Contact-Aware Non-prehensile Manipulation for Object Retrieval in Cluttered Environments

Yongpeng Jiang†, Yongyi Jia†, and Xiang Li

***Abstract*— Non-prehensile manipulation methods refer to manipulation without grasping (i.e., utilizing unilateral contacts). Compared to the grasping method, more compact and flexible end effectors can be applied, making it possible to perform tasks in a constrained workspace; As a trade-off, it has relatively few degrees of freedom (DoFs), resulting in an under-actuation problem with complex constraints for planning and control. This paper proposes a new non-prehensile manipulation method for the task of object retrieval in cluttered environments, using a rod-like pusher. Different from existing methods, the proposed approach is with the contact-aware feature, which enables the synthesized effect of active removal of obstacles, avoidance behavior, and switching contact face for improved dexterity. Hence both the feasibility and efficiency of the task are greatly promoted. The performance of the proposed method is validated in a planar object retrieval task, where the target object, surrounded by many fixed or movable obstacles, is manipulated and isolated. Simulation and experimental results are presented.**

I. INTRODUCTION AND RELATED WORKS

Manipulation in clutter is a skill commonly demanded in daily life and production. Such a task is challenging for a robot manipulator because the dexterity of the robot end effector is often restricted by the cluttered environment and unknown object properties. For example, the stable grasp pose might be occluded by surrounding obstacles; Another example is that fragile or heavy objects are generally dangerous to lift up. Non-prehensile manipulation proposed by Mason [1] only requires no penetration constraints and does not rely on stable grasping [2], which is suitable for manipulation in cluttered environments.

This paper considers a representative scenario in the problem of non-prehensile manipulation, that is, retrieving a target object from clutter with a single rod-like pusher overhead, as seen in Fig. 1. To achieve it, the pusher should contact and move the object (i.e., the planar slider) to the goal location in the presence of multiple obstacles. Such a task is not trivial, and the challenges include complex kinodynamic constraints of the underactuated hybrid system, and the narrow corridor problem which restricts solving efficiency.

To address the problems above, existing methods add extra constraints to reduce the search space, such as demanding the contact mode to be consistent [3] or limiting the slider's movement to a particular pattern (i.e., Dubins path) [4].

† Equal contribution

Y. Jiang, Y. Jia, and X. Li are with the Department of Automation, Tsinghua University, China. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4701401, in part by National Natural Science Foundation of China under Grant U21A20517 and 52075290, and in part by the Beijing National Research Center for Information Science and Technology. Corresponding author: Xiang Li (xiangli@tsinghua.edu.cn)

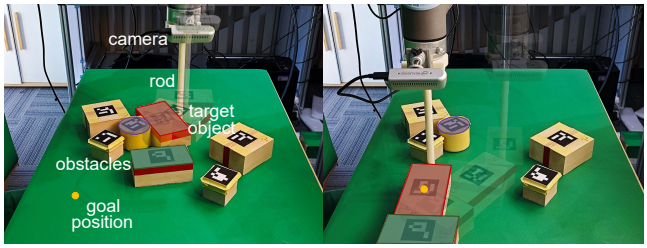


Fig. 1. **Object retrieval task through planar pushing.** The target object in red is separated from the clutter with a rod-like pusher through pushing manipulation. **Left**: The purple cylinder is pushed aside and the target object quickly navigates through fixed obstacles on both sides. **Right**: the cyan cube is pushed away and the target object bursts through the clutter.

However, such methods fail to sufficiently explore the state space, which might affect the solution quality. Besides, most existing works consider avoidance of simple obstacles [5] or implicitly assume an open space is required [3], [6]. However, in cluttered environments, there might be no feasible path to the goal position if the manipulated object merely avoids obstacles, or the total efficiency is unacceptable as it might take a long time to complete all the avoidance.

To improve the feasibility and efficiency of the task in Fig. 1, this paper proposes a contact-aware non-prehensile manipulation method using the rod-like pusher, which integrates multiple actions of active removal of obstacles, avoidance behavior, and switching contact face to create a feasible path to the goal if it is not available at the beginning.

Such a contact-aware feature allows the robot to explore different actions to generate more opportunities in cluttered environments. Moreover, the simplification of pushing dynamics yields the design of reachable set and object interaction model, which efficiently guides motion planning. Simulation results and further robot experiments are presented to verify the effectiveness of the proposed method.

