On Surprising Effects of Risk-Aware Domain Randomization for Contact-Rich Sampling-based Predictive Control

Sergio A. Esteban[1], Junheng Li[1], Vince Kurtz[2], and Aaron D. Ames[1]

***Abstract*— Domain randomization (DR) is widely used in policy learning to improve robustness to modeling error, but remains underexplored in contact-rich sampling-based predictive control (SPC), where rollout quality is highly sensitive to uncertainty. In this work, we take the first step by studying risk-aware DR in predictive sampling on a simple yet representative Push-T task, comparing average, optimistic, and pessimistic rollout aggregations under randomized model instances. Our initial results suggest that DR affects not only robustness to model error, but also the effective cost landscape seen by the sampling-based optimizer, by reshaping the basin of attraction around contact-producing actions. This opens up potential for exploring better grounded risk-aware contact-rich SPC under model uncertainty. Video: https://youtu.be/f1F0ALXxhSM**

I. Introduction

Domain Randomization (DR) has become a standard and powerful tool in reinforcement learning (RL) for robotics, particularly in contact-rich settings where accurate modeling of friction, inertial parameters, and compliance remains difficult [1]. In contrast, the use of DR in trajectory optimization and model predictive control (MPC) is *far less* explored, especially for contact-rich problems. This gap is notable because the same modeling challenges that motivate DR in RL also arise in contact-rich trajectory optimization and MPC, where performance can be highly sensitive to physical parameters and contact outcomes.

Recent advances make this problem timely to study: massively parallel GPU simulators enable efficient evaluation of thousands of rollouts under different model realizations [2], [3], while sampling-based methods, such as Predictive Sampling [4], Model Predictive Path Integral (MPPI) [5], and Cross Entropy Method (CEM)-style approaches [6], have become increasingly attractive for nonlinear contact-rich control because they avoid some of the analytic difficulties of the gradient-based optimization through non-smooth contact and complementarity conditions [7]. Together, these tools enable a simple paradigm: evaluate each candidate control sequence across a randomized ensemble of domains and rank it according to a chosen notion of risk.

In this work, we present a preliminary study of **risk-aware domain randomization for contact-rich sampling-based predictive control (SPC)**. In particular, our preliminary demonstration focuses on the Push-T task. For each candidate input sequence, we evaluate rollout cost across multiple randomized model instances and aggregate the results using three risk operators:

1) **Average**: *mean performance* across domains,
2) **Pessimistic**: *worst-case* rollout cost (Risk-averse),
3) **Optimistic**: *best-case* rollout cost (Risk-seeking).

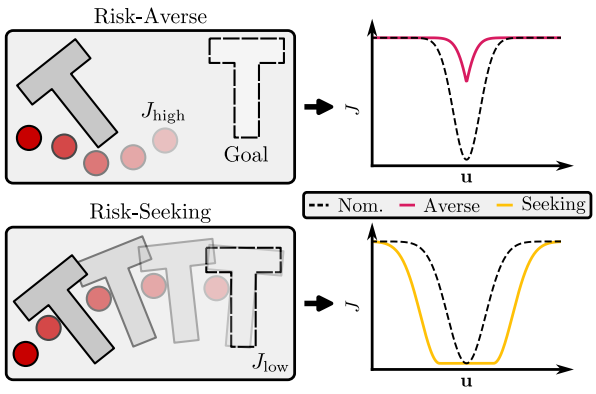


Fig. 1: In contact-rich control settings, successful contact may correspond to a narrow low-cost region. We hypothesize that while risk-averse aggregation can suppress this region and shrink its basin of attraction, risk-seeking aggregation can enlarge the basin around promising contact-producing actions.

The main contribution of this work is a first step toward a systematic study of risk-aware DR in contact-rich sampling-based predictive control through a unified comparison of proposed risk operators. Our motivation is that the intuitive ranking of these strategies does not always hold: although pessimistic aggregation may seem most robust under model mismatch, our initial results suggest that optimistic aggregation can sometimes perform better by promoting high-variance but contact-producing candidates. We hypothesize that this occurs because DR affects not only robustness to modeling error, but also the effective search landscape itself. In contact-rich tasks, pessimistic aggregation can suppress narrow low-cost basins, whereas optimistic aggregation can make them easier for a sampling-based optimizer to discover.

