Risk-Sensitive Reinforcement Learning With Exponential Criteria

Erfaun Noorani, Christos N. Mavridis, *Member, IEEE*, and John S. Baras, *Life Fellow, IEEE*

***Abstract*—While reinforcement learning (RL) has shown experimental success in a number of applications, it is known to be sensitive to noise and perturbations in the parameters of the system, leading to high variability in the total reward amongst different episodes on slightly different environments. To introduce robustness, as well as sample efficiency, risk-sensitive RL methods are being thoroughly studied. In this work, we provide a definition of robust RL policies and formulate a risk-sensitive RL problem to approximate them, by solving an optimization problem with respect to a modified objective based on exponential criteria. In particular, we study a model-free risk-sensitive variation of the widely used Monte Carlo policy gradient algorithm, and introduce a novel risk-sensitive online Actor–Critic algorithm based on solving a multiplicative Bellman equation using stochastic approximation updates. Analytical results suggest that the use of exponential criteria generalizes commonly used ad-hoc regularization approaches, improves sample efficiency, and introduces robustness with respect to perturbations in the model parameters and the environment. The implementation, performance, and robustness properties of the proposed methods are evaluated in simulated experiments.**

***Index Terms*—Actor–critic, risk-sensitive reinforcement learning (RL), robust control.**

I. Introduction

In stochastic decision systems, where uncertainty leads to risk (variability) in a desired performance metric, solving a stochastic optimal control task (viz., reinforcement learning (RL) applications) by optimizing a risk-neutral objective, often leads to control policies that might perform poorly, especially in real-world applications. This is due to the fact that risk-neutral objectives typically consist of a long-run expectation of the desired metric (average performance) which have been shown to be nonrobust to noise and model uncertainties [1]. This phenomenon is observed in widely used RL algorithms, such as Actor–Critic methods, which are often unable to maintain their performance under slight variations in the environment at the testing time. Fig. 1 shows the training and testing performance of an Actor–Critic agent in an inverted pendulum problem (see Section V-A) with perturbed model parameters. While training is conducted with a given pole length, the performance of the trained agent is evaluated in a set of environments with different pole lengths. It is clear that, in the risk-neutral case, the change in the pole length results in significant performance degradation. To mitigate such issues, risk-sensitive RL investigates alternative optimization approaches, by incorporating constraints and alternative objective functions to induce robustness with respect to variations and uncertainties of the environment [2], [3], [4].

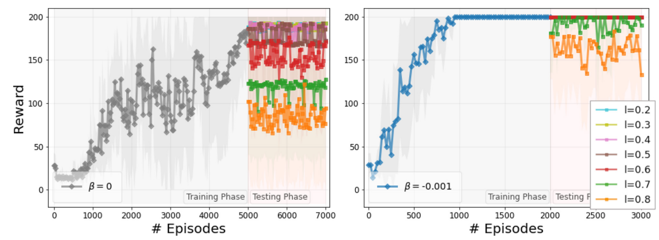


Fig. 1. Generalization performance with respect to perturbations in the model parameters. Risk-neutral (left) and risk-sensitive (right) actor–critic RL algorithms trained in the Cart-Pole environment with pole length $l = 0.5$ are tested for different pole length values $l \in [0.2, 0.8]$. Average reward and 90% confidence intervals over a running window of ten episodes are depicted.

Related Work: Robustness has been studied extensively in optimization and optimal control [5], [6], [7], [8]. In RL problems, where uncertainties in the system demand that distributional information is taken into account, robustness is associated with a stochastic optimization problem of the form

$$\max_{\pi \in \Pi} . \inf_{\rho \in \Psi} \mathbb{E}_{x\sim\pi,\zeta\sim\rho}[R(x,\zeta)]$$

where $x \in X$ are the design parameters with distribution $\pi \in \Pi$, $\zeta \in Z$ is a random vector with distribution $\rho \in \Psi$ representing uncertain system parameters, and $R : X \times Z \to [0, \infty)$ is an objective (reward) function to be maximized. Here, the system's sensitivity to maximum uncertainty (e.g., noise and disturbances) is maximized. [9]. This problem is closely related to mini-max games [6].

A number of risk-sensitive RL approaches have been studied in recent years; from constructing constraint stochastic optimization problems [10], [11], [12] or approximately solving mini-max optimization problems [13], to investigating different statistical measures of the objective function [14].

Received 1 February 2025; revised 6 April 2025; accepted 14 May 2025. Date of publication 18 June 2025; date of current version 24 July 2025. This work was supported in part by the U.S. Office of Naval Research (ONR) under Grant N00014-17-1-2622; in part by the U.S. NSF under Grant ECCS2127605; in part by the Clark Foundation Distinguished Fellowship; and in part by the University of Maryland Graduate School Ann G. Wylie Fellowship. This article was recommended by Associate Editor S. J. Yoo. *(Corresponding author: Erfaun Noorani.)*

Erfaun Noorani and John S. Baras are with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland at College Park, College Park, MD 20742 USA (e-mail: noorani@umd.edu; baras@umd.edu).

Christos N. Mavridis is with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: mavridis@kth.se).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2025.3575240.

Digital Object Identifier 10.1109/TCYB.2025.3575240

The latter approach often provides benefits to algorithmic implementations, since the computational problems associated with constraint optimization and the problems associated with the existence of multiple Nash equilibria are avoided. In particular, the algorithms in [15], [16], [17], and [18] use the conditional value at risk (CVaR) for policy search and the algorithms in [19], [20], [21], and [22] use variance as the desired risk measure.

Although these are ad-hoc approaches developed by experimental observations, there is a duality connection between KL- and entropy-regularized objectives and entropic risk measures [3], [23], [24], [25], [26], [27], [28], associated with exponential criteria of the form

$$\max_{\pi\in\Pi} \quad \frac{1}{\beta}\mathbb{E}_{x\sim\pi}\big[\exp(\beta R(x))\big].$$

In addition to this connection, exponential criteria are well understood in the context of risk-sensitive control [7], [8], [29], where the equivalence of the problem of robust output-feedback control of general nonlinear, set valued, Markov chain, partially observed systems to a risk-sensitive partially observed control problem with an exponential of an integral cost criterion has been shown. In the case of known models, the solution has the known structure of two Hamilton–Jacobi–Bellman (HJB) equations: one forwards in time that computes online the information state of the problem (i.e., the sufficient statistics for the control) and one backwards in time that computes offline the control as a memory-less function of the information state. These results naturally suggest that, in the case of unknown models, RL with exponential criteria may be introduced to potentially increase robustness in RL applications using online data-driven estimation updates.

Contribution: In this work, we study the effect of exponential criteria on the robustness of the learned policy of an RL agent. In particular, we:

1) formulate the risk-sensitive RL problem as an optimization problem with a modified objective using exponential criteria and show its connection to KL-regularized RL methods (Section II-C);
2) provide a definition of robustness for RL policies and show that the use of exponential criteria results in robust RL policies with given probability bounds on the observed cumulative rewards in the form of concentration inequalities (Section II-D);
3) provide new analytic results for the implementation of the risk-sensitive REINFORCE algorithm based on exponential criteria introduced in [4] regarding the update rule and the convergence of the parameters (Section III);
4) develop a risk-sensitive online actor–critic (OAC) algorithm, by approximately solving a risk-sensitive multiplicative Bellman equation with stochastic approximation updates (Section IV);
5) quantify the robustness of the proposed methods in terms of the CVaR values of the total reward in simulated experiments under model parameter perturbations (Section V).

Contributions 1) and 2) provide a formal definition of robust RL and its connection to risk-sensitive RL with exponential criteria, which is lacking from existing work. Contributions 3) and 4) extend the algorithmic and experimental results introduced by Noorani and Baras [4] and Noorani et al. [30], by providing new analytic results (Appendix A, Section IV-A) and appropriate implementation details (Algorithm 2). Potential shortcomings of the implementation of risk-sensitive RL approaches with exponential criteria are also discussed.

Our experimental results support our theoretical analysis and suggest that the proposed problem formulation using exponential criteria is suitable for risk-sensitive RL. The proposed risk-sensitive RL methods inherit computational and convergence properties of widely used RL algorithms, can accelerate the learning process, and can reduce the variance of the learned policies under model uncertainty, resulting in policies that show enhanced robustness with respect to environmental and model perturbations.

II. Risk-Sensitive Reinforcement Learning With Exponential Criteria

In this section, we formulate the problem of risk-sensitive RL and show its connection to exponential criteria. We provide an explicit definition of robustness and risk-sensitivity, and show that the use of exponential criteria is associated with a min–max optimization problem that results in robust RL policies with known concentration bounds on the observed rewards under environmental perturbations.

A. Reinforcement Learning Preliminaries

The RL problem is typically modeled using a Markov decision process (MDP) which is represented by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p_0, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are, respectively, the state and action spaces (may each be discrete or continuous). The probability distribution p_0 is the initial state distribution, prescribing a probability on the starting state. The kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel (unknown to the agent), where $\Delta(\mathcal{S})$ denotes the space of probability distributions on $\mathcal{S}$. The function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function; and $\gamma \in [0, 1)$ is a discounting factor. The behavior of an RL agent is determined by its policy. Here, we consider randomized policies.

A (randomized) policy $\pi(\cdot|s) \in \Pi$ is a probability distribution over the action space given the state, which prescribes the probability of taking an action $a \in \mathcal{A}$ when in state s. Stochastic policies are smooth and continuous functions and therefore are more suitable for gradient-based methods. At each time-step t, the agent perceives the state of the environment s_t, and executes an action a_t according to its policy, a differentiable parameterized policy (e.g., a neural network), $\pi(\cdot|s_t; \theta)$, where $\theta \in \mathbb{R}^d$ is a vector of parameters. Then, the system transitions to a successor state s_{t+1} according to transition probability $p(s_{t+1}|s_t, a_t)$ (unknown to the agent) and the agent receives a reward $r_t := r(s_t, a_t)$. By following policy π, the agent generates trajectory $\tau(\pi)$ (a sequence of states and actions). The agent's policy and the system transition probabilities induce a trajectory distribution, a probability distribution $\rho_\theta \in \Psi$ over the possible trajectories $\mathcal{T}$, given by

$$\rho_\theta(\tau) = p_0 \prod_{t=0}^{|\tau|-1} \pi(a_t|s_t;\theta)p(s_{t+1}|s_t,a_t). \tag{1}$$

The RL agent aims to find a policy that maximizes the sum of rewards over a time period, called episode. Since the observed rewards r_t are random variables, in risk-neutral RL, the typical objective is to optimize for the expected (discounted) cumulative reward

$$\max_\theta . \; J(\theta) := \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\Big], \;\; R(\tau) = \sum_{t=0}^{|\tau|-1} \gamma^t r(s_t,a_t) \tag{2}$$

where $R(\tau)$ is the trajectory's (γ-discounted) total reward. The expectation is taken with respect to the trajectory distribution. That is, the expectation is taken over the space of trajectories $\mathcal{T}$ generated by following the policy, i.e., $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t;\theta)$ and $s_{t+1} \sim p(\cdot|s_t,a_t)$.