II. METHOD

The block diagram of the proposed framework is depicted in Fig. 2.

A. Pushing Task and Pushing Planner

1) Problem Formulation: This paper considers a planar workspace with the target object (also referred to as the planar slider) o^s and multiple fixed or movable obstacles denoted as $\mathcal{O}^t$ and $\mathcal{O}^m$, respectively. We assume all objects are convex polygons with known geometric parameters and initial poses. The target object is actuated by a single rod-like pusher p through frictional point contact. A graphical representation of the task is shown in Fig. 3.

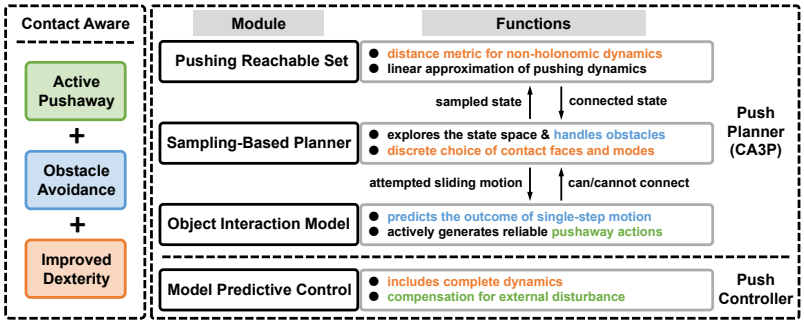


Fig. 2. Block diagram of the proposed framework (the planner and controller). Elements and implementations of the contact-aware feature are highlighted.

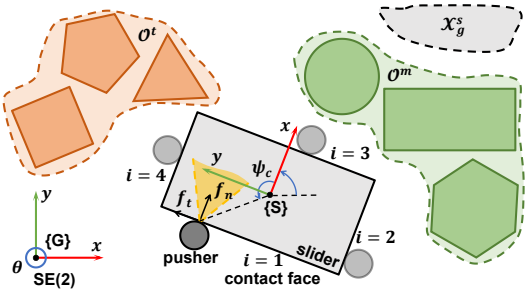


Fig. 3. Graphical representation of the planar object retrieval task, including the target object (planar slider) painted in grey, the goal region $\mathcal{X}_g^s$, movable obstacles $\mathcal{O}^m$ and fixed obstacles $\mathcal{O}^t$, $i \in \{1, 2, 3, 4\}$ denotes the discrete choice of contact face. Yellow shade represents the friction cone constraints.

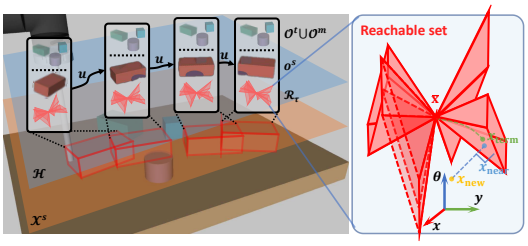


Fig. 4. The search tree and reachable set in CA3P. **Left**: Each node in tree $\mathcal{H}$ consists of the planning scene $\mathcal{O}^t \bigcup \mathcal{O}^m$, the state $\boldsymbol{x}^s \in \mathcal{X}^s$ of slider o^s and the corresponding reachable set $\mathcal{R}_\tau$. **Right**: the reachable set is composed of several convex cones. Generating state $\bar{\boldsymbol{x}}$ of the reachable set, the sampled state $\boldsymbol{x}_{\text{new}}$, nearest neighbor $\boldsymbol{x}_{\text{near}}$, and terminal state $\boldsymbol{x}_{\text{term}}$ reached by state connection are shown.

This paper considers actively utilizing contacts with the environment to create or amplify the pushing path. Hence, we consider solving the task in the joint state space $\mathcal{X}^E = \mathcal{X}^s \times \mathcal{X}_1^m \times \cdots \times \mathcal{X}_{|\mathcal{O}^m|}^m$, where $\times$ denotes the Cartesian product. A state $\boldsymbol{x}^E \in \mathcal{X}^E$ is called feasible if the fixed obstacles are not in contact with other objects (i.e., in case of turning over objects or getting stuck). The state transition is regulated by pushing dynamics (II-B.1) and the object interaction model (II-C).

