

Motion and Force Planning for Manipulating Heavy Objects by Pivoting

Amin Fakhari, Aditya Patankar, and Nilanjan Chakraborty

Abstract—Manipulation of objects by exploiting their contact with the environment can enhance both the dexterity and payload capability of robotic manipulators. A common way to manipulate heavy objects beyond the payload capability of a robot is to use a sequence of pivoting motions, wherein, an object is moved while some contact points between the object and a support surface are kept fixed. The goal of this paper is to develop an algorithmic approach for automated plan generation for object manipulation with a sequence of pivoting motions. A plan for manipulating a heavy object consists of a sequence of joint angles of the manipulator, the corresponding object poses, as well as the joint torques required to move the object. The constraint of maintaining object contact with the ground during manipulation results in nonlinear constraints in the configuration space of the robot, which is challenging for motion planning algorithms. Exploiting the fact that pivoting motion corresponds to movements in a subgroup of the group of rigid body motions, $SE(3)$, we present a novel task-space based planning approach for computing a motion plan for both the manipulator and the object while satisfying contact constraints. We also combine our motion planning algorithm with a grasping force synthesis algorithm to ensure that friction constraints at the contacts and actuator torque constraints are satisfied. We present simulation results with a dual-armed Baxter robot to demonstrate our approach.

I. INTRODUCTION

Manipulation of heavy and bulky objects is a challenging task for manipulators and humanoid robots. An object is considered heavy if the manipulator's joint torques are not large enough to balance the object weight while lifting it off the ground. Thus, heavy objects cannot be manipulated with usual pick-and-place strategy due to actuator saturation. Consider the manipulation scenario shown in Fig. 1, where a heavy object has to be moved from an initial pose C_O to a final pose C_F by a dual-armed robot. The object has to negotiate a step during the manipulation which implies that the final pose cannot be achieved by either pick-and-place strategies or by pushing. One possible way to move the object and negotiate the step is to use a sequence of pivoting motions, which we call object gaiting, and this is a common strategy used by humans to manipulate heavy objects. Therefore, the goal of this paper is to develop an algorithmic approach to compute a plan for manipulating heavy objects by a sequence of pivoting motions.

In a pivoting motion, we move the object while creating a sequence of point or line contacts with the environment (see Figure 1). A point contact acts like a spherical joint, whereas a line contact acts like a revolute joint. The location and axes

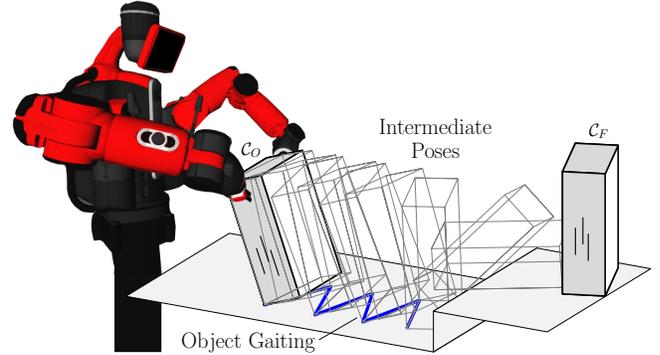


Fig. 1. Schematic sketch of dual-handed manipulation of a heavy object between two given poses C_O and C_F by a sequence of pivoting motions.

of these joints change during a gaiting motion. These joints are force-closed joints and can only be implemented through adequate frictional force at the object-ground contact that prevents slippage. Because of the making and breaking of the contacts the equations of motion of the object change and technically the system of equations form a hybrid dynamical system. Planning of motion through intermittent contact considering the switching dynamics is difficult and computationally costly, in general, although some attempts have been made in this direction (e.g., [1]). To reduce the complexity and cost of planning, one approach is to decouple the overall planning problem and first compute a kinematic motion plan and then compute force inputs that follow the kinematic motion plan and satisfy the dynamics constraints. The challenge in the decoupled approach is to ensure that the kinematic plans are dynamically feasible.

In this paper, we take the decoupled approach to motion planning for pivoting. Thus, a plan for pivoting operations consists of (a) *Motion plan*: a sequence of joint angles of the manipulators and the corresponding object poses that maintains contact with the ground and ensures that the contacts between the hand and the object are maintained (b) *Force plan*: a sequence of joint torques that are within the actuator limits and ensure that there is enough force at the object ground contact to prevent slippage. Furthermore, to ensure that the kinematic plan is dynamically feasible, we also want to ensure that the manipulator does not lose the grasp of the object and there is no slippage at the hand-object contact. In this paper, we will focus on the motion planning problem. We have studied the problem of computing the force plan (or force synthesis problem), for a given motion plan, in [2], and we will combine it with our motion plan to generate torques to achieve the motion.

The key challenge in solving the motion planning problem is that the kinematic constraints of the object maintaining a spherical or a revolute joint with the ground during the motion corresponds to nonlinear manifold constraints in the joint space of the manipulator. For example, in the scenario shown in Fig. 1, the configuration space is 20 dimensional and since the position kinematics equations are nonlinear, the pivoting constraints will form nonlinear manifolds in this 20 dimensional space. In sampling-based motion planning in joint space (\mathbb{J} -space), these constraints are hard to deal with, although there have been some efforts in this direction [3], [4], [5], [6], [7], [8], [9]. Furthermore, in manipulation by gaiting, where we are performing a sequence of pivoting operations, these manifold constraints are not known beforehand since they depend on the choice of the pivot points (or lines) which has to be computed as a part of the plan. In this paper, we present a novel task-space (\mathbb{T} -space) based approach for generating the motion plan that exploits the fact that the kinematic constraints of a revolute or spherical joint constrains the motion of the object to a subgroup of $SE(3)$.

Contributions: We present a two-step approach for computing the motion plan. In the first step, we develop an algorithm to compute a sequence of intermediate poses for the object to go from the initial to the goal pose. Two consecutive intermediate poses implicitly determine a point or line on the object and the ground that stay fixed during motion, thus encoding motion about a revolute or a spherical joint. In the second step, we use Screw Linear Interpolation (ScLERP) to determine a task space path between two intermediate poses, along with resolved motion rate control (RMRC) [10], [11] to convert the task space path to a joint space path. The advantage of using ScLERP is that it (a) automatically satisfies the kinematic motion constraints that the contact points between the object and the ground do not change during the pivoting motion without explicitly encoding it (b) the object does not penetrate the ground (c) the relative pose between the fingers (or manipulators) at the grasping points do not change, i.e., the closed chain constraints formed by the manipulator and object is maintained throughout the whole motion without explicitly encoding it. Thus, the joint space path that we compute along with the object path automatically ensures that the kinematic contact constraints are satisfied and the object remains grasped. This computationally efficient approach for motion planning for manipulation by pivoting, where apparently complicated constraints can be satisfied without explicitly modeling them, is the key contribution of this paper. Assuming that the inertial force are negligible, i.e., the motion is quasi-static, we also show that our motion plan can be combined with the second order cone programming (SOCP) based approach to compute joint torques and grasping forces [2], while ensuring that all no-slip constraints at the contacts and actuator limits are satisfied. Thus, we can ensure that the kinematic plan is feasible when one considers the forces and actuator limits. We demonstrate our approach in simulation using a dual-armed Baxter robot.

