Optimization-Based Human-in-the-Loop Manipulation Using Joint Space Polytopes

Philip Long, Tarik Kelestemur, Aykut Ozgun Onol, Taskin Padir

Optimization-Based Human-in-the-Loop Manipulation Using Joint Space Polytopes

Philip Long, Tarık Keleştemur, Aykut Özgün Önol and Taşkın Padır [1]

***Abstract*— This paper presents a new method of maximizing the free space for a robot operating in a constrained environment under operator supervision. The objective is to make the resulting trajectories more robust to operator commands and/or changes in the environment. To represent the volume of free space, the constrained manipulability polytopes are used. These polytopes embed the distance to obstacles, the distance to joint limits and the distance to singular configurations. The volume of the resulting Cartesian polyhedron is used in an optimization-based motion planner to create the trajectories. Additionally, we show how fast collision-free inverse kinematic solutions can be obtained by exploiting the pre-computed inequality constraints. The proposed algorithm is validated in simulation and experimentally.**

I. INTRODUCTION

Robotics and remote systems are being deployed in nuclear clean-up missions internationally including Japan, Korea, the United Kingdom and the United States. However, their widespread adoption in high-consequence material handling tasks is limited due to the risks associated with failure modes of (semi-)autonomous robots. As a result, most systems are operated in a human-in-the-loop manner, often with multiple levels of operators distributing the risks in decision-making. Robot manipulators for high-consequence material handling often operate in constrained, and cluttered environments, for instance inside nuclear gloveboxes. Due to their critical nature, during manipulation tasks these robots are supervised by an experienced human operator who may or may not override the robot's motions on-line.

This research aims to develop a human-in-the-loop manipulation method to provide the operator with increased situational awareness and control flexibility during task execution. This is achieved by maximizing the allowable (obstacle-free) volume around the robot's end-effector by introducing constraints into the joint-space manipulability polytopes. The overall idea is to define, track and visualize the constrained end-effector manipulability polytopes as the operator moves the robot remotely to provide the operator with a metric-based intuitive manipulation planning tool.

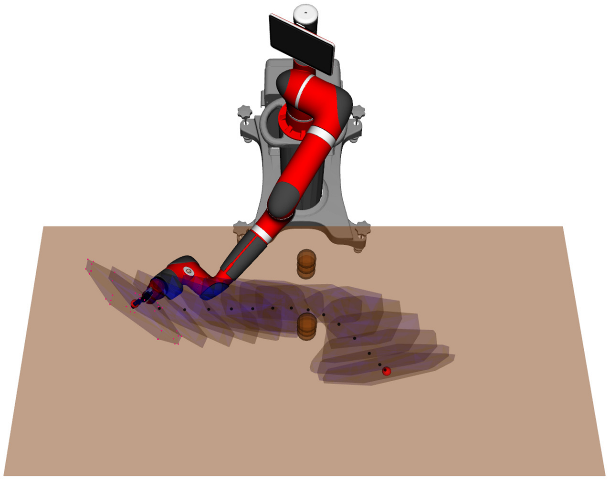


Fig. 1. Visualization of virtual fixtures obtained using the constrained manipulability polytope for Sawyer in a cluttered environment. The brown spheres and table are treated as collision objects. The resulting polytope is shown in blue for the whole trajectory. It can be seen how the polytope shrinks in volume as the tool passes through the constrained environment. The volume represents the constraints due to obstacles in the environment, positional joint limits and transmission ratio from configuration to task space.

A. Related Work

The primary approach of combining planning and control is by deforming a candidate path using a local controller [1]. For instance, in the elastic bands/strips frameworks [2–4], the volume of free space, containing all feasible robot configurations, is obtained from a globally planned candidate path. Thus, during execution a local controller can modify the candidate path providing new solutions within this volume. In contrast, much research [5] has focused on decreasing the on-line reaction time in order to cope with sensor based events. Recently, [6, 7] proposed a quasi-real time quadratic programming based controller to satisfy constraints of a particular event, for instance to satisfy visibility constraints while ensuring velocity and acceleration bounds are respected. Instead of attempting to deform a candidate plan in order to cope with unexpected events, an alternative approach creates a trajectory using optimization techniques [8–11] that is robust to uncertainties.

While the above methods can obtain trajectories that are robust and minimize the likelihood of collision, our objective is to maximize the free space of the end effector during the motion, as such it resembles much work in the field of virtual fixtures (active constraints) most frequently used for surgical

*This research is supported by the Department of Energy under Award Number DE-EM0004482, by the National Aeronautics and Space Administration under Grant No. NNX16AC48A issued through the Science and Technology Mission Directorate, by the National Science Foundation under Award No. 1451427, and by the Office of the Secretary of Defense under Agreement Number W911NF-17-3-0004.