B. Risk-Sensitive Reinforcement Learning

Risk-sensitivity in RL is often associated with the following general problem:

$$\max_\theta \quad \inf_{\rho_\theta\in\Psi} \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau(\theta))\Big] \tag{3}$$

which induces distributional robustness with respect to the probability distribution over the possible trajectories $\mathcal{T}$. Maximization over the parameter space $\theta \in \mathbb{R}^d$ simulates optimization over all policies $\pi \in \Pi$. Minimization over the distributions ρ_θ corresponds to reducing the sensitivity of the uncertainties that affect ρ_θ, which include both the initial state distribution p_0, and the transition probabilities P, i.e., all uncertainties with respect to the model parameters and any noise perturbation of the system dynamics. Typically the space Ψ is constrained to a closed set of distributions that defines a tradeoff between optimality and conservativeness of the policy. However, solving (3) with dynamic programming and game theoretic methods becomes intractable in large state/action spaces, and methods that approximate its solution have been studied [11], [12], [13], including the use of different statistical measures of the objective function to avoid the minimization over the distributions ρ_θ [16], [17], [19], [20], [21].

In this work, we focus on the following definition of a risk-sensitive RL problem that incorporates an inherent regularization term for the set of distributions ρ_θ:

$$\max_\theta . \begin{cases} \sup_{\hat\rho}\Big\{\mathbb{E}_{\tau\sim\hat\rho}[R(\tau)] - \frac{1}{\beta}D_{KL}\big(\hat\rho,\rho_\theta\big)\Big\}, & \beta > 0 \\ \inf_{\hat\rho}\Big\{\mathbb{E}_{\tau\sim\hat\rho}[R(\tau)] - \frac{1}{\beta}D_{KL}\big(\hat\rho,\rho_\theta\big)\Big\}, & \beta < 0 \end{cases} \tag{4}$$

where $D_{KL}(\cdot,\cdot)$ represents the Kullback–Leibler divergence measure defined in (6). The optimization problem (4) is essentially a game between the trajectory distribution $\hat\rho$ that tries to find a worst-case scenario of the cumulative reward while staying close to a baseline distribution ρ_θ, and the parameter vector θ that tries to optimize for the worst-case expected cumulative reward. The use of baseline terms in RL is widely adopted [1] and is further explained in Section III. The parameter β is the risk-sensitive parameter that controls the behavior of the agent and the weight of the regularization term. In particular, $\beta > 0$ induces a risk-seeking (optimistic) approach, while $\beta < 0$ invokes a risk-averse (pessimistic) approach [25], [31].

C. Risk-Sensitive RL With Exponential Criteria

Problem (4) is a game-theoretic formulation of the risk-sensitive RL problem, which can be hard to solve directly. However, it is well known (see [32], [33]) that the following duality relationship, with respect to a Legendre-type transform, holds.

Theorem 1: Consider a measurable space $(\Omega,\mathcal{F})$, where $\mathcal{F}$ is a σ-algebra on Ω. Let $\mathcal{P}(\Omega)$ be a set of probability measures $P : \Omega \to [0,1]$, and $P_\mu, P_\nu \in \mathcal{P}(\Omega)$. In addition, consider a bounded measurable function $Z : \Omega \to \mathbb{R}$. Then, the free energy is defined as

$$J_{l_\beta}(Z) \;= \frac{1}{\beta}\log \mathbb{E}_{P_\mu}\Big[e^{\beta Z}\Big] \tag{5}$$

and the KL divergence measure

$$D_{KL}\big(P_\nu,P_\mu\big) = \begin{cases} \int \log\Big(\frac{dP_\nu}{dP_\mu}\Big)dP_\nu, & \text{if } C_{KL}\big(P_\nu,P_\mu\big) \\ \infty, & \text{otherwise} \end{cases} \tag{6}$$

are in duality with respect to a Legendre-type transform, in the following sense:

$$J_{l_\beta}(Z) \;= \begin{cases} \sup_{P_\nu\in\mathcal{P}(\Omega)}\Big\{\mathbb{E}_{P_\nu}[Z] - \frac{1}{\beta}D_{KL}\big(P_\nu,P_\mu\big)\Big\}, & \beta > 0 \\ \inf_{P_\nu\in\mathcal{P}(\Omega)}\Big\{\mathbb{E}_{P_\nu}[Z] - \frac{1}{\beta}D_{KL}\big(P_\nu,P_\mu\big)\Big\}, & \beta < 0. \end{cases} \tag{7}$$

Here, the conditions $C_{KL}(P_\nu,P_\mu)$ include $P_\nu \ll P_\mu$ and $\int \log(dP_\nu/dP_\mu)dP_\nu \in L^1(P_\nu)$.

Proof: Follows directly from [32, Th. 6] and standard algebraic manipulations. ■

Corollary 1 then follows directly from Theorem 1.

Corollary 1: The problem

$$\max_\theta . \quad J_{l_\beta}(\theta) := \frac{1}{\beta}\log \mathbb{E}_{\tau\sim\rho_\theta}\Big[\exp(\beta R(\theta))\Big] \tag{8}$$

is equivalent to (4), for the baseline distribution ρ_θ being the current trajectory distribution of the algorithm, assuming that the maximum is attained.

Notice that a Taylor expansion of (8) reveals an intuition behind how the exponential criterion incorporates risk into the objective function, since it incorporates an infinite sum of the higher moments of the return, i.e., for small β, we get

$$\frac{1}{\beta}\log\mathbb{E}\Big[e^{\beta R(\theta)}\Big] = \mathbb{E}\;\Big[R(\theta)\Big] + \frac{\beta}{2}\mathrm{Var}\Big[R(\theta)\Big] + \mathcal{O}\Big(\beta^2\Big). \tag{9}$$

Equation (9) shows how the risk-sensitive parameter β controls the weight of the moments of the cumulative reward in the objective function. We note that as the risk-sensitive parameter β approaches zero, the exponential objective (9) approaches the risk-neutral objective (2).

Remark 1: Connection to maximum-entropy RL (see [22]). Notice that we can simplify (7) by making heuristic assumptions on the measure P_μ. In particular, it is known that the maximum-entropy objective is equivalent to the maximization of the KL-regularized objective with respect to a uniform distribution as the reference policy [34]. In other words, by assuming that P_μ is a uniform probability measure, and for $\beta > 0$, (7) and (8) imply that the problem of maximizing $J_{l_\beta}(\theta)$ with respect to θ is equivalent to

$$\max_{\rho} . \left\{\mathbb{E}_{\tau\sim\rho}[R(\tau)] + \frac{1}{\beta}H(\rho)\right\}$$

where $H(\rho)$ represents the Shannon entropy of the distribution ρ. Thus, the maximum-entropy RL objective of [22] is a special case of the objective (8) considered in this work.

D. Policy Robustness

Given an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p_0, P, r, \gamma)$ with transition probabilities P, a fixed policy π, parameterized by θ, defines a trajectory distribution ρ_θ given by (1). RL algorithms try to find the optimal policy $\pi(\theta)$ given observations of the rewards r of $\mathcal{M}$. However, during the testing phase, when the policy $\pi(\theta)$ is applied, environment and model perturbations may alter the transition probabilities. Thus, the agent is asked to operate on a perturbed MDP $\hat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{p}_0, \hat{P}, r, \gamma)$, where $\hat{P}$ represents the perturbed system of transition probabilities. This is especially the case when training takes place in simulation environments while testing is transferred to a real system.

In this case, risk-sensitivity can be associated with a measure of robustness of a policy $\pi(\theta)$, quantified by a lower bound on the probability of good performance when the transition distribution $\hat{\rho}$ during testing deviates from the distribution ρ_θ induced by $\pi(\theta)$. In this work, we will adopt the following definition of robustness of a policy $\pi(\theta)$.

Definition 1: Let $\pi(\theta)$ be a given policy and ρ_θ be its associated trajectory distribution given by (1) with transition probabilities P. In addition, let $\hat{\rho}$ be a trajectory distribution generated by $\pi(\theta)$ given a perturbed system of transition probabilities $\hat{P}$. The policy $\pi(\theta)$ is (ξ, δ, ϵ)-robust if, for $\delta, \epsilon > 0$, and under the condition $D(\hat{\rho}, \rho_\theta) \le \epsilon$, it holds that

$$\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau(\theta)) > \xi] \ge 1 - \delta(\xi, \epsilon) \tag{10}$$

where $D(\cdot, \cdot)$ represents the KL divergence defined in (6).

In general, nontrivial sets of parameters (ξ, ϵ, δ) such that the condition (10) is satisfied cannot be found. However, for optimal policies with respect to (8), we can analytically provide such parameters using standard concentration inequalities. Theorem 2 provides upper bounds on the probability of the tails of the cumulative rewards R, in the case of bounded reward ($R \le R_{\max}$ almost surely). Note that $R_{\max} = [r_{\max}(1-\gamma^T)/1-\gamma]$ when the per step reward is bounded $r \le r_{\max}$.

Theorem 2: Let $\pi(\theta^*)$ be an optimal policy with respect to (8), i.e., $\pi(\theta^*) = \arg\max_\theta J_{l_\beta}(\theta)$, and ρ_{θ^*} be its associated trajectory distribution given by (1) with transition probabilities P. In addition, let $\hat{\rho}$ be a trajectory distribution generated by $\pi(\theta)$ given a perturbed system of transition probabilities $\hat{P}$ such that $D(\hat{\rho}, \rho_{\theta^*}) \le \epsilon$. Then, the following inequalities hold:

$$\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau) \ge \xi] \le \frac{1}{\xi}J^*_{l_\beta} + \frac{\epsilon}{\beta\xi},\ \ \beta > 0 \tag{11}$$

$$\begin{aligned}&\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau) \le \xi] \le \\ &\quad\le \frac{R_{\max}}{R_{\max}-\xi}\left(1 - \frac{1}{R_{\max}}J^*_{l_\beta} + \frac{\epsilon}{|\beta|R_{\max}}\right),\ \ \beta < 0\end{aligned} \tag{12}$$

where $J^*_{l_\beta} = (1/\beta)\log\mathbb{E}_{\tau\sim\rho_{\theta^*}}[\exp(\beta R(\tau))]$.

Proof: For (12), using Markov's inequality, we get

$$\begin{aligned}\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau) \ge \xi] &\le \frac{\mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)]}{\xi} \\ &\le \frac{1}{\xi}\left(J^*_{l_\beta} + \frac{1}{\beta}D\big(\hat{\rho}, \rho_{\theta^*}\big)\right) \\ &\le \frac{1}{\xi}J^*_{l_\beta} + \frac{\epsilon}{\beta\xi}\end{aligned} \tag{13}$$

where we have used (7) for $\beta > 0$ to get

$$\begin{aligned}J^*_{l_\beta} &= \frac{1}{\beta}\log\mathbb{E}_{\tau\sim\rho_{\theta^*}}\Big[\exp(\beta R(\tau))\Big] \\ &= \sup_\rho\left\{\mathbb{E}_{\tau\sim\rho}[R(\tau)] - \frac{1}{\beta}D(\rho, \rho_{\theta^*})\right\} \\ &\ge \mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)] - \frac{1}{\beta}D\big(\hat{\rho}, \rho_{\theta^*}\big)\end{aligned}$$

which implies that $\mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)] \le J^*_{l_\beta} + (1/\beta)D(\hat{\rho}, \rho_{\theta^*})$.