II. RELATED WORK

The use of external environment contacts to enhance the in-hand manipulation capability was first studied by Chavan-Daffe in [12]. More recently Hou *et. al* [13] have developed a planning algorithm for quasi-static reorientation of 3D objects on a table using a parallel-jaw gripper by having 3D mesh model of the objects. However, they are using a gripper with a special mechanism to allow pivoting at the gripper-object contact and compliant rolling at the object-table contact. In another research, they have referred to the use of environment contact as *shared grasping* wherein they treat the environment as an additional finger [14]. They have provided stability analysis of shared grasping by using *Hybrid Force-Velocity Control* (HFVC). Murooka *et. al.* [15] proposed a method for pushing a heavy object by an arbitrary region of a humanoid robot. Polverini *et. al.* [16] also developed a control architecture for a humanoid robot which is able to exploit the complexity of the environment to perform the pushing task of a heavy object. Pivoting was first introduced by Aiyama *et. al.* [17] as a new method of grasplless/non-prehensile manipulation. Based on this method, Yoshida *et. al.* [18], [19], [20] developed a whole-body motion planner for a humanoid robot to autonomously plan a pivoting strategy for manipulating bulky objects. They first planned a sequence of collision-free Reeds and Shepp paths (especially straight and circular paths in \mathbb{R}^2), then convert these paths into a sequence of pivoting motions. However, this method is limited to the motion on Reeds and Shepp curves to satisfy a nonholonomic constraint, which is not always required. Thus, it is not a general, efficient, and optimum way to manipulate objects between two given poses, especially when there are no obstacles in the workspace. Hence, we propose a general gait planning method as an optimization problem by defining the *intermediate poses* and using the ScLERP to manipulate the object by gaiting between any two arbitrary poses. Although there are task-space based force control algorithms [21], [22], they do not consider planning through the hybrid dynamics created by the intermittent contact. Furthermore [21] approximates the second order friction cone as a polyhedral cone, which can lead to infeasible solutions even when feasible plans exist.

III. PRELIMINARIES

Quaternions and Rotations: The quaternions are the set of hypercomplex numbers, \mathbb{H} . A quaternion $Q \in \mathbb{H}$ can be represented as a 4-tuple $Q = (q_0, \mathbf{q}_r) = (q_0, q_1, q_2, q_3)$, $q_0 \in \mathbb{R}$ is the real scalar part, $\mathbf{q}_r = (q_1, q_2, q_3) \in \mathbb{R}^3$ corresponds to the imaginary part. The conjugate, norm, and inverse of a quaternion Q is given by $Q^* = (q_0, -\mathbf{q}_r)$, $\|Q\| = \sqrt{QQ^*} = \sqrt{Q^*Q}$, and $Q^{-1} = Q^*/\|Q\|^2$, respectively. Addition and multiplication of two quaternions $P = (p_0, \mathbf{p}_r)$ and $Q = (q_0, \mathbf{q}_r)$ are performed as $P+Q = (p_0+q_0, \mathbf{p}_r+\mathbf{q}_r)$ and $PQ = (p_0q_0 - \mathbf{p}_r \cdot \mathbf{q}_r, p_0\mathbf{q}_r + q_0\mathbf{p}_r + \mathbf{p}_r \times \mathbf{q}_r)$. The quaternion Q is a *unit quaternion* if $\|Q\| = 1$, and consequently, $Q^{-1} = Q^*$. Unit quaternions are used to represent the set of all rigid body rotations, $SO(3)$, the Special

Orthogonal group of dimension 3. Mathematically, $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^T \mathbf{R} = \mathbf{R} \mathbf{R}^T = \mathbf{I}_3, |\mathbf{R}| = 1\}$, where \mathbf{I}_3 is a 3×3 identity matrix and $|\cdot|$ is the determinant operator. The unit quaternion corresponding to a rotation is $Q_R = (\cos \frac{\theta}{2}, \mathbf{l} \sin \frac{\theta}{2})$, where $\theta \in [0, \pi]$ is the angle of rotation about a unit axis $\mathbf{l} \in \mathbb{R}^3$.

Dual Quaternions and Rigid Displacements: In general, dual numbers are defined as $d = a + \epsilon b$ where a and b are elements of an algebraic field, and ϵ is a *dual unit* with $\epsilon^2 = 0, \epsilon \neq 0$. Similarly, a dual quaternion D is defined as $D = P + \epsilon Q$ where $P, Q \in \mathbb{H}$. The conjugate, norm, and inverse of the dual quaternion D is represented as $D^* = P^* + \epsilon Q^*$, $\|D\| = \sqrt{DD^*} = \sqrt{PP^* + \epsilon(PQ^* + QP^*)}$, and $D^{-1} = D^* / \|D\|^2$, respectively. Another definition for the conjugate of D is represented as $D^\dagger = P^* - \epsilon Q^*$. Addition and multiplication of two dual quaternions $D_1 = P_1 + \epsilon Q_1$ and $D_2 = P_2 + \epsilon Q_2$ are performed as $D_1 + D_2 = (P_1 + P_2) + \epsilon(Q_1 + Q_2)$ and $D_1 D_2 = (P_1 P_2) + \epsilon(P_1 Q_2 + Q_1 P_2)$. The dual quaternion D is a *unit dual quaternion* if $\|D\| = 1$, i.e., $\|P\| = 1$ and $PQ^* + QP^* = 0$, and consequently, $D^{-1} = D^*$. Unit dual quaternions can be used to represent the group of rigid body displacements, $SE(3) = \mathbb{R}^3 \times SO(3)$, $SE(3) = \{(\mathbf{R}, \mathbf{p}) \mid \mathbf{R} \in SO(3), \mathbf{p} \in \mathbb{R}^3\}$. An element $T \in SE(3)$, which is a pose of the rigid body, can also be expressed by a 4×4 homogeneous transformation matrix as $T = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$ where $\mathbf{0}$ is a 1×3 zero vector. A rigid body displacement (or transformation) is represented by a unit dual quaternion $D_T = Q_R + \frac{\epsilon}{2} Q_p Q_R$ where Q_R is the unit quaternion corresponding to rotation and $Q_p = (0, \mathbf{p}) \in \mathbb{H}$ corresponds to the translation.

