

An Optimal Control Formulation of Tool Affordance Applied to Impact Tasks

Boyang Ti, Yongsheng Gao, Jie Zhao, Sylvain Calinon

Abstract—Humans use tools to complete impact-aware tasks such as hammering a nail or playing tennis. The postures adopted to use these tools can significantly influence the performance of these tasks, where the force or velocity of the hand holding a tool plays a crucial role. The underlying motion planning challenge consists of grabbing the tool in preparation for the use of this tool with an optimal body posture. Directional manipulability describes the dexterity of force and velocity in a joint configuration along a specific direction. In order to take directional manipulability and tool affordances into account, we apply an optimal control method combining iterative linear quadratic regulator (iLQR) with the alternating direction method of multipliers (ADMM). Our approach considers the notion of tool affordances to solve motion planning problems, by introducing a cost based on directional velocity manipulability. The proposed approach is applied to impact tasks in simulation and on a real 7-axis robot, specifically in a nail-hammering task with the assistance of a pilot hole. Our comparison study demonstrates the importance of maximizing directional manipulability in impact-aware tasks.

Index Terms—Directional manipulability maximization, impact-aware motion, constrained motion planning, optimal control, iterative linear quadratic regulator (iLQR), alternating direction method of multipliers (ADMM).

I. INTRODUCTION

EFFICIENT manipulation requires choosing a comfortable posture to execute an action, so that it is easy to generate a force or a velocity according to the tool affordances. The intrinsic of the above phenomenon is that the central nervous system (CNS) will transform information about the initial human posture and the task requirement into an appropriate pattern of muscular activity, where the target position is first transformed into a desired arm posture, which is then used to compute the motor commands [1]–[3].

The prevalent research in robot motion planning focuses on imitating the human hand motion, such as some typical point-to-point (pick-and-place) or trajectory following tasks (writing/wiping/polishing). However, as trajectories extend into a

larger spatial area and/or task complexity increases, factors other than kinematics and biomechanics become increasingly relevant [4]. This demonstrates that arm morphology and its other intrinsic properties, such as dynamics, inertia, and muscle viscoelasticity, influence trajectory planning together with the environment geometry. Manipulability ellipsoid [5]–[7] is taken as a geometric descriptor to measure and visualize the capability of a manipulator in a given configuration by considering the kinematic and dynamic information of the system. The manipulability projections along different directions provide insights regarding the directions along which the manipulator has higher or lower dexterity. A higher projection value means that a greater displacement of the end-effector can be produced in this direction, for the given joint angle configuration. Jaquier *et al.* [8] proposed a manipulability ellipsoid tracking method and validated it in a peg-in-hole task, where two kinds of desired manipulability are aligned referring to the central axis of the ellipsoids. In this article, we use directional manipulability [9] as a metric of desired manipulability, which can be flexibly transferred to a broader range of scenarios.

The concept of object/tool affordance originates from the field of psychology [10], whose definition has been extended to describe the relationship between the properties of an object and the capabilities of an agent that determine how the object could possibly be used [11]. By extension, tool affordance refers to the qualities or characteristics of a tool that suggest the potential actions that can be performed with the tool. In this article, the notion of tool affordance will refer to the set of possible actions that could be generated according to the manner in which the tool is handled by the robot. It extends the function of the end-effector to enable the robot to expand its range of applications using tools, where the way in which the tool is grasped will determine in which manner the new end-effector can be used (e.g., using the tip of the hammer as the prolongation of the robot kinematic chain). Humans have the ability to generate skilled movements by exploiting tool affordances according to the demands of different tasks. It has been suggested that the body schema is plastic because it can incorporate external objects [12]. A hand-held tool, for example, may become so familiar to the user that it feels like a natural extension of the hand [13]. The assistance of external tools has dramatically increased the possibility of achieving more intractable tasks. Finding ways to manipulate the tools before reaching the final target pose is crucial. Therefore, uniting these two phases to complete motion planning is essential to transfer such skills to robots.

Fang *et al.* [14] solved a tool-based manipulation as a two-

This work was funded by the National Key Research and Development Program of China (No. 2022YFB4700701), and by the Major Research Plan of the National Natural Science Foundation of China (No. 92048301), supported by the China Scholarship Council, China (CSC, No. 202006120159). (*Corresponding authors: Yongsheng Gao; Sylvain Calinon.*)

Boyang Ti is with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China, and also with Idiap Research Institute, CH-1920, Martigny, Switzerland (e-mail: 17b908043@stu.hit.edu.cn, tiboyang@outlook.com).

Yongsheng Gao and Jie Zhao are with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: gaogs@hit.edu.cn, jzhao@hit.edu.cn).

Sylvain Calinon is with Idiap Research Institute, 1920 Martigny, Switzerland, and also with Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland (e-mail: sylvain.calinon@idiap.ch).

stage problem consisting of a grasping phase and a manipulation phase, where the optimized task success is achieved by jointly training two networks. The dataset used in their network covers diverse shapes of hammers, allowing it to be generalized to many cases. However, there is still a gap between these two separate phases. Namely, how to transition between them naturally. In this article, we solve the motion planning problem by including the two phases, which allows the robot to anticipate in which manner the tool should be seized by the robot to be further used in an efficient way. Solving the optimization problem in such a way allows the viapoint separating the two phases and their transitions to be carefully chosen. In contrast, solving the problem in each phase separately would not provide the tool affordance capability of our approach. The *tool affordance range* loosely refers to the desired range for the grasping of the tool that will determine how it can further be used in the hitting motion, which is specified as a viapoint constraint. The term *viapoint* refers here to a key position that the robot needs to pass through during the movement. An optimal control problem (OCP) can be solved to generate optimal grasping and manipulation movements by balancing the weights of multiple cost terms, mimicking the human handling of the tool in different postures to facilitate the task according to the requirements. We use directional manipulability to describe the capability of using a tool. The maximization of the directional manipulability is taken as a part of the cost function to influence the tool grasping pose. To offer more choices to handle the tool, we represent the viapoint as a desired range treated as a spatial constraint, which allows the manipulator to choose any gripping point within the range while considering tool affordance. The tool handle gesture is adaptively optimized according to the demand of the subsequent manipulation task to be achieved.

An optimal control problem formulation can be used to solve this tool-use planning challenge. The iterative linear quadratic regulator (iLQR) [15], [16] is of particular interest to our work, as it can typically be used to search for a solution in the joint angle configuration space of the robot. To consider the inequality constraints that occur in the task space introduced by the tool affordance, there are several solvers available for solving constrained optimization problems, such as SNOPT [17], SLSQP [18] and IPOPT [19]. The optimization problems in the field of robotics manipulation can be solved efficiently by the above solution methods. The alternating direction method of multipliers (ADMM) [20] is another popular optimization technique for solving problems that can be decomposed into subproblems with different variables. In this article, inspired by Sindhvani *et al.* work [21] on car parking and obstacle avoidance, we embed ADMM into iLQR to solve a constrained motion planning problem and demonstrate its effectiveness in serial manipulator systems through point-to-range and pick-and-place tasks. The ADMM-iLQR approach that we use is similar to the method proposed in [22], except that we employ an approximation for the optimization function to make it approximately equal to the convex problem in the first ADMM iteration. In [22], the control case of iLQR is considered, which can be solved by dynamic programming.

In our work, the planning case is instead considered, where the solution can be obtained analytically through the batch solution of iLQR, which can efficiently computed in matrix form by only relying on linear algebra. Note that in both control and planning cases, the underlying iterative approach is not guaranteed to lead to a global optimum. However, the local optimal solution that it finds remains an appropriate estimate in a wide range of problems in robotics, including redundant problems with sparse costs in the forms of viapoints as the one that we target in this article.

From the perspective of biomechanics, we can observe the characteristics of human behaviors in impact-aware tasks with tool assistance. For example, as shown in Fig. 1, when we select a grasp to seize an object, we consider the relationship among body configuration, tool, and target poses. When we use a claw hammer to pull out a nail driven into a wooden board, we typically prefer to hold the arm in a tucked-in configuration, making it easier to generate substantial static forces between the nail and the claw. In contrast, when we want to drive a nail into a board in a more stretched configuration, we can easily drive the arm to a higher speed and transfer it to the hammer, as a way to generate maximum momentum (provided that the inertia of the hammer remains constant). In addition to high speeds, the inertial and stiffness properties of the arm configuration play a critical role in such impact-aware tasks, where we need our muscles to absorb the vibrations while controlling the tool’s stability during the impact. In the setup that we will consider, instead of absorbing vibrations solely through the properties of the arm, a pilot hole is additionally used to prevent deviations arising from uneven resistance distribution at the tip of the nail. This external assistance guarantees a smooth execution of the hammering task, which is also widely applied in our daily lives.

The contributions of this article can be summarized as follows:

- (1) We employ an ADMM-iLQR strategy to solve an optimal control problem (OCP) considering tool affordance constraints in impact-aware tasks;
- (2) We introduce a maximum directional manipulability cost in the optimal control framework to optimize the grasping and final manipulation posture;
- (3) We make a comprehensive comparison of different approaches to measuring manipulability in directional tasks;

To fully illustrate the significance of these contributions, we provide simulations to validate our approach and demonstrate its application in an impact-aware task (hammering).

Compared to the existing constrained OCP, the use of ADMM solver in our work is employed to include a tool affordance aspect, specified as a range that can be used during robot manipulation. We show that this convex constraint can be used to enlarge the manipulation capability of the robot. The inequality constraint problem is often studied in the context of obstacle avoidance or joint physical constraint avoidance. We show here that it can also be used as a simple model of tool affordance described as a constraint in OCP.

The rest of the paper is organized as follows. In Section II, we summarize the related works on impact-aware hammering and tool-use tasks, constrained optimal control and

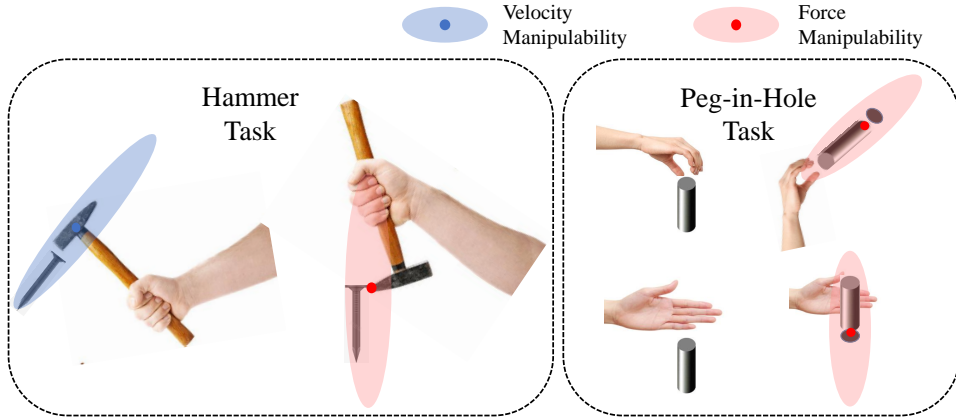


Fig. 1. *Left*: The ways in which tools are seized and used vary according to the characteristics of the task (tool affordances). For hammering a nail, we grasp the hammer in a way that is efficient for the task demands, since the greater the velocity, the deeper the nail can be driven. For nail-pulling, the same hammer is grasped in a different way. Here, the greater the static force applied to the nail, the easier the nail can be pulled out. *Right*: In a peg-in-hole task, we typically choose a comfortable pose to insert a peg into a narrow hole, which depends on the relations between the hand, the peg and the hole (with maximum force manipulability along the insertion direction).

manipulability. In Section III, we introduce the background knowledge related to our method. In Section IV, we introduce our proposed method by deriving the control policy with constraints and analyze the directional manipulability and other soft constraints considered in the cost. In Section V and VI, we compare our approach with methods in which the manipulability is not taken into account, methods using manipulability determinant metrics [6], and method minimizing manipulability distances [8]. In Section VII and VIII, we discuss current limitations and future work.

II. RELATED WORK

A. Impact-Aware Hammering Task and Tool-Use Task

Hammering is a typical impact-aware task, common in daily life, industry, and even in the medical field. The general performance index of hammering is the speed of the hammer head in contact with the object or landmark of interest [23]. Research on hammering has been validated with single joint [24], [25], serial manipulators [26] and humanoid robots [27], [28]. Tsujita *et al.* [27] presented a model of contact dynamic and proposed an evaluation of the impulsive force prediction model. Imran *et al.* [29] focused on minimizing the impulses on robot joints to increase their lifespan. Jujjavarapu *et al.* [25] demonstrated the usability of optimization for exploiting the dynamics of a variable stiffness mechanism (VSM) in hammering tasks to improve task performance with time-varying stiffness profiles. The aspect of vibration absorbing or post-impact system stability has been well investigated in past decades. Fang *et al.* [14] and Tanev *et al.* [30] developed a model that learns policies for task-oriented grasping and task-related tool manipulation with self-supervision. Training neural networks takes time, requires large datasets, and produces results that lack interpretability. It also rarely takes into account the known biomechanical aspects. In our work, we consider the pose configuration for the pre-hammering state, which is achieved by using an optimal control method. This allows the manipulator to limit the control commands while generating important hammering speed in the desired configuration.

