



Manipulation planning under changing external forces

Lipeng Chen¹ · Luis F. C. Figueiredo¹ · Mehmet R. Dogar¹

Received: 2 October 2019 / Accepted: 22 June 2020 / Published online: 9 July 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

This paper presents a planner that enables robots to manipulate objects under changing external forces. Particularly, we focus on the scenario where a human applies a sequence of forceful operations, e.g. cutting and drilling, on an object that is held by a robot. The planner produces an efficient manipulation plan by choosing stable grasps on the object, by intelligently deciding when the robot should change its grasp on the object as the external forces change, and by choosing subsequent grasps such that they minimize the number of regrasps required in the long-term. Furthermore, as it switches from one grasp to the other, the planner solves the bimanual regrasping in the air by using an alternating sequence of bimanual and unimanual grasps. We also present a conic formulation to address force uncertainties inherent in human-applied external forces, using which the planner can robustly assess the stability of a grasp configuration without sacrificing planning efficiency. We provide a planner implementation on a dual-arm robot and present a variety of simulated and real human-robot experiments to show the performance of our planner.

Keywords Manipulation planning · Forceful human-robot collaboration · Task-oriented grasping

1 Introduction

Most manipulation planning focuses on dealing with geometric constraints. In this work, we are interested in the scenario where a robot manipulates an object not only under geometric constraints, but also under the application of changing external forces. Take the cutting task in Fig. 1, where a human is cutting a circular piece off from a rectangular board with the assistance of a robot system (Fig. 1a). Before the task,

human indicates the operation type (cutting) and the desired cutting pattern (a circle) using a graphical interface (Fig. 1a-Left). During the cutting task, the human applies external forces on the board which change position, direction, and even magnitude along the circular path. To assist the human to perform the task, the robot changes its grasp on the object multiple times (Fig. 1a–e) to position the object at expected pose(s) and keep it stable against the changing cutting forces. In this paper, we propose a planner that enables the robot to manipulate objects under changing external forces like this.

This kind of human-robot interaction can be very useful in manufacturing applications, where human workers need to apply a sequence of forceful operations like polishing, cutting and welding on workpieces, or in carpentry where sequential forceful operations like drilling and inserting are widely observed. To achieve this level of interaction, our planner needs to solve three key problems:

First, our planner produces efficient manipulation plans by minimizing the number of times the robot needs to change its grasp on the object, namely *regrasp*. For example in Fig. 1, the robot uses three different grasp configurations to keep the object stable and accordingly changes its grippers' positions on the object only two times (counting each gripper separately) during the whole task. This is also a capability demonstrated by humans in sequential manipulation tasks:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie SkłodowskaCurie Grants Agreement No. 746143 and 795714, and from the UK Engineering and Physical Sciences Research Council under Grant EP/P019560/1.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10514-020-09930-z>) contains supplementary material, which is available to authorized users.

✉ Lipeng Chen
scle@leeds.ac.uk

Luis F. C. Figueiredo
l.figueiredo@leeds.ac.uk

Mehmet R. Dogar
m.r.dogar@leeds.ac.uk

¹ School of Computing, University of Leeds, Leeds, UK

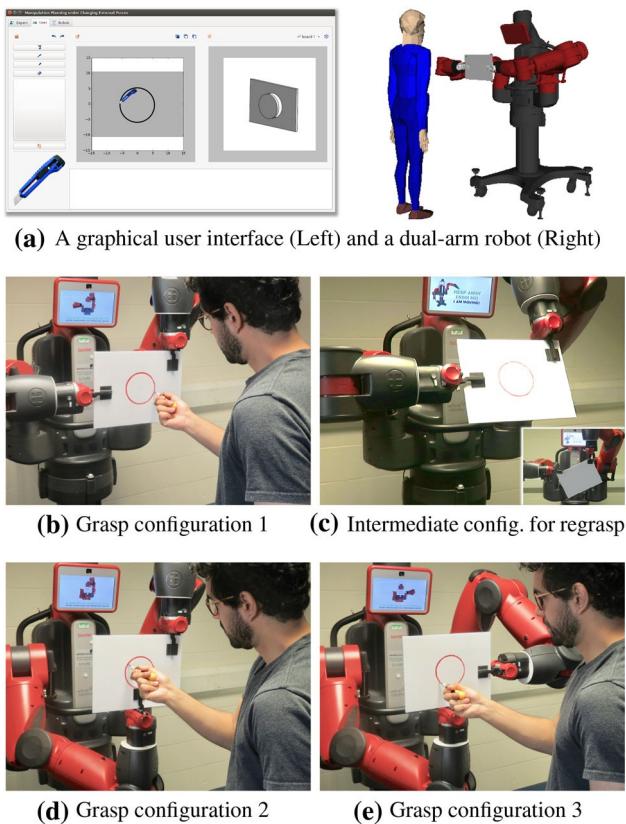


Fig. 1 The human is cutting a circular piece off from a board with the assistance of a robot system

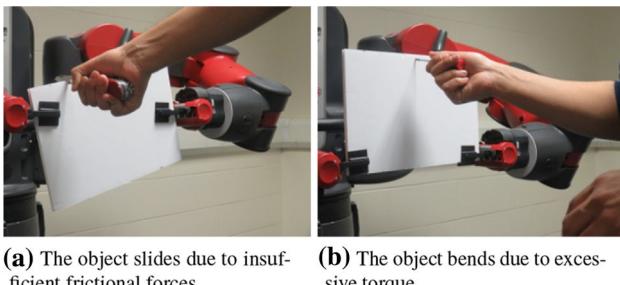


Fig. 2 Task failures during cutting (a) and drilling (b)

we regrasp when we need to, but we are also able to choose grasps which are useful for long durations during a task.

This capability poses two closely related challenges to the planner: *grasp planning* and *regrasp minimization*. Specifically, the planner needs to decide not only **how** to grasp the object, but also **when** to regrasp the object during the course of interaction. A *good* choice of robot grasp on the object may enable the robot to stabilize the object against multiple sequential external forces, and thus reduce the need of regrasping throughout the interaction, while a *bad* grasp, however, would lead to frequent regrasps and therefore task interruptions. Even worse, an inappropriate grasp may not be able to stabilize the object against some external forces, thus

bringing about task failures and risks during execution. For example, the object may slip through the gripper fingers during a cutting operation (Fig. 2a) due to insufficient frictional forces between gripper fingers and object surface. Similarly, a drilling operation may exert excessive torque around the grippers due to a bad choice of grasp (Fig. 2b).

Second, our planner plans each regrasp. A regrasp requires the robot to release its grippers off the object and then to grasp the object at different positions. However, when the robot releases a gripper, the object may become unstable under external forces, e.g. gravity. Even if we assume the human in Fig. 1b stops applying cutting forces during regrasps, the object may still become unstable due to gravity. For example, to regrasp the object from the configuration in Fig. 1b to the one in Fig. 1d, if the robot directly releases its right gripper from the object as shown in the small figure at the right bottom of Fig. 1c, a heavy object may slip within the remaining gripper. Alternatively, the robot can first move the object to an intermediate pose before releasing one gripper, so that the remaining one can still hold the object stable until the robot completes the regrasp. Figure 1c shows such an intermediate pose, at which the object is stable even when the right robot gripper releases from it.

Third, our planner takes a robust approach to efficiently assess the stability of a grasp configuration with the presence of force uncertainties. The primary step towards manipulation planning under changing external forces is to model the external forces. A forceful operation, such as cutting a board, ideally, exerts a determinate external force on a target object. However, in practical applications such a human-applied forceful operation would inevitably deviate from its expected direction, which brings about force uncertainties and thus challenges in finding appropriate robot grasps to keep the target object stable. In this sense, to guarantee effective and robust planning, our planner chooses grasps which can keep the object stable not only under the expected operation force, but also under all possible deviated operation forces. To achieve this, our planner requires a model of the forces to be applied as input in advance.

This work is a significantly extended and improved version of our previous work on manipulation planning under changing external forces (Chen et al. 2018b). Briefly, we build our planner on the following key contributions:

- A graph-based formulation of manipulation planning under changing external forces, which is referred to as the *operation graph* hereafter, and a planning approach which can simultaneously (i) produce a sequence of grasp configurations to keep the object stable, and (ii) minimize the need of regrasping during manipulation (Sect. 5.1).
- An algorithm to plan stable regrasps in the air by using multiple cooperative manipulators. Different from most existing work in regrasp planning (reviewed in Sect. 2),

we focus on regrasping without an extra support structure. This is achieved by reasoning about the object's stability under gravity while moving the object to go through an alternating sequence of intermediate unimanual and bimanual grasps (Sects. 5.2–5.4).

New contributions in this version includes:

- A conic model for external forces with the presence of force deviations and a new theorem with detailed proof, which formulates and significantly simplifies the stability check using the conic model (Sect. 4).
- A graphical user interface which ties in human task specification, on-demand manipulation planning and robot-assisted fabrication together (Sect. 6).
- A new set of simulated and real-robot experiments with an increased number and variety of forceful tasks to verify the performance of our planning framework (Sect. 7).

2 Related work

This work is mainly related to three areas which have been well studied in the literature: *grasp analysis*, *multi-step manipulation planning and regrasping* and *forceful human-robot collaboration*.

2.1 Grasp analysis

The literature of *grasp analysis* investigates the question of how stable a grasp is. General methods using the concept of *force-closure/form-closure* answer whether a grasp would be able to resist external wrenches acting along arbitrary directions. The *grasp wrench space* (Mishra et al. 1987; Borst et al. 2004), for example the volume of its largest inscribed sphere (Ferrari and Canny 1992), can be used as a metric to measure the general quality of a grasp configuration.

The *task-oriented grasping* literature (Dang and Allen 2012; El-Khoury et al. 2015; Nikandrova and Kyrki 2015) studies the problem of grasping an object for a particular task, an important part of which is modelling the particular external wrench expected on the target object during the task. For example, Li and Sastry (1988) presents the *task wrench space* as a metric to measure how good a grasp is under task-relevant external wrenches. Other work in this area proposes faster and more robust ways to compute task-based metrics (Borst et al. 2004; Haschke et al. 2005; Lin and Sun 2015). In general, given an external wrench, a set of contact points on the object, and *contact-models* (Salisbury and Roth 1983) (which provide constraints on the directions and magnitudes of the wrenches that can be applied at the contacts), the question of whether the set of contacts would be able to resist the external wrench can be formulated as a linear matrix inequality

problem (Han et al. 2000). Grasp analysis in the case of compliant contact has also been investigated through modelling the contact between a finger and the object as a spring (Cutkosky and Kao 1989).

While the grasp analysis literature focuses on the stability of a grasp on a target object, our work is also related to *cooperative manipulation*, which focuses on the problem of multiple manipulators cooperatively manipulating a common object (Takase 1974; Zheng and Luh 1989). To exert a resultant wrench on the object, one can solve a set of linear equations to find the force/torque efforts required at the manipulator joints (Uchiyama and Dauchez 1988, 1992).

We build on the formulations of grasp stability and cooperative manipulation to propose our grasp stability check (Sect. 4.1), which involves checking the force/torque limits at both the grip points and the manipulator joints.

2.2 Multi-step manipulation planning and regrasping

In a typical problem of *multi-step manipulation planning*, a robot manipulates an object through geometric obstacles where the robot needs to ungrasp and regrasp the object multiple times. The need to regrasp objects was recognized even in the earliest manipulation systems (Lozano-Pérez et al. 1987; Tournassoud et al. 1987). Later, Siméon et al. (2004) presented a planner via random sampling that solves the problem using an alternating sequence of *transfer* and *transit* actions. More recently, planners have been proposed to solve the planning problem in the case of multiple manipulators for assembly-like tasks (Lertkultanon and Pham 2018; Wan and Harada 2016; Dogar et al. 2019).

Most existing work on manipulation planning focuses on dealing with geometric constraints, generating collision-free robot motions to manipulate target objects. Our planner goes beyond geometric constraints, taking into account the force feature, which can be required in a large variety of sequential manipulation tasks. In our task, for example, the robot is required not only to move a target object to desired goal position(s) under geometric constraints, but also to keep the object stable under changing external forces.

Our work is also related to regrasp planning, especially the case of dual-arm or multi-arm regrasping. Regrasp planning involves finding a connected path over a sequence of sub-manifolds in the composite configuration space. Roughly, the basic flow of regrasp planning follows the pattern of first building a *manipulation graph* (Alami et al. 1990) and then searching the graph for regrasp sequences. Early studies (Rohrdanz and Wahl 1997; Stoeter et al. 1999) employed *grasp-placement* tables to generate a sequence of motions for regrasping. More recent works propose some other graph-based representations, such as the *regrasp graph* (Wan and Harada 2016, 2017). Most existing studies on regrasp plan-

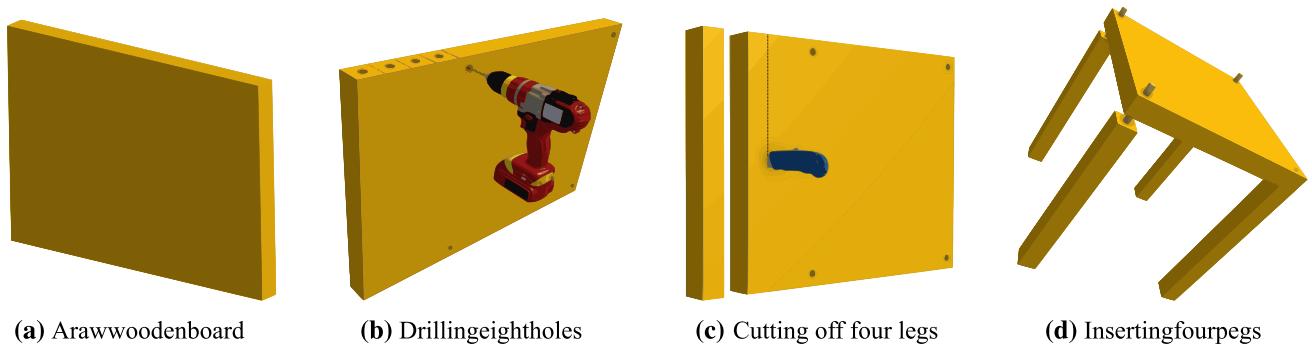


Fig. 3 The table assembly task consists of a continuous (e.g. cutting) or discrete (e.g. drilling) sequence of forceful operations on a target object

ning use object placements on extra supports for regrasping, such as support surfaces (Wan and Harada 2016, 2017; Chavan-Dafie and Rodriguez 2018) and other complex structures (Cao et al. 2016; Ma et al. 2018).

