Towards Collaborative Manipulation with Car-Like Robot Pushers

Shambhavi Singh[1] and Christoforos Mavrogiannis[1]

***Abstract*— We focus on collaborative manipulation with a team of car-like robot pushers. Pushing can be a practical manipulation strategy for rearranging large, heavy, or unstructured objects without needing grippers with high design complexity and cost. Prior work has focused on simplified problem instances, including prehensile manipulation using grippers, and pushing with holonomic robots. However, real-world applications of object rearrangement in construction, mining, or warehouses motivate the need to support manipulation of diverse objects and supply higher torques. Our key insight is to leverage nonprehensile manipulation to accommodate a wide range of object geometries, and, use car-like robot pushers to apply significantly higher torque than holonomic robots of comparable cost. The nonholonomic constraints imposed by car kinematics in conjunction with pushing-based constraints required for object controllability complicate planning, control, and coordination. To this end, we develop an architecture for planning the motion of multiple car-like robots to produce a desired object rearrangement. Given a goal pose for the object, we first extract a trajectory (sequence of twists) taking the object from its current pose to the goal. For each object twist, we solve an optimization instance to optimally distribute pushing forces and contact configurations among robots. We formulate the optimization as a quadratic programming problem and solve to minimize the magnitude of forces required for each object twist. Each robot executes the sequence of pushing forces that it was assigned in a decentralized fashion using a model predictive controller. Preliminary results validate our approach on four pushing scenarios each involving the rearrangement of a long rectangular object by two car-like robots. Ongoing work involves the evaluation of our architecture on hardware, using a team of 1/10th scale robot racecars.**

***Index Terms*—Pushing, Planar Manipulation, Multi-Robot Systems, Model-Based Optimization**

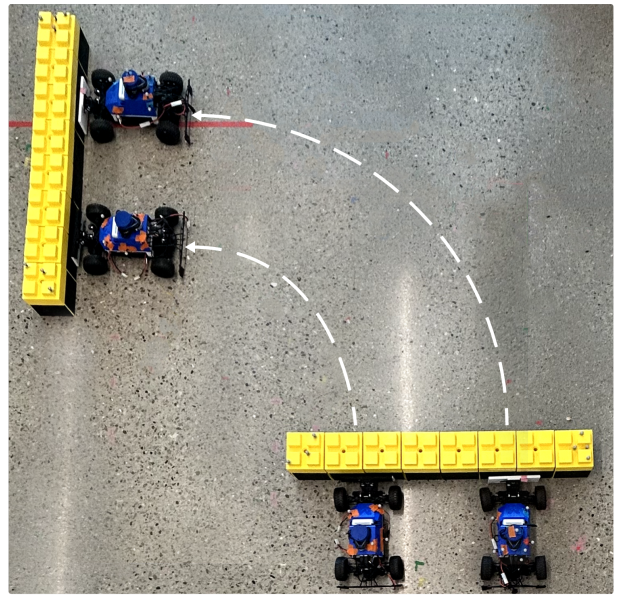

Fig. 1. Example setup of two car-like robots collaboratively pushing a long rectangular object along a curved path.

I. Introduction

Autonomous collaborative manipulation has the potential to transform how robots interact with large or heavy objects in environments like warehouses, construction sites, and factories. While prior approaches rely on holonomic robots or complex grippers, car-like robots offer a compelling alternative due to their higher torque and simpler mechanical design. Nonprehensile manipulation through planar pushing presents a robust strategy for object rearrangement, particularly for objects that are irregularly shaped, heavy, or too large for conventional grippers [1, 3, 6].

Mechanics of planar pushing: Mason (1982) and Peshkin and Sanderson (1988) develop the mechanics of pushing under quasistatic conditions where frictional forces on the object due to the surface μ_s quickly damp out any kinetic energy of the object. Goyal et al. (1991) define the convex boundary of frictional wrenches during contact as the limit surface. With this assumption, Lynch and Mason (1996) develop conditions for a mobile robot pushing an object with line-contact using stable pushes i.e. pushes without relative sliding [14]. Prior work has employed quasistatic stable pushing models to perform planar object rearrangement tasks [1, 19]. In our work, we devise the conditions for planar quasistatic pushing with multiple mobile robots using only stable pushes.

