

Allowing Safe Contact in Robotic Goal-Reaching: Planning and Tracking in Operational and Null Spaces

Xinghao Zhu^{1,2}, Wenzhao Lian³, Bodi Yuan², C. Daniel Freeman⁴, and Masayoshi Tomizuka¹

Abstract—In recent years, impressive results have been achieved in robotic manipulation. While many efforts focus on generating collision-free reference signals, few allow safe contact between the robot bodies and the environment. However, in human’s daily manipulation, contact between arms and obstacles is prevalent and even necessary. This paper investigates the benefit of allowing safe contact during robotic manipulation and advocates generating and tracking compliance reference signals in both operational and null spaces. In addition, to optimize the collision-allowed trajectories, we present a hybrid solver that integrates sampling- and gradient-based approaches. We evaluate the proposed method on a goal-reaching task in five simulated and real-world environments with different collisional conditions. We show that allowing safe contact improves goal-reaching efficiency and provides feasible solutions in highly collisional scenarios where collision-free constraints cannot be enforced. Moreover, we demonstrate that planning in null space, in addition to operational space, improves trajectory safety. Further information is available at <https://rolandzhu.github.io/ContactReach/>.

I. INTRODUCTION

Robotic manipulation in unconstrained environments has received increasing attention in recent years. Although much research focuses on generating and tracking collision-free reference signals (e.g., the desired trajectory) [1]–[4], some manipulation tasks benefit from allowing contact between the robot and the environment. For example, in shelf picking, the robot sometimes must collide with obstacles to grasp the target object [5], [6]; in agricultural harvesting, it is favourable for the robot to push away the occluding petioles and leaves [7].

In this work, we propose to allow contact in robotic manipulation. Allowing contact can enhance the robot’s efficiency and capability by enlarging the feasible set [8]. The advantage of a larger feasible set is two-fold (Fig. 1). First, it yields better trajectories than enforcing collision-free constraints, measured by trajectory length and jerkiness. Second, it can generate feasible solutions in highly collisional scenarios where collision-free solutions cannot be found.

One of the most critical concerns in contact-allowed manipulation is safety. Without considering safety, the robot can hit obstacles heavily and damage the hardware. Safety constraints can be imposed by restricting the contact forces [9] and we impose this constraint by online generating and tracking of the reference signals. For the prior, a constraint

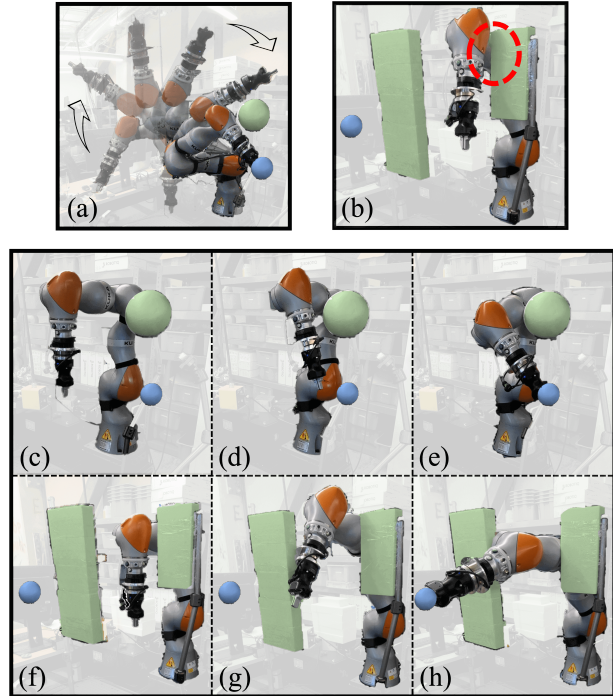


Fig. 1: **(a)** The robot finds a long and unnatural trajectory to reach the goal (blue) avoiding the obstacle (green). **(b)** The goal-reaching robot fails to find a collision-free trajectory; red circle indicates collision with the obstacle. By allowing safe contact, the robot reaches the goal more efficiently **(c-e)**, and finds a trajectory in the highly collisional scenario **(f-h)**.

can be applied during trajectory optimization to limit the contact forces. For the latter, a robot controller can leverage null space compliance to reduce the contact force while tracking the desired trajectory [10].

This paper studies the safe contact-allowed robotic goal-reaching problem with two feedback control loops. The outer loop optimizes the time-varying operational and joint trajectories in a receding horizon manner. The dynamics model of the robot and objects is approximated with a differentiable simulator, Brax [11], and utilized to impose contact constraints. The inner loop tracks the trajectory using an impedance controller. This paper focuses on redundant manipulation where the robot is kinematically redundant to the task, and includes a null space projector in addition to an operational space control. Compared with other works, our method optimizes joint configuration as extra decision variables and actively explores the null space for a safer motion. To solve the trajectory planning, we propose a hybrid

¹ Mechanical Systems Control Lab, Mechanical Engineering, UC Berkeley, CA, USA {zhuxh, tomizuka}@berkeley.edu

² X, The Moonshot Factory, CA, USA bodiyuan@google.com

³ Intrinsic Innovation LLC, CA, USA wenzhaol@google.com

⁴ Google Research, CA, USA cdfreeman@google.com

algorithm that integrates sampling- and gradient-based methods, gaining the benefit of scalability and differentiability.

