

Scenario-Based Trajectory Optimization in Uncertain Dynamic Environments

Oscar de Groot , Bruno Brito , Laura Ferranti , Darius Gavrila , and Javier Alonso-Mora 

Abstract—We present an optimization-based method to plan the motion of an autonomous robot under the uncertainties associated with dynamic obstacles, such as humans. Our method bounds the marginal risk of collisions at each point in time by incorporating chance constraints into the planning problem. This problem is not suitable for online optimization outright for arbitrary probability distributions. Hence, we sample from these chance constraints using an uncertainty model, to generate “scenarios,” which translate the probabilistic constraints into deterministic ones. In practice, each scenario represents the collision constraint for a dynamic obstacle at the location of the sample. The number of theoretically required scenarios can be very large. Nevertheless, by exploiting the geometry of the workspace, we show how to prune most scenarios before optimization and we demonstrate how the reduced scenarios can still provide probabilistic guarantees on the safety of the motion plan. Since our approach is scenario based, we are able to handle arbitrary uncertainty distributions. We apply our method in a Model Predictive Contouring Control framework and demonstrate its benefits in simulations and experiments with a moving robot platform navigating among pedestrians, running in real-time.

Index Terms—Motion and path planning, optimization and optimal control, collision avoidance.

I. INTRODUCTION

MOBILE robots are increasingly becoming part of our society, with applications in warehouses [1], automotive [2], maritime transportation [3], etc. In all these domains, it is essential that the robots can safely operate in dynamic environments (e.g., near humans). However, uncertainty is omnipresent, for example, in the future motion paths of the dynamic obstacles or in sensing (i.e., localization) errors. Our goal is to design a local robot motion planning algorithm able to plan collision-free trajectories in the presence of possibly unbounded and arbitrary uncertainties.

Optimization-based motion planning methods avoid collisions by imposing constraints in the optimization problem. Classical methods consider deterministic obstacle predictions, that

is, they do not account for the presence of uncertainties. When uncertainties come into the picture, deterministic frameworks fail to achieve safety, since they do not consider the possible spread of outcomes. In the case of bounded uncertainties, that is, if the probability density function is non-zero in a bounded domain of the robot’s workspace and is zero elsewhere, then it is possible to set the acceptable level of risk to zero. This approach is referred to as *robust optimization*. On the one hand, this approach allows for the addition of uncertainties in the deterministic framework. On the other hand, the assumption that the distribution is bounded can be limiting (e.g., when obstacle predictions are Gaussian). Additionally, it becomes conservative when the domain of support is large. In the presence of unbounded uncertainties, *chance constraint optimization* allows one to constrain the probability of collisions to be below an acceptable level of risk. In this work, and likewise to [4], [5], we consider the marginal probabilities of collision at each point in time. This is, we constrain the chance of collision for each step of the trajectory, separately.

Directly evaluating these chance constraints is intractable, especially for arbitrary shapes of the distribution. Instead they are often either approximated (e.g., using particle filters [6]) or bounded. Approximation techniques have received most attention, due to their sample efficiency. However, the safety of these approaches cannot be guaranteed, especially when operating in unknown environments.

Contribution. In this work, we assume that a perception module provides predictions of the motion of dynamic obstacles together with a description of their (unbounded, possibly non Gaussian) uncertainty. To provide probabilistic safety for each step of the planned trajectory with respect to the modeled uncertainties, our work presents a novel *probabilistic trajectory optimization* framework for motion planning in uncertain dynamic environments, that is, a Scenario-based Model Predictive Contouring Control (S-MPCC) design. Our S-MPCC builds on nonconvex scenario-optimization framework [7] and the model predictive contouring control (MPCC) design of [8]. We show that in contrast with the general a posteriori results in [7], we obtain the perceived risk of our motion plan *before optimization*. The support subsample, which is the key indicator for the risk in [7], is obtained through the geometry of the problem, leading to efficient evaluation of the samples. While sampling-based chance constrained approaches are generally considered intractable for real-time motion planning, our method is competitive in terms of computation times with state-of-the-art planning methods, while applicable to generic uncertainties. The

Manuscript received October 15, 2020; accepted February 20, 2021. Date of publication April 21, 2021; date of current version May 11, 2021. This letter was recommended for publication by Associate Editor A. Faust and Editor N. Amato upon evaluation of the reviewers’ comments. This work was supported by the Dutch Science Foundation NWO-TTW, within the SafeVRU Project (nr. 14667) and Veni Award (nr. 15916). (Corresponding author: Oscar de Groot.)

The authors are with the Department of Cognitive Robotics, TU Delft, CD Delft 2628, The Netherlands (e-mail: o.m.degroot@tudelft.nl; bruno.debrito@tudelft.nl; l.ferranti@tudelft.nl; D.M.Gavrila@tudelft.nl; j.alonsomora@tudelft.nl).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2021.3074866>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3074866

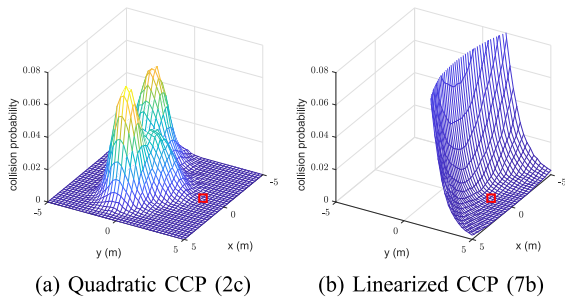


Fig. 1. Grid wise evaluation of the collision probability, with $r = 0.5$ m of chance constraints (2c) and (7b) in an example where δ follows a Mixture-of-Gaussians (MoG) distribution. The red square denotes the linearization point.

approach handles multiple obstacles and accounts for the size of the vehicle and obstacles.

We show how our approach allows the robot to move continuously through its environment while reasoning about its probability of colliding with dynamic obstacles. In our framework, illustrated in Fig. 2, instead of directly solving the chance constrained motion planning problem, we solve an associated deterministic problem obtained as follows. First, we apply a tailored linearization of the chance constraints, then we sample from the linearized chance constraints a large set of deterministic constraints, known as *scenarios*. The number of scenarios drawn is linked with the associated risk of collisions. This allows us to reformulate the original planning problem in a deterministic one, known as a scenario program. Using this approach we effectively resolve the chance constraints in a preprocessing step.

