



HAL
open science

Efficient Set-Based Approaches for the Reliable Computation of Robot Capabilities

Joshua K. Pickard, Vincent Padois, Milan Hladík, David Daney

► **To cite this version:**

Joshua K. Pickard, Vincent Padois, Milan Hladík, David Daney. Efficient Set-Based Approaches for the Reliable Computation of Robot Capabilities. 2021. <hal-03182624>

HAL Id: hal-03182624

<https://inria.hal.science/hal-03182624v1>

Preprint submitted on 31 Mar 2021

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.



HAL Authorization

Efficient Set-Based Approaches for the Reliable Computation of Robot Capabilities

Joshua K. Pickard, Vincent Padois, Milan Hladík, and David Daney,

Abstract—To reliably model real robot characteristics, interval linear systems of equations allow to describe families of problems that consider sets of values. This allows to easily account for typical complexities such as sets of joint states and design parameter uncertainties. Inner approximations of the solutions to the interval linear systems can be used to describe the common capabilities of a robotic manipulator corresponding to the considered sets of values. In this work, several classes of problems are considered. For each class, reliable and efficient polytope, n-cube, and n-ball inner approximations are presented.

The interval approaches usually proposed are inefficient because they are too computationally heavy for certain applications, such as control. We propose efficient new inner approximation theorems for the considered classes of problems. This allows for usage with real-time applications as well as rapid analysis of potential designs. Several applications are presented for a redundant planar manipulator including locally evaluating the manipulator’s velocity, acceleration, and static force capabilities, and evaluating its future acceleration capabilities over a given time horizon.

Index Terms—Interval analysis, robot capabilities, inner approximation, interval linear system, tolerance solution

I. INTRODUCTION

THE capability of a robot to autonomously adapt to an open dynamic environment is one of the ultimate challenges of robotics.

One of the keys to solve this challenge lies in the ability to dynamically control the motion of the robot as a function of the prediction of the environment dynamics in order to both optimally achieve the goals assigned to the robot while ensuring safety at all times. In other words, planning and control can no longer be considered as two separated problems solved sequentially. Indeed, this kind of approach works well only for simple robots evolving in static environments and solving basic tasks, where a large amount of time can be spent planning the motion offline and where the control problem consists of tracking, with almost no adaptation, the planned trajectory.

This work was supported by the Czech Science Foundation Grant P403-18-04735S.

J. K. Pickard was with Auctus, Inria - IMS (Univ. Bordeaux / Bordeaux INP / CNRS UMR 5218), 33405 Talence, France. He is now with Eigen Innovations, 444 Aberdeen St #255, Fredericton, New Brunswick, Canada (email: joshua.pickard@inria.fr).

V. Padois, and D. Daney are with Auctus, Inria - IMS (Univ. Bordeaux / Bordeaux INP / CNRS UMR 5218), 33405 Talence, France (emails: vincent.padois@inria.fr and david.daney@inria.fr).

M. Hladík is with Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Malostranské nám. 25, 11800, Prague, Czech Republic (email: milan_hladik@centrum.cz).

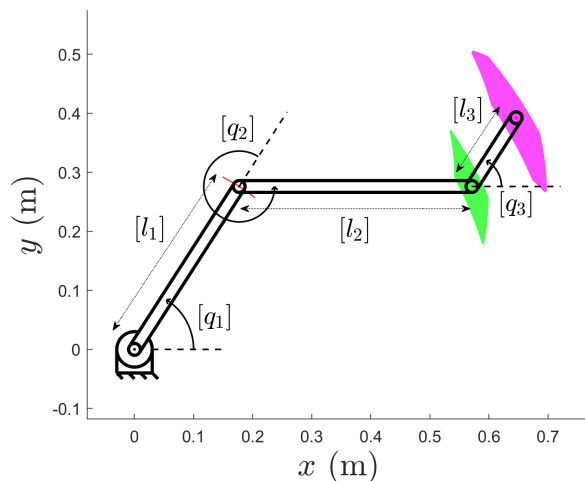


Fig. 1. The sets of positions associated with the distal ends of the links for the set of configurations $[q] = (1.0, -1.0, 1.0) \pm 0.1$ rads for a family of three-link planar manipulators with parameter values $[l] = (0.3280, 0.3940, 0.1385) \pm 0.001$ m. The red, green, and purple regions are the corresponding positions of the distal ends of the first, second, and third links respectively.

Beyond the perception abilities needed to estimate the state of the robot with respect to its environment, dynamic contexts require to adapt the planned motion at all time. For this adaptation to be optimal, it should not be purely reactive but rather adapt the overall plan online, i.e., at each control instant. While the concept of online re-planning is appealing, it raises the question of the real-time tractability of the corresponding optimization problems. As explained in [1], in the very challenging case of humanoid robots, accounting for limited computation power, one can choose to make use of rather precise models of the robot’s dynamics and constraints but the price to pay for this modeling complexity is the limited time horizon over which planning can be considered, leading to feasible but potentially sub-optimal robot behaviours. On the other hand, one can decide to plan over longer time horizons, for example using optimal control approaches or their receding horizon version such as Model Predictive Control (MPC). In return, one has to make use of simpler models to maintain a low computation complexity for online re-planning to be possible. This is typically the approach taken when planning for locomotion in humanoid robotics: given a target position in Cartesian space, one first computes a desired trajectory for an abstracted version of the system, typically its centre of

mass (CoM) and the associated centroidal dynamics. Then one defines a sequence of contacts and associated stances that are both compatible with the desired CoM trajectory and physically viable. Finally, using a more complete model of the system, a whole-body control approach is used to compute the actuation inputs allowing to follow the trajectory joining two consecutive stances and associated contact states [2]. The latter approach seems more appropriate in terms of the optimality of the obtained behaviours but also in terms of the involved computations. Indeed, while recent works on Differential Dynamic Programming [3] tackle the full model problem over the full horizon, these approaches remain computationally too complex for online re-planning on real robots and it actually does not seem quite necessary to plan precisely joint level actuation over a long time horizon.

If approaches resorting to simplified models seem more appropriate, an abstract model of the system still requires to be connected to a more physically realistic one for control purposes. While this connection can be made through task level abstractions, it is also necessary to account for the low level limits of the robot at the planning level in order to produce plans that can be achieved by the robots given its intrinsic (e.g., joint limits) and extrinsic (e.g., collision avoidance) constraints. Often in robotics, accounting for the joint level limitations at the more abstract level of task description is a matter of manual tuning and heuristics as solving this problem formally and in a computationally efficient manner is complex. Going beyond simple heuristics and assuming constant task level acceleration capabilities (e.g. the ones provided by robot manufacturers), some work have been performed in the past to account for acceleration limits at the joint level while accounting for the potential incompatibilities between these acceleration limits and position ones [4]–[6]. While of interest, these works fail short at accounting in a formal way for the true limits of robots which are best expressed at the joint level in terms of joint position, velocity, torque and torque derivative. Indeed, one of the main difficulties with joint level acceleration limits is that they are state dependant. This is all the more true when considering the expression of these limits in Cartesian space as the mappings from joint space to task space, being them at the static or dynamic level, are both state dependant and non-linear. In order to tackle this complexity, there is a need for new methodologies that are able to efficiently, accurately and with a limited computation cost predict the robot’s capabilities in static but more importantly in dynamic scenarios. Furthermore, these methodologies should be able to handle imprecise/variable systems by accounting for system complexities such as sets of joint states and design parameter uncertainties to produce reliable results. Accounting for these uncertainties is of high importance when dealing with floating base systems such as humanoids or mobile robots in general [7]. Thus, modeling errors and state changes over a given time horizon should be properly accounted for when computing capabilities.

With this overall motivation in mind, this work presents

several classes of problems that can be used to reliably and efficiently compute polytope, n-cube, and n-ball inner approximations of the capabilities of robotic systems, being them imprecise or uncertain.

The descriptions of the capabilities of many robotics systems can be formulated in terms of linear systems of equations

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (1)$$

with $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^m$. These linear equations may be used to describe specific capabilities of a robotic manipulator for a unique configuration \mathbf{q} or state $\{\mathbf{q}, \dot{\mathbf{q}}\}$. In offline applications, the evaluation of robot capabilities allows to analyze the performance of a given design and may be used during the dimensional synthesis stage of robotic design. In online applications, the current and/or future robot capabilities can be computed and may be used to autonomously re-plan and control the motion of the robot.

For example, for serial manipulators at the velocity kinematic level one may consider the:

- *velocity kinematics model:*

$$\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}} \quad (2)$$

where $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix, and $\dot{\mathbf{q}}$ and $\dot{\mathbf{x}}$ are the joint and operational velocities respectively.

- *kinetostatic model*

$$\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau} \quad (3)$$

where \mathbf{f} is the operational wrench and $\boldsymbol{\tau}$ is the joint torque/force vector.

At the dynamic level, one may also characterize the capabilities of a robot through the:

- *forward acceleration model:*

$$\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} = \ddot{\mathbf{x}} \quad (4)$$

where $\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})$ is the time derivative of the Jacobian matrix, and $\ddot{\mathbf{q}}$ and $\ddot{\mathbf{x}}$ are the joint and operational accelerations respectively.

- *configuration-space dynamic model:*

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) \quad (5)$$

where $\mathbf{M}(\mathbf{q})$ is the mass matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ is the centrifugal and Coriolis vector and $\mathbf{g}(\mathbf{q})$ is the gravity vector.

- *operational-space dynamic problem:*

$$\boldsymbol{\Lambda}(\mathbf{q})\ddot{\mathbf{x}} = \mathbf{f} - \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{p}(\mathbf{q}) \quad (6)$$

where $\boldsymbol{\Lambda}(\mathbf{q})$ is the Cartesian mass matrix, $\boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}})$ is the centrifugal and Coriolis vector in Cartesian space and $\mathbf{p}(\mathbf{q})$ is the gravity vector in Cartesian space.

Due to the many sources of uncertainties that exist (e.g., noisy sensor data, modeling errors), to reliably predict the capabilities of the robot it can be desirable to consider sets of robot models and states. Using interval analysis (see [8]–[10]), it is possible to extend these problems in order to

describe a family of problems that consider sets of values (e.g., sets of design parameters, sets of configurations $[\mathbf{q}]$, sets of joint velocities $[\dot{\mathbf{q}}]$, etc.). Figure 1 depicts a family of three-link planar manipulators with sets of link lengths, $[l_1]$, $[l_2]$, and $[l_3]$, and sets of configurations, $[q_1]$, $[q_2]$, and $[q_3]$, and shows the corresponding sets of positions associated with the distal ends of the links for the given configurations (53) and design parameters (51). The set of configurations may be used to model dimensional uncertainties in the system, and may also be used to model the temporal evolution of the system. The set-based approaches presented in this work can be used to reliably predict the robot's various capabilities for these type of problems, where the reliable and efficient evaluation of imprecise/variable systems and their current or future capabilities over a given prediction period provides many avenues for improving the performance and safety of robotic manipulators. Classes of problems may be described in terms of quantifiers which allow to describe reliable inner approximations of the capabilities of a robot over sets of values. Alternatively, reliable outer approximations may also be computed [11]–[14], although these types of problems are not considered in this work. Considering velocity kinematics related models, when a set of configurations $[\mathbf{q}]$ are considered the Jacobian matrix \mathbf{J} becomes an interval matrix $[\mathbf{J}]$. Generally, the evaluation of $[\mathbf{J}]$ is overestimated due to the interval wrapping effect [15] and dependency problem [16] and there exists some matrices $\mathbf{J}_o \in [\mathbf{J}]$ such that $\mathbf{J}_o \notin \{\mathbf{J}(\mathbf{q}) \mid \mathbf{q} \in [\mathbf{q}]\}$. Nevertheless, each $\mathbf{J} \in [\mathbf{J}]$ has associated capabilities and the intersection of these capabilities describes the common capabilities for all configurations $\mathbf{q} \in [\mathbf{q}]$. This extends to the dynamic models similarly.

