



Topical Collection Reprint

Machines, Mechanisms and Robots

Theory and Applications—Volume I

Edited by
Raffaele Di Gregorio

mdpi.com/journal/machines/topical_collections



**Machines, Mechanisms and Robots:
Theory and Applications—Volume I**

Machines, Mechanisms and Robots: Theory and Applications—Volume I

Collection Editor

Raffaele Di Gregorio



Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Collection Editor

Raffaele Di Gregorio
Department of Engineering
University of Ferrara
Ferrara
Italy

Editorial Office

MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Topical Collection, published open access by the journal *Machines* (ISSN 2075-1702), freely accessible at: https://www.mdpi.com/journal/machines/topical_collections/machines_mechanisms_robots.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. <i>Journal Name</i> Year , <i>Volume Number</i> , Page Range.
--

ISBN 978-3-7258-2965-1 (Hbk)

ISBN 978-3-7258-2966-8 (PDF)

<https://doi.org/10.3390/books978-3-7258-2966-8>

© 2025 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

Contents

About the Editor	vii
Preface	ix
Raffaele Di Gregorio Theory and Applications on Machines, Mechanisms and Robots, and the Figure of Ettore Pennestri Reprinted from: <i>Machines</i> 2024 , <i>12</i> , 916, https://doi.org/10.3390/machines12120916	1
Matteo Autiero, Mattia Cera, Marco Cirelli, Ettore Pennestri and Pier Paolo Valentini Review with Analytical-Numerical Comparison of Contact Force Models for Slotted Joints in Machines Reprinted from: <i>Machines</i> 2022 , <i>10</i> , 966, https://doi.org/10.3390/machines10110966	4
Adrian Peidro and Edward J. Haug Obstacle Avoidance in Operational Configuration Space Kinematic Control of Redundant Serial Manipulators Reprinted from: <i>Machines</i> 2024 , <i>12</i> , 10, https://doi.org/10.3390/machines12010010	37
Gang Chen, Wenqian Xu, Zixing Li, Yuqiang Liu and Xin Liu Research on the Multi-Robot Cooperative Pursuit Strategy Based on the Zero-Sum Game and Surrounding Points Adjustment Reprinted from: <i>Machines</i> 2021 , <i>9</i> , 187, https://doi.org/10.3390/machines9090187	59
Raffaele Di Gregorio and Tommaso Cinti Geometric Constraint Programming (GCP) Implemented Through GeoGebra to Study/Design Planar Linkages Reprinted from: <i>Machines</i> 2024 , <i>12</i> , 825, https://doi.org/10.3390/machines12110825	77
Xianwen Kong and Jieyu Wang Reconfiguration Analysis and Characteristics of a Novel 8-Link Variable-DOF Planar Mechanism with Five Motion Modes Reprinted from: <i>Machines</i> 2023 , <i>11</i> , 529, https://doi.org/10.3390/machines11050529	117
Li Wu, Xuan Liu, Mingjun Wang, Dengbiao Zhao and Lewei Tang Kinematic Analysis of a Spatial Cable-Driven Mechanism and Its Equivalent Hybrid Mechanism for Elliptical Trajectory Reprinted from: <i>Machines</i> 2023 , <i>11</i> , 710, https://doi.org/10.3390/machines11070710	133
Henrique Simas, Raffaele Di Gregorio, Roberto Simoni and Marco Gatti Parallel Pointing Systems Suitable for Robotic Total Stations: Selection, Dimensional Synthesis, and Accuracy Analysis Reprinted from: <i>Machines</i> 2024 , <i>12</i> , 54, https://doi.org/10.3390/machines12010054	147
Daming Nie, Ruilong Du, Jiangren Tian, Pu Zhang, Fangyan Shen, Jason Gu and Yili Fu Buffering Performance Analysis of an Ostrich-like Leg Based on a Seven-Link Parallel Mechanism Reprinted from: <i>Machines</i> 2022 , <i>10</i> , 306, https://doi.org/10.3390/machines10050306	168
John M. Huss, Malachi Lehman and Arthur G. Erdman Patient-Specific Stent Fabrication Using a Seven-Degree-of-Freedom Additive Manufacturing System Reprinted from: <i>Machines</i> 2022 , <i>10</i> , 1144, https://doi.org/10.3390/machines10121144	181

**Xun Wei Chia, Poh Kiat Ng, Robert Jeyakumar Nathan, Jian Ai Yeow, Way Soong Lim
and Yu Jin Ng**
The Development of an Automated Multi-Spit Lamb Rotisserie Machine for Improved
Productivity
Reprinted from: *Machines* **2021**, *9*, 165, <https://doi.org/10.3390/machines9080165> **200**

Author Index **227**

About the Editor

Raffaele Di Gregorio

Raffaele Di Gregorio is currently a full professor of machine mechanics at the Engineering Dept. of Ferrara University, Italy. He received an M.Sc. degree in nuclear engineering, an M.Sc. degree in mechanical engineering, and an M.Sc. degree in automotive engineering from the Polytechnic University of Turin in 1982, 1985, and 1988, respectively, and a Ph.D. degree in applied mechanics from the University of Bologna in 1992. In 1983, he spent one year at the FIAT Research Center, Orbassano (IT), with an ATA grant. From 1984 to 1992, he was an officer in the Italian Army's Technical Corp at the Military School of Turin and at STAVECO of Bologna. In 1993, he taught at ITIS O. Belluzzi of Bologna. In 1994, he joined the Engineering Dept. of Ferrara University.

His research interests include kinematics and dynamics of mechanisms and machines, biomechanics, robotics, vibration mechanics, and vehicle mechanics.

He has been an "observer/member" of IFToMM TC "Computational Kinematics" since 2007 and a "member" of IFToMM PC "Standardization of Terminology" since 2009. He is an ASME member and was a "general member" of the ASME Mechanisms and Robotics Committee from 2007 to 2012; since then, he has continued to serve this Committee as a past member.

He is the author/co-author of more than 90 technical papers published in refereed international journals, has published as many papers in conference proceedings, and actively collaborates with many international journals of the field.

Preface

“Machines, Mechanisms and Robots: Theory and Applications (MMR-T&A)” is an interdisciplinary collection, published in MDPI’s journal *Machines*, that is focused on Mechanical Systems analyzed from both the theoretical and the applicative points of view. It aims to report scientific advancements in all fields concerning mechanical system design. This reprint collects the first ten papers published in the MMR-T&A collection and is the first volume of the MMR-T&A series, which will release one volume for every ten papers published in the MMR-T&A collection. The ten papers collected in this volume include one review on contact models, two papers addressing obstacle avoidance, one paper on geometric constraint programming (GCP), three papers presenting novel mechanisms, one paper on mechanism performances, and two papers addressing manufacturing issues.

This volume is dedicated to the memory of Prof. Ettore Pennestrì (Reggio Calabria, Italy, May 16, 1957 – Rome, Italy, August 29, 2024).

Raffaele Di Gregorio

Collection Editor

Editorial

Theory and Applications on Machines, Mechanisms and Robots, and the Figure of Ettore Pennestrì

Raffaele Di Gregorio

Laboratory of Mechatronics and Virtual Prototyping (LaMaViP), Department of Engineering, University of Ferrara, 44122 Ferrara, Italy; raffaele.digregorio@unife.it

This editorial presents a book that marks the beginning of the MMR-T&A series, which compiles articles from the collection *Machines, Mechanisms, and Robots: Theory and Applications (MMR-T&A)*, published in MDPI's journal *Machines*. The first volume of this series includes the first ten published papers from the collection. It starts with one review [1] on contact models. Then, it continues with two papers [2,3] addressing obstacle avoidance in different contexts, one more paper [4] on geometric constraint programming, and three papers [5–7] presenting novel mechanisms. Finally, it closes with one paper [8] on mechanism performances and two papers [9,10] addressing manufacturing issues. Going forward, a new volume in the MMR-T&A series will be released for every ten papers published.

The book is dedicated to the memory of Ettore Pennestrì (Reggio Calabria, Italy, 16 May 1957–Rome, Italy, 29 August 2024), who co-authored one [1] of the papers included in this collection. Those who knew him knew how important sharing his knowledge and his findings was to him, especially with students and young researchers (Figure 1 shows him during a lecture).



Figure 1. A picture of Ettore Pennestrì during a lecture.

Citation: Di Gregorio, R. Theory and Applications on Machines, Mechanisms and Robots, and the Figure of Ettore Pennestrì. *Machines* **2024**, *12*, 916. <https://doi.org/10.3390/machines12120916>

Received: 10 December 2024

Accepted: 12 December 2024

Published: 13 December 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Ettore Pennestrì earned his Laurea in Mechanical Engineering from the University La Sapienza of Rome in 1980. He continued his studies at Columbia University, earning a Master's degree (1987) and a Ph.D. (1991) under the guidance of Ferdinand Freudenstein. Upon returning to Italy, he began his academic career at the University of Calabria and joined the University of Rome Tor Vergata faculty in 1991, where he initially served as an Associate Professor and then, from 2001, served as a Full Professor of Applied Mechanics.

Throughout his career, he focused on research into kinematics, dynamics of mechanisms, biomechanics, gear transmission design, and vibration damping. He authored

over 100 scientific papers published in international journals and several textbooks used in university education.

His contributions span theoretical research, practical applications, and advanced numerical methodologies in applied mechanics. In computational kinematics, he developed algorithms using dual numbers and dual quaternions for the analysis of rigid body motion that are widely applied in robotics and biomechanics for motion optimization. In multibody dynamics, he authored/co-authored studies on the simulation and analysis of mechanical systems with multiple interconnected bodies, such as gears and linkages, which include modeling dynamic systems for applications in vehicles and robotics. In gear train efficiency, he published foundational research on the mechanical efficiency of gear trains, including epicyclic and hybrid systems that include closed-form formulas for analyzing power flow and losses in complex gear mechanisms. In biomechanics, he pioneered the development of musculoskeletal models for biomechanical analysis, particularly focusing on the upper limb, which contributed to ergonomics, rehabilitation engineering, and sports science. In vehicle powertrains and vibrations, he investigated vibration isolation techniques and efficiency in vehicle powertrain systems, emphasizing hybrid vehicle applications.

In addition to his scientific work, he held important roles in the editorial boards of international journals; for example, he was an Associate Editor of the *ASME Journal of Mechanical Design*; an honorary member of the Editorial Board of *Mechanism and Machine Theory*; and a member of the Advisory Board of the *Journal of Multibody System Dynamics*.

Ettore Pennestri held numerous prestigious positions and participated actively in international initiatives, such as the following:

- He was a member of the scientific committee for the ECCOMAS Thematic Conferences on Multibody Dynamics in several editions: 2005 (Madrid), 2007 (Milan), 2009 (Warsaw), 2015 (Barcelona), 2017 (Prague), and 2019 (Duisburg). He also contributed to the scientific committees of international workshops in Romania and Italy.
- He was a representative of the National Research Council (CNR) at the International Union of Theoretical and Applied Mechanics (IUTAM).
- He was an elected coordinator of the Italian study group on kinematics and dynamics of multibody systems within AIMETA, the Italian Association of Theoretical and Applied Mechanics.

His passion for teaching led him to publish numerous influential books on kinematics, dynamics, and multibody system simulation methods. His last book, *Design of Approximate Curve Tracing Linkages: The Chebyshev Min-Max Optimality Criterion* (with V.R. Shanmukhasundaram and Marco Cirelli), was published a month before his passing away, and an e-book version was offered as a free download on his ResearchGate profile (<https://www.researchgate.net/profile/Ettore-Pennestri>, accessed on 9 December 2024), which is an example of his strong commitment towards the spread of knowledge.

Ettore Pennestri was appreciated not only for his scientific contributions but also for his ability to support students and colleagues in their academic and professional journeys. As a scientist who loved science and books, his personal library contained over six hundred volumes on kinematics and applied mechanics.

He was a scientist who went beyond science. He never read books simply to understand their scientific content; he was also interested in the authors' lives and the historical context in which they lived. His lectures, rich in scientific content, were also filled with references to the history of great scientists of the past.

His commitment, rigor, and generosity will continue to inspire both the scientific and academic communities. For those who had the privilege of knowing him, Ettore will remain a guide and a lighthouse, lighting the way for research and applied mechanics.

Acknowledgments: The author acknowledges Pier Paolo Valentini for his help in synthesizing the many contributions of Ettore Pennestri to mechanism and machine science.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Autiero, M.; Cera, M.; Cirelli, M.; Pennestrì, E.; Valentini, P.P. Review with Analytical-Numerical Comparison of Contact Force Models for Slotted Joints in Machines. *Machines* **2022**, *10*, 966. [CrossRef]
2. Peidro, A.; Haug, E.J. Obstacle Avoidance in Operational Configuration Space Kinematic Control of Redundant Serial Manipulators. *Machines* **2024**, *12*, 10. [CrossRef]
3. Chen, G.; Xu, W.; Li, Z.; Liu, Y.; Liu, X. Research on the Multi-Robot Cooperative Pursuit Strategy Based on the Zero-Sum Game and Surrounding Points Adjustment. *Machines* **2021**, *9*, 187. [CrossRef]
4. Di Gregorio, R.; Cinti, T. Geometric Constraint Programming (GCP) Implemented Through GeoGebra to Study/Design Planar Linkages. *Machines* **2024**, *12*, 825. [CrossRef]
5. Kong, X.; Wang, J. Reconfiguration Analysis and Characteristics of a Novel 8-Link Variable-DOF Planar Mechanism with Five Motion Modes. *Machines* **2023**, *11*, 529. [CrossRef]
6. Wu, L.; Liu, X.; Wang, M.; Zhao, D.; Tang, L. Kinematic Analysis of a Spatial Cable-Driven Mechanism and Its Equivalent Hybrid Mechanism for Elliptical Trajectory. *Machines* **2023**, *11*, 710. [CrossRef]
7. Simas, H.; Di Gregorio, R.; Simoni, R.; Gatti, M. Parallel Pointing Systems Suitable for Robotic Total Stations: Selection, Dimensional Synthesis, and Accuracy Analysis. *Machines* **2024**, *12*, 54. [CrossRef]
8. Nie, D.; Du, R.; Tian, J.; Zhang, P.; Shen, F.; Gu, J.; Fu, Y. Buffering Performance Analysis of an Ostrich-like Leg Based on a Seven-Link Parallel Mechanism. *Machines* **2022**, *10*, 306. [CrossRef]
9. Huss, J.M.; Lehman, M.; Erdman, A.G. Patient-Specific Stent Fabrication Using a Seven-Degree-of-Freedom Additive Manufacturing System. *Machines* **2022**, *10*, 1144. [CrossRef]
10. Chia, X.W.; Ng, P.K.; Nathan, R.J.; Yeow, J.A.; Lim, W.S.; Ng, Y.J. The Development of an Automated Multi-Spit Lamb Rotisserie Machine for Improved Productivity. *Machines* **2021**, *9*, 165. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Review

Review with Analytical-Numerical Comparison of Contact Force Models for Slotted Joints in Machines

Matteo Autiero ^{1,*}, Mattia Cera ^{1,†}, Marco Cirelli ^{2,‡}, Ettore Pennestri ^{1,‡} and Pier Paolo Valentini ^{1,‡}¹ Department of Enterprise Engineering, University of Rome Tor Vergata, 1 00133 Roma, Italy² Department of Mechanical Engineering, University Niccolò Cusano, 1 00133 Roma, Italy

* Correspondence: matteo.autiero@uniroma2.it

† Current address: Dipartimento di Ingegneria dell'Impresa, via del Politecnico, 1 00133 Rome, Italy.

‡ These authors contributed equally to this work.

Abstract: The pin-in-the-slot joint is a common element in machines, and the dynamics of joints with clearances is an actively investigated topic. Important applications of such a joint can be found in Geneva mechanisms, robotized gear selectors, centrifugal vibration absorbers (CPVA) and other important mechanical devices. The paper will review the main analytical steps required to obtain the equations characterizing the different force contact models. Furthermore, a numerical test bench where such models are introduced for modeling the clearances between the pin and slot is proposed. In this regard, the present study will offer a comparison and discussion of the numerical results obtained with the different force contact models herein reviewed.

Keywords: slotted joints; contact dynamics; CPVA; curvature analysis

Citation: Autiero, M.; Cera, M.; Cirelli, M.; Pennestri, E.; Valentini, P.P. Review with Analytical-Numerical Comparison of Contact Force Models for Slotted Joints in Machines. *Machines* **2022**, *10*, 966. <https://doi.org/10.3390/machines10110966>

Academic Editor: Raffaele Di Gregorio

Received: 3 October 2022

Accepted: 19 October 2022

Published: 22 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reuleaux distinguished between lower and higher kinematic pairs. The first category includes pairs with surface contact between kinematic elements. The second category contains the remaining ones, i.e., all those with line or point contacts. The cited classification is based on an ideal geometry of kinematic elements. In fact, no clearance was assumed between the contacting surfaces of kinematic elements. This assumption is mainly adopted for rigid body kinematic and dynamic analyses. However, the presence of clearances cannot be avoided in actual manufacturing and is a cause of impact forces. There is a broad class of machines, such as robotized gear selectors [1], vibrating conveyers, vibratory diggers, centrifugal vibration absorbers (CPVA) [2–6], etc., where a reliable analysis requires an approach to kinematic pairs modeling consistent with the presence of clearances and elastic couplings between bodies. The clearance allows a tiny vibration displacement governed by the geometry and compliance of both kinematic elements' surface boundaries. The forces between the colliding bodies are characterized by high values acting for a short time interval, much less than the system-free vibration natural period of oscillation. The impact pulsating forces, triggered during the indentation of surfaces, may generate phenomena having a negative effect on system operation. These negative effects are amplified by the increase in operation speed. In the early sixties of the nineteenth century, the modeling of elastic couplings between machine links was pioneered by Kobrinskiy and Babitsky [7]. They recognized that clearances play a fundamental role in machine dynamics and introduced a joint with a one-dimensional clearances model based on the impact of pendula masses.

Many investigations (e.g., [8–11]) on impact loads caused by the presence of joint clearances are on record. Due to their high number and the differences in the theoretical approaches, this review does not have the ambition of being exhaustive.

Strictly related to the problem of impact dynamics is the topic of contact force models (e.g., [10,12–14]), impact with friction (e.g., [15]) and elastodynamic contact (e.g., [16]).

Thoughtful reviews on the contact force models are already available (e.g., Gilardi and Sharf [17], Schwab et al. [18], Haddadi and Hashtrudi-Zaad [19], Zhang and Sharf [13,20], Pereira et al. [21], Machado et al. [22]), as well as dedicated important monographies (e.g., Goldsmith [23], Johnson [24], Flores and Lankarani [25]). In particular, Chapter 3 of this last reference offers an interesting quantitative comparison of different viscous contact force models. Flores and Lankarani [25] addressed the following critics of the linear Kelvin–Voigt contact force model:

- the damping component force has a discontinuity at the beginning of the contact;
- although at the end of restitution phase there is a null indentation, the contact force is negative due to a nonzero relative contact velocity;
- the damping coefficient is constant for the entire impact time interval.

One of the distinctive features of the present review work is the summary and discussion of the main physical conditions imposed and an outline of the analytical steps that establish the contact force models. In other words, each contact force equation is reported after an outline of its theoretical bases. In an effort to maintain the original nomenclature of the investigations reviewed, it is possible that there is an overlap of meaning for different notations. The authors apologize to the readers for any inconvenience. However, the Nomenclature section should solve any ambiguity. Moreover, the reader could skip the analytical details and directly use the model equations herein marked.

The purposes of this paper are:

1. To propose a novel polynomial fitting of implicit elastic contact force models.
2. To offer a summary of the analytical derivations leading to some viscous force contact models available in the literature.
3. To investigate the difference in the different elastic contact force models in a multibody dynamics simulation.

The paper is organized as follows:

- In Section 2, the cylindrical elastic models have been summarized in their original formulation. Then, polynomial fits that explicitly link force and elastic indentation have been summarized in tables for different compliances.
- In Section 3, for different viscous analytical models, the main analytical steps that brought to their deduction have been reported.
- In Section 4, the multibody dynamics simulations of a scotch–yoke linkage with a curved pin in the slot have been discussed. In particular, for each simulation, a different elastic contact model was assumed and tested.
- Finally, Section 5 contains the conclusions.

2. Cylindrical Elastic Contact Models

An extended review of elastic contact force models has been presented by Skrinjar et al. [26] and Lankarani and Flores.

This section is focused on elastic contact force models that establish a polynomial relationship between the normal force and elastic indentation of cylindrical surfaces along a line contact. Table 1 lists some classic formulas as originally reported. A main drawback of such formulas is the often implicit relationship between force and indentation.

Thus, the contact stiffness parameter K is not immediately available, and simulation times increase. To avoid such inconveniences, in this paper, the various force-indentation relationships have been expressed as fitted polynomial equations, and the corresponding stiffness K is numerically reported.

In computer simulation, the value of δ is available and related to the amount of interference between cylindrical shapes. Conversely, the value of F_n corresponding to a prescribed δ must be computed. The use of implicit relationships between δ and F_n , such as those listed in Table 1, requires an iterative computational scheme and increases computing

times. To speed-up the computation, a simplified polynomial version of each model has been deduced. In particular, the Hertz-type elastic contact force model is assumed

$$F_n = K\delta^n \tag{1}$$

the values of K and n have been computed fitting the values of F_n and δ obtained from the implicit relationship reported in Tables 2–9 for different values of ΔR and materials (Steel and Aluminum).

Our numerical tests show that the time to evaluate the polynomial is two orders of magnitude less than the one required for the iterative solution.

Koshy et al. [27] studied, as well as experimental tests, the influence of the use of different contact force models with dissipative damping on the dynamic response of a slider-crank with dry revolute clearance joints.

A general theory for the computation of tangential and torsional compliance during the contact of two isotropic bodies has been proposed by Mindlin [28].

Hertz’s theory of impact between bodies of circular shape (e.g., [29]) considers the indentation governed by the following differential equation:

$$m_r\ddot{\delta} + F_n = 0 \tag{2}$$

where $m_r = \frac{M_1+M_2}{M_1M_2}$ and

$$F_n = K\delta^{\frac{3}{2}} \tag{3}$$

is the compliance force.

Equation (2) can be integrated into the form

$$\frac{1}{2}(\dot{\delta}^2 - v_i^2) = -\frac{2}{5}m_rK\delta^{\frac{5}{2}} \tag{4}$$

where v_i is the value of $\dot{\delta}$ at the beginning of impact. A numerical solution of (4) was discussed by Deresiewicz [30].

Table 1. Cylindrical contact-force models.

Contact Force Model	Formula	Notes
Radzimovsky [31]	$\delta = \frac{W}{\pi E^*} \left[\frac{2}{3} + \ln\left(\frac{4R_i}{b}\right) + \ln\left(\frac{4R_j}{b}\right) \right]$	
Goldsmith [23]	$\delta = W(h_i + h_j) \left[\ln\left(\frac{L^m}{F_n R(h_i + h_j)}\right) + 1 \right]$	$m = 1$
Dubowsky-Freudenstein [32–34]	$\delta = \pi \left(\frac{h_i + h_j}{2a} \right) \left[\ln\left(\frac{(R_i - R_j) 8a^3}{F_n R_i R_j \pi (h_i + h_j)}\right) + 1 \right] F_n$	
Lankarani-Nikravesh [35]	$\delta = \left(\frac{3F_n}{4E^* R^{0.5}} \right)^{\frac{1}{n}}$	
ESDU-78035 [36]	$\delta = W(h_i + h_j) \left[\ln\left(\frac{4L(R_i - R_j)}{F_n (h_i + h_j)}\right) + 1 \right]$	

Table 1. Cont.

Contact Force Model	Formula	Notes
Johnson [24]	$\delta = \frac{W}{\pi E^*} \left[\ln \left(\frac{4\pi E^* \Delta R}{W} \right) - 1 \right]$	
Pereira et al. [37,38]	$F_n = \frac{(a\Delta R + b)LE^*}{\Delta R} \delta^n$	

Pereira et al. [37] (internal contact)

$$a = 0.49 \quad b = 0.10 \quad n = Y\Delta R^{-0.005}$$

with

$$Y = \begin{cases} 1.56[\ln(1000\Delta R)]^{-0.192} & \text{if } \Delta R \in [0.005, 0.750] \text{ mm} \\ 0.0028\Delta R + 1.1083 & \text{if } \Delta R \in [0.750, 10.0] \text{ mm} \end{cases}$$

Pereira et al. [38]

$$a = \begin{cases} 0.965 & \text{for internal contact} \\ 0.39 & \text{for external contact} \end{cases} \quad b = \begin{cases} 0.0965 & \text{for internal contact} \\ 0.85 & \text{for external contact} \end{cases}$$

$$n = \begin{cases} Y\Delta R^{-0.005} & \text{for internal contact} \\ 1.094 & \text{for external contact} \end{cases}$$

with

$$Y = \begin{cases} 1.51[\ln(1000\Delta R)]^{-0.151} & \text{if } \Delta R \in [0.005, 0.34954] \text{ mm} \\ 0.0151\Delta R + 1.151 & \text{if } \Delta R \in [0.34954, 10.0] \text{ mm} \end{cases}$$

Table 2. Polynomial version of Radzimovsky's [31] contact force model (Steel).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$1.04 \cdot 10^5$	1.162	6.1
5.00	$6.24 \cdot 10^4$	1.118	4.1
10.0	$5.55 \cdot 10^4$	1.109	3.8
30.0	$4.72 \cdot 10^4$	1.097	3.2
60.0	$4.32 \cdot 10^4$	1.091	3.0
80.0	$4.17 \cdot 10^4$	1.089	2.9

Table 3. Polynomial version of Radzimovsky's [31] contact force model (Aluminum).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$41.308 \cdot 10^3$	1.195	14.6
5.00	$22.979 \cdot 10^3$	1.135	9.0
10.0	$20.228 \cdot 10^3$	1.123	8.0
30.0	$16.993 \cdot 10^3$	1.108	6.7
60.0	$15.434 \cdot 10^3$	1.100	6.1
80.0	$14.867 \cdot 10^3$	1.098	5.8

Table 4. Polynomial version of Johnson's [24] force model (Steel).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$1.42 \cdot 10^5$	1.192	7.4
5.00	$7.53 \cdot 10^4$	1.133	4.8
10.0	$6.54 \cdot 10^4$	1.122	4.3
30.0	$5.44 \cdot 10^4$	1.107	3.7
60.0	$4.91 \cdot 10^4$	1.100	3.4
80.0	$4.72 \cdot 10^4$	1.097	3.2

Table 5. Polynomial version of Johnson's [24] force model (Aluminum).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$60.628 \cdot 10^3$	1.242	18.9
5.00	$28.324 \cdot 10^3$	1.155	10.9
10.0	$24.286 \cdot 10^3$	1.140	9.5
30.0	$19.782 \cdot 10^3$	1.121	7.8
60.0	$17.703 \cdot 10^3$	1.111	7.0
80.0	$16.962 \cdot 10^3$	1.083	6.7

Table 6. Polynomial version of Goldsmith's [23] contact force model (Steel).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$2.86 \cdot 10^4$	1.066	2.0
5.00	$2.39 \cdot 10^4$	1.057	1.6
10.0	$2.27 \cdot 10^4$	1.055	1.5
30.0	$2.09 \cdot 10^4$	1.051	1.4
60.0	$1.94 \cdot 10^4$	1.048	1.3
80.0	$1.84 \cdot 10^4$	1.046	1.2

Table 7. Polynomial version of Goldsmith's [23] contact force model (Aluminum).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$10.032 \cdot 10^3$	1.071	3.7
5.00	$8.354 \cdot 10^3$	1.061	2.9
10.0	$7.933 \cdot 10^3$	1.058	2.8
30.0	$7.273 \cdot 10^3$	1.054	2.5
60.0	$6.752 \cdot 10^3$	1.050	2.3
80.0	$6.394 \cdot 10^3$	1.048	2.1

Table 8. Polynomial version of the EDSU-78035 [36] contact force model (Steel).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$2.84 \cdot 10^3$	1.165	12.2
5.00	$1.77 \cdot 10^3$	1.119	8.0
10.0	$1.59 \cdot 10^3$	1.110	7.2
30.0	$1.36 \cdot 10^3$	1.098	6.1
60.0	$1.25 \cdot 10^3$	1.092	5.6
80.0	$1.21 \cdot 10^3$	1.090	5.4

Table 9. Polynomial version of the EDSU-78035 [36] contact force model (Aluminum).

ΔR mm	K N/mm ⁿ	n	Max Error %
0.50	$10.868 \cdot 10^3$	1.165	23.6
5.00	$6.395 \cdot 10^3$	1.119	14.4
10.0	$5.691 \cdot 10^3$	1.110	12.8
30.0	$4.849 \cdot 10^3$	1.098	10.7
60.0	$4.437 \cdot 10^3$	1.092	9.6
80.0	$4.286 \cdot 10^3$	1.090	9.3

3. Viscous Contact Models

The damping factors due to the material hysteresis have great importance in dynamic simulations.

Zhang and Sharf [13] and Flores and Lankarani [25] (See p. 44 of [25]) compiled tables where, for different models, the equations of constitutive laws and the corresponding damping factors have been summarized.

The effects of clearances on machine dynamics is a topic of great interest, and many contributions are on record. The monograph authored by Flores et al. [10,39,40] presents methodologies aimed at the simulation of multibody dynamics systems, taking into account joint clearances.

3.1. Dubowsky and Freudenstein (1971)

Dubowsky and Freudenstein [32–34] developed a systematic and unified analysis of the dynamics of general planar mechanisms with clearances. Figure 1 represents their impact-pair model governed by the following differential equations:

- Non-contact period $|X_r| \leq r$

$$\ddot{X}_r = \frac{M_1 F_2(t) - M_2 F_1(t)}{M_1 M_2} \quad (5a)$$

$$\ddot{X}_m = \frac{M_1 F_2(t) + M_2 F_1(t)}{M_1 M_2} \quad (5b)$$

- Contact period $|X_r| \geq r$

$$\ddot{X}_r = -\frac{M_1 + M_2}{M_1 M_2} g(X_r) + \frac{F_2(t)}{M_2} - \frac{F_1(t)}{M_1} \quad (6a)$$

$$\ddot{X}_m = \frac{M_2 - M_1}{M_1 M_2} g(X_r) + \frac{F_2(t)}{M_2} + \frac{F_1(t)}{M_1} \quad (6b)$$

The contact compliance force F_n was computed by means of the following Hertz contact formula:

$$F_n = \frac{4 \left(\frac{R_1 R_2}{R_1 - R_2} \right)^{\frac{1}{2}}}{3(h_1 + h_2)} \delta^{\frac{3}{2}} \quad (7)$$

where the indentation, for the internal-pin configuration, follows from

$$\delta = \pi \left(\frac{h_1 + h_2}{2a} \right) \left[\ln \left(\frac{(R_1 - R_2) 8a^3}{F_n R_1 R_2 \pi (h_1 + h_2)} \right) + 1 \right] F_n \quad (8)$$

where a is half the length of the pin.

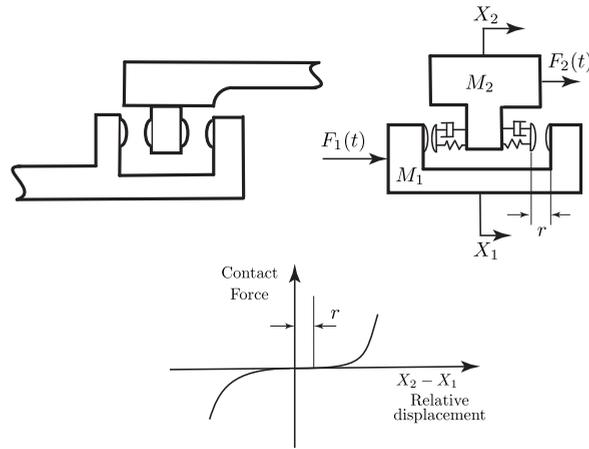


Figure 1. Dubowsky and Freudenstein elastic coupling model. F_1, F_2 : Forces; M_1, M_2 : Masses; X_1, X_2 : Displacements; r : clearance [33].

3.2. *Hunt and Crossley (1975)*

Hunt and Crossley [41] start by expressing the variation of the kinetic energy of a mass impacting against a stationary body as follows:

$$\begin{aligned} \Delta E &= \frac{1}{2} m_r (v_o^2 - v_i^2) \\ &= \frac{1}{2} m_r v_i^2 [1 - (1 - \alpha v_i^2)] \end{aligned} \tag{9}$$

where m is the mass of the moving body, and (The equation is valid for a Maxwell material and low values of v_i . See also [24], p. 368 or [23], plots on p. 259 and discussion on p. 265. The constant α is determined experimentally and has the dimensions of an inverse speed. Hunt and Crossley estimate a value within [0.08, 0.32] s/m)

$$\alpha = \frac{1 - e}{v_i} \tag{10}$$

where e is the coefficient of restitution, and v_i is the initial effective mass-relative speeds. Since $\alpha < 1$, with acceptable accuracy

$$\Delta E = \alpha m_r v_i^3 \tag{11}$$

Consequently, the force versus displacement plot must show a hysteresis loop, as shown in Figure 2.

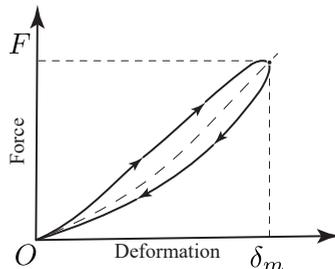


Figure 2. Hunt and Crossley: Indentation force hysteresis loop [41], where the area of such a loop represents the energy loss.

Moreover, they were well aware that the differential equation of the idealized Kelvin–Voigt model

$$m_r \ddot{\delta} + c \dot{\delta} + K \delta = 0 \tag{12}$$

could not reliably reproduce the overall nonlinear pattern of the impact. Thus, they proposed an improvement substituting (12) with

$$m_r \ddot{\delta} + (\lambda \delta^n) \dot{\delta} + K \delta^n = 0 \tag{13}$$

Moreover, they observed that for a small dissipation of energy

$$\frac{1}{2} m_r v_i^2 \approx \int_0^{\delta_m} K \delta^n d\delta \tag{14}$$

or

$$v_i = \sqrt{\frac{2K}{m_r(n+1)} \delta_m^{\frac{n+1}{2}}} \tag{15}$$

where δ_m is the maximum indentation value.

Similarly, for an intermediate position of the impact phase, one has

$$\frac{1}{2} m_r v^2 \approx \frac{1}{2} m_r v_i^2 - \int_0^\delta K \delta^n d\delta \tag{16}$$

or

$$v = \sqrt{\frac{2K}{m_r(n+1)} \sqrt{\delta_m^{n+1} - \delta^{n+1}}} \tag{17}$$

Furthermore, assuming that the linear area of the hysteresis loop is shared sufficiently equally between the inward and outward indentation phases, one has

$$E = \oint c v d\delta \approx 2 \int_0^{\delta_m} c v d\delta \tag{18}$$

or

$$\Delta E = 2 \sqrt{\frac{2K}{m_r(n+1)}} \int_0^{\delta_m} c \sqrt{\delta_m^{n+1} - \delta^{n+1}} d\delta \tag{19}$$

To integrate (19), a damping coefficient of the form

$$c = \lambda \delta^n \tag{20}$$

is assumed, and the following expression is deduced

$$\lambda = \frac{3}{2} \alpha K \tag{21}$$

Thus, one may conclude that (12), representing the free-damped half cycle of vibration of the vibroimpact, can be more realistically modified as follows

$$m_r \ddot{\delta} + K \delta^n \left(1 + \frac{3}{2} \alpha \dot{\delta} \right) = 0 \tag{22}$$

One of the merits of Hunt and Crossley is the recognition that a linear elastic spring cannot accurately represent the physics of the energy transfer process during the impact. Flores and Lankarani (see Section 3.1 of [25]) clearly show, with convincing arguments, the limits of the linear Kelvin–Voigt model.

In conclusion, the contact force equation proposed is the following

$$F_n = K\delta^{\frac{3}{2}} \left(1 + \frac{3(1-e)}{2} \frac{\delta}{\dot{\delta}^{(-)}} \right) \tag{23}$$

Moreover, Hunt and Crossley, well aware of the theoretical limitations of the coefficient of restitution concept, recommended its use in an engineering context provided verifications and extensions for new materials and impacting surfaces properties were available. It is well known that small changes in impact conditions have a strong influence on the coefficient of restitution. Several studies (e.g., Tataru and Moriwaki [42], Thornton [43], Seifried et al. [44], Minamoto and Kawamura [45]) address the theoretical and experimental coefficient of restitution evaluation for the impact of bodies of different materials.

3.3. Herbert and McWhannel (1975)

Herbert and McWhannel [46] followed an approach similar to the one of Hunt and Crossley and deduced

$$\alpha = \frac{6(1-e)}{v_i[(2e-1)^2+3]} \tag{24}$$

In particular, they proposed the following formula for the contact force:

$$F_n = K\delta^{\frac{3}{2}} \left(1 + \frac{6(1-e)}{[(2e-1)^2+3] \frac{\delta}{\dot{\delta}^{(-)}}} \right) \tag{25}$$

For the coefficient of restitution e , Herbert and McWhannel recommended the following empirical equation:

$$e = 1 - 0.026 \left(\frac{\delta}{\dot{\delta}^{(-)}} \right)^{\frac{1}{3}} \tag{26}$$

with the velocity expressed in mm/s.

3.4. Lee and Wang (1983)

Lee [47] observed that, due to their inherent kinematic limitations, Geneva mechanisms are subjected to shock loading. Moreover, pin-slot compliances, elastic deflections and imbalances interact to contribute to the dynamic load between the pin and slot. The design procedure proposed aims to minimize shock loading and contact stresses. In particular, the Hertz formula contacts a plane surface to estimate the pin-slot contact stress.

The physical model proposed by Lee and Wang [48,49] is the same as the one from Dubowsky and Freudenstein [33] (see Figure 1). For the modeling of damping, the differences regard the introduction of two new functions. The first one is apt to be evaluated by means of empirical or published experimental data. The second one is based on a heuristic choice of the damping force consistent with the boundary conditions imposed by the force-deformation hysteresis loop. The first damping function takes the form

$$D_1 = \zeta_1 T_1 \tag{27}$$

where ζ_1 and T_1 represent the damping factor and an indentation function, respectively. The simplest choice of T_1 is linear:

$$T_1 = \delta \tag{28}$$

From the fitting of experimental data, it is possible to express the coefficient of restitution e as a linear polynomial

$$e = \alpha_0 - \alpha_1 v_i \tag{29}$$

where α_0 and α_1 are the polynomial coefficients and v_i . By definition

$$e = -\frac{v_o}{v_i} \tag{30}$$

The differential equation governing the indentation during the contact phase between the body surfaces can be written in the form

$$m_r \ddot{\delta} + (\zeta_1 \dot{\delta}) \dot{\delta} + K\delta = P(t) \tag{31}$$

and its solution is obtained as polynomial approximation

$$\delta(t) \doteq v_i t - \frac{\zeta_1 v_i + K}{6m_r} v_i t^3 + \frac{(4\zeta_1 v_i + K)(\zeta_1 v_i + K)v_i}{120m_r^2} t^5 + \dots \tag{32}$$

Since the relative velocity at the end of the outward contact phase ($t = t_e$) is

$$v_o = \dot{\delta}(t_e) = v_i - \frac{\zeta_1 v_i + K}{2m_r} v_i t_e^2 + \frac{(4\zeta_1 v_i + K)(\zeta_1 v_i + K)v_i}{24m_r^2} t_e^4 + \dots \tag{33}$$

The substitution of (33) into (30) yields the coefficient of restitution

$$e = -1 + \frac{Kt_e^2}{2m_r} + \left(1 - \frac{5Kt_e^2}{12m_r}\right) \frac{\zeta_1 t_e^2}{2m_r} v_i - \frac{\zeta_1^2 t_e^4}{6m_r^2} v_i^2 + \dots \tag{34}$$

Comparing (29) with (34), we obtain

$$\alpha_0 = -1 + \frac{Kt_e^2}{2m_r} \tag{35}$$

To estimate t_e , Lee and Wang set (This choice is consistent with the usual polynomial fitting of coefficient of restitution $e = 1 - \alpha_1 v_i$.) $\alpha_0 = 1$ in (27) and obtained

$$t_e^2 = 4 \frac{m_r}{K} \tag{36}$$

thus

$$\zeta_1 = \frac{2m_r}{\left(\frac{5Kt_e^2}{12m_r} - 1\right) t_e^2} \alpha_1 \approx \frac{3}{4} \alpha_1 K \tag{37}$$

The algebraic structure of the result matches with the one deduced by Hunt and Crossley [41] by means of an energy balance. The first damping coefficient is

$$\zeta_1 \doteq \left(\frac{3}{4} \alpha_1 K\right) \dot{\delta} \tag{38}$$

In conclusion, the first contact force formula proposed by Lee and Wang is

$$f_n = K\delta \left(1 + \frac{3}{4} \alpha_1 \dot{\delta}\right) \tag{39}$$

The derivation of the second damping function follows the same guidelines as the first damping function. In particular, such a function is expressed as

$$D_2 = \zeta_2 T_2 \tag{40}$$

where ζ_2 is a damping factor, and

$$T_2 = \frac{\delta + |\delta|}{2\delta} \exp\left\{[(\delta - \varepsilon) - |\delta - \varepsilon|] \frac{q}{\varepsilon}\right\} \tag{41}$$

is the transition function, with ε being the width of the impact transition zone (see Figure 3) and q a parameter specifying the curve path within the transition zone. Typical values of q are 2, 3 and 4.

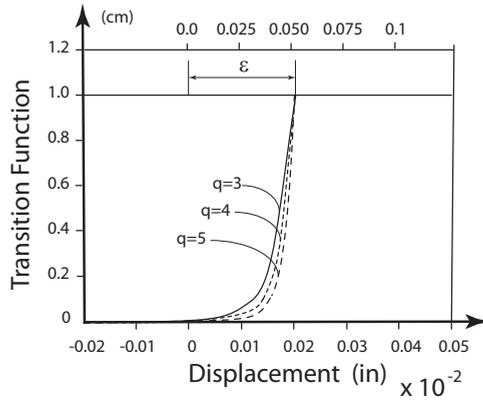


Figure 3. Transition function T_2 [48].

Parameter value ε can be arbitrarily chosen, but the conditions

$$0 \leq \varepsilon \leq \delta_m \tag{42}$$

$$\zeta_2 T_2(\delta) \delta + K_e \delta \geq 0 \tag{43}$$

that ensure the sum of damping and spring forces to be positive must be fulfilled.

With a procedure similar to the one adopted for the first damping factor [50], one obtains

$$\zeta_2 = 2m_r \omega \sqrt{\frac{(\ln e)^2}{(\ln e)^2 + \pi^2}} \tag{44}$$

where ω is the system’s natural frequency.

It has been observed that the simulations based on the second damping function are more stable and have a hysteresis loop wider than the one predicted with the first damping function.

In conclusion, the second contact force model proposed by Lee and Wang is summarized by the following formula:

$$f_n = K_e \delta + \zeta_2 T_2 \dot{\delta} \tag{45}$$

where T_2 and ζ_2 are computed from (41) and (44), respectively.

3.5. Khulief and Shabana (1985)

The approach of Khulief and Shabana [51–53] is aimed to be implemented within a multibody dynamic environment and the nomenclature is thus adapted for the purpose. Their analysis is based on the assumption that the energy dissipated during the impact is much less than the elastic strain energy involved. Therefore, the coefficient of restitution is $e \approx 1$. The bodies are assimilated to point masses or with a relative translation motion.

As shown in Figure 4, the collision process is divided into:

- the compression phase, during which the relative velocity $\dot{\delta}_{ij}$ is gradually reduced to zero and elastic energy is stored;
- the restitution phase, which begins at the release of the stored elastic energy and finishes when the bodies separate.

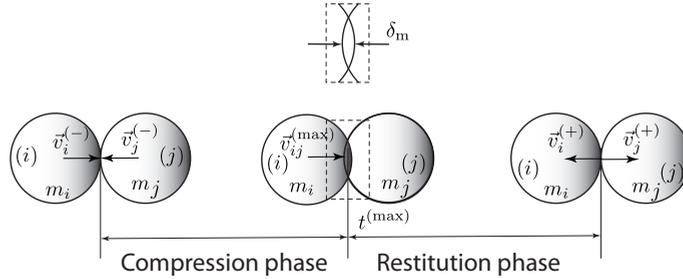


Figure 4. Central collision between two circular-shaped masses: representation of the compression and restitution phase [14,25].

The force between bodies i and j is analytically represented by the Kelvin–Voigt model

$$F_{ij} = (K\delta_{ij} + c\dot{\delta}_{ij})LG_{ij} \tag{46}$$

where

$$LG_{ij} = \begin{cases} 1 & \text{during impact} \\ 0 & \text{after separation} \end{cases} \tag{47}$$

is a logical function. The following nomenclature is introduced: $\widehat{\delta}_{ij}$ is the maximum value of indentation; \bar{u}_i and \bar{u}_j are the velocities of body i at the beginning of the compression phase and at the end of the restitution phase, respectively; u_{ij} is the body velocity when $\dot{\delta}_{ij} = 0$ (end of compression phase); $\bar{\delta}_{ij} = \bar{u}_j - \bar{u}_i$ is the relative velocity at the beginning of compression phase.

The energy conservation principle yields:

$$\frac{1}{2}m_i\bar{u}_i^2 + \frac{1}{2}m_j\bar{u}_j^2 = \frac{1}{2}(m_i + m_j)u_{ij}^2 + \underbrace{\int_0^{\widehat{\delta}_{ij}} K\delta_{ij}d\delta_{ij}}_{\text{Maximum strain energy}} \tag{48}$$

where

$$u_{ij} = \bar{u}_i + \Delta\bar{u}_i \tag{49a}$$

$$u_{ij} = \bar{u}_j + \Delta\bar{u}_j \tag{49b}$$

Using the momentum conservation principle, one has

$$m_i\Delta\bar{u}_i + m_j\Delta\bar{u}_j = 0 \tag{50}$$

or, taking into account (49),

$$u_{ij} = \left[\frac{m_j}{m_i + m_j} \right] \bar{u}_j + \frac{m_i}{m_j} \bar{u}_i \tag{51}$$

The substitution of (51) into (48), with the hypothesis of a constant contact stiffness K , gives [52]:

$$\frac{1}{2}K\delta_{ij}^2 = \frac{1}{2} \left(\frac{m_i m_j}{m_i + m_j} \right) \bar{\delta}_{ij}^2 \tag{52}$$

where

$$\bar{\delta}_{ij} = \bar{u}_i - \bar{u}_j \tag{53}$$

Equation (52) allows a relationship between the contact stiffness upper bound \bar{K}

$$\bar{K} = \frac{m_i m_j}{m_i + m_j} \left(\frac{\bar{\delta}_{ij}}{\delta_{ij}} \right)^2 \tag{54}$$

and the impacting bodies' kinematics.

To obtain an expression for the damping coefficient, the kinetic energy loss ΔE and the coefficient of restitution must be taken into account.

The combination of the following three equations:

$$\Delta E = \frac{1}{2}m_i(\bar{u}_i^2 - \tilde{u}_i^2) - \frac{1}{2}m_j(\bar{u}_j^2 + \tilde{u}_j^2) \quad \text{Energy balance} \tag{55}$$

$$e = -\frac{\tilde{u}_i - \tilde{u}_j}{\bar{u}_i - \bar{u}_j} \quad \text{Coefficient of restitution} \tag{56}$$

$$m_i(\bar{u}_i - \tilde{u}_i) + m_j(\bar{u}_j - \tilde{u}_j) = 0 \quad \text{Momentum conservation} \tag{57}$$

gives

$$\Delta E = \frac{1}{2} \left[\frac{(1-e)m_i m_j}{m_i + m_j} \right] \bar{\delta}_{ij}^2 \tag{58}$$

This energy loss is dissipated by the damping force expressed as $D\dot{\delta}_{ij}$. Therefore, it is

$$\Delta E = \oint D\dot{\delta}_{ij}d\delta_{ij} \tag{59}$$

where \oint denotes the integration around the force-displacement hysteresis loop.

$$\underbrace{\frac{1}{2}m_i\bar{u}_i^2 + \frac{1}{2}m_j\bar{u}_j^2}_{\text{Kinetic energy at start}} = \underbrace{\frac{1}{2}m_i u_i^2 + \frac{1}{2}m_j u_j^2 + \int_0^{\delta_{ij}} \bar{K}d\delta_{ij}}_{\text{Energy at intermediate time}} \tag{60}$$

with

$$u_i = \bar{u}_i + \Delta u_i \tag{61a}$$

$$u_j = \bar{u}_j + \Delta u_j \tag{61b}$$

From the previous one follows:

$$\dot{\delta}_{ij} = \bar{\delta}_{ij} + \Delta u_i - \Delta u_j \tag{62}$$

Combining the momentum conservation equation

$$m_i\Delta u_i + m_j\Delta u_j = 0 \tag{63}$$

with (60) and (61), taking into account (54), one obtains:

$$\int_0^{\delta_{ij}} \bar{K}\dot{\delta}_{ij}d\delta_{ij} = \frac{1}{2} \left(\frac{m_i m_j}{m_i + m_j} \right) \left[\bar{\delta}_{ij}^2 - \dot{\delta}_{ij}^2 \right] \tag{64}$$

or

$$\bar{K}\delta_{ij}^2 = \left(\frac{m_i m_j}{m_i + m_j} \right) \left[\bar{\delta}_{ij}^2 - \delta_{ij}^2 \right] \tag{65}$$

Consequently, from (65) follows

$$\delta_{ij} = \sqrt{\bar{\delta}_{ij}^2 - \left[\frac{\bar{K}(m_i + m_j)}{m_i m_j} \right] \delta_{ij}^2} \tag{66}$$

Finally, the substitution of (66) into (59) and the choice of a damping function of the form (The function satisfies the boundary conditions at both the contact start and separation).

$$D = \mu\delta_{ij} \tag{67}$$

yields (It is assumed the area of the hysteresis loop is equally shared between compression and restitution phase.

$$\Delta E = 2 \int_0^{\hat{\delta}_{ij}} \mu\delta_{ij} \sqrt{\bar{\delta}_{ij}^2 - \bar{\beta}^2 \delta_{ij}^2} d\delta_{ij} \tag{68}$$

After we let

$$\bar{\beta}^2 = \frac{\bar{K}(m_i + m_j)}{m_i m_j} = \left(\frac{\bar{\delta}_{ij}}{\hat{\delta}_{ij}} \right)^2 \tag{69}$$

Equation (68) can be rewritten in the form

$$\Delta E = -\frac{2}{3}\mu\bar{\beta} \left\{ \left[\left(\frac{\bar{\delta}_{ij}}{\bar{\beta}} \right)^2 - \hat{\delta}_{ij}^2 \right]^{\frac{3}{2}} - \left(\frac{\bar{\delta}_{ij}}{\bar{\beta}} \right)^3 \right\} \tag{70}$$

Equating (70) and (58), one obtains

$$\mu = \frac{3}{4} \left(\frac{\bar{K}(1 - e^2)\bar{\delta}_{ij}^2}{\left[\bar{\delta}_{ij}^2 - \left(\bar{\delta}_{ij}^2 - \bar{\beta}^2 \hat{\delta}_{ij}^2 \right)^{\frac{3}{2}} \right]} \right) \tag{71}$$

and the force-approach law, according to Khulief and Shabana [51–53], is expressed in the form

$$F_n = K\delta_{ij} + (\mu\delta_{ij})\dot{\delta}_{ij} \tag{72}$$

where μ is computed from (71).

