

Choosing grasps to enable collision-free post-grasp manipulations

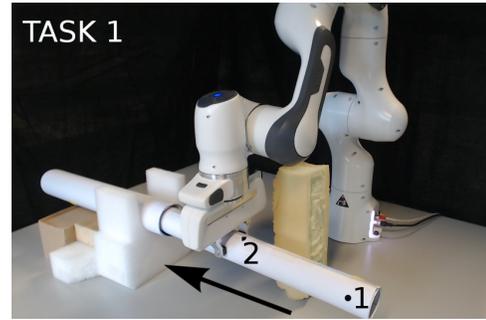
Tommaso Pardi, Rustom Stolkin and Amir M. Ghalamzan E.

Abstract—Consider the task of grasping the handle of a door, and then pushing it until the door opens. These two fundamental robotics problems (selecting secure grasps of a hand on an object, e.g. the door handle, and planning collision-free trajectories of a robot arm that will move that object along a desired path) have predominantly been studied separately from one another. Thus, much of the grasping literature overlooks the fundamental purpose of grasping objects, which is typically to make them move in desirable ways. Given a desired post-grasp trajectory of the object, different choices of grasp will often determine whether or not collision-free post-grasp motions of the arm can be found, which will deliver that trajectory. We address this problem by examining a number of possible stable grasping configurations on an object. For each stable grasp, we explore the motion space of the manipulator which would be needed for post-grasp motions, to deliver the object along the desired trajectory. A criterion, based on potential fields in the post-grasp motion space, is used to assign a collision-cost to each grasp. A grasping configuration is then selected which enables the desired post-grasp object motion while minimising the proximity of all robot parts to obstacles during motion. We demonstrate our method with peg-in-hole and pick-and-place experiments in cluttered scenes, using a Franka Panda robot. Our approach is effective in selecting appropriate grasps, which enable both stable grasp and also desired post-grasp movements without collisions. We also show that, when grasps are selected based on grasp stability alone, without consideration for desired post-grasp manipulations, the corresponding post-grasp movements of the manipulator may result in collisions.

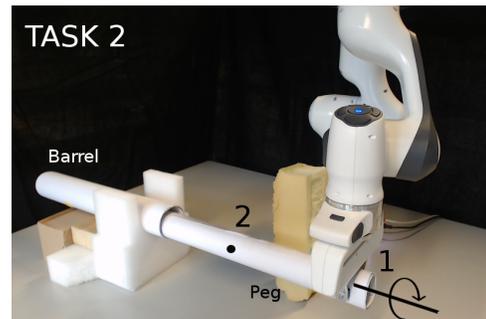
I. INTRODUCTION

The planning of manipulator actions, that are necessary for moving an object, have been predominantly been studied as two separate problems in isolation: (1) detecting end-effector poses which provide stable contacts between end-effector parts and the object [1]–[3], known variously as “grasp planning”, “grasp detection” or “grasp synthesis”; and (2) planning post-grasp manipulative (PGM) movements [4]. Given a set of stable grasp candidates, some approaches plan PGM movements in the presence of obstacles to deliver an object to a desired target pose [5]. However, such methods consider only the final destination pose as the task, and do not address the more complex task of moving an object along an entire desired trajectory or path. In contrast, it is often necessary to achieve specific trajectories of manipulated

All authors are with the Extreme Robotics Lab (ERL) at the University of Birmingham, UK. {txp754, R.Stolkin and A.Ghalamzanesfahani}@bham.ac.uk. Tommaso Pardi is supported by a doctoral bursary of the UK Nuclear Decommissioning Authority. Stolkin is supported by a Royal Society Industry Fellowship. This work forms part of the UK National Centre for Nuclear Robotics initiative, funded by EPSRC EP/R02572X/1. The work was also partially supported by H2020 RoMaNS 645582, EPSRC EP/P017487/1, EP/P01366X/1.



(a)



(b)

Fig. 1. The Panda is tasked with inserting and rotating the cylindrical peg into the cylindrical shell. The manipulator grasps the peg at 1 ((b)) which is suitable for rotating clockwise the peg $\frac{\pi}{6}$ [rad]. However, it is not appropriate for inserting peg 0.15 [m] into the barrel. By contrast, while the manipulator can insert the peg 0.15 [m] into the shell by grasping it at 2, it is not able to rotate the peg clockwise because of the collision.

objects, because: (1) object movements may be constrained, e.g. pushing a sliding door by its handle, or inserting an object into a slot; or (2) we may require the robot to execute motions that are learned from demonstrations, e.g. by dynamic movement primitives. Here, we propose an approach that computes a collision avoidance cost of the desired PGM movements, and we use this cost function to select the most useful choice from multiple possible stable grasp configurations.

Planning an intelligent grasping configuration and post-grasp movements in a cluttered environment include (i) G0 (reach-to-grasp): a manipulator approaches an object, i.e. reach a door handle in the presence of obstacles; (ii) G1 (grasp synthesis): the manipulator makes stable contacts on the object surface, called force-/form-closure grasping configuration, e.g. grab the door handle; and (iii) G2 (*post-grasp manipulator movements*): it moves the object and delivers desired object movements, e.g. opening a sliding

door.

