Real-World Deployment of Massively Parallel Sampling-Based MPC for Contact-Rich Manipulation

Magnus Dierking[1,2], João Carvalho[1,3], An T. Le[1,6], Georgia Chalvatzaki[2,4,5] and Jan Peters[1,3,4,5]

Abstract—Sampling-based Model Predictive Control (SMPC) is a promising strategy for contact-rich robotic manipulation, combining gradient-free optimization with massively parallel GPU simulation. Yet, most prior work relies on simplified dynamics or remains confined to simulation.

We present an MPC framework that leverages JAX for large-scale parallelization and efficient computation, coupled with the high-fidelity MuJoCo MJX simulator, and deploy it on a Franka Research 3 executing the Push-T manipulation task through a complete real-to-sim-to-real pipeline. The MTP variant with structured global sampling outperforms unimodal baselines such as CEM, MPPI, and PS across tasks that require mode switching, both in simulation and on hardware. Furthermore, we evaluate online domain randomization within the MPC sample budget, showing that contact-initiation parameters yield interpretable adaptation signals, whereas global physics parameters provide feedback that is too weak for reliable exploitation at typical replanning frequencies. These findings highlight key challenges for sampling-based MPC in contact-rich manipulation—contact sensitivity, tight compute budgets, and the difficulty of obtaining informative domain-randomization signals in real time.

Index Terms—sampling-based MPC, model tensor planning, contact-rich manipulation, sim-to-real, domain randomization

I. Introduction

Classical gradient-based MPC assumes smooth, differentiable dynamics—an assumption that breaks down in contact-rich manipulation, where dynamics are hybrid and nonsmooth. Sampling-based MPC (SMPC) circumvents this limitation by drawing candidate control trajectories, evaluating them via forward simulation, and updating the control distribution based on outcomes. Combined with JAX [1] and GPU-accelerated physics engines such as MuJoCo MJX [2], we can roll out thousands of trajectories in parallel, making real-time MPC feasible for robotics.

Despite growing interest, real-robot SMPC has focused primarily on locomotion. DIAL-MPC [3] and Whole-Body MPPI [4] achieve impressive results on quadruped jumping and climbing, while [5] extends the approach to drone flight. For manipulation, Pezzato et al. [6] present the most relevant real-world study. However, their demonstrations on a dexterous anthropomorphic hand remain proofs of concept—the authors report limited GPU throughput and real-time control constraints that restricted trajectory smoothness and prevented effective parallelization.

Standard SMPC algorithms such as MPPI [7] and CEM [8] rely on unimodal Gaussian proposals and often become trapped in local minima when the cost landscape is multimodal—a common occurrence in contact-rich manipulation. Model Tensor Planning (MTP) [9] addresses these problems by augmenting local Gaussian sampling with graph-based global candidates, enabling a better exploration vs. exploitation tradeoff. However, MTP has so far been evaluated only in simulation, leaving open questions about real-world applicability, scalability under real-time constraints, and sim-to-real gap sensitivity. Contact-rich manipulation tasks such as Push-T expose these limitations directly, providing reproducible benchmarks in both simulated and physical settings.

This paper makes the following contributions:

- Simulation baseline comparison of MTP, MPPI, CEM, and Predictive Sampling (PS) [10] on tasks requiring multimodal exploration;
- Real-robot deployment of SMPC for contact-rich manipulation using a high-fidelity physics backend (MuJoCo MJX) on the Push-T task with a Franka Research 3.
- An empirical analysis of online domain randomization within SMPC, identifying which physics parameters yield interpretable adaptation signals and why learning from them remains difficult under contact-rich dynamics and tight compute constraints.

II. Background

A. Sampling-Based MPC

We consider a discrete-time system $\mathbf{x}_{t+1} = \mathbf{g}(\mathbf{x}_t, \mathbf{v}_t)$, with state $\mathbf{x}_t$ and control input $\mathbf{v}_t \sim \mathcal{N}(\mathbf{u}_t, \mathbf{\Sigma})$. Sampling-based MPC seeks a mean sequence $\mathbf{U}_k = [\mathbf{u}_0, \dots, \mathbf{u}_{T-1}]$ that minimises an expected trajectory cost $S(\mathbf{V}_k)$ over horizon T by evaluating N candidate trajectories.

