Artificial Intelligence 211 (2014) 34–50

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

A branch and prune algorithm for the computation of generalized aspects of parallel robots [☆]

S. Caro [a], D. Chablat [a], A. Goldsztejn [b,*], D. Ishii [c], C. Jermann [d]

[a] CNRS, IRCCyN, Nantes, France
[b] CNRS, LINA (UMR-6241), Nantes, France
[c] Tokyo Institute of Technology, Tokyo, Japan
[d] Université de Nantes, LINA (UMR-6241), Nantes, France

A R T I C L E   I N F O

Article history:
Received 18 April 2013
Received in revised form 6 February 2014
Accepted 8 February 2014
Available online 18 February 2014

Keywords:
Numerical constraints
Parallel robots
Singularities
Generalized aspects

A B S T R A C T

Parallel robots enjoy enhanced mechanical characteristics that have to be contrasted with a more complicated design. In particular, they often have parallel singularities at some poses, and the robots may become uncontrollable, and could even be damaged, in such configurations. The computation of the connected components in the set of nonsingular reachable configurations, called generalized aspects, is therefore a key issue in their design. This paper introduces a new method, based on numerical constraint programming, to compute a certified enclosure of the generalized aspects. Though this method does not allow counting their number rigorously, it constructs inner approximations of the nonsingular workspace that allow commanding parallel robots safely. It also provides a lower-bound on the exact number of generalized aspects. It is moreover the first general method able to handle any parallel robot in theory, though its computational complexity currently restricts its usage to robots with three degrees of freedom. Finally, the constraint programming paradigm it relies on makes it possible to consider various additional constraints (e.g., collision avoidance), making it suitable for practical considerations.

1. Introduction

Mechanical manipulators, commonly called robots, are widely used in the industry to automatize various tasks. They are mechanical assemblies of rigid links connected by mobile joints. Some joints are actuated and they allow commanding the robot operating link, called its end-effector (or platform). One key characteristic of a robot is its reachable workspace, informally defined as the set of poses its end-effector can reach. Indeed, its size defines the scope of operational trajectories the robot can perform. The workspace can be computed from the set of possible command inputs using the kinematic model of the robot, a system of equations relating the commands and the pose coordinates. The size of this system is often referred to as the degrees of freedom (DOF) of the robot.

Robots comply with either a serial or a parallel (or possibly a hybrid) assembly, whether its links are connected in series or in parallel. Parallel robots [1,2] present several advantages with respect to serial ones: They are naturally stiffer, leading to better accuracy with larger loads, and allow high speed motions. These advantages are contrasted by a more

[☆] This paper is an invited revision of a paper first published at the 18th International Conference on Principles and Practice of Constraint Programming (CP 2012).
[*] Corresponding author.

http://dx.doi.org/10.1016/j.artint.2014.02.001
0004-3702/© 2014 Elsevier B.V. All rights reserved.

complicated design that yields difficulties for the computation and the analysis of their workspace. First, one pose of the robot's end-effector may be reached by several actuated joint commands (which correspond to different *working modes*), and conversely one input command may lead to several poses of its end-effector (which correspond to different *assembly modes*). Second, parallel robots generally have parallel singularities [3], i.e., specific configurations where they become uncontrollable and can even be damaged.

One central issue in designing parallel robots is to compute its nonsingular workspace, together with the corresponding commands, so that the robot can be safely operated. This amounts to computing the connected components of the set of nonsingular configurations, called generalized aspect in [4]. This computation must be certified in terms of non-singularity and connectivity in order to guarantee safe operations. Few frameworks provide such certifications, among which algebraic computations and interval analysis. Algebraic methods are in general too expensive and apply only for polynomial systems. Still, the cylindrical algebraic decomposition was used in [5] with a connectivity analysis limited to robots with 2 DOFs. Though generalized aspects are mathematical objects that cannot, in general, be computed exactly using numerical methods, interval analysis allows the rigorous computation of some approximation. It was used in [6] for robots having a single solution to their inverse kinematic problem; Though limited, this method can still tackle important classes of robots like the Stewart platform. A quad-tree with certification of nonsingularity was built in [7] for some planar robots with 2 DOFs; This method can be extended to higher dimensional robots, but it requires the a priori separation of working modes by ad hoc inequalities, and is not certified with respect to connectivity. Finally, the two works [8,9] propose algorithms based on interval analysis to analyze the connectivity of set defined by inequalities constraints, but cannot be extended to equality constraints. In particular, the developments presented in the present paper somehow extend the interval-based path planning method proposed in [8] for sets defined by inequality constraints only, to manifolds defined by equality, disequality and inequality constraints.