A few works have used a learning from demonstration strategy to achieve tool-based manipulation skills learning. Li *et al.* [31] proposed a hierarchical architecture to embed the use of tools in a learning from demonstration framework. Rajeswaran *et al.* [32] incorporated human demonstrations in deep reinforcement learning to provide robust manipulation strategies. In addition to offline learning, Jain *et al.* [33] proposed an online and incremental approach to make the robot acquire manipulation capabilities by interaction with the human user. Involving humans in the learning process is a very efficient way to exploit the redundancy of robot arms, but it can be challenging to imitate human manipulation postures just through kinesthetic or visual demonstrations, especially when the demonstrated skill requires to be close to the robot limits. Kunz *et al.* [34] presented an RRT motion planner that considers joint acceleration limits and potentially non-zero start and goal velocities in hammering tasks. Holladay *et al.* [35] formulated kinematic and force limits as decision variables with various constraints. In this article, we discuss the influence of the impact-aware task requirements on the tool grasping posture and consider this influence in an optimal control formulation of motion planning.

B. Constrained Optimal Control

For real-world application scenarios, different constraints should be considered in the optimization problem, such as dynamic constraints, task constraints (incl. obstacle avoidance), and physical limit constraints (incl. robot joint position and velocity limits). Control-limited differential dynamic programming [36] can be used to handle box inequality constraints at the control level. Chen *et al.* [37] proposed a constrained iLQR approach that transforms the constraints to a cost function by adding a barrier function, which can also be extended to a logarithmic barrier [38].

Nowadays, the development of numerical optimization has made unprecedented progress. ADMM has been proposed to solve optimal control problems by showcasing excellent efficiency and stability. It has been used in diverse areas such as autonomous driving [22], electricity networks [39] or wind

farms [40]. It can handle problems with multiple constraints and it can be more efficient than the distributed gradient method [41], especially in problems in which the objective function can be separated into two or more subproblems that can be solved independently. Hong *et al.* [42] established the global (linear) convergence of the ADMM method for a class of convex objective functions involving any number of blocks. In some special conditions (strong convexity and Lipschitz differentiable), variants of ADMM can ensure linear convergence. Benefiting from the subproblems split strategy, the ADMM algorithm can relieve the computation burden caused by increasing the system dimensions. Also, ADMM is typically less dependent on parameter settings than many other distributed methods and, therefore, is easy to implement. Overall, ADMM can often offer performance comparable to specialized algorithms, and in most cases, the simple ADMM algorithm will be efficient enough to be useful in many robotics problems [20], see Yang *et al.* [43] for a comprehensive survey on the fundamental property and optimization options in ADMM. In robot manipulation tasks, it has also been used in contact-rich optimization problems. Aydinoglu *et al.* [44], [45] proposed using ADMM to exploit its distribution to solve multi-contact dynamics and complicated contact-rich manipulation tasks. Shirai *et al.* [46] proposed a distributed optimization framework based on ADMM to solve contact dynamics. Shorinwa *et al.* [47] used the ADMM method to solve a contact-implicit trajectory optimization problem in a multi-agent system. Wijayarathne *et al.* [48] employed ADMM to generate real-time optimal control in soft contact problems. The above collection of work focused on contact-rich optimization problems. Our work exploits the power of distributed fast solvers to handle constrained optimization problems encapsulating manipulability and tool affordance information, that we exploit in impact-aware tasks. This problem is critical for ensuring tool affordance constraints and optimizing the manipulation configuration of the robotic manipulator to produce optimal impact actions.

C. Manipulability Ellipsoids

Velocity and force manipulability ellipsoids are two descriptors used to measure the ability of an end-effector to perform velocity or force along all task-space directions for a given joint angle configuration. A typical metric based on these descriptors consists of keeping only the volume of the ellipsoid as the source of information [6]. Marić *et al.* [49] used a Riemannian metric to provide a measure of proximity to singular configurations. A thorough study of manipulability ellipsoid on learning, tracking, and transfer has been conducted by Jaquier *et al.* [8], [50], [51], where a control formulation was developed to track desired manipulability ellipsoids, which can be extended to nullspace control to also reach a target pose as primary goal. Here, we use the ADMM-iLQR method to extend the approach to viapoint tasks characterized by a desired range to reach, which is used to pick and manipulate a hammer, by letting the robot exploits the tool affordances to determine optimal picking locations by anticipating the next part of the task that consists of hammering a nail.

In this article, we propose to exploit directional manipulability in the cost function, defined as the length of the manipulability ellipsoid along a particular direction of interest. Tugal *et al.* [52] implement directional manipulability to extend the interaction capabilities of a mobile manipulator on the valve turning task. Kim *et al.* [53] evaluated the maximum directional kinematic capability using optimization-based methods for redundant manipulators. Besides improving the velocity and force, Marais *et al.* [54] exploited this measure to maximize the dynamic manipulability to reject directional disturbances. For the hammering task, the direction of interest corresponds to the orientation of the nail perpendicular to the board. We incorporate this into the cost function to account for its effect on the grasping posture.

III. PRELIMINARIES

A. Alternating Direction Method of Multipliers (ADMM)

Alternating Direction Method of Multipliers (ADMM) is an algorithm designed to combine the decomposability of dual ascent methods with the superior convergence properties of the method of multipliers [20]. The algorithm solves problems in the form

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \quad & c(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{d}, \end{aligned} \quad (1)$$

with variable $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$, where $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times m}$ and $\mathbf{d} \in \mathbb{R}^p$. $c(\cdot)$ and $g(\cdot)$ are assumed to be convex. The difference from the general linear equality-constrained problem is that the variable has been split into two parts, called \mathbf{x} and \mathbf{z} here, with the objective function separable across this splitting. The augmented Lagrangian function of this is

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \quad & c(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^\top (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{d}) \\ & + (\rho/2) \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{d}\|_2^2, \end{aligned} \quad (2)$$

where $\rho > 0$ is a penalty parameter and \mathbf{y} is the dual variable. The formulation can be described compactly by combining the linear and quadratic terms in the augmented Lagrangian and scaling the dual variable. The scaled dual variable is defined as $\boldsymbol{\lambda} = (1/\rho)\mathbf{y}$. Then, by following [20], the iteration process of ADMM becomes

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg \min_{\mathbf{x}} \left(c(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{d} + \boldsymbol{\lambda}^k\|_2^2 \right), \\ \mathbf{z}^{k+1} &= \arg \min_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z} - \mathbf{d} + \boldsymbol{\lambda}^k\|_2^2 \right), \\ \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{d}. \end{aligned} \quad (3)$$

The convergence condition of ADMM solver is divided into:

- Residual convergence: $\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{d} \rightarrow 0$
- Objective convergence: $c(\mathbf{x}^{k+1}) + g(\mathbf{z}^{k+1})$ approaches the optimal value
- Scaled dual variable convergence: $\boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}^*$, where $\boldsymbol{\lambda}^*$ is a dual optimal variable

Consider the following generic constrained convex optimization problem over the variable \mathbf{x}

$$\begin{aligned} \min_{\mathbf{x}} \quad & c(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{C}, \end{aligned} \quad (4)$$

where $c(\cdot)$ and \mathcal{C} are convex. This problem can be written in an ADMM consensus formulation as

$$\begin{aligned} \min_{\mathbf{x}} \quad & c(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{x} - \mathbf{z} = \mathbf{0}, \end{aligned} \quad (5)$$

where g is the indicator function of \mathcal{C} , *i.e.*, $g(\mathbf{z}) = 0$ for $\mathbf{z} \in \mathcal{C}$ and $g(\mathbf{z}) = +\infty$ otherwise, and \mathbf{z} is related to the constraint \mathcal{C} . The augmented Lagrangian becomes

$$\mathbf{L}_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) = c(\mathbf{x}) + g(\mathbf{z}) + (\rho/2) \|\mathbf{x} - \mathbf{z} + \boldsymbol{\lambda}\|_2^2. \quad (6)$$

Therefore, the scaled form of ADMM for this problem is

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg \min_{\mathbf{x}} \left(c(\mathbf{x}) + \rho \|\mathbf{x} - \mathbf{z}^k + \boldsymbol{\lambda}^k\|_2^2 \right), \\ \mathbf{z}^{k+1} &= \Pi_{\mathcal{C}} \left(\mathbf{x}^{k+1} + \boldsymbol{\lambda}^k \right), \\ \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1}. \end{aligned} \quad (7)$$

The \mathbf{x} update involves minimizing $c(\cdot)$ plus a convex quadratic function. The \mathbf{z} update is the Euclidean projection onto \mathcal{C} . In the Appendix, we show that in the case of affine constraints, the projection step corresponds to the projection onto the sublevel set of a convex function.

B. Manipulability Ellipsoids

The velocity and force manipulability ellipsoids are two mutually complementary metrics, that can be obtained by an inverse transformation of the other. Here, we consider velocity manipulability, which describes the volume that can be reached by the end-effector in Cartesian space for a single bounded command in joint space. It will typically describe how the robot will be able to face perturbations at a given time step, which is an important descriptor for kinematically redundant robots as it describes how the selected robot posture can influence the movement of the end-effector in all directions of the task space.

The velocity manipulability of a robot can be found by using the kinematic relationship between task velocity $\dot{\mathbf{x}}$ and joint velocity $\dot{\mathbf{q}}$

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (8)$$

where $\mathbf{q} \in \mathbb{R}^n$ and $\mathbf{J} \in \mathbb{R}^{6 \times n}$ are the joint position and Jacobian of the forward kinematics function describing the robot, respectively. The derivation of the manipulability ellipsoid typically starts by taking the joint velocity limited in a unit sphere by

$$\|\dot{\mathbf{q}}\| = 1, \quad (9)$$

therefore we can obtain

$$\dot{\mathbf{q}}^\top \dot{\mathbf{q}} = \dot{\mathbf{x}}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} \dot{\mathbf{x}} = 1, \quad (10)$$

by using the least-squares inverse kinematics relation $\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} = \mathbf{J}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} \dot{\mathbf{x}}$. We can observe that under this constraint, the matrix $\mathbf{J}\mathbf{J}^\top$ determines the scalability of the velocity manipulability described by

$$\mathbf{M}(\mathbf{q}) = \mathbf{J}\mathbf{J}^\top. \quad (11)$$

The eigenvector with the maximum eigenvalue of the matrix represents the direction with the greatest velocity that the robot can generate based on its current joint angle configuration

and in the limit of joint velocity contained in the unit sphere. A force manipulability can be defined similarly by replacing velocity commands with torque commands.

The above definition of manipulability ignores the physical property of the robot, which can consist of constraints on the commands, or in the form of preferences to set the contribution of each actuator in the kinematic chain. In [50], a diagonal matrix $\mathbf{W} = \text{diag}(\dot{q}_{1,\max}, \dots, \dot{q}_{n,\max})$ is considered, whose elements correspond to the maximum joint velocities of the robot. Here, we normalize the maximum velocity of each joint to consider this weight using $\tilde{\mathbf{W}} = \text{diag}(\dot{q}_{1,\max}, \dots, \dot{q}_{n,\max}) / \max(\dot{q}_{1,\max}, \dots, \dot{q}_{n,\max})$. Therefore, Eq. (11) becomes

$$\tilde{\mathbf{M}}(\mathbf{q}) = \mathbf{J}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^\top \mathbf{J}^\top, \quad (12)$$

which can describe the flexibility of the manipulator in task space by considering its joint angle velocity limits, see [50] for details.

There are a variety of indicators to measure manipulability, in which the manipulability index is the most commonly used, expressed as

$$m = \sqrt{\det(\mathbf{J}\mathbf{J}^\top)}, \quad (13)$$

which is proportional to the volume of the manipulability ellipsoid. The orientation of the main axis of the ellipsoid depends on the joint angle configuration of the robot. In order to design it according to the required orientation in the task space, transmission ratios [9] has first been proposed to measure directional manipulability as a manipulability measure along a specific direction by projecting the ellipsoid onto the direction vector, defined as

$$\alpha = \sqrt{\mathbf{u}^\top (\mathbf{J}\mathbf{J}^\top) \mathbf{u}}, \quad (14)$$

where \mathbf{u} represents the direction of interest. α corresponds to the distance along the vector \mathbf{u} from the origin to the surface of the ellipsoid.

C. Impact aware analysis of the hammering task

Hammering a nail is a typical example of impact-aware tasks, corresponding to a momentum or energy transfer problem. The difference between humans hammering a nail with and without a tool lies in the dynamic model of the task. Without a tool, the momentum transfer occurs between the human and the nail, while when using a tool, it occurs between the tool and the nail. The dynamic model with the tool does not include the human because of the soft connection between the human and the tool. The role of the human is here to stabilize the tool while absorbing the vibration generated at the moment of impact.