We empirically evaluate the proposed method in various simulation and real-world environments to demonstrate the effectiveness of the contact-allowed planning and control. In summary, our work makes the following contributions.

- We state the contact-allowed robotic goal-reaching task with safety constraints. We provide open-source environments and benchmarks for the task.
- We propose a time-varying trajectory planner and tracking controller for the contact-allowed problem. The planner optimizes both operational and null space reference signals to efficiently and safely achieve the goal.
- We present a hybrid optimization solver for the trajectory planner. The superiority of the proposed algorithm is evaluated in diverse experiments ranging from free space to highly collisional.

The remainder of the article is organized as follows. Related works are discussed in Chapter II. The details of our method and environments are described in Chapter III and IV. The experiments and conclusions are presented in Chapter V and VI.

II. RELATED WORK

A. Impedance and Null Space Control

Recently, problems and algorithms related to compliant behaviour have obtained an increasing amount of interest because of its capability of ensuring safe interaction. Impedance control represents an appealing approach to achieve robot compliance with joint torque command [12]–[14]. The impedance control uses a mass-spring-damper system to model the dynamic relationship between the robot and the contact objects. Another method that focuses on increasing contact safety leverages kinematic redundancy with null space control [15]. In [16]–[19], the null space control was considered for a case where the joint space motion operates in the null space of the operational task. These works showed it is effective to render desired joint impedance behaviour without affecting the operational task. [20], [21] models safety as the primary objective and optimizes the operational motion to diminish contact. In contrast, we optimize the joint and operational space motion simultaneously while preserving the operational task structure.

Receding horizon control strategies are typically utilized to generate and track reference signals with the above controller, such as model predictive control [22], [23] or reinforcement learning [2], [4]. The former defines a cost function and control constraints to optimize a short-term motion. The latter selects actions based on the estimated reward or value at each time step. In [24], the null space behaviour was optimized to avoid collisions while satisfying the task constraints. In [1], a null space posture torque is applied to deal with the unexpected collision. This paper similarly leverages the receding horizon control framework to generate and track the trajectory. But unlike previous works, we treat the null space behaviour as an additional

decision variable for the operational space motion, and optimize them at the same time for safe and efficient robotic manipulation.

B. Contact-Allowed Robotic Manipulation

Allowing safe contact between the robot and the environment can improve performance in specific tasks. Some works focus on contact-allowed trajectory planning with deformable obstacles. Different from collision-free problems, interactions between the robot and the environment impose uncertainties on the dynamics. In [25], a simulator was included to deal with dynamical uncertainty. In [9], the contact force and collision response were approximated with the finite-element method. Instead of approximating the dynamics, contact constraints were represented using voxel grids and intruding volumes in [8]. Although intruding distance avoids the usage of simulation, it only works for static obstacles where the object center stays fixed. This, however, is not true in practice since the obstacle can move along the contact direction. Moreover, parameterizing intruding volumes/distances is less explainable or intuitive than contact forces. Therefore, the above works only focus on finding feasible trajectories for simple deformable geometries. These trajectories are generated by the rapidly-exploring random tree (RRT) or its variants [26], [27] in one shot and won't be updated during trajectory execution.

Other works have included robot control in contact-allowed manipulation using sample-based or optimization-based methods like control barrier functions [28]. These works, however, restrict the contact area to the feet for legged robot jogging [29], [30], or the end effector for manipulator wiping and polishing [31]–[33]. A comparison of control methods and task definitions was presented in [34]. Body contact is typically hard to sense and localize accurately [10], limiting the usage of these controllers in the null space contact scenario. This work aims to allow contact between the full robot body and the environment for a goal-reaching task. We use simulation to approximate the system dynamics [9] and iteratively plan and execute robot motions [31].

III. CONTACT-ALLOWED ROBOTIC GOAL-REACHING

A. Problem Formulation

This paper chooses goal-reaching as the operational manipulation task. It requires the end-effector to reach a given goal pose, while the manipulator can collide with the environment with a maximum permitted contact force. Relating to the formalisms in Chapter I, a hierarchical framework is utilized to generate and track robotic motions, as shown in Fig. 2. The outer planner takes the robotic and environmental state s as input and optimizes operational and joint space motions $a = \{\Delta x, \Delta q\}$; the input includes robot joints and obstacle information¹. The inner robot controller computes joint torques τ based on the references to actuate the robot.

