Projection-Retraction MPPI: Exact Constraint-Manifold Control for Manipulators

Abstract— **Model Predictive Path Integral (MPPI) control is widely used in manipulation for its gradient-free, parallel handling of non-convex costs. Manipulation tasks, however, often impose constraints that hold throughout the motion: a closed kinematic chain that two grasping arms keep exactly, or joint limits and obstacle clearances that are never crossed. MPPI handles such constraints only through the cost, as soft penalties that hold approximately and fail under a strong task cost. To address this, we propose Projection-Retraction MPPI (PR-MPPI), which enforces the constraints inside the sampled dynamics. At every rollout step, the sampled velocity is projected to satisfy both constraint types: the equality restricts it to a subspace, and each inequality to a half-space within that subspace, so inequality handling never breaks the equality. This projection, however, satisfies the constraints only to first order, and a finite step leaves a small drift off the equality. Therefore, we retract the returned command back onto the constraint to numerical tolerance and independent of task weighting. We validate PR-MPPI on 14-DoF dual-arm systems. In simulation, the returned commands satisfy the closed-chain equality to numerical tolerance through a joint-limit stress test and randomized obstacle avoidance. On real hardware, the arms of a Unitree H1-2 humanoid reactively avoid a moving obstacle. Code and experiment videos are available at `https://anonymous.4open.science/w/prmppi/`.**

I. Introduction

Model Predictive Path Integral (MPPI) control [1], [2] is widely used in manipulation. It optimizes the task by sampling control sequences and weighting them by their rollout costs, requiring no gradients, which suits the non-convex, non-smooth costs induced by contact and clutter. The rollouts are also evaluated fast through massive GPU parallelization, as demonstrated by frameworks such as STORM [3]. Manipulation tasks, however, often impose constraints that hold throughout the motion, not merely at the goal. A wiping tool stays on the work surface. Two arms rigidly grasping an object form a closed kinematic chain whose relative pose holds exactly, or internal wrenches build up and the grasp is lost. All along the way, joint limits and obstacle clearances are never crossed. MPPI handles constraints only through soft, penalty-based cost evaluation, and therefore cannot enforce a hard constraint.

To enforce constraints more strictly than a soft penalty, several lines of work augment MPPI with dedicated machinery on the inequality side. Chance-constrained and barrier-based formulations add probabilistic or forward-invariant safety [4]. In the same spirit, discrete-time control-barrier filters such as Shield-MPPI [5] repair the optimized control through a local barrier correction. A parallel line of work constrains the samples themselves rather than the cost. PRIEST [6] projects them toward collision- and kinematic-feasible sets, and MPPI-Tan [7] redirects them along obstacle-gradient tangent spaces. CSC-MPPI [8] pushes them out of obstacles by primal-dual gradient steps and selects the update from a DBSCAN cluster rather than a global average, while π-MPPI [9] projects the sampled inputs onto magnitude and smoothness bounds. Whether by barrier repair or by sample correction, however, these methods consider one-sided *inequality* margins only and offer no means of holding an equality.

Enforcing a kinematic equality along a motion has a long history outside sampling-based MPC. Classical redundancy resolution holds task equalities instantaneously by projecting joint velocities onto the constraint null space [10]–[13]. As a reactive one-step law, however, it performs no horizon optimization and offers no principled coupling to inequality margins. Sampling-based motion planners on constraint manifolds instead project or retract random configurations onto the manifold [14]–[16]. This yields feasible paths offline, but not a closed-loop policy that re-solves the problem at every control cycle. These literatures thus supply the core operations, projection and retraction, but neither embeds them in a sampling-based MPC loop.

Two recent controllers bring this machinery into MPPI, and are closest to our setting. DQ-MPPI [17] adopts the analytical route directly, projecting the samples onto the null space of the chain constraint. It handles the residual off-manifold drift only through null-space feedback at the low level, rather than removing it to a prescribed tolerance. The constraint is also confined to the dual-arm closed chain. MC-MPPI [18] instead replaces the analytic constraint with a feasible manifold learned by a variational autoencoder. It samples within that manifold and corrects the selected command with a single quadratic-program step, which restores feasibility only approximately. Because the feasible set is a learned model, its reliability also depends on the training distribution and model accuracy. Both methods, moreover, enforce the equality only in the sampling stage and leave inequalities to the cost. As a result, neither returns an equality-feasible command to numerical tolerance while handling inequality constraints together, and neither keeps an inequality correction from pushing the state off the equality. Table I summarizes this landscape.

To bridge this gap, we propose Projection-Retraction MPPI (PR-MPPI), which enforces equality and inequality constraints within the rollout itself rather than through the cost. At every rollout step, each sampled velocity passes through a single velocity-space projection: the equality confines it to the tangent subspace of the constraint manifold,