Uncertainty of predictions is generally non Gaussian and appears, for example, when Gaussian uncertainty is propagated through nonlinear dynamics. Our method is applicable to generic uncertainties. We demonstrate our framework for Gaussian and non Gaussian uncertainties using an autonomous ground robot, both in simulation and in experiments.

Related Work: Trajectory optimization with chance constraints for collision avoidance has previously been considered in the case of *Gaussian uncertainties*. For example, [8] defined collision avoidance constraints in an MPC framework, by modeling the dynamic obstacles as ellipses. This representation allows the planner to accommodate Gaussian uncertainties (as the level set of a Gaussian distribution are ellipses) and solve a deterministic nonlinear optimization problem online. The approach was applied for autonomous driving in [9]. In [4] the chance constraint problem is solved explicitly. Their approach linearizes the collision chance constraints, and uses the prior of Gaussian uncertainty to formulate deterministic constraints on the mean and covariance of the distribution.

Literature on motion planning for *non Gaussian* uncertainty distributions is still limited. Inspired by particle filter approaches, [6] introduces a method to approximately evaluate the chance constraints using particles. Their method uses a relatively small sample size, but cannot provide any guarantees on the safety of the solution. In [10], a Rapidly-expanding Random Trees (RRT) algorithm is presented where the uncertainty is evaluated for each node in the tree. The dynamic

obstacle trajectories are predicted using Gaussian Processes. In [11], assuming full knowledge of the probability distribution, polynomial chance constraints are transformed to deterministic inequalities using the statistical moments of the non Gaussian distribution.

Compared to the previous methods, our approach bounds the probability of collision using the scenario optimization framework. This framework is well established for convex optimization ([5], [12]–[16]). A framework for non-convex scenario optimization was recently introduced in [7]. We rely on this framework and extend it in the context of robot motion planning.

II. PROBLEM FORMULATION

We consider the motion planning problem of a mobile robot, whose dynamics can be represented by the following nonlinear discrete-time system:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad (1)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{u}_k \in \mathbb{R}^m$ denote the states and inputs, respectively. The robot can move within a workspace (e.g., the 2D plane when we consider ground robots). In the workspace, the robot must avoid collisions with dynamic obstacles. We model the collision region of the robot \mathcal{V}_k at time k as the union of n_c circles, and the collision region of the dynamic obstacles \mathcal{D}_k^v at time k as a single circle.

The position of dynamic obstacles along the planning horizon of the robot is uncertain. We denote the uncertainty of the obstacles at stage k with a tuple $(\Delta_k, \mathcal{D}_k, \mathbb{P}_{k,\text{real}})$, where Δ_k is a probability space equipped with a σ -algebra \mathcal{D}_k and a probability measure $\mathbb{P}_{k,\text{real}}$. We allow the probability spaces to be unbounded and non Gaussian. We assume that at each step a perception module provides the motion planner with an independent model of the uncertainty, formalized as follows.

Assumption 1: The planner is provided with a model \mathbb{P}_k of the real probability measure $\mathbb{P}_{k,\text{real}}$ for each k .

Assumption 2: Random variables $\delta_j \sim \mathbb{P}_j$ and $\delta_l \sim \mathbb{P}_l$ are independent for all stages $j, l \in \{1, \dots, N\}$, where $j \neq l$.

Assumption 2 implies that the dependency induced, for example, by the dynamics of an obstacle, is handled by the perception module such that the uncertainties are independent as viewed from the perspective of the motion planner. The assumption is common in state-of-the-art perception modules, for example [17]–[19].

Under the, possibly unbounded, uncertainty of the dynamic obstacles, we constrain the marginal probability of collision at each time step of the trajectory using *chance constraints*, similarly to [4], [5]. Each chance constraint is subject to an acceptable risk level ϵ_k , which can be tuned accordingly. This implies that we cannot give a non-conservative bound on the collision risk of the full motion plan. However, by frequently recomputing the motion plan, for example in an MPC framework, the actions in the near future are probabilistically safe and risk in later stages is reconsidered when the robot moves closer. We formulate the

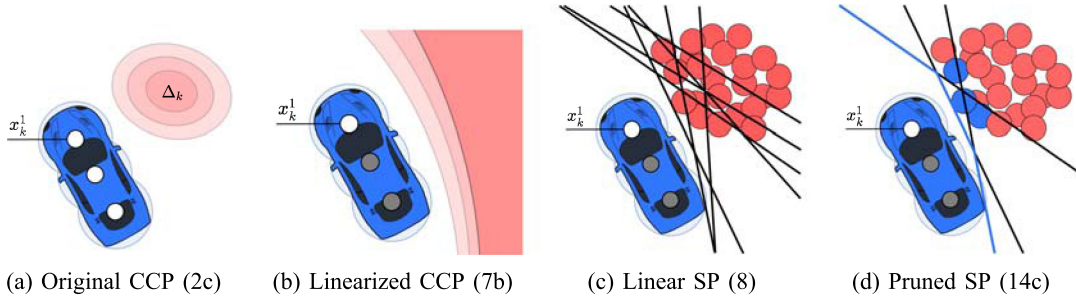


Fig. 2. Our approach exemplified for one robot’s disc and one dynamic obstacle for a single stage. The robot and the obstacle are drawn in blue and red, respectively. Fig. 2(a) shows the 1σ to 3σ interval of the uncertainty in red shades. Fig. 2(b) shows the probabilistic collision region when linearized from the robot disc at the front. Fig. 2(c) shows the sampled locations in red and boundaries of the constraints in black. Fig. 2(d) shows the resulting minimal polytope in blue.

motion planning problem as follows:

$$\min_{\mathbf{u} \in \mathbb{U}} \sum_{k=1}^N J(\mathbf{x}_k, \mathbf{u}_k) \quad (2a)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x} \in \mathbb{X} \quad (2b)$$

$$\mathbb{P}_k [\|\mathbf{x}_k^d - \delta_k^v\|_2 > r, \forall d, v] \geq 1 - \epsilon_k, \quad \forall k, \quad (2c)$$

where $\mathbf{u} = \{\mathbf{u}_1, \dots, \mathbf{u}_N\} \in \mathbb{U}$ are the optimized system inputs subject to input constraints, $\delta_k^v \in \Delta_k^v$ is the uncertain position of obstacle v at stage k and $J(\mathbf{x}_k, \mathbf{u}_k) \geq 0$ is the cost function specifying performance metrics. The radius r is the sum of vehicle and obstacle radii. To simplify the notation, we assume this radius to be a constant. The chance constraint, (2c), constrains the probability of collisions between each collision circle d of the vehicle and the collision circle of each dynamic obstacle v at prediction step k to be below the risk level ϵ_k , as visualized in Fig. 2(a). The probability measure \mathbb{P}_k refers to the modeled uncertainty.