[1] Philip Long, Tarık Keleştemur, Aykut Özgün Önol, and Taşkın Padır are with Department of Electrical and Computer Engineering, Northeastern University, Boston, Massachusetts 02115, USA. {p.long,t.padir}@northeastern.edu {onol.a,kelestemur.t}@husky.neu.edu

robots [12, 13]. Virtual fixtures define zones where the manipulator can operate and thus can aid operators overseeing predefined trajectories. These zones can be obtained using point cloud data [14], shape primitives such as tubes curves or cones, [15] or generated on-line by operator input [16]. Indeed, in [17] a library of virtual fixtures for surgical robots is presented. The authors show how including hyper-planes and polyhedra that approximate non-linear constraints can lead to a decreased computation time when obtaining the inverse kinematic solution. Yet these fixtures often focus exclusively on Cartesian space constraints, neglecting the transform from joint space to task space, joint limits and free space surrounding the manipulator body.

In this work, we aim to maximize the volume of the virtual fixture and present this free space in an intuitive way to an overseeing supervisor. Similar to [18], the performance index itself is part of the optimization criteria. In order to achieve this, we propose combining an optimization method with a representation of robot capacities akin to manipulability maps proposed in [19, 20]. Manipulability maps, or reachability maps, discretize the work-space and label each cell with a value based on the manipulability ellipsoid [21]. The manipulability value can also include additional constraints, such as joint limits for instance in [22] or indeed obstacles in the workspace [23]. While manipulability indicates the robot's capacity to transmit velocities and forces from the configuration space to the task space, relating this to actual allowable end effector displacements can lead to several errors [24]. Instead, we focus on another way of representing manipulability capacities: manipulability polytopes. Manipulability polytopes represent the actual velocity bounds [25] rather than the approximation provided by ellipsoids [26]. Moreover, since a polytope is defined by a set of inequality constraints further constraints can be incorporated into existing polytopes, for instance mobile robot toppling constraints in [27], friction cones [28], or maximum danger values [29].

With this in mind, we propose a new manipulation planning method that aims to increase the feasible space for on-line reactive control based on the idea of constrained manipulability polytopes. While in [29] the manipulability polytopes are constructed to obtain reduced velocity bounds, in this paper we assume that over small distances these polytopes represent positional workspace bounds. Indeed, a similar idea is proposed in [30], where joint limit constraints are imposed by ensuring that the Cartesian position of the robot's foot remains within a conservatively small box.

In contrast to the aforementioned planning methods, the robot's trajectory in a cluttered environment is generated with the explicit goal of obtaining trajectories for on-line control, i.e., trajectories that maximize the possible translational deviations of the end effector. These motions are represented by a convex polytope and thus solutions that lie inside this space can be rapidly obtained. To achieve this, the set of allowable joint deviations, that take into account both obstacles and joint limits, is determined from linear inequality constraints. The resulting convex polytope is transformed into a Cartesian polyhedron to represent the feasible translational displacements. Additionally, by representing the constraints in the task space, the set of feasible robot motions can be intuitively understood by the operator during supervised autonomy. The method is demonstrated for a guarded teleoperation example where the polytopes define virtual fixtures. This paper is organized as follows. In Section II, the methodology, comprising the robot modeling, allowable motion generation and trajectory optimization, is presented. In Section III, simulation and experiments that validate our approach are shown. Finally in Section IV, the conclusions are drawn and future work is discussed.

II. Methodology

A. Robot Model

In this paper, an n degree-of-freedom manipulator operating in 6 dimensional space is considered. The pose of the end effector is represented by the vector $\mathbf{x}_n \in \mathbb{R}^6$, consisting of three translational variables and the minimal representation of orientation. $\mathbf{x}_n$ can be obtained from the manipulator's configuration variables that are denoted as $\mathbf{q} = [q_1, q_2 \dots q_n]$, using the forward kinematics, denoted as fk i.e.,

$$\mathbf{x}_n = fk(\mathbf{q}). \tag{1}$$

The twist at the n^{th} frame, i.e., the end effector, denoted by $\boldsymbol{\nu}_n$, is obtained from the differential kinematic model as

$$\boldsymbol{\nu}_n = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}_n \dot{\mathbf{q}}, \tag{2}$$

where $\mathbf{J}_n \in \mathbb{R}^{6\times n}$ is the kinematic Jacobian matrix defined at frame n. $\mathbf{v}$ and $\boldsymbol{\omega}$ denote the translational and angular velocity respectively. $\dot{\mathbf{q}} = [\dot{q}_1, \dots, \dot{q}_n]^T$ is the joint velocity vector. Finally, the manipulator's dynamic model is given by

$$\boldsymbol{\tau} = \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}), \tag{3}$$

where $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the inertial matrix, $\ddot{\mathbf{q}}$ is the joint acceleration vector, $\ddot{\mathbf{q}} = [\ddot{q}_1, \ddot{q}_2 \dots, \ddot{q}_n]^T$, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ represents the virtual forces of the Coriolis, centrifugal, and gravitational terms and $\boldsymbol{\tau} \in \mathbb{R}^n$ vector of generalized joint torques.

