



HAL
open science

Approximating robot reachable space using convex polytopes

Antun Skuric, Vincent Padois, David Daney

► **To cite this version:**

Antun Skuric, Vincent Padois, David Daney. Approximating robot reachable space using convex polytopes. 2022. hal-03719885v1

HAL Id: hal-03719885

<https://inria.hal.science/hal-03719885v1>

Preprint submitted on 11 Jul 2022 (v1), last revised 29 Nov 2022 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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, if one can be expressed in a form of linear constraints.

The accuracy of the proposed method is evaluated using dynamic simulations of the robot with respect to the non-linear robot simulation and is compared to results obtained using the cartesian space limits, usually provided by manufacturers in their standard datasheet.

The accuracy analysis results show that the polytope based method has good performance for the time horizons up to $t_h \leq 250\text{ms}$, 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[5] have used interval analysis tools to approximate the human arm reachable space, modeled as a serial robotic manipulator, using a set of spheres and cylinders. This work has been extended by Shrepp et al.[9] making this approach more computationally efficient. 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 is proposed using convex polytopes. Convex polytopes have several characteristics benefiting reachable space analysis. They can be represented as a set of linear inequalities $A\mathbf{x} \leq \mathbf{b}$ and in that way they can 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 providing a human operator with valuable information about the robot’s capacity. Finally, polytope enumeration techniques are very efficient for low dimensional polytopes having the potential to make this metric real-time capable. Even though convex polytopes are widely used to characterise different metrics of robot’s physical abilities[7]: 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.

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

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

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 $\dot{\mathbf{q}}$ and joint angles \mathbf{q} are n -dimensional vectors, limited by the

robot's hardware¹

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

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

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

For a given horizon t_h , the robot's joint velocity $\dot{\boldsymbol{q}}_{k+1}$ and position \boldsymbol{q}_{k+1} can then be calculated by integrating the acceleration

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

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 $\dot{J}(\boldsymbol{q})$. For a given instant k the relationship becomes

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

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

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

$\underbrace{\hspace{15em}}_{\boldsymbol{x}_{k+1}^*}$

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 possible cartesian positions \boldsymbol{x}_{k+1} given the horizon of interest t_h , robot's physical limitations (2) and robot's joint velocity $\dot{\boldsymbol{q}}_k$ and position \boldsymbol{q}_k in the instant k .

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 independently one from the other.

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

Figures 1, 2 and 3 showcase the influences of different horizon lengths t_h , robot constraints and robot movement $(\dot{\mathbf{q}}_k, \boldsymbol{\tau}_d)$ on the shape of \mathcal{P}_x .



Fig. 1. Reachable space polytope \mathcal{P}_x for a *Franka Emika Panda* robot's end-effector for three different horizon lengths (from left to right) $t_h = 0.05, 0.15$ and 0.25 seconds



Fig. 2. Reachable space polytope \mathcal{P}_x for a *Franka Emika Panda* robot's end-effector for 3 configurations ($t_h=0.15$ s). On the left, the robot is in its initial position $\mathbf{q} = [0, 0, 0, -\pi/2, 0, 3\pi/5, 0]$, close to the center of the joint ranges. In the middle figure the robot has joints q_1 and q_3 are at their limits, $\mathbf{q} = [0, -1.59, 0, -2.9, 0, 3\pi/5, 0, 0]$. On the right, robot's joints q_0 and q_2 are at their limits $\mathbf{q} = [-2.72, 0 - 2.72, -\pi/2, 0, 13\pi/5, 0, 0]$.