The interval analysis framework has proven to be a useful tool for the analysis and synthesis of robotic mechanisms as it guarantees certified numerical solutions to problems [17]. When combined with classical branch-and-bound approaches, entire parameter spaces can be thoroughly and reliably explored allowing the worst-case conditions throughout the parameter space to be properly understood. Such an approach has been widely used for workspace analysis, e.g., considering flexure-jointed mechanisms [18], evaluating wrench capabilities [19], [20] and positioning errors [21], and computing collision-free poses [22] and singularity-free poses [23], [24].

By considering the intersections of manipulator capabilities for all values in the sets, this allows to compute common capabilities of the robotic manipulator corresponding to the considered sets of values. To reliably describe the capabilities of a robotic manipulator corresponding to a set of values, inner approximations of the intersections of manipulator capabilities must be computed. For a given interval linear system of equations, several classes of problems may be considered. The main contribution of this work is the development of efficient polytope, n-cube, and n-ball inner approximations which are applicable to each class of problem considered. Furthermore, these inner approximations are applied to several relevant robotics problems to demonstrate their capability to

easily handle typical complexities such as sets of joint states and design parameter uncertainties.

The remainder of this paper is as follows. First, in Section II the relevant interval notations are introduced. Next, in Section III the classes of interval linear system of equation problems considered in this work are described. In Section IV the polytope, n-cube, and n-ball inner approximations for each class of problem are presented. Several applications for each class of problem are presented. Several applications are presented in Section V for a redundant planar manipulator to locally evaluate the manipulator's velocity, static force, and acceleration capabilities, and also its future acceleration capabilities over a given time horizon.

II. NOTATION

The interval extension of a variable x is given by

$$[x] = \{x \mid \underline{x} \leq x \leq \bar{x}\} \quad (7)$$

where \underline{x} is the lower bound (infimum) $\underline{x} = \inf(x)$ and \bar{x} is the upper bound (supremum) $\bar{x} = \sup(x)$ of the interval.

A few terms useful in this work are:

- 1) The *midpoint* of an interval is the real value given by

$$\text{mid}([x]) = \frac{1}{2}(\underline{x} + \bar{x}) \quad (8)$$

- 2) The *radius* of an interval is the real value given by

$$\Delta[x] = \frac{1}{2}(\bar{x} - \underline{x}) \quad (9)$$

- 3) The *mignitude* of an interval is the real value given by

$$\text{mig}([x]) = \min(\{|x| \mid x \in [x]\}) \quad (10)$$

- 4) The *magnitude* of an interval is the real value given by

$$\text{mag}([x]) = \max(\{|x| \mid x \in [x]\}) \quad (11)$$

- 5) The *absolute value* of an interval is the interval given by

$$|[x]| = [\text{mig}([x]), \text{mag}([x])] \quad (12)$$

Let a vector and matrix be given by \mathbf{x} and \mathbf{A} respectively. The interval extensions of the vector and matrix are then given by $[\mathbf{x}]$ and $[\mathbf{A}]$ and the previously described terms may be applied to each element in the vector or matrix. Classes of problems may be described with the \forall (for all) and \exists (there exists) quantifiers which allow to describe reliable inner approximations to the interval linear system of equations over the considered sets of values.

III. CLASSES OF PROBLEMS

Several classes of problems are considered in this work which are presented in a general form $\mathbf{A}\mathbf{x} = \mathbf{b}$. These problems and their corresponding sub-problems are as follows.

A. Problems $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ and $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$

The first class of problems considered are of the form

$$\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}]) = \{\mathbf{b} \mid (\forall \mathbf{A} \in [\mathbf{A}])(\exists \mathbf{x} \in [\mathbf{x}])(\mathbf{A}\mathbf{x} = \mathbf{b})\} \quad (13)$$

which applies a set of linear transformations $[\mathbf{A}]$ to a set of vectors $[\mathbf{x}]$. The sub-problem

$$\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}]) = \{\mathbf{b} \mid (\exists \mathbf{x} \in [\mathbf{x}])(\mathbf{A}\mathbf{x} = \mathbf{b})\} \quad (14)$$

is geometrically a convex polytope [25]–[27]. Therefore, (13) is equivalent to the intersection of all sub-problems $\mathbf{A} \in [\mathbf{A}]$. That is,

$$\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}]) = \bigcap_{\mathbf{A} \in [\mathbf{A}]} \Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}]) \quad (15)$$

B. Problems $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ and $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$

The second class of problems considered are of the form

$$\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}]) = \{\mathbf{x} \mid (\forall \mathbf{A} \in [\mathbf{A}])(\exists \mathbf{b} \in [\mathbf{b}])(\mathbf{A}\mathbf{x} = \mathbf{b})\} \quad (16)$$

which is precisely the tolerance solution set of the interval linear system as described in [28]. Considering the sub-problem

$$\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}]) = \{\mathbf{x} \mid (\exists \mathbf{b} \in [\mathbf{b}])(\mathbf{A}\mathbf{x} = \mathbf{b})\} \quad (17)$$

which is also geometrically a convex polytope, it is clear that

$$\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}]) = \bigcap_{\mathbf{A} \in [\mathbf{A}]} \Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}]) \quad (18)$$

C. Robotics Use Cases

The use cases of each class of problem applied to the relevant robotics problems (2) – (6) are outlined in Tables I and II. The classes $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ and $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ consider a unique configuration \mathbf{q} or state $\{\mathbf{q}, \dot{\mathbf{q}}\}$ whereas the classes $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ and $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ can consider a set of configuration $[\mathbf{q}]$ or states $\{[\mathbf{q}], [\dot{\mathbf{q}}]\}$.

IV. INNER APPROXIMATIONS

A. $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ Inner Approximations

Geometrically, polytope $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ represents a so called zonotope [29]. The $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ polytope can be obtained as the convex hull (conv) of the set of \mathbf{b} associated with the 2^n vertices (vert) of $[\mathbf{x}]$. That is, the $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ polytope can be computed by

$$\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}]) = \text{conv}(\{\mathbf{b} \mid (\exists \mathbf{x} \in \text{vert}([\mathbf{x}])(\mathbf{A}\mathbf{x} = \mathbf{b}))\}) \quad (19)$$

An equivalent non-iterative approach called the hyperplane shifting method, proposed in [25] and further improved in [27], may be used as a fast alternative in place of the convex hull routine. It requires to consider sub-matrices formed from all of the possible combinations of $m-1$ columns of \mathbf{A} and directly computes the corresponding set of shifted hyperplanes.

Inner approximations of the largest inscribed n-ball and n-cube can be computed from the half-space representation of

the $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ polytope. Let the half-space representation of the $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$, describing the associated set of \mathbf{b} , be given by

$$\mathbf{H}\mathbf{b} \leq \mathbf{d} \quad (20)$$

with $\mathbf{H} \in \mathbb{R}^{k \times m}$ and $\mathbf{d} \in \mathbb{R}^k$. The hyperplane shifting method provides an efficient means of evaluating the half-space representation, especially when m is close to n as is common for many robotics problems.

Theorem 1 (Largest inscribed n-cube centered at \mathbf{b}_c). *If \mathbf{b}_c belongs to $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$, then the n-cube*

$$[\mathbf{c}] = \mathbf{b}_c + r[\mathbf{e}] \quad (21)$$

where $[\mathbf{e}]$ is a vector of $[-1, 1]$ intervals and

$$r = \min_{i=1, \dots, k} \frac{-\mathbf{h}_i \mathbf{b}_c + d_i}{\|\mathbf{h}_i\|_1} \quad (22)$$

where \mathbf{h}_i is the i^{th} row of \mathbf{H} and $\|\cdot\|_1$ is the 1-norm, is contained in $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$.

Proof. The Chebyshev center \mathbf{b}_c of a polytope yielding the largest inscribed n-cube can be determined by solving the linear program

$$\begin{aligned} & \max && r \\ & \text{subject to} && \mathbf{h}_i \mathbf{b}_c + r \|\mathbf{h}_i\|_1 \leq d_i \\ & && r \geq 0 \\ & && i = 1, \dots, k \end{aligned} \quad (23)$$

The maximum value for r may be computed directly for a given \mathbf{b}_c as (22). ■

Many extensions have been considered for Theorem 1, especially on the topic of tolerance sensitivity analysis (see [30], [31]).

Theorem 2 (Largest inscribed n-ball centered at \mathbf{b}_c). *If \mathbf{b}_c belongs to $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$, then the n-ball*

$$\mathcal{B} = \{\mathbf{b} \mid \|\mathbf{b} - \mathbf{b}_c\|_2 \leq r\} \quad (24)$$

with

$$r = \min_{i=1, \dots, k} \frac{-\mathbf{h}_i \mathbf{b}_c + d_i}{\|\mathbf{h}_i\|_2} \quad (25)$$

where $\|\cdot\|_2$ is the 2-norm, is contained in $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$.

Proof. The Chebyshev center \mathbf{b}_c of a polytope yielding the largest inscribed n-ball can be determined by solving the linear program

$$\begin{aligned} & \max && r \\ & \text{subject to} && \mathbf{h}_i \mathbf{b}_c + r \|\mathbf{h}_i\|_2 \leq d_i \\ & && r \geq 0 \\ & && i = 1, \dots, k \end{aligned} \quad (26)$$

The maximum value for r may be computed directly for a given \mathbf{b}_c as (25). ■

TABLE I
INNER APPROXIMATION USE CASES FOR $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ AND $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$

Problems	Unknown	Description of $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$	Description of $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$
$\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}}$	$\dot{\mathbf{x}}$	$\{\dot{\mathbf{x}} \mid (\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])(\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}})\}$	$\{\dot{\mathbf{x}} \mid (\forall \mathbf{J}(\mathbf{q}) \in \mathbf{J}([\mathbf{q}])(\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])(\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}})\}$
$\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau}$	$\boldsymbol{\tau}$	$\{\boldsymbol{\tau} \mid (\exists \mathbf{f} \in [\mathbf{f}])(\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau})\}$	$\{\boldsymbol{\tau} \mid (\forall \mathbf{J}(\mathbf{q}) \in \mathbf{J}([\mathbf{q}])(\exists \mathbf{f} \in [\mathbf{f}])(\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau})\}$
$\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} = \ddot{\mathbf{x}}$	$\ddot{\mathbf{x}}$	$\{\ddot{\mathbf{x}} \mid (\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])(\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \ddot{\mathbf{x}})\}$ $+ \{\ddot{\mathbf{x}} \mid (\exists \ddot{\mathbf{q}} \in [\ddot{\mathbf{q}}])(\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} = \ddot{\mathbf{x}})\}$	$\{\ddot{\mathbf{x}} \mid (\forall \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}}) \in \dot{\mathbf{J}}([\mathbf{q}], [\dot{\mathbf{q}}])(\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])(\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \ddot{\mathbf{x}})\}$ $+ \{\ddot{\mathbf{x}} \mid (\forall \mathbf{J}(\mathbf{q}) \in \mathbf{J}([\mathbf{q}])(\exists \ddot{\mathbf{q}} \in [\ddot{\mathbf{q}}])(\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} = \ddot{\mathbf{x}})\}$

TABLE II
INNER APPROXIMATION USE CASES FOR $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ AND $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$.