The 3D goal pose is denoted as s_g . The success criteria is defined as $\|s_t - s_g\| \leq \delta$ ($\delta = 0.01m$ in this paper), where

¹Poses for rigid obstacles; node positions for deformable obstacles represented by node graphs, as in Fig. 3(c).

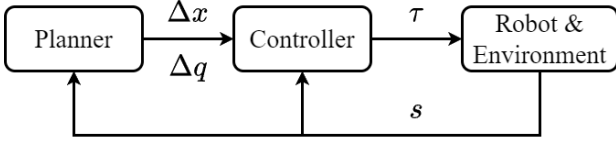


Fig. 2: The planning loop takes in state s and optimizes reference signals (i.e., operational space references Δx and null space references Δq) for the robot controller, which generates joint torques τ as actuation commands.

$\|\cdot\|$ computes the translational distance from the end-effector to s_g . The system propagates with the transfer function $T(s_t, \tau_t)$ as the robot executes the torque command at t , which computes the successive state s_{t+1} and the contact force. The maximum permitted contact force is written as ε .

B. Operational and Null Space Impedance Control

In this work, we adopt operational space computed torque control [12] for the manipulation task and introduce null space projection [10] for safe contact. Previous works have shown the advantage of planning task-related motions in the operational space over the joint space [4]. Not only the manipulation tasks are typically defined in the operational space, but also it is straightforward to adapt the controller among different robots. Moreover, null space projection allows compliance at the joint level for a redundant robot without disturbing the operational task, thus improving safety by reducing contact forces.

This work uses the computed torque control [35] as the operational space controller. It first calculates the wrench for the end-effector based on the desired motion and then maps the wrench to joint torques. The control law is written as (1).

$$\tau_{op} = J(q)^T \Lambda(q) (-K_p \Delta x - K_d \dot{x}) + C(q, \dot{q}) + g(q) \quad (1)$$

For a robotic manipulator with n joints, $q \in \mathbb{R}^n$ is the joint angle, $x \in \mathbb{SE}(3)$ is the end-effector pose, $J(q)$ is the task Jacobian, $\Lambda(q) = (J(q)M^{-1}(q)J(q)^T)^{-1}$ is the operational space inertial matrix and $M(q) \in \mathbb{R}^{n \times n}$ is the robot inertial matrix, $C(q, \dot{q}) \in \mathbb{R}^n$ is the Coriolis and Centrifugal forces, $g(q) \in \mathbb{R}^n$ is the robot gravity vector. We use $\Delta x = x \ominus x_d$ to represent the error in $\mathbb{SE}(3)$, where x_d is a desired operational pose. $K_p, K_d \succeq 0$ are PD gains for the controller. The following sections will drop the dependencies in q, \dot{q} for better readability.

The null space control projects joint torques into the null space with the projector $N = I - J^\dagger J$, where $J^\dagger = M^{-1} J^T \Lambda$ is the dynamically consistent inverse of the Jacobian [1], [17]. The projected torque does not affect the motion in the operational space. We implement a joint space PD controller to generate and project the torque:

$$\tau_{null} = N^T (-K_{qp} \Delta q - K_{qd} \dot{q}) \quad (2)$$

, where $\Delta q = q - q_d$ and q_d is a desired joint posture. $K_{qp}, K_{qd} \succeq 0$ are control gains in the joint space.

Combining the operational space control (1) with the null space projection (2), the final control output is defined as:

$$\tau = \tau_{op} + \tau_{null}. \quad (3)$$

C. Contact-Allowed Motion Planning

Section III-B introduces an operational space controller with null space compliance. The control law (1, 2, 3) takes in the reference signals $a = \{\Delta x, \Delta q\}$ to compute the actuation torque. This section introduces how to find the time-varying reference signals for efficient and safe robotic manipulation.

The problem is formulated in a receding horizon manner. For each time step k , the optimization is written as follows.

$$\min_{a_{k:k+H-1}} \sum_{t=k}^{k+H-1} \lambda_1 \|s_{t+1} - s_g\| - \lambda_2 \|s_{t+1} - s_k\| \quad (4a)$$

$$\text{s.t. } \tau_t = f(a_t) \quad \triangleright (3) \quad (4b)$$

$$s_{t+1} = T(s_t, \tau_t) \quad (4c)$$

$$\text{contact}(s_t, a_t) \leq \varepsilon \quad (4d)$$

$$\text{for } t \in [k, \dots, k+H-1]$$

The objective function (4a) minimizes costs over the horizon H . s_k is the initial state at the current planning horizon. $\lambda_1, \lambda_2 > 0$ are hyper-parameters. $\|\cdot\|$ measures the distance of end-effectors between states. The first term minimizes the distance to the goal, representing the objective to reach the destination efficiently. The second term encourages exploration by forcing the robot away from the initial state. In practice, it helps to escape from the local minimum.

