intersection, averaged on all test trajectories with outcomes M_j, $j = 1, 2, 3$. The black vertical line represents the starting point of the intersection.

C. Scenario Tree

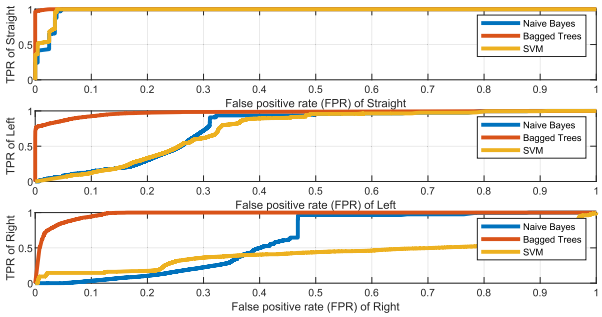
For setting up the SMPC controller, we consider the scenario tree depicted in Figure 2, in which there are three branches and we set $k_{12} = k_{13} < k_{23}$, where $t + k_{12} = t + k_{13}$ is the time at which we can distinguish between scenario1 (straight) and the remaining scenarios, while $t + k_{23}$ is the time at which we can distinguish between the left and right scenarios. The trained classifier generates both the associated probabilities and the times at which we can distinguish the scenarios using the data generated in SUMO. Once the simulation reaches the time instants at which the scenarios can be distinguished, the tree is pruned and the corresponding collision avoidance constraints are removed from SMPC. Note that, in general, pruning is independent of the probabilities assigned to each scenario.

IV. Simulation Results

To compare the PMPC, RMPC, and SMPC approaches introduced in this paper, we consider several realistic examples simulated in SUMO. Firstly, we consider an uncontrolled intersection in Lucca, Italy, where right-hand driving is prescribed

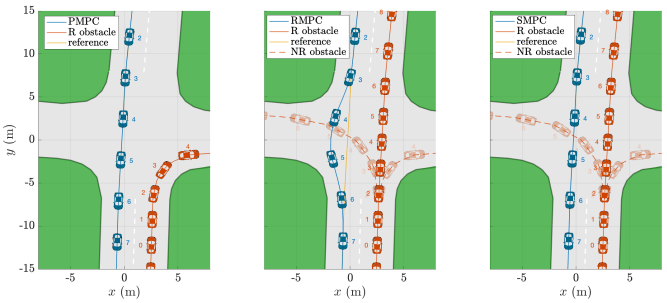


(a) ROC curve of each scenario on the testing dataset within a distance from 100m to 25m from the intersection.

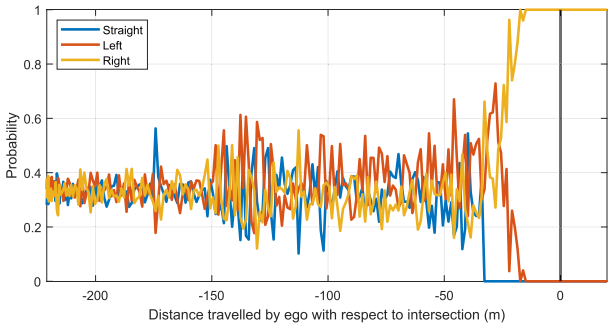


(b) ROC curve for each scenario on the testing dataset from 25m to 5m away from the intersection.

Fig. 5. ROC curve of Naïve-Bayes, Bagged trees, and SVM classifiers on the testing dataset for different ranges of distance from the intersection.



(a) Ego positions, realized (R) obstacle positions and non-realized (NR) obstacle positions near the intersection with right-hand driving.



(b) Estimated scenario probability of the obstacle vehicle as a function of distance traveled by obstacle $d(tT_s)$ from the intersection. The black vertical line represents the starting of the intersection.

