Approximating robot reachable space using convex polytopes

Antun Skuric, Vincent Padois, David Daney

Approximating robot reachable space using convex polytopes

Antun Skuric, Vincent Padois, and David Daney

INRIA Sud-Ouest, Bordeaux, France
`firstname.lastname@inria.fr`
http://team.inria.fr/auctus/

Abstract. This paper presents an approach for approximating the reachable space of robotic manipulators based on convex polytopes. The proposed approach predicts the reachable space over a given time horizon based on the robot's actuation limits and kinematic constraints. The approach is furthermore extended to integrate the robot's environment, assuming it can be expressed in a form of linear constraints, and to account for the robot's link geometry.

The accuracy of the proposed method is evaluated using simulations of robot's nonlinear dynamics and it is compared against the cartesian space limits, usually provided by manufacturers in standard datasheets. The accuracy analysis results show that the proposed method has good performance for the time horizons up to $t_h \leq$250ms, encapsulating most of the simulated robot's reachable space while maintaining comparable volume. For a 7 dof robot, the method has an average execution time of 50ms, independent of the horizon time, potentially enabling real-time applications.

Keywords: Reachability analysis · Convex polytopes · Collaborative robotics.

1 Introduction

Collaborative robots, designed for safe physical interaction with humans, have a great potential to find their way in many spheres of industry, research and potentially become a part of our everyday lives. Apart from replacing rigid and dangerous manipulators in the industrial applications, their main potential is creating new robotic assistance scenarios leveraging high degree of physical human-robot interaction. Such scenarios demand not only many safety guarantees, but a high degree of operator's situation awareness both in terms of robot's behaviour and its physical abilities in real-time.

A metric unifying the robot's physical abilities and its movement capacity, important for safety and performance analysis, is called the robot's *reachable space* [1]. It represents the space of reachable robot's positions (ex. end-effector cartesian positions) for a certain time horizon, given different assumptions on its physical abilities. As robotic systems are highly non-linear and featuring

complex dynamics, the true reachable space of robots is very complex to characterise and calculate. Therefore, approximation techniques are necessary in order to yield practical solutions. Pereira and Athhoff[6] have developed an interval analysis based approach, later improved by Shrepp et al.[7], to approximate the human arm reachable space, modeled as a serial robotic manipulator, using a set of spheres and cylinders. However, in many cases the approximation of the reachable space using these shapes is impractical due to their non-linear nature.

In this paper, an approximation approach of the robot's reachable space using convex polytopes is proposed, leveraging several of their key characteristics. Convex polytopes can be represented as a set of linear inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$, which may be directly used with different optimisation techniques. Additionally, this set of linear inequalities can be intuitively extended with different environmental and user defined constraints. Since the reachable space is often low dimensional ($\leq$3D) these polytopes can be easily visualised, having the structure of a triangulated mesh, and potentially provide a human operator with valuable information about the robot's capacity. Furthermore, operations over polytopes such as Minkowski sum and intersection are well defined and efficient to calculate, enabling for the intuitive extension of the proposed method to account for robot's link geometry. Finally, the efficiency of polytope enumeration techniques for low dimensional polytopes has a potential to make this metric real-time capable.

Even though convex polytopes are widely used to characterise different metrics of robot's physical abilities[9]: force, acceleration and velocity, to the best of our knowledge they have not yet been used to characterise the reachable space of a robotic manipulator. The proposed approach is partially inspired by the works of Long et al. [5] on the constrained manipulability (velocity) polytopes.

Depending on whether the application requires an under or over approximation, or perhaps time-efficiency, the evaluation metrics of the accuracy of the reachable space approximation will differ significantly. In this paper, in addition to the execution time analysis, three accuracy measures are chosen to enable the evaluation of the proposed approach's performance and study its limitations.

The paper has the following structure. In the chapter 2 the formal definition of the approach is introduced, followed by the chapter 3, describing the procedure of benchmarking and the accuracy analysis. In the chapter 4, the results of the accuracy and execution time analysis of the approach are given and the chapter 5 discusses the main limitation as well as potential applications.

2   Convex polytope of reachable space

Robot dynamics in joint space can be expressed as

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \underbrace{C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_g(\boldsymbol{q})}_{\boldsymbol{\tau}_d} = \boldsymbol{\tau} \tag{1}$$

where M is the mass matrix, C is coriolis matrix and $\boldsymbol{\tau}_g$ is the gravity vector. $\boldsymbol{\tau}$ is the applied joint torque vector and $\boldsymbol{\tau}_d$ is the equivalent joint torque vector due to the coriolis and gravity effects.

For a robot with n actuated degrees of freedom (DOF), the robot's joint torque $\boldsymbol{\tau}$, velocity $\dot{\boldsymbol{q}}$ and joint angles $\boldsymbol{q}$ are n-dimensional vectors, limited by the robot's hardware[1]

$$\boldsymbol{\tau} \in [\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}], \quad \dot{\boldsymbol{q}} \in [\dot{\boldsymbol{q}}_{min}, \dot{\boldsymbol{q}}_{max}], \quad \boldsymbol{q} \in [\boldsymbol{q}_{min}, \boldsymbol{q}_{max}] \tag{2}$$

For a given moment in time k and for given robot configuration $\boldsymbol{q}_k$, the affine relationship between joint torques $\boldsymbol{\tau}$ and accelerations $\ddot{\boldsymbol{q}}_k$ can be expressed as

$$\ddot{\boldsymbol{q}}_k = M^{-1}(\boldsymbol{q}_k)(\boldsymbol{\tau} - \boldsymbol{\tau}_d) = M_k^{-1}(\boldsymbol{\tau} - \boldsymbol{\tau}_d) \tag{3}$$

Considering fixed joint acceleration $\ddot{\boldsymbol{q}}_k$ during given horizon length t_h, an approximation of the robot's joint velocity $\dot{\boldsymbol{q}}_{k+1}$ and position $\boldsymbol{q}_{k+1}$, at the end of horizon, can be calculated using numerical integration (forward Euler method)

$$\dot{\boldsymbol{q}}_{k+1} = M_k^{-1} t_h (\boldsymbol{\tau} - \boldsymbol{\tau}_d) + \dot{\boldsymbol{q}}_k, \qquad \boldsymbol{q}_{k+1} = M_k^{-1} \frac{t_h^2}{2}(\boldsymbol{\tau} - \boldsymbol{\tau}_d) + \dot{\boldsymbol{q}}_k t_h + \boldsymbol{q}_k \tag{4}$$

This linear numerical integration (4) considers the robot's dynamics (1), and the applied joint torque $\boldsymbol{\tau}$, fixed during the horizon time t_h. Such assumption is reasonable only for short horizon times t_h which in term represents a limitation of the proposed method.