The Information-Theoretic MPC (IT-MPC) framework [11] grounds this in the Gibbs Variational Principle,

$$-\lambda\,\mathcal{F}(S, p, \lambda) \;\le\; \mathbb{E}_{\mathbb{Q}}[S(\mathbf{V})] + \lambda\,\mathrm{KL}(\mathbb{Q}\|\mathbb{P}), \qquad (1)$$

where $\mathcal{F}$ is the free energy of cost S under prior $\mathbb{P}$ and $\lambda{>}0$ is a temperature. Minimising the right-hand side over Gaussian proposals yields exponential reweighting

$$\mathbf{U}_k \;\leftarrow\; \sum_{i=1}^{N} w^i \mathbf{V}_k^i, \quad w^i \propto \exp\!\big(-S(\mathbf{V}_k^i)/\lambda\big), \qquad (2)$$

[1]Intelligent Autonomous Systems Lab, TU Darmstadt, Germany; [2]Interactive Robot Perception & Learning Lab, TU Darmstadt, Germany; [3]German Research Center for AI (DFKI); [4]Hessian.AI; [5]Robotics Institute Germany (RIG). [6]College of Engineering and Computer Science, VinUniversity, Vietnam. This work was funded by the German Federal Ministry of Education and Research Software Campus project ROBOSTRUCT (16|S23067). Corresponding author: Magnus Dierking, magnus.dierking@stud.tu-darmstadt.de.

which recovers MPPI [7] under Gaussian perturbations. CEM [8] replaces soft reweighting with hard elite selection (top-E trajectories); PS [10] sets the next mean to the lowest cost rollout.

B. Model Tensor Planning

Using a Gaussian distribution to model control sequences can lead to local minima. MTP [9] addresses this issue by mixing local Gaussian samples with global tensor-path samples $\mathbf{V}^G$ drawn from a layered graph over the control space $\mathcal{U}$. These are combined with local perturbations $\mathbf{V}^L \sim \mathcal{N}(\mathbf{U}, \alpha\mathbf{\Sigma})$ into a joint batch of control signals $\mathbf{V} = [\mathbf{V}^G; \mathbf{V}^L]$. The update applies CEM-style elite selection followed by MPPI reweighting over elites, and the best sample is applied (as in PS). A mixing coefficient $\beta \in [0, 1]$ controls the global/local budget split. Near a low-cost mode, local samples dominate the elite set and the algorithm behaves like MPPI; near a local minimum, global samples enter the elites and shift the mean across cost modes.

C. Domain Randomization in Sampling-Based MPC

Domain randomization improves sim-to-real robustness by considering a distribution of physics parameters $h(\boldsymbol{\xi})$. Abraham et al. [12] show that we can extend IT-MPC as

$$-\lambda\,\mathcal{F}(S, p, \lambda) \;\le\; \mathbb{E}_{h(\boldsymbol{\xi})}\Big[\mathbb{E}_{\mathbb{Q}}[S(\mathbf{V})] + \lambda\,\mathrm{KL}(\mathbb{Q}\|\mathbb{P})\Big], \qquad (3)$$

promoting control sequences that are robust across the domain distribution. In practice, this amounts to running parallel rollouts over D randomised environments and aggregating per-domain costs before the weight update (2).

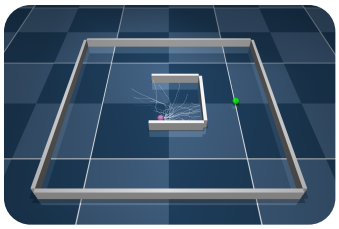
III. Framework & Methodology

A. Framework and Environments

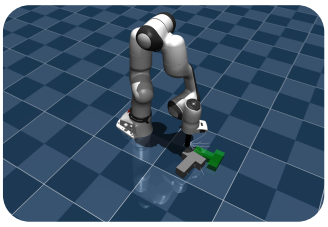
Our framework is built on top of Hydrax [13], a JAX-based SMPC framework using MuJoCo MJX [2] as the dynamics backend, enabling end-to-end JIT compilation and vmap-vectorized rollouts across thousands of parallel trajectories on GPU without hand-crafted dynamics models. We contribute two environments, framework optimizations, and a complete real-robot pipeline.