2) Contact-Aware Pushing Planner: The **C**ontact-**A**ware **P**lanar **P**ush **P**lanner (CA3P) is outlined in Algorithm 1. Above all, the contact-aware feature could be divided into **three levels**. Specifically, the active avoidance and clearance of obstacles, and the contact face-switching technique to increase dexterity. The proposed algorithm a repeated process derived from the RRT planner. The search tree of RRT is stored in $\mathcal{H}$, and each node in $\mathcal{H}$ records a feasible pose of the target object o^s and the corresponding control input. We simultaneously track the changeable poses of $\mathcal{O}^m$ and the

Algorithm 1: CA3P

Input: State space $\mathcal{X}^s$, obstacles $\mathcal{O}^t, \mathcal{O}^m$, initial pose $\boldsymbol{x}[0]$, goal region $\mathcal{X}_g$
Output: Control sequence $\boldsymbol{u}[0:T-1]$
Parameters: Maximum number of nodes $n_{\max}$

1 **while not** maximum number of nodes exceeded **do**
2 Initialize search tree $\mathcal{H}$; $\mathcal{H}$.nodes.add $(\langle \boldsymbol{x}[0], \varnothing \rangle)$;
3 Randomly sample $\boldsymbol{x}_{\text{new}} \in \mathcal{X}^s$;
4 $\boldsymbol{x}_{\text{near}} \leftarrow \texttt{NearestNeighbor}\left(\boldsymbol{x}_{\text{new}}, \bigcup_{\boldsymbol{x} \in \mathcal{H}.\text{nodes}} \mathcal{R}_\tau(\boldsymbol{x})\right)$;
5 $\boldsymbol{x}_{\text{gen}} \leftarrow \texttt{GetGenerateState}(\boldsymbol{x}_{\text{near}})$;
6 $\boldsymbol{x}_{\text{term}}, \boldsymbol{u}_{\text{term}} \leftarrow \texttt{Connect}\,(\boldsymbol{x}_{\text{gen}}, \boldsymbol{x}_{\text{near}})$;
7 $\mathcal{O}_{\text{gen}}^t \bigcup \mathcal{O}_{\text{gen}}^m \leftarrow \texttt{GetEnviron}\,(\mathcal{H}.\text{nodes}, \boldsymbol{x}_{\text{gen}})$;
8 **if** $\texttt{Simulate}(\boldsymbol{x}_{\text{gen}}, \boldsymbol{u}_{\text{term}}, \mathcal{O}_{\text{gen}}^m, \mathcal{O}_{\text{gen}}^t)$ **then**
9 $\mathcal{H}$.nodes.add $(\langle \boldsymbol{x}_{\text{term}}, \boldsymbol{u}_{\text{term}} \rangle)$;
10 $\mathcal{H}$.edges.add $(\langle \boldsymbol{x}_{\text{gen}}, \boldsymbol{x}_{\text{term}} \rangle)$;
11 $\texttt{UpdateEnviron}\,(\mathcal{H}.\text{nodes}, \boldsymbol{x}_{\text{gen}}, \boldsymbol{u}_{\text{term}}, \mathcal{O}_{\text{gen}}^m)$;
12 **if** $\texttt{Connect}(\boldsymbol{x}_{\text{term}}, \mathcal{X}_g)$ **then**
13 **return** $\texttt{ExtractPath}\,(\mathcal{X}_g, \mathcal{H})$;
14 **end**
15 **end**
16 **end**
17 **return** $\varnothing$;

poses of $\mathcal{O}^t$, which is referred to as the **planning scene**. The structure of CA3P search tree is depicted in Fig. 4 **Left**.

Since the pushing system is subject to non-holonomic constraints, the Euclidean distance works poorly; we adopt the concept of reachable sets [7] as a distance metric. Existing works have shown promising effects of this concept in prehensile grasping [8], [9], while we extend the approach to non-prehensile pushing. In Algorithm 1, `NearestNeighbor` computes the nearest state $\boldsymbol{x}_{\text{near}}$ in all reachable sets of states in $\mathcal{H}$ to a newly sampled state $\boldsymbol{x}_{\text{new}}$ (Line 4). We omit the superscript s of the target object for brevity. Then, `GetGenerateState` returns the **generating state** $\boldsymbol{x}_{\text{gen}}$ of the reachable set which contains $\boldsymbol{x}_{\text{near}}$ (Line 5). The notations and computation of reachable sets will be presented in Sec. II-B.2. Next, `Connect` calculates the control input $\boldsymbol{u}_{\text{term}}$ driving the system from $\boldsymbol{x}_{\text{gen}}$ to $\boldsymbol{x}_{\text{near}}$ (Line 6). Due to the linearization error of pushing dynamics, the rollout of $\boldsymbol{u}_{\text{term}}$ usually does not reach $\boldsymbol{x}_{\text{near}}$ exactly; We denote the actual terminal state as $\boldsymbol{x}_{\text{term}}$. The input matrix $\boldsymbol{B}$ will be derived from (1), the subscript i is also ignored for brevity.