3.6. Lankarani and Nikravesh (1988)

Lankarani and Nikravesh [8,9,35] recognized the limits and inconsistencies of the Kelvin–Voigt model and proposed a contact force expressed by the following equation

$$F_n = K\delta^n + D\dot{\delta} \tag{73}$$

for the entire period of contact. They assumed that the energy dissipated during the impact is small compared to the maximum absorbed elastic energy. Moreover, within the contact time interval, they distinguished a compression and a restitution phase. With reference to Figure 5, let $t^{(-)}$, $t^{(m)}$ and $t^{(+)}$ denote the initial time of compression, the time of maximum indentation and the final time of restitution, respectively.

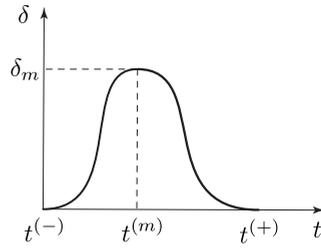


Figure 5. Indentation δ versus time for a Hertz-type contact force model [35].

Lankarani and Nikravesh start by writing the equations expressing:

- the energy loss ΔE computed as a difference between the bodies' kinetic energies at the beginning and at the end of the impact:

$$\begin{aligned} \Delta E &= T^{(-)} - T^{(+)} \\ &= \frac{m_i}{2} \left[(v_i^{(-)})^2 - (v_i^{(+)})^2 \right] + \frac{m_j}{2} \left[(v_j^{(-)})^2 - (v_j^{(+)})^2 \right] \end{aligned} \quad (74)$$

- the conservation of linear momentum:

$$m_i [v_i^{(-)} - v_i^{(+)}] + m_j [v_j^{(-)} - v_j^{(+)}] = 0 \quad (75)$$

- the coefficient of restitution:

$$e = -\frac{v_i^{(+)} - v_j^{(+)}}{v_i^{(-)} - v_j^{(-)}} \quad (76)$$

The combination of (74)–(76) gives:

$$\Delta E = \frac{1}{2} \frac{m_i m_j}{m_i + m_j} [\delta^{(-)}]^2 (1 - e^2) \quad (77)$$

where

$$\delta^{(-)} = v_i^{(-)} - v_j^{(-)} \quad (78)$$

Furthermore, considering the time interval $[t^{(-)}, t^{(m)}]$ between the beginning and end of the compression phase, one can write:

- the energy balance equation

$$\begin{aligned} T^{(-)} &= T^{(m)} + U^{(m)} \\ \frac{1}{2} m_i [v_i^{(-)}]^2 + \frac{1}{2} m_j [v_j^{(-)}]^2 &= \frac{1}{2} (m_i + m_j) [v_{ij}^{(m)}]^2 + U^{(m)} \end{aligned} \quad (79)$$

- the linear momentum conservation equation

$$m_i v_i^{(-)} + m_j v_j^{(-)} = (m_i + m_j) v_{ij}^{(m)} \quad (80)$$

The combination of (79) and (80) yields

$$U^{(m)} = \frac{1}{2} \left(\frac{m_i m_j}{m_i + m_j} \right) [\delta^{(-)}]^2 \quad (81)$$

Such energy can also be evaluated by means of the integral

$$U^{(m)} = \int_0^{\delta_m} K\delta^n d\delta = \frac{K}{n+1} \delta_m^{n+1} \tag{82}$$

The comparison of (81) and (82) gives

$$[\delta^{(-)}]^2 = 2 \frac{(m_i + m_j)K}{m_i m_j (n+1)} \delta_m^{n+1} \tag{83}$$

This relationship shows how the maximum indentation δ_m is influenced by the contact stiffness K and the difference in mass velocities at $t^{(-)}$.

Repeating the previous reasoning for a generic time interval $[t^{(-)}, t]$, with $t^{(-)} \leq t \leq t^{(m)}$, one obtains

$$\delta^2 = [\delta^{(-)}]^2 - 2 \frac{m_i + m_j}{m_i m_j} \frac{K\delta^{n+1}}{n+1} \tag{84}$$

With (84), the energy dissipated by the damping force is computed by means of the integral (It is assumed the area of the hysteresis loop is equally shared between the compression and restitution phase.):

$$\Delta T = \oint D\dot{\delta}d\delta = \oint \mu\delta^n \dot{\delta}d\delta = 2 \int_0^{\delta_m} \mu\delta^n \dot{\delta}d\delta \tag{85}$$

or, taking into account (83) and (84),

$$\Delta E = \frac{2\mu}{3K} \frac{m_i m_j}{m_i + m_j} [\delta^{(-)}]^3 \tag{86}$$

The comparison of (77) and (86) yields

$$\mu = \frac{3K(1 - e^2)}{4\delta^{(-)}} \tag{87}$$

and the contact force is finally expressed by the following formula:

$$F_n = K\delta^n \left[1 + \frac{3(1 - e^2)}{4} \frac{\delta}{\delta^{(-)}} \right] \tag{88}$$

3.7. Tsuji et al. (1992)

Tsuji et al. [54] assumed the indentation during the contact governed by the following differential equation:

$$m_r \ddot{\delta} + c\dot{\delta} + K\delta^{\frac{3}{2}} = 0 \tag{89}$$

The damping coefficient equation

$$c = \chi\delta^{\frac{1}{4}} \sqrt{m_r K} \tag{90}$$

has been found heuristically. Parameter χ is an empirical constant related to the coefficient of restitution, as shown in Figure 6.

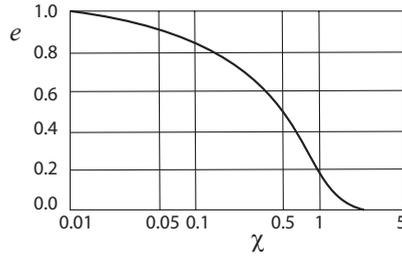


Figure 6. The relationship between χ and the coefficient of restitution e [54].

3.8. Lankarani and Nikravesh (1994)

Lankarani and Nikravesh [55] extended their analysis to the case where, after the impact between two spheres, a permanent indentation δ_p due to local plasticity is observed (see Figure 7). Such a circumstance is realistic during the collision of metallic bodies with an initial relative velocity larger than $10^{-5}V_s$, where V_s is the elastic wave propagation speed in the colliding bodies.

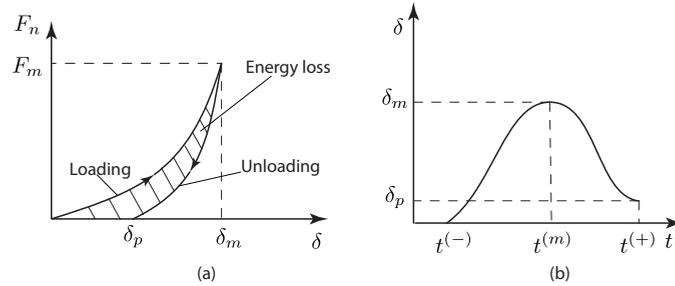


Figure 7. Contact force model with permanent indentation. (a) contact force versus time, (b) indentation versus time [55].

The contact force changes according to the following equation:

$$F_n = \begin{cases} K\delta^n & \text{compression phase} \\ F_m \left(\frac{\delta - \delta_p}{\delta_m - \delta_p} \right)^n & \text{restitution phase} \end{cases} \quad (91)$$

As it will be shown, parameters δ_m , F_m and δ_p are computed by means of (94), (95) and (97), respectively.

To determine δ_m and F_m , we use the equation of motion of the two spheres in contact

$$m_r \ddot{\delta} + K\delta^n = 0 \quad (92)$$

when integrated with the initial conditions $\delta^{(-)} = 0, \dot{\delta}^{(-)} = v_i^{(-)} - v_j^{(-)}$, yields:

$$\frac{1}{2} m_r [\dot{\delta}^2 - \dot{\delta}^{(-)2}] = K \frac{\delta^{n+1}}{n+1} \quad (93)$$

At the instant of maximum compression $\dot{\delta} = 0$ and $\delta = \delta_m$. Thus, from (93), one obtains the maximum indentation:

$$\delta_m = \left[\frac{n+1}{2K} m_r \dot{\delta}^{(-)2} \right]^{\frac{1}{n+1}} \quad (94)$$

and maximum contact force

$$F_m = K\delta_m^n \tag{95}$$

The comparison of the dissipated energy, computed by means of the integration of the contact force, is

$$\Delta T = \int_0^{\delta_m} K\delta^n d\delta + \int_{\delta_m}^{\delta_p} F_m \left[\frac{\delta - \delta_p}{\delta_m - \delta_p} \right]^n d\delta = \frac{F_m\delta_p}{n+1} \tag{96}$$

and (77) yields the permanent indentation

$$\delta_p = \frac{(n+1)m_r\dot{\delta}^{(-)2}}{2F_m} (1 - e^2) \tag{97}$$

In conclusion, the contact force is expressed by (91), where δ_m , F_m and δ_p are expressed by (94), (95) and (97), respectively.

Lankarani and Shivaswamy [56,57] conducted experiments impacting a hardened steel indenter against aluminum and steel plates. The hysteresis loop in the plot of contact-force versus indentation has been experimentally obtained and compared with simulation results.

Rhee and Akay [58] describe the motion of a four-bar rocker described by three sets of equations:

- the *sliding mode*, when the pin and journal are in contact;
- *free-flight mode*, when the pin motion is governed by its own inertia and acting forces;
- *impact mode*, when the pin and journal begin contact.

3.9. Marhefka and Orin (1999)

Marhefka and Orin [59] obtained the same result as Hunt and Crossley but by means of different analytical reasoning. They started assuming the relative motion between masses at contact governed by the differential equation

$$m_r\ddot{\delta} + \lambda\delta^n\dot{\delta} + K\delta^n = 0 \tag{98}$$

or

$$\ddot{\delta} = \Lambda\delta^n\dot{\delta} + \tilde{K}\delta^n \tag{99}$$

where $\tilde{K} = -\frac{K}{m_r}$ and $\Lambda = -\frac{\lambda}{m_r}$.

Introducing the new variable $v = \dot{\delta}$, (99) can be rewritten in the form

$$\frac{dv}{d\delta} = \frac{(\Lambda v + \tilde{K})\delta^n}{v}, \tag{100}$$

or

$$\int \frac{v dv}{(\Lambda v + \tilde{K})} = \int \delta^n d\delta \tag{101}$$

The integration of (101), with initial conditions $\delta(0) = 0$, $v(0) = v_i$, gives:

$$\Lambda v - \tilde{K} \ln|\tilde{K} + \Lambda v| = \frac{1}{n+1} \Lambda^2 \delta^{n+1} + \Lambda v_i - \tilde{K} \ln|\tilde{K} + \Lambda v_i| \tag{102}$$

or

$$\begin{aligned} \Lambda v - \tilde{K} \ln\left|1 + \frac{\Lambda}{\tilde{K}} v\right| \\ = \frac{1}{n+1} \Lambda^2 \delta^{n+1} + \Lambda v_i - \tilde{K} \ln\left|1 + \frac{\Lambda}{\tilde{K}} v_i\right| \end{aligned} \tag{103}$$

Since the values of αv_i , $\frac{\Lambda v}{\tilde{K}}$ and $\frac{\Lambda v_i}{\tilde{K}}$ are $\ll 1$, (103) can be expanded in Taylor series such that

$$\ln |1 + \epsilon| = \epsilon - \frac{\epsilon^2}{2} + \frac{\epsilon^3}{3} + \dots \tag{104}$$

For (See also Equation (10)) $\delta = 0$, $v = -ev_i = -(1 - \alpha v_i)v_i$, solving the Taylor expansion for Λ yields:

$$\Lambda \approx \frac{3\alpha\tilde{K}(2 - \alpha v_i)}{2(2 - 3\alpha v_i + 3\alpha^2 v_i^2 - \alpha^3 v_i^3)} \tag{105}$$

The result of Hunt and Crossley, as expressed in Equation (21), can also be deduced from (105) assuming negligible αv_i . After substitution of $\Lambda = \frac{3}{2}\alpha\tilde{K}$, from the integration of (103), one obtains the deformation of δ related to velocity $\dot{\delta}$ as follows

$$\delta = \left(\frac{-2m_r(n+1)}{9K\alpha^2} \right)^{\frac{1}{n+1}} \left(3\alpha(v - v_i) + 2 \ln \left| \frac{2 + 3\alpha v_i}{2 + 3\alpha v} \right| \right)^{\frac{1}{n+1}} \tag{106}$$

In conclusion, the following contact force equation was proposed:

$$F_n = K\delta^{\frac{3}{2}} \left(1 + \frac{3\alpha(2 - \alpha v_i)}{2(2 - 3\alpha v_i + 3\alpha^2 v_i^2 - \alpha^3 v_i^3)} \right) \dot{\delta} \tag{107}$$

to be simplified into

$$F_n = K\delta^{\frac{3}{2}} \left(1 + \frac{3\alpha(2 - \alpha v_i)}{2(2 - 3\alpha v_i)} \right) \dot{\delta} \tag{108}$$

neglecting the powers of αv_i equal to or greater than two.

3.10. Ghontier et al. (2004)

For Ghontier et al. [60] the differential equation governing the contact is

$$m_r \ddot{\delta} + K\delta^n (1 + \alpha \dot{\delta}) = 0 \tag{109}$$

where

$$\alpha = \frac{K}{\lambda} \tag{110}$$

To avoid tensile forces, the following inequality is required

$$1 + \alpha \dot{\delta} \geq 0 \quad \forall \dot{\delta} \tag{111}$$

After the integration of (109), one obtains

$$\int_{v_i}^{v_o} \frac{\dot{\delta}}{1 + \alpha \dot{\delta}} d\dot{\delta} + \frac{K}{m_r} \int_{\delta_i}^{\delta_o} \delta^n d\delta = 0 \tag{112}$$

where $\delta_i = \delta_o = 0$ are the initial and final penetration depths, respectively.

The algebraic development of (112), taking into account the definition of kinematic coefficient of restitution (30), yields:

$$\alpha v_i - \ln(1 + \alpha v_i) + e\alpha v_i + \ln(1 - e\alpha v_i) = 0 \tag{113}$$

After we introduce the new variable

$$d = \alpha v_i e \tag{114}$$

Equation (113) can be expressed in the following form

$$\frac{1 + \frac{d}{e}}{1 - d} = \exp \left[d \left(1 + \frac{1}{c} \right) \right] \tag{115}$$

which is more amenable for the solution with respect to d .

In conclusion, the force contact formula proposed by Ghontier et al. is

$$F_n = K\delta^n (1 + \alpha\dot{\delta}) \tag{116}$$

where, given e and v_i , the damping factor α is obtained from (114), after the solution of (115) with respect to d . In the numerical solution of (115), one should observe that:

- $d < 1$, a good initial guess is $d = 1 - e^2$;
- the solution must be consistent with (110);
- the solution d_{HC} corresponding to the model of Hunt and Crossley is $d_{HC} = \frac{3}{4}e(1 - e)$.

3.11. Flores et al. (2011–2016)

Flores et al. [14,25] closely followed the work of Lankarani and Nikravesh [35]. One of the novelties of their work is distinguishing between the energy ΔE_c , dissipated in the compression phase, and ΔE_r , dissipated in the restitution phase :

$$T^{(-)} = T^{(m)} + U^{(m)} + \Delta E_c \tag{117}$$

For this purpose, they assumed that, during masses contact, the system dynamics are governed by the differential equation

$$m_r \ddot{\delta} + c\dot{\delta} + K\delta = 0 \tag{118}$$

This, neglecting damping, allowed the expression of the velocity of deformation during the two phases, respectively, as follows:

$$\begin{aligned} \dot{\delta} &= \dot{\delta}^{(-)} \sqrt{1 - \left(\frac{\delta}{\delta_m} \right)^2} \\ \dot{\delta} &= \dot{\delta}^{(+)} \sqrt{1 - \left(\frac{\delta}{\delta_m} \right)^2} \end{aligned} \tag{119}$$

where $\delta_m = \frac{\delta^{(-)}}{\omega}$ and $\omega = \sqrt{\frac{K}{m_r}}$.

Moreover, they established the mathematical relationship

$$\left(\frac{\delta}{\delta_m} \right)^2 + \left(\frac{\dot{\delta}}{\dot{\delta}^{(-)}} \right)^2 = 1 \tag{120}$$

Since

$$\begin{aligned} \Delta E_c &= \int_0^{\delta_m} \lambda \delta^{\frac{3}{2}} \dot{\delta}^{(-)} \sqrt{1 - \left(\frac{\delta}{\delta_m} \right)^2} d\delta \\ \Delta E_r &= \int_0^{\delta_m} \lambda \delta^{\frac{3}{2}} |\dot{\delta}^{(+)}| \sqrt{1 - \left(\frac{\delta}{\delta_m} \right)^2} d\delta \end{aligned} \tag{121}$$

with λ parameter to be determined, the total dissipated energy is computed as follows:

$$\begin{aligned} \Delta E &= \Delta E_c + \Delta E_r \\ &= \chi \left(\dot{\delta}^{(-)} + |\dot{\delta}^{(+)}| \right) \int_0^{\delta_m} \delta^{\frac{3}{2}} \sqrt{1 - \left(\frac{\delta}{\delta_m} \right)^2} d\delta \end{aligned} \tag{122}$$

A numerical evaluation of the integral provides

$$\Delta E = \frac{1}{4} \lambda (1 + e) \dot{\delta}^{(-)} \delta_m^{\frac{5}{2}} \tag{123}$$

Finally, the combined use of Equations (77) and (123), energy balance

$$\frac{1}{2} m_i \left(v_i^{(-)} \right)^2 + \frac{1}{2} m_j \left(v_j^{(-)} \right)^2 = \frac{1}{2} (m_i + m_j) v_{ij}^{(m)} + \frac{2}{5} K \delta_m^{\frac{5}{2}} + \frac{1}{4} \lambda \dot{\delta}^{(-)} \delta_m^{\frac{5}{2}} \tag{124}$$

and linear momentum conservation

$$m_i v_i^{(-)} + m_j v_j^{(-)} = (m_i + m_j) v_{ij}^{(m)} \tag{125}$$

yields:

$$\lambda = \frac{8K(1 - e)}{5e\dot{\delta}^{(-)}} \tag{126}$$

and the proposed contact force formula is as follows:

$$F_n = K \delta^{\frac{3}{2}} \left(1 + \frac{8(1 - e)}{5e\dot{\delta}^{(-)}} \dot{\delta} \right) \tag{127}$$

Machado et al. embodied (127) in a general methodology for 3D-contact problems [61] and in biomechanic analyses of knee joints [62,63]

3.12. Gharib and Hurmuzlu (2012)

Gharib and Hurmuzlu [64] assumed a contact force of the form:

$$F_n = K \delta^n + \lambda \delta^n \dot{\delta} \tag{128}$$

and observed that, at the end of an elastic collision ($t = t^{(+)}$), both the contact force F_N and the indentation δ_f vanish, while the indentation velocity is

$$\dot{\delta}^{(+)} = v_i^{(+)} - v_j^{(+)} \neq 0 \tag{129}$$

Therefore, the following equation holds

$$\delta_f^n \left(K + \lambda \dot{\delta}^{(+)} \right) = 0 \tag{130}$$

the solution of which is

$$\dot{\delta}^{(+)} = -\frac{K}{\lambda} \tag{131}$$

Since by definition of coefficient of restitution

$$\dot{\delta}^{(+)} = -e \dot{\delta}^{(-)} \tag{132}$$

one has the following expression for the damping coefficient

$$\lambda = \frac{1}{e} \frac{K}{\dot{\delta}^{(-)}} \tag{133}$$

In conclusion, the first formula for the elastic contact force proposed by Gharib and Hurmuzlu is

$$F_n = K\delta^n \left(1 + \frac{\dot{\delta}}{e\dot{\delta}^{(-)}} \right) \tag{134}$$

Gharib and Hurmuzlu also analyzed the case of impact with indentation. The contact forces can be mathematically written as follows:

$$F_c = K_c\delta^n \quad \text{compression force} \tag{135}$$

$$F_r = F_m \left[\frac{\delta - \delta_p}{\delta_m - \delta_p} \right]^n \quad \text{restitution force} \tag{136}$$

The restitution force can be alternatively expressed in the form:

$$F_r = K_r(\delta - \delta_p)^n \tag{137}$$

Since

$$F_m \equiv K_c\delta_m^n = K_r(\delta_m - \delta_p)^n \tag{138}$$

or

$$\delta_p = \delta_m \left[1 - \left(\frac{K_c}{K_r} \right)^{\frac{1}{n}} \right] \tag{139}$$

Since

$$\int_0^{\delta_m} F_c d\delta = \frac{K_c\delta_m^{n+1}}{n+1} \quad \text{Work in the compression phase} \tag{140}$$

$$\int_{\delta_m}^{\delta_p} F_r d\delta = \frac{K_r(\delta_m - \delta_p)^{n+1}}{n+1} \quad \text{Work in the restitution phase} \tag{141}$$

using the definition of the energetic coefficient of restitution e_w one obtains

$$e_w^2 = - \frac{\int_0^{\delta_m} F_c d\delta}{\int_{\delta_m}^{\delta_p} F_r d\delta} = \left(\frac{K_c}{K_r} \right)^{\frac{1}{n}} \tag{142}$$

The combination of (139) and (142) gives

$$\delta_p = \delta_m \left[1 - e_w^2 \right] \tag{143}$$

Moreover, the combination of (136) and (137) yields

$$K_r = \frac{K_c}{e_w^{2n}} = \frac{F_m}{(\delta_m - \delta_p)^n} \tag{144}$$

In conclusion, for the case of impact with indentation, the second contact force formula proposed by Gharib and Hurmuzlu is

$$F_n = \begin{cases} K_c\delta^n & \text{Compression phase} \\ K_r(\delta - \delta_p)^n & \text{Restitution phase} \end{cases} \tag{145}$$

with $K_c = K, K_r, F_m$ and δ_p from (144), (139) and (138), respectively.

For the impact of a slender bar against a hard wall, experimental values of the coefficient of restitution are reported in [65].

3.13. Hu and Guo (2015)

Hu and Guo [66] estimate the energy loss by means of (79), whereas the relationship between the deformation and deformation velocity are

$$\dot{\delta} = \begin{cases} \delta^{(-)} \sqrt{1 - \left(\frac{\delta}{\delta_m}\right)^{\frac{5}{2}}} & \text{Compression phase} \\ \delta^{(+)} \sqrt{1 - \left(\frac{\delta}{\delta_m}\right)^{\frac{5}{2}}} & \text{Restitution phase} \end{cases} \quad (146)$$

deduced assuming the differential Equation (2) of the elastic Hertz contact model and symmetry of behavior between compression and restitution phases.

Flores et al. [14,25] instead deduced (119) using a linear differential equation without damping.

The energy losses for the two phases are, respectively:

$$\Delta E_c = \int_0^{\delta_m} \lambda \delta^{\frac{3}{2}} \delta^{(-)} \sqrt{1 - \left(\frac{\delta}{\delta_m}\right)^{\frac{5}{2}}} d\delta = \frac{4}{15} \lambda \delta^{(-)} \delta_m^{\frac{5}{2}} \quad (147a)$$

$$\Delta E_r = \int_0^{\delta_m} \lambda \delta^{\frac{3}{2}} |\delta^{(+)}| \sqrt{1 - \left(\frac{\delta}{\delta_m}\right)^{\frac{5}{2}}} d\delta = \frac{4}{15} \lambda |\delta^{(+)}| \delta_m^{\frac{5}{2}} \quad (147b)$$

and the overall energy loss is

$$\Delta E = \Delta E_c + \Delta E_r = \Delta E = \frac{4}{15} \lambda (1 + e) \delta^{(-)} \delta_m^{\frac{5}{2}} \quad (148)$$

To evaluate the hysteresis damping factor λ , the combination of energy balance

$$T^{(-)} = T^{(m)} + \int_0^{\delta_m} K \delta^{\frac{3}{2}} d\delta + \Delta E_c \quad (149)$$

and momentum conservation (80) yields

$$\delta_m^{\frac{5}{2}} = \frac{15m}{4(2\lambda \delta^{(-)} + 3K)} \delta^{(-)2} \quad (150)$$

or

$$\lambda = \frac{3K(1 - e)}{2e \delta^{(-)}} \quad (151)$$

In conclusion, Hu and Guo proposed the following formula for the normal contact force

$$F_n = K \delta^n \left[1 + \frac{3(1 - e)}{2e} \frac{\dot{\delta}}{\delta^{(-)}} \right] \quad (152)$$

4. Numerical Example

Some of the contact force models listed in the previous section have been tested within a multibody dynamics simulation. In particular, the scotch-yoke linkage with a circular guide, depicted in Figure 8, has been chosen as a test bench. In order to highlight the effect of different formulations for the normal contact force F_n , the only clearance introduced is the one between the pin and the circular slot. All the remaining kinematic joints are frictionless and without clearance.

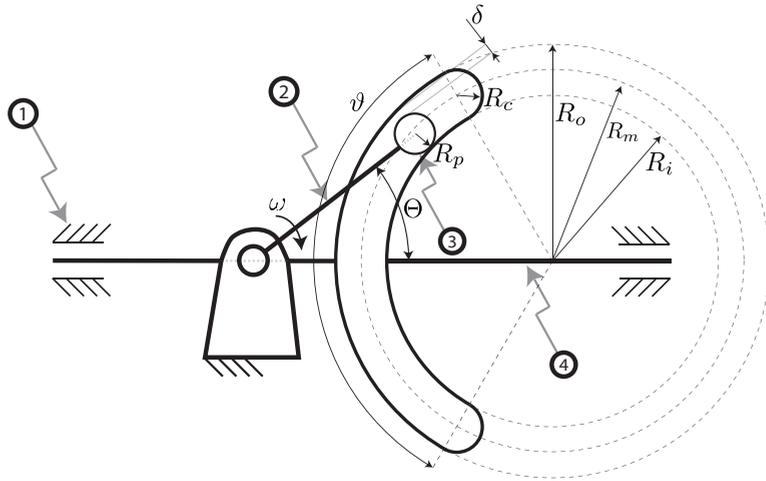


Figure 8. Scotch-yoke linkage.

The geometrical data, initial condition and inertia parameters are listed in Table 10. The crank rotates at a constant angular speed of $24\pi \frac{\text{rad}}{\text{s}}$, and the effect of gravity is omitted. All bodies are made of steel with a Young modulus of 207 GPa and a Poisson ratio of 0.3. The mechanism moves two masses m of 1 kg each fixed at both ends of the slotted slider.

As already been pointed out in the previous sections, the normal force models are mostly implicit. To reduce the computational burden of the simulation, all the formulations are represented as a polynomial function of the type $F_n = K\delta^n$.

The contact between the pin and circular slot has variable stiffness properties depending on the geometry bodies in contact. Three different regions can be observed:

- Pin to inner track: external contact between the pin with radius R_p and the inner track with radius R_i ;
- Pin to outer track: internal contact between the pin with radius R_p and the outer track with radius R_o ;
- Pin to circular track: internal contact between the pin with radius R_p and the circumferential track with radius R_c .

Table 10. Input data.

Geometry			Inertia				Initial Conditions			
			Name	Mass (kg)	Inertia (kg · mm ²)					
R_o	68	mm	1	Frame	\\	\\		Θ_{in}	31	deg
R_m	60	mm	2	Crank	0.090	30.40		ω_{in}	0	$\frac{\text{rad}}{\text{s}}$
R_i	52	mm	3	Pin	0.015	0.48		ω	$24 \cdot \pi$	$\frac{\text{rad}}{\text{s}}$
R_c	8	mm	4	Slotted slider	0.353	3495.6				
R_p	7.9	mm								
θ	120	deg								

The normal contact force is governed by the equation:

$$F_n = K\delta^n + \frac{3}{2}\alpha K|\delta|^n \delta^m \tag{153}$$

The elastic constant K and the indentation exponent n are obtained by fitting five different cylindrical contact force relationships. In particular, the models tested are: Radzi-

mowsky, Johnson, Goldsmith, EDSU-78035 and Lankarani and Nikravesh. Among those considered in this numerical example, the last one is the only model already explicit. Table 11 reports the polynomial fitting results for each model and every contact region.

On the other hand, following the Hunt and Crossley model, the viscous coefficient C is taken proportional to K by means of the coefficient α . For materials such as steel, bronze and ivory, Hunt and Crossley suggest values of alpha between 0.002 and 0.008 s/in (0.08–0.32 s/m). In this numerical example, α is set as equal to 0.32 s/m, and the penetration velocity exponent m is set to unit value.

Table 11. Fitting results.

	ΔR	Lankarani		Radzimowsky		Johnson	
	(mm)	$K [\frac{N}{mm^m}]$	n	$K [\frac{N}{mm^m}]$	n	$K [\frac{N}{mm^m}]$	n
R_i/R_p	60.1	$1.220 \cdot 10^5$	1.50	$4.321 \cdot 10^4$	1.09	$4.914 \cdot 10^4$	1.10
R_o/R_p	59.9	$1.424 \cdot 10^5$	1.50	$4.315 \cdot 10^4$	1.09	$4.907 \cdot 10^4$	1.10
R_c/R_p	0.1	$1.198 \cdot 10^6$	1.50	$1.846 \cdot 10^5$	1.22	$3.147 \cdot 10^5$	1.28

	ΔR	Goldsmith		EDSU-78035	
	(mm)	$K [\frac{N}{mm^m}]$	n	$K [\frac{N}{mm^m}]$	n
R_i/R_p	60.1	$1.717 \cdot 10^4$	1.04	$1.300 \cdot 10^4$	1.09
R_o/R_p	59.9	$1.743 \cdot 10^4$	1.04	$1.254 \cdot 10^4$	1.09
R_c/R_p	0.1	$2.221 \cdot 10^4$	1.05	$4.786 \cdot 10^4$	1.22

As it is possible to observe from the fitting results, the values of K for the inner track region (R_i to R_p) and the outer track region (R_o to R_p) are very similar, especially for Johnson and Radzimowsky formulations. In fact, these models give the same force result for internal and external contact with the same ΔR .

To clarify this statement, both the formulation of Johnson and Radzimowsky are reported below, emphasizing the ΔR dependence

$$\delta = \frac{W}{\pi E^*} \left[\ln \left(\frac{4\pi \Delta R E^*}{W} \right) - 1 \right], \quad \text{Johnson}$$

$$\delta = \frac{W}{\pi E^*} \left[\frac{2}{3} + \ln \left(\frac{8\Delta R E^*}{1.6^2 W} \right) \right], \quad \text{Radzimowsky}$$

In the slot without clearance condition, referring to Figure 9, and minding that $\Delta R = R_i \pm R_j$ (+/− : external/internal contact), one can write:

$$\Delta R_{ext} = R_p^* + R_i$$

$$\Delta R_{int} = R_o - R_p^* = R_i + 2R_p^* - R_p^* = R_i + R_p^* = \Delta R_{ext}$$

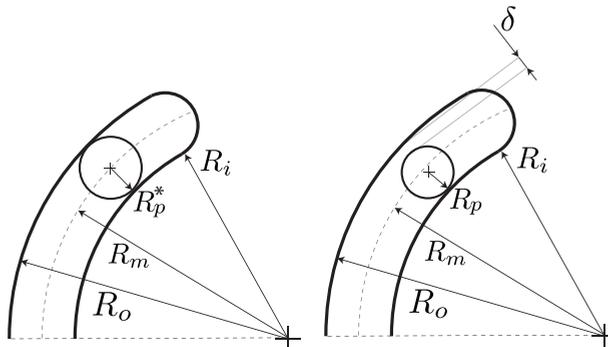


Figure 9. Scotch-yoke linkage, without (left) and with clearance (right).

This relationship, valid for pin-slot coupling without clearance, can be generalized with a clearance δ as

$$\Delta R_{ext} = \Delta R_{int} + \delta$$

This geometrical condition can be used to simplify the dynamic analysis of curved slots, calculating just one of the two values of K and using it for both the inner and outer track. As highlighted in Figure 10a for steel and Figure 10b for aluminum-like material, the error involved in such a simplification will be proportional to the clearance. The penetration reported is computed for a contact force per unit length of 10 kN/mm and variable clearance. The error does not exceed 2%. This supports the righteousness of considering only one stiffness coefficient when the clearance is quite small.

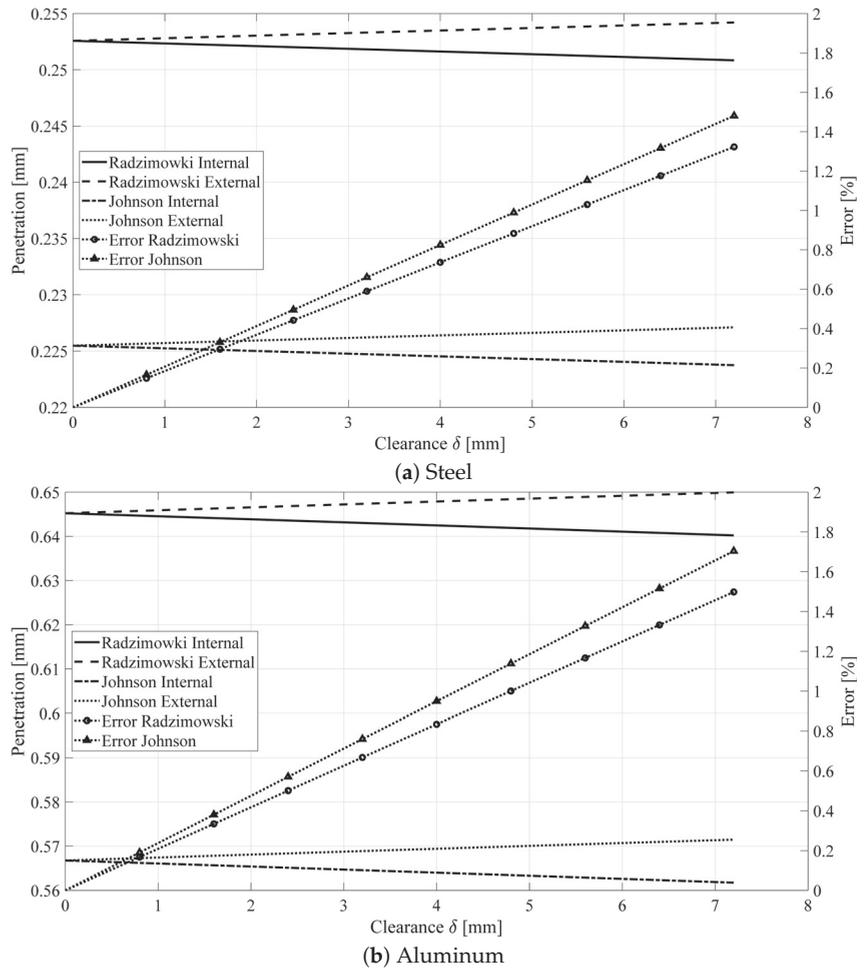


Figure 10. Effect of clearance on the error.

For a complete crank rotation, the comparison of contact forces computed according to the different models is depicted in Figure 11. To minimize the effects of the initial conditions, the second full crank revolution is monitored. A region with a null contact force is visible around 100 degrees of the relative crank angle corresponding to pin-slot separation caused by clearance.

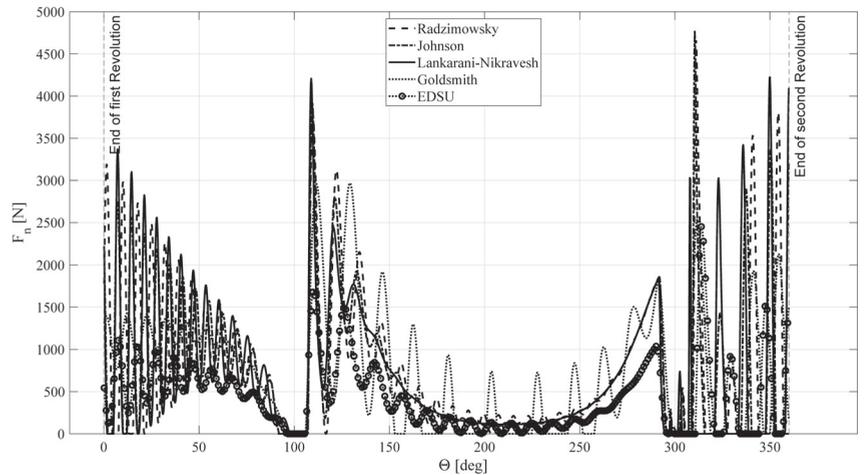


Figure 11. Contact force comparison.

When the contact is continuous (i.e., neither impact nor rebound occur), there is no relevant difference between the models. In fact, the contact force must be consistent with the dynamic equilibrium matching all the other forces acting on the system. On the other hand, the penetration, as well as its oscillation frequency, heavily depends on the contact model due to the different stiffness characteristics, as observed in Figures 12 and 13.

The detail on the impact sections, depicted in Figure 14, is useful to highlight the differences in the model's dynamic behavior. Since the exponent n values are close for all models, except for the Lankarani–Nikravesh model, one can state (for all the remaining formulations) that the higher the K value, the higher the amplitude of contact forces during the impact phases (i.e., until 100 deg of relative crank angle). Moreover, a high value of K , and, consequently, a high value of C , provides contact steadiness. In this regard, the Goldsmith model (the less stiff) is the only model that shows the detachment of the pin from the slot in the range between 120 and 300 degrees.

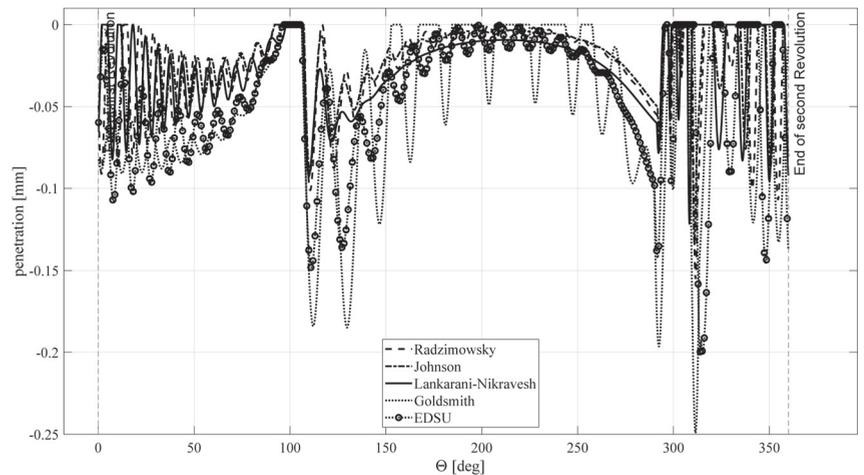


Figure 12. Penetration comparison.

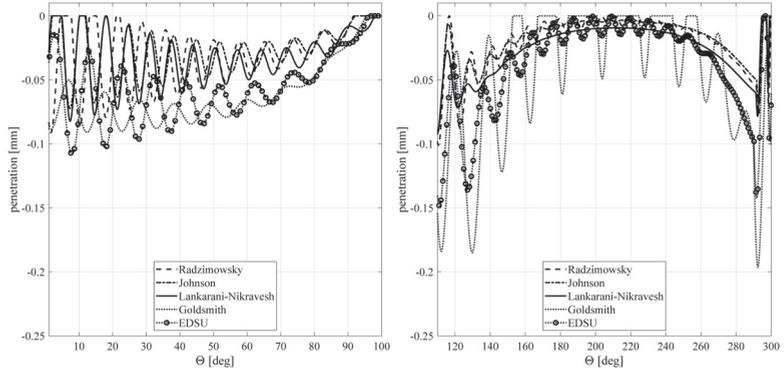


Figure 13. Detail of penetration comparison.

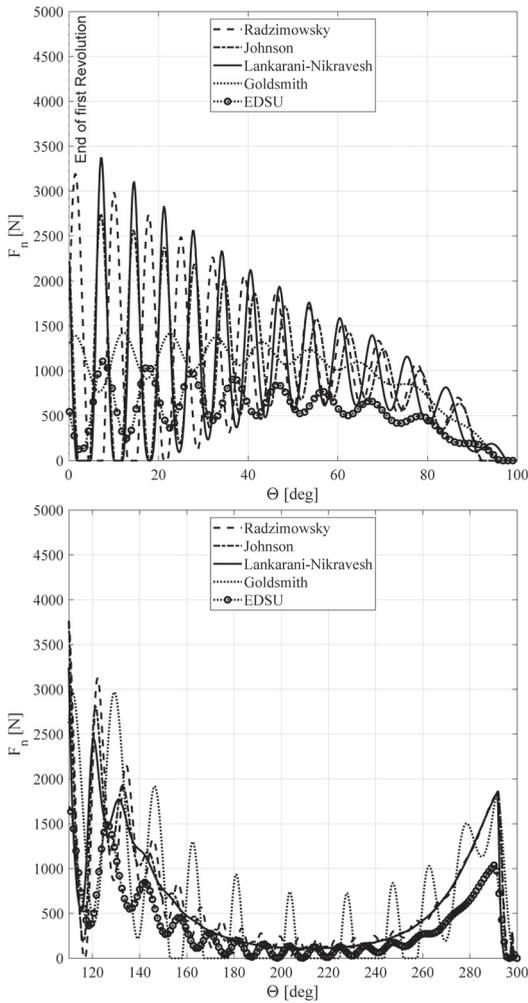


Figure 14. Detail of contact force comparison.

Figure 15 depicts the differences between the hysteresis profile obtained isolating a situation of contact and rebound within the dynamic simulation of interest.

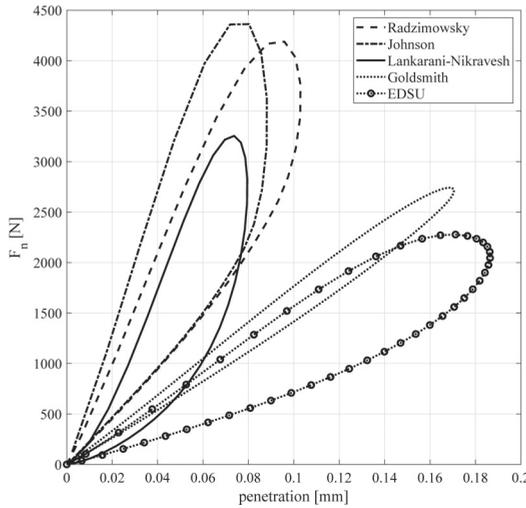


Figure 15. Hysteresis comparison.

In accordance with the previous considerations, higher forces and more steep loops are noticeable in the Johnson and Radzimowsky models. Conversely, Goldsmith and EDSU exert less abrupt profiles. The Lankarani–Nikravesh model provides intermediate behavior.

Finally, the contact formulation affects the position, velocity and acceleration of the slider. However, the small clearance between the pin and slot causes the position to be slightly influenced by the contact formulation. Conversely, in the velocity and acceleration plots versus the relative crank angle, some differences are detectable and are consistent with the effects observed in the contact force variation (See Figure 16).

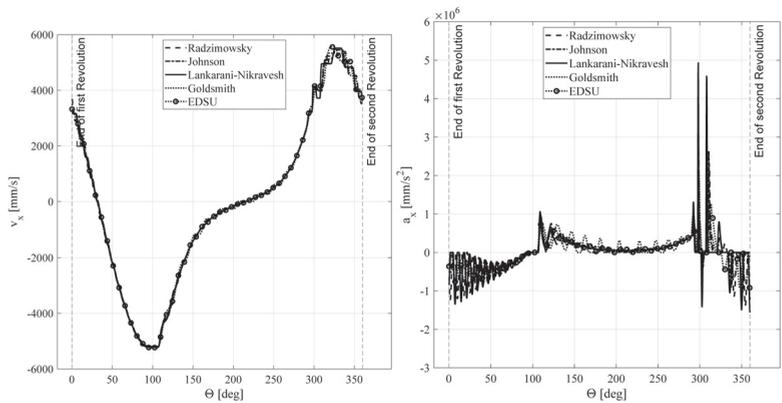


Figure 16. Slotted slider kinematics.

5. Conclusions

The availability of reliable and computationally efficient contact force models is an important requirement in multibody dynamics simulations. A review of methods on the basis of analytical developments and behavior in simulations has been presented herein. Our main focus was the dynamic analysis of mechanisms with the pin-in-the-slot kinematic pairs. The discussion herein offered gives guidelines about the distinctive computational

features of each model, but cannot offer a definitive answer on how faithfully the model reproduces reality. This would require an extensive campaign of experimental validation. As an element of novelty and to speed-up the simulation, the polynomial fitting of implicit equations has been presented in tabular form to compute static indentation with different models. Moreover, the possibility of taking into account just one track stiffness instead of differentiating between internal and external contact has been studied. We inferred that this simplification is feasible with small clearances. Lastly, choosing high-stiffness models will provide severe contact forces with high-frequency oscillation, but with a high value of exponent n the contact proved to be more stable in this particular case study.

Author Contributions: The authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

c	damping coefficient
D	damping coefficient
e	kinematic coefficient of restitution
$E^* = \frac{1}{h_i + h_j}$	composite Young modulus
E_k	Young modulus of body k ($k = i, j, 1, 2$)
F_m	maximum contact force
F_n	normal force
F_1, F_2	forces acting on the masses of the Dubowsky and Freudenstein impact pair
F_m	maximum contact force
$g(X_r)$	contact force as function of relative displacement
$h_k = \frac{1 - v_k^2}{E_k}$	($k = i, j, 1, 2$)
K	contact stiffness parameter
L	length of the contact
m	exponent of penetration velocity δ
m_r	$\frac{M_1 M_2}{M_1 + M_2}$ effective mass
n	Hertz exponent
M_k	masses ($k = 1, 2, i, j$)
R_k	cylinder radius of body k ($k = i, j$)
t	time
t_e	time at the end of outward contact phase
v_i, v_o	relative speeds before and after collision
$\Delta R = R_i \pm R_j$	clearance (+/-: External/Internal contact)
ΔE	variation of kinetic energy
$t^{(-)}, t^{(m)}$ and $t^{(+)}$	initial time of compression, the time of maximum indentation and the final time of restitution, respectively
$T^{(-)}, T^{(+)}$	system kinetic energies at times $t^{(-)}$ and $t^{(+)}$, respectively
$T^{(m)}$	kinetic energy at the end of impact compression phase
$U^{(m)}$	maximum strain energy at the end of impact compression phase
$v_k^{(-)}, v_k^{(+)}$	body k velocities at times $t^{(-)}$ and $t^{(+)}$, respectively ($k = i, j$)
$v_{ij}^{(m)}$	common velocity of the bodies at the end of the contact compression phase

v_i, v_o	relative velocities at the beginning and at the end of contact, respectively
V_s	elastic wave propagation speed in colliding solids
$W = \frac{F_n}{L}$	contact force per unit length
X_1, X_2	masses displacements of the Dubowsky and Freudenstein impact pair
α	a constant based on the slope of the (e, v_i) curve
δ	relative indentation between contacting bodies
δ_f	relative indentation at the end of contact
δ_k	indentation of sphere k , ($k = i, j$)
δ_m	maximum relative indentation value
δ_p	permanent indentation after impact
$\delta^{(-)}$	relative approach velocity (same as v_i)
$\delta^{(+)}$	relative departing velocity (same as v_o)
λ	hysteresis damping factor
ε	width of the transition zone (see Figure 3)
ν_k	Poisson ratio of body k ($k = i, j$)
ω	system natural frequency

References

- Pennestri, E.; Stefanelli, R.; Valentini, P.P.; Vita, L. Efficiency and wear in cam actuated robotized gearbox using virtual model. *Int. J. Veh. Des.* **2008**, *46*, 347. [CrossRef]
- Cera, M.; Cirelli, M.; Pennestri, E.; Valentini, P. The kinematics of curved profiles mating with a caged idle roller—higher-path curvature analysis. *Mech. Mach. Theory* **2021**, *164*, 104414. [CrossRef]
- Cera, M.; Cirelli, M.; Pennestri, E.; Valentini, P.P. Design analysis of torsichrone centrifugal pendulum vibration absorbers. *Nonlinear Dyn.* **2021**, *104*, 1023–1041. [CrossRef]
- Cirelli, M.; Cera, M.; Pennestri, E.; Valentini, P.P. Nonlinear design analysis of centrifugal pendulum vibration absorbers: An intrinsic geometry-based framework. *Nonlinear Dyn.* **2020**, *102*, 1297–1318. [CrossRef]
- Cera, M.; Cirelli, M.; Pennestri, E.; Valentini, P.P. Nonlinear dynamics of torsichrone CPVA with synchronizing form closure constraint. *Nonlinear Dyn.* **2021**, *105*, 2739–2756. [CrossRef]
- Cera, M.; Cirelli, M.; Pennestri, E.; Valentini, P. Design and comparison of centrifugal dampers modern architectures: The influence of roller kinematics on tuning conditions and absorbers nonlinear dynamics. *Mech. Mach. Theory* **2022**, *174*, 104876. [CrossRef]
- Kobrinisky, A. *Mechanisms with Elastic Couplings*, NASA Technical Translation TT F-534; Technical Report; NASA: Washington, DC, USA, 1964.
- Lankarani, H.M. Canonical Equations of Motion and Estimation of Parameters in the Analysis of Impact Problems. Ph.D. Thesis, University of Arizona, Tucson, AZ, USA, 1988.
- Lankarani, H.M.; Nikraves, P.E. A contact force model with hysteresis damping for impact analysis of multibody systems. In Proceedings of the 15th Design Automation Conference: Volume 3—Mechanical Systems Analysis, Design and Simulation. American Society of Mechanical Engineers, Montreal, QC, Canada, 17–21 September 1989; pp. 45–51. [CrossRef]
- Flores, P.; Ambrósio, J.A.C.; Pimenta Claro, J.C.; Lankarani, H.M. *Kinematics and Dynamics of Multibody Systems with Imperfect Joints: Models and Case Studies*; Springer International Publishing: Berlin/Heidelberg, Germany, 2010.
- Li, W.; Wang, C.; Sheng, C.L.; Yang, H.; Lu, E. Pin-slot clearance joints in multibody systems. *Int. J. Precis. Eng. Manuf.* **2017**, *18*, 1719–1729. [CrossRef]
- Han, I.; Gilmore, B.J. Multi-Body Impact Motion with Friction Analysis, Simulation, and Experimental Validation. *J. Mech. Des.* **1993**, *115*, 412–422. [CrossRef]
- Zhang, Y.; Sharf, I. Validation of nonlinear viscoelastic contact force models for low speed impact. *J. Appl. Mech.* **2009**, *76*, 051002. [CrossRef]
- Flores, P.; Machado, M.; Silva, M.T.; Martins, J.M. On the continuous contact force models for soft materials in multibody dynamics. *Multibody Syst. Dyn.* **2011**, *25*, 357–375. [CrossRef]
- Keller, J.B. Impact With Friction. *J. Appl. Mech.* **1986**, *53*, 1–4. [CrossRef]
- Schiehlen, W.; Seifried, R. Three approaches for elastodynamic contact in multibody systems. *Multibody Syst. Dyn.* **2004**, *12*, 1–16. [CrossRef]
- Gilardi, G.; Sharf, I. Literature survey of contact dynamics modelling. *Mech. Mach. Theory* **2002**, *37*, 1213–1239. [CrossRef]
- Schwab, A.; Meijaard, J.; Meijers, P. A comparison of revolute joint clearance models in the dynamic analysis of rigid and elastic mechanical systems. *Mech. Mach. Theory* **2002**, *37*, 895–913. [CrossRef]
- Haddadi, A.; Hashtrudi-Zaad, K. A New Method for Online Parameter Estimation of Hunt-Crossley Environment Dynamic Models. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 981–986.