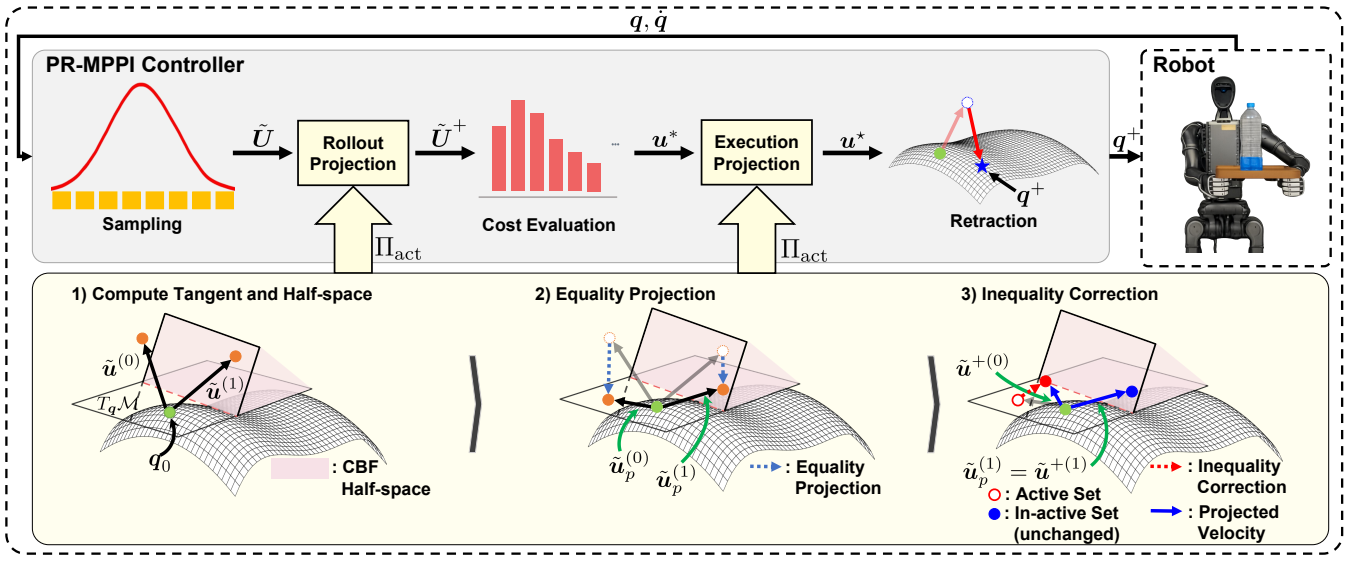


Fig. 1. Overview of PR-MPPI. Sampled control sequences are rolled out through the constraint-aware dynamics, where the active constraint projector Π_{act} is applied at every step of every rollout. The filtered inputs are averaged by the exponential-weighted update, the averaged sequence is filtered once more along its own predicted states, and the first step is retracted onto $\mathcal{M}$. The lower panels illustrate Π_{act} on two rollout samples in three steps: (1) the tangent space of $\mathcal{M}$ and the CBF half-space of an active margin, (2) projection of the sampled inputs onto the tangent space, and (3) correction of the violating input within that tangent space, leaving the admissible one unchanged.

TABLE I

CONSTRAINT HANDLING IN SAMPLING-BASED MPC.

Method	Ineq.[a]	Eq.[a]	Exact[b]	Free[c]
Shield-MPPI [5]	✓	×	×	✓
PRIEST [6]	✓	×	×	✓
π-MPPI [9]	✓	×	×	✓
MPPI-Tan [7]	✓	×	×	✓
CSC-MPPI [8]	✓	×	×	✓
MC-MPPI [18]	×	✓	×	×
DQ-MPPI [17]	×	✓	×	✓
PR-MPPI (ours)	✓	✓	✓	✓

[a] The constraint is enforced within the sampled rollouts or by a dedicated repair layer, rather than only through a cost penalty. Eq. denotes a configuration-space equality maintained throughout the motion.
[b] The returned command satisfies the equality to numerical tolerance.
[c] Constraint satisfaction does not require learned model.

and every active inequality confines it further to a half-space inside that subspace. The two are therefore not combined by weighting or priority but nested: an inequality correction stays inside the equality subspace by construction, for any activation pattern. This matters because a naive fix, such as clamping a joint at its limit, can silently break the closed chain. The projection, however, holds the constraints only to first order, and a finite step leaves a small drift off the equality. A retraction step therefore pulls the commanded state back onto the manifold to the prescribed tolerance. Finally, the rollouts and the returned command pass through the same projection, so the plan is optimized under the same first-order constraint geometry that shapes the command.

The main contributions of this paper are as follows:

- We propose PR-MPPI, a constrained sampling-based MPC framework in which a single velocity-space projection carries both constraint types inside the sampled dynamics: each active inequality is corrected within the equality tangent subspace, so inequality handling cannot break the equality regardless of which margins are active. A retraction step then removes the finite-step drift that the first-order projection leaves on the equality, returning the command to satisfy the equality to numerical tolerance.
- We establish a command-level equality property and characterize the finite-step inequality error. Whenever the retraction terminates, the returned command satisfies the equality constraint to the prescribed numerical tolerance, independent of the MPPI weighting. Under the stated assumptions, the inequality error introduced by the finite step and the retraction is $O(\Delta t^2)$, small enough to be absorbed by the safety margin.
- We validate PR-MPPI on a 14-DoF dual-arm Franka system in simulation, where the retracted commands satisfy the closed-chain equality to numerical tolerance while the measured trajectories retain the joint-limit and obstacle margins under aggressive task commands, and demonstrate reactive avoidance of a moving obstacle on the dual arms of a Unitree H1-2 humanoid.

II. METHODOLOGY

In this section, we present PR-MPPI, which incorporates equality and inequality constraints into MPPI through projection and retraction. During each sampled rollout, the perturbed velocity is projected onto the intersection of the equality-constraint tangent space and the active inequality half-spaces. After the MPPI update, the averaged sequence is filtered once more along its own predicted states, and the finite-step equality drift of the first command is removed by retracting the integrated configuration onto the constraint manifold. The overall framework is illustrated in Fig. 1.

A. Problem Definition

The control input is the joint velocity, $\boldsymbol{u}_t = \dot{\boldsymbol{q}}_t \in \mathbb{R}^n$. We consider the discrete-time system

$$\boldsymbol{q}_{t+1} = f(\boldsymbol{q}_t, \tilde{\boldsymbol{u}}_t), \tag{1}$$

where $\boldsymbol{q}_t \in \mathbb{R}^n$ is the joint configuration and $\tilde{\boldsymbol{u}}_t \in \mathbb{R}^n$ is the stochastic control input. Our objective is to find an optimal nominal control sequence $\boldsymbol{U} = [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{T-1}]$ over a prediction horizon T that minimizes the expected trajectory cost while satisfying both hard equality and inequality constraints under stochastic perturbations. The optimization problem is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{U}} \; & J = \mathbb{E}\left[\phi(\boldsymbol{q}_T) + \sum_{t=0}^{T-1}\left(l(\boldsymbol{q}_t) + \frac{1}{2}\boldsymbol{u}_t^T \boldsymbol{R}\boldsymbol{u}_t\right)\right], \\ \text{s.t.} \quad & \boldsymbol{c}(\boldsymbol{q}_t) = \boldsymbol{0}, \quad \boldsymbol{h}(\boldsymbol{q}_t) \geq \boldsymbol{0}, \\ & \tilde{\boldsymbol{u}}_t = \boldsymbol{u}_t + \delta\boldsymbol{u}_t, \quad \delta\boldsymbol{u}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_u), \\ & \boldsymbol{q}_0 = \boldsymbol{q}_{\text{init}}, \quad \boldsymbol{q}_{t+1} = f(\boldsymbol{q}_t, \tilde{\boldsymbol{u}}_t), \end{aligned} \tag{2}$$

where $\boldsymbol{c}(\boldsymbol{q}_t) \in \mathbb{R}^{n_e}$ and $\boldsymbol{h}(\boldsymbol{q}_t) \in \mathbb{R}^{n_i}$ denote the equality and inequality constraints imposed on the joint space, respectively. The matrix $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ is a positive-definite weight matrix penalizing the nominal control effort, and $\boldsymbol{\Sigma}_u \in \mathbb{R}^{n\times n}$ is the exploration covariance.