The core procedure that enables the active clearance of

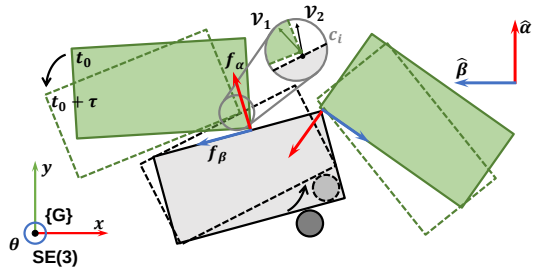


Fig. 5. Configurations of the multi-contact system in consecutive time steps t_0 (solid line border) and $t_0+\tau$ (dashed line border). At contact point c_i, the normal and tangential $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$, the contact forces $f_\alpha, f_\beta = f_{\beta+} - f_{\beta-}$ and the contact velocities $\mathcal{V}_1, \mathcal{V}_2$ are shown.

obstacles is `Simulate` (Line 8), which predicts the outcome of one-step pusher motion. Configuration of the slider and obstacles is updated through the object interaction model in Sec. II-C. The state connection is abandoned if collision with fixed obstacles is detected. Otherwise, the search tree is updated with new **planning scene** `UpdateEnviron`.

B. Pushing Reachable Sets

1) Pushing Dynamics: Under quasi-static assumptions, the system dynamics model can now be given as

$$\dot{\boldsymbol{x}}^s = \boldsymbol{f}_i\left(\boldsymbol{x}^s, \boldsymbol{u}^p\right) \triangleq \begin{bmatrix} \boldsymbol{R}\boldsymbol{A}\boldsymbol{J}_{c,i}^\top & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}^p, \tag{1}$$

where $\boldsymbol{x}^s, \boldsymbol{u}^p$ are the system state and input, $i = 1, \ldots, N$, N is the number of contact faces (see Fig. 3), $\boldsymbol{R}$ is the slider's rotation matrix, $\boldsymbol{A}$ is a positive definite force-motion mapping matrix, $\boldsymbol{J}_{c,i}$ is the contact jacobian.

Besides, the system satisfies workspace constraints $\boldsymbol{x}^s \in \mathcal{X}^s$, box constraints on the input variables and Coulomb friction constraints $\boldsymbol{u}^p \in \mathcal{U}^p$.

2) Computation of Reachable Sets: The reachable sets highlight the states more likely to be connected from the already explored state space. This technique can provide directional guidance to kinodynamic push planning. The reachable sets of arbitrary state $\bar{\boldsymbol{x}} \in \mathcal{X}$ is defined as the set of states reachable from $\bar{\boldsymbol{x}}$ within finite time horizon τ, under the system constraints:

$$\begin{aligned}\mathcal{R}_\tau(\bar{\boldsymbol{x}}) \triangleq \{&\boldsymbol{x} \in \mathcal{X} | \exists (x, u) : [0, t_0] \mapsto (\mathcal{X}, \mathcal{U}), t_0 \in [0, \tau], \\ &x(0) = \bar{\boldsymbol{x}}, x(t_0) = \boldsymbol{x}, \dot{x}(\xi) = \boldsymbol{f}_i\left(x(\xi), u(\xi)\right)\}.\end{aligned} \tag{2}$$

We call $\bar{\boldsymbol{x}}$ the **generating state** of the **reachable set** $\mathcal{R}_\tau(\bar{\boldsymbol{x}})$. The reachable set in this specific problem could be approximated by the union of convex sets, as shown in Fig. 4 **Right**. This special structure enables a discrete choice of contacts and allows computationally efficient methods for nearest neighbor search [7].

