Planning for Multi-stage Forceful Manipulation

Planning for Multi-stage Forceful Manipulation

Rachel Holladay[1], Tomás Lozano-Pérez[1], and Alberto Rodriguez[2]

Abstract— **Multi-stage forceful manipulation tasks, such as twisting a nut on a bolt, require reasoning over interlocking constraints over discrete and continuous choices. The robot must choose a sequence of discrete actions, or strategy, such as whether to pick up an object, and the continuous parameters of each of those actions, such as how to grasp that object. In forceful manipulation tasks, the force requirements substantially impact the choices of both strategy and parameters. To enable planning and executing forceful manipulation, we augment an existing task and motion planner with controllers that exert wrenches and constraints that explicitly consider torque and frictional limits. In two domains, opening a childproof bottle and twisting a nut, we demonstrate how the system considers a combinatorial number of strategies and how choosing actions that are robust to parameter variations impacts the choice of strategy. https://mcube.mit.edu/forceful-manipulation/**

I. Introduction

Our goal is to enable robots to plan and execute *forceful manipulation tasks* such as drilling through a board, cutting a carrot, and twisting a nut. While all tasks that involve contact are technically forceful, we refer to forceful manipulation tasks as those where the ability to generate and transmit the necessary forces to objects and their environment is an active limiting factor which must be considered. Respecting these limits might require a planner, for example, to prefer a more forceful kinematic configuration of the robot arm or a more stable grasp of an object.

Forceful operations, as defined by Chen et al., are the exertion of a wrench (generalized force/torque) at a point on an object [1]. These operations are intended to be quasi-statically stable, i.e. the forces are always in balance and produce relatively slow motions, and will generally require some form of fixturing to balance the applied wrenches. For example, opening a push-and-twist childproof bottle is an example of a forceful manipulation task. To remove the lid the robot must exert a downward force on the lid while applying a torque along the axis of the bottle (Fig.1-bottom left). The robot must be strong enough to deliver the required wrench and also secure the bottle before exerting a wrench on it, to insure that there is enough available friction between the fingers and the lid.

To accomplish complex, multi-step forceful manipulation tasks, robots need to make discrete decisions, such as whether to push on the lid with the fingers, the palm or a tool, and whether to secure the bottle via frictional contact with a surface, with another gripper or with a vise. The robot must also make continuous decisions such as the choice of grasp pose, initial robot configuration and object poses in the environment. Critically, all these decisions interact in relatively complex ways to achieve a valid task execution.

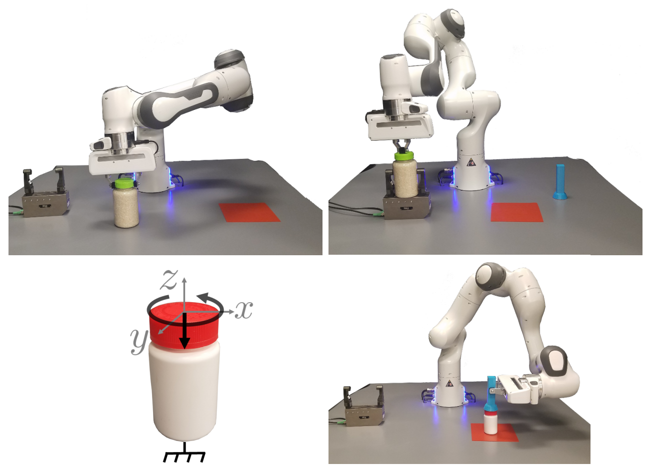


Fig. 1: Opening a childproof bottle involves executing a downward-push on the lid, while securing the bottle (lower left). Our system can reason over a combinatorial number of strategies to accomplish this forceful manipulation task, including twisting with various parts of its end effector, twisting with a tool (in blue), securing with a vise (in grey), securing against the table, or securing against a high-friction rubber mat (in red).

Fig.1 illustrates that there are *different strategies* for completing this task. Each strategy's viability depends on the robot's choices and on the environment. For example, a solution that uses the friction from the table to secure the bottle, as shown in the top left, would fail if the table can only provide a small amount of friction. Instead, the robot would need to find a significantly different strategy, such as securing, or fixturing, the bottle via a vise, as shown in the top right.

Strategies represent sequences of parameterized high-level actions. Each action is implemented as a controller parameterized by a set of constrained continuous values, such as robot configurations, grasp poses, trajectories, etc. Our goal is to find both a sequence of high-level actions (a strategy) and parameter values for those actions, all of which satisfy the coupled constraints.

To produce valid solutions for a wide range of object and environment configurations, the robot must be able to consider a wide range of strategies. As illustrated above, small changes, such as decreasing a friction coefficient, may necessitate an entirely new strategy. Approaches that attempt to explicitly encode solutions in the form of a policy, *e.g.* via a finite state machine or a fixed action sequence, will generally fail to capture the full range of feasible strategies [2], [3]. Methods that attempt to learn such a policy will need a very large number of interactions to explore this rich and highly-constrained solution space.