B. Constraint-Aware Sampled Dynamics

The constrained problem in (2) requires the sampled rollouts to respect both equality constraints and inequality margins. A direct penalty-based treatment is undesirable because a sufficiently large task cost can still make sampled trajectories violate the hard constraints. Instead of treating these constraints only through the cost, PR-MPPI projects each sampled velocity in velocity space before it enters the dynamics. The trajectory costs are then evaluated on near-feasible motions, so the MPPI average is formed over rollouts that pass through the same projection as the final control input. We first express both constraint types as conditions on the velocity and then define the filter and the resulting sampled dynamics.

The equality constraint defines the manifold

$$\mathcal{M} = \{\boldsymbol{q} : \boldsymbol{c}(\boldsymbol{q}) = \boldsymbol{0}\}, \qquad \boldsymbol{J}_c(\boldsymbol{q}) = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{q}} \in \mathbb{R}^{n_e \times n}, \tag{3}$$

and, to first order, a velocity control input $\boldsymbol{u}$ preserves the constraint if and only if $\boldsymbol{J}_c(\boldsymbol{q})\boldsymbol{u} = \boldsymbol{0}$. For the inequalities, the desired safety buffer $\boldsymbol{h}_{\text{safe}} \in \mathbb{R}^{n_i}$, elementwise positive, is embedded directly into the inequality function, and the shifted margins are kept nonnegative to first order by the CBF condition:

$$\bar{\boldsymbol{h}}(\boldsymbol{q}) = \boldsymbol{h}(\boldsymbol{q}) - \boldsymbol{h}_{\text{safe}}, \qquad \boldsymbol{J}_h(\boldsymbol{q})\,\boldsymbol{u} \geq -\boldsymbol{\Gamma}\,\bar{\boldsymbol{h}}(\boldsymbol{q}), \tag{4}$$

where $\boldsymbol{J}_h(\boldsymbol{q}) = \partial\bar{\boldsymbol{h}}/\partial\boldsymbol{q} \in \mathbb{R}^{n_i\times n}$ and $\boldsymbol{\Gamma} = \text{diag}(\gamma_1, \ldots, \gamma_{n_i})$ with $\gamma_i > 0$. The shifted guard condition is $\bar{\boldsymbol{h}}(\boldsymbol{q}) \geq \boldsymbol{0}$, and each row of the CBF condition bounds how fast the velocity may approach the corresponding boundary. Both conditions are linear in the input. The active constraint projector Π_{act} updates the sampled input to its projection onto their intersection:

$$\begin{aligned} \tilde{\boldsymbol{u}}_t^+ = \Pi_{\text{act}}\left(\boldsymbol{q}_t, \tilde{\boldsymbol{u}}_t\right) = \arg\min_{\boldsymbol{u}} \; & \tfrac{1}{2}\left\|\boldsymbol{u} - \tilde{\boldsymbol{u}}_t\right\|^2 \\ \text{s.t.} \quad & \boldsymbol{J}_c(\boldsymbol{q}_t)\boldsymbol{u} = \boldsymbol{0}, \\ & \boldsymbol{J}_h(\boldsymbol{q}_t)\boldsymbol{u} \geq -\boldsymbol{\Gamma}\,\bar{\boldsymbol{h}}(\boldsymbol{q}_t). \end{aligned} \tag{5}$$

The system dynamics in (1) then take the constraint-aware form

$$\boldsymbol{q}_{t+1} = f_{\mathcal{M}}\left(\boldsymbol{q}_t, \tilde{\boldsymbol{u}}_t\right) = \boldsymbol{q}_t + \tilde{\boldsymbol{u}}_t^+\,\Delta t, \tag{6}$$

and every sampled rollout is generated by (6). The rest of this subsection describes how (5) is computed inside the rollouts, in two stages: an equality tangent-space projection and an active inequality correction.

The first stage projects the sampled input onto the tangent space of $\mathcal{M}$ through the null-space projector:

$$\tilde{\boldsymbol{u}}_{p,t} = \boldsymbol{N}(\boldsymbol{q}_t)\,\tilde{\boldsymbol{u}}_t, \qquad \boldsymbol{N}(\boldsymbol{q}) = \boldsymbol{I} - \boldsymbol{J}_c^{\dagger}(\boldsymbol{q})\boldsymbol{J}_c(\boldsymbol{q}), \tag{7}$$

where $\boldsymbol{N} \in \mathbb{R}^{n\times n}$ is the null-space projector and $\boldsymbol{J}_c^{\dagger}$ the pseudoinverse of $\boldsymbol{J}_c$, so that $\boldsymbol{J}_c(\boldsymbol{q}_t)\tilde{\boldsymbol{u}}_{p,t} = \boldsymbol{0}$. Since the feasible set of (5) lies in the tangent space, the distances to $\tilde{\boldsymbol{u}}_t$ and to $\tilde{\boldsymbol{u}}_{p,t}$ differ there only by the constant $\|(\boldsymbol{I} - \boldsymbol{N})\tilde{\boldsymbol{u}}_t\|^2$. The second stage enforces the CBF half-spaces within this tangent space. Since each sampled input violates only a few half-spaces, if any, solving (5) with all n_i half-spaces at every step of every rollout is unnecessary. The projection is therefore solved over a small active set that is grown only when needed from the initial members selected by the CBF residual of each margin at the equality-projected input:

$$r_i\left(\boldsymbol{q}_t, \boldsymbol{u}\right) = -\gamma_i \bar{h}_i(\boldsymbol{q}_t) - \nabla\bar{h}_i(\boldsymbol{q}_t)^T\boldsymbol{u}, \tag{8}$$

where $\nabla\bar{h}_i(\boldsymbol{q})^T$ is the i-th row of $\boldsymbol{J}_h(\boldsymbol{q})$, and $r_i(\boldsymbol{q}_t, \boldsymbol{u}) \leq 0$ recovers the i-th row of the CBF condition in (4). A positive residual means the input is pushing the rollout too fast toward the i-th margin. All violating margins and margins within the activation band form the initial active set,

$$\mathcal{A}_t^{(0)} = \left\{i : r_i(\boldsymbol{q}_t, \tilde{\boldsymbol{u}}_{p,t}) > 0 \;\text{ or }\; \bar{h}_i(\boldsymbol{q}_t) < h_{\text{band}}\right\}. \tag{9}$$

