An Open-Source Toolbox for Motion Analysis of Closed-chain Mechanisms

L. Ros, J. M. Porta, O. Bohigas, M. Manubens, C. Rosales, and L. Jaillet

Abstract Many situations in Robotics require an effective analysis of the motions of a closed-chain mechanism. Despite appearing very often in practice (e.g. in parallel manipulators, reconfigurable robots, or molecular compounds), there is a lack of general tools to effectively analyze the complex configuration spaces of such systems. This paper describes the CUIK suite, an open-source toolbox for motion analysis of general closed-chain mechanisms. The package can determine the motion range of the whole mechanism or of some of its parts, detect singular configurations leading to control or dexterity issues, or find collision- and singularity-free paths between given configurations. The toolbox is the result of several years of research and development within the *Kinematics and Robot Design* group at IRI, Barcelona, and is available under GPLv3 license from `http://www.iri.upc.edu/cuik`.

Key words: Kinematic Constraints, Motion Analysis and Planning, Branch-and-Prune, Higher-dimensional Continuation.

1 Introduction

The notion of configuration space (C-space) is fundamental in Robotics. It allows designing motion planning algorithms for broadly-defined classes of robots or mechanisms, without worrying about their particular geometry or multibody structure. In most Robotics textbooks, this notion is introduced for open-chain mechanisms, where the C-space has an explicit global parametrization. In this way, C-spaces are readily understood and algorithms operating on them can be readily defined. In many cases, however, the C-space can have a more intricate structure, and its analysis is by no means trivial. This is the case of parallel manipulators, re-

L. Ros · J. M. Porta · O. Bohigas · M. Manubens · C. Rosales · L. Jaillet

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, e-mail: {lros,porta,obohigas,mmanuben, crosales,ljaillet}@iri.upc.edu

configurable mechanisms, or robots working under geometric or contact constraints, but similar problems arise when exploring the motions of a protein, or when assembling parts using spatial constraints. Common to these problems is the fact that the feasible configurations are implicitly defined by a nonlinear system of equations

$$\mathbf{F}(\boldsymbol{x}) = \mathbf{0}, \tag{1}$$

and the goal is to understand the motion capabilities of the mechanism by analyzing the solution set $\mathscr{C}$ of this system.

In an extreme case, $\mathscr{C}$ is composed of isolated points only. This is what happens when solving forward or inverse kinematics problems in robot manipulators. Historically, the preferred approach has been to tackle these problems by reducing Eq. (1) to a resultant polynomial, and then solving this polynomial using well-established methods for the univariate case. However, this approach may introduce extraneous roots, and the degree of the resultant grows rapidly with the size and complexity of the mechanism. The CUIK suite circumvents these issues by adopting an opposite approach. Instead of reducing Eq. (1) to a univariate polynomial, we formulate it as a larger system involving linear and quadratic equations. This allows the application of an efficient branch-and-prune technique to fully isolate $\mathscr{C}$ at the desired accuracy. In comparison to general toolboxes for polynomial constraint solving [12, 24], the CUIK suite sacrifices generality to gain simplicity and efficiency in the implementation. Opposite to [24], moreover, it directly isolates the real roots instead of the complex ones, even if they form positive-dimensional sets.

Branch-and-prune methods are exhaustive, which broadens their range of applicability, but in problems such as path planning, it may be sufficient to explore only those configurations that are path-connected to a given point. To this end, the CUIK suite implements higher-dimensional continuation tools allowing to trace arbitrary, implicitly-defined manifolds [6]. Note that while several packages provide state-of-the-art path planning methods, they are oriented to open-chain robots [11, 22, 21, 23], or to particular classes of closed-chain devices [10]. The CUIK suite complements these packages by providing new methods to deal with the general closed-chain case.

The rest of the paper describes the numerical methods implemented in the toolbox, and the several higher-level algorithms relying on them.

2 Branch-and-prune methods

Branch-and-prune methods solve Eq. (1) by refining successive box approximations of $\mathscr{C}$. We outline them for planar mechanisms, but general spatial mechanisms can also be treated with the proper extensions [16].

Initially, the kinematic equations must be formulated in a form allowing simple pruning operations. In our case we depart from the formulation in Fig. 1, which leads to polynomial equations of a simple form with little manipulation. If J_i is a revolute

joint connecting links L_j and L_k, the assembly constraint is equivalent to imposing the coincidence of two points on the axis of the joint, P_i and Q_i, respectively fixed to L_j and L_k. If J_i is prismatic, we further define a unit vector aligned with the joint, d_i, and force P_i to lie on the line of L_k defined by Q_i and d_i, fixing the orientation of L_j relative to L_k. By assigning a reference frame $\mathscr{F}_j$ to every link L_j, and taking $\mathscr{F}_1$ to be the absolute frame, these conditions can then be written as shown in the bottom of the figure.