Problems	Unknown	Description of $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$	Description of $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$
$\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}}$	$\dot{\mathbf{q}}$	$\{\dot{\mathbf{q}} \mid (\exists \dot{\mathbf{x}} \in [\dot{\mathbf{x}}])(\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}})\}$	$\{\dot{\mathbf{q}} \mid (\forall \mathbf{J}(\mathbf{q}) \in \mathbf{J}([\mathbf{q}])(\exists \dot{\mathbf{x}} \in [\dot{\mathbf{x}}])(\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}})\}$
$\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau}$	\mathbf{f}	$\{\mathbf{f} \mid (\exists \boldsymbol{\tau} \in [\boldsymbol{\tau}])(\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau})\}$	$\{\mathbf{f} \mid (\forall \mathbf{J}(\mathbf{q}) \in \mathbf{J}([\mathbf{q}])(\exists \boldsymbol{\tau} \in [\boldsymbol{\tau}])(\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau})\}$
$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})$	$\ddot{\mathbf{q}}$	$\{\ddot{\mathbf{q}} \mid (\exists \boldsymbol{\tau} \in [\boldsymbol{\tau}])(\exists \mathbf{q} \in [\mathbf{q}])(\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])$ $(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}))\}$	$\{\ddot{\mathbf{q}} \mid (\forall \mathbf{M}(\mathbf{q}) \in \mathbf{M}([\mathbf{q}])(\exists \boldsymbol{\tau} \in [\boldsymbol{\tau}])(\exists \mathbf{q} \in [\mathbf{q}])(\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])$ $(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}))\}$
$\Lambda(\mathbf{q})\ddot{\mathbf{x}} = \mathbf{f} - \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{p}(\mathbf{q})$	$\ddot{\mathbf{x}}$	$\{\ddot{\mathbf{x}} \mid (\exists \mathbf{f} \in [\mathbf{f}])(\exists \mathbf{q} \in [\mathbf{q}])(\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])$ $(\Lambda(\mathbf{q})\ddot{\mathbf{x}} = \mathbf{f} - \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{p}(\mathbf{q}))\}$	$\{\ddot{\mathbf{x}} \mid (\forall \Lambda(\mathbf{q}) \in \Lambda([\mathbf{q}])(\exists \mathbf{f} \in [\mathbf{f}])(\exists \mathbf{q} \in [\mathbf{q}])(\exists \dot{\mathbf{q}} \in [\dot{\mathbf{q}}])$ $(\Lambda(\mathbf{q})\ddot{\mathbf{x}} = \mathbf{f} - \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{p}(\mathbf{q}))\}$

As an example, consider the $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ problem given

$$\mathbf{A} = \begin{pmatrix} 0.8947 & 0.6707 & 0.2409 \\ 0.3348 & 0.3899 & 0.6958 \end{pmatrix}, [\mathbf{x}] = \begin{pmatrix} [-1, 1] \\ [-1, 1] \\ [-1, 1] \end{pmatrix}. \quad (27)$$

The following inner approximations of $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ are given in Figure 2:

- the associated polytope (purple);
- the largest n-cube centered at the origin with $r = 0.4686$ (red);
- the largest n-ball centered at the origin with $r = 0.6407$ (green);
- the largest n-cube with variable center with $b_c = (0.4855, 0.2823)$ and $r = 0.4686$ (dashed line);
- the largest n-ball with variable center with $b_c = (0.2921, 0.1698)$ and $r = 0.6407$ (dashed line).

When reliable arithmetic operations are used which account for round-off errors, the computed inner approximations provide a reliable description of the solution of the linear transformation. In this example there is a diagonal line of optimal solutions corresponding to the parallel edges that the variable center inner approximation algorithm can return.

B. $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ Inner Approximations

As stated in (15), the common capabilities are the intersection of the capabilities for all sub-problems $\mathbf{A} \in [\mathbf{A}]$. Since the capabilities associated with each sub-problem are geometrically convex polytopes, the common capabilities are the intersection of the convex polytopes for all sub-problems.

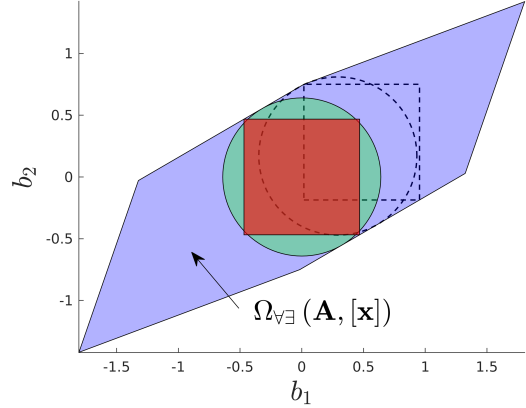


Fig. 2. The associated polytope, n-cube, and n-ball inner approximations of $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$.

Theorem 3. We have $\mathbf{b} \in \Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ if and only if the linear system

$$\begin{aligned} \mathbf{b} &= (\text{mid}([\mathbf{A}]) - \text{diag}(\mathbf{s}) \Delta[\mathbf{A}]) \mathbf{x}_1 \\ &\quad - (\text{mid}([\mathbf{A}]) + \text{diag}(\mathbf{s}) \Delta[\mathbf{A}]) \mathbf{x}_2, \\ \mathbf{x}_1, \mathbf{x}_2 &\geq 0, \\ \underline{\mathbf{x}} &\leq \mathbf{x}_1 - \mathbf{x}_2 \leq \bar{\mathbf{x}} \end{aligned}$$

is solvable for each $\mathbf{s} \in \{\pm 1\}^m$, where $\text{diag}(\mathbf{s})$ generates a diagonal matrix from the vector \mathbf{s} .

Proof. For a fixed \mathbf{b} , we have $\mathbf{b} \in \Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ if and only if the linear system

$$\mathbf{b} = \mathbf{A} \mathbf{x}, \quad \underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}} \quad (28)$$

is solvable in variable \mathbf{x} for each $\mathbf{A} \in [\mathbf{A}]$. By [32], this kind of strong solvability is equivalent to solvability of the system from the theorem statement. ■

The above test requires to solve 2^m linear systems. This can be a large number, however, it can hardly be decreased in general since the problem is intractable.

Theorem 4. *The test $\mathbf{b} \in \Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ is co-NP-hard.*

Proof. Suppose that $[\mathbf{x}]$ is sufficiently large. Then the test $\mathbf{b} \in \Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ is equivalent to stating that for each $\mathbf{A} \in [\mathbf{A}]$ there is \mathbf{x} such that $\mathbf{A}\mathbf{x} = \mathbf{b}$. That is equivalent with claiming that the interval linear system $[\mathbf{A}]\mathbf{x} = \mathbf{b}$ is strongly solvable. Strong solvability is, however, known to be co-NP-hard [33]. The real valued right-hand side \mathbf{b} of the linear system can be easily deduced. ■

To make the problem tractable, we first construct an inner approximation of $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ by a zonotope, that is, by a set of the form $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{y}])$ for some \mathbf{A} and $[\mathbf{y}]$. In particular, we will choose $\mathbf{A} := \text{mid}([\mathbf{A}])$ and $[\mathbf{y}] := \text{mid}([\mathbf{x}]) + [-r, r]\Delta[\mathbf{x}]$. The task is now to compute as large as possible value $r \in [0, 1]$ such that $\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{y}]) \subseteq \Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$. Notice that this need not be satisfied even for $r = 0$, so we present a sufficient condition below.

Theorem 5. *Suppose that $\text{mid}([\mathbf{A}])$ has full row rank and put*

$$r = \min_{i=1, \dots, n} \left(\frac{\left(\Delta[\mathbf{x}] - |\text{mid}([\mathbf{A}])^\dagger| \cdot \Delta[\mathbf{A}] \cdot \text{mag}([\mathbf{x}]) \right)_i}{\Delta[\mathbf{x}]_i} \right), \quad (29)$$

where $(\cdot)^\dagger$ denotes the Moore–Penrose pseudoinverse. If $r \geq 0$, then $\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{y}]) \subseteq \Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$.

Proof. We want $\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{y}]) \subseteq \Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ for each $\mathbf{A} \in [\mathbf{A}]$. That is, for each $\mathbf{y} \in [\mathbf{y}]$ there must be $\mathbf{x} \in [\mathbf{x}]$ such that $\text{mid}([\mathbf{A}])\mathbf{y} = \mathbf{A}\mathbf{x}$. Putting $\mathbf{A}' := \text{mid}([\mathbf{A}]) - \mathbf{A}$, we can write it as

$$\text{mid}([\mathbf{A}])\mathbf{y} + \mathbf{A}'\mathbf{x} = \text{mid}([\mathbf{A}])\mathbf{x}.$$

When $\mathbf{x} \in [\mathbf{x}]$ and $\mathbf{A} \in [\mathbf{A}]$, the value of $\mathbf{A}'\mathbf{x}$ ranges in the interval

$$[\mathbf{z}] := [-1, 1]\Delta[\mathbf{A}]\text{mag}([\mathbf{x}]).$$

Thus it is sufficient to find $[\mathbf{y}]$ such that

$$\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{y}]) + [\mathbf{z}] \subseteq \Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{x}]).$$

That is, for each $\mathbf{y} \in [\mathbf{y}]$ and $\mathbf{z} \in [\mathbf{z}]$, there must be $\mathbf{x} \in [\mathbf{x}]$ such that

$$\text{mid}([\mathbf{A}])\mathbf{y} + \mathbf{z} = \text{mid}([\mathbf{A}])\mathbf{x},$$

or, equivalently

$$\mathbf{z} = \text{mid}([\mathbf{A}])\mathbf{x} - \mathbf{y}.$$

If $\text{mid}([\mathbf{A}])$ has full row rank, this equation can be deduced from

$$\text{mid}([\mathbf{A}])^\dagger \mathbf{z} = \mathbf{x} - \mathbf{y}.$$

By choosing an appropriate \mathbf{x} , the right-hand side can attain any vector in $[-1, 1](1-r)\Delta[\mathbf{x}]$. Since the interval vector $[\mathbf{z}]$ is symmetric around zero, it is sufficient to compare the radii

$$|\text{mid}([\mathbf{A}])^\dagger| \Delta[\mathbf{z}] \leq (1-r)\Delta[\mathbf{x}],$$

from which the statement follows. ■

This approach to computing an inner polytope approximation of $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ by a set of the form $\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{y}])$ is particularly convenient when the intervals in $[\mathbf{A}]$ are narrow relatively to those in $[\mathbf{x}]$. In that case, the value of r is close to 1, and the inner approximation is tight. Inner approximations for the largest inscribed n-ball and n-cube in $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ may then be computed from the inner polytope approximation $\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{y}])$ using Theorems 1 and 2 respectively.

As an example, given \mathbf{A} and $[\mathbf{x}]$ from (27) and adding uncertainties of $[-0.01, 0.01]$ to each matrix element gives the following inner approximation of $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ shown in Figure 3:

- the associated polytope with $r = 0.9289$ (purple);
- the largest n-cube centered at the origin with $r = 0.4353$ (red);
- the largest n-ball centered at the origin with $r = 0.5951$ (green);
- the largest n-cube with variable center with $b_c = (0.4510, 0.2622)$ and $r = 0.4353$ (dashed line);
- the largest n-ball with variable center with $b_c = (0.2714, 0.1577)$ and $r = 0.5951$ (dashed line).

For comparison the polytope $\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{x}])$ is also shown which necessarily contains the polytope $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ due to (15).

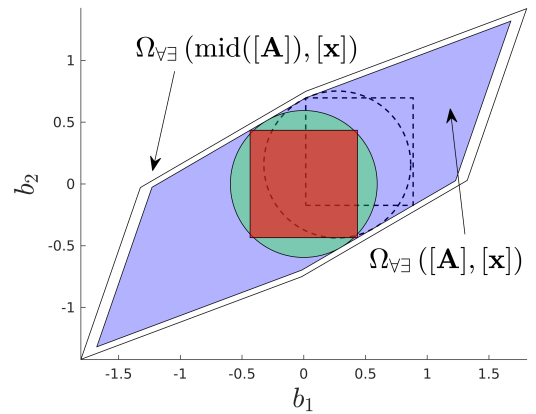


Fig. 3. The associated polytope, n-cube, and n-ball inner approximations for $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$. For comparison the polytope $\Omega_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{x}])$ is also shown.