[1]Rachel Holladay and Tomás Lozano-Pérez are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, {rhollada,tlp@mit}.edu.

[2]Alberto Rodriguez is with the Mechanical Engineering Department, Massachusetts Institute of Technology, albertor@mit.edu.

We propose addressing forceful manipulation problems by planning over a combinatorial set of discete/continuous strategies. We extend an existing task and motion planning (TAMP) system, PDDLStream, which reduces this type of hybrid discrete/continuous planning problem to a sequence of discrete planning problems via focused sampling of the continuous parameters [4]. To apply this method to forceful manipulation, we introduce new actions with controllers that exert wrenches and constraints that explicitly consider torque and frictional limits. In particular, the constraints propagate the wrench of the forceful operations throughout the actions in a strategy. In the childproof bottle example, this propagation is critical to finding a strategy that fixtures the bottle against the wrench applied on the lid.

Our paper makes the following contributions:

- Augment an existing TAMP method [4] for forceful manipulation tasks by adding controllers and constraints.
- Demonstrate, in two domains (opening a childproof bottle and twisting a nut on a bolt), the interplay of the force-based constraints and the geometric and discrete choices.
- Enable the planner to choose strategies that are robust by formulating this as cost-sensitive planning. We demonstrate how accounting for robustness impacts the robot's strategy.

II. Background and Related Work

Our goal of finding a discrete sequence of actions parameterized by continuous values lies at the heart of multi-modal motion planning (MMMP) [5], [6] and task and motion planning (TAMP) [7]. MMMP plans motions that follow modes, e.g. moving through free space, and motions that switch between discrete modes, e.g. grasping, where each mode is a submanifold of configuration space. TAMP extends MMMP by incorporating non-geometric state variables and a structured action representation that supports efficient search [8], [9], [10]. Most, although not all [11], TAMP algorithms have focused on collision and kinematic constraints; this paper focuses on integrating force-based reasoning with an existing TAMP method [4].

Most similar to our work, Toussaint et al. formulates force-centric constraints that integrate into a path-optimization framework (LGP) for manipulation tasks [11]. While LGP can search over strategies, in this paper the strategy was provided and fixed. While Toussaint et al. take a more generic approach to representing interaction, their use of 3D point-of-attack (POA) to represent 6D wrenches prevents the system from considering patch contact, which is critical to our tasks. Levihn and Stilmann present a specific planner that reasons over which combination of objects in the environment will yield the appropriate mechanical advantage for unjamming a door [12]. The type of the door directly specifies which strategy to use (lever or battering ram) and the planner considers the interdependencies of force-based and geometric-based decisions for each application.

Michelman and Allen formalize opening a childproof bottle via a finite state machine, where the overall strategy and some of the continuous parameters, such as the grasp, are fixed [2]. Holladay et al. consider force- and motion-based constraints in planning a fixed sequence of actions to enable tool use [3]. While these systems reason over geometric and force constraints, these constraints do not impact the sequence of actions, i.e. the choice of strategy.

Several recent works have characterized types of force-based motions. Gao et al. defined "force-relevant skills" as a desired position and velocity in task space, along with an interaction wrench and task constraint [13]. Mahschitz et al. termed "sequential forceful interaction tasks" as those characterized by point-to-point movements and an interaction where the robot must actively apply a wrench [14]. Chen et al. define "forceful operations" as a 6D wrench f applied at a pose p with respect to a target object [1]. In this paper, we adopt their definition of forceful operations to characterize the type of interactions our system plans for. Chen et al. focus on finding environmental and robot contacts to stabilize an object while a human applies the forceful operation. In our work, a robot must both stabilize the object and apply the forceful operation.

Stabilizing, or fixturing, an object is a common requirement of any system forcefully operating on those objects. The goal of fixturing is to fully constrain an object or part, while enabling it to be accessible [15]. Fixture planning often relies on a combination of geometric, force and friction analyses [16]. There are various methods of fixturing including using clamps [17], using another robot to directly grasp or grasp via tongs [18], [19], [20], or using the environment and relying on friction, such as in fixtureless fixturing or shared grasping [21], [22]. Within this paper we consider fixturing via grasping and environmental contacts.