Bugtrap Escape (fig. 1a) consists of a 2D point mass inside a U-shaped trap built from physical box primitives. The cost penalizes external contact forces and distance to goal; starting inside the trap, the agent must discover an escape route—a direct test of mode-switching capability.

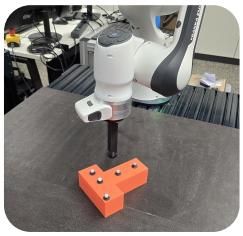
Push-T (fig. 1b) requires pushing a rigid T-shaped block to a target pose using a Franka Research 3 arm (fig. 1c). We use a 7 DoF free-floating rigid body (with orientation as a quaternion) that captures realistic friction, mass distribution, and full table-contact dynamics. The cost combines T-pose alignment, end-effector proximity to the object, and a retraction term to prevent singularities. We sample end-effector twists and map them to joint velocities with differentiable inverse kinematics.



(a) Bugtrap Escape. Green sphere is the goal. Grey are trajectory samples.



(b) Push-T in simulation. The green object is the goal pose.



(c) Push-T real-world setup, with a FR3 arm, custom end-effector, and Mocap markers.

Fig. 1: Simulation environments and real-world setup.

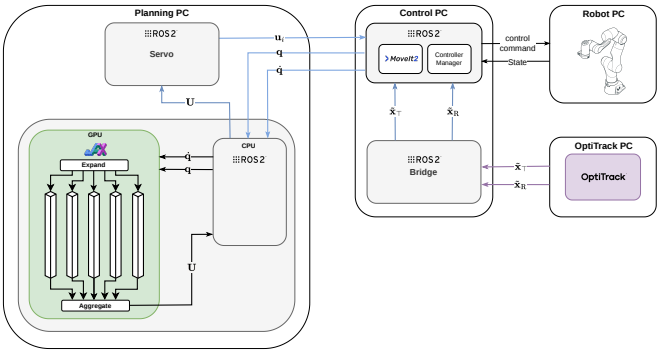


Fig. 2: Real-world MPC experimental system overview based on ROS2. Blue arrows indicate ROS topic-based communication between processes. Purple arrows indicate Mocap streaming.

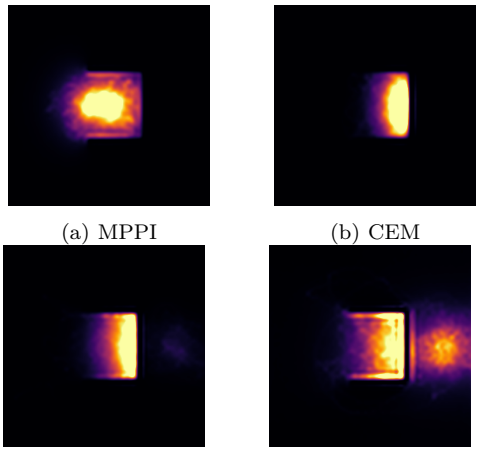
B. Real-to-Sim-to-Real Pipeline

The real-robot system couples the planning loop to a Franka Research 3 and a 3D printed T through four asynchronous ROS 2 [14] processes (fig. 2). Two control nodes run MoveIt2 [15] Servo for 1 kHz Cartesian velocity streaming and select the current horizon step at 50 Hz. A bridge node publishes Mocap poses (via OptiTrack) at 100 Hz. The planning node executes the JIT-compiled MPC on an NVIDIA RTX 5090 GPU at 8–10 Hz.

Because planning and servo nodes are decoupled, the servo node interpolates within the published action trajectory based on elapsed time, maintaining smooth control between replans. Hand–eye calibration [16] aligns the OptiTrack world frame with the robot base frame.

C. Domain Randomization

We implement a structured domain randomization (DR) interface supporting per-body, per-geom, and per-joint randomization. For parameters with derived quantities



(a) MPPI (b) CEM

(c) PS (d) MTP

Fig. 3: Empirical state visitation distributions for the Bugtrap Escape task.