II. Related Work

Domain randomization is most established in robot learning for sim-to-real transfer, where visual or dynamics variation during training improves robustness to the reality gap [1], [8]. Subsequent work has made DR more adaptive [9], [10], but the current literature remains focused on policy learning in general [11].

In contrast, this work is closely related to the growing body of sampling-based predictive control methods for nonlinear planning in robotics, including MPPI [12], Predictive Sampling [4], and recent GPU-parallel frameworks such

[1]The authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, USA. {sesteban, junhengl, ames}@caltech.edu
[2]The author is with the School of Computing, DePaul University, Chicago, IL, USA. vkurtz1@depaul.edu
*This work was supported by Technology Innovation Institute (TII).

as Hydrax [13]. The closest prior works are robust and risk-aware MPPI variants [14]–[18], which mainly study robustness and safety in domains such as rally car racing or obstacle avoidance. In contrast, our focus is on contact-rich predictive control, and specifically on how risk-aware aggregation across randomized domains reshapes the search landscape of a sampling-based optimizer.

More broadly, this paper relates to robust trajectory optimization and motion planning under uncertainty, including Monte Carlo [19], risk-bounded [20], scenario-based [21], and risk-averse planning methods [22]. It is also connected to planning through contact, where recent work has used smoothing or contact-implicit formulations to address the nonsmooth, highly nonconvex structure induced by impacts and friction [23]–[25]. Our perspective is complementary to both lines of work: rather than estimating collision risk or smoothing contact gradients, we study how risk-aware DR reshapes the cost landscape in contact-rich SPC (Fig. 1).

III. Sampling-based Predictive Control

We consider a finite-horizon optimal control problem at planning time t, with current state estimate $\mathbf{x}_0 = \hat{\mathbf{x}}(t)$. Let

$$\mathbf{U} := \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}\} \tag{1}$$

denote a control tape over a horizon of length N. The nominal planning problem can be written compactly as

$$\min_{\mathbf{U}} \quad J(\mathbf{U}; \mathbf{x}_0), \tag{2}$$

where $J(\mathbf{U}; \mathbf{x}_0)$ denotes the rollout cost over the horizon under the dynamics and task objective. For contact-rich systems, this problem is often highly nonconvex and nonsmooth, making gradient-based optimization difficult. SPC instead evaluates candidate control tapes via forward rollouts and updates the nominal plan according to their costs [26].

At each MPC step, we draw K samples $\mathbf{U}^{(k)}$ from some proposal distribution, typically a Gaussian centered around the previous nominal control tape. We then simulate a rollout for each sample, yielding the associated costs

$$J^{(k)} = J\Big(\mathbf{U}^{(k)}; \mathbf{x}_0\Big), \qquad k = 1 \dots K \tag{3}$$

These costs are used to update the nominal control tape $\mathbf{U}$. In general, SPC updates can be written as

$$\mathbf{U} \leftarrow \mathbf{U} + \frac{\sum_{k=1}^{K} g\big(J^{(k)}\big)\,\big(\mathbf{U}^{(k)} - \mathbf{U}\big)}{\sum_{k=1}^{K} g\big(J^{(k)}\big)}, \tag{4}$$

where $g : \mathbb{R} \to \mathbb{R}_+$ is a nonnegative weighting function. Different choices of $g(\cdot)$ recover different SPC algorithms [26]. In this work, we focus on predictive sampling [4], the simplest SPC algorithm; predictive sampling merely sets $\mathbf{U}$ as the lowest-cost sample.

Having updated the control tape $\mathbf{U}$, we apply the first action $\mathbf{u}_0$ and proceed in receding-horizon fashion.