Problem (2) is a chance constrained optimization problem. As discussed in Section IV, to solve this problem, we rely on the nonconvex scenario optimization (NSO) framework of [7], for which we provide an overview in the following section. This framework can in general provide a bound on the risk with respect to the unknown probability distribution \mathbb{P}_{real} , by sampling from the real system. In the real-time setting of this letter, however, collecting samples online is intractable. Instead we propose to sample from the model distribution \mathbb{P} , as defined in Assumption 1. For consistency of notation, the results of [7] are presented here using the model \mathbb{P} .

III. NONCONVEX SCENARIO OPTIMIZATION

The NSO framework allows us to replace chance constraints with deterministic constraints by sampling. Consider the Chance Constrained Problem (CCP)

$$\min_{\mathbf{u} \in \mathbb{U}} J(\mathbf{u}) \quad (3a)$$

$$\text{s.t.} \quad \mathbb{P} [g(\mathbf{u}, \delta) \leq 0] \geq 1 - \epsilon, \quad \delta \in \Delta, \quad (3b)$$

where \mathbf{u} are decision variables, $\delta \in \Delta$ is the realization of the uncertainty and the function $g : \mathbb{X} \times \Delta \rightarrow \mathbb{R}$ is a nonlinear function associated with the nonconvex constraint $g(\mathbf{x}, \delta) \leq 0$.

The authors of [7] established a link between CCP (3) and the deterministic Scenario Program (SP):

$$\min_{\mathbf{u} \in \mathbb{U}} J(\mathbf{u}) \quad (4a)$$

$$\text{s.t.} \quad g(\mathbf{u}, \delta^i) \leq 0, \quad \delta^i \in \Delta, \quad \forall i \in \mathcal{S}. \quad (4b)$$

We denote its solution by \mathbf{u}_{SP}^* . Each of the S constraints in (4b) is constructed by drawing a sample δ^i from Δ , and formulating the constraint $g(\mathbf{u}, \delta^i) \leq 0$ in the scenario where the sample δ^i is a realization of the uncertainty. Since each of the samples specifies a scenario, the samples themselves are called *scenarios* and the constraints (4b) are known as *scenario constraints*. The *violation probability*, $V : \mathbb{U} \rightarrow [0, 1]$, given by

$$V(\mathbf{u}) := \mathbb{P} [\delta \in \Delta : g(\mathbf{u}, \delta) > 0], \quad (5)$$

defines the probability that input \mathbf{u} violates a newly observed scenario. The solution of the SP in (4) depends on randomly sampled scenarios and hence its violation probability is a random variable over the product probability measure, given by $\mathbb{P}^S = \mathbb{P} \times \dots \times \mathbb{P}$ (S times). To link the SP of (4) with the CCP of (3), we are therefore interested in bounding the probability that $V(\mathbf{u}_{SP}^*)$ satisfies our risk bound ϵ , a probability which we refer to as the confidence. A key definition in this direction is the *support subsample*.

Definition [7]: A *support subsample* of an SP is a subset of scenarios $\mathcal{S}_{\text{support}} \subseteq \mathcal{S}$ that results in the same optimizer as the original SP. The cardinality of the support subsample, that is, the support subsample size, is denoted by s . The smallest support subsample size is denoted by s^* .

Theorem 1 in [7] provides the following confidence bound

$$\mathbb{P}^S [V(\mathbf{u}_{SP}^*) > \epsilon(s^*)] \leq \sum_{s=0}^{S-1} \binom{S}{s} [1 - \epsilon(s)]^{S-s} = \beta. \quad (6)$$

Here $\epsilon(s) : \{0, \dots, S\} \rightarrow [0, 1]$ can be designed subject to (6) and $\epsilon(S) = 1$, an example can be found in [7, Sec. II]. Equation (6) theoretically links the sampling size S , confidence parameter β (complement of the confidence) and risk ϵ , based on the observed support sample size. Notice that in this work, as a consequence of using model distribution \mathbb{P} , the bound (6) applies to the modeled uncertainty rather than the real robot, in contrast with [7, Th. 1].

IV. PROPOSED APPROACH

Our method relies on the Model Predictive Contouring Control (MPCC) framework [8] to define the objective to optimize to plan a suitable path for the robots. Our method differs from [8] in the way we deal with dynamic obstacles, as detailed in the rest of the section. As such we will refer to our approach as Scenario-MPCC (S-MPCC). To present the method, we consider a single dynamic obstacle and one of the discs used to represent the vehicle.¹

A. Chance Constraints Linearized in the Robot Position

Chance constraints (2c) are nonconvex in the robot position when sampled (see discs in Fig. 2(c)) and the associated SP may have many local optima and a sizable support subsample. We therefore consider a linearization of the collision regions (depicted by the lines in Fig. 2(c)) before sampling to decrease the support subsample size of the SP. This step reduces the risk of its solution significantly. We modify the constraints as

$$\mathbf{A}_k = \frac{\boldsymbol{\delta}_k - \hat{\mathbf{x}}_k}{\|\boldsymbol{\delta}_k - \hat{\mathbf{x}}_k\|}, \quad b_k = \mathbf{A}_k^T (\boldsymbol{\delta}_k - \mathbf{A}_k \mathbf{r}), \quad (7a)$$

$$\mathbb{P}_k [\mathbf{A}_k^T \mathbf{x}_k \leq b_k] \geq 1 - \epsilon_k, \forall k, \boldsymbol{\delta}_k \in \Delta_k, \quad (7b)$$

where we linearize the collision region with respect to $\hat{\mathbf{x}}_k$, the k -step ahead prediction of the robot position. We employ the trajectory of the previous planning cycle, forward propagated, as predictor. That is,² $\hat{\mathbf{x}}_{t|k} = \mathbf{x}_{t-1|k+1}$ and $\hat{\mathbf{x}}_{t|N} = \mathbf{x}_{t-1|N}$. Hence, we search for collision-free solutions around the planned trajectory of the previous planning cycle. We show in Section IV-C that after linearization, the free-space of the resulting SP is convex in the robot position. A comparison between chance constraints (2c) and (7b) for an example is provided in Fig. 1. The linearized chance constraints capture less of the shape of the distribution, but are accurate near $\hat{\mathbf{x}}_k$ and thus sufficient for motion planning. Note that the linearizations are performed for each stage of the trajectory, as illustrated in Fig. 2(b).