Model-based planning and control: Model-based optimization and data-driven approaches particularly reinforcement learning-based methods have been widely adopted for contact-rich pushing scenarios. Several works formulate the problem of finding pushing contacts and trajectories/forces as constraints of an optimization problem in single [2, 11, 13] or multi-robot scenarios [5, 12, 20] with centralized controllers. In contrast, prior works with decentralized multi-robot control define control laws based on pushing models where robots may take turns to execute pushing actions [16], follow a leader robot [21] or move as a formation in a swarm [4]. Although these works made great leaps in development of push manipulation abilities, their scope was limited to either small holonomic robots or manipulator arms. These works can

[1]Authors are with the Robotics Department, University of Michigan, Ann Arbor, MI, USA. {shasingh, cmavro}@umich.edu

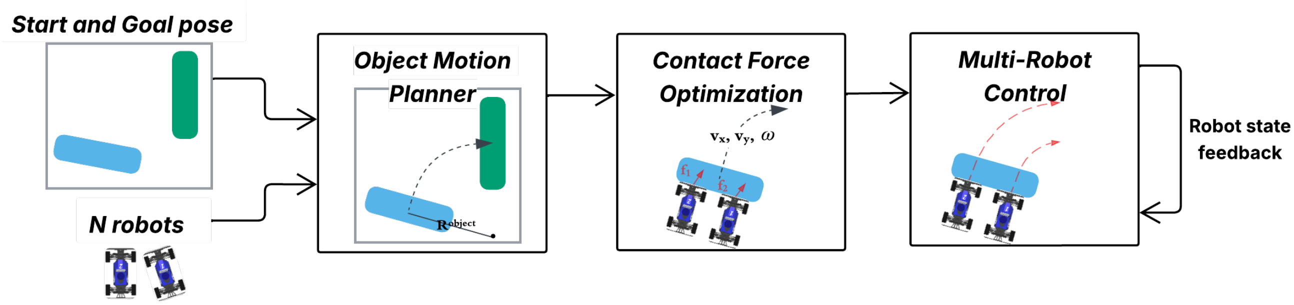

Fig. 2. System Architecture: Given an object start and goal pose, and N robots, we find a valid plan for pushing the object as a sequence of object twists. For each object twist, we distribute pushing forces among the robots. Finally, robots track a desired velocity profile using a closed loop multi-robot controller.

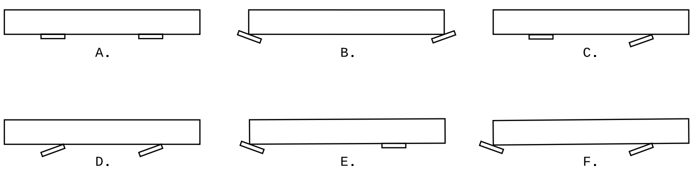

Fig. 3. Various configurations with two pusher robots (0.1m×0.02m) in multiple pushing contact modes with a large rectangular object (0.8m×0.1m).

approximate pushing contact as point-contact due to the small size of the end-effectors/robots compared to the object. In contrast, our work leverages car-like robots to deliver higher torque compared to holonomic robots of comparable cost. This, along with the front bumper of cars forming a line-contact, enables manipulation of objects that are too large, heavy or irregular in shape for point-contact pushing with holonomic robots.

Nonprehensile manipulation of large objects: Recent works highlight growing development of robots for manipulation of large objects. In [8], hierarchical reinforcement learning-based methods are used for obstacle-aware multi-robot manipulation of a large object by quadrupedal robots. Also, in [22] a differential-drive mobile robot navigates through a region cluttered with heavy movable obstacles using a pushing model from a physics engine to efficiently sample rollouts of their controller. Our work is a step towards collaborative non-prehensile manipulation where this reasoning is developed for multiple car-like robots. Our framework decomposes the collaborative pushing problem into three components: object motion planning, optimal force distribution among multiple robots, and low-level robot control. This approach enables manipulation of large objects with kinematically constrained car-like robots.

