Non-Gaussian Chance-Constrained Trajectory Planning for Autonomous Vehicles Under Agent Uncertainty

Allen Wang, Ashkan Jasour, and Brian C. Williams

***Abstract*—Agent behavior is arguably the greatest source of uncertainty in trajectory planning for autonomous vehicles. This problem has motivated significant amounts of work in the behavior prediction community on learning rich distributions of the future states and actions of agents. However, most current works on chance-constrained trajectory planning under agent or obstacle uncertainty either assume Gaussian uncertainty or linear constraints, which is limiting, or requires sampling, which can be computationally intractable to encode in an optimization problem. In this letter, we extend the state-of-the-art by presenting a methodology to upper-bound chance-constraints defined by polynomials and mixture models with potentially non-Gaussian components. Our method achieves its generality by using statistical moments of the distributions in concentration inequalities to upper-bound the probability of constraint violation. With this method, optimization-based trajectory planners can plan trajectories that are chance-constrained with respect to a wide range of distributions representing predictions of agent future positions. In experiments, we show that the resulting optimization problem can be solved with state-of-the-art nonlinear program solvers to plan trajectories fast enough for use online.**

***Index Terms*—Probability and Statistical Methods, Motion and Path Planning, Optimization and Optimal Control, Robot Safety, Intelligent Transportation Systems.**

I. Introduction

In ORDER for autonomous vehicles to drive safely on public roads, they need to plan trajectories that take into account predictions of future positions of other agents (e.g. human driven vehicles, pedestrians, cyclists). However, predictions are inherently uncertain, especially predictions of human behavior. This fact is motivating significant amounts of work in the behavior prediction community to develop methods that predict *distributions* of future agent states and actions, often using a deep neural network (DNN). For example, [1] trains a conditional variational autoencoder to generate samples of possible future trajectories; the DNN in this case essentially becomes the distribution from which samples can be drawn. [2]–[4] learn Gaussian mixture models (GMMs) for the agents' future positions to handle both uncertainty in high level decisions, which tends to be multi-modal, and uncertainty in execution, which tends to be continuous.

While work in behavior prediction can now generate rich distributions of future agent states and actions, most current works in chance-constrained trajectory planning only address the unimodal case and either make Gaussian assumptions or require sampling [5]–[10]. We argue that handling non-Gaussian distributions is important because, for example, almost any distribution for agent action propagated through nonlinear dynamics models will result in non-Gaussian position distributions. Gaussians also have unbounded supports; this is unrealistic as the reachable set of agents is bounded by physical laws in reality. To handle non-Gaussian uncertainty, some prior works take a sampling-based approach. This often makes optimization computationally intractable because thousands of constraints need to be introduced to even enforce chance-constraints on the order of 10^{-2} for any practical problem [6], [10], [11]. To address non-Gaussian uncertainty in obstacle or agent positions without sampling, recent works apply the Cantelli or Chebyshev inequality, but they are currently restricted to linear constraints [12]–[15]. For non-Gaussian uncertainty, sums-of-squares programming has been applied to the problem of trajectory tracking for nonlinear systems and risk assessment in the presence of non-convex obstacles, but current computational limitations restrict it to applications amenable to leveraging offline computation [16]–[20].

Statement of Contributions: In this letter, we present a general chance-constrained trajectory planning formulation for autonomous vehicles that can handle mixtures of non-Gaussian distributions of agent position and, unlike many prior works which make point mass assumptions, accounts for the sizes of the ego vehicle and agents. This is enabled by a general methodology we develop for enforcing non-Gaussian polynomial chance-constraints using concentration inequalities, extending the prior art which can only handle linear constraints. This methodology makes heavy use of symbolic algebra, and we develop and provide a Python package, AlgebraicMoments,[1] to implement

Manuscript received February 24, 2020; accepted July 2, 2020. Date of publication July 21, 2020; date of current version July 31, 2020. This letter was recommended for publication by Associate Editor H. Kurniawati and Editor N. Amato upon evaluation of the reviewers' comments. This work was supported in part by the Boeing under Grant 6943358 and in part by Masdar Institute under Grant 6938857. The work of Allen Wang was supported by NSF Graduate Research Fellowship. Embotech generously supplied academic FORCES Pro licenses. *(Corresponding author: Allen Wang.)*

The authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 01239 USA (e-mail: allenw@mit.edu; jasour@mit.edu; williams@mit.edu).

Digital Object Identifier 10.1109/LRA.2020.3010755

[1] The source code with examples can be found at github.com/allen-adastra/algebraic_moments

it. Given a constraint defined as a polynomial in a random vector, AlgebraicMoments can generate a closed form expression that upper-bounds the probability of the event in terms of statistical moments of the random vector. It can even directly generate MATLAB or Python code to compute the risk bound, given the necessary inputs. Since this approach only depends on statistical moments of the distributions, it can apply to a wide range of prediction distributions including non-Gaussian mixture models of future agent positions. In numerical experiments, we show how our formulation, when solved with advanced interior-point methods, can be used to plan trajectories with horizons of 5 seconds fast enough for use online. While this letter focuses on the problem of trajectory planning for autonomous vehicles under agent uncertainty, the general methodology should be applicable to motion and trajectory planning problems with constraint uncertainty in many other application domains.

II. Representation of Agent Predictions

A. Assumptions

We assume a behavior prediction system provides the distribution of future positions for an agent over a T step horizon, $\mathbf{g}_{1:T}$, in a fixed frame. The distributions can be either unimodal or a mixture of non-Gaussian random vectors. The distributions $\mathbf{g}_t = [g_{x_t}, g_{y_t}]^T$ are assumed to be either independent across time in the unimodal case or independent across time conditioned on the discrete mode in the mixture model case. This is a common assumption used in state-of-the-art behavior prediction systems; thus, it does not significantly restrict our method's range of applicability [2], [3], [21]. We also make the additional assumption that the predicted distributions do not change w.r.t. changes in the ego vehicle trajectory, as accounting for the change requires having the behavior prediction system in the planning loop. In practice, it may be more effective to alternate between the planner and prediction systems by using planned trajectories in the prediction system to generate a new distribution, but we do not explore this interaction in this letter.