Constraint (4b) stands for the control law in (3). Constraint (4c) represents the transfer function. Constraint (4d) encodes the safety requirement, where $\text{contact}(s_t, a_t)$ measures the contact force while executing the robot control; the contact force is constrained by the maximum bound ε .

The transfer function $T(s_t, \tau_t)$ approximates the system propagation in (4c) and estimates the contact force in (4d). We use the Brax physics engine [11] as the transfer function. Brax is designed for performance and parallelism on accelerators, allowing a large sample size for sampling-based optimization solvers. Moreover, it supports auto-differentiation of the dynamics, which makes it possible to use gradient-based solvers. The next section introduces how we solve the optimization problem by leveraging the parallelism and differentiable properties of the Brax engine.

D. Solving the Optimization Problem

The optimization (4) is first written in a non-constraint form by plugging dynamics (4b, 4c) into the objective (4a) and simplifying contact constraints (4d):

$$\min_{a_{k:k+H-1}} \sum_{t=k}^{k+H-1} \lambda_1 \|T(s_t, f(a_t)) - s_g\| - \lambda_2 \|T(s_t, f(a_t)) - s_k\| + \lambda_3 (\text{contact}(s_t, a_t) - \varepsilon) \quad (5)$$

To solve (5), this paper proposes a hybrid solver that integrates the covariance matrix adaptation evolution strategy (CMA-ES, [36]) and the gradient descent method, as outlined in Algorithm 1. The cost function in (5) is denoted as \mathcal{L} .

Compared to others [36], [37], our method uses a larger population size, bounds the decision variables, and applies

Algorithm 1 Hybrid Optimization Solver

```

1: Initialize  $m = 0$ ,  $\sigma = 0.01$ ,  $C = I$ ,  $p_\sigma = p_c = 0$ 
2:  $k = 200$ ,  $k_{top} = 50$  ▷ Population size
3:  $\gamma$ ,  $\beta_{max} = 1$  ▷ Boundaries & max grad. step
4: for  $step = 1$  to  $max\ step$  do
5:   for  $i \in \{1, \dots, k\}$  do
6:      $a_i \sim \mathcal{N}(m, \sigma^2 C)$  ▷ Sample candidates
7:      $a_r = \text{clip}(a_i, -\gamma, \gamma)$  ▷ Repair candidates
8:      $d_i = \|a_i - a_r\|^2$ 
9:      $\alpha = \begin{cases} \exp(\log \mathcal{L}(a_i) - \log d_i) & \text{if } d_i > 0 \\ 0 & \text{otherwise} \end{cases}$ 
10:     $l_i = \mathcal{L}(a_r) + \alpha \cdot d_i$  ▷ (5) w/ penalty
11:  end for
12:   $a_{best} = \text{argsort}(l_{1:k})$  ▷ Find  $k_{top}$  best candidates
13:   $m = \text{mean}(a_{best})$  ▷ CMA-ES mean
14:   $\nabla \mathcal{L} = \frac{\partial \mathcal{L}(m)}{\partial a}$  ▷ Diff. w/ Brax
15:   $\beta_1, \dots, \beta_k \sim \mathcal{U}[0, \beta_{max}]$ 
16:   $m = m + \text{argmin}_{\beta_{1:k}} \mathcal{L}(m + \beta_j \cdot \nabla \mathcal{L}) \cdot \nabla \mathcal{L}$ 
17:   $m = \text{clip}(m, -\gamma, \gamma)$ 
18:  Update  $\sigma$ ,  $C$ ,  $p_\sigma$ ,  $p_c$  ▷ CMA-ES update [36]
19: end for
20: Return  $m$ 

```

an additional gradient descent step. The boundary constrains the robot’s movement in operational and joint space. A penalty is added to the cost function, which is the distance to the boundary. A tradeoff weight α is adaptively selected so that the cost and the penalty are similar in magnitude. Moreover, we add a gradient descent step before refitting distributions in CMA-ES. The gradient step optimizes the solution locally. Similar ideas have been utilized in [37], [38] and have shown improved performance over CMA-ES alone. Nevertheless, our method does not iterate the gradient descent until convergence. We sample a population of neighbours in the gradient direction and find the best one as the local optimal, similar to [37]. Such design gains an advantage in computation speed as the evaluation of the cost is much faster than that of the gradient. Thus, we sample more candidates and only apply the gradient descent once per step. The algorithm is halted if the maximum step has been reached or the cost stagnates for three steps. Readers are referred to [36] for details of the CMA-ES. Although this paper initializes the solution to zeros for convenience, it can be easily replaced by solutions obtained from other motion planners. However, due to the large population size used in the solver, we did not observe significant improvements by changing the initial values.

