

Risk-Sensitive Sequential Action Control with Multi-Modal Human Trajectory Forecasting for Safe Crowd-Robot Interaction

Haruki Nishimura¹, Boris Ivanovic¹, Adrien Gaidon², Marco Pavone¹, and Mac Schwager¹

Abstract—This paper presents a novel online framework for safe crowd-robot interaction based on risk-sensitive stochastic optimal control, wherein the risk is modeled by the entropic risk measure. The sampling-based model predictive control relies on mode insertion gradient optimization for this risk measure as well as Trajectron++, a state-of-the-art generative model that produces multimodal probabilistic trajectory forecasts for multiple interacting agents. Our modular approach decouples the crowd-robot interaction into learning-based prediction and model-based control, which is advantageous compared to end-to-end policy learning methods in that it allows the robot’s desired behavior to be specified at run time. In particular, we show that the robot exhibits diverse interaction behavior by varying the risk sensitivity parameter. A simulation study and a real-world experiment show that the proposed online framework can accomplish safe and efficient navigation while avoiding collisions with more than 50 humans in the scene.

I. INTRODUCTION

As autonomous robots expand their workspace to cage-free, social environments, they must be designed as safety-critical systems; failures in avoiding collisions with humans sharing the workspace result in catastrophic accidents. Safe and efficient robot navigation alongside many humans is still a challenging problem in robotics, especially due to the potentially unpredictable and uncooperative nature of human motion. We propose an effective solution to this problem via risk-sensitive stochastic optimal control, wherein desired collision avoidance and goal reaching motion is achieved by a cost function, a risk-sensitivity parameter, and dynamic optimization. Specifically, we extend the Stochastic Sequential Action Control (SAC) algorithm [1] to a risk-sensitive setting through the use of exponential disutility [2], the objective often referred to as the entropic risk measure [3]. The proposed sampling-based algorithm, which we name Risk-Sensitive Sequential Action Control (RSSAC), is a stochastic nonlinear model predictive control (NMPC) algorithm that optimally improves upon a given nominal control with a series of control perturbations. Unlike many other control-theoretic stochastic NMPCs [4], [5], [6], RSSAC is not limited to a particular class of distributions such as Gaussian, nor does it need to know the analytical form; it only requires a black-box probabilistic generator. Leveraging this property and recent advances in machine learning for modeling multi-agent behavior, we combine RSSAC with Trajectron++ [7],

Toyota Research Institute (“TRI”) provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. This work was also supported in part by ONR grant number N00014-18-1-2830, a JASSO fellowship, and a Masason Foundation fellowship. The authors are grateful for this support.

¹Haruki Nishimura, Boris Ivanovic, Marco Pavone, and Mac Schwager are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA {hnishimura, borisi, pavone, schwager}@stanford.edu

²Adrien Gaidon is with Toyota Research Institute, Los Altos, CA 94022 USA adrien.gaidon@tri.global

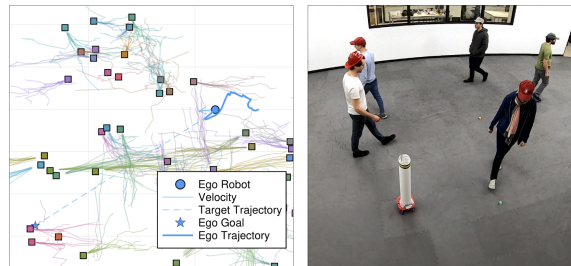


Fig. 1: The proposed RSSAC-Trajectron++ framework is effective for safe robot navigation in a social environment densely populated with humans. (Left) A simulation environment with real human trajectories from the UCY/UNIV scene [8], overlaid with predictions sampled from Trajectron++. (Right) A risk-sensitive robot safely navigating itself alongside 5 humans.

a state-of-the-art generative model for predicting the many possible future trajectories of multiple interacting agents accurately and efficiently. The overall framework constitutes a prediction-control pipeline for safe robot navigation, which is shown to be capable of interacting with more than 50 pedestrians simultaneously while avoiding collisions and steering the robot towards its goal.

The contributions of this paper are two-fold. First, the extension of the Stochastic SAC to risk-sensitive optimal control is presented as a theoretical contribution. This is achieved by generalizing the mode insertion gradient [9], [10] to the entropic risk measure. Second, the combined RSSAC-Trajectron++ framework is presented as a novel modular approach to safe crowd-robot interaction with dynamic collision avoidance. RSSAC takes advantage of the parallelizability of Monte Carlo sampling, leading to an efficient GPU implementation with real-time performance. We conduct simulation studies as well as a real-world experiment to demonstrate that the explicit probabilistic prediction of Trajectron++ and risk-sensitive optimization of RSSAC allow a robot to interact with many humans safely and efficiently. They also reveal that the risk sensitivity parameter adds an additional degree of freedom in determining the type of interaction behavior exhibited by the robot, e.g. yielding to or crossing in front of oncoming humans; this diverse behavior is not achieved by tuning cost function parameters.

II. RELATED WORK

We first review relevant work in dynamic collision avoidance and safe robot navigation in Section II-A. We then focus on Stochastic SAC (Section II-B) and data-driven trajectory forecasting methods (Section II-C), on which our RSSAC-Trajectron++ framework is built.

A. Dynamic Collision Avoidance and Safe Robot Navigation

There exists a vast body of literature in the field of dynamic collision avoidance and safe robot navigation. Coordinated collision avoidance is widely studied in the multi-robot

systems literature, with methods including mixed integer programming [11], reciprocal velocity obstacles [12], buffered Voronoi cells [13], [14], Nash-equilibria-based differential games [15], and safety barrier certificates [16]. These methods do not explicitly model uncoordinated agents or motion uncertainty, which are both crucial aspects in human motion; later in Section V we observe collisions for [14] due to humans breaking the coordination assumption. A notable exception is [17], where the authors propose probabilistic safety barrier certificates for bounded motion uncertainty. An alternative approach suited to bounded uncertainty is the Hamilton-Jacobi (HJ) reachability analysis [18], [19], although it suffers from the curse of dimensionality in solving the Hamilton-Jacobi-Isaacs equation.

