Density-Based Stochastic Reachability Computation for Occupancy Prediction in Automated Driving

Shadi Haddad, Abhishek Halder, *Senior Member, IEEE*, and Baljeet Singh

***Abstract*—We propose a stochastic reachability computation framework for occupancy prediction in automated driving by directly solving the underlying transport partial differential equation (PDE) governing the advection of the closed-loop joint density functions. The resulting nonparametric gridless computation is based on integration along the characteristic curves, and it allows online computation of the time-varying collision probabilities. We provide numerical simulations for multi-lane highway driving scenarios to highlight the scope of the proposed method.**

***Index Terms*—Automated driving, collision probability, occupancy prediction, stochastic reachability.**

I. Introduction

GUARANTEEING safety without sacrificing performance in automated driving, especially in a mixed traffic of autonomous, semi-autonomous, and human-driven vehicles, requires real-time planning and decision-making against time-varying uncertainties. From a decision-making (ego) vehicle's perspective, these uncertainties stem not only from the environment (states and intents of neighboring traffic, road geometry, weather, and lighting conditions) but also from the intrinsic sources (sensing and localization errors, communication packet drop, computational latency, model mismatch between the planning and control software). Thus, timely and accurate forecasting of the states of the ego vehicle and its environment is critical for provably safe decision-making. At the same time, given the complexity of the dynamics and environment, it is extremely challenging to perform such forecasting computation at a time scale much smaller than the physical dynamics time scale. Therefore, a critical gap exists in real-time forecasting to enable provably safe automation for achieving objectives such as collision avoidance, safe lane change, and separation management. This article presents a framework for automated navigation by designing novel computational tools for stochastic uncertainty propagation enabling fast occupancy prediction in multi-lane driving scenarios.

The importance of forecasts in automated driving while accounting for the dynamic uncertainties has been well-advocated [1], [2] in the literature. The problem of stochastic occupancy prediction has been addressed before using finite state Markov chain abstractions [3], [4] on a discretization of the joint state and action space, and also by the Monte Carlo methods [5]–[7]. The purpose of the present work is to harness recent developments [8]–[11] in theory and algorithms for solving the transport partial differential equation (PDE) governing the evolution of the joint state probability density function (PDF) subject to a known trajectory-level dynamics, to enable stochastic occupancy prediction in automated driving. This PDE-based computational framework has a number of attractive features pertinent to occupancy prediction. For instance, exact pointwise evaluation of the time-varying joint state PDFs can be performed along the trajectories of the closed-loop dynamics conditioned on the (joint) stochastic uncertainties in the initial conditions and parameters–a sharp juxtaposition with the discretization-based methods such as Markov chain abstraction or Monte Carlo, where the functional approximation of the joint PDFs are computed–the quality and computational cost of approximation being dependent on the resolution of the discretization grid. In the proposed framework, the exact evaluation is possible since the underlying transport PDE is linear and its characteristic curves happen to coincide with the closed-loop trajectories, which can be algorithmically leveraged for scattered weighted point cloud-based computation, as shown in Section III.

Another notable feature of the proposed framework is that the computation is nonparametric: we use neither statistical (e.g., Gaussian, Gaussian mixture, exponential family) nor dynamical (e.g., polynomial vector field) approximation. In contrast to the moment-based uncertainty propagation [12], [13], the proposed method does not need *a priori* knowledge of the sufficient statistic of the transient stochastic states, which are difficult to obtain in settings such as ours. We discuss these features in Section III-B.

The research results presented here advance the state-of-the-art for stochastic prediction to tackle challenges in the automated driving application domain. Our specific contributions include: 1) a novel nonparametric computational method to co-evolve the joint PDF values with the stochastic states in a gridless manner subject to the exact vehicle dynamics and 2) a detailed analysis demonstrating how the same

Manuscript received 14 September 2021; revised 22 December 2021; accepted 16 January 2022. Date of publication 10 February 2022; date of current version 21 October 2022. This work was supported in part by the 2019 Ford University Research Project and in part by the Chancellor Fellowship from the University of California at Santa Cruz. Recommended by Associate Editor M. Oishi. *(Corresponding author: Abhishek Halder.)*

Shadi Haddad and Abhishek Halder are with the Department of Applied Mathematics, University of California at Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: shhaddad@ucsc.edu; ahalder@ucsc.edu).

Baljeet Singh is with Ford Greenfield Labs, Palo Alto, CA 94304 USA (e-mail: bsing124@ford.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2022.3145976.

Digital Object Identifier 10.1109/TCST.2022.3145976

computation can be used for estimating the time-varying collision probabilities, which, in turn, may be used for decision-making and/or driving assist.

We clarify here that in this work, we are not designing a control policy. Instead, our focus is on density-based stochastic reachability computation for given control policy. We perform this computation in a manner that is not specific to the structural aspects of the given control policy, i.e., the computation is generalizable. In a different endeavor [36], we have pursued stochastic optimal control design that steers a given state PDF to a safe state PDF.

The rest of this article is organized as follows. In Section II, we describe two models for vehicle dynamics. Section III explains the transport PDE-based stochastic reachability computation. In Section IV, we introduce the technical ingredients to promote probabilistic safety against collisions. Numerical simulations using the aforesaid models are detailed in Section V. Section VI concludes the article.

Notations: We use bold-faced small letters for vectors and bold-faced capital letters for matrices. The symbol diag denotes a diagonal matrix, and blkdiag denotes a block-diagonal matrix. We use the notation $\boldsymbol{I}_n$ for the $n \times n$ identity matrix. For $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, the notation $\boldsymbol{x} \sim \rho$ means that the random vector $\boldsymbol{x}$ has the joint PDF ρ. By definition,

$$\rho \geq 0 \quad \text{for all} \quad \boldsymbol{x} \in \mathcal{X}, \quad \text{and} \quad \int_{\mathcal{X}} \rho \, \mathrm{d}\boldsymbol{x} = 1.$$

The symbol $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a joint Gaussian PDF with the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. The support of a joint PDF ρ is denoted as $\operatorname{spt}(\rho) := \{\boldsymbol{x} \mid \rho(\boldsymbol{x}) \neq 0\}$. The notation $\mathbb{1}_{\mathcal{X}}$ stands for the indicator function of a set $\mathcal{X}$. We use standard abbreviations such as ODE for ordinary differential equation, LTI for linear time invariant, MPC for model predictive control, QP for quadratic programming, and w.r.t. for "with respect to."

II. Models

In this section, we detail two vehicle dynamics models that will be used in this article. The first is a kinematic bicycle model with four states and two controls, and the second is a dynamic bicycle model in road aligned coordinates with six states and three controls.

Our rationale behind considering two different models is to demonstrate the generalizability of the proposed computational framework in the sense that the proposed method of stochastic prediction is not contingent on the structural aspects of dynamical nonlinearity. Both the models are widely used in the automated driving literature. The model in Section II-A is simpler while that in Section II-B is more complex and is of higher fidelity.

Which model is to be used when depends on the application at hand. For instance, when the computational resource is scarce, we expect usage of a model of the sort given in Section II-A. If a reasonable amount of computational resource is available, and/or accounting road curvature, tire forces are important for the application at hand, and then we expect usage of a model of the sort given in Section II-B.

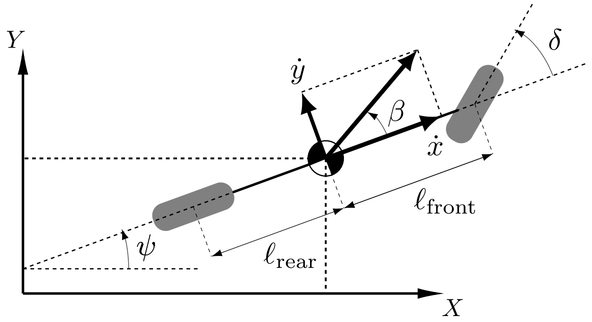


Fig. 1. Schematic of notations used in the kinematic bicycle model in Section II-A. Here, (X, Y) denotes an inertial coordinate system.

In Section III and thereafter, we will consider stochastic uncertainties in the initial states of the vehicular dynamics models given below. We suppose that the initial condition is the only source of stochasticity over the time horizon of interest. In other words, given a random realization of the initial state, the time evolution of the state vector follows an ODE model as in Section II-A or II-B. For a given initial distribution, the deterministic trajectory-level dynamics will induce a dynamics on the probability distribution of the state vector, which will be discussed in Section III. In this section, we focus on the ODE models for vehicle dynamics with given controls. To highlight that the proposed computational framework is agnostic to both dynamical nonlinearity and controller structures, in Section V, we will simulate the model in Section II-A with an open-loop control and the model in Section II-B with a closed-loop control policy.

A. Kinematic Bicycle Model

We consider the kinematic bicycle model [14] with the state vector $\boldsymbol{x} := (x, y, v, \psi)^\top$ and control vector $\boldsymbol{u} := (a_c, \delta)^\top$, given by (see Fig. 1)

$$\dot{x} = v\cos(\psi + \beta) \tag{1a}$$
$$\dot{y} = v\sin(\psi + \beta) \tag{1b}$$
$$\dot{v} = a_c \tag{1c}$$
$$\dot{\psi} = \frac{v}{\ell_{\text{rear}}}\sin\beta \tag{1d}$$

wherein the sideslip angle

$$\beta = \arctan\left(\frac{\ell_{\text{rear}}}{\ell_{\text{front}} + \ell_{\text{rear}}}\tan\delta\right) \tag{2}$$

where the parameters $\ell_{\text{front}} and \ell_{\text{rear}}$ are the distances of the vehicle's center of mass to the front and rear axles, respectively. The state vector $\boldsymbol{x}$ comprises the inertial position (x, y) for the vehicle's center of mass, its speed v, and the vehicle's inertial heading angle ψ. The control vector $\boldsymbol{u}$ comprises the acceleration a_c and the front steering wheel angle δ. In Section V-A1, we will report stochastic prediction using this model with an open-loop control.

B. Dynamic Bicycle Model in Road Aligned Coordinates

1) Open-Loop Dynamics: We now consider the open-loop dynamic bicycle model [15], [16] in road aligned coordinates.

TABLE I

PARAMETERS USED IN THE MODEL (4)

Symbol	Description	Numerical value
a	distance of the vehicle's center-of-gravity from its front axle	1.432 m
b	distance of the vehicle's center-of-gravity from its rear axle	1.472 m
c	the vehicle's half-track width	0.8125 m
m	mass of the vehicle	2050 kg
I_z	yawing moment of inertia of the vehicle	3344 kg m^2

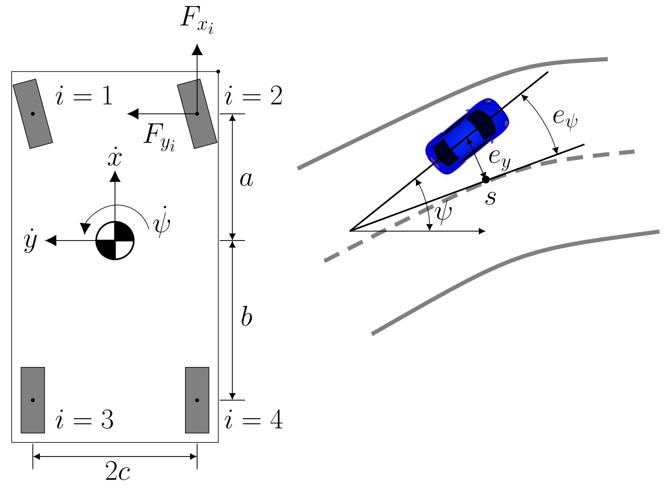


Fig. 2. For the dynamic bicycle model in Section II-B, shown above are (left:) the forces in the body frame and (right:) the vehicle states depicted in the road coordinates.