The relationship between the m-dimensional task space velocity and acceleration of certain frame on the robot (for example end-effector frame) and the n-dimensional joint space equivalents is defined through the corresponding jacobian matrix $J(\boldsymbol{q})$ and its time derivative $\dot{J}(\boldsymbol{q})$.

$$\dot{\boldsymbol{x}}_k = J(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k = J_k \dot{\boldsymbol{q}}_k, \qquad \ddot{\boldsymbol{x}}_k = J_k \ddot{\boldsymbol{q}}_k + \dot{J}_k \dot{\boldsymbol{q}}_k \tag{5}$$

Finally, given the horizon of interest t_h, the predicted cartesian position $\boldsymbol{x}_{k+1}$ can then be expressed as

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \ddot{\boldsymbol{x}}_k \frac{t_h^2}{2} && + \dot{\boldsymbol{x}}_k t_h && + \boldsymbol{x}_k \\
&= J_k M_k^{-1} \frac{t_h^2}{2} \boldsymbol{\tau} \quad \underbrace{-J_k M_k^{-1} \frac{t_h^2}{2} \boldsymbol{\tau}_d}_{} \; \underbrace{+\dot{J}_k \dot{\boldsymbol{q}}_k \frac{t_h^2}{2} \quad + \dot{\boldsymbol{x}}_k t_h}_{} \quad + \boldsymbol{x}_k
\end{aligned}
\tag{6}
$$

$$\underbrace{\phantom{-J_k M_k^{-1} \frac{t_h^2}{2}\boldsymbol{\tau}_d}}_{\Delta\boldsymbol{x}_{k,dyn}} \underbrace{\phantom{+\dot{J}_k \dot{\boldsymbol{q}}_k \frac{t_h^2}{2} + \dot{\boldsymbol{x}}_k t_h}}_{\Delta\boldsymbol{x}_{k,vel}}$$

$$\underbrace{}_{\boldsymbol{x}_{k+1}^*}$$

Where the $\boldsymbol{x}_{k+1}^* = \Delta\boldsymbol{x}_{k,dyn} + \Delta\boldsymbol{x}_{k,vel} + \boldsymbol{x}_k$ is a predicted position vector based on the joint configuration $\boldsymbol{q}_k$, joint velocity $\dot{\boldsymbol{q}}_k$ (or cartesian velocity $\dot{\boldsymbol{x}}_k = J\dot{\boldsymbol{q}}_k$) and the joint torque $\boldsymbol{\tau}_d$ in the current time-step k, while the first term describes the influence of the applied joint torques $\boldsymbol{\tau}$.

The convex polytope of reachable cartesian space $\mathcal{P}_x$ can then be defined as a set of all the possible cartesian positions $\boldsymbol{x}_{k+1}$ at the end of the horizon t_h, achieved by any combination of joint torques $\boldsymbol{\tau}$ within robot's physical limitations (2), given the robot's current state $\dot{\boldsymbol{q}}_k$, $\boldsymbol{q}_k$ and $\boldsymbol{\tau}_d$.

---

[1] There is a coupling between the velocity and torque limits of an actuator related to its power. Yet, it is common practice to consider a subset of these limits such as torque and velocity can be chosen independantly one from the other.
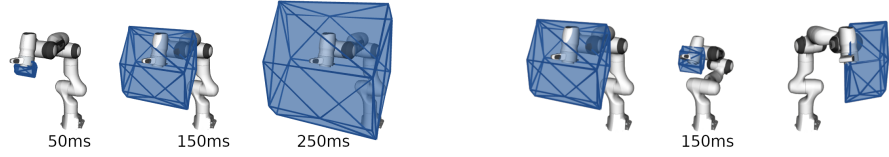
Fig. 1. Three images on the left show the comparison of the size of the reachable space polytope $\mathcal{P}_x$ of a *Franka Emika Panda*[3]robot's end-effector for 3 horizons (left to right) $t_h = 0.05$, 0.15 and 0.25s, for the same configuration. Whereas the three figures on the right show the constraining effect of the joint position limits on the $\mathcal{P}_x$ for a fixed horizon time (t_h=0.15s). Third image from the right shows the robot in its initial position $\boldsymbol{q} = [0, 0, 0, -\pi/2, 0, 3\pi/5, 0]$, close to the center of the joint ranges. In the second image form the right, the robot's joints q_1 and q_3 at their limits, $\boldsymbol{q} = [0, -1.59, 0, -2.9, 0, 3\pi/5, 0]$. On the last image, robot's joints q_0 and q_2 are at their limits $\boldsymbol{q} = [-2.72, 0, -2.72, -\pi/2, 0, 13\pi/5, 0]$, preventing the robot to rotate around the z axis in one direction.
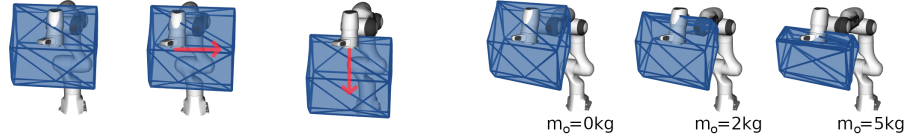

Fig. 2. Three images on the left show the shifting effect on the reachable space polytope $\mathcal{P}_x$ produced by certain cartesian velocity at the beginning of the horizon, from left to right: $\dot{\boldsymbol{x}}_k = [0, 0, 0]$, $[0, 1, 0]$ and $[0, 0, -1]m/s$. Three images on the right show the reducing effect on the polytope $\mathcal{P}_x$ by three different carried object masses (left to right) $m_o = 0$, 2 and 5kg. Horizon used is t_h=0.15s and the robot is in initial configuration.