Margins near the boundary are included preemptively through the band, which covers half-spaces that the correction itself could push into violation. If $\mathcal{A}_t^{(0)} = \emptyset$, the equality-projected input already satisfies every half-space, so $\tilde{\boldsymbol{u}}_t^+ = \tilde{\boldsymbol{u}}_{p,t}$. Otherwise, at iteration ℓ, (5) is solved with only the half-spaces in $\mathcal{A}_t^{(\ell)}$, and by the KKT conditions its solution can be written as

$$\boldsymbol{u}_t^{(\ell)} = \tilde{\boldsymbol{u}}_{p,t} + \sum_{i\in\mathcal{A}_t^{(\ell)}} \mu_{i,t}\,\boldsymbol{N}(\boldsymbol{q}_t)\nabla\bar{h}_i(\boldsymbol{q}_t), \quad \mu_{i,t} \geq 0, \tag{10}$$

where $\mu_{i,t}$ are the dual variables associated with the selected CBF half-spaces. By complementary slackness, a band-selected half-space that remains nonbinding at the solution has $\mu_{i,t} = 0$. The residuals of the omitted half-spaces are then checked at $\boldsymbol{u}_t^{(\ell)}$. If any is violated, the most violated one is added,

$$\mathcal{A}_t^{(\ell+1)} = \mathcal{A}_t^{(\ell)} \cup \left\{\arg\max_{i\notin\mathcal{A}_t^{(\ell)}} r_i(\boldsymbol{q}_t, \boldsymbol{u}_t^{(\ell)})\right\}, \tag{11}$$

and the restricted solve is repeated. The iteration terminates when every omitted residual satisfies $r_i(\boldsymbol{q}_t, \boldsymbol{u}_t^{(\ell)}) \le 0$, and the final iterate is taken as $\tilde{\boldsymbol{u}}_t^{+}$. At termination the iterate satisfies the selected half-spaces by construction and the omitted ones by the check, so it is the minimizer of (5) with all n_i half-spaces. Since every correction term carries $\boldsymbol{N}$ as a left factor and $\boldsymbol{J}_c\boldsymbol{N} = \boldsymbol{0}$, applying $\boldsymbol{J}_c$ to (10) gives $\boldsymbol{J}_c\tilde{\boldsymbol{u}}_t^{+} = \boldsymbol{J}_c\tilde{\boldsymbol{u}}_{p,t} = \boldsymbol{0}$: the filtered input preserves the equality to first order.

Each rollout generated by (6) is assigned the trajectory cost of (2), yielding $\{S^{(k)}\}_{k=1}^{K}$. Let $\tilde{\boldsymbol{U}}^{+(k)} = [\tilde{\boldsymbol{u}}_0^{+(k)}, \dots, \tilde{\boldsymbol{u}}_{T-1}^{+(k)}]$ denote the filtered input sequence of the k-th rollout. The nominal sequence is then updated by exponential-weighted averaging of the filtered sequences:

$$\boldsymbol{U}^{*} = \sum_{k=1}^{K} w_k\, \tilde{\boldsymbol{U}}^{+(k)}, \qquad w_k \propto \exp\!\Big(-\tfrac{1}{\lambda} S^{(k)}\Big), \tag{12}$$

where $\lambda > 0$ is the temperature parameter and the weights are normalized to sum to one.

C. Execution-Layer Projection and Retraction

The MPPI update (12) returns $\boldsymbol{U}^{*} = [\boldsymbol{u}_0^{*}, \dots, \boldsymbol{u}_{T-1}^{*}]$ by exponential-weighted averaging of the projected inputs. Each sample, however, was projected at the states of its own rollout, so the averaged inputs carry no constraint guarantee along the trajectory that $\boldsymbol{U}^{*}$ itself generates. The averaged sequence therefore requires filtering before actuation, and the system dynamics (6) serve this role. Starting from the measured state $\boldsymbol{q}_0$, the sequence is rolled forward, and each input is filtered into the final control input for $t = 0, \dots, T-1$:

$$\boldsymbol{u}_t^{\star} = \Pi_{\mathrm{act}}\left(\boldsymbol{q}_t, \boldsymbol{u}_t^{*}\right). \tag{13}$$

This execution filtering corrects each input on the state it will actually visit, with the same projector as in the rollouts. The initial execution active set follows the same rule,

$$\mathcal{A}_t^{(0)} = \big\{ i : r_i(\boldsymbol{q}_t, \boldsymbol{N}\boldsymbol{u}_t^{*}) > 0 \text{ or } \bar{h}_i(\boldsymbol{q}_t) < h_{\mathrm{band}} \big\}, \tag{14}$$

so simultaneously critical margins are enforced jointly, and the active-set iteration (11) refines this set until every omitted half-space is satisfied.

Although (13) enforces $\boldsymbol{J}_c(\boldsymbol{q}_0)\boldsymbol{u}_0^{\star} = \boldsymbol{0}$, this is a first-order condition on the commanded input. Starting from $\boldsymbol{q}_0 \in \mathcal{M}$, the finite step therefore leaves $\boldsymbol{c}(\boldsymbol{q}_0 + \Delta t\, \boldsymbol{u}_0^{\star}) = O(\Delta t^2)$. We remove this finite-step drift by initializing $\boldsymbol{q} = \boldsymbol{q}_0 + \Delta t\, \boldsymbol{u}_0^{\star}$ and applying the Gauss–Newton iteration

$$\boldsymbol{q} \leftarrow \boldsymbol{q} - \boldsymbol{J}_c^{\dagger}(\boldsymbol{q})\, \boldsymbol{c}(\boldsymbol{q}) \qquad \text{until} \quad \|\boldsymbol{c}(\boldsymbol{q})\| < \varepsilon_{\mathrm{tol}}. \tag{15}$$

The iteration defines the retraction map $\mathcal{R}_{\mathcal{M}}$, yielding the final command $\boldsymbol{q}^{+} = \mathcal{R}_{\mathcal{M}}(\boldsymbol{q}_0 + \Delta t\, \boldsymbol{u}_0^{\star})$.

Remark 1 (Retraction exactness). The stopping rule of (15) makes the commanded tolerance a design property rather than a theorem: whenever the iteration terminates, the retracted command satisfies $\|\boldsymbol{c}(\boldsymbol{q}^{+})\| < \varepsilon_{\mathrm{tol}}$, independently of the task cost, the MPPI temperature and covariance, and the inequality-handling parameters.