Different from the regrasping work mentioned above, our work specifically focuses on the planning problem where the robot can not place the object down on an extra support surface, but only use its manipulators to cooperatively regrasp the object under external forces, e.g. gravity.

Our problem can also be interpreted as an instance of *multi-modal manipulation planning* (Bretl 2006; Hauser and Latombe 2010; Lee et al. 2015), where each modality corresponds to a bimanual or unimanual grasp. In developing a planner, we follow a similar strategy of first identifying intersections among these different grasp modalities/manifolds, and then planning motions to connect them. Our problem can also be interpreted as a constrained *set-cover* problem (Slavík 1996; Feige 1998), where the planner needs to find a minimal sequence of grasp configurations to keep the target object stable under changing external forces in order. To address the sequential nature inherent in the force sequence, we formulate a weighted directed graph to search for the optimal grasp sequence efficiently in Sect. 5.

2.3 Forceful human-robot collaboration

We are also interested in addressing multi-step manipulation planning in a *human-robot collaboration* setting. Existing work in forceful human-robot collaboration mostly focuses on the control problem (Kosuge and Kazamura 1997; Rozo et al. 2016; Abi-Farraj et al. 2017), solving for necessary stiffness of manipulator joints as an external force is applied, and assumes the object to be already stably grasped by the robot. We approach the problem from the manipulation planning point of view and address the decision of what grasps to use and when/how to switch between these grasps.

Other work in planning for human-robot collaboration exists which mostly focuses on object transportation (Rozo et al. 2016), handover (Sisbot and Alami 2012; Strabala

et al. 2013; Maeda et al. 2017), or other applications where robots attempt to avoid colliding with humans in the shared workspace (Luo et al. 2018). To the best of our knowledge, our work is the first one to take a planning approach to solve the collaboration problem where the human applies sequential changing forces on an object.

2.4 Other work on robotic assembly

There is also recent work focusing on assembly planning. Lipton et al. (2017, 2018) present a system for robot-assisted carpentry. The system uses a team of mobile robots to fabricate human-customised parts with standard carpentry tools and assumes two specialized stands to stabilize lumbars against cutting forces. In another recent work by Moriyama et al. (2019), a sampling-based assembly planner was proposed to generate stable assembly pose under gravitational constraint. The main difference in our work is that we consider changing external forces applied on an object manipulated by a multi-arm robot.

3 Problem formulation

This section presents the definitions and fundamentals of the planning problem discussed in this work.

3.1 Problem background

In this work, we refer to a complete forceful interaction as a *forceful task*, which consists of a continuous or discrete sequence of *forceful operations*. An example is the circular cutting task shown in Fig. 1, which we discretize into a sequence of cutting operations tangential to the circle path. Another example is the table assembly task illustrated in Fig. 3, which requires the human to apply eight drilling operations on a wooden board to create holes (Fig. 3b), a continuous sequence of cutting operations to get chair legs (Fig. 3c), and four inserting operations (Fig. 3d) to assemble

the legs. In Sect. 6, we present a graphical user interface, using which human users can easily specify such forceful tasks, i.e. sequences of forceful operations, in an interactive manner.

We define a *forceful operation* \mathbf{F} as a generalized force (force/torque)¹ f w.r.t. a tool frame, applied at a pose \mathbf{p} on a target object w.r.t. an object frame, which is at a desired pose $\mathbf{T} \in \mathbb{SE}(3)$ w.r.t. a robot frame during the course of the operation. That is, a forceful operation can be specified as $\mathbf{F} = (f, \mathbf{p}, \mathbf{T})$. Accordingly, a *forceful task*, comprising a sequence of forceful operations, can be represented as

$$\{\mathbf{F}_i\}_{i=1}^m = \{(f_i, \mathbf{p}_i, \mathbf{T}_i)\}_{i=1}^m \quad (1)$$

where m indicates the number of involved operations. For example, as illustrated in Fig. 4, we treat the circular cutting task as a sequence of 20 cuttings via discretization.

We assume the robot has two manipulators, and each manipulator is equipped with a parallel gripper.² Let $\mathcal{C}_l, \mathcal{C}_r$ be the configuration space of the left and right arm respectively, and $\mathcal{C}_o \subseteq \mathbb{SE}(3)$ be the object's configuration space. The system's composite configuration space \mathcal{C} can be then defined as their Cartesian product $\mathcal{C} = \mathcal{C}_l \times \mathcal{C}_r \times \mathcal{C}_o$, while each composite configuration $\mathbf{q} \in \mathcal{C}$ can be denoted as $\mathbf{q} = (\mathbf{q}_l, \mathbf{q}_r, \mathbf{T})$, where $\mathbf{q}_l \in \mathcal{C}_l$, $\mathbf{q}_r \in \mathcal{C}_r$, and $\mathbf{T} \in \mathcal{C}_o$.

We define a robot *grasp* \mathbf{g} , using the relative pose(s) of gripper(s) on the target object. Specifically, a bimanual grasp $(\mathbf{g}_l, \mathbf{g}_r)$ specifies poses of both left and right grippers, while the unimanual grasps $(\mathbf{g}_l), (\mathbf{g}_r)$ specify pose of only left and right gripper respectively.

Note that there is redundancy in this definition. Specifically, a system configuration $\mathbf{q} = (\mathbf{q}_l, \mathbf{q}_r, \mathbf{T})$ can be mapped to its corresponding grasp configuration \mathbf{g} via forward kinematics. In this sense, the composite configuration space \mathcal{C} can be regarded as a collection of lower-dimensional *grasp manifolds*, in which each manifold $\mathcal{M}(\mathbf{g})$ corresponds to a particular robot grasp \mathbf{g} on the object.

3.2 Overview of problem

Figure 4 illustrates our planning problem in detail using the circular cutting task (Fig. 1).

The robot is supposed to position and stabilize a target object under the application of a sequence of forceful operations $\{\mathbf{F}_i\}_{i=1}^m$. Given a single forceful operation \mathbf{F} , the planner can find a *feasible* configuration \mathbf{q} by first searching for a

kinematically valid configuration \mathbf{q} and then checking the force stability of the system, i.e. whether the robot and object are stable under the operation force f at the configuration \mathbf{q} . This problem has been widely discussed in the literature on grasp stability (Mishra et al. 1987; Borst et al. 2004; Ferrari and Canny 1992) and cooperative manipulation (Uchiyama and Dauchez 1988, 1992). We explain in detail how we perform the stability check in Sect. 4.

Then, given a forceful task consisting of a sequence of forceful operations $\{\mathbf{F}_i\}_{i=1}^m$, the planner can simply find one feasible grasp configuration \mathbf{q}_i for each operation $\mathbf{F}_i \in \{\mathbf{F}_i\}_{i=1}^m$ and accordingly, impose one configuration switch, or broadly, a *regrasp*, between every two sequential operations. In this case, the robot would need to perform two regrasps (one regrasp for each gripper for a dual-arm robot) for each operation \mathbf{F}_i and thus at least $2m$ regrasps in total for whole task.

Alternatively, the robot can make the utmost of one configuration \mathbf{q} against multiple forceful operations in a row, which, as a result, would reduce the need of regrasping. This, **regrasp minimization**, imposes an additional but practically necessary requirement for efficient and smooth manipulation. In this work, we explicitly address this as a main objective, building planners to find stable manipulation plans with a minimal number of regrasps.

We say a system configuration \mathbf{q} is *stable against* a sequence of k forceful operations $\{\mathbf{F}_i\}_{i=1}^k$ if, at \mathbf{q} , the robot and object are stable under any operation in $\{\mathbf{F}_i\}_{i=1}^k$. Further, we say a sequence of configurations $\{\mathbf{q}_j\}_{j=1}^n$ is stable against a sequence of forceful operations $\{\mathbf{F}_i\}_{i=1}^m$, if the configurations in $\{\mathbf{q}_j\}_{j=1}^n$ cover all forceful operations in $\{\mathbf{F}_i\}_{i=1}^m$ in order, i.e. if \mathbf{q}_1 is stable against $\{\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_k\}$, \mathbf{q}_2 is stable against $\{\mathbf{F}_{k+1}, \mathbf{F}_{k+2}, \dots, \mathbf{F}_l\}$, and so on, until \mathbf{q}_n is stable against $\{\mathbf{F}_{s+1}, \mathbf{F}_{s+2}, \dots, \mathbf{F}_m\}$, where $1 \leq k < l < \dots < s < m$.

For example, the three configurations $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}$ shown in Fig. 4 are stable against the 20 circular cutting operations (\mathbf{q}_1 is stable against \mathbf{F}_1 to \mathbf{F}_8 , \mathbf{q}_2 is stable against \mathbf{F}_9 to \mathbf{F}_{12} , \mathbf{q}_3 is stable against \mathbf{F}_{13} to \mathbf{F}_{20}). Note that different configurations correspond to different grasps on the object. In this sense, regrasp minimization can be achieved by finding a minimal sequence of configurations (i.e. a minimal n) $\{\mathbf{q}_j\}_{j=1}^n$, stable against the operations $\{\mathbf{F}_i\}_{i=1}^m$.

In addition, the robot needs to move the object to go through the planned configurations in $\{\mathbf{q}_j\}_{j=1}^n$ successively, using *collision-free* and *stable* trajectories $\{\mathbf{t}_j\}_{j=1}^n$. Specifically, each trajectory \mathbf{t}_j moves the system from \mathbf{q}_{j-1} to \mathbf{q}_j (\mathbf{q}_0 is the initial system configuration), which corresponds to a constrained regrasping task.

In this context, we define a *manipulation query* as a forceful task consisting of a sequence of forceful operations $\{\mathbf{F}_i\}_{i=1}^m$ to be applied on the object, together with a starting

¹ Later in Sect. 4.1, we present a more realistic model where f is a distribution of a set of possible generalized forces that can be applied during a forceful operation, instead of a single idealized force.

² This is for clarity of explanation and because the robot we use in our experiments has two arms. However, our formulation is general and can be easily extended to systems with more manipulators.

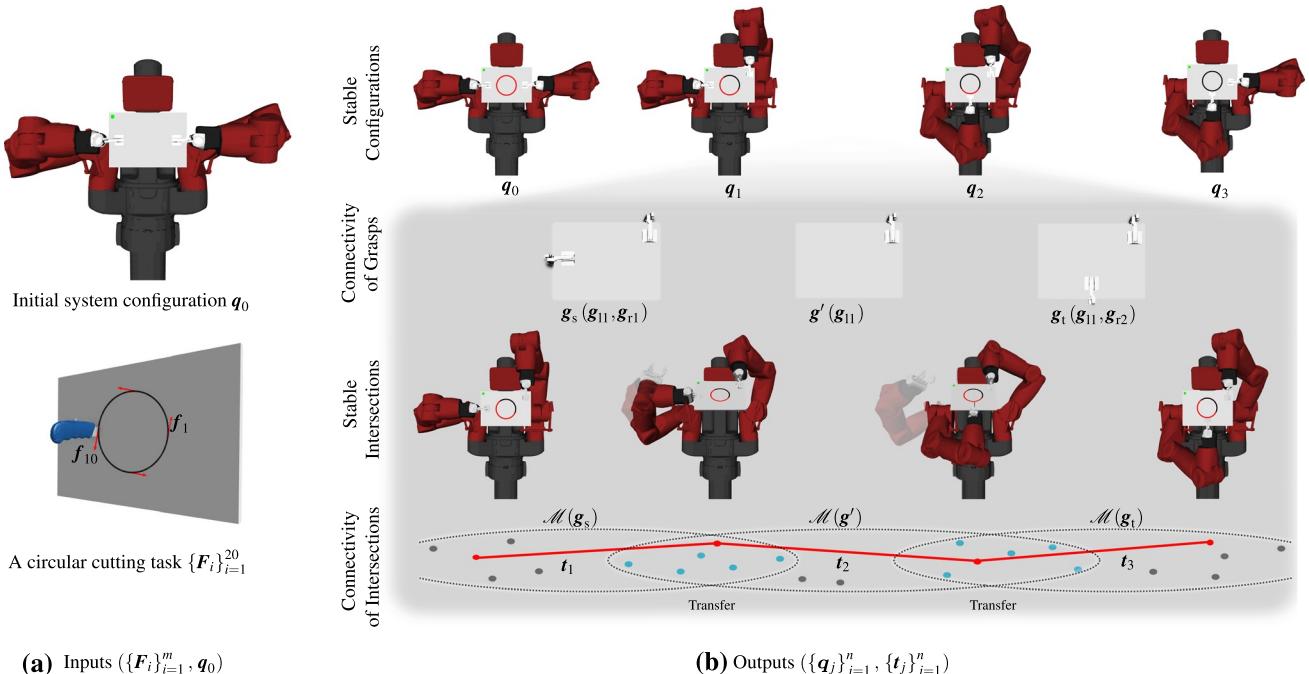


Fig. 4 Overview of the approach. **a** A circular cutting task is represented as a sequence of 20 cutting operations $\{F_i\}_{i=1}^{20}$ tangential to the desired circle. **b** Layers of our planning approach

system configuration q_0 . Then, the *manipulation planning problem under changing external forces* can be stated as:

Given the description of manipulation system and a manipulation query $(\{F_i\}_{i=1}^m, q_0)$, find a minimal sequence of grasp configurations $\{q_j\}_{j=1}^n$ and stable connecting trajectories $\{t_j\}_{j=1}^n$ to position and stabilize the object under forceful operations in $\{F_i\}_{i=1}^m$ in order.