We selected a plastic foam material for the hammering task. The material is tight and soft, making it easy to break through. However, the interior of the foam is composed of spherical particles, making the internal material distribution non-uniform, resulting in possible sideslip deviation during the nailing process, as shown in Fig. 2(a). When considering materials such as wood, concrete, or plastic, pilot holes (smaller than the nails) can be used as a way to drive screws (e.g., for furniture assembly). Despite the presence of a pilot hole, the

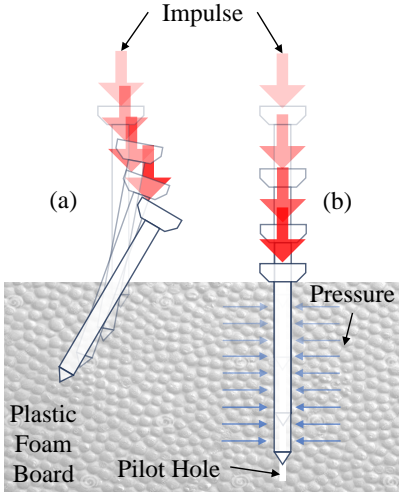


Fig. 2. Illustration of a nail driven into a plastic foam. (a) shows the situation without a preformed insertion shape (pilot hole), where sideslip problems can occur; (b) shows the typical situation with a pilot hole, where the nail can be driven more easily so that it remains straight.

resistance gradually increases during the insertion process due to the compactness and elasticity of the material, as shown in Fig. 2(b), where the blue arrow represents the pressure.

According to the above analysis, the impulse model of hammering with a tool is constructed between the hammer and a nail. With the impulse-momentum theorem [55], the impulse during the hammering task is defined as

$$\int_0^t \mathbf{F}(t) dt = m\mathbf{v}^h(t) - m\mathbf{v}^h(0), \quad (15)$$

where $\mathbf{F}(t)$ represents the varying impulsive force along the nail orientation with the duration of the collision t . $\mathbf{v}^h(0)$ and $\mathbf{v}^h(t)$ represent the velocity along the driving direction of the hammer head before and after the collision. The impulse-momentum theorem states that the impulse applied to an object will equate to the change in its momentum. With the help of the pilot hole, we can neglect other additional momentum losses such as rotational kinetic energy, plastic deformation, and heat. Even though these losses are still present, the presence of the pilot hole greatly reduces their effects. In addition, as the depth of the nail into the board gradually increases, the force due to the squeezed foam increases, which creates resistance during insertion. Based on the above assumption, in the nail hammering task, we assume $\mathbf{v}^h(t) = 0$. Therefore, for a given hammer mass, the velocity before the collision is the key factor to generate a powerful hammering motion.

IV. PROBLEM FORMULATION

A. ADMM for Constrained iLQR

This section uses ADMM optimization to solve an OCP by considering inequality constraints with non-linear dynamical systems and non-quadratic cost functions through an iLQR method. We separated the problem into two parts:

- 1) non-convex problem without inequality constraints;
- 2) an analytical projection step resulting into a fast solver.

The constrained problem is defined as

$$\begin{aligned} \min_{\mathbf{x}_t, \mathbf{u}_t} \quad & \sum_{t=0}^T c_t(\mathbf{x}_t) + \|\mathbf{u}_t\|_{\mathbf{R}_t}^2 \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \\ & \mathcal{C}_x : \quad \mathbf{a}_x \leq \mathbf{l}(\mathbf{x}_t) \leq \mathbf{b}_x, \\ & \mathcal{C}_u : \quad \mathbf{a}_u \leq \mathbf{h}(\mathbf{u}_t) \leq \mathbf{b}_u, \end{aligned} \quad (16)$$

where $c(\mathbf{x})$ is a non-quadratic function of the state and represents the general formulation of the cost related to the hammering task. We introduce the components of the state cost function in Section IV-E (see complete cost function in Eq. (30)). $c(\mathbf{x}, \mathbf{u}) = \sum_{t=0}^T c_t(\mathbf{x}_t) + \|\mathbf{u}_t\|_{\mathbf{R}_t}^2$ corresponds to the term $c(\mathbf{x})$ in Eq. (4), where the variable \mathbf{x} in $c(\mathbf{x})$ covers the system state \mathbf{x} and control \mathbf{u} in our problem. $\mathbf{f}(\cdot)$ is a nonlinear function on the state and control variables. We consider the tool affordance as a convex constraint in the task space. For example, the tool affordance of a hammer can be approximated to the prismatic range of the handle that can be grasped in the task space. The state \mathbf{x}_t is composed of the joint position of manipulator \mathbf{q}_t and its end-effector position $\mathbf{p}_t = \mathbf{f}^{\text{kin}}(\mathbf{q}_t)$, where $\mathbf{f}^{\text{kin}}(\cdot)$ represents the forward kinematics of the manipulator. Here, we use velocity control, so the control command \mathbf{u}_t consists of the joint angle velocity $\dot{\mathbf{q}}_t$. Thus, with this combination of state and control variables, the system is nonlinear. We also added control boundaries for joint velocity limits according to the robot used in the experiment. $\mathbf{l}(\cdot)$ and $\mathbf{h}(\cdot)$ represent the general form of the function between the system state and the task state where the constraint occurs, in our case $\mathbf{l}(\mathbf{x}_t) = \mathbf{p}_t$ and $\mathbf{h}(\mathbf{u}_t) = \mathbf{u}_t$. We define \mathcal{C}_x and \mathcal{C}_u as the inequality constraint on the state and control variables, respectively.

1) *Step 1 of ADMM*: To solve the above optimization, we can separate the inequality constraints from the problem according to Eq. (6) and solve a regularized version of the problem in Eq. (16) without inequality constraints as

$$\begin{aligned} \min_{\mathbf{x}_t, \mathbf{u}_t} \quad & \sum_{t=0}^T c_t(\mathbf{x}_t) + \|\mathbf{u}_t\|_{\mathbf{R}_t}^2 + \frac{\rho_x}{2} \|\mathbf{x}_t - \mathbf{z}_{x,t} + \boldsymbol{\lambda}_{x,t}\|^2 \\ & + \frac{\rho_u}{2} \|\mathbf{u}_t - \mathbf{z}_{u,t} + \boldsymbol{\lambda}_{u,t}\|^2 \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \end{aligned} \quad (17)$$

where the update of $\mathbf{z}_{x/u}$ is the Euclidean projection onto the state and control boundary and $\boldsymbol{\lambda}_{x/u}$ is the scaled dual variable as in Eq. (3).

The constraints for the state of the end-effector position can be defined to occur at any timestep, which in this article corresponds to the timestep of picking up the tool. Therefore, the projection on the state inequality constraint \mathcal{C}_x is active when the pick-up action occurs. In addition, the projection of the control inequality constraint \mathcal{C}_u is valid throughout the motion. We simplify the above formula by adding the following abbreviation to provide compact expressions. $\mathbf{Q}_{r,t}$ and $\mathbf{R}_{r,t}$ are penalty parameters in the form of diagonal matrices for the state \mathbf{x}_t and control command \mathbf{u}_t . We define $\mathbf{x}_{r,t} = \mathbf{z}_{x,t} - \boldsymbol{\lambda}_{x,t}$, $\mathbf{u}_{r,t} = \mathbf{z}_{u,t} - \boldsymbol{\lambda}_{u,t}$. Then, we get the

following expression

$$\begin{aligned} \min_{\mathbf{x}_t, \mathbf{u}_t} \quad & \sum_{t=0}^T c_t(\mathbf{x}_t) + \|\mathbf{u}_t\|_{\mathbf{R}_t}^2 + \|\mathbf{x}_t - \mathbf{x}_{r,t}\|_{\mathbf{Q}_{r,t}}^2 \\ & + \|\mathbf{u}_t - \mathbf{u}_{r,t}\|_{\mathbf{R}_{r,t}}^2 \quad (18) \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t). \end{aligned}$$

We perform a first-order Taylor expansion of the dynamical system $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t)$ around some nominal realization of the plant denoted as $(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t)$, namely

$$\mathbf{x}_{t+1} \approx \mathbf{f}(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}_t}(\mathbf{x}_t - \hat{\mathbf{x}}_t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}_t}(\mathbf{u}_t - \hat{\mathbf{u}}_t)$$

$$\iff \Delta \mathbf{x}_{t+1} \approx \mathbf{A}_t \Delta \mathbf{x}_t + \mathbf{B}_t \Delta \mathbf{u}_t, \quad (19)$$

with $\Delta \mathbf{x}_t = \mathbf{x}_t - \hat{\mathbf{x}}_t$, $\Delta \mathbf{u}_t = \mathbf{u}_t - \hat{\mathbf{u}}_t$, and Jacobian matrices $\mathbf{A}_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_t} \big|_{\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t}$, $\mathbf{B}_t = \frac{\partial \mathbf{f}}{\partial \mathbf{u}_t} \big|_{\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t}$. The concatenated form of the linearized dynamics model can be written as $\Delta \mathbf{x} = \mathbf{S}_x \Delta \mathbf{x}_1 + \mathbf{S}_u \Delta \mathbf{u}$. Since in our problem, we start from $\Delta \mathbf{x}_1 = 0$, therefore, $\Delta \mathbf{x} = \mathbf{S}_u \Delta \mathbf{u}$, see Appendix for details. We then approximate $c(\mathbf{x}_t)$ by a second-order Taylor expansion around $\hat{\mathbf{x}}_t$, namely

$$\begin{aligned} c_t(\mathbf{x}_t) &\approx c_t(\hat{\mathbf{x}}_t) + \mathbf{k}_t^{x\top} (\mathbf{x}_t - \hat{\mathbf{x}}_t) \\ &\quad + \frac{1}{2} (\mathbf{x}_t - \hat{\mathbf{x}}_t)^\top \mathbf{K}_t^{xx} (\mathbf{x}_t - \hat{\mathbf{x}}_t) \quad (20) \\ &\approx \frac{1}{2} (\Delta \mathbf{x}_t - \boldsymbol{\mu}_t)^\top \mathbf{K}_t^{xx} (\Delta \mathbf{x}_t - \boldsymbol{\mu}_t) + \text{const.}, \end{aligned}$$

where $\mathbf{k}_t^{x\top} = \nabla_{\mathbf{x}_t} c_t(\mathbf{x}_t)$ and $\mathbf{K}_t^{xx} = \nabla_{\mathbf{x}_t}^2 c_t(\mathbf{x}_t)$ are the Jacobian and the Hessian matrices, and $\boldsymbol{\mu}_t = -\mathbf{K}_t^{xx-1} \mathbf{k}_t^{x\top}$.

With the above approximation, the cost of the state $c(\mathbf{x})$ can be changed from a non-convex to a convex form. Then, the cost $c(\mathbf{x})$ embedding the control part can be rewritten in batch form by removing the constant terms as $c(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \|\Delta \mathbf{x} - \mathbf{x}_d\|_{\mathbf{K}_t^{xx}}^2 + \|\Delta \mathbf{u} - \mathbf{u}_d\|_{\mathbf{R}}^2$, with $\mathbf{u}_d = -\hat{\mathbf{u}}$. We define $\Delta \mathbf{x}_r = -(\hat{\mathbf{x}} - \mathbf{x}_r)$, $\Delta \mathbf{u}_r = -(\hat{\mathbf{u}} - \mathbf{u}_r)$, where variables without indices t denote the concatenation of the variables for the whole task duration. One iteration of iLQR consists of the following problem:

$$\begin{aligned} \min_{\Delta \mathbf{x}, \Delta \mathbf{u}} \quad & \frac{1}{2} \|\Delta \mathbf{x} - \mathbf{x}_d\|_{\mathbf{K}^{xx}}^2 + \|\Delta \mathbf{u} - \mathbf{u}_d\|_{\mathbf{R}}^2 \\ & + \|\Delta \mathbf{x} - \Delta \mathbf{x}_r\|_{\mathbf{Q}_r}^2 + \|\Delta \mathbf{u} - \Delta \mathbf{u}_r\|_{\mathbf{R}_r}^2 \quad (21) \\ \text{s.t.} \quad & \Delta \mathbf{x} = \mathbf{S}_u \Delta \mathbf{u}. \end{aligned}$$

For such class of problems, the solution can be obtained analytically. This problem follows a similar LQR formulation as Eq. (36) in the Appendix. We focus here on the batch least-squares solution of LQR. Its analytic solution is given by

$$\Delta \hat{\mathbf{u}} = \left(\mathbf{S}_u^\top \left(\frac{1}{2} \mathbf{K}^{xx} + \mathbf{Q}_r \right) \mathbf{S}_u + \mathbf{R} + \mathbf{R}_r \right)^{-1} \left(\frac{1}{2} \mathbf{S}_u^\top \mathbf{K}^{xx} \mathbf{x}_d + \mathbf{S}_u^\top \mathbf{Q}_r \Delta \mathbf{x}_r + \mathbf{R}_r \Delta \mathbf{u}_r \right). \quad (22)$$

Although the problem in this paper is solved using an approximation into a convex problem, the optimization problem for the original problem is still non-convex. Therefore, with the help of the ADMM algorithm in solving such non-convex problems, it does not need to give a feasible solution.