$$
\begin{aligned}
\mathcal{P}_x = \{\boldsymbol{x}_{k+1} \in \mathbf{R}^m \quad | \quad & \boldsymbol{x}_{k+1} = J_k M_k^{-1} \frac{t_h^2}{2} \boldsymbol{\tau} + \boldsymbol{x}_{k+1}^*, \\
& \boldsymbol{\tau} \in [\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}], \\
M_k^{-1} t_h (\boldsymbol{\tau} - \boldsymbol{\tau}_d) + \dot{\boldsymbol{q}}_k & \in [\dot{\boldsymbol{q}}_{min}, \dot{\boldsymbol{q}}_{max}], \\
M_k^{-1} \frac{t_h^2}{2} (\boldsymbol{\tau} - \boldsymbol{\tau}_d) + \dot{\boldsymbol{q}}_k t_h + \boldsymbol{q}_k & \in [\boldsymbol{q}_{min}, \boldsymbol{q}_{max}]\}
\end{aligned}
\tag{7}
$$

Figures 1 and 2 showcase the influences of different horizon lengths t_h, robot constraints and robot movement ($\dot{\boldsymbol{q}}_k, \boldsymbol{\tau}_d$) on the shape of $\mathcal{P}_x$. Chapter 2.4 describes an approach to enumerate the reachable space polytope.

2.1   Influence of the carried object

If the robot is carrying a payload, an object with a mass m_o and inertia i_o attached to its end-effector, this object will have an influence on the robot's dynamics, modifying the mass matrix M, coriolis matrix C and the gravity vector $\boldsymbol{\tau}_g$ [4]. The augmented dynamical model of the robot will have reduced acceleration capabilities due to the added effort necessary for the object's movements.

---

[3] More information about the Panda robot at https://frankaemika.github.io/docs/
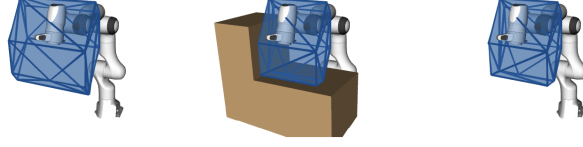
Fig. 3. Resulting reachable space polytope $\mathcal{P}_x$ for a *Franka Emika Panda* robot's end-effector when integrating the environmental constraints, horizon time used is t_h=0.15s. Environment is defined as $z \geq 0.5$m and $y \geq -0.2$m.

The figure 2 shows the influence of different object masses m_o, on the resulting reachable space polytope $\mathcal{P}_x$ for the *Franka Emika Panda* robot.

2.2   Integration of the environment

If the robot's environment can be defined as a set of convex inequalities

$$A_e \boldsymbol{x} \leq b_e \tag{8}$$

these constraints can be directly transformed to the constraints of the joint torque $\boldsymbol{\tau}$ using the equation (6)

$$A_e \boldsymbol{x}_{k+1} = A_e J_k M_k^{-1} \frac{t_h^2}{2} (\boldsymbol{\tau} - \boldsymbol{\tau}_d) + A_e \boldsymbol{x}_{k+1}^* \leq b_e \tag{9}$$

and then included in the $\mathcal{P}_x$ calculation (7)

$$
\begin{aligned}
\mathcal{P}_x = \{ \boldsymbol{x}_{k+1} \in \mathbf{R}^m \quad | \quad & \boldsymbol{x}_{k+1} = J_k M_k^{-1} \frac{t_h^2}{2} \boldsymbol{\tau} + \boldsymbol{x}_{k+1}^*, \\
& \boldsymbol{\tau} \in [\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}], \\
M_k^{-1} t_h (\boldsymbol{\tau} - \boldsymbol{\tau}_d) + \dot{\boldsymbol{q}}_k & \in [\dot{\boldsymbol{q}}_{min}, \dot{\boldsymbol{q}}_{max}], \\
M_k^{-1} \frac{t_h^2}{2} (\boldsymbol{\tau} - \boldsymbol{\tau}_d) + \dot{\boldsymbol{q}}_k t_h + \boldsymbol{q}_k & \in [\boldsymbol{q}_{min}, \boldsymbol{q}_{max}], \\
A_e J_k M_k^{-1} \frac{t_h^2}{2} (\boldsymbol{\tau} - \boldsymbol{\tau}_d) + A_e \boldsymbol{x}_{k+1}^* & \leq b_e \quad \}
\end{aligned}
\tag{10}
$$

Figure 3 showcases the influence of the environmental constraints on the shape of the $\mathcal{P}_x$.

2.3   Integration of robot's link geometry

If the robot link l_i is considered to be a straight line, to calculate the predicted reachable space of a robot's link l_i the polytope $\mathcal{P}_x$ can be calculated at the start $\mathcal{P}_{xs}$ and the end $\mathcal{P}_{xe}$ of the line l_i. Finally the reachable space of this idealised robot link l_i can then be calculated as the *convex-hull* (CH) of the polytopes $\mathcal{P}_{xs}$ and $\mathcal{P}_{xe}$

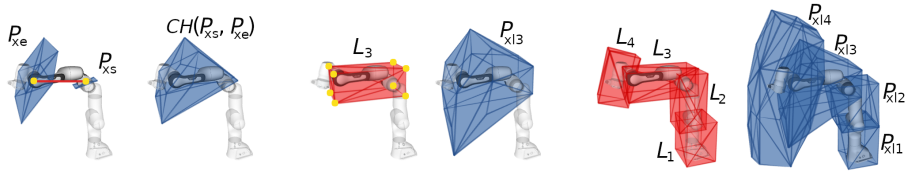$$\mathcal{P}_{xli} = CH(\mathcal{P}_{xs}, \ \mathcal{P}_{xe}) \tag{11}$$

Fig. 4. First four images show the construction of the reachable space polytope of the panda robot's link 3, $\mathcal{P}_{xl3}$. From left to right they show, robot's link 3 as a line segment (in red) with polytopes $\mathcal{P}_{xs}$ and $\mathcal{P}_{xe}$, their convex-hull, an example of the enveloping space $\mathcal{L}_3$ of the link 3 (in red), and the final polytope $\mathcal{P}_{xl3}$ calculated as a convex hull of the reachable space polytopes of each one of the 8 vertices (shown in yellow) of $\mathcal{L}_3$. Last two images further show an example of 4 enveloping spaces $\mathcal{L}_i$ for each one of the robot's links, and their reachable space polytopes $\mathcal{P}_{xli}$. Robot is in its initial configuration, and the horizon time used is 150ms.