B. Computing Moments of Distributions

In some cases, the statistical moments of distributions are known in closed form. In other cases, statistical moments of a distribution can be rapidly computed by applying automatic or numerical differentiation to its characteristic function (CF). This is a very general approach to computing moments of distributions as CFs always exist, and, from a more practical standpoint, there are extensive tables of CFs for common distributions [22]. Letting X denote a random variable and $\Phi_X(t)$ denote its characteristic function, the n_{th} moment of X can be computed by:

$$\mathbb{E}[X^n] = i^{-n}\left[\frac{d^n}{dt^n}\Phi_X(t)\right]_{t=0} \tag{1}$$

Similarly, moments of a random vector, $\mathbf{w}$, can be computed via partial differentiation of its joint characteristic function $\Phi_{\mathbf{w}}(t)$, although we note catalogues of CFs are less extensive for multivariate distributions. Alternatively, moments may also be estimated with Monte Carlo type methods; this is useful for approaches where the DNN is a distribution from which sampled trajectories are drawn.

C. Statistics of Mixture Models

A mixture model is a rich way of expressing multi-modal uncertainty by combining multiple continuous distributions [23]. In this work, we work with random vectors and define random vector mixture models as:

Definition 1: An n component random vector mixture model $\mathbf{w}$ is a random vector with components $\mathbf{w}_i$ and mixture weights w_i for $i = 1, \ldots, n$ s.t. $\sum_{i=1}^n w_i = 1$. Its pdf $f_{\mathbf{w}}$ is related to those of its components $f_{\mathbf{w}_i}$ by:

$$f_{\mathbf{w}}(\cdot) = \sum_{i=1}^{n} w_i f_{\mathbf{w}_i}(\cdot) \tag{2}$$

A useful property of mixture models is its statistics can be computed in terms of statistics of its components [23]. Proposition 1 states this fact in the general case and only makes the mild assumption that $g(\cdot)$ is measurable, allowing it to apply to most functions used in practice. As a simple corollary, by letting $g(\mathbf{w})$ be some moment of $\mathbf{w}$, moments of $\mathbf{w}$ can be expressed as the weighted sum of the moments of its components.

Proposition 1: For any n component random vector mixture model $\mathbf{w}$ with components $\mathbf{w}_i$ with mixture weights w_i and any measurable function g, we have that $\mathbb{E}[g(\mathbf{w})] = \sum_{i=1}^n w_i \mathbb{E}[g(\mathbf{w}_i)]$.

Proof: By the law of the unconscious statistician and the definition of the mixture model pdf:

$$\mathbb{E}[g(\mathbf{w})] = \int g(\mathbf{x}) \sum_{i=1}^{n} w_i f_{\mathbf{w}_i}(\mathbf{x}) d\mathbf{x} \tag{3}$$

Applying the linearity of expectation we then have:

$$\int g(\mathbf{x}) \sum_{i=1}^{n} w_i f_{\mathbf{w}_i}(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{n} w_i \int g(\mathbf{x}) f_{\mathbf{w}_i}(\mathbf{x}) d\mathbf{x} \tag{4}$$

$$= \sum_{i=1}^{n} w_i \mathbb{E}[g(\mathbf{w}_i)] \tag{5}$$

■

III. Problem Formulation

A. Definition of Risk

We define risk in a way that accounts for the size of the agent and ego vehicle. The general idea is to fit circles of radius r to an agent and constrain the probability that the centers of the circles are inside an appropriately scaled "collision ellipsoid" around the vehicle. Fig. 1 illustrates an example; note that if the centers of the circles are not in the ellipsoid, the vehicles are not in collision.

We essentially treat each circle as a separate agent. We will define risk in the "planned body frames," which are depicted by Fig. 2. While the prediction is given in some fixed frame, we eventually show in Section IV-D how moments in the planned

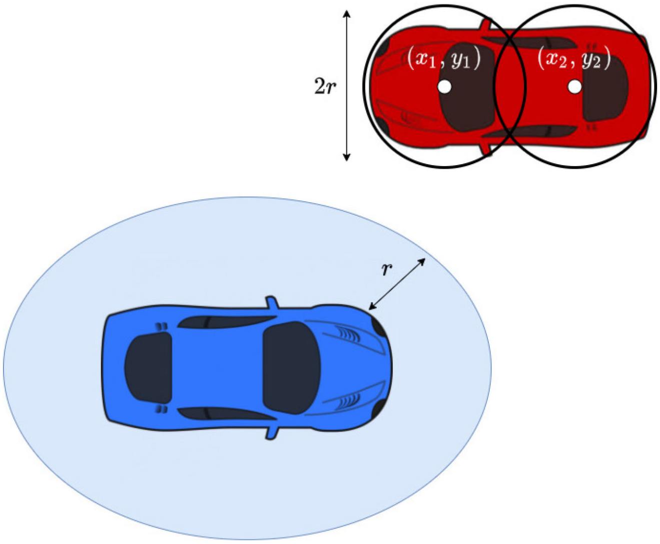

Fig. 1. An example showing a collision ellipsoid around the ego vehicle and corresponding circles around the agent. Note that if the points (x_i, y_i) are not in the ellipsoid, then the vehicles are not in collision.