Screw Displacement: Chasles-Mozzi theorem states that the general Euclidean displacement/motion of a rigid body from the origin \mathbf{I} to $T = (\mathbf{R}, \mathbf{p}) \in SE(3)$ can be expressed as a rotation θ about a fixed axis \mathcal{S} , called the *screw axis*, and a translation d along that axis (see Fig. 2). Plücker coordinates can be used to represent the screw axis by \mathbf{l} and \mathbf{m} , where $\mathbf{l} \in \mathbb{R}^3$ is a unit vector that represents the direction of the screw axis \mathcal{S} , $\mathbf{m} = \mathbf{r} \times \mathbf{l}$, and $\mathbf{r} \in \mathbb{R}^3$ is an arbitrary point on the axis. Thus, the screw parameters are defined as $\mathbf{l}, \mathbf{m}, \theta, d$. The screw displacements can be expressed by the dual quaternions as $D_T = Q_R + \frac{\epsilon}{2} Q_p Q_R = (\cos \frac{\Phi}{2}, L \sin \frac{\Phi}{2})$ where $\Phi = \theta + \epsilon d$ is a dual number and $L = \mathbf{l} + \epsilon \mathbf{m}$ is a dual vector. A power of the dual quaternion D_T is then defined as $D_T^\tau = (\cos \frac{\tau\Phi}{2}, L \sin \frac{\tau\Phi}{2})$, $\tau > 0$.

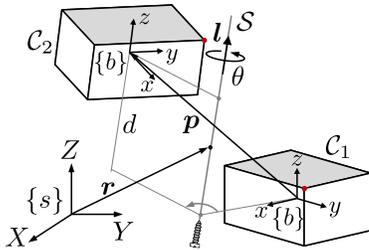


Fig. 2. Screw displacement from pose C_1 to pose C_2 .

Screw Linear Interpolation (ScLERP): To perform a one

degree-of-freedom smooth screw motion (with a constant rotation and translation rate) between two object poses in $SE(3)$, the screw linear interpolation (ScLERP) can be used. The ScLERP provides a *straight line* in $SE(3)$ which is the closest path between two given poses in $SE(3)$. If the poses are represented by unit dual quaternions D_1 and D_2 , the path provided by the ScLERP is derived by $D(\tau) = D_1(D_1^{-1}D_2)^\tau$ where $\tau \in [0, 1]$ is a scalar path parameter. As τ increases from 0 to 1, the object moves between two poses along the path $D(\tau)$ by the rotation $\tau\theta$ and translation τd . Let $D_{12} = D_1^{-1}D_2$. To compute D_{12}^τ , the screw coordinates $\mathbf{l}, \mathbf{m}, \theta, d$ are first extracted from $D_{12} = P + \epsilon Q = (p_0, \mathbf{p}_r) + \epsilon(q_0, \mathbf{q}_r) = (\cos \frac{\theta}{2}, \mathbf{l} \sin \frac{\theta}{2}) + \epsilon Q$ by $\mathbf{l} = \mathbf{p}_r / \|\mathbf{p}_r\|$, $\theta = 2 \operatorname{atan2}(\|\mathbf{p}_r\|, p_0)$, $\mathbf{d} = \mathbf{p} \cdot \mathbf{l}$, and $\mathbf{m} = \frac{1}{2}(\mathbf{p} \times \mathbf{l} + (\mathbf{p} - d\mathbf{l}) \cot \frac{\theta}{2})$ where \mathbf{p} is derived from $2QP^* = (0, \mathbf{p})$ and $\operatorname{atan2}(\cdot)$ is the two-argument arctangent. Then, $D_{12}^\tau = (\cos \frac{\tau\Phi}{2}, L \sin \frac{\tau\Phi}{2})$ is directly derived from $(\cos \frac{\tau\theta}{2}, \sin \frac{\tau\theta}{2} \mathbf{l}) + \epsilon(-\frac{\tau d}{2} \sin \frac{\tau\theta}{2}, \frac{\tau d}{2} \cos \frac{\tau\theta}{2} \mathbf{l} + \sin \frac{\tau\theta}{2} \mathbf{m})$. Note that $\theta = 0, \pi$ corresponds to pure translation between two poses and the screw axis is at infinity.

IV. PROBLEM STATEMENT

Let us assume that we want to manipulate a heavy cuboid object quasi-statically by using n manipulators, while maintaining contact with environment, from an initial pose $C_O \in SE(3)$ to a final pose $C_F \in SE(3)$. We assume that both C_O and C_F are in the robot's workspace. Let $\Theta^i = [\theta_1^i, \theta_2^i, \dots, \theta_{l_i}^i] \in \mathbb{R}^{l_i}$ be the vector of joint angles of the i -th l_i -DoF manipulator, which represents the *joint space* (\mathbb{J} -space) or the *configuration space* (\mathbb{C} -space) of the manipulator. Moreover, $\mathcal{E}^i \in SE(3)$ is defined as the pose of the end-effector of the i -th manipulator where $\mathcal{E}^i = \mathcal{FK}(\Theta^i)$ and $\mathcal{FK}(\cdot)$ is the manipulator forward kinematics map. Therefore, $\Theta_O^i \in \mathbb{R}^{l_i}$ and $\mathcal{E}_O^i \in SE(3)$ represent the initial configuration of the i -th manipulator (in \mathbb{J} -space) and pose of i -th end-effector, respectively, corresponding to the object initial pose C_O and $\Theta_F^i \in \mathbb{R}^{l_i}$ and $\mathcal{E}_F^i \in SE(3)$ represent the final configuration of the i -th manipulator (in \mathbb{J} -space) and pose of i -th end-effector, respectively, corresponding to the object final pose C_F . We assume that the position of the manipulator-object contact c_i is given and the transformation between the frames $\{e_i\}$ and $\{c_i\}$ remains constant during the manipulation, i.e., there is no relative motion (slippage) at the contact interface.