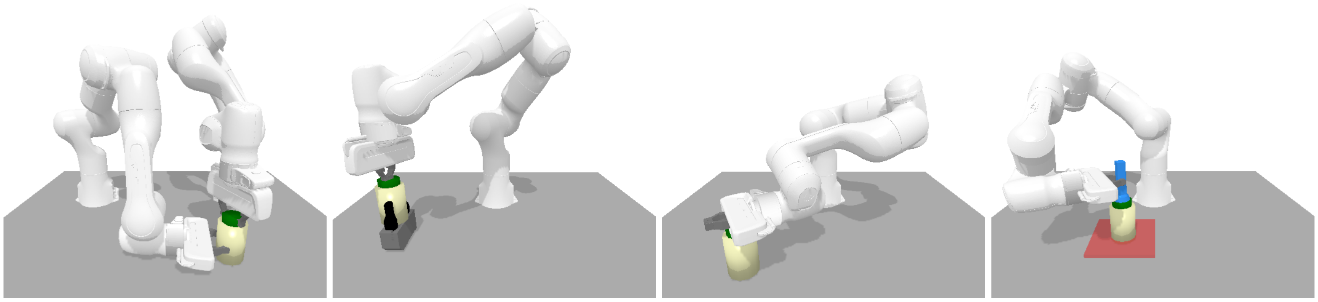
III. Forceful Manipulation

In this section, we characterize the class of problems considered in this paper and we introduce two domains that illustrate the class.

A. Problem Definition

We are interested in solving forceful manipulation tasks via a sequence of high-level actions, i.e. a strategy. Each action is implemented by a parameterized controller and is associated with constraints relating the continuous parameter values, such as robot configurations, poses, paths, etc. that must be satisfied for the controller to achieve its desired effect. The choice of a strategy requires finding satisfying values for all of these constraints.

Borrowing from Chen et al., we are interested in tasks that involve *forceful operations*, where the robot exerts a 6D wrench ($[f_x, f_y, f_z, t_x, t_y, t_z]$) on an object [1]. Actions that perform forceful operations (1) require the robot to use a controller that exerts force and (2) plan for the robotic system to be able to exert that force. We view the robotic system, composed of the robot joints, grasps and other possible frictional contacts, as a *forceful kinematic chain*. To exert a wrench, the forceful kinematic chain must be maintained, i.e. each joint must be stable under the imparted wrench. We informally use the word stable to refer to stable equilibrium of the forces and torques at all the "joints". Likewise, if



(a) GT + RF (b) FT + VF (c) PT + SF(T) (d) TT + SF(M)

Fig. 2: To open a childproof bottle, the robot must forcefully push and twist the lid, while fixturing the bottle. We define four strategies for twisting the lid: grasping on the lid (GT), pushing through the fingertips (FT) or the palm (PT) or grasping a pusher-tool that contacts the lid (TT). The system may fixture the bottle via another robot's grasp (RF), a vise's grasp (VF) or by securing it between the table (SF(T)) or a high-friction rubber mat, shown in red, (SF(M)).

we are exerting force upon an object, we need to fixture that object, hence creating a second forceful kinematic chain. In Sec. IV we explain how to incorporate stability into the planner as a constraint and then describe several types of joints in forceful kinematic chains and their mathematical models.

In this paper we assume a quasi-static physics model and, as input, are given geometric models of the robot, the objects and the environment along with the poses of each object. Physical parameters, such as the object's mass and center of mass, and friction coefficients, are known.

B. Example Domains

To ground our work in concrete problems, we consider two example domains: (1) opening a childproof bottle and (2) twisting a nut on a bolt (Fig.2 and Fig.4 respectively). For each domain, we define the forceful operation that represents the task, what object(s) must be fixtured, and a set of possible actions for imparting that forceful operation and fixturing the objects. In both domains we also include generic actions such as: move, move while holding, pick, and place.

1) Childproof Bottle Opening: In the first domain, the objective is to open a push-and-twist childproof bottle, as introduced in Sec. I. We specify the push-twisting, required before removing the lid, as the forceful operation of applying wrench $(0, 0, -f_z, 0, 0, t_z)$ in the frame of the lid (Fig.1-bottom left), where we assume f_z and t_z are given. While performing this action, the robot must fixture the bottle to prevent its motion. We define four push-twisting actions and three fixturing methods.

Each of the four push-twisting actions, illustrated in Fig.2, vary in how the robot contacts the lid and thus the construction of the forceful kinematic chain: grasping on the lid, pushing through fingertips or the palm or grasping a pusher-tool that contacts the lid. We refer to these push-twisting strategies as GT (grasp-twist), FT (finger-twist), PT (palm-twist) and TT (tool-twist). For the last three actions, the robot can reason over applying additional downward force, since this can safely increase the stability of the action.