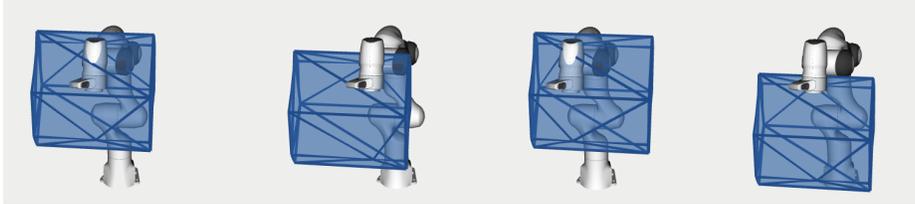


Fig. 3. Reachable space polytope \mathcal{P}_x for a *Franka Emika Panda* robot's end-effector for 4 different end-effector velocities $\dot{\mathbf{x}}_k = J \dot{\mathbf{q}}_k$ (from left to right) $\dot{\mathbf{x}}_k = [0, 0, 0], [0, -1, -0.5], [0, 1, 0]$ and $[0, 0, -1]m/s$.

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 the robot's 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 τ_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.

The figure 4 shows the influence of different object's masses m_o , on the resulting reachable space polytope \mathcal{P}_x for the *Franka Emika Panda* robot.

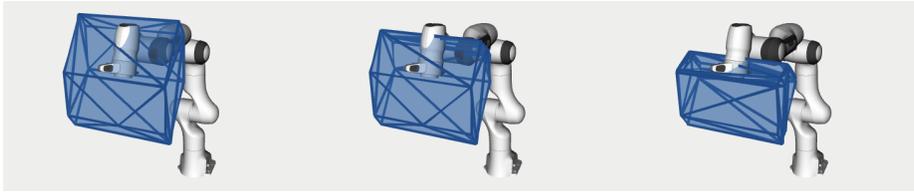


Fig. 4. Reachable space polytope \mathcal{P}_x for a *Franka Emika Panda* robot for three different carried object masses (left to right) $m_o = 0, 2$ and 5kg , horizon time used $t_h=0.15\text{s}$.

2.2 Integration of the environment

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

$$A_e \mathbf{x} \leq b_e \quad (8)$$

these constraints can be directly transformed to the constraints of the joint torque τ using the equation (6)

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

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

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

Figure 5 showcases the influence of the environmental constraints on the shape of the \mathcal{P}_x .

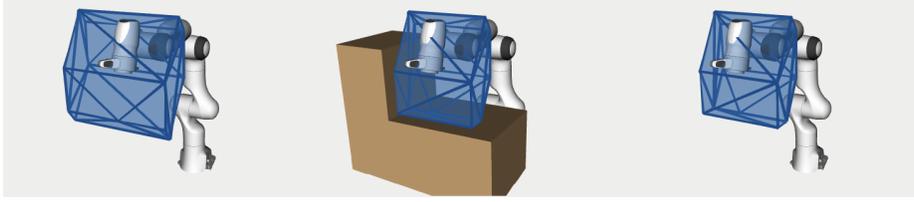


Fig. 5. Resulting reachable space polytope \mathcal{P}_x for a *Franka Emika Panda* robot's end-effector when integrating the environmental constraints, horizon time used is $t_h=0.15s$. Environment is defined as $z \geq 0.5m$ and $y \geq -0.2m$.

2.3 Enumerating reachable space polytope

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

$$\mathcal{P}_x = \{ \mathbf{x} \in \mathbf{R}^m \mid \mathbf{x} = P\boldsymbol{\tau}, \quad A\boldsymbol{\tau} \leq \mathbf{b} \}, \quad \boldsymbol{\tau} \in \mathbf{R}^n, \quad n \geq m \quad (11)$$

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

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 \mathbf{v}_i and is commonly used for the visualisation purposes $\mathcal{V} = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots\}$, while half-space representation is defined as a list of inequalities $H\mathbf{x} \leq \mathbf{d}$, 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 (11) is a particular case of the projection of a high-dimensional polytope onto a lower dimensional space. Therefore, the set of constraints $A\boldsymbol{\tau} \leq \mathbf{b}$ form a \mathcal{H} -representation of a polytope \mathcal{P}_τ in the n -dimensional joint torque space.

$$\mathcal{P}_\tau = \{ \boldsymbol{\tau} \in \mathbf{R}^n \mid A\boldsymbol{\tau} \leq \mathbf{b} \} \quad (12)$$

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 .