Algorithm 1 Projection-Retraction MPPI Framework

Require:
- $\boldsymbol{q}_0$: Current measured configuration
- $\boldsymbol{U} = [\boldsymbol{u}_0, \dots, \boldsymbol{u}_{T-1}]$: Nominal control sequence
- $\boldsymbol{\Sigma}_u$: Exploration covariance
- K, T: Number of samples and horizon
- $\boldsymbol{c}, \bar{\boldsymbol{h}}$: Equality and shifted inequality constraints

for $k = 1, \dots, K$ **do** ▷ Parallel sampling
  $\delta\boldsymbol{u}_{0:T-1}^{(k)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_u)$
  $\boldsymbol{q}_0^{(k)} \leftarrow \boldsymbol{q}_0$
  for $t = 0, \dots, T-1$ **do**
    $\tilde{\boldsymbol{u}}_t^{(k)} \leftarrow \boldsymbol{u}_t + \delta\boldsymbol{u}_t^{(k)}$
    $\tilde{\boldsymbol{u}}_t^{+(k)} \leftarrow \Pi_{\mathrm{act}}(\boldsymbol{q}_t^{(k)}, \tilde{\boldsymbol{u}}_t^{(k)})$ ▷ Eq. (5)
    $\boldsymbol{q}_{t+1}^{(k)} \leftarrow \boldsymbol{q}_t^{(k)} + \tilde{\boldsymbol{u}}_t^{+(k)}\Delta t$ ▷ Eq. (6)
  end for
  Evaluate trajectory cost $S^{(k)}$
end for
Compute exponential weights $\{w^{(k)}\}_{k=1}^{K}$ via (12)
$\boldsymbol{U}^{*} \leftarrow \sum_{k=1}^{K} w^{(k)}\tilde{\boldsymbol{U}}^{+(k)}$ ▷ MPPI update
for $t = 0, \dots, T-1$ **do**
  $\boldsymbol{u}_t^{\star} \leftarrow \Pi_{\mathrm{act}}(\boldsymbol{q}_t, \boldsymbol{u}_t^{*})$ ▷ Execution filtering, Eq. (13)
  $\boldsymbol{q}_{t+1} \leftarrow \boldsymbol{q}_t + \boldsymbol{u}_t^{\star}\,\Delta t$
end for
$\boldsymbol{q}^{+} \leftarrow \mathcal{R}_{\mathcal{M}}(\boldsymbol{q}_0 + \Delta t\, \boldsymbol{u}_0^{\star})$ ▷ Retraction
$\boldsymbol{U} \leftarrow [\boldsymbol{u}_1^{\star}, \dots, \boldsymbol{u}_{T-1}^{\star}, \boldsymbol{0}]$ ▷ Warm start
return $\boldsymbol{q}^{+}$

Lemma 1 (Finite-step margin bound)**.** *Let $\boldsymbol{c}$ and each $\bar{h}_i$ be twice continuously differentiable with bounded second derivatives on the operating set, on which $\boldsymbol{J}_c$ has constant row rank and $\|\boldsymbol{J}_c^{\dagger}\|$ is bounded. Suppose that, at every control cycle, the retraction* (15) *terminates at its tolerance, so that the executed configuration satisfies $\|\boldsymbol{c}(\boldsymbol{q}_t)\| < \varepsilon_{\mathrm{tol}}$ with $\varepsilon_{\mathrm{tol}} = O(\Delta t^2)$. Here $\boldsymbol{q}_t$ denotes the configuration at cycle t and $\boldsymbol{u}_t^{\star}$ the actuated input of that cycle. If* (5) *is feasible at $\boldsymbol{q}_t$ at every cycle, $\|\boldsymbol{u}_t^{\star}\| \le \bar{u}$, $\|\nabla\bar{h}_i\| \le \bar{L}$, and $\gamma_i\Delta t \le 1$, then there exist constants $\kappa_i \ge 0$, independent of Δt, such that*

$$\bar{h}_i(\boldsymbol{q}_{t+1}) \ge (1 - \gamma_i\Delta t)\bar{h}_i(\boldsymbol{q}_t) - \kappa_i\Delta t^2, \tag{16}$$

so $\bar{h}_i(\boldsymbol{q}_t) \ge 0$ implies $\bar{h}_i(\boldsymbol{q}_{t+1}) \ge -\kappa_i\Delta t^2$.

Proof. $\|\boldsymbol{c}(\boldsymbol{q}_t)\| < \varepsilon_{\mathrm{tol}}$ and the equality constraint enforced by (13), $\boldsymbol{J}_c(\boldsymbol{q}_t)\boldsymbol{u}_t^{\star} = \boldsymbol{0}$, give an integrated step whose equality residual is $O(\Delta t^2) + O(\varepsilon_{\mathrm{tol}}) = O(\Delta t^2)$. Under the rank and pseudoinverse bounds, the retraction displacement is of the same order. At termination of the active-set iteration, the actuated input satisfies every CBF half-space, $\nabla\bar{h}_i(\boldsymbol{q}_t)^T\boldsymbol{u}_t^{\star} \ge -\gamma_i\bar{h}_i(\boldsymbol{q}_t)$. A Taylor expansion of $\bar{h}_i$ along the step, with the tolerance term and the retraction displacement absorbed into κ_i, then gives (16). With $\gamma_i\Delta t \le 1$ and $\bar{h}_i(\boldsymbol{q}_t) \ge 0$, the right side is at least $-\kappa_i\Delta t^2$. □

The bound makes the layer $\{\boldsymbol{q} : \bar{h}_i(\boldsymbol{q}) \ge -(\kappa_i/\gamma_i)\Delta t\}$ forward invariant, since $-(\kappa_i/\gamma_i)\Delta t$ is a fixed point of the recursion (16). A safety buffer $h_{\mathrm{safe},i} \ge (\kappa_i/\gamma_i)\Delta t$ with

TABLE II

JOINT-LIMIT VIOLATIONS, MEASURED CONSTRAINT RESIDUALS, AND MPPI-INNER COMPUTATION TIMES OVER THE 0–6 s INTERVAL.

Method	Chain trans. (m)	Chain rot. (rad)	Tilt (rad)	Max. bound viol. (rad)	Max. margin pen. (rad)	Compute time (ms)
DQ-MPPI [17]	0.004 ± 0.000	0.004 ± 0.001	0.061 ± 0.047	0.082	0.132	238.948 ± 19.473
PR-MPPI Full[a]	0.003 ± 0.000	0.003 ± 0.000	0.002 ± 0.000	0.000	0.003	19.351 ± 1.216
PR-MPPI w/o retraction	0.007 ± 0.002	0.008 ± 0.003	0.011 ± 0.004	0.000	< 0.001	18.288 ± 0.514
PR-MPPI w/o inequality[a]	0.003 ± 0.000	0.003 ± 0.000	0.002 ± 0.000	0.022	0.072	8.975 ± 0.092

[a] The returned command satisfies every equality channel to below 10^{-9} throughout. The table entries are measured-state residuals.

$\bar{h}_i(\boldsymbol{q}_0) \geq 0$ therefore keeps $h_i(\boldsymbol{q}_t) \geq 0$ at every cycle. If the per-step parameter $\bar{\gamma}_i = \gamma_i \Delta t \in (0, 1]$ is held fixed, the invariant-layer width is $(\kappa_i/\bar{\gamma}_i)\Delta t^2$, recovering the $O(\Delta t^2)$ finite-step layer.