If, instead of a straight line, a space captured by a robot's link l_i can be expressed as a convex set of constraints or a polytope

$$\mathcal{L}_i = \left\{ \boldsymbol{x}_l \mid A_l \boldsymbol{x}_l \leq b_l \right\} \tag{12}$$

the reachable space polytope $\mathcal{P}_{xli}$ can be extended to account for this link geometry by first finding the vertices of the polytope $\mathcal{L}$, followed by computing the reachable space polytope $\mathcal{P}_x$ for each one of the k vertices, and finally computing their convex hull.

$$\mathcal{P}_{xl,i} = CH \left(\mathcal{P}_{x1}, \ \dots \ , \mathcal{P}_{xk} \right) \tag{13}$$

By evaluating the polytope $\mathcal{P}_{xli}$ for each robot's link one can obtain an efficient approximation of the envelope of the reachable space of the entire robot, as shown on figure 4.

2.4   Enumerating reachable space polytope

The reachable space polytope $\mathcal{P}_x$ belongs to the family of over-determined problems

$$\mathcal{P}_x = \left\{ \boldsymbol{x} \in \mathbf{R}^m \ \mid \ \boldsymbol{x} = P\boldsymbol{\tau}, \quad A\boldsymbol{\tau} \leq \boldsymbol{b} \ \right\}, \quad \boldsymbol{\tau} \in R^n, \quad n \geq m \tag{14}$$

where the inequality constraint $A\boldsymbol{\tau} \leq \boldsymbol{b}$ encapsulates all the robot constraints (2) as well as the environmental constraints (9), and the matrix $J_k M_k^{-1} \frac{t_h^2}{2}$ becomes the projection matrix P. Vector $\boldsymbol{\tau}$ corresponds to the joint torque vector and the vector $\boldsymbol{x}$ corresponds to the difference $\boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^*$.

In order to fully define and simplify this formulation, the set of all the vertices ($\mathcal{V}$-representation) or all the half-spaces ($\mathcal{H}$-representation) of this polytope has to be found. Vertex representation of a polytope is a list of all of its vertices $\boldsymbol{v}_i$ and is commonly used for visualisation purposes $\mathcal{V} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \dots \}$, while half-space representation is defined as a list of inequalities $H\boldsymbol{x} \leq \boldsymbol{d}$, each corresponding to

one of the polytope facets. This matrix inequality equation is more suitable for applications involving different optimisation strategies.

The reachable space polytope (14) is a particular case of the projection of a high-dimensional polytope onto a lower dimensional space. Therefore, the set of constraints $A\boldsymbol{\tau} \leq \boldsymbol{b}$ form a $\mathcal{H}$-representation of a polytope $\mathcal{P}_\tau$ in the n-dimensional joint torque space.

$$\mathcal{P}_\tau = \{\boldsymbol{\tau} \in \mathbf{R}^n \quad | \quad A\boldsymbol{\tau} \leq \boldsymbol{b} \ \} \tag{15}$$

This polytope is then projected using the projection matrix P into the m-dimensional cartesian space, usually much lower dimensional, forming the reachable space polytope $\mathcal{P}_x$.

$$\mathcal{P}_x = \{\boldsymbol{x} \in \mathbf{R}^m \quad | \quad \boldsymbol{x} = P\boldsymbol{\tau}, \quad \boldsymbol{\tau} \in \mathcal{P}_\tau \ \} \tag{16}$$

The most straight forward way of approaching the enumeration of this polytope is to first enumerate the vertices of the polytope $\mathcal{P}_\tau$, project them to the cartesian space using the projection matrix P, followed by a Convex-Hull algorithm to extract the real vertices $\boldsymbol{v}_i$ of the polytope $\mathcal{P}_x$ from the projected points and its half-space representation.

In order to avoid enumerating all the vertices of the high-dimensional joint torque polytope $\mathcal{P}_\tau$, as the most complex operation of this polytope enumeration, *Iterative convex hull method*[10] (ICHM) is used, allowing to directly enumerate low-dimensional cartesian polytope $\mathcal{P}_x$. This iterative method is based on successively applying *Linear programming* (LP) and *Convex-Hull* (CH) algorithms and is able to find both vertex and half-space representation of this polytope at the same time. ICHM algorithm is defined for a family of sets:

$$\{ \ \boldsymbol{x} \in R^m \mid A\boldsymbol{x} = B\boldsymbol{y}, \quad C\boldsymbol{y} \leq \boldsymbol{d} \ \}, \qquad \boldsymbol{y} \in R^n, \quad m \leq n \tag{17}$$

Therefore the usage of the ICHM algorithm for enumerating the reachable space polytope is rather straight forward, by setting its matrix A to identity $A = I_{m\times m}$, matrix B becomes the projection matrix P, the inequality constraint $C\boldsymbol{y} \leq \boldsymbol{d}$ then becomes the set of the constraints $A\boldsymbol{\tau} \leq \boldsymbol{b}$.

3   Analysing the approximation accuracy

Defining metrics of interest for accuracy analysis is highly dependent of the applications these methods will be used on. If the application requires an under-approximation of the reachable set or an over-approximation, the metrics of interest will not be the same. In this paper, the algorithms accuracy is analysed using three different quantitative metrics, each one providing an insight in different characteristics and limitations of the method.

Additionally as this approximation method has a potential to be used for real-time applications, execution time analysis is performed as well.

In the extent of the proposed experiments, the considered reachable space is the reachable space of the robot's end effector.
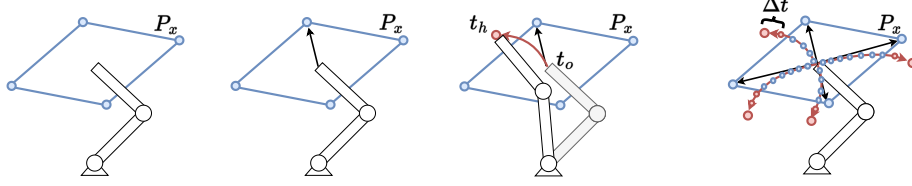
Fig. 5. Images show the steps (left to right) of the nonlinear simulation procedure on a simplified 2 link planar robot. First the polytope $\mathcal{P}_x$ of the end-effector is determined. For each vertex of the $\mathcal{P}_x$, joint torque τ generating the vertex is applied to the robot simulation (figures 2 and 3). All the robot's end-effector positions in all the simulation time steps Δt are saved for further analysis.