Similarly, for (12), using reverse Markov's inequality and assuming that $R < R_{\max},\ a.s.$, we get

$$\begin{aligned}\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau) \le \xi] &\le \frac{R_{\max} - \mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)]}{R_{\max}-\xi} \\ &\le \frac{R_{\max}}{R_{\max}-\xi}\left(1 - \frac{1}{R_{\max}}J^*_{l_\beta} - \frac{1}{R_{\max}\beta}D\big(\hat{\rho}, \rho_{\theta^*}\big)\right) \\ &\le \frac{R_{\max}}{R_{\max}-\xi}\left(1 - \frac{1}{R_{\max}}J^*_{l_\beta} + \frac{\epsilon}{|\beta|R_{\max}}\right)\end{aligned} \tag{14}$$

where we have used (7) for $\beta < 0$ to get

$$\begin{aligned}J^*_{l_\beta} &= \frac{1}{\beta}\log\mathbb{E}_{\tau\sim\rho_{\theta^*}}\Big[\exp(\beta R(\tau))\Big] \\ &= \inf_\rho\left\{\mathbb{E}_{\tau\sim\rho}[R(\tau)] - \frac{1}{\beta}D(\rho, \rho_{\theta^*})\right\} \\ &\le \mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)] - \frac{1}{\beta}D\big(\hat{\rho}, \rho_{\theta^*}\big),\ \ \beta < 0\end{aligned}$$

which implies that $-\mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)] \le -J^*_{l_\beta} - [1/\beta]D(\hat{\rho}, \rho_{\theta^*})$. ■

Remark 2: Note that in Theorem 2, the term $J^*_{l_\beta}$ does not depend on the perturbed system of transition probabilities $\hat{P}$ in the test environment.

Equations (11) and (12) give upper bounds on the probability of the two tails of the cumulative rewards R. In particular, a risk-averse agent tries to optimize for the maximum average reward weighing in the maximization of the decay of the left tail of the distribution of the total reward, while a risk-seeking agent weights in the maximization of the decay of the right tail of the reward distribution. This is consistent with the following theorem proven in [31] using the Gartner–Ellis theorem of large deviation.

Theorem 3 [31]: For a given negative risk parameter (risk-aversion) $\beta < 0$, the maximization of the risk-sensitive exponential criterion J_{l_β} in (8) is equivalent to the maximization of the exponential rate of decay of the left tail of the system's trajectory reward distribution, i.e., for a given $\beta < 0$, there exists a constant $\psi \in \mathbb{R}$ such that

$$\arg\max_\pi J_{l_\beta}(\pi) = \lim_{|\tau|\to\infty} \arg\min_\pi \mathbb{P}[R(\tau) < \psi]$$

where $\mathbb{P}[R(\tau) < \psi]$ denotes the probability of the event $R < \psi$. Similarly, for a given positive risk parameter (risk-seeking) $\beta > 0$, the maximization of the risk-sensitive exponential criterion J_{l_β} in (8) is equivalent to minimization of the exponential rate of decay of the right tail of the system's trajectory reward distribution, that is, for a given $\beta > 0$, there exists a constant ψ such that

$$\arg\max_\pi J_{l_\beta}(\pi) = \lim_{|\tau|\to\infty} \arg\max_\pi \mathbb{P}[R(\tau)) > \psi].$$

Based on Theorem 2, Corollary 2 shows that the risk-averse policy ($\beta < 0$) with respect to (8) is a (ξ, δ, ϵ)-robust policy according to Definition 1.

Corollary 2: Let an optimal policy $\pi(\theta^*) = \arg max_\theta J_{l_\beta}(\theta)$ with respect to (8) for $\beta < 0$. Then, $\pi(\theta^*)$ is (ξ, δ, ϵ)-robust according to Definition 1 with

$$\delta(\xi, \epsilon) = \frac{R_{\max}}{R_{\max} - \xi}\left(1 - \frac{1}{R_{\max}} J^*_{l_\beta} + \frac{\epsilon}{|\beta| R_{\max}}\right). \tag{15}$$

In addition, for a given $\delta = \bar{\delta}$, we can quantify ξ by

$$\xi = R_{\max} - \frac{R_{\max}}{\bar{\delta}}\left(1 - J^*_\beta + \frac{\epsilon}{\beta R_{\max}}\right). \tag{16}$$

Proof: It follows from (12) since $\mathbb{P}[R > \xi] = 1 - \mathbb{P}[R \leq \xi]$. ■

As a last remark, we have shown how the optimization problem (8) is connected to risk-sensitivity and robustness of the learned policy with respect to model perturbations. However, as will be discussed in Section IV-A, the presence of the logarithmic nonlinearity in (8) creates computational problems in algorithmic implementations. For this reason, throughout the remainder of this article, we will study the equivalent (in terms of optimal policy) problem

$$\max_\theta . \; J_\beta(\theta) := \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\exp(\beta R(\theta))\Big]. \tag{17}$$

III. Policy Gradient With Exponential Criteria

In this section, we present a brief overview of the risk-sensitive REINFORCE algorithm introduced in [4] and provide new analytic results for its implementation regarding its update rule and the convergence of its parameters.

A. Policy Gradient and the REINFORCE Method

Policy gradient (PG) methods are a class of policy search methods that use gradient ascend/descend schemes to search for the optimal policy [35]. That is, at each iteration of the algorithm t, the parameters of the policy are updated using the following update rule:

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)} \tag{18}$$

where $\alpha \in \mathbb{R}$ is a constant step size, i.e., learning rate, and $\widehat{\nabla J(\theta_t)} \in \mathbb{R}^d$ is an unbiased estimate of the gradient with respect to the policy parameter θ. The well-known REINFORCE [36] and Actor–Critic [37] algorithms are examples of Monte-Carlo and recursive on-policy PG algorithms, respectively, particularly suitable for continuous action spaces.

An estimate of the gradient of J in (18) with respect to the policy parameters can be obtained using the PG theorem [35], that is

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\sum_{t=0}^{|\tau|-1} \nabla \log \pi_\theta(a_t|s_t;\theta)\Big]. \tag{19}$$

The PG theorem suggests that the gradient estimate in (18) can be computed by Monte Carlo estimation of the expectation in (19). Equation (19) can be rewritten in terms of the reward-to-go $R_t := \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$ as follows [1]:

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1} R_t \nabla \log \pi_\theta(a_t|s_t;\theta)\Big]. \tag{20}$$

Using (20), the update rule in the standard REINFORCE algorithm is obtained and is given by

$$\theta_{t+1} = \theta_t + \alpha R_t \frac{\nabla \pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta)}. \tag{21}$$

To further reduce the variance associated with the gradient estimations of (19) and (20), which is imperative in complex environments, baseline methods, based on subtracting an appropriately chosen baseline from the reward-to-go R_t, have been proposed. Using baselines, one gets

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\Big(R_t - b(s_t)\Big)\nabla \log \pi_\theta(a_t|s_t;\theta)\Big] \tag{22}$$

where $b(s_t)$ is a state-dependent function [1]. State-dependent baselines are guaranteed to exist, introduce no bias, and show better convergence properties in practice. However, they are hard to find [38]. A common baseline in practice is the estimate of the value function, i.e., $b(s_t) = V^{\pi_\theta}(s_t) := \mathbb{E}_{\tau\sim\rho_\theta}[R_t|s_t]$. As we will show, a particularly convenient property of using exponential criteria is that it alleviates the need for such approaches [25].

B. Risk-Sensitive REINFORCE (R-REINFORCE)

Risk-sensitive REINFORCE (R-REINFORCE) [4] is the risk-sensitive counterpart of REINFORCE based on the objective (17). In R-REINFORCE, the update rule (20) is replaced by

$$\nabla J_\theta(\theta) \propto \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1} e^{\beta R_t}\nabla \log \pi_t(\theta)\Big]. \tag{23}$$

The derivation of this formula is based on a risk-sensitive variation of the PG theorem [35]. These results are provided in Appendix A. Given (23), the R-REINFORCE update rule reads as

$$\theta_{t+1} = \theta_t + \frac{\alpha}{\beta} e^{\beta R_t}\frac{\nabla \pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta)} \tag{24}$$

Algorithm 1 Risk-Sensitive REINFORCE

Input: a differentiable policy $\pi(a|s;\theta)$.
Algorithm parameters: step-size $\alpha > 0$, discount factor $\gamma > 0$, risk parameter β.
Initialization: $\theta = \theta_0 \in \mathbb{R}^d$.
while θ **not converged do**
  Generate an episode $s_0, a_0, \ldots, s_{|\tau|-1}, a_{|\tau|-1}$ **by** $s_0 \sim p_0$**,** $a_t \sim \pi(\cdot|s_t;\theta)$**,** $s_{t+1} \sim p(\cdot|s_t, a_t)$
  for $t = 0$ **to** $|\tau| - 1$ **do**
    $\hat{R} \leftarrow \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r_t$
    $\theta_{t+1} \leftarrow \theta_t + \alpha\gamma^t \ \frac{1}{\beta} e^{\beta\hat{R}} \ \nabla \log \pi(a_t|s_t;\theta_t)$
  end for
end while

and is a stochastic approximation algorithm (see [39]). We provide the convergence analysis of the parameters θ in Appendix A-A. The implementation of the Risk-sensitive REINFORCE algorithm is given in Algorithm 1. For more details, the readers are referred to [4] and the references therein.

Note that the update rule is not proportional to the reward-to-go $R_t := \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$, but to the exponential

$$\beta e^{\beta R_t} = \frac{1}{\beta} \prod_{t'=t}^{|\tau|-1} \exp\{\gamma^{t'-t} \beta r(s_{t'}, a_{t'})\}. \tag{25}$$

Remark 3: Substituting the exponential with its Taylor series expansion (see (9)), reveals that the risk-sensitive objective provides a natural baseline (see Section III-A). This is empirically shown in [4]. The baseline term can be derived by expanding the exponential function and combining all terms, except for the one proportional to R_t, i.e., $\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}[\sum_{t=0}^{|\tau|-1}(R_t - b(s_t))\nabla \log \pi_\theta(a_t|s_t;\theta)]$, where $b(s_t) = -([1/\beta] + [\beta R_t^2/2] + \ldots)$. In Section V, we show that such baseline leads to significant variance reduction and acceleration of learning process.

IV. Actor–Critic With Exponential Criteria

Actor–Critic methods improve the policy using gradient methods and use a critic network to estimate the value function and use it to bootstrap an estimate of the reward-to-go [37]. The value function $V^{\pi_\theta}(s_t) \simeq \mathbb{E}_{\tau\sim\rho_\theta}[R_t|s_t]$, satisfies Bellman's equation

$$V^{\pi_{\theta^*}}(s) = \mathbb{E}_{a\sim\pi_{\theta^*}}\Big[r(s,a) + \gamma V^{\pi_{\theta^*}}(s')|s\Big] \tag{26}$$

which is a contraction mapping that gives rise to stochastic approximation algorithms that try to asymptotically minimize the mean-squared error

$$\min{}_\theta \ \mathbb{E}_{a\sim\pi_\theta}\Big[\|r(s,a) + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)\|^2 \mid s\Big]$$

forming temporal-difference actor–critic methods that employ learning models (e.g., neural networks or other models [40], [41]). Such methods use two learning systems to estimate the parameters θ_t of the optimal policy $\pi(a_t|s_t;\theta_t)$ (actor) and the parameters w_t of the value function $V(s_t; w_t)$ (critic), that is

$$\begin{cases} \theta_{t+1} = \theta_t + \alpha\Big(\hat{R}_t - V(s_t; w_t)\Big)\frac{\nabla\pi(a_t|s_t;\theta_t)}{\pi(a_t|s_t;\theta_t)} \\ w_{t+1} = w_t - \bar{\alpha}\nabla J_c(s_t; w_t, \theta_t) \end{cases} \tag{27}$$

where $J_c(s_t; w_t, \theta_t) = \|\hat{R}_t - V(s; w_t)\|^2$. In this case, $\hat{R}_t$ is an estimate of the reward-to-go R_t given by

$$\hat{R}_t := r(s, \pi_{\theta_t}) + \gamma V(s'; w_t).$$

A. Risk-Sensitive Online Actor–Critic

In this section, we develop a risk-sensitive counterpart of the temporal-difference actor–critic method. In contrast to the risk-neutral case, in the risk-sensitive RL setting the optimal control problem is often associated with an undiscounted version of the cost function J_{l_β} in (8)

$$\max{}_\pi \bar{J}_{l_\beta}(\pi) := \limsup_{n\to\infty} \frac{1}{n} \log \mathbb{E}\Big[e^{\beta\sum_{l=0}^{n-1} r(s_l,a_l)}|s_0\Big]. \tag{28}$$

