Python-based Toolbox for Generating Rational Single-loop Linkages

Daniel Huczala, Martin Pfurner, and Hans-Peter Schröcker
University of Innsbruck, Unit of Geometry and Surveying, Innsbruck, Austria
Corresponding author's email: daniel.huczala@uibk.ac.at

Abstract

In this work, we present an easy-accessible Python package that simplifies the design generation of rational linkages, available as an open-source repository [1] on the Gitlab instance of the University of Innsbruck, Austria. The purpose of the toolbox is to bring the rational linkages closer to engineering applications, for example to deploy them as cheap single-purpose manipulators, when the positioning is done only with a single motor (1 degree of freedom mechanism).
The rational motion factorization method introduced by Hegedüs, Schicho, and Schröcker [2] is capable of factorizing a parametric rational curve, which represents a rigid-body motion in the Special Euclidean group SE(3). Most importantly, it can be used for the synthesis of custom mechanisms, since the method allows one to design single-loop linkages as the same authors show in [3], that is, the resulting factors can represent the joint axes of a mechanism.
While the mathematical background is described in the papers mentioned above, the implementation itself may be not straightforward for engineers since it deals with dual quaternions (also known as Study's parameters) and the related algebra, which are not commonly taught at engineering university study programs. Furthermore, if one wants to produce a mechanism from a factorization, converting factors to Plücker/screw coordinates, Denavit-Hartenberg notation, or another representation of a mechanism [4] could be tricky. And probably the crucial problem of industrial deployment of spatial linkages are their self-collisions: purely by means of the kinematics representation parameters it is not possible to analyze and evaluate the suggested design if it is collision-free under the full-cycle motion, while CAD assemblies and kinematic or dynamic analysis are not easy (often even impossible) to perform because of numerical errors and singularities that occur when closing a linkage.
The package presented provides tools that address the problems mentioned above. It uses an open-source implementation of the motion factorization method, i.e. the package BiQuaternionPy [5], which allows calculations to be performed in biquaternion algebra. The Rational Linkages package [1] can visualize the factorization results (mechanism that can be interactively moved), as seen in the figure below. Furthermore, it assesses the self-collisions of the line model, where computations extensively benefit from the rationality of the mathematical model.
The toolbox is still in development; however, a first major release can be found on PyPI repository under the name *Rational Linkages*. Many features are planned to be added, for example, the CAD export interface, direct STL model generation, or a fast collision-free design optimization methodology. All this with the aim of joining the advanced mathematical methods with practical applications via an easy-to-use Python interface.

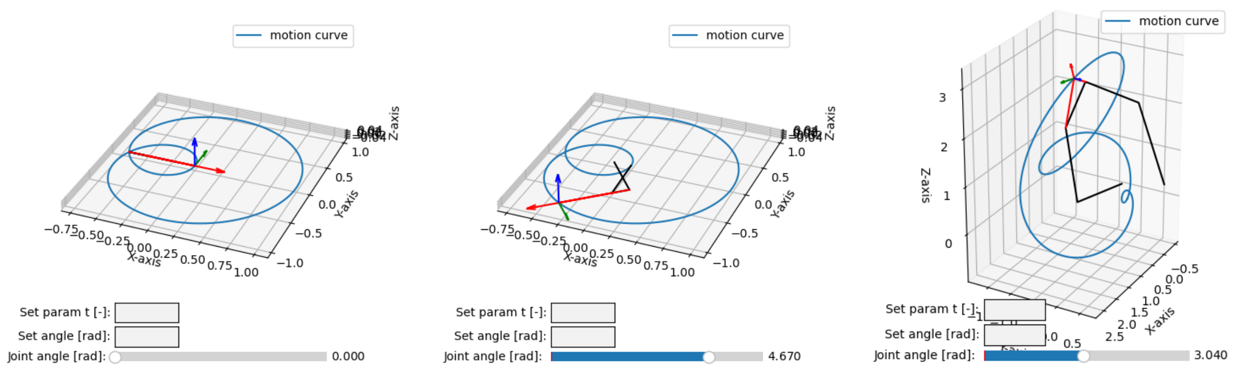

Figure: visualization of the mechanisms in presented toolbox: (left) planar 4-bar mechanism in folded configuration; (center) planar 4-bar in a random configuration; (right) spatial 6-bar in a random configuration.

Literature

[1] Huczala D.: *Rational Linkages*. Available online at [Accessed on 30th Nov 2023]: https://git.uibk.ac.at/geometrie-vermessung/rational-linkages

[2] Hegedüs, G. and Schicho, J. and Schröcker, H.-P.: Factorization of rational curves in the study quadric. *Mechanism and Machine Theory*, 69 (2013) 142-152. DOI: 10.1016/j.mechmachtheory.2013.05.010. Elsevier BV.

DOI: 10.17185/duepublico/81656

[3] Hegedüs, G. and Schicho, J. and Schröcker, H.-P.: Four-Pose Synthesis of Angle-Symmetric 6R Linkages. *Journal of Mechanisms and Robotics*, 7(4), (2015). DOI: 10.1115/1.4029186. ASME International.

[4] Huczala, D., Kot, T., Mlotek, J., Suder, J., and Pfurner, M.: An Automated Conversion Between Selected Robot Kinematic Representations. In *10th International Conference on Control, Mechatronics and Automation (ICCMA)*, Belval, Luxembourg, 2022, pp. 47-52. DOI: 10.1109/ICCMA56665.2022.10011595.

[5] Thimm D.: *biquaternion-py*. Available online at [Accessed on 30th Nov 2023]: https://pypi.org/project/biquaternion-py/

[6] Li, Z. and Nawratil, G. and Rist, F. and Hensel, M.: Invertible Paradoxic Loop Structures for Transformable Design. *Computer Graphics Forum*, 39(2) (2020) 261-275. DOI: 10.1111/cgf.13928.

Acknowledgements

Funded by
the European Union

Disclaimer
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

DOI: 10.17185/duepublico/81656
URN: urn:nbn:de:hbz:465-20240304-111836-1