There are three methods to fixture the bottle, as shown in Fig.2: grasping it with another robot, grasping it with a vise (here implemented via a rigidly mounted robot hand)

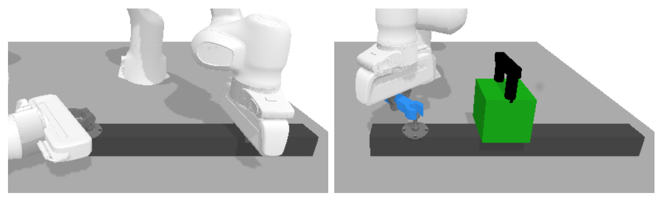


Fig. 4: To twist a nut on the bolt, the robot can use either its fingers or a spanner (in blue). While twisting, the robot must fixture the beam that the bolt is attached to. The robot can fixture either via another robot's grasp or by weighing down the beam with a large mass (in green).

or exerting additional downward force to secure the bottle with friction. For the last method, the frictional surface can either be the table, or a high-friction rubber mat. We refer to these fixturing strategies as RF (robot-fixture), VF (vise-fixture), SF(T) (surface-fixture using the table) and SF(M) (surface-fixture using the mat).

2) Nut Twisting: In the second domain, the robot twists a nut on a bolt by applying the wrench $(0, 0, 0, 0, 0, t_z)$ in the frame of the nut (Fig.3), with t_z given as input, while fixturing the beam holding the bolt. We do not consider the more general task of twisting a nut *until* it is tight. This version of the task, which would require monitoring and re-planning, is discussed as future work in Sec. VI

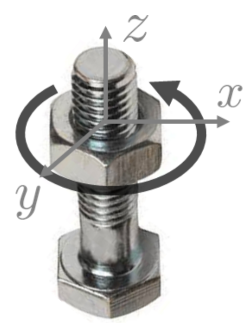


Fig. 3: In the nut twisting domain, the forceful operation is to apply a torque in z, in the frame of the nut.

The robot can either twist the nut, executing a controller that exerts force, with its fingers or by using a spanner (Fig.4). The beam can be fixtured either by having a second robot securely grasp it or by placing a heavy block on the beam to weigh it down.

In each domain, the variety of push-twisting (or twisting) and fixturing methods gives rise to a combinatorial set of possible strategies to search over. For example, one such strategy for the childproof bottle domain that uses the grasp-twist (GT) and robot-fixture (RF) methods, as partly visualized in Fig.2a, is: move(robot$_0$, path$_0$), pick(robot$_0$, bottle, grasp$_0$), move(robot$_1$, path$_1$), grasp-twist(robot$_1$, bottle, path$_2$, wrench), move(robot$_1$, path$_3$), pick(robot$_1$, cap, grasp$_0$), where some of the parameters of the actions have been omitted for brevity. The goal of the planner, described in Sec. IV, is to both search through the set of feasible strategies (action sequences) and solve for the parameters of each action, all while respecting the geometric and force-based constraints.

IV. Force-Based TAMP

We next detail in Sec. IV-A how we introduce force-based reasoning into an existing TAMP method, including adding a new type of variable, new controllers and new constraint functions, and how we extend it to generate robust strategies via cost-sensitive planning. We then discuss in Sec. IV-B the controller for exerting wrenches and the mathematical models used to verify forceful kinematic chains.

A. Planning Approach

To solve for both the strategy and the parameters of the strategy in a forceful manipulation problem, we use PDDLStream, a task and motion planning algorithm [4] that has been used in a variety of robotics domains, including pick-and-place in observed and partially-observed setting [23].

1) Extending PDDLStream: PDDLStream solves hybrid (discrete/continuous) problems by sampling, in a focused fashion, the continuous parameters and generating and solving a sequence of discrete planning problems until a solution to the original hybrid problem is found. The focused sampling is achieved by combining a small set of samplers that are conditional on the output of other samplers, such as a sampler for kinematic solutions conditioned on sampled poses of objects.

We define each of our example domains using the PDDLStream planning language, an extension of the PDDL language [24], by specifying variable types (e.g. poses), predicates (e.g. HandEmpty) and actions (e.g. pick) that are relevant to the domain. Each action is a parameterized controller and is characterized by preconditions on the state, or requirements for executing that controller, and its resulting effects to the state. Given this specification, along with an initial state and goal state, PDDLStream finds a sequence of actions and their parameters.

In order to leverage PDDLStream for forceful manipulation we add a wrench type, define actions that exert wrenches and introduce force-based constraints. First, we add `(Wrench ?w)` as a state variable type, where `?w` is a 6D wrench, specified with respect to a reference frame, which is usually attached to an object. This addition enables wrenches to be directly and easily incorporated into the specification.

Actions and their controllers which exert the forceful operations, such as the push-twist actions in the childproof lid domain, are parameterized, in part, by this wrench variable and we can characterize an effect of the action as accomplishing the forceful operation. Additionally, a precondition for executing these wrench-exerting actions, as mentioned in Sec. III-A, is that the forceful kinematic chain is maintained and that the objects the robot is acting upon are fixtured. This constraint propagates the planned exerted wrench through the joints of the relevant forceful kinematic chain and is deemed satisifed if all the joints (robot and contacts) are stable with respect to the application of that propagated wrench.