TABLE I

STATISTICAL RESULTS OF MOTION PLANNING

Scene		0	1	2	3	4
Success	CA3P	26/30	25/30	29/30	29/30	30/30
	CA3P-s	28/30	28/30	30/30	28/30	30/30
	RRTc	19/30	7/30	30/30	30/30	30/30
Planning Time (s)[1]	CA3P	5.73±5.17	8.20±6.64	4.45±2.34	10.66±11.57	4.35±1.94
	CA3P-s	6.90±5.84	15.95±10.36	4.36±1.86	8.88±7.87	6.44±4.53
Track Length (m)[1]	CA3P	0.57±0.08	0.70±0.08	0.55±0.06	0.72±0.09	0.60±0.04
	CA3P-s	0.55±0.06	0.70±0.07	0.54±0.05	0.67±0.08	0.61±0.09

[1] The results are provided as mean±standard deviation.

C. Object Interaction Model

The object interaction model is designed to forecast the motion of movable objects in contact with the planar slider, as a resolution to heavy physics engine. Inspired by [10], for each point of contact c_i in the system consisting of the slider and obstacles, the feasible contact force that satisfies the non-penetration constraints is equivalent to the solution to the Linear Complementarity Problem (LCP):

$$\left.\begin{matrix}\mathcal{C}_1 \\ \vdots \\ \mathcal{C}_{n_c}\end{matrix}\right\} \Leftrightarrow \begin{cases} \boldsymbol{z} = \boldsymbol{M}\left(\boldsymbol{x}^E, \boldsymbol{\Gamma}\right)\begin{bmatrix}\boldsymbol{f} \\ \boldsymbol{\lambda}\end{bmatrix} + \boldsymbol{N}\left(\boldsymbol{x}^E, \boldsymbol{u}^p, \boldsymbol{\Gamma}\right) \\ \boldsymbol{0} \preceq \boldsymbol{z} \perp \begin{bmatrix}\boldsymbol{f} \\ \boldsymbol{\lambda}\end{bmatrix} \succeq \boldsymbol{0} \end{cases}, \tag{3}$$

where $\boldsymbol{f}$ is the contact force, $\boldsymbol{z}, \boldsymbol{\lambda}$ are auxiliary variables, and $\boldsymbol{\Gamma}$ includes geometries and limit surface parameters, $\boldsymbol{M}, \boldsymbol{N}$ are nonlinear. The LCP (3) can be efficiently solved with Newton-based methods [11]. Then, the poses of movable objects can be updated through forward integration.

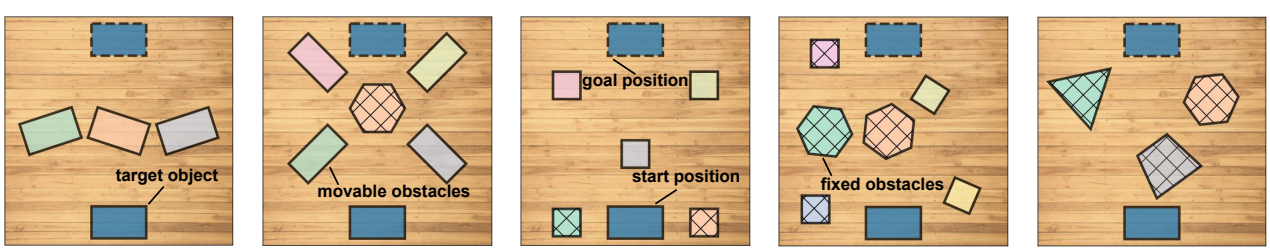
III. RESULTS

In this section, we show that CA3P generates a shorter path in less time compared with baselines, utilizing the contact-aware feature. Extended robot experiments are carried out with an integrated model predictive pushing controller.

A. Simulation Studies

The simulations were conducted on a 64-bit Intel Core i7-12700 4.9 GHz Ubuntu workstation with 32 GB RAM. We compared CA3P with the RRTc baseline proposed in [4]. RRTc is an RRT-based planner utilizing the differential flatness properties. The slider is manipulated through sticking contacts and forced to follow trajectories with constant curvature, i.e., Dubins path.