Force-closure [6], or form-closure analysis [7] are classical approach to compute stable grasping configurations based on a 3-D model of an object. However, building 3-D model of all unknown objects is not easy and may not be practically feasible. State-of-the-art approaches efficiently compute stable grasping configurations (G1) from point clouds of a scene with a high success rate (e.g. using a probabilistic learning algorithm [2], artificial neural networks [1], [8], or efficient geometric matching between hand and object parts [3]). However, a manipulator may not be able to reach such grasps, because it collides with obstacles during G0. Other approaches therefore consider jointly planning G0 (planning the reach-to-grasp) and G1 (grasp synthesis). Berenson [5] stores a set of off-line computed grasping configurations in a database and uses them in an online collision-free G0 search. GraspRRT [4] simultaneously finds a feasible grasping configuration (G1), solves inverse kinematic (IK) and searches a collision-free reach-to-grasp trajectory (G0). Adjigble [3] detects stable grasps, while also searching the most stable grasps to find those that are reachable, in real-time. Kitaev et al. [9] address this problem in a differently way, using rollouts in physics-based simulation to grasp the target object in a cluttered environment by pushing obstructing objects aside, [10].

However, none of the above work address the different problem of selecting grasps that will enable a desired post-grasp manipulative trajectory of the target object, without any parts of the robot or object encountering collisions. While most stable grasp configurations will typically allow some post-grasp manipulative movements without collision, these may not include a specific desired G2 object trajectory.

In Fig. 1(a), the Panda can successfully push a peg into a barrel if it grasps the peg at point 2 and moves it along the black arrow. However, if it rotates the peg $\frac{\pi}{6}$ [rad] clockwise, it will collide with the yellow cuboid despite the existing collision-free movements of the peg. Therefore, a synthesised grasping configuration may not permit the robot to perform desired post-grasp movements. Although G1 is mostly studied regardless of G2, some have studied them jointly. For instance, [11], [12] studied the problem of selecting a grasp configuration that yields minimum manipulation torque effort and minimum impact force in case of a collision. [13] presented an approach that exploits a semantic and geometric understanding of a scene for task-oriented grasping. This approach addresses the problem of grasping an object according to its affordance. Likewise, we propose an approach of selecting a grasp that yields collision-free post-grasp manipulative movements.

The previous works [3]–[5], [9] addressed joint planning of G0 and G1. Nonetheless, they did not consider post-grasp movements of the manipulator for synthesising grasp configuration. The trajectory of an object is sometimes known prior to the reach-to-grasp phase, e.g. welding metal parts, sealing glasses to a frame, painting, or opening a sliding door. Hence, the manipulator must plan a grasping configuration that allows collision-free post-grasp movements. We, therefore,

present an approach of grasp selection (G1) by considering collision of robot parts during a post-grasp manipulative movement of the target object (G2).

The contribution of this paper is that we compute a collision cost during G2 movements and use it to select a grasp candidate (G1) with the minimum collision cost for performing G2 movements. To show the effectiveness of our approach, we perform three real manipulation experiments with a real robot (“Panda” robot manufactured by FRANKA EMIKA GmbH). The experiments include (1) push a peg in a barrel, (2) rotate the peg in the barrel, and (3) pick-and-place task. Our experimental results demonstrate that our proposed cost effectively differentiates grasping configurations resulting in a collision. In a constrained environment, sample-based approaches, which randomly explore the environment, are very time-consuming. These approaches need more than a few seconds to find a solution whereas our approach finds a solution in a few milliseconds.

II. PROBLEM FORMULATION

Here we present an approach to synthesising grasp (G1) taking into account obstacle avoidance during the desired post-grasp movements (G2). Nevertheless, one can also use another state-of-the-art grasp synthesising approach.

A. Learning and generation of grasps for arbitrarily shaped objects

In this paper, we formalise the problem for parallel gripper which is originally presented in [2]. Nonetheless, the formulation for arbitrary hand kinematics can be found in original work, which proposes grasp models trained using only a single demonstrated grasp on a single object and computes grasping configurations for new objects of arbitrary shape, extracted from 3D depth images. The surface features $x \in SE(3) \times \mathbb{R}^2$ consist of curvature $r \in \mathbb{R}^2$ and a local frame attached to surface point $r x_g \in SE(3)$, where $SE(3)$ denotes the group of 3D poses (3D position p and 3D orientation q). $SE(3) = \mathbb{R}^3 \times SO(3)$ and $SO(3) \subset \mathbb{R}^{3 \times 3}$ denotes the group of rotations in three dimensions:

$$SO(n) = \{R \in \mathbb{R}^{n \times n} : RR^T = I, \det(R) = +1\}.$$

The object model is defined as a joint probability distribution of a set of features, modelled as kernel density estimation:

$$O(r x_g, r) = \sum_{j=1}^{K_0} w_j \mathcal{K}(r x_g, r | x_j, \sigma_x) \quad (1)$$

where $w_j \in \mathbb{R}^+$ are kernel weights, and

$\mathcal{K}(r x_g, r | x_j, \sigma_x) = \mathcal{N}_3(p | \mu_p, \sigma_p) \theta(q | \mu_q, \sigma_q) \mathcal{N}_2(r | \mu_r, \sigma_r)$
 μ and σ are the kernel mean and kernel bandwidth. \mathcal{N}_* is an $*$ -variate isotropic Gaussian kernel, and θ is a Gaussian-like distribution in $SO(3)$.

Contact model pdf $_i^M(u, r)$ encodes the joint probability distribution of surface features and of the 3D pose of the i_{th} hand link, where $u_{ij} = v_j^{-1} \circ s_i$, s_i and \circ denote the pose of link L_i ($i = 1, 2, 3$, representing two fingers and a palm

of end-effector) and the composition operator. The contact model of link L_i is

$$\mathcal{B}_i(u, r) = \frac{1}{Z} \sum_{j=1}^{K_0} \mathcal{N}_3(p|\mu_p, \sigma_p) \theta(q|\mu_q, \sigma_q) \mathcal{N}_2(r|\mu_r, \sigma_r) \quad (2)$$

where $Z \in \mathbb{R}^+$, $u_{ij} = (p_{ij}, q_{ij})$ is the pose of link L_i relative to the pose of v_j of the j th surface feature.

