Sampling-Based and Contact-Implicit Nonprehensile Manipulation using Signed Distance Functions

Nikola Raicevic

PI: Nikolay Atanasov

Department of Electrical and Computer Engineering
University of California San Diego

Thursday 4th September, 2025

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

Motivation

Real-world help requires touch

- Homes are cluttered and unpredictable, intentional contact makes behavior robust
- Safe assistance around people and fragile items needs gentle, controlled touch

Everyday tasks that depend on contact

- *Cleaning & organizing*
- *Making a smoothie*
- *Assisting the elderly*
- *Quick household fixes*
- *Kitchen help*

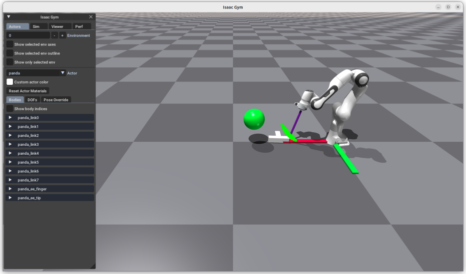
Background

- **Contact changes the world state**
  Touch creates forces and *mode switches* (no contact / pushing / pulling / lifting-off) $\rightarrow$ discontinuities

- **Classical planners fall short**
  They operate in continuous spaces and assume collision-free motion, treating objects as obstacles, not tools

- **What we need**
  Let the robot *discover* useful contacts (when/where/how) as part of the plan—no brittle, pre-fixed contact scripts

Xarm6 with Tee



Example: an arm uses touch to achieve a goal rather than avoiding it

Related Work

Learning-based

- *Pros:* learns rich, non-linear policies; end-to-end; adapts with more data
- *Cons:* data-demanding; reward shaping; sim-to-real/transfer gaps
- *Limits:* tends to overfit object/task; generalization to new shapes is hard
- *Example:* Visual Manipulation with Legs [1]

Sampling-based

- *Pros:* handles multi-modal choices; real-time re-planning; no gradients needed
- *Cons:* model-free variants *sample blindly*; cost shaping heavily influences outcomes
- *Limits:* no built-in guidance for *where/when* to contact; many rollouts for delicate timing
- *Example:* DIAL-MPC [2]

Model-based

- *Pros:* interpretable; geometry/dynamics aware; efficient when models match reality
- *Cons:* explicit contact/complementarity can be brittle
- *Limits:* fine-tuning per object shape/scene; manual constraint engineering
- *Example:* model-based pushing CRISP [3]

[1] He, X., et al. (2025). *Visual Manipulation with Legs*. Proceedings of The 8th Conference on Robot Learning, PMLR

[2] Xue, H., et al.(2024). *Full-Order Sampling-Based MPC for Torque-Level Locomotion Control via Diffusion-Style Annealing*

[3] Li, Y., et al. (2025). *On the Surprising Robustness of Sequential Convex Optimization for Contact-Implicit MP*

Related Work: Our Approach

What Is Missing?

- Geometry-aware guidance without per-object re-training or manual mode schedules
- A planner that *discovers* contact strategy (where/when/how) based on the task
- Generalization across shapes/layouts with sample efficiency and interpretability

Our Approach:

- Contact-implicit, geometry-aware sampling model-predictive control using Signed Distance Function (SDF) for guidance
- Push from start $\rightarrow$ goal with *no re-training* for new shapes, *no blind sampling*, minimal manual constraints, and no task-specific data

Problem Formulation

Objective: Given a robot's initial configuration q_0 and target object's signed distance function $\phi(p)$, find a sequence of controls u_0 to relocate an object from start pose T_0 to goal pose T^*

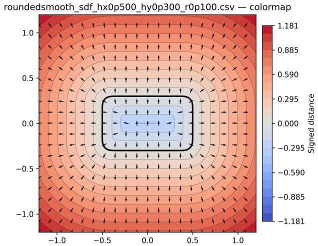
Given:

- T_0 — Initial object pose
- T^* — Goal object pose
- q_0 — Initial robot configuration
- u_0 — Initial robot controls (i.e. velocities,torques)
- $\phi(p)$ — Signed distance function (SDF) of the target object

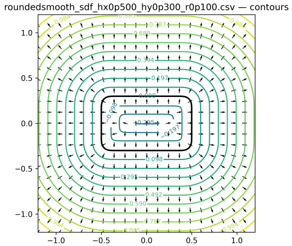
Problem Setup

SDF sign convention

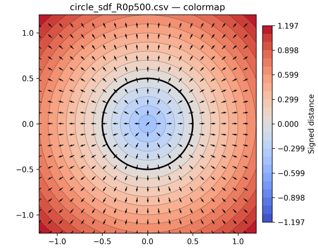
$\phi(\mathbf{p}) > 0$ outside the object,
$\phi(\mathbf{p}) = 0$ on the surface,
$\phi(\mathbf{p}) < 0$ inside.