Our motion planning problem is now defined as computing a sequence of joint angles $\Theta^i(j)$, where $j = 1, \dots, m$, $\Theta^i(1) = \Theta_O^i$, $\Theta^i(m) = \Theta_F^i$, to manipulate the object while maintaining contact with the environment from its initial pose C_O to a final pose C_F when $(C_O, \mathcal{E}_O^i, \Theta_O^i)$ and (C_F, \mathcal{E}_F^i) ($i = 1, \dots, n$) are given. Moreover, our force planning problem is computing the minimum contact wrenches required to be applied at c_i during the object manipulation to balance the external wrenches (e.g., gravity) and also the environment contact wrenches using the method we have presented in [2].

Solution Approach Overview: Generally speaking, to move an object while maintaining contact we can use two primitive motions, namely, (1) *sliding* on a vertex, edge, or

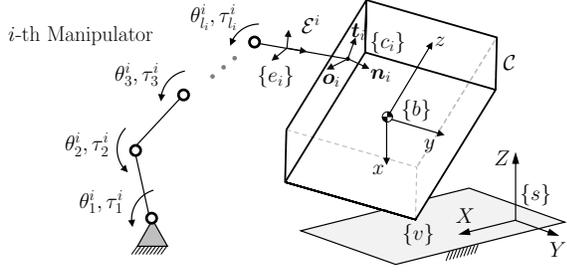


Fig. 3. An cuboid-shape object being tilted at one of its vertices.

face of the object in contact with the environment (Fig. 5-a) and (2) *pivoting* about an axis passing through a vertex, edge, or face of the object in contact with the environment (Fig. 5-b,c, Fig. 1). All other motions can be made by combining these primitive motions. Note that we consider *tumbling* as a special case of pivoting when the axis of rotation passes through an object edge or face. Manipulation by sliding (or pushing) can be useful in many scenarios like picking a penny off a table. However, in heavy and bulky object manipulation scenarios, sliding may not give feasible solutions. Thus, in this paper, we will focus on manipulation using the pivoting primitive.

Our *manipulation strategy* can be described briefly as follows. (i) Given the initial and final pose of the object, we first determine if multiple pivoting moves have to be made and, if necessary, compute intermediate poses of the object. (ii) Using the dual quaternion representation of these poses, we compute paths in $SE(3)$ using the ScLERP for the object and end-effectors. These paths automatically satisfies all the basic task related constraints (without any additional explicit representation of the constraints). (iii) We use the (weighted) pseudoinverse of the Jacobian to derive the joint angles in the \mathbb{J} -space from the computed \mathbb{T} -space path. (iv) Finally, we compute the minimum required contact wrenches and manipulators' joint torques required for object manipulation. Note that the steps (ii) to (iv) can be done either sequentially or they can be interleaved in a single discrete time-step

V. FORCE PLANNING

In this section, we briefly review our force planning algorithm proposed in [2]. Consider an object which is in contact with the environment at m contact positions and being manipulated quasi-statically by n l_i -DOF manipulators ($i = 1, \dots, n$) an n contact positions (Fig. 3). We define the grasping force optimization problem (GFOP) as a convex optimization problem as

$$\underset{\mathbf{F}_C, \mathbf{F}_E, \boldsymbol{\tau}}{\text{minimize}} \quad \max(\mathbf{F}_n) \quad (1a)$$

$$\text{subject to} \quad \mathbf{G}_C \mathbf{F}_C + \mathbf{G}_E \mathbf{F}_E + \mathbf{f}_{\text{ext}} = \mathbf{0}, \quad (1b)$$

$$\mathbf{F}_C \in \mathcal{K}_C, \quad (1c)$$

$$\mathbf{F}_E \in \mathcal{K}_E, \quad (1d)$$

$$\boldsymbol{\tau} + \mathbf{J}^T \mathbf{F}_C - \boldsymbol{\tau}_g = \mathbf{0}, \quad (1e)$$

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}. \quad (1f)$$

The objective function (1a) is the maximum normal force at the object-manipulator contacts, where $\mathbf{F}_n \in \mathbb{R}^n$ is the vector of normal contact forces. Equation (1b) is the equilibrium constraint where $\mathbf{G}_C \mathbf{F}_C \in \mathbb{R}^6$ and $\mathbf{G}_E \mathbf{F}_E \in \mathbb{R}^6$ are the total wrenches exerted by the manipulators and environment through the contacts, with respect to the body frame $\{b\}$, respectively, $\mathbf{F}_C \in \mathbb{R}^{6n}$ and $\mathbf{F}_E \in \mathbb{R}^{6m}$ are the vectors containing all the contact wrenches at the manipulators and environment contacts, respectively, \mathbf{G}_C is the grasp matrix and \mathbf{G}_E is the matrix that converts the environment contact wrench to the body frame, and $\mathbf{f}_{\text{ext}} \in \mathbb{R}^6$ is the total external wrench (including the object weight) applied to the object at the body frame $\{b\}$. (1c) and (1d) represent the friction cone constraints at all the manipulators and environment contacts, respectively, where \mathcal{K}_C and \mathcal{K}_E are the second-order cones (SOC). (1e) represents the manipulator joint torque constraints, where $\boldsymbol{\tau} \in \mathbb{R}^l$ is a vector containing the joint torques of all the manipulators and computing from (1e), $\boldsymbol{\tau}_g \in \mathbb{R}^l$ is the vector of joint torques due to gravity, and $\mathbf{J} = \text{diag}(\mathbf{J}_1, \dots, \mathbf{J}_n) \in \mathbb{R}^{6n \times l}$ is the overall Jacobian matrix of the manipulators ($l = \sum_{i=1}^n l_i$).

VI. PIVOTING

Pivoting is a motion where an object is moved while maintaining a point or line contact with a support surface. When an object maintains a point contact, the constraints on motions are same as those imposed by a spherical joint. Thus, the motion of the object is restricted to $SO(3)$, which is a subgroup of $SE(3)$, and the axis of rotation passes through the contact point. During pivoting with line contact (or tumbling), the constraint on the motion is same as that imposed by a revolute joint with the axis of the joint being the line of contact. Thus, in this case, the motion of the object is restricted to $SO(2)$, which is also a subgroup of $SE(3)$. This mathematical structure of pivoting motions is key to our ScLERP approach as we discuss below.

Suppose an object can reach a goal pose from a start pose using a single pivoting motion. This can happen when the start and the goal poses are such that there is a common vertex, say v , between the start and goal poses that lie on the support surface (Fig. 4, Fig. 5-c). To plan the motion of the object between the start and goal pose via interpolation such that the contact at the vertex v is maintained and the object does not penetrate the surface one should be careful about the interpolation scheme.