Another common approach to modeling motion uncertainty is via the use of probability distributions. MDP and POMDP frameworks have been applied to robot navigation and collision avoidance [20], [21] and are best suited to discrete action spaces. Stochastic Optimal Control (SOC), especially sampling-based model predictive control (MPC), is also gaining increasing attention due to its ability to handle stochastic uncertainty and incorporate prediction of possible future outcomes. Recent methods applied to safe robot navigation and dynamic collision avoidance include exhaustive search with motion primitives [22], path-integral control [5], and information-theoretic control [23]. While the latter two approaches can find approximately optimal solutions under certain assumptions, they require the form of distribution to be Gaussian. In addition, it is often not satisfactory to only optimize for the expected cost, especially for safety-critical systems. There exist many possible ways to endow robots with risk-awareness [3], such as chance-constrained optimization [24], conditional value at risk (CVaR) [25], and entropic risk measure [26]. All of these methods come with advantages and drawbacks, but in general there is a trade-off between the complexity of modeling assumptions and the statistical safety assurances. Our work employs the entropic risk measure which has been extensively studied in control theory [27], [28]. Albeit not yielding statistical safety assurances, our algorithm can compute control inputs in real-time for highly dynamic environments with multimodal predictive distributions, which is quite challenging for chance constrained optimization or CVaR-based methods.

Lastly, we briefly highlight similarities and differences of our work from policy learning methods such as imitation learning [29] or end-to-end deep reinforcement learning [30], [31]. Similar to these methods, our framework applies machine learning to extract features from multi-agent interactions. However, we use these features to make explicit probabilistic predictions about other agents which are then incorporated into optimization by the model-based controller. Our modular approach has two advantages. First, explicit prediction brings about better interpretability and explainability of the resulting interaction. Second, our model-based controller allows the robot's behavior to be changed at run time by simply modifying the cost function design or the risk sensitivity parameter, while other methods would need re-training to do so as the policy itself is data-driven.

B. Stochastic Sequential Action Control

SAC in its original form is a deterministic model predictive control algorithm for nonlinear hybrid systems [10], [32].

In contrast to local trajectory optimization, at each time the algorithm perturbs a given (open-loop or closed-loop) nominal control to optimize a local improvement of the total receding-horizon cost, a quantity known as the mode insertion gradient [9]. SAC is shown to yield highly efficient control computation with high-frequency replanning, which often achieves better performance than local trajectory optimization methods. In prior work [1], [33], we have generalized the mode insertion gradient to stochastic systems and solved challenging belief space planning problems, where our approach significantly outperforms other state-of-the-art methods under both sensing and motion uncertainty. The proposed RSSAC algorithm is an extension of this algorithm to risk-sensitive optimal control in which the cost functional is the entropic risk measure. This makes the algorithm more suitable for problems where safety is concerned, such as the crowd-robot interaction one that we address in this paper.

C. Multi-Agent Trajectory Modeling from Data

Since human behavior is rarely deterministic or unimodal, deep learning-based generative approaches have emerged as state-of-the-art trajectory forecasting methods, due to recent advancements in deep generative models [34], [35]. They mostly use a recurrent neural network architecture with a latent variable model, such as a Conditional Variational Autoencoder (CVAE) [34], to explicitly encode multimodality (e.g. [36]), or a Generative Adversarial Network (GAN) [35] to do so implicitly (e.g. [37]). Trajectron++ [7] belongs to the former, making use of a CVAE to explicitly model the multimodality of human behavior with a latent variable. Specifically, it uses a discrete Categorical latent variable which aids in performance as well as interpretability, since different behavior modes are explicitly represented and able to be visualized. Designed for downstream robotic motion planning and control tasks, Trajectron++ can also produce predictions that are conditioned on candidate future motion plans of the ego robot. We leverage this capability in a real-world experiment in Section V and show that the robot-future-conditional prediction significantly improves both safety and efficiency of the robot navigation.

III. PROBLEM STATEMENT

A. Dynamics Model

Following [1], [33], this work assumes a discrete-time dynamical system model for other agents (i.e. humans) while employing a continuous-time one for the robot. This modeling assumption is practical, as 1) in general the robot receives information of its surrounding environment at a much lower frequency than actuation; and 2) with this representation one can naturally handle potentially agile robot dynamics without coarse time discretization. Let $x(t) \in \mathbb{R}^n$ denote the state of the robot at time t , whose dynamics are modeled by

$$\dot{x}(t) = f(x(t)) + H(x(t))u(t), \quad (1)$$

where $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control and \mathcal{U} is a bounded convex set. We assume that the dynamics are deterministic and control-affine. As the robot navigates in the environment, it receives position information on humans in the scene and updates it at discrete times $\{t_k\}_{k \geq 0}$, with interval time Δt_o . Similarly to related works [13], [14] we do not assume velocity information to be available to the robot. Let $p^i(t_k) \in \mathbb{R}^2$

denote the position of the human with label $i \in \{1, \dots, N\}$. From t_{k-1} to t_k , the position $p^i(t_{k-1})$ changes to $p^i(t_k)$ with difference y_k^i . From the robot's continuous-time perspective, this is viewed as a periodic jump discontinuity given by

$$\begin{cases} p^i(t_k) = p^i(t_k^-) + y_k^i \\ p^i(t) = p^i(t_k) \end{cases} \quad \forall t \in [t_k, t_{k+1}), \quad (2)$$

where t_k^- is infinitesimally smaller than t_k . The dynamics of the joint state $s(t) = (x(t), p^1(t), \dots, p^N(t))$ are specified by (1) and (2), which together constitute a hybrid dynamical system with time-driven switching. We treat N as a fixed number in this section, but relax this assumption later and handle a time-varying number of humans via replanning.

The robot plans its control actions over a finite horizon $[t_0, t_T]$ while the humans make stochastic transitions $\{y_k^i\}_{(1 \leq i \leq N, 1 \leq k \leq T)}$. Stacking these transition variables, we obtain a random vector with distribution \mathcal{D} . This distribution can be dependent on other variables such as past history of interactions and/or future motion plans of the robot. For the sake of planning, we only require that samples drawn from \mathcal{D} are available. In this work we choose to model \mathcal{D} with Trajectron++ [7], due to its superior performance over other trajectory forecasting methods as well as its capability to produce robot-future-conditional predictions. The reader is referred to [7] for further details on Trajectron++.

