Risk-Sensitive Model Predictive Control for Interaction-Aware Planning–A Sequential Convexification Algorithm

Renzi Wang, Mathijs Schuurmans, and Panagiotis Patrinos, *Member, IEEE*

***Abstract*—This letter considers risk-sensitive model predictive control for stochastic systems with a decision-dependent distribution. This class of systems is commonly found in human-robot interaction scenarios. We derive computationally tractable convex upper bounds to both the objective function, and to frequently used penalty terms for collision avoidance, allowing us to efficiently solve the generally nonconvex optimal control problem as a sequence of convex problems. Simulations of a robot navigating a corridor demonstrate the effectiveness and the computational advantage of the proposed approach.**

***Index Terms*—Predictive control for nonlinear systems, optimization algorithms, stochastic optimal control.**

I. Introduction

STOCHASTIC dynamical systems are widely used in control tasks, particularly in applications such as autonomous driving and robot navigation, where predicting non-deterministic human behavior is crucial [1]. While stochastic influences were traditionally modeled as exogenous disturbances, mixture models have gained popularity for this purpose [2], [3]. In contrast to the exogenous stochastic switching models such as Markov jump models [4], this class of models introduces a dependency of the probability on the system's state, significantly boosting its ability to model interaction between the controlled system and environmental uncertainty.

Conventional stochastic model predictive control (MPC) methods optimize the expected value of the cost function (known as *risk-neutral* cost). This typically leads to tractable formulations, as the linearity of the expectation preserves convexity of the cost function for exogenous uncertainty. However, this is no longer the case for decision-dependent distributions, which generally lead to non-convex problems, even when the underlying cost function is convex. Furthermore, studies in human behavior modeling [5], [6] show that humans are risk-sensitive. This insight suggests that designing risk-sensitive controller would enable more natural human-robot interaction. Based on those two observations, we aim to develop a risk-sensitive optimal control algorithm using the exponential utility function. Besides being beneficial from a modeling point of view, we observe that the exponential utility function introduces additional structure that facilitates the development of a tailored solution method. The method we propose is specifically designed to address the nonconvexity arising from the state-dependent distributions.

Contributions: We summarize the contributions of this letter as follows: (i) We propose a risk-sensitive MPC formulation for a mixture-of-experts (MoE) model [7] that expands beyond traditional Gaussian-only uncertainty models while explicitly accounting for the state-dependency—a crucial property for modeling interactive behaviors. This formulation can be interpreted as a smoothed version of a two-player game, which may be either cooperative or adversarial, based on the value of a hyperparameter; (ii) We develop an algorithm based on the Majorization-Minimization (MM) principle [8] that explicitly exploits the structure of the proposed risk-sensitive formulation. The algorithm converts the nonconvex optimization problem into a sequence of convex problems. This includes a novel reformulation for the nonconvex collision penalty, which is a critical component in navigation systems. (iii) Our numerical experiments validate the effectiveness of both our algorithm and the formulation, demonstrate how the risk-sensitivity parameter impacts interaction behavior, and show significant computational advantages over off-the-shelf solvers for nonlinear programming.

A. Related Work

1) Controlling a Stochastic Mixture Model: Human behavior modeling in interactive settings has increasingly employed mixture models to capture its multi-modal nature. When formulating optimal control problems, scenario trees are commonly employed to roll out the dynamics. Several

Received 17 March 2025; revised 29 May 2025; accepted 22 June 2025. Date of publication 7 July 2025; date of current version 29 July 2025. This work was supported in part by the Research Foundation Flanders (FWO) Projects under Grant G081222N, Grant G033822N, and Grant G0A0920N; in part by the Research Council KU Leuven C1 Project under Grant C14/24/103; in part by the European Union's Horizon 2020 Research and Innovation Programme through the Marie Skłodowska-Curie Grant under Agreement 953348. The work of Mathijs Schuurmans was supported by an FWO postdoctoral fellowship under Project 1248626N. Recommended by Senior Editor L. Zhang. *(Corresponding author: Renzi Wang.)*

The authors are with the Department of Electrical Engineering, KU Leuven, 3001 Leuven, Belgium (e-mail: renzi.wang@kuleuven.be; mathijs.schuurmans@kuleuven.be; panos.patrinos@kuleuven.be).

Digital Object Identifier 10.1109/LCSYS.2025.3586667

2475-1456 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 23,2025 at 13:52:05 UTC from IEEE Xplore. Restrictions apply.

works [9], [10], [11] focus on minimizing the expected value of the cost function, with [9], [10] employing general nonlinear programming solvers and [11] utilizing sampling-based methods. Reference [12] considers both expectation and Conditional Value-at-Risk in their objective functions, and solves the resulting problems via sequential quadratic programming.

2) Risk-Sensitive Control Using the Exponential Utility Function: The risk sensitive control problem with the exponential utility function has a long history [13], [14], [15], [16]. In recent years, this type of method has also been applied in reinforcement learning [17], [18], [19]. When applied to interactive settings, [20], [21] consider the joint dynamics to be linear with an additive Gaussian noise and solve the problem based on Ricatti recursion. Reference [22] employs a probabilistic machine learning model [3] that can be conditioned on future trajectories, and solves the optimal control problem using an approach based on sensitivity analysis. However, their sensitivity analysis does not account for how perturbed future trajectories affect the distribution, and consequently the cost. In contrast, our work explicitly considers the decision-dependent distribution in the mixture model.

3) Majorization-Minimization-Based Solver for Control Problems: The expectation-maximization (EM) [23] algorithm has already been used to solve optimal control problems [18], [24], [25], where Jensen's inequality is used to derive the majorizer. Our work extends this beyond Jensen's inequality. Unlike [26], which applies the MM principle only as a substep to find the optimal disturbance sequence in linear systems, we apply MM to the entire optimal control problem for mixture models.

This letter is organized as follows: We first present the risk-sensitive formulation and its interpretation in Section II. We then derive the solution method in Section III. Finally, we propose an additional formulation for collision penalty and validate the proposed method through a numerical experiment in Section IV.