B. Scenario Program

For each of the chance constraints in (7b) we construct a set of deterministic constraints by sampling from the uncertainty. The red circles in Fig. 2(c) represent these samples and the black lines are the *scenarios* (Section III). The resulting SP is given by

$$\min_{\mathbf{u} \in \mathbb{U}} \sum_{k=1}^N J(\mathbf{x}_k, \mathbf{u}_k) \quad (8a)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x} \in \mathbb{X} \quad (8b)$$

$$\mathbf{A}_k^T (\boldsymbol{\delta}_k^i, \hat{\mathbf{x}}_k) \mathbf{x}_k \leq b_k(\boldsymbol{\delta}_k^i, \hat{\mathbf{x}}_k), \forall i \in \mathcal{S}_k, \forall k. \quad (8c)$$

The theoretic properties of SPs, discussed in Section III, are limited to CCPs with one chance constraint. However, (7b)

¹Section IV-D shows how this case extends linearly to multiple dynamic obstacles and multiple discs.

²We denote by $\mathbf{x}_{t|k}$ the k -step ahead prediction of the robot trajectory for the MPC planning cycle at time t

describes multiple chance constraints, one for every stage of the planned trajectory. We now show that multiple chance constraints can be handled separately, resulting in a probabilistic feasibility property per stage.

Theorem 1: Under Assumption 2, the probability that the solution of SP (8) violates its associated chance constraint at stage k , satisfies

$$\mathbb{P}_k^{\mathcal{S}_k} [V_k(\mathbf{u}_{SP}^*) > \epsilon_k(s_k^*)] \leq \beta_k(S_k), \quad (9)$$

where

$$\beta_k(S_k) := \sum_{s=0}^{S_k-1} \binom{S_k}{s} [1 - \epsilon_k(s)]^{S_k-s}. \quad (10)$$

Proof of Th. 1: The proof follows along the lines of the convex proof [5, Th. 4.1]. In the following, we derive the result for $k = 1$. The proof is analogous for all other k . We use the notation $\boldsymbol{\omega}_k = \{\boldsymbol{\delta}_k^1, \dots, \boldsymbol{\delta}_k^{S_k}\}$ to denote the collection of all samples per stage. Consider the complement of the confidence of the first stage, when the samples of all other stages have been drawn,

$$\mathbb{P}_1^{\mathcal{S}_1} [V_1(\mathbf{u}_{SP}^*(\boldsymbol{\omega}_1)) > \epsilon_1(s_1^*) \mid \boldsymbol{\omega}_2, \dots, \boldsymbol{\omega}_N], \quad \boldsymbol{\omega}_1 \in \Delta^{\mathcal{S}_1}. \quad (11)$$

Under Assumption 2, the samples $\boldsymbol{\omega}_1$ are drawn independently from the samples $\boldsymbol{\omega}_2, \dots, \boldsymbol{\omega}_N$. Moreover, since $\boldsymbol{\omega}_2, \dots, \boldsymbol{\omega}_N$ have been observed, we can merge their respective constraints into the feasible set

$$\tilde{\mathbb{X}}^{2:N} = \prod_{k=2}^N \{\mathbf{x}_k \mid g(\mathbf{x}_k, \boldsymbol{\omega}_k) \leq 0\}. \quad (12)$$

This results in the following modified optimisation problem

$$\min_{\mathbf{u} \in \mathbb{U}} \sum_{k=1}^N J(\mathbf{x}_k, \mathbf{u}_k) \quad (13a)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x} \in \mathbb{X} \quad (13b)$$

$$\mathbb{P}_1[g(\mathbf{x}_1, \boldsymbol{\delta}_1) \leq 0] \geq 1 - \epsilon_1 \quad (13c)$$

$$\mathbf{x}_{2:N} \in \tilde{\mathbb{X}}^{2:N}. \quad (13d)$$

This problem is a nonconvex CCP of the form (3) with one chance constraint, hence we can apply (6), which shows that the confidence of the first stage satisfies the proposed theorem for $k = 1$ and analogous derivations apply for $k = 2, \dots, N$. Even though constraints (13d) are deterministic, the solution to the optimization problem has not changed compared to SP (8). We therefore conclude that the result holds. ■

C. Probabilistic Safety Guarantees

The key insight that makes our approach tractable is that due to the geometric structure of the problem, the free space may be described by only a small subset of the scenarios. To see this, first note that each scenario constraint in (8c) defines a half-space. The collision-free space, if it exists, is formed by the intersection of half-spaces and is convex, as i) each half-space is convex and ii) the intersection of convex constraints is convex. This results in a free space polytope \mathcal{P}_k (see Fig. 2(d)), spanned by those half-spaces that form the boundary of the polytope. We may

Algorithm 1: S-MPCC.

```

1: Compute  $S_k$  from  $\epsilon_k, \bar{s}$ , for all  $k$ 
2: for all  $t = 1, 2, \dots$  do
3:    $\Delta_k^t \leftarrow$  Retrieve uncertainty from perception module
4:   for all  $k = 1, \dots, N$  do
5:     Sample  $\delta_k^i \in \Delta_k^t, i = \{1, \dots, S_k\}$ 
6:     Compute  $\mathbf{A}_k^i, b_k^i$  from (7a) for all samples
7:     Find  $\mathcal{H}_k$  and verify  $|\mathcal{H}_k| \leq \bar{s}$ 
8:   end for
9:    $\mathbf{u}_t \leftarrow$  Solve (14)
10:  Output:  $\mathbf{u}_{t|1}$ 
11: end for