C. Inner Approximations of $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$

According to [34], a vector \mathbf{x} belongs to $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ (i.e., is a tolerance solution) if and only if

$$|\text{mid}([\mathbf{A}])\mathbf{x} - \text{mid}([\mathbf{b}])| \leq -\Delta[\mathbf{A}]\mathbf{x} + \Delta[\mathbf{b}] \quad (30)$$

This allows to quickly verify if a desired property exists locally. Furthermore, the following theorems are proposed for directly computing inner approximations of the local capabilities, which may be used for real-time applications. The problems of computing outer approximations of the tolerable solution set have been addressed in [11].

Theorem 6 (Largest inscribed n-cube centered at \mathbf{x}_c). *If \mathbf{x}_c belongs to $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ then an n-cube*

$$[\mathbf{c}] = \mathbf{x}_c + r[\mathbf{e}] \quad (31)$$

with

$$r = \min_{i=1,\dots,m} \inf \left(\frac{\Delta b_i - |\text{mid}([b_i]) - [\mathbf{a}_i]\mathbf{x}_c|}{\|[\mathbf{a}_i]\|_1} \right) \quad (32)$$

where $[\mathbf{a}_i]$ is the i^{th} row of $[\mathbf{A}]$, is contained in $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$.

The proof for Theorem 6 is given in [35]. Notice that it provides the largest n-cube centered in \mathbf{x}_c . Notice also that it requires that the right-hand side of (32) is evaluated exactly. If we evaluate it by interval arithmetic, it provides a lower bound on the largest cube only. In [36], the problems of computing inner approximations of the tolerable solution set are also considered.

On the other hand, the test $[\mathbf{c}] \subseteq \Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ can be simply performed by interval evaluation $[\mathbf{A}][\mathbf{c}] \subseteq [\mathbf{b}]$. Therefore, we can extend the above inner n-cube $[\mathbf{c}]$ by ε -inflation approach to $[\mathbf{c}] := [\mathbf{c}] + \varepsilon[\mathbf{e}]$ and repeat it while the test does not break.

To overcome the above drawbacks, we propose a more general method. It finds the optimal center point of the n-cube and it computes exactly its radius, too.

Theorem 7 (Largest inscribed n-cube with variable center). *The largest inscribed n-cube to $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ reads $[\mathbf{c}] = \mathbf{x}_c + r[\mathbf{e}]$, where \mathbf{x}_c, r is an optimal solution to*

$$\max r \quad (33a)$$

$$\text{subject to } \sum_j y_j^i \leq \bar{b}_i, \quad \forall i, \quad (33b)$$

$$y_j^i \geq \bar{a}_{ij}(x_{cj} + r), \quad (33c)$$

$$y_j^i \geq \bar{a}_{ij}(x_{cj} - r), \quad (33d)$$

$$y_j^i \geq \underline{a}_{ij}(x_{cj} + r), \quad (33e)$$

$$y_j^i \geq \underline{a}_{ij}(x_{cj} - r), \quad (33f)$$

$$\sum_j z_j^i \geq \underline{b}_i, \quad \forall i, \quad (33g)$$

$$z_j^i \leq \bar{a}_{ij}(x_{cj} + r), \quad (33h)$$

$$z_j^i \leq \bar{a}_{ij}(x_{cj} - r), \quad (33i)$$

$$z_j^i \leq \underline{a}_{ij}(x_{cj} + r), \quad (33j)$$

$$z_j^i \leq \underline{a}_{ij}(x_{cj} - r). \quad (33k)$$

Proof. Our problem reads

$$\begin{aligned} & \max r \\ & \text{subject to } \sup([\mathbf{a}_i](\mathbf{x}_c + r[\mathbf{e}])) \leq \bar{b}_i \quad \forall i, \\ & \quad \quad \quad \inf([\mathbf{a}_i](\mathbf{x}_c + r[\mathbf{e}])) \geq \underline{b}_i \quad \forall i. \end{aligned}$$

Denoting by y_j^i and z_j^i the supremum and the infimum of the product $[a_{ij}](x_{cj} + r[-1, 1])$, respectively, the resulting optimization model follows. ■

Notice that (33) is a linear programming problem, so it can be solved efficiently. The number of variables is $1 + n + 2mn$, so the problem is still polynomially solvable.

Theorem 8 (Largest inscribed n-ball centered at \mathbf{x}_c). *If \mathbf{x}_c belongs to $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$, then the n-ball*

$$\mathcal{B} = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_c\| \leq r\} \quad (34)$$

with

$$\begin{aligned} r = \min_{i=1,\dots,m} & \\ \min \left(\inf \left(\frac{-[\mathbf{a}_i]\mathbf{x}_c + \bar{b}_i}{\|[\mathbf{a}_i]\|_2} \right), \inf \left(\frac{[\mathbf{a}_i]\mathbf{x}_c - \underline{b}_i}{\|[\mathbf{a}_i]\|_2} \right) \right) & \quad (35) \end{aligned}$$

where $\|\cdot\|_2$ is the 2-norm, is contained in $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$.

Proof. According to [34], we know that any $\mathbf{x} \in \Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ must satisfy

$$\{\mathbf{A}\mathbf{x} \mid \mathbf{A} \in [\mathbf{A}]\} \subseteq [\underline{\mathbf{b}}, \bar{\mathbf{b}}] \quad (36)$$

Therefore, $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ is described by the convex polytope given by the following inequalities

$$\begin{aligned} & \sup([\mathbf{a}_i]\mathbf{x}) \leq \bar{b}_i, \\ & \sup(-[\mathbf{a}_i]\mathbf{x}) \leq -\underline{b}_i, \\ & \mathbf{x} \in \mathbb{R}^n \\ & i = 1, \dots, m \end{aligned} \quad (37)$$

where sup ensures the inequalities are well-defined. In other words, (37) describes the intersection of a set of half-spaces.

For a given $\mathbf{A} \in [\mathbf{A}]$, the largest n-ball centered at \mathbf{x}_c and inscribed to $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ is computed by the linear program

$$\begin{aligned} & \max r \\ & \text{subject to } \mathbf{a}_i \mathbf{x}_c + r \|\mathbf{a}_i\|_2 \leq \bar{b}_i, \quad \forall i, \\ & \quad \quad \quad -\mathbf{a}_i \mathbf{x}_c + r \|\mathbf{a}_i\|_2 \leq -\underline{b}_i, \quad \forall i, \\ & \quad \quad \quad r \geq 0, \end{aligned}$$

from which

$$r = \min_{i=1,\dots,m} \min \left(\frac{-\mathbf{a}_i \mathbf{x}_c + \bar{b}_i}{\|\mathbf{a}_i\|_2}, \frac{\mathbf{a}_i \mathbf{x}_c - \underline{b}_i}{\|\mathbf{a}_i\|_2} \right).$$

Now, for the radius of the largest n-ball centered at \mathbf{x}_c inscribed to $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ we just minimize subject to $\mathbf{a}_i \in [\mathbf{a}_i]$. ■

The theorem provides the largest n-ball which is centered in a given point \mathbf{x}_c . Notice also that it requires that the right-hand

side of (35) is evaluated exactly. If we evaluate it by interval arithmetic, it provides a lower bound on the largest ball only.

Lemma 1. *The function $f(a) = \frac{a^T x + b}{\|a\|}$ is quasiconcave on the domain $a \neq 0$, $a^T x + b \geq 0$.*

Proof. Let $\alpha \geq 0$ be fixed. The set of $a \neq 0$ satisfying $f(a) \geq \alpha$ is characterized by

$$a^T x + b \geq \alpha \|a\|,$$

which describes a convex set. ■

The elements of the right-hand side of (35) fulfill the assumptions of the lemma. Since quasiconcave functions attain the minima on the border of the domain, we have the corollary below; the second item follows from the fact that (35) is monotone in some cases.

Corollary 1. *For every i, j we have:*

- (1) *The right-hand side of (35) is attained for $a_{ij} \in \{\underline{a}_{ij}, \bar{a}_{ij}\}$.*
- (2) *If $x_{cj} \geq 0$ and $a_{ij} \geq 0$, then we can fix $a_{ij} := \bar{a}_{ij}$ in the first infimum of (35).*
- (3) *If $x_{cj} \leq 0$ and $a_{ij} \leq 0$, then we can fix $a_{ij} := \underline{a}_{ij}$ in the first infimum of (35).*
- (4) *If $x_{cj} \leq 0$ and $a_{ij} \geq 0$, then we can fix $a_{ij} := \bar{a}_{ij}$ in the second infimum of (35).*
- (5) *If $x_{cj} \geq 0$ and $a_{ij} \leq 0$, then we can fix $a_{ij} := \underline{a}_{ij}$ in the second infimum of (35).*

Therefore, we may quickly check if \mathbf{x}_c belongs to $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ by using (30). If so, the corresponding inscribed n-cube and n-ball are computed directly from (32) and (35) respectively. The proposed theorems are also valid for the $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ sub-problems.

Theorem 9 (Inscribed n-ball with variable center). *Denote $s_i := \sup(\|[\mathbf{a}_i]\|_2) \forall i$. Let $r, \mathbf{x}_c, \mathbf{y}$ be an optimal solution of the linear programming problem*

$$\begin{aligned} \max \quad & r \\ \text{subject to} \quad & \text{mid}(\mathbf{a}_i) \mathbf{x}_c + \Delta \mathbf{a}_i \mathbf{y} + r s_i \leq \bar{b}_i, \quad \forall i, \\ & -\text{mid}(\mathbf{a}_i) \mathbf{x}_c + \Delta \mathbf{a}_i \mathbf{y} + r s_i \leq -\underline{b}_i, \quad \forall i, \\ & r \geq 0, \mathbf{y} \geq \mathbf{x}_c, \mathbf{y} \geq -\mathbf{x}_c. \end{aligned} \quad (38)$$

Then the ball (34) is contained in $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$.

Proof. As in the proof of Theorem 8, $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ is described by (37). The Chebyshev center \mathbf{x}_c of a polytope can be determined by the linear programming problem which reads

$$\begin{aligned} \max \quad & r \\ \text{subject to} \quad & \sup([\mathbf{a}_i] \mathbf{x}_c + r \|\mathbf{a}_i\|_2) \leq \bar{b}_i, \quad \forall i, \\ & \sup(-[\mathbf{a}_i] \mathbf{x}_c + r \|\mathbf{a}_i\|_2) \leq -\underline{b}_i, \quad \forall i, \\ & r \geq 0 \end{aligned} \quad (39)$$

It is hard to express the suprema explicitly, so we estimate them from above, resulting in a lower bound to the maximum inscribed n-ball

$$\begin{aligned} \max \quad & r \\ \text{subject to} \quad & \sup([\mathbf{a}_i] \mathbf{x}_c) + r s_i \leq \bar{b}_i, \quad \forall i, \\ & \sup(-[\mathbf{a}_i] \mathbf{x}_c) + r s_i \leq -\underline{b}_i, \quad \forall i, \\ & r \geq 0 \end{aligned} \quad (40)$$

Since $\sup([\mathbf{a}_i] \mathbf{x}_c) = \text{mid}(\mathbf{a}_i) \mathbf{x}_c + \Delta \mathbf{a}_i | \mathbf{x}_c |$, we equivalently have (38), where \mathbf{y} substitutes for $| \mathbf{x}_c |$. ■

By the first item of Corollary 1, the largest inscribed n-ball is attained for a vertex of \mathbf{A} . This is true not only for the fixed center \mathbf{x}_c , but even when the center is variable (because we can fix the center at the optimal one). This yields the following method of exponential complexity (with respect to the size of $[\mathbf{A}]$). It remains an open question if the problem is computationally tractable or not.