Notation: Let $\mathbb{R}^d_+$ denote the d-dimensional positive orthant. Let $\mathbf{1}_d = [1 \dots 1] \in \mathbb{R}^d$. We denote the cone of $m \times m$ positive definite matrices by $\mathbb{S}^m_{++}$, and the probability simplex $\Delta^d = \{p \in \mathbb{R}^d_+ \mid \mathbf{1}_d^\top p = 1\}$. Define $\mathrm{lse} : \mathbb{R}^d \to \mathbb{R}$ as $\mathrm{lse}(x) = \ln(\sum_{j=1}^d \exp(x_j))$, and the softmax function $\sigma : \mathbb{R}^d \to \Delta_d$ with components $\sigma_i(x) = \exp(x_i)/\sum_{j=1}^d \exp(x_j)$. For discrete distribution $\Pi \in \Delta^n$ and $X \in \mathbb{R}^n$ representing realizations of random variable x, we denote $\mathbb{E}_\Pi[X] = \sum_{i=1}^n \Pi_i X_i$ and $\mathbb{V}\mathrm{ar}_\Pi[X] = \sum_{i=1}^n \Pi_i (X_i - \mathbb{E}_\Pi[X])^2$. Let $\mathbb{N}_{[a,b]} = \mathbb{N} \cap [a, b]$. We denote by $\operatorname{vecmin} L$ ($\operatorname{vecmax} L$), the smallest (largest) elements of a vector L.

II. Problem Formulation

We consider a class of stochastic systems in the form of an MoE model [7]:

$$\xi_t \sim \sigma(\Theta x_t), \tag{1a}$$

$$x_{t+1} = A_{\xi_t} x_t + B_{\xi_t} u_t, \tag{1b}$$

where for all $t \geq 0$, $x_t \in \mathbb{X} \subset \mathbb{R}^{n_x}$ denotes the continuous system state, and $u_t \in \mathbb{U} \subset \mathbb{R}^{n_u}$ is the control input. The sets $\mathbb{X}$ and $\mathbb{U}$ are nonempty closed convex sets. The state x_t is governed by d linear experts that selected according to the value of the random variable $\xi_t \in \Xi = \{1, \dots, d\}$. We refer to ξ_t as the *mode* at time t. At every time step t, ξ_t is sampled randomly according to the gate distribution (1a), whose argument is allowed to depend linearly on the state x_t. In this letter, we assume the parameters $\Theta \in \mathbb{R}^{d \times n_x}$, $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_x \times n_u}$, $i \in \Xi$ to be known. In practice, these can be estimated from data using [27].

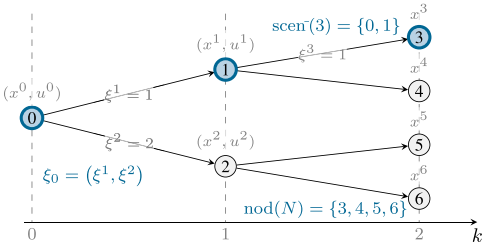


Fig. 1. A fully branched scenario tree with $d = 2$ and $N = 2$.

Remark 1: Note that the argument of σ in (1a) may as well be an affine function of x_t, allowing to model exogenous switching of ξ_t, for instance. However, we will not account for this case explicitly as it can simply be modeled by augmenting the state x_t with a constant 1.

Let N be the prediction horizon. Since Ξ is a finite set, the realization of the stochastic process $\{x_t, u_t\}_{t=1}^N$ satisfying (1) can be represented on a scenario tree [28], as illustrated in Fig. 1. We briefly introduce the scenario tree notation needed for this letter. For detailed notation, we refer to [9].

A scenario tree captures all possible outcomes of a stochastic process over N time steps. We denote the nodes at step $k \in \mathbb{N}_{[0,N]}$ by $\mathrm{nod}(k)$ and $\mathrm{nod}(a, b) = \bigcup_{k=a}^b \mathrm{nod}(k)$ for $a < b \in \mathbb{N}_{[0,N]}$. Let $\boldsymbol{u} := \{u^\iota\}_{\iota \in \mathrm{nod}(0,N-1)}$, and states $\boldsymbol{x} := \{x^\iota\}_{\iota \in \mathrm{nod}(0,N)}$. The dynamics (1) can be represented as

$$x^{\iota+} = A_{\xi^{\iota+}} x^\iota + B_{\xi^{\iota+}} u^\iota,$$

where $\iota+$ denotes the child node of ι corresponding to the realization $\xi^{\iota+}$. Each leaf node $s \in \mathrm{nod}(N)$ corresponds to a unique scenario—a path from root to leaf representing one realization of the stochastic process. For $s \in \mathrm{nod}(N)$, we denote by $\mathrm{scen}^-(s)$ all nodes on this path excluding s itself. Let the convex functions $\ell(x, u) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and $\ell_N(x) : \mathbb{R}^{n_x} \to \mathbb{R}$ denote the stage costs and terminal cost respectively. Then, the cost associated with a scenario s is

$$L_s(\boldsymbol{x}, \boldsymbol{u}) = \ell_N(x^s) + \sum_{\iota \in \mathrm{scen}^-(s)} \ell(x^\iota, u^\iota). \tag{2}$$

A conventional stochastic MPC formulation considers minimizing the expected cost in the optimal control problem (OCP), i.e., for a given initial state $x_t \in \mathbb{X}$,

$$\underset{(\boldsymbol{x},\boldsymbol{u}) \in \mathbb{C}(x_t)}{\text{minimize}} \quad \mathcal{L}^{\mathrm{n}}(\boldsymbol{x}, \boldsymbol{u}) := \sum_{s \in \mathrm{nod}(N)} p(s; \boldsymbol{x}) L_s(\boldsymbol{x}, \boldsymbol{u}), \tag{3}$$

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 23,2025 at 13:52:05 UTC from IEEE Xplore. Restrictions apply.

where the set $\mathbb{C}(x_t)$ is defined as

$$\mathbb{C}(x_t) := \left\{ (\boldsymbol{x}, \boldsymbol{u}) \;\middle|\; \begin{array}{l} x^0 = x_t, \\ x^{\iota+} = A_{\xi^{\iota+}} x^\iota + B_{\xi^{\iota+}} u^\iota, \\ u^\iota \in \mathbb{U},\ \forall \iota \in \mathrm{nod}(0, N-1), \\ x^\iota \in \mathbb{X},\ \forall \iota \in \mathrm{nod}(0, N), \end{array} \right\}.$$

The probability for a given scenario is defined as

$$p(s\,;\boldsymbol{x}) = \prod_{\iota \in \mathrm{scen}^-(s)} p(\xi^{\iota+}\,;\boldsymbol{x}) = \prod_{\iota \in \mathrm{scen}^-(s)} \sigma_{\xi^{\iota+}}\big(\Theta x^\iota\big). \tag{4}$$

We denote by $P(\boldsymbol{x}) \in \Delta^{d^N}$ the vector whose elements represent the scenario probability $p(s\,;\boldsymbol{x})$. We refer to (3) as the *risk-neutral formulation*. Due to the state-dependent distribution $P(\boldsymbol{x})$, problem (3) is nonconvex, even when the loss function L_s is convex, making the problem hard to solve.