Denoting the solution to Eq. (21) at iteration k_j as $\Delta \hat{\mathbf{u}}_{k_j}$ and given the current nominal state and control $\{\hat{\mathbf{x}}_{k_j}^{k_i}, \hat{\mathbf{u}}_{k_j}^{k_i}\}$, we perform a line search as in Algorithm 2 to determine the next nominal state and control command $\{\hat{\mathbf{x}}_{k_{j+1}}^{k_i}, \hat{\mathbf{u}}_{k_{j+1}}^{k_i}\}$

Algorithm 1: ADMM with constrained iLQR

Input: Set $\mathbf{Q}_r, \mathbf{R}_r, k_{\max}$, primal and dual residual thresholds $r_{p,\max}, r_{d,\max}, c_{\max}$;

Initialize

$k = 0, \mathbf{z}_x^0 = \mathbf{0}, \boldsymbol{\lambda}_x^0 = \mathbf{0}, \mathbf{z}_u^0 = \mathbf{0}, \boldsymbol{\lambda}_u^0 = \mathbf{0}$ and

$r_p^0 \geq r_{p,\max}, r_d^0 \geq r_{d,\max}$

Initialize the nominal state $\hat{\mathbf{x}}^0$ and control $\hat{\mathbf{u}}^0$

Output: optimal $\mathbf{x}^*, \mathbf{u}^*$

while $k_i < k_{\max}^{\text{ADMM}}$ and $r_p^k > r_{p,\max}$ and $r_d^k > r_{d,\max}$ **do**

while $k_j < k_{\max}^{\text{iLQR}}$ and $c(\hat{\mathbf{x}}_{k_j}^{k_i}, \hat{\mathbf{u}}_{k_j}^{k_i}) > c_{\max}$ **do**

 Solve Eq. (21) for $\Delta \mathbf{u}^{k_i+1}, \Delta \mathbf{x}^{k_i+1}$ with

$\mathbf{x}_r^{k_i} = \mathbf{z}_x^{k_i} - \boldsymbol{\lambda}_x^{k_i}$ and $\mathbf{u}_r^{k_i} = \mathbf{z}_u^{k_i} - \boldsymbol{\lambda}_u^{k_i}$;

 Do line search to find $\hat{\mathbf{u}}_{k_j}^{k_i+1}$ and $\hat{\mathbf{x}}_{k_j}^{k_i+1}$;

end

$\mathbf{z}_x^{k_i+1} = \Pi_{C_x}(\hat{\mathbf{x}}^{k_i+1} + \boldsymbol{\lambda}_x^{k_i})$;

$\mathbf{z}_u^{k_i+1} = \Pi_{C_u}(\hat{\mathbf{u}}^{k_i+1} + \boldsymbol{\lambda}_u^{k_i})$;

$\boldsymbol{\lambda}_x^{k_i+1} = \boldsymbol{\lambda}_x^{k_i} + \hat{\mathbf{x}}^{k_i+1} - \mathbf{z}_x^{k_i+1}$;

$\boldsymbol{\lambda}_u^{k_i+1} = \boldsymbol{\lambda}_u^{k_i} + \hat{\mathbf{u}}^{k_i+1} - \mathbf{z}_u^{k_i+1}$;

$r_p^{k_i+1} = \|\hat{\mathbf{u}}^{k_i+1} - \mathbf{z}_u^{k_i+1}\|_2^2 + \|\hat{\mathbf{x}}^{k_i+1} - \mathbf{z}_x^{k_i+1}\|_2^2$;

$r_d^{k_i+1} = \|\mathbf{z}_x^{k_i} - \mathbf{z}_x^{k_i+1}\|_2^2 + \|\mathbf{z}_u^{k_i} - \mathbf{z}_u^{k_i+1}\|_2^2$;

end

Algorithm 2: Line search method with parameter α_{\min}

$\alpha \leftarrow 1$

while

$c(F(\hat{\mathbf{u}} + \alpha \Delta \hat{\mathbf{u}}), \hat{\mathbf{u}} + \alpha \Delta \hat{\mathbf{u}}) > c(\hat{\mathbf{x}}, \hat{\mathbf{u}})$ and $\alpha > \alpha_{\min}$

do

$\alpha \leftarrow \frac{\alpha}{2}$

end

in the iLQR loop, where $\mathbf{x} = F(\mathbf{u})$ is the forward pass function in vector form, built from $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t)$. k_i and k_j represent the iteration indices in the ADMM and iLQR loops, respectively (see Appendix and [56] for details of the derivation).

2) *Step 2 of ADMM:* After Step 1 of ADMM for iLQR, we obtain the converged nominal solution $\{\hat{\mathbf{x}}^{k_i+1}, \hat{\mathbf{u}}^{k_i+1}\}$, where the projection step is derived below. In the case of control bounds, $\mathbf{h}(\mathbf{u}) = \mathbf{u}$ and the projection step corresponds to clipping the values of \mathbf{u} between the bounds \mathbf{a}_u and \mathbf{b}_u , as shown in the Appendix. The projection is $\Pi_{C_u}(\mathbf{u}) = \text{clip}(\mathbf{u}, \text{lower} = \mathbf{a}_u, \text{upper} = \mathbf{b}_u)$. In the case of state bounds, $\mathbf{l}(\mathbf{x}) = \mathbf{x}$ and the projection step corresponds to clipping the values of \mathbf{x} between the bounds \mathbf{a}_x and \mathbf{b}_x , as shown in the Appendix. The projection is $\Pi_{C_x}(\mathbf{x}) = \text{clip}(\mathbf{x}, \text{lower} = \mathbf{a}_x, \text{upper} = \mathbf{b}_x)$. Above all, the projection step ensures that the solution remains feasible with respect to the hard constraints. If a candidate solution violates the constraints, the projection step clips it to satisfy the constraints. Then, the hard constraints problem can be naturally handled in the dual update step. The state projections in our work correspond to the tool affordance range represented by a prismatic constraint. The whole ADMM-iLQR algorithm process is summarized in Algorithm 1.

B. ADMM-iLQR on tool-use motion planning

We consider the tool-use motion planning problem as a viapoint task with a desired range. We use the batch iLQR form in our experiment by considering joint angle velocity commands, with the dynamics expressed as a single integrator to update the joint angle states given velocity commands. A common and intuitive way to incorporate constraints into an optimization problem is to use soft constraints as a weighted cost. For redundant tasks, the constraints can be satisfied for a large range of weights and do not need fine tuning of the weights. However, for more complex tasks requiring trade-offs, the weights need to be adjusted until a satisfactory result is obtained. In such a case, with this adjustment through weights, it can be difficult to ensure that this constraint is strictly satisfied. The tool affordance in our work can be taken as a constraint in the task space at a specific picking time. Since the tool handling is a necessary prerequisite for the completion of the entire manipulation task, it must be taken as a hard constraint. However, the control commands are described in the joint space of the manipulator. We introduce the tool affordance constraint as a hard constraint into the task space and embed the end-effector position into the state function. Thus, the state function consists of joint position and end-effector position $\mathbf{x} = \{\mathbf{q}, \mathbf{p}\}$. For a robot with D articulations, the system consists of a large vector $\{\mathbf{q}, \mathbf{p}\}$ lengths $(D + 3)T$ with control commands length DT , by concatenating the joint angle states \mathbf{q}_t and end-effector states \mathbf{p}_t at different time steps $t \in \{1, \dots, T\}$.

In Eq. (16), the state $\mathbf{l}(\mathbf{x})$ representing the inequality constraint of the problem is defined as the range of the handle which is represented as a prismatic shape, and the affine projection is used to solve this kind of constraint in the task space to ensure that the robot reaches the hammer handle within the available range. Besides the basic requirement of the position reaching cost, the cost term is also composed of the task requirement, including the orientation of grasping, the final hammering direction and the maximum directional manipulability (introduced in the following subsection).

C. Tool usage as an extended link in the robot kinematic chain

After a tool is grasped, the tool can be seen as an external link to the robot, where the new end-effector is closely related to the notion of tool affordance. Before the hammering motion is executed, the soft connection can be seen as a fixed joint, in which the original kinematic chain changes depending on the grasping location. When handling tools, the manipulability of the end-effector of the robot (e.g., gripper) and the end-effector of the tool it holds can change considerably. This kinematic chain can be brought into the optimal control problem to maximize the manipulability of the extended kinematic chain. As shown in Fig. 3, the gripping point location along the tool and the gripper pose used to grasp this tool can result in drastically different manipulability. In contrast, if we use the same grasping orientation, the projection along the end-effector direction is constant, independently of the grasping point on the tool. This phenomenon can be observed by the projection value in Fig. 3(a).

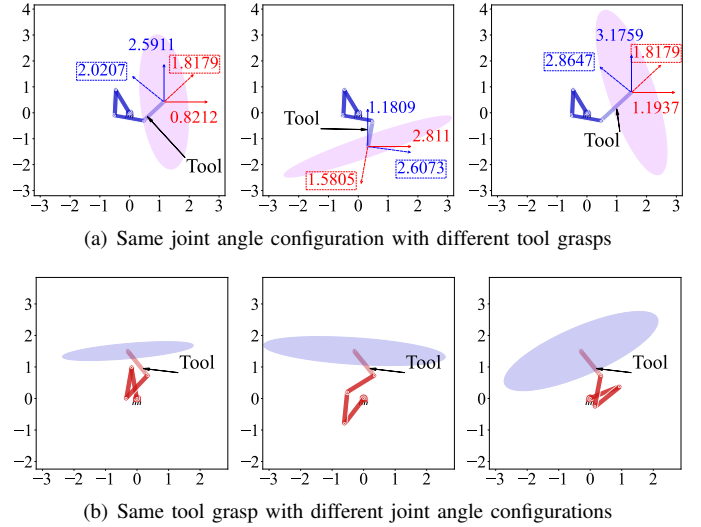


Fig. 3. Influence of the grasping pose on the velocity manipulability of a 3-axis planar arm, where the last link with light color represents the tool. In subfigure (a), a planar arm grasps a tool with the same posture but a different grasping position (left and right plot) and grasping orientation (left and middle plot). The number shows the projection value along the world frame and local end-effector frame (dashed box). Subfigure (b) shows the robot grasping the same tool with different joint angle configurations.

D. Maximization of Velocity Manipulability

In this section, we discuss the different manipulability metrics on the directional task. Directional manipulability is an appropriate metric for the measurement and optimization of manipulability in a particular direction. In addition, there are also other commonly used metrics for measuring the manipulability of manipulators. The method for maximizing the end-effector volume of a manipulability ellipsoid is the most commonly used metric, which does not take into account tool affordance and directionality. Its isotropic expansion can typically lead to failure to meet the specified direction required by the task. In the following, we abbreviate this approach as *Common Metric*. The third method is to set a desired ellipsoid whose main axis is large in the desired direction, as in [50]. This approach cannot guarantee that the manually specified ellipsoid is reachable by the robot. The robot will also consider that matching the secondary axes is equally important as matching the main axis of the ellipsoid. We introduce the above three methods in the comparative study, with respective cost functions defined as

$$c_{\text{man},T}^1 = w_{\text{man}} \|\mathbf{W}^\top \mathbf{J}(\mathbf{q}_T)^\top \mathbf{n}\|_2^{-2}, \quad (23)$$

$$c_{\text{man},T}^2 = w_{\text{man}} (\det(\mathbf{J}(\mathbf{q}_T) \mathbf{W} \mathbf{W}^\top \mathbf{J}(\mathbf{q}_T)^\top))^{-2}, \quad (24)$$

$$c_{\text{man},T}^3 = w_{\text{man}} \left\| \log(\mathbf{M}_d^{-\frac{1}{2}} \tilde{\mathbf{M}}(\mathbf{q}_T) \mathbf{M}_d^{-\frac{1}{2}}) \right\|_F^2, \quad (25)$$

where $c_{\text{man},T}^{\{1,2,3\}}$ represents the cost of different strategies for maximizing manipulability (respectively, directional manipulability, *common metric* and tracking of desired manipulability) at the final time step T , and at other time steps $c_{\text{man},T} = 0$. w_{man} represents the weight in the cost function. $\mathbf{n} \in \mathbb{R}^3$ in $c_{\text{man},T}^1$ represents the desired direction of maximization in the task space. \mathbf{W} represents the velocity capacity of each joint and $\mathbf{M}(\mathbf{q}_T)$ represents the manipulability at the final time

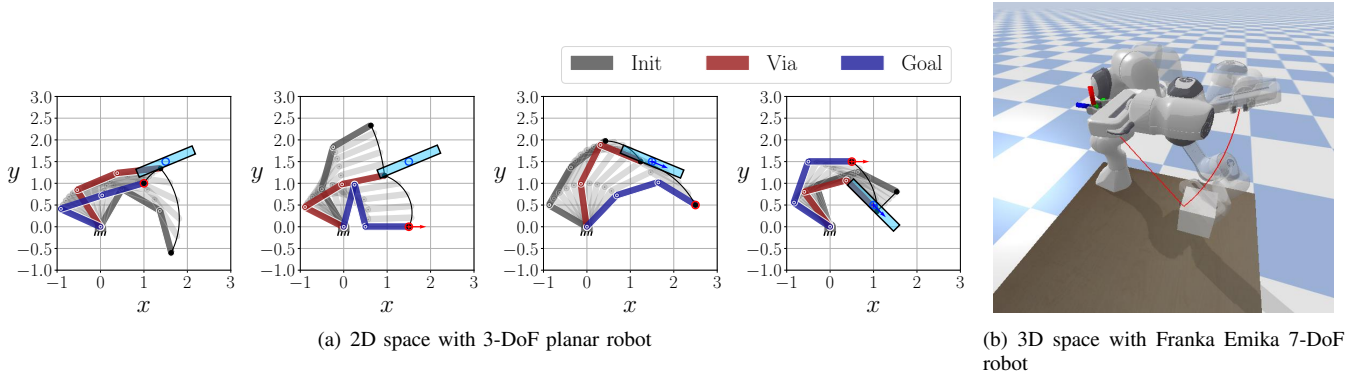


Fig. 4. ADMM-iLQR applied to a viapoint task with a desired range task. In subplot (a), we show four examples of the viapoint with a desired range, where the grey, red and blue robots represent the initial, via and final states of the robot, respectively. The first plot requires the robot to reach the range in cyan and then the final position (red ring). We increase the complexity of the task by setting the final pose specified after passing through the range, arriving at the range while maintaining a desired pose, and finally combining both constraints above into the fourth task scenario. The arrows in the last three plots (blue: viapoint with a desired range; red: final) represent the specified direction. In subplot (b), our approach is verified on a simulated Franka Emika robot.

step T . Both values are defined in Eq. (12). M_d represents the desired manipulability (set manually with an axis having a large value along the task direction).