IV. Risk-Aware Domain Randomization

In practice, the model we use to simulate rollouts is never perfect. Instead, we aim to improve robustness via DR. In particular, we assume that some model parameters

Algorithm 1 Risk-Aware Sampling Predictive Control

Initialize $\mathbf{U} = \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_N\}$
$\{\boldsymbol{\theta}^{(r)}\}_{r=1}^{R} \sim \mathcal{D}$ ▷ sample model randomizations
while planning **do**
  $\mathbf{x}_0 \leftarrow \hat{\mathbf{x}}(t)$ ▷ update state
  for $k = 1$ to K **do**
    $\mathbf{U}^{(k)} \sim \mathcal{N}(\mathbf{U}, \sigma^2)$ ▷ sample control tapes
    for $r = 1$ to R **do**
      $J^{(k,r)} \leftarrow J\big(\mathbf{U}^{(k)}; \mathbf{x}_0, \boldsymbol{\theta}^{(r)}\big)$ ▷ rollouts
    end for
    $\bar{J}^{(k)} \leftarrow \mathrm{Risk}\big(J^{(k,1)}, J^{(k,2)}, \dots, J^{(k,R)}\big)$
  end for
  $\mathbf{U} \leftarrow \mathbf{U} + \frac{\sum_{k=1}^{K} g(\bar{J}^{(k)})(\mathbf{U}^{(k)} - \mathbf{U})}{\sum_{k=1}^{K} g(\bar{J}^{(k)})}$ ▷ SPC step (4)
  Apply $\mathbf{u}_0$
end while

$\boldsymbol{\theta}$ (friction coefficients, body masses, etc.) are drawn from some distribution $\mathcal{D}$. In this setting, the cost

$$J(\mathbf{U}; \mathbf{x}_0, \boldsymbol{\theta}) \tag{5}$$

is determined by the value of these parameters as well as the initial condition $\mathbf{x}_0$.

A. Sampled Control Inputs and Domain Randomization

To apply DR to SPC, we randomly sample R sets of parameters

$$\boldsymbol{\theta}^{(r)} \sim \mathcal{D} \quad r = 1 \dots R, \tag{6}$$

resulting in R "domains", each with different dynamics.

We roll out each control tape $\mathbf{U}^{(k)}$ in each domain, performing $K \times R$ total rollouts to produce costs

$$J^{(k,r)} = J(\mathbf{U}^{(k)}; \mathbf{x}_0, \boldsymbol{\theta}^{(r)}) \tag{7}$$

indexed by both sample k and randomized domain r. All $R \times K$ rollouts can be performed in parallel.

B. Risk Strategies

To apply SPC, we must aggregate the rollout costs across domains. We do so via a *risk operator*

$$\bar{J}^{(k)} := \mathrm{Risk}\Big(J^{(k,1)}, J^{(k,2)}, \dots, J^{(k,R)}\Big). \tag{8}$$

This provides a single scalar score for each sample, which we can then feed into (4) to perform SPC.

While there are many possibilities for the $\mathrm{Risk}(\cdot)$ operator, we focus here on three simple strategies: *average*, *pessimistic*, and *optimistic*.

1) Average: Strategy that uses the average cost over the randomized domains:

$$\bar{J}^{(k)}_{\mathrm{avg}} \approx \frac{1}{R} \sum_{r=1}^{R} J(k, r). \tag{9}$$

This is most similar to the DR used in RL, which optimizes performance in expectation over randomized domains.

2) Pessimistic: Strategy that assumes the worst, and uses the highest cost across the randomized models:

$$\bar{J}^{(k)}_{\mathrm{pes}} := \max_{r} J^{(k,r)}. \tag{10}$$

This is a risk-averse strategy: costs are only low if they are low across all domains.

3) Optimistic: Strategy that assumes the best, and uses the lowest cost across the randomized models:

$$\bar{J}_{\text{opt}}^{(k)} := \min_{r} J^{(k,r)}. \tag{11}$$

This is a risk-seeking strategy and may seem like a strictly bad idea, especially under modeling error. However, as we show below, risk-seeking DR can produce strong and even superior performance on contact-rich tasks.

Remark 1. Risk metrics such as VaR and CVaR [16] can interpolate between risk-seeking (optimistic), risk-neutral (average), and risk-averse behavior (pessimistic), and are a natural direction for future study.

Algorithm 1 summarizes the overall procedure. Note that both loops in this algorithm are trivially parallelizable. In practice, performant SPC implementation requires spline-based dimensionality reduction and careful treatment of time shifts between replanning steps. We omit these details for space and refer the interested reader to [4], [13], [27].