3.1   Metrics definition

To analyse the approximation accuracy of the reachable space polytope $\mathcal{P}_x$, a simple discrete non-linear robot dynamics simulation, subject to the constraints (2) is employed. Simulation is carried out by discretizing the equation (1) to determine the instantaneous joint acceleration $\ddot{\boldsymbol{q}}_k$ and followed by its numerical integration to determine joint velocity $\dot{\boldsymbol{q}}_{k+1}$ and position $\boldsymbol{q}_{k+1}$ in the step $k+1$.

$$\ddot{\boldsymbol{q}}_k = M^{-1}(\boldsymbol{q}_k)\left(\boldsymbol{\tau} - C(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)\dot{\boldsymbol{q}}_k + \boldsymbol{\tau}_g(\boldsymbol{q}_k)\right)$$
$$\dot{\boldsymbol{q}}_{k+1} = \ddot{\boldsymbol{q}}_k \Delta t + \dot{\boldsymbol{q}}_k \tag{18}$$
$$\boldsymbol{q}_{k+1} = \ddot{\boldsymbol{q}}_k \frac{\Delta t^2}{2} + \dot{\boldsymbol{q}}_k \Delta t + \boldsymbol{q}_k$$

Where Δt is a fixed simulation sampling time. The cartesian position in each step $\boldsymbol{x}_k$ is then calculated by evaluating the robot's forward kinematics

$$\boldsymbol{x}_k = f_{DK}(\boldsymbol{q}_k). \tag{19}$$

Each simulation roll-out considers a fixed joint torque vector $\boldsymbol{\tau}$ and simulates the robot dynamics through the horizon time t_h performing $N = t_h/\Delta t$ steps.

Each vertex $\boldsymbol{v}_i$ of reachable space polytope $\mathcal{P}_x$, calculated for a joint position $\boldsymbol{q}_k$ and a horizon t_h, is generated by a joint torque vector $\boldsymbol{\tau}_i$, belonging to a subset of vertices of the joint torque polytope $\mathcal{P}_\tau$.

To evaluate the accuracy of the proposed method, the difference in reached space produced by the constant liner model (polytope $\mathcal{P}_x$) and time varying non-linear model of the robot is evaluated, for the same set of applied joint torques. The applied joint torques chosen for the experiments are the joint torques $\boldsymbol{\tau}_i$ generating the vertices $\boldsymbol{v}_i$ of the reachable space polytope $\mathcal{P}_x$ calculated by the linear model.

As shown on figure 5, for each of the n_v vertices $\boldsymbol{v}_i$ of the polytope $\mathcal{P}_x$ the simulation (18) is performed and all the cartesian positions $\boldsymbol{x}_k$ of the robot in each of the $N = t_h/\Delta t$ sample times are retained for the further analysis.

$$\mathcal{X} = \{\underbrace{\boldsymbol{x}_{0,0} \ \ldots \ \boldsymbol{x}_{0,N}}_{\boldsymbol{\tau}_0}, \ \underbrace{\boldsymbol{x}_{1,0} \ \ldots \ \boldsymbol{x}_{1,N}}_{\boldsymbol{\tau}_1}, \ \ldots, \ \underbrace{\boldsymbol{x}_{n_v,0} \ \ldots \ \boldsymbol{x}_{n_v,N}}_{\boldsymbol{\tau}_{n_v}}\}$$

$$m_1 = \frac{n\{\circ\}}{n\{\circ\}+n\{\circ\}} \qquad m_2 = \frac{Vol\{R_1\}}{Vol\{P_x\}} \qquad m_3 = \frac{Vol\{P_x\}}{Vol\{R_2\}}$$
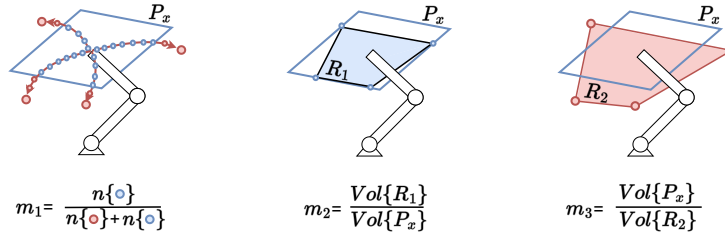
Fig. 6. Depiction of the defined quantitative metrics for a simple 2 link planar robot.

First metric, m_1, is defined as a ratio of number of the simulated cartesian robot positions $\boldsymbol{x}_k \in \mathcal{X}$ inside the polytope $\mathcal{P}_x$ with respect to the overall number of simulated positions.

$$m_1 = \frac{|\mathcal{X} \cap \mathcal{P}_x|}{|\mathcal{X}|} = \frac{|\mathcal{X} \cap \mathcal{P}_x|}{N \cdot n_v} \tag{20}$$

The metric m_1 gives an insight on how well the polytope $\mathcal{P}_x$ encapsulates the real reachable space determined with simulations. This metric might be used to asses the confidence in the polytope $\mathcal{P}_x$ for safety applications, the closer this metric is to 1, the less chance that the robot can exit the space bounded by $\mathcal{P}_x$.

The second metric, m_2, is defined as the ratio of the volumes of the polytope $\mathcal{P}_x$ and the convex hull CH of the points $\boldsymbol{x}_k \in \mathcal{X} \cap \mathcal{P}_x$,

$$R_1 = CH(\mathcal{X} \cap \mathcal{P}_x)$$

representing a subset of the points reached by the robot in the simulations found inside of the polytope $\mathcal{P}_x$.

$$m_2 = \frac{Vol\{R_1\}}{Vol\{P_x\}} \tag{21}$$

The metric m_2 gives an insight on how much volume of the polytope $\mathcal{P}_x$ is actually attainable by the robot, with respect to the simulations. This metric is designed to asses the quality of the approximation, if the score on this metric is low, this means that the large part of the $\mathcal{P}_x$ is not actually reachable.