3.3 Overview of approach

The primary step towards efficient object manipulation under sequential forceful operations is modelling and checking the stability of forceful operations. We present how we model and perform the stability check of forceful operations with and without the presence of force deviations in Sect. 4.

- *Idealized forceful operations and stability check*: We formulate an idealized operation model for forceful operations that can be applied exactly as expected. Using the idealized operation model, we formulate the stability check as a linear programming problem in Sect. 4.1.
- *Deviated forceful operations and stability check*: We formulate a spherical cone model to address forceful operations with the presence of force deviations. Further, we propose a polyhedral cone approximation and prove a theorem simplifying the stability check using the spherical cone model in Sect. 4.2.

Having a model of forceful operations as planning input, we illustrate how our planner solves a manipulation query in Fig. 4 with four layers and present details of each layer in Sect. 5. Here we present a brief overview and explain how these layers fit together:

– *Stable configurations*: Given an input manipulation query $(\{F_i\}_{i=1}^m, q_0)$, the planner first identifies a sequence of configurations $\{q_j\}_{j=1}^n$ which are stable against $\{F_i\}_{i=1}^m$, and minimize the number of regrasps during manipulation. In Fig. 4, the three configurations $\{q_1, q_2, q_3\}$ shown in the top layer is such an example sequence.

The configurations generated by this layer are discrete over the configuration space. The lower layers of the planner try to generate a sequence of stable motion trajectories $\{t_j\}_{j=1}^n$ to connect every two subsequent configurations in $\{q_j\}_{j=1}^n$ starting from q_0 , which corresponds to a sequence of constrained regrasping tasks. This layer is explained in detail in Sect. 5.1.

– *Connectivity of grasps*: Given any two subsequent configurations $q_s, q_t \in \{q_j\}_{j=1}^n$ produced by the previous layer (e.g. q_1 and q_2 in Fig. 4), the planner identifies a sequence of intermediate grasps $\{g_i\}_{i=1}^{n_g}$, which moves the robot grippers from the grasp g_s in q_s to the grasp g_t in q_t (denoted as g_1 and g_{n_g} respectively in $\{g_i\}_{i=1}^{n_g}$).

The grasp sequence acts as an abstract plan to guide the

subsequent search. The second layer in Fig. 4 shows such an example grasp sequence $\{g_s, g', g_t\}$. It connects the grasps in configurations q_1 and q_2 of the previous layer. Note that there might be other feasible grasp sequences, which go through different intermediate gripper contacts as shown in Fig. 10. This layer is explained in detail in Sect. 5.2.

- *Sampling stable intersections of grasp manifolds:* Given any two neighbouring grasps $g_i, g_{i+1} \in \{g_i\}_{i=1}^{n_g}$, the planner identifies a set of candidate stable regrasping configurations by sampling within the intersection of their grasp manifolds $\mathcal{M}(g_i) \cap \mathcal{M}(g_{i+1})$ (illustrated as the blue points in Fig. 4). These configurations are checked for stability against object gravity such that at each configuration in the set, the *transition* from g_i to g_{i+1} can be performed stably. The second configuration in the third layer of Fig. 4 is such an example. Note that the object is deliberately tilted from its initial pose, such that at the configuration both unimanual and bimanual grasps can hold the object stable under object gravity. That is, the configuration is a stable transition/regrasping configuration from grasp g_i to grasp g_{i+1} . This layer is explained in detail in Sect. 5.3.
- *Connectivity of manifold intersections:* After obtaining a sequence of stable regrasping configurations in the intersections of the sequence of grasp manifolds, the fourth layer performs collision-free and stability-constrained motion planning within these manifolds, namely generating a sequence of stable and collision-free trajectories $\{t_j\}_{j=1}^n$ (illustrated as the red solid lines in Fig. 4). This layer is explained in detail in Sect. 5.4.

Overall, the layered structure enables the planner to minimize the number of regrasps at the top layer. The planner takes some form of lazy planning (Bohlin and Kavraki 2000; Sánchez and Latombe 2003; Hauser 2015): It generates high-level plans in upper layers to provide significant search guidance to lower layers, leaving the time-consuming motion planning to the final layer. If lower layers fail to find a plan, the planner goes back to higher layers to generate new and different high-level plans.

4 Force modelling and stability

This section presents our mathematical formulations of forceful operations, and explains in detail how the planner efficiently checks the force stability of a candidate configuration q under a certain forceful operation F , while with the presence of force uncertainties. We refer to this process as *stability check* herein.

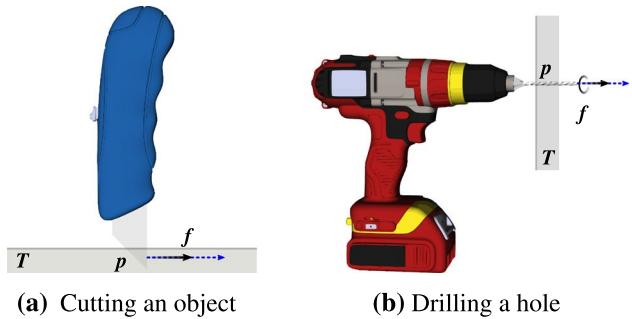


Fig. 5 Ideally, a forceful operation F would generate a determinate force f onto a target object along/about an expected operation axis

4.1 Idealized operations modelling and stability check

In this section, we present an idealized operation model for forceful operations that can be applied exactly as expected. Later in the following section, we introduce a conic operation model to address forceful operations with the presence of force deviations.

Idealized Operation Model: Ideally, a forceful operation F , e.g. cutting and drilling as illustrated in Fig. 5, qualitatively involves moving a certain tool(object) along/about an expected operation axis (depicted as blue axes in Fig. 5) to interact with a target object. Accordingly, if applied as expected, the operation F would produce a determinate operation force f onto the target object along/about the expected operation axis.

In this sense, the forceful operation F can be simply modelled as a single generalized force f applied on the target object. For example, ideally, a cutting operation (Fig. 5a) would generate a force along a cutting axis along the cutting direction. Similarly, a drilling operation (Fig. 5b) would generate a drilling force together with a rotational torque along/about an axis perpendicular to the object surface.

Herein, we refer to this formulation $F : (f, p, T)$ as the *idealized* operation model. It assumes the operation F to be exactly applied along/about an expected operation axis, at a pose p on the target object, ideally modelling F as a single generalized point in the wrench space.

Stability Check with Idealized Operation Model: *Stability check* refers to checking if a grasp configuration q (along with its corresponding grasp g) is stable against a certain forceful operation F . Specifically, we are interested in checking whether:

- The robot manipulators are able to provide sufficient stiffness to keep the robot and target object stable against F . This requires the planner to check whether the required torques τ at manipulator joints exceed the torque limits.

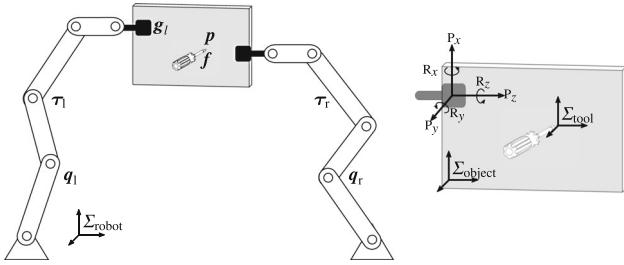


Fig. 6 Left: The planner checks if a candidate grasp configuration \mathbf{q} is able to provide a solution of torques $\boldsymbol{\tau}$ at manipulator joints and wrenches \mathbf{f}_g at grip points to keep the target object stable against a forceful operation \mathbf{F} . Right: We approximate the grasp wrench space of a grasp \mathbf{g} with an axis-aligned box in the 6D wrench space

- The grippers are able to provide sufficient wrenches \mathbf{f}_g at grip points to stabilize the target object in hand. This requires the generalized external force (force/torque) applied by \mathbf{F} onto the object is inside the grasp wrench space (Mishra et al. 1987), namely the set of all external wrenches \mathbf{g} can resist.

Consider a generalized force $\mathbf{f}_{g,i}$ acting at the gripper of the i -th manipulator, the required torques $\boldsymbol{\tau}_i$ at manipulator joints can be derived by $\boldsymbol{\tau}_i = \mathbf{J}_i^T \mathbf{f}_{g,i}$, where \mathbf{J}_i is the Jacobian matrix of i -th manipulator at a configuration \mathbf{q} .

Furthermore, the symmetric formulation by Uchiyama and Dauchez (1988, 1992), generalized the above model to multiple manipulators cooperatively holding a common object, describing the kinematic and static relationships between an external force \mathbf{f} applied at the object and its counterparts acting at manipulator joints.

The symmetric formulation, however, leaves the forces \mathbf{f}_g at grip points unconstrained. For the case of parallel plate grippers we use in this work, as illustrated in Fig. 6-right, we approximate the grasp wrench space of a gripper-object system with a 6D axis-aligned box in the wrench space. Specifically, we take the maximum forces/torques along/about the three principal axes ($X-Y-Z$) that are resistible at grip point as its limits $\mathbf{f}_{g,i}^{\max}, \mathbf{f}_{g,i}^{\min}$, where $\mathbf{f}_{g,i}^{\max} = [\mathbf{P}_x^+, \mathbf{P}_y^+, \mathbf{P}_z^+, \mathbf{R}_x^+, \mathbf{R}_y^+, \mathbf{R}_z^+]^T$ and $\mathbf{f}_{g,i}^{\min} = [\mathbf{P}_x^-, \mathbf{P}_y^-, \mathbf{P}_z^-, \mathbf{R}_x^-, \mathbf{R}_y^-, \mathbf{R}_z^-]^T$ are the vectors of estimated upper and lower limits at the i -th grip point. $\mathbf{P}_{x/y/z}^{+/-}$ and $\mathbf{R}_{x/y/z}^{+/-}$ are the force and torque limits respectively.

Imposing the additional constraints onto above formulations, we model the stability check as finding a distribution of $\boldsymbol{\tau}$ and \mathbf{f}_g that satisfies:

$$\begin{cases} \mathbf{J}^T \mathbf{f}_g = \boldsymbol{\tau} \\ \mathbf{W} \mathbf{f}_g = -\mathbf{R}(\mathbf{p}) \mathbf{f} \end{cases} \quad (2a)$$

$$\boldsymbol{\tau}^{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}^{\max}, \mathbf{f}_g^{\min} \leq \mathbf{f}_g \leq \mathbf{f}_g^{\max} \quad (2b)$$

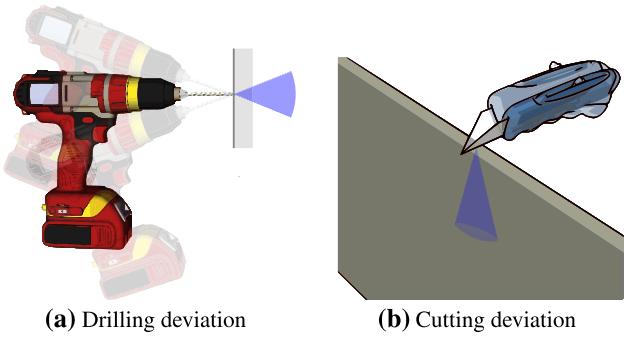


Fig. 7 The operation force \mathbf{f} may deviate from the expected operation axis by a certain angle

where

- $\mathbf{J} = \text{diag}(\mathbf{J}_1, \dots, \mathbf{J}_n)$ is the composite Jacobian matrix at configuration \mathbf{q} ;
- $\mathbf{f}_g = [\mathbf{f}_{g,1}^T, \mathbf{f}_{g,2}^T, \dots, \mathbf{f}_{g,n}^T]^T$ and $\mathbf{f}_{g,i}^T$ is the generalized force acting at i -th gripper;
- $\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T, \boldsymbol{\tau}_2^T, \dots, \boldsymbol{\tau}_n^T]^T$ and $\boldsymbol{\tau}_i$ is the joint torque distribution of i -th manipulator;
- $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_n]$ (usually termed as the grasp matrix (Mishra et al. 1987; Borst et al. 2004; Ferrari and Canny 1992)) is a $(6 \times 6n)$ matrix mapping forces at robot grippers to a resultant wrench onto the object w.r.t. the robot frame;
- $\boldsymbol{\tau}^{\max/\min}$ are the upper and lower joint torque limits respectively;
- $\mathbf{f}_g^{\max/\min}$ are the estimates of upper and lower wrench limits at grip points respectively (i.e. our estimate of grasp wrench space).
- $\mathbf{R}(\mathbf{p})$ transforms the external force \mathbf{f} from the tool frame to the common robot frame.