III. EXPERIMENTS

A. Experimental Setup

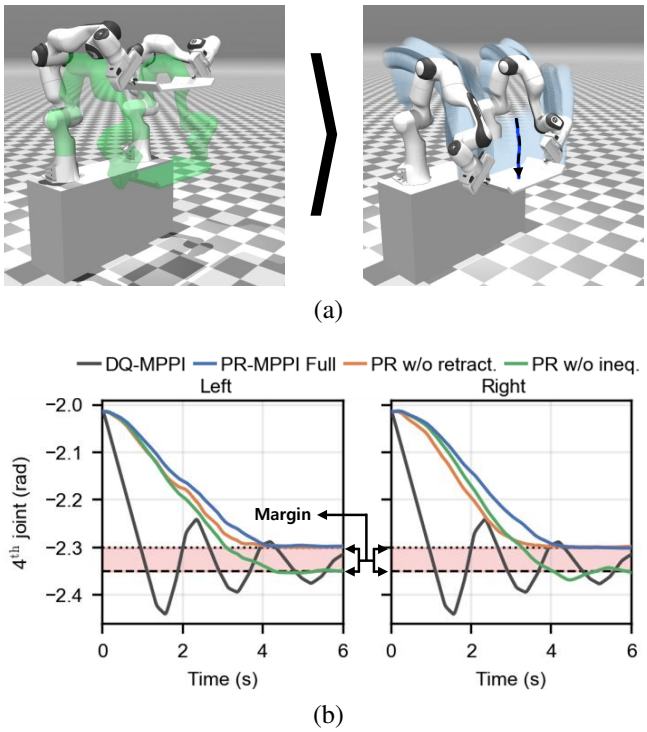
In this section, we present three experiments designed to evaluate the proposed method. Experiment 1 examines the multi-inequality projection when an aggressive joint-tracking objective activates multiple joint-limit margins simultaneously, and assesses the respective roles of inequality projection and equality retraction. Experiment 2 investigates how incorporating inequality projection into the sampled dynamics affects the generation of near-feasible rollout trajectories in a randomized obstacle-avoidance task; and Experiment 3 evaluates hardware applicability and reactive avoidance of a moving obstacle on a humanoid robot. Experiments 1 and 2 use a dual-arm Franka Emika Panda in MuJoCo [19], whereas Experiment 3 uses the two 7-DoF arms of a Unitree H1-2 humanoid.

In every experiment, the arms rigidly grasp a tray with both end-effectors, and the enforced equality constraint is the resulting closed-chain condition

$$\boldsymbol{c}(\boldsymbol{q}) = \begin{bmatrix} \log\big(\boldsymbol{E}(\boldsymbol{q})\big)^{\vee} \\ \varphi_o \\ \vartheta_o \end{bmatrix} \in \mathbb{R}^8, \qquad \boldsymbol{E}(\boldsymbol{q}) = \boldsymbol{T}_l \, {}^{l}\boldsymbol{T}_r \, \boldsymbol{T}_r^{-1}, \tag{17}$$

where $\boldsymbol{T}_l, \boldsymbol{T}_r \in SE(3)$ are the end-effector poses, ${}^{l}\boldsymbol{T}_r$ and ${}^{l}\boldsymbol{T}_o$ are the grasp-fixed relative poses of the right gripper and the tray, and (φ_o, ϑ_o) are the roll and pitch of the tray pose $\boldsymbol{T}_l \, {}^{l}\boldsymbol{T}_o$. The inequalities $\boldsymbol{h}(\boldsymbol{q})$ collect the joint-limit and obstacle margins specified in each scenario. All experiments run on an Intel Core i5-13400F CPU with 16 GB of RAM and an NVIDIA GeForce RTX 4060 Ti GPU (8 GB VRAM).

PR-MPPI is configured identically in all experiments: $K = 1000$ rollouts over a horizon of $T = 30$ steps with $\Delta t = 1/30$ s, and an isotropic exploration covariance $\boldsymbol{\Sigma}_u = \sigma^2 \boldsymbol{I}$ with $\sigma = 0.03$ rad/s on the commanded joint velocities, and every inequality uses the class-$\mathcal{K}$ gain $\gamma_i = 5$. PR-MPPI runs at 30 Hz, and its retracted joint command is tracked by a computed-torque law at the 500 Hz state rate. The task costs, their weights, and the detailed parameters for each experiment are provided on the project page, `https://anonymous.4open.science/w/prmppi/`.



(a)

(b)

Fig. 2. Joint-limit stress test. **(a)** Dual-arm tray-lowering motion: the goal joint configuration in green and the trajectory executed by Full PR-MPPI in blue. **(b)** Measured fourth-joint trajectories of both arms for the four controllers. The dashed line is the artificial lower bound $q_{4,\min} = -2.35$ rad and the dotted line the shifted guard boundary at -2.30 rad; the shaded band between them is the 0.05 rad CBF margin.

B. Experiment 1: Joint-Limit Stress Test

The system is initialized at a configuration corresponding to a level tray at $(0.45, 0, 1.0)$ m and tracks an IK solution for $(0.45, 0, 0.55)$ m. Both arms start from $q_4 = -2.01$ rad, whereas the IK target is $q_{4,g} = -2.41$ rad. We impose the artificial lower bound $q_{4,\min} = -2.35$ rad and a 0.05 rad CBF safety margin, which shifts the guard boundary to -2.30 rad. Because the target lies 0.06 rad beyond the bound in both arms, the two joint-limit constraints become active nearly simultaneously.

We compare Full PR-MPPI with DQ-MPPI and two ablations: PR-MPPI without retraction, which bypasses the final equality retraction, and PR-MPPI without inequality projection, which disables joint-limit projection in both the sampled dynamics and the execution layer. In this experiment, the joint-tracking cost weight was set to ten times the joint-limit violation penalty weight. No joint-limit violation penalty was

applied to Full PR-MPPI or PR-MPPI without retraction.

Figure 2 and Table II summarize the outcome. Because the tracking objective aggressively drives both fourth joints toward the bound, a soft joint-limit penalty can be traded against a sufficiently aggressive task objective. A cost-based treatment alone therefore cannot guarantee the limit, and the measurements confirm it: PR-MPPI without inequality projection and DQ-MPPI penetrate the bound by 0.022 and 0.082 rad, respectively. Full PR-MPPI uses no joint-limit violation penalty and instead removes the bound-violating component from every sampled and executed velocity: both joints settle onto the guard boundary and slide along it, the bound is never crossed, and the residual 0.003 rad margin penetration is absorbed by the 0.05 rad safety shift. Lemma 1 characterizes the remaining finite-step command-space margin deviation as $O(\Delta t^2)$. The 0.05 rad safety shift is selected conservatively to absorb this deviation and tracking errors; the absence of a measured-state bound crossing remains an empirical result.