IV. ENVIRONMENTS

This chapter introduces the simulation and the real-world environments in which we evaluate and compare different control and planning methods, as visualized in Fig. 3. These environments span different levels of collision conditions, ranging from free space to highly collisional.

The simulation environments are built with Brax [11], where the robot is modelled with cylinder links to imitate the kinematics and dynamics of a real one (i.e., Kuka iiwa

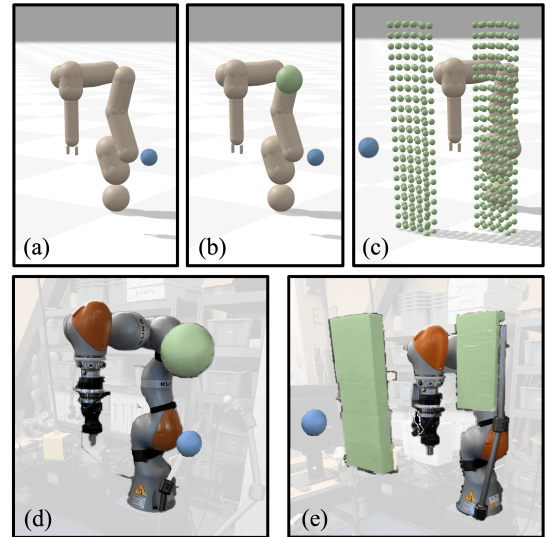


Fig. 3: Goal-reaching task environments of different collision complexity levels. The target is blue, and the obstacles are green. (a) Free space, (b) ball obstacle, (c) wall obstacle, (d) real-world ball obstacle, and (e) real-world wall obstacle.

14). Such modelling significantly reduces the computation time with simpler contact dynamics. The following sections introduce the detailed configurations for each environment.

A. Free Space

This environment aims to measure the performance of the proposed method in a collision-free condition. The target is randomly placed inside the robot manipulability ellipsoid. Since there is no obstacle, the optimal trajectory in the operational space is a straight line, as defined in (4, 5).

B. Ball Obstacle

This environment intends to show that it is more efficient to accomplish certain manipulation tasks by allowing safe contacts. In this environment, a ball with $0.1m$ radius is added as an obstacle. The target is randomly placed within a workspace, varying from $[0.2, -0.3, 0.0]$ to $[0.6, 0.3, 0.5]$ in $[x, y, z]$, respectively. The obstacle is placed to collide with the robot in the free space trajectory. Specifically, the optimal path is first computed without the obstacle. Then the ball is added to hamper the optimal path by colliding with the robot’s middle bodies. Although the obstacle makes the free space trajectory infeasible, other collision-free paths exist.

C. Wall Obstacle

This environment demonstrates that allowing safe contacts enables highly constrained manipulation tasks where collision-free paths cannot be found [9]. The environment puts two $1 \times 0.1 \times 0.2m$ walls between the robot and the target. The walls obstruct all collision-free paths.

The deformable wall is modelled as a mass-spring system in the Brax simulator. The wall is tessellated to volumetric finite element meshes, including vertices and edges. We use small rigid spheres to represent the mesh vertices and spring joints to represent the mesh edges. The simulation can mimic

the physical deformation by carefully parameterizing the sphere mass and the spring stiffness. The robot's initial joint configuration is fixed through trials, while the target pose is randomized in front of the robot.

D. Real-World

Besides the simulation, we create physical environments with foam balls and walls to evaluate the proposed method in the real world. The foam ball has a $0.1m$ radius, whose position is generated similarly as in simulation. The foam walls are positioned with fixtures as shown in Fig. 3(e), and have the same effective dimensions as in simulation. Since tracking the real-world system is not the main focus of this work, we assume the position and deformation of the foam obstacles are known. We manually tuned the coefficients in the dynamics approximator (i.e., Brax) to obtain an accurate transfer function (4c). In practice, the transfer function and the system state can be estimated with an additional observer, as suggested in [39], [40]. In our real-world experiments, we utilized joint torque sensors to measure external contact torques and scaled the threshold in (4d) to address the gap between simulation and reality.

V. EXPERIMENTS

This chapter provides experiments for the proposed method in environments described in Chapter IV. The goal of the experiments is three-fold. First, we demonstrate the advantage of allowing contacts in multiple collisional scenarios. Second, we show the benefit of generating and tracking both the operational and null space reference signals. Third, we provide empirical evaluations for the proposed hybrid optimization solver.

The hyperparameters used throughout our experiments are set as follows. $\lambda_{1,2,3} = \{1, 0.2, 5\}$, $H = 3$, $\varepsilon = 10N$, $max\ step = 100$, $\gamma = [0.01m, 0.2rad, 0.2rad]$ where the first two represent the end-effector maximum movement and the last limits the null space motion. In this paper, we do not infer control gains; instead, they are set to constants: $K_p = 880I$, $K_d = 100I$, $K_{qp} = 30I$, $K_{qd} = I$. On the one hand, gains work as a scaling factor for $\Delta x, \Delta q$ in (3), thus implicitly optimized. We did not observe improvement by including gain to the decision variables in our experiments. On the other hand, including gain inference makes the algorithm less stable and harder to solve. The control torque is prone to going unbounded with the inferred scaling.