The resulting equations can be algebraized by performing the term substitutions $s_j = \sin\theta_j$ and $c_j = \cos\theta_j$, and introducing the equations $s_j^2 + c_j^2 = 1$, which leads to a formulation of Eq. (1) in which the scalar equations are either linear in x, or take one of the forms $x_k = x_i x_j$ or $x_k = x_i^2$. It is easy to see that the variables in x can only take values within prescribed intervals, so that one can define an initial box $\mathscr{B}$ bounding the location of all solutions of Eq. (1) from the Cartesian product of such intervals.

The algorithm for solving Eq. (1) iteratively removes portions of $\mathscr{B}$ that contain no solution. To this end, we use the linear equations in Eq. (1), and linearly-relaxed versions of the equations $x_k = x_i x_j$ and $x_k = x_i^2$ [Fig. 2 (a)], in order to define a polytope $\mathscr{P}$ bounding the solution set within $\mathscr{B}$. This box is then reduced by computing the smallest axis-aligned orthotope circumscribing $\mathscr{P}$, using the Simplex method. The process is repeated until (1) the box is either reduced to an empty set, in which case it contains no solution, or (2) it is sufficiently small, in which case it is returned as a *solution* box, or (3) it cannot be further reduced, in which case it is bisected into two sub-boxes. The whole process is recursively applied to the new sub-boxes, until a fine-enough box approximation of the solution set is finally obtained [Fig. 2 (b)].

The tools of the *CUIK* suite automate the previous process. They obtain the equations from a high-level file describing the mechanism, compute the solutions
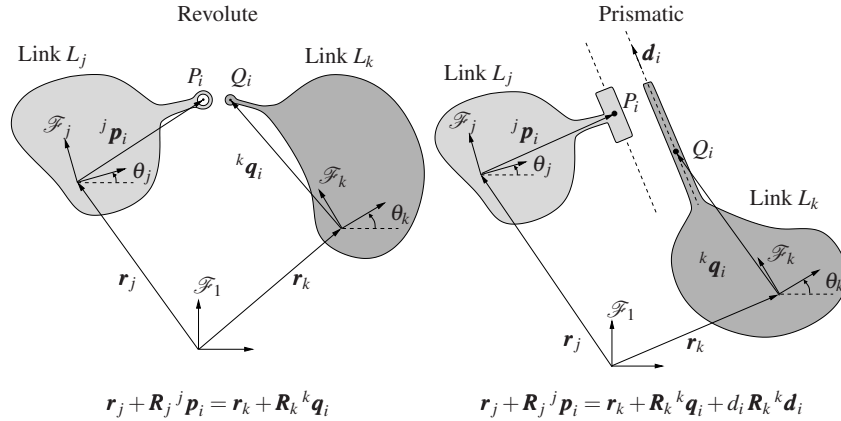

Fig. 1 Assembly constraints for planar pairs. In the equations, r_j is the position of $\mathscr{F}_j$ relative to $\mathscr{F}_1$, R_j is the rotation matrix of angle θ_j, $^j p_i$ is the position of P_i in $\mathscr{F}_j$, and $^k d_i$ provides d_i in the basis of $\mathscr{F}_k$. $R_j = R \cdot R_k$ in the prismatic joint, where R is a constant rotation matrix.
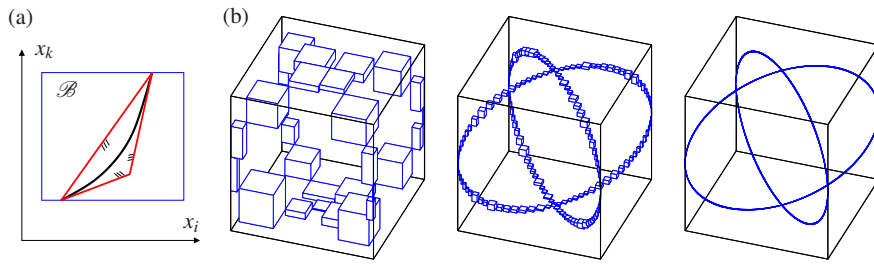
(a)

(b)