```

define this subset of half-spaces by their indices as

$$\mathcal{H}_k := \{i \mid \exists \mathbf{x}_k \in \mathcal{P}_k, \mathbf{A}_k^T(\delta_k^i, \hat{\mathbf{x}}_k)\mathbf{x}_k = b_k(\delta_k^i, \hat{\mathbf{x}}_k)\}.$$

The usefulness of the set \mathcal{H}_k is twofold. First, we may replace (8c) with only those half-spaces that span polytope \mathcal{P}_k , greatly reducing the size of the online optimization problem. Second, the set \mathcal{H}_k contains indices of the constraints that may be active during optimization and hence the support subsample is bounded by its cardinality, that is, $s_k^* \leq |\mathcal{H}_k|$. We use the latter fact to establish the link between the CCP subject to (7b) and SP (8). There always exists an upper bound, \bar{s} , for the cardinality of \mathcal{H}_k and for our problem we find experimentally that this upper bound \bar{s} is much smaller than the sample size. That is, for uncertainty distributions where the samples are not cluttered at the boundary, only few scenarios are active.

We can now compute sampling size S_k offline, using (i) Theorem 1, (ii) upper bound \bar{s} , (iii) confidence parameter β_k , and (iv) risk ϵ_k . The SP we solve online is given by:

$$\min_{\mathbf{u} \in \mathbb{U}} \sum_{k=1}^N J(\mathbf{x}_k, \mathbf{u}_k) \quad (14a)$$

$$\text{s.t. } \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \mathbf{x} \in \mathbb{X} \quad (14b)$$

$$\mathbf{A}_k^T(\delta_k^i, \hat{\mathbf{x}}_k)\mathbf{x}_k \leq b_k(\delta_k^i, \hat{\mathbf{x}}_k), i \in \mathcal{H}_k. \quad (14c)$$

Algorithm 1 summarizes our method. Online, we sample from the distribution and identify the minimal polytope and the support subsample size (line 4-9). We then solve optimization problem (14) (line 10) and use the first input as control input (line 11). In the following we provide a result for improving performance by discarding outlier scenarios.

Theorem 2: Consider solving the CCP (3) using the SP (4), where after sampling, part of the scenarios are discarded. Suppose that we have a discarding algorithm \mathcal{R} that removes R of the S scenarios, leaving $P = S - R$ scenarios to be considered for the optimization. Let $\epsilon(s)$ be a function such that $\epsilon(P) = 1$ and

$$\beta(S, P) = \binom{S}{P} \sum_{s=0}^{P-1} \binom{P}{s} [1 - \epsilon(s)]^{P-s}.$$

Then the probability that the solution of the SP (4) is infeasible for the original CCP (3) satisfies the upper bound

$$\mathbb{P}^S[V(\mathbf{u}_{SP}^*) > \epsilon(s^*)] \leq \beta(S, P). \quad (15)$$

Proof: Consider the partitioning of the probability space:

$$\Delta_{\mathcal{I}_p}^S = \{\delta^S \in \Delta^S \mid \mathcal{R}(\delta^S) = \mathcal{I}_p\}, \quad (16)$$

The sets $\Delta_{\mathcal{I}_p}^S$ are events where the picking algorithm selected the indices \mathcal{I}_p . Define the set where the risk bound is violated

$$\mathcal{B}_{\mathcal{I}_p} = \{\delta^S \mid \mathcal{R}(\delta^S) = \mathcal{I}_p, V(\mathbf{u}_{\mathcal{I}_p}^*) > \epsilon(s_{\mathcal{I}_p}^*)\}. \quad (17)$$

Notice that the last condition is upper bounded by (6) with $S = P$. But the distribution of the samples is biased due to the samples that were removed from the iid sample set. We obtain the following bound on the biased sample set

$$\mathbb{P}^S[\mathcal{B}_{\mathcal{I}_p}] \leq \sum_{s=0}^{P-1} \binom{P}{s} [1 - \epsilon(s)]^{P-s}. \quad (18)$$

This result holds for all index sets which also contains all the possible biases introduced by \mathcal{R} . Hence, the upper bound

$$\mathbb{P}^S[\mathcal{B}] = \mathbb{P}^S \left[\bigcup_{\mathcal{I}_p} \mathcal{B}_{\mathcal{I}_p} \right] \leq \beta(S, P), \quad (19)$$

is attained by independence of the samples. \blacksquare

Remark 1: Bound (19) is conservative. For example if we pick a random discarding algorithm for \mathcal{R} , then the samples are still iid and we can use (6) directly with $S = P$, giving

$$\sum_{s=0}^{P-1} \binom{P}{s} [1 - \epsilon(s)]^{P-s} = \beta,$$

which is generally much tighter than (19). However, even if the bound is conservative we can use it to remove extreme scenarios, leading to generally better performance.

D. Multiple Dynamic Obstacles and Discs

To apply the strategy above to more than one obstacle, we use the fact that scenario optimization is distribution agnostic. We combine the predictions of the obstacles into a probability space $\Delta_k = [\Delta_k^0 \dots \Delta_k^V]^T$, where samples are denoted $\delta_k = [\delta_k^0 \dots \delta_k^V]^T$. Although the stacked distribution δ_k could be used to model the correlation between the movement of obstacles, we will sample each component separately from individual probability distributions. The chance constraints (7b) need to include all obstacles and are modified as follows

$$\mathbb{P}_k [\mathbf{A}_k^T(\delta_k^v, \hat{\mathbf{x}}_k)\mathbf{x}_k \leq b_k(\delta_k^v, \hat{\mathbf{x}}_k), \forall v] \geq 1 - \epsilon_k, \delta_k \in \Delta_k, \forall k.$$

The rest of the method follows analogously to the single obstacle approach but where the scenarios are drawn for each obstacle, resulting in more scenarios to process before obtaining the free space polytope. In the case of multiple vehicle discs, we formulate multiple chance constraints of the form (2c), one for each collision disc. We apply the method described in this Section per disc as samples for each of the discs are independent.