A. Collision-Free or Contact-Allowed

This experiment compares the performance of our contact-allowed planner with a collision-free trajectory planner. We used the RRT* [26] to generate the collision-free trajectory in the joint space and our controller to track the path without replanning. Each experiment was repeated three times to record the average task completion time in Table I. The maximum contact force throughout the execution is reported in simulated environments. A torque observer was implemented to measure the external torque in the real world [1].

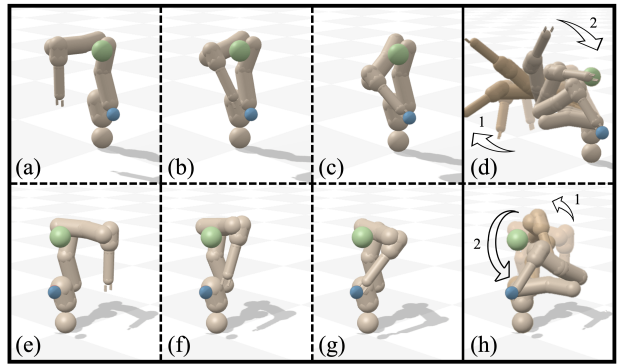


Fig. 4: Execution trajectories in simulated ball environments. (a-c, e-g) show two trajectories generated by our method, (d, h) show the collision-free trajectories with the same environmental settings.

For contact-allowed goal-reaching, an alternative to the receding horizon planning is to plan the whole trajectory before moving the robot [9], [25]. However, the large search space, from allowing contact and additional null space motion, makes the planning problem intractable. Thus, we do not include such a planning framework in the comparison.

As seen in the ball obstacle experiment in Table I, allowing safe contact improves the task efficiency with the ball obstacle. Although contact forces appeared during the execution, our method reduces the task completion time by 2.6x. Fig. 4 visualizes two example executions in the simulated environment. In the highly collisional environment, i.e., wall obstacles, a collision-free trajectory cannot be found to reach the target. In contrast, our method completes the task by pushing the obstacles away. These results indicate that allowing collision relaxes the optimization constraints and provides a larger feasible motion set. Lastly, in the free-space environment comparison, we observed the same performance, suggesting that allowing contact does not change the behaviour in unconstrained scenarios.

B. Different Control Laws and Decision Variables

This experiment provides ablation studies to the control laws and decision variables in (4). The ablation method applies a reference posture torque in the null space to track a joint reference. We set the desired joint posture $q_d = 0$ in (2). The control law is written as:

$$\tau_{posture} = \tau_{op} + N^T (-K_{qp}q - K_{qd}\dot{q}) \quad (6)$$

The trajectory planner only optimizes the operational space motion Δx and has no control over the null space.

Based on the results in Table I, our method outperforms the reference posture control method in all environments. Fig. 5 shows the contact force profile for an experiment in the wall environment. Our method reduces the contact force by almost 4x in the simulated wall environments and 1.7x in the ball environments. Our method also reduces the task execution time by generating shorter operational trajectories. The real-world experiments further support the findings that considering additional null space motions increases task safety. These results validate the necessity of optimizing

TABLE I: Comparison of trajectory efficiency and safety metrics in simulated and real-world environments. In each cell, the first element is the task execution time (s); the second is the maximum contact force on the robot body in simulation (N) or the maximum computed external torque on robot joints in real-world (Nm).

	Free Space		Ball Obstacle		Wall Obstacle		Ball Obstacle Real		Wall Obstacle Real	
Collision-Free	11.4 ± 1.4	0	37.8 ± 3.9	0	Fail	Fail	38.4 ± 7.9	0	Fail	Fail
Ref. Posture	11.4 ± 1.4	0	13.1 ± 0.4	4.7 ± 2.3	21.3 ± 2.0	1.9 ± 0.7	15.2 ± 1.7	1.6 ± 0.2	18.7 ± 1.1	1.1 ± 0.1
Ours	11.4 ± 1.6	0	12.8 ± 0.5	2.8 ± 1.2	17.5 ± 1.5	0.5 ± 0.3	14.7 ± 1.6	0.7 ± 0.1	15.0 ± 1.0	0.5 ± 0.1