Fig. 6. Example 1: Simulation results.

by the road code, importing the road layout from Open Street Map. We model the ego vehicle as a passenger car of 5 m length, maximum speed v_{max}, and speed factor $s = 1$. Closed-loop performance is assessed by the following measure

$$J_{\text{cl}} = \sum_{t=0}^{t_s} \|X_t - X^r_{0|t}\|^2_Q + \|U_t - U^r_{0|t}\|^2_R \tag{10}$$

consistently with the MPC cost function, where t_s is the total simulation time. SUMO's IDM model is used offline to generate a discrete set of reference samples for X^r and U^r, given a desired maximum speed $v_{\max}$ of the ego vehicle. Such samples are used to define B-spline interpolation functions $f_{X^r} : \mathbb{R} \mapsto \mathbb{R}^5$, $f_{U^r} : \mathbb{R} \mapsto \mathbb{R}^2$, such that, for a generic distance d along the central line of the road, $f_{X^r}(d) = [x_r(d)\ y_r(d)\ \theta_r(d)\ v_r(d)\ \delta_r(d)]'$ and $f_{U^r}(d) = [a_r(d)\ \omega_r(d)]'$ provide the corresponding state and input references, respectively. Then, at each controller execution step t, given the piecewise-constant velocity and orientation profiles

$$\begin{aligned} v(\tau) &= v^\star_{k+1|t-1} \\ \theta(\tau) &= \theta^\star_{k+1|t-1}, \quad \forall \tau \in [(t+k)T_s, (t+k+1)T_s] \end{aligned}$$

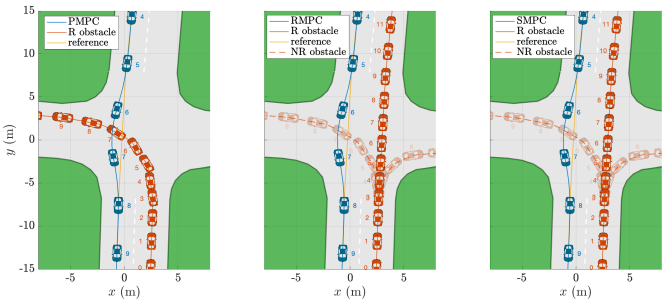
obtained from the previous MPC solution $\{v^\star_{k|t-1}, \theta^\star_{k|t-1}\}$, $k = 0, \dots, N$, we use explicit Runge-Kutta 4 to integrate the differential equation

$$\dot{d}(\tau) = v(\tau)\cos(\theta(\tau) - \theta_r(d(\tau)))$$

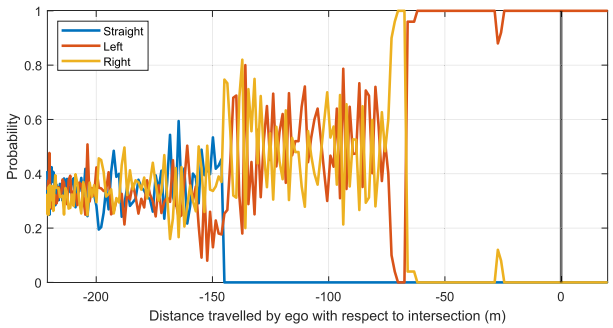
over the prediction horizon $[tT_s, (t+N)T_s]$ to get samples $d_{k|t}$, $k = 0, \dots, N$, of the distance d traveled by the ego vehicle, starting from the initial condition $d_{0|t} = d(tT_s)$ (i.e., the current longitudinal distance traveled by the ego vehicle). Finally, the corresponding MPC references are obtained as $X^r_{k|t} = f_{X^r}(d_{k|t})$ and $U^r_{k|t} = f_{U^r}(d_{k|t})$.

We use sample time $T_s = 0.1$ s for both discretizing the continuous-time ego-vehicle model and executing the MPC controller, which has prediction horizon $N = 40$. The CPU time for solving a single Prescient, Robust, and Stochastic MPC problem is on average, respectively, 0.7 s, 0.9 s, and 2 s using CasADi [48] and the interior-point nonlinear programming solver IPOPT [49]. Note that no particular care was taken in solving the MPC problems efficiently, and dedicated software will produce significantly lower computational times.