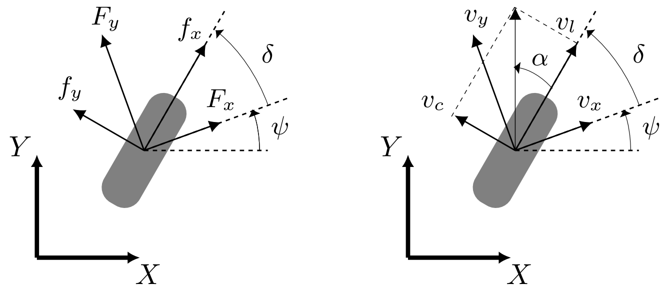


Fig. 3. Schematic of the notations used in the dynamic bicycle model in Section II-B.

The state vector is

$$\boldsymbol{x} := \left(v_x, v_y, v_\psi, e_\psi, e_y, s\right)^\top \tag{3}$$

where v_x and v_y denote the longitudinal and lateral speeds, respectively, in m/s. Furthermore, v_ψ denotes the yaw rate in rad/s. The states ($e_\psi, e_y,$ and s), respectively, denote the heading angle error in rad, the lateral position error measured w.r.t. the center of the road in m, and the longitudinal position of the vehicle along the road in m, as illustrated in Fig. 2. The control vector

$$\boldsymbol{u} := \left(\delta_{\text{front}}, \beta_{\text{left}}, \beta_{\text{right}}\right)^\top$$

comprises of front wheel steering angle δ_{front} and the braking ratios for the left and right wheels: β_{left} and β_{right}. The dynamics is given by

$$\dot{v}_x = v_y v_\psi + \frac{1}{m}\sum_{i=1}^{4} F_{x_i} \tag{4a}$$

$$\dot{v}_y = -v_x v_\psi + \frac{1}{m}\sum_{i=1}^{4} F_{y_i} \tag{4b}$$

$$\dot{v}_\psi = \frac{1}{I_z}\left(a\left(F_{y_1}+F_{y_2}\right) - b\left(F_{y_3}+F_{y_4}\right) + c\sum_{i=1}^{4}(-1)^i F_{x_i}\right) \tag{4c}$$

$$\dot{e}_\psi = v_\psi - \frac{\kappa}{1-\kappa e_y}\left(v_x\cos\left(e_\psi\right) - v_y\sin\left(e_\psi\right)\right) \tag{4d}$$

$$\dot{e}_y = v_x\sin\left(e_\psi\right) + v_y\cos\left(e_\psi\right) \tag{4e}$$

$$\dot{s} = \frac{1}{1-\kappa e_y}\left(v_x\cos\left(e_\psi\right) - v_y\sin\left(e_\psi\right)\right) \tag{4f}$$

wherein the parameters pertaining to the vehicle geometry are listed in Table I. Here, κ denotes the curvature of the road.

To describe the external forces in (4), define the longitudinal and lateral forces (see Fig. 3), denoted by f_{x_i} and f_{y_i} respectively, as

$$f_{x_i} := \zeta\beta_i F_{z_i} \tag{5a}$$

$$f_{y_i} := -C_\alpha \tan\alpha_i \tag{5b}$$

where the braking ratio $\beta_i := \beta_{\text{left}}$ for $i \in \{1,3\}$ and $\beta_i := \beta_{\text{right}}$ for $i \in \{2,4\}$. In particular, $\beta_i \in [-1,1]$ where $\beta_i = -1$ corresponds to the maximum available braking, and $\beta_i = +1$ corresponds to the maximum available throttle. We assume that the friction coefficient ζ between the tires and the road surface is the same for all the four tires, and it remains constant over the time horizon of interest. In (5a), the normal forces F_{z_i} are determined from the force and moment balance as

$$F_{z_1} = F_{z_2} = \frac{mg}{2}\frac{b}{a+b}, \quad F_{z_3} = F_{z_4} = \frac{mg}{2}\frac{a}{a+b} \tag{6}$$

where $g = 9.81$ m/s^2 is the acceleration due to gravity. In (5b), the tire cornering stiffness parameter $C_\alpha = 250$ kN/rad, and the slip angles α_i satisfy

$$\tan\alpha_i = \frac{v_{c_i}}{v_{\ell_i}}, \quad i \in \{1,2,3,4\} \tag{7}$$

wherein v_{c_i} and v_{ℓ_i}, for $i \in \{1,2,3,4\}$, respectively, denote the components of the 4×1 vectors $\boldsymbol{v}_c$ and $\boldsymbol{v}_\ell$, given via the augmented equation

$$\begin{pmatrix}\boldsymbol{v}_\ell\\ \boldsymbol{v}_c\end{pmatrix} = \text{blkdiag}(\boldsymbol{U}(\delta_{\text{front}}), \boldsymbol{U}(\delta_{\text{front}}), \boldsymbol{I}_2, \boldsymbol{I}_2) \times \begin{pmatrix}1 & 0 & -c\\ 0 & 1 & a\\ 1 & 0 & c\\ 0 & 1 & a\\ 1 & 0 & -c\\ 0 & 1 & -b\\ 1 & 0 & c\\ 0 & 1 & -b\end{pmatrix}\begin{pmatrix}v_x\\ v_y\\ v_\psi\end{pmatrix}. \tag{8}$$

In (8), the orthogonal matrix

$$\boldsymbol{U}(\delta_{\text{front}}) := \begin{pmatrix}\cos(\delta_{\text{front}}) & \sin(\delta_{\text{front}})\\ -\sin(\delta_{\text{front}}) & \cos(\delta_{\text{front}})\end{pmatrix} \tag{9}$$

where δ_{front} is the first component of the control $\boldsymbol{u}$ and satisfies $-10° \leq \delta_{\text{front}} \leq 10°$. The longitudinal and lateral forces, F_{x_i} and F_{y_i}, respectively, are defined in the body frame by

$$\begin{pmatrix}F_{x_i}\\ F_{y_i}\end{pmatrix} = \boldsymbol{M}_i\begin{pmatrix}f_{x_i}\\ f_{y_i}\end{pmatrix} \tag{10}$$

where $\boldsymbol{M}_i = \boldsymbol{U}^\top(\delta_{\text{front}})$ for $i = 1, 2$, and $\boldsymbol{M}_i = \boldsymbol{I}_2$ for $i = 3, 4$. The forces f_{x_i} and f_{y_i} in (10) are given by (5).

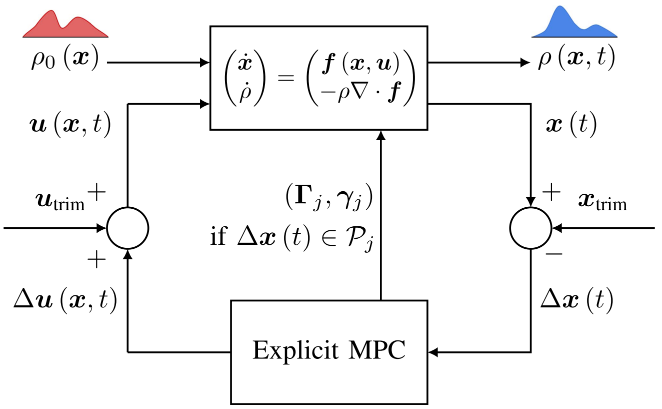
2) Feedback Synthesis: For feedback design, we first linearize the dynamics (4) around the "velocity-hold" trim condition $v_x = 20$ m/s. We use findop in SIMULINK to compute the trim condition[1] $(\boldsymbol{x}_{\text{trim}}, \boldsymbol{u}_{\text{trim}})$ via sequential QP, where in addition to the equality constraint $v_x = 20$ m/s, we impose the inequality constraints

$$\begin{pmatrix}-10°\\ -1\\ -1\end{pmatrix} \leq \boldsymbol{u} \leq \begin{pmatrix}10°\\ 1\\ 1\end{pmatrix} \tag{11a}$$

$$\left(\mu_{e_y}\right)_0 - 1.5 \leq e_y \leq \left(\mu_{e_y}\right)_0 + 1.5. \tag{11b}$$

The constraints (11a) enforce bounded controls. The path constraint (11b) enforces the lateral position error e_y to be within ±1.5 m of the initial mean lateral position error $(\mu_{e_y})_0$. Since we allow stochastic uncertainties in initial condition, the mean and hence the computation of the trim will depend on the joint state PDF $\rho_0(\boldsymbol{x})$ supported over the initial states at time $t = 0$, where the state vector is given by (3).

Linearizing (4) around the resulting trim point $(\boldsymbol{x}_{\text{trim}}, \boldsymbol{u}_{\text{trim}})$, we obtain the LTI matrix pair $(\boldsymbol{A}_{\text{trim}}, \boldsymbol{B}_{\text{trim}})$, which is then used to obtain the explicit MPC feedback $\boldsymbol{u} = \boldsymbol{\pi}(\boldsymbol{x}, t)$ for this linearized system to minimize the deviation from $(\boldsymbol{x}_{\text{trim}}, \boldsymbol{u}_{\text{trim}})$ over a prediction horizon subject to the constraints (11) via the MPC toolbox [17]. Specifically, let $\Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_{\text{trim}}$, $\Delta\boldsymbol{u} := \boldsymbol{u} - \boldsymbol{u}_{\text{trim}}$. The MPC objective penalizes the cumulative cost



$\rho_0(\boldsymbol{x})$
$\begin{pmatrix}\dot{\boldsymbol{x}}\\ \dot{\rho}\end{pmatrix} = \begin{pmatrix}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})\\ -\rho\nabla\cdot\boldsymbol{f}\end{pmatrix}$
$\rho(\boldsymbol{x},t)$
$\boldsymbol{u}(\boldsymbol{x},t)$
$\boldsymbol{x}(t)$
$\boldsymbol{u}_{\text{trim}}$ +
+
$(\boldsymbol{\Gamma}_j, \boldsymbol{\gamma}_j)$
if $\Delta\boldsymbol{x}(t) \in \mathcal{P}_j$
+ $\boldsymbol{x}_{\text{trim}}$
−
$\Delta\boldsymbol{u}(\boldsymbol{x},t)$
Explicit MPC
$\Delta\boldsymbol{x}(t)$

Fig. 4. Block diagram for the computation of the time-varying joint state PDF $\rho(\boldsymbol{x}, t)$ with open-loop dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ and explicit MPC feedback, subject to the initial joint state PDF $\rho_0(\boldsymbol{x})$. The linearized explicit MPC is synthesized about the trim state–control pair $(\boldsymbol{x}_{\text{trim}}, \boldsymbol{u}_{\text{trim}})$ (see Section II-B2).

$$\int_t^{t+t_p} (\Delta\boldsymbol{x}(\tau))^\top \boldsymbol{Q}\Delta\boldsymbol{x}(\tau)\,\mathrm{d}\tau + \int_t^{t+t_p}\left((\Delta\boldsymbol{u}(\tau))^\top\boldsymbol{R}\Delta\boldsymbol{u}(\tau) + (\Delta\dot{\boldsymbol{u}}(\tau))^\top\boldsymbol{S}\Delta\dot{\boldsymbol{u}}(\tau)\right)\mathrm{d}\tau \tag{12}$$

over a prediction horizon t_p subject to the trimmed LTI dynamics and the aforesaid constraints; the weight matrices $\boldsymbol{Q}, \boldsymbol{R},$ and $\boldsymbol{S}$ correspond to the state penalty, control penalty, and slew rate penalty, respectively.