The third metric, m_3, is defined as the ratio of the volumes of the polytope $\mathcal{P}_x$ and the convex hull CH of all the points $\boldsymbol{x}_k \in \mathcal{X}$,

$$R_2 = CH(\mathcal{X})$$

reached by the robot in the simulations.

$$m_3 = \frac{Vol\{P_x\}}{Vol\{R_2\}} \tag{22}$$

The metric m_3 gives insight about the ratio of the volumes of the polytope $\mathcal{P}_x$ and the simulated reachable space of the robot $CH(\mathcal{X})$. This metric is designed to asses the quality of the volume of the $\mathcal{P}_x$ as a metric. The further this metric from 1 the less confidence one has in the volume of the $\mathcal{P}_x$.
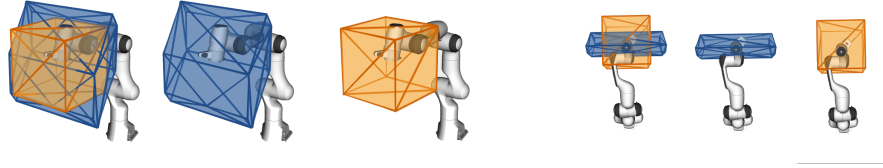
Fig. 7. Comparison of the end-effector reachable space approximation using the convex polytope $\mathcal{P}_x$ (blue) and the cube $\mathcal{C}_x$ (orange) for the *Franka Emika Panda* robot with the horizon time t_h=200ms. They are both evaluated in robot's initial position and the figures show two different views.

3.2   Benchmark comparison

To benchmark the accuracy of convex polytope approach $\mathcal{P}_x$, it is compared with the coarse approximation of the robot's end-effector reachable space using fixed cartesian velocity $\dot{\boldsymbol{x}}$ and acceleration $\ddot{\boldsymbol{x}}$ limits specified by robot manufacturer.

$$\ddot{\boldsymbol{x}} \in [\ \ddot{\boldsymbol{x}}_{min}, \ddot{\boldsymbol{x}}_{max}], \qquad \dot{\boldsymbol{x}} \in [\ \dot{\boldsymbol{x}}_{min}, \dot{\boldsymbol{x}}_{max}] \tag{23}$$

The cartesian space limitations (23) assume that the robot's acceleration $\ddot{\boldsymbol{x}}$ and velocity $\dot{\boldsymbol{x}}$ capacity is constant, which is a not true since the robot's capacity depends highly on robot's joint configuration $\boldsymbol{q}$ and velocity $\dot{\boldsymbol{q}}$ [2][9]. However, they are very convenient and widely used in many applications. A simple convex space $\mathcal{C}_x$ (cube in 3d) of reachable end-effector positions given the cartesian space limitations (23) can be calculated

$$\mathcal{C}_x = \{\ \boldsymbol{x}_{k+1} \in R^m \mid \boldsymbol{x}_{k+1} = \ddot{\boldsymbol{x}}_k \frac{t_h^2}{2} + \dot{\boldsymbol{x}}_k t_h + \boldsymbol{x}_k$$
$$\ddot{\boldsymbol{x}}_k \in [\ \ddot{\boldsymbol{x}}_{min}, \ddot{\boldsymbol{x}}_{max}] \tag{24}$$
$$\ddot{\boldsymbol{x}}_k t_h \in [\ \dot{\boldsymbol{x}}_{min}, \dot{\boldsymbol{x}}_{max}]\ \}$$

Figure 7 shows the visual comparison of the two approaches: $\mathcal{P}_x$ and $\mathcal{C}_x$ calculated for one pose of the *Franka Emika Panda* robot.

4   Results

The proposed analysis of the reachable space polytope accuracy is illustrated on the simulation of the collaborative robotic manipulator *Franka Emika Panda* with 7 degrees of freedom. Robot modeling and kinematics as well as simulations of robot dynamics are built using the python implementation of the ROBOTICS-TOOLBOX[3] package.

The evaluation of the reachable space polytope $\mathcal{P}_x$ and $\mathcal{C}_x$ is carried out using the ICHM algorithm's efficient python implementation within the PYCAPAC-ITY[8] package. The ICHM's approximation accuracy used for all the evaluations is δ=1mm.

Eight different horizon lengths t_h are chosen for the experiments

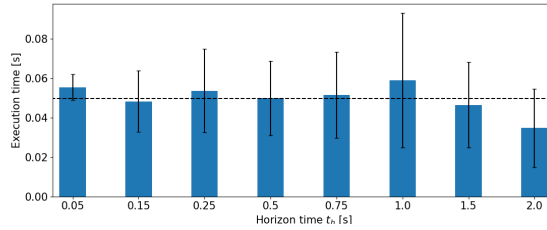$$t_h = [0.05s,\ 0.15s,\ 0.25s,\ 0.5s,\ 0.75s,\ 1.0s,\ 1.5s,\ 2.0s]$$

Fig. 8. Execution time of the polytope $\mathcal{P}_x$ enumeration using ICHM algorithm. Average and standard deviation calculated over 1000 executions for each of the 8 horizon lengths.

and the simulation sample time has been set to Δt=5ms.

The proposed method implementation as well as the code used for the experiments is fully open-source and can be found in the gitlab repository[4]. The gitlab repository additionally contains a Robot operating system (ROS) implementation of the proposed method for more interactive evaluation.

All the simulations are run on a computer with 1.90GHz Intel i7-8650U processor and 32Gb of RAM memory.

4.1   Execution time

Figure 8 shows the polytope $\mathcal{P}_x$ execution time averaged over 1000 random robot configurations for each of the 8 horizon lengths. From the figure it can be seen that the execution time is relatively consistent through all the horizon lengths, with the average value of around 50ms. The constant execution time is expected because the horizon length t_h does not have a big influence on the geometrical complexity of the polytope (number of vertices and faces) but its scale, and the scale itself does have a big influence the execution time.