For example, when generating a grasp, the grasp sampler transforms the exerted wrench into the grasp frame and only considers the grasp feasible if it is stable. Sec. IV-B.1 discusses the controller that exerts wrenches and Sec. IV-B.2 details the forceful kinematic chain constraint.

2) Robust Planning: Given the ability to generate plans that satisfy geometric- and force-based constraints, we now aim to produce *robust* plans. We search for plans that maximize the probability of succeeding during open-loop execution. We focus on protecting against stability-based failures along the forceful kinematic chains due to uncertainty in physical parameters. For example, we want to discourage the system from selecting a grasp where a small change in the friction coefficient would break the stability of the grasp.

We formulate this as cost-sensitive planning, where the cost of an action is given as $-\log(\Pr[\text{success(action)}])$. Minimizing this cost is equivalent to maximizing the plan success likelihood. We define the probability of action success by sampling 100 sets of parameter values, each with a random epsilon perturbation, and evaluating our models (detailed in Sec. IV-B.2) to assess the stability of the forceful kinematic chain. We perturb, when applicable, parameters such as the friction coefficient, the planned applied wrench, the contact frame, the effective size of the contact patches, etc. By reasoning over perturbations in the parameters of the problem, we assess the robustness of the forceful kinematic chain.

B. Constraints on Forceful Operations

We next discuss the controller used to exert wrenches and the mechanics models used to asses the stability of forceful kinematic chains. While we overview a few types of joints, and their corresponding models, alternative joints or models could easily be integrated.

1) Exerting Force: Forceful interaction involves exerting a wrench at a pose. While there are several control methods for this explicitly, such as force control [25] or hybrid position-force control [26], [27], we opt to use a Cartesian impedance controller. Cartesian impedance control treats the interplay of interaction forces and motion deviations as a mass-spring-damper system [28]. We can exert forces and torques by offsetting the target Cartesian pose and adjusting the impedance parameters [29]. We chose to only vary the stiffness parameter, K_p, and set the damping parameter, K_d, to be critically damped, i.e. $K_d = 2\sqrt{K_p}$.

To have a degree of open-loop control over the magnitude of the exerted wrench, we characterized experimentally the relation between wrench magnitude, pose offset and stiffness. During planning, we use this experimental relation to select, given the stiffest possible setting, the desired pose offset.

2) Maintaining Forceful Kinematic Chain: With the actions that exert force, the planner must ensure that each joint in the forceful kinematic chain is stable. For our two domains we consider several classes of joints for which we model the space of wrenches they can withstand.

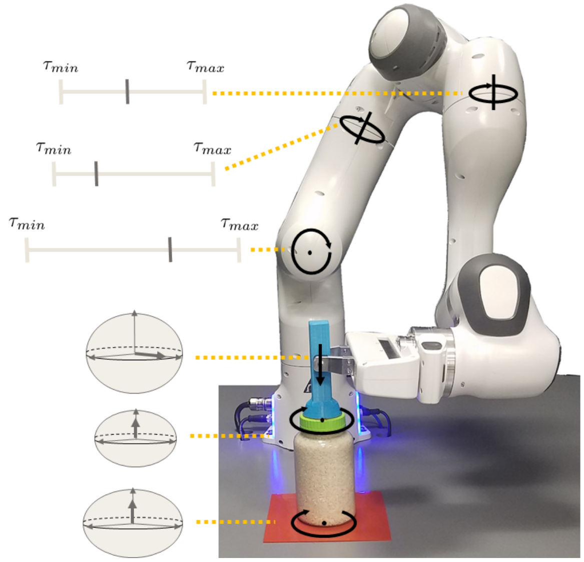


Fig. 5: Along each joint of the forceful kinematic chain, we first project the expected wrench (in black) into the subspace defined by each joint and then verify if the joint is stable under that wrench. Each type of joint connects to a mathematical model that visualizes this space. For circular patch contacts, we check the friction force against a limit surface ellipsoidal model and for each robot joint we check against the 1D torque limits.

For frictional planar joints, the friction wrenches are 3D, and we represent the boundary of the set of frictional wrenches in the three dimensional friction subspace of the plane of contact with a limit surface [30]. We use two ways to approximate the limit surface, depending on the characteristics of the planar joint (see a) and b) below). In the other three dimensions, we assume any wrenches are resisted kinematically by non-pentration reaction forces. For example, in Fig.5, the joint between the gripper and tool is maintained along three axes by friction and in the other three axes by the gripping force. For the robot's joints, the set of wrenches that can be transmitted are directly bound via the joint torque limits.