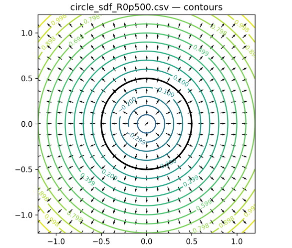
Rounded box — field



Rounded box — contour



Circle — field



Circle — contour

Method Overview

Object-side optimization

$$\min_{\{c_{t:T}\},\{\lambda_{t:T}\},\{T_{t:T}\}} \quad J_O(T_0, T^*, c, \lambda)$$

$$\text{s.t.} \quad \begin{aligned} & f(c,\lambda) = 0 \\ & g_i(c,\lambda) \geq 0, \; i \in \mathcal{I} \\ & g_i(c,\lambda) = 0, \; i \in \mathcal{E} \\ & 0 \leq \phi(c) \;\perp\; \lambda \geq 0 \end{aligned}$$

Where (object):

- J_O object cost function
- T_0 (initial object pose), T^* (goal object pose)
- c_t (contact point), λ_t (contact force)
- $f(\cdot)$ (object's dynamics), $g(\cdot)$ (constraints)
- $\phi(\cdot)$ (object's signed distance function)

Robot-side optimization

$$\min_{\{u_{t:T}\}} \quad J_R(T_0, T^*, c, \lambda, \phi(\cdot), q_0, u_0)$$

$$\text{s.t.} \quad \begin{aligned} & u_{\min} \leq u_t \leq u_{\max} \\ & q_{\min} \leq q_t \leq q_{\max} \\ & \|v_t\| \leq v_{\max} \end{aligned}$$

Where (robot):

- J_R robot cost function
- T_0 (initial object pose), T^* (goal object pose)
- c_t (contact point), λ_t (contact force)
- $\phi(\cdot)$ (object's signed distance function)
- q_0 (initial joints), u_0 (initial joint controls)

Robot-level Optimization

MPC Formulation

1. **Problem Setup**
   - State dynamics: $x_{t+1} = f(x_t, u_t), \quad t = 0, \ldots, T-1$
   - Objective: $\min \ J = \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$
2. **Stage Costs**
   - Goal tracking: $\ell_{\text{goal}} = \|T_{\text{goal}} - T_{\text{obj},k}\|_2^2$
   - Stabilizers: effort ℓ_{eff}, smoothness ℓ_{smooth}
   - Speed caps near contact: ℓ_{spd}
   - End-Effector Pose: ℓ_{rpy}
3. **Closed-Loop Receding Horizon**
   3.1 Sample K control sequences $\{u_{t:t+T-1}^{(k)}\}$
   3.2 Roll out dynamics
   3.3 Score with cost terms
   3.4 Update parameters & controls $\{u_{t:t+T-1}^{(k)}\}$
4. **Safety & Constraints**
   - Hard bounds on u_t and q_t

MPC Formulation

1. **Problem Setup**
   - State dynamics: $x_{t+1} = f(x_t, u_t), \quad t = 0, \ldots, T-1$
   - Objective: $\min \; J = \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$
2. **Stage Costs**
   - Goal tracking: $\ell_{\text{goal}} = \|T_{\text{goal}} - T_{\text{obj},k}\|_2^2$
   - Stabilizers: effort ℓ_{eff}, smoothness ℓ_{smooth}
   - Speed caps near contact: ℓ_{spd}
   - End-Effector Pose: ℓ_{rpy}
3. **Closed-Loop Receding Horizon**