20. Zhang, Y. Contact Dynamics for Rigid Bodies: Modeling and Experiments. Ph.D. Thesis, McGill University, Montreal, QC, Canada, 2007.
21. Pereira, C.M.; Ramalho, A.L.; Ambrósio, J.A.C. A critical overview of internal and external cylinder contact force models. *Nonlinear Dyn.* **2011**, *63*, 681–697. [CrossRef]
22. Machado, M.; Moreira, P.; Flores, P.; Lankarani, H.M. Compliant contact force models in multibody dynamics: Evolution of the Hertz contact theory. *Mech. Mach. Theory* **2012**, *53*, 99–121. [CrossRef]
23. Goldsmith, W. *Impact (Dover Civil and Mechanical Engineering)*, Annotated ed.; Dover Publications: Mineola, NY, USA, 2001; p. 416.
24. Johnson, K.L. *Contact Mechanics*, Reprint ed.; Cambridge University Press: Cambridge, UK, 2008; p. 468.
25. Flores, P.; Lankarani, H.M. Contact force models for multibody dynamics. In *Solid Mechanics and Its Applications*; Springer International Publishing: Cham, Switzerland, 2016; Volume 226. [CrossRef]
26. Skrinjar, L.; Slavič, J.; Boltežar, M. A review of continuous contact-force models in multibody dynamics. *Int. J. Mech. Sci.* **2018**, *145*, 171–187. [CrossRef]
27. Koshy, C.S.; Flores, P.; Lankarani, H.M. Study of the effect of contact force model on the dynamic response of mechanical systems with dry clearance joints: Computational and experimental approaches. *Nonlinear Dyn.* **2013**, *73*, 325–338. [CrossRef]
28. Mindlin, R.D. Compliance of elastic bodies in contact. *J. Appl. Mech.* **1949**, *16*, 259–268. [CrossRef]
29. Love, A.E.H. *A Treatise on the Mathematical Theory of Elasticity (Dover Books on Engineering)*, 4th ed.; Dover Publications: Mineola, NY, USA, 2011; p. 672.
30. Deresiewicz, H. A note on Hertz's theory of impact. *Acta Mech.* **1968**, *6*, 110–112. [CrossRef]
31. Radzimovsky, E.I. *Stress Distribution and Strength Condition of Two Rolling Cylinders Pressed Together*; Technical Report; University of Illinois Urbana-Champaign: Champaign, IL, USA, 1953.
32. Dubowsky, S. The Dynamic Response of Mechanical and Electromechanical Systems with Clearance and Nonlinear Impact Characteristics. Ph.D. Thesis, Columbia University, New York, NY, USA, 1971.
33. Dubowsky, S.; Freudenstein, F. Dynamic analysis of mechanical systems with clearances Part 1: formation of dynamic model. *J. Eng. Ind.* **1971**, *93*, 305–309. [CrossRef]
34. Dubowsky, S.; Freudenstein, F. Dynamic analysis of mechanical systems with clearances Part 2: dynamic response. *J. Eng. Ind.* **1971**, *93*, 310–316. [CrossRef]
35. Lankarani, H.M.; Nikravesh, P.E. A contact force model with hysteresis damping for impact analysis of multibody systems. *J. Mech. Des.* **1990**, *112*, 369–376. [CrossRef]
36. ESDU-78035. *Contact Phenomena. I: Stresses, Deflections and Contact Dimensions for Normally-Loaded Unlubricated Elastic Components*, Engineering Sciences Data Unit; Technical Report; 1978; ISBN 9780856792397.
37. Pereira, C.; Ramalho, A.; Ambrósio, J.A.C. Experimental and numerical validation of an enhanced cylindrical contact force model. In *Surface Effects and Contact Mechanics X*; WIT Transactions on Engineering Sciences; De Hosson, J.M., Brebbia, C., Eds.; WIT Press: Southampton, UK, 2011; Volume 1, pp. 49–60. [CrossRef]
38. Pereira, C.; Ramalho, A.; Ambrósio, J.A.C. An enhanced cylindrical contact force model. *Multibody Syst. Dyn.* **2015**, *35*, 277–298. [CrossRef]
39. Flores, P.; Ambrósio, J.A.C. Revolute joints with clearance in multibody systems. *Comput. Struct.* **2004**, *82*, 1359–1369. [CrossRef]
40. Flores, P.; Ambrósio, J.A.C.; Pimenta Claro, J.C.; Lankarani, H.M.; Koshy, C.S. A study on dynamics of mechanical systems including joints with clearance and lubrication. *Mech. Mach. Theory* **2006**, *41*, 247–261. [CrossRef]
41. Hunt, K.H.; Crossley, F.R.E. Coefficient of restitution interpreted as damping in vibroimpact. *J. Appl. Mech.* **1975**, *42*, 440–445. [CrossRef]
42. Tataru, Y.; Moriawaki, N. Study on impact of equivalent two bodies: Coefficients of restitution of spheres of brass, lead, glass, porcelain and agate, and the material properties. *Bull. JSME* **1982**, *25*, 631–637. [CrossRef]
43. Thornton, C. Coefficient of Restitution for Collinear Collisions of Elastic-Perfectly Plastic Spheres. *J. Appl. Mech.* **1997**, *64*, 383–386. [CrossRef]
44. Seifried, R.; Schiehlen, W.; Eberhard, P. Numerical and experimental evaluation of the coefficient of restitution for repeated impacts. *Int. J. Impact Eng.* **2005**, *32*, 508–524. [CrossRef]
45. Minamoto, H.; Kawamura, S. Moderately high speed impact of two identical spheres. *Int. J. Impact Eng.* **2011**, *38*, 123–129. [CrossRef]
46. Herbert, R.G.; McWhannell, D.C. Shape and frequency composition of pulses from an impact pair. *J. Eng. Ind.* **1977**, *99*, 513–518. [CrossRef]
47. Lee, T.W. Optimization of high speed geneva mechanisms. *J. Mech. Des.* **1981**, *103*, 621–630. [CrossRef]
48. Lee, T.W.; Wang, A.C. On the Dynamics of Intermittent-Motion Mechanisms. Part 1: Dynamic Model and Response. *J. Mech. Transm. Autom. Des.* **1983**, *105*, 534–540. [CrossRef]
49. Wang, A.C.; Lee, T.W. On the Dynamics of Intermittent-Motion Mechanisms. Part 2: Geneva Mechanisms, Ratchets, and Escapements. *J. Mech. Transm. Autom. Des.* **1983**, *105*, 541–551. [CrossRef]
50. Wang, A.C. On the Kinematics and Dynamics of Intermittent Motion Mechanisms. Ph.D. Thesis, Rutgers University, New Brunswick, NJ, USA, 1983.
51. Khulief, Y.A. Dynamic Analysis of Multibody Systems with Intermittent Motion. Ph.D. Thesis, University of Illinois at Chicago, Chicago, IL, USA, 1985.

52. Khulief, Y.A.; Shabana, A.A. A continuous force model for the impact analysis of flexible multibody systems. *Mech. Mach. Theory* **1987**, *22*, 213–224. [CrossRef]
53. Khulief, Y.A.; Shabana, A.A. Dynamic analysis of constrained system of rigid and flexible bodies with intermittent motion. *J. Mech. Transm. Autom. Des.* **1986**, *108*, 38–45. [CrossRef]
54. Tsuji, Y.; Tanaka, T.; Ishida, T. Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. *Powder Technol.* **1992**, *71*, 239–250. [CrossRef]
55. Lankarani, H.M.; Nikravesh, P.E. Continuous contact force models for impact analysis in multibody systems. *Nonlinear Dyn.* **1994**, *5*, 193–207. [CrossRef]
56. Shivaswamy, S. Modeling Contact Forces and Energy Dissipation during Impact in Mechanical Systems. Ph.D. Thesis, Wichita State University, Wichita, KS, USA, 1997.
57. Shivaswamy, S.; Lankarani, H.M. Impact Analysis of Plates Using Quasi-Static Approach. *J. Mech. Des.* **1997**, *119*, 376–381. [CrossRef]
58. Rhee, J.; Akay, A. Dynamic response of a revolute joint with clearance. *Mech. Mach. Theory* **1996**, *31*, 121–134. [CrossRef]
59. Marhefka, D.; Orin, D. A compliant contact model with nonlinear damping for simulation of robotic systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **1999**, *29*, 566–572. [CrossRef]
60. Gonthier, Y.; McPhee, J.; Lange, C.; Piedœuf, J.C. A Regularized Contact Model with Asymmetric Damping and Dwell-Time Dependent Friction. *Multibody Syst. Dyn.* **2004**, *11*, 209–233. [CrossRef]
61. Machado, M.; Flores, P.; Ambrósio, J.A.C. A Lookup-Table-Based Approach for Spatial Analysis of Contact Problems. *J. Comput. Nonlinear Dyn.* **2014**, *9*, 041010. [CrossRef]
62. Machado, M.; Flores, P.; Pimenta Claro, J.C.; Ambrósio, J.A.C.; Silva, M.; Completo, A.; Lankarani, H.M. Development of a planar multibody model of the human knee joint. *Nonlinear Dyn.* **2010**, *60*, 459–478. [CrossRef]
63. Machado, M.; Flores, P.; Ambrósio, J.A.C. Influence of the contact model on the dynamic response of the human knee joint. *Proc. Inst. Mech. Eng. Part K J. Multi-Body Dyn.* **2011**, *225*, 344–358. [CrossRef]
64. Gharib, M.; Hurmuzlu, Y. A new contact force model for low coefficient of restitution impact. *J. Appl. Mech.* **2012**, *79*. [CrossRef]
65. Stoianovici, D.; Hurmuzlu, Y. A Critical Study of the Applicability of Rigid-Body Collision Theory. *J. Appl. Mech.* **1996**, *63*, 307–316. [CrossRef]
66. Hu, S.; Guo, X. A dissipative contact force model for impact analysis in multibody dynamics. *Multibody Syst. Dyn.* **2015**, *35*, 131–151. [CrossRef]



Article

Obstacle Avoidance in Operational Configuration Space Kinematic Control of Redundant Serial Manipulators

Adrian Peidro ¹ and Edward J. Haug ^{2,*}

¹ Systems Engineering and Automation Department, Miguel Hernandez University, Avda de la Universidad s/n, 03202 Elche, Spain; apeidro@umh.es

² Department of Mechanical and Industrial Engineering, The University of Iowa, Iowa City, IA 52242, USA

* Correspondence: echaug@gmail.com

Abstract: Kinematic control of redundant serial manipulators has been carried out for the past half century based primarily on a generalized inverse velocity formulation that is known to have mathematical deficiencies. A recently developed inverse kinematic configuration mapping is employed in an operational configuration space differentiable manifold formulation for redundant-manipulator kinematic control with obstacle avoidance. This formulation is shown to resolve deficiencies in the generalized inverse velocity formulation, especially for high-degree-of-redundancy manipulators. Tracking a specified output trajectory while avoiding obstacles for four- and twenty-degree-of-redundancy manipulators is carried out to demonstrate the effectiveness of the differentiable manifold approach for applications with a high degree of redundancy and to show that it indeed resolves deficiencies of the conventional generalized inverse velocity formulation in challenging applications.

Keywords: obstacle avoidance; kinematic control; redundant manipulators; manipulator differentiable manifold

1. Introduction

1.1. Basics of Redundant-Serial-Manipulator Kinematics

A serial manipulator is comprised of a chain of bodies that are connected by single-degree-of-freedom joints. Joint relative input coordinates y_i between bodies in the chain define the configuration of outboard bodies relative to their inboard counterparts. The terminal body in the chain is the end effector, whose output coordinates characterize manipulator working capability and are defined as twice continuously differentiable functions of input coordinates, in the form of forward kinematic mapping.

$$z = G(y) \quad (1)$$

Input coordinates $y \in \mathbb{R}^n$ are independent generalized coordinates [1] that define the configuration of the underlying mechanism, and output coordinates $z \in \mathbb{R}^m$ of the end effector, with $m < n$, define functionality of the kinematically redundant manipulator. Here, \mathbb{R}^k refers to k -dimensional Euclidean vector space with elements $x \in \mathbb{R}^k$ in the form of column vectors $x = [x_1 \ \cdots \ x_k]^T$, where superscript T denotes matrix transpose. Bold characters denote vectors and matrices.

An input–output pair (y, z) that satisfies Equation (1) defines a manipulator configuration, denoted $x = [y^T \ z^T]^T \in \mathbb{R}^{n+m}$. The manipulator configuration space is defined as

$$X = \{x \in \mathbb{R}^{n+m} : G(y) - z = 0\} \quad (2)$$

As observed in [2], X is the graph of Equation (1). As such, under moderate regularity conditions, it is a differentiable manifold with the single chart $(\mathbb{R}^n, \phi(x))$, where

Citation: Peidro, A.; Haug, E.J. Obstacle Avoidance in Operational Configuration Space Kinematic Control of Redundant Serial Manipulators. *Machines* **2024**, *12*, 10. <https://doi.org/10.3390/machines12010010>

Academic Editor: Xinjun Liu

Received: 12 November 2023

Revised: 20 December 2023

Accepted: 20 December 2023

Published: 23 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

$\phi(x) = y$ [2]. This manifold, however, contains information only on forward kinematics and no information regarding inverse kinematics.

The manipulator configuration space often contains *singular configurations* relative to inverse kinematics. If the rank of the *Jacobian matrix* of derivatives of $G(y)$ with respect to y , defined as the $m \times n$ matrix $G'(y) \equiv [\partial G_i(y)/\partial y_j]$, is less than m at a configuration $\bar{x} = [\bar{y}^T \ \bar{z}^T]^T \in X$, then in a neighborhood of \bar{x} , there exists no continuously differentiable set-valued inverse kinematic mapping with $n - m$ arbitrary parameters [3,4]. To avoid difficulties associated with inverse kinematic singular configurations in X , the *regular configuration space* is defined as

$$\tilde{X} = \{x \in X : \text{rank}(G'(y)) = m\} \tag{3}$$

In \tilde{X} , the manipulator *degree of redundancy* is $r = n - m$.

1.2. A Four-Degree-of-Redundancy Manipulator

Most applications of redundant manipulators have just one degree of redundancy, i.e., $r = n - m = 1$. To relate the mathematical formulation to a real manipulator, consider the planar serial manipulator of Figure 1 that has $n = 6$ inputs, $y \in \mathbb{R}^6$, and two outputs, $z \in \mathbb{R}^2$, hence $r = 6 - 2 = 4$ degrees of redundancy.

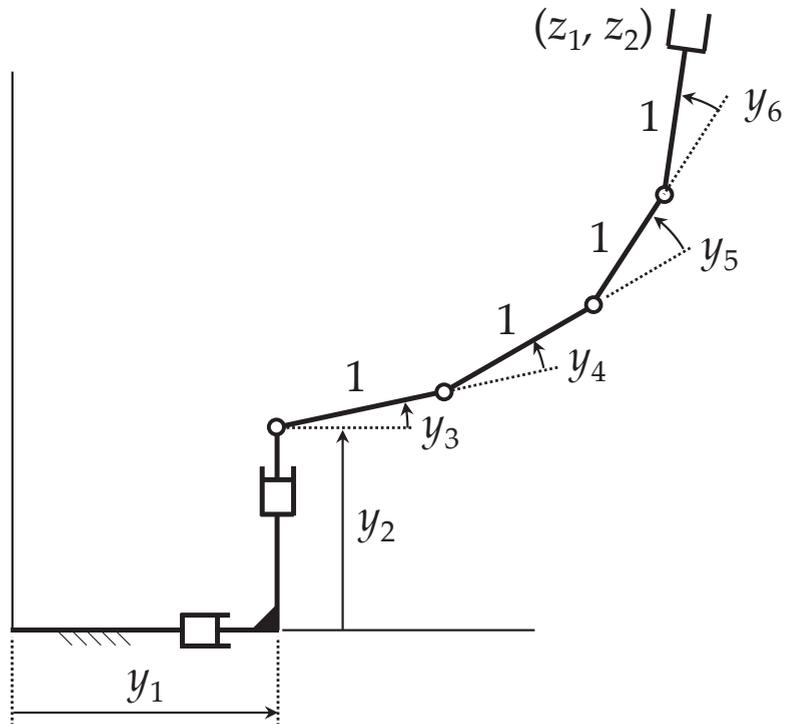


Figure 1. Serial manipulator with a degree of redundancy of 4.

Using the notation $y_{ij} = y_i + \dots + y_j$, the forward kinematic mapping of Equation (1) is

$$z = G(y) = \begin{bmatrix} y_1 + \cos y_3 + \cos y_{34} + \cos y_{35} + \cos y_{36} \\ y_2 + \sin y_3 + \sin y_{34} + \sin y_{35} + \sin y_{36} \end{bmatrix} \tag{4}$$

with the full-rank Jacobian, for all $x \in X = \tilde{X}$,

$$\begin{aligned} G'(\mathbf{y}) &= \begin{bmatrix} 1 & 0 & a & b & -(\sin y_{35} + \sin y_{36}) & -\sin y_{36} \\ 0 & 1 & c & d & (\cos y_{35} + \cos y_{36}) & \cos y_{36} \end{bmatrix} \\ a &= -(\sin y_3 + \sin y_{34} + \sin y_{35} + \sin y_{36}) \\ b &= -(\sin y_{34} + \sin y_{35} + \sin y_{36}) \\ c &= (\cos y_3 + \cos y_{34} + \cos y_{35} + \cos y_{36}) \\ d &= (\cos y_3 + \cos y_{34} + \cos y_{35}) \end{aligned} \quad (5)$$

1.3. Generalized-Inverse-Velocity-Based Redundancy Resolution

Differentiating Equation (1) with respect to time yields the *kinematic velocity equation*:

$$\dot{\mathbf{z}} = G'(\mathbf{y})\dot{\mathbf{y}} \quad (6)$$

Since Equation (6) is linear in velocity, for a given configuration, it may be thought that analysis in the linear velocity space is easier than in the nonlinear regular configuration space of Equation (3). When $m = n$ and the manipulator is nonredundant, this may in fact be the case. In \tilde{X} , if $m = n$, the manipulator is nonredundant, $G'(\mathbf{y})$ is square and nonsingular, and there is a unique solution of Equation (6) for $\dot{\mathbf{y}}$,

$$\dot{\mathbf{y}} = G'^{(-1)}(\mathbf{y})\dot{\mathbf{z}} \quad (7)$$

In this special case, Equation (7) is an ordinary differential equation (ODE). For specified continuous $\dot{\mathbf{z}}(t)$ and initial condition $\mathbf{y}(0) = \mathbf{y}^0$, the initial-value problem with Equation (7) has a unique solution in a neighborhood of $t = 0$.

For redundant manipulators with $n > m$, attractive properties of ODE for Equation (6) are no longer available. At a redundant manipulator configuration $x \in \tilde{X}$, $G'(\mathbf{y})$ has full rank, but $G'(\mathbf{y})$ is not a square matrix, and Equation (6) with given $\dot{\mathbf{z}}(t)$ cannot be solved for a unique value of $\dot{\mathbf{y}}$. In this case, Equation (6) is not an ODE in \mathbf{y} . It is a *Pfaffian differential equation* [5] that behaves more like a partial differential equation than an ODE. While the ODE of Equation (7) with $m = n$ has a general solution for \mathbf{y} that depends on n arbitrary constants, a general solution of Equation (6) for \mathbf{y} with $n > m$ depends on arbitrary vector functions $\mathbf{p}(t)$ [5]. As shown in [4], trading the nonlinear manipulator equations of Equations (2) and (3) for the Pfaffian differential equation of Equation (6) is a questionable decision.

A generalized-inverse-velocity-based redundancy resolution formulation for Equation (6) was introduced by Whitney [6] half a century ago in an attempt to create an ODE formulation for redundant-manipulator kinematics. At a configuration $x \in \tilde{X}$ with $n > m$, $G'(\mathbf{y})$ has full rank and the *Moore–Penrose generalized inverse* [6–8] is defined as

$$G'^{\dagger}(\mathbf{y}) = G'^{\text{T}}(\mathbf{y}) \left(G'(\mathbf{y})G'^{\text{T}}(\mathbf{y}) \right)^{-1} \quad (8)$$

Direct manipulation verifies that $G'(\mathbf{y})G'^{\dagger}(\mathbf{y}) = \mathbf{I}$, and

$$\dot{\mathbf{y}} = G'^{\dagger}(\mathbf{y})\dot{\mathbf{z}} + \left(\mathbf{I} - G'^{\dagger}(\mathbf{y})G'(\mathbf{y}) \right) \dot{\mathbf{y}}^0(t) \quad (9)$$

satisfies Equation (6) for arbitrary $\dot{\mathbf{y}}^0(t) \in \mathbb{R}^n$. While $\dot{\mathbf{y}}$ of Equation (9) satisfies Equation (6), it does not provide a solution $\mathbf{y}(t)$ of Equation (1), and there is no reason to believe Equations (6) and (9) are equivalent.

1.4. Problems with Generalized-Inverse-Velocity-Based Redundancy Resolution

In the redundant-manipulator literature, e.g., [6–10] and references cited therein, Equation (9) has been treated as a differential equation that relates output to input. It has been analytically and computationally shown, however, that use of Equation (9) leads to

irregularities in predicted manipulator performance [4,11–13], including numerical drift, nonholonomic behavior, and divergence of computation. Analysis has shown that use of the generalized inverse of Equation (8) leads to nonholonomic equations of redundancy resolution that create systematic noncyclicity of the manipulator [12,13], i.e., a periodic output trajectory maps into a nonperiodic input trajectory. While some effort has been devoted to creating generalized inverses that overcome these problems, e.g., [14,15] and references cited therein, no comprehensive result has been reported. A treatment based on concepts similar to those employed herein is shown to partially resolve the noncyclicity problem [16]. Theoretical and numerical results presented in [4,16] complement the literature cited and show that the generalized inverse velocity approach to manipulator redundancy resolution is fundamentally flawed.

While the foregoing analyses of deficiencies of the generalized inverse velocity approach do not focus specifically on obstacle avoidance, the inaccuracies implied cast doubt on use of the approach for kinematic control with obstacle avoidance. In particular, no inverse kinematic mapping at the configuration level is defined with free parameters, which means that an indirect velocity approach to obstacle avoidance must be employed. Since obstacle avoidance occurs in configuration space, not in velocity space, an adaptation of generalized inverse velocity differential equations must be created to treat an application that is stated in terms of manipulator configuration.

1.5. Manipulator Trajectory Planning and Obstacle Avoidance

An extensive literature on kinematically redundant manipulator trajectory planning and kinematic control appeared in the last quarter of the 20th century, based primarily on the generalized inverse velocity representation of Equation (9), e.g., [6–10] and references cited therein. The most basic kinematic control formulation seeks to use manipulator redundancy to cause manipulator output $\mathbf{z}(t)$ to follow a desired *output trajectory* $\mathbf{z}_d(t)$ and to satisfy constraints such as avoiding collision of manipulator links with obstacles. Objectives and constraints in these applications are most naturally stated in terms of manipulator input and output coordinates, rather than velocities. The apparent simplicity of the velocity equations of Section 1.3, with the redundancy-related arbitrary velocity $\dot{\mathbf{y}}^0(t)$ of Equation (9), however, has led research and development engineers to adopt the generalized inverse velocity mapping of Equation (9) to address the kinematic control objective, e.g., [7,8] and references cited therein.

Quite recently, a configuration-based inverse kinematic mapping for kinematically redundant serial manipulators has been presented that avoids deficiencies associated with the generalized inverse velocity mapping approach [3,4]. This method is summarized in Section 3 and applied for manipulator path planning with obstacle avoidance.

2. Generalized-Inverse-Velocity-Based Kinematic Control and Obstacle Avoidance

The conventional approach to satisfying constraints such as obstacle avoidance using the generalized inverse velocity approach is the *projected-gradient method*, employing the second term of Equation (9), i.e., the null space term, to minimize a cost function $f(\mathbf{y})$ that models the secondary criterion. This is performed by defining

$$\dot{\mathbf{y}}^0(t) = -k\nabla_{\mathbf{y}}f(\mathbf{y}(t)) \quad (10)$$

with $k > 0$, where $\nabla_{\mathbf{y}}f(\mathbf{y}) = [\partial f(\mathbf{y})/\partial y_1, \dots, \partial f(\mathbf{y})/\partial y_n]^T$, and projecting it onto the null space of the Jacobian matrix by means of the projector $(\mathbf{I} - \mathbf{G}'^{\dagger}(\mathbf{y})\mathbf{G}'(\mathbf{y}))$ of Equation (9) [17]. As a result, $\mathbf{y}(t)$ will move in a direction that decreases $f(\mathbf{y})$, without affecting the desired output trajectory $\mathbf{z}_d(t)$, i.e., *self-motion*. When the constraint is to avoid ob-

stacles, $f(\mathbf{y})$ can be chosen as any function that penalizes manipulator link proximity to obstacles [18]. For example, if $f(\mathbf{y})$ is defined as

$$f(\mathbf{y}) = \sum_{i,j} 1/d_{ij}(\mathbf{y}) \quad (11)$$

where $d_{ij}(\mathbf{y})$ is the minimum distance between link i and obstacle j , manipulator links will tend to move away from obstacles.

Another approach to obstacle avoidance using the generalized inverse velocity method consists in choosing $\dot{\mathbf{y}}^0(t)$ as an input velocity that attempts to generate an escape velocity $\dot{\mathbf{z}}_o(t)$ at the point P_0 of the manipulator that is closest to the obstacle, where such escape velocity is directed away from the obstacle. This requires yet another generalized inversion of the Jacobian matrix J_o that maps input velocities to the velocity of P_0 [19]. Alternatively, Lee and Buss [18] use the transpose of J_o that has lower computational burden than generalized matrix inversion and avoids problems with singularities, at the expense of accuracy in generation of the desired escape velocity. This is not crucial, as long as the link moves away from the obstacle.

When a hierarchy of tasks exists, i.e., there is a primary task and several secondary tasks with different levels of priority, where each task has an associate Jacobian, lower-priority tasks are optimized by projecting velocities on the null space of higher-priority tasks [20]. This hierarchy of projections is used in the method of Saturation in the Null Space (SNS) [21], which saturates input velocities that reach their limits and slows the task, i.e., scales time, to restore feasibility if such saturations render the desired task infeasible. The SNS method can achieve obstacle avoidance by adapting limits of joint velocities near obstacles [22] or by augmenting joint velocities with velocities of control points on links of the manipulator and limiting these velocities in the vicinity of obstacles [23]. The SNS method is usually formulated at the velocity level, but it has been extended in [24,25] to function also at acceleration and torque levels.

In any case, the foregoing methods are based on use of the generalized inverse in the velocity equation, which leads to deficiencies discussed in Section 1.4. Moreover, these methods operate at velocity or acceleration levels, whereas constraints such as obstacle avoidance are naturally defined at the configuration level. This requires recasting these constraints approximately to set bounds on velocities, e.g., joint range limits are typically reformulated as bounds on joint velocities by using a forward difference approximation of the derivative of configuration [22], which may lead to violation of these constraints.

Use of the generalized inverse in the velocity equation is not a robust way to handle obstacle avoidance, because it is based on a velocity equation that needs to be numerically integrated to obtain the configuration of the manipulator. For example, if $\dot{\mathbf{y}}^0(t)$ of Equation (10) is inserted into Equation (9), which is integrated numerically with a given time step Δt , the resulting trajectory of the manipulator may not be feasible. This is because, as time progresses, Equation (9) is integrated, and the cost function $f(\mathbf{y})$ may not be minimized at a sufficiently high rate to avoid obstacles. Indeed, if the minimization of $f(\mathbf{y})$ is not performed sufficiently quickly as the manipulator approaches obstacles, penetration of an obstacle will occur that will produce a zero denominator in Equation (11), i.e., $d_{ij}(\mathbf{y}) = 0$ for some (i,j) , and this will lead to failure of numerical integration. As is demonstrated in Sections 4.1.2 and 4.2.3, it is necessary to carefully select the values of Δt (time step) and k (gain appearing in Equation (10)). In fact, as shown, it may be necessary to use extremely small values of Δt and k to obtain motion that is free of obstacle penetration. On the one hand, simulating the full trajectory using a small Δt means long computation times. On the other hand, when using Equation (9) for real-time kinematic control of the manipulator, a small Δt imposes extreme demands on the control hardware, which must be able to operate at such small sample times.

3. Operational Configuration Space Kinematic Control and Obstacle Avoidance

To replace the generalized inverse velocity formulation of Section 2 with a configuration space formulation, it is required that a set-valued inverse configuration kinematic mapping be constructed, i.e., that a solution $\mathbf{y} = \mathbf{g}(\mathbf{z}, \mathbf{v})$ of Equation (1) be found with an arbitrary vector of parameters $\mathbf{v} \in \mathbb{R}^{n-m}$. Such a mapping has been presented in [3] and extended in [4] on an operational space differentiable manifold, as summarized in this section.

3.1. Inverse Configuration Kinematics

The entire regular configuration space $\tilde{\mathcal{X}}$ cannot, in general, be characterized by a single continuously differentiable inverse kinematic mapping. The only practical global inverse kinematic representation is based on concepts of differential geometry [2] that employ local representations on open subsets $\mathbb{N}^j \in \tilde{\mathcal{X}}$, whose union is the entire regular configuration space, i.e., $\cup_j \mathbb{N}^j = \tilde{\mathcal{X}}$. In each \mathbb{N}^j , there is a *base point* $\bar{\mathbf{x}}^j$ about which the inverse kinematic mapping is constructed. The inverse kinematic mapping process that follows takes place on each \mathbb{N}^j . For a given j and base point $\bar{\mathbf{x}}^j$ in \mathbb{N}^j , define the $n \times m$ matrix \mathbf{U}^j and an $n \times (n - m)$ matrix \mathbf{V}^j such that

$$\mathbf{U}^j = \mathbf{G}'^T(\bar{\mathbf{y}}^j) \quad \mathbf{G}'(\bar{\mathbf{y}}^j)\mathbf{V}^j = \mathbf{0} \quad \mathbf{V}^{jT}\mathbf{V}^j = \mathbf{I} \quad (12)$$

where \mathbf{V}^j is computed as a matrix whose columns form an orthonormal basis of the null space of $\mathbf{G}'(\bar{\mathbf{y}}^j)$, e.g., in MATLAB, using *singular-value decomposition* [26]. The matrices \mathbf{U}^j and \mathbf{V}^j are defined to be constant on \mathbb{N}^j . Note from the second equation of Equation (12) that $\mathbf{U}^{jT}\mathbf{V}^j = \mathbf{0}$ and $\mathbf{V}^{jT}\mathbf{U}^j = \mathbf{0}$. Since $\mathbf{G}'(\bar{\mathbf{y}}^j)$ has full rank, so do \mathbf{U}^j and \mathbf{V}^j . Further, since $\mathbf{V}^{jT}\mathbf{U}^j = \mathbf{0}$, the columns of \mathbf{V}^j are orthogonal to the columns of \mathbf{U}^j and vice-versa. The n linearly independent columns of \mathbf{U}^j (m columns) and \mathbf{V}^j ($n - m$ columns) therefore span \mathbb{R}^n .

Using \mathbf{V}^j and \mathbf{U}^j of Equation (12), any solution of Equation (1) for \mathbf{y} in a neighborhood of $\bar{\mathbf{y}}^j$ can be written in the form

$$\mathbf{y} = \bar{\mathbf{y}}^j + \mathbf{V}^j(\mathbf{v} - \bar{\mathbf{v}}^j) - \mathbf{U}^j(\mathbf{u} - \bar{\mathbf{u}}^j) \quad (13)$$

where $\bar{\mathbf{v}}^j$ and $\bar{\mathbf{u}}^j$ are values of \mathbf{v} and \mathbf{u} associated with $\bar{\mathbf{x}}^j$ on the trajectory that first enters \mathbb{N}^j . They are introduced to assure continuity of \mathbf{y} as a function of \mathbf{v} and \mathbf{z} . On \mathbb{N}^1 , $\bar{\mathbf{v}}^1 = \mathbf{0}$ and $\bar{\mathbf{u}}^1 = \mathbf{0}$. Note that, at $\mathbf{y} = \bar{\mathbf{y}}^j$ in Equation (13), $\mathbf{v} = \bar{\mathbf{v}}^j$ and $\mathbf{u} = \bar{\mathbf{u}}^j$. To see that there is a unique solution of Equation (1) with \mathbf{y} of Equation (13), i.e., a unique solution of

$$\mathbf{G}(\bar{\mathbf{y}}^j + \mathbf{V}^j(\mathbf{v} - \bar{\mathbf{v}}^j) - \mathbf{U}^j(\mathbf{u} - \bar{\mathbf{u}}^j)) - \mathbf{z} = \mathbf{0} \quad (14)$$

for \mathbf{u} as a function of \mathbf{z} and \mathbf{v} in a neighborhood of $\mathbf{z} = \bar{\mathbf{z}}^j$ and $\mathbf{v} = \bar{\mathbf{v}}^j$, the derivative of the left side of Equation (14) with respect to \mathbf{u} , evaluated at $\bar{\mathbf{x}}^j$, i.e., at $\mathbf{z} = \bar{\mathbf{z}}^j$ and $\mathbf{v} = \bar{\mathbf{v}}^j$, using the chain rule of differentiation, is $-\mathbf{G}'(\bar{\mathbf{y}}^j)\mathbf{U}^j = -\mathbf{U}^{jT}\mathbf{U}^j$, which is nonsingular. Thus, the implicit-function theorem [27] implies existence of a unique, twice continuously differentiable solution,

$$\mathbf{u} = \mathbf{h}^j(\mathbf{z}, \mathbf{v}) \quad (15)$$

of Equation (14) in a neighborhood of $\bar{\mathbf{x}}^j$. From Equation (13),

$$\mathbf{y}(\mathbf{z}, \mathbf{v}) = \bar{\mathbf{y}}^j + \mathbf{V}^j(\mathbf{v} - \bar{\mathbf{v}}^j) - \mathbf{U}^j(\mathbf{h}^j(\mathbf{z}, \mathbf{v}) - \bar{\mathbf{u}}^j) \quad (16)$$

This is the desired *set-valued inverse configuration kinematic mapping* on \mathbb{N}^j .

If $\mathbf{z}(t)$ and $\mathbf{v}(t)$ are periodic of period t_p on \mathbb{N}^j , i.e., $\mathbf{z}(t + t_p) = \mathbf{z}(t)$ and $\mathbf{v}(t + t_p) = \mathbf{v}(t)$, since Equation (16) holds throughout \mathbb{N}^j ,

$$\mathbf{y}(t + t_p) = \bar{\mathbf{y}}^j + \mathbf{V}^j(\mathbf{v}(t + t_p) - \bar{\mathbf{v}}^j) - \mathbf{U}^j(\mathbf{h}(\mathbf{z}(t + t_p), \mathbf{v}(t + t_p)) - \bar{\mathbf{u}}^j) = \mathbf{y}(t) \quad (17)$$

Thus, $y(t)$ is periodic of period t_p , and the manipulator is *cyclic* on N^j . This shows that, with the differentiable manifold formulation, the manipulator is *locally cyclic* [4,16]. In kinematic control with obstacle avoidance, however, during a transient control time interval in which $v(t)$ is activated to resist penetration of an obstacle, it will generally not be periodic. During this time interval, the manipulator will not be locally cyclic. If, after transient action to avoid obstacles, the desired periodic output goes through cycles without control action required to avoid obstacles, i.e., $v(t)$ becomes constant, then during these cycles, the manipulator will be locally cyclic.

The role of the function $h^j(z, v)$ is important in assuring satisfaction of Equation (1). This is in contrast with the generalized inverse velocity approach presented in Section 2, in which Equation (1) is ignored. A computationally efficient iterative method for evaluation of $h^j(z, v)$ is presented in [3,4]. In this computation, when the number of iterations required for convergence exceeds a specified tolerance, the associated configuration $x = [y^T \ z^T]^T$ is designated \bar{x}^{j+1} , the associated y, v , and u are designated $\bar{y}^{j+1}, \bar{v}^{j+1}$, and \bar{w}^{j+1} , a new neighborhood N^{j+1} is entered, and the parameterization is redefined. As shown in [28] for dynamic system simulation, less than 0.1% of CPU time and no user interaction are required for reparameterization. For more detail on the process of selecting configurations \bar{x}^j and reparameterization calculations, see [3,4,28].

For a given output z , with $v \in R^{n-m}$ arbitrary in a neighborhood of $v = \bar{v}^j$, Equation (16) defines a set of input coordinates,

$$SMM(z) = \left\{ y = \bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{w}^j) : v \text{ in a neighborhood of } \bar{v}^j \right\} \quad (18)$$

called the *manipulator self-motion manifold* in input space associated with output z . Since $h^j(z, v)$ is the solution of Equation (14), $G(\bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{w}^j)) - z = 0$, for all v in a neighborhood of $v = \bar{v}^j$, $y(z, v)$ of Equation (16) maps into z , i.e., $z = G(y(z, v))$. Components of the vector $v \in R^{n-m}$ are called *self-motion coordinates*. With arbitrary self-motion coordinates v in a neighborhood of \bar{v}^j , Equation (16) defines $n - m$ *redundant degrees of freedom* $v \in R^{n-m}$ of the manipulator that enables it to meet requirements that could not be met with a nonredundant manipulator. The restriction of v to a neighborhood of \bar{v}^j in the foregoing is to meet hypotheses of the implicit-function theorem. As shown in Sections 4.1.1 and 4.2.2, Equations (16) and (18) may hold for large v in applications.

It is important to note that the self-motion manifold of Equation (18) is defined at the configuration level. Since configuration information is not defined in the generalized inverse velocity formulation, the self-motion manifold and self-motion coordinates cannot be explicitly defined and are not available for obstacle avoidance in that formulation.

3.2. Operational Space Differentiable Manifold

Defining manipulator *operational coordinates* $w = [z^T \ v^T]^T \in R^n$ and *functional coordinates* $s = [y^T \ w^T]^T \in R^{2n}$, the manipulator *regular functional configuration space* is

$$\tilde{S} = \bigcup_{N^j} \left\{ s \in R^{2n} : \bar{x}^j = [\bar{y}^{jT} \ \bar{z}^{jT}]^T \in \tilde{X}, y \equiv \psi^j(w) = \bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{w}^j) \right\} \quad (19)$$

Similar to the definition of the manipulator configuration space of Equation (2) as the product of input and output spaces, the serial manipulator regular functional configuration space is defined as the product of input and operational spaces. This is important in establishing \tilde{S} as a *differentiable manifold*, as presented in [4]. A family of charts $(N^j, \psi^j(w))$ (an *atlas*) is defined to cover \tilde{S} , such that the mappings ψ^j are *compatible* and \tilde{S} is a differentiable manifold that is parameterized by w [2,4]. The differentiable manifold can thus be parameterized by either y or w . It cannot, however, be parameterized by only output z .

Kinematics on \tilde{S} must be carried out on individual charts $(N^j, \psi^j(w))$ and transitioned to adjacent charts as manipulator configurations progress along a trajectory in \tilde{S} , as shown

schematically in Figure 2. Piecewise analysis on charts is unavoidable since, in general, there is no globally valid operational coordinate parameterization $\psi(w)$ of \tilde{S} [2]. This attribute of differential geometry that transforms local to global properties of sets and mappings is one of its greatest contributions. The unavoidable reality, however, is that one must adopt local operational space parameterizations, since no global parameterization generally exists.

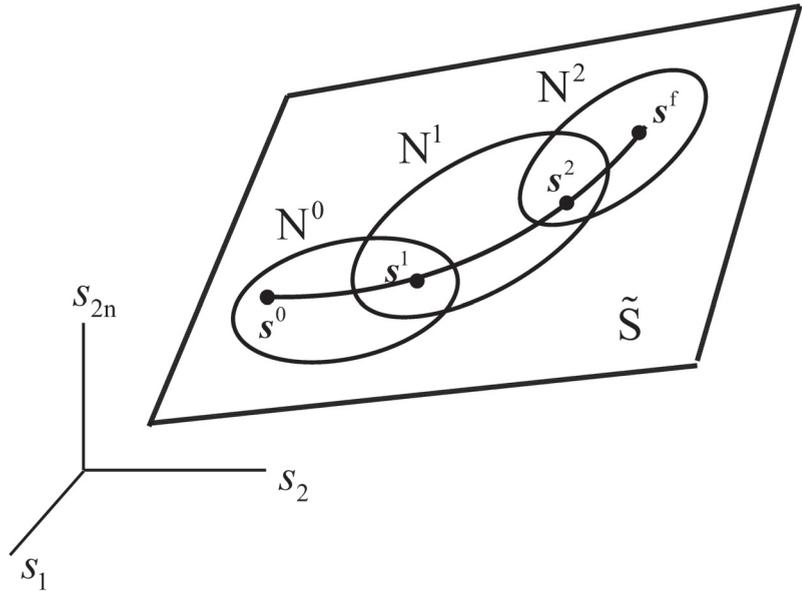


Figure 2. Trajectory along charts in \tilde{S} .

3.3. Differentiable-Manifold Output Trajectory Tracking and Obstacle Avoidance Algorithm

An output trajectory tracking and obstacle avoidance algorithm based on the inverse kinematic mapping of Equation (16) is presented to carry out kinematic control more rigorously and efficiently than with the generalized-inverse-velocity-based method of Section 2. In [3], the inverse mapping was employed to avoid collisions in redundant manipulators using an algorithm that marched along one-dimensional self-motion manifolds. That algorithm, however, is limited to manipulators with a degree of redundancy of one. First, a nominal trajectory $y_n(t)$ for the inputs y is selected that yields the desired output trajectory $z_d(t)$, without exploiting kinematic redundancy. When the manipulator starts to move, it follows this nominal trajectory until a collision is detected at some z^* of the desired output trajectory, i.e., until the intersection of two bodies is not empty. When a collision is detected, the algorithm presented in [3] performs self-motions defined by Equation (18) that keeps $z = z^*$, i.e., the self-motion coordinate v is varied on a grid with a given step Δv along one direction (or the opposite) of the self-motion manifold until the interference disappears. This algorithm is feasible only when the self-motion manifold is one-dimensional, because one only needs to march along self-motion curves, exhaustively searching until a collision-free configuration is found.

In manipulators with higher degrees of redundancy, such as the manipulator of Section 1.2, the obstacle avoidance algorithm used in [3] is not feasible, because one would need to perform an exhaustive grid search in the higher-dimensional space of self-motion coordinates v , which would be computationally prohibitive, e.g., a four-dimensional grid for the manipulator of Section 1.2. The new algorithm employed herein efficiently treats obstacle avoidance using the inverse mapping of Equation (16) for higher degrees of redundancy, as the examples of Sections 4.1.1 and 4.2.2 demonstrate.

The new algorithm is based on the fact that, in the vicinity of a contact between two bodies, it is possible to define a gap function g [29,30] that is the signed distance between the closest points of the two bodies along their common normal. This works even when one of the bodies has a nonsmooth shape and does not have a unique normal at the point of contact [29,30]. Figure 3 illustrates this gap function g . Note that g is a *signed* distance, which is positive when the bodies do not intersect, zero when their boundaries touch, and negative when one body has penetrated the other. The gap function g depends on the relative pose of the contacting bodies, which is defined by input coordinates, i.e., $g = g(y)$. Using the inverse mapping of Equation (16), one can write the gap function in terms of the operational coordinates, i.e., $g = g(y(z, v))$.

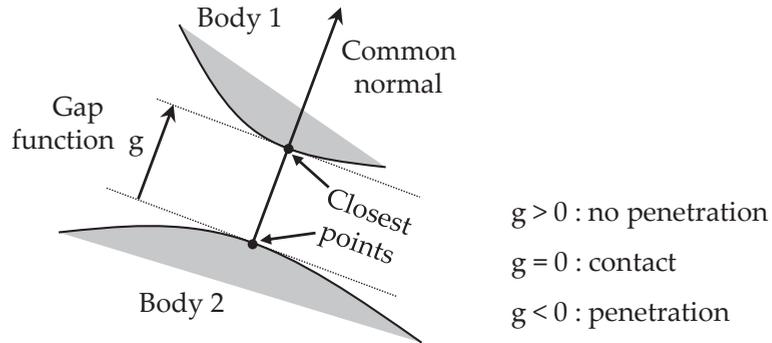


Figure 3. Definition of gap function g .

Taking the foregoing into account, the new algorithm is as follows:

STEP 1: Initialize $t = t_0$ and select a time step Δt . The desired output trajectory is $z_d(t)$ for $t_0 < t < t_1$. The initial configuration y_0 yields the initial desired z , i.e., $G(y_0) = z_d(t_0)$. Construct the inverse mapping of Equation (16) in a neighborhood of $\bar{y} = y_0$, i.e., compute U and V and initialize $v = 0$ and $u = 0$. Throughout execution of the algorithm, the self-motion vector v is updated only when necessary to avoid obstacles. If other secondary goals are to be met, the self-motion vector v can be used to optimize these secondary goals. In the present algorithm, the focus is on the use of redundancy only for obstacle avoidance.

STEP 2: Set $z = z_d(t)$, solve for $u = h(z, v)$ as in Section 3.1, and evaluate $y = y(v, u)$ of Equation (16). Whenever necessary, perform a reparameterization, i.e., a transition between charts illustrated in Figure 2, as outlined in Section 3.2.

STEP 3: For the computed configuration y , check if interferences occur, i.e., if any link of the manipulator intersects an obstacle. If no intersections occur, continue with step 4. Otherwise, go to step 5.

STEP 4: Send the collision-free configuration y to the controller of the manipulator and proceed to the next time step, i.e., update $t = t + \Delta t$. If $t \leq t_1$, return to step 2. Otherwise, the desired trajectory has been completed and the algorithm ends.

STEP 5: If the algorithm has reached this step, there exists mechanical interference between the manipulator and at least one obstacle for the current value of $y = y^*$, which is obtained from $z = z^*$ and $v = v^*$. Assume that $c > 0$ contacts have occurred between links of the manipulator and obstacles. In that case, contacts define gap functions $g_i(z, v)$, $i = 1, \dots, c$, which will be negative because obstacles are penetrated. The objective is to find a value of v that renders the gap functions non-negative. This can be formulated as the following nonlinear system:

$$\begin{aligned} g_1(z, v) &= k_1^2 \\ \dots \\ g_c(z, v) &= k_c^2 \end{aligned} \quad (20)$$

where $k = [k_1, \dots, k_c]^T$ is a vector of auxiliary variables that are introduced to transform the desired inequalities $g_i(z, v) \geq 0$ into equivalent equalities $g_i(z, v) = k_i^2$. Next, Equation

(20) is solved for $r = [v^T, k^T]^T$, using the Newton–Raphson (N-R) method [26], where the starting values of v and k for N-R iteration are $v = v^*$, i.e., the value that produced the interference that led to this step of the algorithm, and $k = 0$. Since Equation (20) is c equations in $(c + n - m)$ unknowns, it comprises an underdetermined system. Therefore, when inverting the Jacobian used in the N-R method, the minimum-norm Moore–Penrose pseudoinverse is used, which updates the unknowns r with the vector Δr of minimum norm. The N-R method used to solve Equation (20) is described by Algorithm 1 shown below, where

$$\tilde{g}(r) = [g_1(z, v) - k_1^2, \dots, g_c(z, v) - k_c^2]^T \quad (21)$$

and $\tilde{g}'(r)$ is the $c \times (c + n - m)$ Jacobian of $\tilde{g}(r)$ with respect to r .