Figure 9 shows the ICHM execution time for the 8 horizon lengths with different numbers of environmental constraints $A_e \boldsymbol{x} \leq \boldsymbol{b}_e$. All the environmental constraints are generated randomly and all the results are averaged over 1000 random robot configurations for each horizon length t_h and all 4 different numbers of environmental constraints: 0 (no environment constraints), 10, 100, 500 and 1000. Results show that up to 10 added environmental constraints the execution time of the method does not change significantly, having the average execution time around 50ms. With 100 environmental constraints the average execution time increases around 20% to around 70ms, for 500 the average execution is 100ms and with 1000 constraints the average execution time more than doubles, to 150ms.

The execution time analysis of the ICHM algorithm has shown that the average polytope $\mathcal{P}_x$ enumeration takes between 50 and 150ms, which opens many doors for potential real-time applications.

The average execution time of the nonlinear simulation for t_h=50ms and Δt=5ms is around 60s, and it grows to about 240s for t_h=2s. The cube $\mathcal{C}_x$ average execution time stays constant for all horizon lengths at around 2ms.

---

[4] https://gitlab.inria.fr/auctus-team/people/antunskuric/reachable_space/
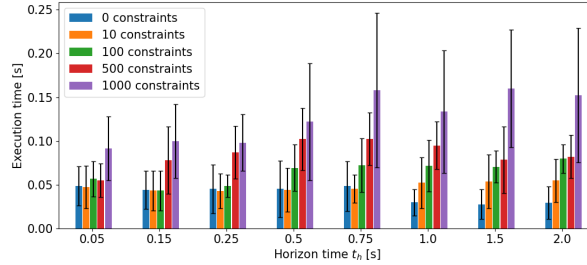
Fig. 9. Average execution time and standard deviation calculated over 1000 executions for each of the 8 horizon lengths, with different number of added environmental constraints, ranging from 0 to 1000.

4.2   Accuracy analysis

Figure 10 shows the evolution of the three metrics m_1, m_2 and m_3 with respect to the horizon length t_h for the $\mathcal{P}_x$ and $\mathcal{C}_x$, averaged over 100 random robot configurations.

The first graph on the figure 10 shows the metric m_1. It can be seen that the ratio of simulated robot positions $\boldsymbol{x}_k \in \mathcal{X}$ inside the polytope $\mathcal{P}_x$ decreases with the increasing horizon length t_h, while it increases for the $\mathcal{C}_x$. For larger horizon lengths the robot is able to move far from the initial joint configuration $\boldsymbol{q}$ at the beginning of the horizon, making the assumption of linearity of the robot model, made for calculating $\mathcal{P}_x$, largely inaccurate. The approach $\mathcal{C}_x$, on the other hand, improves with the increase in horizon length, since the space bounded by the cube $\mathcal{C}_x$ increases linearly with the horizon length t_h, at some point it encapsulates the whole workspace of the robot. However, it can be seen that even though the ratio m_1 decreases, the polytope $\mathcal{P}_x$ still contains the most ($\geq 60\%$) of the simulated robot's reached positions $\boldsymbol{x}_k \in \mathcal{X}$. Finally, it can also be seen that the variance of the m_1 increases considerably, lowering the confidence of the polytope $\mathcal{P}_x$ for longer horizons.

Evolution of the metric m_2 is shown on the middle graph of the figure 10 . It can be seen that the volume of all the reached robot positions inside polytope $\mathcal{P}_x$ and $\mathcal{C}_x$ decreases considerably with the increase of the horizon time t_h. For the horizon times under $t_h \leq 0.25$s the majority ($\geq 50\%$) of the reachable space polytope $\mathcal{P}_x$ has been reached by the robot simulations. However for the longer horizon times $t_h \geq 0.5$s the reached volume ratio m_2 drops under 20% of the total volume of the $\mathcal{P}_x$. In the case of $\mathcal{C}_x$, the majority ($\geq 50\%$) of the volume of the $\mathcal{C}_x$ is never reached by the robot for all tested horizon lengths t_h.

The right graph of the figure 10 shows the evolution of the metric m_3. The graph shows that the volume of the polytope $\mathcal{P}_x$ and the cube $\mathcal{C}_x$ becomes several times higher than the volume of the R_2 with the increase in the horizon time t_h. For the horizon time of t_h=2s the average volume of $\mathcal{P}_x$ is 12 times larger than the volume of R_2 and the volume $\mathcal{C}_x$ is more than 50 times larger than R_2. At the same time the average reached volume R_1 inside the $\mathcal{P}_x$ and $\mathcal{C}_x$ consists of
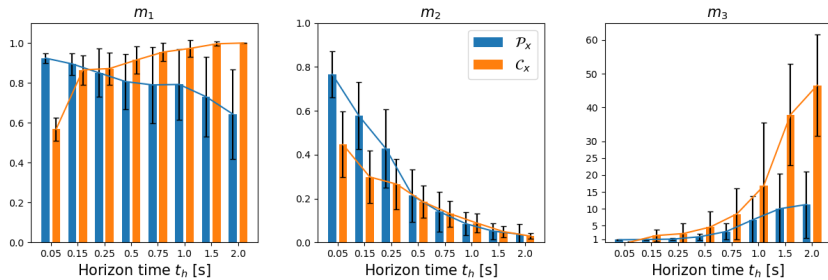
Fig. 10. Accuracy measures calculated for $\mathcal{P}_x$ and $\mathcal{C}_x$, averaged over 100 random poses of the *Franka Emika panda* robot for each of 8 horizon lengths.

under 10% of their volume (metric m_2), which potentially makes both polytope $\mathcal{P}_x$ and $\mathcal{C}_x$ impractical for many of applications.

However, for the shorter horizon times $t_h \leq 0.25$s the ratio m_3 is very close to 1 for $\mathcal{P}_x$, which means that the reachable space polytope's $\mathcal{P}_x$ volume corresponds well to the actual reached volume of the robot R_2.

Overall, this numerical analysis shows that the reachable space approximation based on the convex polytope $\mathcal{P}_x$ works reasonably well for low horizon times $t_h \leq 250$ms, while for the longer horizon times $t_h > 250$ms the assumption of the linearity of the system produces a large error in the reachable space estimation, producing the polytope $\mathcal{P}_x$ with volume several times higher than the real reachable space of the robot (shown by the metric m_3), most of which is not reachable by the robot at all (showed by the metric m_2). Finally, the results show that the polytope $\mathcal{P}_x$, even though a coarse approximation, has better performance across all the defined metrics then the $\mathcal{C}_x$ based on the constant cartesian limits, at the same time having smaller average volume for the all the horizon times. These results remind, once again, that robot's cartesian capacity is not constant and by modeling their evolution, even in a coarse manner, a substantial improvement in the estimation accuracy can be reached.