The measured equality residuals in Table II are similar across the controllers. The chain channels of all four methods remain at the 10^{-3} level because the two methods without retraction still take a null-space projection step at every cycle. DQ-MPPI uses its low-level null-space servo, whereas PR-MPPI without retraction uses the execution-layer projection (13). The commanded control inputs, however, show that the retraction step is required for command-space equality satisfaction to numerical tolerance. The command of Full PR-MPPI satisfies every channel of (17) to within $(0.170 \pm 0.070) \times 10^{-10}$ m in chain translation, $(0.280 \pm 0.020) \times 10^{-10}$ rad in chain rotation, and $(0.055 \pm 0.052) \times 10^{-10}$ rad in tilt. These values are the numerical floor of the projection and retraction pipeline, while the measured residuals reflect low-level tracking error. For PR-MPPI without retraction and DQ-MPPI, the commanded residuals coincide with the measured ones because finite-step drift from tangent-space integration remains in the command and cannot be removed by the low-level controller. DQ-MPPI additionally handles tray tilt only as a soft cost, resulting in a larger mean tilt residual (0.061 versus 0.002 rad).

As shown in Table II, the computational differences reflect the structure of each method. PR-MPPI applies the inequality projection at every step of every sampled rollout, so its cost increases with the number of inequality constraints, while the retraction is applied only once to the selected command and contributes little overhead. DQ-MPPI, evaluated with the authors' official implementation[1], performs two MPPI stages together with a clustering step. On the same hardware and at the same sampling budget, its planning update takes 238.948 ms against 19.351 ms for Full PR-MPPI. Both methods run their low-level null-space controller at 500 Hz between planning updates; the difference lies in the replanning rate, about 4.2 Hz for DQ-MPPI versus the sustained 30 Hz of PR-MPPI.

[1]Available at `https://dq-gpu-mpc.github.io/`.

TABLE III

OUTCOME COUNTS OVER 30 RANDOMIZED OBSTACLE PLACEMENTS.

Method	Success/ Total	Stuck	Tray drop	Collision
MC-MPPI [18]	13/30	17	0	0
DQ-MPPI [17]	18/30	1	11	0
PR-MPPI Full	30/30	0	0	0
PR-MPPI Exec.-Only Ineq.	29/30	1	0	0

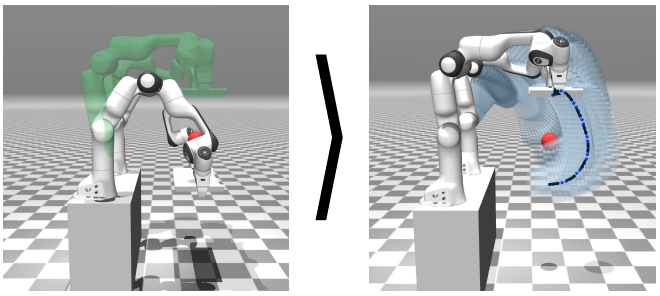


Fig. 3. Randomized obstacle-avoidance task at the central lateral offset. The tray is transported past the spherical obstacle (red) under the closed-chain grasp; green marks the goal tray pose and blue a representative collision-free detour by Full PR-MPPI, with its dashed tray-center path.

C. Experiment 2: Randomized Obstacle Avoidance

The level tray moves from $(0.5, 0, 0.5)$ m to $(0.5, 0, 1.0)$ m past a spherical obstacle of radius 0.05 m centered at $(x_i, y_i, 0.75)$ m, where x_i and the lateral offset y_i are drawn at random within $[0.40, 0.60]$ m and $[-0.10, 0.10]$ m, respectively, for each of the 30 paired trials and shared by all controllers, so that each must find a lateral detour around the blocked straight path in Fig. 3. All methods use task-space tracking. PR-MPPI takes $h(\boldsymbol{q})$ as the tray–obstacle distance with safety margin $h_{\text{safe}} = 0.02$ m, whereas DQ-MPPI and MC-MPPI [18] rely on a soft collision cost alone. Full PR-MPPI handles the obstacle through the projection only, with no collision penalty in its cost. An execution-only ablation of PR-MPPI disables the inequality projection inside the sampled dynamics while keeping the execution filtering and the retraction, and instead carries the same squared penetration penalty as the baselines; all other settings are identical.

All controllers run for 120 s against the same outcome geometry: a collision is declared when the tray contacts the obstacle sphere, which exerts no physical force, and a run succeeds when the tray pose error $e = \| \log(\boldsymbol{T}^{-1}\boldsymbol{T}_g) \|_2$, which stacks the translational (m) and rotational (rad) components of the relative twist, stays below 0.02 [2] for 1 s; failed runs are classified by their first failure event as collision, tray drop, or stuck.

Table III summarizes the outcomes, which separate primarily by how each method handles the obstacle inequality. MC-MPPI completes 13 of 30 trials, while all 17 unsuccessful trials are classified as stuck: no collision or tray drop occurs, but the controller does not attain the goal dwell

[2]Matching the criterion of [17], which thresholds the dual-quaternion pose-error norm—approximately $e/2$ for small errors—at 0.01.

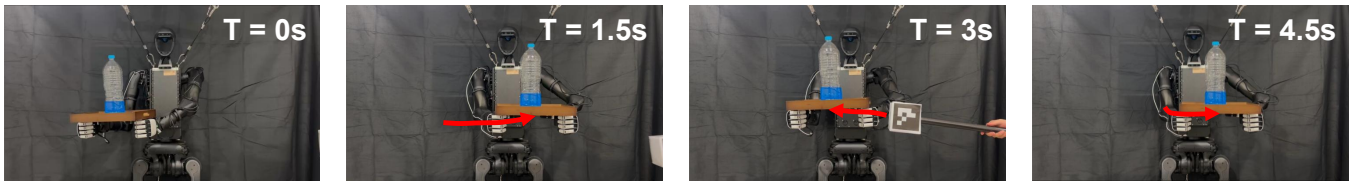


Fig. 4. Hardware demonstration on the Unitree H1-2. The two arms hold a tray carrying a payload under the closed-chain grasp while a hand-held obstacle is moved toward it; each frame is labelled with the elapsed time.

within the horizon. The stuck behavior is most evident when the obstacle lies near the center of the nominal path, around $(0.5, 0, 0.75)\ m$. Because the tray is wide relative to the available clearance, a large fraction of the sampled trajectories then intersect the obstacle, leaving too few informative feasible samples that pass around either side. The cost-only update consequently fails to establish a consistent lateral detour, and the motion stalls near the obstacle.