Two popular choices for interpolation between two given poses (Q_1, \mathbf{p}_1) and (Q_2, \mathbf{p}_2) (using unit quaternion parameterization of orientation) are (a) linear interpolation for both orientation and position, i.e., $Q(\tau) = [Q_1 + (Q_2 - Q_1)\tau] / \|Q_1 + (Q_2 - Q_1)\tau\|$, $\mathbf{p}(\tau) = \mathbf{p}_1 + (\mathbf{p}_2 - \mathbf{p}_1)\tau$ (Fig. 4-a), and (b) spherical linear interpolation (SLERP) for orientation and linear interpolation for position, i.e., $Q(\tau) = Q_1(Q_1^{-1}Q_2)^\tau$, $\mathbf{p}(\tau) = \mathbf{p}_1 + (\mathbf{p}_2 - \mathbf{p}_1)\tau$, $\tau \in [0, 1]$ (Fig. 4-b). If we use these linear interpolations between the end poses in the space of parameters, the object will penetrate the support surface, so the motion plan will not be feasible. The motion obtained will also change

with the choice of the coordinate frames for the initial and final pose. The advantage of using ScLERP is that it is coordinate invariant. Furthermore, since the pivoting motions also belongs to a subgroup of $SE(3)$, ScLERP ensures that all the intermediate poses will lie in the same subgroup that contains the initial and goal pose (i.e., all intermediate poses will have the vertex v fixed to the support surface). Thus, for motion planning, it is not necessary to explicitly enforce the pivoting constraint. Lemma 1 formalizes this discussion. Furthermore, ScLERP results in a motion along the shortest path, which in the pivoting scenario leads to a rotation about the axis through the pivot point(s) by the smallest angle. Thus, the non-penetration constraint of the object with the ground is also satisfied without explicitly enforcing it.

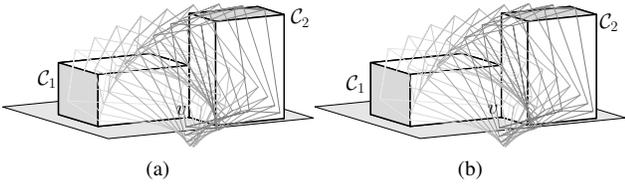


Fig. 4. Interpolation between two given poses C_1 and C_2 . (a) linear interpolation for both orientation and position, (b) spherical linear interpolation (SLERP) for orientation and linear interpolation for position.

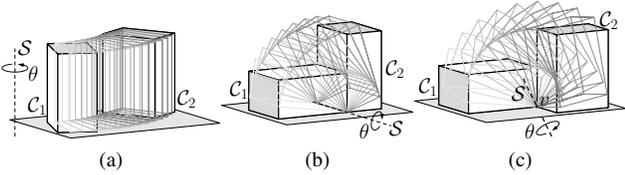


Fig. 5. Examples of the primitive motions for manipulating cuboid-shape objects by exploiting the environment contact, (a) sliding or pushing on a face, (b) pivoting about an edge (tumbling), (c) pivoting about a vertex.

Lemma 1: Let $D_1 = Q_{R1} + \frac{\epsilon}{2}Q_{p1}Q_{R1}$ and $D_2 = Q_{R2} + \frac{\epsilon}{2}Q_{p2}Q_{R2}$ be two unit dual quaternions representing two poses of a rigid body. If a point $v \in \mathbb{R}^3$ in the rigid body has the same position in both poses, the position of this point remains the same in all the poses provided by the ScLERP $D(\tau) = D_1(D_1^{-1}D_2)^\tau$ where $\tau \in [0, 1]$.

Proof: Let $Q_v = (0, v) \in \mathbb{H}$ be a pure quaternion representing the point v . Since the point v has the same position in both poses D_1 and D_2 , therefore

$$D_1(1 + \epsilon Q_v)D_1^\dagger = D_2(1 + \epsilon Q_v)D_2^\dagger, \quad (2)$$

$$\therefore Q_{p2} - Q_{p1} = Q_{R1}Q_vQ_{R1}^* - Q_{R2}Q_vQ_{R2}^*. \quad (3)$$

Therefore, the transformation from D_1 to D_2 is derived as

$$\begin{aligned} D_{12} &= D_1^*D_2 = Q_{R1}^*Q_{R2} + \frac{\epsilon}{2}Q_{R1}^*(Q_{p2} - Q_{p1})Q_{R2} \\ &= Q_{R1}^*Q_{R2} + \frac{\epsilon}{2}(Q_vQ_{R1}^*Q_{R2} - Q_{R1}^*Q_{R2}Q_v). \end{aligned} \quad (4)$$

By representing the rotation $Q_{R1}^*Q_{R2}$ as $(\cos \frac{\theta}{2}, \mathbf{l} \sin \frac{\theta}{2}) \in \mathbb{H}$ (where \mathbf{l} is a unit vector along the screw axis and θ is rotation

about the screw axis), (4) can be simplified as

$$D_{12} = (\cos \frac{\theta}{2}, \mathbf{l} \sin \frac{\theta}{2}) + \epsilon(0, \mathbf{v} \times \mathbf{l} \sin \frac{\theta}{2}) = P + \epsilon Q. \quad (5)$$

The translation d along the screw axis is determined by $d = \mathbf{p} \cdot \mathbf{l}$ where \mathbf{p} is derived from $2QP^* = (0, \mathbf{p})$. By using (5),

$$\mathbf{p} = \mathbf{v} \times \mathbf{l} \sin \frac{\theta}{2} \cos \frac{\theta}{2} - (\mathbf{v} \times \mathbf{l}) \times \mathbf{l} \sin^2 \frac{\theta}{2}, \quad (6)$$

and $d = \mathbf{p} \cdot \mathbf{l} = 0$. Therefore, the transformation $D(\tau)$ is a pure rotation about the fixed point \mathbf{v} on the screw axis. ■

Furthermore, when using multiple manipulators to pivot an object and assuming that there is no relative motion at the hand-object contact, the motion of each end-effector can be obtained independently by ScLERP using a shared interpolation parameter. This will ensure that the constraint that the relative end-effector poses of the manipulators are unchanged during motion is maintained without explicitly encoding it (this follows from Lemma 3 of [23] and so we do not repeat the formal statements and proofs here). In other words, it is guaranteed that the manipulators and the object always form a closed chain during the entire motion without explicitly encoding the constraint. In the next section, we use pivoting as a primitive motion for motion planning between any two given poses in \mathbb{T} -space.