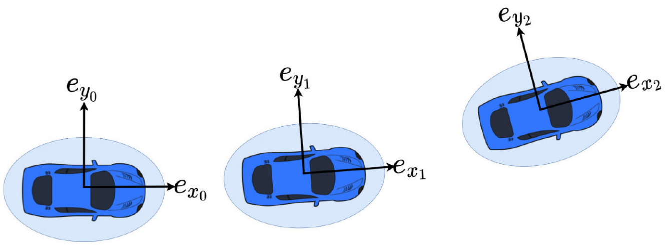

Fig. 2. A planned trajectory for the ego vehicle along with the "planned body frames" and collision ellipsoids drawn around the vehicle.

body frames can be expressed as a function of the fixed frame moments and the planned ego vehicle pose. In the body frame, the ellipsoid is the set:

$$\{\mathbf{x} : \mathbf{x}^T Q \mathbf{x} \le 1\} \tag{6}$$

Where Q is a constant 2×2 positive definite matrix. The following notation will be used for quadratic forms as it better reflects the polynomial nature of quadratic forms, and many results in this letter will be for polynomials in general:

$$Q(\mathbf{x}) := \mathbf{x}^T Q \mathbf{x} \tag{7}$$

We denote the distribution of the i_{th} agent in the planned frame at time t as $\mathbf{a}_{t,i}$; recall this distribution is ultimately a function of the global frame prediction $\mathbf{g}_{t,i}$ and the planned ego vehicle pose. Risk along the horizon is thus defined as:

$$\mathcal{R} := \mathbb{P}\left(\bigcup_{i=1}^{n_a}\bigcup_{t=1}^{T}\{Q(\mathbf{a}_{t,i}) \le 1\}\right) \tag{8}$$

However, evaluating (8) without excessive conservatism is a non-trivial problem itself meriting a separate treatment. For this letter, we simply apply Boole's Inequality which states:

$$\mathcal{R} \le \sum_{i=1}^{n_a}\sum_{t=1}^{T}\mathbb{P}(Q(\mathbf{a}_{t,i}) \le 1) \tag{9}$$

To upper bound the total risk $\mathcal{R}$ with some Δ, we define a risk-allocation $\epsilon_{t,i}$ s.t.:

$$\sum_{i=1}^{n_a}\sum_{t=1}^{T}\epsilon_{t,i} \le \Delta \tag{10}$$

And then upper-bound the marginal probabilities as such:

$$\mathbb{P}(Q(\mathbf{a}_{t,i}) \le 1) \le \epsilon_{t,i} \tag{11}$$

It is possible to encode the risk allocation as decision variables in an optimization problem, but doing so can often be computationally intractable. In this letter, we solve the trajectory planning problem with a fixed risk allocation; other works have presented approaches for "outer loops" that optimize the risk allocation [15], [24], [25]. In addition, throughout the rest of this letter, we present mathematical statements for the single agent case for the sake of notational simplicity; it is straight-forward to simply repeat the chance-constraint in the multi-agent case.

B. The Trajectory Planning Problem

In this letter, we derive results for the general cc-trajectory planning problem defined below.

$$\min_{\mathbf{x}_{1:T},\mathbf{u}_{0:T}} \quad c(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}) \tag{12a}$$

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), t \in [T-1] \tag{12b}$$

$$\mathbb{P}(Q(\mathbf{a}_t) \le 1) \le \epsilon_t, t \in [T] \tag{12c}$$

$$\mathbf{u}_{min} \le \mathbf{u}_t \le \mathbf{u}_{max}, t \in [T] \tag{12d}$$

$$\mathbf{x}_{min} \le \mathbf{x}_t \le \mathbf{x}_{max}, t \in [T] \tag{12e}$$

Where $\mathbf{x}_t$ is the state vector, $\mathbf{u}_t$ is the control vector, c is some cost function, f is a discrete time system modeling ego vehicle dynamics, $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ are control limits and $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ are state limits. We assume that the ego vehicle dynamics are deterministic as modern feedback control systems for autonomous vehicles are effective at tracking trajectories with positional error on the order of ten centimeters [26], [27]. Thus, the only difference between this problem and standard deterministic trajectory planning formulations is the chance constraint (12c) which ensures that the probability of the vehicle colliding with an agent is no more than ϵ at each time step. Section IV presents our approach to enforcing a risk-bound on this chance constraint.

IV. Bounding Non-Gaussian Polynomial Chance-Constraints

In this section, we present a general methodology for bounding polynomial non-Gaussian chance-constraints; that is, for a random vector $\mathbf{w}$ and polynomial p, we present a method to establish a bound on the probability:

$$\mathbb{P}(p(\mathbf{w}) \le 0) \tag{13}$$

Recall that the chance-constraint in our trajectory planning problem (12c) takes this form as quadratic forms are polynomials. By viewing $p(\mathbf{w})$ as a random variable, (13) is simply the cumulative distribution function (cdf) of $p(\mathbf{w})$ evaluated at 0. Unfortunately, even in the relatively simple case where $\mathbf{w}$ is a multivariate Gaussian and p is a quadratic, $p(\mathbf{w})$ does not have a closed form cdf [28]. In general, it can very challenging to characterize the cdfs of distributions that arise from nonlinear transformations, so our approach is to leverage one-tailed concentration inequalities which bound (13) using the mean and variance of $p(\mathbf{w})$. Sub section IV-A begins by showing how moments of the random variable $p(\mathbf{w})$ can be expressed in closed form in terms of moments of $\mathbf{w}$. This provides us with a way to compute the mean and variance of $p(\mathbf{w})$ given the distribution of $\mathbf{w}$. Subsection IV-B then introduces several concentration inequalities that can be used to bound (13) using the mean and variance of $p(\mathbf{w})$. We can directly apply these approaches to mixture models by computing the moments of the mixture models with the approach from Subsection II-C. However, we show in Subsection IV-C that tighter bounds can be achieved by instead bounding the components of the mixture model. Finally, Subsection IV-D applies these techniques to bound the chance-constraint (12c).