Notice that it has been assumed that $\gamma = 1$, and the time-average limit has been added to ensure boundedness of the cost. It has been shown (see [23], [42]) that by defining a value function $\bar{V}_{l_\beta}^*(s_k) = \max_\pi \mathbb{E}\Big[e^{\beta\sum_{l=k}^{t_h} r(s_l,a_l) - \log J_{l_\beta}^*}|s_k\Big]$, with t_h being the first hitting time of a distinguished state, problem (28) is equivalent to a multiplicative version of the Bellman equation which defines a nonlinear eigenvalue problem

$$\bar{V}_{l_\beta}^*(s_k) = \max_\pi \frac{e^{\beta r(s_k,a_k)}}{\bar{J}_{l_\beta}^*}\mathbb{E}\Big[V_{l_\beta}^*(s_{k+1})|s_k\Big], \quad a_k \sim \pi(\cdot|s_k). \tag{29}$$

For sufficiently small β, stochastic approximation updates in two timescales can be designed to solve the eigenvalue problem recursively implementing a policy iteration scheme and converging to an optimal stationary control that attains the optimal reward $J_{l_\beta}^* < \infty$. It is important to point out that substituting for the logarithmic value function $W(\cdot) = \log V_{l_\beta}(\cdot)$ results in an additive dynamic programming equation, that has similarities with the classical equation for average reward

$$W^*(s_k) := \max_\pi\Big\{r(s_k,a_k) + \log\mathbb{E}\Big[e^{W^*(s_{k+1})} \mid s_k\Big]\Big\} - \log J_{l_\beta}^*. \tag{30}$$

While this seems like a compelling formulation, and has indeed been followed by some authors (see [43], [44]), the problem arises when attempting to formulate an RL algorithm out of the latter dynamic programming equation. In particular, notice that, in (30), the conditional expectation with respect to the transition probabilities appears inside a logarithm. This typically leads to violation of the assumptions of the stochastic approximation algorithm used to train temporal-difference RL algorithms (e.g., stochastic gradient descent if using neural networks) [23]. As a result, the form of (30) is not convenient for Q-learning and most temporal-difference RL methods.

In this work, we consider the discounted optimal control problem in (17). According to the cost function J_β, we define the risk-sensitive value function of a policy π as $V_\beta^\pi(s_k) := (1/\beta)\mathbb{E}\Big[e^{\beta\sum_{l=k}^{\infty}\gamma^{l-k}r(s_l,a_l)}|s_k\Big]$. We further define

$$\bar{V}_\beta^\pi(s_k) := \beta V_\beta^\pi(s_k) = \mathbb{E}\Big[e^{\beta\sum_{l=k}^{\infty}\gamma^{l-k}r(s_l,a_l)}|s_k\Big]. \tag{31}$$

By definition, we get that $\bar{V}_\beta^\pi(\cdot) \geq 0$, and the following relationship holds:

$$\begin{aligned}\bar{V}_\beta^*(s_k) &:= \max_\pi \ \mathbb{E}\Big[e^{\beta\sum_{l=k}^{\infty}\gamma^{l-k}r(s_l,a_l)}|s_k\Big]\\ &= \max_\pi \ \mathbb{E}\Big[e^{\beta\left(r(s_k,a_k)+\gamma\sum_{l=k+1}^{\infty}\gamma^{l-(k+1)}r(s_l,a_l)\right)}|s_k\Big]\\ &= \max_\pi \ e^{\beta r(s_k,a_k)}\mathbb{E}\Big[\left(\bar{V}_\beta^*\right)^\gamma(s_{k+1})|s_k\Big] + \bar{\epsilon}_k(\gamma)\\ &= \max_\pi \ \mathbb{E}\Big[e^{\beta r(s_k,a_k)+\gamma\log\bar{V}_\beta^*(s_{k+1})}|s_k\Big] + \bar{\epsilon}_k(\gamma) \quad (32)\end{aligned}$$

where $\bar{V}^*(\cdot) = \bar{V}^{\pi^*}(\cdot)$ is the optimal value function resulting by the optimal control policy, and the term $\bar{\epsilon}(\gamma)$ is given by

$$\begin{aligned}\bar{\epsilon}_k(\gamma) &= e^{\beta r(s_k,a_k)}\mathbb{E}[\Big(e^{\beta\sum_{l=k+1}^{\infty}\gamma^{l-(k+1)}r(s_l,a_l)}\Big)^\gamma\\ &\qquad -\mathbb{E}\Big[e^{\beta\sum_{l=k+1}^{\infty}\gamma^{l-(k+1)}r(s_l,a_l)}|s_{k+1}\Big]^\gamma|s_k]\\ &= e^{\beta r(s_k,a_k)}\mathbb{E}[e^{\gamma\sum_{l=k+1}^{\infty}\gamma^{l-(k+1)}r(s_l,a_l)}\\ &\qquad -\left(\bar{V}_\beta^*\right)^\gamma(s_{k+1})|s_k].\end{aligned} \tag{33}$$

Note that the existence of the term $\bar{\epsilon}(\gamma)$ implies that (32) holds only approximately. The approximation error $\bar{\epsilon}(\gamma)$ depends on the statistics of the problem at hand, as well as the value of γ. A good approximation can be achieved for $\gamma \approx 1$, since strict equality in (32) holds in the case of $\gamma = 1$, since $\bar{\epsilon}_k(\gamma) = 0\ \forall k \geq 0$. This follows from the law of total expectation such that $\mathbb{E}\Big[e^{\beta\sum_{l=k+1}^{\infty}r(s_l,a_l)}|s_k\Big] = \mathbb{E}\Big[\mathbb{E}\Big[e^{\beta\sum_{l=k+1}^{\infty}r(s_l,a_l)}|s_{k+1}\Big]|s_k\Big] = \mathbb{E}\Big[(\bar{V}_\beta^*)(s_{k+1})|s_k\Big]$. Notice also how the use of the exponential has resulted in a multiplicative Bellman equation. Finally, note that the exponent γ is assumed a rational number such that the term $(\bar{V}_\beta^*)^\gamma$ is well defined. This is not restrictive, as in practice the term $\exp(\gamma\log\bar{V}_\beta^*)$ is used, leading to a similar update law to the risk-neutral case.

To develop a risk-sensitive temporal-difference RL algorithm, we use two learning systems, similar to (27), as follows:

$$\begin{cases}\theta_{t+1} = \theta_t + \alpha\frac{1}{|\beta|}\left(R_t^\beta - \bar{V}_\beta(s_t;w_t)\right)\frac{\nabla\pi(a_t|s_t;\theta_t)}{\pi(a_t|s_t;\theta_t)}\\ w_{t+1} = w_t - \bar{\alpha}\nabla J_r(s_t;w_t,\theta_t)\end{cases} \tag{34}$$

where, in contrast to the risk-neutral case in (27), here we define

$$R_t^\beta = \exp\big[\beta r(s_t,a_t) + \gamma\log\bar{V}_\beta(s_{t+1};w_t)\big] \tag{35}$$

$$\begin{aligned}J_r(s_t;w_t,\theta_t) = \|&\exp\big[\beta r(s_t,a_t) + \gamma\log\bar{V}_\beta(s_{t+1};w_t)\big]\\ &-\bar{V}_\beta(s_t;w_t)\|^2, \quad a\sim\pi_{\theta_t}\end{aligned} \tag{36}$$

forming a stochastic gradient descent approach to asymptotically minimize the mean-squared error

$$\min_w \ \mathbb{E}\Big[\|e^{\beta r(s_t,a_t)}\left(\bar{V}_\beta\right)^\gamma(s_{t+1};w) - \bar{V}_\beta(s_t;w)\|^2 \mid s_t\Big].$$

The actor parameter updates constitute a stochastic approximation algorithm based on (24), where the average reward-to-go $V_\beta(s_t;w_t) = [1/\beta]\mathbb{E}\big[e^{\beta R_k}|s_k\big]$ is estimated by the critic model. The critic parameter updates are also a stochastic approximation scheme that run at a slower timescale (see [23]). Notice that this recursion does not correspond to a fixed-point iteration but to a stochastic gradient descent approach. The algorithmic implementation is based on the updates (34) and the objective function in (36) and is provided in Algorithm 2.

Remark 4: Note that simply minimizing the error $\|\beta e^{\beta r(s,a)} + \gamma V(s';w_t) - V(s;w_t)\|$, $a\sim\pi_{\theta_t}$, for the risk-neutral value function V is not equivalent to the update rule (34), but to simply scaling the initial rewards r_t to $\beta e^{\beta r_t}$.

Algorithm 2 Risk-Sensitive OAC (R-AC)

Input: a differentiable policy $\pi(a|s;\theta)$.
Algorithm parameters:
step-sizes $\alpha>0$, $\bar{\alpha}>0$,
discount factor $\gamma>0$, risk parameter β.
Initialization: $\theta = \theta_0 \in \mathbb{R}^d$, $w = w_0 \in \mathbb{R}^{d'}$.
while (θ, w) **not converged do**
  for $t = 0$ **to** $|\tau| - 1$ **do**
    $a_t\sim\pi(\cdot|s_t;\theta)$, $s_{t+1}\sim p(\cdot|s_t,a_t)$
    $\hat{R}_\beta \leftarrow \beta r_t + \gamma\log\bar{V}_\beta(s_{t+1};w_t)$
    $\theta_{t+1} \leftarrow \theta_t + \alpha\gamma^t\frac{1}{|\beta|}(e^{\hat{R}_\beta} - \bar{V}_\beta(s_t;w_t))\nabla\log\pi(a_t|s_t;\theta)$
    $w_{t+1} \leftarrow w_t + \bar{\alpha}\gamma^t(e^{\hat{R}_\beta} - \bar{V}_\beta(s_t;w_t))\nabla\bar{V}_\beta(s_t;w_t)$
  end for
end while

V. Simulation Results

We compare the proposed risk-sensitive RL algorithms against their risk-neutral counterparts on two baseline RL problems, namely, the inverted pendulum (Cart-Pole) and the underactuated double pendulum (Acrobot) [45].

In addition, we also test our methodology in the planar lunar landing control problem, which is an appropriate testing environment for a risk-sensitive RL algorithm due to the level of uncertainty and disturbances present.