a) Limit Surface for Small Circular Patch Contacts: For small circular patch contacts with uniform pressure distributions (e.g. the fingers-bottle, bottle-surface contacts), we leverage the ellipsoidal approximation of the limit surface [31]. The ellipsoid is centered in the contact frame, $w = [f_x, f_z, m_y]$, and, for isotropic friction, is defined by $w^T A w = 1$ where:

$$A = \begin{bmatrix} \frac{1}{(N\mu)^2} & & 0 \\ & \frac{1}{(N\mu)^2} & \\ 0 & & \frac{1}{(Nk\mu)^2} \end{bmatrix}$$

such that μ is the friction coefficient, N is the normal force and, given our assumptions, $k \approx 0.6r$ where r is the radius of the contact [31], [32]. Having propagated the exerted wrench into the contact frame, we check if this wrench lies within the ellipsoid, which would indicate a stable contact:

$$\frac{f_x^2}{(N\mu)^2} + \frac{f_z^2}{(N\mu)^2} + \frac{m_y^2}{(Nk\mu)^2} < 1 \qquad (1)$$

b) Limit Surface for More General Patch Contacts: For contacts with more general shapes and less uniform pressure distributions, we directly model the contact patch as a set of point contact, each with its own normal force (localized pressure) and its own friction limits. Given a contact patch we model the force it can transmit as the convex hull of generalized friction cones placed at the corners of the patch. Generalized friction cones, based on the Coulomb friction model, represent the frictional wrench that a point contact can offer [33]. We represent the friction cone, FC, at each point contact with a polyhedral approximation of generators:

$$FC = \{(\mu, 0, 1), (-\mu, 0, 1), (0, \mu, 1), (0, -\mu, 1)\} \qquad (2)$$

for a friction coefficient, μ [34]. These generators can be scaled by the applied normal force. Given this approximation, the generalized friction cone can be written as:

$$V = \{v = J_f^T F \mid F \in FC\} \qquad (3)$$

where J_f^T is the Jacobian that maps contact forces f from the contact frame, where FC is defined, to the object frame. If the exerted wrench, in the reference frame of the patch contact, lies within the convex hull of V, the frictional wrench can resist the exerted wrench and the contact is stable.

In the nut-twisting domain (Fig.4), we use the generalized friction cone to model the contact between the table and the beam holding the bolt, placing friction cones at the four corners of the beam. In evaluating the stability of fixturing via a heavy weight, the applied normal force, determined by the mass and location of the weight, is modeled as a simply supported 1D beam with a partially distributed uniform load.

c) Torque Limits: The last type of joint we consider are the joints of the robot, where the limit of each joint is expressed via its torque limits. We relate the wrenches at the end effector to robot joint torques through the manipulator Jacobian, J_m. Specifically, given a joint configuration q and wrench w, the torque τ experienced at the joints is modeled by $\tau = J_m^T(q)w$. The forceful kinematic chain is stable if the expected vector of torques τ does not exceed the robot's torque limits τ_{lim}:

$$J_m^T(q) w_{ext} < \tau_{lim}. \qquad (4)$$

In summary, we have three models, each of which captures the spaces of wrenches that a particular joint can resist. The frictional contact models create ellipsoids and polytopes in wrench space. The torque limit creates a n-d box in torque space, where n is the number of joints, which can be thought of as a series of 1D bounds for each joint. Fig.5 visualizes how these varying models are used to asses the stability of the forceful kinematic chains.

V. Empirical Evaluation

We provide a few illustrative examples of how the system reasons over strategies. We also further demonstrate the system in the supplemental video [35].

A. Breadth of Solutions

We conduct two ablation studies in the childproof bottle domain (Table I). In, the first study, A1, we search over all possible fixturing strategies, with the twisting strategy

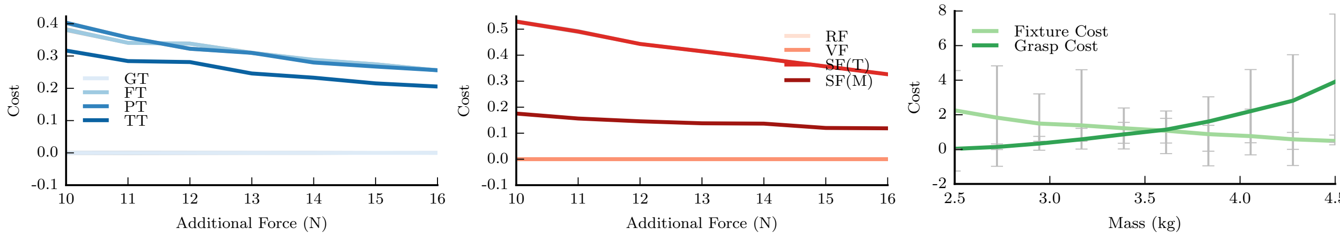


Fig. 6: Robustness Evaluation. In the childproof bottle domain, we evaluate the cost of each twisting method (left) and fixture method (center) across varying values of downward force. For each point, we average one hundred samples, randomizing over the grasp (when applicable). The standard error is plotted, but minuscule. From our cost definition, a higher cost corresponds to a less robust action. In the nut-twisting domain (right) we consider the trade-off in the grasp cost versus the fixturing cost. At each weigh value, we randomly sample the pose of the weight along the beam and the grasp on the weight. Since, at the extremes, some costs evaluate to infinity, we plot the median and a 95% confidence interval.