E. Other task requirements defined in the cost function

We simplify the optimization problem by treating orientation constraints as soft constraints in the cost function by assigning appropriate weights to them. For a two-finger gripper used as robot hand, the best way to grasp a cylindrical object is to keep the grasping direction perpendicular to its axis, ensuring the hammer does not fall off during transportation. Therefore, to define a suitable viapoint with a desired grasping range, we add an orientation constraint

$$c_{o,t'} = w_o e_o^2, \quad \text{with} \quad e_o = \mathbf{R}_r^{y\top} \mathbf{R}_h^z, \quad (26)$$

where \mathbf{R} with y and z superscripts represent y and z column vector of the rotation matrix, the subscript r represents robot and h represents the hammer. $c_{o,t'}$ is used to force the gripper grasping direction y to be perpendicular to the axis z of the hammer handle at hammer picking time step t' , and for other time step $c_{o,t} = 0$. w_o is the weight of the cost.

Before hammering, the robot should take the hammer to the accurate desired pose at the final time step T , where the hammer should be above the nail with the direction of the head parallel to the nail. For the position part, the corresponding cost with weight w_{pos} is defined as

$$c_{\text{pos},T} = w_{\text{pos}} \mathbf{e}_{\text{pos}}^\top \mathbf{e}_{\text{pos}}, \quad \text{with} \quad \mathbf{e}_{\text{pos}} = \mathbf{f}(\mathbf{q}_T) - \mathbf{f}_d. \quad (27)$$

The direction vector, which is a unit vector in three-dimensional space, can be viewed as a point in a Riemannian manifold \mathcal{S}^2 , representing a unit sphere. The distance between two directions is computed on \mathcal{S}^2 , see Appendix for details. The corresponding cost with weight w_{dir} is

$$c_{\text{dir},T} = w_{\text{dir}} \mathbf{e}_{\text{dir}}^\top \mathbf{e}_{\text{dir}}, \quad \text{with} \\ \mathbf{e}_{\text{dir}} = \text{Log}_{\mathbf{v}_T^h}^{\mathcal{S}^2}(\mathbf{v}_T^r) = \arccos(\mathbf{v}_T^h \mathbf{v}_T^r) \frac{\mathbf{v}_T^r - \mathbf{v}_T^h \mathbf{v}_T^r \mathbf{v}_T^h}{\|\mathbf{v}_T^r - \mathbf{v}_T^h \mathbf{v}_T^r \mathbf{v}_T^h\|}. \quad (28)$$

We also need to consider the physical joint limits during motion planning. These joint angle constraints are usually viewed

as a hard constraint in optimization problems. However, it can also restrict the exploration of joint configuration possibilities during iteration, especially when exploring maximum manipulability, where joint angle configurations significantly impact the final result. We noticed that taking joint physical limits as hard constraints can easily make the solution fall into a local optimum. Soft constraints have thus been considered for this part of the cost, represented by an activation function as weight in the cost function, shown as

$$c_{\text{lim},t} = \|\mathbf{q}_L - \mathbf{q}_t\|_{\Lambda_t}^2, \quad \text{with} \\ \Lambda_t^i = \begin{cases} 1 & \text{if } \mathbf{q}_t^i \geq \max(\mathbf{q}_{L_i}) \quad \text{or} \quad \mathbf{q}_t^i \leq \min(\mathbf{q}_{L_i}) \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

so that the constraint cost is activated if the joint position exceeds the limit, where this cost c_{lim} is valid throughout the timeline.

Summarizing the presentation of all the above constraints and cost functions, the final optimization problem related to Eq. (16) becomes

$$\begin{aligned} \min_{\mathbf{x}_t, \mathbf{u}_t} \quad & \sum_{t=0}^T c_{o,t}(\mathbf{x}_t) + c_{\text{pos},t}(\mathbf{x}_t) + c_{\text{dir},t}(\mathbf{x}_t) \\ & + c_{\text{lim},t}(\mathbf{x}_t) + c_{\text{man},t}(\mathbf{x}_t) + \|\mathbf{u}_t\|_R^2 \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \\ & \mathbf{C}_x : \mathbf{a}_x \leq \mathbf{l}(\mathbf{x}_t) \leq \mathbf{b}_x, \\ & \mathbf{C}_u : \mathbf{a}_u \leq \mathbf{h}(\mathbf{u}_t) \leq \mathbf{b}_u, \end{aligned} \quad (30)$$

where $\{\mathbf{a}_x, \mathbf{b}_x\}$ represents the prismatic constraint of the tool affordance range occurring at the tool pick-up timestep, and $\{\mathbf{a}_u, \mathbf{b}_u\}$ represents the joint velocity limit of the manipulator through the entire motion.

V. SIMULATION

To validate ADMM-iLQR in the task space constrained motion planning, we design a viapoint task with a desired range and extend its application to the pick-and-place task in the 2D and 3D space on a planar robot and Franka Emika robot in simulation. Then, we introduce the maximization of manipulability along a desired direction to the optimal control problem and show the difference in manipulability metrics.

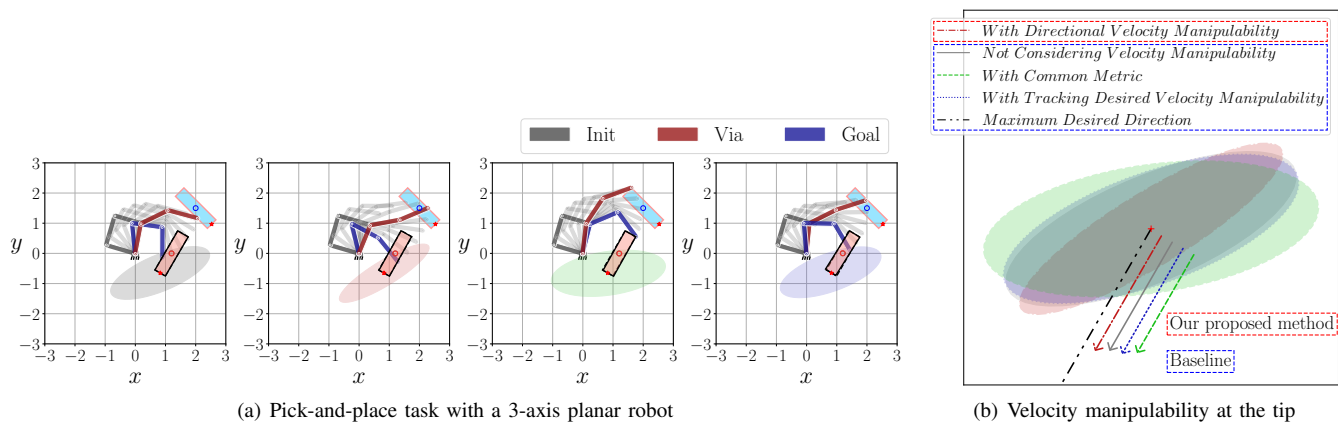


Fig. 5. ADMM-iLQR applied to a pick-and-place task. In (a), we show the results of considering the maximization of manipulability, where the grey, red and blue robots correspond to the descriptions in Fig. 4(a). The robot is required to pick the blue cube and place it in the black dashed cube region while maximizing the velocity manipulability of the robot-tool system along the object orientation. The arrow in the last plot represents the desired direction and the red dot represents the object tip. In (b), we display the velocity manipulability of the robot-tool system separately on the tool tip generated by different strategies. The projections along the desired direction generated by the different approaches are represented with arrows of different line styles. To observe the differences between each projection more clearly, we lined them up in parallel. For a better view of the overlaid ellipses, see the color version of this article.

The optimization algorithm is implemented in Python 3.8, running on a personal computer with Intel(R) i7-10750H CPU 2.60GHz.

A. ADMM-iLQR in a viapoint task with a desired range

We define a square range at the intermediate timestep ($t' = T/2$) to simulate the available range to grasp the hammer handle, then set a desired pose at the final timestep, as shown in Fig. 4. In the planar task, the length of each link of the planar robot is 1.0, and the size of the available range for the hammer handle is defined as a rectangular shape with length 1.4 and width 0.2. The control boundary is defined as $[-4, 4](\text{rad/s})$ for the 2D examples and as $[-3, 3](\text{rad/s})$ for the 3D examples. In the 3D simulation, we import the URDF model of the Frank Emika robot, where the available range is defined as a cube with size 1. The initial control states in both cases are set as $\mathbf{u} = \mathbf{0}$. In the iLQR parameter setting, the final pose reaching weight is $\{2D\text{-position and orientation: } 1e2; 3D\text{-position: } 1e1, \text{orientation: } 1e0\}$, and the control weight for both spaces is $\mathbf{R} = \text{diag}(1e-5)$. The cost threshold is $c_{max} = 1$. The penalty parameter for state and control of the ADMM are set as $\{\mathbf{Q}_r\text{-}2D: \text{diag}(1e1); 3D: \text{diag}(1e0)\}$, $\{\mathbf{R}_r\text{-}2D/3D: \text{diag}(1e-3)\}$, respectively. The threshold is $r_{p,max} = r_{d,max} = 1e-4$. Also, the time step is $\{dt\text{-}2D: 0.01s; 3D: 0.06s\}$, and the trajectory horizon is $T = 100$. The maximum iteration numbers of the iLQR algorithm and ADMM algorithm are $k_{max}^{iLQR} = 10$ and $k_{max}^{ADMM} = 20$, respectively. We initialize the robot close to this range. In the first task, we can observe that the robot can reach the range and the target position. We then add the orientation constraints to increase the complexity of the task, where the robot needs to maintain a specified orientation at the viapoint (with a desired range) and at the final state. The results show that the ADMM-iLQR approach performs well in solving OCP with the state constraint in task space. The computation time in the four scenarios of Fig. 4(a) is 10.03s, 21.43s, 18.57s and 5.34s, respectively. The planar robot can find an optimal joint angle trajectory depending on

the relations between its initial configuration, the viapoint with a desired range and the final target, effectively anticipating the grasp by exploiting tool affordance.

B. ADMM-iLQR with maximization of manipulability in a pick-and-place task

Based on the above simulation results, we add the consideration of the manipulability in the motion planning problem and design a pick-and-place motion scenario as shown in Fig. 5(a). We also compare the maximization of different metrics to show the advantage of directional manipulability, as shown in Fig. 5(b). Here, we treat the tool as an external link in the kinematic chain of the robot after the pick-up stage. In this experiment, the weight of each joint angle actuator is set to be unitary (i.e., with same importance). The algorithm parameter setting is kept the same as in Section V-A, except for the control weight of iLQR, which is changed to $1e-5$. The state and control penalty of ADMM are changed to $\{\mathbf{Q}_r\text{-}2D/3D: \text{diag}(1e-1)\}$, $\{\mathbf{R}_r\text{-}2D: \text{diag}(1e-2); 3D: \text{diag}(1e-3)\}$, respectively. The directional manipulability maximization cost weight is set as $\{w_{man}\text{-}2D: 1e0; 3D: 1e-1\}$. Due to the 3D space simulation task setup close to the actual hammering, its cost function is defined as Eq. (30). The final position reaching weight $w_{pos} = 1e2$, the weight of the via grasping orientation and final direction reaching are $w_o = 1e1$ and $w_{dir} = 1e1$. We perform simulations for different methods in ten different experimental settings. The projection value of the velocity manipulability is shown in Fig. 6. We also plot the primal residual error and cost during ADMM iterations of considering directional manipulability in Fig. 7. The average computation time for these four different methods is 49.349s (directional velocity manipulability), 20.014s (not considering velocity manipulability), 48.607s (*common metric*), 70.134s (tracking desired velocity manipulability).

We verified the results in a 3D space simulation, as shown in Fig. 8. We can observe that directional manipulability has a better performance. The method considering directional

manipulability can sometimes get stuck in poor local minima, which explains that the volume maximization can sometimes perform better in these cases. The method maximizing the *common metric* enlarges the ellipsoid in an isotropic way. For the approach of tracking a desired manipulability [50], it is hard to design a perfect desired manipulability to track, which may be beyond the robot’s physical configuration or put too many constraints on the ellipsoid axes that are not relevant for the task. The average computation time for these four different methods is 161.7s (directional velocity manipulability), 147.3s (not considering velocity manipulability), 163.1s (*common metric*), 210.5s (tracking desired velocity manipulability).