Fig. 2 (a) A linear relaxation is a collection of half-spaces bounding the graph of a given equation within the box $\mathscr{B}$ of interest. The figure shows a relaxation of $x_k = x_i^2$ involving three half-spaces. (b) Progression of the branch-and-prune method on a one-dimensional C-space.
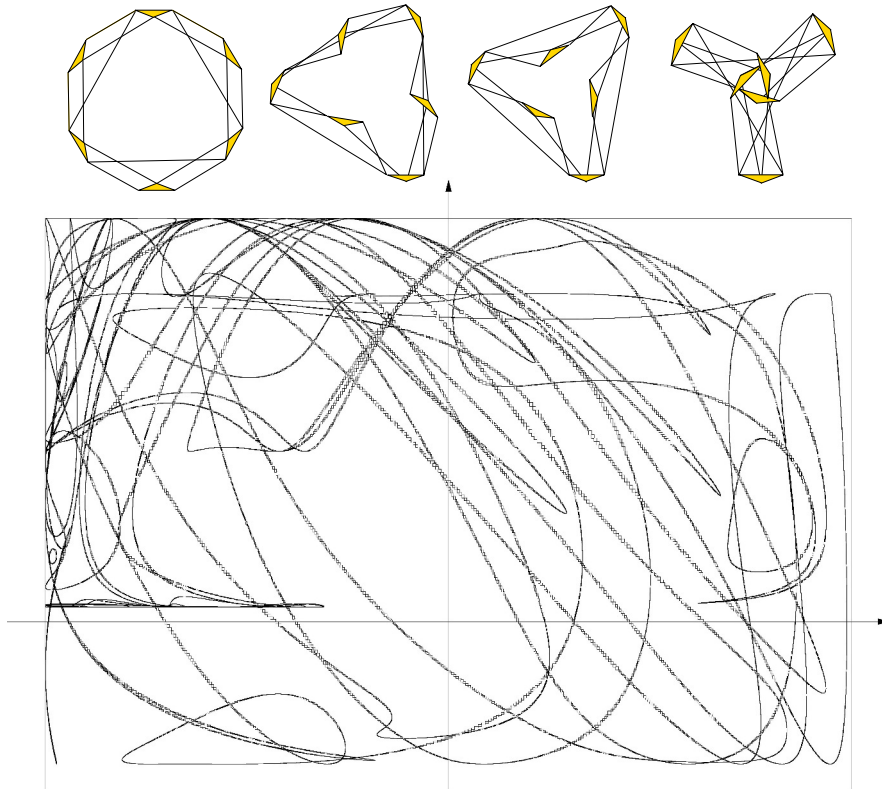

Fig. 3 Top: The rigid *wheel* linkage on the left involves 10 loops. For the shown bar lengths, the tools of the CUIK suite determine that it can be assembled into 19 different configurations, including the four ones shown herein. Bottom: The C-space of the wheel linkage with one bar removed, as computed by the mentioned tools. The C-space involves one isolated point and fifteen one-dimensional components, which are here projected onto the sines of two link angles. A resultant polynomial describing such curve is expected to be of a very high degree.

in single- or multiple-CPU machines, and allow the visualization of the results by producing 2- or 3-dimensional plots. The operations involving ranges are implemented using interval arithmetic, and the Simplex program outputs are adjusted in order not to lose solutions. The example in Fig. 3 is simple, but the tools have proved successful in analysing the C-spaces of general 6R robots and Stewart platforms [16], multiloop molecules [17], robot hands with contact constraints [20], or complex multi-loop linkages like the one in Fig. 3. To compare, resultant methods in the plane are finding their limit in mechanisms of much smaller size [18].

C-spaces of robotic systems typically exhibit singularity subsets. These are loci of critical configurations where control or dexterity losses arise, leading to malfunction or a breakage of the structure [25]. These loci provide the boundaries of the task and joint workspaces too, and all possible motion barriers within them [3]. By adequately defining the equations passed to the solver, the CUIK suite can isolate any of such loci [3, 5, 2], becoming the first general tool able to do so, up to the authors' knowledge.

3 Continuation methods

Continuation methods generate atlases of the C-space region that is path connected to a given point. To see how such atlases are constructed, let x_i be an initial point in $\mathscr{C}$. The tangent space of $\mathscr{C}$ at $\mathbf{x}_i$, $\mathscr{T}_i$, can be parametrized by