Theorem 10 (Largest inscribed n-ball with variable center). *Let r, \mathbf{x}_c be an optimal solution of the linear programming problem*

$$\begin{aligned} \max \quad & r \\ \text{subject to} \quad & \mathbf{a}_i \mathbf{x}_c + r \|\mathbf{a}_i\|_2 \leq \bar{b}_i, \quad \forall i, \forall \mathbf{a}_i \in \text{vert}([\mathbf{a}_i]), \\ & -\mathbf{a}_i \mathbf{x}_c + r \|\mathbf{a}_i\|_2 \leq -\underline{b}_i, \quad \forall i, \forall \mathbf{a}_i \in \text{vert}([\mathbf{a}_i]), \\ & r \geq 0. \end{aligned} \quad (41)$$

Then the ball (34) is the largest ball contained in $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$.

The polytope $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ may be represented as the set of all feasible solutions of a linear programming problem. The corresponding linear program (42) was first presented in [37].

Theorem 11. $\mathbf{x} \in \Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ *if and only if $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$ is a solution to the system of linear inequalities*

$$\begin{aligned} \bar{\mathbf{A}} \mathbf{x}_1 - \underline{\mathbf{A}} \mathbf{x}_2 &\leq \bar{\mathbf{b}} \\ -\underline{\mathbf{A}} \mathbf{x}_1 + \bar{\mathbf{A}} \mathbf{x}_2 &\leq \underline{\mathbf{b}} \\ \mathbf{x}_1 &\geq 0, \quad \mathbf{x}_2 \geq 0 \end{aligned} \quad (42)$$

The proof for Theorem 11 is given in [37].

For example, given \mathbf{A} as the transpose from (27) and adding uncertainties of $[-0.01, 0.01]$ to each matrix element and an arbitrary non-symmetric box $[\mathbf{b}]$

$$[\mathbf{b}] = \begin{pmatrix} [-74.0, 95.0] \\ [-24.0, 20.0] \\ [-22.0, 33.0] \end{pmatrix} \quad (43)$$

the following inner approximations of $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ are given in Figure 4:

- the associated polytope (purple);
- the largest n-cube centered at the origin with $r = 18.5082$ (red);
- the largest n-ball centered at the origin with $r = 25.3333$ (green);

- the largest n-cube with variable center with $x_c = (-1.0599, -3.1972)$ and $r = 20.3591$ (dashed line);
- the largest n-ball with variable center with $x_c = (-2.5291, -0.6963)$ and $r = 27.8666$ (dashed line).

For comparison the polytope $\Sigma_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{b}])$ is also shown which necessarily contains the polytope $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ due to (18).

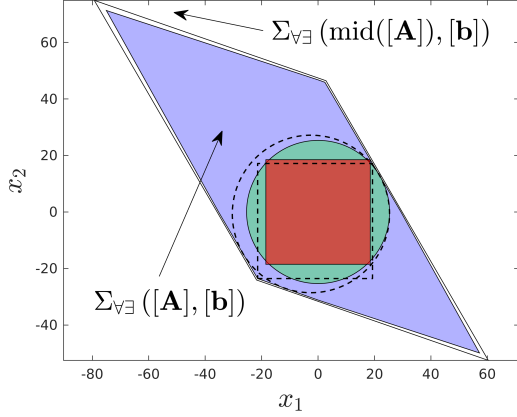


Fig. 4. The associated polytope, n-cube, and n-ball inner approximations of $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$. For comparison the polytope $\Sigma_{\forall\exists}(\text{mid}([\mathbf{A}]), [\mathbf{b}])$ is also shown.

D. $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ Inner Approximations

When \mathbf{A} is square, the $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ polytope can be obtained as the convex hull of the set of \mathbf{x} associated with the 2^m vertices of $[\mathbf{b}]$ as

$$\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}]) = \text{conv}(\{\mathbf{x} \mid (\exists \mathbf{b} \in \text{vert}([\mathbf{b}]))(\mathbf{x} = \mathbf{A}^{-1} \mathbf{b})\}) \quad (44)$$

When \mathbf{A} is non-square and $m > n$ the linear system of equations is over-determined and the Moore–Penrose pseudo-inverse solution

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b}, \quad (45)$$

where $\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$, only ensures that the norm of the error $\mathbf{b} - \mathbf{A} \mathbf{x}$ is minimized. This error is zero and (45) has a solution if and only if the set of \mathbf{b} belongs to the image (Im) of \mathbf{A} . In [38] the authors propose finding a reduced polytope that is given by the intersection $[\mathbf{b}] \cap \text{Im}(\mathbf{A})$ and apply this method to solve the wrench problem of redundant serial manipulators. This approach has been recently extended to allow for online computation of the polytope [39]. For many robotics applications, this is equivalent to considering only the set of \mathbf{b} that do not cause internal motions. For example, when considering the velocity kinematics model, only the joint states resulting in motions at the end-effector are considered, as states in the null space produce only internal motions.

The associated \mathbf{x} for each vertex of the reduced polytope can be computed from (45) and the convex hull of the resulting

set gives the polytope $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$. The $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ polytope for over-determined systems is given by

$$\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}]) = \text{conv}(\{\mathbf{x} \mid (\exists \mathbf{b} \in \text{vert}([\mathbf{b}] \cap \text{Im}(\mathbf{A}))) (\mathbf{x} = \mathbf{A}^\dagger \mathbf{b})\}) \quad (46)$$

The proposed n-cube and n-ball theorems (Theorems 6, 7, and 8) are also valid and applicable to the $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ sub-problems.

For example, given \mathbf{A} as the transpose from (27) and $[\mathbf{b}]$ from (43), the box $[\mathbf{b}]$ and corresponding reduced polytope $[\mathbf{b}] \cap \text{Im}(\mathbf{A})$, and the following inner approximations of $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ are given in Figure 5:

- the associated polytope (purple);
- the largest n-cube centered at the origin with $r = 18.8573$ (red);
- the largest n-ball centered at the origin with $r = 25.7800$ (green);
- the largest n-cube with variable center with $x_c = (-1.0450, -3.3319)$ and $r = 20.7430$ (dashed line);
- the largest n-ball with variable center with $x_c = (-2.5624, -0.7216)$ and $r = 28.3579$ (dashed line).

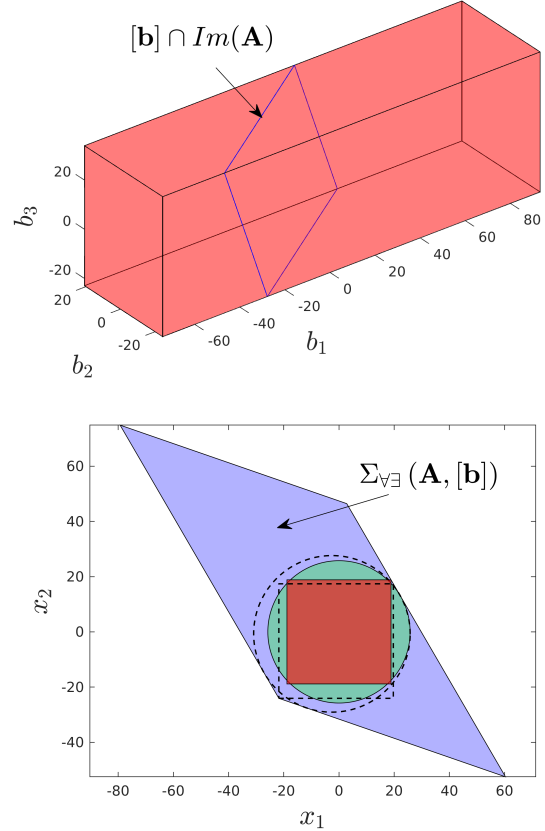


Fig. 5. (Top) The box $[\mathbf{b}]$ and the corresponding reduced polytope $[\mathbf{b}] \cap \text{Im}(\mathbf{A})$. (Bottom) The associated polytope, n-cube, and n-ball inner approximations of $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$.

E. Inner Approximations Summary

The equations associated with each inner approximation for each class of problem are summarized in Table III. Software implementations of the proposed inner approximation algorithms are made available at [40] for Matlab. To get an estimate of the computational times for typical 7-degree-of-freedom redundant robotics applications, for each problem the mean computational times¹ and standard deviations (over 20 runs) evaluated for a random system with $n = 7$ and $m = 6$ are provided in Table III. The fixed center point is taken as the origin. It can be noted that the Matlab implementations are not optimized for speed and the computational times can be significantly reduced with more efficient implementations. However, the already low computational times for many of the inner approximations demonstrates the suitability for real-time applications.

V. APPLICATIONS

One application of this work is to improve the reliability of the computed capabilities, since all sources of error can be easily modeled and managed. Methods for reliably modeling sources of error are described within the appropriate design framework [21], [41]. This allows to compute certifiable results providing confidence of the calculated capabilities of a manipulator. Other applications of this work are to provide reliable inner approximations of the common capabilities of a robotic manipulator over a given time horizon. This allows to reliably compute local capabilities, in real-time in many cases, that can be used online to evaluate the manipulator's current capabilities as well as its future capabilities within a time horizon.

As a simple example, consider a redundant three-link planar manipulator for a positioning task with point masses m_1, m_2, m_3 at distal ends of links of lengths l_1, l_2, l_3 . The manipulator is depicted in Figure 6.

The manipulator's 2×3 Jacobian matrix is given by

$$\mathbf{J}(\mathbf{q}) = \begin{pmatrix} J_1 - sq_{12}l_2 - sq_1l_1 & J_1 - sq_{12}l_2 & J_1 \\ J_2 + cq_{12}l_2 + cq_1l_1 & J_2 + cq_{12}l_2 & J_2 \end{pmatrix} \quad (47)$$

where $q_{ijk} = (q_i + q_j + q_k)$, $sq_i = \sin(q_i)$, $cq_i = \cos(q_i)$, $J_1 = -sq_{123}l_3$, and $J_2 = cq_{123}l_3$.

The 2×3 time derivative of the Jacobian matrix is given by

$$\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{pmatrix} -J_6 - J_3 - cq_1l_1\dot{q}_1 & -J_6 - J_3 & -J_6 \\ J_5 - J_4 - sq_1l_1\dot{q}_1 & J_5 - J_4 & J_5 \end{pmatrix} \quad (48)$$

where $\dot{q}_{ijk} = \dot{q}_i + \dot{q}_j + \dot{q}_k$, $J_3 = cq_{12}l_2\dot{q}_{12}$, $J_4 = sq_{12}l_2\dot{q}_{12}$, $J_5 = J_1\dot{q}_{123}$, and $J_6 = J_2\dot{q}_{123}$.

¹A desktop computer with a AMD Phenom(tm) II X6 1045T 2.70 GHz processor and 16.0 GB of RAM is used to compute mean computational times.

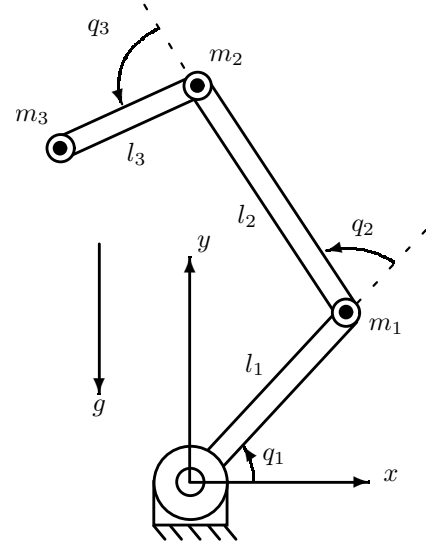


Fig. 6. Three-link planar manipulator with point masses at distal ends of links.