Risk-sensitive formulations use an exponential utility function [14], [15] in the OCP, which is defined as $\rho_\gamma(L, P) := \frac{1}{\gamma} \ln \mathbb{E}_P[\exp(\gamma L)]$, where $\gamma \neq 0$ determines the degree of risk-sensitivity. For $\gamma > 0$, it is also known as the entropic risk measure [29, Example 6.20]. It will be convenient to let $\gamma > 0$ and define the OCPs

$$\underset{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{C}(x_t)}{\text{minimize}}\ \mathcal{L}^{\mathrm{o}}_\gamma(\boldsymbol{x},\boldsymbol{u}) := -\tfrac{1}{\gamma} \ln \mathbb{E}_{P(\boldsymbol{x})}\big[\exp\big(-\gamma L(\boldsymbol{x},\boldsymbol{u})\big)\big], \tag{5}$$

$$\underset{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{C}(x_t)}{\text{minimize}}\ \mathcal{L}^{\mathrm{p}}_\gamma(\boldsymbol{x},\boldsymbol{u}) := \tfrac{1}{\gamma} \ln \mathbb{E}_{P(\boldsymbol{x})}\big[\exp\big(\gamma L(\boldsymbol{x},\boldsymbol{u})\big)\big]. \tag{6}$$

We refer to (5) as the *optimistic formulation* and to (6) as the *pessimistic formulation*.

A. Interpretation of the Risk-Sensitive Formulation

1) Regularized Risk-Neutral Problem: As presented in [30, Remark 2], the risk-sensitive formulation can be viewed as a regularized risk-neutral problem, in the form:

$$\mathcal{L}^{\mathrm{o}}_\gamma = \mathbb{E}_{P(\boldsymbol{x})}(L) - \tfrac{\gamma}{2}\mathbb{V}\mathrm{ar}_{P(\boldsymbol{x})}(L) + o(\gamma),$$
$$\mathcal{L}^{\mathrm{p}}_\gamma = \mathbb{E}_{P(\boldsymbol{x})}(L) + \tfrac{\gamma}{2}\mathbb{V}\mathrm{ar}_{P(\boldsymbol{x})}(L) + o(\gamma),$$

when γ is around 0. The approximation is obtained by applying Taylor expansion. For the optimistic case, higher cost variance across the scenario distribution $P(\boldsymbol{x})$ is preferred. In contrast, the pessimistic case penalizes the cost variance across scenarios.

2) Game-Theoretical Interpretation: Risk-sensitive formulations (5) and (6) can be viewed as smoothed versions of a collaborative and a competitive game, respectively, as we show in the following result.

Lemma 1: For any $x_t \in \mathbb{X}$, and $(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{C}(x_t)$, we have

$$\underline{L}(\boldsymbol{x},\boldsymbol{u}) \leq \mathcal{L}^{\mathrm{o}}_\gamma(\boldsymbol{x},\boldsymbol{u}) \leq \mathbb{E}_{P(\boldsymbol{x})}[L(\boldsymbol{x},\boldsymbol{u})] \leq \mathcal{L}^{\mathrm{p}}_\gamma(\boldsymbol{x},\boldsymbol{u}) \leq \bar{L}(\boldsymbol{x},\boldsymbol{u}),$$

where $\underline{L} := \mathrm{vecmin}\, L$ and $\bar{L} := \mathrm{vecmax}\, L$. Furthermore,

(i) $\lim_{\gamma\to 0} \mathcal{L}^{\mathrm{o}}_\gamma(\boldsymbol{x},\boldsymbol{u}) = \mathbb{E}_{P(\boldsymbol{x})}[L(\boldsymbol{x},\boldsymbol{u})]$;
(ii) $\lim_{\gamma\to\infty} \mathcal{L}^{\mathrm{o}}_\gamma(\boldsymbol{x},\boldsymbol{u}) = \mathrm{vecmin}\, L(\boldsymbol{x},\boldsymbol{u})$;
(iii) $\lim_{\gamma\to 0} \mathcal{L}^{\mathrm{p}}_\gamma(\boldsymbol{x},\boldsymbol{u}) = \mathbb{E}_{P(\boldsymbol{x})}[L(\boldsymbol{x},\boldsymbol{u})]$;
(iv) $\lim_{\gamma\to\infty} \mathcal{L}^{\mathrm{p}}_\gamma(\boldsymbol{x},\boldsymbol{u}) = \mathrm{vecmax}\, L(\boldsymbol{x},\boldsymbol{u})$.

This result follows from elementary arguments. An explicit proof is given in [31].

From Lemma 1, it follows that as γ increases, problems (5) and (6) more closely approximate the two-player games

$$\underset{\boldsymbol{x},\boldsymbol{u}\in\mathbb{C}(x_t)}{\text{minimize}}\ \max_{s\in\mathrm{nod}(N)} L_s(\boldsymbol{x},\boldsymbol{u}),\quad \underset{\boldsymbol{x},\boldsymbol{u}\in\mathbb{C}(x_t)}{\text{minimize}}\ \max_{s\in\mathrm{nod}(N)} L_s(\boldsymbol{x},\boldsymbol{u}),$$

respectively, where in the former case, the controller takes the optimistic assumption that the uncertain environment will act cooperatively, whereas in the latter, it assumes an adversarial environment.

Remark 2: Note that Lemma 1 only provided point-wise convergence w.r.t. γ. Under some mild regularity assumptions, epigraphical versions of these limits can also be shown, which provide a more complete justification for this interpretation [32, Ch. 7].

III. Sequential Convexification

The problems (5) and (6) are nonconvex problems. To solve them efficiently, we propose a sequential convexification scheme based on the MM principle [8], where we iteratively construct a convex surrogate function $\mathcal{Q}^m$ at the current iterate $(\boldsymbol{x}^m, \boldsymbol{u}^m) \in \mathbb{C}(x_t)$, satisfying

$$\mathcal{Q}^m(\boldsymbol{x},\boldsymbol{u}) \geq \mathcal{L}_\gamma(\boldsymbol{x},\boldsymbol{u}),\ \forall (\boldsymbol{x},\boldsymbol{u}) \in \mathbb{C}(x_t), \tag{7a}$$
$$\mathcal{Q}^m(\boldsymbol{x}^m,\boldsymbol{u}^m) = \mathcal{L}_\gamma(\boldsymbol{x}^m,\boldsymbol{u}^m). \tag{7b}$$

which is minimized to yield the next iterate $(\boldsymbol{x}^{m+1}, \boldsymbol{u}^{m+1})$.