Equation 2 models stability check of an idealized forceful operation as a *linear programming*, and thus can be solved, e.g. using the Simplex method, to see if there exists any feasible solution of torques $\boldsymbol{\tau}$ at manipulator joints and wrenches \mathbf{f}_g at grip points. If it fails, we regard the configuration \mathbf{q} (and its corresponding grasp \mathbf{g}) unstable against the operation \mathbf{F} .

4.2 Deviated operations modelling and stability check

Conic operation model Obviously, the idealized model is only applicable to cases where forceful operations can be precisely applied as expected. However, a forceful operation in actual applications will inevitably deviate from its expected operation axis to some extent, especially for human-applied forceful operations. For example, as illustrated in Fig. 7a, for a drilling operation, rather than an idealized force along the

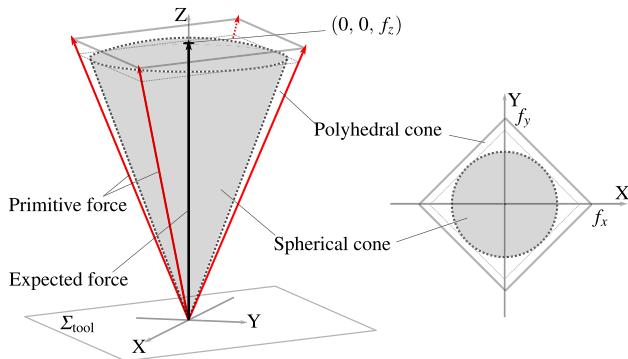


Fig. 8 The spherical cone models the distribution of an operation force f under deviations. The polyhedral cone circumscribes the spherical cone, approximating it with a limited number of primitive forces

expected drilling axis, the actual applied force can deviate towards any direction within a certain cone. Figure 7b shows a similar observation of a cutting operation.

We take such force uncertainties into account, formulating a *spherical cone* model to address deviations in forceful operations. As shown in Fig. 8, geometrically, the force deviation is bounded by a spherical cone (illustrated as the grey cone) centred with the expected operation force f (the dark solid vector along the $+z$ axis), while the actual operation force can be any force within the cone.

Stability check with conic model The spherical cone models a forceful operation F as a continuous set of forces that can possibly be applied by the operation, while the shape of the cone can be extracted from experimental data. However, such a continuous conic model poses a challenge for stability check. Specifically, to check if a candidate configuration q is able to keep an object stable under an operation F , all possible deviated forces within the continuous spherical cone must be checked for the sake of robustness, for which a naive discretization approach would be computationally extremely expensive.

To reduce the computational complexity but without degenerating robustness of stability check using the spherical cone model, we propose to approximate the spherical cone with a n_F -edged *circumscribed* polyhedral cone (illustrated as the outer polyhedron in Fig. 8).

As shown in Fig. 8, the polyhedral cone circumscribes and bounds the spherical cone, including all deviated forces in the spherical cone. It also contains a small set of additional forces (the space between the polyhedral cone and the spherical cone) due to the geometric relation between two cones, which makes the polyhedral cone conservative by enlarging the actual force distribution. This is advantageous in the sense of producing no false positives, while the cost we pay may be false negatives.

More importantly, the polyhedral cone is a convex cone rooted at the origin. Then, given a forceful operation F , any

deviated force f' within its corresponding spherical cone, including the idealized operation force, can be denoted as a linear combination of the n_F edge forces $\{\hat{f}_1, \dots, \hat{f}_{n_F}\}$ (illustrated as the red edge vectors in Fig. 8) of its corresponding polyhedral cone:

$$f' = \sum_{i=1}^{n_F} k_i \hat{f}_i \quad \text{and} \quad \sum_{i=1}^{n_F} k_i \leq 1, \quad k_i \geq 0 \quad (3)$$

Here we refer to $\{\hat{f}_1, \dots, \hat{f}_{n_F}\}$ as *primitive forces* for the operation F . Then, we can easily define:

Theorem 1 Give a forceful operation F , if a grasp configuration q is stable against all its primitive forces $\{\hat{f}_1, \dots, \hat{f}_{n_F}\}$, then the configuration q is stable against any possible deviated force f' within its corresponding spherical cone, i.e. the configuration q is stable against the operation F .

Proof Consider a grasp configuration q and a determinate external force f , in Sect. 4.1 we formulate the stability check problem as a linear programming in Eq. 2. For the sake of simplicity, here we denote the linear mapping from operation force f to robot solution $\tilde{f} = (\tau, f_g)$ (i.e., manipulator joint torques τ and grasp wrenches f_g) as

$$\tilde{f} = \text{LP}(f), \quad \tilde{f}_{\min} \leq \tilde{f} \leq \tilde{f}_{\max} \quad (4)$$

where LP denotes the linear mapping in Eq. 2a and $\tilde{f}_{\min}, \tilde{f}_{\max}$ denote robot limits in Eq. 2b. Herein, we assume $\tilde{f}_{\min} \leq 0$ and $\tilde{f}_{\max} \geq 0$ for simplicity.

Then, given a forceful operation F , if q is stable against all its primitive forces, the planner can find a solution \tilde{f}_i for each primitive force \hat{f}_i meeting Eq. 4 ($i = 1, 2, \dots, n_F$).

In this context, for any force $f' = \sum_{i=1}^{n_F} k_i \hat{f}_i$ within the spherical cone, we can always find a robot solution $\tilde{f}' = \sum_{i=1}^{n_F} k_i \tilde{f}_i$ that satisfies:

$$\begin{aligned} \tilde{f}' &= \sum_{i=1}^{n_F} k_i \tilde{f}_i \\ &= \sum_{i=1}^{n_F} k_i \text{LP}(\hat{f}_i) = \text{LP}(\sum_{i=1}^{n_F} k_i \hat{f}_i) = \text{LP}(f') \end{aligned}$$

and

$$\begin{aligned} \tilde{f}_{\min} &\leq \sum_{i=1}^{n_F} k_i \tilde{f}_{\min} \\ &\leq \tilde{f}' = \sum_{i=1}^{n_F} k_i \tilde{f}_i \leq \sum_{i=1}^{n_F} k_i \tilde{f}_{\max} \leq \tilde{f}_{\max} \end{aligned}$$

That is, the configuration q can provide a solution \tilde{f}' meeting Eq. 4. In other words, the configuration q is stable against any force f' within the spherical cone. \square

This theorem dramatically simplifies the stability check using the spherical cone model but with guaranteed robustness. Specifically, give a forceful operation F and a grasp

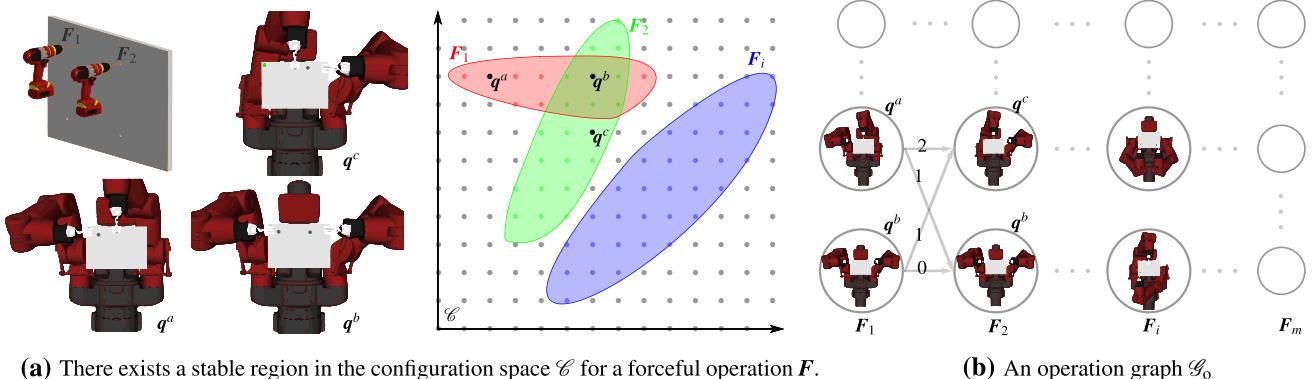


Fig. 9 We build an operation graph to search for a minimal sequence of configurations $\{\mathbf{q}_j\}_{j=1}^n$ stable against $\{\mathbf{F}_i\}_{i=1}^m$

configuration \mathbf{q} , rather than checking all forces within a continuous spherical deviation cone, the planner can check only a limited number of primitive forces $\{\hat{f}_1, \dots, \hat{f}_{n_F}\}$. If all n_F stability checks succeed, according to Theorem 1, the configuration \mathbf{q} is stable against any force in the spherical cone and thus can be returned as a feasible configuration. In contrast, if any of the n_F stability checks fails, the planner can stop and return \mathbf{q} as unstable without further checking.

The number n_F can be chosen empirically. Note that, a larger n_F would make the polyhedral cone closer to the spherical cone. However, this may require more time for stability check, since for a forceful operation F and a candidate configuration \mathbf{q} , in the worst case, stability check involves checking all n_F primitive forces. A smaller n_F , in contrast, would make the polyhedral cone more conservative by containing additional forces outside the spherical cone. This might lead to the loss of feasible solutions, since the polyhedral cone imposes a relatively stronger constraint by covering additional forces into stability check. In this sense, the choice of n_F is more or less a trade-off between planning efficiency and the loss of feasible solutions due to conservative approximation. In Sect. 7.1, we present experiment results to show how the choice of n_F will affect the planning process.

5 Planning

This section presents details of our planner layer by layer as illustrated in Fig. 4.

5.1 Generating $\{\mathbf{q}_j\}_{j=1}^n$ Stable against $\{\mathbf{F}_i\}_{i=1}^m$

The planner starts by generating a minimal sequence of grasp configurations $\{\mathbf{q}_j\}_{j=1}^n$ that are stable against $\{\mathbf{F}_i\}_{i=1}^m$ (illustrated as the sequence of three grasp configurations in the top layer of Fig. 4).

Given a forceful operation F , theoretically, there exists a set of configurations in the configuration space \mathcal{C} , i.e. a *stable region*, in which all configurations are stable against the operation F . For example, in Fig. 9a, we show two subsequent drilling operations F_1 and F_2 of the table assembly task in Fig. 9b, while the red and green regions in the configuration space \mathcal{C} illustrate such stable regions for them respectively (red region for F_1 and green region for F_2). In this sense, finding a sequence of system configurations stable against $\{\mathbf{F}_i\}_{i=1}^m$ can be regarded as finding a sequence of configurations to visit all stable regions for the operations in $\{\mathbf{F}_i\}_{i=1}^m$ in order.

Further, there might be *intersections* between these stable regions. Within each intersection, any configuration would be stable against the corresponding multiple operations. For example, the configuration \mathbf{q}^b in Fig. 9 is stable against both F_1 and F_2 , since \mathbf{q}^b lies inside the intersection of stable regions for the operations F_1, F_2 . We use these intersections to minimize the number of regrasps in the sequence.

Specifically, to create such a minimal sequence of configurations $\{\mathbf{q}_j\}_{j=1}^n$, our planner first samples a set of candidate configurations in \mathcal{C} . To sample configurations that are likely to be stable against a variety of operations, i.e. configurations in the intersections, the planner starts by sampling grasps uniformly on the object. Then, using such a sampled grasp configuration \mathbf{g} and the desired object pose \mathbf{T} , the planner solves inverse-kinematics problem, which may output multiple achievable solutions, and randomly picks one configuration \mathbf{q} .

For each sampled configuration \mathbf{q} , the planner identifies the operations in $\{\mathbf{F}_i\}_{i=1}^m$ that the configuration \mathbf{q} is stable against. We then build an **operation graph** \mathcal{G}_o using these stable configurations as shown in Fig. 9b. The operation graph is an acyclic directed weighted graph. Specifically, in the operation graph, each column corresponds to a forceful operation. That is, the nodes in the i -th column are sampled configurations that are stable against the operation F_i . Further,

we define a link between every two nodes in neighbouring columns, and associate the link with a weight using *the number of gripper moves* required from one configuration to the other. For example, the weight for the link between the node q^b in the first column and the node q^b in the second column is zero, since they correspond to a same configuration and thus no regrasp is required. Similarly, if two configurations differ only by one gripper contact on the object, the weight for their corresponding link is set to be one (e.g. q^b and q^c). Otherwise, the weight would be two for a dual arm robot. Note that one can come up with other weighting schemes, e.g. one that takes the overall motion trajectories for regrasping into account.

At this point, our problem in this layer is formulated as a graph search problem. Given a manipulation query, the expected output is a path that starts from one node in the leftmost column for operation F_1 and ends with a node in the rightmost column for operation F_m .

By searching the operation graph \mathcal{G}_o , e.g. using *Dijkstra's algorithm*, the planner can generate a candidate sequence $\{q_j\}_{j=1}^n$ with the least number of gripper moves based on the current set of samples. Hereafter, we call this planner the *min-regrasp* planner.