Example 1. The ego vehicle travels at $v_{\max} = 43$ km/h and passes the crossroad without turning, the obstacle vehicle is a motorcycle taking a right turn at $v^o_{\max} = 44$ km/h with $speedfactor = 1.1$. Figure 6 shows the positions of both ego and obstacle vehicle near the intersection, and the probabilities $\omega^1_t, \omega^2_t, \omega^3_t$ estimated by the classifier with respect to the distance traveled by ego $d(tT_s)$ from the intersection. It is apparent that, initially, RMPC and SMPC behave similarly until the scenarios start becoming distinguishable, i.e., the probability distribution starts getting non-uniform. In Figure 6b the black vertical line represents the point at which the ego vehicle enters the intersection. Note that, as remarked earlier, a right turn can only be predicted with high probability by the classifier when the obstacle is close to the intersection. Thanks to the use of the scenario tree, SMPC tends to behave more similarly to PMPC than RMPC, which instead remains conservative. This is a crucial benefit



(a) Ego positions, realized (R) obstacle positions and non-realized (NR) obstacle positions near the intersection with right-hand driving.



(b) Estimated scenario probability of the obstacle vehicle as a function of distance traveled by obstacle $d(tT_s)$ from the intersection. The black vertical line represents the starting of the intersection.

Fig. 7. Example 2: Simulation results.

of using SMPC compared to other schemes: even without an *exact* information about the future positions of the obstacle, past information is used in a more clever way to infer *possible* obstacle positions and the related chances to realize. Thanks to the probabilistic model and the use of additional degrees of freedom associated with it, SMPC takes control decisions that, by construction, are best in expectation. This behavior is reflected in Table I, which shows that the cost associated with SMPC is significantly smaller than that of RMPC, which instead gives the same importance to all possible scenarios.

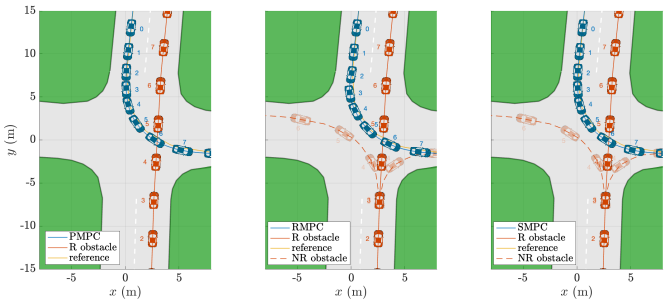
Example 2. The second example refers to the case in which the ego vehicle aims at passing the intersection without turning at a maximum speed $v_{\max} = 50$ km/h, while the obstacle is a bus coming from the opposite direction at $v^o_{\max} = 54$ km/h, with speed factor $s = 1.3$, taking a left turn. The results obtained using the different MPC formulations in this scenario where a collision is possible are described in Figure 7 and Table I, from which we can infer that all the control algorithms result in the same closed-loop costs since the ego vehicle deviates from the reference path to avoid the collision with the bus. Figure 7b shows the probabilities ω_t^1, ω_t^2, ω_t^3 estimated by the classifier with respect to the distance traveled by the ego vehicle $d(tT_s)$ from the intersection and the black vertical line at 0 m represent the starting point of the intersection.

Example 3. The ego vehicle travels at $v_{\max} = 45$ km/h, takes a left turn, and the obstacle is a car with speed factor $s = 1$ driving straight across the intersection in the opposite direction at a maximum speed of $v^o_{\max} = 40$ km/h as shown in Figure 8a. There is no potential collision between the ego

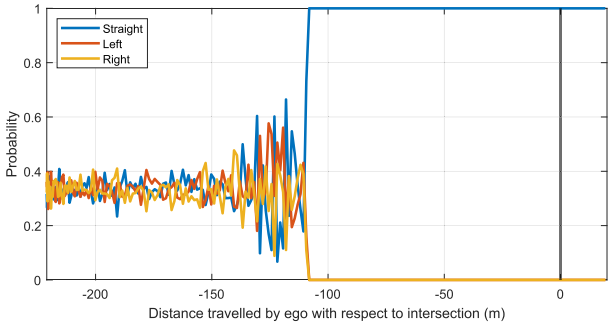
TABLE I

CLOSED-LOOP COST J_{cl} (10) OF PRESCIENT, ROBUST, AND STOCHASTIC MPC OBTAINED IN THE THREE TEST EXAMPLES

Example	Prescient MPC	Robust MPC	Stochastic MPC
Example 1	14.8514	14.8514	14.8514
Example 2	6.9371	33.1935	7.4447
Example 3	20.4692	40.6931	20.4692
Example 4	6.9371	117.291	10.3261
Example 5	10.5519	10.5519	10.5519



(a) Ego positions, realized (R) obstacle positions and non-realized (NR) obstacle positions near the intersection with right-hand driving.