The experiments are designed to investigate the performance and robustness of the proposed risk-sensitive algorithms against model perturbations. We quantify the performance of the algorithms using the mean values of the observed cumulative rewards R during testing in different environments, and their robustness using the variance and the CVaR[1]

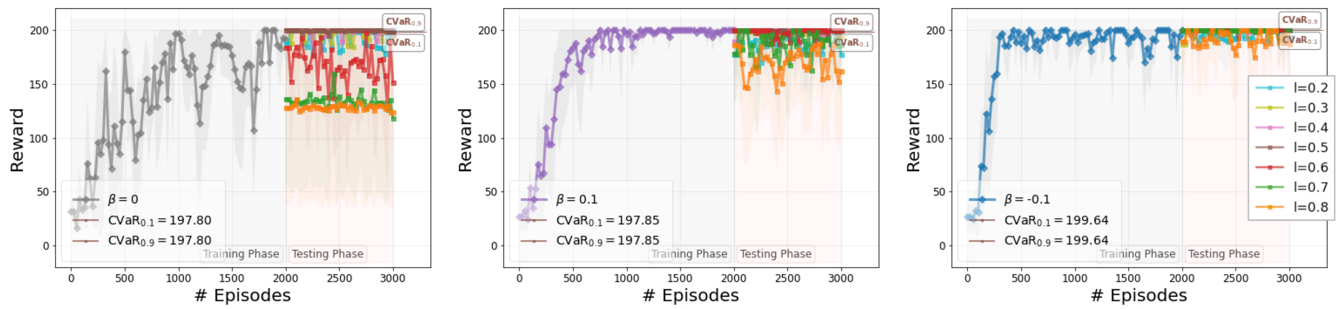
$$\text{CVaR}_p(R) = \mathbb{E}\big[R|R \leq VaR_p(R)\big] \tag{37}$$

where p denotes the confidence interval and the Value at Risk $\text{VaR}_p(R)$ is the p-quantile of the trajectory reward given by

$$\text{VaR}_p(R) = \inf\{r\in\mathbb{R} : P(R\leq r) \ > p\}.$$

In particular, we make use of two p-quantiles for $p \in \{0.1, 0.9\}$ to capture the two tails of the distribution of R (see Section VI-A). In all figures, the average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values are computed over ten independent training and testing runs with different random seeds.

[1]Equation (37) captures the intuition behind the statistical meaning of CVaR and holds if there is no probability atom at $VaR_p(R)$. For a formal definition, the readers are referred to [15] and the references therein.



(a) (b) (c)

Fig. 2. Training and testing behavior of the risk-neutral REINFORCE algorithm against the proposed risk-sensitive R-REINFORCE algorithm (Algorithm 1) for $\beta = -0.1$ and $\beta = +0.1$ in the Cart-Pole problem. (a) Risk-neutral. (b) Risk-seeking. (c) Risk-averse.

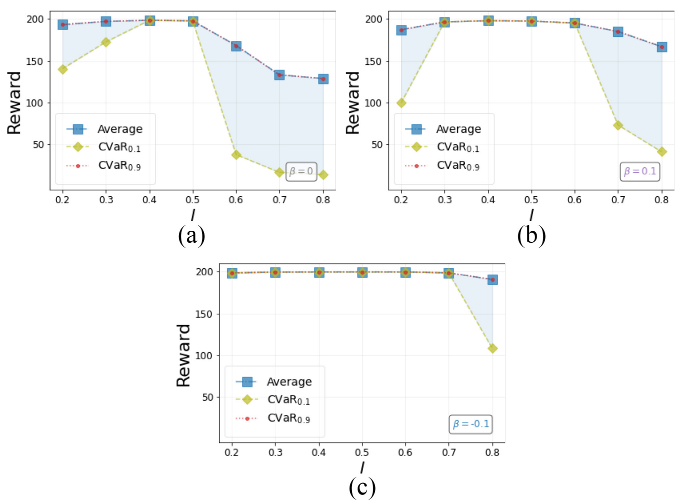


Fig. 3. Robustness of risk-neutral REINFORCE and risk-sensitive R-REINFORCE (Algorithm 1) algorithms in a cart-pole environment with respect to varying pole length. The training environment is modeled with pole length $l = 0.5$. The testing environments have perturbed pole length values of $l \in [0.2, 0.8]$. (a) $\beta = 0$. (b) $\beta = 0.1$. (c) $\beta = -0.1$.

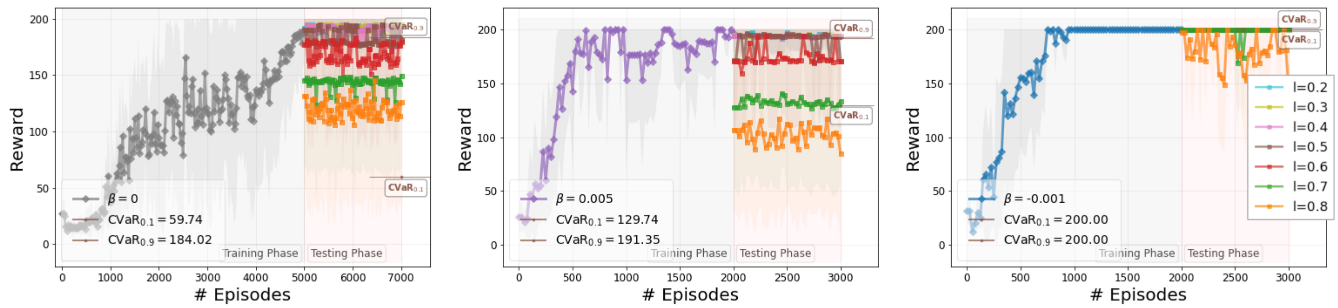
A. Inverted Pendulum (Cart-Pole)

In Fig. 2, we present the training and testing behavior of the risk-neutral REINFORCE algorithm against the proposed risk-sensitive R-REINFORCE algorithm (Algorithm 1) for $\beta = -0.1$ and $\beta = +0.1$ in the Cart-Pole problem. We note that the use of REINFORCE with baseline yielded no statistically significant results compared to risk-neutral REINFORCE. The policy networks of the algorithms are modeled as fully connected artificial neural networks with one hidden layer of only $h = 16$ neurons and a "ReLU" activation function. The objective functions to be optimized are as defined in Section III. We use a discount factor of $\gamma = 0.99$ and the "Adam" optimizer. The best-performing learning rate within the set $\{0.001, 0.003, 0.005, 0.007, 0.01\}$ across all algorithms are selected for the visual inspection of the learning curves. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length is $l = 0.5$ and tested in different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.2, 0.8]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment without perturbations ($l = 0.5$) are computed over ten independent training and testing runs with different random seeds.

We notice that although the mean, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ metrics are not significantly different across the three algorithms, the risk-sensitive algorithms in Fig. 2(c) and (b) converge faster to a near-optimal policy that shows increased robustness with respect to model perturbations. This is further assessed in Fig. 3, where the robustness of the algorithms with respect to model perturbations is quantified by the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for all testing environments. In Fig. 3(a), we observe that the risk-neutral REINFORCE algorithm is performing very well near $l = 0.5$, i.e., where no model perturbations exist, but the performance is quickly deteriorated ($\text{CVaR}_{0.1}$ values decrease) in the presence of perturbations. Fig. 3(b) and (c) shows that the risk-sensitive approaches increase the domain of perturbations where the behavior of the RL agent is stable, with the risk-averse approach ($\beta < 0$) showcasing the best behavior.

In Fig. 4, we present the training and testing behavior of the risk-neutral OAC and risk-sensitive actor–critic (R-AC) (Algorithm 2) algorithms in the cart-pole environment with respect to varying pole length. The policy networks of the algorithms are modeled as fully connected artificial neural networks with one hidden layer of only $h = 16$ neurons and a ReLU activation function. The objective functions to be optimized are as defined in Section IV-A. We use a discount factor of $\gamma = 0.99$ and the Adam optimizer with the best-performing learning rates within the set $\{0.0003, 0.0005, 0.0007, 0.001\}$ across all algorithms. The best-performing learning rate is chosen best on visual inspection of the learning curves. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length is $l = 0.5$ and tested in different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.2, 0.8]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment without perturbations ($l = 0.5$) are computed over ten independent training and testing runs with different random seeds.

We notice that although the mean value performance is not significantly different across the three algorithms, the risk-sensitive algorithms in Fig. 4(c) and (b) converge to a near-optimal policy (in the risk-averse case the performance is optimal) that shows reduced variation across different runs,

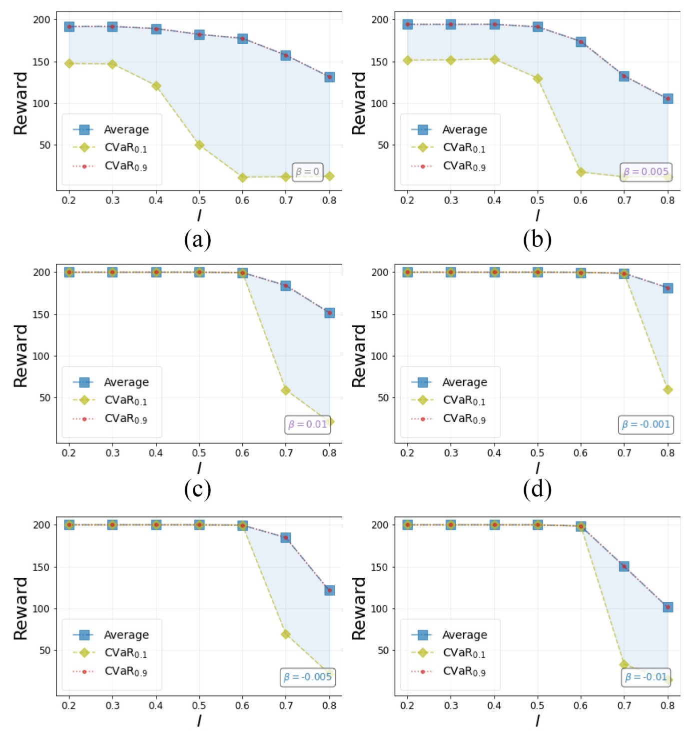


(a) (b) (c)

Fig. 4. Training and testing behavior of the risk-neutral OAC algorithm against the proposed risk-sensitive R-AC algorithm (Algorithm 2) for $\beta = -0.001$ and $\beta = +0.005$ in the Cart-Pole problem. (a) Risk-neutral. (b) Risk-seeking. (c) Risk-averse.

as indicated by the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values calculated for $l = 0.5$ (no model perturbations). Moreover, notice that the risk-neutral algorithm in 4(a) is trained for $n_e = 5000$ episodes to achieve similar performance to the risk-sensitive algorithms. This indicates better sample efficiency for the proposed risk-sensitive algorithms in Algorithm 2. The robustness of the algorithms with respect to model perturbation is further assessed in Fig. 5. Fig. 5(a) shows how the $\text{CVaR}_{0.1}$ values decrease as the pole length increases in the risk-neutral case ($\beta = 0$). Fig. 5(b) and (c) shows that the risk-seeking approaches slightly increase the robustness of the learned policies. However, as shown in Fig. 5(d)–(f), the risk-averse approach ($\beta < 0$) showcases significantly increased robustness with respect to perturbations in the pole length.

Fig. 6 presents a sensitivity analysis of the algorithms with respect to the risk-sensitive parameter $\beta \in [-0.01, 0.01]$. Three testing environments are studied for $l = 0.5$ (no perturbation), $l = 0.3$ (overestimation during training), and $l = 0.7$ (underestimation during training). Negative values for β showcase a more stable behavior across the testing environments. Moreover, notice that $sgn(\beta) < 0$ is roughly adequate for a stable behavior regardless of the numerical value of β, as long as it is close to zero, i.e., no precise estimation of the optimal β is required.



(a) (b) (c) (d) (e) (f)

Fig. 5. Robustness of risk-neutral OAC and risk-sensitive R-AC (Algorithm 2) algorithms in a cart-pole environment with respect to varying pole length. The training environment is modeled with pole length $l = 0.5$. The testing environments have perturbed pole length values of $l \in [0.2, 0.8]$. (a) $\beta = 0$. (b) $\beta = 0.005$. (c) $\beta = 0.01$. (d) $\beta = -0.001$. (e) $\beta = -0.005$. (f) $\beta = -0.01$.