Query density Q_i is a density over possible i th link poses s given a new object point cloud. Q_i is computed by convolving the corresponding contact model B_i with a new object point cloud O .

$$Q_i(s) \simeq \sum_{j=1}^{K_{Q_i}} w_{ij} \mathcal{N}_3(p|\hat{p}_{ij}, \sigma_p) \theta(q|\hat{q}_{ij}, \sigma_q) \quad (3)$$

To generate a grasp for a new object, a finger link is selected at random, and a link pose is sampled from the query density. Hence, the corresponding hand configuration is determined by $h_c = \text{FK}({}^r x_g, h_i)$ where ${}^r x_g$ is the poses of the end-effector and h_i , is the pose of L_i , $i = 1, 2, 3$. We now compute the likelihood of a grasp using the kernels

$$\mathcal{L}_Q(h_c) = \prod_{Q_i \in Q} Q_i(\text{FK}({}^r x_g, h_i)) \quad (4)$$

where FK_i denotes the forward kinematics corresponding to i th link of the hand. The objective of grasp optimisation is to find a grasp that maximises the product of the likelihood of the query densities and the hand configuration density

$$[{}^r x_g, \bar{h}_i] = \text{argmax} \mathcal{L}(h_c) \quad (5)$$

We compute the kernels based on an available example of a successful grasp. Then, we sample a set of grasp candidates from the kernels using eq. (4). In the next section, we study this set of grasp candidates and analyse them to find which one results in collision-free post-grasp movements.

B. Post-grasp movements (G2) and obstacle avoidance

We consider reference frame $\mathbf{x}_r \in SE(3)$ (the black frame in Fig. 2). Frame ${}^r \mathbf{x}_g \in SE(3)$ (in eq. 5) is attached to the end-effector at each time (shown with blue thick frames). We use operational space trajectory to refer to successive poses of this frame that correspond with a sequence of poses attached to the centre of mass of the object ${}^r \mathbf{x}_c = \{{}^r \mathbf{o}_c, {}^r \mathbf{x}_c, {}^r \mathbf{y}_c, {}^r \mathbf{z}_c\}$. ${}^r \mathbf{x}_c(t) = \{{}^r \mathbf{t}_c(t), {}^r \mathbf{R}_c(t) \mid \forall 0 \leq t \leq T\}$ defines a desired trajectory for the object, t denotes a time and t_f is time-to-completion for the manipulative movements, ${}^r \mathbf{R}_c(t)$ is rotation matrix from ${}^r \mathbf{x}_c(t)$ to \mathbf{x}_r , and ${}^r \mathbf{t}_c(t)$ represents translation of ${}^r \mathbf{x}_c(t)$ expressed in \mathbf{x}_r . We assume the object is non-deformable. Hence, when the robot comes into contact with the object, a trajectory of the corresponding end effector pose at grasping configuration can be expressed based on ${}^r \mathbf{x}_c$ and a fixed transformation from ${}^r \mathbf{x}_g$ into ${}^r \mathbf{x}_c$, as follows:

$$\begin{aligned} {}^r \mathbf{R}_g(t) &= {}^r \mathbf{R}_c(t) {}^c \mathbf{R}_g \\ {}^r \mathbf{t}_g(t) &= {}^r \mathbf{t}_c(t) + {}^r \mathbf{R}_c(t) {}^c \mathbf{t}_g \end{aligned} \quad (6)$$

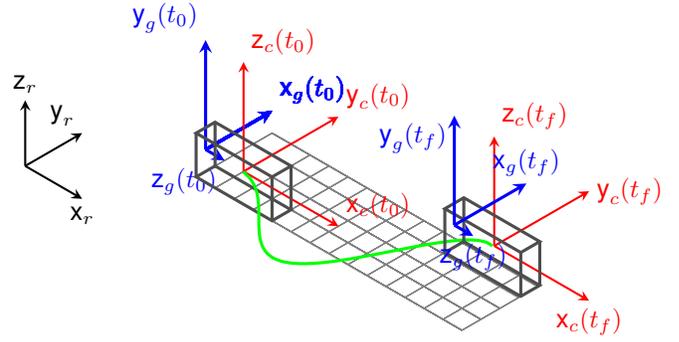


Fig. 2. An object is shown in global coordinate frame $\mathbf{x}_r = \{\mathbf{o}_r, \mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r\}$ at initial and final time (t_0 and t_f) in the left and right side of the figure. Local coordinate frame ${}^r \mathbf{x}_c = \{\mathbf{o}_c, \mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c\}$ is attached to the centre of mass of the object, shown in red colour. This frame follows trajectory ${}^r \mathbf{x}_c(t)$ shown with green line during manipulative movements. The blue frame ${}^r \mathbf{x}_g = \{\mathbf{o}_g, \mathbf{x}_g, \mathbf{y}_g, \mathbf{z}_g\}$ represents a coordinate frame corresponding with a end effector pose of one feasible grasp configuration at time t_0 and t_f