C. Implementation Details

We implement the proposed MPC framework in Hydrax [13], using MuJoCo MJX [2] as the backend for parallel rollout evaluation across sampled input tapes and domain-randomized model instances.

D. Experiment Setup

The Push-T task consists of two bodies: a T-shaped block and a spherical pusher, both governed by second-order mechanical dynamics. The goal is for the spherical pusher to drive the T-shaped block to a desired pose through contact. As illustrated in Fig. 1, the spherical pusher has configuration $\mathbf{q}_p = \mathbf{p}_p \in \mathbb{R}^2$, where $\mathbf{p}_p$ denotes its planar position. The T-block has configuration $\mathbf{q}_b = [\mathbf{p}_b, \phi] \in SE(2)$, where $\mathbf{p}_b \in \mathbb{R}^2$ denotes its planar position and $\phi \in \mathbb{S}^1$ its orientation.

We compare the three risk-aware DR strategies from Sec. IV using predictive sampling [4], with controller parameters given in Table I and cost terms given in Table II. To model parametric uncertainty, we define the DR terms as

$$\boldsymbol{\theta} = \{\boldsymbol{\lambda},\ \boldsymbol{\tau},\ \mathbf{m},\ \mathbf{k}_v\}, \tag{12}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\tau}$ are the sliding-friction and contact-time-constant vectors, and $\mathbf{m}$ and $\mathbf{k}_v$ are the body-mass and actuator-gain vectors. The sampling strategy associated with each component of $\boldsymbol{\theta}$ is summarized in Table III.

At each MPC step, predictive sampling returns a pusher planar velocity command $\mathbf{u}_0$, which is applied to the pusher through a first-order closed-loop velocity servo:

$$\mathbf{v}_p = \dot{\mathbf{p}}, \quad \dot{\mathbf{v}}_p = \mathbf{k}_v(\mathbf{u}_0 - \mathbf{v}_p). \tag{13}$$

To evaluate robustness to model mismatch, each trial is associated with a unique seed that specifies both the collision-free initial condition (Table IV) and a fixed "true" model realization sampled from the randomization distribution. At planning time, the controller evaluates candidate actions over an ensemble of R randomized models, whereas execution takes place on the single fixed true model realization, thereby introducing model mismatch. We consider $R \in \{0, 4, 16, 32, 64\}$ together with the three risk strategies. For $S = 20$ seeded trials, we simulate for $T_{\text{sim}} = 7.0$ seconds.

TABLE I: Predictive sampling parameters used in the experiments.

Samples	Noise	Horizon	Spline	Knots
$K = 128$	$\sigma = 0.4$	$T = 0.5$ sec	Zero-order	$n = 6$

TABLE II: Cost terms used in the experiments, where $\|\cdot\|$ is the Euclidean norm and $\ominus$ denotes the orientation difference operator.

Cost Term	Weight	Cost Function
Block Position	$w_p = 2.0$	$\|\mathbf{p}_b^{\text{des}} - \mathbf{p}_b\|^2$
Block Orientation	$w_q = 1.0$	$\|\phi^{\text{des}} \ominus \phi\|^2$
Pusher Close-to-Block	$w_c = 0.01$	$\|\mathbf{p}_p - \mathbf{p}_b\|^2$
Pusher Velocity	$w_v = 0.01$	$\|\mathbf{v}_p\|^2$

TABLE III: DR variables for the task. Contact parameters are sampled directly, while body mass and actuator velocity gain are multiplicatively scaled. $\mathcal{U}(\cdot,\cdot)$ denotes the uniform distribution and $\odot$ denotes the Hadamard product for element-wise multiplication.

Randomization Variable	Application
Sliding Friction	$\boldsymbol{\lambda} \sim \mathcal{U}(0.5,\ 1.5)$
Contact Time Const.	$\boldsymbol{\tau} \sim \mathcal{U}(0.01,\ 0.03)$
Body Mass	$\mathbf{s}_m \odot \mathbf{m}, \quad \mathbf{s}_m \sim \mathcal{U}(0.8,\ 1.2)$
Actuator Velocity Gain	$\mathbf{s}_k \odot \mathbf{k}_v, \quad \mathbf{s}_k \sim \mathcal{U}(0.8,\ 1.2)$

TABLE IV: Initial-condition sampling used in the Push-T experiments. All initial velocities are zero.