We provide the pseudo-code for this layer of the planner in Algorithm 1 in the procedure PlanStableSequence. In line 1, the planner constructs the operation graph \mathcal{G}_o as described above. In line 2, the planner searches the graph (e.g. using Dijkstra's algorithm) for a candidate sequence $\{q_j\}_{j=1}^n$. Then the planner iterates over every subsequent pair of configurations in $\{q_j\}_{j=0}^n$ (line 4), attempting to plan a regrasp between them, which is explained below. If the regrasp planning fails between any two configurations (line 6), the planner removes the failing link from the graph in Fig. 9b (line 7), and then re-searches the graph to generate a new candidate sequence (line 8).

Note that, building the whole operation graph \mathcal{G}_o requires knowing all forceful operations $\{F_i\}_{i=1}^m$ beforehand. However, there may be cases for which the forceful operations are revealed progressively, e.g. one by one. In such cases, the operation graph \mathcal{G}_o can be constructed as the next operation(s) is specified, and then be searched greedily. We call this version the *greedy* planner.

In this layer, the planner generates a minimal sequence of grasp configurations $\{q_j\}_{j=1}^n$ to keep the target stable under the application of forceful operations $\{F_i\}_{i=1}^m$. Hereafter, the planner generates collision-free and stable regrasping trajectories $\{t_j\}_{j=1}^n$ to connect every two subsequent grasp configurations in $\{q_j\}_{j=0}^n$ (q_0 is the initial system configuration), while t_j corresponds to a contained regrasping task from q_{j-1} to q_j .

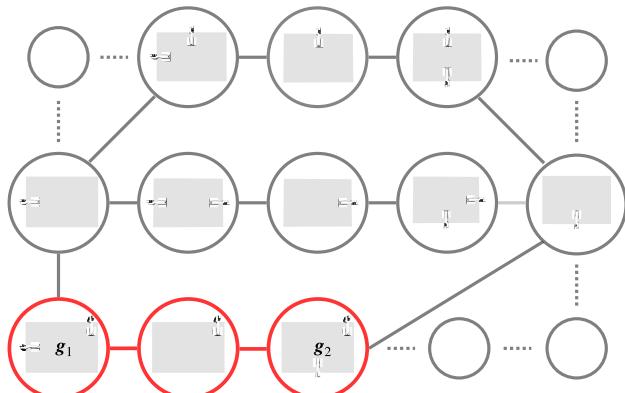


Fig. 10 A grasp graph \mathcal{G}_g : Each node in the grasp graph represents a bimanual or a unimanual grasp

5.2 Connectivity of grasps

Given any two subsequent configurations q_s , $q_t \in \{q_j\}_{j=1}^n$ generated above, rather than directly searching in the high-dimensional composite configuration space for a regrasping motion trajectory, the planner first generates a grasp sequence to guide and constrain the following search into a limited sequence of grasp manifolds.

Specifically, the planner starts by finding a sequence of grasps $\{g_i\}_{i=1}^{n_g}$ on the object, which moves the system from the grasp g_s to the grasp g_t , where g_s, g_t are the grasps at configurations q_s, q_t and denoted as g_1 and g_{n_g} in the sequence $\{g_i\}_{i=1}^{n_g}$ respectively. For example, to regrasp from g_1 to g_2 in the top layer of Fig. 4, the robot must go through a sequence of intermediate grasps (e.g. $\{g'\}$) to switch from the grasp g_s to the grasp g_t on the object.

In the case of dual-arm system used in this work, these intermediate grasps are either bimanual or unimanual. We represent the connectivity of these grasps as a *grasp graph* \mathcal{G}_g as illustrated in Fig. 10. Each node in the graph \mathcal{G}_g represents a grasp on the object. A bimanual and a unimanual grasp are connected if the unimanual grasp is contained by the bimanual grasp. For example, we say a bimanual grasp (g_l, g_r) contains a unimanual grasp (g_l), since they share a common left gripper contact on the object. Building such a grasp graph requires the generation of feasible grasps on the object, which can be pre-specified or can be generated using a general grasp planner, e.g. Miller and Allen (2004).

Then, the planner searches the grasp graph \mathcal{G}_g to get a sequence of grasps $\{g_i\}_{i=1}^{n_g}$, which connects the grasps g_s and g_t (denoted as g_1 and g_{n_g} respectively in the sequence $\{g_i\}_{i=1}^{n_g}$) with an alternating sequence of bimanual and unimanual grasps. Figure 10 highlights in red the shortest grasp sequence to move from the grasp g_s to the grasp g_t . Note that there might be other longer feasible grasp sequences in the graph as well.

The grasp sequence $\{\mathbf{g}_i\}_{i=1}^{n_g}$ acts as an abstract plan to guide and constrain the following motion planning into a limited sequence of grasp manifolds $\{\mathcal{M}(\mathbf{g}_i)\}_{i=1}^{n_g}$ (illustrated as the three foliations in the bottom of Fig. 4). In Algorithm 1, the procedure PlanRegrasp outlines this process. On lines 1–2, the planner builds the grasp graph \mathcal{G}_g and searches it to obtain a sequence of grasps $\{\mathbf{g}_i\}_{i=1}^{n_g}$ as described above.

The lower layers of the planner then try to plan the motion from \mathbf{q}_s to \mathbf{q}_t through the planned grasps in $\{\mathbf{g}_i\}_{i=1}^{n_g}$ (line 3). If the planner returns with a failure to connect any two grasps \mathbf{g}_i and \mathbf{g}_{i+1} in $\{\mathbf{g}_i\}_{i=1}^{n_g}$ (line 4), then it removes the link between these grasps (line 7), and perform the search again to generate a new grasp sequence (line 8). If the connection is successful, the planner returns the motion trajectory t (line 10) for regrasping.

5.3 Sampling stable intersections of grasp manifolds

Given a grasp sequence $\{\mathbf{g}_i\}_{i=1}^{n_g}$ generated in above layer, the following layers of the planner attempt to generate stable regrasping motions within the grasp manifolds $\{\mathcal{M}(\mathbf{g}_i)\}_{i=1}^{n_g}$. However, the grasp sequence $\{\mathbf{g}_i\}_{i=1}^{n_g}$ provides necessary but not sufficient condition of the connectivity of their corresponding grasp manifolds.

Specifically, to regrasp from \mathbf{g}_i to \mathbf{g}_{i+1} , the planner needs to find at least one transition/regrasping configuration \mathbf{q} within the intersection of their grasp manifolds $\mathcal{M}(\mathbf{g}_i) \cap \mathcal{M}(\mathbf{g}_{i+1})$ (illustrated as the blue points in Fig. 4), such that the configuration \mathbf{q} can be both kinematically feasible and stable against the object gravity. Particularly in our task, the transition from a bimanual grasp to a unimanual grasp may fail, as the object might become unstable against object gravity if the robot directly releases one gripper from the object. Instead, the second configuration in the third layer of Fig. 4 shows a stable regrasping configuration, at which the robot deliberately tilts the object pose before releasing its right gripper for regrasping, such that the remaining gripper can still hold the object stable (as the object gravity will be partially resisted by the gripper structure).

Our planner searches for such stable regrasping configurations in the intersection of their grasp manifolds $\mathcal{M}(\mathbf{g}_i)$, $\mathcal{M}(\mathbf{g}_{i+1})$ by random sampling. Specifically, in Algorithm 1, the procedure SampleIntersection samples a set of k such configurations. To find one such configuration, the planner first samples an object pose in the robot's reachable space (line 4). Then, it solves the inverse-kinematics for the bimanual grasp at the sampled object pose to get a fully-assigned configuration \mathbf{q} (line 5). The planner checks (line 6) whether both grasps \mathbf{g}_i and \mathbf{g}_{i+1} are stable against gravity at \mathbf{q} , using the stability check described in Sect. 4.1. The stable config-

uration \mathbf{q} is then saved as a candidate regrasping connection in the final solution path (line 7).

Algorithm 1 Manipulation Planning under Changing Forces

```

PlanStableSequence ( $\{F_i\}_{i=1}^m, \mathbf{q}_0$ ):
1:  $\mathcal{G}_o \leftarrow$  Sample stable configurations in  $\mathcal{C}$  and build an operation graph as shown in Fig. 9b
2:  $\{\mathbf{q}_j\}_{j=1}^n \leftarrow$  GraphSearch( $\mathcal{G}_o$ )
3:  $\{\mathbf{q}_j\}_{j=0}^n \leftarrow$  Insert  $\mathbf{q}_0$  to the beginning of  $\{\mathbf{q}_j\}_{j=1}^n$ 
4: for each subsequent  $\mathbf{q}_j$  and  $\mathbf{q}_{j+1}$  in  $\{\mathbf{q}_j\}_{j=0}^n$  do
5:    $t_{j+1} \leftarrow$  PlanRegrasp( $\mathbf{q}_j, \mathbf{q}_{j+1}$ )
6:   if PlanRegrasp failed then
7:      $\mathcal{G}_o \leftarrow$  Remove failing edge from graph  $\mathcal{G}_o$ 
8:   Go to line 2
9: return ( $\{\mathbf{q}_j\}_{j=1}^n, \{t_j\}_{j=1}^n$ )

PlanRegrasp ( $\mathbf{q}_s, \mathbf{q}_t$ ):
1:  $\mathcal{G}_g \leftarrow$  Sample grasps and build graph in Fig. 10
2:  $\{\mathbf{g}_i\}_{i=1}^{n_g} \leftarrow$  GraphSearch( $\mathcal{G}_g, \mathbf{q}_s, \mathbf{q}_t$ )
3:  $t \leftarrow$  Connect( $\mathbf{q}_s, \{\mathbf{g}_i\}_{i=1}^{n_g}, \mathbf{q}_t$ )
4: if Connect failed or  $t$  is None then
5:   if maximum number of attempts reached then
6:     return failure
7:    $\mathcal{G}_g \leftarrow$  Remove failing edge from graph  $\mathcal{G}_g$ 
8:   Go to line 2
9: else
10:  return  $t$ 

Connect ( $\mathbf{q}_s, \{\mathbf{g}_i\}_{i=1}^{n_g}, \mathbf{q}_t$ ):
1: if  $n_g = 1$  then
2:    $t \leftarrow$  MotionPlan( $\mathbf{q}_s, \mathbf{q}_t$ ) using grasp  $\mathbf{g}_{n_g}$ 
3:   if MotionPlan successful then
4:     return  $t$ 
5:   else
6:     return failure
7:  $S \leftarrow$  SampleIntersection( $\mathbf{g}_1, \mathbf{g}_2$ )
8: for each  $\mathbf{q}$  in  $S$  do
9:    $t \leftarrow$  MotionPlan( $\mathbf{q}_s, \mathbf{q}$ ) using grasp  $\mathbf{g}_1$ 
10:  if MotionPlan successful then
11:    return  $t +$  Connect( $\mathbf{q}, \{\mathbf{g}_i\}_{i=2}^{n_g}, \mathbf{q}_t$ )
12: return failure

SampleIntersection ( $\mathbf{g}, \mathbf{g}'$ ):
1: One of  $\mathbf{g}$  and  $\mathbf{g}'$  must be bimanual. Assuming  $\mathbf{g}$ .
2:  $S \leftarrow \{\}$ 
3: while  $S$  contains less than  $k$  elements do
4:    $\mathbf{x} \leftarrow$  Sample pose for object
5:    $\mathbf{q} \leftarrow$  Solve IK with object at  $\mathbf{x}$  and grippers at  $\mathbf{g}$ 
6:   if  $\mathbf{q}$  is stable against gravity with both  $\mathbf{g}$  and  $\mathbf{g}'$  then
7:     Add  $\mathbf{q}$  to  $S$ 
8: return  $S$ 

```

5.4 Connectivity of sequence of manifold intersections

In this layer, given two configurations \mathbf{q}_s and \mathbf{q}_t , and stable configurations sampled at the intersections of a sequence of manifolds (i.e. the grasp manifolds corresponding to $\{\mathbf{g}_i\}_{i=1}^{n_g}$),

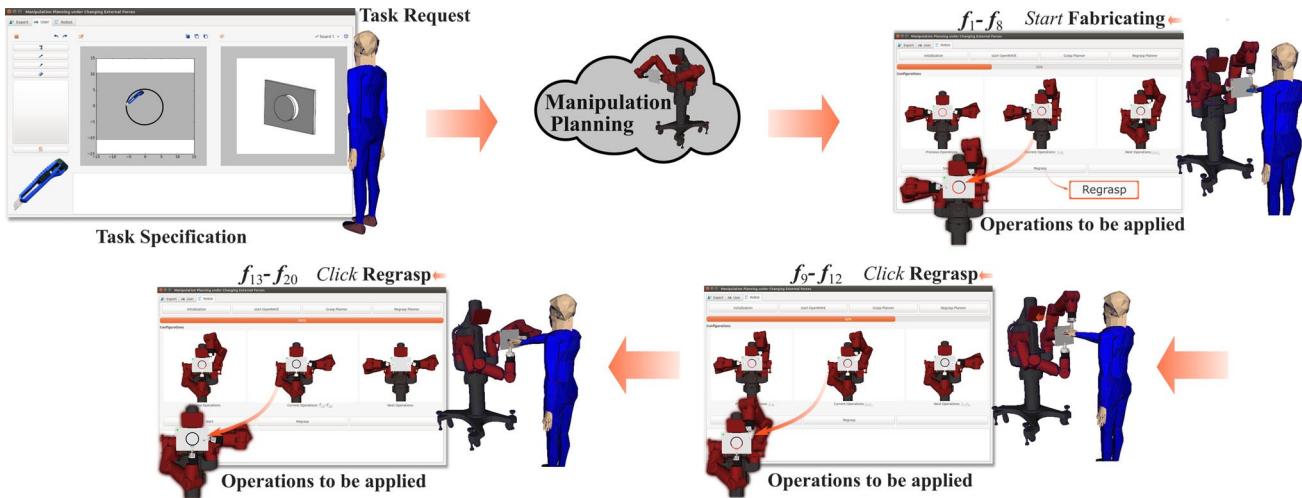


Fig. 11 The work-flow of human-robot collaboration in performing collaborative forceful tasks with the graphical user interface

the planner attempts to generate a collision-free and stable trajectory t that connects q_s to q_t through these manifolds (illustrated as the red connected line segments in Fig. 4).