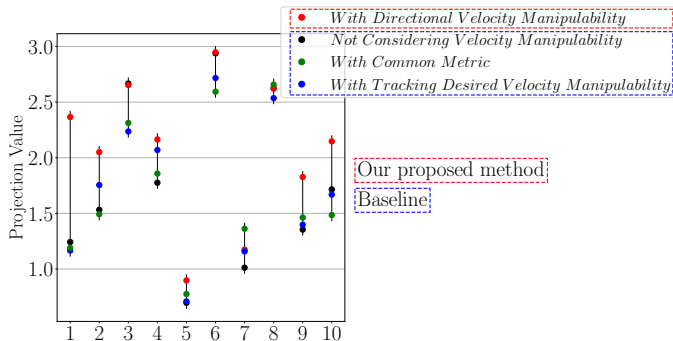


Fig. 6. Projection of the velocity manipulability along the desired direction in ten simulations.

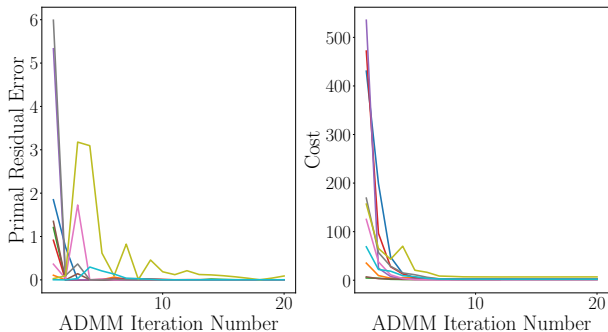
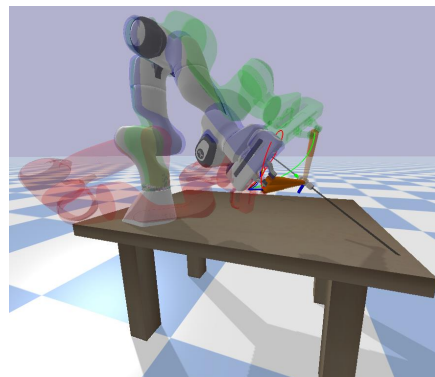


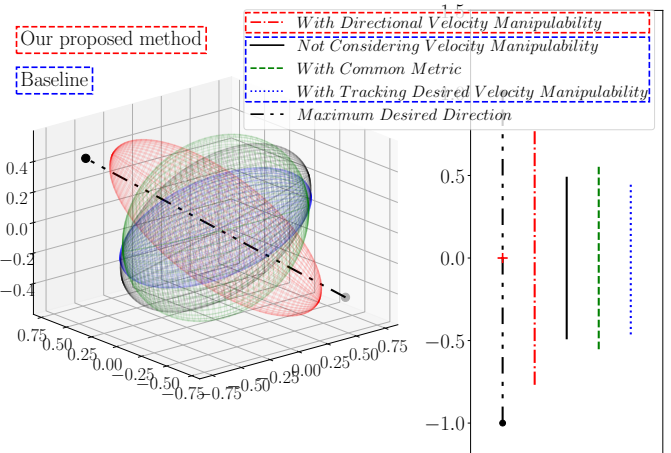
Fig. 7. Primal residual error and cost during ADMM iterations of considering directional manipulability in ten simulations.

VI. ROBOT EXPERIMENT

We applied our approach to an impact-aware task requiring a robot to hammer a nail into a foamed plastic base. We evaluated the approach with a 7-axis Franka Emika manipulator by using a joint velocity controller on the same PC platform as the simulation. We chose the foamed plastic as the pegboard, which is a soft material with a structure that is still firm inside. This material can mitigate the impulse of the hammering impact on the type of robot we used for the experiment. Then, we used a nail to create a pilot hole by inserting a nail fully and straight into the vacant space of the foam platform. To simulate the connection between the human hand and the hammer, we stuck two pieces of sponge on the two grippers of the robot hand. The hammer is made of ABS material by 3D printing, with the handle designed as a cylinder



(a) Joint angle configurations in the pre-hammering state using different methods to maximize the velocity manipulability along the desired direction.



(b) Velocity manipulability visualized in 3D space (left), with its projection along the desired direction (right).

Fig. 8. ADMM-iLQR with maximization of different manipulability metrics in a simulated pick-and-place task. In (a), the red robot corresponds to the use of the directional velocity manipulability. The robot in solid lines does not consider velocity manipulability. The green robot employed the *common metric*. The blue robot tracked a desired velocity manipulability. For a better view of the transparent overlays of the robots, see the color version of this article.

TABLE I
MAXIMUM VELOCITY FOR EACH JOINT OF THE FRANKA EMIKA ROBOT*

Joint index	Joint 1-4	Joint 5-7
\dot{q}_{max}	2.1750 rad/s	2.610 rad/s

*https://frankaemika.github.io/docs/control_parameters.html

for easy grasp by a simple two-finger gripper. The grasping range is defined as a cuboid whose length is from 50 mm to 230 mm starting from the head and whose width and height are 1 mm. The setup is depicted in Fig. 9. The control inequality constraint of the problem is set as the Franka Emika safety joint velocity limit. The time step, trajectory horizon, initial control command, threshold, and other parameters setting of the iLQR algorithm and ADMM algorithm are the same as those of the 3D space simulation parameters setting in Section V-B. The hammer picking-up time step is set as $t' = T/2$.

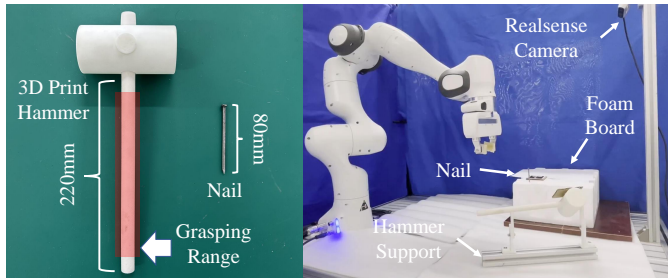
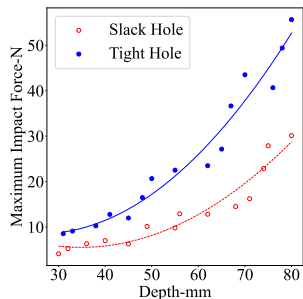
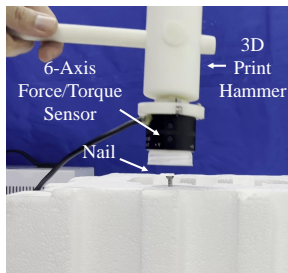


Fig. 9. Experimental setup. The red rectangle represents the available grasping points.



(a) Relation between the maximum impact force and the nail insertion depth



(b) Experimental setup for impact force sampling

Fig. 10. Relationship between the insertion depth of the nails and the impact force for two levels of tightness. The red dashed line and the blue solid line represent the modeled relation based on the 15 sets of data sampled in the slack and tight cases.

A. Hammering Force Test

We chose two kinds of tightness to evaluate our approach. Due to the elasticity of the hammering platform, we could adjust the tightness by repeatedly inserting and pulling the nail to enlarge the pilot hole. Due to experimental equipment constraints, we could not record impact forces accurately. Indeed, as the connection between the grippers and the tool remains soft, it is hard to fully transmit the impact force to the robot’s sensors. To cope with this issue, we modeled the relationship between the insertion depth of the nails and the maximum impact force. We then exploited this model to derive the impact force from the observed insertion depth of the nails. We sampled 15 sets of impact forces with the insertion depth of the nails, for each different tightness state (blue dots represent tight holes, red circles represent slack holes). A quadratic regression model was used to encode this relationship. The sampling data and setup are shown in Fig. 10. Note here that if an external force sensor were mounted on the head of the hammer, it would increase the weight of the hammer head and cause a bias for the gravity center, further increasing the grip instability and making subsequent impact experiments difficult.

B. Hammering Nails

For the real robot experiment, we additionally used the insertion depth of the nails as a way to visualize the differences in velocity manipulability produced by different strategies during impact. We randomly set the position of the hammer and nail within the workspace of the Franka Emika robot

and fixed the initial joint angle configuration of the robot. Inspired by the definition of manipulability, we normalized the joint velocity vector, which is calculated by inverse kinematics to keep the hammering direction along the nail axis. We neglected the direction bias during the impact process. The normalized motion in joint space results in a difference in the manipulability ellipsoid on the end-effector. In the experiment, we used a joint velocity controller to drive the system. We kept the hammering velocity of each joint constant during the impact process. In the hitting stage, the robot first drives the hammer back two time steps at strike speed so that there is enough space to reach the target speed at the state of hitting the nail.

Screenshots of the hammering process are shown in Fig. 11, depicting four typical cases, including the entire trajectory and the grasping and pre-hammering state in Fig. 12. Additional experiment details are presented in the supplementary video. Due to the limited workspace available for the experiment, we chose twelve locations of hammer and nail that were reachable to accomplish the entire impact-aware task (six experiments for each manipulability metric, using two different tightness values). We used the insertion depth of the nails as a performance indicator for each method. In addition, we contrasted the use of ADMM-iLQR for motion planning to the use of kinesthetic demonstrations (Human Demonstration baseline). The use of kinesthetic demonstrations is another intuitive and effective way to accomplish manipulation tasks when operating a robotic manipulator. Therefore, we used this as one of the baselines. But unlike demonstrations in point-to-point movements, impact-aware tasks not only require reaching a set of desired viapoints but also ensuring efficient motion of the manipulator and efficient exploitation of the kinematic redundancy to improve manipulability. To realize that, the user dragged the arm of the robot to grasp the hammer and then put the arm into a pre-hammer state.

TABLE II
THE PROJECTION VALUES ALONG THE DESIRED DIRECTION OF THE TWELVE TIMES HITTING EXPERIMENT

Index	A	B	C	D	E
S1	0.5894	0.7567	0.6677	0.6276	0.6084
S2	0.5224	0.7168	0.6322	×	0.5057
S3	0.6267	0.7346	0.5601	0.5158	0.5843
S4	0.7443	0.8800	0.8285	0.8257	0.8534
S5	0.6838	0.7700	0.6868	0.5744	0.5934
S6	0.7252	0.7505	0.6715	0.6305	0.6942
T1	0.5674	0.7579	0.6652	0.6223	0.6197
T2	0.5970	0.7507	0.7395	0.6921	0.6485
T3	0.6900	0.7962	0.6157	0.6234	0.5456
T4	0.5849	0.7367	0.6375	0.5875	0.6813
T5	0.7367	0.8884	0.8207	0.8303	0.8434
T6	0.4948	0.6227	0.5863	×	0.4923

(A) Not considering velocity manipulability; (B) Maximization of directional manipulability; (C) Common metric; (D) Desired ellipsoid tracking method; (E) Human demonstration.

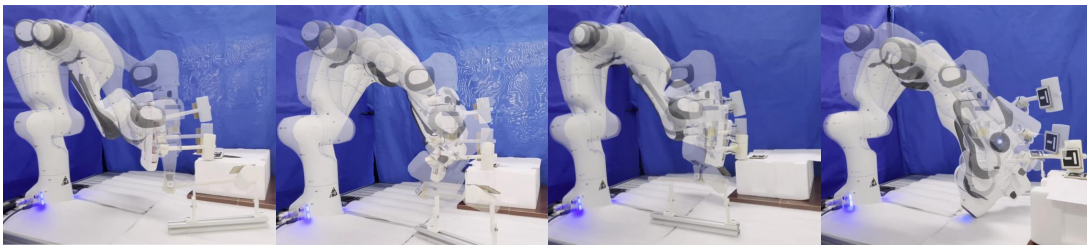


Fig. 11. Screenshot of four typical cases using the ADMM-iLQR with maximizing the directional velocity manipulability in the hammering task.

TABLE III
THE INSERTION DEPTH OF THE NAILS

Index	A	B	C	D	E
S1	63	75	70	70	71
S2	65	71	69	×	63
S3	68	73	69	60	65
S4	72	80	80	76	80
S5	70	80	75	68	70
S6	71	74	75	73	73
T1	39	66	55	43	45
T2	40	65	65	64	54
T3	45	70	48	47	34
T4	41	61	45	46	63
T5	45	75	68	68	68
T6	35	58	42	×	31

Indices A/B/C/D/E correspond to the indices in Table II; values in *mm*.

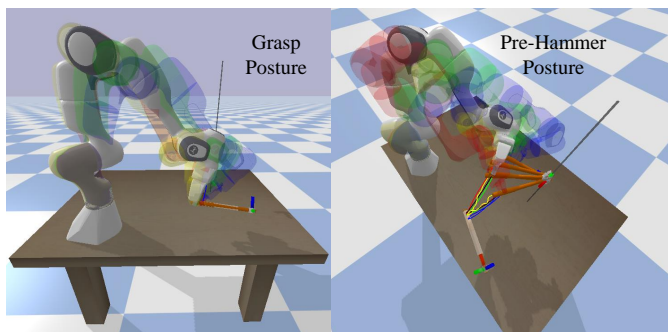


Fig. 12. Trajectories in the case of different maximization strategies. The left and right plot represent the grasping posture and the final pre-hammering state, respectively. The methods for different colored robots are the same as shown in Fig. 8(a), with the addition of the baseline using human demonstration (yellow robot). For a better view of the transparent overlay of the robots, see the color version of this article.