As is well-known [18], computing the explicit MPC feedback for an LTI system amounts to solving parametric QP, whose solution is a continuous piecewise affine policy

$$\boldsymbol{\pi}(\boldsymbol{x}, t) = \boldsymbol{\Gamma}_j\boldsymbol{x} + \boldsymbol{\gamma}_j, \quad \text{for} \quad \boldsymbol{x}(t) \in \mathcal{P}_j \tag{13}$$

where $j = 1, 2, \ldots, \nu$, and the disjoint polytopic partition $\cup_{j=1}^{\nu}\mathcal{P}_j$ of the reach set of the closed-loop constrained LTI system depends on the time horizon in MPC synthesis and on the constraints (11) and the pair $(\boldsymbol{A}_{\text{trim}}, \boldsymbol{B}_{\text{trim}})$. We perform the explicit MPC synthesis offline and store the feedback gain matrix–vector pairs $\{\boldsymbol{\Gamma}_j, \boldsymbol{\gamma}_j\}_{j=1}^{\nu}$ and the polytopic partition $\cup_{j=1}^{\nu}\mathcal{P}_j$.

In Section V-A2, we will discuss a multi-lane driving scenario where different vehicles will have different initial joint state PDFs, thus different trim points, and consequently different MPC feedback policies.

III. Stochastic Reachability via Liouville PDE

The control system (1), (2) is of the standard form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)$ with the initial condition $\boldsymbol{x}_0 := \boldsymbol{x}(t = 0)$. We would like to compute the evolution of the transient joint state PDF $\rho(\boldsymbol{x}, t)$ subject to this controlled dynamics for a given nonrandomized control policy $\boldsymbol{u} = \boldsymbol{\pi}(\boldsymbol{x}, t)$, and uncertainties in the initial conditions $\boldsymbol{x}_0 \sim \rho_0$ (given).

We consider stochastic uncertainties in the initial states of both ego and nonego vehicles. In our framework, the notion of "initial condition uncertainty" is to be interpreted in a moving horizon context. Depending on the update rates of the on-board

[1] Over the horizon $[t_0, t_f]$, the trim input $\boldsymbol{u}_{\text{trim}}$ is assumed to be constant. In other words, $\boldsymbol{u}_{\text{trim}}$ has a deterministic value which is derived from the nominal operating point, the mean of the initial joint state PDF ρ_0.

sensor fusion or filtering algorithms, the ego would estimate the uncertainties in its own and other nonego vehicles' states at the beginning of these prediction horizons. The ego would then perform the predictions recursively in a moving horizon manner. Since this moving horizon length is typically 3–5 s, it is not too restrictive to suppose that the intent of the nonego vehicles will remain the same over that horizon.

Denoting the closed-loop dynamics as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\pi}(\boldsymbol{x}, t), t) =: \boldsymbol{g}(\boldsymbol{x}, t), \quad \boldsymbol{x}(t=0) = \boldsymbol{x}_0 \sim \rho_0 \tag{14}$$

the associated dynamics of the joint state PDF is governed by the Liouville PDE initial value problem (IVP)

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \boldsymbol{g}), \quad \rho(\boldsymbol{x}, t=0) = \rho_0(\boldsymbol{x}) \tag{15}$$

where $\nabla \cdot$ denotes the divergence operator w.r.t. the state vector $\boldsymbol{x}$, and the function ρ_0 is the joint PDF describing the initial condition uncertainties. The solution of the IVP (15) is the transient joint state PDF $\rho(\boldsymbol{x}, t)$ which characterizes the time-varying state uncertainties subject to the closed-loop dynamics (14). Fig. 4 shows the block diagram for the case of MPC feedback discussed in Section II-B2. For regularity of the solution of (15), see Appendix and the references therein. Our interest is to perform fast nonparametric computation of $\rho(\boldsymbol{x}, t)$.

Since the IVP (15) involves a first-order PDE, it is possible to solve the same using the method-of-characteristics. Specifically, it can be shown [10, Sec. 2.1] that the characteristic curves associated with (15) are the flows generated by (14), and hence must be nonintersecting due to standard existence uniqueness of the flow for an ODE of the form (14) with Lipschitz $\boldsymbol{g}$. This can be exploited to design a weighted scattered point cloud-based computation of the transient joint PDFs $\rho(\boldsymbol{x}, t)$ by integrating the following system of characteristic ODEs (see [10]) w.r.t. time t:

$$\begin{pmatrix} \dot{\boldsymbol{x}}^i \\ \dot{\rho}^i \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}(\boldsymbol{x}^i, t) \\ -\rho^i \nabla \cdot \boldsymbol{g}(\boldsymbol{x}^i, t) \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{x}^i(t=0) \\ \rho^i(t=0) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_0^i \\ \rho_0^i \end{pmatrix} \tag{16}$$

where the superscript $i = 1, \ldots, N$ denotes the sample index. In other words, one can first generate N random samples from the known initial joint state PDF ρ_0, and then the ODE (16) can be integrated along the characteristics for each of the N samples, resulting in a weighted scattered point cloud representation $\{\boldsymbol{x}^i(t), \rho^i(t)\}_{i=1}^N$ of the transient joint state PDF $\rho(\boldsymbol{x}, t)$. For $i = 1, \ldots, N$, the vector $\boldsymbol{x}^i(t)$ tells the location of the ith sample in the state space at time t, and the nonnegative weight $\rho^i(t)$ gives the value of the joint state PDF evaluated at that state space location at that time. Thus, at any given time $t > 0$, the higher (resp. lower) value of ρ^i quantifies the higher (resp. lower) likelihood of the state $\boldsymbol{x}^i$.

A. Computation of the Divergence of $\boldsymbol{g}$

We note that (16) involves the divergence $\nabla \cdot \boldsymbol{g}$. If the control is open loop, as will be the case in Section V-A1, then $\boldsymbol{g} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\pi}(t), t)$.

If the control is a feedback policy such as (13), as will be the case in Section V-A2, then using chain rule to compute the divergence of the composition $\boldsymbol{f} \circ \boldsymbol{\pi}(\boldsymbol{x}, t)$ at $\boldsymbol{x} = \boldsymbol{x}^i$ requires specifying the generalized gradient of $\boldsymbol{\pi}(\boldsymbol{x})$ (i.e., generalized gradient of each component of $\boldsymbol{\pi}(\boldsymbol{x})$ w.r.t. the vector $\boldsymbol{x}$) evaluated at a boundary point $\boldsymbol{x}^i$. This can be written in terms of the Clarke subdifferential [26] as follows. The Clarke subdifferential, in general, returns a set whose elements are the generalized gradients of the nonconvex nonsmooth function under consideration.

Since $\boldsymbol{\pi}(\boldsymbol{x})$ given by (13) is continuous (possibly nonconvex) piecewise affine with the index j in (13) denoting the "active partition index," it is Lipschitz continuous, and the Lipschitz constant is equal to $\max_{j=1,\ldots,\nu} \|\boldsymbol{\Gamma}_j\|_2$; e.g., [27, Ch. 2.2.3]. Thanks to this Lipschitz continuity, we can define its Clarke subdifferential as the convex hull of the limit of the gradients taken at differentiable points [26]. Specifically, the Clarke subdifferential of the ℓth component of $\boldsymbol{\pi}$, denoted as $\partial \pi_\ell(\boldsymbol{x})$, is given by

$$\partial \pi_\ell(\boldsymbol{x}) = \operatorname{conv}\Big(\lim_{r\to\infty} \nabla \pi_\ell(\boldsymbol{x}_r) \mid \boldsymbol{x}_r \to \boldsymbol{x}, \text{ and } \pi_\ell \text{ is differentiable at } \boldsymbol{x}_r\Big)$$

where $\operatorname{conv}(\cdot)$ denotes the convex hull, and ∇ denotes the standard Euclidean gradient. The set inside $\operatorname{conv}(\cdot)$ is often referred to as the Bouligand subdifferential. In our case, $\nabla \pi_\ell(\cdot)$ equals the ℓth row of the matrix $\boldsymbol{\Gamma}_j$, where the index j now runs over all possible neighbors (in the pointwise limit sense) of the partition boundary under consideration.

Our numerical implementation, consistent with the above, is done as follows. The divergence of $\boldsymbol{g}$ evaluated at any nonboundary point uses the "active" matrix–vector pairs $(\boldsymbol{\Gamma}_j, \boldsymbol{\gamma}_j)$ associated with the active partition $\mathcal{P}_j$ to which the sample state at that time belongs to. While numerically integrating the state ODE, if any state sample is at a partition boundary, then we use the last "interior point" matrix–vector pairs $(\boldsymbol{\Gamma}_j, \boldsymbol{\gamma}_j)$ in evaluating the divergence. This is theoretically consistent since the Clarke subdifferential's elements (i.e., generalized subgradients) include all neighboring matrices of that specific partition boundary in the limiting sense. We use variable time-step ODE integrator with fifth-order truncation error, and thus, the last "interior state" is indeed very close to the partition boundary, and thus approximates the limit in the generalized subgradient. Numerical computation of the Clarke subdifferentials in similar vein can be found in [28] and [29].

B. Features of the Proposed Computation

Several features of the scattered point cloud-based computation of the joint PDF $\rho(\boldsymbol{x}, t)$ described above make it particularly attractive for real-time occupancy prediction in automated driving scenarios as follows.

1) The computation along each characteristic curve is independent of the other, and hence the integration of (16) is massively parallelizable.
2) The computation is nonparametric and does not suffer from the moment closure problem. From a practical standpoint, one cannot guarantee *a priori* if the joint PDF evolution subject to (14) may admit a finite-dimensional sufficient statistic, and hence a nonparametric computation is desired. From an information theoretic viewpoint, nonparametric computation (as opposed to moment-based approximation [12], [13])

is preferred since the joint PDF subsumes all its moment information.

3) Unlike the traditional Monte Carlo methods, the proposed method explicitly computes $\rho^i(t)$ via the characteristic ODE in (16). This allows gridless computation. In contrast, the traditional Monte Carlo methods including density estimation algorithms [30] only propagate the states $\boldsymbol{x}^i(t)$ and then approximate the joint state PDF from the state samples alone. This function approximation requires discretizing the state space and suffers from "curse of dimensionality" [19]. The explicit time integration of the characteristic ODE circumvents this difficulty.

We remark here that it is possible to use the characteristic ODE (16) to write the solution of PDE IVP (15) in semi-analytical form

$$\rho(\boldsymbol{x},t) = \rho_0(\boldsymbol{x}_0(\boldsymbol{x},t))\exp\left(-\int_0^t (\nabla\cdot\boldsymbol{g})(\tau)\,\mathrm{d}\tau\right)$$

where $\boldsymbol{x}_0(\boldsymbol{x},t)$ is the inverse flow map for the closed-loop dynamics (14). In most vehicular models such as those described in Section II, one does not have analytical handle on this inverse flow map, and hence it is computationally beneficial to directly integrate the system of characteristic ODEs (16) along the sample trajectories. In the numerical simulations, we do so using the standard ode45 integrator in Matlab.

We also point out that even though the proposed framework is a finite sample computation, it allows evaluating the joint state PDF at a given time at any arbitrary state of interest. To see this, let $\boldsymbol{x}_q$ be a user-supplied arbitrary query state at time t, where none of our finitely many weighted samples [evolved by integrating (16)] has landed. We can backward integrate the dynamics from $\boldsymbol{x}_q$ at time t to $t=0$, determining the initial condition $\boldsymbol{x}_q^0$ from where the query point $\boldsymbol{x}_q$ could have come from. This can be done by any backward-stable numerical integrator. Since we know the initial joint PDF ρ_0 at $t=0$, we can test whether $\boldsymbol{x}_q^0$ belongs to the known support of ρ_0 or not. If not, then we conclude that the joint PDF value $\rho(t,\boldsymbol{x}_q)\equiv 0$. Otherwise, $\boldsymbol{x}_q^0$ is in the known support of ρ_0 with certain nonzero value $\rho_q^0 = \rho_0(\boldsymbol{x}_q^0)$ (this evaluation is possible, again, because the function ρ_0 is known). Then, we can forward integrate the dynamics and the method-of-characteristics ODE for the single-sample initial condition $(\boldsymbol{x}_q^0,\rho_q^0)$ over time $[0,t]$ to obtain $(\boldsymbol{x}_q,\rho(t,\boldsymbol{x}_q))$, where $\rho(t,\boldsymbol{x}_q)$ is the value of the joint PDF at time t at the query state $\boldsymbol{x}_q$, as desired. Note that since the choice of $\boldsymbol{x}_q$ is arbitrary, this procedure can be performed for as many query points as needed.