Algorithm 2: Detailed S-MPCC for (Truncated) Gaussian.

```

1: Determine  $S_k$  from  $\epsilon_k, \beta_k, \bar{s}, R$ 
2:  $u^i \leftarrow \mathbb{U} \times \mathbb{U}, \forall i = \{1, \dots, S_k\}$  (uniform random)
3:  $z_0^i = \sqrt{-2 \ln u_1^i} \cos(2\pi u_2^i), \forall i = \{1, \dots, S_k\}$ 
4:  $z_1^i = \sqrt{-2 \ln u_1^i} \sin(2\pi u_2^i), \forall i = \{1, \dots, S_k\}$ 
5: Verify relevance of samples  $z$ , prune irrelevant
6: for all  $t = 1, 2, \dots$  do
7:   for all  $k = 1, \dots, N$  do
8:      $\delta_k^i \leftarrow (20)$ 
9:      $\hat{\delta}_k^l \leftarrow$  apply  $\mathcal{R}$  to closest  $R + l$  scenarios in  $\delta_k^i$ 
10:     $\mathcal{P}_k \leftarrow$  intersection algorithm on  $\mathcal{H}(\hat{\delta}_k^l) \cup \mathcal{H}_k^{\text{range}}$ 
11:   end for
12: end for

```

V. S-MPCC WITH GAUSSIAN UNCERTAINTIES

A common class of uncertainties are the (truncated) Gaussian uncertainties. This section presents a detailed formulation of Algorithm 1, namely Algorithm 2, one can use in the case of (truncated) Gaussian uncertainty.

The first step of Algorithm 2 is to determine the sample size. We set $\epsilon_k = 1 - 0.9889$, equivalent to the probability mass under the 3σ interval of a bivariate Gaussian (generally considered as safe). Since the risk has logarithmic dependency on β_k [7], β_k is generally small. We pick $\beta_k = 1 \cdot 10^{-6}$, i.e., one in a million SPs may not be feasible for the original CCP³. The removal size $R = 50$ is empirically determined, verifying that outliers are removed. Upper bound \bar{s} is guessed and increased until it is never exceeded in practice. We find $\bar{s} = 20$. Evaluating (19), we are able to pick $S_k \approx 53050$ (line 1). We note that the main dependency of the sample size is the acceptable risk ϵ_k . Sampling more scenarios results in a higher probability of safety, but at the cost of more conservative trajectories and increased computation times.

Instead of online sampling, we may sample a set of parameterized samples offline followed by an online transformation. This reduces the online operations, resulting in lower computation times. We describe this approach for the (truncated) Gaussian case. We generate offline a number of batches with S_k bivariate Gaussian samples, centered at the origin and with $\Sigma = \mathbf{I}$, where \mathbf{I} is the identity matrix (line 2-4). These samples are obtained using the Box-Muller Transformation (BMT) [20], which also allows us to draw radially truncated Gaussian samples by simply changing the support domain of u_1 to $[e^{-\frac{r^2}{2}}, 1]$ [21]. Most of the samples will be in the center of the distribution and will not be relevant online. Hence, we run our online algorithm for scenario selection (explained later), offline and aggregate the set of selected scenarios. Scenarios that are not in this set are pruned offline (line 5). In the 3σ example, approximately 95% of the scenarios are removed offline.

Online, we are only required to transform the offline samples from the standard bivariate normal distribution to the estimated

mean and variance of the uncertainty (line 8), which is computed using

$$\delta_k^i = \mathbf{A}_k^T z_k^i + \mu_k, \quad \mathbf{A}_k^T \mathbf{A}_k = \Sigma_k. \quad (20)$$

We select for each obstacle only one batch of samples. The obstacle predictions are sampled with that batch for all stages and all time steps. This provides the motion planner with consistent constraints. To further reduce the computational load, we search online only for the $l + R$ scenarios closest to considered vehicle position, where we use $l = 150$ in the following experiments, we assume that this set contains the support subsample. We then apply the discarding algorithm \mathcal{R} , which removes the R scenarios furthest from the mean of the distribution (line 9). We construct half-spaces from the remaining l scenario and add four half-spaces to constrain the vehicle in a square workspace. To find the minimal polygon in 2D from this set of half-spaces, we use an intersection based algorithm (line 10). The algorithm explores the intersections in the inner polygon in a counter-clockwise fashion. The lines traveled form the minimal polygon. In the following simulations and experiments, we incorporate our dynamic obstacle avoidance method in the MPCC framework [8]. We introduce a cost term that activates when the robot gets close to the boundaries of the free space polygon, to penalize movement close to pedestrians.

VI. RESULTS

In this section, we present simulation and real-world results for a mobile robot navigating among pedestrians. Moreover, we present a qualitative analysis and performance results of our method against two baselines: MPCC [8] and Collision Avoidance with Deep RL (CADRL) [22].

A. Experimental Settings

Our experimental platform is the Clearpath Jackal robot equipped with an Intel i5 CPU@2.6 GHz. For the robot and pedestrian's localization we have used the OptiTrack system [23]. Our simulations use the open-source ROS implementation of the Jackal Gazebo for the robot simulation and Social Forces model [24] for pedestrian simulation.

To solve SP (14), we use the ForcesPro [25] solver. The robot dynamics are described by a continuous-time second-order unicycle model [26]. The model is discretized with steps of 200 ms. The time horizon is set to 3 seconds divided into 15 stages. The sampling period for control is 50 ms.

B. Simulation Results

We compare the proposed method against two methods for Gaussian uncertainties. The first is a baseline MPCC approach [8] in which the ellipses used to represent the obstacles are obtained from the level sets of a known Gaussian distribution of the uncertainties. For comparison, we use the same tuning for both approaches (the interested reader can refer to [8] for details on the definition of the cost function and general constraints). The main difference between the two approaches is the handling of dynamic obstacles (i.e., ellipsoidal level sets vs. scenario

³Note that the designer can choose to keep safety margin in the obstacle radius such that a failure does not have to result in a collision.