where ${}^c \mathbf{x}_g = \{{}^c \mathbf{R}_g, {}^c \mathbf{t}_g\}$ are rotation and translation from ${}^r \mathbf{x}_g$ to ${}^r \mathbf{x}_c$, respectively. We consider the trajectory for object movements, namely ${}^r \mathbf{x}_c(t)$, is known, e.g. moving/opening a sliding door. Hence, for each grasping configuration, namely ${}^c \mathbf{x}_g$, the operational space trajectory of the end-effector, namely ${}^r \mathbf{x}_g(t) = \{{}^r \mathbf{t}_g(t), {}^r \mathbf{R}_g(t)\}$, is fixed and can be computed using eq.(6), i.e. ${}^r \mathbf{x}_g = {}^r \mathbf{x}_c {}^c \mathbf{x}_g$. An IK algorithm is then utilised to compute the corresponding joint space trajectory of the manipulator as follow:

$$\mathbf{q}(t|{}^r \mathbf{x}_c, {}^c \mathbf{x}_g) = \text{IK}({}^r \mathbf{x}_g(t)) \quad (7)$$

where the vertical line in $f(y|x)$ mean f is a function of y given x . In eq. (7), $\mathbf{q}(t|{}^r \mathbf{x}_g) = \{q_i(t) \mid q_i(t|{}^r \mathbf{x}_g) \in \mathbb{R}, i = 1, \dots, n_q\}$ at each time step and n_q is the number of joints

We consider some body points attached to manipulator's links ($b_j = \{b_{j,1} \in \mathbb{N}, b_{j,2} \in \mathbb{R}\}$ where $b_{j,1}$ denotes the link number to which the body point is attached and $b_{j,2}$ represents the corresponding distance between the j th joint and the body point.), e.g. one body point is attached to every joint and one is attached to the middle of every link. Thus, the trajectory of each body point is:

$$\mathbf{b}_j(t|{}^r \mathbf{x}_g) = \text{FK}_j(\mathbf{q}(t|{}^r \mathbf{x}_g)), \quad j = 1, \dots, n_{bp} \quad (8)$$

where n_{bp} is the number of body points and FK_s is the forward kinematics showing the position of j th body point, namely $\mathbf{b}_j \in \mathbb{R}^3$, at each time step. A group of trajectories expressing the movements of the body points, as per eq. (8), for a known grasping candidate ${}^r \mathbf{x}_g$ and a given sampling time of a desired post-grasp trajectory is

$$\mathbf{X}|{}^r \mathbf{x}_g = \{\mathbf{b}_j(k) \mid \mathbf{b}_j(k)|{}^r \mathbf{x}_g = \text{FK}_j(\mathbf{q}(k)|{}^r \mathbf{x}_g); \quad j = 1, \dots, n_{bp}; k = 1, \dots, K\} \quad (9)$$

where $f(y)|_x$ means f is a function of y given x , $\mathbf{X}|{}^r \mathbf{x}_g \in \mathbb{R}^{n_{bp} \times K}$, n_{bp} is the number of body points and K represent the total number of discrete time steps k . For the sake of simplicity, we write \mathbf{b}_j instead of $\mathbf{b}_j(k)|{}^r \mathbf{x}_g$ in the following.

We specify an obstacle by a set of linear constraints as follows:

$$g_h^i(x) \leq 0, g_h^i \in L^m, x \in \mathbb{R}^3 \quad (10)$$

This expresses a set of inequalities describing a convex region including i_{th} obstacle where Ls are linear functions. [14] proposed a function that is arbitrarily large in the region inside the obstacle and decreases sharply proportional to d that is the distance between \mathbf{b} and $g_h(x) = 0$ using eq. (9) and (10).

$$d_i(\mathbf{b}_j, x) = \sum_{h=1}^{n_h} \{ \hat{d}(g_h^i(x) = 0, \mathbf{b}_j) + | \hat{d}(g_h^i(x) = 0, \mathbf{b}_j) | \} \quad (11)$$

where n_h is the number of linear constraints and \hat{d} is the distance between \mathbf{b} and the plane $g_h(x) = 0$. If \mathbf{b}_j is on or inside the convex region of the i_{th} obstacle, then $d_i \leq 0$. Hence, we express the costs for all body points, obstacles and sample points of G2 by using eq. (10) and (11) as follows:

$$C_i({}^r \mathbf{x}_g | g_h) = \begin{bmatrix} c_{11}^i & c_{12}^i & c_{13}^i & \cdots & c_{1T}^i \\ c_{21}^i & c_{22}^i & c_{23}^i & \cdots & c_{2T}^i \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n_{bp}1}^i & c_{n_{bp}2}^i & c_{n_{bp}3}^i & \cdots & c_{n_{bp}T}^i \end{bmatrix} \quad (12)$$

Where $c_{j,k}^i | {}^r \mathbf{x}_g, g_h$ represents the cost $d_i(\mathbf{b} | {}^r \mathbf{x}_g, g_h)$ for the i_{th} obstacle, j_{th} body point at k_{th} time step. We can easily extend our cost computation to more than one obstacle $C = [C_1, C_2, \dots, C_{n_o}]$ where n_o is the number of obstacles. Finally, eq. (13) defines the collision cost for a desired post-grasp trajectory of the object and an obstacle. Given the desired trajectory of the object, as per eq. (6), and an obstacle as per eq. (10), we can write $C({}^r \mathbf{x}_g, g_h)$ in eq. (12) as a function of ${}^c \mathbf{x}_g$, i.e. $C({}^c \mathbf{x}_g | {}^r \mathbf{x}_c, g_h)$. Hence, we write the obstacle avoidance cost $J(C({}^r \mathbf{x}_g, g_h))$ as a function of ${}^c \mathbf{x}_g$, as per eq. (13).