B. Optimal Control Problem

Consider a finite horizon optimal control cost of the form

$$J = \int_{t_0}^{t_T} c(s(t), u(t)) dt + h(s(t_T)), \quad (3)$$

where $c(\cdot)$ is the instantaneous cost function and $h(\cdot)$ is the terminal cost function. In this work we assume that $c(\cdot)$ is the well-known LQ tracking cost plus a collision cost term:

$$c(s(t), u(t)) = \frac{1}{2}(x(t) - r(t))^T Q(x(t) - r(t)) + \frac{1}{2}u(t)^T R u(t) + c_{\text{col}}(s(t)), \quad (4)$$

where $Q = Q^T \succeq 0$ and $R = R^T \succ 0$ are weight matrices, and $c_{\text{col}}(\cdot) \geq 0$ is a collision penalty function that is continuously differentiable, bounded, and has bounded gradient with respect to x . The reference trajectory r is assumed to be given, possibly from a high level global planner. Similarly, the terminal cost $h(\cdot)$ is defined by

$$h(s(t_T)) = \frac{\beta}{2}(x(t_T) - r(t_T))^T Q(x(t_T) - r(t_T)) + \beta c_{\text{col}}(s(t_T)), \quad (5)$$

where $\beta \geq 0$ determines the relative weight between the terminal and instantaneous costs.

Having specified the cost functional J , we define the (risk-neutral) stochastic optimal control problem as follows.

Problem 1 (Risk-Neutral Stochastic Optimal Control):

$$\begin{aligned} & \underset{u}{\text{minimize}} && \mathbb{E}_{\mathcal{D}}[J] \\ & \text{subject to} && (1), (2) \quad \forall i \in \{1, \dots, N\} \quad \forall k \in \{0, \dots, T\} \\ & && x(t_0) = x_0, p^i(t_0) = p_0^i \quad \forall i \in \{1, \dots, N\} \\ & && u(t) \in \mathcal{U} \quad \forall t \in [t_0, t_T] \end{aligned}$$

C. Entropic Risk Measure

The formulation of Problem 1 ignores the safety-critical aspect of collision avoidance, as simply optimizing the expected value fails to take into account the shape of the distribution. A remedy is to introduce the following entropic risk measure with risk sensitivity parameter $\sigma > 0$:

$$R_{\mathcal{D}, \sigma}(J) \triangleq \frac{1}{\sigma} \log(\mathbb{E}_{\mathcal{D}}[e^{\sigma J}]). \quad (6)$$

As long as the robot trajectory x is bounded for all admissible control, J becomes a bounded random variable and $R_{\mathcal{D}, \sigma}(J)$ is finite. It is known that this transformation approximately decouples the mean and variance:

$$R_{\mathcal{D}, \sigma}(J) \approx \mathbb{E}_{\mathcal{D}}[J] + \frac{\sigma}{2} \text{Var}_{\mathcal{D}}(J) \quad (7)$$

for small $\sigma \text{Var}_{\mathcal{D}}(J)$ [2]. The meaning of σ is now clear; it is a parameter that determines how much we care about the variability of the cost in addition to the mean. Larger σ increases risk sensitivity, while the risk-neutral objective $\mathbb{E}_{\mathcal{D}}[J]$ is recovered as $\sigma \rightarrow 0^+$.

Replacing the expectation in Problem 1 with (6), we obtain the following risk-sensitive optimal control problem:

Problem 2 (Risk-Sensitive Stochastic Optimal Control):

$$\begin{aligned} & \underset{u}{\text{minimize}} && R_{\mathcal{D}, \sigma}(J) \\ & \text{subject to} && \text{the same constraints as in Problem 1} \end{aligned}$$

IV. RISK-SENSITIVE SEQUENTIAL ACTION CONTROL

A. Review of Stochastic Sequential Action Control

Even solving Problem 1 is intractable due to potential non-convexity in c_{col} and complexity in \mathcal{D} . An efficient, approximate MPC solution can be obtained via Stochastic SAC [1], [33]. In this framework, we seek the optimal perturbation of a given nominal control such that the expected value of the mode insertion gradient [9], [10], which quantifies local effects of the perturbation on the cost functional J , is minimized. We assume that the nominal control is an open-loop control schedule u (although it is straightforward to extend our method to deal with closed-loop nominal control policies [1], [33]). The perturbed control is defined as

$$u^\epsilon(t) \triangleq \begin{cases} v & \text{if } t \in (\tau - \epsilon, \tau] \\ u(t) & \text{otherwise,} \end{cases} \quad (8)$$

where $v \in \mathcal{U}$, $\tau \in (t_0, t_T)$, $\epsilon \geq 0$ are the perturbation parameters. The perturbation u^ϵ yields deterministic, perturbed state trajectory x^ϵ and cost J^ϵ under a specific sample from \mathcal{D} . The mode insertion gradient is the sensitivity of the cost J to the perturbation duration ϵ , with v and τ fixed:

$$\left. \frac{\partial_+ J}{\partial \epsilon} \right|_{\epsilon=0} \triangleq \lim_{\epsilon \rightarrow 0^+} \frac{J^\epsilon - J}{\epsilon}. \quad (9)$$

The value of the mode insertion gradient is given by

$$\begin{aligned} \left. \frac{\partial_+ J}{\partial \epsilon} \right|_{\epsilon=0} &= \frac{1}{2} v^T R v + \rho(\tau)^T H(x(\tau))(v - u(\tau)) \\ &\quad - \frac{1}{2} u(\tau)^T R u(\tau), \end{aligned} \quad (10)$$

where x is the robot state trajectory induced by nominal control u , and ρ is the adjoint variable matching the dimensionality of the robot state. Specifically, it follows the ODE:

$$\dot{\rho}(t) = -\frac{\partial}{\partial x} c(s(t), u(t)) - \left(\frac{\partial}{\partial x} f(x(t)) + \frac{\partial}{\partial x} H(x(t))u(t) \right)^T \rho(t), \quad (11)$$

with boundary condition $\rho(t_T) = \frac{\partial}{\partial x} h(s(t_T))$. As the joint state $s(t)$ is a random vector due to stochasticity in human motion, so is $\rho(t)$ and (11) is valid under the specific sample from \mathcal{D} . Taking the expectation yields the expected mode insertion gradient $\mathbb{E}_{\mathcal{D}}[\partial_+ J / \partial \epsilon |_{\epsilon=0}]$, which is obtained by replacing $\rho(\tau)$ in (10) with $\mathbb{E}_{\mathcal{D}}[\rho(\tau)]$.