T-block position	T-block orientation	Pusher position
$\mathbf{p}_b \sim \mathcal{U}(-0.1,\ 0.1)$	$\phi \sim \mathcal{U}(-3.14,\ 3.14)$	$\mathbf{p}_p \sim \mathcal{U}(-0.1,\ 0.1)$

E. Results

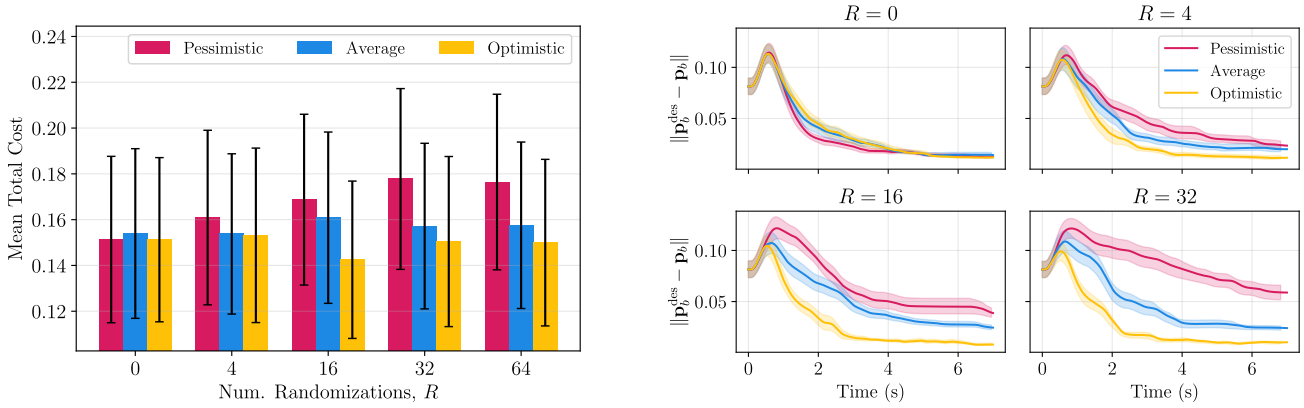
All reported metrics are computed on the executed closed-loop simulation rather than on the predicted rollouts, so the risk operator affects performance only indirectly through the planner's sample ranking. Fig. 2a shows that the *pessimistic* strategy generally performs worst and degrades as the number of randomized domains increases, while the average strategy remains intermediate. In contrast, the *optimistic* strategy consistently achieves the best or near-best mean total cost, with the strongest performance around $R = 16$. Fig. 2b shows a similar trend in block position error over time. As R increases, the *optimistic* strategy drives the block toward the goal more quickly, whereas the *pessimistic* strategy often stalls and fails to make meaningful progress. Accordingly, in difficult Push-T cases, the *optimistic* controller reaches the low-cost target region much earlier, whereas the *pessimistic* controller often fails to do so.

V. Discussion

Intuitively, one might expect a risk-averse strategy to perform best under model error. But this expectation is not supported by data, at least in the case of the Push-T. In fact, the trend that we see in Fig. 2 is quite the opposite: the risk-*seeking* strategy instead achieves the best performance.

We hypothesize that this surprising result is due to the fact that domain randomization plays two unique and sometimes conflicting roles: (i) improving robustness to model error, and (ii) shaping the basin of attraction around local minima. The first role is well-known, while the second is less so.

To illustrate the basin-shaping effect, consider a simple scalar cost $J(u)$. Domain randomization warps the cost



(a) **(b)**

Fig. 2: Comparison of risk-sensitive domain randomization strategies on the Push-T task, averaged over $S = 20$ simulations with distinct randomization seeds. (a) Time-averaged total cost over $T_{\text{sim}} = 7.0$ second trajectories ($\pm$ SE). (b) Block position error over time ($\pm$ SE) for selected values of R.