IV. Probabilistic Safety

Using the joint PDF computation framework explained thus far, we now propose an approach for estimating the risk of collision and finding the unsafe domain on the time horizon of interest, $[t_0,t_f]$. Suppose that at t_0, the ego vehicle has an estimate of its own and the nonego vehicle's state in terms of the respective beliefs or joint state PDFs: $\rho_0^{\text{ego}}(\boldsymbol{x})$ and $\rho_0^{\text{nonego}}(\boldsymbol{x})$, possibly computed via some sensor fusion algorithm using the particle filter or the extended Kalman filter. The ego vehicle would like to predict the transient beliefs $\rho^{\text{ego}}(\boldsymbol{x},t)$ and $\rho^{\text{nonego}}(\boldsymbol{x},t)$ for $t\in[t_0,t_f]$. The horizon length $t_f - t_0$ is typically few seconds, and the ego vehicle may repeat this predictive computation once new estimate (i.e., belief) arrives at the end of this horizon.

To compute $\rho^{\text{ego}}(\boldsymbol{x},t)$ and $\rho^{\text{nonego}}(\boldsymbol{x},t)$ for $t\in[t_0,t_f]$, conditioned on the initial beliefs $\rho_0^{\text{ego}}(\boldsymbol{x})$ and $\rho_0^{\text{nonego}}(\boldsymbol{x})$ at t_0, the ego vehicle utilizes its own and the nonego vehicle's closed-loop dynamics. The latter may only be an approximation since the exact nonego dynamics is not known to the ego vehicle, but the approximation may be refined and/or learned from observations in real-time. Using these predictions, we develop a collision avoidance approach detailed in Sections IV-A and IV-B.

Remark 1: Note that starting from the initial joint PDF at time $t=0$, we not only propagate the samples, but we also explicitly propagate the joint PDF values evaluated along each of those sample paths.

A. Collision Probability

At any time $t\in[t_0,t_f]$, nonzero collision probability results only if the distance between the supports of the (s,e_y) bivariate marginal PDFs of the ego and the nonego vehicles violates a safe lower bound. Let us denote these bivariate marginal PDFs as $\xi^{\text{ego}}(s^{\text{ego}},e_y^{\text{ego}},t)$ and $\xi^{\text{nonego}}(s^{\text{nonego}},e_y^{\text{nonego}},t)$, respectively. Making the standard assumption [20, Sec. 4] that the dynamics of the ego and the nonego vehicles are independent in $[t_0,t_f]$, the collision probability $p_{\text{collision}}(t)$ is given by

$$p_{\text{collision}}(t) = \int_{\mathcal{D}(t)} \xi^{\text{ego}}\left(s^{\text{ego}},e_y^{\text{ego}},t\right)\xi^{\text{nonego}}\left(s^{\text{nonego}},e_y^{\text{nonego}},t\right) \times \mathrm{d}s^{\text{ego}}\mathrm{d}e_y^{\text{ego}}\mathrm{d}s^{\text{nonego}}\mathrm{d}e_y^{\text{nonego}}. \quad (17)$$

We define the time-varying domain $\mathcal{D}(t)$ in (17) as

$$\mathcal{D}(t) := \left\{\left(s^{\text{ego}},e_y^{\text{ego}},s^{\text{nonego}},e_y^{\text{nonego}}\right)\in\mathbb{R}^4 \mid \left|s^{\text{ego}}-s^{\text{nonego}}\right|\le L_s,\ \left|e_y^{\text{ego}}-e_y^{\text{nonego}}\right|\le L_{e_y}\right\} \quad (18)$$

wherein L_s and $L_{e_y}\ge 0$ are some suitable "safe distances" in the longitudinal and lateral directions, respectively. For instance, if ℓ_0 and w_0 denote the nominal length and width of the vehicles, and ε_s and ε_{e_y} are the factors of safety, then one may set

$$L_s = \ell_0 + \varepsilon_s, \quad L_{e_y} = w_0 + \varepsilon_{e_y}.$$

Thus, $\mathcal{D}(t)$ can be seen as the "unsafe set" at time t. For related works on accounting the vehicle geometry in collision probability computation, see [31], [32].

Note that we can rewrite (17) as

$$p_{\text{collision}}(t) = \int_{\mathbb{R}^4} \mathbb{1}_{\mathcal{D}(t)}\,\xi^{\text{ego}}\left(s^{\text{ego}},e_y^{\text{ego}},t\right) \times\xi^{\text{nonego}}\left(s^{\text{nonego}},e_y^{\text{nonego}},t\right)\mathrm{d}s^{\text{ego}}\mathrm{d}e_y^{\text{ego}}\mathrm{d}s^{\text{nonego}}\mathrm{d}e_y^{\text{nonego}}. \quad (19)$$

In [20], the integral (19) was approximated as an empirical (i.e., uniform) average using the state samples. For our computational framework, in addition to the state samples, the

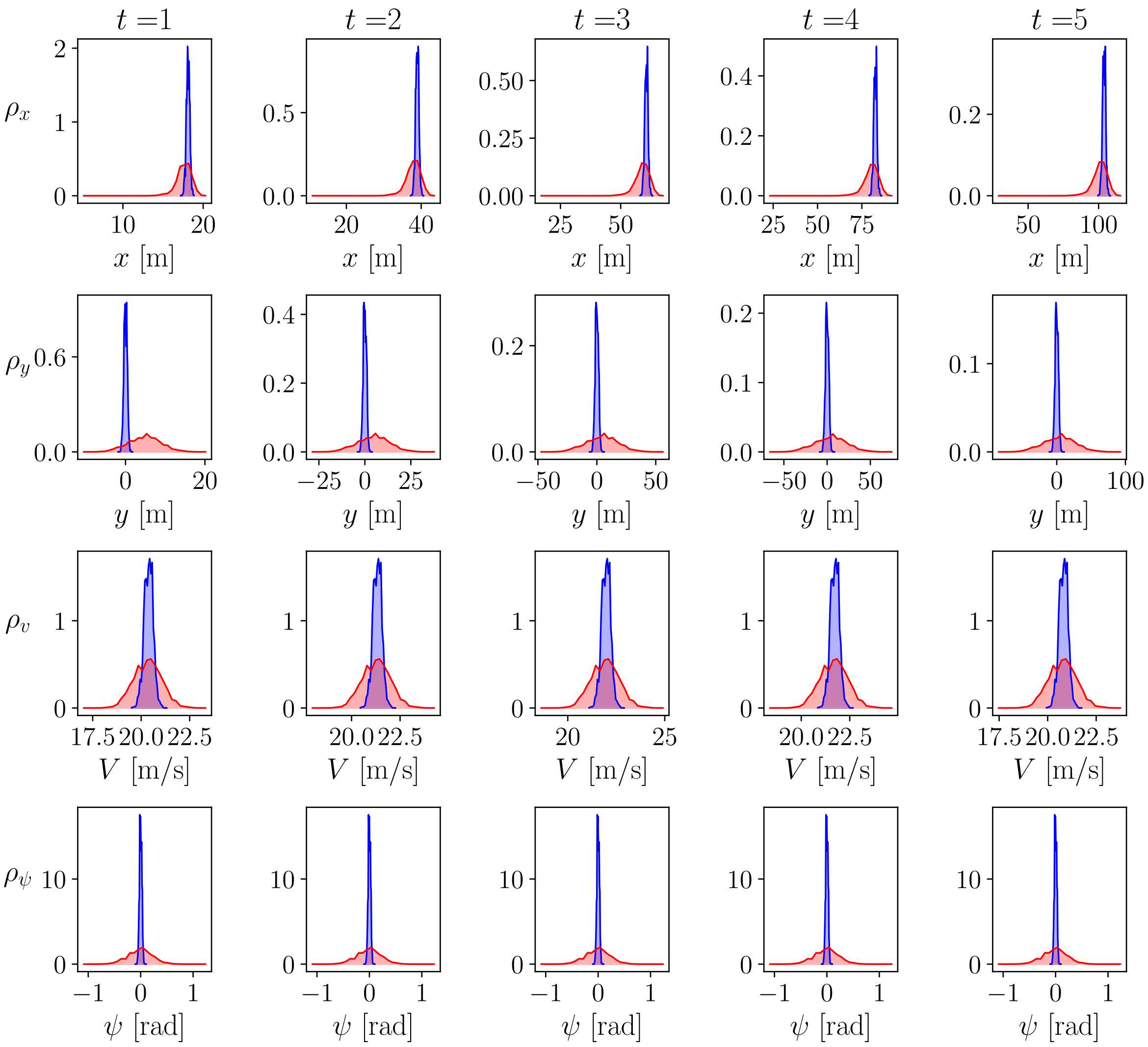


Fig. 5. For the simulation setup detailed in Section V-A1, the evolution of the transient univariate marginal state PDFs for vehicle 1 (ego, in blue) and vehicle 2 (nonego, in red).

transient bivariate marginals ξ^{ego} and ξ^{nonego} are numerically available, which allow us to approximate (19) as a weighted average.

Specifically, the unsafe set $\mathcal{D}(t)$ is a set of points where the intersection of the supports of the bivariate (e_y, s) marginals for the two vehicles is nonzero. In our context, a support is represented via a finite collection of sampling points in the (e_y, s) coordinates. As suggested by (18), we compute $\mathcal{D}(t)$ by comparing the coordinate-wise minimum and maximum of support points of the marginal of each car added with the size of the vehicles and a safety factor. We then interpolate (using cubic splines) the marginal PDFs over the $\mathcal{D}(t)$ thus computed and estimate the collision probability using (19).

B. Probabilistic Safety via Wasserstein Barycenters

The natural question for the ego vehicle is how should it plan its probabilistically safest path in the (s, e_y) variables over the horizon $[t_0, t_f]$ taking into account the stochastic predictions $\xi^{\text{nonego }(i)}(s^{\text{nonego }(i)}, e_y^{\text{nonego }(i)}, t)$, $i \in \{1, 2, \dots, N_{\text{nonego}}\}$, $t \in [t_0, t_f]$, where N_{nonego} is the number of neighboring nonego vehicles. Specifically, the ego would like to plan a path in the (s, e_y) marginal PDF or belief space, which is maximally separated from each predicted belief trajectory $\xi^{\text{nonego }(i)}(s^{\text{nonego }(i)}, e_y^{\text{nonego }(i)}, t)$, $i \in \{1, 2, \dots, N_{\text{nonego}}\}$ in its neighborhood, thus minimizing the chance of collision with any of the nonego vehicles for all $t \in [t_0, t_f]$.