We show in the following proposition that a surrogate function satisfying (7a) for problem (5) can be obtained by applying Jensen's inequality.

Proposition 1: Let $\Pi \in \Delta^{d^N}$ be an arbitrary distribution, and $\mathrm{KL}\big(\Pi \,\|\, P\big) = \sum_i \Pi_i(\ln \Pi_i - \ln P_i)$ denote the Kullback-Leibler (KL) divergence, we define

$$\mathcal{Q}^{\mathrm{o}}(\boldsymbol{x},\boldsymbol{u}\,;\Pi) := \tfrac{1}{\gamma}\mathrm{KL}\big(\Pi \,\|\, P(\boldsymbol{x})\big) + \Pi^\top L(\boldsymbol{x},\boldsymbol{u}). \tag{8}$$

For all $\boldsymbol{x}, \boldsymbol{u}$, we have $\mathcal{L}^{\mathrm{o}}_\gamma(\boldsymbol{x},\boldsymbol{u}) \leq \mathcal{Q}^{\mathrm{o}}(\boldsymbol{x},\boldsymbol{u}\,;\Pi)$.

Proof: Recalling the definition (5) and noting that $x \mapsto -\ln(x)$ is convex, we expand

$$\begin{aligned} \mathcal{L}^{\mathrm{o}}_\gamma(\boldsymbol{x},\boldsymbol{u}) &= -\tfrac{1}{\gamma} \ln\Big[\sum_{s\in\mathrm{nod}(N)} \tfrac{\Pi_s}{\Pi_s} p(s\,;\boldsymbol{x}) \exp(-\gamma L_s(\boldsymbol{x},\boldsymbol{u}))\Big] \\ &\leq -\tfrac{1}{\gamma} \sum_{s\in\mathrm{nod}(N)} \Pi_s \ln\big[\tfrac{p(s\,;\boldsymbol{x})}{\Pi_s} \exp(-\gamma L_s(\boldsymbol{x},\boldsymbol{u}))\big] \\ &= -\tfrac{1}{\gamma} \sum_{s\in\mathrm{nod}(N)} \Pi_s \ln\big[\tfrac{p(s\,;\boldsymbol{x})}{\Pi_s}\big] + \sum_{s\in\mathrm{nod}(N)} \Pi_s L_s(\boldsymbol{x},\boldsymbol{u}), \end{aligned}$$

where the inequality follows the Jensen's inequality. The claim follows by definition of the KL divergence. ∎

Finding Π satisfying (7b) is less straightforward. Since $\mathcal{Q}^{\mathrm{o}}(\cdot,\cdot\,;\Pi)$ upper bounds the cost for any Π, a reasonable estimate is to find the smallest such upper bound:

$$\Pi^\star = \underset{\Pi\in\Delta^{d^N}}{\mathrm{argmin}}\, \mathcal{Q}(\boldsymbol{x}^m,\boldsymbol{u}^m\,;\Pi). \tag{9}$$

Particularly, this convex problem has a closed-form solution.

Lemma 2: The solution of problem (9) is

$$\Pi^\star_s = \frac{\exp\big(\ln p(s\,;\boldsymbol{x}^m) - \gamma L_s(\boldsymbol{x}^m,\boldsymbol{u}^m)\big)}{\sum_{s'\in\mathrm{nod}(N)} \exp\big(\ln p(s'\,;\boldsymbol{x}^m) - \gamma L_{s'}(\boldsymbol{x}^m,\boldsymbol{u}^m)\big)}. \tag{10}$$

Proof: Since $(\boldsymbol{x}^m, \boldsymbol{u}^m)$ are constants, we will use P and L to denote $P(\boldsymbol{x}^m)$, $L(\boldsymbol{x}^m, \boldsymbol{u}^m)$, respectively. Problem (9) is equivalent to

$$\begin{aligned} \underset{\Pi}{\mathrm{argmin}}\quad & \tfrac{1}{\gamma}\mathrm{KL}\big(\Pi \,\|\, P\big) + \Pi^\top L, \\ \text{subject to}\quad & \Pi^\top \mathbf{1}_{d^N} = 1, \quad \Pi_s \geq 0 \quad \forall s \in \mathrm{nod}(N). \end{aligned} \tag{11}$$

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 23,2025 at 13:52:05 UTC from IEEE Xplore. Restrictions apply.

As the KL divergence is convex w.r.t. both arguments, the problem (11) is convex. Define the Lagrangian

$$\mathscr{L}(\Pi, \mu, \boldsymbol{\lambda}) := \tfrac{1}{\gamma}\mathrm{KL}\big(\Pi \,\|\, P\big) + \Pi^\top L - \boldsymbol{\lambda}^\top \Pi - \mu\big(1 - \Pi^\top \mathbf{1}_{d^N}\big).$$

The KKT conditions of problem (11) yield:

$$\nabla_{\Pi_s}\mathscr{L}(\Pi, \mu, \boldsymbol{\lambda}) = \tfrac{1}{\gamma}(\ln \Pi_s + 1 - \ln P_s) + L_s - \lambda_s + \mu = 0,$$
$$\lambda_s \Pi_s = 0,\ \lambda_s \geq 0,\ \Pi_s \geq 0,\ \Pi^\top \mathbf{1}_{d^N} = 1,\ \forall s \in \mathrm{nod}(N).$$