B. Allowable Motions

At an instant k during a trajectory, the end effector pose is given by $\mathbf{x}_k$ and the instantaneous joint velocity is given as $\dot{\mathbf{q}}_k$. Over a period of δt seconds, assume the end effector travels a distance of $\delta\mathbf{x}_k$, then the resulting end-effector pose at instant $k+1$ can be obtained by linearizing (2) using (1)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta\mathbf{x}_k \approx fk(\mathbf{q}_k + \delta\mathbf{q}_k), \tag{4}$$

$$\delta\mathbf{x}_k = \mathbf{J}_n \delta\mathbf{q}_k, \tag{5}$$

where the displacement of the joint variables, denoted $\delta\mathbf{q}_k$, is given as

$$\delta\mathbf{q}_k = \dot{\mathbf{q}}_k \; \delta t. \tag{6}$$

Within a cluttered environment the allowable motion of the end effector is dependent on the links' allowable motion. This means that even if the end effector is nominally located in free space, certain attainable motions may be restricted by obstacles close to the antecedent links of the chain. Evidently, the joint deviation (4) generates motion of points all along the manipulator's body. Consider a point $\mathcal{I}$ on the robot's kinematic chain whose position with respect to the world frame is denoted by the vector $\mathbf{r}_i$. The motion of a point $\mathcal{I}$ on the robot's kinematic chain is obtained from (5) as

$$\delta\mathbf{x}_k^i = \mathbf{J}_i^v \delta\mathbf{q}_k, \tag{7}$$

where $\mathbf{J}_i^v \in \mathbb{R}^{3\times n}$ denotes the kinematic Jacobian matrix that relates the velocities of the preceding joints in kinematic chain to the linear velocity at point $\mathcal{I}$, with columns of zeros corresponding to joints which do not generate velocity at $\mathcal{I}$. The superscript v indicates the translational components of the Jacobian i.e, the first three rows and will be dropped in the following as we only consider translational motions.

Suppose there is an object $\mathcal{O}$ somewhere in the robot's proximity, whose location with respect to the world frame, is described by the vector $\mathbf{r}_o$. At an instant k the translational motion of point $\mathcal{I}$ *towards* $\mathbf{r}_o$, denoted as $\delta\mathbf{x}_{o,k}^i$, is defined as

$$\delta\mathbf{x}_{o,k}^i = \hat{\mathbf{r}}_{io}^T \; \mathbf{J}_i \; \delta\mathbf{q}_k, \tag{8}$$

where $\mathbf{r}_{io} = \mathbf{r}_i - \mathbf{r}_o$ defines the relative vector between the object $\mathcal{O}$ and point $\mathcal{I}$ while $\hat{\mathbf{r}}_{io}$ denotes the corresponding normalized unit vector. Therefore, in order to prevent a potential collision between $\mathcal{I}$ and $\mathcal{O}$, an allowable motion constraint, defining the maximum displacement of $\mathcal{I}$, can be expressed as follows

$$\hat{\mathbf{r}}_{io}^T \; \mathbf{J}_i \; \delta\mathbf{q}_k \leq \|\mathbf{r}_{io}\|, \tag{9}$$

where $\|\mathbf{r}_{io}\|$ is the norm of the vector denoting the distance to the object. The inequality constraint (9) is enforced for l discretized points, at a suitable resolution, along the robot's body, leading to the following set of inequalities

$$\begin{bmatrix} \hat{\mathbf{r}}_{1o}^T \; \mathbf{J}_1 \\ \vdots \\ \hat{\mathbf{r}}_{lo}^T \; \mathbf{J}_l \end{bmatrix} \delta\mathbf{q}_k \leq \begin{bmatrix} \|\mathbf{r}_{1o}\| \\ \vdots \\ \|\mathbf{r}_{lo}\| \end{bmatrix}. \tag{10}$$

Equation (10) gives the set of instantaneous collision-free joint deviations for all discretized points on the robot's kinematic chain, considering obstacle $\mathcal{O}$. Extending this formulation to m obstacles in the robot's workspace, where for convenience the inequality constraints for the $j^{th} = 1 \ldots m$ obstacle is written as $\mathbf{J}_o^j \; \delta\mathbf{q}_k \leq \mathbf{r}_o^j$, results in the following expression

$$\begin{bmatrix} \mathbf{J}_o^1 \\ \vdots \\ \mathbf{J}_o^m \end{bmatrix} \delta\mathbf{q}_k \leq \begin{bmatrix} \mathbf{r}_o^1 \\ \vdots \\ \mathbf{r}_o^m \end{bmatrix}. \tag{11}$$

The above expression defines $(l \times m)$ constraints to ensure collision-free motions. In practice this can be reduced by neglecting obstacles where $\|\mathbf{r}_{io}\|$ is greater than a defined threshold.