$$x_j^i = x_i + \Phi_i u, \tag{2}$$

where Φ_i is a matrix providing an orthonormal basis of $\mathscr{T}_i$, and u is a parameter vector with the same dimension as $\mathscr{C}$. By choosing a value for u in Eq. (2) we obtain a point $x_j^i \in \mathscr{T}_i$, which can be projected down to $\mathscr{C}$ by solving the system formed by $\mathbf{F}(x_j) = \mathbf{0}$ and $\Phi_i^\top (x_j - x_j^i) = \mathbf{0}$, which provides the point $x_j \in \mathscr{C}$ lying in the normal line through x_j^i [Fig. 4 (a)]. The point x_j is then used to define a new chart that is coordinated with the previous chart, and the process is iterated until the whole component of $\mathscr{C}$ reachable from x_i gets fully covered [Fig. 4 (b)]. The construction
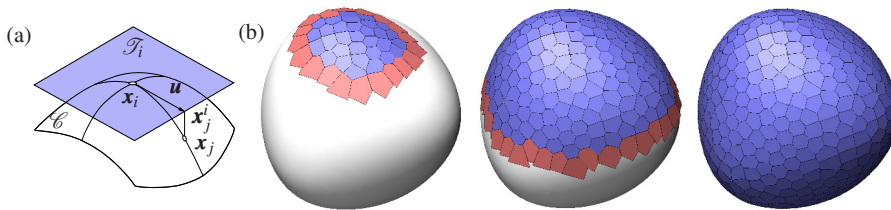

Fig. 4 (a) A chart is used to obtain new C-space points by projecting points from the tangent space. (b) Progress of the atlas construction method on a manifold. Red polygons represent the charts to be extended in subsequent iterations.
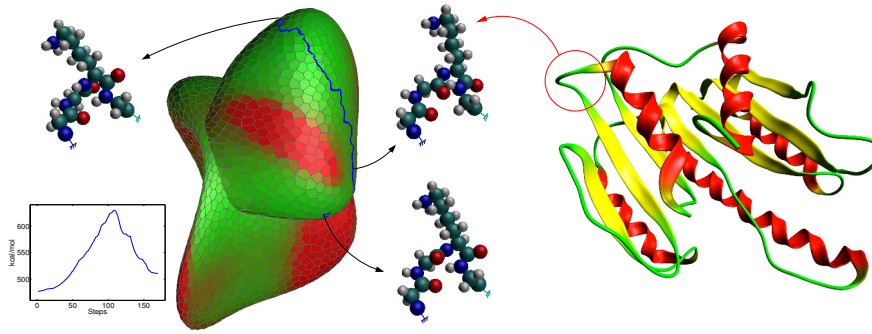
Fig. 5 A low-cost path (in blue) computed in the conformational space of a loop of the FTSJ protein of Escherichia Coli (in ribbon diagram on the right). The cost is the potential energy of each conformation. The insets show the initial conformation, the transition state (i.e., the conformation with the highest potential energy along the path), and the final conformation. Only the atoms in the loop are shown in such conformations. The plot shows the energy profile along the transition path.

of a whole atlas is typically fast in 1- or 2-dimensional C-spaces, which allows tackling difficult optimization problems involving large multibody systems [19].

To solve path planning problems, the CUIK suite exploits the fact that an atlas defines a roadmap in configuration space, whose nodes are located at the chart centers, and whose edges are given by the collision-free transitions between neighboring charts. The roadmap can be used to resolve multiple planning queries between different configurations. However, for cases where only one query needs to be resolved, the suite provides a method to construct only the charts required to define a short path between the given configurations [15]. Based on this tool, singularity-free, resolution-complete path planners for general closed-chain [1] and cable-driven manipulators [4] have been developed, solving problems with no prior satisfactory solution.

Resolution-complete strategies can be inefficient in cluttered environments, and they do not scale gracefully to higher dimensions. To avoid these weaknesses, the CUIK suite implements a sampling method where a partial atlas is used to extend a rapidly-exploring random tree (RRT), which in turn is exploited to decide extension directions for the atlas [9]. Using this technique it is possible to solve problems in pretty high dimensions in a few seconds. Although the paths generated with RRT-like algorithms are typically jerky, the CUIK suite provides procedures to smooth them, and to generate near-optimal paths when there is a cost function defined over the C-space. If the cost is defined for each configuration, the suite implements an extended version of the T-RRT algorithm in [7]. For instance, Fig. 5 shows a low cost path computed by this method in the case of a short loop of a protein [14]. If the cost is associated with the length of the path, the suite adapts the RRT* asymptotically-optimal path planner to the case of implicitly-defined C-spaces [8].

4 Conclusions

This paper has described the CUIK suite, a comprehensive set of tools to analyze configuration spaces implicitly defined by systems of kinematic equations. We provided a brief account of the implemented techniques and their possible applications. Since problems involving kinematic constraints are ubiquitous in Robotics, the suite may potentially be used to address many other problems beyond those described in the paper, in contexts like robot positioning and mapping [13], motion analysis and synthesis of robot formations, tensegrity or deployable structures, or programmable surfaces, to name a few. The suite is an open source package under continuous development, and we invite the community to use it, and to help us improve it by sending feedback and suggestions.