The mass matrix, centrifugal and Coriolis vector, and gravity vector are given by

$$\begin{aligned} \mathbf{M}(\mathbf{q}) &= \begin{pmatrix} K_7 & K_1 + K_3 + K_4 & K_2 + K_6 \\ K_1 + K_3 + K_4 & 2K_5 + K_4 & K_5 + K_6 \\ K_2 + K_6 & K_5 + K_6 & K_6 \end{pmatrix} \\ \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{pmatrix} K_{10}(K_8 + K_{12}) + K_{11}(K_8 + K_9) \\ \dot{q}_1^2(K_8 + K_{12}) + K_9K_{11} \\ -K_9K_{10} + \dot{q}_1^2(K_8 + K_9) \end{pmatrix} \\ \mathbf{g}(\mathbf{q}) &= g \begin{pmatrix} cq_{123}l_3m_3 + l_2cq_{12}m_{23} + l_1cq_1m_{123} \\ cq_{123}l_3m_3 + l_2cq_{12}m_{23} \\ cq_{123}l_3m_3 \end{pmatrix} \end{aligned} \quad (49)$$

where

$$\begin{aligned} K_1 &= l_1l_2m_{23}cq_2 \\ K_2 &= l_3m_3(l_1cq_{23} + l_2cq_3) \\ K_3 &= l_3m_3(l_1cq_{23} + 2l_2cq_3) \\ K_4 &= l_2^2m_{23} + K_6 \\ K_5 &= l_2l_3m_3cq_3 \\ K_6 &= l_3^2m_3 \\ K_7 &= 2K_1 + 2K_2 + K_4 + l_1^2m_{123} \\ K_8 &= l_1l_3m_3sq_{23} \\ K_9 &= l_2l_3m_3sq_3 \\ K_{10} &= -(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\ K_{11} &= -(2\dot{q}_1\dot{q}_3 + 2\dot{q}_2\dot{q}_3 + \dot{q}_3^2) \\ K_{12} &= l_1l_2m_{23}sq_2 \end{aligned} \quad (50)$$

with $m_{ijk} = m_i + m_j + m_k$, and g is the gravitational constant.

For the following examples, realistic parameters and joint state limits are selected to approximate a simplified planar model of joints 2, 4, and 6 of the Franka Emika Panda robot.

TABLE III
INNER APPROXIMATION SUMMARY.

Approximations	$\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$	ms \pm std	$\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$	ms \pm std	$\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$	ms \pm std	$\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$	ms \pm std
Polytope inner approximation	(19)	56.842 ± 1.090	(29) + (19)	56.719 ± 0.430	(44) or (46)	187.374 ± 0.982	(42)	381.568 ± 3.544
Largest inscribed n-cube centered at a point	(19) + (22)	1.496 ± 0.033	(29) + (22)	1.537 ± 0.030	(32)	0.024 ± 0.002	(32)	0.024 ± 0.001
Largest inscribed n-cube with variable center	(19) + (23)	20.748 ± 0.389	(29) + (23)	20.807 ± 0.310	(33) – exact	30.690 ± 0.414	(33) – exact	33.154 ± 0.675
Largest inscribed n-ball centered at a point	(19) + (25)	1.509 ± 0.029	(29) + (25)	1.541 ± 0.023	(35)	0.036 ± 0.004	(35)	0.036 ± 0.002
Largest inscribed n-ball with variable center	(19) + (26)	19.872 ± 0.322	(29) + (26)	19.883 ± 0.294	(38) (41) – exact	17.287 ± 0.181 34.701 ± 0.466	(38) (41) – exact	17.560 ± 0.434 48.459 ± 0.481

The parameter values with assumed uncertainties are

$$\begin{aligned}
 [l_1] &= 0.328 \pm 0.0001 \text{ m} \\
 [l_2] &= 0.394 \pm 0.0001 \text{ m} \\
 [l_3] &= 0.1385 \pm 0.0001 \text{ m} \\
 [m_1] &= 1.9 \pm 0.001 \text{ kg} \\
 [m_2] &= 1.6 \pm 0.001 \text{ kg} \\
 [m_3] &= 1.3 \pm 0.001 \text{ kg}
 \end{aligned} \tag{51}$$

and the corresponding joint state limits (*i.e.*, $[\mathbf{q}_{lim}]$, $[\dot{\mathbf{q}}_{lim}]$, $[\ddot{\mathbf{q}}_{lim}]$, $[\boldsymbol{\tau}_{lim}]$, $[\dot{\boldsymbol{\tau}}_{lim}]$) are taken from the manufacturer's specifications as

$$\begin{aligned}
 [\mathbf{q}_{lim}] &= \begin{pmatrix} [-1.7628, 1.7628] \\ [-3.0718, -0.0698] \\ [-0.0175, 3.7525] \end{pmatrix} \text{ rad} \\
 [\dot{\mathbf{q}}_{lim}] &= \begin{pmatrix} [-2.175, 2.175] \\ [-2.175, 2.175] \\ [-2.610, 2.610] \end{pmatrix} \text{ rad/s} \\
 [\ddot{\mathbf{q}}_{lim}] &= \begin{pmatrix} [-7.5, 7.5] \\ [-12.5, 12.5] \\ [-20, 20] \end{pmatrix} \text{ rad/s}^2 \\
 [\boldsymbol{\tau}_{lim}] &= \begin{pmatrix} [-87, 87] \\ [-87, 87] \\ [-12, 12] \end{pmatrix} \text{ Nm} \\
 [\dot{\boldsymbol{\tau}}_{lim}] &= \begin{pmatrix} [-1000, 1000] \\ [-1000, 1000] \\ [-1000, 1000] \end{pmatrix} \text{ Nm/s}
 \end{aligned} \tag{52}$$

To visualize the effects of considering a set of configurations, for

$$[\mathbf{q}] = (1.0, -1.0, 1.0) \pm 0.1 \text{ rads} \tag{53}$$

an approximation of the set of positions associated with the distal ends of the links is depicted in Figure 1. The proposed inner approximations provide a convenient means of evaluating the robot's common capabilities over the corresponding set of poses, provided that the intersection of capabilities is not empty.

A. Evaluating the Velocity Problem

A necessary procedure when analyzing the performance of a robotic manipulator is to evaluate the possible end-effector velocities through the mapping of known joint velocity limits with (2). The velocity problem is of the class $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ and the associated inner approximations of section IV-A may be used to evaluate the possible end-effector velocities of the manipulator at a given configuration \mathbf{q} . When considering a set of configurations $[\mathbf{q}]$, the manipulator's Jacobian matrix becomes an interval matrix and the associated inner approximations of section IV-B may be used.

At the following configurations

$$[\mathbf{q}] = (0.0, -1.5708, 1.8675) \pm 0.01 \text{ rads} \tag{54}$$

and with the joint velocities $[\dot{\mathbf{q}}] = [\dot{\mathbf{q}}_{lim}]$ the inner approximations – associated polytope with $r = 0.9136$, largest n-cube centered at the origin with $r = 0.4748$ m/s, and largest n-ball centered at the origin with $r = 0.6657$ m/s approximations of $\Omega_{\forall\exists}(\mathbf{J}([\mathbf{q}]), [\dot{\mathbf{q}}])$ – are given in Figure 7.

Analysis of the manipulator's velocity capabilities is performed by bisecting the joint configuration space (given in terms of an interval vector $[\mathbf{q}]$) into a set of k sub-intervals $[\mathbf{q}_1], \dots, [\mathbf{q}_k]$ at a desired resolution. Each sub-interval is used to evaluate an inner approximation of the corresponding n-ball centered at the origin, thus the continuous configuration space is completely explored without sampling. Plots of the values of r throughout the configuration space (see Figure 7) can be quickly generated and due to the independence of each sub-interval the performance can benefit significantly from parallel computing. Furthermore, if given a desired velocity capability, a branch-and-bound loop can iteratively refine the joint configuration space $[\mathbf{q}]$ finding all satisfying configurations for a desired resolution. This has applications for trajectory planning in the redundant configuration space, where feasibility requires a connected path of satisfying configurations and planning involves selection of an optimal path [42]. Note that the use of outer approximations of the manipulator's velocity capabilities, which are outside of the scope of this current work, can provide a complementary test allowing to quickly

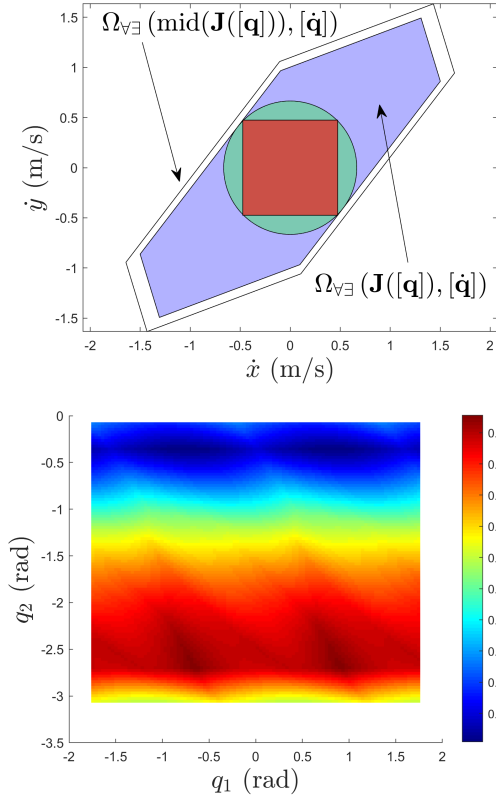


Fig. 7. (Top) The associated polytope, largest n-cube centered at the origin, and largest n-ball centered at the origin for the example velocity problem $\Omega_{\forall\exists}(\mathbf{J}([\mathbf{q}]), [\dot{\mathbf{q}}])$. (Bottom) A plot of the values of r throughout the configuration space (with $[q_3] = 1.8675 \pm 0.01$ rads) for the largest n-ball centered at the origin.

remove non-satisfying configurations from the branch-and-bound loop.

Considering the relationship between the velocity kinematics model (2) and manipulability, the largest n-cube and n-ball inner approximations can be used to rapidly evaluate manipulability measures that correspond to given sets of robot states.

B. Evaluating the Kinetostatic Problem

Another necessary procedure when analyzing the performance of a robotic manipulator is to evaluate the possible end-effector wrenches through the mapping of known joint torque/force limits with (3). The kinetostatic problem is of the class $\Sigma_{\forall\exists}(\mathbf{A}, [\mathbf{b}])$ and the associated inner approximations of section IV-D may be used to evaluate the possible end-effector wrenches of the manipulator at a given configuration \mathbf{q} . For a set of configurations $[\mathbf{q}]$, the kinetostatic problem is of the class $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ and the associated inner approximations of section IV-C may be used.

At the same configurations (54) and with the joint torques $[\boldsymbol{\tau}] = [\boldsymbol{\tau}_{lim}]$, the inner approximations – associated polytope, largest n-cube centered at the origin with $r = 67.3479$ N, and largest n-ball centered at the origin with $r = 85.1640$ N

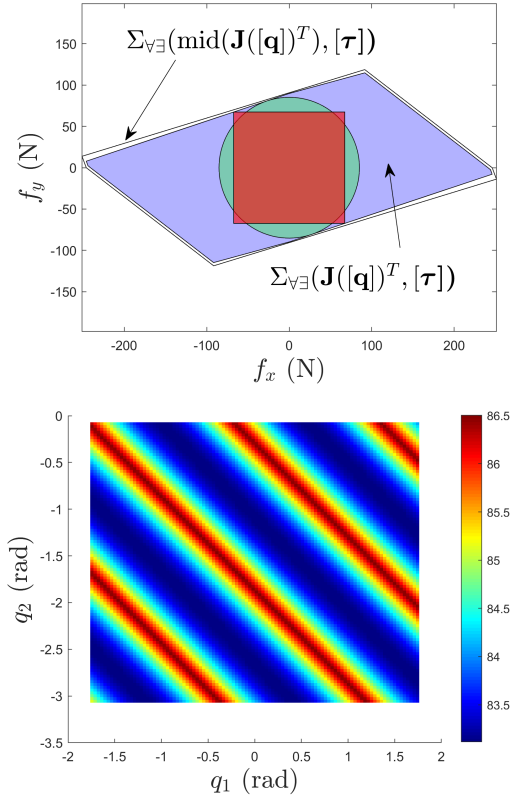


Fig. 8. (Top) The associated polytope, largest n-cube centered at the origin, and largest n-ball centered at the origin for the example kinetostatic problem $\Sigma_{\forall\exists}(\mathbf{J}([\mathbf{q}]^T, [\boldsymbol{\tau}])$. (Bottom) A plot of the values of r throughout the configuration space (with $[q_3] = 1.8675 \pm 0.01$ rads) for the largest n-ball centered at the origin.

approximations of $\Sigma_{\forall\exists}(\mathbf{J}([\mathbf{q}]^T, [\boldsymbol{\tau}])$ – are given in Figure 8. In addition, the joint configuration space is bisected into sub-intervals and inner approximations of the corresponding n-ball centered at the origin allow to analyze the manipulator's force capabilities.