II. Approach

We formulate the problem for pushing an object in a planar workspace $W \subset SE(2)$ using two non-holonomic car-like robots or *pushers*. The 3D object has a mass M and moment of inertia I, a low center-of-gravity such that $\mathbf{o} = (x, y, \theta) \in W$ represents the object pose, and O represents the boundary of the object. We consider objects that are too heavy for a single robot but sufficiently large to allow multiple robots to push simultaneously. Although there are numerous configurations for pushing the object with car-like pusher robots as illustrated in Fig 3, for the scope of this work we focus on Fig 3-A. In configuration A, robots execute stable pushes maintaining *line-contact* with the object.

We represent the state of the robots as $\mathbf{p_1}, \mathbf{p_2} \in W$, the robot follows rear-axle simple-car kinematics:

$$\dot{\mathbf{p}}_\mathbf{i} = [s_i \cos(\theta_i), s_i \sin(\theta_i), s_i \tan(\phi_i)/L] \quad (1)$$

where $\mathbf{u_i}$ is the control input, s_i is the speed, ϕ_i is the steering angle, and L is the wheelbase of the robots with $i \in \{1, 2\}$. Additionally, we assume uniform pressure distribution across the object and uniform frictional properties, where μ_s and μ_c give the friction between the object and the support ground, and the object and the robot respectively.

A. Object Motion Planner

The path of the object consists of a sequence of stable pushes, and the space of stable pushing directions imposes non-holonomic constraints on the motion of the object[14]. We use a hybrid-A^* planner to construct stable pushing paths for the object among obstacles. Our planner generates a Dubins Curve[7] as the shortest path between the start and goal location using L, R, and S primitives corresponding respectively to left, right, and straight motion. The left and right motion primitives are calculated using a minimum turning radius that ensures stable pushing under the quasistatic assumption similar to the planner in [1].

Consider robots $\mathbf{p_1}, \mathbf{p_2} \in W$ pushing an object $\mathbf{o} \in W$ with a constant velocity. Tang et al. (2024) prove that any object transformation with a constant velocity can be represented as an arc transformation. Let the turning radius of the arc be R^{object} as illustrated in Fig. 4, and the contact lines be represented by $\mathbf{r_1}, \mathbf{r_2} \in W$ as the coordinates of their mid-points. Since each car performs stable pushes, the radius of curvature of each car must be larger than R^{car}_{min}. Here, R^{car}_{min} is the minimum radius for sticking contact with stable pushing. Thus, the minimum radius of curvature for the object must be offset such that each robot can perform stable pushes, it is given by:

$$R^{object}_{min} = max(R^{car}_{min} + \mathbf{r_1}.\hat{\imath} - \mathbf{o}.\hat{\imath}, \quad R^{car}_{min} + \mathbf{r_2}.\hat{\imath} - \mathbf{o}.\hat{\imath}) \quad (2)$$

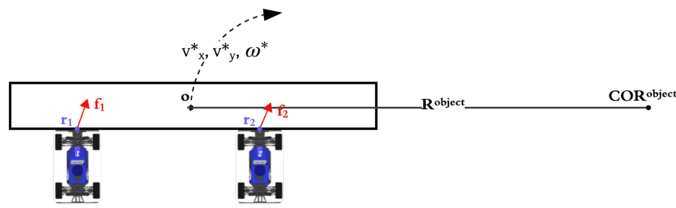

Fig. 4. The object at $o = (x, y, \theta)$ with the desired object twist (v_x^*, v_y^*, ω^*) traces an arc of radius R^{object} while robots apply forces f_1 and f_2 through contacts at r_1 and r_2 respectively.

where $\hat{\imath}$ is the unit vector along the X-axis. For the scope of this work, we use the farthest contact point $max(\mathbf{r_i}.\hat{\imath} - \mathbf{o}.\hat{\imath})$ to find the R^{object}_{min} that can allow robots anywhere along the edge of the object to complete that object transformation.