Algorithm 1. Newton–Raphson iterations to solve Equation (20)

```

1:       $r \leftarrow [(v^*)^T, 0]^T$ 
2:      iterations  $\leftarrow 0$ 
3:      do
4:           $\Delta r \leftarrow -\tilde{g}'^{\dagger}(r)\tilde{g}(r)$ 
5:           $r \leftarrow r + \Delta r$ 
6:          iterations  $\leftarrow$  iterations + 1
7:      while  $\|\tilde{g}(r)\| > \varepsilon$  AND iterations  $<$  max_iterations

```

Since the increment Δr with minimum norm is used in lines 4–5 of Algorithm 1 to update the unknowns, this favors continuity of the configuration of the manipulator. In fact, as the examples of Section 4 demonstrate, the proposed algorithm generates continuous trajectories for the manipulator, and execution of Algorithm 1 (whose objective is to find a new value of self-motion coordinates v that avoids collisions) takes only a few milliseconds on a modern computer. The N-R method stops when a solution $r = [v^T, k^T]^T$ that satisfies Equation (20) is found, or when a maximum of max_iterations is exceeded. In the first case, the algorithm yields a configuration y that is collision-free, and the algorithm jumps to step 4. In the second case, the desired trajectory $z_d(t)$ can be considered as infeasible, because no configuration (sufficiently near to the configuration at the previous time step) can be found to continue executing the trajectory while avoiding obstacles, and the algorithm ends.

4. Numerical Examples

The foregoing theory and computational algorithms are tested in this section using planar examples with four and twenty degrees of redundancy. The selection of planar examples is for ease of analysis and visualization of results. The theory and algorithm of Section 3.3 is equally applicable for spatial and planar manipulators. For an application of the differentiable manifold formulation with a seven degree of freedom spatial redundant serial manipulator, see [3].

4.1. Four-Degree-of-Redundancy Trajectory Tracking and Obstacle Avoidance

4.1.1. Differentiable-Manifold Output Tracking and Obstacle Avoidance

In this section, the method described in Section 3.3 is applied to plan the motion of the manipulator of Section 1.2 that has four degrees of redundancy. The manipulator starts with input $y_0 = [0.5688, 0.4694, 0.0119, 0.3371, 0.1622, 0.7943]^T$, and its end-effector is required to follow the periodic elliptic trajectory

$$z_d(t) = [a \cos(\theta) \cos(t) - b \sin(\theta) \sin(t) + x_0, a \sin(\theta) \cos(t) + b \cos(\theta) \sin(t) + y_0]^T \quad (22)$$

while its links must avoid a circular obstacle centered at (1.5, 2) with radius 0.5. Equation (22) is the parametric equation of an ellipse centered at $(x_0 = 1.7, y_0 = 2.9)$ with semiaxes $a = 2$ and $b = 0.25$, whose longer semiaxis forms an angle of $\theta = -0.15$ rad with the x axis. The end-effector is required to follow three cycles on this ellipse, i.e., t runs from $t = 0$ to $t = 6\pi$, with a time step of $\Delta t = 0.01$ s.

The algorithm described in Section 3.3 is carried out with the following parameters:

- (1) Iterations in the method presented in [4] to evaluate $\mathbf{u} = h^i(z, v)$ stop when $\|\Delta \mathbf{u}\| < 0.0001$.
- (2) Iterations in Algorithm 1 stop when $\|\tilde{g}(r)\| < 0.001$ or when iterations $> 10 = \text{max_iterations}$.

With these parameters, the algorithm generates the time histories of $\mathbf{y}(t)$ and $v(t)$ shown in Figure 4. As this figure shows, the time history of these variables is periodic after the first cycle, where vertical dashed lines separate cycles. In particular, v is constant after the first cycle. It can be seen in Figure 4 that the trajectory of $\mathbf{y}(t)$ becomes cyclic when v becomes constant, even before the output trajectory of Equation (22) has completed its first cycle. This locally cyclic behavior is consistent with the theory presented in Section 3.1.

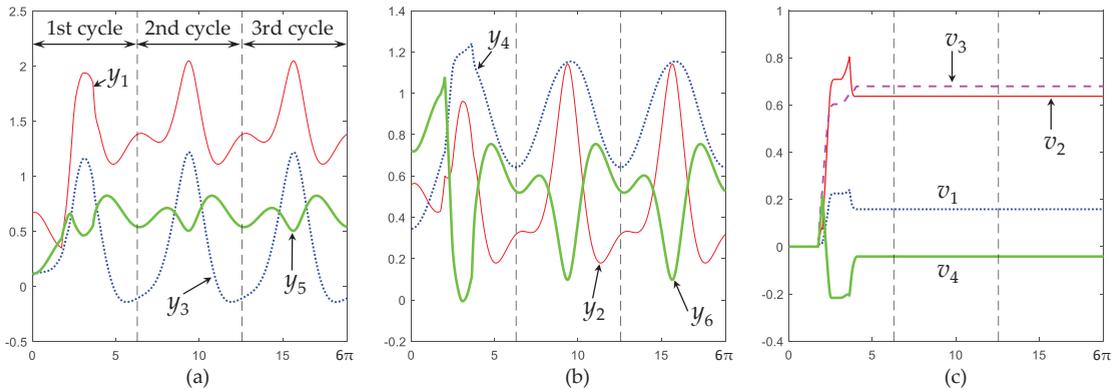


Figure 4. (a–c) Time history of $\mathbf{y}(t)$ and $v(t)$ becomes cyclic after the first cycle, when the end-effector follows Equation (22) while avoiding an obstacle using the obstacle avoidance algorithm of Section 3.3.

Snapshots of the manipulator executing this trajectory are shown in Figure 5. An animation is shown in supplementary video S1 attached. Figure 5a–i show execution of the first cycle. In Figure 5a, the manipulator moves with $v = 0$ as it approaches the obstacle, until touching it. Then, the algorithm of Section 3.3 starts to actively update $v(t)$ to avoid penetration of the obstacle. Figure 5b–g show how the manipulator continues tracking the desired elliptical trajectory, while some of its links touch the circular obstacle, until contact is lost in Figure 5h and v is no longer updated. After this, the manipulator continues executing the trajectory far from the obstacle until it completes the first cycle, as shown in Figure 5i.

Execution of the second cycle begins with the manipulator approaching the obstacle, as illustrated in Figure 5j. As shown in Figure 5k–n, during the second cycle, the manipulator is not in contact with the obstacle except during a very short period of time shown in Figure 5m, in which one of the links becomes tangent to the obstacle. This contact period is very short, as can be better observed in the supplementary animation S1 attached to the paper. After this short contact is lost (Figure 5n), the manipulator continues moving far from the obstacle until completing the second cycle (Figure 5o). The third and subsequent cycles are identical to the second cycle, as shown in Figure 4.

As Figures 4 and 5 show, during the first cycle of the desired trajectory given by Equation (22), the algorithm of Section 3.3 actively updates $v(t)$ to prevent penetration of the obstacle during contact. After contact is lost, $v(t)$ does not need to be further updated and the input trajectory becomes cyclic, by virtue of Equation (17). Interestingly, the constant value that $v(t)$ adopts after contact is lost in the first cycle allows the manipulator to continue repeating subsequent cycles during which contact without penetration occurs only during a very short fraction of the cycle, as illustrated in Figure 5m.

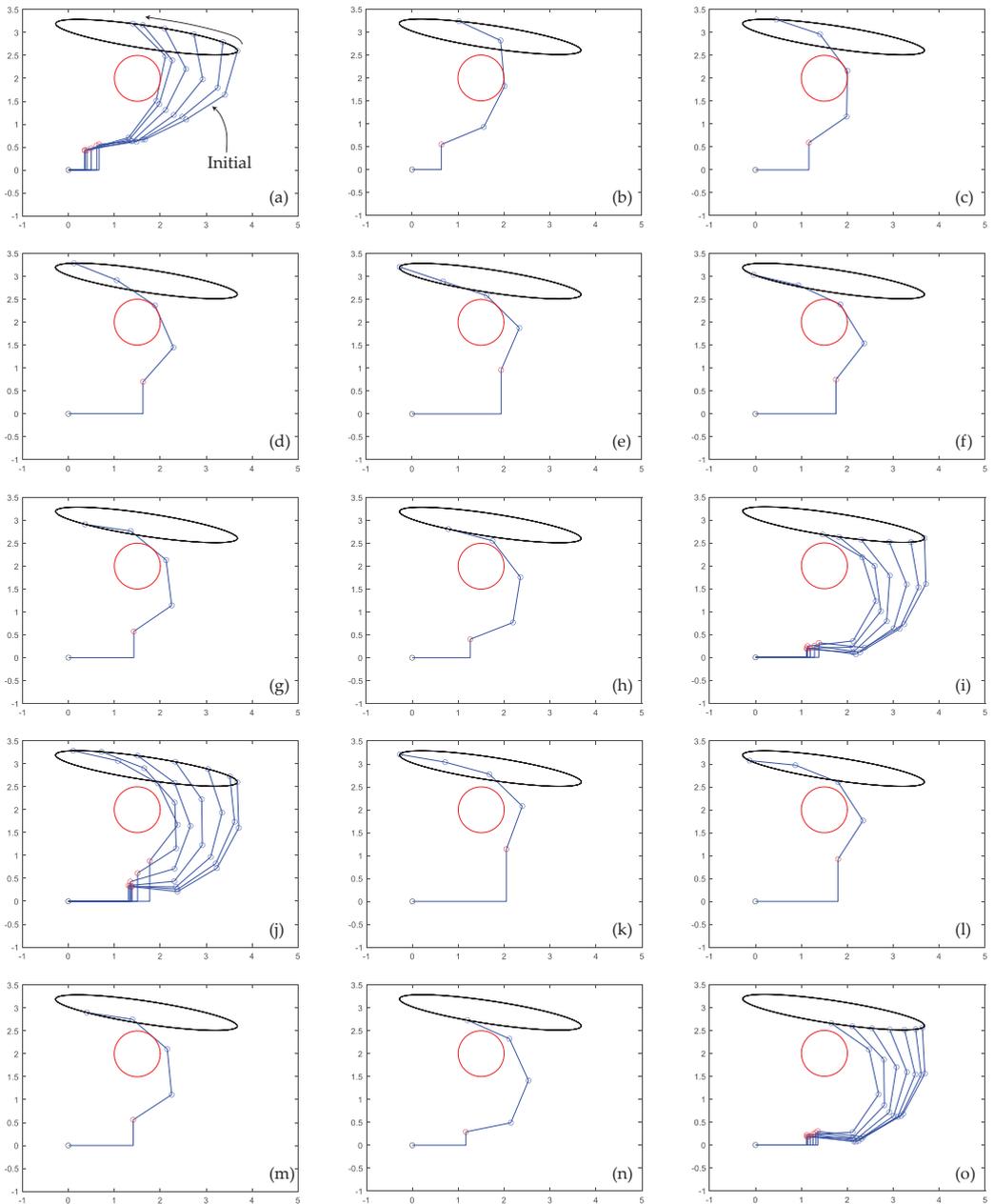


Figure 5. (a–i) Execution of the first cycle of the desired output trajectory given by Equation (22). (j–o) Execution of the second cycle. Supplementary video S1 attached shows this figure in motion, including a third cycle that is identical to the second one.

The average time required to run Algorithm 1 is 1.2 milliseconds (time measured on the following CPU running MATLAB R2022b: Intel Core i7-8750H at 2.20 GHz). This is the time required to find a new value of self-motion coordinates v that avoids penetration

of obstacles every time that such penetration is detected, which demonstrates a rapid correction of the trajectory to avoid obstacles, compatible with real time requirements.

As shown in Figure 5, to prevent obstacle penetration, the algorithm of Section 3.3 keeps the links in contact with the obstacle and moving smoothly around it until contact is lost, as required by the trajectory of the end-effector. If it were necessary to avoid the obstacle without contacting it, leaving some safety distance d_s , then one would only need to change the gap functions in Equation (20) to $g_i(z, v) - d_s = k_i^2$, making it easier to accurately regulate distance to obstacles than with the generalized-inverse-velocity-based approach, which is illustrated next.

4.1.2. Generalized-Inverse-Velocity Output Tracking and Obstacle Avoidance

In this section, the approach described in Section 2 is used to solve the problem treated in Section 4.1.1, where the manipulator with four degrees of redundancy must track the periodic trajectory given in Equation (22). The manipulator starts at the same configuration y_0 as in Section 4.1.1. The same time step, $\Delta t = 0.01$ seconds, is used in the present section to numerically integrate Equation (9) using a fourth-order Runge–Kutta integrator [26]. In contrast to the generalized-inverse-velocity-based method, the algorithm of Section 3.3 does not need to perform any numerical integration. The time step in that case is used only to progress in time according to step 4 of the algorithm.

The method described In Section 2 is carried out by numerically integrating with a fixed time step of $\Delta t = 0.01$ s. For a given time step Δt , the main parameter that can be tuned in the method of Section 2 is the coefficient $k > 0$, which adjusts the intensity of the penalty function $f(y)$. A higher k will give greater weight to minimization of the cost function, which means that the manipulator links will tend to have larger distances from the obstacles.

Figure 6 presents a sequence of configurations adopted by the manipulator for three different values of k when executing the first half of the first cycle of the periodic trajectory given by Equation (22), i.e., when integrating Equation (9) from $t = 0$ to $t = \pi$ seconds. According to Figure 6c, for higher values of k , the manipulator adopts configurations that are unnecessarily far from the obstacle. Thus, in the following, the value $k = 0.01$, which generates the result shown in Figure 6a, is used. Note, however, that if k is chosen too small, the rate of minimization of $f(y)$ may be too slow for the motion of the robot, and one or more links may penetrate the obstacle before the term $\dot{y}^0(t)$ of Equation (10) leads to a trajectory that avoids the obstacle.

When the manipulator is required to transit the elliptic trajectory three times by integrating Equation (9) with $k = 0.01$ from $t = 0$ to $t = 6\pi$, the time evolution of inputs $y_i(t)$ presented in Figure 7 is obtained. As this figure shows, the time history of $y_i(t)$ is not cyclic, because the value of each $y_i(t)$ at the end of each cycle of the output trajectory of Equation (22) does not coincide with its value at the beginning. This is an example of noncyclicity of the generalized inverse velocity approach to redundancy resolution, as discussed in Section 1.4. In contrast, as shown in Section 4.1.1, the differentiable manifold approach leads to a locally cyclic trajectory after nonzero control action is required to avoid the obstacle.

To better illustrate this noncyclic trajectory, Figure 8 presents a sequence of snapshots of the manipulator as its inputs $y_i(t)$ traverse the trajectories graphed in Figure 7. Only the first two cycles are presented in Figure 8. Supplementary video S2 attached displays the animation of these two cycles, as well as a third cycle.

Figure 8a–i show execution of the first cycle. In Figure 8a, the manipulator is approaching the obstacle. Between Figure 8b,e, the gradient term of Equation (10) acts to avoid the obstacle. Between Figure 8f,i, the links of the manipulator are sufficiently far from the obstacle, and the cost f of Equation (11) is negligible (with negligible derivative), so the corrective term of Equation (10) does not have significant effect. Figure 8i shows the configuration of the manipulator at the end of the first cycle of the elliptic trajectory, which does not coincide with the initial configuration shown in Figure 8a.

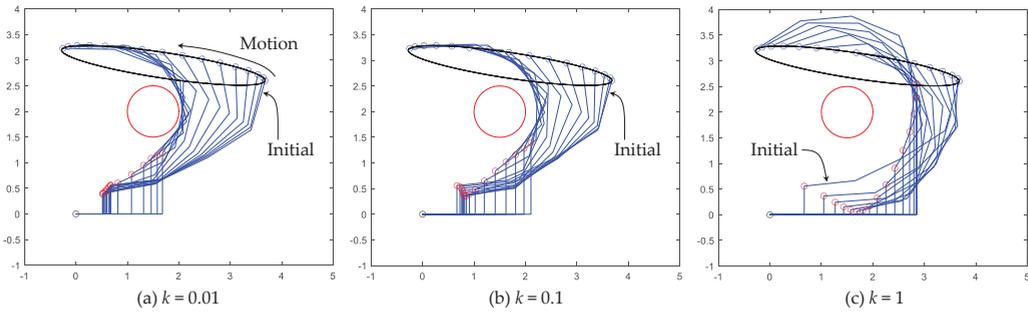


Figure 6. (a–c) Comparison of the postures adopted by the manipulator when integrating Equation (9) using Equation (10) for different values of k while describing half of the ellipse defined by Equation (22).

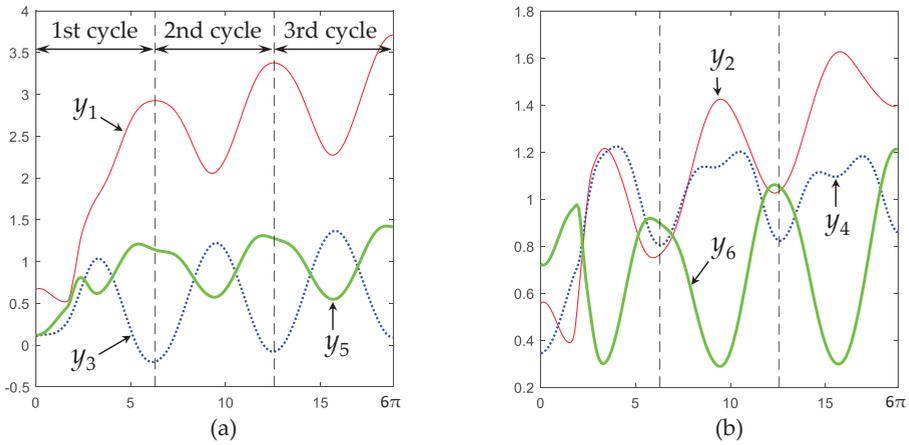


Figure 7. (a,b) Time evolution of $y_i(t)$ when tracking the trajectory of Equation (22) with $k = 0.01$.

Figure 8j–o show a sequence of snapshots of the configuration of the manipulator during execution of the second cycle, completing the second cycle in Figure 8o. When comparing these snapshots to those of the first cycle, one observes the noncyclicity of the input trajectory, where the manipulator is further away from the obstacle than in Figure 8a,i. A third cycle can be observed in attached video S2, which ends with the manipulator even further away from the obstacle.

As this example shows, generalized-inverse-velocity-based kinematic control can be used for obstacle avoidance, but its performance is not easy to tune or predict. For this example, a time step of $\Delta t = 0.01$ seconds was sufficient to achieve obstacle avoidance for different values of k , and it was possible to avoid obstacles without excessively conservative distances between links and obstacles by choosing a small value of $k = 0.01$. However, this worked well only for the first cycle. After completing the first cycle, due to the nonholonomic nature of Equation (9), the manipulator drifts and executes subsequent cycles far from the obstacle, where $\dot{y}^0(t)$ of Equation (10) becomes negligible, yielding postures similar to those that would have been adopted if the first cycle had been executed using a larger value of k , as demonstrated in Figure 6. This suggests that the choice of parameters k and Δt (which have been kept constant throughout this example) may actually be rather arbitrary, because nonholonomy may end up wasting or undoing a careful choice of such parameters. This is even more important in difficult scenarios such as the one treated in Section 4.2.3, where a highly redundant manipulator must avoid several obstacles,

and choice of parameters k and Δt is critical for performance of generalized-inverse-velocity-based kinematic control.

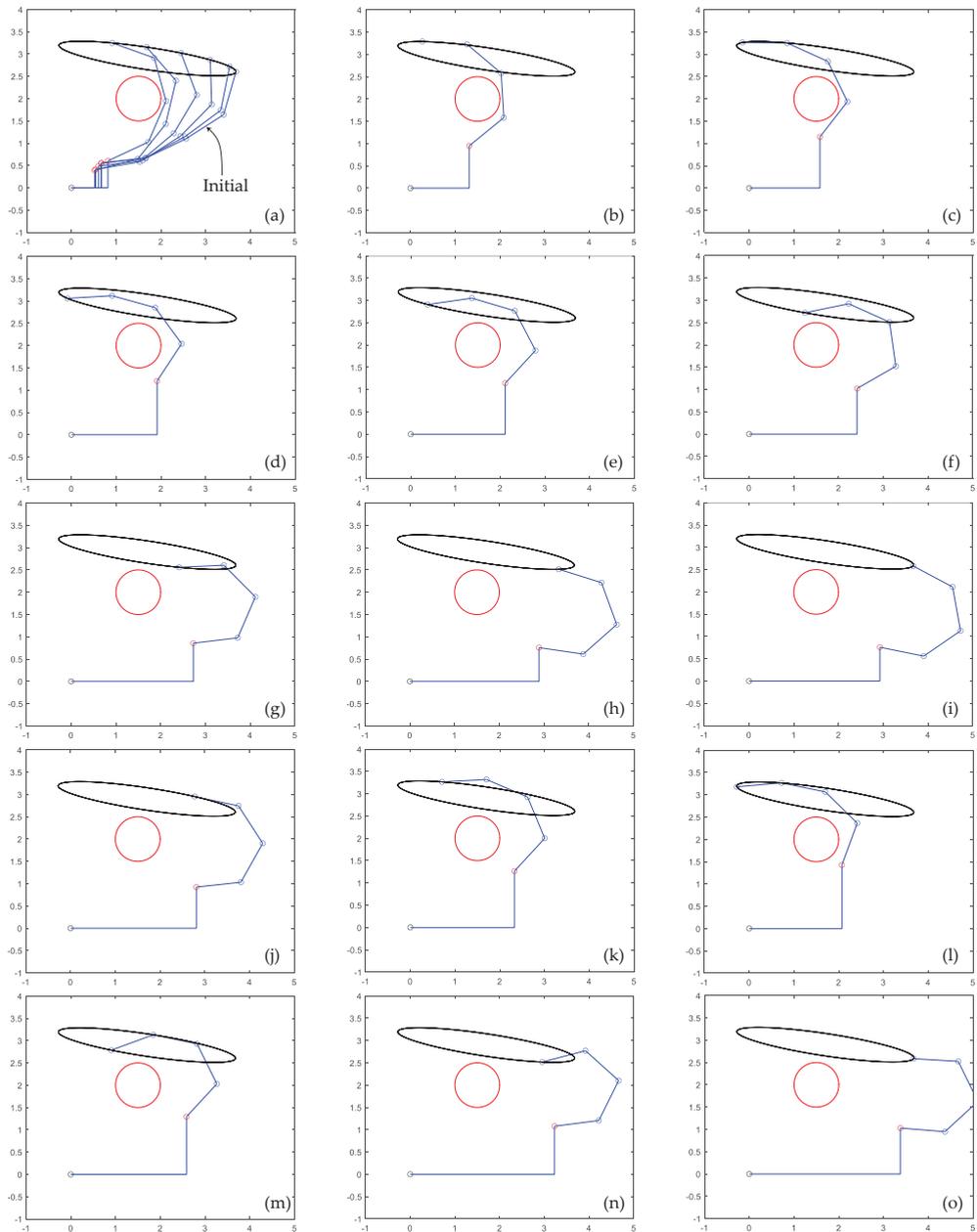


Figure 8. (a–i) Execution of the first cycle of the desired output trajectory given by Equation (22), for $k = 0.01$. (j–o) Execution of the second cycle. Supplementary video S2 attached shows this figure in motion, including a third cycle that is different from the previous two cycles.

4.2. Twenty-Degree-of-Redundancy Trajectory Tracking and Obstacle Avoidance

4.2.1. A Twenty-Degree-of-Redundancy Manipulator

Consider a highly redundant variant of the manipulator of Figure 1 that is shown in Figure 9. Instead of having four links with relative angles $y_3 \dots y_6$, it has 21 links with relative angles $y_3 \dots y_{23}$, where all links have unit length.

The initial configuration of the manipulator is shown in Figure 10, given by the following input angles.

$$y_0 = \begin{bmatrix} 0.7727 & 0.1546 & 0.1040 & 0.4934 & 0.9715 & 0.3697 & 0.6284 & \dots \\ \dots & 0.2710 & 0.7981 & 0.2884 & 0.5227 & 0.6964 & 0.8729 & 0.9409 & \dots \\ \dots & 0.5358 & 0.1209 & 0.1221 & 0.2128 & 0.7854 & 0.2131 & 0.7936 & 0.2532 & 0.9685 \end{bmatrix}^T \quad (23)$$

This manipulator has $n = 23$ degrees of freedom. Outputs of the manipulator are the position coordinates of the outboard end of the last link and its orientation, i.e.,

$$z = G(y) = \begin{bmatrix} y_1 + \sum_{i=3}^{23} \cos\left(\sum_{j=3}^i y_j\right) \\ y_2 + \sum_{i=3}^{23} \sin\left(\sum_{j=3}^i y_j\right) \\ \sum_{i=3}^{23} y_i \end{bmatrix} \quad (24)$$

Since there are $m = 3$ outputs, the degree of redundancy is $r = n - m = 23 - 3 = 20$, so this can be considered a *hyper-redundant manipulator*.

The task of this manipulator is to follow the output trajectory

$$z_d(t) = [-2.1164 + 2.5(\cos(4t) - 1), 4.0455 + \sin(3t), -1.6035 \text{ rad}]^T \quad (25)$$

which means that the outboard endpoint of the manipulator must describe the desired path given by the first two coordinates of Equation (25), and the last link maintains a constant orientation given by the last coordinate of Equation (25). The desired path is displayed in Figure 10. The desired trajectory must be achieved from $t = 0$ to $t = 1.2$, while the manipulator avoids three circular obstacles that are shown in Figure 10. Two obstacles are stationary, labeled O1 and O2 in Figure 10, where O1 is a circle centered at (0, 2) with unit radius, and O2 is another circle of unit radius centered at (-2, 7). Note that, initially, the manipulator wraps around obstacle O1. There is a third circular obstacle labeled Om in Figure 10 with radius 0.75 that is mobile. Its center starts at (-4, 2) and translates with constant speed until reaching the point (-3, 5.5) at $t = 1$, remaining static at that point for $t > 1$.

4.2.2. Differentiable-Manifold Output Tracking and Obstacle Avoidance

The path-planning obstacle avoidance problem is first treated using the differentiable manifold algorithm of Section 3.3 with a time step of $\Delta t = 0.001$ s, which results in the sequence of postures shown in Figure 11 (supplementary video S3 is attached that illustrates the continuous motion). As demonstrated by Figure 11, the algorithm is able to resolve redundancy and obtain a feasible motion that tracks the desired output trajectory of Equation (25), while avoiding penetration of all obstacles for all time. First, the manipulator contacts obstacle O1 (Figure 11b), and then it contacts obstacle O2 (Figure 11c–d), to lose contact later (Figure 11e–h). In Figure 11i, the manipulator contacts the mobile obstacle Om and subsequently adjusts its configuration continuously to avoid penetrating this mobile obstacle as it approaches stationary obstacle O2 (Figure 11i–l). A difficult situation occurs, as shown in Figure 11m, when the mobile obstacle Om stops close to obstacle O2, after which Om and O2 create a narrow corridor through which the manipulator must maneuver (see the magnified view of the corridor in the inset of Figure 11m). The differentiable

manifold algorithm of Section 3.3 is able to handle this situation seamlessly. It completes the desired trajectory, as demonstrated in Figure 11m–o.

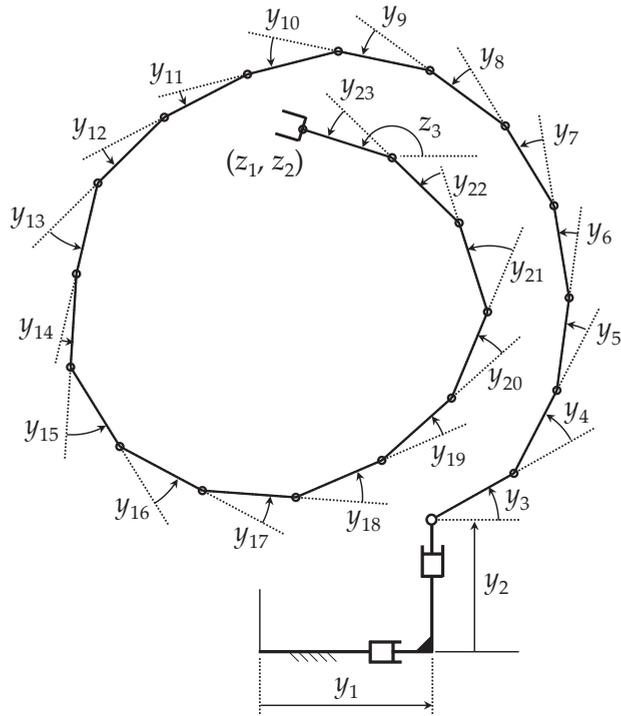


Figure 9. A manipulator with $r = 20$ degrees of redundancy.

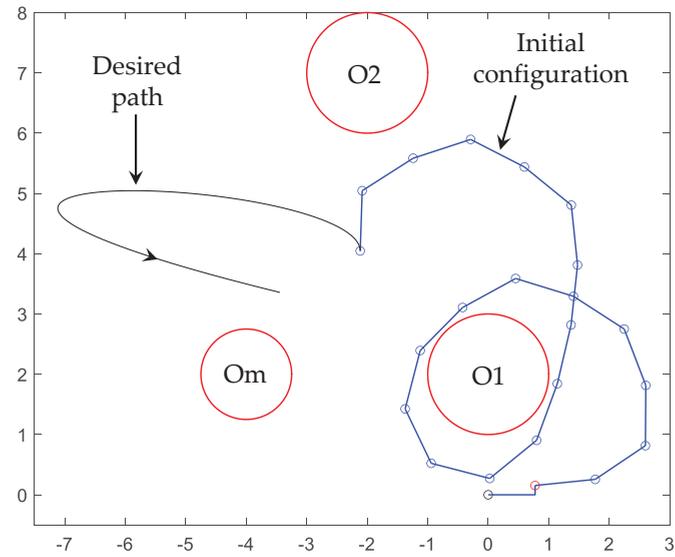


Figure 10. Obstacles and manipulator initial configuration.

4.2.3. Generalized-Inverse-Velocity Output Tracking and Obstacle Avoidance

The problem treated in Section 4.2.2 is attempted using the generalized inverse velocity method, numerically integrating Equation (9) with the term of Equation (10) to avoid obstacles. For this method, it is necessary to select two parameters, the time step Δt used to numerically integrate Equation (9) (using again the fourth-order Runge–Kutta method) and the coefficient k used in Equation (10) to tune the rate of minimization of the cost function that penalizes proximity to obstacles. For a less complex example such as the one studied in Section 4.1.2, these two parameters were successfully set. However, for this more complex example with three obstacles (one of them mobile) and, more importantly, a narrow corridor such as the one shown in Figure 11m, the choice of parameters becomes critical. A smaller time step Δt allows accurate numerical integration and smooth motions, but the simulation takes more CPU time to complete the full trajectory (or, in case the method is used to update the configuration in a real-time controller, it requires a more demanding controller that is able to operate at smaller sample times Δt). A smaller k performs the minimization of the cost function more slowly, which yields smoother motions that come closer to obstacles. However, a small value of k requires using a very small time step Δt to enable the algorithm to avoid penetrating obstacles.

Table 1 shows 16 combinations of $(k, \Delta t)$ and the result of obstacle avoidance for each combination. The result of each combination is represented by a grade F, D, or C, with the following meaning:

Grade F means that the simulation fails when the first obstacles (O1 or O2) are contacted, long before the narrow corridor indicated in Figure 11m is formed. The failure consists in the manipulator suffering a discontinuity in its configuration upon contact.

Grade D means that the simulation fails when the manipulator is trapped by the narrow corridor. The failure consists in the manipulator suffering a discontinuity in its configuration when the corridor is formed, without penetrating obstacles.

Grade C means that, when the narrow corridor is formed, the manipulator starts to penetrate obstacles and suffers discontinuous jumps in its configuration.

Table 1. Results of the generalized inverse velocity method for different pairs $(k, \Delta t)$.

$k \setminus \Delta t$ (Seconds)	0.01	0.001	0.0001	0.00001 (Very Slow Simulation)
0.1	F	D	D	D
0.01	F	D	D	D
0.001	F	C	D	C
0.0001	F	F	D	C

The lower-right case of Table 1 ($k = 0.0001$ and $\Delta t = 0.00001$) is illustrated in Figure 12. This case has been chosen for illustration because, despite not being fully successful (in fact, no case of Table 1 is fully successful), it exhibits the most similar behavior to Figure 11. It should be noted that the result of Figure 12 required a very slow simulation, due to the small time step of 0.00001 s, in contrast to the simulation of Figure 11 that worked well with a time step of 0.001 s. As Figure 12 shows, the generalized-inverse-velocity-based method yields a solution that is not very different from the one shown in Figure 11 during the beginning of the trajectory. The links contact obstacles and wrap around them as the manipulator progresses, including the mobile obstacle O_m . The problem occurs when the mobile obstacle forms the narrow corridor with static obstacle O2. In that case, as Figure 12m shows, the manipulator starts to penetrate obstacles. It also suffers discontinuities in its configuration that occur between Figure 12l and m. This behavior can be better observed in supplementary video S4 attached. In any case, the trajectory becomes infeasible. In fact, according to Table 1, no valid combination of $(k, \Delta t)$ was found that accomplishes this trajectory while avoiding obstacles with the generalized inverse velocity method. In

contrast, the differentiable manifold method of Section 3.3 was able to easily solve this problem, as demonstrated in Figure 11 and Supplementary video S3.

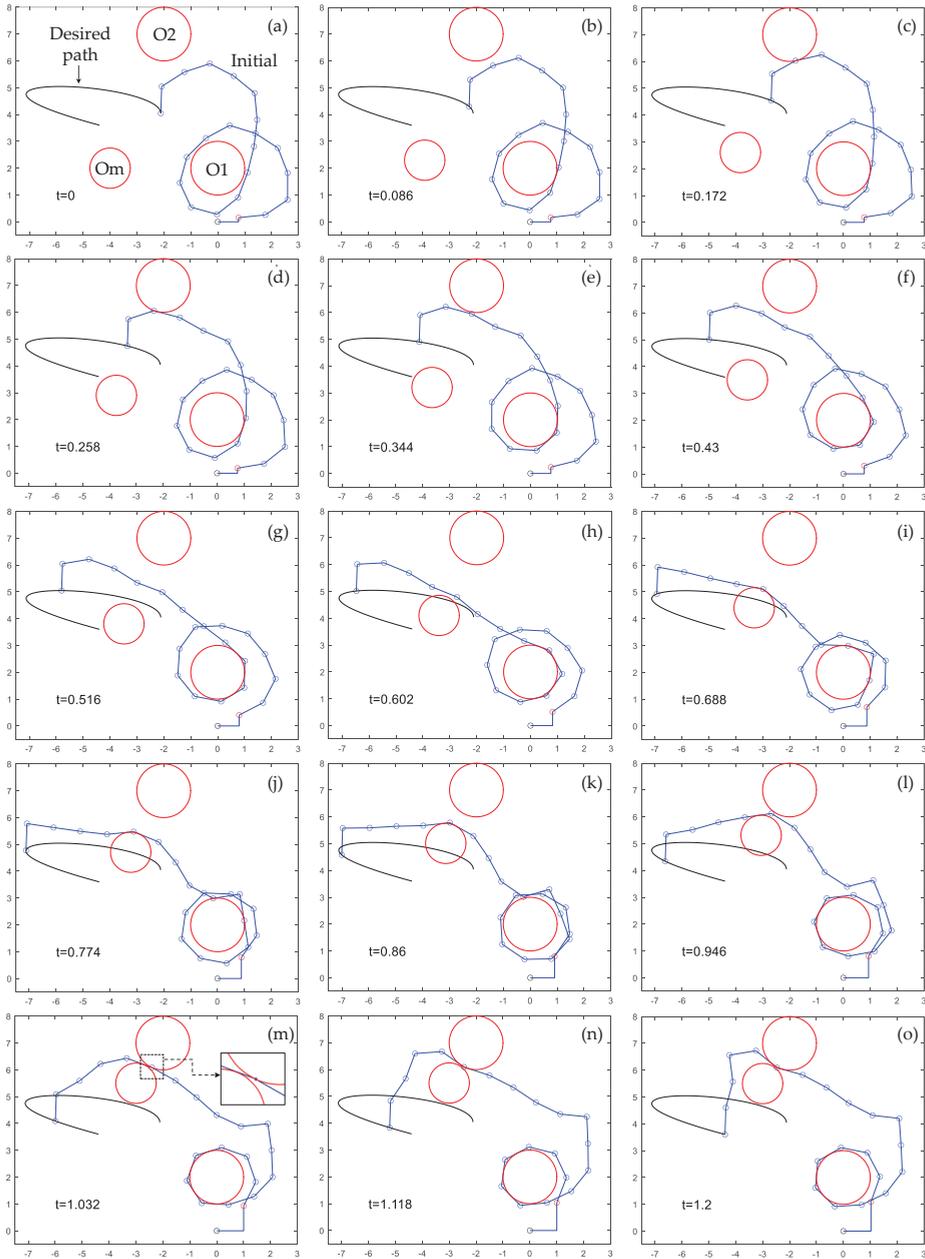


Figure 11. (a–o) Successful redundancy resolution in a highly redundant manipulator with obstacle avoidance, using the differentiable manifold algorithm of Section 3.3. Supplementary video S3 of this sequence is attached.

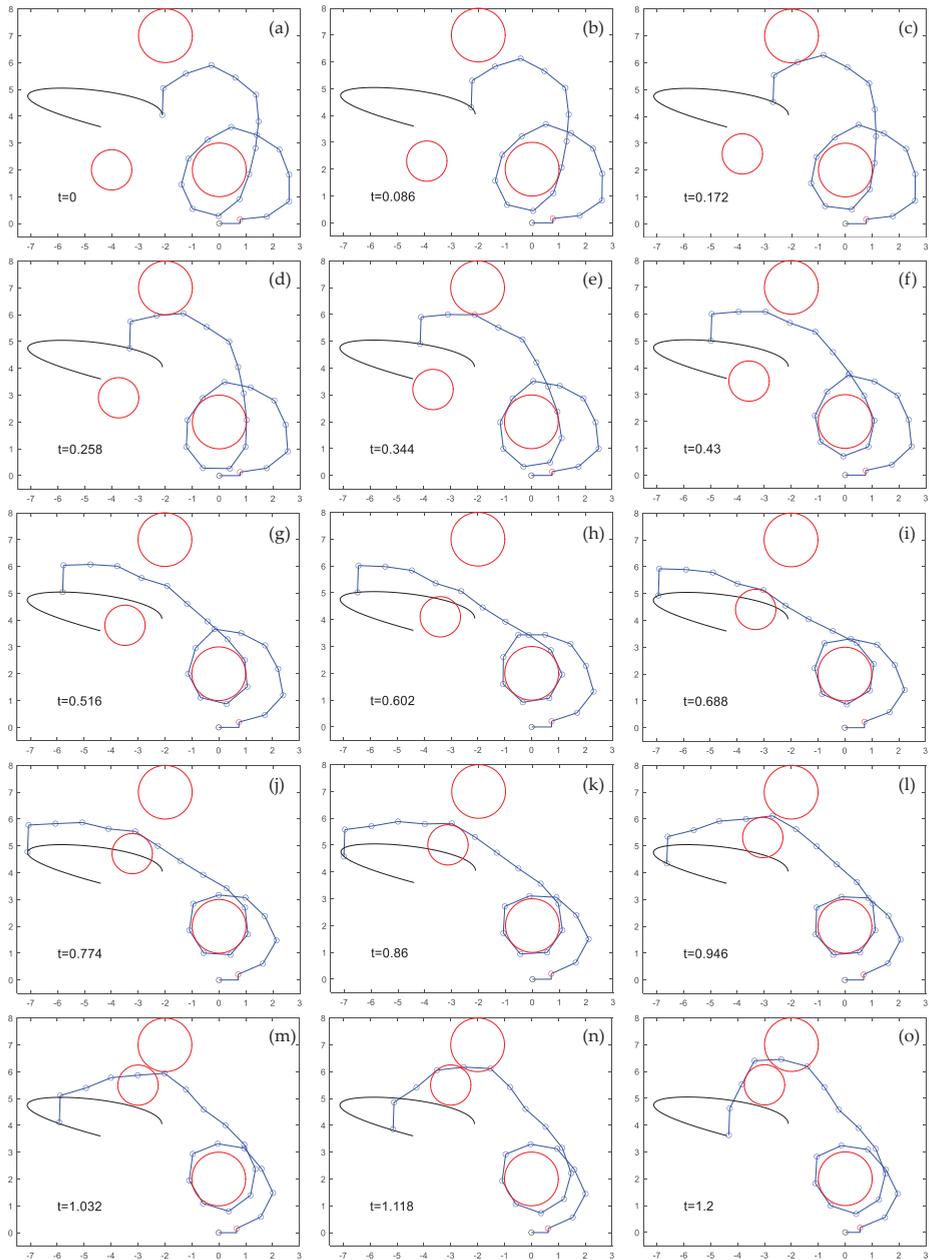


Figure 12. (a–o) Unsuccessful redundancy resolution in a highly redundant manipulator with obstacle avoidance, using the generalized inverse velocity method of Equation (9). Supplementary video S4 of this sequence is attached.

5. Discussion

Kinematic control of redundant manipulators has traditionally been conducted at the velocity level, where a generalized inverse is used with the velocity equation of Equation (9) to obtain joint velocities \dot{j} in terms of the task (or output) velocities \dot{z} and arbitrary

input velocities. Joint trajectories are then determined by numerical integration of \dot{y} . Obstacle avoidance is traditionally treated by performing self-motions that approximately minimize a cost function of Equation (11) that penalizes proximity to obstacles. As shown in Section 4.1.2, this approach leads to difficult-to-predict behavior, due to nonholonomy, as the manipulator performs quite different postures between cycles of specified output trajectories for the four-degree-of-redundancy manipulator, despite fine-tuning the parameters of the method (the rate of minimization of the cost function) to avoid obstacles during the first cycle. As shown in Section 4.2.3, regardless of the values of tunable parameters, the generalized inverse velocity approach encounters insurmountable difficulties for the twenty-degree-of-redundancy manipulator in a complicated situation such as a narrow corridor between obstacles, leading to infeasible trajectories that exhibit discontinuities or penetration of obstacles. These examples show that theoretical deficiencies of the generalized-inverse-velocity-based redundancy resolution method identified in Section 1.4 lead to severe problems in manipulator control applications.

An operational configuration space differentiable manifold formulation is presented for kinematic control of redundant manipulators during obstacle avoidance. In contrast with the traditional generalized-inverse-velocity-based approach, the differentiable manifold formulation resolves kinematic redundancy at the configuration level, constructing an inverse mapping that parameterizes input coordinates y as functions of output coordinates z and self-motion coordinates v . This mapping is holonomic, locally cyclic, and can be used to generate global obstacle-free motion plans, as described in Section 3.3. The algorithm presented defines a gap function g in the vicinity of every collision that is a function of self-motion coordinates v . The algorithm actively adjusts self-motion coordinates v to avoid penetration of obstacles, keeping g non-negative by solving an under-constrained system of nonlinear equations. The example of Section 4.1.1 demonstrates the cyclicity and real-time capability of the algorithm with the four-degree-of-redundancy manipulator. The example of Section 4.2.2 demonstrates its ability to treat very restrictive obstacle situations with the twenty-degree-of-redundancy manipulator, negotiating a narrow corridor while accurately completing the desired task.

Although this paper has focused on redundant serial manipulators, the differentiable manifold formulation is presented for non-serial redundant manipulators in [3], which makes it possible to extend the algorithm presented to non-serial manipulators. Future applications of the formulation presented will not only consider control at the kinematics level but also control at the dynamics level [4], where accurate holonomic control of obstacle avoidance can be exploited for controlling contact forces between links and obstacles. Kinematic control applications involving constraints on velocities and accelerations can also be addressed.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/machines12010010/s1>: Video S1: Figure 5.mp4 (video of Figure 5); Video S2: Figure 8.mp4 (video of Figure 8); Video S3: Figure 11.mp4 (video of Figure 11); Video S4: Figure 12.mp4 (video of Figure 12).

Author Contributions: Conceptualization, E.J.H.; methodology, E.J.H. and A.P.; software, A.P.; validation, E.J.H. and A.P.; data curation, A.P.; writing—original draft preparation, E.J.H.; writing, review and editing, E.J.H. and A.P.; visualization, A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Pars, L.A. *A Treatise on Analytical Dynamics*, 1965; Reprint by Ox Bow Press: Woodbridge, UK, 1979.
2. Robbin, J.W.; Salamon, D.A. *Introduction to Differential Geometry*; Springer: Berlin/Heidelberg, Germany, 2022.

3. Haug, E.J.; Peidro, A. Redundant Manipulator Kinematics and Dynamics on Differentiable Manifolds. *J. Comput. Nonlinear Dyn.* **2022**, *17*, 111008. [CrossRef]
4. Haug, E.J. Redundant Serial Manipulator Inverse Position Kinematics and Dynamics. *J. Mech. Robot.* **2024**, *16*, 081008. [CrossRef]
5. Haack, W.; Wendland, W. *Lectures on Partial and Pfaffian Differential Equations*; Pergamon Press: Oxford, UK, 1972.
6. Whitney, D.E. Resolved Motion Rate Control of Manipulators and Human Prostheses. *IEEE Trans. Man-Mach. Systems.* **1969**, *10*, 47–53. [CrossRef]
7. Siciliano, B. Kinematic Control of Redundant Robot Manipulators: A Tutorial. *J. Intell. Robot. Systems.* **1990**, *3*, 201–212. [CrossRef]
8. Chilverini, S.; Oriolo, G.; Maciejewski, A.A. *Springer Handbook of Robotics*, 2nd ed.; Robots, R., Siciliano, B., Khatib, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2016.
9. Khatib, O. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *IEEE J. Robot. Automation.* **1987**, *RA-3*, 43–53. [CrossRef]
10. Khatib, O. The Operational Space Framework. *JSME Int. J. Ser. C.* **1993**, *36*, 277–287. [CrossRef]
11. Klein, C.A.; Huang, C.-H. Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators. *IEEE Trans. Syst. Man Cybern.* **1983**, *SMC-13*, 245–250. [CrossRef]
12. De Luca, A.; Oriolo, G. Nonholonomic Behavior in Redundant Robots Under Kinematic Control. *IEEE Trans. Robot. Autom.* **1997**, *13*, 776–782. [CrossRef]
13. Shamir, T.; Yomdin, Y. Repeatability of Redundant Manipulators: Mathematical Solution of the Problem. *IEEE Trans. Autom. Control* **1988**, *33*, 1004–1009. [CrossRef]
14. Simas, H.; Di Gregorio, R. A Technique Based on Adaptive Extended Jacobians for Improving the Robustness of the Inverse Numerical Kinematics of Redundant Robots. *J. Mech. Robot.* **2010**, *11*, 020913. [CrossRef]
15. Mussa-Ivaldi, F.A.; Hogan, N. Integrable Solutions of Kinematic Redundancy via Impedance Control. *Int. J. Robot. Res.* **1991**, *10*, 481–491. [CrossRef]
16. Haug, E.J. A Cyclic Differentiable Manifold Representation of Redundant Manipulator Kinematics. *J. Mech. Robot.* **2024**, *16*, 061005. [CrossRef]
17. Liegeois, A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. Syst. Man Cybern.* **1977**, *7*, 868–871.
18. Lee, K.K.; Buss, M. Obstacle avoidance for redundant robots using Jacobian transpose method. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 3509–3514.
19. Maciejewski, A.A.; Klein, C.A. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *Int. J. Robot. Res.* **1985**, *4*, 109–117. [CrossRef]
20. Slotine, S.B.; Siciliano, B. A general framework for managing multiple tasks in highly redundant robotic systems. In Proceedings of the 5th International Conference on Advanced Robotics, Pisa, Italy, 19–22 June 1991; Volume 2, pp. 1211–1216.
21. Flacco, F.; De Luca, A.; Khatib, O. Control of redundant robots under hard joint constraints: Saturation in the null space. *IEEE Trans. Robot.* **2015**, *31*, 637–654. [CrossRef]
22. Flacco, F.; De Luca, A.; Khatib, O. Motion control of redundant robots under joint constraints: Saturation in the null space. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 285–292.
23. Kazemipour, A.; Khatib, M.; Al Khudir, K.; Gaz, C.; De Luca, A. Kinematic control of redundant robots with online handling of variable generalized hard constraints. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9279–9286. [CrossRef]
24. Ziese, A.; Fiore, M.D.; Peters, J.; Zimmermann, U.E.; Adamy, J. Redundancy resolution under hard joint constraints: A generalized approach to rank updates. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 7447–7453.
25. Fiore, M.D.; Meli, G.; Ziese, A.; Siciliano, B.; Natale, C. A general framework for hierarchical redundancy resolution under arbitrary constraints. *IEEE Trans. Robot.* **2023**, *39*, 2468–2487. [CrossRef]
26. Atkinson, K.E. *An Introduction to Numerical Analysis*, 2nd ed.; Wiley: New York, NY, USA, 1989.
27. Corwin, L.J.; Szczarba, R.H. *Multivariable Calculus*; Marcel Dekker: New York, NY, USA, 1982.
28. Haug, E.J. *Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume II: Modern Methods*, 3rd ed.; ResearchGate: Berlin, Germany, 2022; Available online: www.researchgate.net (accessed on 11 November 2023).
29. Brogliato, B. *Nonsmooth Mechanics—Models, Dynamics and Control*, 3rd ed.; Springer International Publishing: Cham, Switzerland, 2016.
30. Peidr , A.; P rez-Navarro, P.D.; Puerto, R.; Pay , L.; Reinoso, O. Locking underactuated robots by shrinking their manifolds of free-swinging motion. *Mech. Mach. Theory* **2023**, *188*, 105403. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Research on the Multi-Robot Cooperative Pursuit Strategy Based on the Zero-Sum Game and Surrounding Points Adjustment

Gang Chen ^{1,*}, Wenqian Xu ¹, Zixing Li ¹, Yuqiang Liu ² and Xin Liu ²

¹ School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China; xuwenqian@buptrobot.com (W.X.); lizixing@bupt.edu.cn (Z.L.)