B. Underactuated Double Pendulum (Acrobot)

In Fig. 7, we present the training and testing behavior of the risk-neutral REINFORCE with and without baseline algorithms against the proposed risk-sensitive R-REINFORCE algorithm (Algorithm 1) for $\beta = -0.1$ and $\beta = +0.1$ in the Acrobot problem. The policy networks of the algorithms are modeled as fully connected artificial neural networks with one hidden layer of only $h = 64$ neurons and a ReLU activation function. The objective functions to be optimized are as defined in Section III. We use a discount factor of $\gamma = 0.99$ and the Adam optimizer with the best-performing learning rates within the set $\{0.001, 0.003, 0.005, 0.007, 0.01\}$ across all algorithms. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length of the first link is $l = 1.0$ and tested in different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.7, 1.3]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment without perturbations ($l = 1.0$) are computed over ten independent training and testing runs with different random seeds.

First, we notice [Fig. 7(b)] that risk-neutral REINFORCE without baseline is not able to learn a policy that solves the Acrobot problem. On the remaining algorithms, although the mean performance is not significantly different, the risk-sensitive algorithms in Fig. 7(d) and (c) showcase increased $\text{CVaR}_{0.1}$ values that suggest reduced variation across different runs. The fact that the risk-sensitive approaches perform on par, and slightly better, compared to REINFORCE with baseline, is indicative of the implicit baseline term present in

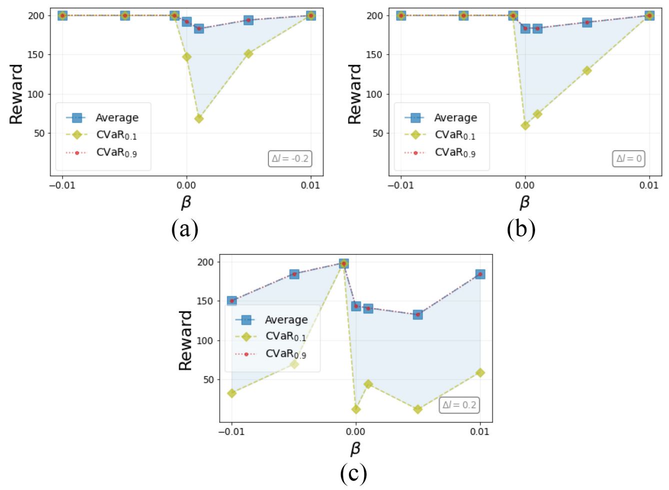


Fig. 6. Sensitivity analysis of the risk-sensitive R-AC algorithm (Algorithm 2) with respect to the risk-sensitive parameter $\beta \in [-0.01, 0.01]$ in the Cart-Pole problem. $\beta = 0$ corresponds to the risk-neutral OAC. The training environment is modeled with pole length $l = 0.5$. (a) $l = 0.3$. (b) $l = 0.5$. (c) $l = 0.7$.

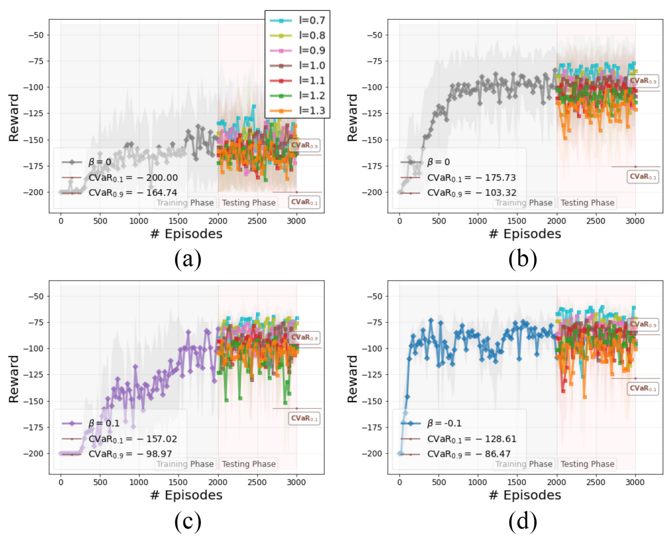


Fig. 7. Training and testing behavior of the risk-neutral REINFORCE and risk-neutral REINFORCE with baseline algorithms against the proposed risk-sensitive R-REINFORCE algorithm (Algorithm 1) for $\beta = -0.1$ and $\beta = +0.1$ in the Acrobot problem. (a) Risk-neutral. (b) Risk-neutral with Baseline. (c) Risk-seeking. (d) Risk-averse.

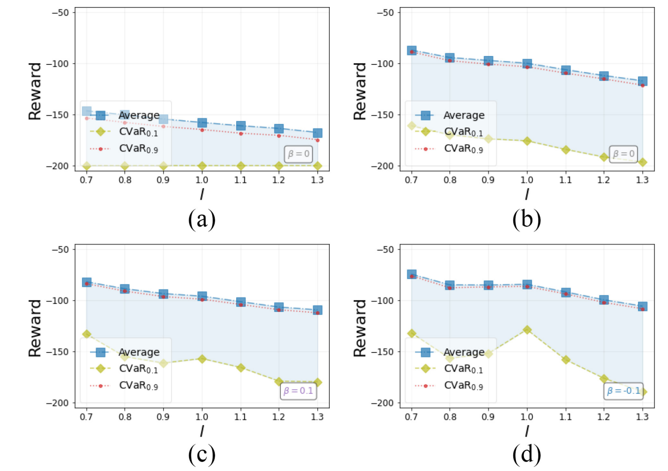


Fig. 8. Robustness of risk-neutral REINFORCE risk-neutral REINFORCE with baseline, and risk-sensitive R-REINFORCE in the Acrobot environment with respect to varying pole length. The training environment is modeled with pole length $l = 1.0$. The testing environments have perturbed pole length values of $l \in [0.7, 1.3]$. (a) $\beta = 0$. (b) $\beta = 0$ with Baseline. (c) $\beta = 0.005$. (d) $\beta = -0.001$.

optimizing for the exponential objective function as explained in Section II-C. The robustness of the algorithms with respect to model perturbation is further assessed in Fig. 8 for all testing environments. Similar to the Cart-Pole problem, Fig. 8 suggests that the risk-sensitive approaches can increase the domain of perturbations where the behavior of the RL agent is more stable, with the risk-averse approach ($\beta < 0$) showcasing the best behavior.

Finally, in Fig. 9, we present the training and testing behavior of the risk-neutral OAC and R-AC (Algorithm 2) algorithms in the Acrobot environment with respect to varying pole length. The policy networks of the algorithms are modeled as fully connected artificial neural networks with one hidden layer of only $h = 64$ neurons and a ReLU activation function. The objective functions to be optimized are as defined in Section IV-A. We use a discount factor of $\gamma = 0.99$ and the Adam optimizer with the best-performing learning rates within the set $\{0.0003, 0.0005, 0.0007, 0.001\}$ across all algorithms. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length of the first link is $l = 1.0$ and tested in different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.7, 1.3]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment without perturbations ($l = 1.0$) are computed over ten independent training and testing runs with different random seeds.

Similar to the Cart-Pole case, we notice that although the mean value performance is not significantly different across the three algorithms, the risk-sensitive algorithms in Fig. 9(c) and (b) converge to a near-optimal policy that shows reduced variation across different runs, as indicated by the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values calculated for $l = 1.0$ (no model perturbations). The robustness of the algorithms with respect to model perturbation is further assessed in Fig. 10. Fig. 10(a) shows how the $\text{CVaR}_{0.1}$ values decrease as the pole length increases in the risk-neutral case ($\beta = 0$). Fig. 10(c) shows that the risk-averse approach can increase the robustness of the learned policies for small perturbations.

C. Planar Rocket Trajectory Control (Lunar Lander)

The lunar lander problem is a simplified rocket trajectory control problem in 2-D [45]. The rocket needs to land safely in a predefined region using a version of a bang-bang control under wind and turbulence disturbances.

In Fig. 11, we present the training and testing behavior of the risk-neutral OAC and R-AC (Algorithm 2) algorithms in the Lunar Lander Gymnasium environment [45] with respect to varying width strength $w \in \{5.0, 10.0, 15.0\}$. The policy networks of the algorithms are modeled as fully connected

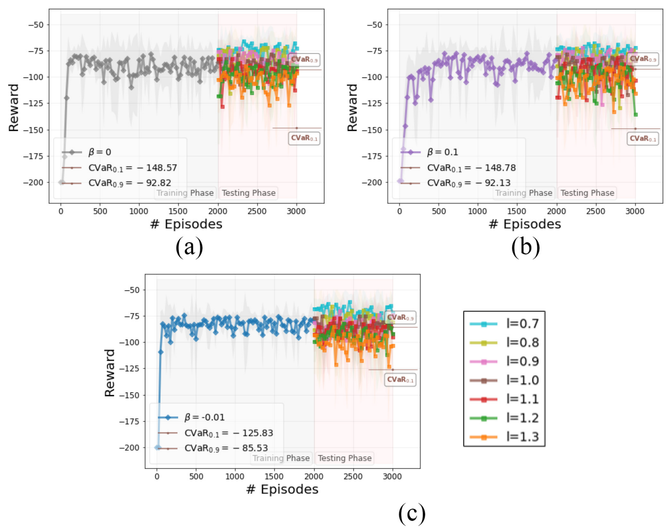


Fig. 9. Training and testing behavior of the risk-neutral OAC algorithm against the proposed risk-sensitive R-AC algorithm (Algorithm 2) for $\beta = -0.01$ and $\beta = +0.1$ in the Acrobot problem. (a) Risk-neutral. (b) Risk-seeking. (c) Risk-averse.

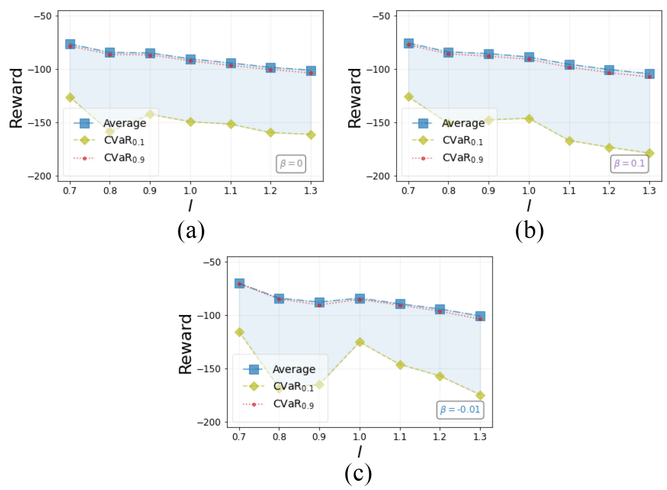


Fig. 10. Robustness of risk-neutral OAC and risk-sensitive R-AC (Algorithm 2) algorithms in the Acrobot environment with respect to varying pole length. The training environment is modeled with pole length $l = 1.0$. The testing environments have perturbed pole length values of $l \in [0.7, 1.3]$. (a) $\beta = 0$. (b) $\beta = 0.005$. (c) $\beta = -0.001$.

artificial neural networks with one hidden layer of only $h = 16$ neurons and a ReLU activation function. The objective functions to be optimized are as defined in Section IV-A. We use a discount factor of $\gamma = 0.99$ and the Adam optimizer with the best-performing learning rates within the set $\{0.0003, 0.0005, 0.0007, 0.001\}$ across all algorithms. The algorithms are trained for $n_e = 1750$ episodes in a training environment with wind strength $w = 5.0$ and tested in different testing environments for $n_e = 750$ testing runs with wind strength $w \in [10.0, 15.0]$. The average reward for the different testing environments are computed over 10 independent training and testing runs.