The first equation is equivalent to

$$\Pi_s = \exp\big(-1 + \ln P_s + \gamma(\lambda_s - L_s - \mu)\big). \tag{13}$$

Thus, $\Pi_s > 0$ for all $s \in \mathrm{nod}(N)$ and $\lambda_s = 0$. Because

$$\sum_{s\in\mathrm{nod}(N)} \Pi_s = \exp(-\gamma\mu) \sum_{s\in\mathrm{nod}(N)} \exp\big(-1 + \ln P_s - \gamma L_s\big) = 1,$$

it follows that $\mu = \frac{1}{\gamma}\ln(\sum_{s\in\mathrm{nod}(N)} \exp(-1 + \ln P_s - \gamma L_s))$. Plugging the expression into (13), we obtain

$$\Pi_s = \frac{\exp\big(-1+\ln P_s - \gamma L_s\big)}{\sum_{s'\in\mathrm{nod}(N)} \exp\big(-1+\ln P_{s'} - \gamma L_{s'}\big)},$$

which is equivalent to (10). ∎

Plugging (10) into the definition of $\mathcal{Q}^{\mathrm{o}}$ (8), it is straightforward to verify that for this value of Π, $\mathcal{Q}^{\mathrm{o}}$ satisfies (7b). To summarize, instead of solving (5) directly, we solve

$$(\boldsymbol{x}^{m+1}, \boldsymbol{u}^{m+1}) \leftarrow \operatorname*{argmin}_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{C}(x_t)} \mathcal{Q}^{\mathrm{o}}(\boldsymbol{x}, \boldsymbol{u}\,; \Pi^m) \tag{14}$$

at each iteration m, where the parameter Π^m is computed via (10) using the current iterate $(\boldsymbol{x}^m, \boldsymbol{u}^m)$.

Proposition 1 relies on the convexity of function $x \mapsto -\ln(x)$ to derive the surrogate function, which is not applicable for the pessimistic formulation. The following proposition provides an alternative surrogate function satisfying (7) for problem (6).

Proposition 2: Let $\tilde{\boldsymbol{x}} = \{\tilde{x}^\iota\}_{\iota\in\mathrm{nod}(0,N)}$, with $\tilde{x}^\iota \in \mathbb{X}$ chosen arbitrarily for all $\iota \in \mathrm{nod}(0, N)$, and let

$$\mathcal{Q}^{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{u}\,; \tilde{\boldsymbol{x}}) := \tfrac{1}{\gamma}\mathrm{lse}\big[\hat{P}(\boldsymbol{x}\,; \tilde{\boldsymbol{x}}) + \gamma L(\boldsymbol{x}, \boldsymbol{u})\big],$$

where the vector $\hat{P}(\boldsymbol{x}\,; \tilde{\boldsymbol{x}}) \in \mathbb{R}^{d^N}$ has entries $\hat{P}_s(\boldsymbol{x}\,; \tilde{\boldsymbol{x}}) := \rho(\boldsymbol{\xi}_s, \Theta\boldsymbol{x}_s\,; \Theta\tilde{\boldsymbol{x}}_s)$, for all $s \in \mathrm{nod}(N)$. Here, $\boldsymbol{\xi}_s := \{\xi^\iota\}_{\iota\in\mathrm{scen}^-(s)}$ is the mode realization of scenario s, $\boldsymbol{x}_s := \{x^\iota\}_{\iota\in\mathrm{scen}^-(s)}$ represents the states along the scenario, and $\rho : \Xi^N \times \mathbb{R}^{d\times N} \to \mathbb{R}$ is defined as

$$\rho(i,\ Z\,; \tilde{Z}) := \sum_{t=0}^{N-1} \tilde{Z}_{i_t,t} - \mathrm{lse}(\tilde{Z}_t) - \sigma^\top(\tilde{Z}_t)(Z_t - \tilde{Z}_t). \tag{15}$$

For all $\boldsymbol{x}, \boldsymbol{u}$, we have $\mathcal{L}^{\mathrm{p}}_\gamma(\boldsymbol{x}, \boldsymbol{u}) \leq \mathcal{Q}^{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{u}\,; \tilde{\boldsymbol{x}})$. If, furthermore, $\tilde{\boldsymbol{x}} = \boldsymbol{x}$, this holds with equality.

Proof: Function $\mathcal{L}^{\mathrm{p}}_\gamma$ defined in (6) can be written as

$$\begin{aligned}\mathcal{L}^{\mathrm{p}}_\gamma(\boldsymbol{x}, \boldsymbol{u}) &= \tfrac{1}{\gamma}\ln\Big[\sum_{s\in\mathrm{nod}(N)} \exp\big(\ln p(s\,; \boldsymbol{x}) + \gamma L_s(\boldsymbol{x}, \boldsymbol{u})\big)\Big],\\ &= \tfrac{1}{\gamma}\mathrm{lse}\big(\ln P(\boldsymbol{x}) + \gamma L(\boldsymbol{x}, \boldsymbol{u})\big).\end{aligned}$$

By definition (4), we rewrite the log-probability

$$\ln p(s\,; \boldsymbol{x}) = \sum_{\iota\in\mathrm{scen}^-(s)} \ln \sigma_{\xi^\iota}\big(\Theta x^\iota\big) = \sum_{\iota\in\mathrm{scen}^-(s)} \Theta_{\xi^\iota} x^\iota - \mathrm{lse}\big(\Theta x^\iota\big), \tag{16}$$

where the second equality follows by definition of the softmax function σ. Since lse is convex and differentiable, it satisfies $\mathrm{lse}(x) \geq \mathrm{lse}(y) + \sigma^\top(y)(x-y)$. Applying this inequality to (16), it follows that $\ln p(s\,; \boldsymbol{x}) \leq \rho(\boldsymbol{\xi}_s, \Theta\boldsymbol{x}_s\,; \Theta\tilde{\boldsymbol{x}}_s)$. Consequently, $\ln P(\boldsymbol{x}) \leq \hat{P}(\boldsymbol{x}\,; \tilde{\boldsymbol{x}})$ element-wise. Since lse is monontonically increasing, the inequality implies that $\mathcal{L}^{\mathrm{p}}_\gamma(\boldsymbol{x}, \boldsymbol{u}) \leq \frac{1}{\gamma}\mathrm{lse}(\hat{P}(\boldsymbol{x}\,; \tilde{\boldsymbol{x}}) + \gamma L(\boldsymbol{x}, \boldsymbol{u}))$. ∎

To summarize, instead of solving (6) directly, we solve

$$\begin{aligned}(\boldsymbol{x}^{m+1}, \boldsymbol{u}^{m+1}) \leftarrow\ & \operatorname{argmin}_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{C}(x_t)} \mathcal{Q}^{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{u}\,; \boldsymbol{x}^m)\\ :=\ & \tfrac{1}{\gamma}\mathrm{lse}\big[\hat{P}(\boldsymbol{x}\,; \boldsymbol{x}^m) + \gamma L(\boldsymbol{x}, \boldsymbol{u})\big],\end{aligned} \tag{17}$$

where each entry $\hat{P}_s(\boldsymbol{x}\,; \boldsymbol{x}^m) := \rho(\boldsymbol{\xi}_s, \Theta\boldsymbol{x}_s\,; \Theta\boldsymbol{x}^m_s)$ with ρ defined in (15).