In addition to ensuring collision-free motions, the robot is restricted by positional joint limits, of which the minimum and maximum are denoted as $\mathbf{q}_{min}$ and $\mathbf{q}_{max}$ respectively. To include these limits in our virtual fixture and thereby restrict displacements that would lead to a joint limit violation, the following constraints are defined

$$\begin{bmatrix} \mathbb{I}_n \\ -\mathbb{I}_n \end{bmatrix} \delta\mathbf{q}_k \leq \begin{bmatrix} \mathbf{q}_{max} - \mathbf{q}_k \\ \mathbf{q}_k - \mathbf{q}_{min} \end{bmatrix}, \tag{12}$$

where $\mathbb{I}_n$ is the $n \times n$ identity matrix. Finally, the linearization error increases as the joint displacement increases, as discussed in Section III-A. To limit this error a maximum linearization limit is imposed

$$\begin{bmatrix} \mathbb{I}_n \\ -\mathbb{I}_n \end{bmatrix} \delta\mathbf{q}_k \leq \begin{bmatrix} \delta\mathbf{q}_{max} \\ \delta\mathbf{q}_{max} \end{bmatrix}, \tag{13}$$

under the assumption that if this constraint is satisfied, (4) holds.

When combined, the inequality constraints given by (11), (12) and (13), form the $\mathcal{H}$-representation of a configuration-space polytope, i.e., the intersection of a finite number of half-spaces, denoted $\mathcal{Q}_k$, written as

$$\mathcal{Q}_k = \mathbf{A}_k \delta\mathbf{q}_k \leq \mathbf{b}_k, \tag{14}$$

where $\mathbf{A}_k$ contains the normals to the half-spaces and $\mathbf{b}_k$ contains the shifted distance from the origin along the normal for the obstacle, joint limit and linearization constraints, i.e., (11), (12) and (13), respectively. For a robot of n joints, $\mathcal{Q}_k$, is an $n-$dimensional polytope that encapsulates allowable joint displacements. A polytope may similarly be represented as a convex hull of its vertex set ($\mathcal{V}$-representation)

$$\mathcal{Q} = \{\mathbf{x} : \mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{y}_i \Big| \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i = 1\}, \tag{15}$$

where $\mathbf{y}_i$ is the i^{th} element of the vertex set and $\mathbf{x}$ denotes any point contained in $\mathcal{Q}$. The $\mathcal{V}$-representation of $\mathcal{Q}_k$ is a polytope with q vertices in n-dimensional space, whose matrix form is given as

$$\mathbf{Q}_v = \begin{bmatrix} \delta\mathbf{q}_1^v & \ldots & \delta\mathbf{q}_q^v \end{bmatrix} \tag{16}$$

In this work, the conversion between $\mathcal{V}$-representation and $\mathcal{H}$-representation is carried out using the double description method [31][1].

A linear transformation of a polytope results in another polytope. It follows that a linear transformation applied to $\delta\mathbf{q}_k$ is a convex combination of the same linear transformation applied to the vertex set. Thus by applying (2) to the vertex set a polytope, denoted $\mathcal{P}$, is formed for the task-space displacements, whose vertex set is described by a $3 \times p$ matrix given by

$$\boldsymbol{X}_v = \begin{bmatrix} \delta\mathbf{x}_1^v & \ldots & \delta\mathbf{x}_p^v \end{bmatrix}^T, \quad \text{where} \tag{17}$$

$$\delta\mathbf{x}_j^v = \mathbf{J}_n \; \delta\mathbf{q}_j^v \qquad \text{and} \quad p \leq q. \tag{18}$$

$\mathcal{P}$ defines a set of Cartesian displacements for the manipulator's end effector, at an instant k for which an inverse kinematic solution can be obtained that is collision free and within the positional joint limits.

[1] In this work, we use the C++ wrapper for Fukuda's cdd library available https://github.com/vsamy/eigen-cdd

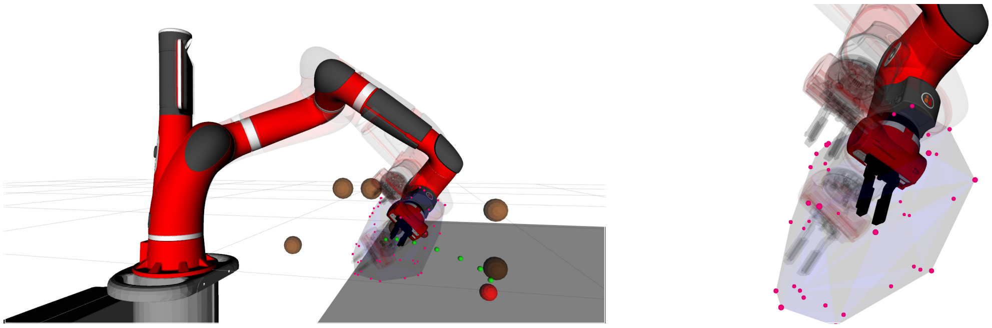


Fig. 2. Inverse kinematic solutions exploiting the off-line generated convex polytope. (Left) Global view of environment. (Right) A focused view of the feasible solutions the system can find for any desired Cartesian pose within polytope.

As in the case of the manipulability polytope, a performance metric, denoted as w_p, is obtained by calculating the enclosed volume. In this case, the enclosed volume inherently embeds information about the kinematic transmission ratio, the distance to joint limits and the proximity to obstacles. While the polytope is typically used to represent velocity capabilities, in this case over the linearization limits it gives a good representation of the amount of *free space* the end effector can move in as shown in Fig. 1. If w_p approaches zero it signifies that the end effector is losing mobility in one or more Cartesian directions. A library that calculates the constrained manipulability polytope in *ROS* is available here https://github.com/philip-long/constrained_manipulability.