$$\mathcal{P}_x = \{ \mathbf{x} \in \mathbf{R}^m \mid \mathbf{x} = P\boldsymbol{\tau}, \quad \boldsymbol{\tau} \in \mathcal{P}_\tau \} \quad (13)$$

The most straight forward way of approaching the enumeration of this polytope is to first enumerate the vertices of the polytope \mathcal{P}_τ , project them to the cartesian space using the projection matrix P , followed by a Convex-Hull algorithm to extract the real vertices \mathbf{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}_τ , as the most complex operation of this polytope enumeration, *Iterative convex hull method*[8] (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:

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

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 $C\mathbf{y} \leq \mathbf{d}$ then becomes the set of the constraints $A\boldsymbol{\tau} \leq \mathbf{b}$.

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.

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 $\ddot{\mathbf{q}}_k$ and followed by the its integration to determine joint velocity $\dot{\mathbf{q}}_{k+1}$ and position \mathbf{q}_{k+1} in the future step $k + 1$.

$$\begin{aligned} \ddot{\mathbf{q}}_k &= M^{-1}(\mathbf{q}_k) (\boldsymbol{\tau} - C(\mathbf{q}_k, \dot{\mathbf{q}}_k)\dot{\mathbf{q}}_k + \boldsymbol{\tau}_g(\mathbf{q}_k)) \\ \dot{\mathbf{q}}_{k+1} &= \ddot{\mathbf{q}}_k \Delta t + \dot{\mathbf{q}}_k \\ \mathbf{q}_{k+1} &= \ddot{\mathbf{q}}_k \frac{\Delta t^2}{2} + \dot{\mathbf{q}}_k \Delta t + \mathbf{q}_k \end{aligned} \tag{14}$$

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

$$\mathbf{x}_k = f_{DK}(\mathbf{q}_k). \tag{15}$$

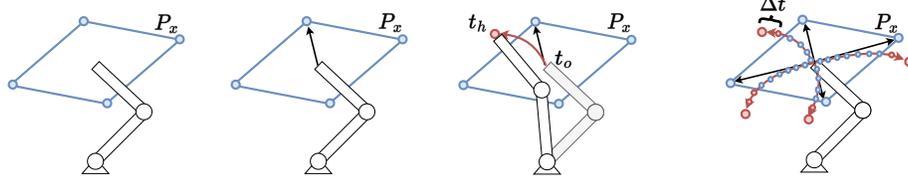


Fig. 6. Nonlinear simulation procedure

Each simulation roll-out considers a fixed joint torque vector τ and simulates the robot dynamics through the horizon time t_h , for $N = t_h/\Delta t$ steps.

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

To evaluate the accuracy of the approximation, the difference in reached space produced by the constant linear 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 τ_i generating the vertices \mathbf{v}_i of the reachable space polytope \mathcal{P}_x calculated by the linear model.

As shown on figure 6, for each of the n_v vertices \mathbf{v}_i of the polytope \mathcal{P}_x the simulation (14) is performed and all the cartesian positions \mathbf{x}_k of the robot in each of the $N = t_h/\Delta t$ sample times are retained for the further analysis.

$$\mathcal{X} = \left\{ \underbrace{\mathbf{x}_{0,0} \dots \mathbf{x}_{0,N}}_{\tau_0}, \underbrace{\mathbf{x}_{1,0} \dots \mathbf{x}_{1,N}}_{\tau_1}, \dots, \underbrace{\mathbf{x}_{n_v,0} \dots \mathbf{x}_{n_v,N}}_{\tau_{n_v}} \right\}$$

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

$$m_1 = \frac{|\mathcal{X} \cap \mathcal{P}_x|}{|\mathcal{X}|} = \frac{|\mathcal{X} \cap \mathcal{P}_x|}{N \cdot n_v} \quad (16)$$

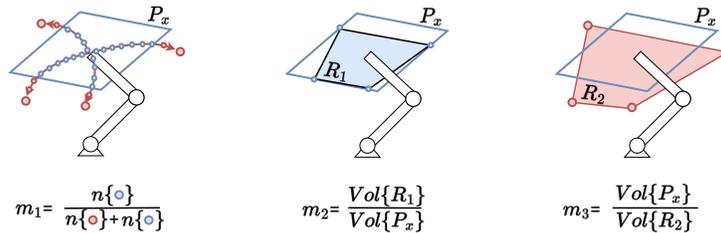


Fig. 7. Quantitative metrics definition

The metric m_1 gives an insight on how well the polytope \mathcal{P}_x encapsulates the real reachable space determined with simulations. If the polytope \mathcal{P}_x was an overapproximation of the real reachable set, m_1 would be always equal to 1.