A. Moments of Polynomials in Random Vectors

An important property of p being a polynomial is that the n_{th} moment of $p(\mathbf{w})$ can be computed as the weighted sum of moments of $\mathbf{w}$. This is true because $p(\mathbf{w})^n$ is, itself, a polynomial to which the linearity of expectation can be applied. To see this, consider the following simple example where $p(\mathbf{w}) = w_1^2 + w_2^2$:

$$\mathbb{E}[p(\mathbf{w})^2] = \mathbb{E}[w_1^4 + 2w_1^2 w_2^2 + w_2^4] \tag{14a}$$

$$= \mathbb{E}[w_1^4] + 2\mathbb{E}[w_1^2 w_2^2] + \mathbb{E}[w_2^4] \tag{14b}$$

To state the general case, we adopt multi-index notation which allows us to much more succinctly express moments of random vectors. For example, $\mathbb{E}[w_1^2 w_2^2]$ can be represented with the vector $\mathbf{w}$ and a multi-index $\alpha = (2, 2, 0, \ldots, 0)$. Letting $\mathbf{w}$ be an n dimensional random vector, we can express any moment of $\mathbf{w}$ with a multi-index $\alpha \in \mathbb{N}^n$ as such where α_i is the i_{th} element of α:

$$\mathbb{E}[\mathbf{w}^\alpha] := \mathbb{E}\left[\prod_{i=1}^{n} w_i^{\alpha_i}\right] \tag{15}$$

Proposition 2 uses multi-index notation to express the idea that any moment of $p(\mathbf{w})$ can be expressed in terms of moments of $\mathbf{w}$. AlgebraicMoments can be used to easily derive expressions of the form (16), and it even leverages independence in the random vector $\mathbf{w}$ to further decompose terms of the form $\mathbb{E}[\mathbf{w}^\alpha]$.

Proposition 2: For a n dimensional random vector $\mathbf{w}$, a polynomial p and $m \in \mathbb{N}$, there exists a set of multi-indices $\mathcal{A} \subset \mathbb{N}^n$ and coefficients $C_\mathcal{A} = \{c_\alpha \in \mathbb{R} : \alpha \in \mathcal{A}\}$ s.t.:

$$\mathbb{E}[p(\mathbf{w})^m] = \sum_{\alpha \in \mathcal{A}} c_\alpha \mathbb{E}[\mathbf{w}^\alpha] \tag{16}$$

Proof: Since p is a polynomial, $p(\mathbf{w})^m$ is also a polynomial in $\mathbf{w}$ since the ring of polynomials is closed under multiplication. Thus, we have the existence of $\mathcal{A}$ and $C_\mathcal{A}$ s.t. $p(\mathbf{w})^m = \sum_{\alpha \in \mathcal{A}} c_\alpha \mathbf{w}^\alpha$. Applying the expectation operator to both sides and the linearity of expectation, we arrive at the result. ■

TABLE I
ASSUMPTIONS REQUIRED FOR CONCENTRATION INEQUALITIES

Inequality	**Assumptions**
Cantelli	$p(\mathbf{w})$ has finite mean + variance
Vysochanskij-Petunin (VP)	Cantelli assumptions + unimodal
Gauss	VP assumptions + symmetric pdf

B. Bounding Risk With Concentration Inequalities

The prior subsection shows how moments of $p(\mathbf{w})$ can be expressed in terms of moments of $\mathbf{w}$; in this section we show how the mean and variance of $p(\mathbf{w})$, $\mu_{p(\mathbf{w})}$ and $\sigma^2_{p(\mathbf{w})}$, can be used to bound risk using concentration inequalities. We start with Cantelli's inequality [29], also known as the one-tailed Chebyshev Inequality, which bounds the probability of constraint violation as such:

$$\mathbb{P}(p(\mathbf{w}) \leq 0) \begin{cases} \leq \frac{\sigma^2_{p(\mathbf{w})}}{\sigma^2_{p(\mathbf{w})} + \mu^2_{p(\mathbf{w})}} & \mu_{p(\mathbf{w})} \geq 0 \\ \geq 1 - \frac{\sigma^2_{p(\mathbf{w})}}{\sigma^2_{p(\mathbf{w})} + \mu^2_{p(\mathbf{w})}} & \mu_{p(\mathbf{w})} < 0 \end{cases} \tag{17}$$

However, for many applications, Cantelli's inequality can be excessively conservative; in fact, it is often sharp only for discrete distributions. By making additional mild assumptions, we can arrive at tighter bounds with the Vysochanskij-Petunin (VP) and Gauss inequalities, which are very similar [30]–[32]. Our proposed strategy is to adopt the tightest inequality for which $p(\mathbf{w})$ meets the given assumptions; the table below summarizes the assumptions for each inequality.Since the inequalities are very similar, we simplify future notation by defining the Conc$[\cdot]$ operator to denote the tightest appropriate concentration inequality and Conc$^*[\cdot]$ to denote the corresponding necessary condition for the inequality to hold. In practice, the practitioner would have to select the correct inequality to use on a case-by-case basis.

$$\text{Conc}[p(\mathbf{w})] = \begin{cases} \frac{\sigma^2_{p(\mathbf{w})}}{\sigma^2_{p(\mathbf{w})} + \mu^2_{p(\mathbf{w})}} & \text{Cantelli Holds} \\ \frac{4}{9}\frac{\sigma^2_{p(\mathbf{w})}}{\sigma^2_{p(\mathbf{w})} + \mu^2_{p(\mathbf{w})}} & \text{VP Holds} \\ \frac{2}{9}\frac{\sigma^2_{p(\mathbf{w})}}{\mu^2_{p(\mathbf{w})}} & \text{Gauss Holds} \end{cases} \tag{18}$$

$$\text{Conc}^*[p(\mathbf{w})] = \begin{cases} -\mu_{p(\mathbf{w})} & \text{Cantelli Holds} \\ -\mu_{p(\mathbf{w})} + \sqrt{\frac{5}{3}}\sigma_{p(\mathbf{w})} & \text{VP Holds} \\ -\mu_{p(\mathbf{w})} + \frac{2}{3}\sigma_{p(\mathbf{w})} & \text{Gauss Holds} \end{cases} \tag{19}$$

Thus, an ϵ chance-constrain can be bounded in an optimization problem by enforcing the constraints:

$$\text{Conc}[p(\mathbf{w})] \leq \epsilon \tag{20}$$

$$\text{Conc}^*[p(\mathbf{w})] \leq 0 \tag{21}$$

Note that the Conc* conditions are not particularly restrictive, as they are only violated when risk is relatively high. In fact, the Conc* condition only breaks when the upper-bound is at least $1, 1/6$, or $1/2$ for the Cantelli, VP, and Gauss cases respectively, which is well above values usually specified for chance-constraints in practice.