Acknowledgments

References

[1] Bohigas, O., Henderson, M.E., Ros, L., Manubens, M., Porta, J.M.: Planning singularity-free paths on closed-chain manipulators. IEEE Transactions on Robotics Conditionally accepted

[2] Bohigas, O., Manubens, M., Ros, L.: Singularities of non-redundant manipulators: A short account and a method for their computation in the planar case Submitted (under minor review)

[3] Bohigas, O., Manubens, M., Ros, L.: A complete method for workspace boundary determination on general structure manipulators. IEEE Transactions on Robotics **28**(5), 993–1006 (2012)

[4] Bohigas, O., Manubens, M., Ros, L.: Navigating the wrench-feasible C-space of cable-driven hexapods. In: T. Bruckmann, A. Pott (eds.) Cable-Driven Parallel Robots, *Mechanisms and Machine Science*, vol. 12, pp. 53–68. Springer (2013)

[5] Bohigas, O., Zlatanov, D., Ros, L., Manubens, M., Porta, J.M.: Numerical computation of manipulator singularities. In: IEEE International Conference on Robotics and Automation, pp. 1351–1358 (2012)

[6] Henderson, M.E.: Multiple parameter continuation: Computing implicitly defined k-manifolds. International Journal of Bifurcation and Chaos **12**(3), 451–476 (2002)

[7] Jaillet, L., Cortés, J., Siméon, T.: Sampling-based path planning on configuration-space costmaps. IEEE Transactions on Robotics **26**(4), 635–646 (2010)

[8] Jaillet, L., Porta, J.M.: Asymptotically-optimal path planning on manifolds. In: Robotics: Science and Systems, pp. 145–152 (2012)

[9] Jaillet, L., Porta, J.M.: Path planning under kinematic constraints by rapidly exploring manifolds. IEEE Transactions on Robotics **29**(1), 105–117 (2013)

[10] LAAS-CNRS: Move 3D. http://www.openrobots.org/wiki/move3d

[11] Lavalle, S.: The Motion Strategy Library. http://msl.cs.uiuc.edu/msl

[12] Merlet, J.P.: The Alias software. ftp://ftp-sop.inria.fr/coprin/ALIAS

[13] Porta, J.M.: CuikSLAM: A Kinematic-based approach to SLAM. In: IEEE International Conference on Robotics and Automation, pp. 2425–2431 (2005)

[14] Porta, J.M., Jaillet, L.: Exploring the energy landscapes of flexible molecular loops using higher-dimensional continuation. Journal of Computational Chemistry **34**(3), 234–244 (2013)

[15] Porta, J.M., Jaillet, L., Bohigas, O.: Randomized path planning on manifolds based on higher-dimensional continuation. The International Journal of Robotics Research **31**(2), 201–215 (2012)

[16] Porta, J.M., Ros, L., Thomas, F.: A linear relaxation technique for the position analysis of multiloop linkages. IEEE Transactions on Robotics **25**, 225–239 (2009)

[17] Porta, J.M., Ros, L., Thomas, F., Corcho, F., Cantó, J., Pérez, J.J.: Complete maps of molecular-loop conformational spaces. Journal of Computational Chemistry **28**(13), 2170–2189 (2007)

[18] Rojas, N., Thomas, F.: Closed-form solution to the position analysis of Watt–Baranov trusses using the bilateration method. Journal of Mechanisms and Robotics **3**(3), 031,001 (2011)

[19] Rosales, C., Porta, J.M., Ros, L.: Grasp optimization under specific contact constraints. IEEE Transactions on Robotics (2013). In press

[20] Rosales, C., Ros, L., Porta, J.M., Suárez, R.: Synthesizing grasp configurations with specified contact regions. The International Journal of Robotics Research **30**(4), 431–443 (2011)

[21] Saha, M.: The Motion Planning Kit. http://ai.stanford.edu/~mitul/mpk

[22] Sucan, I.A., Moll, M., Kavraki, L.E.: The Open Motion Planning Library. IEEE Robotics Automation Magazine **19**(4), 72–82 (2012)

[23] Vahrenkamp, N.: Simox. http://simox.sourceforge.net

[24] Verschelde, J.: The PHC software. http://homepages.math.uic.edu/~jan/PHCpack/phcpack.html

[25] Zlatanov, D.S.: Generalized singularity analysis of mechanisms. Ph.D. thesis, University of Toronto (1998)