B. Generalized Mode Insertion Gradient

Under a weak regularity condition on the dynamics function (1) and for the cost J defined by (3), (4), (5), one can show¹ that the robot trajectory x is bounded for all admissible control, and that the perturbed cost J^ϵ is Lipschitz continuous in $\epsilon \geq 0$:

$$|J^\epsilon - J| \leq \epsilon K, \quad (12)$$

under the same random sample from \mathcal{D} for J and J^ϵ . The value of the constant K is not dependent on the samples. Therefore, the dominated convergence theorem allows us to interchange the derivative and the expectation:

$$\mathbb{E}_{\mathcal{D}} \left[\frac{\partial_+ J}{\partial \epsilon} \Big|_{\epsilon=0} \right] = \frac{\partial_+ \mathbb{E}_{\mathcal{D}}[J]}{\partial \epsilon} \Big|_{\epsilon=0}. \quad (13)$$

The right-hand side of (13) is the mode insertion gradient generalized to the case of risk-neutral stochastic optimal control (i.e. Problem 1). If the optimal value with respect to v is negative for some τ , then there exists a sufficiently small ϵ for which the perturbation defined by (v, τ, ϵ) will reduce the expected cost. If instead it is zero, we can think of the nominal control as satisfying a local optimality condition.

C. Extension to Entropic Risk Measure

Sections IV-A and IV-B have provided a summary of prior work [1], [33]. Now we are set to derive the generalized mode insertion gradient for the entropic risk measure $R_{\mathcal{D}, \sigma}(J)$, which is a novel contribution of this paper.

Lemma 1: Suppose that the total cost J satisfies (12). Then, for $\sigma > 0$ the following relation holds:

$$\mathbb{E}_{\mathcal{D}} \left[\frac{\partial_+ e^{\sigma J}}{\partial \epsilon} \Big|_{\epsilon=0} \right] = \frac{\partial_+ \mathbb{E}_{\mathcal{D}}[e^{\sigma J}]}{\partial \epsilon} \Big|_{\epsilon=0}. \quad (14)$$

Proof: Let J^ϵ and \bar{J} be the perturbed and the nominal cost, respectively. We have

$$|e^{\sigma J^\epsilon} - e^{\sigma \bar{J}}| = \frac{|e^{\sigma(J^\epsilon - \bar{J})} - 1|}{e^{\sigma \bar{J}}} \leq e^{\sigma |J^\epsilon - \bar{J}|} - 1, \quad (15)$$

where we used $|e^x - 1| \leq e^{|x|} - 1$ and $e^{\sigma J} \geq 1$. Substituting (12) and dividing by $\epsilon > 0$, we obtain

$$\frac{|e^{\sigma J^\epsilon} - e^{\sigma \bar{J}}|}{\epsilon} \leq \frac{e^{\sigma K \epsilon} - 1}{\epsilon} \quad (16)$$

¹A more general and detailed analysis is provided in [33] where we provide sufficient conditions for (13) to hold.

Let $g(\epsilon)$ denote the right-hand side of this inequality, which is a strictly monotone increasing function for $\epsilon > 0$. Now let $|f_\epsilon(J)|$ denote the left-hand side as a function of J parameterized by ϵ . Take some positive ϵ_0 and a sequence $\{\epsilon_n\}_{n \geq 0}$ converging to 0^+ as $n \rightarrow \infty$. It follows that $\forall n \geq 0, |f_{\epsilon_n}(J)| \leq g(\epsilon_0)$. The dominated convergence theorem applies as $g(\epsilon_0)$ is a finite, deterministic constant:

$$\mathbb{E}_{\mathcal{D}}[\lim_{n \rightarrow \infty} f_{\epsilon_n}(J)] = \lim_{n \rightarrow \infty} \mathbb{E}_{\mathcal{D}}[f_{\epsilon_n}(J)], \quad (17)$$

which is equivalent to (14). \blacksquare

Theorem 1 (Mode Insertion Gradient of Entropic Risk):

Suppose that the regularity condition mentioned in Section IV-B is met so the robot trajectory x is bounded under admissible control and the cost J satisfies (12). Then, for fixed v and τ the mode insertion gradient of the entropic risk measure $\frac{\partial_+}{\partial \epsilon} R_{\mathcal{D}, \sigma}(J) |_{\epsilon=0}$ exists and is given by

$$\frac{\partial_+}{\partial \epsilon} R_{\mathcal{D}, \sigma}(J) \Big|_{\epsilon=0} = \frac{1}{2} v^T R v + \frac{\mathbb{E}_{\mathcal{D}}[e^{\sigma J} \rho(\tau)]^T}{\mathbb{E}_{\mathcal{D}}[e^{\sigma J}]} H(x(\tau))(v - u(\tau)) - \frac{1}{2} u(\tau)^T R u(\tau), \quad (18)$$

where J inside the expectations is the cost value under the nominal control u .

Proof: As the robot trajectory is bounded, J is a bounded random variable and the value of $R_{\mathcal{D}, \sigma}(J)$ is finite. The chain rule gives

$$\frac{\partial_+}{\partial \epsilon} R_{\mathcal{D}, \sigma}(J) \Big|_{\epsilon=0} = \frac{1}{\sigma \mathbb{E}_{\mathcal{D}}[e^{\sigma J}]} \frac{\partial_+ \mathbb{E}_{\mathcal{D}}[e^{\sigma J}]}{\partial \epsilon} \Big|_{\epsilon=0} \quad (19)$$

$$= \frac{1}{\mathbb{E}_{\mathcal{D}}[e^{\sigma J}]} \mathbb{E}_{\mathcal{D}} \left[e^{\sigma J} \frac{\partial_+ J}{\partial \epsilon} \Big|_{\epsilon=0} \right], \quad (20)$$

where we also used Lemma 1. Substituting (10) and simplifying the terms complete the proof. \blacksquare

The mode insertion gradient of entropic risk (18) is a generalization of the stochastic mode insertion gradient (13). Indeed, for the risk neutral case (i.e. $\sigma = 0$) the two equations match. This enables us to extend the Stochastic SAC algorithm to risk-sensitive optimal control problems without changing the structure of the algorithm.

D. RSSAC Algorithm

The core of RSSAC is the following optimization problem, which substitutes Problem 1 (if $\sigma = 0$) and 2 (if $\sigma > 0$).