In this case, the mean value performance is significantly different across the three algorithms. The risk-averse algorithm in Fig. 11(c) is able to learn a near-optimal policy (average

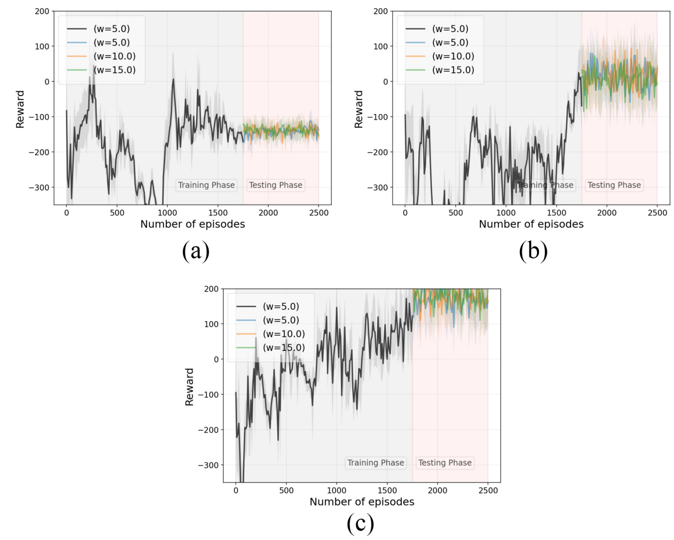


Fig. 11. Training and testing behavior of the risk-neutral OAC algorithm against the proposed risk-sensitive R-AC algorithm (Algorithm 2) for $\beta = -0.01$ and $\beta = +0.01$ in the Lunar Lander problem. (a) Risk-neutral. (b) Risk-seeking. (c) Risk-averse.

mean value of 200 solves the problem), while the risk-neutral algorithm [Fig. 11(a)] does not manage to learn a policy at all. The risk-seeking algorithm in Fig. 9(b) performs better than the risk-neutral algorithm, but does not provide an acceptable controller either.

VI. Discussion

The experimental results are consistent with the analysis presented in Section II and suggest that the use of exponential criteria accelerate the learning process, leading in increased sample efficiency, while learning policies with increased robustness with respect to environmental and model perturbations, similar to entropy- and KL-regularized RL methods (Remark 1). However, in contrast to alternative risk-sensitive RL methods that are based on constrained optimization or optimize for the statistical measures (e.g., CVaR values) [10], [11], [12], [15], [16], [17], [18], [19], [20], [21], [22], Algorithm 1 inherits the computational properties of the REINFORCE algorithm, while Algorithm 2 has similar computational complexity with traditional actor–critic methods. The updates of the proposed algorithms have been designed intentionally similar to traditional RL algorithms to inherit their complexity and convergence properties.

A. On the Sign and Values of the Parameter β

The sign of the risk parameter β determines the optimization problem that is being solved according to (4). Thus, in the simulated experiments of Sections V-A and V-B, it is expected that the risk-averse approach ($\beta < 0$) reduces the variance (and CVaR_p values for $p > 0.5$) of the distribution of the total reward. In addition, the risk-seeking approach ($\beta > 0$) does not guarantee, but can also help reduce the variance (and CVaR_p values for $p < 0.5$) of the distribution of the total reward. Such a reduction can be indicative of a better suited learning behavior for the RL policies estimated by the proposed algorithm compared to the risk-neutral RL methods. Since in the risk-seeking (or "optimistic") case of $\beta > 0$, emphasis is given on the right tail of the distribution of the total reward, convergence to policies with high average return can be accelerated under certain values of the hyperparameters of the system and certain sequences of random exploratory actions. The hyperparameters that can affect this behavior include the learning rate of the actor and critic models, and the random sequences that generate the exploratory actions. The selection of the policies that yield the best performance among different runs (e.g., runs with different learning rates) is often adopted. In this case, the risk-seeking approach can also lead to better policies in terms of reduced variance.

We note that in the experiments of Sections V-A and V-B, we do not optimize for the risk-sensitive hyperparameter β. Rather, we include a comparison and a sensitivity analysis for different values of β close to zero. Further analysis and experimentation regarding the choice of the value of $\beta \in (0, \infty)$ is beyond the scope of this article and will be addressed by the authors elsewhere. It is worth mentioning, that the choice of the hyperparameter β can have an impact on the implementation of the proposed algorithms. In particular, large values for β can create numerical issues by forcing the updates (24) and (27) to handle small/large values near the precision limits of the machine. To avoid this issue, it is suggested to chose a value as close to zero as possible, while large enough to have an effect to the risk-sensitivity of the algorithm.

VII. Conclusion

We formulated a risk-sensitive RL approach as an optimization problem with respect to a modified objective based on exponential criteria. In particular, we study a model-free risk-sensitive variation of the widely used Monte Carlo PG algorithm, and introduce a novel risk-sensitive OAC algorithm based on solving a multiplicative Bellman equation using stochastic approximation updates. Analytical results suggest that the use of exponential criteria generalizes commonly used ad-hoc regularization approaches, improves sample efficiency, and introduces robustness with respect to perturbations in the model parameters and the environment. The implementation, performance, and robustness properties of the proposed methods are evaluated in simulated experiments, and suggest suitability for real-life applications where learning in simulation is followed by transferring the learned policies to real agents in noisy environments.

Appendix A
Risk-Sensitive Policy Gradient Update Rule

In this section, we provide a risk-sensitive version of the policy gradient theorem [35] using exponential criteria, which is used to derive the update rule for the Risk-Sensitive REINFORCE algorithm in (24). The exponential objective can be written as an integral (summation for finite state and action spaces) over all possible trajectories, i.e.,

$$\begin{aligned}\nabla_\theta J_\beta(\theta) &= \nabla \frac{1}{\beta}\int \rho_\theta(\tau)\exp\{\beta R(\tau)\}d\tau \\ &= \frac{1}{\beta}\int \rho_\theta(\tau)\frac{\nabla\rho_\theta(\tau)}{\rho_\theta(\tau)}\exp\{\beta R(\tau)\}d\tau \\ &= \frac{1}{\beta}\int \rho_\theta(\tau)\nabla\log\rho_\theta(\tau)\exp\{\beta R(\tau)\}d\tau \\ &= \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\nabla\log\rho_\theta(\tau)\exp\{\beta R(\tau)\}\Big]. \end{aligned} \tag{38}$$

Using the "log-trick" [1], the gradient of $J_\beta(\theta)$ with respect to the policy parameter θ can be obtained as follows:

$$\begin{aligned}\nabla_\theta J_\beta(\theta) &= \nabla \frac{1}{\beta}\int \rho_\theta(\tau)\exp\{\beta R(\tau)\}d\tau \\ &= \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\nabla\log\rho_\theta(\tau)\exp\{\beta R(\tau)\}\Big]. \end{aligned} \tag{39}$$

Recall that $\rho_\theta(\tau) = p_0\prod_{t=0}^{|\tau|-1}\pi(a_t|s_t;\theta)p(s_{t+1}|s_t,a_t)$. Then, by first taking the logarithm and then the gradient of both sides, we get

$$\nabla\log\rho_\theta(\tau) = \sum_{t=0}^{|\tau|-1}\nabla\log\pi(a_t|s_t;\theta). \tag{40}$$

For brevity, we use $\pi_t(\theta) := \pi(a_t|s_t;\theta)$. Thus, by substituting (40) in (38), we get

$$\nabla J_\theta(\theta) = \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\nabla\log\pi_t(\theta)\exp\{\beta R(\tau)\}\Big]. \tag{41}$$

Recall that $R(\tau) = \sum_{t=0}^{|\tau|-1}\gamma^t r(s_t,a_t)$. Using this fact and the property of exponential, we have

$$\begin{aligned}\nabla J_\theta(\theta) = \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\ \nabla\log\pi_t(\theta)\exp\{\beta\sum_{t'=0}^{t-1}\gamma^{t'}r(s_t,a_t)\} \\ \exp\{\beta\sum_{t'=t}^{|\tau|-1}\gamma^{t'}r(s_{t'},a_{t'})\}\Big]. \end{aligned} \tag{42}$$

By using the temporal structure of the problem and causality, it can be argued that the rewards prior to time t are not dependent on the actions that the policy will take in a future state s_t, that is, $\sum_{t'=0}^{t-1}\gamma^{t'}r_t(s_{t'},a_{t'})$ is independent of $\nabla\log\pi(a_t|s_t;\theta)$. Thus, by using the independence property, we have

$$\begin{aligned}\nabla J_\theta(\theta) = \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\exp\{\beta\sum_{t'=0}^{t-1}\gamma^{t'}r(s_{t'},a_{t'})\}\Big] \\ \cdot\mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\nabla\log\pi_t(\theta)\exp\{\beta\sum_{t'=t}^{|\tau|-1}\gamma^{t'}r(s_{t'},a_{t'})\}\Big]. \end{aligned} \tag{43}$$

Note that the first expectation is a constant, therefore

$$\nabla J_\theta(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\frac{1}{\beta}e^{\beta R_t}\nabla\log\pi_t(\theta)\Big] \tag{44}$$

where $R_t := \sum_{t'=t}^{|\tau|-1}\gamma^{t'-t}r(s_{t'},a_{t'})$.

As a final remark, notice that from (43), we can see that the first term on the right-hand side of the equation provides an inherent way of adjusting the step size, effectively making the constant step size adaptive.

A. Convergence Analysis

In this section, we show that the parameter vector θ updated by the risk-sensitive REINFORCE algorithm in (24) converges to the optimal parameter vector θ^* in expectation, for sufficiently small values of the risk-parameter β. First note the following identity:

$$\begin{aligned}\|\theta_{t+1} &- \theta^*\|^2 - \|\theta_t - \theta^*\|^2\\ &= \|\theta_{t+1} - \theta_t + \theta_t - \theta^*\|^2 - \|\theta_t - \theta^*\|^2\\ &= \|\theta_{t+1} - \theta_t\|^2 - 2(\theta_{t+1} - \theta_t)\cdot\big(\theta_t - \theta^*\big).\end{aligned}$$

Using the R-REINFORCE update rule in (24), i.e.,

$$\theta_{t+1} = \theta_t + \frac{\eta}{\beta}e^{\beta R_t^+}\nabla\log\pi_{\theta_t}(a_t|s_t)$$

we get

$$\begin{aligned}\|\theta_{t+1} - \theta^*\|^2 - \|\theta_t &- \theta^*\|^2 = \left(\frac{\eta}{\beta}e^{\beta R_t^+}\right)^2\|\nabla\log\pi_{\theta_t}(a_t|s_t)\|^2\\ &- 2\frac{\eta}{\beta}e^{\beta R_t^+}\nabla\log\pi_{\theta_t}(a_t|s_t)\cdot\big(\theta_t - \theta^*\big).\end{aligned}$$

By taking the conditional expectation with filtration $\mathcal{F}_t$ from both sides of the equation, we have

$$\begin{aligned}\mathbb{E}\Big[\|\theta_{t+1} - \theta^*\|^2 \mid \mathcal{F}_t\Big] &= \|\theta_t - \theta^*\|^2\\ &+\left(\frac{\eta}{\beta}\right)^2\mathbb{E}\Big[e^{2\beta R_t^+}\|\nabla\log\pi_{\theta_t}(a_t|s_t)\|^2 \mid \mathcal{F}_t\Big]\\ &-2\frac{\eta}{\beta}e^{-\beta R_t^-}\mathbb{E}\Big[e^{\beta R}\nabla\log\pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t\Big]\cdot\big(\theta_t - \theta^*\big)\\ = \|\theta_t - \theta^*\|^2 &+ \left(\frac{\eta}{\beta}\right)^2\mathbb{E}\Big[e^{2\beta R_t^+}\|\nabla\log\pi_{\theta_t}(a_t|s_t)\|^2 \mid \mathcal{F}_t\Big]\\ &-2\eta e^{-\beta R_t^-}\nabla J_\gamma(\theta_t)\cdot\big(\theta_t - \theta^*\big).\end{aligned}$$