C. Tighter Bounds for Mixture Models

In the case that $\mathbf{w}$ is a mixture model, we can simply compute the moments of $p(\mathbf{w})$ using the result of Proposition 1 in terms of moments of the components $p(\mathbf{w}_i)$ and treat the mixture model as any other random variable. However, Proposition 1 also seems to suggest that we can instead bound the component probabilities $\mathbb{P}(p(\mathbf{w}_i) \leq 0)$ and bound the overall risk with the weighted sum of the component bounds. Intuition suggests this approach may be better because it involves applying concentration inequalities at the most detailed level possible. Since concentration inequalities are, in a sense, blanket statements about distributions with a given set of moments, it makes sense to apply them at the most detailed level possible. In fact, we show with Theorem 1 that applying concentration inequalities to the mixture components individually will almost certainly produce a less conservative risk bound.

Theorem 1: For any random vector mixture model $\mathbf{w}$ with n components $\mathbf{w}_i$ and weights w_i and any measurable function g, if $\text{Conc}^*[g(\mathbf{w}_i)] \leq 0, \forall i \in [n]$, then:

$$\mathbb{P}(g(\mathbf{w}) \leq 0) \leq \sum_{i=1}^{n} w_i \text{Conc}(g(\mathbf{w}_i)) \tag{22a}$$

$$\leq \text{Conc}(g(\mathbf{w})) \tag{22b}$$

For Cantelli and VP, almost surely, we have:

$$\sum_{i=1}^{n} w_i \text{Conc}(g(\mathbf{w}_i)) < \text{Conc}(g(\mathbf{w})) \tag{23}$$

Proof: See the version of this letter with appendices [33]. ■

Remark 1: "Almost surely" in Theorem 1 means that if the first and second moments of each component are randomly chosen from $\mathbb{R}^2$ according to any distribution supported on a subset of $\mathbb{R}^2$, then the result holds with probability one.

We arrive at the "almost surely" result by establishing a geometric sufficient condition in the space of first and second moments. We show that each component of a mixture model generates a line in the first-second moment space. If the first-second moment point of each component does not lie on the line generated by any other component, the result holds. Since the union of a finite number of lines has Lebesgue measure zero with respect to $\mathbb{R}^2$, the probability of randomly choosing first and second moments that violate the condition is zero. Even if the condition is violated, an arbitrarily small perturbation of the moments would satisfy the condition. Fig. 3 illustrates an example that satisfies this condition.

D. Bounding Risk for the Trajectory Planning Problem

We now return to the problem of establishing a bound on:

$$\mathbb{P}(Q(\mathbf{a}_t) \leq 1) \tag{24}$$

Letting $\mathbf{a}_t^{(i)}$ denote the i_{th} component of $\mathbf{a}_t$, this essentially consists of applying the methods of the prior subsections to the polynomial:

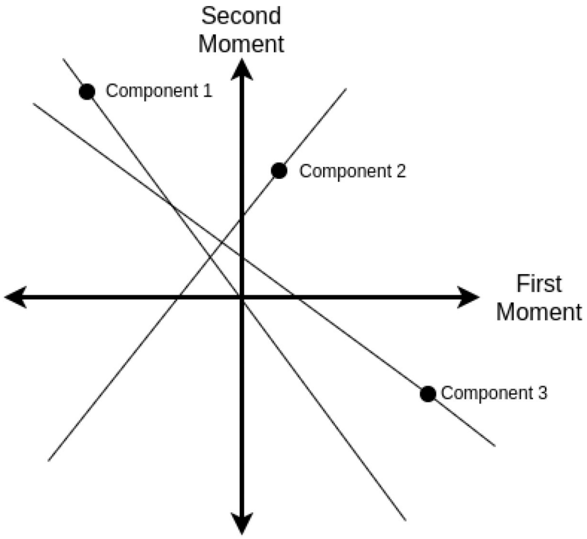

Fig. 3. The first and second moments of three components of a mixture model are plotted along with the lines they generate. Note how every component lies on the line it generates, but does not lie on the lines generated by any other component. Thus, for this example, the "almost surely" result of Theorem 1 applies.

$$Q(\mathbf{a}_t^{(i)}) - 1 \tag{25}$$

However, there is one small additional complication: $\mathbf{a}_t^{(i)}$ is in the body frame, making it a function of the planned ego vehicle pose and the global frame distribution $\mathbf{g}_t^{(i)}$. Letting $\mathbf{y}_t = [x_t, y_t]$ denote the ego vehicle position, θ_t denote the ego vehicle heading, and $R(\cdot)$ denote the 2D rotation matrix, they are related by:

$$\mathbf{a}_t^{(i)} = R(\theta_t)^T(\mathbf{g}_t^{(i)} - \mathbf{y}_t) \tag{26}$$

The idea here is to view $\mathbf{a}_t^{(i)}$ as a polynomial in $\mathbf{g}_t^{(i)}$ and to view the elements of both $R(\theta_t)^T$ and $\mathbf{y}_t$ as elements of the coefficients. From this perspective, we apply AlgebraicMoments to express moments of $\mathbf{a}_t^{(i)}$ in terms of moments of $\mathbf{g}_t^{(i)}$. The moments of $Q(\mathbf{a}_t^{(i)}) - 1$ can, in turn, be expressed in terms of the moments of $\mathbf{a}_t^{(i)}$. With the first and second moments of $Q(\mathbf{a}_t^{(i)}) - 1$ expressed in closed form, we can enforce the following constraints in the optimization problem where n is the number of mixture components:

$$\sum_{i=1}^{n} \text{Conc}[Q(\mathbf{a}_t^{(i)}) - 1] \leq \epsilon_t, t \in [T] \tag{27}$$

$$\text{Conc}^*[Q(\mathbf{a}_t^{(i)}) - 1] \leq 0, i \in [n] t \in [T] \tag{28}$$

AlgebraicMoments provides the functionality to directly generate code for the above expressions given a specification.