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 $\mathbf{x}_k \in \mathcal{X} \cap \mathcal{P}_x$,

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

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

$$m_2 = \frac{Vol\{R_1\}}{Vol\{\mathcal{P}_x\}} \quad (17)$$

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. If a polytope \mathcal{P}_x was the exact reachable set of the robot, the metric m_2 would be equal to 1.

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 $\mathbf{x}_k \in \mathcal{X}$,

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

reached by the robot in the simulations.

$$m_3 = \frac{Vol\{\mathcal{P}_x\}}{Vol\{R_2\}} \quad (18)$$

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 $CH(\mathcal{X})$. If a polytope \mathcal{P}_x was the exact reachable set of the robot, the metric m_3 would be equal to 1.

3.2 Benchmark comparison

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

$$\ddot{\mathbf{x}} \in [\ddot{\mathbf{x}}_{min}, \ddot{\mathbf{x}}_{max}], \quad \dot{\mathbf{x}} \in [\dot{\mathbf{x}}_{min}, \dot{\mathbf{x}}_{max}] \quad (19)$$

The cartesian space limitations (19) assume that the robot's acceleration $\ddot{\mathbf{x}}$ and velocity $\dot{\mathbf{x}}$ capacity is constant, which is a strong assumption since the robot's capacity depends highly on robot's joint configuration \mathbf{q} and velocity $\dot{\mathbf{q}}$ [2][7]. 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 (19) can be calculated

$$\begin{aligned} \mathcal{C}_x = \{ \mathbf{x}_{k+1} \in R^m \mid \mathbf{x}_{k+1} = \ddot{\mathbf{x}}_k \frac{t_h^2}{2} + \dot{\mathbf{x}}_k t_h + \mathbf{x}_k \\ \ddot{\mathbf{x}}_k \in [\ddot{\mathbf{x}}_{min}, \ddot{\mathbf{x}}_{max}] \\ \dot{\mathbf{x}}_k t_h \in [\dot{\mathbf{x}}_{min}, \dot{\mathbf{x}}_{max}] \} \end{aligned} \quad (20)$$

Figure 8 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.

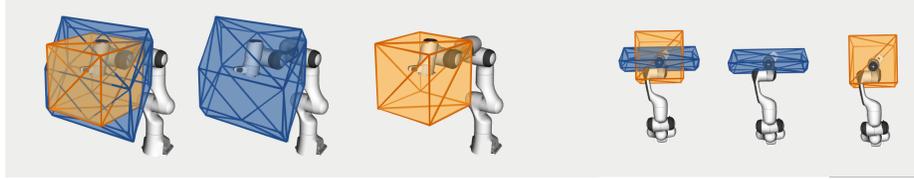


Fig. 8. Comparison of the end-effector reachable space approximation using the convex polytope \mathcal{P}_x (blue) and the cube \mathcal{C}_x (orange) for the *Franka Emika Panda* robot with the horizon time $t_h=200\text{ms}$.

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.

Eight different horizon lengths t_h are chosen for the experiments

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

and the simulation sample time has been set to $\Delta t=5\text{ms}$.

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 PYCAPACITY[6] package. The ICHM's approximation accuracy used for all the evaluations is $\delta=1\text{mm}$.