C. Trajectory Generation

The trajectory is generated by formulating a nonlinear optimization problem using SNOPT [32] , where the decision variables are denoted by $\mathbf{u} \in \mathbb{R}^n$, given by

$$\mathbf{u} = \boldsymbol{\tau} - \hat{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}). \tag{19}$$

The decision variables define the evolution of the robot's state over the trajectory. The end effector pose at N is evaluated using the forward kinematics. This is known as the shooting method which is used to transcribe the optimal control problem, where the cost function is written as follows

$$\underset{\boldsymbol{u}_{1\dots N}}{\text{minimize}} \quad w_1 \frac{\|\mathbf{x}^d - \mathbf{x}(\mathbf{u})\|^2}{\|\mathbf{x}^d - \mathbf{x}_0\|^2} + w_2 \sum_{k=1}^{N} \frac{1}{\hat{w}_p^k(\mathbf{u}) + \lambda} \tag{20}$$
$$\text{subject to } \boldsymbol{\tau}_k^{min} \leq \mathbf{u}_k + \hat{\mathbf{c}}_k(\mathbf{q}, \dot{\mathbf{q}}) \leq \boldsymbol{\tau}_k^{max}, \quad k = 1, ..., N.$$

The first cost term $\|\mathbf{x}^d - \mathbf{x}\|^2$ denotes the Cartesian error between the resulting end-effector pose and the desired pose and is normalized with respect to the initial pose $\mathbf{x}_0$. The second cost term penalizes trajectories during which the volume of the constrained polytope, evaluated at timestep k, approaches zero, where

$$\hat{w}_p^k = \frac{w_p^k}{w_{max}} \tag{21}$$

and w_{max} is the maximum volume obtained off-line without considering obstacles or joint limits using the specified linearization constraints. λ is a regularization parameter used to prevent infinite costs. The volume of the constrained polytope is obtained by firstly writing the inequality constraints (14), secondly obtaining the joint-space vertex set (15) and finally transforming the vertex set to the Cartesian space using (18) and calculating the resulting enclosed volume with the *quickhull* algorithm [33]. Thus, the optimization procedure aims to obtain trajectories leading to the desired pose while penalizing trajectories close to joint limits, obstacles, or singular configurations. A successful optimization procedure outputs the robot state from an initial configuration to a desired configuration. The decision variables are initialized at zero, while $w_1 = 1000$, $w_2 = 0.1$. The trajectory will ensure that the volume of the Cartesian free-space polytope is maximized which in turn increases the free space around the end effector and finally as a by-product the manipulator's manipulability. Additionally, a set of joint polytopes in $\mathcal{H}$-representation i.e. $\mathbb{A} = \{\mathbf{A}_1 \dots \mathbf{A}_k \dots \mathbf{A}_N\}$ and $\mathbb{B} = \{\mathbf{b}_1 \dots \mathbf{b}_k \dots \mathbf{b}_N\}$ can be obtained. Therefore for a given joint configuration, these inequalities can be used on-line during the trajectory to obtain a feasible inverse kinematic (IK) solution. The convexity of the constraint equations allows collision-free, feasible IK solutions to be obtained rapidly with global convergence.

III. Experiments

In the following section, the concept is demonstrated using Rethink Robotics's Sawyer robot.[2]

A. Free Space Approximation

The use of the constrained manipulability polytope to measure the free space around the end effector is valid provided the linearization is approximately correct. Figure 3 shows the percentage linearization error defined as

$$\frac{10^2 \times \|(\mathbf{x} + \mathbf{J}_n \delta\mathbf{q}) - fk(\mathbf{q} + \delta\mathbf{q})\|}{\|\mathbf{J}_n \delta\mathbf{q}\|}, \tag{22}$$

where $\mathbf{x}$ is evaluated at the end effector.

For a random joint configuration $\mathcal{Q}$ is constructed using only (13), defining a hypercube in 7 dimensional space, with

[2] In this example objects are represented as a union of spheres. However, we have implemented our framework for object primitives and meshes using the FCL library [34] to compute object's normals, with minimal changes.

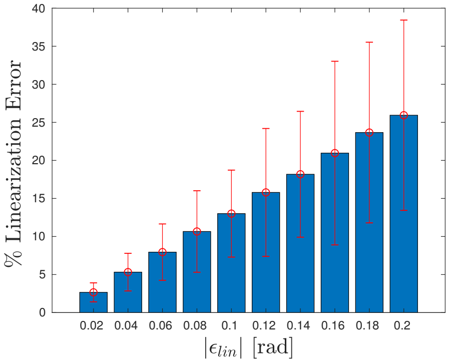


Fig. 3. Percentage error versus ϵ_{lin}, the joint deviation.