IV. Numerical Experiment

We compare the performance of different formulations using a corridor scenario where a robot and a human approach from opposite directions. The robot is represented by discrete-time double integrators (in x and y) with state $x^{\mathrm{bot}}_k = \big[p_{x,k}\ p_{y,k}\ v_{x,k}\ v_{y,k}\big]^\top$, where p_x, p_y, v_x, v_y denote the position and the velocity in the x and y directions, respectively. The sample time is $\Delta t = 0.1$ s. The robot controls its acceleration $u \in \mathbb{R}^2$ in both directions. For simplicity, we assume the human (with state $x^{\mathrm{h}}_k = \big[p^{\mathrm{h}}_{x,k}\ p^{\mathrm{h}}_{y,k}\big]$) maintains a constant x-velocity $v^{\mathrm{h}}_x = -0.8$ m/ s. The y-velocity is controlled by $v^h_{y,k} = -0.3(p^h_{y,k} - y^{\mathrm{ref}}_\xi)$, to track a reference position y^{ref}_ξ, taking values $y^{\mathrm{ref}}_1 = 0$ m, $y^{\mathrm{ref}}_2 = -1$ m, and $y^{\mathrm{ref}}_3 = 1$ m, depending on ξ. The distribution of ξ depends on the relative distance:

$$\xi \sim \sigma\left(\Theta\begin{bmatrix} p_x - p^{\mathrm{h}}_x \\ p_y - p^{\mathrm{h}}_y \\ 1\end{bmatrix}\right), \quad \text{with } \Theta = \begin{bmatrix} -5 & -1 & -1 \\ 0 & 1 & -1 \\ -12.5 & 0 & 0\end{bmatrix}.$$

We define the joint state $x_k := \big[x^{\mathrm{bot}\top}_k\ x^{\mathrm{h}\top}_k\big]^\top \in \mathbb{R}^6$. The robot aims to move forward as fast as possible while avoiding collision either with the wall or the human. To this end, the robot tracks the reference $x^{\mathrm{ref}}_k = \big[p_{x,t} + k v^{\max}_x\ 0\ v^{\max}_x\ 0\big]$ using cost functions:

$$\ell^{\mathrm{track}}(x, u) = \tfrac{1}{2}\left\|x^{\mathrm{bot}} - x^{\mathrm{ref}}\right\|^2_Q + \tfrac{1}{2}\|u\|^2_R,$$
$$\text{and} \quad \ell^{\mathrm{track}}_N(x) = \tfrac{1}{2}\left\|x^{\mathrm{bot}} - x^{\mathrm{ref}}\right\|^2_{Q_{\mathrm{f}}},$$

and $Q := \mathbf{diag}\{50, 50, 2, 2\}$, $R := \mathbf{diag}\{2, 2\}$, and $Q_{\mathrm{f}} := 5Q$. To avoid colliding with the human, we introduce a penalty:

$$c(x) = \alpha \exp\left(-\beta\left\|\begin{bmatrix} p_x - p^{\mathrm{h}}_x \\ p_y - p^{\mathrm{h}}_y\end{bmatrix}\right\|\right) \tag{18}$$

with $\alpha = 500$ and $\beta = 5$. Thus, the stage and the terminal cost are given by $\ell(x_k, u_k) = \ell^{\mathrm{track}}(x_k, u_k) + c(x_k)$, $\ell_N(x_N) = \ell^{\mathrm{track}}_N(x_N) + c(x_N)$. Finally, $\mathbb{X}$ and $\mathbb{U}$ are defined by the box constraints $\big[-1\ -0.6\big]^\top \leq u \leq \big[1\ 0.6\big]^\top$, $p_y \in [-1.5, 1.5]$, $v_x \in [0, 1.5]$, and $v_y \in [-1, 1]$.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 23,2025 at 13:52:05 UTC from IEEE Xplore. Restrictions apply.

A. Upper Bound of Collision Penalty

The stage and terminal costs are nonconvex due to the collision penalty term (18). To apply our proposed formulation, we derive a convex upper bound for this collision penalty.

Lemma 3: Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ defined as $f(x) = f_1(f_2(x))$, where:

- $f_1 : \mathbb{R} \to \mathbb{R}$ is convex, monotonically decreasing,
- $f_2 : \mathbb{R}^n \to \mathbb{R}$ is convex.

For any $\tilde{x} \in \mathbb{R}^n$ and $w \in \partial f_2(\tilde{x})$, where $\partial f_2(\tilde{x})$ is the subdifferential of f_2 at $\tilde{x}$ [33, Sec. 23], the function $\hat{f}(x\,;w) = f_1(f_2(\tilde{x}) + w^\top(x-\tilde{x}))$ is convex and satisfies: (i) $\hat{f}(x\,;w) \geq f(x)$ for all $x \in \mathbb{R}^n$, (ii) $\hat{f}(\tilde{x}\,;w) = f(\tilde{x})$.

Proof: Function $\hat{f}$ is a composition of an affine mapping and a convex function and is therefore convex [32, Ex. 2.20(a)]. Inequality (i) follows directly from the subgradient inequality [32, Proposition 8.12], combined with the monotone decrease property of f_1. The equality (ii) is readily verified by substituting $\tilde{x}$ into the definitions of f and $\hat{f}$. ■

Several common collision penalty functions satisfy the conditions in Lemma 3, including:

- $c(x) = \alpha \exp(-\beta \|x\|_\Sigma^2)$ with $\Sigma \succ 0$, such as in [22], [34], where $f_1(z) := \alpha \exp(-\beta z)$, and $f_2(x) := \|x\|_\Sigma^2$;
- $c(x) = \alpha \exp(-\beta \|x\|)$ where $f_1(z) := \alpha \exp(-\beta z)$, and $f_2(x) := \|x\|$;
- $c(x) = 1/(\alpha + \beta \|x\|)^p$ with $p > 0$, such as in [21], where $f_1(z) := 1/(\alpha + \beta z)^p$, and $f_2(x) := \|x\|$;

where $\alpha, \beta > 0$. In our experiment, we select the form $c(x) = \alpha \exp(-\beta \|x\|)$ since it exhibits small approximation gap with the upper bound in Lemma 3 across the domain. To apply our formulation, we first majorize the loss function $\mathcal{L}_\gamma^{\mathrm{o}}$ or $\mathcal{L}_\gamma^{\mathrm{p}}$ using Proposition 1 or Proposition 2, respectively. Then, we replace the collision penalty with its convex upper bound by applying Lemma 3, using the previous iterate as $\tilde{x}$.