   3.1 Sample K control sequences $\{u^{(k)}_{t:t+T-1}\}$

   3.2 Roll out dynamics

   3.3 Score with cost terms

   3.4 Update parameters & controls $\{u^{(k)}_{t:t+T-1}\}$
4. **Safety & Constraints**
   - Hard bounds on u_t and q_t

MPC Formulation

1. **Problem Setup**
   - State dynamics: $x_{t+1} = f(x_t, u_t), \quad t = 0, \ldots, T-1$
   - Objective: $\min \; J = \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$
2. **Stage Costs**
   - Goal tracking: $\ell_{\text{goal}} = \|T_{\text{goal}} - T_{\text{obj},k}\|_2^2$
   - Stabilizers: effort ℓ_{eff}, smoothness ℓ_{smooth}
   - Speed caps near contact: ℓ_{spd}
   - End-Effector Pose: ℓ_{rpy}
3. **Closed-Loop Receding Horizon**
   - 3.1 Sample K control sequences $\{u^{(k)}_{t:t+T-1}\}$
   - 3.2 Roll out dynamics
   - 3.3 Score with cost terms
   - 3.4 Update parameters & controls $\{u^{(k)}_{t:t+T-1}\}$
4. **Safety & Constraints**
   - Hard bounds on u_t and q_t

MPC Formulation

1. **Problem Setup**
   - State dynamics: $x_{t+1} = f(x_t, u_t), \quad t = 0, \ldots, T-1$
   - Objective: $\min \; J = \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T)$
2. **Stage Costs**
   - Goal tracking: $\ell_{\text{goal}} = \|T_{\text{goal}} - T_{\text{obj},k}\|_2^2$
   - Stabilizers: effort ℓ_{eff}, smoothness ℓ_{smooth}
   - Speed caps near contact: ℓ_{spd}
   - End-Effector Pose: ℓ_{rpy}
3. **Closed-Loop Receding Horizon**
   3.1 Sample K control sequences $\{u_{t:t+T-1}^{(k)}\}$
   3.2 Roll out dynamics
   3.3 Score with cost terms
   3.4 Update parameters & controls $\{u_{t:t+T-1}^{(k)}\}$
4. **Safety & Constraints**
   - Hard bounds on u_t and q_t

Model Predictive Path Integral (MPPI) [1]

1. **Idea:** Treat control optimization as a weighted average over noisy rollouts
2. **Algorithm:**
   - Sample K control sequences: $\{u^{(k)}_{t:t+T-1}\}$
   - Roll out dynamics: $x_{t+1} = f(x_t, u_t)$
   - Evaluate trajectory cost $J^{(k)} = \sum \ell(x_t, u_t)$
   - Weight by exponential transform: $w^{(k)} = \exp\Big(-\frac{1}{\lambda} J^{(k)}\Big)$
   - Update control: $u_t \leftarrow \frac{\sum_k w^{(k)} u^{(k)}_t}{\sum_k w^{(k)}}$
3. **Properties:**
   - No need for gradients $\Rightarrow$ robust to nonsmooth costs
   - Naturally parallelizable (GPU rollouts)
   - Handles nonconvex landscapes

[1] Williams, G., Aldrich, A., & Theodorou, E. A. (2015). *Model Predictive Path Integral Control using Covariance Variable Importance Sampling*. arXiv:1509.01149.

Model Predictive Path Integral (MPPI)

Reward Design

1. **Goal Shaping** $J_{\text{goal}} = \|T_{\text{goal}} - T_{\text{obj}}\|_2$
2. **Smoothness & Effort** $J_{\text{eff}} = \|u_t\|_2^2, \quad J_{\text{smooth}} = \|\Delta u_t\|_2^2$
3. **Distance-Adaptive Speed Cap**

$$d_t = \max\{0,\ \phi(p_t^{\text{ee}}; T_{\text{obj}})\} \quad \text{(object SDF)}$$

$$v_{\text{cap}}(d_t) = v_{\text{near}} + (v_{\text{far}} - v_{\text{near}})\,\sigma\left(\frac{d_t - d_{\text{near}}}{\tau}\right), \quad \sigma(z) = \tfrac{1}{2}(1 + \tanh z)$$

$$\boxed{J_{\text{spd}} = \left[\max(0,\ \|v_t^{\text{ee}}\| - v_{\text{cap}}(d_t))\right]^2}$$

4. **End-Effector Uprightness (roll/pitch)** $J_{\text{rpy}} = \phi_t^2 + \theta_t^2$

Total: $J = w_{\text{goal}} J_{\text{goal}} + w_{\text{eff}} J_{\text{eff}} + w_{\text{smooth}} J_{\text{smooth}} + w_{\text{speed}} J_{\text{speed}} + w_{\text{rpy}} J_{\text{rpy}}$