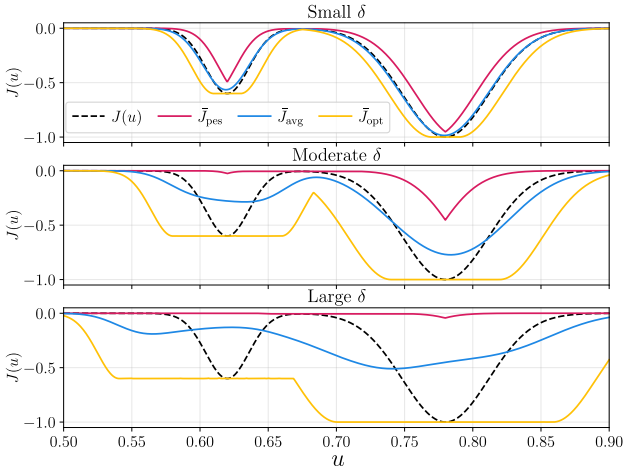


Fig. 3: Effect of risk-sensitive domain randomization on a scalar cost landscape $J(u)$ with two local minima. Each row corresponds to a different perturbation magnitude δ. The optimistic strategy $\bar{J}_{\text{opt}}$ widens the basins of attraction around local minima, while the pessimistic strategy $\bar{J}_{\text{pes}}$ narrows them. As δ increases, these basin-shaping effects become more pronounced.

landscape in complicated ways, but for ease of illustration, assume that DR merely shifts the cost landscape left and right. More precisely, for each randomized domain r we have

$$J^{(r)}(u) = J(u + \epsilon), \quad \epsilon \in \mathcal{U}(-\delta, \delta). \tag{14}$$

An example is shown in Fig. 3, along with aggregated costs $\bar{J}_{\text{avg}}$, $\bar{J}_{\text{pes}}$, $\bar{J}_{\text{opt}}$ from each of our three risk strategies.

The risk-seeking optimistic strategy ($\bar{J}_{\text{opt}}$) expands the basin of attraction around local minima, while the risk-averse pessimistic strategy ($\bar{J}_{\text{pes}}$) makes local minima smaller and narrower. Because contact-rich tasks are dominated by large (nearly) flat regions and narrow minima, clearer information about the location of a local minimum, as provided by a risk-seeking strategy, may be more important than preferring a very robust local minima. However, a risk-seeking strategy can also degrade performance by obscuring the location of a true minimum. This is quite similar to randomized smoothing [23], though DR perturbs the cost function itself rather than simply adding noise to its arguments.

This effect can be seen from a simple rollout-ranking example in Push-T. Suppose candidate A has randomized costs $\{1, 8, 9\}$, while candidate B has costs $\{4, 4, 4\}$, where lower is better. Candidate B is consistent but mediocre, and is preferred by pessimistic aggregation as $\max(A) > \max(B)$. Candidate A, however, succeeds in one randomized domain and is preferred by optimistic aggregation as $\min(A) < \min(B)$. Fig. 1 illustrates such a scenario. Since local minima are particularly narrow and difficult to hit in contact-rich tasks like the Push-T, taking this risky rollout is often worthwhile; it moves the system closer to a local minimum, where further replanning steps will be more effective.

VI. Future Work and Open Questions

The work presented here is extremely preliminary—a single task, one SPC algorithm, and only a few risk strategies. Future experimental work will focus other contact-rich tasks, more sophisticated SPC algorithms (MPPI, CEM, CMA-ES, etc.), more advanced risk strategies (VaR, CVaR, etc.), and sim-to-real transfer.

Our preliminary results also point to a pressing need for more fundamental understanding of risk-aware domain randomization for contact-rich control. What theoretical framework(s) should we leverage? Can we quantify the tradeoff between model robustness and a friendlier cost landscape? Do such tradeoffs occur in policy learning as well as predictive control? How do system dynamics (quasi-static vs unstable, high-dimensional vs low-dimensional, etc.) influence these tradeoffs? And can we use this information to design better SPC algorithms?

VII. Conclusion

Thanks to recent advances in hardware-accelerated parallel simulation and sampling-based optimization, we finally have the tools to investigate domain randomization in the predictive control setting. In this preliminary study, we demonstrated that risk-aware domain randomization produces some surprising and counter-intuitive effects for contact-rich tasks. While preliminary and limited in scope, these initial results demonstrate striking qualitative differences from the well-known impact of domain randomization on policy learning, pointing toward domain randomized predictive control as an important and fruitful area for further study.