In Algorithm 1, the procedure *Connect* implements this process as depth-first-search. Given a current configuration q_s and a sequence of grasps $\{g_i\}_{i=1}^{n_g}$ (where g_1 is the grasp in q_s), the planner samples the intersection of the first two grasp manifolds in the sequence for a set of stable configurations S (line 7). Then it tries to plan a motion from q_s to a sampled configuration $q \in S$ (line 9). Note that this is a motion plan within a single manifold and thus can be solved by existing closed-chain or single-arm motion planners. However, the object motion must be also stable against gravity, for which the constrained motion planners (Berenson et al. 2011; Jaillet and Porta 2013) can be used. If the motion planning succeeds, the trajectory is returned along with a recursive call to the depth-first-search. Lines 1–6 handle the simple case where q_s and q_t are already in a same manifold.

6 A graphical user interface

Before presenting experiments to show the performance of our planner, we present a graphical user interface to close the loop of robot-assisted forceful manipulation.

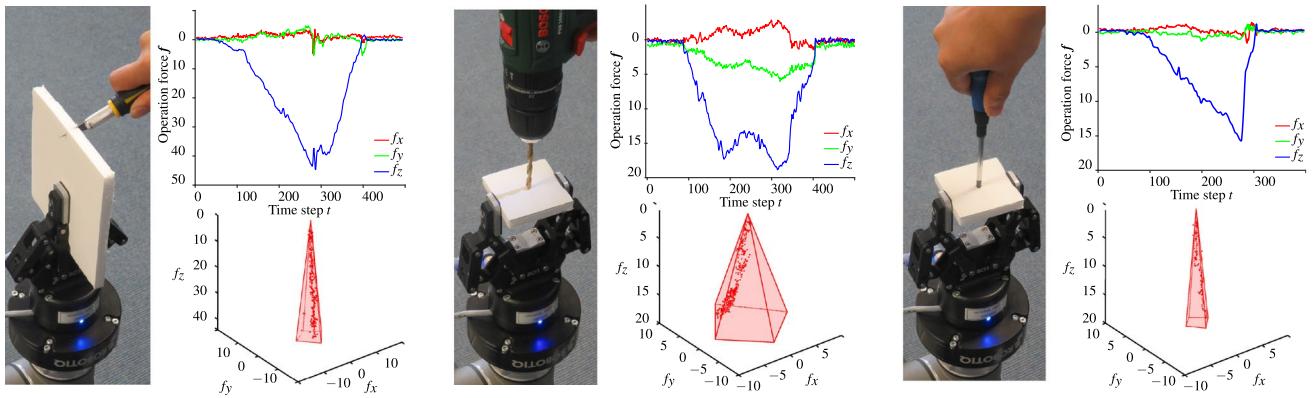
Using the interface, (1) A human user can easily specify forceful tasks, i.e. sequences of operations on selected objects; (2) The robot connected to the interface assists the user in performing customized tasks by stably manipulating the selected objects and providing *operation instructions*.

Fig. 11 illustrates the overall work-flow of the robot-assisted manipulation with the user interface using the circular cutting task. First, the human user specifies a desired forceful task by choosing a tool(s) (e.g. a cutter or a drill)

to draw on a selected object. For example in Fig. 11, the human first selects a cutter and a rectangular board, and then draws a circle on the board to specify the circular cutting task (Fig. 11-Task Specification). Once receiving confirmation, the interface triggers a planning process (with our planners acting as the underlying planners) to generate efficient manipulation plans as discussed in previous sections (Fig. 11-Manipulation Planning).

After planning, the interface controls the robot to assist the human in performing the specified forceful operations, as well as providing *operation instructions* to the human according to the manipulation plan (Fig. 11-Fabricating).

Specifically, during manipulation, the robot assistant manipulates the target object to the planned configurations in sequence and stabilizes it under the application of forceful operations. At each planned configuration, the interface instructs the human user to apply a subsequence of resistible forceful operations by visually displaying the subsequence in both the interface and the robot head monitor (Fig. 11-Operations to Be Applied). After completing the instructed operations, the human presses a Regrasp button provided by the interface to command the robot to go the next planned configuration(s) (Fig. 11-Regrasp). The regrasp button is how the human notifies the system that the subsequence of forceful operations are applied (In the future, we aim to improve the system by automatic perception of human operations). In this manner, the interface connects the robot assistant and the human users to perform forceful tasks interactively. In the next section, we present real robot experiments using the interface.



(a) The distribution of a cutting force.

(b) The distribution of a drilling force.

(c) The distribution of a puncturing force.

Fig. 12 We collected experimental data from 30 operation trials to build data-driven conic models for forceful operations in our experiments. The red polyhedral cones are the extracted models. The red dots inside the cones are the force data over one operation trial

7 Experiments and analysis

We present a variety of experiments in this section to verify the performance of our proposed planners.

Experimental Setting: We applied our planners to the Baxter robot from Rethink Robotics. Baxter has two 7-DOF manipulators, each equipped with a parallel plate gripper. We tested our planners in OpenRAVE (Diankov and Kuffner 2008) for simulated experiments.

For Algorithm 1, we used the NetworkX (Hagberg et al. 2008) for graph construction and search, and BiRRT (Kuffner and LaValle 2000) for motion planning. In our implementation, we set the maximum number of attempts to be 3 for the procedure PlanRegrasp and the number of samples to be 20 for the procedure SampleIntersection.

We measured the grip force/torque limits (as explained previously in Sect. 4.1) of a Baxter gripper gripping a foam board. Specifically, along each principal axis, we applied an incremental amount of forces and torques on the foam board that is gripped by the Baxter gripper, to find the point when the object started to slide between the parallel gripper plates or when the object tilted more than 5° due to finger structure deformation. In this manner, we tested the grip limits as $f_{g,i}^{\max} = [13\text{ N}, 40\text{ N}, 100\text{ N}, 0.5\text{ Nm}, 0.1\text{ Nm}, 0.15\text{ Nm}]$ and $f_{g,i}^{\min} = [-13\text{ N}, -40\text{ N}, -13\text{ N}, -0.5\text{ Nm}, -0.1\text{ Nm}, -0.15\text{ Nm}]$ ³.

Experiments Overview: We conducted three categories of experimental studies, including:

- Modelling Forceful Operations: We collected experimental data to capture the conic distributions of forceful

operations involved in our experiments and studied the effect of using conic model on stability check (Sect. 7.1);

- Simulated Experiments: We tested our planner on a variety of forceful tasks to verify its performance in terms of minimizing the number of regrasps, planning stable regrasps and time efficiency (Sect. 7.2);
- Real Experiments: We did a set of real human-robot experiments to further study the feasibility of our planner in real forceful human-robot applications. We used the graphical user interface presented in Sect. 6 for task specification and robot-assisted fabrication (Sect. 7.3).

7.1 Modelling forceful operations

We tested our planners on three types of forceful operations, cutting, puncturing and drilling on rigid foam boards as shown in Fig. 12. We collected experimental data to capture their force distributions in the wrench space.

As discussed in Sect. 4.1, using the idealized model:

- A cutting operation applies a pure cutting force along an expected cutting axis;
- A puncturing operation applies a pure intruding force along an expected puncturing axis;
- A drilling operation applies a rotational torque about an expected drilling axis plus a drilling force along the axis.

Therefore, the idealized model requires identifying the maximum operation forces to extract values of parameters f_z and τ_z from experimental data.

While for the conic model, we need to determine their conic deviation ranges, i.e. extracting values of parameters $f_{x/y/z}$ and $\tau_{x/y/z}$. To do this, we applied each type of operations 30 times separately, collecting force data using a 6D force/torque sensor (FT150 from Robotiq) as shown in

³ Along the $+z$ direction, the object can rest against the gripper palm, therefore the planner adopted a large force limit (100 N) for P_z^+ .

Table 1 Numbers of candidate configurations remaining stable after being checked using the conic model with different n_F

	Cutting	Drilling	Puncturing
Idealized model	50	50	50
Conic model	$n_F = 4$	35	40
	$n_F = 8$	40	42
			46

Fig. 12. During each trial, we recorded the force and torque values at different time steps at the rate reported by the sensor. This means, for each operation trial, we collected between 400–500 force data points. For each category, we compute the polyhedral cone as discussed in Sect. 4.1 which contains the force distributions over all time steps over the 30 operation trials. In Fig. 12, each sub-figure shows the distribution of operation forces in one operation trial⁴. The red polyhedral cone in the lower right of each sub-figure is the corresponding polyhedral cone model ($n_F = 4$) extracted for from all force data of 30 trials.

Specifically, for cutting operations, we measured $f_z = 45\text{ N}$, $f_x = 4\text{ N}$ and $f_y = 6\text{ N}$. For drilling operations, $f_z = 19\text{ N}$, $f_x = 6\text{ N}$ and $f_y = 6\text{ N}$. The torque generated by drill bit rotation is much smaller than the one generated by drilling force, thus we simply neglect it and assume $\tau_x = \tau_y = \tau_z = 0\text{ Nm}$. For puncturing operations, $f_z = 16\text{ N}$, $f_x = 2\text{ N}$ and $f_y = 2\text{ N}$.

Effect of Conic Model on Stability Check: To test the effect of conic model on stability check, for each type of operations above, we test 20 forceful operations evenly distributed on the object surface. For each forceful operation, we first generate 50 stable configurations using the idealized model (i.e. 50 different complete robot configurations grasping the object stably against the idealized force). We then check the stability of these configurations again using the conic model (50 * 20 for the 20 operations, giving a total of 1000 stability checks for each operation type) with $n_F = 4$ and $n_F = 8$ respectively.

The numbers of configurations remaining feasible out of the 50 configurations (which are feasible using the idealized model) are shown in Table 1. Note that the number of feasible configurations for $n_F = 8$ is larger than for $n_F = 4$. This is reasonable, since for a forceful operation, a larger n_F will make the polyhedral cone approximation closer to its real spherical cone distribution.

In the following experiments, we used the conic model for robust stability check and set the number of primitive forces $n_F = 4$ in modelling involved forceful operations.

⁴ In Fig. 12 we show the force distribution of one operation trial for the sake of visual clarity, but the models are extracted from data of 30 trials.

7.2 Planning performance in simulated experiments

In simulated experiments, we implemented our planner on three categories of forceful tasks, including

- Random-puncturing Each random-puncturing task contains 10 puncturings randomly distributed on the surface of a foam board. An example is shown in Fig. 13;
- V-puncturing Each task consists of 40 puncturing operations along two random line segments meeting at a common point. An example is shown in Figs. 14 and 15;
- Drilling and cutting Each task involves 4 drillings and a subsequence of cutting operations as shown in Fig. 18.

We generated 100 random tasks for each task category above. *Analysis of number of regrasps* First, we compared the performance of our planners, *min-regrasp* and *greedy*, with a *random* planner in reducing the number of regrasps. The random planner acts as a baseline approach. For the first forceful operation, the random planner performs sampling in the configuration space until it finds the first stable configuration against the operation. Then, for any subsequent operations, it first checks whether the current configuration is still stable. If not, it falls back to random sampling.

Table 2 shows the average number of regrasps generated by the three planners over 100 random forceful tasks. For the random-puncturing tasks, the random planner generates almost a new grasp and thus one bimanual regrasp for every forceful operation (maximum 20 regrasps for 10 operations). The min-regrasp dramatically reduces the number of regrasps (5.4 regrasps on average for 10 operations, an example solution is shown in Fig. 13). The greedy planner also performs well in reducing regrasps (8.2 regrasps on average).

Similarly, for the V-puncturing tasks, the random planner generates plans with a large number of regrasps (52.9 regrasps for the 40 operations of a V-puncturing task on average), while the min-regrasp planner just needs 1.6 regrasps on average (an example solution is shown in Fig. 15). The greedy planner also shows a much better performance compared with the random planner, but still worse than the min-regrasp planner. For example, as shown in Fig. 14, one solution generated by the greedy planner requires the grippers to climb along the edges of the board up and down frequently to follow the movement of the puncturing operations, while the min-regrasp planner comes up with a plan of just two regrasps in Fig. 15. Similar results can also be found for the drilling and cutting tasks.

We also counted the number of samples the random planner needed before it found a stable grasp. On average, the random planner needed 41.1 samples for each forceful operation above, showing that planning is necessary, since random grasps have little chance of being feasible.