$$J({}^c \mathbf{x}_g | {}^r \mathbf{x}_c, g_h) = | \mathbf{1} ./ (C({}^c \mathbf{x}_g | {}^r \mathbf{x}_c, g_h) + \mathbf{1}\delta) | \mathbf{1} \quad (13)$$

where δ^{-1} is a fixed value for setting the maximum available cost, $\mathbf{1}$ is a matrix whose elements are equal to 1, and $./$ is the element wise division operator. J yields maximum costs on and inside the convex region representing an obstacle. A post-grasp manipulator trajectory that yields minimum cumulative value of cost ideally allows collision-free movement for the robot. By definition, the value of J is a function of grasping configuration, obstacle position and shape and object trajectory.

$${}^c \mathbf{x}_g^* = \underset{{}^c \mathbf{x}_g}{\operatorname{argmin}} J \quad (14)$$

Therefore, we select the optimal grasp candidate ${}^c \mathbf{x}_g^*$ that minimises collision cost for the given post-grasp movements and obstacle position.

2-D link simulated manipulator

Fig. 3(a) shows a 2-D manipulator tasked with grasping a rectangle object from its top edge and moving it from the bottom right to top left of the image. Nonetheless, the grasps at the far left part of the top edge of the object result in

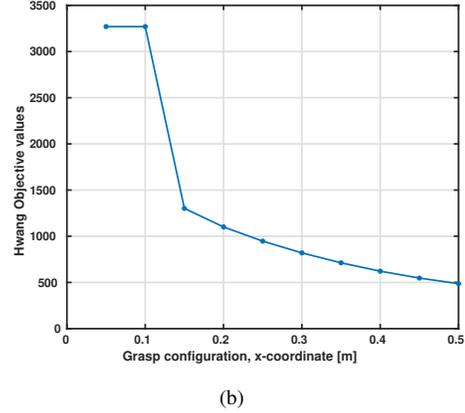
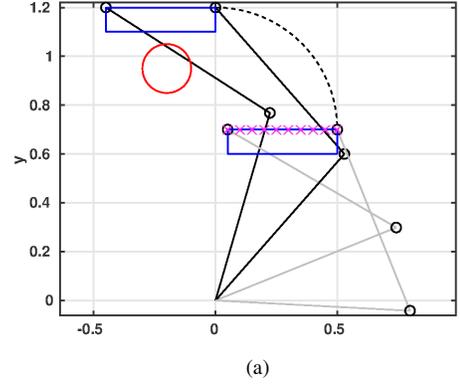


Fig. 3. (a) 2-D planar manipulator moves the cuboid object from bottom right to top left along the black dashed line. The red circle represents an obstacle added to the workspace. (b) shows the cost corresponding with each grasping candidate at the initial position of the object. Grasps at 0.05 [m] and 0.1 [m] yield collision, hence, they result in the maximum value of cost, i.e. δ^{-1} .

collisions between the link of the manipulator and an added obstacle, shown by the red circle. As such, from the set of feasible grasping configurations (shown by green crosses on the blue rectangle in Fig. 3(a)), just a few numbers of them allows the manipulator to perform the desired post-grasp movements. Therefore, the robot must select the one that yields overall collision-free movements.

For each grasping configuration, an obstacle avoidance cost is estimated using eq. (13). Fig. 3(b) shows the estimated costs for selecting a grasp configuration and perform the given task. The lower the cost, the larger is the distance between the manipulator and the obstacles during movements. In 3(b) the vertical axis shows the obstacle avoidance cost J and the horizontal axis is the x coordinate of the grasping configuration on the top edge of the blue object. Each grasping configuration is at 0.05 [m] from another. The cost values suggest that grasping configurations at far right on the object (i.e. $x = 0.5[m]$) is the most suitable one for moving the object along the given trajectory.

III. EXPERIMENTAL RESULTS

We set up a ‘‘Panda’’ manipulator manufactured by FRANKA EMIKA to perform experiments showing the ef-

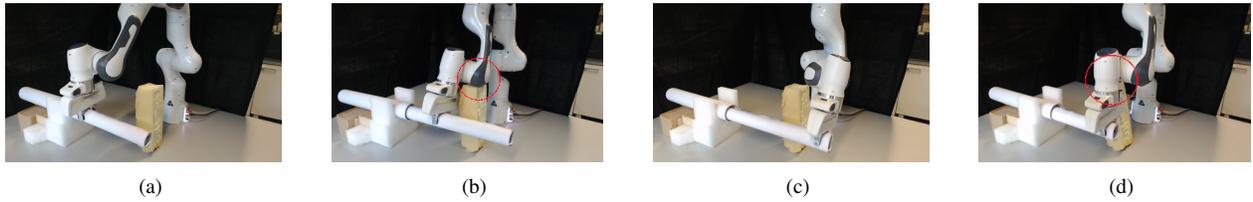


Fig. 4. This figure shows the Panda performs pushing ((a) and (b)) and rotating ((c) and (d)) by grasping the cylindrical peg at 1 and 2 (shown in Figs. 1(a) and 1(b)). Figs. (a) and (b) show the Panda performed the inserting and rotating task without colliding with the obstacle while it grasped the peg at 1 and 2, respectively. On the contrary, grasp 1 and 2 resulted in a collision during performing pushing and rotating task, respectively (Figs. (b) and (d)).

effectiveness of our approach for selecting a grasping configuration. The Panda has 7 revolute joints and an electric parallel jaw gripper. We designed three experiments (i) pushing a peg into a barrel (Fig. 1(a)), (ii) rotating the peg in the barrel (Fig. 1(b)), and (iii) pick-and-place task (Fig. 5).