DQ-MPPI employs a two-stage MPPI scheme whose first stage performs high-variance exploration. Under the common sampling budget, it also has the longer update time reported in Table II, so its closed-loop behavior reflects both the two-stage update and its attainable replanning rate. In the trials, DQ-MPPI produces more abrupt commanded motion and longer executed paths. On the same 18 placements where DQ-MPPI succeeds, its mean path length is $1.338\ m$, compared with $0.686\ \ m$ for Full PR-MPPI. The larger step-to-step motion also magnifies the finite-step error of the first-order constraint correction, increasing the closed-chain residuals. Indeed, the tray-drop trials exhibit larger mean chain translation, chain rotation, and tilt residuals than the successful trials: the respective values increase from $0.0111\ m$, $0.0105\ rad$, and $0.0239\ rad$ to $0.0159\ m$, $0.0181\ rad$, and $0.0301\ rad$. The resulting loss of closed-chain consistency can degrade tray support; 11 of the 12 failures are tray drops, while the remaining failure is stuck.

The execution-only ablation separates the two roles of the projection. With the execution filtering kept, the constraints themselves hold: whatever the update proposes, the actuated command is corrected on the states it visits, and no collision or tray drop occurs in any trial. The plan, however, is still evaluated on rollouts that violate the obstacle margin, and the cost of this appears in how the search behaves. When the obstacle blocks the center of the nominal path, few informative samples pass around either side, so the search takes longer or stalls. The ablation reaches the goal dwell in $18.217 \pm 10.183\ s$ against $17.951 \pm 5.046\ s$ for Full PR-MPPI, with twice the spread, and the same effect appears as path variability: it travels $0.757 \pm 0.107\ m$ against $0.636 \pm 0.040\ m$, with the largest detours at placements close to the nominal path. Its single failure, out of 30 trials, is stuck at exactly such a placement. Full PR-MPPI projects every sampled input inside the rollouts, so the trajectory cost is evaluated on motions that respect the margin and the update averages over rollouts that already pass the obstacle. It runs with the tray–obstacle penalty disabled entirely and completes every trial.

D. Experiment 3: Reactive Avoidance of a Moving Obstacle on Hardware

The two 7-DoF arms of a Unitree H1-2 humanoid hold a loaded tray under the closed-chain grasp while a person advances a hand-held obstacle toward it from varying directions. The current obstacle position is supplied to the controller at each control cycle and held fixed over the rollout horizon, so the controller replans reactively without predicting obstacle motion. The same sampling–projection–retraction pipeline used in simulation is deployed online with only the platform's kinematic model substituted. As shown in Fig. 4, the arms reshape their configuration within the equality tangent subspace to open clearance, while the retraction returns grasp-consistent commands and the measured tray remains level. This experiment demonstrates online hardware deployment and reactive avoidance.

A limitation nevertheless emerged when the obstacle advanced along a direction whose clearance-increasing motion the closed-chain equality forbids. Near such configurations the projected gradient of the margin, $\boldsymbol{N}\nabla\bar{h}$, nearly vanishes, and no admissible input can increase clearance without violating the grasp. The failure is therefore structural rather than an artifact of control rate or latency: it is the price of satisfying the commanded equality to numerical tolerance, since the very projection that guarantees the grasp also excludes the retreating motion. A principled treatment would relax the grasp equality selectively when the admissible tangent motion is insufficient, trading a bounded grasp error for recovered clearance; we leave this to future work.

IV. Conclusion

We presented Projection-Retraction MPPI, a sampling-based MPC that enforces equality and inequality constraints inside its rollouts through one nested velocity-space projection, with a retraction that satisfies the commanded equality to numerical tolerance independently of the task weighting. On a 14-DoF dual-arm system, the returned commands reached this tolerance while the measured trajectories retained the true inequality margins under aggressive task costs. The same pipeline was deployed on the arms of a Unitree H1-2 humanoid for reactive avoidance of a moving obstacle. Approaches whose retreat direction the closed chain forbids exposed the structural cost of command-space equality enforcement.

The proposed formulation operates at the kinematic level: the projection and retraction act on configurations and velocities, and a low-level controller tracks the retracted command.

The structure itself, however, is not tied to this choice. Sampled torques can equally be projected onto the dynamically consistent tangent space of the same constraint manifold, with torque limits and contact-force margins entering as inequalities of the same nested form, which would place the exactness guarantee at the actuation layer. This torque-level extension, together with the selective relaxation of the grasp equality when the admissible tangent motion is insufficient, is left to future work.

REFERENCES

[1] 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)*. IEEE, 2016, pp. 1433–1440.

[2] 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, 2017.

[3] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots, "Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 750–759.

[4] J. Yin, P. Tsiotras, and K. Berntorp, "Chance-constrained information-theoretic stochastic model predictive control with safety shielding," in *Proc. IEEE Conf. Decision and Control (CDC)*, 2024, pp. 653–658.

[5] 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 Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7106–7113, 2023.

[6] F. Rastgar, H. Masnavi, B. Sharma, A. Aabloo, J. Swevers, and A. K. Singh, "PRIEST: Projection guided sampling-based optimization for autonomous navigation," *arXiv preprint arXiv:2309.08235*, 2023.

[7] G. Zhao, N. Jin, J. Wu, and Z. Xiong, "Online motion generation via tangential sampling-based MPC around nonconvex obstacles," *IEEE Robot. Autom. Lett.*, vol. 10, no. 6, pp. 5537–5544, 2025.

[8] L. Park, K. Jang, and S. Kim, "CSC-MPPI: A novel constrained MPPI framework with DBSCAN for reliable obstacle avoidance," *arXiv preprint arXiv:2506.16386*, 2025.

[9] E. M. Andrejev, A. Manoharan, K.-E. Unt, and A. K. Singh, "π-MPPI: A projection-based model predictive path integral scheme for smooth optimal control of fixed-wing aerial vehicles," *IEEE Robot. Autom. Lett.*, vol. 10, no. 6, 2025.

[10] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *J. Dyn. Syst. Meas. Control*, vol. 108, no. 3, pp. 163–171, 1986.

[11] B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, 1991, pp. 1211–1216.

[12] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, 1995.

[13] B. Dariush, Y. Zhu, A. Arumbakkam, and K. Fujimura, "Constrained closed loop inverse kinematics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2010, pp. 2499–2506.

[14] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009, pp. 625–632.

[15] L. Jaillet and J. M. Porta, "Path planning under kinematic constraints by rapidly exploring manifolds," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 105–117, 2013.

[16] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *Int. J. Robot. Res.*, vol. 38, no. 10–11, pp. 1151–1178, 2019.

[17] T. Zhu, J. Mao, J. Yang, and S. Li, "Real-time dual-arm cooperative manipulation under multiple constraints: A two-stage sampling MPC approach," *IEEE Transactions on Robotics*, 2026.

[18] S. Lee and S. Kim, "Manifold-constrained model predictive path integral control," *arXiv preprint arXiv:2605.24813*, 2026.

[19] 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*. IEEE, 2012, pp. 5026–5033.