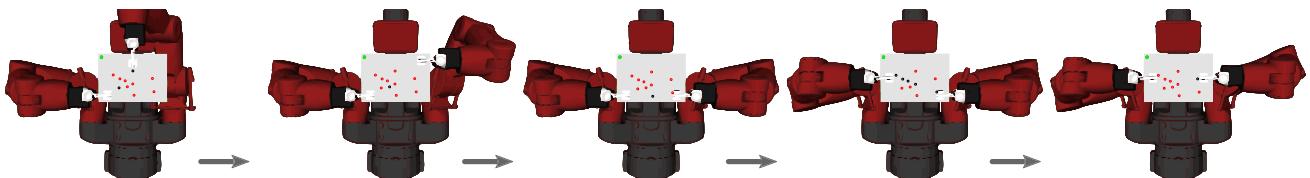


Fig. 13 A grasp sequence by the min-regrasp planner for a random-puncturing task. The dark points indicate the puncturing operations planned to be applied during the current grasp. The arrows indicate regrasp actions

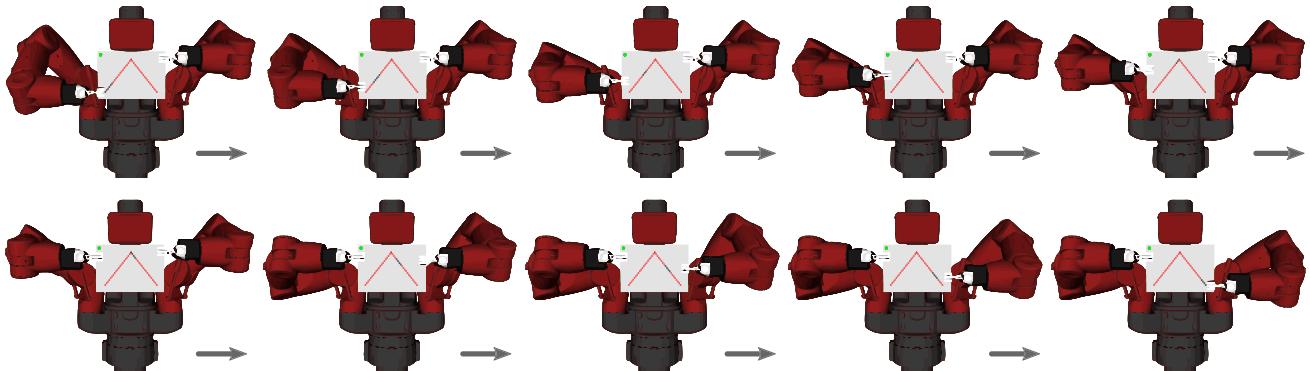


Fig. 14 A grasp sequence by the greedy planner for a V-puncturing task. The dark points indicate the puncturing operations planned to be applied during the current grasp. The arrows indicate regrasp actions

Table 2 Numbers of regrasps (with standard deviations in parentheses) of three planners on three categories of tasks

	Random-puncturing	V-puncturing	Drilling and cutting
Random	19.7(0.7)	52.9(10.1)	5.8(2.1)
Greedy	8.2(2.1)	5.3(1.9)	3.1(0.8)
Min-regrasp	5.4(1.3)	1.6(0.6)	2.0(0.5)

Our planners are generalized to common objects, not limited to grasping only rectangular objects. To demonstrate this, we tested the min-regrasp planner with a sequence of 40 circular puncturing operations applied on a round board. A plan with three grasps (two regrasps) is shown in Fig. 16.

Analysis of Planning Stable Regrasps: We also tested the performance of our planner on *light* and *heavy* objects respectively. For light objects, the robot can perform regrasps by directly releasing and re-placing its grippers on the object, whereas the robot might need to move a heavy object to certain intermediate poses before regrasping. Similarly, we ran the planner on the 100 forceful tasks for each category as discussed above.

Figure 17 shows an example sequence to regrasp a heavy object. For a light object, the robot can stably grasp and move the object using just a single gripper at most reachable configurations. Thus, mostly, the robot can directly release off to regrasp the object, without the need of reorienting it to intermediate configurations. However, for a heavy object, as discussed previously, the object may slip down between gripper fingers if the robot directly releases one gripper. That is, the robot needs to find intermediate configurations at which one single gripper is still enough to keep the object stable. In

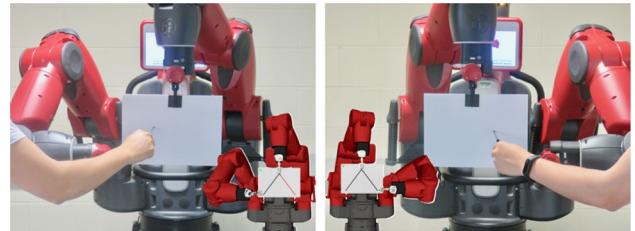


Fig. 15 A plan by the min-regrasp planner for a V-puncturing task which contains two regrasps

Fig. 17, the robot first transfers the object to configurations in Fig. 17b,d before releasing one gripper. After releasing, most object weight will be resisted by the forces arising from gripper finger structure as shown in Fig. 17c, e, which are much larger than the frictional forces between the object and finger surfaces.

Analysis of Planning Time: Table 3 shows the average planning time each layer of the planner takes, including time for generating stable grasp sequences (StabSeq for short in Table 3), time for generating and searching the grasp graph combined with sampling intersections (SampInt, for short) and motion planning (Connect, for short). As the table shows,

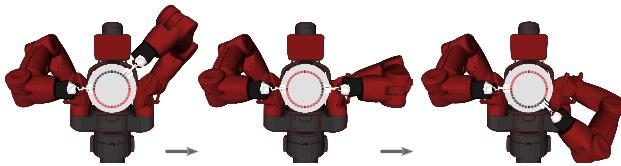


Fig. 16 A grasp sequence by the min-regrasp planner for 40 circular puncturing operations on a round board

most time is spent on motion planning, while the time for planning stable grasp sequences and sampling intersections is negligible. Planning for the heavy objects takes longer time since finding stable regrasp configurations and motion trajectories is more difficult.

In addition, it is notable that given a forceful task, the time for motion planning (and thus the overall planning time) is nearly proportional to the number of regrasps (which can be found in Table 2) required in the corresponding manipulation plan, not to the number of involved forceful operations, as each regrasp corresponds to a motion plan request. For example, for a V-Puncturing task involving 40 forceful operations, the overall planning time is about 85 s for a solution of 1.6 regrasps on average, while for a Random-Puncturing task involving 10 forceful operations, the overall planning time is about 310 s for a solution of 5.4 regrasps.

7.3 Planning performance in human-robot experiments

We did a variety of real human-robot experiments to further verify the feasibility of our system in real applications. A recorded video of all these experiments can be found in the multimedia extension of this paper.

Figures 1, 15 and 18 show the implementations of forceful tasks discussed above. Figure 20 shows a solution by the min-regrasp planner for the table assembly task discussed in Fig. 3, which consists of a large sequence of drilling, cutting

and inserting operations. As shown, the solution involves only 3 different grasp configurations through the whole task.

We also performed 10 human-robot experiments using the graphical user interface introduced in Sect. 6. Before these experiments, 10 human participants were fully explained the usage of the interface and the robot system. Then they specified and performed preferred forceful tasks via the interface as explained in Sect. 6. Figure 19 shows one such experiment, where the user customizes and then cuts a square piece off from the board. During these experiments, we regarded an interaction as failure if it had any unexpected interruption, e.g. unstable operations due to inappropriate grasp. Among these ten experiments, nine interactions succeeded with a small number of regrasps ranging from 1 to 4. One interaction failed due to a collision between the robot gripper and the object during regrasping, which can be seen from time 13:18 to 13:24 in the video accompanying this paper. This is mainly because of the uncertainty in the robot system and can be improved by automatic perception of system motion.

We also collected the interaction time of each part during interactions. Over these 10 experiments, on average, the task specification took 39.5(3.5)s (standard variance is in parentheses). The manipulation planning took 44.5(9.3)s and the Fabrication took 191.3(44.5)s.

8 Conclusion and future work

Our planning approach allows a multi-arm robotic system to stably and fluidly interact with a human co-worker applying forceful operations on an object. Importantly, the planner minimizes the required regrasps—which in turn are performed without any support surface. The proposed planners are capable of addressing uncertainties in the forceful interaction, inherent to human-centred applications. The system has been successfully assessed in different human-robot experi-

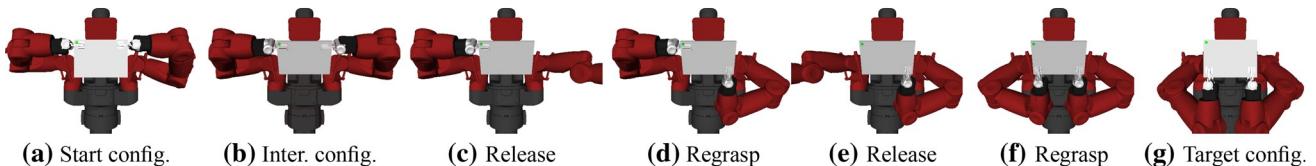


Fig. 17 Regrasping a heavy object. The robot moves the heavy object to some intermediate poses before regrasping

Table 3 Planning time for both heavy and light objects

	Random-puncturing			V-puncturing			Drilling and cutting		
	StabSeq	SampInt	Connect	StabSeq	SampInt	Connect	StabSeq	SampInt	Connect
Heavy	1.8(0.1)	12.6(0.9)	299.1(40.3)	5.1(0.5)	3.3(0.3)	77.2(11.5)	0.7(0.1)	3.5(0.2)	94.1(16.2)
Light	1.6(0.2)	\	107.5(10.1)	4.9(0.6)	\	29.8(5.6)	0.8(0.1)	\	47.5(7.8)

Times are in seconds. Standard deviations are in parentheses

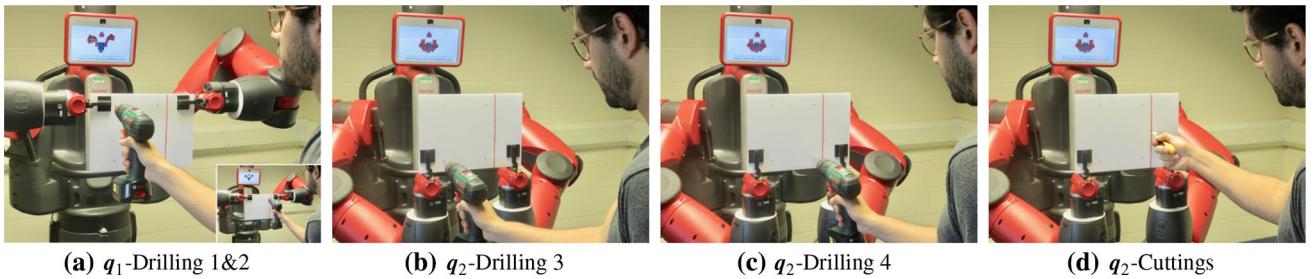


Fig. 18 A grasp sequence by the min-regrasp planner for the Drilling and Cutting task

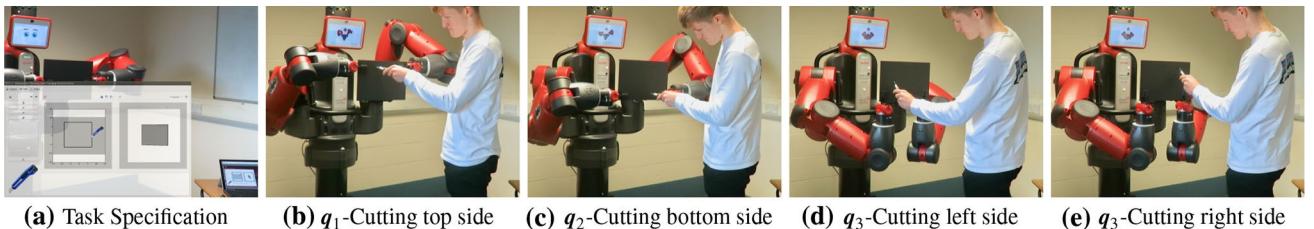


Fig. 19 Human-robot collaboration-a square cutting task

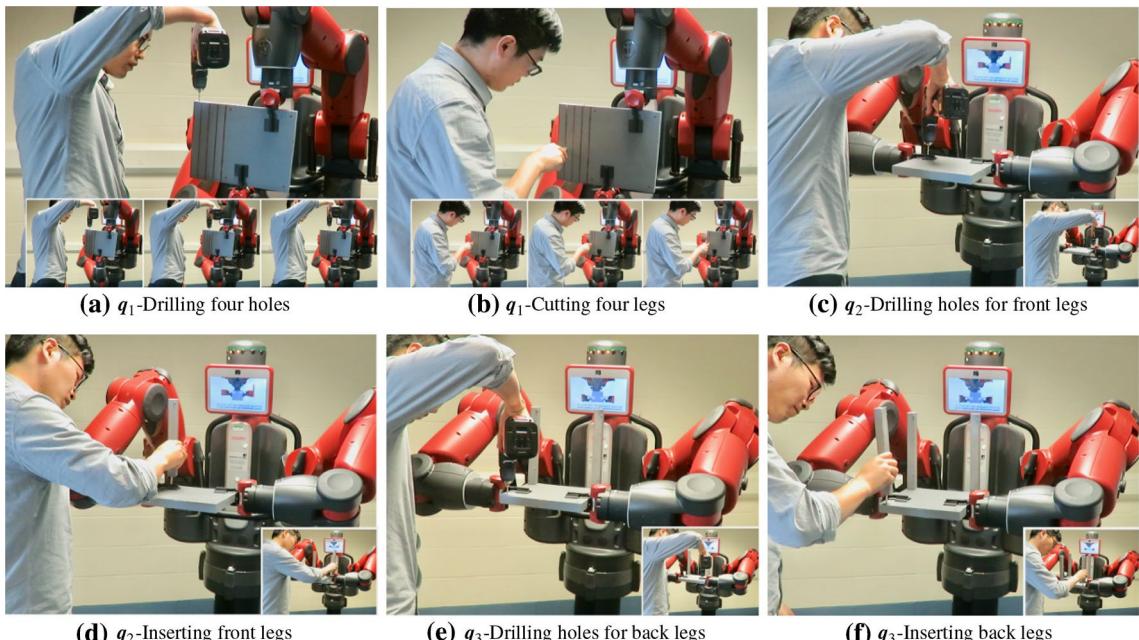


Fig. 20 A solution with only 3 regrasps by the min-regrasp planner for the table assembly task in Fig. 3

ments. We believe the planning system presented here can be a key component in a human-robot collaboration framework.