For a given bivariate marginal PDF $\xi(s, e_y)$, and for $i \in \{1, 2\}$, let

$$W_i(t) := W\left(\xi(s, e_y), \xi^{\text{nonego }(i)}\left(s^{\text{nonego }(i)}, e_y^{\text{nonego }(i)}, t\right)\right)$$

denote the two-Wasserstein distance between ξ and $\xi^{\text{nonego }(i)}$ at time t. The two-Wasserstein distance W is a metric in the space of PDFs (or probability measures, in general), and it quantifies the minimal cost of transporting or reshaping one PDF to another; e.g., [21, Ch. 7]. For any two probability measures ς_0, ς_1 with finite second moments, each supported on $\mathcal{M} \subseteq \mathbb{R}^d$, the two-Wasserstein distance W is defined as

$$W(\varsigma_0, \varsigma_1) := \left[\inf_{\wp \in \Pi(\varsigma_0, \varsigma_1)} \int_{\mathcal{M}\times\mathcal{M}} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \, \mathrm{d}\wp(\boldsymbol{x}, \boldsymbol{y})\right]^{\frac{1}{2}} \quad (20)$$

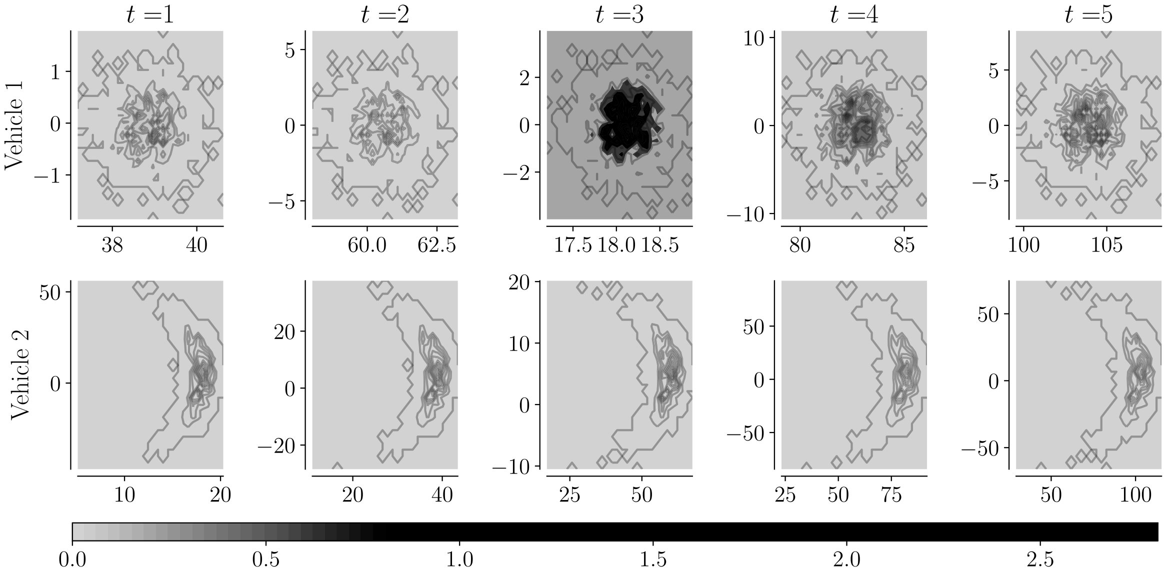


Fig. 6. For the simulation setup detailed in Section V-A1, the evolution of the transient bivariate marginal state PDF in (x, y) variables for vehicle 1 (ego) and vehicle 2 (nonego). The colorbar (light hue = small, dark hue = large) shows the values of the bivariate marginals.

where $\Pi(\varsigma_0, \varsigma_1)$ is the set of joint probability measures on $\mathcal{M}\times\mathcal{M}$ with marginals ς_0, ς_1. Then, maximizing the distance from the nonego marginal beliefs, as motivated above, amounts to computing the barycentric [22] trajectory for $t \in [t_0, t_f]$ in the belief space

$$\xi^{\text{bary}}\big(s, e_y, t\big) := \underset{\xi\in\mathcal{P}_2(\mathbb{R}^2)}{\arg\inf} \sum_{i=1}^{N_{\text{nonego}}} \lambda_i(t)(W_i(t))^2 \tag{21}$$

where the weights $\lambda_i(t)$ codify the ego vehicle's confidence in its stochastic prediction $\xi^{\text{nonego }(i)}(s^{\text{nonego }(i)}, e_y^{\text{nonego }(i)}, t)$, and satisfy $\lambda_i(t) \geq 0, \sum_{i=1}^{N_{\text{nonego}}} \lambda_i(t) = 1$. In (21), $\mathcal{P}_2(\mathbb{R}^2)$ denotes the collection of all joint PDFs over $\mathbb{R}^2$ with finite second moments.

Intuitively, the barycenter $\xi^{\text{bary}}(s, e_y, t)$ is the weighted average of the beliefs

$$\big\{\xi^{\text{nonego }(i)}\big(s^{\text{nonego }(i)}, e_y^{\text{nonego }(i)}, t\big)\big\}_{i=1}^{N_{\text{nonego}}}.$$

In the absence of uncertainties, the beliefs $\xi^{\text{nonego }(i)}$ would be Dirac deltas, and the barycentric PDF trajectory would reduce to a Dirac delta trajectory supported on the graph of the weighted average of the nonego state trajectories.

V. Numerical Simulations

In the following, we elucidate the overall computational framework by first giving some illustrative examples in Section V-A, and then consider a case study in Section V-B to highlight the scope of the proposed method. Section V-C provides a comparison of the numerical performance of the proposed method with the standard Monte Carlo. All simulations were performed in Matlab R2019b running on iMac with 3.4-GHz Intel Core i5 processor and 8-GB memory.

A. Illustrative Examples

We consider a two-lane unidirectional highway driving scenario with two vehicles, one ego and another nonego, driving in the adjacent lanes with stochastic uncertainties in their states. The ego vehicle has better estimation accuracy of its own states compared with the same for the state of the nonego vehicle. Consequently, the uncertainties in the state of the nonego vehicle at any time, as perceived by the ego vehicle, are expected to be more than the uncertainties in its own state at that time. We often refer to the ego vehicle as "vehicle 1," and the nonego vehicle as "vehicle 2." We illustrate the proposed framework in Section IV using the models described in Section II. The computation is carried on for a single horizon $[t_0, t_f]$, set $t_0 \equiv 0$ without loss of generality, and refer to $\boldsymbol{x}(t_0)$ as the "initial" condition.

1) Stochastic Reachability With the Kinematic Bicycle Model: Let us consider the case where both the ego and the nonego vehicles have the dynamics as in Section II-A with $\ell_{\text{front}} = 1$ m and $\ell_{\text{rear}} = 1.5$ m. Suppose that at $t_0 = 0$, their initial joint state PDFs are

$$\rho_0^{\text{ego}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \quad \rho_0^{\text{nonego}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \tag{22}$$

with

$$\begin{aligned}\boldsymbol{\mu}_1 &= (0, 0, 20, 0)^\top, \quad \boldsymbol{\Sigma}_1 = \text{diag}\big(10^{-2}, 10^{-2}, 10^{-1}, 10^{-3}\big)\\ \boldsymbol{\mu}_2 &= (0, 5, 20, 0)^\top, \quad \boldsymbol{\Sigma}_2 = \text{diag}\big(10^{-2}, 10^{-1}, 1, 10^{-1}\big).\end{aligned}$$

For all $t \in [t_0, t_f] \equiv [0, 5]$, we set the acceleration input $a_c(t) \equiv \sin(t)$ and the front steering wheel angle input $\delta(t) \equiv 0$ for both the vehicles. We generate $N = 1000$ random samples for each of the above two initial joint state PDFs, evaluate the resulting samples exactly at the respective known initial joint state PDFs, and then propagate the two weighted scattered point clouds $\{\boldsymbol{x}^{\text{ego},i}(t), \rho^{\text{ego},i}(t)\}_{i=1}^{N}$ and $\{\boldsymbol{x}^{\text{nonego},i}(t), \rho^{\text{nonego},i}(t)\}_{i=1}^{N}$ as described in Section III.

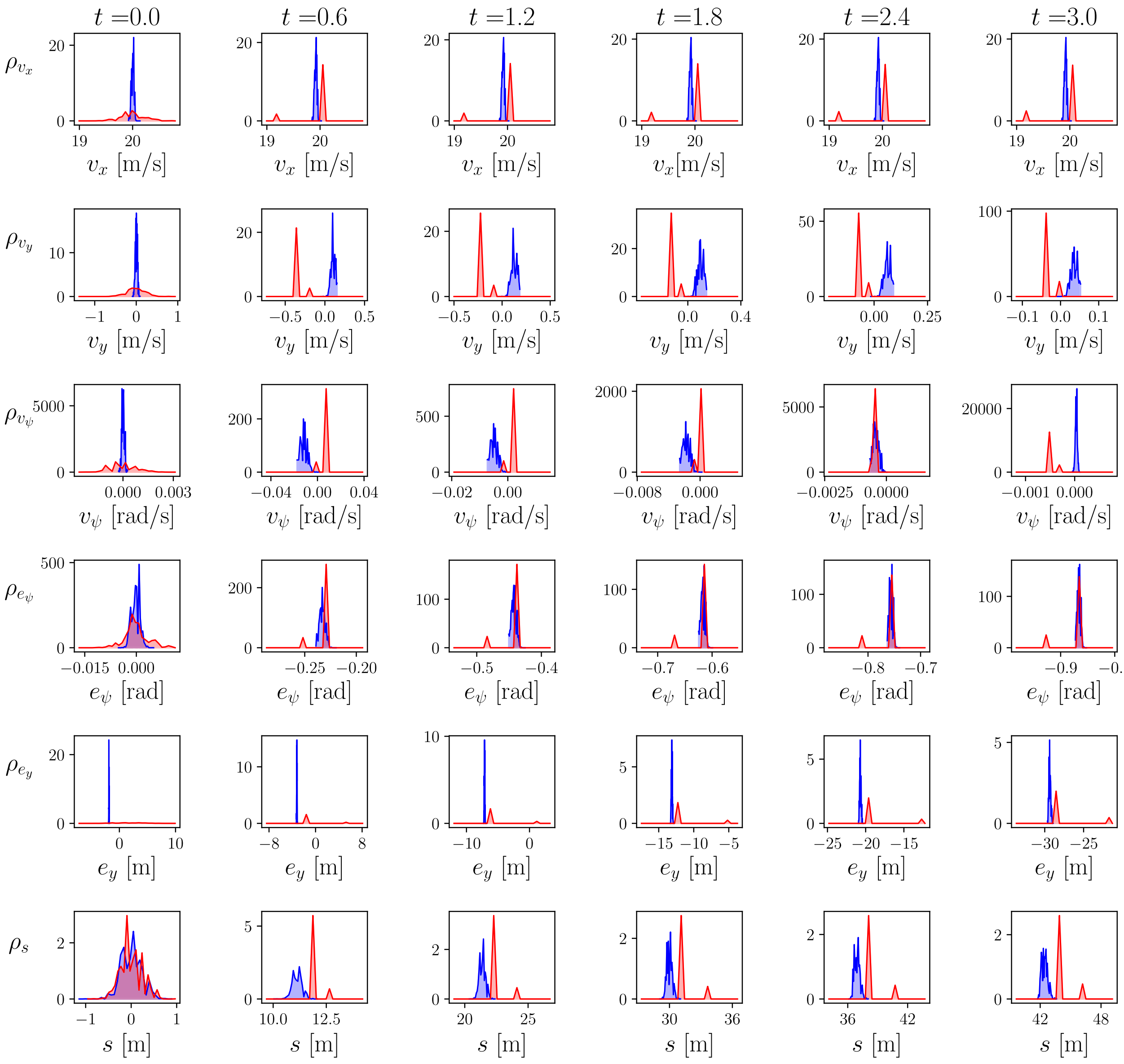


Fig. 7. For the simulation setup detailed in Section V-A2, the evolution of the transient univariate marginal state PDFs for vehicle 1 (ego, in blue) and vehicle 2 (nonego, in red).