² Beijing Institute of Spacecraft System Design Cast, Beijing 100086, China; owubb8@163.com (Y.L.); fjjjwj@163.com (X.L.)

* Correspondence: chengang_zdh@bupt.edu.cn

Abstract: Making full use of the cooperation of multi-robots can improve the success rate of a pursuit task. Therefore, this paper proposes a multi-robot cooperative pursuit strategy based on the zero-sum game and surrounding points adjustment. First, a mathematical description of the multi-robot pursuit problem is constructed, and the zero-sum game model is established considering the cooperation of the pursuit robots and the confrontation between the pursuit robots and the escape robot. By solving the game model, the optimal movement strategies of the pursuit robots and the escape robot are obtained. Then, the position adjustment method of the pursuit robots is studied based on the Hungarian algorithm, and the pursuit robots are controlled to surround the escape robot. Based on this, a multi-robot cooperative pursuit strategy is proposed that divides the pursuit process into two stages: pursuit robot position adjustment and game pursuit. Finally, the correctness and effectiveness of the multi-robot cooperative pursuit strategy are verified with simulation experiments. The multi-robot cooperative pursuit strategy allows the pursuit robots to capture the escape robot successfully without conflicts among the pursuit robots. It can be seen from the documented simulation experiments that the success rate of the pursuit task using the strategy proposed in this paper is 100%.

Keywords: multi-robot cooperative pursuit strategy; zero-sum game; Hungarian algorithm; pursuit robot position adjustment

Citation: Chen, G.; Xu, W.; Li, Z.; Liu, Y.; Liu, X. Research on the Multi-Robot Cooperative Pursuit Strategy Based on the Zero-Sum Game and Surrounding Points Adjustment. *Machines* **2021**, *9*, 187. <https://doi.org/10.3390/machines9090187>

Academic Editors: Raffaele Di Gregorio and Dan Zhang

Received: 3 July 2021

Accepted: 1 September 2021

Published: 3 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A multi-robot system consists of more than two robots, which can improve the efficiency of task completion with cooperation [1]. It has good scalability to adapt to different tasks by adjusting the number and types of robots in the system [2,3]. Multi-robot systems can replace humans to complete tasks such as information collection [3,4], target pursuing [3–6], target capture [7,8], etc. There is usually a certain degree of a confrontation relationship between the targets and the multi-robot system during tasks. The target, namely the escape robot, wants to be away from the multi-robot system, and the multi-robot system, namely the pursuit robot team, tries every means to capture the target at the same time. The multi-robot system cooperative pursuit strategy is the key to improving the task efficiency and success rates. If there is no cooperation, it will lead to low pursuit efficiency, long task completion time, and even task failure, which will threaten the security of the robots involved in the pursuit. Therefore, it is of great significance to conduct research on the multi-robot pursuit problem and to design a multi-robot cooperative pursuit strategy.

Currently, many scholars have researched the multi-robot pursuit problem. The multi-robot pursuit problem includes two cases: multi-pursuit robots capture multi-escape robots (many-vs.-many) and multi-pursuit robots capture a single-escape robot (many-vs.-one). In the case of many-vs.-many, the task assignment should be done first to determine if all of the pursuit robot sets correspond to every escape robot, and the pursuit robots sets are then

used to capture the corresponding escape robots. Thus, it can be seen that a case of many-vs.-many is transformed into a case of many-vs.-one based on the task assignment [9–11]. In addition, in the field of space, there is often only one pursuit robot that is used to capture the escape robot, such as the “Phoenix” and “Shenzhou 12 docking with Chinese space station” projects, etc. Additionally, the research focus is gradually expanded to capture an escape robot with many pursuit robots to reduce the control difficulty of the pursuit robot. A mature research system has not yet been formed for the pursuit problem of many-vs.-one, so this paper conducts research on that subject.

When the escape robot moves faster than the pursuit robots do, it is necessary to give full play to the advantages of the cooperative multi-robot system, which puts forward higher requirements for the design of a multi-robot cooperative pursuit strategy. To realize the pursuit of the escape robot with high speed, some scholars have researched the constraints of the initial position distribution of the pursuit robots and the number of the pursuit robots required to complete the pursuit tasks. Breakwell [12] completed the boundary analysis of two pursuit robots with slower speeds pursuing an escape robot with a faster speed based on a differential game and determined the relative positional relationship of the pursuit robots at the beginning of the pursuit task. Zha [13,14] applied the boundary analysis method to the pursuit constraint analysis of tasks with many pursuit robots and one escape robot. The minimum number of pursuit robots required and the speed relationship between the pursuit robots and the escape robot were further determined. Jin [15] also conducted a pursuit constraints analysis based on the Apollonius circle, which included the minimum number of pursuit robots required and the position distribution at the beginning of the pursuit task. All of the above studies are based on the constraints analysis of the number and initial position of pursuit robots. However, the positions of the pursuit robots are random at the beginning of the pursuit task, so the initial position constraints may not be satisfied, which may lead to task failure. Considering that the perception ranges of the robots are limited and considering that when the pursuit robots are outside the perception range of the escape robot, the process of pursuit and escape will not start immediately. Su [16] proposed a multi-robot cooperative pursuit strategy based on the Q-learning algorithm, where the pursuit robots are moved to a certain distance from the escape robot. However, this strategy does not analyze whether the positions of the pursuit robots meet the initial position constraints, which results in the pursuit task having a high probability of failure. The analysis of the above research shows that the key to improving the success rate of the pursuit task is to determine the initial pursuit conditions of the initial position and the number of pursuit robots. Therefore, this paper designs a multi-robot cooperative pursuit strategy that can adjust the positions of the pursuit robots to form a pursuit configuration that satisfies the pursuit conditions before the pursuit begins.

After the pursuit robots surround the escape robot with the pursuit configuration, the movement strategies of the pursuit robots will directly affect the pursuit result. In the previously conducted research on the movement strategies of pursuit robots, many scholars have adopted discrete methods to simplify the problem and to improve the efficiency. Benda [17] conducted a study with many pursuit robots and one escape robot based on a discrete grid environment, and each robot was restricted to move in the horizontal or vertical direction. On this basis, Korf [18] introduced movement along the diagonal to expand the robots’ movement strategies. Zhou [19] regarded the pursuit process as a multi-stage game and established a game model to obtain the movement strategies of pursuit robots in the pursuit process. However, rasterization of the environment causes the robots’ movement to be limited to the horizontal, vertical, and diagonal directions in these research studies. This causes a big difference between the research and the actual scenario. Skrzypczyk [20] discretized and combined the angular velocity and linear velocity of the robots to obtain the movement strategy of the robots so that the robots can move in more directions. This reduces the limitation influence of the rasterization method on the robots’ movement strategy, thereby reducing the difference between the research and the actual scenario. This method used to conduct this research on the multi-robot target pursuit

problem is based on non-cooperative game theory. However, the escape robot did not take the initiative to escape, which is not consistent with the actual situation.

In research previously conducted on multi-robot pursuit, the movement of the escape robot also needs to be considered. For this problem, Selvakumar [21] regarded the team of pursuit robots as a players in a game and the escape robot as the another player. They proposed a method using the game matrix to determine the movement strategy of the pursuit robots to achieve the pursuit of the active escape robot. However, there is always only one pursuit robot at a time in this study, and the remaining pursuit robots are not well used, resulting in a waste of resources. In order to improve the participation degree of various pursuit robots in the pursuit process, some experts give full play to the cooperation of the multi-robot system. Alexander [22] proposed multiple two-player game decomposition (MTPGD) by changing the player selection method in the game and by considering the pursuit problem as a combination of multiple zero-sum games. Among them, the encapsulated-team two-player game decomposition (ETTPGD) embodies the strict confrontation relationship between the pursuit and escape robot during the pursuit process. Both MTPGD and ETTPGD can complete the pursuit. However, they have not considered the cooperative relationship between the pursuit robots, which may easily lead to conflicts among the pursuit robots. Therefore, it is necessary to not only consider the competitive relationship between the pursuit robots and the escape robot, but also the cooperative relationship of the pursuit robots by analyzing the mutual influence of the pursuit robots in the pursuit process.

In summary, in order to improve the success rate of a pursuit task with a multi-robot system, this paper divides the multi-robot pursuit process into two stages. First, this paper adjusts the position distribution of the pursuit robots to form a pursuit configuration for the escape robot based on the pursuit conditions. Then, a movement strategy is proposed for the pursuit robots that is based on a zero-sum game to complete the pursuit task. The main contributions of this paper are:

1. The problem of multi-robot cooperative pursuit: a multi-robot cooperative pursuit strategy including two stages and considering both pursuit robot position adjustment and the pursuit robots' pursuit of the escape robots based on a zero-sum game is proposed to improve the success rate of the pursuit task;
2. The cooperation of the pursuit robots and the competition between the pursuit robots and the escape robot are considered comprehensively to establish a zero-sum game model in this paper, which avoids conflict among the pursuit robots and improves the safety of the multi-robot system during the pursuit process.

The article structure is as follows. The second part describes the problem of multi-robot pursuit. The third part establishes the zero-sum game model of the multi-robot pursuit problem and proposes a method to solve the game model. Then, a pursuit robot movement strategy is proposed in the game stage in order to pursue the escape robot. The fourth part completes the position adjustment of the pursuit robots based on the Hungarian algorithm and form a pursuit configuration for the escape robot. The overall design of the multi-robot cooperative pursuit strategy is completed in this part. In the fifth part, a simulation experiment for the four pursuit robots and an escape robot are designed to verify the effectiveness of the multi-robot cooperative pursuit strategy mentioned in this paper. Finally, the summary of this paper is presented.

2. The Description of the Multi-Robot Pursuit Problem

In the study of multi-robot cooperative pursuit, the form of the pursuit tasks will have an important impact on the design of the cooperative pursuit strategies. Therefore, the multi-robot pursuit task will be analyzed in this part to determine the movement of the pursuit robots during the pursuit process and to give a mathematical description of the multi-robot pursuit problem.

In the two-dimensional space, the schematic diagram of the multi-robot pursuit task is shown in Figure 1. Where the escape robot is recorded as E , the n pursuit robots are

recorded as $P = \{P_1, P_2, \dots, P_n\}$. Combined with the polygon forming conditions, it can be seen that $n \geq 3$ must be met to ensure that the pursuit robots can form an effective encirclement for the escape robot. Considering the operating safety of the robots, there are n circular areas that represent the safe operation area of each robot with the center of each pursuit robot's position and the radius of d_{safe} . At the same time, the circle area at the center of the escape robot's position and the radius of d_E is the safe area for the escape robot. The escape robot will not play games with pursuit robots outside of this area. During the pursuit process, the position of the pursuit robot team is recorded as $P = [P_1, P_2, \dots, P_n]^T$. P_i is the position of the i -th pursuit robot P_i , and E is the position of escape robot. The distance between the i -th pursuit robot and the escape robot is $d_i(t)$. When Equation (1) is satisfied, the capture condition is met, and the pursuit task is completed.

$$D_{\min}(P, E, t) = \min\{d_1(t), d_2(t), \dots, d_n(t)\} \leq D_{\text{captured}} \tag{1}$$

where $D_{\min}(P, E, t)$ is the minimum distance between the pursuit robots and the escape robot, D_{captured} is the capture distance.

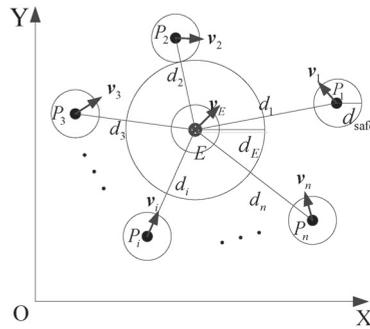


Figure 1. The schematic diagram of the multi-robot pursuit task.

During the pursuit process, the movement of each pursuit robot is shown in Figure 2. Each robot moves at the maximum speed V_p (the magnitude of the velocity), and the velocity of the i -th pursuit robot is denoted as v_i . The velocity of the escape robot is denoted as v_E , and the speed of it is V_E . The position of the pursuit robot P_i at the moment of t is recorded as $P_i(t) = [x_i(t), y_i(t)]$, $i = 1, 2, \dots, n, t \geq 0$. After a decision period T , the position of P_i changes to $P_i(t + T) = [x_i(t + T), y_i(t + T)]$, and it can be calculated by the following equation:

$$\begin{cases} x_i(t + T) = x_i(t) + V_p T \cos(\theta_i + \varphi_i) \\ y_i(t + T) = y_i(t) + V_p T \sin(\theta_i + \varphi_i) \end{cases} \tag{2}$$

where θ_i is the angle between the velocity direction of P_i and the positive direction of the X axis at the moment of t . φ_i is the velocity direction deflection angle of P_i in the decision period T , and the counterclockwise deflection is specified as positive. Due to the limited deflection ability of the robots, $\varphi_i \in [-\varphi_{\text{max}}, \varphi_{\text{max}}]$ needs to be met, and $\varphi_{\text{max}}(\varphi_{\text{max}} > 0)$ is the maximum deflection angle. Since the movement description process of the escape robot E is similar to that of the pursuit robots, it will not be described in detail.

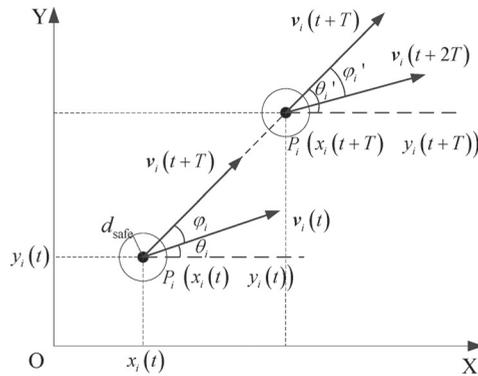


Figure 2. The schematic diagram of the movement of P_i .

Based on the above, the mathematical description of the multi-robot pursuit problem is as follows:

$$\begin{aligned}
 S(t) &= [S_1(t), S_2(t), \dots, S_n(t), S_E(t)], S_i(t) = \varphi_i \\
 s.t. D_{\min}(P, E, kT) &= \min\{d_1(kT), d_2(kT), \dots, d_n(kT)\} \leq D_{\text{captured}} \\
 t &\in [0, kT], k \in \mathbb{N}
 \end{aligned} \tag{3}$$

where $S(t)$ represents the deflection angles of all of the robots at the moment of t , and k is the number of decisions required to complete the pursuit.

3. Movement Strategy Solving Method Based on Zero-Sum Game

After completing the mathematical description of the multi-robot pursuit problem, the multi-robot cooperative pursuit model is conducted and solved based on a zero-sum game in order to achieve the movement strategy of the pursuit robots in this part. The pursuit robot team $P = \{P_1, P_2, \dots, P_n\}$ is regarded as a player in the game, and the escape robot is regarded as another player. The cooperation of the pursuit robots and the competition between the pursuit robots and the escape robot are comprehensively considered to design the game payoff function to improve the zero-sum game model. Then, the optimal movement strategy for the pursuit robots during the pursuit process is designed by solving the game model.

3.1. Establishment of Multi-robot Pursuit Zero-Sum Game Model

Combined with the aforementioned robot movement method, the pursuit process of the multi-robot pursuit is decomposed into multiple decision-making stages. Each stage is regarded as a round of the game. The strategy set of players P and E should be determined first, which represents the possible movements of the pursuit robot team and the escape robot in a round of the game. In each round of the game, the construction process of the strategy set is as follows:

The deflection angle φ_i of the i -th pursuit robot P_i can be discretized as shown in Figure 3. Then, the movement strategy of pursuit robots can be obtained:

$$S_i = \left\{ \varphi_i^1, \varphi_i^2, \dots, \varphi_i^K \mid \varphi_i^j \in [-\varphi_{\max}, \varphi_{\max}], j \in [1, K], i = 1, 2, \dots, n \right\} \tag{4}$$

where φ_{\max} is the maximum deflection angle of the pursuit robots, and K is the number of the deflection angle discretization. φ_i^j represents the j -th deflection angle of P_i , which is optional.

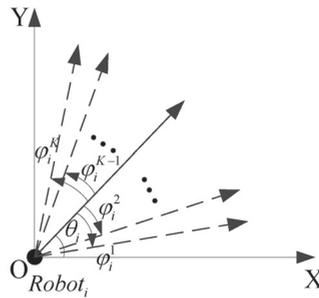


Figure 3. Discretization of robot motion deflection angle.

The strategy set S_P of the pursuit robots team can be obtained by combining the movement strategies of the pursuit robots. Similarly, the strategy set S_E of the escape robot can be obtained:

$$S_E = \{ \varphi_E^1, \varphi_E^2, \dots, \varphi_E^K | \varphi_E^j \in [-\varphi_{max}^E, \varphi_{max}^E], j \in [1, K] \} \tag{5}$$

When the pursuit robots and escape robot adopt a certain strategy from their strategy set, the pursuit situation $\Omega(S_P, S_E)$ is formed. Then, we need to design the payoff function, which is used to determine the optimal solution. Combined with the basic elements of the game, the payoff function $Payoff_P(S_P, S_E)$ of the pursuit robots team and the payoff function $Payoff_E(S_P, S_E)$ of the escape robot can be designed to evaluate the strategy adopted by the two parties. Based on the zero-sum game, it can be known that the sum of the gains and losses of both parties involved in the game is always zero in the case of strict competition. That is, the gains of one party must mean the equal losses of the other party. According to the characteristics of the zero-sum game, the payoff function of the pursuit robot team and the escape robot has the following relationship:

$$Payoff_E(S_P, S_E) = -Payoff_P(S_P, S_E) \tag{6}$$

It can be seen from Equation (6) that when one of the payoff functions is designed completely, another payoff function can be subsequently determined. The payoff function design of the pursuit robot team should take into account the cooperative relationship among the pursuit robots, while the payoff function of the escape robot needs to destroy this cooperative relationship. In the case that each robot operates safely, the pursuit robot team should maintain a compact state to enhance the cooperation capability and to gradually move towards the escape robot. The escape robot needs to break the compactness of the pursuit robot team and move away from the pursuit robots. Therefore, the payoff function of the pursuit robot team and the escape robot are designed as follows:

1. Pursuit robot team distribution and maintenance item F_p^1 :

In order to ensure that the pursuit robot team maintains a compact distribution, we propose the pursuit robot team distribution and maintenance item F_p^1 . It presents the compact degree of the pursuit robots' positions. Adjusting this item can make the pursuit robots distribute compactly, and it is not easy for one robot to remove itself too far away from the team.

In the inertial coordinate system, the virtual center $C_{team}^b = [x_c^b(t), y_c^b(t)]$ of the pursuit robots team, can be represented as follows:

$$\begin{cases} x_c^b(t) = \frac{x_1^b(t) + \dots + x_n^b(t)}{n} \\ y_c^b(t) = \frac{y_1^b(t) + \dots + y_n^b(t)}{n} \end{cases}, (b = 0, 1) \tag{7}$$

where $b = 0$ represents the current pursuit situation, and $b = 1$ represents the new situation with the pursuit strategy.

Comprehensively consider the distance standard deviation $\sigma^b(P, E)$ between each pursuit robot and the escape robot and the distance $d^b(C_{team}^b, E)$ between the virtual center of the pursuit robot team and the escape robot to design a pursuit robot team distribution and maintenance item F_p^1 .

$$F_p^1 = -\eta_1 \tan((\sigma^1(P, E) - \sigma^0(P, E)) / K_A) - \eta_2 \tan((d^1(C_{team}^b, E) - d^0(C_{team}^b, E)) / K_A) \tag{8}$$

where K_A is a constant, and η_1, η_2 are the weight coefficients, $\eta_1 + \eta_2 = 1$. When it is more focused on keeping the distance d_i between the pursuit robots and the escape robot consistent, the η_1 is larger. When it is more focused on creating the distance $d(C_{team}^b, E)$ between the virtual center of each pursuit robot and the escape robot, the η_1 is shorter, and the η_2 is larger. The coefficients of η_1 and η_2 are adjusted according to experience. The distribution of the pursuit robots can be kept compact by adjusting η_1 and η_2 . Meanwhile, the pursuit robots can be prevented from leaving the team to move alone, and the cooperative relationship between the pursuit robots can be enhanced during the pursuit process in this way.

2. Pursuit distance item F_p^2 :

In order to make the pursuit robot team approach the escape robot gradually and to ensure that the escape robot gradually moves away from the pursuit robots, we propose the pursuit distance item F_p^2 . It represents the distance between the pursuit robot team and the escape robot. By adjusting this item, the pursuit robot team can gradually approaches the escape robot. During the pursuit process, $D_{team}^b(P, E)$ is the total distance between the pursuit robot team and the escapee.

$$D_{team}^b(P, E) = \sum_{i=1}^n d_i^b(t) \tag{9}$$

$D_{min}^b(P, E)$ is the closest distance between the pursuit robot team and the escape robot.

$$D_{min}^b(P, E) = \min\{d_1^b(t), d_2^b(t), \dots, d_n^b(t)\} \tag{10}$$

In combination with the above two distances, the pursuit distance item F_p^2 can be designed as

$$F_p^2 = -\beta_1 \tan((D_{team}^1(P, E) - D_{team}^0(P, E)) / K_B) - \beta_2 \tan((D_{min}^1(P, E) - D_{min}^0(P, E)) / K_B) \tag{11}$$

where K_B is a constant, β_1, β_2 are the weight coefficients, and $\beta_1 + \beta_2 = 1$. When the coefficient β_1 is larger, it ensures that the pursuit robots are more inclined to shorten the overall distance and realize the contraction of the surrounding points. When the coefficient β_2 is larger, it ensures that the pursuit robot team is more inclined to continuously control the closest pursuit robot to approach the escape robot. The coefficients β_1 and β_2 are adjusted according to experience.

3. Robot collision avoidance item F_p^3 :

In order to improve the safety of the robots and to avoid a multi-robot collision, we propose the robot collision avoidance item F_p^3 . It represents the collision status of each robot. By adjusting this item, the safe operation of the robots can be ensured, and the robots do not collide with each other over the course of the pursuit task. The robot collision avoidance item F_p^3 is designed as follows.

$$F_p^3 = \begin{cases} -\infty, \exists d_{ij}^1(S_p^w, S_E^v) \leq 2d_{safe} \\ 0, \forall d_{ij}^1(S_p^w, S_E^v) > 2d_{safe} \end{cases}, i \neq j, i, j = 1, 2, \dots, n, E \tag{12}$$

where $d_{ij}^1(S_p^w, S_E^v)$ represents the distance between any two robots after the pursuit robot team adopts the movement strategy S_p^w and after the escape robot adopts the movement strategy S_E^v .

Based on the above sub-items, the payoff function expression is as follows:

$$Payoff_P(S_P, S_E) = \tau_1 \lg(F_P^1 + K_C) + \tau_2 \lg(F_P^2 + K_C) + \tau_3 F_P^3 \tag{13}$$

where K_C is a constant, the weight coefficients τ_1, τ_2, τ_3 are adjusted according to experience, and $\tau_1 + \tau_2 + \tau_3 = 1$.

Equation (13) can be used to evaluate the possible strategy combination adopted by the pursuit robots and the escape robot. Then, a multi-robot team payoff matrix and the escape robot payoff matrix can be built with the evaluation results, as shown in Equations (14) and (15).

$$I_P = \begin{pmatrix} Payoff_P^{11} & \dots & Payoff_P^{1K} \\ \vdots & \ddots & \vdots \\ Payoff_P^{U1} & \dots & Payoff_P^{UK} \end{pmatrix}, U = K^n \tag{14}$$

$$I_E = \begin{pmatrix} Payoff_E^{11} & \dots & Payoff_E^{1K} \\ \vdots & \ddots & \vdots \\ Payoff_E^{U1} & \dots & Payoff_E^{UK} \end{pmatrix} = - \begin{pmatrix} Payoff_P^{11} & \dots & Payoff_P^{1K} \\ \vdots & \ddots & \vdots \\ Payoff_P^{U1} & \dots & Payoff_P^{UK} \end{pmatrix} \tag{15}$$

Among them, after the escape robot adopts the movement strategy S_E^v and the pursuit robots adopt the movement strategy S_p^w , the corresponding payoff function $Payoff_P(S_p^w, S_E^v)$ is abbreviated as $Payoff_P^{wv}$. K is the strategies number obtained by discretizing the movement strategy of the robots.

In this section, the zero-sum game model of the multi-robot pursuit problem is established based on the three basic elements of the game player, strategy sets, and payoff function.

3.2. The Solution Method of Optimal Game Movement Strategy

Based on the multi-robot pursuit zero-sum game model, in order to obtain the optimal movement strategy of the pursuit robots in each round of the game, this section solves the game model by solving the game payoff matrix.

In one round of the game, when the pursuit robots adopt the strategy S_p^* and the escape parties adopt the strategy S_E^* that satisfy Equation (16), the game reaches a Nash equilibrium state. At this time, strategy S_p^* and S_E^* are the pure strategy Nash equilibrium solution of the game. In the Nash equilibrium state, neither the pursuit robots nor the escape robot can unilaterally change their adopted strategy to make the situation more beneficial to itself. S_p^* and S_E^* are the optimal movement strategies that the pursuit robot team and the escape robot can adopt in this round of the game. That is, the movement strategy corresponding to the pure strategy Nash equilibrium solution is the current optimal movement strategy.

$$\begin{aligned} Payoff_P(S_p^*, S_E^*) &\geq Payoff_P(S_p^*, S_E^v), \forall S_E^v \in S_E \\ Payoff_E(S_p^*, S_E^*) &\geq Payoff_E(S_p^w, S_E^*), \forall S_p^w \in S_P \end{aligned} \tag{16}$$

Combining the definition of the Nash equilibrium solution and the method in references [23,24] can solve the model. First, for each of the escape robot's escape strategies, the optimal pursuit strategy of the pursuit robots team to deal with the escape strategy is solved and forms a set T_P .

$$T_P = \left\{ S^{v\omega} \mid \max_{\omega} \{ Payoff_P^{1v}, \dots, Payoff_P^{Uv} \}, v = 1, 2, \dots, K \right\} \tag{17}$$

where $S^{v\omega}$ indicates the strategy combination with the strategy S_E^v that the escape robot adopts and the corresponding optimal strategy S_p^{ω} that the pursuit robots team adopts.

Similarly, for each pursuit robot team movement strategy, the corresponding optimal movement strategy of the escape robot can be determined as shown in Equation (18) and can form a strategy set T_E .

$$T_E = \left\{ S^{\omega v} \mid \max_v \left\{ \text{Payoff}_{f_E^{\omega 1}}, \dots, \text{Payoff}_{f_E^{\omega K}} \right\}, \omega = 1, 2, \dots, U \right\} \quad (18)$$

where $S^{\omega v}$ indicates the strategy combination with the strategy S_p^{ω} that the pursuit robot team adopts and the corresponding optimal escape strategy S_E^v that the escape robot adopts.

The pure strategy Nash equilibrium solution NE of the game can be obtained by solving the intersection of sets T_P and T_E . When there is only one solution for NE , the strategy combination for NE can be considered to be the optimal movement strategy adopted by the pursuit robot team and the escape robot in this round of the game.

In view of the possible situation that there is no solution or a multi-solution, the following processing is conducted to ensure the existence and uniqueness of the movement strategy of the pursuit robot team and the escape robot:

In the case of $T_P \cap T_E = \emptyset$, the pursuit robot team and the escape robot cannot obtain the optimal movement strategy through the pure strategy Nash equilibrium solution in this round of the game. Taking into account the confrontational relationship between the pursuit robot team and the escape robot, the two sides should determine their respective movement strategies according to the principle of avoiding taking on the strategy that would have the most adverse effect on them. By determining the minimum element $e = \min(I_P)$ of the payoff matrix I_P and by removing the row where the e is located, the most unfavorable situation caused by the movement strategy of the escape robot being adopted is avoided. If e is located in multiple rows, the second minimum elements of these rows are compared until only one row is determined. This operation should be repeated until there is only one row left in the payoff matrix, and the corresponding strategy is the movement strategy S_p^* of the pursuit robot team under the aforementioned principles. The movement strategy S_E^* of the escape robot can be determined using similar methods to deal with the columns of the payoff matrix I_E in the current round.

In the case of $T_P \cap T_E = NE$ and NE have multiple solutions, the optimal pursuit strategy is not unique. Taking into account the stability of the pursuit task, the shortest distance $D_{\min}^1(P, E)$ between the pursuit robots team and the escape robot after using the execution strategy $[S_p^{i*}(t), S_E^{i*}(t)]$ is introduced as an additional evaluation criterion. The strategy with the smallest $D_{\min}^1(P, E)$ is selected as the optimal choice of movement strategy.

$$S(t) = \left\{ [S_p^{i*}(t), S_E^{i*}(t)] \mid \min_{S_p^{i*}(t), S_E^{i*}(t)} D_{\min}^1(P, E) \right\} \quad (19)$$

The pursuit robot and escape robot movement strategies can be obtained by the above method for each round.

$$S(t) = [S_p^*(t), S_E^*(t)] = [S_1(t), S_2(t), \dots, S_n(t), S_E(t)] \quad (20)$$

4. Design of Multi-Robot Pursuit Strategy

Based on the establishment of the zero-sum game model and the design of the game model solution method, it is necessary to conduct research on the multi-robot pursuit strategy to improve the success rate of the pursuit task. Additionally, the success rate of the pursuit task can be improved by pre-forming the surrounding configuration of the escape robot.

It can be seen from the literature [15,16] that when the number of pursuit robots meets the condition $V_p/V_E = \lambda \geq \sin(\pi/n)$ and when the pursuit robots are evenly distributed on the circle centered around the escape robot, the pursuit robots with a slower velocity can adopt appropriate strategies to pursue the escape robot with a higher velocity to increase the success rate of the pursuit task. At this time, the positional relationship between the pursuit robots and the escape robot is shown in Figure 4.

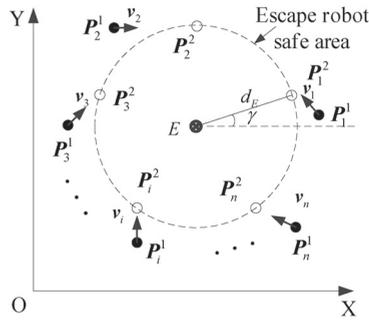


Figure 4. Schematic diagram of robot surround.

Where $P_i^1 = [x_i, y_i]$ and $P_j^2 = [x_j, y_j] (i = 1, 2, \dots, n, j = 1, 2, \dots, n)$ represent the initial position of P_i^1 and the target position after the escape robot is surrounded.

In order to control each pursuit robot to move from the initial position to the target position at the same time and to improve the position adjustment efficiency, this paper adopts the Hungarian algorithm [23], which is widely used to solve small and medium scale assignment problems to complete the surround points allocation of pursuit robots.

In the process of surround points allocation, the task cost matrix $C^{n \times n}$ is constructed using the distance $f(P_i^1, P_j^2)$ between the initial position of the pursuit robots to the target position. By solving the task cost matrix, the target position of each pursuit robot can be determined. Then, each pursuit robot can be controlled to complete the position adjustment before the start of the pursuit game and can form a surrounding configuration for the escaped robot.

After the surround points allocation and the position adjustment of the pursuit robots, combining the aforementioned establishment and solve method of the zero-sum game model, the multi-robot pursuit strategy can be constructed to realize the capture of the escape robot with the pursuit robot team. The detailed process is as follows:

- Step 1: Randomly generate the initial positions of the pursuit robots and the escape robot;
- Step 2: The initial position distribution constraints of the pursuit task are considered to select the target position of the pursuit robots to surround the escape robot and use the Hungarian algorithm to complete the target position allocation of each pursuit robot;
- Step 3: Each pursuit robot moves according to the result of the target position allocation and forms a surrounding configuration for the escape robot;
- Step 4: A zero-sum game model of the multi-robot pursuit problem is established;
- Step 5: The zero-sum game model is solved to obtain the movement strategies of the pursuit robots and the escape robot at each decision-making stage and to complete the pursuit process of the escape robot.

In summary, the overall process of the multi-robot cooperative pursuit strategy designed in this chapter is shown in Figure 5.

So far, this paper has completed research regarding a multi-robot cooperative pursuit strategy. This strategy divides the process of multi-robot pursuit into two stages: pursuit robot position adjustment and the pursuit of the escape robot based on a zero-sum game. In the first stage, this paper realized the position transition of the pursuit robots from the randomly generated initial configuration to the surround configuration that satisfies the pursuit constraints. In the second stage, this paper designed a pursuit strategy to realize the capture of the escape robot based on a zero-sum game.

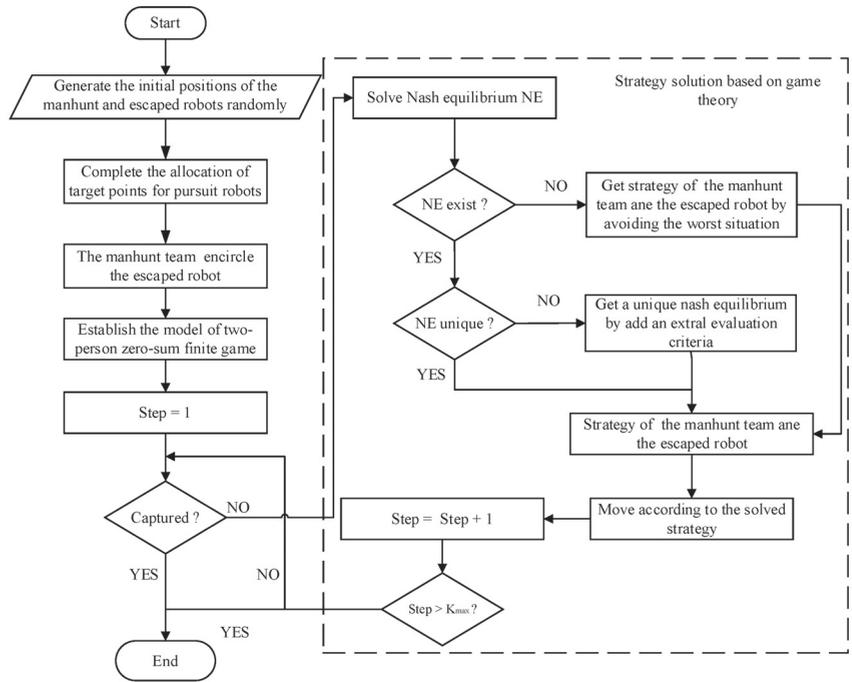


Figure 5. Multi-robot cooperative pursuit strategy.

5. Results and Discussion

In this chapter, the correctness and effectiveness of the multi-robot cooperative pursuit strategy proposed in this paper will be simulated and verified. When the number of pursuit robots is equal to or greater than three, it is sufficient to form the pursuit conditions. The increase in the number has little effect on the success rate of the pursuit task. For the sake of simulation experiments without a loss of generality, there are four pursuit robots and an escape robot that are used as experimental objects. When the speed of the robots meets the condition $V_P/V_E = \lambda \geq \sin(\pi/4)$, the simulation experiments are conducted for the three cases when the speed of the pursuit robots is greater than, equal to, and less than that of the escape robot. The simulation results verify the correctness and effectiveness of the multi-robot cooperative pursuit strategy designed in this paper.

5.1. Verification of the Multi-Robot Game Pursuit Model

In order to verify the generality of the multi-robot game pursuit model, four pursuit robots with the speeds of 1.2 m/s, 1.0 m/s, and 0.8 m/s are used to pursue the escape robot with a speed of 1 m/s in this section.

Before the game’s pursuit process, the position E of the escape robot and the position P of the pursuit robots located on the boundary of the escape robot safe area are randomly generated. The radius d_E of the escape robot safe area is set to 20 m. During the pursuit process, the range of motion deflection angle of each pursuit robot is set to $\varphi \in [-\pi/3 \pi/3]$ rad, and the deflection angle φ is discretized with the interval $\pi/18$ rad. The radius of the robots’ safe operation area d_{safe} is set to 1 m. The capture distance $D_{capture}$ is set to 3m. When the distance between each pursuit robot and the escape robot satisfies Equation (1), the pursuit task is completed successfully.

Each pursuit robot should be made to repeat the game pursuit process 30 times for the aforementioned three different speed situations; the success rate of the pursuit task is shown in Figure 6.

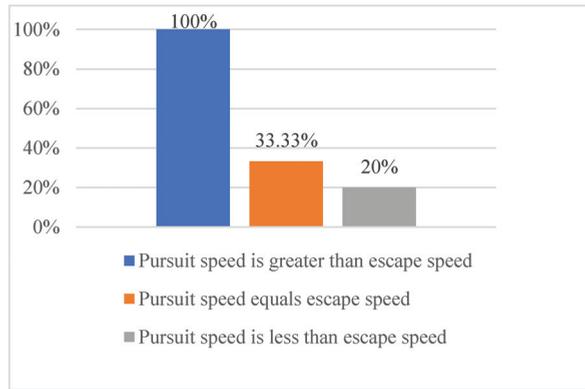


Figure 6. The success rate of pursuit tasks with different pursuit speeds.

During the repeated game pursuit simulation experiment, the pursuit processes of the pursuit robots with different speeds were selected to be shown in Figure 7.

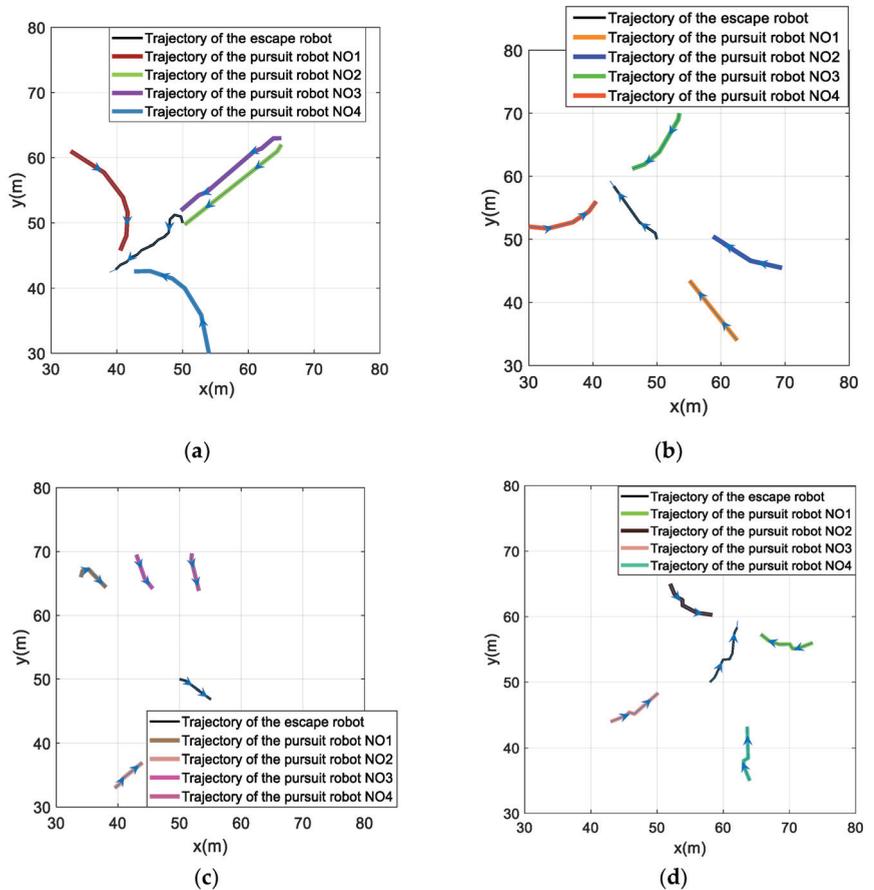


Figure 7. Cont.

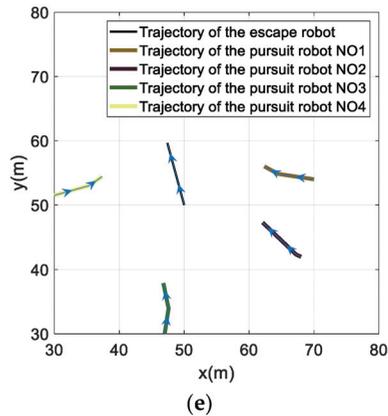


Figure 7. Pursuit robot team pursuit of the escape robot with different speeds: (a) $V_P = 1.2$ m/s, $V_E = 1$ m/s, and successful capture; (b) $V_P = 1$ m/s, $V_E = 1$ m/s, and successful capture; (c) $V_P = 1$ m/s, $V_E = 1$ m/s, and failed capture; (d) $V_P = 0.8$ m/s, $V_E = 1$ m/s, and successful capture; (e) $V_P = 0.8$ m/s, $V_E = 1$ m/s, and failed capture.

Figure 7a,b,d show the movement trajectory of each pursuit robot and the escape robot when the task is completed successfully. Figure 7c,e show the movement trajectories of the robots when the task fails.

Through the comprehensive analysis of the above simulation results, it can be seen that when the pursuit robots move faster than the escape robot, the pursuit robots can always capture the escape robot with their speed advantage. In addition, from the trajectories of Pursuit Robots 2 and 3 in Figure 7a, we know that since the initial positions of Pursuit Robots 2 and 3 are very close, it is a dangerous situation for the pursuit robots. However, under the effect of the collision avoidance term of the payoff function, the pursuit robots adopted a collision-avoidance movement strategy at the initial stage of the pursuit, which increased the distance between Pursuit Robots 2 and 3 and avoided a collision between the pursuit robots. From the simulation results of the successful pursuit tasks in Figure 7b,d, we know that when the escape robot is faster, the pursuit robots maintain the surrounding state of the escape robot through cooperation and gradually shorten the shortest distance between the pursuit robots and the escape robot until the capture conditions are met and the escape robot is captured. It is verified that the multi-robot game pursuit model established in this paper can fully exert the advantage of the cooperative relationship between the pursuit robots. From the simulation results of the failed pursuit tasks in Figure 7c,e, we know that an unreasonable position distribution of the pursuit robots will cause the pursuit task to fail and result in a reduction of the pursuit success rate.

5.2. Validation of the Multi-Robot Cooperative Pursuit Strategy

According to the simulation in the above section, we know that the initial position distribution of the pursuit robots will directly affect the result of the multi-robot pursuit strategy. This section will verify the correctness and effectiveness of the multi-robot cooperative pursuit strategy with the position adjustment of pursuit robots.

In the multi-robot cooperative pursuit process, the initial position E of the escape robot and the initial position P^1 of each pursuit robot are randomly generated. Suppose that the initial positions of the robots are shown in Equation (21).

$$P^1 = \begin{bmatrix} 32 & 93 \\ 50 & 19 \\ 91 & 98 \\ 44 & 12 \end{bmatrix} \text{m}, E = [46 \ 49] \text{m} \quad (21)$$

According to the multi-robot cooperative pursuit strategy designed in this paper, the Hungarian algorithm is first used to allocate the surrounding point of the escape robot, and the pursuit robots adjust their positions according to the allocation results to form the surrounding configuration of the escape robot. After surround points allocation, this paper adopts the A* path-planning algorithm to plan the movement trajectories and to calculate the distance between the initial position and the designated surround position of each pursuit robot. The selection and allocation results of the surrounding points of the escape robot based on the Hungarian algorithm are shown in Figure 8.

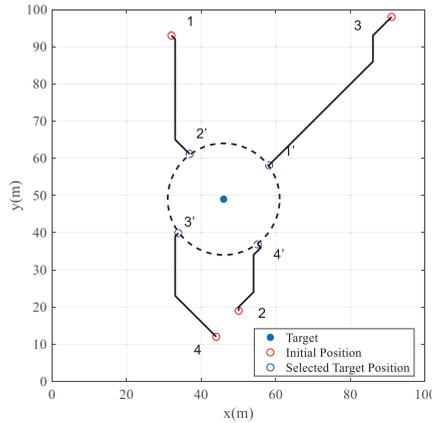


Figure 8. Position adjustment with the Hungarian algorithm.

In order to verify the effect position adjustment on the pursuit robots in this paper in improving the success rate of multi-robot cooperative pursuit, the pursuit robot position adjustment method proposed in [16] is used for comparison. The pursuit conditions only have constraints on the target positions of the pursuit robots, but they do not determine the target positions. In this position adjustment method, each pursuit robot moves from the initial position towards the escape robot until it reaches the escape robot’s safe area boundary. When all of the pursuit robots move to the boundary of the escape robot’s safe area, the robots start the game’s pursuit process. The selection of the surrounding points of the escape robot based on the pursuit conditions is shown in Figure 9.

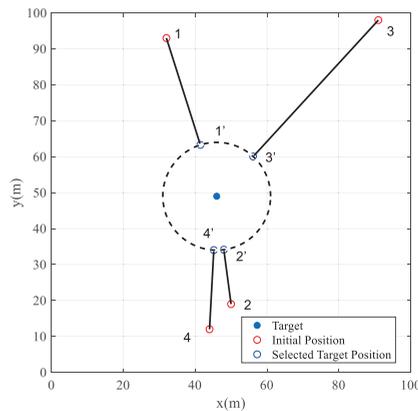


Figure 9. Position adjustment with pursuit conditions.

In Figures 8 and 9, the red and blue circles are the initial and target positions for the pursuit robot position adjustment. The black lines are the trajectory of the pursuit robot

position adjustment. The blue dot is the escape robot, and the dotted circle indicates the escape robot’s safe area. It can be seen from Figures 8 and 9 that during the pursuit robot position adjustment, the trajectory of each pursuit robot is always outside of the escape robot’s safe area, and the game pursuit process will not start.

After the pursuit robot position adjustment with the above two methods, a multi-robot pursuit zero-sum game model is established to obtain the movement strategies of the robots to complete the multi-robot pursuit task. The pursuit robots team have different speeds V_p and pursue the escape robot based on the zero-sum game; the pursuit results are shown in Figure 10.

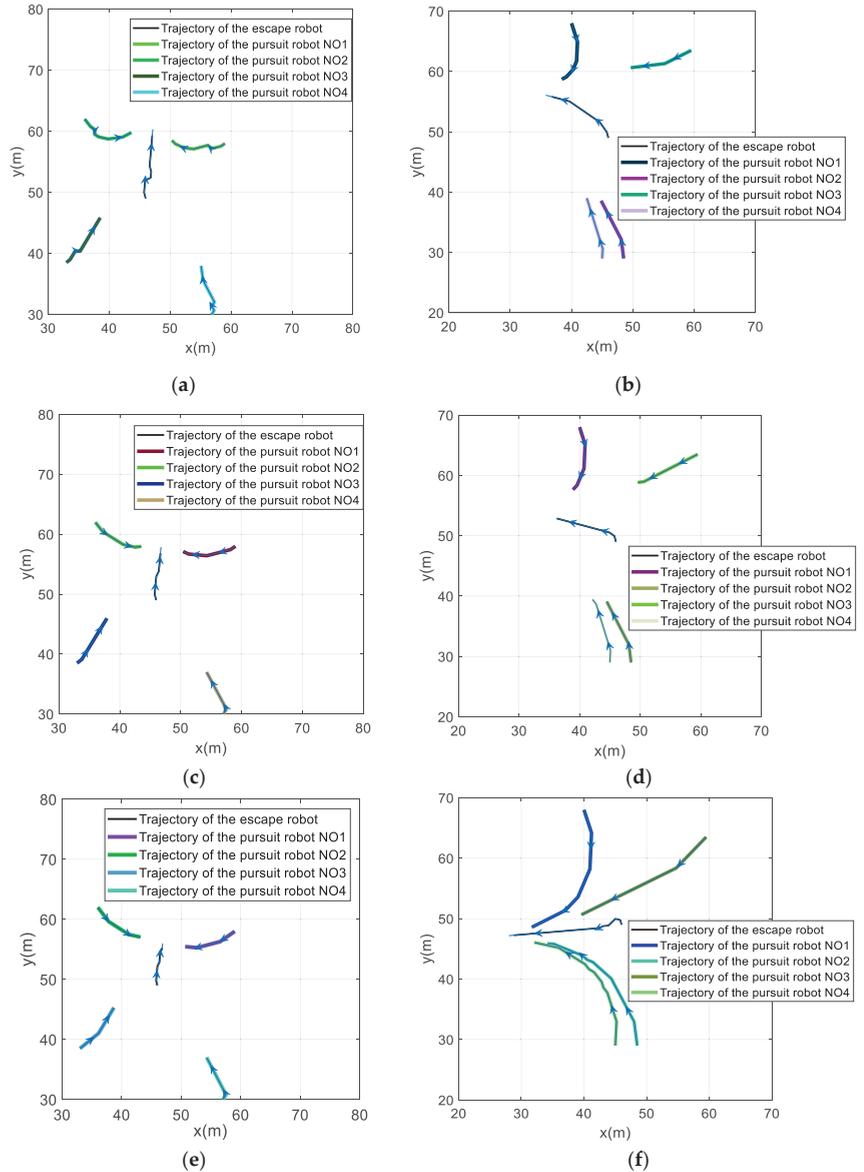


Figure 10. Pursuit results corresponding to the different initial positions and different speeds of pursuit robots: (a) $V_p < V_e$ with the pursuit robot position adjustment method in this paper; (b) $V_p < V_e$

with the pursuit robot position adjustment method in [16]; (c) $V_p = V_e$ with the pursuit robot position adjustment method in this paper; (d) $V_p = V_e$ and the with pursuit robot position adjustment method in [16]; (e) $V_p > V_e$ with the pursuit robot position adjustment method in this paper; (f) $V_p > V_e$ with the pursuit robot position adjustment method in [16].