The proposed method implementation as well as the code used for the experiments is fully open-source and can be found in the gitlab repository². 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 9 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,

² https://gitlab.inria.fr/auctus-team/people/antunskuric/reachable_space/

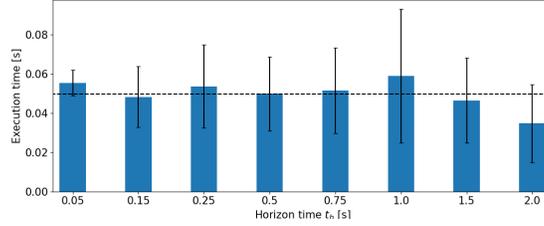


Fig. 9. Execution time of the polytope \mathcal{P}_x enumeration using ICHM algorithm averaged over 1000 executions for each of the 8 horizon lengths.

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 10 shows the ICHM execution time for the 8 horizon lengths with different number of environmental constraints $A_e \mathbf{x} \leq \mathbf{b}_e$. 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.

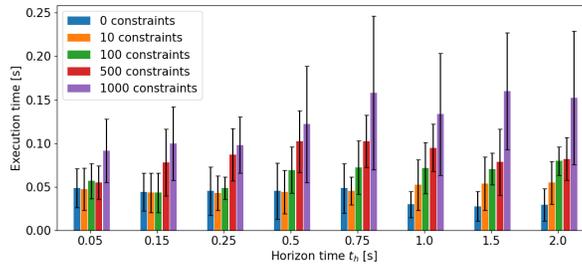


Fig. 10. Execution time of the polytope \mathcal{P}_x enumeration using ICHM algorithm averaged over 1000 executions for each of the 8 horizon lengths, with different number of added environmental constraints, ranging from 0 to 1000.

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.

4.2 Accuracy analysis

Figure 11 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 11 shows the metric m_1 . It can be seen that the ratio of simulated robot positions $\mathbf{x}_k \in \mathcal{X}$ 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 \mathbf{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 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 majority of the simulated robot's reached positions $\mathbf{x}_k \in \mathcal{X}$. 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.

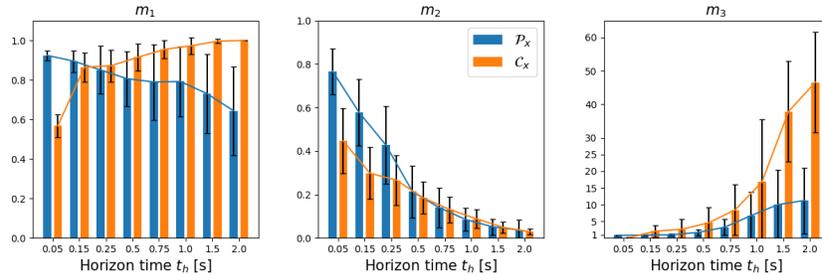


Fig. 11. Accuracy measures calculated for \mathcal{P}_x and \mathcal{C}_x , averaged over 100 random poses of the *Franka Emika panda* robot for each of 8 horizon lengths.

Evolution of the metric m_2 is shown on the middle graph of the figure 11. 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 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 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 11 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 $t_h=2$ s 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 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). Additionally, the results show that the polytope \mathcal{P}_x has better performance on all the metrics for horizon times $t_h \leq 250$ ms, even though the average volume of the \mathcal{P}_x is smaller than the volume of the \mathcal{C}_x for the all the horizon times.

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 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 \mathbf{q}_k and velocity $\dot{\mathbf{q}}_k$ at the beginning of the horizon and this linearised model is considered constant during the horizon length. The consideration of constant robot model and the linearity

of the robot model 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 numerical 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 robots 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 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 $t_h \leq 250\text{ms}$ the volume of the approximated polytope \mathcal{P}_x and the simulated reachable space of the robot R_2 is comparable figure 11, which 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 $H\mathbf{x} \leq \mathbf{d}$, 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 are 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 and other user defined constraints. With the execution time of around 50ms 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.

Acknowledgements This work has been funded by the BPI France Lichie project.

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. 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>
6. Skuric, A.: A real-time capable task-space capacity calculation python package, <https://pypi.org/project/pycapacity>
7. 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>
8. 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>
9. Sven R. Schepp, Jakob Thumm, S.B.L., Althoff, M.: Sara: A tool for safe human-robot coexistence and collaboration through reachability analysis (2022)