VII. MOTION PLANNING IN TASK SPACE

To manipulate a polyhedral object between any two given poses C_O and C_F while maintaining contact with the environment, multiple pivoting moves can be combined by defining a set of appropriate *intermediate poses*. The set of the intermediate poses $C_I = \{C_I^1, C_I^2, \dots, C_I^h\}$ are defined in a way that the motion between any two successive poses $\{C_O, C_I, C_F\}$ can be represented by a single constant screw pivoting move. Thus, we can conveniently represent the motion between any two given object poses C_O and C_F in $SE(3)$ by using ScLERP to ensure that the object maintains its contact with the environment continuously. The object manipulation strategies on a flat surface can be categorized into 3 cases; **(Case I)** If C_O and C_F have a contact edge or vertex in common, the final pose can be achieved by pivoting the object about the common point or edge (Fig. 5-b,c). **(Case II)** If C_O and C_F do not have any edge or vertex in common but the same face of the object is in contact with the environment in both poses, different strategies can be considered. One of the strategies is using a sequence of pivoting motions about the object edges (tumbling). In this motion, the travel distance is discrete and depends on the object size and it may not be suitable for manipulating some objects like furniture. In this situation, we can manipulate the object is *object gaiting* (Fig. 8-a) which is defined as a sequence of pivoting motions on two adjacent object vertices in contact (see VII-A and VII-B). **(Case III)** If the adjacent or opposite faces of the object are in contact with the environment in both poses, a combination of pivoting and gaiting is required to achieve the final pose as shown in Fig. 6. Depending on the manipulators' physical limitations, object gaiting is more efficient only when a specific face of

the object is in contact with the environment. For instance, manipulation on the longer edge of the cuboid shown in Fig. 8-a may be more difficult than two other edges.

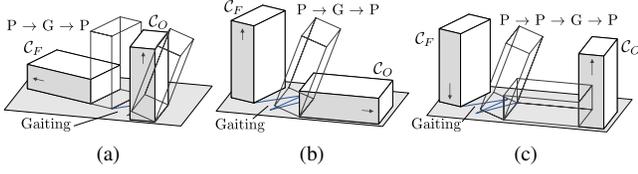


Fig. 6. Examples of the object manipulation with primitive motions when two adjacent (a,b) or opposite (c) object faces are in contact with the environment in initial and final poses (P: Pivoting, G: Gaitting).

A. Intermediate Poses in Object Gaitting

Let us assume that the axes of the body frame $\{b\}$ are parallel to the cuboid edges and the inertia frame $\{s\}$ is attached to the supporting plane such that Z -axis is perpendicular to the plane (Fig. 7). Three successive intermediate poses while pivoting about the vertex a are shown in Fig. 7-a,b. The object is initially in the pose $\mathcal{C}_I^1 = (R_1, p_1)$ (Fig. 7-a) holding on the contact edge ab . The pose $\mathcal{C}_I^2 = (R_2, p_2)$ (Fig. 7-a) is achieved by rotating the object by a small angle β along the edge passing through the vertex a ; therefore, $R_2 = R_1 R_x(-\beta)$ and only the vertex a is in the contact. Finally, the pose $\mathcal{C}_I^3 = (R_3, p_3)$ (Fig. 7-b) is determined by rotating \mathcal{C}_I^1 by an angle α along Z -axis about the vertex a ; therefore, $R_3 = R_Z(\alpha)R_1$ and the edge ab is again in contact with the environment. This procedure can be also repeated for the vertex b . By using ScLERP between these intermediate poses, we can obtain a smooth motion for object gaitting while maintaining contact with the environment. Note that the angles γ and β can be chosen arbitrarily and we do so in this paper. However, if there are secondary objectives like avoiding small obstacles on the ground while pivoting, γ and β can be chosen to satisfy those objectives.

B. Gait Planning

To manipulate the object from an initial pose \mathcal{C}_O to a final pose \mathcal{C}_F by object gaitting, a sequence of the rotation angle α between these two poses should be properly determined (Fig. 8-a). Let k be the number of required edge contacts and $\alpha = [\alpha_1, \dots, \alpha_k] \in \mathbb{R}^k$ be the angles between the contact edges as shown in Fig. 8-b. We can find α using an optimization problem as

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} && \|\alpha\| \\ & \text{subject to} && \mathbf{x} = \pm w \sum_{i=1}^k (-1)^i \begin{bmatrix} \cos(\alpha_O \pm \bar{\alpha}) \\ \sin(\alpha_O \pm \bar{\alpha}) \end{bmatrix}, \\ & && \alpha_F - \alpha_O = \pm \sum_{i=1}^k (-1)^i \alpha_i, \\ & && |\alpha_i| \leq \alpha_{\max}, \quad i = 1, \dots, k, \end{aligned} \quad (7)$$

where $\bar{\alpha} = \sum_{j=1}^i (-1)^j \alpha_j$, α_{\max} is the maximum allowed rotation angle, w is the length of the edge contact, and α_O and α_F represent the orientation of the contact edges $a_O b_O$

and $a_F b_F$ relative to X -axis, respectively. The negative sign correspond to the case that the first gait begins from the edge a_O , where $\mathbf{x} = \mathbf{b}_F - \mathbf{a}_O$ if k is an odd number and $\mathbf{x} = \mathbf{a}_F - \mathbf{a}_O$ if k is an even number, moreover, the positive sign correspond to the case that the first gait begins from the edge b_O , where $\mathbf{x} = \mathbf{a}_F - \mathbf{b}_O$ if k is an odd number and $\mathbf{x} = \mathbf{b}_F - \mathbf{b}_O$ if k is an even number. $\mathbf{a}_O, \mathbf{b}_O, \mathbf{a}_F, \mathbf{b}_F \in \mathbb{R}^2$ are the coordinates of the contact vertices in \mathcal{C}_O and \mathcal{C}_F poses along X - and Y -axis of the frame $\{s\}$. In the optimization problems (7), the first constraint represents the distance of the last contact vertex (a_F or b_F) relative to the first contact vertex (a_O or b_O) in X and Y directions. The second constraint represents the relative angle between the contact edges $a_O b_O$ and $a_F b_F$, and the last constraint considers the manipulators' limitations to rotate the object. In order to find the feasible minimum number of edge contacts, k , required to manipulate the object between two poses \mathcal{C}_O and \mathcal{C}_F , we need to repeat (7) for different values of k .