Where: $v_{\text{far}} > v_{\text{near}} > 0$; d_{near} is where slowing begins; τ controls transition sharpness

Object-level Optimization

Object Modeling

1. **Dynamic Model (Equality Constraints)**

$$x_{t+1} = f(x_t, u_t), \quad x = [p_x, p_y, \theta], \quad u = [c_x, c_y, \lambda]$$

2. **Contact Constraints (Inequalities)**

$$\phi(x) \geq 0, \quad \lambda \geq 0, \quad -\phi(x)\,\lambda \geq 0$$

3. **Initial Constraints**

$$x(0) = x_0$$

Object Kinematics and Dynamics (Equality Constraints)

State & control (per step t):

$$x_t = \begin{bmatrix} p_{x,t} \\ p_{y,t} \\ \theta_t \end{bmatrix}, \qquad u_t = \begin{bmatrix} c_{x,t} \\ c_{y,t} \\ \lambda_{1-4,t} \end{bmatrix} \text{ (analytical)}$$

with objects center p, object orientation θ, contact location c and applied force λ

Discrete-time dynamics (explicit Euler):

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \Delta t\, \dot{\mathbf{p}}_t, \qquad \theta_{t+1} = \theta_t + \Delta t\, \dot{\theta}_t,$$

where $\mathbf{p}_t = [p_{x,t}, p_{y,t}]^\top$ and time step Δt

Forces/torque from contact:

$$\dot{\mathbf{p}}_t = \frac{1}{\mu m g}\, \mathbf{F}_t^{\text{world}}, \qquad \dot{\theta}_t = \frac{\tau_t}{J_z}$$

where J_z is angular scalar $1/(\mu m g)$ (m-mass, μ-friction, and g-gravity)
Equality constraints (per $t = 0{:}T-1$):

$$g_t^{\text{dyn}}(x_t, u_t) \;=\; \begin{bmatrix} \mathbf{p}_{t+1} - \mathbf{p}_t - \Delta t\, \dot{\mathbf{p}}_t \\ \theta_{t+1} - \theta_t - \Delta t\, \dot{\theta}_t \end{bmatrix} = \mathbf{0}.$$

Object Dynamics Model (Analytical)

Analytical "face" model uses face-aligned direction

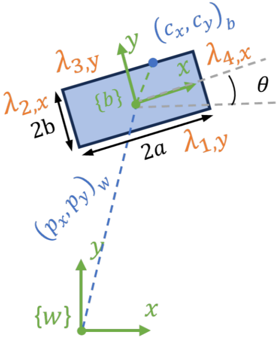
$$\underbrace{\mathbf{F}_t^{\text{body}}}_{\in\mathbb{R}^2} = \begin{bmatrix}\lambda_{2,t}+\lambda_{4,t}\\ \lambda_{1,t}+\lambda_{3,t}\end{bmatrix}, \qquad \mathbf{F}_t^{\text{world}} = R(\theta_t)\,\mathbf{F}_t^{\text{body}}$$

$$\tau_t = \underbrace{c_{x,t}}_{r_x}\underbrace{(\mathbf{F}_t^{\text{body}})_y}_{\lambda_{1,t}+\lambda_{3,t}} - \underbrace{c_{y,t}}_{r_y}\underbrace{(\mathbf{F}_t^{\text{body}})_x}_{\lambda_{2,t}+\lambda_{4,t}}$$

Where:

$$R(\theta) = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$$

Problem Setup[1]



[1]Image adapted from Li et al. (2025), *CRISP*, arXiv:2502.01055.

Object Dynamics Model (SDF)

SDF-normal model for arbitrary shapes

$$\textbf{(controls)} \quad u_t = \begin{bmatrix} c_{x,t} \\ c_{y,t} \\ \lambda_t \end{bmatrix}$$

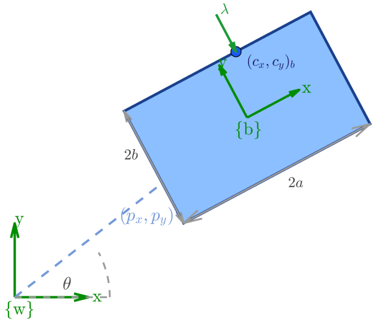
$$\textbf{(normal)} \quad \mathbf{n}_t^b = -\frac{\nabla\phi(\mathbf{c}_t)}{\|\nabla\phi(\mathbf{c}_t)\|}$$

$$\textbf{(linear force)} \quad \mathbf{F}_t^b = \lambda_t\,\mathbf{n}_t^b, \quad \mathbf{F}_t^w = R(\theta_t)\,\mathbf{F}_t^b$$

$$\textbf{(torque about COM)} \quad \tau_t = \left(c_{x,t}n_{y,t}^b - c_{y,t}n_{x,t}^b\right)\lambda_t$$

Where: the sign on $\mathbf{n}_k^b$ is flipped so that SDF gradient points inward.