As explained before, this joint PDF propagation via the method-of-characteristics is a gridless computation.

Fig. 5 shows the evolution of the univariate marginal state PDFs for the two vehicles, obtained by numerically integrating (see [10, Sec. 5.2]) the weighted scattered point cloud data described above. The evolution of the bivariate marginal PDFs in (x, y) variables is depicted in Fig. 6. These plots indeed show that the uncertainties in (nonego) vehicle 2's state at any given time has more dispersion than (ego) vehicle 1's state.

For this simulation (with no parallel computing), the computational time for joint PDF propagation for each vehicle took approximately 0.7805 s.

2) Stochastic Reachability With the Dynamic Bicycle Model: We now consider the case where both the ego and nonego vehicles have the dynamics as in Section II-B with trim-linearized explicit MPC feedback satisfying the constraints (11). We set $\kappa = 0.02$. As mentioned in Section II-B2, the computation of the trim depends on the initial joint state PDF due to the constraint (11b). We suppose that the initial joint state PDFs are of the form (22) with

$$\begin{aligned}\boldsymbol{\mu}_1 &= (20, 0, 0, 0, -1.85, 0)^\top\\ \boldsymbol{\Sigma}_1 &= \operatorname{diag}\big(0.11, 0.11, 1.24\times 10^{-8}, 2.78\times 10^{-6}, 10^{-2}, 0.11\big)\\ \boldsymbol{\mu}_2 &= (20, 0, 0, 0, 1.85, 0)^\top\\ \boldsymbol{\Sigma}_2 &= \operatorname{diag}\big(1.11, 1.11, 1.24\times 10^{-7}, 2.5\times 10^{-5}, 1.11, 0.11\big).\end{aligned}$$

We perform the offline linearized explicit MPC synthesis using the MPC toolbox [17] with sampling time 0.01 s, prediction horizon $t_p = 3$ s, and control horizon of length 2 s. The weight matrices in (12) are chosen as below so that the vehicle closed-loop response for speed control and lane centering has the desired settling time of $O(1)$ s

$$\boldsymbol{Q} = 10\boldsymbol{I}_6, \quad \boldsymbol{R} = \boldsymbol{I}_3, \quad \boldsymbol{S} = 10^{-1}\boldsymbol{I}_3.$$

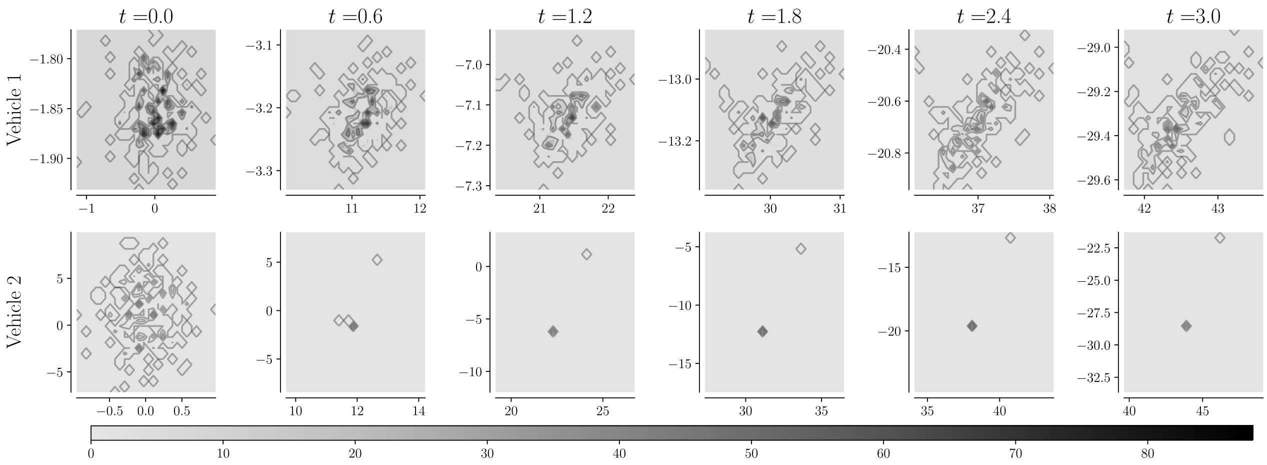


Fig. 8. For the simulation setup detailed in Section V-A2, the evolution of the transient bivariate marginal state PDF in (s, e_y) variables for vehicle 1 (ego) and vehicle 2 (nonego). The colorbar (light hue = small, dark hue = large) shows the values of the bivariate marginals.

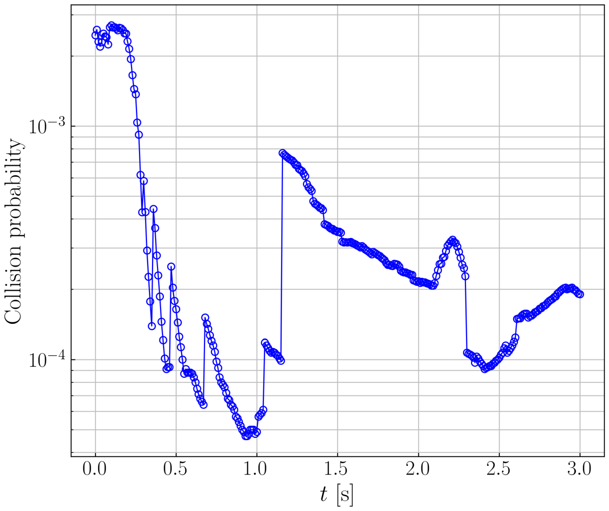


Fig. 9. Collision probability $p_{\text{collision}}(t)$ between ego and nonego vehicle for the simulation setup detailed in Section V-A2.

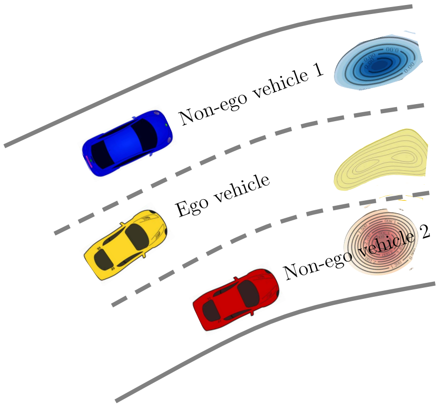


Fig. 10. Schematic of the three-lane highway driving scenario described in Section V-B. As shown, the ego vehicle is in the middle lane, while the two nonego vehicles are in its left and right lanes, respectively. The three contour plots show the respective bivariate marginals ξ^{ego}, $\xi^{\text{nonego}\,(1)}$, and $\xi^{\text{nonego}\,(2)}$ in (s, e_y) variables at an instance t within the prediction horizon $[t_0, t_f]$.

The resulting continuous piecewise affine feedback (13) admits a disjoint polytopic partition $\cup_{j=1}^{3610}\mathcal{P}_j$ in the six-dimensional state space. In our simulation environment mentioned before, the offline MPC synthesis takes approximately 15 min.

We generate $N = 200$ random samples for each of the two initial joint state PDFs, and as before, evaluate the resulting samples exactly at the respective known initial joint state PDFs. For $t \in [t_0, t_f] \equiv [0, 3]$, we propagate the two weighted scattered point clouds $\{\boldsymbol{x}^{\text{ego},i}(t), \rho^{\text{ego},i}(t)\}_{i=1}^{N}$ and $\{\boldsymbol{x}^{\text{nonego},i}(t), \rho^{\text{nonego},i}(t)\}_{i=1}^{N}$ for the closed-loop dynamics with respective explicit MPC feedback. Thus, the joint PDF computation depends on the feedback gain matrix–vector pairs $\{\boldsymbol{\Gamma}_j, \boldsymbol{\gamma}_j\}_{j=1}^{\nu}$ and the polytopic partition $\cup_{j=1}^{\nu}\mathcal{P}_j$ (in our simulation, $\nu = 3610$).

Fig. 7 shows the evolution of the univariate marginal state PDFs for the two vehicles, obtained by numerically integrating the weighted scattered point cloud data described above. The evolution of the bivariate marginal PDFs in (s, e_y) variables is depicted in Fig. 8. From these plots, we observe that the uncertainties in (nonego) vehicle 2's state at any given time has more dispersion than (ego) vehicle 1's state, as expected. We next use the bivariate (s, e_y) marginal PDFs to compute the time-varying collision probability over the simulation interval $[t_0, t_f] \equiv [0, 3]$, for $L_s = 4.36$ m and $L_y = 2.44$ m, using (19). With the simulation setup described above, Fig. 9 shows $p_{\text{collision}}(t)$ for $t \in [t_0, t_f]$.

For this simulation with no parallel computation, the computational time for joint PDF propagation for each vehicle took approximately 287.0702 s. Since our joint PDF propagation is completely parallelizable, we expect that implementing our algorithm in a realistic automated driving platform will enjoy a significant speed up in runtime by leveraging the available parallel computation resources, either on-board or from the enterprise cloud.

B. Case Study: Barycentric Planning in the Belief Space

To illustrate the scope of the proposed computational framework, we now consider a three-lane (unidirectional) highway

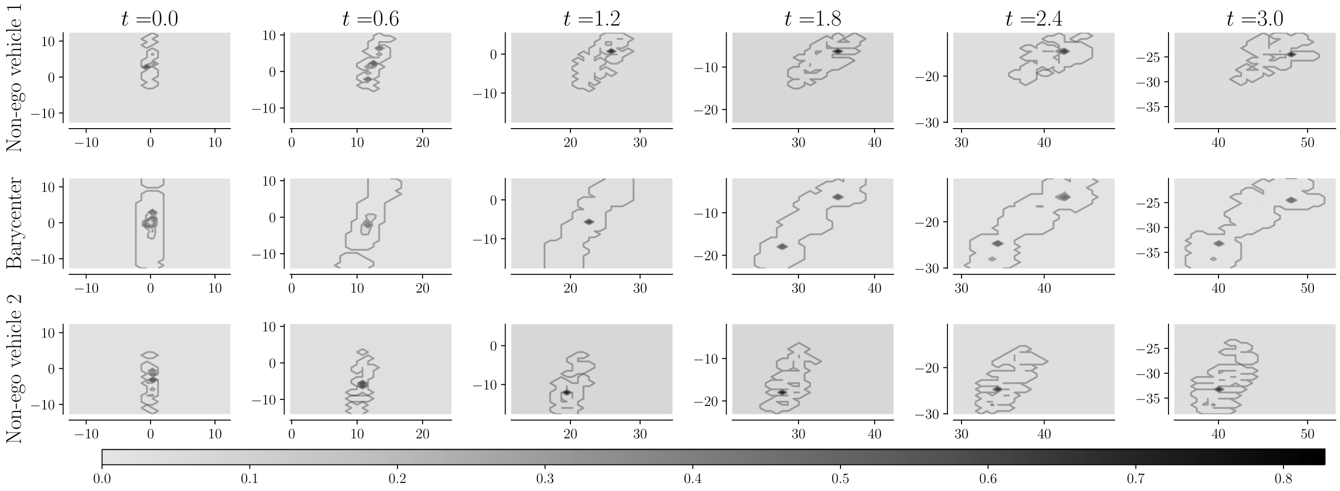


Fig. 11. For the simulation setup detailed in Section V-B, the evolution of the transient bivariate marginal state PDFs in (s, e_y) variables for the nonego vehicle 1 (top row), the nonego vehicle 2 (bottom row), and their barycenters (middle row). The colorbar (light hue = small, dark hue = large) shows the values of these bivariate marginals.