First, the Panda is tasked with pushing and rotating the cylindrical peg in the barrel (Fig. 4). This is a sample of common tasks in industry in which the trajectory of the object is constrained and is known prior to G_0 and G_1 . Two stable grasps are considered (Fig. 1) that have high likelihood as per eq. (5). We use Trac-IK [15] and eq. (6) to compute the trajectory of the end-effector corresponding with G_2 (shown with the black arrows in Fig. 1(a) and 1(b)) and grasp 1 and 2. In the first task, the manipulator pushes the peg into a cylindrical barrel by moving the peg from right to left, as shown in Fig. 4(a). In the second task, the manipulator rotates the peg clockwise in the barrel by $\frac{\pi}{6}$ [rad] (Fig. 4(c)). An obstacle in the workspace, however, makes one of the grasp unsuccessful as it results in collision between the manipulator and the obstacle. The collision costs are $J_{(t_1, g_1)} = 11126$, $J_{(t_1, g_2)} = 3512$, $J_{(t_2, g_1)} = 6887$ and $J_{(t_2, g_2)} = 15318$ where t_* and g_* represent $*_{th}$ task and grasp, respectively. These cost values indicate, grasp 2 results in collision-free movements during performing task 1, as shown in Fig. 4(a), while it collides with the obstacle during executing task 2 as shown in Fig. 4(b). On the other hand, grasp 1 results in successful completion of task 2 (Fig. 4(c)) while yields collision during post-grasp movements of task 1 (Fig. 4(d)). This shows the cost values correctly predict collisions during post-grasp movements.

Next, the Panda is tasked with picking up a white cuboid object (located on the table in front of the robot) and placing it at the desired target pose shown with G in Fig. 5(a)). The desired trajectory corresponding with the object movements (G_2) is shown with red dashed line and is known prior to the experiment. A set of 6 grasping configurations with the high likelihood are also considered (using eq. (5)).

We add an obstacle to the robot’s workspace. The obstacle position is captured by an Astra Pro RGB-D sensor, tracked using a marker-based Aruco tracker and is represented in the robot frame by camera calibration transformation. As the collision-free object movements are known, only a collision between the manipulator and the obstacle may cause a problem during movements. We compute the collision cost for all 6 grasps (shown in Fig. 5(a)) which are shown in

Fig. 5(d)). These costs show that grasp 5 and 6 are collision-free (Fig. 5(c)) and grasp 1 to 4 yield collision between the manipulator and yellow cuboid during post-grasp movements (Fig. 5(b)). We performed the desired movements and Fig. 5(b) and 5(c) show the manipulator during post-grasp movements with unsuccessful (grasp 1) and successful (grasp 6) grasping configurations, respectively.

A. Time evaluation

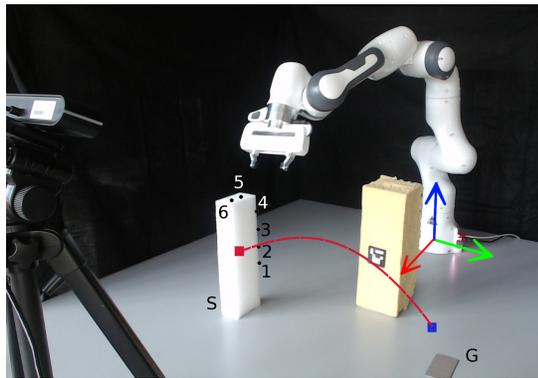
The time for computing 10,000 IK and the obstacle avoidance cost is shown in Fig. 6. The mean value of IK and the cost computation time is 0.0104 [s] and 0.0107 [s], respectively. Although the maximum time of computing IK and J are 0.0177 [s] and 0.0171 [s], these represent outliers of data. 25_{th} and 75_{th} percentiles of the computation time samples are in the interval of 0.0102 [s] and 0.0103 [s] for IK and 0.0105 [s] and 0.0107 [s] for the whole cost computation. This includes computing the IK and obstacle avoidance cost. To sum, our expected mean and median cost computation frequency are 93 [Hz] and 94 [Hz], respectively. In the worst case, the expected cost computation frequency is 58 [Hz].

To show the correlation between the obstacle position and the computed cost, we performed another experiment shown in the paper’s video. In this experiment, a human is moving an obstacle in the robot’s workspace while our proposed approach computes the collision cost. Fig. 7(a) and 7(b) show the trajectory of the obstacle moved by a human. Fig. 7(c) depicts the corresponding obstacle avoidance costs for all 6 grasp configurations at each time step. Three parts can be identified in the trajectory of costs in Fig. 7. In the first and the last parts of the trajectory, grasping configurations 3, 4, 5 and 6 yields high value of costs indicating possible collision of robot and obstacle whereas they obtain very small cost values in the mid part of the trajectory. On the other hand, grasps 1 and 2 yield collision-free post-grasp movements during the initial and final parts of the obstacle trajectory.

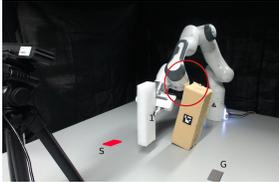
This paper is accompanied by a video of the corresponding experiments. The video can be watched on youtube (<https://youtu.be/TqaPLtiiUck>).