Problem 3 (Mode Insertion Gradient Optimization):

$$\begin{aligned} & \underset{v, \tau}{\text{minimize}} && \frac{\partial_+}{\partial \epsilon} R_{\mathcal{D}, \sigma}(J) \Big|_{\epsilon=0} \\ & \text{subject to} && \tau \in (t_0 + t_{\text{calc}}, t_T) \\ & && v(\tau) \in \mathcal{U} \quad \forall \tau \in (t_0, t_T), \end{aligned}$$

where t_{calc} is a computation time budget. The objective can be evaluated by Monte Carlo sampling the joint dynamics (1), (2) under the nominal control and backward integration of the adjoint dynamics (11). Fixing τ , Problem 3 is a quadratic minimization over v under a convex constraint. The optimal value $v^*(\tau)$ can be obtained analytically for simple constraints such as a box constraint or a norm inequality constraint with a scaled identity matrix R . Optimizing τ is

Algorithm 1 Control Computation with RSSAC

INPUT: Initial joint state $s(t_0)$, reference trajectory r , nominal control schedule $u(t)$ for $t \in [t_0, t_T]$.

OUTPUT: Optimally perturbed control schedule u^ϵ

- 1: **for** $j = 1:M$ **do**
 - 2: Forward-simulate the joint dynamics (1), (2) while sampling human transitions $\{y_k^i\}_{(1 \leq i \leq N, 1 \leq k \leq T)}^j$ from \mathcal{D}
 - 3: Compute sampled cost J^j
 - 4: Backward-simulate adjoint robot trajectory ρ^j with sampled human transitions (11)
 - 5: **end for**
 - 6: Compute Monte Carlo estimate: $\mathbb{E}_{\mathcal{D}}[e^{\sigma J} \rho] \approx \frac{1}{M} \sum_{j=1}^M e^{\sigma J^j} \rho^j$ and $\mathbb{E}_{\mathcal{D}}[e^{\sigma J}] \approx \frac{1}{M} \sum_{j=1}^M e^{\sigma J^j}$
 - 7: Solve Problem 3
 - 8: Specify ϵ or search by re-simulating the dynamics
 - 9: $u^\epsilon \leftarrow \text{PerturbControlSchedule}(u, v^*, \tau^*, \epsilon)$ (8)
 - 10: **return** u^ϵ
-

achieved by solving $v^*(\tau)$ multiple times and searching for the minimum value. There is only a finite number of τ to consider since in practice we use numerical integration, such as the Euler scheme with some discrete step size, to integrate the robot dynamics. The optimal mode insertion gradient is always non-positive, and if it is negative then there exists some $\epsilon > 0$ such that the entropic risk measure is reduced as a result of the control perturbation. The value of ϵ can be either specified or searched. If instead the mode insertion gradient is zero, we set $\epsilon = 0$ and choose not to perturb.

The pseudo-code of the RSSAC algorithm is presented in Algorithm 1. Importantly, the Monte Carlo sampling part is naturally parallelizable. Note that the risk sensitivity has no effect unless we draw at least $M = 2$ samples. The replanning happens at every Δt_r seconds, allowing for a variable number of humans to be considered over time. The complexity of the algorithm is $\mathcal{O}(NMT)$ where N denotes the number of humans, M the number of samples, and T the planning horizon.

E. Implementation Details

In our implementation, we use the double integrator model for the robot's dynamics: $x(t) = (x_p(t), x_v(t))$, $\dot{x}_p(t) = x_v(t)$, $\dot{x}_v(t) = u(t)$. These dynamics are integrated using the Euler scheme with step size $\Delta t_c = 0.02[s]$. The cost weight matrices are $Q = \text{Diag}(0.5, 0.5, 0, 0)$ and $R = 0.2I_{2 \times 2}$. The collision cost is the following sum of Gaussian functions centered at each human:

$$c_{\text{col}}(s(t)) = \sum_{i=1}^N \alpha \exp\left(-\frac{\|x_p(t) - p^i(t)\|_2^2}{2\lambda}\right), \quad (21)$$

with peak parameter $\alpha = 100$ and bandwidth parameter $\lambda = 0.2$. The relative weight of the terminal cost is set to $\beta = 0.1$. The reference trajectory r is a straight line connecting the initial robot position to the goal position at a constant target speed, and is replanned whenever $\|x(t) - r(t)\|_2 > 2[m]$. The control constraint is $\|u(t)\|_2 \leq u_{\text{max}}$ with $u_{\text{max}} = 5.0, 2.5[m/s^2]$ in the simulation and the real-world experiment, respectively. Monte Carlo sampling is parallelized on a GPU with sample size $M = 30$. The planning horizon is $4.8[s]$, which corresponds to $T = 12$

steps of human motion prediction with measurement interval $\Delta t_o = 0.4[s]$.

In contrast to prior work [1], [33], we searched for ϵ from $\{0, 1e^{-3}, 2e^{-3}, 4e^{-3}, 8e^{-3}, 1.6e^{-2}, 2e^{-2}, 4e^{-2}, 8e^{-2}\}[s]$ by re-simulating the perturbed dynamics. The nominal control u is a simple MPC-style search algorithm and takes the form

$$u(t) = \begin{cases} u_s(t) & \text{if } t \in [t_0 + t_{\text{calc}}, t_0 + t_{\text{calc}} + \Delta t_o] \\ u_{\text{pr}}(t) & \text{otherwise,} \end{cases} \quad (22)$$

where u_{pr} is the perturbed control from the previous iteration, and u_s is either the same as u_{pr} or chosen from a set of constant control inputs $\{(a \cos(\theta), a \sin(\theta))\}$ with $a \in \{0.4u_{\text{max}}, 0.8u_{\text{max}}\}$ and $\theta \in \{0, \pi/4, \pi/2, \dots, 2\pi\}$. The best u_s is chosen based on the evaluation of $R_{\mathcal{D}, \sigma}(J)$ for each nominal control candidate using the Monte Carlo samples. This nominal search is similar to the iterative update scheme presented in [32] in that the previously-computed perturbation is used in the next iteration, but we insert u_s prior to running Algorithm 1. It can also be considered as a simplified version of a tree search with motion primitives [22] with only 17 control choices and tree depth 1. Note that if robot-future-conditional prediction is used, the distribution \mathcal{D} is different for each candidate nominal control, allowing the robot to consider the effect of various future robot trajectories on human behavior. We found this simple nominal search to be effective compared to constant nominal control [10], [32] while retaining low computational cost.

We have implemented all the control code in Julia and achieved real-time performance with replanning interval $\Delta t_r = t_{\text{calc}} = 0.1[s]$.