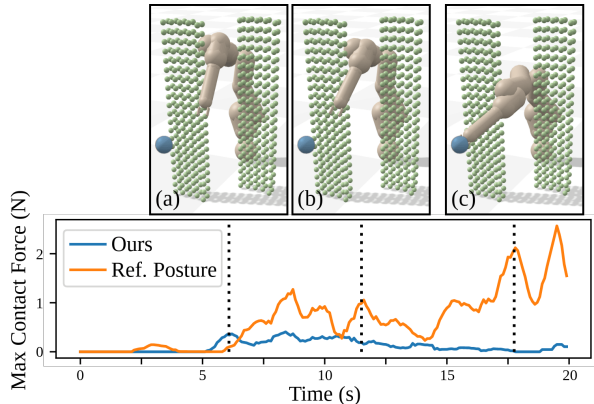


Fig. 5: Example contact force profile when reaching a goal in the wall environment. Pictures correspond to the robot configuration induced by our method at time stamps marked by dotted vertical lines; (a), (b) the robot tracks trajectories to push obstacles in a compliant manner and adjusts its joint configuration in the null space; (c) the robot reaches the goal while maintaining a minimum contact force. Compared to the ablations, our method, which controls both operational and null spaces, results in lower overall contact forces.

the null space motion. Since the null space behaviour is neither optimized nor controlled in the ablation method, it has a tighter search space than ours and often cannot find a trajectory with the same performance as ours.

Another approach to planning robot trajectories is to directly optimize the motor torque. We implemented a joint torque planner and observed similar performance compared to our method. In contrast, our approach involves mapping the planned trajectory to motor torque using a control law defined in (3). This structured design can be viewed as a specific instance of direct torque planning. However, optimizing joint torques directly leads to a more complex optimization problem and requires more computation to solve. For instance, in the wall obstacle environment, our method took an average of $0.47s$ to converge for each step, whereas direct torque planning took $1.34s$. Similar results were reported in [4]. Furthermore, as manipulation tasks like our goal-reaching problem are typically defined in the operational space, it is more intuitive and explicit to search for actions in the operational space.

C. With or Without Gradient Descent

This experiment analyzes the proposed hybrid optimization solver in the wall obstacle environment. We demonstrate the effectiveness of the bounded CMA-ES and single-step gradient descent, line 7-9 and 16 in Algorithm 1. Two baselines were used for comparison. The first only uses

the vanilla CMA-ES to search for the optimal reference signals [36], while the second involves multiple gradient steps until convergence, similar to [37].

Interestingly, all solvers yield similar trajectories given the same environmental configuration, suggesting that they have converged to close solutions. Thus, we use the average cost decrease after 10 optimization steps as the comparison metric. Results showed that our method (0.34) outperforms the vanilla CMA-ES (0.22). This validates the benefit of integrating gradient descent with the sampling-based approach; it expedites the convergence by locally refining the solution. Meanwhile, we observed further improved performance with multiple gradient steps (0.40). However, the iterative gradient calculation requires much longer computation time and significantly reduces the planning frequency. With the consideration of real-time performance, we applied single-step gradient descent in our proposed solver.

VI. CONCLUSION

This paper investigates the problem of contact-allowed robotic goal-reaching with operational and null space control and makes several key contributions. Firstly, we formulate the contact-allowed robotic manipulation problem with safety constraints and provide a set of open-source environments for contact-allowed goal-reaching. These environments have different collision conditions, ranging from free space to highly collisional. Secondly, we propose a receding horizon trajectory planner to generate and track reference signals for collision-allowed motion. This planner optimizes the operational and null space reference signals and tracks the reference using an impedance controller. Finally, we present a hybrid solver to optimize the reference signals. Simulation and real-world experiments demonstrate that by allowing contact, our proposed algorithm enables efficient and safe achievement of the manipulation goal.

The present work has some limitations. First, we do not demonstrate a more complex operational task than goal-reaching. It is interesting to generalize the optimization objective to other tasks by combining reinforcement learning or model predictive control techniques. Second, the current algorithm assumes known environmental states. There is no state observer during the real-world execution and thus is susceptible to sim-to-real gaps. In future works, we plan to address these limitations and develop contact-allowed algorithms for diverse robotic manipulation tasks, such as assembly [41], table wiping, bin-picking [39], [42], and other human-robot interaction tasks.