B. Sensitivity of γ

To test the impact of different γ values, we set $\gamma = 10^i$, $i \in \mathbb{N}_{[-3,2]}$. This range spans from small values that approximate the risk-neutral case to larger values that approach joint minimization and minimax problems. To compare with the risk-neutral case, we apply the same termination criterion measuring the optimality error [35, Eq. (6)] for the proposed method, with the tolerance $\epsilon_{\mathrm{tol}} = 0.003$. To reduce the computational complexity, we branch the scenario tree for first $N_b = 5$ steps for prediction horizon $N = 20$. The final selected mode is maintained for the last 15 steps. The robot is initialized at $x_0^{\mathrm{bot}} = \begin{bmatrix} -3 & 0 & 0 & 0 \end{bmatrix}^\top$ while the human position is randomly initialized. Each experiment is repeated 10 times to mitigate randomness effects. The closed-loop performance is evaluated by the accumulated velocity tracking error and the minimal distance between the robot and the human. The result is summarized in Fig. 2.

As shown in Fig. 2, the optimistic formulation yields reduced velocity tracking error but smaller minimal distance than its pessimistic counterparts. Both formulations exhibit similar behavior as they converge towards the risk-neutral formulation ($\gamma \to 0$). For the optimistic formulation, initial increases in γ simultaneously decreases velocity tracking error and minimal distance, as the optimistic formulation accounts

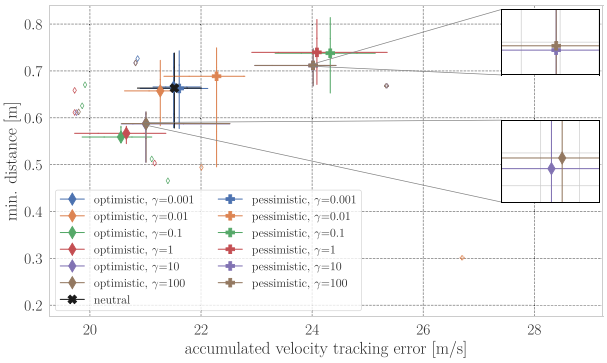


Fig. 2. Performance comparison of different formulations with $\gamma = 10^i$, $i \in \mathbb{N}_{[-3,2]}$. The solid markers represent the median. The lines extend from $Q_1 - \mathrm{IQR}$ to $Q_3 + \mathrm{IQR}$, where IQR is the difference between the first and the third quartile (Q_1 and Q_3). Data points beyond this range are plotted individually as outliers. Same color indicates the same value of γ.

for cooperative behavior that jointly minimizes the loss. Though at large γ values, tracking error deteriorates due to the finite-infinite prediction horizon discrepancies. Conversely, the pessimistic formulation initially exhibits increased tracking error and distance with rising γ, as it accounts for unfavorable behavior that maximizes the loss. At high γ values, it demonstrates improved tracking as a result of the discrepancy between the finite-horizon worst-case approximation and the underlying infinite-horizon problem.

C. Runtime Analysis

As suggested in Section II-A1 and in Section IV-B, the risk-sensitive formulation approximates the risk-neutral formulation with a small γ. We thus compare the runtime performance of the risk-sensitive formulation with $\gamma = 10^{-3}$ and the risk-neutral formulation to demonstrate its computational advantage. We implement the convex problem of the proposed MM algorithm with `CVXPY` [36], and solve it via `MOSEK` [37]. The risk-neutral problem is solved with `IPOPT` [35]. For problems (3), (5), and (6), we let the initial state $x_t = [\,-2.5\ 0\ 1\ 0\ 1\ 0.2\,]^\top$ to ensure constraints are activated within the horizon. We branch the scenario tree for the first $N_b = 5$ steps with varying prediction horizons. Using the same set of 10 random initial guesses for all algorithms, the experiment is conducted on an Intel Core i7-11700@2.50 GHz machine.

To conduct a fair comparison, we apply the same termination criterion measuring the optimality error [35, Eq. (6)] for all algorithms with the tolerance $\epsilon_{\mathrm{tol}} = 0.003$. Since this criterion requires the dual variables, as a heuristic, we use the optimal dual variables from the surrogate problem when evaluating the termination criterion for our approach. The optimal loss value and the mean of the runtime are summarized in Table I. From Table I, we observe that each solver converges to a solution of similar quality measured by the loss value. However, the risk-sensitive formulations demonstrate a significant improvement in runtime performance compared to the risk-neutral formulation.

The computational complexity grows exponentially w.r.t. the branching horizon N_b. In practice, different heuristics can be applied to reduce the runtime, such as using a loose termination criterion, or reducing the branching horizon

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 23,2025 at 13:52:05 UTC from IEEE Xplore. Restrictions apply.

TABLE I
RUNTIME COMPARISON USING TERMINATION CRITERION [35, EQ. (6)] WITH $\epsilon_{\text{tol}} = 0.003$. RISK-NEUTRAL PROBLEM IS SOLVED BY IPOPT. THE SURROGATE PROBLEMS IN MM ALGORITHM ARE SOLVED BY MOSEK

	$N = 15$		$N = 20$	
	$\mathcal{L}_\gamma$	runtime [s]	$\mathcal{L}_\gamma$	runtime [s]
risk-neutral	85.3 ± 1.1	128.2 ± 15.2	187.4 ± 3.8	285.0 ± 69.5
optimistic	83.8 ± 0.0	11.6 ± 0.6	183.6 ± 2.9	33.8 ± 3.8
pessimistic	84.2 ± 0.0	17.1 ± 0.8	189.8 ± 4.3	60.1 ± 4.9

TABLE II
PERFORMANCE COMPARISON OF OPTIMISTIC FORMULATION USING DIFFERENT SCENARIO TREE CONFIGURATIONS

	Runtime per time step [s]		min. dist. (↑) [m]	AVTE† (↓) [m/s]
	mean	max	mean ± std.	mean ± std.
$N = 15$, $N_b = 2^*$	0.091	0.241	0.532 ± 0.039	20.280 ± 0.436
$N = 20$, $N_b = 5$	8.773	44.342	0.565 ± 0.053	20.544 ± 0.467

† Accumulated velocity tracking error * Run 1 MM iteration

N_b. Table II shows an example with 10 simulations, where these heuristics effectively reduce the runtime while maintaining similar closed-loop performance. However, this requires application-specific tuning. We leave the development of a general approach to reduce the complexity for future work.