C. Discussion of results

We list the maximum projection along the desired direction in Table II and the corresponding insertion depth of the nails in Table III. We can observe that the optimization considering the directional manipulability has the highest value in the twelve experiments. However, in the depth case S4, B/C/E all have the same performance because the impact force has exceeded the maximum limit of definable depths. In Table III case S6, these methods all perform similarly, but C has the highest value. Indeed, although the pilot hole smooths out the hammering task, the rotational and plastic deformation during hammering remains, which could still cause in some cases the nail to be

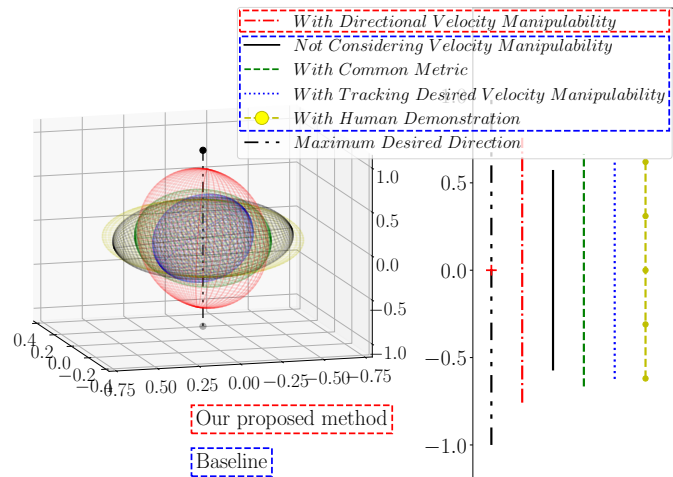


Fig. 13. Velocity manipulability of the robot-hammer system at the impact point, generated by different methods. These ellipsoids represent the velocity manipulability of the robot-hammer system, whose end-effector is located at the hammer head. The projection results are displayed on the right part for the method for directional velocity manipulability (in red dash-dotted line) without considering velocity manipulability (in black solid line), for *common metric* velocity manipulability (in green dashed line), for desired velocity manipulability tracking (in blue dotted line), and from human demonstration (in yellow dash-dotted line). For a better view of the transparent overlaid ellipsoids, see the color version of this article.

rotated by the hammering deflection. It is also the reason for the inconsistency in the relationship between the projection values and the insertion depth of the nails for the different methods. We selected one case from the four typical cases in Fig. 11 to illustrate the entire trajectory and plot the grasping tool's gestures and the pre-hammering robot's configuration in Fig. 12. We plot the velocity manipulability of the robot in its hammering pose, and its projection along the desired direction of the hammering on the right, where we can find that the projection of the directional manipulability is the largest. The primal residual error and cost during ADMM iterations of considering directional manipulability are shown in Fig. 14. We also plot the relation between the projection and the insertion depth in Fig. 15, where we can find that its trend is similar to Fig. 10(a). The gradient of the maximum impact force increases gradually as the insertion depth increases. To observe the performance of the different methods, the box plot in Fig. 16 additionally compares the insertion depth of the nails using different manipulability metrics. We can observe that almost all the methods perform well in the slack nail driving case. In the tight case, the human demonstration method is randomly distributed due to the robot configuration not being

considered, where the user typically prefers to use an easily dragged posture in the demonstration. The directional manipulability performs best, and the *tracking desired ellipsoid* strategy does not work in some task settings. The *common metric* strategy focuses on volume maximization, which only worked efficiently in some cases. The average computation time for these four different methods is 56.8s (directional velocity manipulability), 44.1s (not considering velocity manipulability), 50.7s (common metric), 97.5s (tracking desired velocity manipulability). There is a big difference between the actual scene and the simulation on the hammer's placement and the hitting direction setup, significantly impacting the computation time. In the simulation, the hammer and the hitting direction are randomly set in the table workspace. However, in the actual task, the hitting direction should be set perpendicular to the ground at a specific height. Those result in a significant difference in the computation time of both cases. Since the motion planning of the final hitting motion is not considered in the optimization process, the failures due to reaching the joint limits (listed in the table) all occurred during the impact phase. We can observe that tracking the desired manipulability always leads the robot to singular configurations close to joint angle limits, which makes it easy for the impact motion to exceed the joint physical limits (see hollow ring of the blue box in Fig. 16). This kind of configuration has a significant influence on the subsequent hammering action. Therefore, we defined this situation as a failure case.

In addition to the above successful cases, we also encountered some failure cases that were not considered in the analysis of the results. In the hammer grasping phase, due to the limitations of the experimental equipment, we employed a simple gripper. When handling the hammer, the gripper cannot guarantee a very stable grasp because the gravity is mainly distributed on the head of the hammer. Therefore, the held hammer can sometimes be rotated or shifted when grasping it or in the subsequent movement process. In addition, when the robot picks up the hammer away from the support frame, the grasp can also be affected by perturbations caused by the support frame. In the proposed approach, the tool handled by the robot is considered as an external link of the robot kinematic, so the above disturbances can degrade the reaching of the final hitting pose. Also, as our contribution focused on the tool affordance aspects, we did not consider the collision avoidance problem extensively. While collision is an additional possible failure factor, it rarely happened in practice with the problem setup that we proposed, except in the comparison case of tracking a desired manipulability.

VII. DISCUSSION

The approach presented in this paper adopts the ADMM-iLQR approach to solve motion planning problems exploiting tool affordances by comparing different strategies for maximizing manipulability along specific task directions with a constrained optimal control approach. The gripping point for the tool is treated as a viapoint to reach within a desired range, which provides the flexibility of choosing the grasping

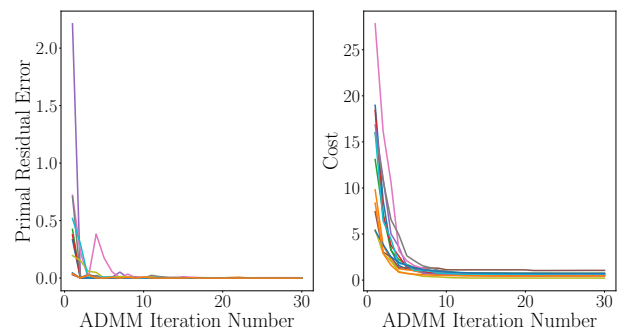


Fig. 14. Primal residual error and cost during ADMM iterations when considering directional manipulability in twelve hammering experiments.

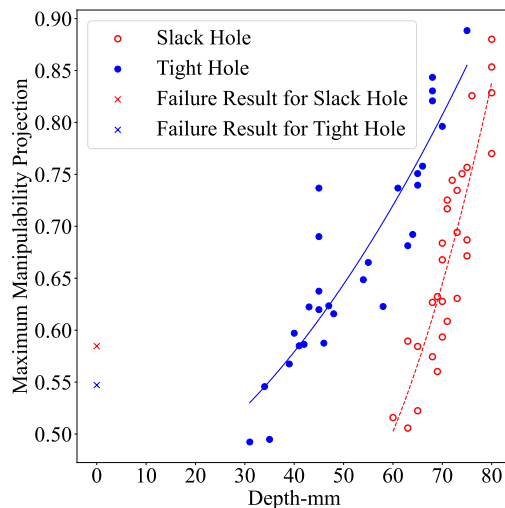


Fig. 15. Relations between the nail insertion depths and the maximum projection of manipulability along the desired direction.

posture, so that the robot can automatically select an ideal gripping point by anticipating how the tool will be further used after grasping it. Thus, the grasping hand pose and the body configuration are also determined by the final hitting direction, allowing the motion planning process to use the tool by considering the requirements of both the task and the environment in which the motion takes place. By considering the directional velocity manipulability, the manipulator can

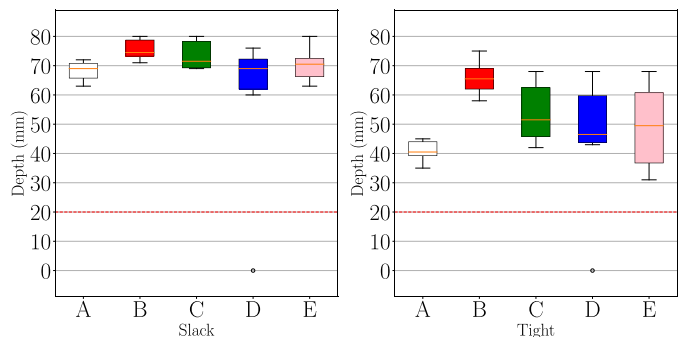


Fig. 16. Nail insertion depths in slack and tight cases. The boxplots show the results for the ADMM-iLQR motion planning using different methods, where the indices A/B/C/D/E correspond to the indices in Table II. Circle points correspond to the cases where the robot finds solutions over the joint limits in the motion planning process. The red dashed line represents the pre-depth of the nail in the plastic foam.

give a large local speed to the handheld tool, effectively generating a large momentum during the impact.

In contrast to the scenario where a robot hand would directly act on the nail, using a tool such as a hammer involves an additional soft connection in the kinematic chain, where the tool can be used as an additional link in the kinematic chain. The soft connection leads to a quasi isolation of the body inertia from the system during the impact, where the tool inertia has the most important influence. The assumption for this category of hitting tasks using tools is that the robot inertia can be ignored during the impact. In contrast, the robot posture will influence the capability of the robot to use the handheld tool. Velocity manipulability is thus maximized rather than inertial or force manipulability.

In the experiment, we used a simple two-finger gripper that cannot ensure perfectly stable grasping and hammering. Although the two-finger can meet our simple requirement, we still sporadically observed unexpected outcomes, such as the hammer falling off during the impact motion, or a deviation in the hitting impact. In future work, we aim to ameliorate this aspect by combining the approach with a visual corrective feedback component.

VIII. CONCLUSION

In this article, we proposed a constrained optimal control approach considering directional manipulability as a way to improve the performance of using tools in impact-aware tasks. We relied on ADMM-iLQR to efficiently solve constrained optimal control problems, by considering the notion of tool affordance, in the form of task space constraints in which the robot can determine an optimal configuration to grasp the tool by taking into account what the tool can offer when held by the robot. The approach has been tested for tasks requiring impacts. We demonstrated in various simulations and on a real robot experiment that the robot was able to solve this challenge with viapoint references specified as a desired range to reach with preferred orientations, while maximizing the velocity manipulability along desired task directions. Our experimental results showed that the consideration of directional manipulability in our approach achieved better hammering performance than the baselines, including the use of human demonstrations, the use of a common scalar manipulability index, and the tracking of desired manipulability ellipsoids.

ACKNOWLEDGMENTS

This work was funded by the National Key Research and Development Program of China (No. 2022YFB4700701) and by the Major Research Plan of the National Natural Science Foundation of China (No.92048301), supported by the China Scholarship Council (CSC, No.202006120159). The authors would like to thank Dr Hakan Girgin for his suggestions and support on the use of ADMM.

APPENDIX

PROJECTION ONTO SUBLEVEL SET OF A CONVEX FUNCTION

Solutions to the optimization problems in the form $\min_x \|x - z\|$ s.t. $f(x) \leq u$ is called a projection onto the sublevel set

of a convex function and denoted as $\Pi_C(z)$ with the convex set $C = \{x \mid f(x) - u \leq 0\}$. This projection is given by $\Pi_C(z) = (I + \mu \partial f)^{-1}z$, where μ is an arbitrary solution of $f(\Pi_C(z)) = t$.

Affine projections: If we let $f(x) = a^\top x$ with $C = \{x \mid a^\top x \leq u\}$, then $\partial f = a$, hence

$$\begin{aligned} (I + \mu \partial f)\Pi_C(z) &= z, \\ \Pi_C(z) + \mu a &= z, \\ \Pi_C(z) &= z - \mu a. \end{aligned} \quad (31)$$

Then, we find an arbitrary solution for μ from $f(\Pi_C(z)) = u$ as

$$\begin{aligned} f(z - \mu a) &= u, \\ a^\top (z - \mu a) &= u, \\ \mu &= \frac{a^\top z - u}{\|a\|_2^2}. \end{aligned} \quad (32)$$

One can find the projection onto $C = \{x \mid l \leq a^\top x\}$ by letting $f(x) = -a^\top x$ and by replacing u with $-l$.