V. Application to Trajectory Planning With Model Predictive Contouring Control

In the context of autonomous driving, an approximate coarse-grained path is almost always available to trajectory planners as most autonomous driving systems have maps of lane geometries *a priori* and also have routers and higher level planners that provide a discrete plan (e.g: a sequence of waypoints along the road). This availability of a "reference path" makes model predictive contouring control (MPCC) a useful approach. MPCC is a methodology for expressing an approximation of the minimum distance from a point to a third order polynomial in closed form. This allows for deviations from the reference path to be included in the cost function of an optimization problem. By jointly applying MPCC and our chance-constraint formulation, we arrive at a trajectory planner that can find trajectories with low deviation from the reference path while satisfying some desired level of safety, allowing for the generation of rich qualitative behavior. In the following subsections, we present a brief overview of the MPCC formulation; the interested reader is referred to [5], [34] for additional details.

A. Reference Path

Following the standard contouring control formulation, the reference path is represented as third order polynomials in an arc-length parameter $s \in [0, L]$ where L is the length:

$$\begin{bmatrix} x_{ref}(s) \\ y_{ref}(s) \end{bmatrix} = \begin{bmatrix} c_{x0} + c_{x1}s + c_{x2}s^2 + c_{x3}s^3 \\ c_{y0} + c_{y1}s + c_{y2}s^2 + c_{y3}s^3 \end{bmatrix} \tag{29}$$

Where c_{xi} and c_{yi} for $i \in [3]$ are the polynomial coefficients. Parameterizing a third order polynomial path with arc-length is a difficult problem itself without exact solutions, but approximation methods are well-studied [35], [36]. For our experiments, we applied a simple approximation by initially generating the polynomials with $s \in [0, 1]$ and then the lengths of the polynomials were computed by numerical integration. The coefficients are then scaled s.t. $s \in [0, L]$. The heading at each point on the reference path, denoted $\Theta(s)$, can be expressed as such:

$$\Theta(s) = \arctan\left(\frac{\partial y_{ref}}{\partial x_{ref}}(s)\right) \tag{30}$$

B. Contouring Deviation and Lag Error

Ideally, the Euclidean distance from the ego vehicle to the nearest point on the reference path, which we will refer to as *contouring deviation*,[2] would be used as the measure of deviation from the reference path, but doing so requires a minimization over the path parameter s that is computationally intractable to perform in an optimization routine. The standard solution is to approximate the contouring deviation by using the distance the vehicle has travelled, denoted Δ, as an approximation. This only requires adding an additional integrator variable to the dynamics model as the time derivative of Δ is the vehicles speed. Letting $\bar{x}_t = x_t - x_{ref}(\Delta_t)$ and $\bar{y}_t = y_t - y_{ref}(\Delta_t)$, contouring deviation can be approximated with:

$$D_t = \sin(\Theta(\Delta_t))\bar{x}_t - \cos(\Theta(\Delta_t))\bar{y}_t \tag{31}$$

It is also important to penalize error between Δ and the true parameter corresponding to the closest point on the path to the vehicle. This quantity is known as the *lag error* and can be approximated by:

$$L_t = -\cos(\Theta(\Delta_t))\bar{x}_t - \sin(\Theta(\Delta_t))\bar{y}_t \tag{32}$$

C. Ego Vehicle Model

For driving in nominal conditions, the kinematic bicycle model is known to provide a high level of fidelity while requiring less computational cost than a dynamics model making it well suited for trajectory planning [37]. The state of the vehicle is defined as $\mathbf{x} = [x, y, \theta, v, \delta, \Delta]^T$ where x, y denotes the vehicle's position and θ denotes the heading in the global coordinates. v denotes speed, δ denotes the front steering angle, and Δ denotes the distance traveled. The control inputs are $\mathbf{u} = [u_a, u_\delta]^T$ where u_a is acceleration and u_δ is the rate of change of the steering angle. The relevant physical parameters of the vehicle in this model are the distances from the center of gravity to the front and rear axles; we denote them l_f and l_r respectively. The continuous time model is thus:

$$\dot{\mathbf{x}} = \begin{bmatrix} v\cos(\theta + \beta) \\ v\sin(\theta + \beta) \\ \frac{v}{l_r}\sin(\beta) \\ u_a \\ u_\delta \\ v \end{bmatrix} \quad \beta = \arctan\left(\frac{l_r}{l_f + l_r}\tan(\delta)\right) \tag{33}$$

D. Optimization Statement

In the cost function, we penalize contouring deviation, lag error, control effort, and deviation from a reference speed (e.g: the roads speed limit):

$$c(\mathbf{x}_t, \mathbf{u}_t) = c_D D_t^2 + c_L L_t^2 + \mathbf{u}_t^T R \mathbf{u}_t + c_v(v_t - v^*)^2 \tag{34}$$

Where $c_D, c_L, c_v \in \mathbb{R}$ and $R \in \mathcal{S}^2_{++}$ are cost function parameters and v^* is the reference speed. Letting f_{RK4} denote the RK4 approximation of the bicycle model, the full problem we solve is:

$$\min \sum_{t=0}^{T} c(\mathbf{x}_t, \mathbf{u}_t) \tag{35a}$$

$$\mathbf{x}_{t+1} = f_{RK4}(\mathbf{x}_t, \mathbf{u}_t), t \in [T-1] \tag{35b}$$

$$\sum_{i=1}^{n} \mathrm{Conc}[Q(\mathbf{a}_t^{(i)}) - 1] \leq \epsilon_t, t \in [T] \tag{35c}$$

$$\mathrm{Conc}^*[Q(\mathbf{a}_t^{(i)}) - 1] \leq 0, i \in [n] t \in [T] \tag{35d}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_t \leq \mathbf{x}_{max}, t \in [T] \tag{35e}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_t \leq \mathbf{u}_{max}, t \in [T-1] \tag{35f}$$

[2] In the literature, this is usually known as *contouring error* but we call it contouring deviation as deviation from the reference path to satisfy chance-constraints is not necessarily undesirable.