In this paper we propose a branch and prune algorithm incorporating the certification of the solutions and of their connectivity. This allows a fully automated and certified computation of what we call *connected sets of nonsingular configurations* (CSNCs), i.e., certified approximations of generalized aspects, from the model of arbitrary parallel robots, including robots with multiple solutions to their direct and inverse kinematic problems, without requiring any a priori study to separate their working modes. Though the proposed method does not allow counting the number of CSNCs rigorously, it constructs inner approximations of the nonsingular workspace that allow commanding parallel robots safely. Although less important in practice, a more accurate and costly connectivity analysis is also proposed, which enables separating non-connected CSNCs, hence providing a lower-bound on the exact number of generalized aspects. The algorithm is applicable to robots with an arbitrary number of DOF, although the complexity of the computations currently restricts its application to robots with three DOFs. It is also very flexible as it can naturally take into account additional constraints such as, e.g., arm collisions, obstacle avoidance or joint limits. It is thus the first method able to handle such a large class of robots for the problem of computing connected sets of nonsingular configurations. Its main limitation is its performances, due to the combinatorial explosion of the number of computed boxes with the dimension of the problem and the prescribed computational precision. As a consequence, we have applied it to planar robots only at the moment.

A motivating example is presented in Section 2 followed by some preliminaries about numerical constraint programming and robotics in Section 3. The proposed algorithm for certified singularity free connected components computation is presented in Section 4. Finally, experiments on planar robots with 2 and 3 degrees of freedom are presented in Section 5.

Notations. Boldface letters denote vectors. Thus $\mathbf{f}(\mathbf{x}) = 0$ denotes a system of equations $\mathbf{f}$ on a vector of variables $\mathbf{x}$: $f_1(x_1, \ldots, x_n) = 0, \ldots, f_k(x_1, \ldots, x_n) = 0$. The Jacobian matrix of $\mathbf{f}(\mathbf{x})$ with respect to the subset $\mathbf{x}'$ of the variables $\mathbf{x}$ is denoted $\mathbf{F}_{\mathbf{x}'}(\mathbf{x})$. Interval variables are denoted using bracketed symbols, e.g., $[x] = [\underline{x}, \bar{x}] := \{x \in \mathbb{R} \mid \underline{x} \leqslant x \leqslant \bar{x}\}$. Hence, $[\mathbf{x}]$ is an interval vector (box) and $[A] = ([a_{ij}])$ is an interval matrix. $\mathbb{IR}$ denotes the set of intervals and $\mathbb{IR}^n$ the set of n-dimensional boxes. For an interval $[x]$, we denote $\mathrm{wid}[x] := \bar{x} - \underline{x}$ its width, $\mathrm{int}[x] := \{x \in \mathbb{R} \mid \underline{x} < x < \bar{x}\}$ its interior, and $\mathrm{mid}[x] := (\underline{x} + \bar{x})/2$ its midpoint. These notations are extended to interval vectors.

2. Motivating example

Description. Consider the simple $\underline{\mathrm{P}}$RRP[1] planar robot depicted in Fig. 1 (left), which involves two prismatic joints (gray rectangles) sliding along two perpendicular directions. These prismatic joints are connected through three rigid bars (black lines) linked by two revolute joints (circles) that allow free rotations between the rigid bars. The lengths of the prismatic joints are respectively denoted by x and q, the end-effector pose x being along the horizontal direction and the command q corresponding to the height along the vertical direction. Fig. 1 (left) shows one generic configuration of the robot. Note that there is another symmetric (negative) pose x associated to the same command q, which is typical of parallel robots. From this configuration, every (vertical) change in q induces a unique corresponding (horizontal) change in x, hence this configuration is nonsingular. Fig. 1 (right) shows two singular configurations. In the plain green pose (where the robot's main rigid bar is horizontal), increasing or decreasing the command q both entails a decrease of x. In the dashed red pose (where

[1] In robotics, manipulators are typically named according to the sequence of joints they are made of, e.g., P stands for prismatic joint and R stands for revolute joint, actuated joints being underlined.