B. Force Distribution Optimization

For each desired object twist $\dot{\mathbf{o}}^* = (v_x, v_y, \omega)$ in the planned trajectory, we determine how to optimally distribute pushing forces among the robots. We formulate this as a Mixed-Integer Quadratically Constrained Programming (MIQCP) problem. Robots are allowed to push with their flat bumpers anywhere on the perimeter of the object, given that the line-contact does not intersect with a corner point of the object (to maintain configuration A from Fig. 3), or collide with another robot.

1) Friction Constraints: For two robots with sticking line-contacts, the configuration is defined as: $\xi \triangleq \mathbf{r}_1\mathbf{r}_2$ Further, for the i-th robot where $i \in \{1, 2\}$, the force $\mathbf{f}_i$ applied by the robot, can be decomposed into normal and tangential components: $\mathbf{f}_i \triangleq \mathbf{f}_i^{\mathrm{n}} + \mathbf{f}_i^{\mathrm{t}} \triangleq f_i^{\mathrm{n}}\mathbf{n}_i + f_i^{\mathrm{t}}\boldsymbol{t}_i$ where $\mathbf{n_i}$ and $\mathbf{t_i}$ represent the unit vectors and f_i^{n} and f_i^{t} represent the frictional force magnitudes in the normal and tangential directions respectively. With the following constraints due to Coulomb's law of friction for sticking contact:

$$0 \le f_i^{\mathrm{n}} \le f_{i,\max}; \quad 0 \le |f_i^{\mathrm{t}}| \le \mu_s f_i^{\mathrm{n}} \tag{3}$$

2) Generalized force for two robots: The total force applied by the two robots can be represented by $\mathbf{F}_\xi$ as:

$$\mathbf{F}_\xi \triangleq (\mathbf{F}_\xi^{\mathrm{n}}, \mathbf{F}_\xi^{\mathrm{t}}) \triangleq (f_1^{\mathrm{n}}, f_2^{\mathrm{n}}, f_1^{\mathrm{t}}, f_2^{\mathrm{t}}) \in \mathbb{R}^4$$

3) Object Dynamics: The combined generalized force applied on the object is:

$$\mathbf{q}_\xi \triangleq (f_x, f_y, m)$$

where:

$$(f_x, f_y) = \mathbf{f}_1 + \mathbf{f}_2$$
$$m = (\mathbf{r}_1 - \mathbf{o}) \times \mathbf{f}_1 + (\mathbf{r}_2 - \mathbf{o}) \times \mathbf{f}_2$$

In matrix form $\mathbf{q}_\xi = \mathbf{J}\mathbf{F}_\xi$ where $\mathbf{J} = \nabla_{\mathbf{F}_\xi}\mathbf{q}_\xi$ is the Jacobian.

4) Limit Surface Model: Under the quasistatic assumption, the total generalized force $\mathbf{q}\xi$ is constrained on a limit surface approximated by an ellipsoid:

$$(f_x/f_{\max})^2 + (f_y/f_{\max})^2 + (m/m_{\max})^2 = 1 \tag{4}$$

Following the approach from the limit surface theory, the gradient of the limit surface is proportional to the desired object velocity:

$$\nabla\mathcal{L}(\mathbf{q}\xi) = \lambda\dot{\mathbf{o}}^* \tag{5}$$

where $\lambda > 0$ is a scaling factor and:

$$\nabla\mathcal{L}(\mathbf{q}\xi) = \left(\frac{2f_x}{f_{\max}^2}, \frac{2f_y}{f_{\max}^2}, \frac{2m}{m_{\max}^2}\right)$$

We also constrain the motion of the robots to their respective contact points, ensuring the motion of the contact point aligns with the kinematically constrained motion of the robot.

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} + \begin{bmatrix} -\omega(r_{i,y} - o_y) \\ \omega(r_{i,x} - o_x) \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{p}_{i,x} \\ \dot{p}_{i,y} \\ \dot{p}_{i,\theta} \end{bmatrix}, \quad \forall i \in \{1, 2\} \tag{6}$$

5) Stable Pushing Conditions: A stable pushing trajectory exists if the required generalized force $\mathbf{q}\xi$ lies within the achievable force set Q_ξ given the constraints on the robots' individual forces. In the objective function, we minimize a weighted sum of the L_1-norm along with the $L\infty$-norm of the individual force magnitudes $\mathbf{F}_\xi$. We thus minimize the total magnitude of forces applied on the object, and ensure that the pushing forces are distributed between the robots.