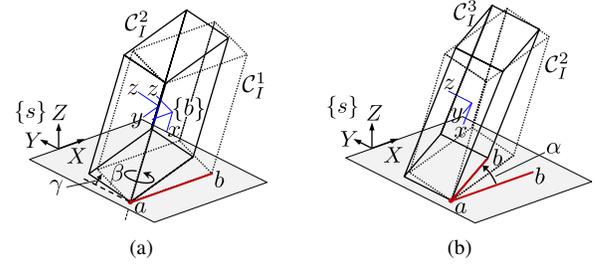


Fig. 7. Intermediate poses in object gaitting while pivoting.

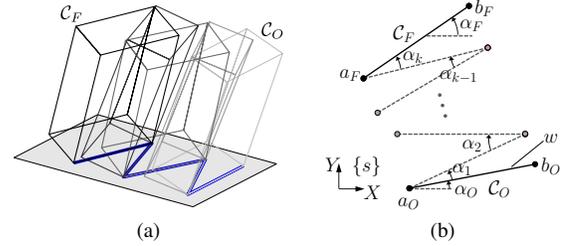


Fig. 8. A sequence of contact edges for object gaitting between two poses \mathcal{C}_O and \mathcal{C}_F when the first gait begins from the edge a_O .

VIII. MAPPING FROM \mathbb{T} -SPACE TO \mathbb{J} -SPACE

Since it is assumed that the transformation between the end-effector frame $\{e_i\}$ and contact frame $\{c_i\}$ remains constant, after planning a path in the \mathbb{T} -space, we can compute the end-effector poses \mathcal{E}_i for each object intermediate pose. Then, we use the ScLERP for each of these end-effector poses individually with a shared screw parameter. To find the joint angles of the manipulators in \mathbb{J} -space, we use the (weighted) pseudoinverse of the manipulators' Jacobian [24]. Let Θ_t and χ_t be the vector of joint angles and end-effector's pose at the step t , respectively. For each manipulator, given the current end effector pose χ_t and the target end effector pose χ_{t+1} (obtained from ScLERP) we

have the corresponding joint angles Θ_{t+1} as

$$\Theta_{t+1} = \Theta_t + \lambda \mathbf{J}(\Theta_t)(\chi_{t+1} - \chi_t), \quad (8)$$

where $0 < \lambda \leq 1$ is a step length parameter (see [23] for a complete algorithm). Here \mathbf{J} is the (weighted) pseudo-inverse of the manipulator Jacobian. By using (8) between any two successive poses in $\{\mathcal{C}_O, \mathcal{C}_I, \mathcal{C}_F\}$, $\Theta^i(j)$ ($j = 1, \dots, m$) for the i -th manipulator is computed. Note that we have just presented the simplest way of converting from task space to joint space. To avoid joint limits, we can augment Equation 8 with additional terms on the right hand side that belongs to the null space of the manipulator Jacobian.

IX. IMPLEMENTATION AND RESULTS

In this section, we briefly present the simulation results for manipulating a heavy cuboid object on a flat surface and over a step. Videos of our simulations are presented in the video attachment to the paper.

Manipulation on a Flat Surface: In this example, we plan motion to reorient a heavy object from an initial pose \mathcal{C}_O to a final pose \mathcal{C}_F , in its vicinity, by object gaiting as shown in Fig. 9-a. Existing planning algorithms [20] cannot efficiently solve this problem, because their motion plan is essentially restricted to move on Reeds and Shepp curves. By using the proposed optimization problem (7), we can find the minimum number of contact edges required to manipulate the object between these two poses. The simulation results for $a_O = [0, 0]$, $\alpha_O = 0^\circ$, $a_F = [0.13, 0.13]$ m, $\alpha_F = -80^\circ$, $w = 0.2$ m, $\alpha_{\max} = 35^\circ$ are shown in Fig. 9-b. As shown, at least 3 contact edges $\alpha = [-10.55^\circ, 29.56^\circ, -12.63^\circ, 27.25^\circ]$ (in total 10 intermediate poses) are required to reach the final pose by starting pivoting from the edge a .

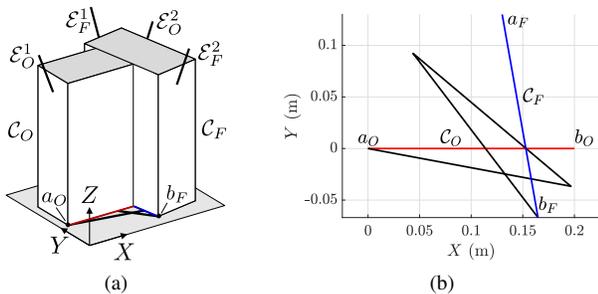


Fig. 9. Object gaiting on a flat surface.

Manipulation over a Step: In this example, we plan motion and force to manipulate a heavy object of uniform density over a step (Fig. 10) by both 7-DoF arms of the Baxter robot. The computed motion plan includes 3 stages: (1) pivoting about the object edge (\mathcal{C}_I^1), (2) pivoting about the vertex v (\mathcal{C}_I^2), where the object face and only the vertex v are in contact with the environment, (3) changing the location of the end-effectors' contacts and pivoting about the step edge (\mathcal{C}_F). Thus, there are two intermediate poses $\{\mathcal{C}_I^1, \mathcal{C}_I^2\}$. For all the 3 stages, we assume that the contact locations are known beforehand. We implemented \mathbb{T} -space planning, conversion to \mathbb{J} -space, and force planning to find

the minimum required normal forces $f_{c_{n,1}}$ and $f_{c_{n,2}}$ at both object-end-effector contacts $\{c_1\}$ and $\{c_2\}$ in each motion stage. The simulation parameters are given in Table I. Moreover, the friction coefficients at the manipulator contact is 0.3 and at the environment contact is 0.4. The optimization formulation has been implemented in MATLAB and solved using the CVX toolbox [25] with the default solver (SDPT3) on a PC with 1.8GHz processor and 16GB RAM. In Fig. 11, the variation of joint angles in the right and left arms of the Baxter robot in the 3 stages of manipulating the object is shown. Note that as mentioned above, in stage (3), we change the end-effector contact locations before pivoting the object about the step edge. This has been reflected in the ‘Stage 3’ of Fig. 11(a) and (b), as the joint angle values, for all the 7 joints of the Baxter robot, change abruptly as compared to the first two stages. In Fig. 12, the variations of the normal contact forces with respect to the number of iterations to reach the goal pose in the 3 stages of object manipulation over a step are shown. In stage 1, $f_{c_{n,1}}$ and $f_{c_{n,2}}$ first decrease and become negligible at a particular object tilting angle where the weight of the object passes through its support edge, and then increases. In stage 2, since the motion is not symmetric, there is a difference between the right and left end-effector normal contact forces in order to balance the object weight. In stage 3, the object-environment contact points are initially located closer to the object center of mass; thus, less contact forces are initially required and by pivoting the object, these forces increase.