The multi-robot cooperative pursuit comparison experiment with the different pursuit robot position adjustment methods is repeated 30 times. Additionally, the success rate statistics are shown in Figure 11.

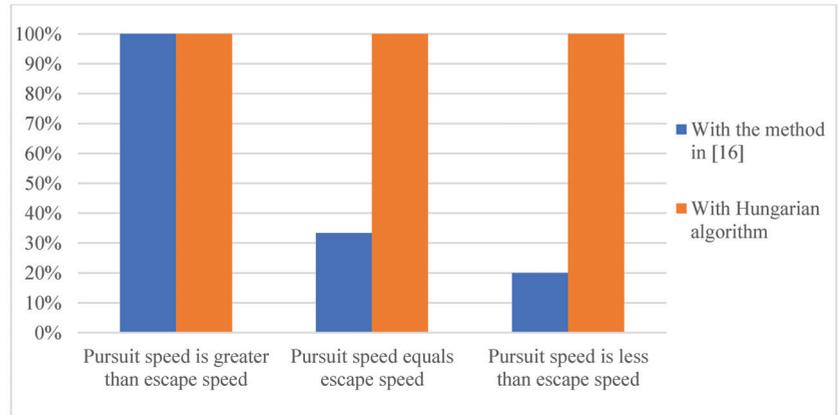


Figure 11. Statistical table of the pursuit results of the different pursuit strategies.

Figure 10a,c,e are the simulation results of the multi-robot cooperative pursuit strategy with different pursuit robot speeds and the same position adjustment method proposed in this paper. The figure shows that the pursuit task can be successfully completed by forming a surrounding configuration for the escape robot based on the Hungarian algorithm before the start of the pursuit game. Figure 10b,d,f are the simulation results of the multi-robot cooperative pursuit strategy, which adjusts the pursuit robot positions using the method in [16]. The pursuit tasks in Figure 10b,d failed because the pursuit robots did not form a surrounding configuration that satisfied the position distribution constraints for the escape robot. This pursuit robot position adjustment method leads to a reduction in the pursuit success rate. In Figure 10f, the pursuit duration is longer than in Figure 10e. This shows that the position adjustment method in [16] leads to a lower pursuit efficiency. The repeated experiment results in Figure 11 show that the success rate of the pursuit tasks using the position adjustment method in this paper can be improved to 100%. This is because when the pursuit robots adjust their positions based on the Hungarian algorithm, the escape robot is tightly surrounded by the pursuit robots. Additionally, even if the escape robot is faster than the pursuit robot, the pursuit robot can capture it pretty successfully. This further proves that the multi-robot cooperative pursuit strategy in this paper can guarantee the capture of the escape robot and can effectively improve the success rate of multi-robot cooperative pursuit tasks.

6. Conclusions

Aiming to solve the problem of multi-robot pursuit, this paper proposed a multi-robot cooperative pursuit strategy based on a zero-sum game and surrounding points adjustment. First, this paper describes the problem of multi-robot pursuit mathematically and abstracts the actual problem into a theoretical model. Second, this paper discretizes the multi-robot cooperative pursuit process and establishes a zero-sum game model for each decision-making stage to obtain the movement strategies of the pursuit and escape robot. Third, this paper controls the pursuit robots to form a surround configuration that satisfies the initial

position distribution constraints of the pursuit task based on the Hungarian algorithm and designs a multi-robot cooperative pursuit strategy that divides the multi-robot pursuit task into two stages: pursuit robot position adjustment and game pursuit. Finally, the simulation shows that the multi-robot cooperative pursuit strategy proposed in this paper can realize the capture of the escape robot by means of the cooperation of pursuit robots and can effectively improve the pursuit success rate. The multi-robot cooperative pursuit strategy proposed in this paper mainly has the following innovations:

1. A multi-robot cooperative pursuit strategy was designed, and the pursuit task was divided into two stages: pursuit robot position adjustment and game pursuit, which improved the success rate of multi-robot cooperative pursuit tasks;
2. The game model of the multi-robot cooperative pursuit tasks was optimized based on a zero-sum game, which comprehensively considered the cooperative relationship between the pursuit robots and the confrontation relationship between the pursuit robot team and the escape robot in the multi-robot cooperative pursuit process. Three pursuit robots team distribution payoff functions and a maintenance item, a pursuit distance item, and a robot collision avoidance item were constructed to give full weight to the advantages of the multi-robot cooperation and to ensure that the multi-robot cooperative pursuit task was completed based on the safe operation of the robots.

The multi-robot cooperative pursuit strategy proposed in this paper is versatile. In follow-up research, the many-to-one confrontation problem of multiple aircraft pursuit can be considered based on the strategy proposed in this paper. In addition, in the position adjustment stage for the pursuit robots, the movement state of the escape robot is assumed to be stationary and was not considered adequately in this paper. This is not in line with a realistic scenario, and we will pay attention to this problem in follow-up research.

Author Contributions: Conceptualization, G.C. and W.X.; methodology, W.X.; software, Z.L.; validation, G.C., W.X. and Z.L.; formal analysis, Y.L.; investigation, X.L.; resources, G.C.; data curation, W.X.; writing—original draft preparation, Z.L.; writing—review and editing, W.X.; visualization, G.C.; supervision, W.X.; project administration, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by State Administration of Science, Technology and Industry for National Defense, grant number No.HTKJ2019KL502012 and BUPT Action Plan to Enhance Capacity for Scientific and Technological Innovation, grant number No. 2021XD-A10-1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Gautam, A.; Mohan, S. A review of research in multi-robot systems. In Proceedings of the 2012 IEEE 7th International Conference on Industrial and Information Systems, Chennai, India, 6–9 August 2012; pp. 1–5.
2. Hung, P.D.; Vinh, T.Q.; Ngo, T.D. A Scalable, Decentralised Large-Scale Network of Mobile Robots for Multi-target Tracking. In Proceedings of the Intelligent Autonomous Systems 13, Padova, Italy, 15–18 July 2014; pp. 621–637.
3. Nestmeyer, T.; Giordano, P.R.; Bühlhoff, H.H.; Franchi, A. Decentralized Simultaneous Multi-target Exploration using a Connected Network of Multiple Robots. *Autonotat. Robot.* **2017**, *41*, 989–1011. [CrossRef]
4. Emoto, S.; Akkaya, I.; Lee, E.A. Information seeking and model predictive control of a cooperative multi-robot system. *Artifi-Cial Life Robot.* **2016**, *21*, 393–398. [CrossRef]
5. Vidal, R.; Shakernia, O.; Kim, H.J.; Shim, D.H.; Sastry, S. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. *IEEE Trans. Robot. Autom.* **2002**, *18*, 662–669. [CrossRef]

6. Ulam, P.D.; Kira, Z.; Arkin, R.C.; Collins, T.R. Mission Specification and Control for Unmanned Aerial and Ground Vehicles for Indoor Target Discovery and Tracking. In Proceedings of the SPIE Defense, Security, and Sensing Symposium, Orlando, FL, USA, 6–9 April 2010; p. 769414.
7. Kehagias, A. Generalized Cops and Robbers: A Multi-player Pursuit Game on Graphs. *Dyn. Games Appl.* **2019**, *9*, 1076–1099. [CrossRef]
8. Bhattacharya, S.; Paul, G.; Sanyal, S. A cops and robber game in multidimensional grids. *Discret. Appl. Math.* **2010**, *158*, 1745–1751. [CrossRef]
9. Makkapati, V.R.; Tsiotras, P.; Zaccour, G. Optimal Evading Strategies and Task Allocation in Multi-player Pursuit–Evasion Problems. *Dyn. Games Appl.* **2019**, *9*, 1168–1187. [CrossRef]
10. Chen, M.; Zhou, Z.; Tomlin, C.J. Multiplayer reach-avoid games via pairwise outcomes. *IEEE Trans. Autom. Control* **2017**, *62*, 1451–1457. [CrossRef]
11. Garcia, E.; Casbeer, D.W.; Moll, A.V.; Pachter, M. Multiple Pursuer Multiple Evader Differential Games. *IEEE Trans. Autom. Control* **2020**, *66*, 2345–2350. [CrossRef]
12. Lewin, J.; Breakwell, J.V. The surveillance-evasion game of degree. *J. Optim. Theory Appl.* **1975**, *16*, 339–353. [CrossRef]
13. Zha, W.; Chen, J.; Peng, Z.; Gu, D. Construction of Barrier in a Fishing Game With Point Capture. *IEEE Trans. Cybern.* **2016**, *47*, 1409–1422. [CrossRef] [PubMed]
14. Chen, J.; Zha, W.Z.; Peng, Z.; Gu, D. Multi-player pursuit–evasion games with one superior evader. *Automatica* **2016**, *71*, 24–32. [CrossRef]
15. Jin, S.; Qu, Z. Pursuit-evasion games with multi-pursuer vs. one fast evader. In Proceedings of the 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010; pp. 3184–3189.
16. Su, Z.B.; Su, J.L.; Tong, L. Strategy of Cooperative Hunting by Multiple Mobile Robots. *J. Beijing Inst. Technol.* **2004**, *5*, 32–35.
17. Benda, M. *On Optimal Cooperation of Knowledge Sources—an Empirical Investigation*; Technical Report; Boeing Advanced Technology Center: Washington, DC, USA, 1986.
18. Korf, R. A Simple Solution to Pursuit Games. In Proceedings of the International Workshop on DAI, Delhi, India, 5–7 February 1992; pp. 83–194.
19. Zhou, P.C.; Hong, B.R. Group robot pursuit-evasion problem based on game theory. *J. Harbin Inst. Technol.* **2003**, *35*, 1056–1059.
20. Skrzypczyk, K. Control of a Team of Mobile Robots Based on Non-cooperative Equilibria with Partial Coordination. *Int. J. Appl. Math. Comput. Sci.* **2005**, *15*, 89–97.
21. Selvakumar, J.; Bakolas, E. Evasion with Terminal Constraints from a Group of Pursuers using a Matrix Game Formulation. In Proceedings of the 2017 American Control Conference, New York, NY, USA, 24–26 May 2017; pp. 1604–1609.
22. Alexopoulos, A.; Badreddin, E. Decomposition of multi-player games on the example of pursuit-evasion games with unmanned aerial vehicles. In Proceedings of the 2016 American Control Conference (ACC), New York, NY, USA, 6–8 July 2016; pp. 3789–3795.
23. Shao, J.; Xu, Y. Cooperative Combat Decision-making Research for Multi UAVs. *Inf. Control* **2018**, *47*, 347–354.
24. Nessah, R.; Tian, G. On the existence of Nash equilibrium in discontinuous games. *Econ. Theory* **2016**, *61*, 515–540. [CrossRef]



Article

Geometric Constraint Programming (GCP) Implemented Through GeoGebra to Study/Design Planar Linkages

Raffaele Di Gregorio * and Tommaso Cinti

Laboratory of Mechatronics and Virtual Prototyping (LaMaViP), Department of Engineering, University of Ferrara, Via Saragat 1, 44122 Ferrara, Italy; tommaso.cinti@edu.unife.it

* Correspondence: raffaele.digregorio@unife.it; Tel.: +39-0532-974828

Abstract: In the study and design of planar mechanisms, graphical techniques for solving kinematic analysis/synthesis and kinetostatics problems have regained interest due to the availability of advanced drawing tools (e.g., CAD software). These techniques offer a deeper physical understanding of a mechanism's behavior, which can enhance a designer's intuition and help students develop their skills. Geometric Constraint Programming (GCP) is the term used to describe this modern approach to implementing these techniques. GeoGebra is an open-source platform designed for the interactive learning and teaching of mathematics and related STEM disciplines. It offers an object-oriented programming language and a wide range of geometric tools that can be leveraged to implement GCP. This work presents a systematic technique for studying and designing planar linkages, based on Assur's groups and GeoGebra's tools. Although some kinematic analyses and syntheses of planar linkages using GeoGebra have been previously introduced, the proposed systematic approach is novel and could serve as a guide for implementing similar problem-solving methods in other graphical environments. Several case studies will be presented to illustrate this novel approach in detail.

Keywords: planar linkage; kinematics; kinetostatics; parametric modeling; geometric constraint programming; Assur group; GeoGebra

Citation: Di Gregorio, R.; Cinti, T. Geometric Constraint Programming (GCP) Implemented Through GeoGebra to Study/Design Planar Linkages. *Machines* **2024**, *12*, 825. <https://doi.org/10.3390/machines12110825>

Academic Editors: Zheng Chen and Fugui Xie

Received: 11 September 2024

Revised: 31 October 2024

Accepted: 12 November 2024

Published: 18 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Graphical techniques [1] were the primary tools used for kinematic analysis and synthesis of planar mechanisms until about the mid-twentieth century [2], when the first kinematic synthesis programs for digital computers appeared [3]. From then on, many software packages for the kinematic analysis and synthesis of planar linkages were developed during the second half of the twentieth century. Such programs moved the central interest of researchers from graphical to numerical techniques. Unfortunately, even though numerical techniques fully exploited the power of digital computers, they were less prone than graphical ones to guide the intuition of students and designers toward a full understanding of the practicable solutions of kinematic problems.

In the meantime, solid-modeling software packages enhanced both their user interfaces and their integrated numerical solvers. These tools, together with the greatly increased computational speeds of digital computers, created advanced drafting platforms where the parametric programming of linkage sketches [1] made implementing graphical techniques for solving kinematic problems easy and fast through trial-and-error procedures. In these software packages, imposing geometric conditions (e.g., coincidence, parallelism, etc.) on points and lines does not require any computation, and redrawing the whole sketch, after the value of a parameter changes, is automatic and fast.

The evolution of solid-modeling software packages led, in the first decade of the twenty-first century, to the rediscovery of graphical techniques as central tools for solving kinematics and kinetostatics problems. In 2006, Kinzel et al. [4] referred to the implementation of graphical techniques for solving kinematic synthesis problems using the sketchers

of CAD software as ‘Geometric Constraint Programming’ (GCP). Since then, a body of literature [5–15] has emerged, focusing on GCP-based solution methods for both classical and novel kinematic synthesis problems.

By extension, the term GCP can be used to indicate the implementation of any graphical technique related to kinematic or kinetostatic problems using any advanced graphical platform [1,16,17]. GeoGebra ([geogebra.org](https://www.geogebra.org) (accessed on 31 October 2024)) [18] is an open-source environment designed for interactively learning and teaching mathematics and related STEM disciplines. It features an object-oriented programming language and a wide array of geometric tools, which can be used for implementing GCP.

In the context of planar mechanism sketches, GeoGebra allows users to insert geometric objects (e.g., points, lines, vectors, etc.) containing variable parameters and then generate additional objects through geometric operations (e.g., intersection of curves, etc.) on the inserted objects. These features are sufficient to build and animate mechanism sketches where the variable parameters allow both the definition of motion inputs (i.e., actuated-joint variables) and the mechanism geometry changes needed for matching design requirements.

Moreover, by defining geometric constraints (e.g., parallelism/perpendicularity between a line and a vector, etc.) among the objects, GeoGebra allows for drawing vector diagrams, tied to the mechanism sketch, which dynamically vary during the mechanism animation or because of a geometric constant change. Such a property is sufficient to draw velocity/acceleration vector diagrams corresponding to the velocity/acceleration loop equations of the mechanism and static-equilibrium vector diagrams of the forces applied to the mechanism’s links. On this point, it is worth noting that sketchers of CAD software are not able to insert vectors [2], which makes building vector diagrams much more difficult/uncertain in those platforms than it is in GeoGebra.

Eventually, GeoGebra contains a long list of predefined functions and algebraic operators that are usable to state (even cumbersome) math relationships among geometric parameters (i.e., analytic constraints).

Drawing a mechanism’s sketch is the prerequisite for implementing, manually or through GCP, any graphical technique. Such an operation involves the translation of the kinematic constraints into geometric relationships among relevant points/lines/segments used to build the sketch. From a conceptual point of view, once the sketch is completed, the position analysis of the mechanism is graphically solved at a particular configuration of the mechanism. Therefore, formulating the position analysis solution as a sequence of geometric operations to implement is the central task to address using either traditional drafting tools or any advanced graphical platform.

The decomposition of a planar linkage into Assur groups [19–22] is a technique proposed in the literature for transforming the position analysis solution of planar linkages into a sequence of modular operations [23–28]. An Assur group is a kinematic chain with zero mobility that does not contain any smaller zero-mobility kinematic chain. Artobolevskii [21] stated that any linkage is obtainable by connecting a number of Assur groups.

The position analysis solution through linkage decomposition starts from the data of the frame and of the input links (driving links), whose poses (positions and orientations) are known. Successively, it solves the position analysis of the Assur groups adjacent to the driving links, and it continues by solving the position analysis of the Assur groups adjacent to the already-solved part of the linkage until covering the whole linkage. It is applicable and has advantages when the solution algorithms of the smaller modules (i.e., the Assur groups) are known and simpler than the solution of the whole linkage. This is true for the Assur groups named dyads (Figure 1), which feature two binary links (links i and j in Figure 1) connected through single-degree-of-freedom (DOF) lower kinematic pairs (i.e., either a revolute (R) pair or a prismatic (P) pair) both to one another at one end and to the remaining part of the linkage at the other end. A large family of planar linkages, which contains the most common and studied ones, are decomposable by using only dyads over driving links and frame.

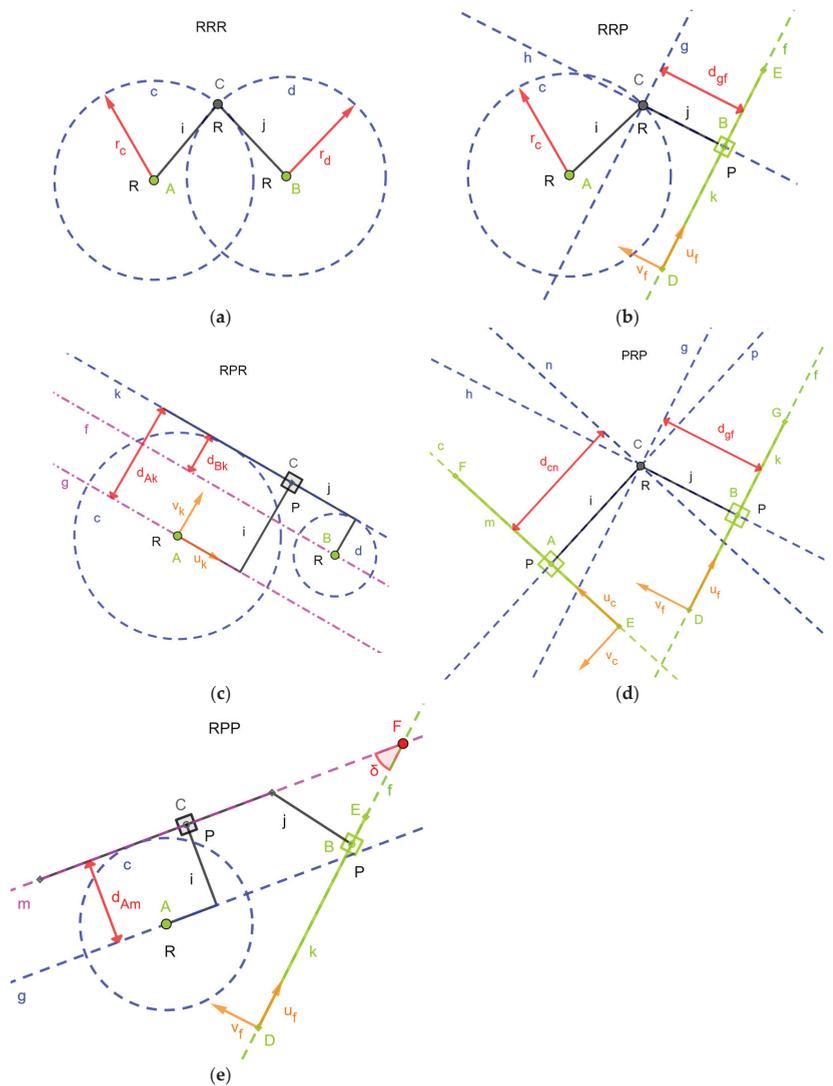


Figure 1. Dyad types (i and j are the two links of the dyad, the green elements are the free endings of the dyad, the red parameters are the geometric constants of the dyad): (a) RRR, (b) RRP, (c) RPR, (d) RRP, and (e) RPP.

This paper addresses the kinematic analysis/synthesis and kinetostatic analysis of this family of linkages using GeoGebra. In particular, firstly, how dyads' kinematic analysis is solvable by using purely geometric conditions, which are easy to implement through GeoGebra, is shown. Secondly, how such geometric solutions allow building parametric sketches of linkages, which can also be animated, and the associated vector diagrams necessary to study/design the linkage from kinematic and kinetostatic points of view, is also addressed. Eventually, a number of case studies is presented to better illustrate the novel GCP methodology. Even though the kinematic analyses/syntheses of some planar linkages with GeoGebra have already been presented, the proposed systematic approach is novel and could guide the implementation of the solutions of the same problems even in

other graphical environments. The results of these works are useful both to mechanism designers and to conceiving a novel teaching methodology for higher education courses.

The paper is organized as follows. Section 2 presents the methodology together with some background materials. Section 3 illustrates a number of relevant case studies. Successively, Section 4 discusses the obtained results and Section 5 draws the conclusions.

2. Materials and Methods

In GeoGebra (see [18] and the options “Manual” or “Tutorial” of the “Help & Feedback” menu of <https://www.geogebra.org/classic> (accessed on 31 October 2024)), geometric object and geometric operations can be introduced either by typing commands in the “Algebra” menu or by choosing on a graphic menu, named “Tools”, the specific object (e.g., point or line symbols) or operation, and, then, clicking, with the mouse, on the position of the Cartesian plane window where the object/operation must be located/executed. Here, for the sake of clarity, the procedures are illustrated through the commands of the “Algebra” menu. The use of the graphic menu “Tools” is intuitive, and the reader can autonomously master it after having understood the commands of the “Algebra” menu.

Also, in GeoGebra, scalar parameters are introducible by typing the command “SliderName = Slider(minimum value, maximum value, increment)” or, simply, “SliderName”. They are, by definition, variable and are usable for animating the drawing (i.e., as joint variable) or for changing the geometry of the introduced objects (i.e., as a design parameter to adjust according to the design requirements).

Eventually, in the Cartesian plane window of GeoGebra, by right-clicking on an already-introduced geometric object, a submenu appears where the option “Show Trace” can be chosen to provide the trace of the object during the animation (e.g., the path of a point of a link during the link motion). In the same submenu, by choosing the option “Settings”, another window appears on the right where all the properties of the object can be modified. In this window, by deselecting the option “Show Object”, the object remains defined, but it is not shown in the Cartesian plane window. Such a choice is useful, when a linkage has multiple assembly modes, to hide all the assembly modes the user is not interested in.

2.1. Building the Linkage and Generating Its Motion

The planar linkages of the family under study contain only dyads, driving links and the frame.

In GeoGebra, modeling the frame corresponds to the introduction of the geometric objects that define the joints connecting the links to the frame. Such joints are either R-pairs or P-pairs. Consequently, for R-pairs, the coordinates of the R-pair centers must be introduced through the command “PointName = Point({x,y})” or, simply, “PointName = (x,y)”, where x and y can be either numeric values or names of scalar parameters. For P-pairs, the lines parallel to the P-pair sliding directions, chosen as joint axis, must be introduced through the command “LineName = Line(<Point>, <Point>)” or “LineName = Line(<Point>, <Direction>)”, where the points belong to the joint axis and the direction is assigned by means of a vector or a line/segment parallel to the joint axis. Since the frame is at rest, all these points and lines have null velocity and acceleration during the linkage animation, even though their position can be modified by using the scalar parameters that have been introduced during the object definitions.

Differently, the introduction of dyads and driving links can be achieved through the short command lists presented below.

2.1.1. Dyads

Considering all the possible combinations of R- and P-pairs in a dyad brings one to the conclusion that there are only five types of dyads (Figure 1): (a) RRR, (b) RRP, (c) RPR, (d) PRP, and (e) RPP. From a kinematic point of view, each type of dyad is considerable as a module whose inputs and outputs are the kinematic data of the two free endings (i.e., the

green elements in Figure 1) and of the intermediate joint between the two links (i.e., the black joint in Figure 1), respectively. For an R-pair, such data refer to the position, velocity and acceleration of the R-pair center, whereas, for a P-pair, they are pose, velocity characteristics (e.g., a given point velocity and the angular velocity) and the acceleration characteristics (e.g., a given point acceleration and the angular acceleration) of the line parallel to the P-pair slider direction that has been chosen as the joint axis. The analytic relationships that express the kinematic data outputs as functions of the kinematic data inputs are known and used in analytical/numerical modular solution algorithms for planar linkages (see [16], for instance). Reinterpreting such relationships in term of geometric conditions easy to implement in GeoGebra is feasible, as follows. In the zipped file “GGB files.zip” uploaded as “Supplementary Materials”, the folder “Dyads” contains the GeoGebra files (extension ggb) with the GeoGebra programs that build the five types of dyads together with their velocity/acceleration vector diagrams. In Appendix A, these GeoGebra programs are reported together with comments that explain what each command line generates in the Cartesian plane window of GeoGebra.

2.1.1.a. RRR Dyad

For RRR dyads, with reference to Figure 1a, the data inputs are the coordinates of the R-pair centers at the free endings, that is, points A and B, and the link lengths, that is, r_c and r_d , whereas the data outputs are the coordinates of the center of the intermediate R-pair, that is, point C.

Since point C must keep a constant distance r_c (r_d) from A (from B) during motion, point C must be located at one of the two possible intersections of two circumferences (c and d in Figure 1a), one centered at A with radius r_c and the other centered at B with radius r_d . Accordingly, the list of GeoGebra commands that implement this geometric condition is reported in Appendix A. Such a GeoGebra program generates in the Cartesian plane window of GeoGebra the RRR dynamic sketch of Figure 1a, where the style properties (colors of lines and points, etc.) of the introduced objects have been modified by selecting the object and clicking on “Settings”, as explained above.

In this draft, the dyad configuration is modifiable by clicking on one free ending A or B and moving the mouse. Also, the dyad geometry is modifiable through the defined “Sliders” r_c and r_d . When an RRR dyad is added to an already-built linkage, points A and B are already defined as linkage points, and the first two commands of the above cited list must be omitted.

2.1.1.b. RRP Dyad

For RRP dyads, with reference to Figure 1b, the data inputs are the coordinates of the R-pair center at the left free ending, that is, point A; the pose data of the P-pair joint axis at the right free ending, that is, line f; the length of link i , that is, r_c , and the distance from line f of the center of the intermediate R-pair, that is, d_{gf} . The data outputs are the coordinates of the center of the intermediate R-pair, that is, point C.

Since point C must keep a constant distance r_c (d_{gf}) from point A (from line f) during motion, point C must be located at one of the two possible intersections between the circumference (c in Figure 1b) centered at A with radius r_c and a line (line g in Figure 1b) parallel to the P-pair joint axis and distant d_{gf} from it. Accordingly, the list of GeoGebra commands that implement this geometric condition is reported in Appendix A. Such a GeoGebra program generates in the Cartesian plane window of GeoGebra the RRP dynamic sketch of Figure 1b, where the style properties (colors of lines and points, etc.) of the introduced objects have been modified by selecting the object and clicking on “Settings”, as explained above.

In this draft, the dyad configuration is modifiable by clicking on one free ending A or D and moving the mouse, and by acting on the defined “Slider” θ to change the slope of line f. Also, the dyad geometry is modifiable through the defined “Sliders” r_c and d_{gf} . When an

RRP dyad is added to an already built linkage, point A and line f are already defined as linkage parts and the first three commands of the above-cited list must be omitted.

2.1.1.c. RPR Dyad

For RPR dyads, with reference to Figure 1c, the data inputs are the coordinates of the R-pair centers at the free endings, that is, points A and B, and their distances from the P-pair joint axis, that is, d_{Ak} and d_{Bk} , whereas the data outputs are the pose of P-pair joint axis, line k, and the position on this axis of the P-pair's slider (i.e., of point C).

Since line k must keep at a constant distance d_{Ak} (d_{Bk}) from A (from B) during motion, line k must be always tangential to two circumferences (c and d in Figure 1c), one centered at A with radius d_{Ak} and the other centered at B with radius d_{Bk} . Once the pose of line k is determined, the pose of the P-pair slider immediately comes out from the geometry of the two links i and j . Accordingly, the list of GeoGebra commands that implement this geometric condition is reported in Appendix A. Such a GeoGebra program generates in the Cartesian plane window of GeoGebra the RPR dynamic sketch of Figure 1c, where the style properties (colors of lines and points, etc.) of the introduced objects have been modified by selecting the object and clicking on "Settings", as explained above.

In this draft, the dyad configuration is modifiable by clicking on one free ending A or B and moving the mouse. Also, the dyad geometry is modifiable through the defined "Sliders" d_{Ak} , d_{Bk} , x_k , and y_k . When an RPR dyad is added to an already built linkage, points A and B are already defined as linkage points, and the first two commands of the above-cited list must be omitted.

2.1.1.d. PRP Dyad

For PRP dyads, with reference to Figure 1d, the data inputs are the pose data of the two P-pair joint axes at the free endings, that is, lines c and f, and the distance from line c (line f) of the center of the intermediate R-pair, that is, d_{cn} (d_{gf}). The data outputs are the coordinates of the center of the intermediate R-pair, that is, point C.

Since point C must keep a constant distance d_{cn} (d_{gf}) from line c (from line f) during motion, point C must be located at the intersection between two lines, one (line n in Figure 1d) parallel to line c and distant d_{cn} from it and the other (line g in Figure 1d) parallel to line f and distant d_{gf} from it. Accordingly, the list of GeoGebra commands that implement this geometric condition is reported in Appendix A. Such a GeoGebra program generates in the Cartesian plane window of GeoGebra the PRP dynamic sketch of Figure 1d, where the style properties (colors of lines and points, etc.) of the introduced objects have been modified by selecting the object and clicking on "Settings", as explained above.

In this draft, the dyad configuration is modifiable by clicking on one free ending E or D and moving the mouse, and by acting on the defined "Slider" θ (φ) to change the slope of line f (line c). Also, the dyad geometry is modifiable through the defined "Sliders" d_{cn} and d_{gf} . When a PRP dyad is added to an already built linkage, lines c and f are already defined as linkage parts, and the first four commands of the above-cited list must be omitted.

2.1.1.e. RPP Dyad

For RPP dyads, with reference to Figure 1e, the data inputs are the coordinates of the R-pair center at the left free ending, that is, point A; the pose data of the P-pair joint axis at the right free ending, that is, line f; the distance, d_{Am} , from A of the joint axis (line m in Figure 1e) of the intermediate P-pair and the angle, δ , that the same joint axis forms with line f. The data outputs are the pose data of the joint axis of the intermediate P-pair, that is, line m, and the positions of the two P-pair sliders on lines f and m.

During motion, line m must keep a constant distance, d_{Am} , from point A, that is, it must be tangential to a circle (circle c in Figure 1e) centered at A with radius d_{Am} , because of the R-pair; and it must keep a constant slope angle, δ , with respect to line f, since lines f and m are both fixed to link j . Accordingly, the list of GeoGebra commands that implement this geometric condition is reported in Appendix A. Such a GeoGebra program generates

in the Cartesian plane window of GeoGebra the RPP dynamic sketch of Figure 1e, where the style properties (colors of lines and points, etc.) of the introduced objects have been modified by selecting the object and clicking on “Settings”, as explained above.

In this draft, the dyad configuration is modifiable by clicking on one free ending A or D and moving the mouse, and by acting on the defined “Slider” θ to change the slope of line f. Also, the dyad geometry is modifiable through the defined “Sliders” d_{Am} and δ , whereas the links’ geometry is modifiable through the defined “Sliders” d_{BF} , x_j and d_j , for link j , and x_i , for link i . When an RPP dyad is added to an already-built linkage, point A and line f are already defined as linkage parts, and the first three commands of the above-cited list must be omitted.

2.1.2. Driving Links

The driving link (link j in Figure 2) is a binary link that at one ending is connected to a link (link i in Figure 2) of the linkage, whose motion is known, through either an actuated R-pair (Figure 2a) or an actuated P-pair (Figure 2b), and at the other ending (free ending) through a non-actuated joint. These two cases can be modeled in GeoGebra as follows. In the zipped file “GGB files.zip” uploaded as “Supplementary Material”, the folder “Driving Link” contains the GeoGebra files (extension ggb) with the GeoGebra programs that build the two types of driving links. In Appendix B, these GeoGebra programs are reported together with comments that explain what each command line generates in the Cartesian plane window of GeoGebra.

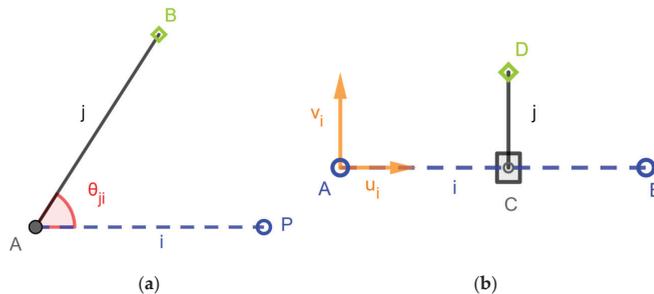


Figure 2. Driving links (link j is the driving link; link i is a link of the studied linkage whose motion is known): (a) driving link with actuated R-pair and (b) driving link with actuated P-pair.

2.1.2.a. Driving Link with Actuated R-pair

For a driving link with actuated R-pair (see Figure 2a), the input data are the value of the actuated-joint variable, that is, the angle θ_{ji} , and the positions of points A and P of link i , whereas the output data are the coordinates of point B, that is, of the free ending of link j . Moreover, the length, say r , of the segment AB is a geometric parameter of link j . Accordingly, the list of GeoGebra commands that generates point B and link j as functions of the data input is reported in Appendix B.

The parametric draft of Figure 2a was generated in GeoGebra through the above-cited command list followed by the adjustment of the style properties (colors of lines and points, etc.) of the introduced objects. When this driving link is added to an already-built linkage, points A and P and link i , together with their motion, are already defined as linkage parts, and the first three commands of the list must be omitted.

2.1.2.b. Driving Link with Actuated P-pair

For a driving link with an actuated P-pair (see Figure 2b), the input data are the value of the actuated joint variable, that is, the length of segment AC, and the positions of points A and B of link i , whereas the output data are the position of the P-pair slider, that is of point C, and the coordinates of the free ending of link j , that is, of point D. Moreover, the distance,

say d , of point D from the segment AB is a geometric parameter of link j . Accordingly, the list of GeoGebra commands that generates point C and link j as functions of the data input is reported in Appendix B.

The parametric draft of Figure 2b was generated in GeoGebra through the above-cited command list followed by the adjustment of the style properties (colors of lines and points, etc.) of the introduced objects. When this driving link is added to an already-built linkage, points A and B and link i , together with their motion, are already defined as linkage parts, and the first three commands of the list must be omitted.

2.1.3. Position Analysis' Geometric Solution

The geometric solution of the position analysis through linkage decomposition starts from the data of the frame and of the driving links adjacent to the frame, and builds them in GeoGebra through the above-described lists of commands. This is always possible since their poses (positions and orientations) and motion are known. Successively, by using the same lists of GeoGebra commands, it builds the dyads/driving links adjacent to the frame and to the already-inserted driving links, and it continues by building the dyads/driving links adjacent to the already-built part of the linkage until completing the GeoGebra sketch of the linkage.

Once the linkage sketch has been completed, all the poses of the links are known, that is, the position analysis has been geometrically solved. Also, the GeoGebra sketch can be animated by selecting "Animation on" in the properties of the "Slider()" used to define the actuated joint variables. Eventually, the geometric constants can be adjusted by selecting "Show object" in the properties of the "Slider()" used to define the geometric constants, and by clicking on the slider that appears in the Cartesian plane window and then moving the mouse. The positions of points assigned with coordinates that are parameters of the object can be moved by simply clicking on the points and then moving the mouse.

2.2. Velocity and Acceleration Analyses' Vector Diagrams

After the linkage's dynamic sketch has been built, the geometric solution of velocity and acceleration analyses, that is, the construction of the velocity and acceleration vector diagrams, proceeds by following the same sequence of substructure analyses used to build the linkage sketch.

Indeed, it starts from the velocity/acceleration data of the frame and of the driving links adjacent to the frame, and draws the velocity/acceleration vectors of their free endings. Successively, it uses these vectors to geometrically determine the velocity/acceleration vectors of the reference points/lines of the intermediate joint (for dyads) or of the output endings (for driving links) of the dyads/driving links adjacent to the frame and to the already solved driving links. Then, it continues by drawing velocity/acceleration vector diagrams of the dyads/driving-links adjacent to the already-solved part of the linkage until covering the whole linkage. Therefore, the modules to repeatedly implement in this methodology are velocity/acceleration analyses of frame, driving links and dyads.

Velocity/acceleration analyses are linear problems. As a consequence, the geometric determination of the output velocity/acceleration vectors involves only intersections of lines and the introduction of dependent variables related to the already-determined vectors by simple analytic formulas. Moreover, the motion input data of a linkage are the time histories of the actuated joint variables, and the frame is assumed either at rest or with known motion. Consequently, the velocity/acceleration analyses of frame and driving links are straightforward; they only involve the introduction of well-known simple analytic formulas reported in textbooks (see, for instance, [29]). For the sake of brevity, such formulas are not recalled, and the following part of this subsection is devoted to the graphical implementation in GeoGebra of the velocity/acceleration analyses of the dyads.

Hereafter, the following notations are used:

- $v_{p,g}$ ($a_{p,g}$) denotes the velocity (acceleration) of point P considered fixed to link g when measured from the frame;

- $v_{P,R,g}$ ($a_{P,R,g}$) denotes the velocity (acceleration) difference $v_{P,g} - v_{R,g}$ ($a_{P,g} - a_{R,g}$);
- $v_{P,gm}$ ($a_{P,gm}$) denotes the velocity (acceleration) of point P considered fixed to link g when measured from link m;
- ω_g (α_g) denotes the signed magnitude, positive if counterclockwise, of the angular velocity (acceleration) of link g when measured from the frame;
- ω_{gm} (α_{gm}) denotes the signed magnitude, positive if counterclockwise, of the angular velocity (acceleration) of link g when measured from link m.

With these notations, the following relationships hold:

$$v_{P,R,g} = \omega_g |P - R| v_g; a_{P,R,g} = \alpha_g |P - R| v_g - \omega_g^2 |P - R| u_g = a_{P,R,\perp,g} + a_{P,R,\parallel,g}; \quad (1a)$$

$$v_{P,gm} = v_{P,g} - v_{P,m}; a_{P,gm} = a_{P,g} - a_{P,m} - 2\omega_m (k \times v_{P,gm}); \quad (1b)$$

$$\omega_{gm} = \omega_g - \omega_m; \alpha_{gm} = \alpha_g - \alpha_m; \quad (1c)$$

where k is a unit vector perpendicular to the sheet and pointing toward the reader, and the following definitions have been introduced:

$$u_g = \frac{P - R}{|P - R|}; v_g = k \times u_g; a_{P,R,\perp,g} = \alpha_g |P - R| v_g; a_{P,R,\parallel,g} = -\omega_g^2 |P - R| u_g. \quad (2)$$

2.2.1. RRR Dyad

Figure 3 shows the reference sketch of the RRR dyad (Figure 3a) previously built in GeoGebra and the vector diagrams of the velocity/acceleration loops (Figure 3b,c) built in GeoGebra, with the lists of commands reported below.

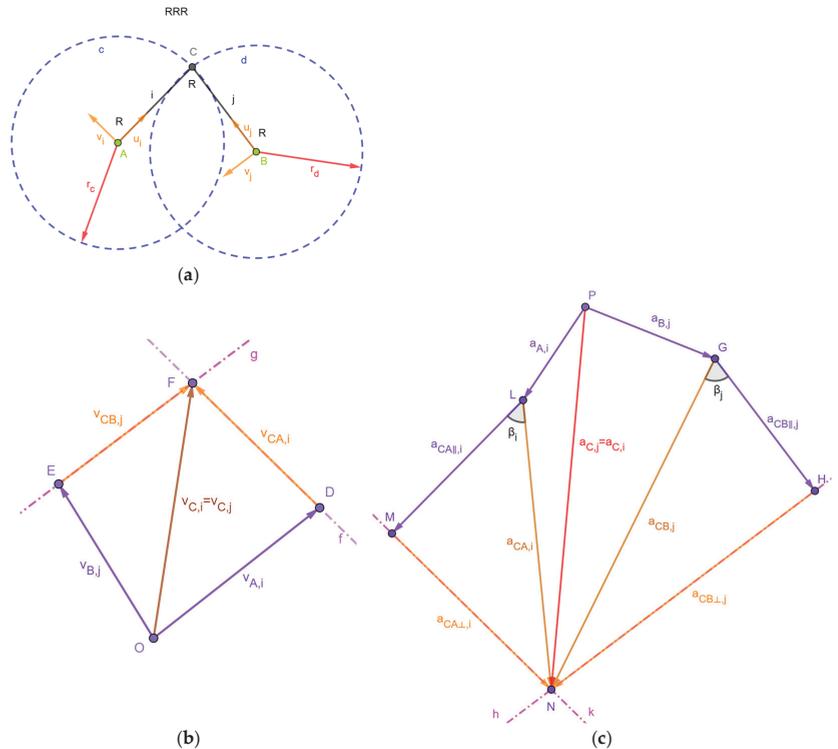


Figure 3. RRR dyad: (a) reference dynamic sketch, (b) vector diagram of the velocity loop associated to the sketch, and (c) vector diagram of the acceleration loop associated to the sketch.

2.2.1.a. Vector Diagram of RRR Dyad's Velocities

The intermediate R-pair with center at C (Figure 3a) brings one to write the following velocity loop equation:

$$v_{C,i} = v_{C,j} \Rightarrow v_{A,i} + v_{CA,i} = v_{B,j} + v_{CB,j} \quad (3)$$

where $v_{A,i}$ and $v_{B,j}$ are the known velocities (i.e., the data inputs) of points A and B, respectively, whereas $v_{CA,i}$ and $v_{CB,j}$ are velocity differences whose directions must be always perpendicular to segment AC and BC (i.e., they have known directions), respectively, and whose signed magnitudes must be determined by solving Equation (3). The vector diagram that geometrically solves Equation (3) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 3b has been generated in GeoGebra by implementing the above-cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point O, and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad, and automatically changes its shape when either the dyad changes its configuration and/or its geometry or the data inputs (i.e., $v_{A,i}$ and $v_{B,j}$) change their values.

2.2.1.b. Vector Diagram of RRR Dyad's Accelerations

The intermediate R-pair with center at C (Figure 3a) brings one to write the following acceleration loop equation:

$$a_{C,i} = a_{C,j} \Rightarrow a_{A,i} + a_{CA,i} = a_{B,j} + a_{CB,j} \Rightarrow a_{A,i} + a_{CA\parallel,i} + a_{CA\perp,i} = a_{B,j} + a_{CB\parallel,j} + a_{CB\perp,j} \quad (4)$$

where $a_{A,i}$ and $a_{B,j}$ are the known accelerations (i.e., the data inputs) of points A and B, respectively, and the vector component $a_{CA\parallel,i}$ ($a_{CB\parallel,j}$) is computable by using the already-computed ω_i (ω_j) and the last of Formulas (2). Also, the vector component $a_{CA\perp,i}$ ($a_{CB\perp,j}$) must always be perpendicular to segment AC (BC), that is, it has a known direction. As a consequence, only the signed magnitudes of $a_{CA\perp,i}$ and $a_{CB\perp,j}$ are unknown and must be determined by solving Equation (4). The vector diagram that geometrically solves Equation (4) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 3c has been generated in GeoGebra by implementing the above-cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point P and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad and automatically changes its shape when either the dyad changes its configuration and/or its geometry or the data inputs (i.e., $a_{A,i}$ and $a_{B,j}$) change their values.

2.2.2. RRP Dyad

Figure 4 shows the reference sketch of the RRP dyad (Figure 4a) previously built in GeoGebra and the vector diagrams of the velocity/acceleration loops (Figure 4b,c) built in GeoGebra, with the lists of commands reported below.

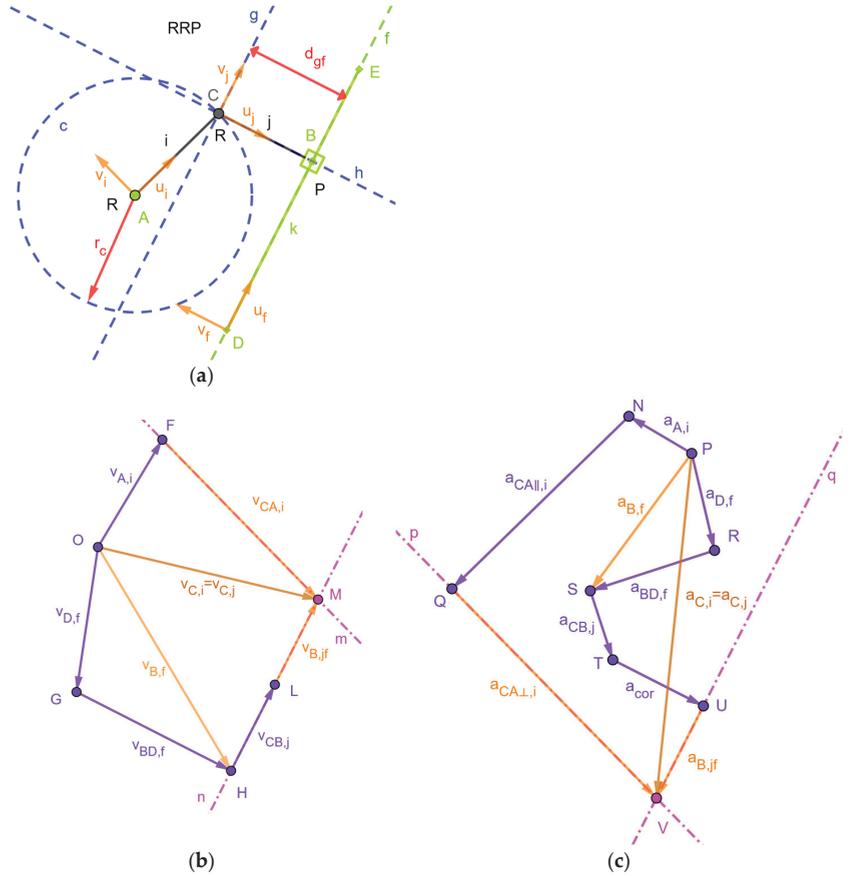


Figure 4. RRP dyad: (a) reference dynamic sketch, (b) vector diagram of the velocity loop associated to the sketch, and (c) vector diagram of the acceleration loop associated to the sketch.

2.2.2.a. Vector Diagram of RRP Dyad’s Velocities

The intermediate R-pair with center at C (Figure 4a) connecting links *j* and *i* and the P-pair with line *f* as the joint axis bring one to write the following velocity loop equation:

$$\left. \begin{aligned}
 v_{C,i} = v_{C,j} &\Rightarrow v_{A,i} + v_{CA,i} = v_{B,j} + v_{CB,j} \\
 \omega_j = \omega_f \\
 v_{B,j} = v_{B,f} + v_{B,jf} \\
 v_{B,f} = v_{D,f} + v_{BD,f}
 \end{aligned} \right\} \Rightarrow v_{A,i} + v_{CA,i} = v_{D,f} + v_{BD,f} + v_{B,jf} + v_{CB,j} \quad (5)$$

where $v_{A,i}$, $v_{D,f}$ and $\omega_f (= \omega_j)$ are known (i.e., they are the data inputs) and the velocity difference $v_{BD,f}$ ($v_{CB,j}$) is computable by using the known ω_f (ω_j) and Formula (2). Also, the velocity difference $v_{CA,i}$ (the relative velocity $v_{B,jf}$) has a known direction since it must be always perpendicular to segment AC (parallel to line *f*). As a consequence, the signed magnitudes of $v_{CA,i}$ and $v_{B,jf}$ are the unknowns of Equation (5) and are determinable by solving Equation (5). The vector diagram that geometrically solves Equation (5) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 4b has been generated in GeoGebra by implementing the above-cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point O, and then moving the mouse. Also, it is

geometrically constrained to the dynamic sketch of the dyad and automatically changes its shape when the dyad changes its configuration and/or its geometry or the data inputs (i.e., $v_{A,i}$, $v_{D,f}$ and ω_f) change their values.

2.2.2.b. Vector Diagram of RRP Dyad’s Acceleration

The intermediate R-pair with center at C (Figure 4a) connecting links j and i , and the P-pair with line f as the joint axis bring one to write the following acceleration loop equation:

$$\left. \begin{aligned} a_{C,i} &= a_{C,j} \Rightarrow a_{A,i} + a_{CA||i} + a_{CA\perp,i} = a_{B,j} + a_{CB,j} \\ \alpha_j &= \alpha_f; a_{cor} = 2\omega_f(k \times v_{B,jf}) \\ a_{B,j} &= a_{B,f} + a_{B,jf} + a_{cor} \\ a_{B,f} &= a_{D,f} + a_{BD,f} \end{aligned} \right\} \Rightarrow a_{A,i} + a_{CA||i} + a_{CA\perp,i} = a_{D,f} + a_{BD,f} + a_{B,jf} + a_{cor} + a_{CB,j} \quad (6)$$

where $a_{A,i}$, $a_{D,f}$ and $\alpha_f (= \alpha_i)$ are known (i.e., they are the data inputs) and the accelerations $a_{BD,f}$, $a_{CB,j}$, a_{cor} , and $a_{CA||i}$ are computable by using ω_f , α_f , ω_j , α_j and $v_{B,jf}$, which are either data inputs or already-computed data, and Formulas (1) and (2). Also, the acceleration difference component $a_{CA\perp,i}$ (the relative acceleration $a_{B,jf}$) has a known direction, since it must be always perpendicular to segment AC (parallel to line f). As a consequence, the signed magnitudes of $a_{CA\perp,i}$ and $a_{B,jf}$ are the unknowns of Equation (6) and are determinable by solving Equation (6). The vector diagram that geometrically solves Equation (6) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 4c has been generated in GeoGebra by implementing the above-cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point P, and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad and automatically changes its shape when either the dyad changes its configuration and/or its geometry, or the data inputs (i.e., $a_{A,i}$, $a_{D,f}$ and α_f) change their values.