C. Multi-Robot Controller

Given pushing forces and pushing poses of robots for a desired object twist, we extract a velocity-profile for the motion of each robot. Under quasistatic conditions, the robots exert these pushing forces by tracking desired velocities for the extracted trajectory. We use a model predictive controller to ensure that robots maintain their desired velocities, thereby performing the object twist. For the scope of this work, the controller accounts for kinematic constraints and the robots maintain their velocities relative to each other, the controller does not account for motion of the object in the controller within its rollouts.

We assume the robots start in stable-pushing contact with the object at the start pose. When the robots successfully complete pushing the object along a desired twist, they reposition to pushing positions for the next object twist. We use car-like conflict based search (CL-CBS) [23] to plan the repositioning paths of the two robots. Our controller thus, switches between two modes of contact: stable pushing and repositioning. Each robot uses a Model Predictive Path Integral (MPPI) [24] controller that optimizes control inputs over a receding horizon to maintain its pushing configuration relative to the other robot. In future work, we aim to use an approximate pushing model to increase robustness and allow recovery from failures while pushing. We implement this path tracking model-predictive controller for each robot using the `pytorch_mppi` library.

III. Results

We demonstrate our framework on MuSHR [18], an open-source 1/10th-scale mobile robot racecar, augmented with a 3D-printed flat bumper for pushing in a Mujoco simulation environment shown in Fig. 5. Our test cases include the four

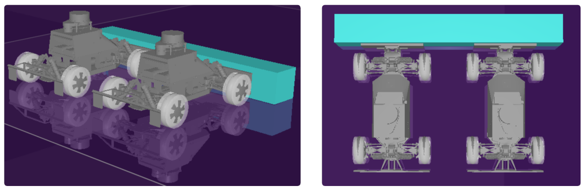

Fig. 5. Two MuSHR robot cars pushing a large object in a Mujoco simulation environment.

TABLE I
COMPARISON OF MEAN ERROR, STANDARD ERROR, AND PATH LENGTH ACROSS DIFFERENT TEST CASES OVER 100 TRIALS.

Test Case	**Mean Error (m)**	**Std Error (m)**	**Path Length (m)**
1	0.004	0.0016	4
2	-0.057	0.009	3.92
3	0.129	0.0045	9.84
4	-0.086	0.0039	7.84

pushing trajectories in Fig. 6 where robots track a desired object trajectory for an object of size 0.1m$\times$0.8m$\times$0.1m, we present the average metrics over 100 trials in Table I. The metrics show that error accumulation correlates with path length and complexity. The longer paths in cases 3 and 4 exhibit cumulative errors as robots execute turns while maintaining pushing contact. Case 3 has the largest path length, resulting in the largest error. Additionally, Case 2 has a large mean error given its small path length because of sliding motion observed while rotating the object. The object slides outwards during the transformation, resulting in the negative bias of the mean. Despite this, the low standard error across all test cases demonstrates the strengths of leveraging stable pushing even for extended trajectories.

IV. DISCUSSION

In the future, we aim to explore additional pushing configurations, including the ones illustrated in Fig. 3. These pushing configurations use a diverse range of contact-rich interactions with corner contacts, opposing forces and caging strategies [9] which, in turn, should help in finding more cost-effective object maneuvers. We also aim to demonstrate these pushing configurations with more than two robots.

Additionally, we plan to develop an approximate analytical pushing model to reduce error accumulation along longer pushing paths. In this work, we observe higher errors along the longer trajectories or larger number of segments. However, when using an approximate model to predict the state of the pushed object in the controller, we can enable robots to take corrective measures while pushing and improve pushing accuracy. Ongoing work involves the development of analytical and learning-based models for pushing with car-like robots.