IV. CONCLUSION AND DISCUSSION

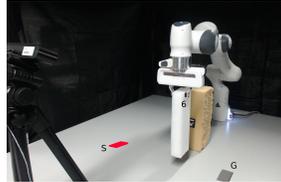
This paper presents a method for selecting a grasp enabling post-grasp manipulative trajectories to be achieved without collisions. This work complements other work on algorithms for grasp synthesis. In principle, any grasp synthesis method



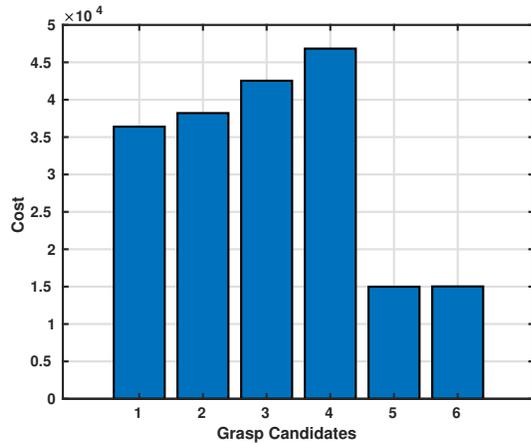
(a)



(b)



(c)



(d)

Fig. 5. Experimental set up: (a) the Panda robot picks up the white cuboid object at initial position S and places it on grey rectangle G . The movement trajectory is assumed to be given and is shown with the red dashed line. The robot reference frame is shown with RGB colour corresponding with $\{X, Y, Z\}$. A marker-based tracking system computes the position of the obstacle in the robot reference frame. Based on this position a cost for each grasp candidates is evaluated as per eq. (13). (b) the robot collides with the obstacle (marked with the red circle) as it randomly selected the grasping configuration without using our predictive grasping approach; (c) the robot successfully performs the task by using our predictive grasp planning approach. (d) collision costs corresponding with each grasp candidate of the pick-and-place task shown in Fig. 5(a).

can be combined with this work in a modular way, i.e. our method will select between a variety of stable grasps, proposed by an arbitrary grasp synthesis method, and choose the best to enable the post-grasp manipulation. Our method achieves this by computing, for each proposed grasp, a corresponding collision cost that would be incurred by a manipulator in order to deliver the target object along its desired post-grasp trajectory.

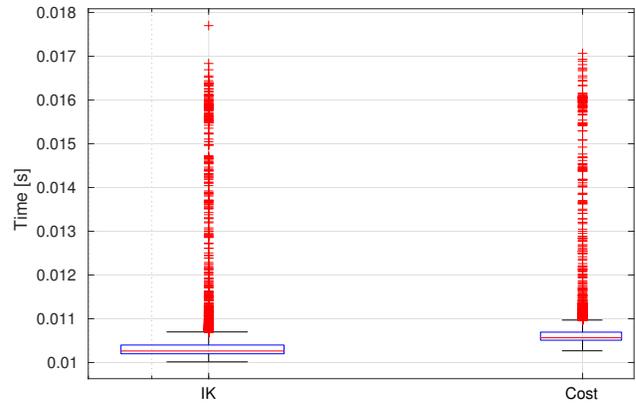


Fig. 6. Time of computing 10,000 times the IK and the obstacle avoidance cost values. The mean value of computation time is 0.0104 and 0.0107 [s] for IK and cost, respectively. The post-grasp trajectory includes 100 sample points. Nonetheless, the number of sample points of a post-grasp trajectory can be significantly reduced as long as the distance between two consecutive sample points is less than the minimum dimension of the obstacle, e.g. length, height and width of a cuboid. This allows us to be sure there is at least one sample capturing a collision.

Previous studies have predominantly considered (1) forming a stable grasp on an object and (2) collision-free reach-to-grasp trajectory planning. In contrast, this paper has studied how intelligent grasp selection enables collision-free post-grasp movements. We have demonstrated the effectiveness of our approach in three different common robotics tasks (1) pushing a cylindrical peg inside a tube, and (2) rotating the peg inside the tube, and (3) a pick-and-place task. Our results show how a robot can intelligently select a grasping configuration by analysing possible collisions during post-grasp motions.

A statistical analysis of computation time and our previous study [16] suggest that we can use this system to guide a human operator (using a haptic teleoperation device), by providing real-time haptic cues towards a grasp that is collision-free during known post-grasp trajectories. For this application, a cost-computation frequency greater than 15 [Hz] was shown to be sufficient for providing real-time perception of haptic cues to the human. In this paper, our empirical experiments showed that the frequency of computing collision costs, on a standard 5-core Intel PC, is greater than 50 [Hz] in the worst case. Therefore, our future work will extend this approach, and use it for generating haptic operator-assistance cues in a teleoperation scenario. Such cues will help the human operator to steer a master device, to effect grasps with a remote slave robot which are both stable and also facilitate collision-free post-grasp movements, which may be also learned from demonstrations [17]. Since our method is based on computations which can readily be parallelized, the approach is computationally scalable to cope with the case of more obstacles and highly cluttered scenes in real-time.