TABLE I
KDL VERSUS CP IN CLUTTERED ENVIRONMENT FOR $\approx$ 2500 IK CALLS

	Mean success time [ms]	% Success
CP	3.1650	96.7417%
KDL	26.7397	77.1140%

ϵ_{lin} defined as the maximum displacement for each joint. $\mathbf{Q}_v$, the $\mathcal{V}$- representation, contains $2^n = 128$ vertices. $\delta\mathbf{q}$ is obtained as the distance from the origin to a vertex; for instance taking $\epsilon_{lin} = 0.2$, one vertex would be defined as $\delta\mathbf{q}^v = [0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2]^T$.

Figure 3 shows that, as expected, the linearization error increases as the limit increases, therefore ϵ_{lin} needs to be judiciously chosen as a trade-off between free space for on-line IK solutions and total error.

B. On-line IK solutions

At a given configuration $\mathbf{q}_k$, a feasible IK solution can be obtained using $\mathbf{A}_k$ and $\mathbf{b}_k$ as linear inequality constraints, by formulating the following optimization problem

$$\begin{aligned} &\underset{\delta\mathbf{q_k}}{\text{minimize}} \| fk(\mathbf{q}_k + \delta\mathbf{q}_k) - \mathbf{x}_k^d \| \\ &\text{subject to } \mathbf{A}_k \delta\mathbf{q}_k \leq \mathbf{b}_k. \end{aligned} \tag{23}$$

Table I compares Orocos KDL Kinematics and Dynamics Library (KDL)[3] accessed via Moveit![4] and initially seeded at $\mathbf{q}_k$, with our proposed method that uses the convex polytope (CP). An IK solution is deemed successful if the error defined (23) is less than 10^{-5} (the default tolerance of KDL). The trajectory shown in Fig. 1 is used as a sample case. At each trajectory way-point, an IK solution is obtained for all vertices of the constrained polytope, leading to approximately 2500 tests.

KDL uses a gradient-descent method where the resulting solutions are tested for collisions and joint limits. Violations are rejected and a new search is initialized with a random configuration a maximum of 30 times. The repeated polling accounts for the large mean time of KDL. During our experiments, a one-shot successful solution with KDL without repolling lasted on average 0.311 ms, albeit with a low success rate (47.09%) due to the obstacles and joint limits. The CP method has a high success rate which increases to over 99% if the tolerance is increased to 1 mm. However, if desired poses are outside the free space polytope, CP is unable is find solutions with the requisite precision, due to the polytope constraints. Finally, we believe that the mean success time can be further decreased by using a specialized optimization procedure rather than SNOPT, which is typically used for large-scale sparse optimization problems.

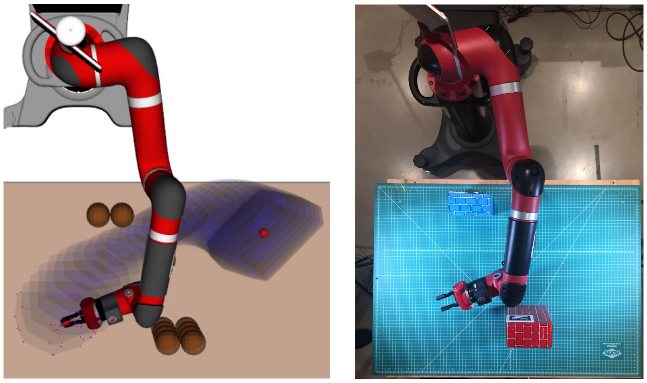


Fig. 4. (Left) The trajectory in the planning environment, the swept polytopes and the objects. (Right) Experimental setup for guarded teleoperation.

C. Guarded Teleoperation Mode

To show the concept of allowable motion, we propose a guarded teleoperation demonstration as shown in Fig. 4. The constrained polytope is generated by considering the two box objects and the table. The objective is to allow the teleoperator to move the end effector freely within the swept volume but threshold motion at the boundary. Hence, motion between overlapping polytopes is authorized while operator commands that would cause the system to exit the swept volume is forbidden.

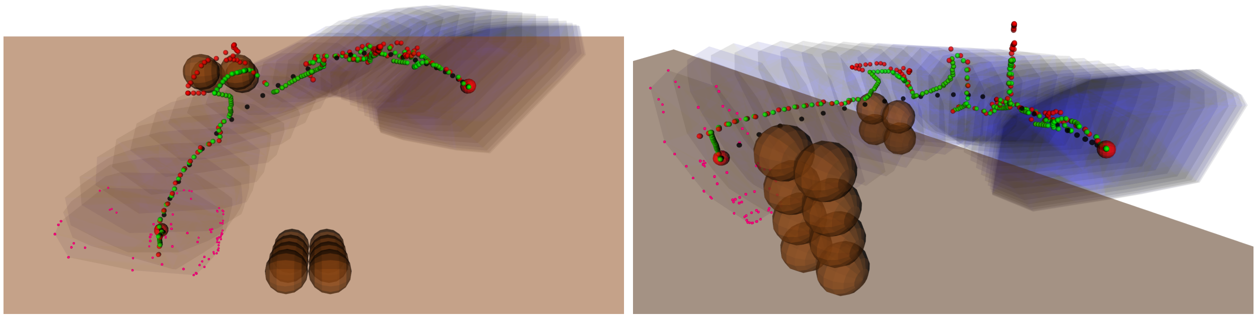
1) Control definition: The user command is defined as a Cartesian deviation denoted as $\delta\mathbf{x}^d$, which is added to the current position to define a desired pose $\mathbf{x}^d$. Prior to solving (23), the inequality equations must be defined. While each polytope is convex, the cumulative swept volume is non-convex, therefore simplifications by polytope merging is not possible. Suppose the robot's current position is given as $\mathbf{q}^c$, the following constraint equation is verified for each reference point in the trajectory, i.e., for $k = 1 \ldots N$