driving scenario as shown in Fig. 10. The ego vehicle, shown in the middle lane in Fig. 10, at time t_0, computes the evolution of its own joint $\rho^{\text{ego}}(\boldsymbol{x}, t)$ and bivariate marginal $\xi^{\text{ego}}(s^{\text{ego}}, e_y^{\text{ego}}, t)$, as well as its prediction of the same for its neighboring nonego vehicles, i.e.,

$$\rho^{\text{nonego }(i)}(\boldsymbol{x}, t),\ \xi^{\text{nonego }(i)}\left(s^{\text{nonego }(i)}, e_y^{\text{nonego }(i)}, t\right), \quad i \in \{1, 2\}$$

over a prediction horizon $[t_0, t_f]$. Then, the ideas in Section V-A2 allow the ego to predict the time-varying collision probabilities between each ego–nonego pair. Moreover, if the ego would like to continue in its (here, middle) lane, then it needs to be maximally separated from both the predicted belief trajectories $\xi^{\text{nonego }(i)}(s^{\text{nonego }(i)}, e_y^{\text{nonego }(i)}, t)$, $i \in \{1, 2\}$, to minimize the chance of collision with either of nonegos for all $t \in [t_0, t_f]$, i.e., it needs to follow the barycenter of marginal PDFs $\xi^{\text{nonego }(1)}$, $\xi^{\text{nonego }(2)}$:

$$\xi^{\text{bary}}\left(s, e_y, t\right) := \underset{\xi \in \mathcal{P}_2\left(\mathbb{R}^2\right)}{\arg\inf} \sum_{i=1}^{2} \lambda_i(t)(W_i(t))^2$$

with $\lambda_1(t) = \lambda_2(t) = (1/2)$, and $W_i(t)$ is the two-Wasserstein distance of marginal PDFs $\xi^{\text{nonego }(1)}$, $\xi^{\text{nonego }(2)}$ defined in (20). Now, to numerically simulate the scenario shown in Fig. 10, let

$$\begin{aligned} \rho_0^{\text{ego}}(\boldsymbol{x}) &= \mathcal{N}\left(\boldsymbol{\mu}_0^{\text{ego}}, \boldsymbol{\Sigma}_0^{\text{ego}}\right) \\ \rho_0^{\text{nonego }(i)}(\boldsymbol{x}) &= \mathcal{N}\left(\boldsymbol{\mu}_0^{\text{nonego }(i)}, \boldsymbol{\Sigma}_0^{\text{nonego }(i)}\right), \quad i \in \{1, 2\} \end{aligned}$$

with

$$\begin{aligned} \boldsymbol{\mu}_0^{\text{ego}} &= (20, 0, 0, 0, 0, 0)^\top \\ \boldsymbol{\Sigma}_0^{\text{ego}} &= \operatorname{diag}\big(1.11 \times 10^{-3}, 1.11 \times 10^{-3}, 1.23 \times 10^{-8}, \\ &\qquad 2.78 \times 10^{-6}, 1.11 \times 10^{-3}, 1.11 \times 10^{-1}\big) \\ \boldsymbol{\mu}_0^{\text{nonego }(i)} &= (20, 0, 0, 0, (-1)^{i+1} \times 3.7, 0)^\top \\ \boldsymbol{\Sigma}_0^{\text{nonego }(1)} &= \boldsymbol{\Sigma}_0^{\text{nonego }(2)} = \operatorname{diag}\big(0.11, 0.11, 1.23 \times 10^{-6}, \\ &\qquad 2.5 \times 10^{-5}, 11.11, 0.11\big). \end{aligned}$$

Following Section II-B2, we compute the v_x velocity-hold trim conditions for each of the three vehicles and perform the offline linearized MPC synthesis about the same as in Section V-A2. We use $N = 200$ random samples for each of the three initial joint state PDFs given above, and as in Section V-A2, generate weighted scattered point clouds $\{\boldsymbol{x}^{\text{ego},i}(t), \rho^{\text{ego},i}(t)\}_{i=1}^{N}$, $\{\boldsymbol{x}^{\text{nonego }1,i}(t), \rho^{\text{nonego }1,i}(t)\}_{i=1}^{N}$, $\{\boldsymbol{x}^{\text{nonego }2,i}(t), \rho^{\text{nonego }2,i}(t)\}_{i=1}^{N}$, which are then propagated via the Liouville PDE in Section III over $t \in [t_0, t_f] \equiv [0, 3]$ with the respective explicit MPC feedbacks in the loop. As before, we then use the resulting joint PDFs to compute the time-varying bivariate marginals $\xi^{\text{ego}}, \xi^{\text{nonego }1}, \xi^{\text{nonego }2}$.

To compute the barycentric marginals in (21), we use the multi-marginal entropy-regularized optimal transport using the iterative Bregman projection proposed in [23, Sec. 4.2] with $\lambda_1(t) = \lambda_2(t) = 1/2$ for all $t \in [t_0, t_f]$. In our MATLAB implementation, the average computational time to obtain the transient barycenters was approximately 1.82 s. For recent works improving the computational complexity of the iterative Bregman projection for the Wasserstein barycenter computation, we refer the readers to [33]–[35].

Fig. 11 shows the evolution of the transient marginal state PDFs in (s, e_y) variables for the nonego vehicles, as well as their barycenters. Fig. 12 highlights that the collision probabilities between the barycenter–nonego pairs are negligible compared with the collision probabilities between the ego–nonego pairs, as expected. The values of L_s and L_y are as given in Section V-A2. Also shown in Fig. 12 are the ego versus nonego collision probabilities when the ego (starting from stochastic initial conditions in the middle lane, as before) turns either to its left or to its right lane.

C. Comparison With the Monte Carlo Method

We now provide a quantitative comparison of the proposed Liouville PDE method with the standard Monte Carlo. To this end, we compare the computational time and accuracy for the ego vehicle's joint PDF prediction via these two methods in the context of the simulation setup in Section V-A1.

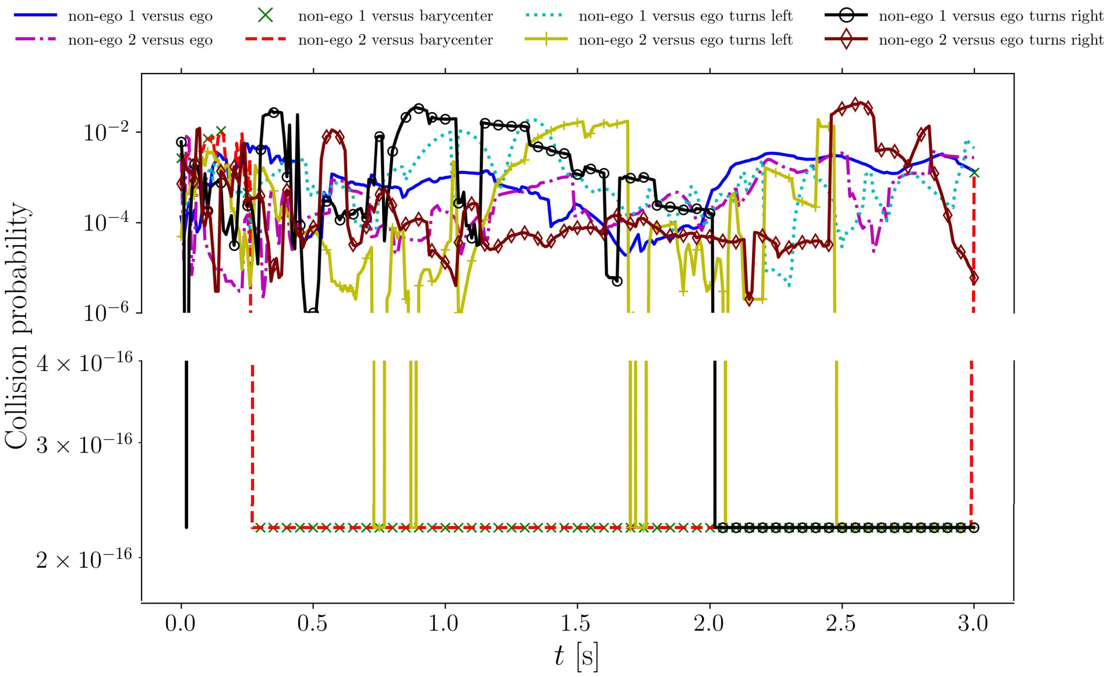


Fig. 12. Time-varying collision probabilities for the simulation setup detailed in Section V-B, plotted with a broken vertical axis as some collision probabilities drop to (machine precision) zero.

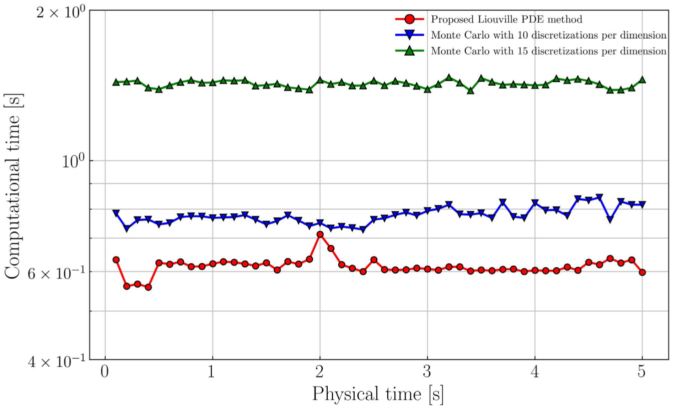


Fig. 13. Comparison of the computational times for calculating the transient joint PDFs of the ego vehicle in Section V-A1 using the proposed method and the standard Monte Carlo.

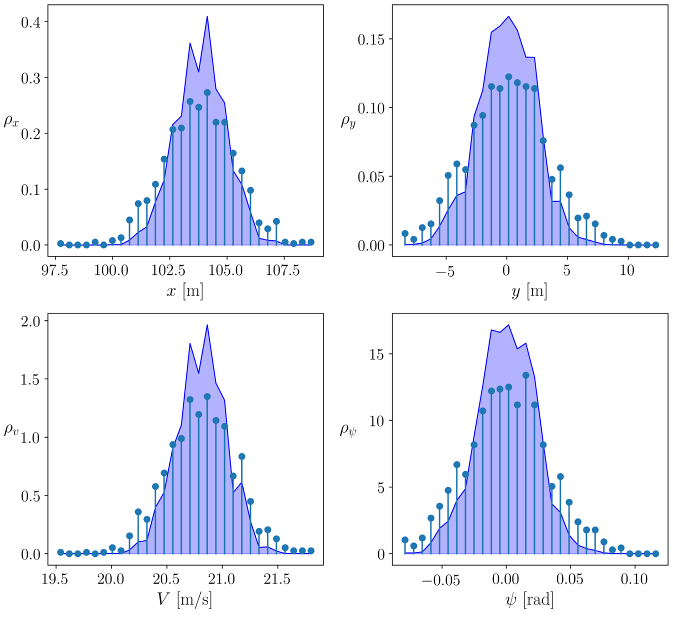


Fig. 14. Comparison of the univariate marginal PDFs of the ego vehicle in Section V-A1 at $t = 5$ s, using the proposed method (solid lines with filled area) and the standard Monte Carlo (stem plot).

Specifically, Fig. 13 shows that the PDE characteristic-based PDF propagation, as outlined in Section III, is computationally faster compared with the standard Monte Carlo method. The latter requires constructing a grid over the (in this case, 4-D) state space to approximate the joint state PDF. Even though the state samples can be propagated in a meshless manner, grids are necessary in the Monte Carlo method for constructing histograms representing the piecewise constant function approximation for instantaneous joint PDFs. As mentioned in Section III, the proposed method–not being a function approximation–obviates the construction of grid, but requires integrating an extra ODE per sample; see (16).