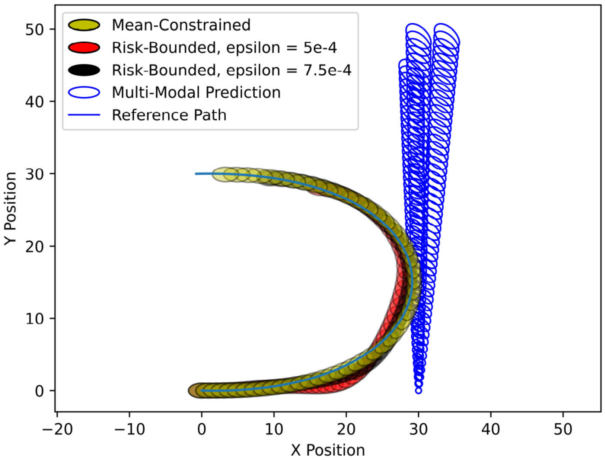

Fig. 4. Comparison between trajectories planned with a deterministic mean-constrained formulation and the risk-bounded formulation with two different chance-constraints. Both risk-bounded trajectories exhibit more cautious behavior when performing the U-turn, by taking a wider and slower turn to satisfy chance-constraints with the lower chance-constraint producing a noticeably more conservative result. The prediction is represented with two standard deviation confidence ellipses.

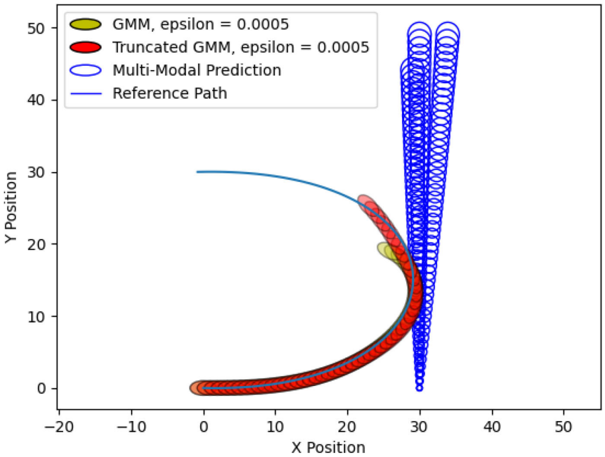

Fig. 5. Planned trajectories from using a GMM and a truncated GMM with the same mean vector and covariance.

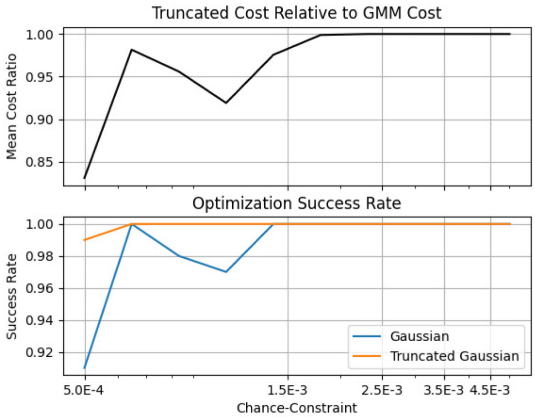

Fig. 6. Plots of the mean cost ratio and solution success rates across the range of chance-constraints tested.

VI. Numerical Experiments

For numerical experiments, we used optimizers generated with FORCES Pro, a software package that generates high performance interior-point solvers for optimal control problems [38], [39]. A third order polynomial path emulating a U-turn was generated to serve as the reference path. Cost parameters are included in version of this letter with appendices [33]. Example GMM predictions were created by linearly interpolating the mean vectors and covariance matrices of manually specified end-point GMMs. To demonstrate the advantages of using non-Gaussian distributions with our approach, predictions consisting of mixtures of truncated multivariate Gaussians, which we will refer to as truncated GMMs, were also created by performing a box truncation of the GMM predictions at two standard deviations. We believe truncated GMMs are effective models for agents as reachability analysis can be applied in practice to determine bounds on the positions of agents. 50 time steps were used with intervals of 0.1 seconds for a planning horizon of 5 seconds. Initial guesses for the optimizer were produced by simulating the system (35c) with constant control inputs.

Fig. 4 shows two trajectories planned with two different formulations: one deterministic and the other risk-bounded. The deterministic formulation uses the means of the agent predictions to enforce a deterministic constraint on collision. The risk-bounded formulation set $\epsilon = 0.0005$ and $\epsilon = 0.00075$ and utilized the VP inequality. Note how the risk-bounded formulation plans qualitatively more cautious trajectories by taking wider turns and slower speeds.

As mentioned earlier, the unbounded support of multivariate Gaussians limits their realism as models for agent position as the forward reachable set of agents is bounded by the laws of physics. To demonstrate the advantages of using a distribution with bounded support, we compared trajectories planned using moments of GMM predictions with trajectories planned using truncated GMMs with the same mean and covariance. While the distributions' means and covariances match, their higher order moments differ significantly as the kurtosis, a metric for the "heaviness" of tails, is expressed by the fourth order moment. These higher order moments are captured in our formulation because the variance of the random variable $Q(\mathbf{a}_t) - 1$ is a function of moments of the prediction up to order four. By leveraging this additional moment information, the planner can plan less conservative trajectories with truncated GMMs that are still safe; Fig. 5 shows an example.

To test at a larger scale, 100 different U-turn scenarios were generated by perturbing the parameters of the polynomial path and 10 chance-constraint settings were tested for each formulation on each scenario, totaling 2000 optimizations. As shown in Fig. 6, the truncated GMM formulation solves the problem more reliably than the GMM formulation and also produces lower cost trajectories. Over 2000 optimizations, the average solution time was 12.9 ms, making this formulation fast enough for use online.