Ablation # 1		
Strategy	# Steps	Planning Time (SE)
GT+SF(T)*	4	177 (51)
GT+RF	6	60 (9.4)
GT+SF(M)**	8	142 (73)
GT+VF**	9	95 (35)

Ablation # 2		
Strategy	# Steps	Planning Time (SE)
GT+SF(M)	4	37 (1.8)
PT+SF(M)	4	25 (3.1)
FT+SF(M)*	4	63 (35)
TT+SF(M)**	8	40 (5.1)

TABLE I: For each ablation study, we provide the number of steps for each strategy and the average planning time in seconds (and standard error) over five runs. *: Utilized a higher friction coefficient μ to increase feasibility **: Invalidated shorter strategies to force to planner to find these longer strategies.

fixed (GT, grasp-twist). In A1, the bottle starts at a random location on the table. In the second study, A2, we search over all possible twisting strategies, starting the bottle on the rubber mat to use the surface fixturing strategy (SF(M), surface-fixturing on the mat).

Because the underlying search over strategies within PDDLStream biases towards plans with the fewest actions, we incrementally invalidated the shorter strategies in order to force exploration of the alternative, longer strategies. For example, in A1 we invalidated fixturing via the table or a robot grasp (GT+SF(T) and GT+RF) as feasible strategies by decreasing the friction coefficient between the bottle and the table and by removing the second arm, respectively. Accounting for this, the planner chose the longer, feasible strategies of fixturing with either the vise (GT+VF) or with the rubber mat (GT+SF(M)).

B. Robustness of Solutions

We next consider how accounting for robustness impacts the choice of strategy. In Fig.6-left we evaluate each of the push-twisting methods in the childproof bottle domain, across varying magnitudes of additional downward force, a parameter the planner must choose. Except for the grasp twist (GT), all of the methods are fairly susceptible to perturbations in their parameters. Each of these methods decreases in cost, and therefore increases in robustness, with additional downward force. We repeat this analysis for the fixturing methods in the same domain (Fig.6-center), where the robot and vise fixturing methods (RF, VF) both have a cost of zero. In comparing fixturing with the table (SF(T)) and the mat (SF(M)), the mat's higher friction coefficient makes it more robust. The robust planner is incentivized to avoid shorter but more brittle plans, such as fixturing with the table, and to opt for longer, more reliable plans.

In the nut-twisting domain we explore robustness by considering a scenario where the robot must chose between several weights, of varying mass, to fixture the beam with. For a given mass, we sample 100 placement locations along the beam holding the bolt and evaluate two robustness metrics: how robustly the weight fixtures the beam and how robustly the robot is able to grasp (and therefore move) the weight to this placement. Fig.6-right shows the trade-off: a heavier weight more easily fixtures the beam but is harder to robustly grasp. In finding a robust plan, and hence a low cost plan, the planner is incentivized to act like Goldilocks and pick the weight that best balances this trade-off.

VI. Discussion

Our goal is to solve complex multi-step manipulation problems that involve forceful operations between the robot and the environment, where forceful operations are defined as the robot applying a wrench at a pose. We leverage an existing task and motion planning system, augmenting it to reason over controllers that exert that wrench, maintain the forceful kinematic chain, and fixture the object we are acting on. We demonstrate our system in two example domains: opening a push-and-twist childproof bottle and twisting a nut on a bolt. We also demonstrate the use of cost-sensitive planning to prioritize actions that are robust to perturbations in the parameters of our stability metrics.

The current system has several limitations, notably that the the compliant controllers have fixed compliance, the computational cost is high, and the planner generates a fixed sequence of actions rather than a policy, e.g. for turning a nut until it is tight. These are fertile ground for future work.

Acknowledgments

This work was supported by the NSF Graduate Research Fellowship. We thank the members of the MCube Lab and LIS Group for their helpful feedback and especially Caelan Garrett for the many PDDLStream discussions.

References

[1] L. Chen, L. F. C. Figueredo, and M. Dogar, "Manipulation Planning Using Environmental Contacts to Keep Objects Stable under External Forces," in *Humanoids*, pp. 417–424, IEEE, 2019.

[2] P. Michelman and P. Allen, "Forming complex dextrous manipulations from task primitives," in *ICRA*, pp. 3383–3388, IEEE, 1994.

[3] R. Holladay, T. Lozano-Pérez, and A. Rodriguez, "Force-and-Motion Constrained Planning for Tool Use," in *IROS*, pp. 7409–7416, IEEE, 2019.