(b) Estimated scenario probability of the obstacle vehicle as a function of distance traveled by obstacle $d(tT_s)$ from the intersection. The black vertical line represents the starting of the intersection.

Fig. 8. Example 3: Simulation results.

vehicle and the obstacle in this example. From Table I it is evident that SMPC works closer to PMPC than RMPC thanks to the exploitation of the scenario tree. As shown in Figure 8b, the straight scenario is detected early by the classifier, thus SMPC lifts the collision avoidance constraints from the other two scenarios. Instead, the actions commanded by RMPC suffer from the presence of the irrelevant scenario (right turn).

Example 4. The ego vehicle travels straight across the intersection at $v_{\max} = 43$ km/h. A motorcycle with a speed factor $s = 1.3$ traveling at a maximum speed of 44 km/h takes a right turn. This is a non-collision scenario, and in Table I we observe that the RMPC has the highest closed-loop cost as it takes into account the scenario (left turn) that leads to a collision as well, while SMPC will consider such a scenario very unlikely, resulting in an overall control action that has a cost closer to that of PMPC.

Example 5. The ego vehicle crosses the intersection without turning at $v_{\max} = 45$ km/h and the obstacle is a bus taking a left turn at a speed $v^o_{\max} = 50$ km/h with speed factor $s = 1$. This is also a collision scenario where all the control algorithms have the same closed-loop cost.

V. Conclusion

This paper has proposed a stochastic MPC approach to autonomous driving, particularly focusing on the case of uncontrolled intersections. By training a classifier on obstacle trajectories that can also return the probabilities corresponding to each possible predictable categorical value, we could set up a stochastic optimization problem whose solution, at each time step t, provides the required command action on the ego vehicle. We have shown that using such probabilities is beneficial, as it makes SMPC less conservative than a robust MPC approach, in which they are not taken into account.

Future work will be devoted to extending the idea to multiple obstacles; in such a case, it will be important to develop methods to automatically reduce the number of scenarios considered in the SMPC formulation to keep the computation complexity of the approach manageable. Another important variation to analyze is the use of more meaningful MPC performance indices to optimize, such as considering driving quality.

Acknowledgment

The authors are grateful to the SUMO Traffic Simulator community, and in particular to Jackob Erdman, for their help.

References

[1] C. Badue et al., "Self-driving cars: A survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113816.

[2] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using optical sensors: A review," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2004, pp. 585–590.

[3] M. Gulzar, Y. Muhammad, and N. Muhammad, "A survey on motion prediction of pedestrians and vehicles for autonomous driving," *IEEE Access*, vol. 9, pp. 137957–137969, 2021.

[4] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *Proc. 12th Int. Symp. Adv. Vehicle Control*, 2014, pp. 712–719.

[5] Y. Jeong and K. Yi, "Target vehicle motion prediction-based motion planning framework for autonomous driving in uncontrolled intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 168–177, Jan. 2021.

[6] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 2, pp. 129–140, Jun. 2018.

[7] D. Li, Y. Wu, B. Bai, and Q. Hao, "Behavior and interaction-aware motion planning for autonomous driving vehicles based on hierarchical intention and motion prediction," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.

[8] I. Batkovic, M. Zanon, N. Lubbe, and P. Falcone, "A computationally efficient model for pedestrian motion prediction," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 374–379.

[9] Y. Hu, W. Zhan, and M. Tomizuka, "Probabilistic prediction of vehicle semantic intention and motion," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 307–313.

[10] X. Liu, G. Jin, Y. Wang, and C. Yin, "A deep learning-based approach to line crossing prediction for lane change maneuver of adjacent target vehicles," in *Proc. IEEE Int. Conf. Mechatron. (ICM)*, Mar. 2021, pp. 1–6.

[11] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 1307–1312.

[12] T.-M. Hsu, Y.-R. Chen, and C.-H. Wang, "Decision making process of autonomous vehicle with intention-aware prediction at unsignalized intersections," in *Proc. Int. Autom. Control Conf. (CACS)*, Nov. 2020, pp. 1–4.