References

[1] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30, IEEE, 2017.

[2] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, IEEE, 2012.

[3] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Munoz, X. Yao, R. Zurbrügg, N. Rudin, *et al.*, "Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning," *arXiv preprint arXiv:2511.04831*, 2025.

[4] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," *arXiv preprint arXiv:2212.00541*, 2022.

[5] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1433–1440, 2016.

[6] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and computing in applied probability*, vol. 1, no. 2, pp. 127–190, 1999.

[7] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, *et al.*, "Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning," *arXiv preprint arXiv:2501.02116*, 2025.

[8] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[9] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," in *Conference on Robot Learning*, pp. 1162–1176, PMLR, 2020.

[10] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 international conference on robotics and automation (ICRA)*, pp. 8973–8979, IEEE, 2019.

[11] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," *Frontiers in Robotics and AI*, vol. 9, p. 799893, 2022.

[12] G. Williams, A. Aldrich, and E. Theodorou, "Model predictive path integral control using covariance variable importance sampling," *arXiv preprint arXiv:1509.01149*, 2015.

[13] V. Kurtz, "Hydrax: Sampling-based model predictive control on gpu with jax and mujoco mjx," 2024. https://github.com/vincekurtz/hydrax.

[14] I. M. Balci, E. Bakolas, B. Vlahov, and E. A. Theodorou, "Constrained covariance steering based tube-mppi," in *2022 American Control Conference (ACC)*, pp. 4197–4202, IEEE, 2022.

[15] M. S. Gandhi, B. Vlahov, J. Gibson, G. Williams, and E. A. Theodorou, "Robust model predictive path integral control: Analysis and performance guarantees," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1423–1430, 2021.

[16] J. Yin, Z. Zhang, and P. Tsiotras, "Risk-aware model predictive path integral control using conditional value-at-risk," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7937–7943, IEEE, 2023.

[17] J. Yin, C. Dawson, C. Fan, and P. Tsiotras, "Shield model predictive path integral: A computationally efficient robust mpc method using control barrier functions," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7106–7113, 2023.

[18] M. Vahs, J. Choi, N. Schmid, J. Tumova, and C. Fan, "Parameter-robust mppi for safe online learning of unknown parameters," *arXiv preprint arXiv:2601.02948*, 2026.

[19] L. Janson, E. Schmerling, and M. Pavone, "Monte carlo motion planning for robot trajectory optimization under uncertainty," in *Robotics Research: Volume 2*, pp. 343–361, Springer, 2017.

[20] A. M. Jasour and B. C. Williams, "Risk contours map for risk bounded motion planning under perception uncertainties.," in *Robotics: Science and Systems*, pp. 22–26, 2019.

[21] O. De Groot, L. Ferranti, D. M. Gavrila, and J. Alonso-Mora, "Scenario-based motion planning with bounded probability of collision," *The International Journal of Robotics Research*, vol. 44, no. 9, pp. 1507–1525, 2025.

[22] T. Lew, R. Bonalli, and M. Pavone, "Risk-averse trajectory optimization via sample average approximation," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1500–1507, 2023.

[23] H. J. T. Suh, T. Pang, and R. Tedrake, "Bundled gradients through contact via randomized smoothing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4000–4007, 2022.

[24] T. Pang, H. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Transactions on robotics*, vol. 39, no. 6, pp. 4691–4711, 2023.

[25] W. Yang and M. Posa, "Dynamic On-Palm Manipulation via Controlled Sliding," in *Proceedings of Robotics: Science and Systems*, (Delft, Netherlands), July 2024.

[26] V. Kurtz and J. W. Burdick, "Generative predictive control: Flow matching policies for dynamic and difficult-to-demonstrate tasks," *arXiv preprint arXiv:2502.13406*, 2025.

[27] A. Jordana, J. Zhang, J. Amigo, and L. Righetti, "An introduction to zero-order optimization techniques for robotics," *arXiv preprint arXiv:2506.22087*, 2025.