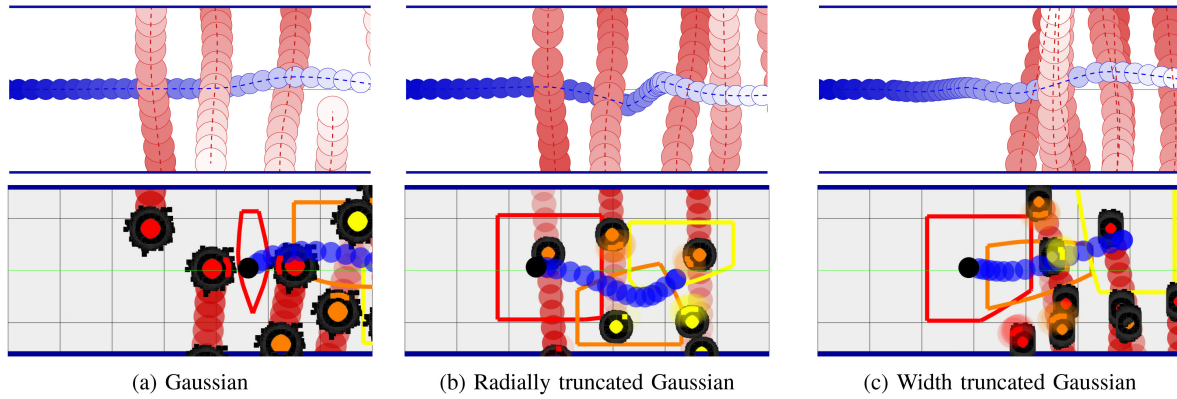


Fig. 3. Simulations using our S-MPCC with 6 crossing pedestrians for 3 types of uncertainties. The top row visualizes the robot (blue) and pedestrian (red) trajectories, where newer positions are depicted with lighter shades. The bottom row visualizes the free space and active samples at stages 1, 8 and 15 in red, orange and yellow. All samples considered online are shown in black. The robot’s current and predicted occupied area are denoted in black and blue, respectively.

TABLE I

STATISTIC RESULTS OF THE PROBABILITY OF COLLISION WITH RESPECT TO THE ESTIMATED UNCERTAINTY FOR THE FIRST STAGE (EVALUATED USING MONTE CARLO SAMPLING) AND VIOLATIONS OF THE SPECIFIED RISK, THE TASK COMPLETION TIME AND THE COMPUTATION TIMES. THE RESULTS ARE COLLECTED FROM 100 SIMULATIONS OF A CROSSING SCENARIO FOR $n \in \{2, 4, 6\}$ PEDESTRIANS

Ped.	Max Collision Prob. Stage 1 (# Violations)			Time to Completion Mean (Std.) [m]			Computation Time Mean (Max) [ms]		
	CADRL	MPCC	S-MPCC	CADRL	MPCC	S-MPCC	CADRL	MPCC	S-MPCC
2	0.71 (12)	0.13 (11)	0.00007 (0)	7.67 (0.97)	7.61 (0.10)	7.14 (0.33)	3.72 (15.49)	1.47 (16.48)	6.48 (22.88)
4	0.83 (17)	0.14 (4)	0.00006 (0)	7.94 (1.04)	8.13 (0.49)	7.54 (0.32)	4.07 (22.29)	1.80 (19.44)	10.32 (43.91)
6	0.86 (43)	0.12 (13)	0.00034 (0)	8.68 (1.99)	8.27 (0.76)	7.40 (0.45)	4.83 (30.31)	2.12 (20.17)	18.37 (65.56)

constraints). The second method for comparison is CADRL [22]. We use the open source ROS implementation in the following simulations. Similar to MPCC we employ ellipsoidal level sets as the collision region of the obstacles.

The simulation environment consists of a straight road where pedestrians are crossing freely, as depicted in Fig. 3. The robot objective is to follow the centerline of the road. We evaluate our method for 2, 4 and 6 pedestrians. The uncertainty of the pedestrian predictions is Gaussian with a variance of $\Sigma = 0.1^2 \mathbf{I}$. We set a pedestrian radius of zero. Fig. 3(a) depicts one simulation of S-MPCC with 6 pedestrians. Aggregated results over 100 simulations are presented in Table I. In all tested cases, collisions are prevented by S-MPCC, while additionally the risk, evaluated over the perceived uncertainty, remains below the specified 3σ threshold. The MPCC method frequently switches between locally optimal trajectories resulting in collisions when it becomes infeasible. CADRL is reactive, which in the simulated environment leads it to positions where collisions may not be avoided. This behavior becomes worse with more obstacles. Interestingly, we find that S-MPCC results in smoother trajectories than both methods which results in earlier arrival at the goal. The downside is that the computation time of our method is higher. The computation time may be decreased by considering only the pedestrians close to the estimate \hat{x}_k . We repeated the simulation with 6 pedestrians in this case. The computation time was reduced to 6.86 ms mean and 40.94 ms maximum.

Evaluation of S-MPCC for non Gaussian uncertainties is depicted in Fig. 3. Here, the previous Gaussian predictions are radially truncated at 3.5σ (Fig. 3(b)) and truncated in their width at 2.5σ (Fig. 3(c)). In this scenario, width truncated uncertainties incorporate the domain knowledge that pedestrians

are expected to cross at a crosswalk. Level set based approaches are not applicable in this case, as the geometry of the level sets depends on the specified risk threshold. We adapt the pedestrian locations to simulate a crosswalk. In contrast to the previous simulations, we specify an obstacle radius of 0.3 m and a variance of $\Sigma = 0.08^2 \mathbf{I}$. We evaluate the probability of collision in the first stage, with respect to the estimated uncertainty over 100 tests using Monte Carlo sampling. We find a maximum risk of 0.00 305 for radial truncation and 0.02 038 for width truncation. The violation of our method in the case of width truncation corresponds to a single case where the horizon is not long enough to correctly assess the risk of the full task a priori. This leads the robot to a state where our method cannot find a trajectory that satisfies the risk bound along the horizon and the optimization becomes infeasible. By increasing the horizon, the risk can be anticipated earlier, improving feasibility at the cost of larger computation times. The maximum risk over the other simulations was at most 0.0070.