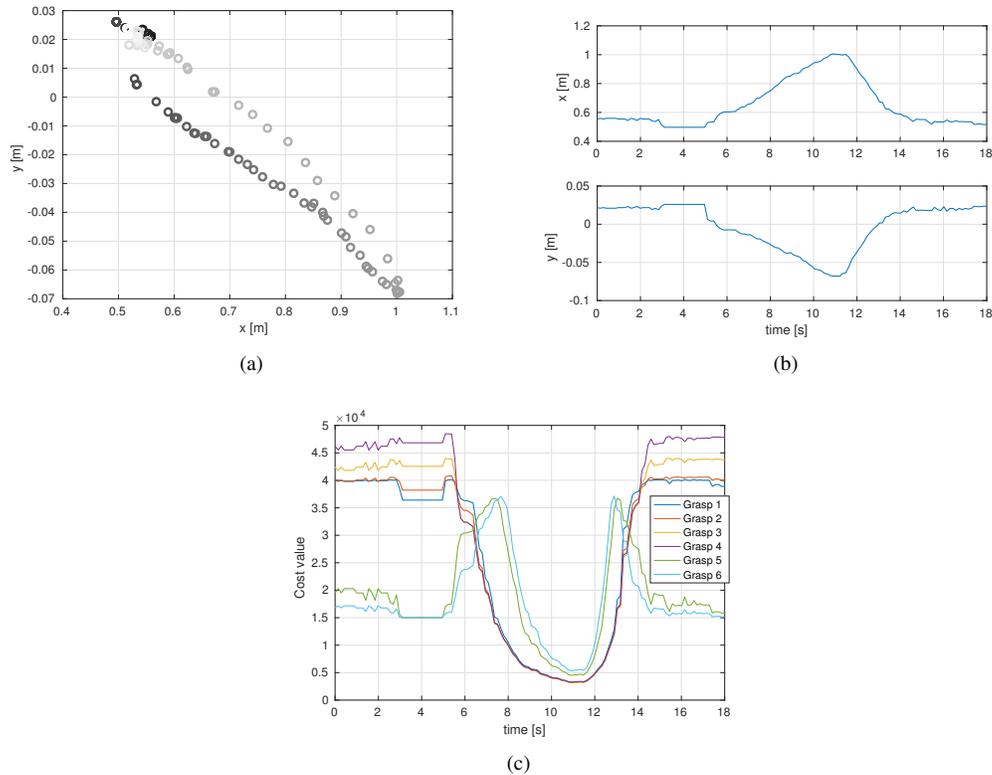


Fig. 7. These figures show the computed cost values for performing the pick-and-place task with the Panda robot. (a) and (b) shows the trajectory of the obstacle that is moved by a human during the experiment. (c) shows evaluated values for each grasp configurations during the experiment. These figures demonstrate how the values of costs at different grasping configurations vary with the change in position of the obstacle. In the middle of the experiment, the obstacle is away from the robot’s workspace which yields a drop in all the costs values. This makes even grasps 5 and 6 less costly than grasps 1 to 4. However, if the obstacle is in the way of manipulative movements, as per $t = 0 - 5[s]$ to $t = 14 - 18[s]$, grasps 5 and 6 yield very large value of costs.

REFERENCES

- [1] K. S. A. ten Pas, M. Gualtieri and R. Platt, “Grasp pose detection in point clouds,” *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.
- [2] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. Wyatt, “One-shot learning and generation of dexterous grasps for novel objects,” *International Journal of Robotics Research*, vol. 35, no. 8, pp. 959–976, 2016.
- [3] M. Adjigble, N. Marturi, V. Ortenzi, V. Rajasekaran, P. Corke, and R. Stolkin, “Model-free and learning-free grasping by local contact moment matching,” in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018.
- [4] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann, “Integrated grasp and motion planning,” in *IEEE International Conference on Robotics and Automation*, pp. 2883–2888, IEEE, 2010.
- [5] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, “Grasp planning in complex scenes,” in *IEEE-RAS International Conference on Humanoid Robots*, pp. 42–48, Nov 2007.
- [6] C. Ferrari and J. Canny, “Planning optimal grasps,” in *IEEE-RSJ International Conference on Robotics and Automation*, pp. 2290–2295, IEEE, 1992.
- [7] D. Ding, Y.-H. Lee, and S. Wang, “Computation of 3-d form-closure grasps,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 515–522, 2001.
- [8] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, “High precision grasp pose detection in dense clutter,” in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, pp. 598–605, 2016.
- [9] N. Kitaev, I. Mordatch, S. Patil, and P. Abbeel, “Physics-based trajectory optimization for grasping in cluttered environments,” in *IEEE International Conference on Robotics and Automation*, pp. 3102–3109, 2015.
- [10] A. C. H. Quispe, H. B. Amor, H. I. Christensen, and M. Stilman, “Grasping for a purpose: Using task goals for efficient manipulation planning,” *Computing Research Repository*, vol. abs/1603.04338, 2016.
- [11] N. Mavrakis, A. M. Ghalamzan E., and S. Rustam, “Safe robotic grasping: Minimum impact-force grasp selection,” in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, pp. 4034–4041, IEEE, 2017.
- [12] N. Mavrakis, Amir M. Ghalamzan E., R. Stolkin, L. Baronti, M. Kopicki, and M. Castellani, “Analysis of the inertia and dynamics of grasped objects, for choosing optimal grasps to enable torque-efficient post-grasp manipulations,” in *IEEE-RAS International Conference on Humanoid Robots*, pp. 171–178, IEEE, 2016.
- [13] R. Detry, J. Papon, and L. Matthies, “Task oriented grasping with semantic and geometric scene understanding,” in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, pp. 3266–3273, IEEE, 2017.
- [14] Y. K. Hwang and N. Ahuja, “A potential field approach to path planning,” *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 23–32, Feb 1992.
- [15] P. Beeson and B. Ames, “Trac-ik: An open-source library for improved solving of generic inverse kinematics,” in *IEEE-RAS International Conference on Humanoid Robots*, pp. 928–935, IEEE, 2015.
- [16] A. M. Ghalamzan E., F. Abi-Faraj, P. Robuffo Giordano, and R. Stolkin, “Human-in-the-loop optimisation: mixed initiative grasping for optimally facilitating post-grasp manipulative actions,” in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, p. submitted, 2017.
- [17] A. M. Ghalamzan E. and M. Ragaglia, “Robot learning from demonstrations: Emulation learning in environments with moving obstacles,” *Robotics and Autonomous Systems*, vol. 101, pp. 45–56, 2018.