2.2.3. RPR Dyad

Figure 5 shows a reference sketch of the RPR dyad (Figure 5a) previously built in GeoGebra and the vector diagrams of the velocity/acceleration loops (Figure 5b,c) built in GeoGebra, with the lists of commands reported below.

2.2.3.a. Vector Diagram of RPR Dyad’s Velocities

The intermediate P-pair joining links i and j (Figure 5a) brings one to write the relationships

$$v_{A,ji} = v_{C,ji} = -v_{C,ij}; \omega_j = \omega_i \quad (7)$$

and the equations

$$\left. \begin{aligned} v_{A,j} &= v_{A,i} + v_{A,ji} \\ v_{A,j} &= v_{B,j} + v_{AB,j} \end{aligned} \right\} \Rightarrow v_{A,i} + v_{A,ji} = v_{B,j} + v_{AB,j} \quad (8a)$$

$$v_{C,j} = v_{A,j} + v_{CA,j} = v_{B,j} + v_{CB,j} \Rightarrow v_{AB,j} = v_{CB,j} - v_{CA,j} \quad (8b)$$

$$v_{C,i} = v_{C,j} + v_{C,ij} \quad (8c)$$

where $v_{A,i}$ and $v_{B,j}$ are the known velocities (i.e., the data inputs) of points A and B, respectively, whereas the relative velocity $v_{A,ji}$ ($=v_{C,ji} = -v_{C,ij}$) must always be parallel to the P-pair joint axis and the velocity differences $v_{AB,j}$, $v_{CA,j}$ and $v_{CB,j}$ must always be perpendicular to segments AB, CA, and CB, respectively. Consequently, only the signed magnitudes of $v_{A,ji}$ and $v_{AB,j}$ are unknown in Equation (8a) and can be determined by solving Equation (8a). Then, by introducing the so-determined $v_{AB,j}$ into Equation (8b), $v_{CA,j}$, $v_{CB,j}$ and $v_{C,j}$ can also be determined and, eventually, the introduction of the so-determined $v_{A,ji}$ ($=v_{C,ji} = -v_{C,ij}$) and $v_{C,j}$ into Equation (8c) gives $v_{C,i}$. This solution

procedure can be geometrically implemented through vector diagrams that can be built in GeoGebra with the list of commands reported in Appendix A.

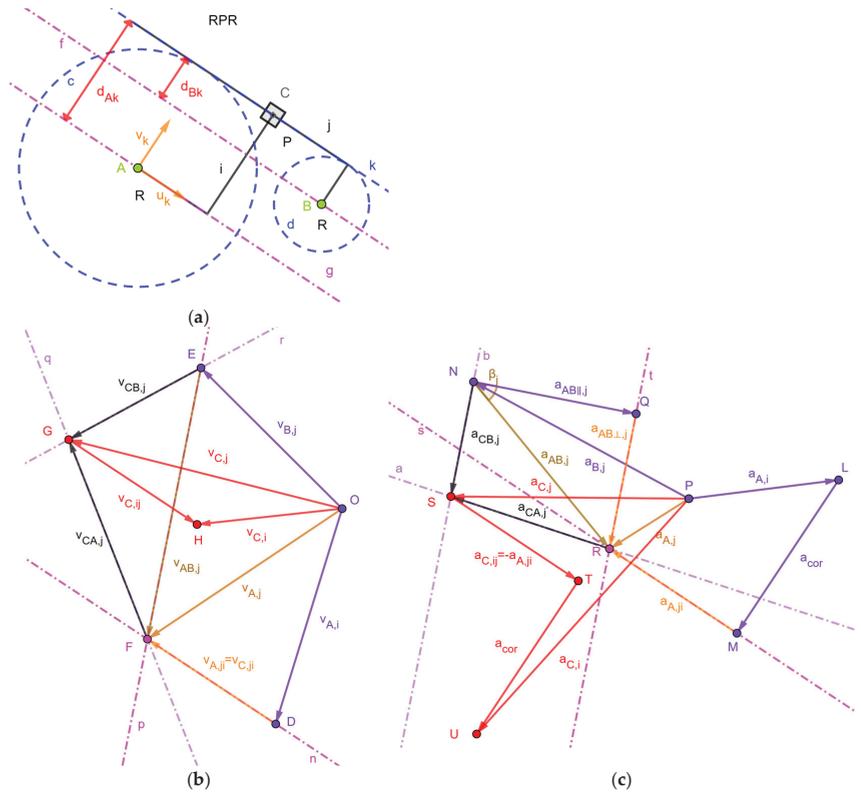


Figure 5. RPR dyad: (a) reference dynamic sketch, (b) vector diagram of the velocity loop associated to the sketch, and (c) vector diagram of the acceleration loop associated to the sketch.

The vector diagram of Figure 5b has been generated in GeoGebra by implementing the above cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point O and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad, and automatically changes its shape when either the dyad changes its configuration and/or its geometry, or the data inputs (i.e., $v_{A,i}$ and $v_{B,j}$) change their values.

2.2.3.b. Vector Diagram of RPR Dyad’s Acceleration

The intermediate P-pair joining links i and j (Figure 5a) brings one to write the following relationships

$$a_{A,ji} = a_{C,ji} = -a_{C,ij}; \alpha_j = \alpha_i \tag{9}$$

and the following equations

$$\left. \begin{aligned} a_{A,j} &= a_{A,i} + a_{A,ji} + a_{cor} \\ a_{cor} &= 2\omega_i(k \times v_{A,ji}) \\ a_{A,j} &= a_{B,j} + a_{AB,j} \\ a_{AB,j} &= a_{AB\perp,j} + a_{AB||,j} \\ a_{AB||,j} &= -\omega_j^2(A - B) \end{aligned} \right\} \Rightarrow a_{A,i} + a_{A,ji} + a_{cor} = a_{B,j} + a_{AB\perp,j} + a_{AB||,j} \tag{10a}$$

$$\left. \begin{aligned} a_{C,j} &= a_{A,j} + a_{CA,j} = a_{B,j} + a_{CB,j} \\ \alpha_j &= \frac{a_{AB\perp,j} \cdot [k \times (A-B)]}{\|A-B\|^2} \\ \beta_j &= \arctan\left(\frac{\alpha_j}{\omega_j^2}\right) \end{aligned} \right\} \Rightarrow a_{AB,j} = a_{CB,j} - a_{CA,j} \quad (10b)$$

$$a_{C,i} = a_{C,j} + a_{C,ij} + a_{cor} \quad (10c)$$

where $a_{A,i}$ and $a_{B,j}$ are the known accelerations (i.e., the data inputs) of points A and B, respectively, and a_{cor} ($a_{AB\parallel,j}$) is computable from the velocity analysis results. Moreover, the relative acceleration $a_{A,ji}$ ($=a_{C,ji} = -a_{C,ij}$) must always be parallel to the P-pair joint axis and the acceleration differences $a_{AB,j}$, $a_{CA,j}$ and $a_{CB,j}$ must always form the angle β_j with segments AB, CA, and CB, respectively. Consequently, only the signed magnitudes of $a_{A,ji}$ and $a_{AB\perp,j}$ are unknown in Equation (10a), and can be determined by solving Equation (10a). Then, introducing the so-determined $a_{AB\perp,j}$ into Equation (10b) brings one firstly to compute α_j and β_j , and then, to determine $a_{CA,j}$, $a_{CB,j}$ and $a_{C,j}$. Eventually, the introduction of the so-determined $a_{A,ji}$ ($=a_{C,ji} = -a_{C,ij}$) and $a_{C,j}$ into Equation (10c) gives $a_{C,i}$. This solution procedure can be geometrically implemented through vector diagrams that can be built in GeoGebra with the list of commands reported in Appendix A.

The vector diagram of Figure 5c has been generated in GeoGebra by implementing the above-cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point P, and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad, and automatically changes its shape when the dyad changes its configuration and/or its geometry or the data inputs (i.e., $a_{A,i}$ and $a_{B,j}$) change their values.

2.2.4. PRP Dyad

Figure 6 shows the reference sketch of the PRP dyad (Figure 6a) previously built in GeoGebra and the vector diagrams of the velocity/acceleration loops (Figure 6b,c) built in GeoGebra, with the lists of commands reported below.

2.2.4.a. Vector Diagram of PRP Dyad’s Velocities

The intermediate R-pair with center at C (Figure 6a) connecting links j and i and the P-pairs with line f and line c as joint axes bring one to write the following velocity loop equation:

$$\left. \begin{aligned} v_{C,i} &= v_{C,j} \Rightarrow v_{A,i} + v_{CA,i} = v_{B,j} + v_{CB,j} \\ \omega_j &= \omega_f; \omega_i = \omega_c; \\ v_{B,j} &= v_{B,f} + v_{B,jf}; v_{A,i} = v_{A,c} + v_{A,ic}; \\ v_{B,f} &= v_{D,f} + v_{BD,f}; v_{A,c} = v_{E,c} + v_{AE,c}; \end{aligned} \right\} \Rightarrow v_{E,c} + v_{AE,c} + v_{A,ic} + v_{CA,i} = v_{D,f} + v_{BD,f} + v_{B,jf} + v_{CB,j} \quad (11)$$

where $v_{E,c}$, $v_{D,f}$, ω_c ($=\omega_i$) and ω_f ($=\omega_j$) are known (i.e., they are the data inputs) and the velocity differences $v_{BD,f}$ and $v_{CB,j}$ ($v_{AE,c}$ and $v_{CA,i}$) are computable by using the known ω_f (ω_c) and Formulas (2). Also, the relative velocity $v_{A,ic}$ (the relative velocity $v_{B,jf}$) has a known direction since it must always be parallel to line c (to line f). As a consequence, the signed magnitudes of $v_{A,ic}$ and $v_{B,jf}$ are the unknowns of Equation (11) and are determinable by solving Equation (11). The vector diagram that geometrically solves Equation (11) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 6b has been generated in GeoGebra by implementing the above-cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point O and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad and automatically changes

its shape when either the dyad changes its configuration and/or its geometry, or the data inputs (i.e., $v_{E,c}$, $v_{D,f}$, ω_c and ω_f) change their values.

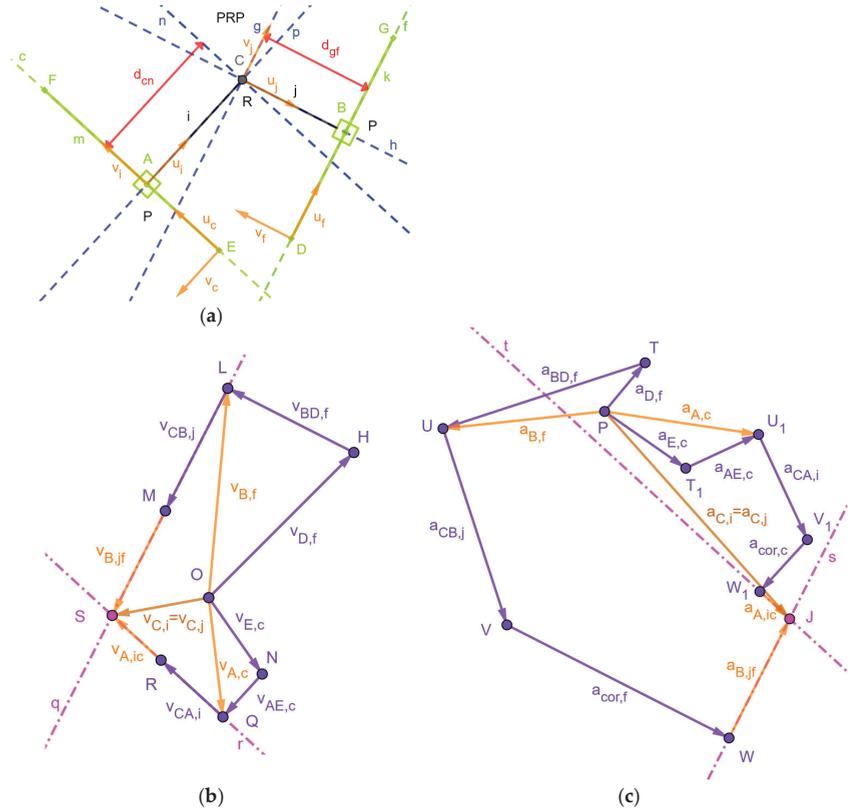


Figure 6. PRP dyad: (a) reference dynamic sketch, (b) vector diagram of the velocity loop associated to the sketch, and (c) vector diagram of the acceleration loop associated to the sketch.

2.2.4.b. Vector Diagram of PRP Dyad’s Acceleration

The intermediate R-pair with center at C (Figure 6a) connecting links j and i and the P-pairs with line c and f as joint axes bring one to write the following acceleration loop equation:

$$\left. \begin{aligned}
 a_{C,i} &= a_{C,j} \Rightarrow a_{A,i} + a_{CA,i} = a_{B,j} + a_{CB,j} \\
 \alpha_j &= \alpha_f; a_{cor,f} = 2\omega_f(k \times v_{B,jf}) \\
 a_{B,j} &= a_{B,f} + a_{B,jf} + a_{cor,f} \\
 a_{B,f} &= a_{D,f} + a_{BD,f} \\
 \alpha_i &= \alpha_c; a_{cor,c} = 2\omega_c(k \times v_{A,ic}) \\
 a_{A,i} &= a_{A,c} + a_{A,ic} + a_{cor,c} \\
 a_{A,c} &= a_{E,c} + a_{AE,c}
 \end{aligned} \right\} \Rightarrow a_{E,c} + a_{AE,c} + a_{A,ic} + a_{cor,c} + a_{CA,i} = a_{D,f} + a_{BD,f} + a_{B,jf} + a_{cor,f} + a_{CB,j} \quad (12)$$

where $a_{E,c}$, $a_{D,f}$, $\alpha_c (= \alpha_i)$ and $\alpha_f (= \alpha_j)$ are known (i.e., they are the data inputs) and the accelerations $a_{BD,f}$, $a_{CB,j}$, $a_{cor,f}$, $a_{AE,c}$, $a_{CA,i}$ and $a_{cor,c}$ are computable using ω_f , α_f , ω_c , α_c , $v_{B,jf}$, and $v_{A,ic}$, which are either data inputs or already-computed data, along with Formulas (1) and (2). Also, the relative acceleration $a_{A,ic}$ ($a_{B,jf}$) has a known direction since it must always be parallel to line c (line f). As a consequence, the signed magnitudes of $a_{A,ic}$ and $a_{B,jf}$ are the unknowns of Equation (12) and are determinable by solving Equation (12).

The vector diagram that geometrically solves Equation (12) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 6c has been generated in GeoGebra by implementing the above-cited list of commands and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point P and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad and automatically changes its shape when either the dyad changes its configuration and/or its geometry or the data inputs (i.e., $a_{E,C}$, $a_{D,f}$, α_f and α_f) change their values.

2.2.5. RPP Dyad

Figure 7 shows the reference sketch of the RPP dyad (Figure 7a) previously built in GeoGebra and the vector diagrams of the velocity/acceleration loops (Figure 7b,c) built in GeoGebra, with the list of commands reported below.

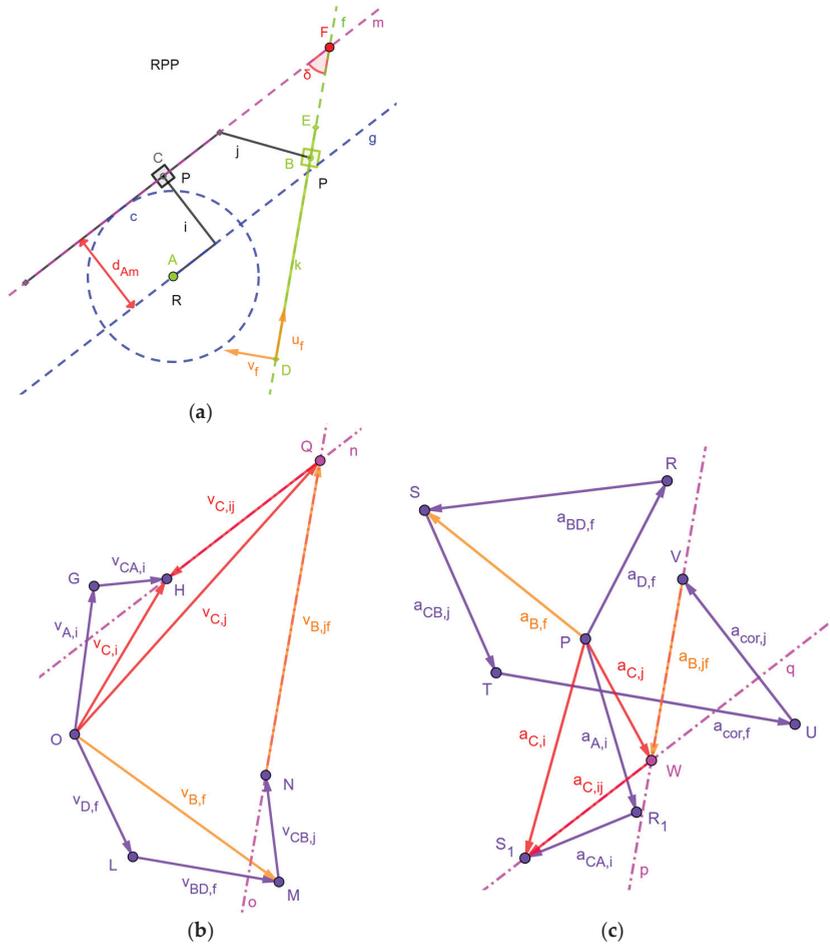


Figure 7. RPP dyad: (a) reference dynamic sketch, (b) vector diagram of the velocity loop associated to the sketch, and (c) vector diagram of the acceleration loop associated to the sketch.

2.2.5.a. Vector Diagram of RPP Dyad’s Velocities

The intermediate P-pair with line m as joint axis (Figure 7a) connecting links *j* and *i* and the P-pair with line f as joint axis bring one to write the following velocity loop equation

$$\left. \begin{aligned} v_{C,i} &= v_{C,j} + v_{C,ij} \Rightarrow v_{A,i} + v_{CA,i} = v_{B,j} + v_{CB,j} + v_{C,ij} \\ \omega_i &= \omega_j = \omega_f \\ v_{B,j} &= v_{B,f} + v_{B,jf} \\ v_{B,f} &= v_{D,f} + v_{BD,f} \end{aligned} \right\} \Rightarrow v_{A,i} + v_{CA,i} = v_{D,f} + v_{BD,f} + v_{B,jf} + v_{CB,j} + v_{C,ij} \quad (13)$$

where $v_{A,i}$, $v_{D,f}$ and $\omega_f (= \omega_i = \omega_j)$ are known (i.e., they are the data inputs) and the velocity differences $v_{CA,i}$, $v_{BD,f}$, and $v_{CB,j}$ are computable by using the known $\omega_f (= \omega_i = \omega_j)$ along with Formulas (2). Also, the relative velocity $v_{C,ij}$ ($v_{B,jf}$) has a known direction since it must be always parallel to line m (parallel to line f). As a consequence, the signed magnitudes of $v_{C,ij}$ and $v_{B,jf}$ are the unknowns of Equation (13), and are determinable by solving Equation (13). The vector diagram that geometrically solves Equation (13) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 7b has been generated in GeoGebra by implementing the above-cited list of commands, and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point O and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad and automatically changes its shape when either the dyad changes its configuration and/or its geometry or the data inputs (i.e., $v_{A,i}$, $v_{D,f}$ and ω_f) change their values.

2.2.5.b. Vector Diagram of RPP Dyad’s Acceleration

The intermediate P-pair with line m as joint axis (Figure 7a) connecting links *j* and *i* and the P-pair with line f as joint axis bring one to write the following acceleration loop equation:

$$\left. \begin{aligned} a_{C,i} &= a_{C,j} + a_{C,ij} + a_{cor,j} \Rightarrow a_{A,i} + a_{CA,i} = a_{B,j} + a_{CB,j} + a_{C,ij} + a_{cor,j} \\ a_{B,j} &= a_{B,f} + a_{B,jf} + a_{cor,f} \\ a_{B,f} &= a_{D,f} + a_{BD,f} \\ a_{cor,j} &= 2\omega_j(k \times v_{C,ij}); a_{cor,f} = 2\omega_f(k \times v_{B,jf}) \\ \omega_i &= \omega_j = \omega_f; \alpha_i = \alpha_j = \alpha_f \end{aligned} \right\} \Rightarrow a_{A,i} + a_{CA,i} = a_{D,f} + a_{BD,f} + a_{B,jf} + a_{cor,f} + a_{CB,j} + a_{C,ij} + a_{cor,j} \quad (14)$$

where $a_{A,i}$, $a_{D,f}$ and $\alpha_f (= \alpha_i = \alpha_j)$ are known (i.e., they are the data inputs). The acceleration differences $a_{CA,i}$, $a_{BD,f}$, and $a_{CB,j}$ together with the Coriolis accelerations $a_{cor,j}$ and $a_{cor,f}$ are computable by using $\omega_f (= \omega_i = \omega_j)$, $\alpha_f (= \alpha_i = \alpha_j)$, $v_{B,jf}$, and $v_{C,ij}$, which are either data inputs or already-computed data, and Formulas (1) and (2). Also, the relative acceleration $a_{C,ij}$ ($a_{B,jf}$) has a known direction since it must be always parallel to line m (parallel to line f). As a consequence, the signed magnitudes of $a_{C,ij}$ and $a_{B,jf}$ are the unknowns of Equation (14) and are determinable by solving Equation (14). The vector diagram that geometrically solves Equation (14) in GeoGebra can be generated with the list of commands reported in Appendix A.

The vector diagram of Figure 7c has been generated in GeoGebra by implementing the above-cited list of commands and then adjusting the properties (i.e., colors, captions, etc.) of the introduced objects. The so-generated vector diagram can be rigidly moved on the Cartesian plane window by clicking on point P and then moving the mouse. Also, it is geometrically constrained to the dynamic sketch of the dyad, and automatically changes its shape when either the dyad changes its configuration and/or its geometry or the data inputs (i.e., $a_{A,i}$, $a_{D,f}$ and α_f) change their values.

2.3. Kinetostatics Analysis’ Vector Diagram

Kinetostatics analysis (KA) of a linkage is the determination of the generalized torques, applied by the actuators in the actuated joints, when the external forces applied to the linkage and the linkage motion (i.e., the inertial forces) are known. The classic approaches for solving this problem are two: (a) the free-body method and (b) the virtual work

principle. Both these approaches are graphically implementable when the mechanism sketch is available, the first one using free-body diagrams [29] and the latter with active-load diagrams [30–32]. Since both the methods require only sums of vectors and intersections of lines, they can be easily implemented in GeoGebra.

2.3.1. Free-Body Method

The graphic solution of the free-body method is based on graphically solving the force equilibrium equation of each link, when separated from the rest of the linkage and loaded by the external forces and the constraint reactions coming from the joints, after having satisfied the moment equilibrium equation by imposing geometric conditions on the line of actions of those forces. In particular, the conditions that make the moment equilibrium equation satisfied [29] are as follows:

- (a) in the case of only two forces, if the two forces share the same line of action;
- (b) in the case of two forces and a pure moment, if the two forces have parallel and non-coincident lines of action;
- (c) in the case of only three forces, if the lines of action of the three forces share a common intersection point;
- (d) in the case of four forces, if the four forces, separated into two subsystems of two forces, gives two resultants, one for each subsystem, that are aligned along the line passing through the two intersections of the two lines of action of each subsystem.

The analysis of the joint types provides the known pieces of information regarding the action line of the constraint reaction forces, that is, in ideal constraints, the center of an R-pair (the direction perpendicular to the P-pair sliding direction) belongs to (is the direction of) the line of action of the R-pair's (P-pair's) constraint reaction force.

It is worth remembering that a planar system of forces with non-null resultant (with null resultant and non-null resultant moment) can always be reduced to a unique force on the system's central axis (to a unique pure moment).

This method is always applicable to directly solve the linkage's static analysis when only one link is loaded by either one external force, which replaces a general system of planar forces with non-null resultant, or by one pure moment, which replaces a general system of planar forces with null resultant and non-null resultant moment. Also, it is able to indirectly solve any case of a linkage static analysis with the superposition principle, that is, by summing up the results obtained when only one loaded link at a time is analyzed.

2.3.2. Method Based on Active Load Diagrams

The method based on active-load diagrams has been recently proposed [30–32]. This method analyzes any multi-degrees-of-freedom (DOF) linkage passing through the single-DOF linkages generated from it by locking all the actuated joints but one. In doing so, it uses the velocity coefficients, expressed using the instant centers (ICs), of the so-generated single-DOF linkages together with suitable moment arms to write the geometric and the analytic formulas that give the generalized torques. The interested reader can refer to [30–32] for details.

Since, in the vast majority of single-DOF linkages, the IC locations are geometrically determinable by intersecting suitable couples of lines [32] that move with the mechanism sketch, this approach is easily implementable in GeoGebra after the dynamic sketch of the linkage has been built.

3. Results

In this section, the systematic approach presented above to build dynamic sketches of planar linkages and to study their kinematics and kinetostatics is applied to three case studies: a) generation of coupler curves, b) four-bar kinetostatics, and c) kinematic analysis of shaper mechanisms. In the zipped file "GGB files.zip" uploaded as "Supplementary Material", the folder "Case Studies" contains the GeoGebra files (extension ggb) with the GeoGebra programs of the three case studies. In Appendix C, these GeoGebra programs

are reported together with comments that explain what each command line generates in the Cartesian plane window of GeoGebra.

3.1. Generation of Coupler Curves

Four-bar linkages' coupler curves play a role in designing four-bar and six-bar (Stephenson) linkages that satisfy a number of design requirements (one or two dwells, path with an about-linear arc, etc.). In [2], the graphical procedures for solving these design problems using coupler curves are illustrated; all those graphical procedures can be implemented in GeoGebra after having built a four-bar dynamic sketch that parametrically generates the coupler curves.

With the above illustrated methodology, such a sketch is buildable in GeoGebra by recognizing that a four-bar linkage is obtained through the assemblage of one driving link (the crank) with actuated R-pair (Figure 2a) and one dyad of RRR type (Figure 1a) constituted by the coupler and the rocker of the four bar. Also, in GeoGebra, the coupler point, say E, that traces the coupler curve is parametrically generable, for instance, as follows. Two mutually perpendicular unit vectors, say u_c and v_c , fixed to the coupler are introduced, and the position of point E is defined as $E = C + x_c * u_c + y_c * v_c$, where C is the moveable ending of the crank, whereas x_c and y_c are two scalar parameters defined through "Slider" commands. Eventually, by setting "Show Trace" in the properties of point E, when the crank is animated, the coupler curve is generated. The commented list of GeoGebra commands that build this dynamic sketch is reported in Appendix C.

Figure 8 shows the dynamic sketch and the generated coupler curve together with the "Sliders" necessary to change the mobile links' geometry and to animate the sketch. In this sketch, the frame geometry can be modified by clicking on point A or B and moving the mouse.

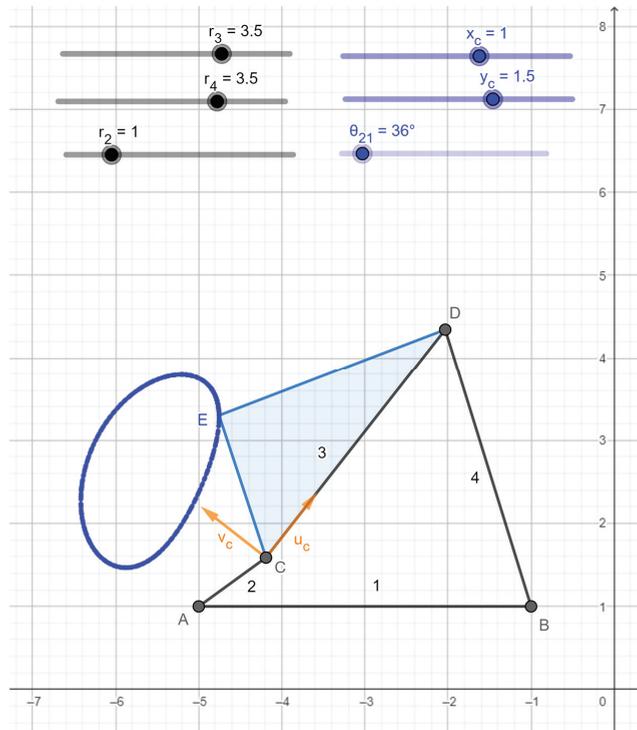


Figure 8. Coupler curve generated in GeoGebra.

3.2. Four-Bar Kinetostatics

A four-bar dynamic sketch, built as illustrated in the previous subsection, can be exploited to make graphical analyses of four-bar linkages' kinetostatics. Indeed, the mass distribution data of the mobile links can be assigned through "Slider" commands; moreover, the barycenter accelerations and the angular accelerations of the same links can be graphically determined with the proposed methodology and successively used to apply the inertia forces to the mobile links; eventually, the kinetostatic analysis of the so-loaded four bar can be graphically solved. In GeoGebra, such a procedure is implementable, for instance, with the list of commands reported in Appendix C. Hereafter, F_{ij} denotes the constraint reaction force applied by link i to link j , while F_{ij-n} denotes the contribution to F_{ij} due to the inertial and external load applied to link n .

Figure 9 shows a dynamic sketch and the velocity/acceleration diagrams generated in GeoGebra for the case of the constant angular velocity of the crank ($\omega_2 = 2\pi/3$ rad/s and $\alpha_2 = 0$ rad/s²) at the instant of motion $t = 0.5$ s (all the measurement units are in SI).

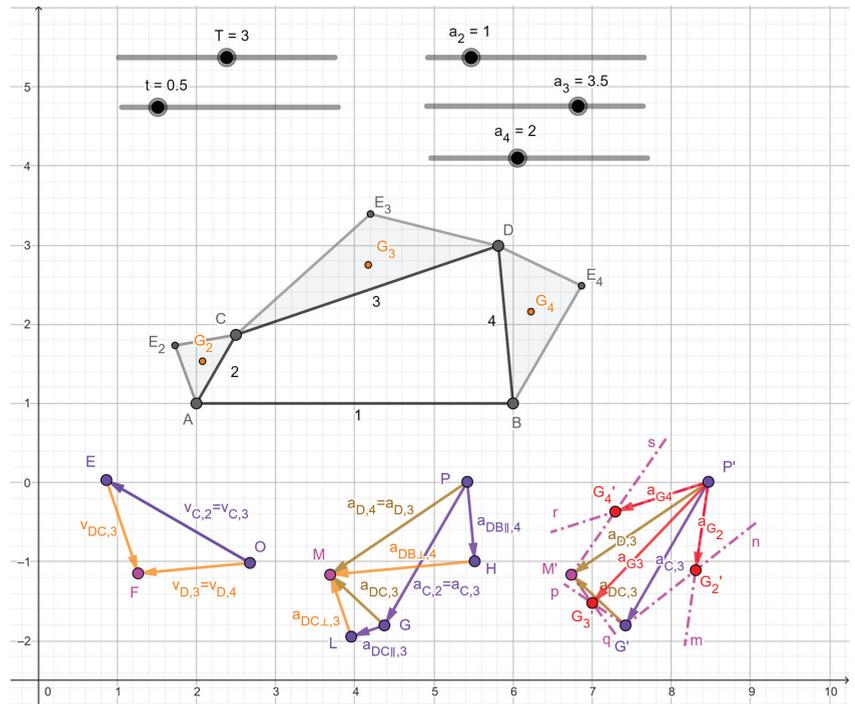


Figure 9. Four-bar linkage: dynamic sketch and velocity/acceleration diagrams generated in GeoGebra for the case of constant angular velocity of the crank ($\omega_2 = 2\pi/3$ rad/s and $\alpha_2 = 0$ rad/s²) at the instant of motion $t = 0.5$ s (all the measurement units are in SI).

Figure 10 shows the geometric solution of the kinetostatic analysis of four-bar linkages, at the instant of motion $t = 0.54$ s (all the measurement units are in SI), built in GeoGebra with the above-cited list of commands, after having adjusted the properties (i.e., colors, captions, etc.) of the introduced objects in their "Settings" menu. The motion inputs are as follows: the crank rotates at a constant angular velocity ($\omega_2 = 2\pi/3$ rad/s and $\alpha_2 = 0$ rad/s²). The static inputs are as follows: only the inertial forces load the links. M_{12} is the generalized torque that must be computed; it is applied by the unique actuator mounted in the R-pair centered at A. On the linkage sketch, the blue (green (violet)) lines of action and forces refer to the case in which only link 3 (link 4 (link 2)) is loaded, whereas the

red forces are the resultants obtained through the superposition principle. The sketch is animated by the “Slider” t , which is the time in seconds. The zipped file uploaded as “Supplementary Material” contains the video “4-bar_KinetostaticsAnimation.wmv” that shows this animation. The mobile links’ lengths can be modified through the “Slider”’s a_2 (for link 2), a_3 (for link 3), and a_4 (for link 4). The frame geometry can be modified by clicking on point A (point B) and then moving the mouse.

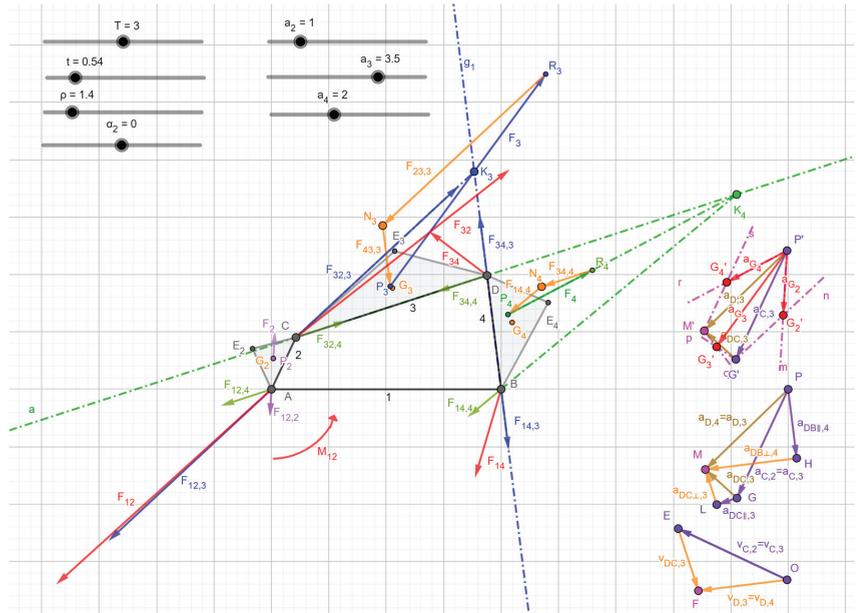


Figure 10. General geometric solution of the kinetostatic analysis of four-bar linkages, at the instant of motion $t = 0.54$ s (all the measurement units are in SI), implemented in GeoGebra in the case in which the crank rotates at constant angular velocity ($\omega_2 = 2\pi/3$ rad/s and $\alpha_2 = 0$ rad/s²) and the R-pair centered at A is actuated. Only the inertial loads are applied to the links. On the linkage sketch, the blue (green (violet)) lines of action and forces refer to the case in which only link 3 (link 4 (link 2)) is loaded, whereas the red forces are the resultants obtained through the superposition principle.

In GeoGebra (version geogebra.org/classic), one “SpreadSheet” window and a second graphic window, named “Graphic2”, can be generated. The values assumed during animation by the introduced independent or dependent variables are recordable on the “SpreadSheet” window, and then are usable to generate a polyline (“Polyline” command) in the window “Graphic2”, which represents their diagrams. Figure 11 shows the diagram of the generalized torque M_{12} as a function of time, t , during one full cycle of crank motion, that is, for t ranging from 0 s to 3 s, generated in GeoGebra with this procedure (the recorded variables are t and M_{12} , and the polyline refers to the sequence of points with coordinates (t, M_{12})).

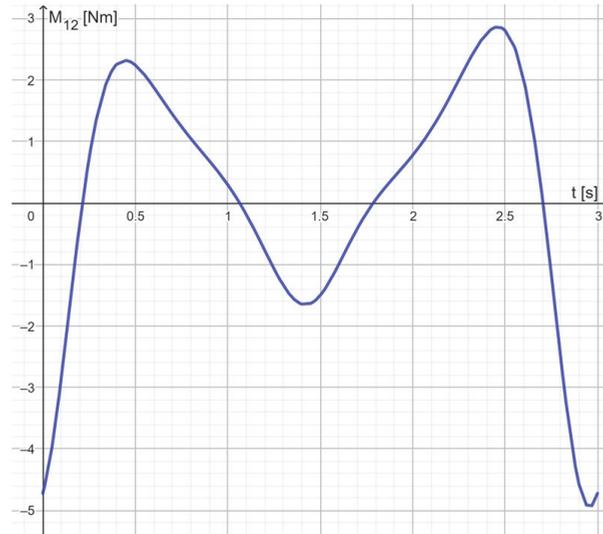


Figure 11. Diagram of the generalized torque M_{12} as a function of time, t , during one full cycle of crank motion, that is, for t ranging from 0 s to 3 s, generated in GeoGebra.

3.3. Shaper Mechanism's Kinematic Analysis

A shaper mechanism (see Figure 12) is a single-DOF two-looped linkage obtainable by assembling in sequence one driving link, the frame and two dyads, one of RPR type and the other of RRP type. The driving link is a crank (link 2 in Figure 12) hinged to the frame (link 1 in Figure 12) at one ending (point A in Figure 12) through an actuated R-pair, and to the RPR dyad (the one constituted by links 3 and 4 in Figure 12) on the other ending (point C in Figure 12) through a passive R-pair. The RPR dyad, at its other ending (point B in Figure 12), is joined to the frame. Also, the same RPR dyad, through a third passive R-pair (the one centered at point D in Figure 12), is joined to the RRP dyad (the one constituted by links 5 and 6 in Figure 12). The RRP dyad closes the second loop of the mechanism through its P-pair (the one with horizontal sliding direction in Figure 12) that joins it to the frame.

Figure 12 shows the dynamic sketch and the velocity/acceleration diagrams generated in GeoGebra for the case of constant angular velocity of the crank ($\omega_2 = 1 \text{ rad/s}$ and $\alpha_2 = 0 \text{ rad/s}^2$) at the instant of motion $t = 4.27 \text{ s}$ (all the measurement units are in SI). With reference to Figure 12, the geometric parameters, introduced through “Slider” commands, are the following ones: a_1 , a_2 , a_4 , and a_5 are the lengths of segments AB, AC, BD, and DE, respectively, and b_1 is the distance of point A from the black dash-dot line. Also, the slider T assigns the motion period (i.e., $\omega_2 = 2\pi/T \text{ rad/s}$) in seconds and the sketch is animated through the slider t , which is the time in seconds. The zipped file uploaded as “Supplementary Material” contains the video “ShaperMechanismAnimation.wmv” that shows this animation. The list of GeoGebra commands used to generate the dynamic sketch and the velocity/acceleration diagrams of Figure 12 are reported in Appendix C.

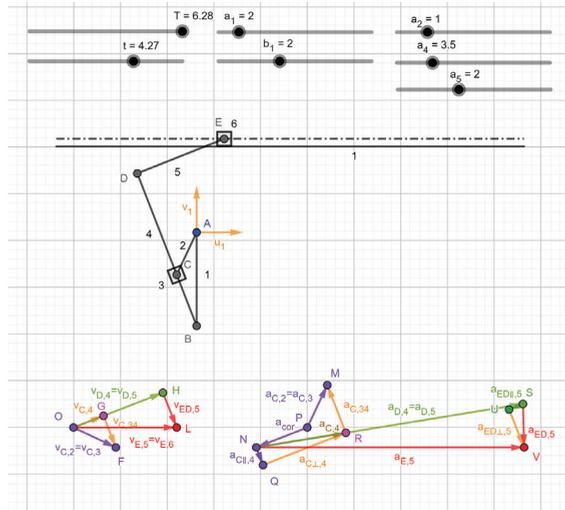


Figure 12. Shaper mechanism: dynamic sketch and velocity/acceleration diagrams generated in GeoGebra for the case of constant angular velocity of the crank ($\omega_2 = 1 \text{ rad/s}$ and $\alpha_2 = 0 \text{ rad/s}^2$) at the instant of motion $t = 4.27 \text{ s}$ (all the measurement units are in SI).

4. Discussion

The presented technique for building dynamic sketches in GeoGebra is applicable to all the planar linkages decomposable into driving links, frame and dyads. This approach is extendable to planar mechanisms that also contain other types of Assur groups (e.g., triads, etc.) provided that the position analysis of those other groups be solved in analytical form. Indeed, for complex Assur groups, the geometric construction starting from the free endings’ data is cumbersome and needs to be bypassed by directly inserting formulas. Once the dynamic sketch has been built, either geometrically or with specific formulas, all the other (velocity/acceleration/kinetostatic) analyses are geometrically implementable since they always correspond to the solution of linear problems (i.e., they only need to find the intersections of suitable lines).

The presented case studies show that the systematic fully graphic solution of kinematic and kinetostatic problems based on dynamic sketches of mechanisms with dyads is able to provide pieces of information useful during design that are not easy to extract from purely analytic/numeric approaches. In particular, it is worth noting that the diagrams of Figure 10 immediately show how to change the external loads on the links to modify particular components of the constraint reactions, and how accelerations directly influence the loads on the links. Indeed (see Figure 10), the acceleration diagrams generate the loads on the links, which, for each link, are reduced to one resultant force lying on the central axis. Then, the static analysis of the mechanism is graphically solved considering only one force at a time, which generates the green, the blue and the violet action lines and constraint reactions for F_4 , F_3 , and F_2 , respectively. Eventually, by using the superposition principle, the actual (red) constraint reactions and generalized torque are determined by summing up their green, blue and violet components.

The usefulness of these diagrams is mainly related to the fact that they are constrained to follow the mechanism motion. In fact, their manual construction was not suitable to make them an efficient design tool, as they are when they automatically follow a dynamic sketch whose geometry can also be varied. As far as these authors are aware, this is the first time that the fully graphical solution, tied to a dynamic sketch, of four bars’ kinetostatic analysis has been presented; also, the adopted methodology is general, and can be used for any linkage decomposable into dyads, frame and driving links.

Regarding linkages' synthesis, the possibility of tracing the paths of points (e.g., the coupler curves built in Section 3.1) and of other geometric objects during the mechanism animation allows the implementation of all the synthesis techniques that use those paths [2]. Also, since the eight geometric constraint tools listed in [2] as necessary to implement all the GCP synthesis procedures reported in the literatures are present in GeoGebra, together with the possibility of writing equations [4], all these synthesis techniques can also be implemented in GeoGebra.

Differently from CAD platforms, vectors and captions are easy to generate in GeoGebra, which greatly facilitates building vector diagrams and finding graphical solutions of vector equations.

5. Conclusions

Geometric constraint programming (GCP) has enabled us to rediscover graphic techniques for solving the kinematics/kinetostatics analysis/synthesis of planar linkages. Such techniques, when easy to implement, are superior with respect to analytic/numeric techniques both in a design context and in the higher-education didactic. Indeed, over the numeric solution of the specific problem, they provide a clear physical representation of linkages' behavior that helps designers to satisfy design requirements and students to reach a deeper comprehension of linkages' mechanics.

The use of GeoGebra for implementing GCP has been explored in this work. In particular, the systematic construction of dynamic sketches of linkages composed by frame, driving links and dyads has been presented together with how to use them for graphically solving kinematics and kinetostatics problems.

This study proves that GeoGebra has all the tools necessary for implementing any GCP synthesis procedure already presented in the literature for this type of linkage, and for building any vector diagram that solves kinematic/kinetostatic analysis problems. This result is useful both to designers and in mechanical engineering higher education.

The three case studies, which have been illustrated, show the effectiveness of the proposed approach and of GeoGebra for GCP implementations.

As far as these authors are aware, the proposed systematic approach for the graphical implementation of the analytic techniques that solve the kinematic/kinetostatic analyses of these linkages is presented here for the first time.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/machines12110825/s1>: one zipped file containing the ggb files with the list of GeoGebra commands presented in the paper and two videos showing the animation of Figures 10 and 12.

Author Contributions: Conceptualization, R.D.G.; methodology, R.D.G.; software, R.D.G. and T.C.; validation, R.D.G. and T.C.; formal analysis, R.D.G. and T.C.; investigation, R.D.G. and T.C.; resources, R.D.G. and T.C.; data curation, R.D.G. and T.C.; writing—original draft preparation, R.D.G. and T.C.; writing—review and editing, R.D.G. and T.C.; supervision, R.D.G.; project administration, R.D.G.; funding acquisition, R.D.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was developed at the Laboratory of Mechatronics and Virtual Prototyping (LaMaViP) of Ferrara University (UNIFE) and was funded by FAR2022 UNIFE and FIRD2023 UNIFE.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

In this appendix, the commented lists of GeoGebra commands necessary to build both the dynamic sketches of the dyads and the associated velocity/acceleration vector diagrams are reported.