5   Discussion

Approximating the reachable space of the robot using convex polytopes $\mathcal{P}_x$ is an efficient and relatively fast way of gaining information about the potential robot's position for the horizon of interest. Even though this approximation approach is neither an under or an over approximation of the true reachable space of the robot, as shown in the numerical analysis, for shorter horizon times $t_h \leq 250$ms this metric is reasonably precise. Due to its efficiency, easy and intuitive integration of different environmental constraints, possibility to consider the robot's link geometry and the fact that its output has the form of a convex set of inequality constraints (convex polytope $\mathcal{P}_x$), it has a potential to be a useful tool for robot performance and safety analysis. However there are several important limitations of this approach.

5.1 Main limitations

First of all, the reachable space of a robot is highly non-convex and non-linear. As shown in the numerical analysis section 4, approximating this space with convex polytopes is relatively precise only for small horizon times $t_h \leq 250$ms.

Secondly, robot dynamics is highly non-linear and time variant as well. When calculating the reachable space polytope $\mathcal{P}_x$, as formulated in section 2, the robot model is linearised in the robot's joint configuration $\boldsymbol{q}_k$ and velocity $\dot{\boldsymbol{q}}_k$ at the beginning of the horizon and this linearised model is considered constant during the horizon length. The consideration of constant robot model and its linearity, is only valid for small robot displacements around the current robot state, which is again a strong assumption and only valid for short horizon times.

Finally, the polytope formulation from the section 2, considers only constant joint torques $\boldsymbol{\tau}$ applied to the robot during the whole horizon length t_h. The polytope $\mathcal{P}_x$ calculates the joint torque vectors $\boldsymbol{\tau}$ that will produce the largest possible displacements in cartesian space given the robot and environment constraints. This calculation is based on the robot's linearised model at the beginning of the horizon, which in term, once again, limits the precision of the approximation to the small displacements around the initial robot state, preventing longer horizon lengths t_h, as confirmed in the analysis section 4.

5.2 Potential applications

One promising field of applications for the polytope based reachable space metric is the human-robot interaction domain. The polytope is essentially a triangulated mesh and can be easily visualised, which could be an easy and intuitive way to give insight to the human operators about robot's real-time capabilities. It has a potential to be used for the haptic control applications, as human operators might not be always aware of how close the robot is to the limitation of its capabilities or what is the possible space the robot can be in while teleoperating. Furthermore, this metric could be potentially used to to increase human operators awareness about the robot's state during the co-manipulation by visualising the polytope $\mathcal{P}_x$ in real-time and even potentially increase the interaction safety.

The numerical analysis has shown that for shorter horizon times $t_h \leq 250$ms the volume of the polytope $\mathcal{P}_x$ is comparable to the volume of the simulated reached space R_2, as shown on the figure 10. The volume of the $\mathcal{P}_x$ could potentially be used as a performance metric for robotic workspace analysis and for applications such as robot design.

Finally, as the polytope $\mathcal{P}_x$ can be represented as a set of convex inequalities $H\boldsymbol{x} \leq \boldsymbol{d}$, this metric could potentially be used as a set of constraints for robot control. For example in case of the Model Predictive Control (MPC) where, in many cases, robot's dynamics is already considered linear and constant during given horizon time, which corresponds well with the assumptions of the polytope $\mathcal{P}_x$ formulation.

6   Conclusion

In this paper a convex polytope based robot's reachable space approximation metric is introduced. The metric predicts the robot's reachable space for a given time horizon based on the robot's actuation capabilities and its kinematic constraints. The proposed metric can intuitively integrate the environmental constraints as well as the robot's link geometry. With the execution time of around 50ms, for standard 7dof collaborative robot, it has a potential to be used for real-time applications.

However, the metric relies on a linearised robot model, which limits its use to shorter horizon times. The analysis has shown that the accuracy of the approximation decreases significantly for horizon times longer than 250ms.

References

1. Althoff, M., Krogh, B.H.: Reachability analysis of nonlinear differential-algebraic systems. IEEE Transactions on Automatic Control **59**(2), 371–383 (2014). https://doi.org/10.1109/TAC.2013.2285751
2. Bowling, A., Khatib, O.: The dynamic capability equations: a new tool for analyzing robotic manipulator performance. IEEE Transactions on Robotics **21**(1), 115–123 (2005). https://doi.org/10.1109/TRO.2004.837243
3. Corke, P., Haviland, J.: Not your grandmother's toolbox–the robotics toolbox reinvented for python. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 11357–11363. IEEE (2021)
4. Hamad, M., Mansfeld, N., Abdolshah, S., Haddadin, S.: The role of robot payload in the safety map framework. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 195–200 (2019). https://doi.org/10.1109/IROS40897.2019.8968022
5. Long, P., Padir, T.: Evaluating robot manipulability in constrained environments by velocity polytope reduction. In: 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids). pp. 1–9 (2018). https://doi.org/10.1109/HUMANOIDS.2018.8624962
6. Pereira, A., Althoff, M.: Calculating human reachable occupancy for guaranteed collision-free planning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4473–4480 (2017). https://doi.org/10.1109/IROS.2017.8206314
7. Schepp, S.R., Thumm, J., Liu, S.B., Althoff, M.: Sara: A tool for safe human-robot coexistence and collaboration through reachability analysis. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 4312–4317 (2022). https://doi.org/10.1109/ICRA46639.2022.9811952
8. Skuric, A.: A real-time capable task-space capacity calculation python package, https://pypi.org/project/pycapacity
9. Skuric, A., Padois, V., Daney, D.: On-line force capability evaluation based on efficient polytope vertex search. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 1700–1706 (2021). https://doi.org/10.1109/ICRA48506.2021.9562050
10. Skuric, A., Padois, V., Rezzoug, N., Daney, D.: On-line feasible wrench polytope evaluation based on human musculoskeletal models: An iterative convex hull method. IEEE Robotics and Automation Letters **7**(2), 5206–5213 (2022). https://doi.org/10.1109/LRA.2022.3155374