C. Evaluating the Forward Acceleration Problem

The local operational acceleration capabilities can be determined from the forward acceleration model (4). The problem can be split into two interval linear systems of equations where each system needs to be solved to give a corresponding set of operational accelerations (see Tables I). The Minkowski sum of the two sets then gives the local operational acceleration capabilities. Inner polytope approximations of the two sets of class $\Omega_{\forall\exists}(\mathbf{A}, [\mathbf{x}])$ or $\Omega_{\forall\exists}([\mathbf{A}], [\mathbf{x}])$ are obtained from:

$$\begin{aligned} \ddot{\mathbf{x}}_1 &= \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ \ddot{\mathbf{x}}_2 &= \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} \end{aligned} \quad (55)$$

Since each inner polytope approximation is convex, the Minkowski sum of the two sets is easily computed as the convex hull of all combinations of polytope vertices. The inner approximations (22) and (25) can then be applied.

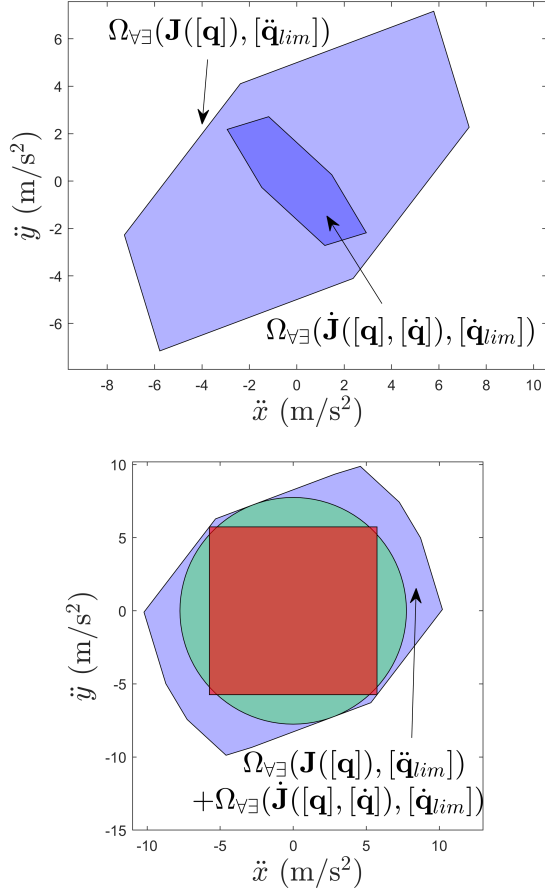


Fig. 9. (Top) The associated inner polytope approximations for $\Omega_{\forall\exists}(\mathbf{J}(\mathbf{q}, \dot{\mathbf{q}}), [\ddot{\mathbf{q}}_{lim}])$ and $\Omega_{\forall\exists}(\mathbf{J}(\mathbf{q}), [\ddot{\mathbf{q}}_{lim}])$. (Bottom) The associated inner approximations for $\Omega_{\forall\exists}(\mathbf{J}(\mathbf{q}, \dot{\mathbf{q}}), [\ddot{\mathbf{q}}_{lim}]) + \Omega_{\forall\exists}(\mathbf{J}(\mathbf{q}), [\ddot{\mathbf{q}}_{lim}])$.

Let $\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{J}(\mathbf{q})$ be evaluated at the same configurations (54) and with the joint velocities

$$[\dot{\mathbf{q}}] = (1.0, 1.0, 1.0) \pm 0.01 \text{ rads/s} \quad (56)$$

As more uncertainties or variabilities in the system are considered, the greater the widths of the intervals in the matrices $[\mathbf{A}]$ become and therefore the greater the widths of the vectors $[\mathbf{x}]$ must be to in order to determine an inner polytope approximation with r greater than zero. Considering the joint state limits $[\dot{\mathbf{q}}_{lim}]$ and $[\ddot{\mathbf{q}}_{lim}]$, the resulting inner polytope approximations of $\Omega_{\forall\exists}(\mathbf{J}(\mathbf{q}, \dot{\mathbf{q}}), [\ddot{\mathbf{q}}_{lim}])$ and $\Omega_{\forall\exists}(\mathbf{J}(\mathbf{q}), [\ddot{\mathbf{q}}_{lim}])$ are $r = 0.8446$ and $r = 0.9237$, respectively (see Figure 9 top). The inner approximations of the operational acceleration capabilities is Minkowski sum of the two sets (see Figure 9 bottom) with: largest n-cube centered at the origin with $r = 5.7386 \text{ m/s}^2$, and largest n-ball centered at the origin with $r = 7.7526 \text{ m/s}^2$. While uncertainties may be largely ignored from a control point of view, accounting for these uncertainties is the only way to truly certify the capabilities of the system.

D. Evaluating Future Dynamic Capabilities

Let the current state of a robot be given by $\{\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}\}$ and let the joint state limits be given by the intervals $\{[\mathbf{q}_{lim}], [\dot{\mathbf{q}}_{lim}], [\boldsymbol{\tau}_{lim}]\}$ and the control limit be given by the interval $[\boldsymbol{\tau}_{lim}]$. Consider a prediction period interval $[t] = [0, t_{max}]$. The future joint torques $[\boldsymbol{\tau}_f]$ are bounded by

$$[\boldsymbol{\tau}_f] = (\boldsymbol{\tau} + [\dot{\boldsymbol{\tau}}_{lim}][t]) \cap [\boldsymbol{\tau}_{lim}] \quad (57)$$

The future joint accelerations $[\ddot{\mathbf{q}}_f]$ are estimated from joint torque at the current state by

$$[\ddot{\mathbf{q}}_f] = \mathbf{M}(\mathbf{q})^{-1} ([\boldsymbol{\tau}_f] - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})). \quad (58)$$

The future joint velocities $[\dot{\mathbf{q}}_f]$ are given by

$$[\dot{\mathbf{q}}_f] = (\dot{\mathbf{q}} + [\ddot{\mathbf{q}}_f][t]) \cap [\dot{\mathbf{q}}_{lim}]. \quad (59)$$

The future joint configurations $[\mathbf{q}_f]$ are given by

$$[\mathbf{q}_f] = \left(\mathbf{q} + [\dot{\mathbf{q}}_f][t] + \frac{1}{2}[\ddot{\mathbf{q}}_f][t]^2 \right) \cap [\mathbf{q}_{lim}]. \quad (60)$$

The joint state limit intersections ensure that the limits are not exceeded during the prediction of the future set of states.

It is clear that joint acceleration capabilities are state dependent and therefore if constant joint acceleration limits are enforced, the controller may either require infeasible accelerations from the system or sub-optimally use its actual capabilities. Furthermore, due to the noise associated with the estimation of the current $\dot{\mathbf{q}}$, enforcing constant joint acceleration limits is difficult in practice. Alternatively, the use of joint torques to evaluate joint acceleration capabilities can provide a much better estimate of the capabilities of the system.

The joint acceleration capabilities achievable during the prediction period $[t]$ can be determined from the configuration-space dynamic model (5). However, the right hand side of (5) should not be overestimated. Therefore, let the effective joint torques be given by $[\boldsymbol{\tau}_e] = [\boldsymbol{\tau}_f + \mathbf{c}\bar{\mathbf{g}}, \bar{\boldsymbol{\tau}}_f + \mathbf{c}\bar{\mathbf{g}}]$, where $[\mathbf{c}\bar{\mathbf{g}}] = -\mathbf{c}([\mathbf{q}_f], [\dot{\mathbf{q}}_f]) - \mathbf{g}([\mathbf{q}_f])$. Inner approximations of the joint acceleration capabilities are then computed from

$$\mathbf{M}([\mathbf{q}_f])\ddot{\mathbf{q}} = [\boldsymbol{\tau}_e]. \quad (61)$$

The problem is of class $\Sigma_{\forall\exists}([\mathbf{A}], [\mathbf{b}])$ and the associated inner approximations of section IV-C may be used.

As an example, let the current states be given by (54) and (56) and

$$\boldsymbol{\tau} = (18.0, 1.0, 2.0) \pm 0.01 \text{ Nm} \quad (62)$$

Selecting the time $[t] = [0, 0.01] \text{ s}$, the corresponding joint acceleration inner approximation polytope, the largest n-cube centered at the origin with $r = 2.3382 \text{ rad/s}^2$, and the largest n-ball centered at the origin with $r = 3.2315 \text{ rad/s}^2$ are shown in Figure 10. For visibility reasons only, the polytope is bounded by the manufacturer's limits $[\dot{\mathbf{q}}_{lim}]$.

The inner polytope approximation of the joint acceleration capabilities represents a set of valid joint accelerations which

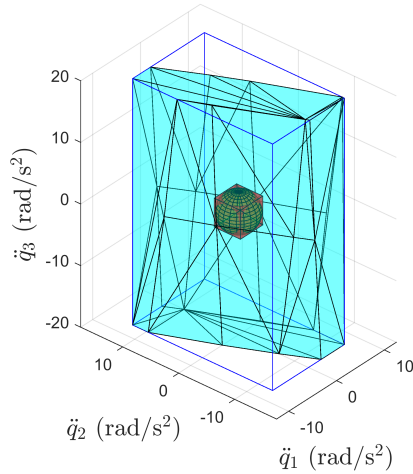


Fig. 10. Future joint acceleration capability inner approximations during a prediction period $[t]$. For visibility, the polytope is bounded by the manufacturer's limits $[\ddot{\mathbf{q}}_{lim}]$.

better describe the future operational accelerations during a prediction period $[t]$. This can be used in the form of inequality constraints inside the robot controller, augmenting the manufacturer's default limits, to ensure that the robot's capabilities are properly considered to select valid control actions.

VI. CONCLUSION

Efficient set-based approaches for computing reliable inner approximations of a robot's common capabilities over sets of joint states and with design parameter uncertainties are presented in this work. These approaches allow to not only consider the temporal evolution of the system over a small time horizon when computing the localized capabilities, but also allow to manage uncertainties to handle an imprecise/variable system describing a family of robotic manipulators.

The use of set-based approaches to evaluate capabilities provides many benefits over conventional approaches. One key benefit is the computation of certified inner approximations of the robot's capabilities, allowing these approaches to be used in offline and online scenarios to improve the safety of robotic systems. The ability to automatically handle design parameter uncertainties and other system uncertainties affecting performance (e.g., inertial parameter errors [43]) allows for the use of simpler imprecise/variable models that accurately describe a complex system (e.g., modeling flexure-jointed mechanisms [18]). Also, the ability to consider sets of joint states allows to locally evaluate capabilities over a small time horizon, such that the robot's performance in the near future can be better understood. The sets of joint states may also be used to analyze the capabilities along continuous trajectories via the use of branch-and-bound methods, as opposed to the standard sampling-based approach, allowing to

properly understand worst-case conditions which can be used to improve the overall performance of the robot.