In Fig. 13, we used $N = 1000$ samples with the same parameters and initial joint PDFs detailed in Section V-A1. For the same data, two Monte Carlo simulations were performed: one with 10 and another with 15 uniform discretizations per state dimension, between the transient minimum and maximum along each state coordinate. This resulted in histograms supported over 10^4 and 15^4 orthotopes, respectively. Fig. 13 also shows that finer discretization for Monte Carlo requires more computational time, as expected.

Fig. 14 shows the comparison of the univariate marginal PDFs for the ego vehicle at $t = 5$ computed via the proposed method and the standard Monte Carlo. While the overall trends are similar, we point out that the solid lines in Fig. 14 are obtained by numerically integrating the joint PDF values via (16), which are exact modulo the floating point arithmetic. In contrast, the Monte Carlo marginals are obtained by numerically integrating the piecewise constant approximations of the joint PDFs, which are in turn obtained by constructing histograms over the propagated states. Of course, finer discretization in Monte Carlo can increase its accuracy but at the expense of computational time; cf. Fig. 13.

VI. Conclusion

We have proposed a density-based stochastic reachability computation framework for occupancy prediction in automated driving. The main idea is to evolve the joint state PDFs along the characteristic curves of the underlying transport PDE via gridless computation. These transient joint PDFs are then used to compute the marginals and the time-varying collision probabilities for the closed-loop vehicle dynamics. We envision that such computational results can be passed to a higher level decision-making module such as driver assistance or early collision warning systems. Using these predictions to design feedback control that steers the ego vehicle's stochastic state to a probabilistically safe state has been pursued in our follow-up work [36]. Although our simulations in this article focused on the multi-lane highway driving scenarios, the overall framework can be adapted to applications that involve interactions with other vehicles and pedestrians in complex traffic environment. Another possible direction of future research is to incorporate the process noise (see [24], [25]) in closed-loop vehicle dynamics.

Appendix
Regularity of the Solution of (15)

The solution $\rho(\boldsymbol{x},t)$ for the Liouville PDE IVP (15), where $t \in [0,T]$ and $\boldsymbol{x} \in \mathcal{X}$ (the state space), is understood in the weak sense. This means that for all compactly supported smooth test functions $\phi \in C_c^\infty(\mathcal{X},[0,T])$, the function $\rho(\boldsymbol{x},t)$ satisfies

$$\int_0^T \int_{\mathcal{X}} \left(\rho \frac{\partial \phi}{\partial t} + \rho \boldsymbol{g} \cdot \nabla \phi\right) \mathrm{d}\boldsymbol{x}\, \mathrm{d}t + \int_{\mathcal{X}} \rho_0(\boldsymbol{x}) \phi(\boldsymbol{x},0) \mathrm{d}\boldsymbol{x} = 0.$$

The regularity of the transient joint PDF $\rho(\boldsymbol{x},t)$ that solves (15) in the weak sense depends on the regularity of the initial joint PDF $\rho_0(\boldsymbol{x})$ and that of the closed-loop vector field $\boldsymbol{g}(\boldsymbol{x},t)$. To discuss such results, we need some notations.

Denote the space of square integrable functions on $\mathcal{X}$ as $L^2(\mathcal{X})$. Given a positive integer m, let $H^m(\mathcal{X})$ denote the Sobolev space of functions in $L^2(\mathcal{X})$ for which all mixed partial derivatives up to order m exist in the weak sense and they are in $L^2(\mathcal{X})$. A regularity result [37, Th. 3.19] is that if $\rho_0 \in H^m(\mathcal{X})$, and $\boldsymbol{g}$ is integrable and is Lipschitz w.r.t. the state, then $\rho \in C(H^m(\mathcal{X}),[0,T])$. The results by DiPerna and Lions [38] also allow unbounded vector fields which roughly speaking grow at most linearly w.r.t. the norm of state at infinity, and get similar regularity estimates; see also [39]. The marginals have at least the same spatial smoothness (in the reduced states) as the joint. In our simulations, both $\boldsymbol{f}$ and $\boldsymbol{\pi}$ are Lipschitz w.r.t. state, and in particular, $\rho_0 \in H^2(\mathcal{X})$. Thus, the marginals are in $C(H^2,[0,T])$.

References

[1] A. Carvalho, S. Lefévre, G. Schildbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty—A control perspective," *Eur. J. Control*, vol. 24, pp. 14–32, Jul. 2015.

[2] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 1, pp. 5–17, Mar. 2018.

[3] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 299–310, Jun. 2009.

[4] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1237–1247, Dec. 2011.

[5] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2005, pp. 319–324.

[6] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 137–147, Mar. 2008.

[7] S. Fünfgeld, M. Holzäpfel, M. Frey, and F. Gauterin, "Stochastic forecasting of vehicle dynamics using sequential Monte Carlo simulation," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 2, pp. 111–122, Jun. 2017.

[8] M. Ehrendorfer, "The Liouville equation and its potential usefulness for the prediction of forecast skill. Part I: Theory," *Monthly Weather Rev.*, vol. 122, no. 4, pp. 703–713, Apr. 1994.

[9] A. Halder and R. Bhattacharya, "Beyond Monte Carlo: A computational framework for uncertainty propagation in planetary entry, descent and landing," in *Proc. AIAA Guid., Navigat., Control Conf.*, Aug. 2010, p. 8029.

[10] A. Halder and R. Bhattacharya, "Dispersion analysis in hypersonic flight during planetary entry using stochastic Liouville equation," *J. Guid., Control, Dyn.*, vol. 34, no. 2, pp. 459–474, Mar. 2011.

[11] A. Halder, K. Lee, and R. Bhattacharya, "Optimal transport approach for probabilistic robustness analysis of F-16 controllers," *J. Guid., Control, Dyn.*, vol. 38, no. 10, pp. 1935–1946, Oct. 2015.

[12] A. Jasour, A. Wang, and B. C. Williams, "Moment-based exact uncertainty propagation through nonlinear stochastic autonomous systems," 2021, *arXiv:2101.12490.*

[13] A. Wang, X. Huang, A. Jasour, and B. Williams, "Fast risk assessment for autonomous vehicles using learned models of agent futures," 2020, *arXiv:2005.13458*.

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

[15] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "Predictive control of an autonomous ground vehicle using an iterative linearization approach," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2335–2340.

[16] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 1094–1099.

[17] A. Bemporad, M. Morari, and N. L. Ricker, *Model Predictive Control Toolbox: User's Guide*. Natick, MA, USA: The Mathworks, 2010.

[18] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.

[19] R. E. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[20] A. Lambert, D. Gruyer, G. S. Pierre, and A. N. Ndjeng, "Collision probability assessment for speed control," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 1043–1048.

[21] C. Villani, *Topics in Optimal Transportation*, no. 58. Providence, RI, USA: American Mathematical Society, 2003.

[22] M. Agueh and G. Carlier, "Barycenters in the Wasserstein space," *SIAM J. Math. Anal.*, vol. 43, no. 2, pp. 904–924, 2011.

[23] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative Bregman projections for regularized transportation problems," *SIAM J. Sci. Comput.*, vol. 37, no. 2, pp. A1111–A1138, Jan. 2015.

[24] K. F. Caluya and A. Halder, “Proximal recursion for solving the fokker-Planck equation,” in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 4098–4103.

[25] K. F. Caluya and A. Halder, “Gradient flow algorithms for density propagation in stochastic systems,” *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 3991–4004, Oct. 2020, doi: 10.1109/TAC.2019.2951348.

[26] F. H. Clarke, “Generalized gradients and applications,” *Trans. Amer. Math. Soc.*, vol. 205, pp. 247–262, Apr. 1975.

[27] S. Scholtes, *Introduction to Piecewise Differentiable Equations*. New York, NY, USA: Springer, 2012.

[28] K. Kubota, “Enumeration of subdifferentials of piecewise linear functions with abs-normal form,” *Optim. Methods Softw.*, vol. 33, nos. 4–6, pp. 1156–1172, Nov. 2018.

[29] J. V. Burke, A. S. Lewis, and M. L. Overton, “Approximating subdifferentials by random sampling of gradients,” *Math. Oper. Res.*, vol. 27, no. 3, pp. 567–584, Aug. 2002.

[30] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Evanston, IL, USA: Routledge, 2008.

[31] N. E. Du Toit and J. W. Burdick, “Probabilistic collision checking with chance constraints,” *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 809–815, Mar. 2011.

[32] A. P. Vinod and M. M. K. Oishi, “Probabilistic occupancy via forward stochastic reachability for Markov jump affine systems,” *IEEE Trans. Autom. Control*, vol. 66, no. 7, pp. 3068–3083, Jul. 2021.

[33] A. Kroshnin, N. Tupitsa, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and C. Uribe, “On the complexity of approximating Wasserstein barycenters,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3530–3540.

[34] T. Lin, N. Ho, X. Chen, M. Cuturi, and M. Jordan, “Fixed-support Wasserstein barycenters: Computational hardness and fast algorithm,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–35.

[35] D. Dvinskikh and D. Tiapkin, “Improved complexity bounds in Wasserstein barycenter problem,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1738–1746.

[36] S. Haddad, K. F. Caluya, A. Halder, and B. Singh, “Prediction and optimal feedback steering of probability density functions for safe automated driving,” *IEEE Control Syst. Lett.*, vol. 5, no. 6, pp. 2168–2173, Dec. 2021.

[37] H. Bahouri, J.-Y. Chemin, and R. Danchin, *Fourier Analysis and Nonlinear Partial Differential Equations*, vol. 343. Berlin, Germany: Springer, 2011.

[38] R. J. DiPerna and P. L. Lions, “Ordinary differential equations, transport theory and sobolev spaces,” *Inventiones Mathematicae*, vol. 98, no. 3, pp. 511–547, Oct. 1989.

[39] L. Ambrosio, “Transport equation and Cauchy problem for BV vector fields,” *Inventiones Mathematicae*, vol. 158, no. 2, pp. 227–260, Nov. 2004.

Shadi Haddad received the M.S. degree in mechanical engineering from the University of Tehran, Tehran, Iran, in 2018. She is currently pursuing the Ph.D. degree with the Department of Applied Mathematics, University of California at Santa Cruz, Santa Cruz, CA, USA.

Her research interests include control and optimization.

Abhishek Halder (Senior Member, IEEE) received the bachelor’s and master’s degrees from the Indian Institute of Technology Kharagpur (IIT Kharagpur), Kharagpur, India, in 2008, and the Ph.D. degree from Texas A&M University, College Station, TX, USA, in 2014, all in aerospace engineering.

He is currently an Assistant Professor with the Department of Applied Mathematics, and an Affiliated Faculty with the Department of Electrical and Computer Engineering, University of California at Santa Cruz, Santa Cruz, CA, USA. Before that, he held Post-Doctoral Researcher positions at the Department of Mechanical and Aerospace Engineering, University of California at Irvine, Irvine, CA, USA, and the Department of Electrical and Computer Engineering, Texas A&M University. His research interests include stochastic systems, control, and optimization with application focus on large-scale cyber-physical systems.

Baljeet Singh received the bachelor’s and master’s degrees in mechanical engineering from the Indian Institute of Technology Bombay (IIT Bombay), Mumbai, India, in 2003, and the Ph.D. degree in aerospace engineering from Texas A&M University, College Station, TX, USA, in 2010.

He is currently a Research Scientist of automated driving with the Ford Motor Company’s Research and Advanced Engineering Division, Palo Alto, CA, USA. In the past, he worked as a Senior Algorithms Engineer with Cymer, LLC., San Diego, CA, USA, and as a Control Tools Developer with MathWorks, Inc., Natick, MA, USA. His research interests include decision-making and motion planning under uncertainty for autonomous driving.