Problem Setup



$\{w\}$ world frame $\quad$ $\{b\}$ body frame

Object Shape and Contact

SDF sign convention
$\phi(\mathbf{p}) > 0$ outside the object
$\phi(\mathbf{p}) = 0$ on the surface
$\phi(\mathbf{p}) < 0$ inside

Contact constraints (per step t)

$$\phi(\mathbf{c}_t) \geq 0, \quad \lambda_t \geq 0, \quad \lambda_t \, \phi(\mathbf{c}_t) \approx 0$$

(non-penetration, nonnegative normal force, force only at contact; "≈ 0")

Initial constraints

$$x(0) = x_0$$

(enforce start pose and any prescribed initial contacts)

Object-Level Optimization via CRISP [1]

Problem Formulation (Eq. 6)

$$\min_{c,\lambda} \; J(c,\lambda)$$

s.t. $f(c,\lambda) = 0$ (dynamics/defects)

$g_i(c,\lambda) \geq 0, \; i \in \mathcal{I}$ (ineq.)

$g_i(c,\lambda) = 0, \; i \in \mathcal{E}$ (eq.)

$0 \leq \phi(c,\lambda) \;\perp\; \lambda \geq 0$ (complementarity)

Where:

- $x = (v, \lambda)$ trajectory and contact forces
- $\phi(\cdot)$ is the gap function (SDF)
- $\lambda \geq 0$ are normal forces
- f, g_i dynamics and other constraints

Merit function (weighted μ) (Eq. 8)

$$\varphi_1(x;\mu) \;=\; J(x) + \sum_{i\in\mathcal{E}} \mu_i \, |g_i(x)| + \sum_{i\in\mathcal{I}} \mu_i \, [g_i(x)]^-,$$

$$[g]^- = \max\{0, -g\}, \qquad \mu_i > 0$$

Algorithm sketch (per iteration t)

1. Linearize f, g at $x^{(t)}$; build trust-region SQP
2. Handle complementarity via penalties in φ_1
3. Solve for step $p^{(t)}$; compute predicted vs. actual decrease of φ_1
4. Accept/Reject step; shrink/expand trust region
5. Update penalties $\mu \uparrow$ for violated constraints
6. Check convergence: merit decrease + constraint violation/KKT residual

[1] Li, Y., et al. (2025). *On the Surprising Robustness of Sequential Convex Optimization for Contact-Implicit MP*

CRISP: Optimization — SQP with Merit (I)

Build the subproblem (linearize constraints at $x^{(k)}$):

$$\min_{p_k} \ \tfrac{1}{2} p_k^\top H^{(k)} p_k + \nabla J(x^{(k)})^\top p_k \quad \text{s.t.} \quad \begin{aligned} & f(x^{(k)}) + \nabla f(x^{(k)})\, p_k = 0, \\ & g(x^{(k)}) + \nabla g(x^{(k)})\, p_k \geq 0, \\ & \|p_k\| \leq \Delta^{(k)}. \end{aligned}$$

CRISP quadratic model of the merit:

$$\min_{p_k} \ \underbrace{J_k + \nabla J_k^\top p_k + \tfrac{1}{2} p_k^\top \nabla^2_{xx} J_k \, p_k}_{\text{quadratic model of } J} + \sum_{i \in \mathcal{E}} \mu_i \left| g_i(x_k) + \nabla g_i(x_k)^\top p_k \right| + \sum_{i \in \mathcal{I}} \mu_i \left[g_i(x_k) + \nabla g_i(x_k)^\top p_k \right]^-$$

$$\text{s.t.} \quad f(x_k) + \nabla f(x_k)\, p_k = 0, \qquad \|p_k\| \leq \Delta^{(k)}.$$

$J_k, \nabla J_k, \nabla^2_{xx} J_k$ evaluated at x_k.