(mass, inertia), we compile a separate MuJoCo model per domain tuple and extract the resulting arrays into the batched pytree, ensuring physical consistency. Each planning step rolls out N trajectories over D randomized domains simultaneously; per-domain costs are aggregated via a weighted average, splitting the sample budget between trajectory samples and domains.

Per-domain weights w_d are either kept uniform or updated online based on the discrepancy between expected and observed system state. Concretely, we store the predicted object pose $\hat{\mathbf{T}}^{(d)}$ under each domain for the elite sample, and compare it against the observation $\mathbf{T}_{\text{obs}}$ at the next replanning step:

$$w_d \propto \exp\!\big(-d_{\text{SE(3)}}(\hat{\mathbf{T}}^{(d)}, \mathbf{T}_{\text{obs}})/\sigma^2\big), \tag{4}$$

where $d_{\text{SE(3)}}$ is the geodesic pose error, and σ a scaling parameter.

IV. Experiments

Our experiments address the following research questions: (RQ1) Does MTP outperform baselines that rely only on unimodal Gaussians on tasks that require switching between exploration and exploitation? (RQ2) How do results transfer to real hardware under real-time constraints? (RQ3) Which physics parameters yield informative domain-randomization feedback signals?

A. Bugtrap Escape – Simulation Experiment

All SMPC algorithms run at 25 Hz over a 0.72 sec horizon for a total of 10,000 planning steps. We run 128 samples across 6 seeds with noise on the initial particle position; control sequences are sampled in velocity space. The algorithm's behaviors can be seen by inspecting the state (position) visitations in Fig. 3. MPPI and CEM fail in every seed: CEM's greedy elite selection traps it near the inner wall in a contact-avoiding limit cycle, while MPPI's full-sample averaging drives the particle toward the trap center where wall-collision costs and the goal attractor equilibrate. PS escapes once but otherwise mirrors CEM.

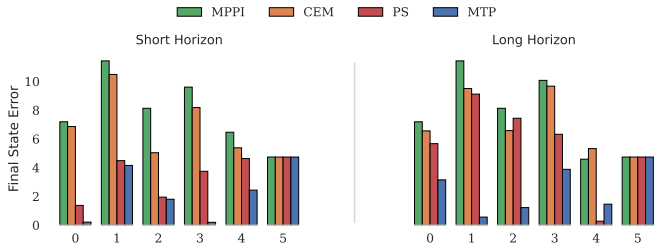


Fig. 4: Sim-to-sim results for Push-T with long and short horizons. The number in the horizontal axis corresponds to a different initial state of the system.

Due to its high exploration capability, MTP escapes and reaches the goal in all seeds.

B. Push-T – Simulation and Real-World Experiments

In simulation, all methods use 256 control samples and plan at 20 Hz with long (0.625 sec) and short (0.375 sec) horizons over 100 planning steps. Fig. 4 presents the results. To test the exploration capabilities, we deliberately set the end-effector and T's initial pose to hard-to-solve configurations. MTP achieves the lowest final pose error across seeds that require mode switching, while unimodal baselines stall in local minima or degrade at shorter horizons.

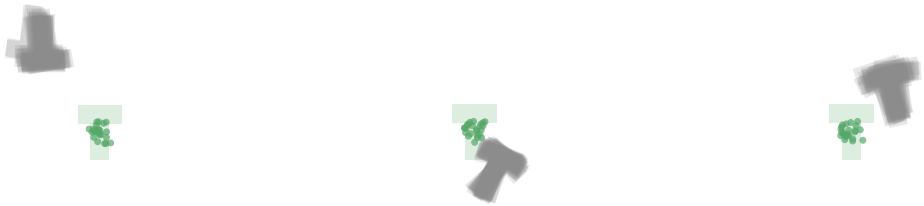
In real-robot evaluations, we run 6 seeds with small variations in T's initial pose and end-effector position. We use 1024 control sequence samples, planning at 8–10 Hz. The MJX physics solver dominates runtime, requiring coarser timesteps and shorter horizons to meet replanning targets. Integration steps above 0.025 s consistently lead to instability via deep single-step penetrations that activate constraint-resolution accelerations rather than physical friction. This forces a fundamental tradeoff: contact fidelity requires small steps, and real-time replanning requires shorter horizons, and both degrade receding-horizon performance if tasks need longer horizons for planning.