There are multiple ways the proposed methods and system can be improved.

In the human-robot experiments, we can see sometimes that the planner generates system configurations that are relatively uncomfortable for humans. For example, the robot in Fig. 20a holds the board at a configuration stable against a large sequence of drilling and cutting operations. However, this configuration makes the human raise a heavy drill at a

laborious arm configuration. In future work, we aim to take human comfort factors (e.g. the human kinematics (Chen et al. 2018a)) into account during planning, improving the human experience both physically and psychologically while in collaboration with the robot.

In addition, our experiments show that planning time for such tasks can still take tens of seconds. Our quantitative assessment of the simulated experiments has shown that the time efficiency of the planner is limited mostly by the speed of the low-level constrained motion planners, which leaves

room for improvement in future work to either speed up these individual motion plans, or to reduce the number of such motion plan queries, i.e. the number of regrasps.

The system presented here can also be improved such that both the robot and the human move simultaneously to adapt to each other's motion. For example, in Fig. 1, while the human cuts the board, the robot can rotate the board actively to reduce human motion, as well as improving human comfort. This requires real-time tracking of the human operations and quickly computing system configurations that are stable against them.

A perception system can also allow us to improve the communication and coordination between the human and the robot. Particularly, in our setting, the robot needs to detect when the human performs certain forceful operations. In this work this is indicated by the human via a graphical interface. Even though a satisfying level of coordination has been achieved through the interface, we aim to integrate a perception component into our system to further improve the overall fluidity, e.g. using haptic feedback of the robot grippers to detect when the human performs the planned operations. Furthermore, not only the immediate perception of human action, but also the prediction of human intention/motion (Mainprice and Berenson 2013; Knepper et al. 2017) can also benefit forceful human-robot collaboration.

A perception system for the object and robot motion, e.g. a vision-based tracking system, can also be easily integrated to improve the motion accuracy for regrasping.

References

- Abi-Farrag, F., Osa, T., Peters, N. P. J., Neumann, G., & Giordano, P. R. (2017). A learning-based shared control architecture for interactive task execution. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE.
- Alami, R., Simeon, T., & Laumond, J. P. (1990). A geometrical approach to planning manipulation tasks. the case of discrete placements and grasps. In *The fifth international symposium on Robotics research* (pp 453–463). MIT Press.
- Berenson, D., Srinivasa, S., & Kuffner, J. (2011). Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 30(12), 1435–1460.
- Bohlin, R., & Kavraki, L. E. (2000). Path planning using lazy prm. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)* (Vol. 1, pp. 521–528). IEEE.
- Borst, C., Fischer, M., & Hirzinger, G. (2004). Grasp planning: How to choose a suitable task wrench space. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004* (Vol. 1, pp. 319–325). IEEE.
- Bretl, T. (2006). Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. *The International Journal of Robotics Research*, 25(4), 317–342.
- Cao, C., Wan, W., Pan, J., & Harada, K. (2016). Analyzing the utility of a support pin in sequential robotic manipulation. In *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE.
- Chavan-Dafie, N., & Rodriguez, A. (2018). Regrasping by fixtureless fixturing. In: *2018 IEEE 14th international conference on automation science and engineering (CASE)* (pp. 122–129). IEEE.
- Chen, L., Figueiredo, L., & Dogar, M. (2018a). Planning for muscular and peripersonal-space comfort during human-robot forceful collaboration. In *Proceedings of Humanoids 2018*. IEEE.
- Chen, L., Figueiredo, L.F., & Dogar, M. (2018b). Manipulation planning under changing external forces. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3503–3510). IEEE.
- Cutkosky, M. R., & Kao, I. (1989). Computing and controlling compliance of a robotic hand. *IEEE Transactions on Robotics and Automation*, 5(2), 151–165.
- Dang, H., & Allen, P.K. (2012). Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1311–1317). IEEE.
- Diankov, R., & Kuffner, J. (2008). Openrave: A planning architecture for autonomous robotics. Tech Rep CMU-RI-TR-08-34 79 (Robotics Institute, Pittsburgh, PA).
- Dogar, M., Spielberg, A., Baker, S., & Rus, D. (2019). Multi-robot grasp planning for sequential assembly operations. *Autonomous Robots*, 43(3), 649–664.
- El-Khoury, S., De Souza, R., & Billard, A. (2015). On computing task-oriented grasps. *Robotics and Autonomous Systems*, 66, 145–158.
- Feige, U. (1998). A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4), 634–652.
- Ferrari, C., & Canny, J.F. (1992). Planning optimal grasps. In *IEEE international conference on robotics and automation, 1992. Proceedings* (Vol. 3, pp. 2290–2295). IEEE.
- Hagberg, A., Swart, P. & Chult, D. S. (2008). Exploring network structure, dynamics, and function using networkx. In *Tech. rep.*. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
- Han, L., Trinkle, J. C., & Li, Z. X. (2000). Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*, 16(6), 663–674.
- Haschke, R., Steil, J. J., Steuwer, I., & Ritter, H. J. (2005) Task-oriented quality measures for dexterous grasping. In: *CIRA* (pp. 689–694). Citeseer.
- Hauser K (2015) Lazy collision checking in asymptotically-optimal motion planning. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 2951–2957). IEEE.
- Hauser, K., & Latombe, J. C. (2010). Multi-modal motion planning in non-expansive spaces. *The International Journal of Robotics Research*, 29(7), 897–915.
- Jaillet, L., & Porta, J. M. (2013). Path planning under kinematic constraints by rapidly exploring manifolds. *IEEE Transactions on Robotics*, 29(1), 105–117.
- Knepper, R. A., Mavrogiannis, C. I., Proft, J., & Liang, C. (2017). Implicit communication in a joint action. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction* (pp. 283–292).
- Kosuge, K. & Kazamura, N. (1997) Control of a robot handling an object in cooperation with a human. In *Proceedings 6th IEEE international workshop on robot and human communication. RO-MAN'97 SENDAI*. IEEE.
- Kuffner Jr, J. J., & LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning. In *ICRA* (Vol. 2).
- Lee, G., Lozano-Pérez, T. & Kaelbling, L. P. (2015) Hierarchical planning for multi-contact non-prehensile manipulation. In *2015 IEEE/RSJ international conference on intelligent robots and systems*. IEEE.
- Lertkultanon, P., & Pham, Q. C. (2018). A certified-complete bimanual manipulation planner. *IEEE Transactions on Automation Science and Engineering*, 15(3), 1355–1368.

- Li, Z., & Sastry, S. S. (1988). Task-oriented optimal grasping by multi-fingered robot hands. *IEEE Journal on Robotics and Automation*, 4(1), 32–44.
- Lin, Y., & Sun, Y. (2015). Grasp planning to maximize task coverage. *The International Journal of Robotics Research*, 34(9), 1195–1210.
- Lipton, J.I., Manchester, Z., & Rus, D. (2017). Planning cuts for mobile robots with bladed tools. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE.
- Lipton, J.I., Schulz, A., Spielberg, A., Trueba, L.H., Matusik, W., & Rus, D. (2018). Robot assisted carpentry for mass customization. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 1–8). IEEE.
- Lozano-Pérez, T., Jones, J., Mazer, E., O'Donnell, P., Grimson, W., Tournassoud, P. & Lanusse, A. (1987) Handey: A robot system that recognizes, plans, and manipulates. In *Proceedings. 1987 IEEE international conference on robotics and automation* (Vol. 4, pp. 843–849). IEEE.
- Luo, R., Hayne, R., & Berenson, D. (2018). Unsupervised early prediction of human reaching for human-robot collaboration in shared workspaces. *Autonomous Robots*, 42(3), 631–648.
- Ma, J., Wan, W., Harada, K., Zhu, Q., & Liu, H. (2018). Regrasp planning using stable object poses supported by complex structure. *IEEE Transactions on Cognitive and Developmental Systems*, 11, 257–269.
- Maeda, G. J., Neumann, G., Ewertton, M., Lioutikov, R., Kroemer, O., & Peters, J. (2017). Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks. *Autonomous Robots*, 41(3), 593–612.
- Mainprice, J. & Berenson, D. (2013). Human-robot collaborative manipulation planning using early prediction of human motion. In *2013 IEEE/RSJ international conference on intelligent robots and systems* (pp. 299–306). IEEE.
- Miller, A. T., & Allen, P. K. (2004). Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4), 110–122.
- Mishra, B., Schwartz, J. T., & Sharir, M. (1987). On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(1–4), 541–558.
- Moriyama, R., Wan, W. & Harada, K. (2019). Dual-arm assembly planning considering gravitational constraints. arXiv preprint: [arXiv:1903.00646](https://arxiv.org/abs/1903.00646).
- Nikandrova, E., & Kyrki, V. (2015). Category-based task specific grasping. *Robotics and Autonomous Systems*, 70, 25–35.
- Rohrdanz, F., & Wahl, F. M. (1997). Generating and evaluating regrasp operations. In *Proceedings of international conference on robotics and automation* (Vol. 3). IEEE.
- Rozo, L., Calinon, S., Caldwell, D. G., Jimenez, P., & Torras, C. (2016). Learning physical collaborative robot behaviors from human demonstrations. *IEEE Transactions on Robotics*.
- Salisbury, J. K., & Roth, B. (1983). Kinematic and force analysis of articulated mechanical hands. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105(1), 35–41.
- Sánchez, G., & Latombe, J. C. (2003). A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In: *Robotics research* (pp. 403–417). Springer.
- Siméon, T., Laumond, J. P., Cortés, J., & Sahbani, A. (2004). Manipulation planning with probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7–8), 729–746.
- Sisbot, E. A., & Alami, R. (2012). A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5), 1045–1057.
- Slavík, P. (1996). A tight analysis of the greedy algorithm for set cover. In *Proceedings of the twenty-eighth annual ACM symposium on theory of computing* (pp. 435–441).
- Stoeter, S. A., Voss, S., Papanikopoulos, N. P., & Mosemann, H. (1999). Planning of regrasp operations. In *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)* (Vol 1, pp. 245–250). IEEE.
- Strabala, K. W., Lee, M. K., Dragan, A. D., Forlizzi, J. L., Srinivasa, S., Cakmak, M., et al. (2013). Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2(1), 112–132.
- Takase, K. (1974). The design of an articulated manipulator with torque control ability. In *Proc. 4th int. symp. on industrial robots*. Tokyo.
- Tournassoud, P., Lozano-Pérez, T. & Mazer, E. (1987). Regrasping. In *Proceedings. 1987 IEEE international conference on robotics and automation* (Vol. 4, pp. 1924–1928). IEEE.
- Uchiyama M. & Dauchez, P. (1988) A symmetric hybrid position/force control scheme for the coordination of two robots. In *Proceedings. 1988 IEEE international conference on robotics and automation* (pp. 350–356). IEEE.
- Uchiyama, M., & Dauchez, P. (1992). Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots. *Advanced Robotics*, 7, 361–383.
- Wan, W., & Harada, K. (2016). Integrated assembly and motion planning using regrasp graphs. *Robotics and Biomimetics*, 3(1), 1–11.
- Wan, W., & Harada, K. (2017). Regrasp planning using 10,000 s of grasps. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1929–1936). IEEE.
- Zheng, Y. F., & Luh, J. (1989). Optimal load distribution for two industrial robots handling a single object. *Journal of Dynamic Systems, Measurement, and Control*, 111(2), 232–237.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Lipeng Chen is now perusing his Ph.D. degree at the School of Computing in the University of Leeds, UK. Previously, he received the M.E. degree in control theory and control engineering from the Northeastern University, Shenyang, China in 2015. His research focuses on robotic manipulation and grasping, physical human-robot collaboration, etc.



Luis F. C. Figueiredo is a Marie-Sklodowska Curie Action research fellow at the University of Leeds. Previously, he was a post-doctoral researcher at Federal University of Minas Gerais, Brazil. He completed his Ph.D. in 2016 being awarded for the Best Ph.D. Thesis in Engineering at the University of Brasilia, Brazil. He also worked at MERS group at CSAIL-MIT in 2013–14 and received awards for robotic demonstrations at ICAPS-14 and IROS-14.



Mehmet R. Dogar is a University Academic Fellow at the School of Computing at the University of Leeds, where he leads the Robotic Manipulation Lab. Previously, he was a postdoctoral researcher at CSAIL, MIT. He received his Ph.D. from the Robotics Institute at Carnegie Mellon University. His research focuses on autonomous robotic manipulation. He envisions a future where robots autonomously perform complex manipulation tasks in human environments.