Note that if we take x to be one-dimensional and $a = 1$, then the problem becomes the projection onto bounds $C = \{x \mid x \leq u\}$, which can be solved with the *clipping* operator defined by $\Pi_C(z) = \min(z, u)$. This can be extended to lower bounds with $\Pi_C(z) = \max(\min(z, u), l)$. The projection of affine hyperplane is defined as

$$\begin{aligned} \text{Constraints: } & l \leq a^\top x \leq u \\ \text{Projection: } \Pi_C &= \begin{cases} x & \text{if } l \leq a^\top x \leq u, \\ x - (a^\top x - u) \frac{a}{\|a\|_2^2} & \text{if } a^\top x > u, \\ x - (a^\top x - l) \frac{a}{\|a\|_2^2} & \text{if } a^\top x < l. \end{cases} \end{aligned} \quad (33)$$

BATCH SOLUTION OF LINEAR QUADRATIC REGULATOR (LQR)

We can describe all states x in a trajectory, concatenated in a big vector, as a linear function of the initial state x_1 and the applied control commands u as

$$x = S_x x_1 + S_u u, \quad (34)$$

where

$$S_x = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{T-1} \end{bmatrix}, S_u = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ B & \mathbf{0} & \dots & \mathbf{0} \\ AB & B & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A^{T-2}B & A^{T-3}B & \dots & B \end{bmatrix}. \quad (35)$$

To track a desired reference x_d , the cost function can be defined as

$$c(x, u) = (x - x_d)^\top Q (x - x_d) + u^\top R u, \quad (36)$$

where $Q = \text{blockdiag}(Q_1, Q_2, \dots, Q_T)$ and $R = \text{blockdiag}(R_1, R_2, \dots, R_{T-1})$. Substituting (34) and (35) to (36), we obtain

$$\begin{aligned} c(x, u) &= u^\top (S_u^\top Q S_u + R) u + 2u^\top S_u^\top Q (S_x x_1 - x_d) \\ &\quad + (S_x x_1 - x_d)^\top Q (S_x x_1 - x_d). \end{aligned} \quad (37)$$

The optimal control command \mathbf{u} is computed by differentiating (37) with respect to \mathbf{u} and equating to zero, providing the batch form solution

$$\mathbf{u} = (\mathbf{S}_u^\top \mathbf{Q} \mathbf{S}_u + \mathbf{R})^{-1} \mathbf{S}_u^\top \mathbf{Q} (\mathbf{x}_d - \mathbf{S}_x \mathbf{x}_1). \quad (38)$$

\mathcal{S}^d MANIFOLD

The exponential and logarithmic maps corresponding to the distance

$$d(\mathbf{x}, \mathbf{y}) = \arccos(\mathbf{x}^\top \mathbf{y}), \quad (39)$$

with $\mathbf{x}, \mathbf{y} \in \mathcal{S}^d$ can be computed as (see also [57])

$$\mathbf{y} = \text{Exp}_{\mathbf{x}}(\mathbf{u}) = \mathbf{x} \cos(\|\mathbf{u}\|) + \frac{\mathbf{u}}{\|\mathbf{u}\|} \sin(\|\mathbf{u}\|), \quad (40)$$

$$\mathbf{u} = \text{Log}_{\mathbf{x}}(\mathbf{y}) = d(\mathbf{x}, \mathbf{y}) \frac{\mathbf{y} - \mathbf{x}^\top \mathbf{y} \mathbf{x}}{\|\mathbf{y} - \mathbf{x}^\top \mathbf{y} \mathbf{x}\|}. \quad (41)$$

REFERENCES

- [1] E. Bizzi, N. Accornero, W. Chapple, and N. Hogan, "Posture control and trajectory formation during arm movement," *Journal of Neuroscience*, vol. 4, no. 11, pp. 2738–2744, 1984.
- [2] D. A. Rosenbaum, L. D. Loukopoulos, R. G. Meulenbroek, J. Vaughan, and S. E. Engelbrecht, "Planning reaches by evaluating stored postures," *Psychological review*, vol. 102, no. 1, p. 28, 1995.
- [3] M. Desmurget and C. Prablanc, "Postural control of three-dimensional prehension movements," *Journal of neurophysiology*, vol. 77, no. 1, pp. 452–464, 1997.
- [4] I. Cos, N. Bélanger, and P. Cisek, "The influence of predicted arm biomechanics on decision making," *Journal of neurophysiology*, vol. 105, no. 6, pp. 3022–3033, 2011.
- [5] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *1st Int. Symp. on Robotics Research, Bretton Woods, NH, 1984*, 1984, pp. 735–348.
- [6] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.
- [7] T. Yoshikawa, "Dynamic manipulability of robot manipulators," *Transactions of the Society of Instrument and Control Engineers*, vol. 21, no. 9, pp. 970–975, 1985.
- [8] N. Jaquier, L. D. Roza, D. G. Caldwell, and S. Calinon, "Geometry-aware tracking of manipulability ellipsoids," in *Robotics: Science and Systems*, no. CONF, 2018.
- [9] S. L. Chiu, "Task compatibility of manipulator postures," *The international journal of robotics research*, vol. 7, no. 5, pp. 13–21, 1988.
- [10] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.
- [11] D. A. Norman, *The psychology of everyday things*. Basic books, 1988.
- [12] H. Head and G. Holmes, "Sensory disturbances from cerebral lesions," *Brain*, vol. 34, no. 2-3, pp. 102–254, 1911.
- [13] K. Vaesen, "The cognitive bases of human tool use," *Behavioral and brain sciences*, vol. 35, no. 4, pp. 203–218, 2012.
- [14] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 202–216, 2020.
- [15] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.
- [16] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *First International Conference on Informatics in Control, Automation and Robotics*, vol. 2. SciTePress, 2004, pp. 222–229.
- [17] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.
- [18] D. Kraft, "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- [19] A. Wachter, "An interior point algorithm for large-scale nonlinear optimization with applications in process engineering," Ph.D. dissertation, Carnegie Mellon University, 2002.
- [20] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [21] V. Sindhwani, R. Roelofs, and M. Kalakrishnan, "Sequential operator splitting for constrained nonlinear optimal control," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 4864–4871.
- [22] J. Ma, Z. Cheng, X. Zhang, M. Tomizuka, and T. H. Lee, "Alternating direction method of multipliers for constrained iterative lqr in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 031–23 042, 2022.
- [23] T. Izumi and Y. Hitaka, "Hitting from any direction in 3-d space by a robot with a flexible link hammer," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 296–301, 1997.
- [24] M. Garabini, A. Passaglia, F. Belo, P. Salaris, and A. Bicchi, "Optimality principles in variable stiffness control: The vsa hammer," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 3770–3775.
- [25] S. S. Jujjavarapu, H. Manjunatha, and E. T. Esfahani, "A variable stiffness end-of-arm tooling mechanism to enhance dynamic task capabilities of robotic manipulators," *Journal of Mechanisms and Robotics*, vol. 15, no. 6, p. 061003, 2023.
- [26] V. Romanyuk, S. Soleymanpour, and G. Liu, "A multiple working mode approach to robotic hammering: Analysis and experiments," *Robotica*, vol. 40, no. 3, pp. 570–582, 2022.
- [27] T. Tsujita, A. Konno, S. Komizunai, Y. Nomura, T. Owa, T. Myojin, Y. Ayaz, and M. Uchiyama, "Analysis of nailing task motion for a humanoid robot," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1570–1575.
- [28] T. Matsumoto, A. Konno, L. Gou, and M. Uchiyama, "A humanoid robot that breaks wooden boards applying impulsive force," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5919–5924.
- [29] A. Imran and B.-J. Yi, "A closed-form analytical modeling of internal impulses with application to dynamic machining task: Biologically inspired dual-arm robotic approach," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 442–449, 2017.
- [30] A. Tanev, P. Köhne, and R. Dillmann, "Joint policy optimization for task-oriented grasping with a humanoid gripper," in *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE, 2022, pp. 1–6.
- [31] W. Li and M. Fritz, "Teaching robots the use of human tools from demonstration with non-dexterous end-effectors," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 547–553.
- [32] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.
- [33] R. Jain and T. Inamura, "Learning of usage of tools based on interaction between humans and robots," in *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*. IEEE, 2014, pp. 597–602.
- [34] T. Kunz and M. Stilman, "Probabilistically complete kinodynamic planning for robot manipulators with acceleration limits," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3713–3719.
- [35] R. Holladay, T. Lozano-Pérez, and A. Rodriguez, "Force-and-motion constrained planning for tool use," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7409–7416.
- [36] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.
- [37] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative lqr for on-road autonomous driving motion planning," in *2017 IEEE 20th International conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 1–7.
- [38] J. Chen, W. Zhan, and M. Tomizuka, "Autonomous driving motion planning with constrained iterative lqr," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 244–254, 2019.
- [39] P. Braun, T. Faulwasser, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann, "Hierarchical distributed admm for predictive control with applications in power networks," *IFAC Journal of Systems and Control*, vol. 3, pp. 10–22, 2018.
- [40] W. Liao, P. Li, Q. Wu, S. Huang, G. Wu, and F. Rong, "Distributed optimal active and reactive power control for wind farms based on

admm,” *International Journal of Electrical Power & Energy Systems*, vol. 129, p. 106799, 2021.

- [41] Q. Ling and A. Ribeiro, “Decentralized linearized alternating direction method of multipliers,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5447–5451.
- [42] M. Hong and Z.-Q. Luo, “On the linear convergence of the alternating direction method of multipliers,” *Mathematical Programming*, vol. 162, no. 1-2, pp. 165–199, 2017.
- [43] Y. Yang, X. Guan, Q.-S. Jia, L. Yu, B. Xu, and C. J. Spanos, “A survey of admm variants for distributed optimization: Problems, algorithms and features,” *arXiv preprint arXiv:2208.03700*, 2022.
- [44] A. Aydinoglu and M. Posa, “Real-time multi-contact model predictive control via admm,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3414–3421.
- [45] A. Aydinoglu, A. Wei, and M. Posa, “Consensus complementarity control for multi-contact mpc,” *arXiv preprint arXiv:2304.11259*, 2023.
- [46] Y. Shirai, X. Lin, A. Schperberg, Y. Tanaka, H. Kato, V. Vichathorn, and D. Hong, “Simultaneous contact-rich grasping and locomotion via distributed optimization enabling free-climbing for multi-limbed robots,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 563–13 570.
- [47] O. Shorinwa and M. Schwager, “Distributed contact-implicit trajectory optimization for collaborative manipulation,” in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2021, pp. 56–65.
- [48] L. Wijayarathne, Z. Zhou, Y. Zhao, and F. L. Hammond, “Real-time deformable-contact-aware model predictive control for force-modulated manipulation,” *IEEE Transactions on Robotics*, 2023.
- [49] F. Marić, L. Petrović, M. Guberina, J. Kelly, and I. Petrović, “A Riemannian metric for geometry-aware singularity avoidance by articulated robots,” *Robotics and Autonomous Systems*, vol. 145, p. 103865, 2021.
- [50] N. Jaquier, L. Rozo, D. G. Caldwell, and S. Calinon, “Geometry-aware manipulability learning, tracking, and transfer,” *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 624–650, 2021.
- [51] L. Rozo, N. Jaquier, S. Calinon, and D. G. Caldwell, “Learning manipulability ellipsoids for task compatibility in robot manipulation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3183–3189.
- [52] H. Tugal, K. Cetin, Y. Petillot, M. Dunnigan, and M. S. Erden, “Manipulation at optimum locations for maximum force transmission with mobile robots under environmental disturbances,” *Autonomous Robots*, vol. 46, no. 6, pp. 769–782, 2022.
- [53] J. H. Kim, Y. J. Shin, and I. G. Jang, “Evaluating the maximum directional kinematic capability of a redundant manipulator based on allowable velocity and force,” *IEEE Access*, vol. 9, pp. 88 085–88 097, 2021.
- [54] W. J. Marais, S. B. Williams, and O. Pizarro, “Anisotropic disturbance rejection for kinematically redundant systems with applications on an uvms,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7017–7024, 2021.
- [55] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of physics*. John Wiley & Sons, 2013.
- [56] T. S. Lembono and S. Calinon, “Probabilistic iterative lqr for short time horizon mpc,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 579–585.
- [57] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.



Boyang Ti received the B.S. degree in mechanical engineering from the Dalian University of Technology (DUT), Dalian, China, in 2017. From 2021 to 2022, he was an intern at the Robot Learning & Interaction group of Idiap Research Institute. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology (HIT). His research interests include human-robot collaboration, learning from demonstration, robotic skill learning, and optimal control. Website: <https://tflqw.github.io>



Yongsheng Gao received the B.Sc., M.Sc., and Ph.D. degrees from the State Key Laboratory of Robotics Institute, Harbin Institute of Technology (HIT), Harbin, China, in 1994, 2001, and 2007, respectively, where he is currently an Associate Professor. His research interests include pathological tremor suppress, tele-operation robot, and biomedical signal processing.



Jie Zhao received the B.S. and Ph.D. degrees in mechatronics engineering from the Harbin Institute of Technology (HIT), Harbin, China, in 1990 and 1996, respectively. He is currently a Professor with the School of Mechatronics Engineering, HIT, where he is also the Director of the State Key Laboratory of Robotics and Systems. He is the Leader of the Subject Matter Expert Group of Intelligent Robots in the National 863 Program supervised by the Ministry of Science and Technology of China. His research interests include industrial robots and bionic robots.



Sylvain Calinon received the BSc, MSc, and PhD degrees from Ecole Polytechnique Fédérale de Lausanne (EPFL) in 2001, 2003 and 2007, respectively. He is currently a Senior Research Scientist at the Idiap Research Institute and a Lecturer at EPFL. From 2009 to 2014, he was a Team Leader at the Italian Institute of Technology. From 2007 to 2009, he was a Postdoc at EPFL. His research interests cover robot learning, human-robot collaboration, optimal control and model-based optimization. Website: <https://calinon.ch>