$$\mathbf{A}_k \left(\mathbf{q}^c - \mathbf{q}_k \right) \leq \mathbf{b}_k, \tag{24}$$

where $\mathbf{A}_k$ and $\mathbf{b}_k$ are taken from the off-line stored polytope set $\mathbb{A}$ and $\mathbb{B}$. If this constraint is respected, the current joint configuration lies within the joint space bounded by $\mathcal{Q}_k$ and hence the IK problem (23) can be solved to obtain $\delta\mathbf{q}_k$. This process is then repeated for every polytope at which (24) is satisfied, for the given desired pose, $\mathbf{x}^d$. The $\delta\mathbf{q}$ that corresponds to solution of minimum error is selected. This

[3] For comparison we are using the Newton method with joint limits available at http://www.orocos.org/wiki/orocos/kdl-wiki/

[4] http://moveit.ros.org



(a)

(b)

Fig. 5. Guarded teleoperation. Red, green and black represent the user's input, on-line IK solver's trajectory, and the planned trajectory, respectively.

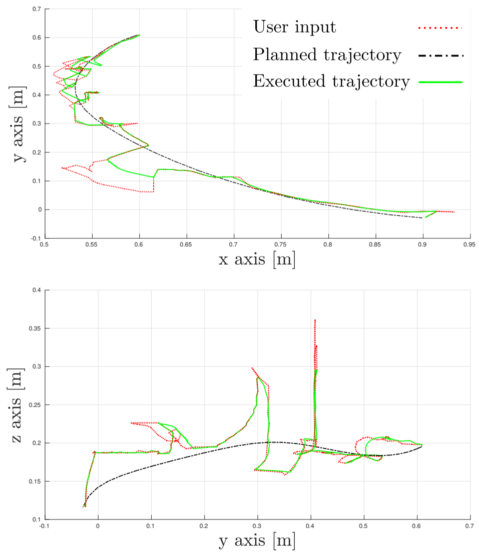


Fig. 6. Guarded teleoperation. Red, green and black represent the user's input, on-line IK solver's trajectory, and the planned trajectory, respectively.

procedure enables the user to move from one polytope to another while respecting the total swept volume bounds. The controller sends a desired joint position defined as $\mathbf{q}^d = \mathbf{q}_k + \delta\mathbf{q}_k$ to the trajectory interpolation function. Since the initial and final positions of the interpolated trajectory are guaranteed to lie within $\mathcal{Q}_k$, by definition a straight line (in joint space) also lies within the space. Finally, in order to aid the operator follow the reference trajectory, a stiffness controller is implemented, if the user input falls below a threshold, the desired position is changed as follows

$$\mathbf{q}^d = \mathbf{q}^c + \mathbf{K}_p \left(\mathbf{q}_{k+1} - \mathbf{q}^c \right), \tag{25}$$

where $\mathbf{K}_p$ is a diagonal gain matrix analogous to a spring constant. Hence, in the absence of operator input, the controller pushes the end effector along the reference trajectory.

2) Results & Discussion: The results of an experimental trial are shown in Figs. 5 and 6[5]. The black line shows the planned trajectory generated from the optimization procedure, the red line the user input and the green line the executed trajectory. Figure 5 shows the resulting curves, the Cartesian polytopes, and the obstacles in the planning environment. The user deliberately tries to collide with the objects and exit the polytope in the z direction during the experiment as shown by the red line user input. The polytope thresholds this motion and bounds the motion to the polytope's facets. Nevertheless the user is able to move freely between polytopes due to multiple IK solution procedure detailed in the previous section. While the polytope bounds are visualized in task space, the constraints are enforced in joint space ensuring that the whole robot respects obstacle bounds. Thus the end effector's orientation is also bounded, however in this work the free-space approximation of possible orientations is not shown to the user.

IV. Conclusion

In this paper, we have proposed a novel method of generating a virtual fixture for an end-effector operating in cluttered environments. We have employed the manipulability polytope with embedded obstacle and joint limit constraints. Within certain bounds, this polytope can represent the end effector's free-space motion. By directly maximizing the Cartesian volume in the trajectory generation algorithm, the allowable motions for a supervisory operator are increased. The proposed method facilitates the calculation of inverse kinematic solutions on-line.