[13] Y. Yoon and K. Yi, "Design of longitudinal control for autonomous vehicles based on interactive intention inference of surrounding vehicle behavior using long short-term memory," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 196–203.

[14] T. Zhang, W. Song, M. Fu, Y. Yang, and M. Wang, "Vehicle motion prediction at intersections based on the turning intention and prior trajectories model," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1657–1666, Oct. 2021.

[15] A. Benterki, M. Boukhnifer, V. Judalet, and C. Maaoui, "Artificial intelligence for vehicle behavior anticipation: Hybrid approach based on maneuver classification and trajectory prediction," *IEEE Access*, vol. 8, pp. 56992–57002, 2020.

[16] A. Benterki, M. Boukhnifer, V. Judalet, and M. Choubeila, "Prediction of surrounding vehicles lane change intention using machine learning," in *Proc. 10th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, vol. 2, Sep. 2019, pp. 839–843.

[17] H. Weinreuter, N.-R. Strelau, B. Deml, and M. Heizmann, "Intention prediction of car drivers at inner city junctions," in *Proc. Int. Symp. Electron. Telecommun. (ISETC)*, Nov. 2020, pp. 1–4.

[18] T. Zhang, M. Fu, and W. Song, "Risk-aware decision-making and planning using prediction-guided strategy tree for the uncontrolled intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10791–10803, Oct. 2023.

[19] D. Mayne, J. Rawlings, and M. Diehl, *Model Predictive Control: Theory and Design*, 2nd ed. Madison, WI, USa: Nob Hill, 2018.

[20] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: Bridging the gap via the real-time iteration," *Int. J. Control*, vol. 93, no. 1, pp. 62–80, Jan. 2020.

[21] A. Bemporad, D. Bernardini, R. Long, and J. Verdejo, "Model predictive control of turbocharged gasoline engines for mass production," in *Proc. WCX: SAE World Congr. Exper.*, Detroit, MI, USA, Apr. 2018.

[22] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2017, pp. 174–179.

[23] M. G. Plessen, D. Bernardini, H. Esen, and A. Bemporad, "Spatial-based predictive control and geometric corridor planning for adaptive cruise control coupled with obstacle avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 38–50, Jan. 2018.

[24] L. D'Alfonso, F. Giannini, G. Franzè, G. Fedele, F. Pupo, and G. Fortino, "Autonomous vehicle platoons in urban road networks: A joint distributed reinforcement learning and model predictive control approach," *IEEE/CAA J. Automatica Sinica*, vol. 11, no. 1, pp. 141–156, Jan. 2024.

[25] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Syst. Mag.*, vol. 36, no. 6, pp. 30–44, Dec. 2016.

[26] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1468–1480, Jul. 2012.

[27] S. D. Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1018–1031, May 2014.

[28] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Atlanta, GA, USA, Dec. 2010, pp. 6077–6082.

[29] G. Ripaccioli, D. Bernardini, S. D. Cairano, A. Bemporad, and I. Kolmanovsky, "A stochastic model predictive control approach for series hybrid electric vehicle power management," in *Proc. Amer. Control Conf.*, Baltimore, MD, USA, Jun./Jul. 2010, pp. 5844–5849.

[30] Y. Jeong, "Probabilistic game theory and stochastic model predictive control-based decision making and motion planning in uncontrolled intersections for autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 15254–15267, Dec. 2023.

[31] H. Bao et al., "Moment-based model predictive control of autonomous systems," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 4, pp. 2939–2953, Jan. 2023.

[32] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic MPC with multi-modal predictions for traffic intersections," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 635–640.

[33] I. Batkovic, M. Zanon, M. Ali, and P. Falcone, "Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 256–262.

[34] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario MPC approach for uncertain multi-modal obstacles," *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 947–952, Jul. 2021.

[35] I. Batkovic, M. Ali, P. Falcone, and M. Zanon, "Safe trajectory tracking in uncertain environments," *IEEE Trans. Autom. Control*, vol. 68, no. 7, pp. 4204–4217, Jul. 2023.