V. RESULTS

A. Simulation Results

We evaluated the performance of RSSAC in simulation and compare against three baseline collision avoidance algorithms: 1) LQ-Tracking with Buffered Input Cell (BIC) [14], 2) Nominal Search (Section IV-E) Only, 3) Exhaustive Tree Search with Motion Primitives [22]. BIC is a reciprocal collision avoidance method that has similar computational complexity to the Velocity Obstacle (VO) approaches. Unlike VO, BIC does not require the velocity information of other agents to be available and is applicable to high-order linear dynamics. As our robot is a double-integrator and only obtains position measurements, BIC is a more suitable reciprocal collision avoidance approach than VO. BIC is minimally invasive in that an arbitrary control input is projected onto a convex polygon in the control space. Collision avoidance is guaranteed as long as all the agents respect their own BIC constraint. We use the LQ-Tracking cost (without c_{col}) to solve a deterministic optimal control problem and the BIC constraint adjusts the control input based on the current positions of humans. The robot is said to be in collision if it is within $40[cm]$ from a human, but we set this threshold to $80[cm]$ for planning with BIC only to give it an extra safety margin. The exhaustive search is similar to the nominal search in that it uses samples from \mathcal{D} , but the tree depth is $T = 4$ and at each depth the constant control is chosen from $\{(a \cos(\theta), a \sin(\theta))\}$ with $a \in \{0, 0.6u_{\text{max}}\}$ and $\theta \in \{0, \pi/4, \pi/2, \dots, 2\pi\}$. This results in $9^4 = 6561$ sequences of control inputs to be considered. Despite parallelization on a GPU and a long replanning interval of

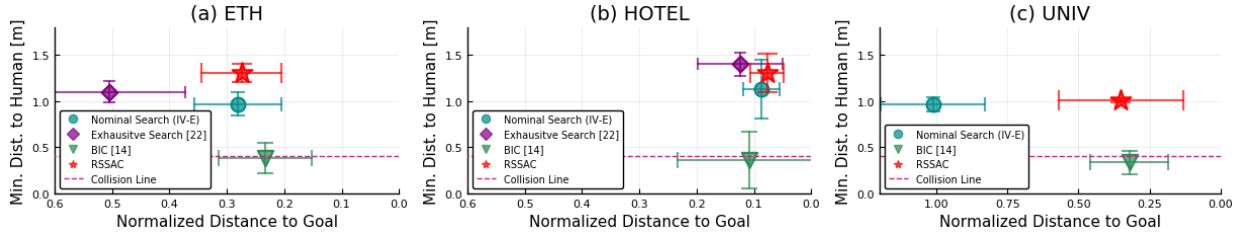


Fig. 2: Quantitative results from 100 runs show that risk-neutral (i.e. $\sigma = 0$) RSSAC further improves the performance of Nominal Search Only as the theory suggests, achieving both safety and efficiency. Note that the farther up and to the right, the better, as the x-axis is flipped. Exhaustive Search could not scale to the UNIV scene with more than 50 humans. BIC resulted in multiple collisions. Error bars show standard deviation.

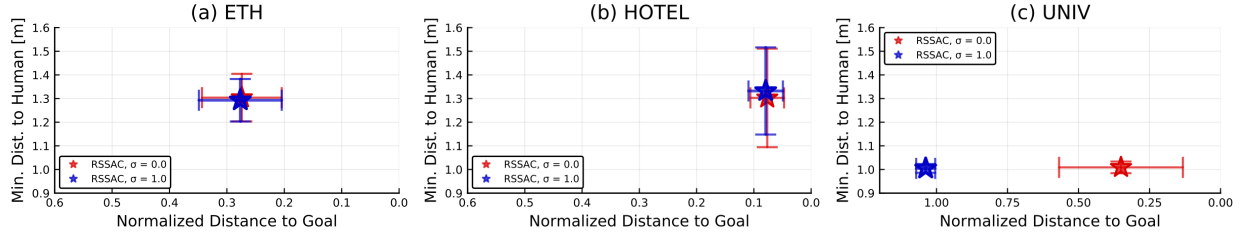


Fig. 3: Compared to the risk-neutral case in Fig. 2, RSSAC with $\sigma = 1.0$ significantly reduces the standard deviation of the minimum robot-human distance by 11%, 12%, and 24% in (a), (b), and (c), respectively. The risk sensitivity trades off the stochastic collision cost and the deterministic tracking cost, which results in increased standard deviation in the x-axis in (a) and (b), and overall distance increase in (c) where the scene was most densely-populated.

$\Delta t_r = t_{\text{calc}} = 0.4[s]$, Exhaustive Search never achieved real-time performance. RSSAC, Nominal Search, and Exhaustive Search all used $M = 30$ prediction samples (per human) drawn from the Trajectron++ model [7].

For this evaluation, we used three distinct sequences from the publicly available ETH [38] and UCY [8] pedestrian motion datasets. They consist of real pedestrian trajectories with rich multi-human interaction scenarios. Each sequence that we used is a series of consecutive frames clipped from the ETH/ETH Test scene, the ETH/HOTEL Test scene, and the UCY/UNIV Test scene. For brevity, we refer to these as the “ETH,” “HOTEL,” and “UNIV” sequences, respectively. In each scene, the robot starts from a collision free configuration and moves towards a specified goal while avoiding collisions. The ETH sequence is $10[s]$ long and has 16 total pedestrians. The HOTEL sequence is also $10[s]$ long and has 8 total pedestrians. The UNIV sequence is the most challenging with 95 total pedestrians over a horizon of $20[s]$; there always exists 36 to 54 pedestrians simultaneously in each frame (see Fig. 1). Note that none of those pedestrians recognize or react to the robot as their motion is a replay from the data. This is the so-called “invisible robot” setting [31] where the robot has to predict Human-Human interaction only. For this reason, we did not condition the prediction of Trajectron++ on the nominal control candidates of the robot.

Trajectron++ was trained following the same procedure as detailed in [7]. The model was trained for 2000 steps with the Adam [39] optimizer and a leave-one-out approach. That is, the model was trained on all datasets but the one used for evaluation. This training was done off-line prior to running RSSAC. At run time, RSSAC fetches prediction samples from the trained model every $\Delta t_o = 0.4[s]$ and uses them for the mode insertion gradient optimization. All the online computation was performed on a desktop computer with an Intel Xeon(R) CPU ES-2650 v3, 62.7 GiB memory, and a GeForce GTX 1080 graphics card.