V. CONCLUSION

We propose a risk-sensitive MPC formulation for an MoE model that enables the application of the MM principle. This approach derives a convex upper bound of the nonconvex optimal control problem with a nonconvex collision penalty, allowing us to solve the original nonconvex problem through a sequence of convex surrogate problems. A numerical experiment validates the effectiveness and the runtime benefits of the proposed approach, though the solver currently faces computational challenges due to the exponential growth in complexity from scenario tree branching. Future work will focus on developing an efficient convex inner solver and scenario reduction techniques to address this computational burden. Moreover, since the closed-loop behavior is highly sensitive to the risk-sensitivity parameter, we plan to investigate on adaptive parameter tuning based on the specific control scenario.

REFERENCES

[1] N. Montali et al., "The Waymo open Sim agents challenge," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 59151–59171.

[2] R. Wang, F. S. Acerbo, T. D. Son, and P. Patrinos, "Imitation learning from observations: An autoregressive mixture of experts approach," 2024, *arXiv:2411.08232*.

[3] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. ECCV*, 2020, pp. 683–700.

[4] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markov Jump Linear Systems*. London, U.K.: Springer, 2005.

[5] A. Majumdar, S. Singh, A. Mandlekar, and M. Pavone, "Risk-sensitive inverse reinforcement learning via coherent risk models," in *Proc. Robot., Sci. Syst.*, vol. 16, 2017, p. 117.

[6] L. J. Ratliff and E. Mazumdar, "Inverse risk-sensitive reinforcement learning," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1256–1263, Mar. 2020.

[7] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, 1994.

[8] K. Lange, *MM Optimization Algorithms*. Philadelphia, PA, USA: SIAM, 2016.

[9] R. Wang, M. Schuurmans, and P. Patrinos, "Interaction-aware model predictive control for autonomous driving," in *Proc. ECC*, 2023, pp. 1–6.

[10] H. Hu and J. F. Fisac, "Active uncertainty reduction for human-robot interaction: An implicit dual control approach," in *Proc. 15th Algorithmic Found. Robot.*, 2023, pp. 385–401.

[11] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *Proc. IEEE ICRA*, 2018, pp. 3399–3406.

[12] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin, and A. D. Ames, "Interactive multi-modal motion planning with branch model predictive control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5365–5372, Apr. 2022.

[13] R. A. Howard and J. E. Matheson, "Risk-sensitive Markov decision processes," *Manag. Sci.*, vol. 18, no. 7, pp. 356–369, 1972.

[14] D. Jacobson, "Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games," *IEEE Trans. Autom. control*, vol. 18, no. 2, pp. 124–131, Apr. 1973.

[15] P. Whittle, "Entropy-minimising and risk-sensitive control rules," *Syst. Control Lett.*, vol. 13, no. 1, pp. 1–7, 1989.

[16] P. Whittle, "Risk sensitivity, a strangely pervasive concept," *Macroeconomic Dynamics*, vol. 6, no. 1, pp. 5–18, 2002.

[17] J. Moos, K. Hansel, H. Abdulsamad, S. Stark, D. Clever, and J. Peters, "Robust reinforcement learning: A review of foundations and recent advances," *Mach. Learn. Knowl. Extract.*, vol. 4, no. 1, pp. 276–315, 2022.

[18] E. Noorani, J. S. Baras, and K. H. Johansson, "Risk-sensitive RL using sampling-based expectation-maximization," in *Proc. 62nd IEEE CDC*, 2023, pp. 7015–7020.

[19] H. Liang and Z.-Q. Luo, "Bridging distributional and risk-sensitive reinforcement learning with provable regret bounds," *J. Mach. Learn. Res.*, vol. 25, no. 221, pp. 1–56, 2024.

[20] J. R. Medina, D. Sieber, and S. Hirche, "Risk-sensitive interaction control in uncertain manipulation tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 502–507.

[21] M. Wang, N. Mehr, A. Gaidon, and M. Schwager, "Game-theoretic planning for risk-aware interactive agents," in *Proc. IEEE/RSJ IROS*, 2020, pp. 6998–7005.

[22] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager, "Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction," in *Proc. IEEE/RSJ IROS*, 2020, pp. 11205–11212.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc., Series B*, vol. 39, no. 1, pp. 1–22, 1977.

[24] J. Peters, K. Mulling, and Y. Altun, "Relative entropy policy search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 24, 2010, pp. 1607–1612.

[25] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," 2018, *arXiv:1805.00909*.

[26] N. Moehle, "Risk-sensitive model predictive control," 2021, *arXiv:2101.11166*.

[27] R. Wang, A. Bodard, M. Schuurmans, and P. Patrinos, "EM++: A parameter learning framework for stochastic switching systems," 2024, *arXiv:2407.16359*.

[28] G. C. Pflug and A. Pichler, *Multistage Stochastic Optimization* (Springer Series in Operations Research and Financial Engineering). Cham, Switzerland: Springer Int. Publ., 2014.

[29] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*. Philadelphia, PA, USA: SIAM, 2021.

[30] H. Tembine, Q. Zhu, and T. Başar, "Risk-sensitive mean-field games," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 835–850, Apr. 2014.

[31] R. Wang, M. Schuurmans, and P. Patrinos, "Risk-Sensitive model predictive control for interaction-aware planning—A sequential convexification algorithm," early access, Jul. 7, 2025, doi: 10.1109/LCSYS.2025.3586667.

[32] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis* (Grundlehren Der Mathematischen Wissenschaften), vol. 317. Berlin, Germany: Springer, 1998.

[33] R. T. Rockafellar, *Convex Analysis*, vol. 11. Princeton, NJ, USA: Princeton Univ., 1997.

[34] B. Evens, M. Schuurmans, and P. Patrinos, "Learning MPC for interaction-aware autonomous driving: A game-theoretic approach," in *Proc. ECC*, 2022, pp. 34–39.

[35] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, pp. 25–57, Mar. 2006.

[36] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.

[37] "MOSEK optimizer API for Python 11.0.12." 2024. [Online]. Available: https://docs.mosek.com/11.0/pythonapi/intro_info.html

Authorized licensed use limited to: KU Leuven University Library. Downloaded on September 23,2025 at 13:52:05 UTC from IEEE Xplore. Restrictions apply.