REFERENCES

- [1] X. Zhu, S. Kang, and J. Chen, "A contact-safe reinforcement learning framework for contact-rich robot manipulation," in *arXiv Preprint*, 2022.
- [2] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka, "Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9881–9887.
- [3] X. Zhu, L. Sun, Y. Fan, and M. Tomizuka, "6-dof contrastive grasp proposal network," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6371–6377.
- [4] R. Martín-Martín, M. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space. an action space for reinforcement learning in contact rich tasks," in *International Conference of Intelligent Robots and Systems (IROS)*, 2019.
- [5] M. R. Dogar and S. S. Srinivasa, "Push-grasping with dexterous hands: Mechanics and a method," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2123–2130.
- [6] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IROS*, 2018.
- [7] Y. Onishi and et al., "An automated fruit harvesting robot by using deep learning," in *Robomech*, 2019.
- [8] C. Phillips-Grafflin and D. Berenson, "A representation of deformable objects for motion planning with no physical simulation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 98–105.
- [9] S. Rodriguez, J.-M. Lien, and N. Amato, "Planning motion in completely deformable environments," in *2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2466–2471.
- [10] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Multi-priority control in redundant robotic systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3752–3757.
- [11] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, "Brax - a differentiable physics engine for large scale rigid body simulation," 2021. [Online]. Available: <http://github.com/google/brax>
- [12] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I—Theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 03 1985.
- [13] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, 1st ed. Springer Publishing Company, Incorporated, 2008.
- [14] S. M. Abu-Dakka FJ, "Variable Impedance Control and Learning-A Review," *Front Robot AI*, 12 2020.
- [15] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.
- [16] R. Platt, M. Abdallah, and C. Wampler, "Multiple-priority impedance control," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 6033–6038.
- [17] M. Jorda, E. G. Herrero, and O. Khatib, "Contact-driven posture behavior for safe and interactive robot operation," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9243–9249.
- [18] H. Sadeghian and et al., "Null-space impedance control with disturbance observer," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2795–2800.
- [19] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2014.
- [20] A. De Luca and L. Ferrajoli, "Exploiting robot redundancy in collision detection and reaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3299–3305.
- [21] E. Magrini and A. De Luca, "Human-robot coexistence and contact handling with redundant robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4611–4617.
- [22] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [23] M. R. Maly and et al., "Iterative temporal motion planning for hybrid systems in partially unknown environments," in *International Conference on Hybrid Systems*, 2013, p. 353–362.
- [24] L. Zhu, H. Mao, X. Luo, and J. Xiao, "Determining null-space motion to satisfy both task constraints and obstacle avoidance," in *IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 2016, pp. 112–119.
- [25] H. Durrant-Whyte, N. Roy, and P. Abbeel, "Motion planning under uncertainty in highly deformable environments," in *Robotics: Science and Systems VII*, 2012, pp. 241–248.
- [26] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," in *arXiv Preprint*, 2011.
- [27] K. Naderi, J. Rajamäki, and P. Hämäläinen, "Rt-rrt*: A real-time path planning algorithm based on rrt*," in *8th ACM SIGGRAPH Conference on Motion in Games*, 2015, p. 113–118.
- [28] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8352–8358.
- [29] Y. Zhu, Z. Pan, and K. Hauser, "Contact-implicit trajectory optimization with learned deformable contacts using bilevel optimization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9921–9927.
- [30] F. Ruscelli, M. P. Polverini, A. Laurenzi, E. M. Hoffman, and N. G. Tsagarakis, "A multi-contact motion planning and control strategy for physical interaction tasks using a humanoid robot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3869–3876.
- [31] Beltran-Hernandez and et al., "Learning force control for contact-rich manipulation tasks with rigid position-controlled robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5709–5716, 2020.
- [32] V. Petrone, E. Ferrentino, and P. Chiacchio, "Time-optimal trajectory planning with interaction with the environment," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 399–10 405, 2022.
- [33] Y. Fan, X. Zhu, and M. Tomizuka, "optimization model for planning precision grasps with multi-fingered hands," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1548–1554.
- [34] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022.
- [35] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.
- [36] N. Hansen, "The CMA evolution strategy: A tutorial," in *arXiv Preprint*, 2016.
- [37] R. Antonova, J. Yang, K. M. Jatavallabhula, and J. Bohg, "Rethinking optimization with differentiable simulation from a global perspective," in *arXiv Preprint*, 2022.
- [38] X. Chen, X. Liu, and Y. Jia, "Combining evolution strategy and gradient descent method for discriminative learning of bayesian classifiers," in *11th Annual Genetic and Evolutionary Computation Conference, GECCO-2009*, 01 2009, pp. 507–514.
- [39] X. Zhu, S. Jain, M. Tomizuka, and J. Van Baar, "Learning to synthesize volumetric meshes from vision-based tactile imprints," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4833–4839.
- [40] X. Zhu, S. Jain, M. Tomizuka, and J. Vanbaar, "Synthesizing and simulating volumetric meshes from vision-based tactile imprints," in *ICRA 2022 Workshop: Reinforcement Learning for Contact-Rich Manipulation*, 2022.
- [41] W. Lian, T. Kelch, D. Holz, A. Norton, and S. Schaal, "Benchmarking off-the-shelf solutions to robotic assembly tasks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1046–1053.
- [42] X. Zhu, Y. Zhou, Y. Fan, L. Sun, J. Chen, and M. Tomizuka, "Learn to grasp with less supervision: A data-efficient maximum likelihood grasp sampling loss," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 721–727.