VII. Conclusion

In this letter, we presented a chance-constrained trajectory planning formulation for autonomous vehicles that can handle mixtures of non-Gaussian distributions of agent positions and does not need to make point mass assumptions. To enforce the chance-constraint, we present a general framework that leverages symbolic algebra to generate expressions that upper-bound polynomial chance-constraints in terms of statistical moments of the underlying distribution. We demonstrated this approach with numerical experiments and show the optimization problem is fast enough for use online. We also provide a Python package, AlgebraicMoments, to enable other members of the community to adopt our approach to bounding chance-constraints. We note that while enforcing chance-constraints using the presented concentration inequalities provides great generality, this generality can also produce excessively conservative results. Future works should consider leveraging more distribution specific information to establish tighter bounds.

References

[1] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 336–345.

[2] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," 2019, in *Proc. IEEE Conf. Robot. Learn.*, 2020, pp. 86–99.

[3] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *Proc. IEEE Intell. Vehicles Symp.*, 2018, pp. 1179–1184.

[4] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8454–8462.

[5] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1928–1935.

[6] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 742–753, May 2006.

[7] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 502–517, Jun. 2010.

[8] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proc. AIAA Guid., Navigation, Control Conf.*, 2009, Art. no. 5876.

[9] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *Proc. AIAA Guid., Navigation, Control Conf.*, 2010, Art. no. 8160.

[10] M. Cannon, "Chance-constrained optimization with tight confidence bounds," 2017, *arXiv:1711.03747*.

[11] J. Norden, M. O'Kelly, and A. Sinha, "Efficient black-box assessment of autonomous vehicle safety," in *Workshop Mach. Learn. Auton.*, 2019.

[12] T. Summers, "Distributionally robust sampling-based motion planning under uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 6518–6523.

[13] V. Renganathan, I. Shames, and T. H. Summers, "Towards integrated perception and motion planning with distributionally robust risk constraints," 2020, *arXiv:2002.02928*.

[14] 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.

[15] J. A. Paulson, E. A. Buehler, R. D. Braatz, and A. Mesbah, "Stochastic model predictive control with joint chance constraints," *Int. J. Control*, vol. 93, no. 1, pp. 126–139, 2020.

[16] A. M. Jasour and B. Williams, "Risk contours map for risk bounded motion planning under perception uncertainties," in *Proc. Robot.: Sci. Syst.*, 2019.

[17] A. M. Jasour and B. C. Williams, "Sequential chance optimization for flow-tube based control of probabilistic nonlinear systems," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 5392–5399.

[18] A. Wang, X. Huang, A. Jasour, and B. Williams, "Fast risk assessment for autonomous vehicles using learned models of agent futures," in *Robot.: Sci. Syst.*, 2020.

[19] A. M. Jasour, A. Hofmann, and B. C. Williams, "Moment-sum-of-squares approach for fast risk estimation in uncertain environments," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 2445–2451.

[20] J. Steinhardt and R. Tedrake, "Finite-time regional verification of stochastic non-linear systems," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 901–923, 2012.

[21] N. Rhinehart, K. M. Kitani, and P. Vernaza, "R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 772–788.

[22] V. Witkovskỳ, G. Wimmer, Z. Ďurišová, S. Ďuriš, and R. Palenčár, "Brief overview of methods for measurement uncertainty analysis: Gum uncertainty framework, monte carlo method, characteristic function approach," in *Proc. IEEE 11th Int. Conf. Meas.*, 2017, pp. 35–38.

[23] S. Frühwirth-Schnatter, *Finite Mixture and Markov Switching Models*. Berlin, Germany: Springer, 2006.

[24] M. Ono and B. C. Williams, "Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint," in *Proc. 47th IEEE Conf. Decis. Control*, 2008, pp. 3427–3432.

[25] Y. Ma, S. Vichik, and F. Borrelli, "Fast stochastic mpc with optimal risk allocation applied to building control systems," in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 7559–7564.

[26] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *Proc. IEEE Amer. Control Conf.*, 2007, pp. 2296–2301.

[27] J. Wang, J. Steiber, and B. Surampudi, "Autonomous ground vehicle control system for high-speed and safe operation," in *Proc. IEEE Amer. Control Conf.*, 2008, pp. 218–223.

[28] P. Duchesne and P. L. De Micheaux, "Computing the distribution of quadratic forms: Further comparisons between the liu–tang–zhang approximation and exact methods," *Comput. Stat. Data Anal.*, vol. 54, no. 4, pp. 858–862, 2010.

[29] Y. Y. Haimes and R. E. Steuer, *Research and Practice in Multiple Criteria Decision Making: Proceedings of the 14th Int. Conf. on Multiple Criteria Decis. Making (MCDM) Charlottesville*. Berlin, Germany: Springer, 2012, vol. 487.

[30] M. Padulo and M. D. Guenov, "Worst-case robust design optimization under distributional assumptions," *Int. J. Numer. Methods Eng.*, vol. 88, no. 8, pp. 797–816, 2011.

[31] I. Popescu, "A semidefinite programming approach to optimal-moment bounds for convex classes of distributions," *Math. Operations Res.*, vol. 30, no. 3, pp. 632–657, 2005.

[32] D. Vysochanskij and Y. I. Petunin, "Justification of the 3σ rule for unimodal distributions," *Theory Probability Math. Stat.*, vol. 21, pp. 25–36, 1980.

[33] A. Wang, A. Jasour, and B. Williams, "Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," 2020, *arXiv:2002.10999*.

[34] D. Lam, C. Manzie, and M. C. Good, "Model predictive contouring control for biaxial systems," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 2, pp. 552–559, Mar. 2013.

[35] J. W. Peterson, "Arc length parameterization of spline curves," 2002. [Online]. Available: https://saccade.com/writing/graphics/RE-PARAM.PDF

[36] M. Floater, "Arc length estimation and the convergence of polynomial curve interpolation," *BIT Numer. Math.*, vol. 45, no. 4, pp. 679–694, 2005.

[37] 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.*, 2015, pp. 1094–1099.

[38] A. Domahidi and J. Jerez, "Forces professional," Embotech AG, 2014-2020, http://embotech. com/forces-pro

[39] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: An efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *Int. J. Control*, vol. 93, no. 1, pp. 13–29, 2020.