The low computation costs of many of the proposed inner approximations allow for real-time applications, enabling their use for rapid design analysis as well as their potential integration directly in to the robot controller. Therefore, a future application of this work may be the development of a certifiably safe robot controller which will use a properly formulated imprecise/variable system to safely estimate the robot's true capabilities over a given time horizon. These capabilities may then be used directly in the controller to ensure that each controller action is achievable. Furthermore, by computing and incorporating the robot's deceleration capabilities into the controller, it is possible to enforce strict braking criteria to guarantee that collaborative systems are safe when in close proximity to a human.

REFERENCES

- [1] A. Ibanez, P. Bidaud, and V. Padois, *Optimization-Based Control Approaches to Humanoid Balancing*. Springer Netherlands, 2017, pp. 1–27.
- [2] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.
- [3] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocodyl: An efficient and versatile framework for multi-contact optimal control," in *Proceedings of the IEEE RAS International Conference in Robotics and Automation*, Paris, France, May 2020.
- [4] A. D. Prete, "Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 281–288, 2018.
- [5] S. Rubrecht, V. Padois, P. Bidaud, M. De Broissia, and M. D. S. Simoes, "Motion safety and constraints compatibility for multibody robots," *Autonomous Robots*, vol. 32, no. 3, pp. 333–349, 2012.
- [6] W. Decre, R. Smits, H. Bruyninckx, and J. De Schutter, "Extending itasc to support inequality constraints and non-instantaneous task specification," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 964–971.
- [7] J. G. N. D. Carvalho Filho, E. A. N. Carvalho, L. Molina, and E. O. Freire, "The impact of parametric uncertainties on mobile robots velocities and pose estimation," *IEEE Access*, vol. 7, pp. 69 070–69 086, 2019.
- [8] R. E. Moore, *Interval analysis*, ser. Prentice-Hall series in automatic computation. Englewood Cliffs, NJ: Prentice-Hall, 1966.
- [9] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer London, Aug. 2001.
- [10] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Philadelphia, PA: SIAM, 2009.
- [11] E. D. Popova and M. Hladík, "Outer enclosures to the parametric AE solution set," *Soft Comput.*, vol. 17, no. 8, pp. 1403–1414, 2013.
- [12] S. P. Shary, "Interval gauss-seidel method for generalized solution sets to interval linear systems," *Reliable Computing*, vol. 7, no. 2, pp. 141–155, Apr 2001.
- [13] L. V. Kolev, "A method for outer interval solution of linear parametric systems," *Reliab. Comput.*, vol. 10, no. 3, pp. 227–239, 2004.
- [14] E. D. Popova and W. Krämer, "Inner and outer bounds for the solution set of parametric linear systems," *J. Comput. Appl. Math.*, vol. 199, no. 2, pp. 310–316, 2007.
- [15] A. Neumaier, "The wrapping effect, ellipsoid arithmetic, stability and confidence regions," in *Validation Numerics: Theory and Applications*, R. Albrecht, G. Alefeld, and H. J. Stetter, Eds. Vienna: Springer, 1993, pp. 175–190.
- [16] N. Nedialkov, V. Kreinovich, and S. Starks, "Interval arithmetic, affine arithmetic, Taylor series methods: Why, what next?" *Numerical Algorithms*, vol. 37, pp. 325–336, 12 2004.

- [17] J.-P. Merlet, "Interval analysis for Certified Numerical Solution of Problems in Robotics," *International Journal of Applied Mathematics and Computer Science*, 2009.
- [18] D. Oetomo, D. Daney, B. Shirinzadeh, and J.-P. Merlet, "An Interval-Based Method for Workspace Analysis of Planar Flexure-Jointed Mechanism," *Journal of Mechanical Design*, vol. 131, no. 1, 12 2008.
- [19] M. Gouttefarde, D. Daney, and J.-P. Merlet, "Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 1–13, Feb 2011.
- [20] J. K. Pickard and J. A. Carretero, "An interval analysis method for wrench workspace determination of parallel manipulator architectures," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 40, no. 2, pp. 139–154, 2016.
- [21] J.-P. Merlet and D. Daney, "Appropriate design of parallel manipulators," in *Smart Devices and Machines for Advanced Manufacturing*, L. Wang and J. Xi, Eds. London: Springer, 2008, pp. 1–25.
- [22] M. H. F. Kaloorazi, M. T. Masouleh, and S. Caro, "Collision-free workspace of parallel mechanisms based on an interval analysis approach," *Robotica*, vol. 35, no. 8, p. 1747–1760, 2017.
- [23] J.-P. Merlet, "Interval analysis and robotics," in *Robotics Research*, M. Kaneko and Y. Nakamura, Eds. Berlin, Heidelberg: Springer, 2011, pp. 147–156.
- [24] J.-P. Merlet and P. Donelan, "On the regularity of the inverse jacobian of parallel robots," in *Advances in Robot Kinematics*, J. Lenarčič and B. Roth, Eds. Dordrecht: Springer Netherlands, 2006, pp. 41–48.
- [25] S. Bouchard, C. M. Gosselin, and B. Moore, "On the ability of a cable-driven robot to generate a prescribed set of wrenches," *ASME. J. Mechanisms Robotics*, vol. 2, no. 1, 2009.
- [26] J. A. Carretero and C. M. Gosselin, "Wrench capabilities of cable-driven parallel mechanisms using wrench polytopes," in *Proceeding of the 2010 IFToMM Symposium on Mechanism Design for Robotics*, Universidad Panamericana, Mexico City, Mexico, September 28–30 2010.
- [27] M. Gouttefarde and S. Krut, "Characterization of parallel manipulator available wrench set facets," in *Advances in Robot Kinematics: Motion in Man and Machine*, J. Lenarčič and M. M. Stanisic, Eds. Dordrecht: Springer Netherlands, 2010, pp. 475–482.
- [28] S. P. Shary, "A new technique in systems analysis under interval uncertainty and ambiguity," *Reliable Computing*, vol. 8, no. 5, pp. 321–418, Oct 2002.
- [29] G. M. Ziegler, *Lectures on Polytopes*, ser. Graduate Texts in Mathematics. Berlin: Springer, 1995, vol. 152.
- [30] R. E. Wendell and W. Chen, "Tolerance sensitivity analysis: Thirty years later," *Croatian Oper. Res. Rev.*, vol. 1, pp. 12–21, 2010.
- [31] M. Hladík, "Tolerance analysis in linear systems and linear programming," *Optim. Methods Softw.*, vol. 26, no. 3, pp. 381–396, 2011.
- [32] —, "Weak and strong solvability of interval linear systems of equations and inequalities," *Linear Algebra Appl.*, vol. 438, no. 11, pp. 4156–4165, 2013.
- [33] J. Rohn, "Linear programming with inexact data is NP-hard," *ZAMM, Z. Angew. Math. Mech.*, vol. 78, no. Supplement 3, pp. S1051–S1052, 1998.
- [34] —, "Solvability of systems of interval linear equations and inequalities," in *Linear Optimization Problems with Inexact Data*, M. Fiedler et al., Ed. Boston, MA: Springer, 2006, ch. 2, pp. 35–77.
- [35] S. P. Shary, "Solving the linear interval tolerance problem," *Mathematics and Computers in Simulation*, vol. 39, no. 1, pp. 53 – 85, 1995.
- [36] E. D. Popova, "Inner estimation of the parametric tolerable solution set," *Comput. Math. Appl.*, vol. 66, no. 9, pp. 1655–1665, 2013.
- [37] J. Rohn, "Inner solutions of linear interval systems," in *Proceedings of the International Symposium on Interval Mathematics on Interval Mathematics 1985*. Berlin, Heidelberg: Springer, 1986, pp. 157–158.
- [38] P. Chiacchio, Y. Bouffard-Vercelli, and F. Pierrot, "Evaluation of force capabilities for redundant manipulators," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, April 1996, pp. 3520–3525.
- [39] A. Skuric, V. Padois, and D. Daney, "On-line force capability evaluation based on efficient polytope vertex search," 2020.
- [40] J. K. Pickard, "Software: Inner Approximation Algorithms for Interval Linear Systems of Equations," June 2020. [Online]. Available: https://figshare.com/articles/Software_Inner_Approximation_Algorithms_for_Interval_Linear_Systems_of_Equations/12472748
- [41] J. K. Pickard, J. A. Carretero, and J.-P. Merlet, "Appropriate analysis of the four-bar linkage," *Mechanism and Machine Theory*, vol. 139, pp. 237 – 250, 2019.
- [42] L. Jaulin, "Path planning using intervals and graphs," *Reliable Computing*, vol. 7, pp. 1–15, 02 2001.
- [43] N. Giftsun, A. Del Prete, and F. Lamiroux, "Robustness to Inertial Parameter Errors for Legged Robots Balancing on Level Ground," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017)*, Madrid, Spain, Jul. 2017.



Joshua K. Pickard received a B.Sc.E. degree in mechanical engineering with a mechatronics option in 2012, and a Ph.D. in mechanical engineering on uncertainty analysis and mechanism synthesis in 2018 from the University of New Brunswick, Fredericton, New Brunswick, Canada. From 2018 to 2020 he was a postdoctoral fellow at Inria Bordeaux Sud-Ouest, Bordeaux, France, with the Auctus Team working on a modeling and analysis framework for imprecise and variable kinematic chains for the study of human motor-variabilities. Currently, he is a postdoctoral fellow at the University of New Brunswick working on research related to factories of the future and Industry 4.0.



Vincent Padois is a senior research scientist at Inria, centre Bordeaux Sud-Ouest in the Auctus team (Talence, France). From 2007 to 2020, he was an associate professor of Robotics at Sorbonne Université and ISIR (Paris, France). From 2006 to 2007 he was a postdoctoral researcher at the Stanford Artificial Intelligence Laboratory in the group of Prof. O. Khatib (Stanford, USA). He received his Ph.D. degree in Robotics and Automatic Control from INP de Toulouse in 2005. His main research interest is robot control in constrained contexts and dynamic environments, including wheeled mobile manipulators, humanoid robots and collaborative robots. Beyond control, his research activities in collaborative robotics are focused on virtual human models for the ergonomic quantification of the physical assistance of collaborative robots. He is also involved in research activities aiming at bridging the gap between adaptation and decision making techniques and model-based control through the use of machine learning and evolutionary algorithms.



Milan Hladík is an associate professor at the Department of Applied Mathematics of the Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic. His research interests include interval computation, numerical analysis, optimization and operations research. He is a member of the editorial board of four international journals, including European Journal of Operational Research.



David Daney is a researcher at Inria Bordeaux Sud-Ouest, and leads the Project-Team Auctus, Inria-IMS (Univ. Bordeaux, Bordeaux INP, CNRS UMR 5218), Talence, France. He received the B. Eng in applied mathematics and computer science from University of Toulouse in 1996 and the M.Sc. and Ph.D. degrees in robotics from the University of Nice, Sophia Antipolis, France, in 1997 and 2000, respectively. He spent two years as a postdoctoral associate with C.M.W., France, to design a robotics system (2000) and with the LORIA Lab, Nancy,

France, on computer arithmetic (2001). In 2002, he was an invited researcher with McGill University, Rutgers University, Laval University. From 2003 to 2013, he has been an Inria Research Scientist with the Inria Sophia Antipolis Research Center. His research interests are in the areas of parallel robotics, calibration, interval analysis, mechanical design, and assistive systems. Since 2013, he has been at the Inria research center in Bordeaux and in 2017 he founded the Auctus team in collaboration with the ENSC and the IMS. The aim of this team is to design collaborative robotic systems for man at work in industrial environments.