Fig. 2 shows the statistical simulation results of RSSAC

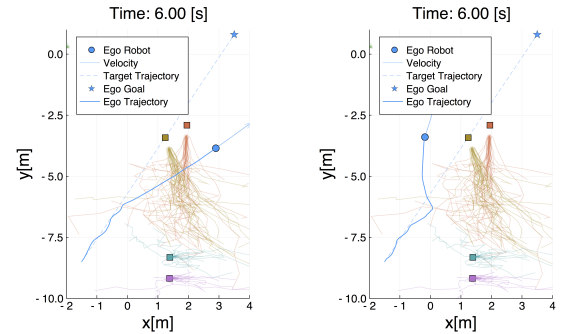


Fig. 4: Qualitative comparison of RSSAC with $\sigma = 0$ (left) and $\sigma = 1.0$ (right) in the HOTEL scene. These results differ in the minimum robot-human distance by only $3[cm]$ and the normalized goal distance by 0.01, but the risk-sensitive robot (right) yields to potentially conflicting humans as opposed to the risk-neutral robot (left). Both simulations used the same random seed. Sampled predictions from Trajectron++ are also depicted.

and the baseline methods. RSSAC, Nominal Search, Exhaustive Search were implemented with $\sigma = 0$. The x-axis shows the final distance between the robot and its goal normalized by the initial distance. The y-axis is the minimum distance between the robot and a human. These performance metrics measure the safety and efficiency of robot navigation, and are often in conflict. For each method we performed 100 runs with different random seeds. The goal of the robot was also randomized. As can be seen, RSSAC was both the safest and the most efficient among those methods that ran in real-time. BIC ended up in multiple collisions in all the three cases as the humans violated the reciprocity assumption. We did not test Exhaustive Search for the UNIV case due to its poor scalability; the computation took about $10\times$ longer than the allocated time budget. We also note that RSSAC did further improve the performance of the Nominal Search, which itself was already achieving reasonably safe navigation.

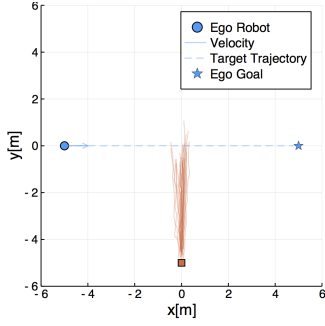


Fig. 5: A synthetic intersection scenario with a human. The prediction is drawn from a linear Gaussian model with a constant mean velocity.

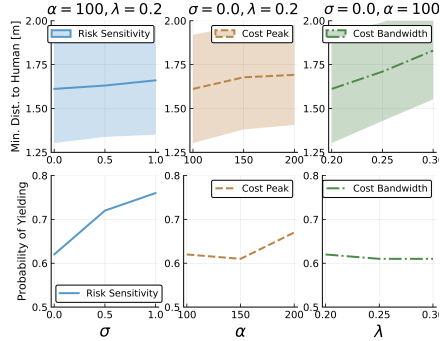


Fig. 6: Minimum robot-human distance (top) and empirical probability of yielding (bottom) for the synthetic intersection scenario. Changing the risk-sensitivity (left) consistently affected whether or not the robot yields, while the other two cost tuning parameters (middle and right) did not.

B. Effects of Risk Sensitivity

Next, we studied the effects of risk sensitivity in the context of safe robot navigation among humans. Fig. 3 compares RSSAC with $\sigma = 0$ (the same data as in Fig. 2) and $\sigma = 1.0$. Our risk-sensitive optimization significantly reduced the standard deviation of the minimum robot-human distance by 11%, 12%, and 24% for the ETH, HOTEL, and UNIV scenes, respectively. The risk sensitivity trades off the stochastic collision cost and the deterministic tracking cost, which appears in the ETH and HOTEL results where the standard deviation of the normalized goal distance slightly increased by 3.9% and 2.3%. On the other hand, the UNIV scene was so densely populated with humans that the risk-sensitive robot was unable to approach the goal. Although not necessarily captured by the quantitative metrics, overall the risk-sensitive robot also exhibited another intriguing behavioral difference as depicted in Fig. 4; the risk-sensitive robot tends to yield to potentially conflicting humans.

To gain a better understanding of the effects of risk-sensitivity, we designed a simplified intersection scenario as illustrated in Fig. 5. The human in this toy example followed a linear Gaussian model with a constant mean velocity and was in a potential collision course with the robot. We ran 100 simulations for each σ in $\{0.0, 0.5, 1.0\}$. The goal position was not randomized in this study. We also ran two additional sets of 100 simulations wherein we kept $\sigma = 0.0$ but varied the collision cost peak parameter α and the bandwidth λ , respectively. This was to elucidate the difference between risk sensitivity tuning and cost function design. The results are summarized in Fig. 6 and confirm that the more risk-sensitive the robot becomes, the more likely it is to yield to the human. This consistent trend was not present when changing α or λ only. On the other hand, there was a positive correlation between the minimum robot-human distance and all the three parameters. This observation suggests that the risk sensitivity parameter σ affects the global behavior of the robot that determines the “type” of interaction, whereas cost function tuning affects the local behavior only (i.e. minimum robot-human distance). Thus, the risk sensitivity parameter can be considered as an additional degree of freedom in choosing desirable robot behavior.

C. Real-World Experiment

Our simulation study was accompanied by a real-world experiment with a holonomic robot [40] and 5 human subjects in an indoor environment (see Fig. 1). Those subjects were assigned a specific start and goal positions, and were instructed to walk to each individual goal at normal speeds. Although the start-goal pairs remained the same, we changed their assignment to the subjects after each run so the type of interaction remained diverse. The robot started at a known position, but its exact goal was randomized and not known to the subjects. The positions of all the subjects as well as the robot were measured by a motion capture system. We used the same parameters as in the simulation study except u_{max} . A major difference between the simulation and the real-world experiment is that the robot is “visible” to humans in the experiment, which requires the robot to take into account resulting human-robot interaction in addition to human-human interaction. This was achieved by conditioning the prediction of Trajectron++ by the nominal control candidates for RSSAC, similar to [22]. As explained in Section IV-E, this robot-future-conditional prediction lets the robot reason about the effect of different nominal control candidates on human behavior, prior to RSSAC perturbation on the best one. We compared the performance of the unconditional prediction (as in the simulation study) to the conditional prediction for both risk-neutral and risk-sensitive cases.