C. Real-World Results

We evaluated our method on real navigation situations with pedestrians. In the experiment, the robot navigates on a road following the lane central line when two pedestrians cross the robot’s path. We modeled the noise on the pedestrian predictions as Gaussian distributions truncated at 3.5σ . Fig. 4 provides snapshots of one experiment.⁴

⁴A video of the experiments and simulations accompanies this letter.

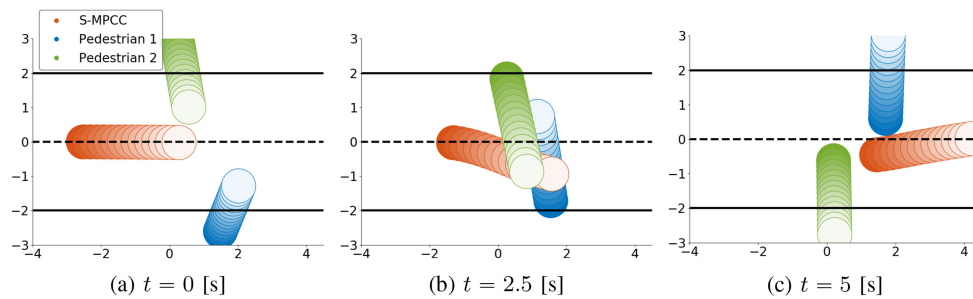


Fig. 4. Experimental results with the robot avoiding two crossing pedestrians. The orange circles depict the robot's plan, while the blue and green circles depict the pedestrians' (constant velocity) predictions. The solid black lines depict the road boundaries.

VII. CONCLUSIONS AND FUTURE WORK

In this letter we presented a Scenario-based Model Predictive Contouring Control (S-MPCC) method for mobile robot motion planning in the presence of dynamic obstacles with arbitrary position distributions. The main idea was to pursue a scenario-based method (translating probabilistic constraints into deterministic ones), generating scenarios from a model of the uncertainty. By using geometry considerations we were able to prune the possible outcomes (scenarios), while providing a bound on the marginal risk with respect to the modeled probability distribution. We demonstrated in simulations that the proposed method outperformed two recent baselines, in the sense that it generated trajectories that were significantly safer and more efficient. This came at a higher processing cost, but the method is still real-time capable. We furthermore illustrated the proposed method in a real-world experiment with a moving robot platform navigating among pedestrians. To further reduce the uncertainties and improve the navigation of the robot, incorporating the interactions between robot and pedestrians would be useful. The risk bounds that our method provides on the modeled uncertainty can still be improved by alleviating the standing assumption that requires our uncertainty models per stage to be independent. Additionally, the risk bound on the planned trajectory is relatively conservative. A tighter bound can be useful for planning safer long term motion, especially when the robot dynamics are slow. Alleviating these limitations are part of our future work.

REFERENCES

- [1] M. Simon, *Inside Amazon Warehouse Where Hum. Machines Become One*, 2019. [Online]. Available: <https://www.wired.com/story/amazon-warehouse-robots/>
- [2] J. Walker, *Self-Driving Car Timeline - Predictions from Top 11 Global Automakers*, 2019. [Online]. Available: <https://emerj.com/ai-adoption-timelines/self-driving-car-timeline-themselves-top-11-automakers/>
- [3] MI News Network, *7 Major Developments Auton. Shipping in 2018*, 2018. [Online]. Available: <https://www.marineinsight.com/know-more/7-major-developments-in-autonomous-shipping-in-2018/>
- [4] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for MAVs in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 776–783, Apr. 2019.
- [5] G. Schildbach, L. Fagiano, and M. Morari, "Randomized solutions to convex programs with multiple chance constraints," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2479–2501, Jan. 2013.
- [6] 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.
- [7] M. C. Campi, S. Garatti, and F. A. Ramponi, "A general scenario theory for nonconvex optimization and decision making," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4067–4078, Dec. 2018.
- [8] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4459–4466, Oct. 2019.
- [9] L. Ferranti *et al.*, "SafeVRU: A research platform for the interaction of self-driving vehicles with vulnerable road users," in *Proc. IEEE Intell. Veh.*, 2019, pp. 1660–1666.
- [10] K. Berntorp, P. Inani, R. Quirynen, and S. Di Cairano, "Motion planning of autonomous road vehicles by particle filtering: Implementation and validation," in *Proc. Amer. Control Conf.*, Jul. 2019, pp. 1382–1387.
- [11] A. Wang, A. Jasour, and B. C. Williams, "Non-Gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6041–6048, Oct. 2020.
- [12] G. Calafiore and M. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 742–753, May 2006.
- [13] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1211–1230, Jan. 2008.
- [14] G. C. Calafiore, "Random convex programs," *SIAM J. Optim.*, vol. 20, no. 6, pp. 3427–3464, Jan. 2010.
- [15] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality," *J. Optim. Theory Appl.*, vol. 148, no. 2, pp. 257–280, Feb. 2011.
- [16] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, Dec. 2014.
- [17] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," Oct. 2019. [Online]. Available: <http://arxiv.org/abs/1910.05449>
- [18] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *Proc. IEEE Intell. Veh.*, Jun. 2018, pp. 1179–1184, iSSN: 1931–0587.
- [19] J. F. P. Kooij, F. Flohr, E. A. I. Pool, and D. M. Gavrila, "Context-based path prediction for targets with switching dynamics," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 239–262, Mar. 2019.
- [20] M. E. Muller and G. E. P. Box, "A note on the generation of random numbers," *Ann. Math. Stat.*, vol. 29, pp. 610–611, 1958.
- [21] L. Martinet, D. Luengo, and J. Miguez, "Efficient sampling from truncated bivariate Gaussians via the box-muller transformation," *Electron. Lett.*, vol. 48, pp. 1533–1534, 2012.
- [22] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *Proc. IEEE/RSS Int. Conf. Intell. Robots Syst.*, Oct. 2018, pp. 3052–3059.
- [23] NaturalPoint, "Optitrack," *Natural Point, Inc.*, 2011. [Online]. Available: <https://optitrack.com/>
- [24] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, 1995, Art. no. 4282, publisher: APS.
- [25] A. Domahidi and J. Jerez, "Forces professional," Jul. 2014. [Online]. Available: <https://embotech.com/FORCES-Pro>
- [26] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots, 2nd ed.* The MIT Press, 2011.