[4] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *ICAPS*, vol. 30, pp. 440–448, 2020.

[5] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *IJRR*, vol. 29, no. 7, pp. 897–915, 2010.

[6] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task," *IJRR*, vol. 30, no. 6, pp. 678–698, 2011.

[7] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated Task and Motion Planning," *Annual review of control, robotics, and autonomous systems*, 2020.

[8] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Sample-Based Methods for Factored Task and Motion Planning," in *RSS*, 2017.

[9] L. Pack Kaelbling and T. Lozano-Perez, "Hierarchical task and motion planning in the now," in *ICRA*, pp. 1470–1477, IEEE, 2011.

[10] M. Toussaint, "Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning," in *IJCAI*, pp. 1930–1936, 2015.

[11] M. Toussaint, J.-S. Ha, and D. Driess, "Describing Physics For Physical Reasoning: Force-based Sequential Manipulation Planning," in *IROS*, 2020.

[12] M. Levihn and M. Stilman, "Using Environment Objects as Tools: Unconventional Door Opening," in *IROS*, pp. 2502–2508, IEEE, 2014.

[13] X. Gao, J. Ling, X. Xiao, and M. Li, "Learning Force-Relevant Skills from Human Demonstration," *Complexity*, 2019.

[14] S. Manschitz, M. Gienger, J. Kober, and J. Peters, "Learning Sequential Force Interaction Skills," *Robotics*, vol. 9, no. 2, p. 45, 2020.

[15] H. Asada and A. By, "Kinematics of Workpart Fixturing," in *ICRA*, vol. 2, pp. 337–345, IEEE, 1985.

[16] S. Hong Lee and M. Cutkosky, "Fixture planning with friction," *Journal of Engineering for Industry*, vol. 113, no. 3, pp. 320–327, 1991.

[17] I. Mitsioni, Y. Karayiannidis, J. Stork, and D. Kragic, "Data-Driven Model Predictive Control for the Contact-Rich Task of Food Cutting," in *Humanoids*, pp. 244–250, IEEE, 2019.

[18] Y. Watanabe, K. Nagahama, K. Yamazaki, K. Okada, and M. Inaba, "Cooking Behavior with Handling General Cooking Tools based on a System Integration for a Life-sized Humanoid Robot," *Journal of Behavioral Robotics Paladyn*, vol. 4, no. 2, pp. 63–72, 2013.

[19] J. Stuckler, M. Schwarz, and S. Behnke, "Mobile Manipulation, Tool Use, and Intuitive Interaction for cognitive Service Robot cosero," *Frontiers in Robotics and AI*, vol. 3, p. 58, 2016.

[20] K. Zhang, M. Sharma, M. Veloso, and O. Kroemer, "Leveraging Multimodal Haptic Sensory Data for Robust Cutting," in *Humanoids*, pp. 409–416, IEEE, 2019.

[21] N. Chavan Dafle and A. Rodriguez, "Regrasping by Fixtureless Fixturing," in *CASE*, pp. 122–129, 2018.

[22] Y. Hou, Z. Jia, and M. Mason, "Manipulation with Shared Grasping," in *RSS*, 2020.

[23] C. R. Garrett, C. Paxton, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox, "Online replanning in belief space for partially observable task and motion problems," in *ICRA*, pp. 5678–5684, IEEE, 2020.

[24] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "PDDL-the planning domain definition language," 1998.

[25] G. Zeng and A. Hemami, "An overview of robot force control," *Robotica*, vol. 15, no. 5, pp. 473–482, 1997.

[26] M. T. Mason, "Compliance and force control for computer controlled manipulators," *Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[27] Y. Hou and M. T. Mason, "Robust execution of contact-rich motion plans by hybrid force-velocity control," in *ICRA*, pp. 1933–1939, IEEE, 2019.

[28] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I - Theory," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 1–7, 1985.

[29] I. Kresse and M. Beetz, "Movement aware Action Control - Integrating Symbolic and Control-theoretic Action Execution," in *ICRA*, pp. 3245–3251, IEEE, 2012.

[30] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction part 1. limit surface and moment function," *Wear*, vol. 143, no. 2, pp. 307–330, 1991.

[31] N. Xydas and I. Kao, "Modeling of contact mechanics and friction limit surfaces for soft fingers in robotics, with experimental results," *IJRR*, vol. 18, no. 9, pp. 941–950, 1999.

[32] J. Shi, J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch, "Dynamic in-hand sliding manipulation," *Transactions on Robotics*, vol. 33, no. 4, pp. 778–795, 2017.

[33] M. Erdmann, "On a representation of friction in configuration space," *IJRR*, vol. 13, no. 3, pp. 240–271, 1994.

[34] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.

[35] M. Lab, "Forceful manipulation webpage," *URL: https://mcube.mit.edu/forceful-manipulation/*, 2021.