The proposed method has some drawbacks. The absence of a closed-form solution for the polytope's volume means the gradient for optimization must be obtained by a finite-difference method. Moreover, each constraint adds a hyperplane into the polytope increasing the computational burden. This may be mitigated by ignoring objects beyond a threshold or considering each links' point of minimum distance rather than the discretized robot body.

While we have demonstrated the method with a guarded teleoperation example, we believe that this approach can be used for reactive control with autonomous robots. Future work will aim to develop a controller to allow the system to cope with dynamic events. Furthermore, we aim to incorporate the idea of allowable contacts and movable objects by defining contact candidates [35].

[5] A video is available at https://www.youtube.com/watch?v=oeqj-m25t9c

References

[1] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE transactions on robotics*, vol. 20, no. 6, pp. 967–977, 2004.

[2] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[3] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *1993 IEEE International Conference on Robotics and Automation, 1993. (ICRA)*, pp. 802–807, IEEE, 1993.

[4] C. Rösmann, F. Hoffmann, and T. Bertram, "Kinodynamic trajectory optimization and control for car-like robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5681–5686, IEEE, 2017.

[5] T. Kröger, "Opening the door to new sensor-based robot applicationsthe reflexxes motion libraries," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–4, IEEE, 2011.

[6] A. M. Zanchettin and P. Rocco, "Near time-optimal and sensor-based motion planning for robotic manipulators," in *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, pp. 965–970, IEEE, 2013.

[7] A. M. Zanchettin and P. Rocco, "Robust constraint-based control of robot manipulators: An application to a visual aided grasping task," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*, pp. 3634–3639, IEEE, 2016.

[8] W. Sun, L. G. Torres, J. Van Den Berg, and R. Alterovitz, "Safe motion planning for imprecise robotic manipulators by minimizing probability of collision," in *Robotics Research*, pp. 685–701, Springer, 2016.

[9] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[10] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4569–4574, IEEE, 2011.

[11] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE International Conference on Robotics and Automation*, pp. 489–494, IEEE, 2009.

[12] L. B. Rosenberg, "The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments.," tech. rep., Stanford Univ Ca Center for Design Research, 1992.

[13] S. A. Bowyer, B. L. Davies, and F. R. y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2014.

[14] T. Yamamoto, N. Abolhassani, S. Jung, A. M. Okamura, and T. N. Judkins, "Augmented reality and haptic interfaces for robot-assisted surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 8, no. 1, pp. 45–56, 2012.

[15] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 953–966, 2004.

[16] C. P. Quintero, M. Dehghan, O. Ramirez, M. H. Ang, and M. Jagersand, "Flexible virtual fixture interface for path specification in telemanipulation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5363–5368, IEEE, 2017.

[17] A. Kapoor, M. Li, and R. H. Taylor, "Constrained control for surgical assistant robots," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 231–236, May 2006.

[18] F. Dimeas, V. C. Moulianitis, C. Papakonstantinou, and N. Aspragathos, "Manipulator performance constraints in cartesian admittance control for human-robot cooperation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3049–3054, IEEE, 2016.

[19] N. Vahrenkamp and T. Asfour, "Representing the robot's workspace through constrained manipulability analysis," *Autonomous Robots*, vol. 38, pp. 17–30, Jan 2015.

[20] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Efficient inverse kinematics computation based on reachability analysis," *International Journal of Humanoid Robotics*, vol. 9, no. 04, p. 1250035, 2012.

[21] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics research: the first international symposium*, pp. 735–747, Mit Press Cambridge, MA, 1984.

[22] M.-J. Tsai, *Workspace geometric characterization and manipulability of industrial robots*. PhD thesis, The Ohio State University, 1986.

[23] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis.," in *Humanoids*, pp. 568–573, 2012.

[24] J.-P. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, 2006.

[25] T. Kokkinis and B. Paden, "Kinetostatic performance limits of cooperating robot manipulators using force-velocity polytopes," in *ASME Winter Annual Meeting*, pp. 151–155, 1989.

[26] R. Finotello, T. Grasso, G. Rossi, and A. Terribile, "Computation of kinetostatic performances of robot manipulators with polytopes," in *IEEE Int. Conf. on Robotics and Automation, 1998*, vol. 4, pp. 3241–3246, IEEE, 1998.

[27] T. Rasheed, P. Long, D. Marquez-Gamez, and S. Caro, "Tension distribution algorithm for planar mobile cable-driven parallel robots," in *Cable-Driven Parallel Robots*, pp. 268–279, Springer, 2018.

[28] S. Caron, Q.-C. Pham, and Y. Nakamura, "Zmp support areas for multicontact mobility under frictional constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, 2017.

[29] P. Long and T. Padir, "Evaluating robot manipulability in constrained environments by velocity polytope reduction," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 1–9, IEEE, 2018.

[30] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[31] K. Fukuda and A. Prodon, "Double description method revisited," in *Combinatorics and computer science*, pp. 91–111, Springer, 1996.

[32] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[33] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[34] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3859–3866, IEEE, 2012.

[35] A. Onol, P. Long, and T. Padir, "A comparative analysis of contact models in trajectory optimization for manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018.