TABLE I
SIMULATION PARAMETERS FOR MANIPULATION OVER A STEP.

Parameter	Value
Weight	20 (N)
Object & Step Dimensions	0.15 m \times 0.15 m \times 0.1 m, $H_s = 0.07$ (m)

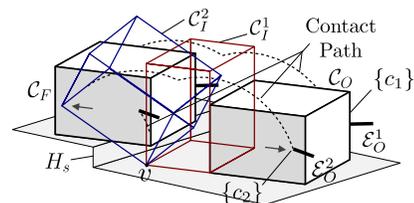
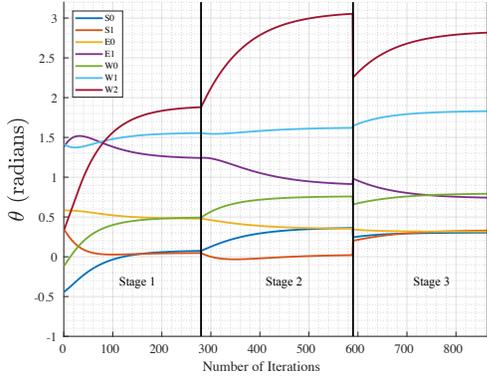


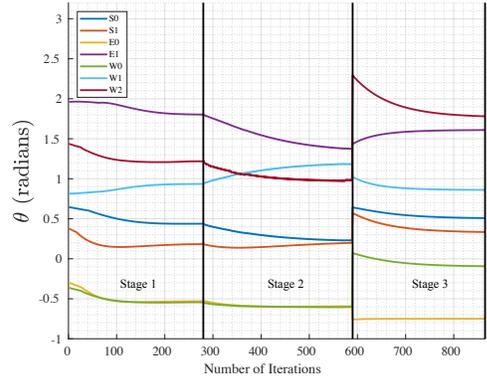
Fig. 10. Object manipulation over a step.

X. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel approach for manipulating heavy objects while using a sequence of pivoting motions. We have implemented our proposed motion and force planning on two different scenarios; reorienting an object using gaiting and also manipulating a heavy object over a step. Given the initial and final poses of the object, we first compute the required intermediate poses. These poses can be derived by an optimization problem which computes the optimal values of the rotation angles between contact edges while *object gaiting*. Then, by using ScLERP, we can

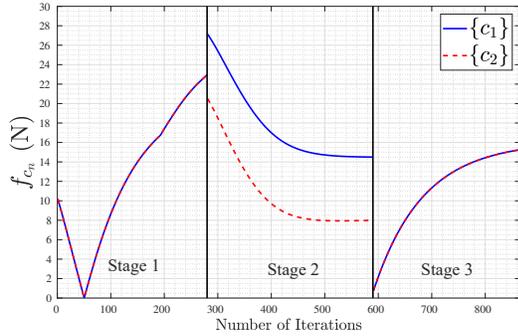


(a)



(b)

Fig. 11. Joint angle changes of the two arms, (a) right arm, (b) left arm.

Fig. 12. The normal contact forces at $\{c_1\}$ and $\{c_2\}$ where the object weight is $m = 2\text{kg}$, maximum joint torque for shoulder and elbow joints is $\tau_{\max} = 50\text{Nm}$, and maximum joint torque for wrist joints is $\tau_{\max} = 15\text{Nm}$. The time taken for one iteration of force and motion planning is 1.5 seconds.

interpolate between these intermediate poses while satisfying all the task-related constraints. Using RMRC we can map the task-space based plan to the joint-space allowing us to compute the contact forces and the joint torques required to manipulate the object. Future work includes the relaxation of the quasi-static assumption for the force planning and experimental evaluation of the proposed approach.

REFERENCES

[1] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal*

of *Robotics Research (IJRR)*, vol. 33, no. 1, pp. 69–81, 2014.

[2] A. Patankar, A. Fakhari, and N. Chakraborty, "Hand-object contact force synthesis for manipulating objects by exploiting environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[3] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.

[4] L. Jaillet and J. M. Porta, "Path planning under kinematic constraints by rapidly exploring manifolds," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 105–117, Feb 2013.

[5] M. Stilman, "Global manipulation planning in robot joint space with task constraints," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 576–584, June 2010.

[6] B. Kim, T. T. Um, C. Suh, and F. C. Park, "Tangent bundle rrt: A randomized algorithm for constrained motion planning," *Robotica*, vol. 34, no. 1, p. 202–225, 2016.

[7] Z. Yao and K. Gupta, "Path planning with general end-effector constraints," *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 316 – 327, 2007.

[8] M. Bonilla, E. Farnioli, L. Pallottino, and A. Bicchi, "Sample-based motion planning for soft robot manipulators under task constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[9] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *International Journal of Robotics Research*, vol. 38, no. 10-11, pp. 1151–1178, 2019.

[10] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on man-machine systems*, vol. 10, no. 2, pp. 47–53, 1969.

[11] D. L. Pieper, "The kinematics of manipulators under computer control," Ph.D. dissertation, Stanford University, 1968.

[12] N. C. Daffe, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1578–1585.

[13] Y. Hou, Z. Jia, and M. Mason, "Fast planning for 3d any-pose-reorienting using pivoting," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1631–1638.

[14] —, "Manipulation with Shared Grasping," in *Proceedings of Robotics: Science and Systems*, Corvallis, Oregon, USA, July 2020.

[15] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5682–5689.

[16] M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli, and N. G. Tsagarakis, "Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 859–866, 2020.

[17] Y. Aiyama, M. Inaba, and H. Inoue, "Pivoting: A new method of graspless manipulation of object by robot fingers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1993, pp. 136–143.

[18] E. Yoshida, M. Poirier, J.-P. Laumond, R. Alami, and K. Yokoi, "Pivoting based manipulation by humanoids: a controllability analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 1130–1135.

[19] E. Yoshida, M. Poirier, J.-P. Laumond, O. Kanoun, F. Lamiraux, R. Alami, and K. Yokoi, "Whole-body motion planning for pivoting based manipulation by humanoids," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 3181–3186.

[20] —, "Pivoting based manipulation by a humanoid robot," *Autonomous Robots*, vol. 28, no. 1, p. 77, 2010.

[21] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 64–77, 2018.

[22] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The international journal of robotics research*, vol. 14, no. 1, pp. 19–36, 1995.

[23] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

- [24] C. A. Klein and C. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 2, pp. 245–250, 1983.
- [25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.