The results for different initial sampling distributions are reported in Fig. 5. MTP achieves the lowest average final pose error and variance. MPPI frequently avoids contact entirely. PS initiates contact aggressively but degrades from overshooting and brittle contact exploitation. CEM produces the smoothest motions but stalls in local minima after initial pose correction.

C. Domain Randomization Analysis

Domain randomization splits the sample budget across D parallel domains. We sweep over global physics parameters (friction, mass) and contact-initiation parameters (end-effector collision margins) across 6 diverse seeds with 8 domains, 160 samples per domain, planning at 5 Hz.

Mass and friction yield no stable adaptation signal. Physics is dominated by contact timing and constraint resolution forces rather than by smooth variation in the randomized parameter. The resulting error signal is asymmetric: low-mass/friction domains overshoot and accumulate large pose errors, while high-mass domains



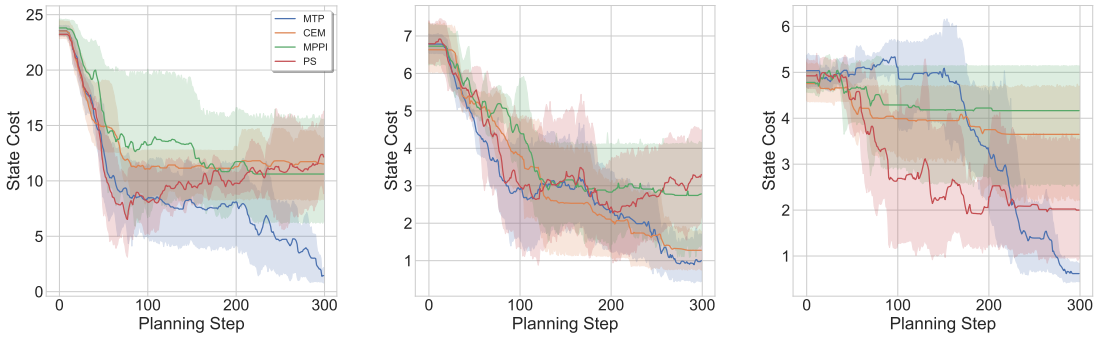


Fig. 5: Real-to-sim-to-real results on the Push T task.

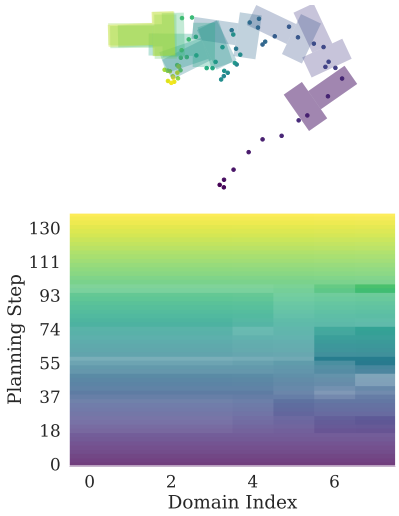


Fig. 6: Exemplary evolution of domain weights for the domain-randomized Push-T task under margin randomization. Time is encoded by color and weight magnitude by transparency. Margins are randomized uniformly over 8 domains within ±1 cm around the end-effector.

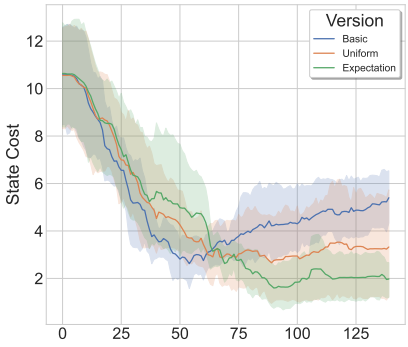


Fig. 7: Real-world results of the MPC cost over time steps for different domain-randomization strategies.

move less and can appear spuriously consistent with the observation — causing adaptive weights to drift toward heavy/high friction domains.