[36] I. Batkovic, A. Gupta, M. Zanon, and P. Falcone, "Experimental validation of safe MPC for autonomous driving in uncertain environments," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 5, pp. 2027–2042, Sep. 2023.

[37] D. Madås et al., "On path planning methods for automotive collision avoidance," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 931–937.

[38] K. Berntorp, "Path planning and integrated collision avoidance for autonomous vehicles," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 4023–4028.

[39] C. H. Ulfsjöö and D. Axehill, "On integrating POMDP and scenario MPC for planning under uncertainty—With applications to highway driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 1152–1160.

[40] R. Oliveira, S. H. Nair, and B. Wahlberg, "Interaction and decision making-aware motion planning using branch model predictive control," 2023, *arXiv:2302.00060*.

[41] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.

[42] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer, 2011.

[43] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, p. 1805, Aug. 2000.

[44] N. V. Chawla and D. A. Cieslak, "Evaluating probability estimates from decision trees," in *Proc. AAAI*, 2006, pp. 1–6.

[45] F.-J. Yang, "An implementation of naive Bayes classifier," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2018, pp. 301–306.

[46] E. Mayoraz and E. Alpaydin, "Support vector machines for multi-class classification," in *Proc. Int. Work-Conf. Artif. Neural Netw.* Berlin, Germany: Springer, 1999, pp. 833–842.

[47] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 625–632.

[48] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

[49] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.

Surya Soman received the master's degree in control systems from the PSG College of Engineering and Technology, India, in 2016. She is currently pursuing the Ph.D. degree in systems science with the IMT School for Advanced Studies, Lucca, Italy. From 2021 to 2023, she was a Software Engineer at Baker Hughes, Florence, Italy, where she became a Control Software Engineer in 2023. Her research interests include model predictive control, robust control, and machine learning for automotive and aerospace applications.

Mario Zanon (Senior Member, IEEE) received the Diplôme d'Ingénieur degree from École Centrale Paris, France, in 2010, the master's degree in mechatronics from the University of Trento, Italy, and the Ph.D. degree in electrical engineering from KU Leuven, Belgium, in November 2015. After research stays at KU Leuven, the University of Bayreuth, Chalmers University, and the University of Freiburg, he received the Ph.D. degree. He held a Post-Doctoral Researcher at Chalmers University until the end of 2017. In 2018, he became an Assistant Professor at the IMT School for Advanced Studies Lucca, Italy, where he became an Associate Professor in 2021. His research interests include numerical methods for optimization, economic MPC, reinforcement learning, and the optimal control and estimation of nonlinear dynamic systems, in particular for aerospace and automotive applications.

Alberto Bemporad (Fellow, IEEE) received the master's degree (cum laude) in electrical engineering and the Ph.D. degree in control engineering from the University of Florence, Italy, in 1993 and 1997, respectively. From 1996 to 1997, he was with the Center for Robotics and Automation, Department of Systems Science and Mathematics, Washington University in St. Louis. From 1997 to 1999, he was a Post-Doctoral Researcher with the Automatic Control Laboratory, ETH Zürich, Switzerland, where he collaborated as a Senior Researcher until 2002. From 1999 to 2009, he was with the Department of Information Engineering, University of Siena, Italy, becoming an Associate Professor in 2005. From 2010 to 2011, he was with the Department of Mechanical and Structural Engineering, University of Trento, Italy. Since 2011, he has been a Full Professor with the IMT School for Advanced Studies Lucca, Italy, where he was the Director of the institute from 2012 to 2015. He spent visiting periods at Stanford University, the University of Michigan, and Zhejiang University. In 2011, he co-founded ODYS S.r.l., a company specialized in developing model predictive control systems for industrial production. He has published more than 400 articles in the areas of model predictive control, hybrid systems, optimization, and automotive control, and is the co-inventor of 21 patents. He is the author or co-author of various software packages for model predictive control design and implementation, including the Model Predictive Control Toolbox (The Mathworks, Inc.) and the Hybrid Toolbox for MATLAB. He received the IFAC High-Impact Paper Award for the 2011–2014 triennial, the IEEE CSS Transition to Practice Award in 2019, and the 2021 SAE Environmental Excellence in Transportation Award. He was an Associate Editor of IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 2001 to 2004 and the Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society from 2002 to 2010.