For CA3P, we set the goal sampling bias as 0.1. For all methods, we define the stopping criterion as maximum planning time 1×10^3 s or maximum number of nodes 1×10^3. A method will return failure if the goal region is unreachable from explored states when the criterion is met. To evaluate



(a) scene 0 (b) scene 1 (c) scene 2 (d) scene 3 (e) scene 4

Fig. 6. Five representative problem instances. Start and goal position of the target objects are painted with solid and dashed borders, respectively. Fixed obstacles are marked with net mesh. The tasks in **scene 0** and **scene 1** require the obstacles to be cleared away. The search process for **scene 2** and **scene 3** can be accelerated by pushing aside the obstacles, although not essential. **Scene 4** only needs obstacle avoidance.

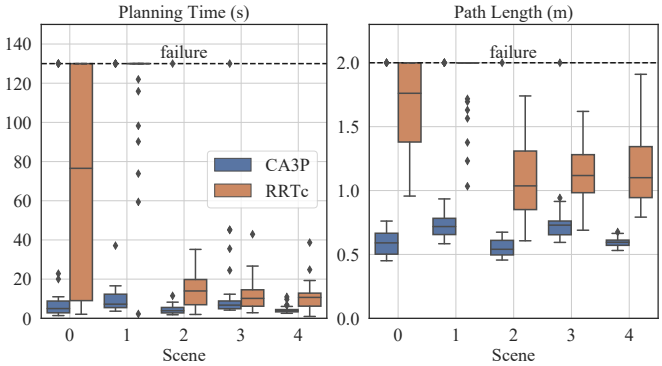


Fig. 7. Results of CA3P and RRTc on five instances across 30 trials. The planning time and path length are set as 130 s and 2.0 m for failure trials. The median, lower and upper quartiles, error bars, and outlier values are reported. Less planning time means the method is more efficient. A shorter path length means the result is more optimal.

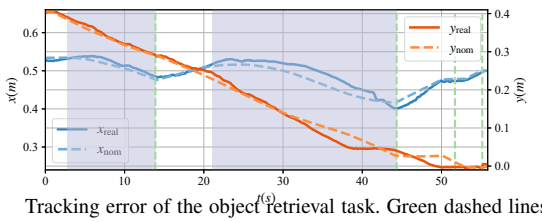


Fig. 8. Tracking error of the object retrieval task. Green dashed lines mark the moments of switching faces; purple shadows report the intervals of contact. Blue and orange curves represent the x and y dimensions, respectively.

the effectiveness of the proposed method, we generated 5 representative problem instances, as shown in Fig. 6. The planning time and total path length for each instance across 30 trials are shown in Fig. 7, other statistical results are presented in Table. I

The proposed CA3P greatly reduced planning time and generated shorter trajectories. Moreover, CA3P reports narrower interquartile ranges for all problem instances, indicating that the method achieves more stable performance with random scenes and trials. For more complicated scene 0 and scene 1, CA3P increased the success rate by 23% and 60%. RRTc completed these scenes by utilizing extra workspace at the expense of growth in the trajectory length. Results of scene 2 and scene 3 showed that actively removing obstacles is an effective way to obtain consistent path length; since avoiding obstacles is pretty demanding in the sampling sequence and quality. Results of scene 4 proved that the reachable set offers preferable directions in the search process.

In Table. I, CA3P-s simplifies the original algorithm by assuming the slider's motion is unaffected even when in contact. Simulations have found that CA3P-s improves success rate with no visible effect on the planned path. Thus CA3P-s will be adopted in Sec. III-B to suppress replanning.

B. Real-World Experiments

We implemented the robot experiments on a 64-bit Intel Core i7-9700 4.7 GHz Ubuntu workstation with 16 GB RAM. We mounted a $\Phi 15 \times 250\,\mathrm{mm}$ resin pusher on a UR5 robot. The perception system was composed of an Intel Realsense D435i camera and several ArUco markers.

Five keyframes of a complete planar object retrieval task are depicted in Fig. 9. The initial and goal positions of the planar slider are $\boldsymbol{x}^s[0] = [0.53\,\mathrm{m}, 0.41\,\mathrm{m}, -1.57\,\mathrm{rad}]^\top$ and $\boldsymbol{x}^s[T] = [0.50\,\mathrm{m}, 0.00\,\mathrm{m}, -3.14\,\mathrm{rad}]^\top$, respectively. The task scenerio contains fixed obstacles of $10.0 \times 10.2 \times 5.0\,\mathrm{cm}^3$ and $5.0 \times 5.0 \times 5.0\,\mathrm{cm}^3$, the additional cylindrical movable obstacle is of $\Phi 7 \times 6\,\mathrm{cm}^3$. Obstacles are simplified as their minimum bounding rectangles in CA3P, with an estimated frictional coefficient $\mu = 0.3$ between all pairs. One notable fact is that all friction-related parameters are not accurately measured. The MPC and relatively simple dynamics [12] suppressed the incurred error in sim-to-real transfer.