CRISP: Optimization — Local Merit Model (II)

Step acceptance via merit ratio:

$$\text{pred} = \varphi_1(x^{(k)};\mu) - m^{(k)}(p_k), \quad \text{ared} = \varphi_1(x^{(k)};\mu) - \varphi_1(x^{(k)}+p_k;\mu), \quad \rho = \frac{\text{ared}}{\text{pred}}$$

Trust-region (γ) acceptance & radius update:

$$x^{(k+1)} = \begin{cases} x^{(k)} + p_k, & \rho \geq \eta \\ x^{(k)}, & \rho < \eta \end{cases} \qquad \Delta^{(k+1)} = \begin{cases} \gamma_{\uparrow}\Delta^{(k)}, & \rho \text{ large} \\ \gamma_{\downarrow}\Delta^{(k)}, & \rho \text{ small} \\ \Delta^{(k)}, & \text{otherwise} \end{cases}$$

Penalty update & convergence:

- Increase μ_i for constraints whose violations stagnate (cap at $\mu_{\max}$)
- Stop when merit decrease and KKT/violations are small (exact-penalty $\to$ near a KKT point)

Complementarity handled via relaxation/penalty inside φ_1 or an nonlinear complementarity problem (NCP)

Reward Design from SDF + CRISP

1. **SDF for Geometric Safety**
   - **Collision & Clearance** $J_{\text{obs}} = \sum \max(0, d_{\text{safe}} - \phi(c_t))$
   - **Contact Directionality** Encourage pushes aligned with surface normal $\nabla\phi$
   - **Penetration Handling** Large penalty for $\phi(c_t) < 0$ (inside obstacle)
2. **Trajectory-based Reward from CRISP**
   Compare predicted contact-point trajectories with object trajectory from MPPI rollout $\Rightarrow$ reward signal for control sequence
3. **Hybrid Reward**
   $$J = \alpha J_{\text{CRISP}} + (1 - \alpha) J_{\text{SDF}}$$
4. **Benefits**
   - CRISP: task-aware, interpretable phases
   - SDF: geometry-aware, safe by design
   - Combined: structured and robust

Reward Design from SDF + CRISP

1. **SDF for Geometric Safety**
   - **Collision & Clearance** $J_{\text{obs}} = \sum \max(0, d_{\text{safe}} - \phi(c_t))$
   - **Contact Directionality** Encourage pushes aligned with surface normal $\nabla\phi$
   - **Penetration Handling** Large penalty for $\phi(c_t) < 0$ (inside obstacle)
2. **Trajectory-based Reward from CRISP**
   Compare predicted contact-point trajectories with object trajectory from MPPI rollout $\Rightarrow$ reward signal for control sequence
3. **Hybrid Reward**

$$J = \alpha J_{\text{CRISP}} + (1-\alpha) J_{\text{SDF}}$$

4. **Benefits**
   - CRISP: task-aware, interpretable phases
   - SDF: geometry-aware, safe by design
   - Combined: structured and robust

Reward Design from SDF + CRISP

1. **SDF for Geometric Safety**
   - **Collision & Clearance** $J_{\text{obs}} = \sum \max(0, d_{\text{safe}} - \phi(c_t))$
   - **Contact Directionality** Encourage pushes aligned with surface normal $\nabla\phi$
   - **Penetration Handling** Large penalty for $\phi(c_t) < 0$ (inside obstacle)
2. **Trajectory-based Reward from CRISP**
   Compare predicted contact-point trajectories with object trajectory from MPPI rollout $\Rightarrow$ reward signal for control sequence
3. **Hybrid Reward**

$$J = \alpha J_{\text{CRISP}} + (1 - \alpha) J_{\text{SDF}}$$

4. **Benefits**
   - CRISP: task-aware, interpretable phases
   - SDF: geometry-aware, safe by design
   - Combined: structured and robust

Reward Design from SDF + CRISP

1. **SDF for Geometric Safety**
   - **Collision & Clearance** $J_{\text{obs}} = \sum \max(0, d_{\text{safe}} - \phi(c_t))$
   - **Contact Directionality** Encourage pushes aligned with surface normal $\nabla\phi$
   - **Penetration Handling** Large penalty for $\phi(c_t) < 0$ (inside obstacle)
2. **Trajectory-based Reward from CRISP**
   Compare predicted contact-point trajectories with object trajectory from MPPI rollout $\Rightarrow$ reward signal for control sequence
3. **Hybrid Reward**
   $$J = \alpha J_{\text{CRISP}} + (1 - \alpha) J_{\text{SDF}}$$
4. **Benefits**
   - CRISP: task-aware, interpretable phases
   - SDF: geometry-aware, safe by design
   - Combined: structured and robust