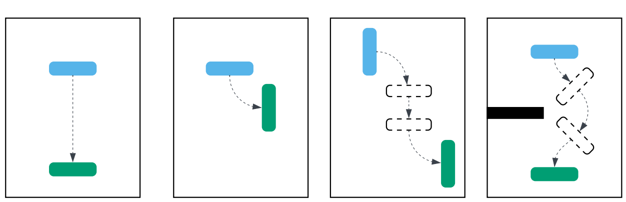

Fig. 6. Start (blue) and goal (green) positions of four test cases with intermediate transitions illustrated.

Lastly, we plan to demonstrate our methods on hardware using 1/10th scale MuSHR robot racecars to identify and address practical challenges not captured in simulation.

REFERENCES

[1] J. Ahn and C. Mavrogiannis. ReloPush: Multi-object rearrangement in confined spaces with a nonholonomic mobile robot pusher. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[2] M. Bauza, F. R. Hogan, and A. Rodriguez. A Data-Efficient Approach to Precise and Controlled Pushing. In *Proceedings of The 2nd Conference on Robot Learning*. PMLR, 2018.

[3] O. Ben-Shahar and E. Rivlin. Practical pushing planning for rearrangement tasks. *IEEE Transactions on Robotics and Automation*, 1998.

[4] F. Bertoncelli and L. Sabattini. Planar pushing manipulation with a group of mobile robots. In *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021.

[5] F. Bertoncelli, M. Selvaggio, F. Ruggiero, and L. Sabattini. Task-oriented contact optimization for pushing manipulation with mobile robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[6] M. Dogar and S. Srinivasa. A framework for push-grasping in clutter. *Robotics: Science and systems VII*, 1:65–72, 2011.

[7] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 1957.

[8] Y. Feng, C. Hong, Y. Niu, S. Liu, Y. Yang, W. Yu, T. Zhang, J. Tan, and D. Zhao. Learning multi-agent loco-manipulation for long-horizon quadrupedal pushing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[9] J. Fink, M. A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[10] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction part 1. limit surface and moment function. *Wear*, 143:307–330, 1991.

[11] F. R. Hogan and A. Rodriguez. Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics. In *Algorithmic Foundations of*

Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics, 2020.

[12] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer. Cooperative multi-robot object transportation system based on hierarchical quadratic programming. *IEEE Robotics and Automation Letters*, 2021.

[13] K.-C. Kuo and K.-Y. Chan. Stable pushing in narrow passage environment using a modified hybrid A* algorithm. *J Intell Manuf*, 2024.

[14] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *The international journal of robotics research*, 1996.

[15] M. Mason. Manipulator grasping and pushing operations. *Ph.D. Thesis*, 1982.

[16] M. J. Mataric, M. Nilsson, and K. T. Simsarin. Cooperative multi-robot box-pushing. In *Proceedings 1995 IEEE/RSJ international conference on intelligent robots and systems. Human robot interaction and cooperative robots*, 1995.

[17] M. Peshkin and A. Sanderson. The motion of a pushed, sliding workpiece. *IEEE Journal on Robotics and Automation*, 4(6):569–598, 1988.

[18] S. S. Srinivasa, P. Lancaster, J. Michalove, M. Schmittle, C. Summers, M. Rockett, J. R. Smith, S. Choudhury, C. I. Mavrogiannis, and F. Sadeghi. Mushr: A low-cost, open-source robotic racecar for education and research. *CoRR*, abs/1908.08031, 2019.

[19] S. Talia, A. Thareja, C. Mavrogiannis, M. Schmittle, and S. S. Srinivasa. PuSHR: A Multirobot System for Nonprehensile Rearrangement. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[20] Z. Tang, Y. Feng, and M. Guo. Collaborative planar pushing of polytopic objects with multiple robots in complex scenes. In *Robotics: Science and Systems (RSS)*, 2024.

[21] Z. Wang and M. Schwager. Multi-robot manipulation without communication. In *Distributed Autonomous Robotic Systems*, 2016.

[22] J. J. Weeda, S. Bakker, G. Chen, and J. Alonso-Mora. Pushing through clutter with movability awareness of blocking obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[23] L. Wen, Y. Liu, and H. Li. Cl-mapf: Multi-agent path finding for car-like robots with kinematic and spatiotemporal constraints. *Robotics and Autonomous Systems*, 2022.

[24] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.