Since it is challenging to control contact forces directly, we converted the control input $\boldsymbol{u}^p$ to speed command. The pusher is initialized at the center of each contact face the robot has switched to. As shown in Fig. 9(a) and Fig. 9(b), the slider pushed the cylindrical object aside to enlarge the space ahead; instead of passively performing a time-consuming avoidance behavior. Later the slider passed through the narrow corridor and came into contact with another obstacle, as Fig. 9(c) depicts. We observed the fast-moving behavior of the pusher on the slider's periphery as an anti-disturbance mechanism (Fig. 9(d)). The slider eventually broke out of the clutter in 50.6 s (Fig. 9(e)). The tracking error in the x and y directions are reported in Fig. 8. When a large deviation from motion planning is detected, replanning is required for error recovery.

IV. CONCLUSIONS

This work proposes a new manipulation method for non-prehensile planar pushing in a constrained workspace. We combine sampling-based approaches with a simplified object interaction model for motion planning. With the use of those techniques together, the proposed planner CA3P is with the novel contact-aware feature, which allows the robot to actively avoid obstacles, switch contacts, or remove obstacles simultaneously. Effectiveness of the proposed method has been comprehensively validated in the task of object retrieval. Future works will be devoted to post-processing the planned trajectory through optimization-based methods; and to taking account of higher-order dynamics for preferable dynamic non-prehensile manipulation. Improving the method's scalability across masses of obstacles is also a meaningful direction.

(a) t =1.03 s (b) t =12.84 s (c) t =22.49 s (d) t =39.36 s (e) t =50.57 s

Fig. 9. Snapshots of the planar object retrieval task executed on a UR5 robot arm. The slider was manipulated to consecutively push aside a cylindrical (a-b) and a cubic obstacle (c-d) and was finally pushed to the goal position after switching contact faces. The time consumed on switching faces is ignored.

REFERENCES

[1] M. T. Mason, "Progress in nonprehensile manipulation," *The International Journal of Robotics Research*, vol. 18, pp. 1129 – 1141, 1999.

[2] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1711–1718, 2018.

[3] N. Doshi, F. R. Hogan, and A. Rodriguez, "Hybrid differential dynamic programming for planar manipulation primitives," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6759–6765, 2019.

[4] J. Zhou, Y. Hou, and M. T. Mason, "Pushing revisited: Differential flatness, trajectory planning, and stabilization," *International Journal of Robotics Research: Special Issue on ISRR '17*, vol. 38, no. 12, pp. 1477 – 1489, September 2019.

[5] J. Moura, T. Stouraitis, and S. Vijayakumar, "Non-prehensile planar manipulation via trajectory optimization with complementarity constraints," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 970–976, 2021.

[6] C.-Y. Chai, W. Peng, and S.-L. Tsao, "Object rearrangement through planar pushing: A theoretical analysis and validation," *IEEE Transactions on Robotics*, vol. 38, pp. 2703–2719, 2022.

[7] A. Wu, S. Sadraddini, and R. Tedrake, "R3t: Rapidly-exploring random reachable set tree for optimal kinodynamic planning of nonlinear hybrid systems," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4245–4251, 2020.

[8] N. C. Dafle, R. Holladay, and A. Rodriguez, "In-hand manipulation via motion cones," *ArXiv*, vol. abs/1810.00219, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:46994279

[9] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *ArXiv*, vol. abs/2206.10787, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:249926394

[10] J. Zhou, J. Bagnell, and M. Mason, "A fast stochastic contact model for planar pushing and grasping: Theory and experimental validation," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017.

[11] A. Fischer, "A newton-type method for positive-semidefinite linear complementarity problems," *Journal of Optimization Theory and Applications*, vol. 86, pp. 585–608, 1995.

[12] E. Huang, Z. Jia, and M. T. Mason, "Large-scale multi-object rearrangement," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 211–218, 2019.