The results of the experiment are presented in Fig. 7. For each setting we performed 5 runs. The robot with unconditional predictions was either too conservative ($\sigma = 1.0$) or unsafe ($\sigma = 0.0$). Using conditional predictions improved overall performance; we did not observe a single collision with conditional predictions. This supports our hypothesis that the robot-future-conditional prediction facilitates appropriate human-robot interaction. Of the four cases tested, using conditional predictions with $\sigma = 1.0$ achieved the best average performance, as well as the smallest variance in minimum robot-human distance.

VI. CONCLUSIONS

This paper presents a novel online framework for safe crowd-robot interaction with dynamic collision avoidance. Our theoretical contribution is a derivation of the mode insertion gradient for the entropic risk measure. This theory

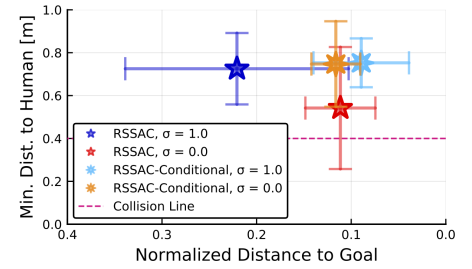


Fig. 7: Quantitative results of the real-world experiment with 5 human subjects. Using robot-future-conditional predictions with $\sigma = 1.0$ achieves the best average performance. Error bars show the standard deviation of 5 runs with a randomized robot goal and human start-goal assignment.

leads to an efficient and high-performance implementation of RSSAC, the proposed model predictive control algorithm for safety-critical systems. As a major practical contribution, we show that the probabilistic, robot-future-conditional predictions of Trajectron++ combined with the risk-sensitive optimization of RSSAC lead to safe and efficient robot navigation among many mutually-interacting humans. Furthermore, the risk sensitivity parameter is found to play a crucial role in determining the type of interaction behavior, such as yielding. In future work, we plan to apply the RSSAC-Trajectron++ framework to vehicle dynamics and scenes with heterogeneous agents. We are also interested in automatically adapting the risk sensitivity parameter so that any specifically-desired robot behavior emerges.

REFERENCES

- [1] H. Nishimura and M. Schwager, "SACBP: Belief Space Planning for Continuous-Time Dynamical Systems via Stochastic Sequential Action Control," in *The 13th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.
- [2] P. Whittle, "Risk sensitivity, a strangely pervasive concept," *Macroeconomic Dynamics*, vol. 6, no. 1, pp. 5–18, 2002.
- [3] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Robotics Research*. Springer, 2020, pp. 75–84.
- [4] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 300–306.
- [5] V. Gómez, S. Thijssen, A. Symington, S. Hailes, and H. J. Kappen, "Real-time stochastic optimal control for multi-agent quadrotor systems," in *Twenty-Sixth International Conference on Automated Planning and Scheduling (ICAPS)*, 2016.
- [6] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.
- [7] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conf. on Computer Vision*, Aug. 2020.
- [8] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.
- [9] Y. Wardi and M. Egerstedt, "Algorithm for optimal mode scheduling in switched systems," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 4546–4551.
- [10] A. R. Ansari and T. D. Murphey, "Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1196–1214, 2016.
- [11] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 477–483.
- [12] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.
- [13] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, 2017.
- [14] M. Wang, Z. Wang, S. Paudel, and M. Schwager, "Safe distributed lane change maneuvers for multiple autonomous vehicles using buffered input cells," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.
- [15] T. Mylvaganam, M. Sassano, and A. Astolfi, "A differential game approach to multi-agent collision avoidance," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4229–4235, 2017.
- [16] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, June 2017.
- [17] W. Luo and A. Kapoor, "Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates," *arXiv preprint arXiv:1912.09957*, 2019.
- [18] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on automatic control*, vol. 43, no. 4, pp. 509–521, 1998.
- [19] M. Chen, J. C. Shih, and C. J. Tomlin, "Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 1695–1700.
- [20] S. Temizer, M. Kochenderfer, L. Kaelbling, T. Lozano-Pérez, and J. Kuchar, "Collision avoidance for unmanned aircraft using markov decision processes," in *AIAA guidance, navigation, and control conference*, 2010, p. 8040.
- [21] P. Cai, Y. Luo, D. Hsu, and W. S. Lee, "Hyp-despot: A hybrid parallel algorithm for online planning under uncertainty," in *Robotics: Science and Systems (RSS)*, 2018.
- [22] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.
- [23] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Best response model predictive control for agile interactions between autonomous ground vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2403–2410.
- [24] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.
- [25] S. Samuelson and I. Yang, "Safety-aware optimal control of stochastic systems using conditional value-at-risk," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 6285–6290.
- [26] J. R. Medina, D. Lee, and S. Hirche, "Risk-sensitive optimal feedback control for haptic assistance," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 1025–1031.
- [27] P. Whittle, "Risk-sensitive linear/quadratic/gaussian control," *Advances in Applied Probability*, vol. 13, no. 4, pp. 764–777, 1981.
- [28] K. Glover and J. C. Doyle, "Relations between h_{∞} and risk sensitive controllers," in *Analysis and Optimization of Systems*. Springer, 1988, pp. 1–10.
- [29] A. Pokle, R. Martín-Martín, P. Goebel, V. Chow, H. M. Ewald, J. Yang, Z. Wang, A. Sadeghian, D. Sadigh, S. Savarese, et al., "Deep local trajectory replanning and control for robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5815–5822.
- [30] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.
- [31] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.
- [32] E. Tzorakoleftherakis and T. D. Murphey, "Iterative sequential action control for stable, model-based control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3170–3183, 2018.
- [33] H. Nishimura and M. Schwager, "SACBP: Belief Space Planning for Continuous-Time Dynamical Systems via Stochastic Sequential Action Control," in *The International journal of robotics research*, submitted. [Online]. Available: <https://arxiv.org/abs/2002.11775>
- [34] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Conf. on Neural Information Processing Systems*, 2015.
- [35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Conf. on Neural Information Processing Systems*, 2014.
- [36] B. Ivanovic and M. Pavone, "The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2375–2384.
- [37] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.
- [38] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 261–268.
- [39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [40] Z. Wang, G. Yang, X. Su, and M. Schwager, "Ouijabs: Omnidirectional robots for cooperative object transport with rotation control using no communication," in *Distributed Autonomous Robotic Systems*. Springer, 2018, pp. 117–131.