Task: Push Rectangle

Task: Push Box

Task: Push Box

Task: Push Box SDF

Task: Pull Circle SDF

Task: Push Circle SDF

Approach: Chart

Task A: Push Cube with Cube

Objective

Translate the object of interest to the goal with minimal collisions and in a short time

Task: MPPI

Task: MPPI + CRISP without tip

Task: MPPI + CRISP with tip

Task: MPPI + CRISP with tip

Task: MPPI + CRISP challange MPPI

Task: MPPI + CRISP challange MPPI

Task: MPPI + CRISP challange CRISP

Task: MPPI + CRISP SDF

Task B: Push Cube with Xarm6

Objective

Perform **end-effector mediated pushing** of the cube while respecting manipulator kinematic limits

Key Challenges

- **Link clearance** — avoid collisions along arm geometry
- **Contact placement** — selecting stable push points
- **Wrist alignment** — aligning end-effector normal with surface

Task: MPPI

Task: CRISP + IK

Evaluation Protocol

Metrics

- **Success rate** — fraction of trials reaching goal within some translation and orientation margin
- **Time-to-goal** — execution time until task completion
- **Path length** — distance traveled by object/EE
- **Peak penetration** — maximum signed distance violation
- **Constraint violations** - equality and inequality constraint violation
- **Contact smoothness** — stability of forces/trajectories

Comparisons

- Baselines: (i) CRISP + IK, (ii) MPPI SDF-only, (iii) CRISP+MPPI
- Ablations: horizon T, samples K, temperature λ, clearance margin, normal-alignment

Runtime Reporting

- Average latency per step
- Wall-clock task completion time

Quantitative Results

- **Success vs. samples** K: MPPI $\Rightarrow$ higher K improves success
- **Baseline comparison**: (i) CRISP + IK, (ii) MPPI SDF-only, (iii) CRISP+MPPI $\Rightarrow$ CRISP improves robustness by guiding the contact, resulting in fewer failure modes
- **Runtime scaling**: parallel rollouts (CPU/GPU) $\Rightarrow$ near real-time performance

Qualitative Results

- Trajectories of contact points and object motion
- Surface normals aligned at contact
- SDF slices show safe clearance and penetration handling

Limitations & Failure Modes

Observed Issues

- **Suboptimal CRISP trajectory:**
  CRISP could provide a suboptimal path of the contact trajectory that leads to a longer way, even when obstacles are not present
- **MPPI:**
  MPPI samples do not have to be optimal and are highly dependent on the samples and the horizon
- **CRISP + MPPI:**
  Leads to a decent trajectory, but the reward sometimes causes the MPPI to focus too much on following CRISP rather than improving the initial goal objective

Takeaway

Robustness improves with CRISP+MPPI, but failure modes remain in cases where the contact has to produce a large force to move the object

Conclusion

Key Insights

- **Contact-implicit planning is hard:** non-convex geometry, mode switches, friction cones
- **Signed Distance Functions (SDF):** provide distances and normals $\nabla\phi$ for smooth, geometry-aware rewards
- **Sample-based control (MPPI):** robust in non-convex landscapes, avoids mode enumeration
- **CRISP + MPPI:** combines structured task guidance with geometric safety $\Rightarrow$ higher success rates

Takeaway

CRISP provides a **guided contact trajectory**, which MPPI follows using SDF-based rewards (goal distance, clearance, alignment). Together, they yield contact-rich behaviors that are *structured, safer, and more effective*

Future Work

Near-Term Extensions

- Perform simulations on **XArm6** with SDF-based MPC
- Extend to **multi-object tasks**, bimanual setups, and mobile manipulators

Modeling Improvements

- Upgrade CRISP with SDF and inertia-aware dynamics for general shapes
- Explore learned/neural SDFs for richer geometry
- Incorporate tactile sensing for normal estimation

Towards Real Systems

- **Sim-to-real transfer:** domain randomization, online SDF updates
- Extend to grasping and re-grasping with contact switching

Reachy 2

Thank You!

Questions?

nmarinosraitsevits@ucsd.edu