The first line follows from the conditioning on the filtration $\mathcal{F}_t$. The second line follows from the fact that $\nabla J_\gamma(\theta_t) = \mathbb{E}[(1/\beta)e^{\beta R}\nabla\log\pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t]$. It should be noted that since $\theta^* = \text{argmax}_\theta J_\gamma(\theta)$, it follows that $\nabla J_\gamma(\theta_t)\cdot(\theta_t - \theta^*) > 0$. Finally, it follows that θ_t converges to θ^*, as long as the following condition holds:

$$\begin{aligned}&\left(\frac{\eta}{\beta}\right)^2\mathbb{E}\Big[e^{2\beta R_t^+}\|\nabla\log\pi_{\theta_t}(a_t|s_t)\|^2 \mid \mathcal{F}_t\Big]\\ &\quad- 2\eta e^{-\beta R_t^+}\nabla J_\gamma(\theta_t)\cdot\big(\theta_t - \theta^*\big) < 0.\end{aligned}$$

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[2] 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.
[3] T. Osogami, “Robustness and risk-sensitivity in Markov decision processes,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–9.
[4] E. Noorani and J. S. Baras, “Risk-sensitive REINFORCE: A monte carlo policy gradient algorithm for exponential performance criteria,” in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 1522–1527.
[5] L. P. Hansen and T. J. Sargent, “Robustness,” in *Robustness*. Princeton, NJ, USA: Princeton Univ. Press, 2011.
[6] M. R. James and J. S. Baras, “Robust H/sub /spl infin// output feedback control for nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 40, no. 6, pp. 1007–1017, Jun. 1995.
[7] J. S. Baras and M. R. James, “Robust and risk-sensitive output feedback control for finite state machines and hidden Markov models,” *J. Math. Syst., Estim., Control*, vol. 7, no. 3, pp. 371–374, 1997.
[8] J. S. Baras and N. S. Patel, “Robust control of set-valued discrete-time dynamical systems,” *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 61–75, Jan. 1998.
[9] H. E. Scarf, *A Min-Max Solution of an Inventory Problem*. Santa Monica, CA, USA: Rand Co., 1957.
[10] S. Paternain, L. Chamon, M. Calvo-Fullana, and A. Ribeiro, “Constrained reinforcement learning has zero duality gap,” in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
[11] E. Delage and S. Mannor, “Percentile optimization for Markov decision processes with parameter uncertainty,” *Oper. Res.*, vol. 58, no. 1, pp. 203–213, 2010.
[12] M. A. Abdullah et al., “Wasserstein robust reinforcement learning,” in *Proc. ICLR*, 2019, pp. 1–17.
[13] A. Tamar, H. Xu, and S. Mannor, “Scaling up robust MDPs by reinforcement learning,” 2013, *arXiv:1306.6189*.
[14] M. Fu et al., “Risk-sensitive reinforcement learning via policy gradient search,” 2018, *arXiv:1810.09126*.
[15] Y. Chow and M. Ghavamzadeh, “Algorithms for CVaR optimization in MDPs,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3509–3517.
[16] L. A. Prashanth, “Policy gradients for CVaR-constrained MDPs,” in *Algorithmic Learning Theory*, P. Auer, A. Clark, T. Zeugmann, and S. Zilles, Eds., Cham, Switzerland: Springer Int. Publ., 2014, pp. 155–169.
[17] A. Tamar, “Risk-sensitive and efficient reinforcement learning algorithms,” Ph.D. dissertation, Dept. Elect. Eng., Technion-Israel Inst. Technol., Haifa, Israel, 2015.
[18] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, “Robust adversarial reinforcement learning,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2817–2826.
[19] X. Pan, D. Seita, Y. Gao, and J. Canny, “Risk averse robust adversarial reinforcement learning,” in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 8522–8528.
[20] A. T. Dotan Di Castro and S. Mannor, “Policy gradients with variance related risk criteria,” in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, U.K., 2012.
[21] B. Liu, J. Liu, and K. Xiao, “R2PG: Risk-sensitive and reliable policy gradient,” in *Proc. Workshops 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 682–686.
[22] B. Eysenbach and S. Levine, “Maximum entropy RL (provably) solves some robust RL problems,” 2021, *arXiv:2103.06257*.
[23] V. S. Borkar, “Learning algorithms for risk-sensitive control,” in *Proc. 19th Int. Symp. Math. Theory Netw. Syst.*, 2010, pp. 1327–1332.
[24] D. Nass, B. Belousov, and J. Peters, “Entropic risk measure in policy search,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 1101–1106.
[25] E. Noorani and J. S. Baras, “Risk-sensitive reinforcement learning and robust learning for control,” in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 2976–2981.
[26] Z. Shang, R. Li, C. Zheng, H. Li, and Y. Cui, “Relative entropy regularized sample-efficient reinforcement learning with continuous actions,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 475–485, Jan. 2025.
[27] M. G. Azar, V. Gómez, and H. J. Kappen, “Dynamic policy programming,” *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 3207–3245, 2012.
[28] Y. Cui, T. Matsubara, and K. Sugimoto, “Kernel dynamic policy programming: Applicable reinforcement learning to robot systems with high dimensional states,” *Neural Netw.*, vol. 94, pp. 13–23, Oct. 2017.
[29] M. R. James, J. S. Baras, and R. J. Elliott, “Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems,” *IEEE Trans. Automat. control*, vol. 39, no. 4, pp. 780–792, Apr. 1994.
[30] E. Noorani, C. N. Mavridis, and J. S. Baras, “Exponential TD learning: A risk-sensitive actor-critic reinforcement learning algorithm,” in *Proc. Am. Control Conf. (ACC)*, 2023, pp. 4104–4109.
[31] E. Noorani and J. S. Baras, “Embracing risk in reinforcement learning: The connection between risk-sensitive exponential and distributionally robust criteria,” in *Proc. Am. Control Conf. (ACC)*, 2022, pp. 2703–2708.
[32] H. Föllmer and A. Schied, “Convex measures of risk and trading constraints,” *Finan. Stochast.*, vol. 6, no. 4, pp. 429–447, 2002.

[33] C. Mavridis, E. Noorani, and J. S. Baras, “Risk sensitivity and entropy regularization in prototype-based learning,” in *Proc. 30th Mediterr. Conf. Control Autom. (MED)*, 2022, pp. 194–199.
[34] A. Galashov et al., “Information asymmetry in KL-regularized RL,” 2019, *arXiv:1905.01240*.
[35] R. S. Sutton, D. McAllester, S. P. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.
[36] R. J. Williams and J. Peng, “Function optimization using Connectionist reinforcement learning algorithms,” *Connect. Sci.*, vol. 3, no. 3, pp. 241–268, 1991.
[37] V. Konda and J. Tsitsiklis, “Actor-critic algorithms,” in *Proc. Adv. Neural Inf. Process. Syst.*, 199, pp. 1008–1014.
[38] L. Weaver and N. Tao, “The optimal reward baseline for gradient-based reinforcement learning,” in *Proc. 17th Conf. Uncertainty Artif. Intell.*, 2013, pp. 538–545.
[39] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, vol. 48. Singapore: Springer, 2009.
[40] C. N. Mavridis and J. S. Baras, “Vector Quantization for adaptive state aggregation in reinforcement learning,” in *Proc. Am. Control Conf. (ACC)*, 2021, pp. 2187–2192.
[41] C. N. Mavridis, N. Suriyarachchi, and J. S. Baras, “Maximum-entropy progressive state aggregation for reinforcement learning,” in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 5144–5149.
[42] V. S. Borkar, “Q-learning for risk-sensitive control,” *Math. Oper. Res.*, vol. 27, no. 2, pp. 294–311, 2002.
[43] Y. Fei, Z. Yang, Y. Chen, Z. Wang, and Q. Xie, “Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret,” in *Proc. 34th Conf. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 22384–22395.
[44] Y. Fei, Z. Yang, Y. Chen, and Z. Wang, “Exponential bellman equation and improved regret bounds for risk-sensitive reinforcement learning,” in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20436–20446.
[45] M. Towers et al., “Gymnasium: A standard interface for reinforcement learning environments,” 2024, *arXiv:2407.17032*.

Erfaun Noorani received the B.Sc. degree in electrical engineering from Drexel University, Philadelphia, PA, USA, in 2018, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland at College Park, College Park, MD, USA, in 2021 and 2023, respectively.

He was a Clark Doctoral Fellow with the University of Maryland at College Park. He is currently a Technical Staff with MIT Lincoln Laboratory, Lexington, MA, USA. Prior to this role, he was a Postdoctoral Associate within the Institute for Systems Research, University of Maryland at College Park. His research focuses on robust and risk-sensitive reinforcement learning.

Christos N. Mavridis (Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2017, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland at College Park, College Park, MD, USA, in 2021.

He is currently a Postdoctoral Fellow with KTH Royal Institute of Technology, Stockholm, Sweden, and has been affiliated as a Research Scientist with the Institute for Systems Research, University of Maryland at College Park; Nokia Bell Labs, Murray Hill, NJ, USA; Xerox Palo Alto Research Center, Palo Alto, CA, USA; and Ericsson AB, Stockholm. His research interests include hybrid systems and control theory, stochastic optimization, and learning theory.

Dr. Mavridis has received the A. James Clark School of Engineering Distinguished Graduate Fellowship and the Ann G. Wylie Dissertation Fellowship in 2017 and 2021, respectively. He has been a finalist in the Qualcomm Innovation Fellowship, San Diego, CA, USA, 2018, and he has received the Best Student Paper Award at the IEEE International Conference on Intelligent Transportation Systems 2021. He is a member of the IEEE/CSS Technical Committee on Security and Privacy.

John S. Baras (Life Fellow, IEEE) received the Diploma degree in electrical and mechanical engineering from the National Technical University of Athens, Athens, Greece, in 1970, and the M.S. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, MA, USA, in 1971 and 1973, respectively.

He is a Distinguished University Professor and holds the Lockheed Martin Chair in Systems Engineering with the Department of Electrical and Computer Engineering and the Institute for Systems Research (ISR), University of Maryland at College Park, College Park, MD, USA. From 1985 to 1991, he was the Founding Director of ISR. Since 1992, he has been the Director of the Maryland Center for Hybrid Networks (HYNET), which he co-founded. His research interests include systems and control, optimization, communication networks, applied mathematics, machine learning, artificial intelligence, signal processing, robotics, computing systems, security, trust, systems biology, healthcare systems, and model-based systems engineering.

Dr. Baras’s honors include the 1980 George Axelby Award from the IEEE Control Systems Society, the 2006 Leonard Abraham Prize from the IEEE Communications Society, the 2017 IEEE Simon Ramo Medal, the 2017 AACC Richard E. Bellman Control Heritage Award, and the 2018 AIAA Aerospace Communications Award. In 2016, he was inducted in the A. J. Clark School of Engineering Innovation Hall of Fame. In 2018, he was awarded a Doctorate Honoris Causa by his alma mater, National Technical University of Athens. He is a Fellow of SIAM, AAAS, NAI, IFAC, AMS, and AIAA, a member of the National Academy of Inventors, and a foreign member of the Royal Swedish Academy of Engineering Sciences.