Contact margins produce interpretable, mode-dependent weight evolution (Fig. 6): weights shift toward larger margins during turning and away during straight pushing, consistent with margins directly gating contact onset. On hardware, kinematic misalignment causes the non-randomized baseline to accumulate error after initial pose correction; both uniform DR and adaptive reweighting improve final pose quality (Fig. 7), though the advantage of adaptive over uniform weighting is small and not yet statistically conclusive.

V. Conclusion

We presented a high-fidelity MuJoCo MJX-based SMPC framework and systematically evaluated in simulation and on real hardware. The MTP algorithm consistently outperforms unimodal baselines in the Bugtrap Escape task, since its mixed global-local sampling is the only strategy that reliably escapes the local minimum. On Push-T, it achieves the lowest final pose error and best variance in both simulation and real-robot trials.

Real-time deployment exposes a fundamental tension: small timesteps are required to avoid deep interpenetrations, yet shorter horizons degrade receding-horizon performance on tasks where long-range credit assignment matters.

Additionally, global physics parameters (mass, friction) are effectively non-identifiable within a single replanning interval, whereas contact-initiation parameters (collision margins) directly gate contact onset, yielding interpretable adaptation signals and measurable robustness gains. This is a structural limitation of the online SMPC setting that offline RL-based DR [17] does not share, since training amortizes contact noise across thousands of episodes.

As future work, to mitigate the longer-horizon problem, we will explore integrating learned value-function surrogates and generative sampling priors.

References

[1] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. (2018) Jax: Composable transformations of python+numpy programs. Software repository. [Online]. Available: https://github.com/jax-ml/jax

[2] DeepMind. (2025) Mujoco documentation: Computation. [Online]. Available: https://mujoco.readthedocs.io/en/stable/computation/index.html#geintegration

[3] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-Order Sampling-Based MPC for Torque-Level Locomotion Control via Diffusion-Style Annealing," Sep. 2024.

[4] J. Alvarez-Padilla, J. Z. Zhang, S. Kwok, J. M. Dolan, and Z. Manchester, "Real-time whole-body control of legged robots with model-predictive path integral control," in 2025 IEEE International Conference on Robotics and Automation (ICRA), 2025, pp. 14 721–14 727.

[5] T. Belvedere, M. Ziegltrum, G. Turrisi, and V. Modugno, "Feedback-mppi: Fast sampling-based mpc via rollout differentiation – adios low-level controllers," IEEE Robotics and Automation Letters, vol. 11, no. 1, pp. 1–8, 2026.

[6] C. Pezzato, C. Salmi, E. Trevisan, M. Spahn, J. Alonso-Mora, and C. H. Corbato, "Sampling-based Model Predictive Control Leveraging Parallelizable Physics Simulations," Jan. 2025.

[7] G. Williams, A. Aldrich, and E. A. Theodorou, "Model Predictive Path Integral Control: From Theory to Parallel Computation," Journal of Guidance, Control, and Dynamics, vol. 40, no. 2, pp. 344–357, Feb. 2017.

[8] C. Pinneri, S. Sawant, S. Blaes, J. Achterhold, J. Stueckler, M. Rolinek, and G. Martius, "Sample-efficient Cross-Entropy Method for Real-time Planning," Aug. 2020.

[9] A. T. Le, K. Nguyen, M. N. Vu, J. Carvalho, and J. Peters, "Model Tensor Planning," May 2025.

[10] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo," Dec. 2022.

[11] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic MPC for model-based reinforcement learning," in 2017 IEEE International Conference on Robotics and Automation (ICRA). Singapore: IEEE, May 2017, pp. 1714–1721.

[12] I. Abraham, A. Handa, N. Ratliff, K. Lowrey, T. D. Murphey, and D. Fox, "Model-Based Generalization Under Parameter Uncertainty Using Path Integral Control," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2864–2871, Apr. 2020.

[13] V. Kurtz. (2024) Hydrax. Software repository. [Online]. Available: https://github.com/vincekurtz/hydrax

[14] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074

[15] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "Moveit! task constructor for task-level motion planning," in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 190–196.

[16] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pp. 345–358, 1989.

[17] 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). IEEE, 2017, pp. 23–30.