- RRR dyad (Figures 1a and 3):

***** Commented list of commands that generates the RRR dynamic sketch (Figure 1a) *****

```
A = (xA, yA)
B = (xB, yB)
c = Circle(A,rC)
d = Circle(B,rD)
C = Intersect(c,d)

i = Segment(A,C)
j = Segment(B,C)
```

(it generates point A and its coordinates as parameters of the object)
 (it generates point B and its coordinates as parameters of the object)
 (it generates the circle centered at A with radius r_c and r_c as parameter of the object)
 (it generates the circle centered at B with radius r_d and r_d as parameter of the object)
 (it locates point C at one intersection of circles c and d; here, the user must choose which intersection (i.e., dyad's assembly mode) he/she is interested in)
 (it generates the segment representing link i)
 (it generates the segment representing link j)

***** Commented list of commands that generates the velocity diagram (Figure 3b) constrained to follow the RRR dynamic sketch (Figure 3a) *****

```
ui = UnitVector(i)
vi = UnitPerpendicularVector(i)
uj = UnitVector(j)
vj = UnitPerpendicularVector(j)
O = (xO,yO)
VA,i = Translate(Vector((vA,x,vA,y)),O)
D = O + VA,i
f = Line(D, vi)
VB,j = Translate(Vector((vB,x,vB,y)),O)
E = O + VB,j
g = Line(E, vj)
F = Intersect(f,g)
VCA,i = Vector(D,F)
VCB,j = Vector(E,F)
VCA = Vector(O,F)
ωi = Dot((VCA,i,vi)/abs(C - A))
ωj = Dot((VCB,j,vj)/abs(C - B))
```

(it generates the unit vector u_i, parallel to segment i (i.e., AC (Figure 3a)) and located on it)
 (it generates the unit vector v_i, perpendicular to segment i)
 (it generates the unit vector u_j, parallel to segment j (i.e., BC (Figure 3a)) and located on it)
 (it generates the unit vector v_j, perpendicular to segment j)
 (it generates point O and its coordinates as parameters of the object)
 (it generates the known velocity vector v_{A,i} together with its components as parameters of the object and applies it to point O (see Figure 3b))
 (it generates point D located at the point of the arrow representing v_{A,i})
 (it generates a line passing through D and parallel to v_i, that is, parallel to the velocity difference v_{CA,i})
 (it generates the known velocity vector v_{B,j} together with its components as parameters of the object and applies it to point O (see Figure 3b))
 (it generates point E located at the point of the arrow representing v_{B,j})
 (it generates a line passing through E and parallel to v_j, that is, parallel to the velocity difference v_{CB,j})
 (it generates point F at the intersection of the two lines f and g (point F graphically solves Equation (3))
 (it generates the arrow representing the velocity difference v_{CA,i})
 (it generates the arrow representing the velocity difference v_{CB,j})
 (it generates the arrow representing the sought-after velocity v_{C,i}(=v_{C,i}))
 (it computes the signed magnitude, ω_i, of the angular velocity of link i)
 (it computes the signed magnitude, ω_j, of the angular velocity of link j)

***** Commented list of commands that generates the acceleration diagram (Figure 3c) constrained to follow the RRR dynamic sketch (Figure 3a) *****

```
P = (xP,yP)
aB,j = Translate(Vector((aB,x,aB,y)),P)

G = P + aB,j
aCB,i,j = Translate(Vector((-ωi2 * |C - B| * ui), G))
H = G + aCB,i,j
h = Line(H, vi)
aA,i = Translate(Vector((aA,x,aA,y)),P)
```

(it generates point P and its coordinates as parameters of the object)
 (it generates the known acceleration vector a_{B,j} together with its components as parameters of the object and applies it to point P (see Figure 3c))
 (it generates point G located at the point of the arrow representing a_{B,j})
 (it computes the vector representing the acceleration component a_{CB,i,j} and applies it to point G)
 (it generates point H located at the point of the arrow representing a_{CB,i,j})
 (it generates a line passing through H and parallel to v_i, that is, parallel to the acceleration component a_{CB,i,j})
 (it generates the known acceleration vector a_{A,i} together with its components as parameters of the object and applies it to point P (see Figure 3c))

```

L = P + aA,i
aCA,i = Translate(Vector((-ωi²*(C - A) * ui), L)
M = L + aCA,i
k = Line(M, vi)
N = Intersect(k,h)
aCA,L,i = Vector(M,N)
aCB,L,i = Vector(H,N)
aCA,i = Vector(L,N)
aCB,i = Vector(G,N)
aC,i = Vector(P,N)
βi = Angle(M,L,N)
βi = Angle(N,L,N)
αi = Angle(N,G,H)
αi = Dot(aCA,L,i,vi)/abs(C - A)
αi = Dot(aCB,L,i,vi)/abs(C - B)
    
```

- RRP dyad (Figures 1b and 4):

***** Commented list of commands that generates the RRP dynamic sketch (Figure 1b) *****

```

A = (XA, YA)
D = (XD, YD)
f = Line(D,Vector(cos(θ),sin(θ)))
uf = UnitVector(f)
vf = UnitPerpendicularVector(f)
c = Circle(A,rc)
g = Line(D + dgf*vf, uf)
C = Intersect(c,g)

i = Segment(A,C)
h = Line(C,vf)
B = Intersect(h,f)
j = Segment(B,C)
Polygon(B + 0.5*aB*(vf - uf), B - 0.5*aB*(uf + vf),
4)
    
```

***** Commented list of commands that generates the velocity diagram (Figure 4b) constrained to follow the RRP dynamic sketch (Figure 4a) *****

```

ui = UnitVector(i)
vi = UnitPerpendicularVector(i)
uj = UnitVector(j)
vj = UnitPerpendicularVector(j)
O = (XO,YO)
VA,i = Translate(Vector((vA,x,vA,y)),O)
F = O + VA,i
m = Line(F, vi)
ωf = Slider(ωf,min, ωf,max, Δωf)
ωf = ωf
VD,f = Translate(Vector((vD,x,vD,y)),O)
G = O + VD,f
VBD,f = Translate(Vector(ωf*(B - D) * vf),G)
    
```

(it generates point L located at the point of the arrow representing a_{A,i})
(it computes the vector representing the acceleration component a_{CA,i} and applies it to point L)
(it generates point M located at the point of the arrow representing a_{CA,i})
(it generates a line passing through M and parallel to v_i, that is, parallel to the acceleration component a_{CA,L,i})
(it generates point N at the intersection of the two lines k and h (point N graphically solves Equation (4))
(it generates the arrow representing the acceleration component a_{CA,L,i})
(it generates the arrow representing the acceleration component a_{CB,L,i})
(it generates the arrow representing the acceleration difference a_{CA,i})
(it generates the arrow representing the acceleration difference a_{CB,i})
(it generates the arrow representing the sought-after acceleration a_{C,i} (=a_{C,i}))
(it generates the angle β_i, which is the constant angle between a_{CA,i} and (A - C))
(it generates the angle β_i, which is the constant angle between a_{CB,i} and (B - C))
(it computes the signed magnitude, α_i, of the angular acceleration of link i)
(it computes the signed magnitude, α_i, of the angular acceleration of link j)

(it generates point A and its coordinates as parameters of the object)
(it generates point D and its coordinates as parameters of the object)
(it generates line f passing through point D with the direction of the unit vector (cos(θ),sin(θ))^T and the angle θ as a parameter of the object)
(it generates a unit vector parallel to line f and located on line f)
(it generates a unit vector perpendicular to line f)
(it generates the circle centered at A with radius r_c and r_c as parameters of the object)
(it generates a line parallel to line f whose distance from f is d_{gf} and d_{gf} as a parameter of the object)
(it locates point C at one intersection of circle c with line g; here, the user must choose which intersection (i.e., dyad's assembly mode) he/she is interested in)
(it generates the segment representing link i)
(it generates a line passing through point C and perpendicular to line f)
(it generates the foot of the perpendicular from C to f)
(it generates the segment representing link j)
(it generates a regular polygon centered at B and with four sides (i.e., a square), parallel to u_f or v_f, whose length is a_B and a_B as a parameter of the object)

(it generates the unit vector u_i, parallel to segment i (i.e., AC (Figure 4a) and located on it)
(it generates the unit vector v_i, perpendicular to segment i)
(it generates the unit vector u_j, parallel to segment j (i.e., BC (Figure 4a)) and located on it)
(it generates the unit vector v_j, perpendicular to segment j)
(it generates point O and its coordinates as parameters of the object)
(it generates the known velocity vector v_{A,i} together with its components as parameters of the object and applies it to point O (see Figure 4b))
(it generates point F located at the point of the arrow representing v_{A,i})
(it generates a line passing through F and parallel to v_i, that is, parallel to the velocity difference v_{CA,i})
(it generates the variable ω_f (necessary to assign the angular velocity, ω_f, of line f) ranging from ω_{f,min} to ω_{f,max} with increment Δω_f)
(it generates the dependent variable ω_f (i.e., the signed magnitude of the angular velocity of link f))
(it generates the known velocity vector v_{D,f} together with its components as parameters of the object and applies it to point O (see Figure 4b))
(it generates point G located at the point of the arrow representing v_{D,f})
(it generates the vector representing the velocity difference v_{BD,f} and applies it to G (see Figure 4b))

```

H = G + vBD,f
vB,f = Vector(O,H)
vCB,j = Translate(Vector(-ω1*C - B1*v1),H)
L = H + vCB,j
n = Line(L, ur)
M = Intersect(m,n)
vCA,i = Vector(F,M)
vB,j,f = Vector(L,M)
vC,i = Vector(O,M)
ω1 = Dot(vCA,i,v1)/abs(C - A)

```

***** Commented list of commands that generates the acceleration diagram (Figure 4c) constrained to follow the RPR dynamic sketch (Figure 4a) *****

```

P = (xP,yP)
aA,i = Translate(Vector(aA,x,aA,y),P)
N = P + aA,i
aCA,i,j = Translate(Vector(-ω12*C - A1*u1),N)
Q = N + aCA,i,j
p = Line(Q, v1)
αr = Slider(αr,min, αr,max, Δαr)
α1 = αr
aD,f = Translate(Vector(aD,x,aD,y),P)

R = P + aD,f
aBD,f = Translate(Vector(αr*vr - ω12*u1),R - D1)R)
S = R + aBD,f
aB,j = Vector(P,S)
aCB,j = Translate(Vector(-αr*vj - ω12*u1),S - B1*v1),S)
T = S + aCB,j
acor = Translate(Vector(2*ω1*Dot(vB,j,f,u1)*v1),T)
U = T + acor
q = Line(U, ur)
V = Intersect(p,q)
aCA,L,i = Vector(Q,V)
aB,j,f = Vector(U,V)
aC,i = Vector(P,V)
α1 = Dot(aCA,L,i,v1)/abs(C - A)

```

- RPR dyad (Figures 1c and 5):

```

***** Commented list of commands that generates the RPR dynamic sketch (Figure 1c) *****
A = (XA, YA)
B = (XB, YB)
c = Circle(A,dAK)
d = Circle(B,dBK)
k = Tangent(c,d)
g = Line(A,k)

```

(it generates point H located at the point of the arrow representing v_{BD,f})
(it generates the arrow representing the velocity v_{B,f})
(it generates the vector representing the velocity difference v_{CB,j} and applies it to H (see Figure 4b))
(it generates point L located at the point of the arrow representing v_{CB,j})
(it generates a line passing through L and parallel to u_r, that is, parallel to the relative velocity v_{B,j,f})
(it generates point M at the intersection of the two lines m and n (point M graphically solves Equation (5)))
(it generates the arrow representing the velocity difference v_{CA,i})
(it generates the arrow representing the relative velocity v_{B,j,f})
(it generates the arrow representing the sought-after velocity v_{C,i}(=v_{C,i})
(it computes the signed magnitude, ω₁, of the angular velocity of link 1)

(it generates point P and its coordinates as parameters of the object)
(it generates the known acceleration vector a_{A,i} together with its components as parameters of the object and applies it to point P (see Figure 4c))
(it generates point N located at the point of the arrow representing a_{A,i})
(it generates the vector representing the acceleration-difference component a_{CA,i,j} and applies it to N (see Figure 4c))
(it generates point Q located at the point of the arrow representing a_{CA,i,j})
(it generates a line passing through N and parallel to v₁, that is, parallel to the acceleration difference component a_{CA,L,i})
(it generates the variable α_r (necessary to assign the angular acceleration, α_r, of line f) ranging from α_{r,min} to α_{r,max} with increment Δα_r)
(it generates the dependent variable α₁ (i.e., the signed magnitude of the angular acceleration of link f))
(it generates the known acceleration vector a_{D,f} together with its components as parameters of the object and applies it to point P (see Figure 4c))
(it generates point R located at the point of the arrow representing a_{D,f})
(it generates the vector representing the acceleration difference a_{BD,f} and applies it to R (see Figure 4c))
(it generates point S located at the point of the arrow representing a_{BD,f})
(it generates the vector representing the acceleration difference a_{CB,j} and applies it to S (see Figure 4c))
(it generates point T located at the point of the arrow representing a_{CB,j})
(it generates the vector representing the Coriolis acceleration a_{cor} and applies it to T (see Figure 4c))
(it generates point U located at the point of the arrow representing a_{cor})
(it generates a line passing through U and parallel to u_r, that is, parallel to the relative acceleration a_{B,j,f})
(it generates point V at the intersection of the two lines p and q (point V graphically solves Equation (6)))
(it generates the arrow representing the acceleration difference component a_{CA,L,i})
(it generates the arrow representing the relative acceleration a_{B,j,f})
(it generates the arrow representing the sought-after acceleration a_{C,i}(=a_{C,i})
(it computes the signed magnitude, α₁, of the angular acceleration of link 1)

(it generates point A and its coordinates as parameters of the object)
(it generates point B and its coordinates as parameters of the object)
(it generates the circle centered at A with radius d_{AK} and d_{AK} as parameter of the object)
(it generates the circle centered at B with radius d_{BK} and d_{BK} as parameter of the object)
(it generates line k as one of the possible common tangents to both the circles c and d; here, the user must choose which tangent (i.e., dyad's assembly mode) he/she is interested in)
(it generates a line, named g, passing through point A and parallel to line k)

(it generates a line, named *l*, passing through point B and parallel to line *k*)
 (it generates a unit vector parallel to line B and perpendicular to line *k*)
 (it generates a unit vector perpendicular to line *g*)
 (it generates link *l* as a polyline with two sides, one parallel to line *k* with length $d_{A,k}$ and Y_k as a parameter of the object)
 (it generates link *j* as a polyline with two sides, one parallel to line *k* with length $d_{B,k}$ and X_k as a parameter of the object)
 (it generates the center, named C, of the P-pair slider as the intersection of the polylines *i* and *j*)
 (it generates a regular polygon centered at C and with four sides (i.e., a square), parallel to u_k or v_k , whose length is a_c and a_c as parameter of the object)

```
f = Line(B,k)
u_k = UnitVector(g)
v_k = UnitPerpendicularVector(g)
i = Polyline(A,A + Y_k*u_k,A + d_Ak*v_k + Y_k*u_k)
j = Polyline(B,B + X_k*u_k,B + d_Bk*v_k - X_k*u_k)
C = Intersect(i,j)
Polygon(C + 0.5*a_c*(v_k - u_k), C - 0.5*a_c*(u_k + v_k), 4)
```

***** Commented list of commands that generates the velocity diagram (Figure 5b) constrained to follow the RPR dynamic sketch (Figure 5a) *****

O = (xO,yO)
 V_A,i = Translate(Vector((v_A,x,v_A,y)),O)
 D = O + V_A,i
 n = Line(D, u_k)
 v_B,j = Translate(Vector((v_B,x,v_B,y)),O)
 E = O + v_B,j
 P = Line(E, UnitPerpendicularVector(Segment(A,B)))
 F = Intersect(p,n)
 V_A,j = Vector(D,F)
 V_A,i = Vector(E,F)
 V_A,j = Vector(O,F)
 ω_j = -Dot(V_A,B,j)
 UnitPerpendicularVector(Segment(A,B))/abs(B - A)
 ω_j = ω_j
 q = Line(F, UnitPerpendicularVector(Segment(A,C)))
 r = Line(E, UnitPerpendicularVector(Segment(B,C)))
 G = Intersect(q,r)
 V_C,i = Translate(Vector((v_C,i)),G)
 H = O + V_C,i

***** Commented list of commands that generates the acceleration diagram (Figure 5c) constrained to follow the RPR dynamic sketch (Figure 5a) *****

P = (xp,yp)
 a_A,i = Translate(Vector((a_A,x,a_A,y)),P)
 L = P + a_A,i
 a_cor = Translate(Vector(2*ω_j*Dot(v_A,j,r, u_k)*v_k),L)
 M = L + a_cor
 S = Line(M, u_k)
 a_B,j = Translate(Vector((a_B,x,a_B,y)),P)

(it generates point O and its coordinates as parameters of the object)
 (it generates the known velocity vector $v_{A,i}$ together with its components as parameters of the object and applies it to point O (see Figure 5b))
 (it generates point D located at the point of the arrow representing $v_{A,i}$)
 (it generates a line passing through D and parallel to u_k , that is, parallel to the relative velocity $v_{A,j}$ ($=v_{C,j}$ $= -v_{C,j}$))
 (it generates the known velocity vector $v_{B,j}$ together with its components as parameters of the object and applies it to point O (see Figure 5b))
 (it generates point E located at the point of the arrow representing $v_{B,j}$)
 (it generates a line passing through E and perpendicular to segment AB, that is, parallel to the velocity difference $v_{AB,j}$)
 (it generates point F at the intersection of the two lines n and p (point F graphically solves Equation (8a)))
 (it generates the arrow representing the relative velocity $v_{A,j}$)
 (it generates the arrow representing the velocity difference $v_{AB,j}$)
 (it generates the arrow representing the velocity $v_{A,i}$)
 (it computes the signed magnitude, ω_j , of the angular velocity of link *j*)
 (it generates the dependent variable ω_j (i.e., the signed magnitude of the angular velocity of link *j*)
 (it generates a line passing through F and perpendicular to segment AC, that is, parallel to $v_{C,i}$)
 (it generates a line passing through E and perpendicular to segment BC, that is, parallel to $v_{CB,j}$)
 (it generates point G at the intersection of the two lines q and r (point G graphically solves Equation (8b)))
 (it generates the arrow representing the velocity $v_{C,i}$)
 (it generates the arrow representing the velocity $v_{CB,j}$)
 (it generates the arrow representing the velocity $v_{C,i}$)
 (it generates the arrow representing $v_{C,j}$ ($= -v_{A,j}$) and applies it at G)
 (it generates the arrow representing the sought-after velocity $v_{C,i}$ that solves Equation (8c))
 (it generates point H located at the point of the arrow representing $v_{C,i}$)

(it generates point P and its coordinates as parameters of the object)
 (it generates the known acceleration vector $a_{A,i}$ together with its components as parameters of the object and applies it to point P (see Figure 5c))
 (it generates point L located at the point of the arrow representing $a_{A,i}$)
 (it generates the vector representing the Coriolis acceleration a_{cor} and applies it to point L (see Figure 5c))
 (it generates point M located at the point of the arrow representing a_{cor})
 (it generates a line passing through M and parallel to u_k , that is, parallel to the relative acceleration $a_{A,j}$ ($=a_{C,j}$ $= -a_{C,j}$))
 (it generates the known acceleration vector $a_{B,j}$ together with its components as parameters of the object and applies it to point P (see Figure 5c))

$N = M + a_{Bj}$
 $a_{AB|j} = \text{Translate}(\text{Vector}(-\omega_1^2 \cdot (A - B)), N)$
 $Q = N + a_{A|j}$
 $t = \text{Line}(Q, \text{Vector}(\text{UnitPerpendicularVector}(\text{Segment}(B, A))))$
 $R = \text{Intersect}(s, t)$
 $a_{A|j} = \text{Vector}(M, R)$
 $a_{A|j} = \text{Vector}(Q, R)$
 $a_{ABj} = \text{Vector}(N, R)$
 $a_{Aj} = \text{Vector}(P, R)$
 $\alpha_j = \text{Dot}(a_{AB, L, j}, \text{UnitPerpendicularVector}(\text{Segment}(B, A))) / \text{abs}(B - A)$
 $\alpha_i = \alpha_j$
 $\beta_j = \arctan(\alpha_j / \omega_1^2)$
 $a = \text{Rotate}(\text{Line}(R, \text{Segment}(A, C)), -\beta_j, R)$
 $b = \text{Rotate}(\text{Line}(N, \text{Segment}(B, C)), -\beta_j, N)$
 $S = \text{Intersect}(a, b)$
 $a_{CAj} = \text{Vector}(R, S)$
 $a_{CBj} = \text{Vector}(N, S)$
 $a_{Cj} = \text{Vector}(P, S)$
 $a_{Cj} = \text{Translate}(-a_{A, j}, S)$
 $T = S + a_{C, j}$
 $U = T + a_{cor}$
 $a_{C, translated} = \text{Vector}(T, U)$
 $a_{Cj} = \text{Vector}(P, U)$

(it generates point N located at the point of the arrow representing a_{Bj})
(it computes $a_{AB|j}$ and generates the vector representing it applied at N (see Figure 5c))
(it generates point Q located at the point of the arrow representing $a_{A|j}$)
(it generates a line passing through Q and perpendicular to segment BA, that is, parallel to $a_{AB, L, j}$)
(it generates point R at the intersection of the two lines s and t (point F graphically solves Equation (10a))
(it generates the arrow representing the relative acceleration $a_{A|j}$)
(it generates the arrow representing the acceleration difference $a_{AB, j}$)
(it generates the arrow representing the acceleration difference a_{ABj})
(it generates the arrow representing the acceleration a_{Aj})
(it computes the signed magnitude, α_j , of the angular acceleration of link j)
(it generates the dependent variable α_i (i.e., the signed magnitude of the angular acceleration of link i)
(it computes the angle β_j)
(it generates a line passing through R and parallel to a_{CAj})
(it generates a line passing through N and parallel to a_{CBj})
(it generates point S at the intersection of the two lines a and b (point S graphically solves Equation (10b))
(it generates the vector representing a_{CAj})
(it generates the vector representing a_{CBj})
(it generates the vector a_{Cj} ($= -a_{A, j}$), and applies it to S)
(it generates point T located at the point of the arrow representing $a_{C, j}$)
(it generates point U at the point of a_{cor} translated into T)
(it generates a vector equal to a_{cor} translated into T)
(it generates the vector representing $a_{C, j}$ that solves Equation (10c))

- PRP dyad (Figures 1d and 6):

***** Commented list of commands that generates the PRP dynamic sketch (Figure 1d) *****

$D = (x_D, y_D)$
 $f = \text{Line}(D, \text{Vector}((\cos(\theta)), \sin(\theta))))$
 $E = (x_E, y_E)$
 $c = \text{Line}(E, \text{Vector}((\cos(\varphi)), \sin(\varphi))))$
 $u_f = \text{UnitVector}(f)$
 $v_f = \text{UnitPerpendicularVector}(f)$
 $u_c = \text{UnitVector}(c)$
 $v_c = \text{UnitPerpendicularVector}(c)$
 $g = \text{Line}(D + d_{df} \cdot v_f, u_f)$
 $n = \text{Line}(E - d_{ce} \cdot v_c, u_c)$
 $C = \text{Intersect}(n, g)$
 $h = \text{Line}(C, v_f)$
 $B = \text{Intersect}(h, f)$
 $J = \text{Segment}(B, C)$
 $\text{Polygon}(B + 0.5 \cdot a_p \cdot (v_f - u_f), B - 0.5 \cdot a_p \cdot (u_f + v_f), 4)$
 $p = \text{Line}(C, v_c)$
 $A = \text{Intersect}(p, c)$
 $I = \text{Segment}(A, C)$
 $\text{Polygon}(A + 0.5 \cdot a_p \cdot (v_c - u_c), A - 0.5 \cdot a_p \cdot (u_c + v_c), 4)$

(it generates point D and its coordinates as parameters of the object)
(it generates line f passing through point D with the direction of the unit vector $(\cos(\theta), \sin(\theta))^T$ and the angle θ as parameter of the object)
(it generates point E and its coordinates as parameters of the object)
(it generates line c passing through point E with the direction of the unit vector $(\cos(\varphi), \sin(\varphi))^T$ and the angle φ as parameter of the object)
(it generates a unit vector parallel to line f and located on line f)
(it generates a unit vector perpendicular to line f)
(it generates a unit vector parallel to line c and located on line c)
(it generates a unit vector perpendicular to line c)
(it generates a line parallel to line f whose distance from f is d_{df} and d_{df} as parameter of the object)
(it generates a line parallel to line c whose distance from c is d_{ce} and d_{ce} as parameter of the object)
(it locates point C at the intersection of lines n and g)
(it generates a line passing through point C and perpendicular to line f)
(it generates the foot of the perpendicular from C to line f)
(it generates the segment representing link J)
(it generates a regular polygon centered at B and with four sides (i.e., a square), parallel to u_f or v_f , whose length is a_p and a_p as a parameter of the object)
(it generates a line passing through point C and perpendicular to line c)
(it generates the foot of the perpendicular from C to line c)
(it generates the segment representing link I)
(it generates a regular polygon centered at A and with four sides (i.e., a square), parallel to u_c or v_c , whose length is the already-defined parameter a_p)

***** Commented list of commands that generates the velocity diagram (Figure 6b) constrained to follow the PRP dynamic sketch (Figure 6a) *****

- $u_i = \text{UnitVector}(i)$
 - $v_i = \text{UnitPerpendicularVector}(i)$
 - $u_j = \text{UnitVector}(j)$
 - $v_j = \text{UnitPerpendicularVector}(j)$
 - $\omega_f = \text{Slider}(\omega_{f,\min}, \omega_{f,\max}, \Delta\omega_f)$
 - $\omega_i = \omega_f$
 - $\omega_c = \text{Slider}(\omega_{c,\min}, \omega_{c,\max}, \Delta\omega_c)$
 - $\omega_l = \omega_c$
 - $O = (\text{XO}, \text{YO})$
 - $v_{D,f} = \text{Translate}(\text{Vector}((v_{D,x}, v_{D,y})), O)$
 - $H = O + v_{D,f}$
 - $v_{BD,f} = \text{Translate}(\text{Vector}(\omega_f * |B - D| * v_i), H)$
 - $L = H + v_{BD,f}$
 - $v_{B,f} = \text{Vector}(O, L)$
 - $(v_{CB,j}) = \text{Translate}(\text{Vector}(-\omega_j * |C - B| * v_j), L)$
 - $M = H + v_{CB,j}$
 - $q = \text{Line}(M, u_i)$
 - $v_{E,c} = \text{Translate}(\text{Vector}((v_{E,x}, v_{E,y})), O)$
 - $N = O + v_{E,c}$
 - $v_{A,E,c} = \text{Translate}(\text{Vector}(\omega_c * |A - E| * v_c), N)$
 - $Q = N + v_{A,E,c}$
 - $v_{A,c} = \text{Vector}(O, Q)$
 - $v_{CA,i} = \text{Translate}(\text{Vector}(\omega_i * |C - A| * v_i), Q)$
 - $R = Q + v_{CA,i}$
 - $r = \text{Line}(R, u_c)$
 - $S = \text{Intersect}(r, q)$
 - $v_{A,i,c} = \text{Vector}(R, S)$
 - $v_{B,j,f} = \text{Vector}(M, S)$
 - $v_{C,j} = \text{Vector}(O, S)$
- (it generates the unit vector u_i , parallel to segment i (i.e., AC (Figure 6a) and located on it)
- (it generates the unit vector v_i , perpendicular to segment i)
- (it generates the unit vector u_j , parallel to segment j (i.e., BC (Figure 6a) and located on it)
- (it generates the unit vector v_j , perpendicular to segment j)
- (it generates the variable ω_f (necessary to assign the angular velocity, ω_f , of line f) ranging from $\omega_{f,\min}$ to $\omega_{f,\max}$ with increment $\Delta\omega_f$)
- (it generates the dependent variable ω_i (i.e., the signed magnitude of the angular velocity of link i))
- (it generates the variable ω_c (necessary to assign the angular velocity, ω_c , of line c) ranging from $\omega_{c,\min}$ to $\omega_{c,\max}$ with increment $\Delta\omega_c$)
- (it generates the dependent variable ω_l (i.e., the signed magnitude of the angular velocity of link l))
- (it generates point O and its coordinates as parameters of the object)
- (it generates the known velocity vector $v_{D,f}$ together with its components as parameters of the object and applies it to point O (see Figure 6b))
- (it generates point H located at the point of the arrow representing $v_{D,f}$ and applies it to G (see Figure 6b))
- (it generates the vector representing the velocity difference $v_{BD,f}$ and applies it to G (see Figure 6b))
- (it generates point L located at the point of the arrow representing $v_{BD,f}$)
- (it generates the arrow representing the velocity $v_{B,f}$)
- (it generates the vector representing the velocity difference $v_{CB,j}$ and applies it to L (see Figure 6b))
- (it generates point M located at the point of the arrow representing $v_{CB,j}$)
- (it generates a line passing through M and parallel to u_i , that is, parallel to the relative velocity $v_{B,j,f}$)
- (it generates the known velocity vector $v_{E,c}$ together with its components as parameters of the object and applies it to point O (see Figure 6b))
- (it generates point N located at the point of the arrow representing $v_{E,c}$)
- (it generates the vector representing the velocity difference $v_{A,E,c}$ and applies it to N (see Figure 6b))
- (it generates point Q located at the point of the arrow representing $v_{A,E,c}$)
- (it generates the arrow representing the velocity $v_{A,c}$)
- (it generates point R located at the point of the arrow representing $v_{CA,i}$ and applies it to Q (see Figure 6b))
- (it generates a line passing through R and parallel to u_c , that is, parallel to the relative velocity $v_{A,i,c}$)
- (it generates point S at the intersection of lines r and q (point S geometrically solves Equation (11)))
- (it generates the vector representing $v_{A,i,c}$)
- (it generates the vector representing $v_{B,j,f}$)
- (it generates the vector representing the sought-after velocity $v_{C,j}$ ($=v_{C,j}$))

***** Commented list of commands that generates the acceleration diagram (Figure 6c) constrained to follow the PRP dynamic sketch (Figure 6a) *****

- $P = (\text{XP}, \text{YP})$
 - $\alpha_f = \text{Slider}(\alpha_{f,\min}, \alpha_{f,\max}, \Delta\alpha_f)$
 - $\alpha_l = \alpha_f$
 - $a_{D,f} = \text{Translate}(\text{Vector}((a_{D,x}, a_{D,y})), P)$
 - $T = P + a_{D,f}$
 - $a_{BD,f} = \text{Translate}(\text{Vector}((\alpha_f * v_i - \omega_f^2 * u_i), |B - D|), T)$
 - $U = T + a_{BD,f}$
 - $a_{B,f} = \text{Vector}(P, U)$
 - $a_{CB,j} = \text{Translate}(\text{Vector}(-(\alpha_j * v_j - \omega_j^2 * u_j), |C - B|), U)$
 - $V = U + a_{CB,j}$
 - $a_{cor,f} = \text{Translate}(\text{Vector}(2 * \omega_f * \text{Dot}(v_{B,j,f}, u_i) * v_j), V)$
- (it generates point P and its coordinates as parameters of the object)
- (it generates the variable α_f (necessary to assign the angular acceleration, α_f , of line f) ranging from $\alpha_{f,\min}$ to $\alpha_{f,\max}$ with increment $\Delta\alpha_f$)
- (it generates the dependent variable α_l (i.e., the signed magnitude of the angular acceleration of link l))
- (it generates the known acceleration vector $a_{D,f}$ together with its components as parameters of the object and applies it to point P (see Figure 6c))
- (it generates point T located at the point of the arrow representing $a_{D,f}$)
- (it generates the vector representing the acceleration difference $a_{BD,f}$ and applies it to T (see Figure 6c))
- (it generates point U located at the point of the arrow representing $a_{BD,f}$)
- (it generates the arrow representing the acceleration $a_{B,f}$)
- (it generates the vector representing the acceleration difference $a_{CB,j}$ and applies it to U (see Figure 6c))
- (it generates point V located at the point of the arrow representing $a_{CB,j}$)
- (it generates the vector representing the Coriolis acceleration $a_{cor,f}$ and applies it to V (see Figure 6c))

(it generates point W located at the point of the arrow representing a_{conf})
 (it generates a line passing through W and parallel to u_f , that is, parallel to the relative acceleration $a_{Bj,f}$)
 (it generates the variable α_c (necessary to assign the angular acceleration, α_c , of line c) ranging from $\alpha_{c,min}$ to $\alpha_{c,max}$ with increment $\Delta\alpha_c$)
 (it generates the dependent variable α_i (i.e., the signed magnitude of the angular acceleration of link i)
 (it generates the known acceleration vector $a_{E,c}$ together with its components as parameters of the object and applies it to point P (see Figure 6c))
 (it generates point T_1 located at the point of the arrow representing $a_{E,c}$)
 (it generates the vector representing the acceleration difference $a_{AE,c}$ and applies it to T_1 (see Figure 6c))
 (it generates point U_1 located at the point of the arrow representing $a_{AE,c}$)
 (it generates the arrow representing the velocity $a_{A,c}$)
 (it generates the vector representing the acceleration difference $a_{CA,i}$ and applies it to U_1 (see Figure 6c))
 (it generates point V_1 located at the point of the arrow representing $a_{CA,i}$)
 (it generates the vector representing the Coriolis acceleration a_{corc} and applies it to V_1 (see Figure 6c))
 (it generates point W_1 located at the point of the arrow representing a_{corc})
 (it generates a line passing through W_1 and parallel to u_c , that is, parallel to the relative acceleration $a_{A,i,c}$)
 (it generates point J at the intersection of lines s and t (point J geometrically solves Equation (12))
 (it generates a vector representing the relative acceleration $a_{Bj,f}$)
 (it generates a vector representing the relative acceleration $a_{A,i,c}$)
 (it generates a vector representing the sought-after acceleration $a_{C,i}$ ($=a_{C,j}$))

- RPP dyad (Figures 1e and 7):
 ***** Commented list of commands that generates the RPP dynamic sketch (Figure 1e) *****

```

A = (xA, yA)
D = (xD, yD)
f = Line(D, Vector((cos(theta), sin(theta))))

u_f = UnitVector(f)
v_f = UnitPerpendicularVector(f)
c = Circle(A, d_Am)
u_m = u_f*cos(pi - delta) + v_f*sin(pi - delta)
v_m = -u_f*sin(pi - delta) + v_f*cos(pi - delta)
g = Line(A, u_m)
m = Tangent(g, c)

F = Intersect(m, f)
B = F - d_Bf*u_f
j = Polyline(B, F + x_i*u_m, F + (x_j + d_j)*u_m)
Polygon(B + 0.5*a_Bf*(v_f - u_f), B - 0.5*a_Bf*(u_f + v_f), 4)

i = Polyline(A, A - x_i*u_m, A - x_j*u_m - d_Am*u_m)
C = Intersect(i, j)
Polygon(C + 0.5*a_Bf*(v_m - u_m), C - 0.5*a_Bf*(u_m + v_m), 4)
    
```

***** Commented list of commands that generates the velocity diagram (Figure 7b) constrained to follow the RPP dynamic sketch (Figure 7a) *****

(it generates point A and its coordinates as parameters of the object)
 (it generates point D and its coordinates as parameters of the object)
 (it generates line f passing through point D with the direction of the unit vector $(\cos(\theta), \sin(\theta))^T$ and the angle θ) as parameters of the object)
 (it generates a unit vector parallel to line f and located on line f)
 (it generates a unit vector perpendicular to line f)
 (it generates the circle centered at A with radius d_{Am} and d_{Am} as parameter of the object)
 (it generates a unit vector, named u_m , whose slope angle with respect to line f is equal to δ)
 (it generates a unit vector, named v_m , that is perpendicular to u_m)
 (it generates a line, named g, passing through A and parallel to u_m)
 (it generates all the tangent to curve c that are parallel to line m; here, the user must choose which tangent (i.e., dyad's assembly mode) he/she is interested in)
 (it generates the intersection point between lines m and f)
 (it generates a point of line f that keeps a fixed distance, named d_{Bf} , from point F and d_{Bf} as parameter of the object)
 (it generates a polyline with two sides, (one of which lies on line m) that represents link j and the object's parameters x_i and d_j)
 (it generates a regular polygon centered at B and with four sides (i.e., a square), parallel to u_f or v_f , whose length is a_B , and a_B as a parameter of the object)
 (it generates a polyline with two sides, one parallel and the other perpendicular to line m, that represents link i and x_i as parameter of the object)
 (it generates point C as the intersection between the two polylines i and j)
 (it generates a regular polygon centered at C and with four sides (i.e., a square), parallel to u_m or v_m , whose length is the already-defined parameter a_P)

(it generates point O and its coordinates as parameters of the object)
 (it generates the known velocity vector $V_{A,i}$ together with its components as parameters of the object and applies it to point O (see Figure 7b))
 (it generates point G located at the point of the arrow representing $V_{A,i}$)
 (it generates the variable ω_f (necessary to assign the angular velocity, ω_f of line f) ranging from $\omega_{f,min}$ to $\omega_{f,max}$ with increment $\Delta\omega_f$)
 (it generates the dependent variable α_f (i.e., the signed magnitude of the angular velocity of link f))
 (it generates the dependent variable ω_i (i.e., the signed magnitude of the angular velocity of link i))
 (it generates the vector representing the velocity difference $V_{C,A,i}$ and applies it to H (see Figure 7b))
 (it generates point H located at the point of the arrow representing $V_{C,A,i}$)
 (it generates a line passing through H and parallel to line m)
 (it generates a vector representing $V_{C,i}$ and applies it to H)
 (it generates the known velocity vector $V_{D,i}$ together with its components as parameters of the object and applies it to point O (see Figure 7b))
 (it generates point L located at the point of the arrow representing $V_{D,i}$)
 (it generates point H located at the point of the velocity difference $V_{BD,i}$ and applies it to L (see Figure 7b))
 (it generates the arrow representing the velocity $V_{B,i}$)
 (it generates the vector representing the velocity difference $V_{CB,j}$ and applies it to M (see Figure 7b))
 (it generates point N located at the point of the arrow representing $V_{CB,j}$)
 (it generates a line passing through N and parallel to u_f , that is, parallel to the relative velocity $V_{B,i}$)
 (it generates point Q at the intersection of lines o and n (point Q geometrically solves Equation (13))
 (it generates the arrow representing the relative velocity $V_{B,j}$)
 (it generates the arrow representing the relative velocity $V_{C,j}$)
 (it generates the arrow representing the sought-after velocity $V_{C,i}$)

```
O = (XO,YO)
VA,i = Translate(Vector((VA,x,VA,y)),O)
G = O + VA,i
ωf = Slider(ωf,min, ωf,max, Δωf)
ωi = ωf
ωf = ωi
VC,A,i = Translate(Vector(ωf*I C -
AI*UnitPerpendicularVector(Segment(A,C))),G)
H = G + VC,A,i
n = Line(H,m)
VC,i = Vector(O,H)
VD,i = Translate(Vector((VD,x,VD,y)),O)
L = O + VD,i
VBD,i = Translate(Vector(ωf*|B - D|*vf),L)
M = L + VBD,i
VB,i = Vector(O,M)
VCB,j = Translate(Vector(ωf*I C -
BI*UnitPerpendicularVector(Segment(B,C))),M)
N = M + VCB,j
O = Line(N,uf)
Q = Intersect(o,n)
VB,j = Vector(N,Q)
VC,j = Vector(Q,H)
VC,i = Vector(O,H)
```

***** Commented list of commands that generates the acceleration diagram (Figure 7b) constrained to follow the RPP dynamic sketch (Figure 7a) *****

```
P = (xP,yP)
αf = Slider(αf,min, αf,max, Δαf)
αf = αf
αf = αf
aD,f = Translate(Vector((aD,x,aD,y)),P)
R = P + aD,f
aBD,f = Translate(Vector((αf*vf - ωf2*uf) | B - D |),R)
S = R + aBD,f
aB,f = Vector(P,S)
aCB,j = Translate(Vector((αf* UnitPerpendicularVector(Segment(B,C)) * C - B |),
S)
T = S + aCB,j
acor,f = Translate(Vector(2*ωf*Dot(vB,j,uf)*vf),T)
U = T + acor,f
acor,j = Translate(Vector(2*ωf*Dot(vC,j,um)*vm),U)
```

(it generates point P and its coordinates as parameters of the object)
 (it generates the variable α_f (necessary to assign the angular acceleration, α_f of line f) ranging from $\alpha_{f,min}$ to $\alpha_{f,max}$ with increment $\Delta\alpha_f$)
 (it generates the dependent variable α_i (i.e., the signed magnitude of the angular acceleration of link i))
 (it generates the dependent variable α_f (i.e., the signed magnitude of the angular acceleration of link f))
 (it generates the known acceleration vector $a_{D,f}$ together with its components as parameters of the object and applies it to point P (see Figure 7c))
 (it generates point R located at the point of the arrow representing $a_{D,f}$)
 (it generates the vector representing the acceleration difference $a_{BD,f}$ and applies it to R (see Figure 7c))
 (it generates point S located at the point of the arrow representing $a_{BD,f}$)
 (it generates the vector representing the acceleration difference $a_{CB,j}$ and applies it to S (see Figure 7c))
 (it generates point T located at the point of the arrow representing $a_{CB,j}$)
 (it generates the vector representing the Coriolis acceleration $a_{cor,f}$ and applies it to T (see Figure 7c))
 (it generates point U located at the point of the arrow representing $a_{cor,f}$)
 (it generates the vector representing the Coriolis acceleration $a_{cor,j}$ and applies it to U (see Figure 7c))

$V = U + a_{conj}$
 $P = \text{Line}(V, u_i)$
 $a_{A,i} = \text{Translate}(\text{Vector}((a_{A,x}, a_{A,y})), P)$
 $R_i = P + a_{A,i}$
 $a_{CA,i} = \text{Translate}(\text{Vector}((\alpha_i * \text{UnitPerpendicularVector}(\text{Segment}(A,C)) - \omega_i^{2*} * \text{UnitVector}(\text{Segment}(A,C))) * (C - A))), R_i)$
 $S_i = R_i + a_{CA,i}$
 $a_{C,i} = \text{Vector}(P, S_i)$
 $q = \text{Line}(S_i, A_m)$
 $W = \text{Intersect}(p, q)$
 $a_{B,j} = \text{Vector}(W, W)$
 $a_{C,j} = \text{Vector}(W, S_i)$
 $a_{C_j} = \text{Vector}(P, W)$

(it generates point V located at the point of the arrow representing a_{conj})
 (it generates a line passing through U and parallel to u_i , that is, parallel to the relative acceleration $a_{B,j}$)
 (it generates the known acceleration vector $a_{A,i}$ together with its components as parameters of the object and applies it to point P (see Figure 7c))
 (it generates point R_i located at the point of the arrow representing $a_{A,i}$)
 (it generates the vector representing the acceleration difference $a_{CA,i}$ and applies it to R_i (see Figure 7c))
 (it generates point S_i located at the point of the arrow representing $a_{CA,i}$)
 (it generates the arrow representing the acceleration $a_{C,i}$)
 (it generates a line passing through S_i and parallel to u_m , that is, parallel to the relative acceleration $a_{C,j}$)
 (it generates point W at the intersection of lines p and q (point W geometrically solves Equation (14))
 (it generates the arrow representing the relative acceleration $a_{B,j}$)
 (it generates the arrow representing the relative acceleration $a_{C,j}$)
 (it generates the arrow representing the acceleration a_{C_j})

Appendix B

In this appendix, the commented lists of GeoGebra commands necessary to build the dynamic sketches of the driving links (Figure 2) are reported.

- Driving link with actuated R-pair (Figure 2a):

$A = (x_A, y_A)$
 $P = (x_P, y_P)$
 $i = \text{Segment}(A, P)$
 $r = \text{Slider}(r_{min}, r_{max}, \Delta r)$
 $\theta_i = \text{Slider}(0, 360, 1)$
 $u_i = \text{UnitVector}(i)$
 $B = \text{Rotate}(A + r * u_i, \theta_i, A)$
 $j = \text{Segment}(A, B)$
 $\text{Angle}(P, A, B)$

(it generates point A and its coordinates as parameters of the object)
 (it generates point P and its coordinates as parameters of the object)
 (it generates the segment AP representing link i)
 (it generates the linear parameter r, ranging from r_{min} to r_{max} with increments of Δr , to use for changing the length of link j)
 (it generates the actuated joint variable θ_i , ranging from 0° to 360° with increment of 1° as a scalar parameter; such a command is replaceable through the analytic expression that gives θ_i as a function of the time, say t, after having defined t with the command "Slider()")
 (it generates the unit vector u_i , parallel to segment i and located on it)
 (it generates point B as the new position of point P' = A + r * u_i , after the segment AP' has rotated around point A of θ_i)
 (it generates the segment AB representing link j)
 (this command just makes the angle θ_i visible on the sketch)

- Driving link with actuated P-pair (Figure 2b):

$A = (x_A, y_A)$
 $B = (x_B, y_B)$
 $i = \text{Segment}(A, B)$
 $d = \text{Slider}(d_{min}, d_{max}, \Delta d)$
 $AC = \text{Slider}(0, \text{abs}(B - A), \Delta AC)$
 $u_i = \text{UnitVector}(i)$
 $v_i = \text{UnitPerpendicularVector}(i)$
 $C = A + AC * u_i$
 $D = C + d * v_i$

(it generates point A and its coordinates as parameters of the object)
 (it generates point B and its coordinates as parameters of the object)
 (it generates the segment AB representing link i)
 (it generates the linear parameter d, ranging from d_{min} to d_{max} with an increments of Δd , to use for changing the geometry of link j)
 (it generates the actuated joint variable AC as a scalar parameter, ranging from 0 to $\text{abs}(B - A)$, which computes the length of segment AB, with an increment of ΔAC . Such a command is replaceable through the analytic expression that gives AC as a function of the time, say t, after having defined t with the command "Slider()")
 (it generates the unit vector u_i , parallel to segment i and located on it)
 (it generates the unit vector v_i , perpendicular to segment i)
 (it generates point C)
 (it generates point D)

(it generates the segment CD representing link j)
 (it generates a regular polygon centered at C and with four sides (i.e., a square), parallel to u_1 or v_1 , whose length is a_p and a_p as parameter of the object)

Appendix C

In this appendix, the commented lists of GeoGebra commands developed for the three case studies illustrated in Section 3 are reported.

- Generation of Coupler Curves (Figure 8)

```
A = (xA,yA)
B = (xB,yB)
f = Segment(A,B)
r2 = Slider(r2,min, r2,max, Δr2)
θ21 = Slider(0°, 360°, 1°)
C = Rotate(A + r2*UnitVector(Segment(A,B)), θ21,A)
g = Segment(A,C)
r3 = Slider(r3,min, r3,max, Δr3)
r4 = Slider(r4,min, r4,max, Δr4)
D = Intersect(Circle(C,r3), Circle(B,r4))
h = Segment(C,D)
uc = UnitVector(h)
vc = UnitPerpendicularVector(h)
i = Segment(B,D)
E = C + xc*uc + yc*vc
```

Polygon(C,D,E)

- Four-Bar Kinestatics (Figures 9 and 10)

***** Commands that build the dynamic Sketch (see Figure 9) *****

```
T = Slider(Tmin, Tmax, ΔT)
t = Slider(0, T, 0.01)
ω2 = 2*π/T
θ21 = ω2*t*180°/π
A = (xA,yA)
B = (xB,yB)
f = Segment(A,B)
a2 = Slider(a2,min, a2,max, Δa2)
C = Rotate(A + a2*UnitVector(f), θ21,A)
g = Segment(A,C)
a3 = Slider(a3,min, a3,max, Δa3)
a4 = Slider(a4,min, a4,max, Δa4)
D = Intersect(Circle(C,a3), Circle(B,a4))
h = Segment(C,D)
i = Segment(D,B)
E2 = A + x2*UnitVector(g) +
y2*UnitPerpendicularVector(g)
```

(it generates point A and its coordinates as parameters of the object)
 (it generates point B and its coordinates as parameters of the object)
 (it generates the segment AB representing link 1, that is, the frame)
 (it generates the variable r₂ (necessary to assign the crank length) ranging from r_{2,min} to r_{2,max} with increment Δr₂)
 (it generates the crank angle θ₂₁ to use for animating the sketch)
 (it generates the movable ending of the crank)
 (it generates the segment AC representing link 2, that is, the crank)
 (it generates the variable r₃ (necessary to assign the coupler length) ranging from r_{3,min} to r_{3,max} with increment Δr₃)
 (it generates the variable r₄ (necessary to assign the rocker length) ranging from r_{4,min} to r_{4,max} with increment Δr₄)
 (it generates the center, D, of the R-pair joining the coupler to the rocker)
 (it generates the segment CD representing link 3, that is, the coupler)
 (it generates a unit vector u_c parallel to segment CD)
 (it generates a unit vector v_c perpendicular to segment CD)
 (it generates the segment BD representing link 4, that is, the rocker)
 (it generates the tracing point, E, that draws the coupler curve and the “Slider”s of the variable parameters x_c and y_c necessary to change its position on the coupler)
 (it generates the triangle CDE fixed to the coupler)

(it generates the variable T)
 (it generates the variable t (time), necessary to give a reference time to the linkage animation, ranging from 0 s to T s with increment 0.01 s)
 (it generates the dependent variable ω₂, necessary to assign the signed magnitude of the crank angular velocity)
 (it generates the crank angle θ₂₁ in degrees)
 (it generates point A and its coordinates as parameters of the object)
 (it generates point B and its coordinates as parameters of the object)
 (it generates the segment AB representing link 1, that is, the frame)
 (it generates the variable a₂ (necessary to assign the crank length) ranging from a_{2,min} to a_{2,max} with increment Δa₂)
 (it generates the movable ending of the crank)
 (it generates the segment AC representing link 2, that is, the crank)
 (it generates the variable a₃ (necessary to assign the coupler length) ranging from a_{3,min} to a_{3,max} with increment Δa₃)
 (it generates the variable a₄ (necessary to assign the rocker length) ranging from a_{4,min} to a_{4,max} with increment Δa₄)
 (it generates the center, D, of the R-pair joining the coupler to the rocker)
 (it generates the segment CD representing link 3, that is, the coupler)
 (it generates the segment BD representing link 4, that is, the rocker)
 (it generates point E₂ fixed to the crank and the associated variable parameters x₂ and y₂)

(it generates a triangular shape of the crank)
 (it computes the area of t2)
 (it generates point G₂ located at the centroid of t2)
 (it computes the area moment of inertia of t2 with respect to its centroid)
 (it generates point E₃ fixed to the coupler and the associated variable parameters x₃ and y₃)
 (it generates the triangular shape of the coupler)
 (it computes the area of t3)
 (it generates point G₃ located at the centroid of t3)
 (it computes the area moment of inertia of t3 with respect to its centroid)
 (it generates point E₄ fixed to the rocker and the associated variable parameters x₄ and y₄)
 (it generates the triangular shape of the rocker)
 (it computes the area of t4)
 (it generates point G₄ located at the centroid of t4)
 (it computes the area moment of inertia of t4 with respect to its centroid)

***** Commands that build the velocity/acceleration vector diagrams (see Figure 9) *****

(it generates point O and its coordinates as parameters of the object)
 (it generates a unit vector with the same direction as parameters of the object (C - A))
 (it generates a unit vector with the same direction as vector k × (C - A))
 (it generates a unit vector with the same direction as vector k × (D - C))
 (it generates a unit vector with the same direction as vector k × (D - C))
 (it generates a unit vector with the same direction as vector k × (D - C))
 (it generates a unit vector with the same direction as vector k × (D - C))
 (it generates the arrow representing the velocity v_{C,2} and applies it at point O)
 (it generates point E located at the point of the arrow representing v_{C,2})
 (it generates point F, which geometrically solves the velocity loop equation)
 (it generates the arrow representing the velocity v_{D,3})
 (it generates the arrow representing the velocity v_{D,3})
 (it computes the signed magnitude, ω₃, of link-3' s angular velocity)
 (it computes the signed magnitude, ω₄, of link-4' s angular velocity)
 (it generates point P and its coordinates as parameters of the object)
 (it generates the variable α₂, necessary to assign the signed magnitude of crank's angular acceleration, ranging from α_{2,min} to α_{2,max} with increment Δα₂)

t2 = Polygon(A,C,E2)
 A2 = Area(t2)
 G2 = Centroid(t2)
 I2 = (1/36)*(a2*y2³ + y2*a2³ - y2*x2*a2² + a2*y2*x2²)
 E3 = C + x3*UnitVector(t)+
 Y3*UnitPerpendicularVector(t)
 t3 = Polygon(C,D,E3)
 A3 = Area(t3)
 G3 = Centroid(t3)
 I3 = (1/36)*(a3*y3³ + y3*a3³ - y3*x3*a3² + a3*y3*x3²)
 E4 = D + x4*UnitVector(t)+
 Y4*UnitPerpendicularVector(t)
 t4 = Polygon(D,B,E4)
 A4 = Area(t4)
 G4 = Centroid(t4)
 I4 = (1/36)*(a4*y4³ + y4*a4³ - y4*x4*a4² + a4*y4*x4²)

O = (xO,yO)
 u2 = UnitVector(Segment(A,C))
 v2 = UnitPerpendicularVector(Segment(A,C))
 u3 = UnitVector(Segment(C,D))
 v3 = UnitPerpendicularVector(Segment(C,D))
 u4 = UnitVector(Segment(B,D))
 v4 = UnitPerpendicularVector(Segment(B,D))
 VC2 = Translate(Vector(u2*(C - A) + v2),O)
 E = O + VC2
 F = Intersect(Line(E,v3),Line(O,v4))
 vD3 = Vector(OF)
 vDC3 = Vector(E,F)
 ω3 = Dot(vDC3,v3)/ID - C I
 ω4 = Dot(vD3,v4)/ID - B I
 P = (xP,yP)
 α2 = Slider(α2,min, α2,max,Δα2)

ac2 = Translate(Vector(α2*v2 - ω2²*u2*(C - A)),P)
 G = P + ac2
 apB||4 = Translate(Vector(-ω4²*(D - B)),P)
 H = P + apB||4
 apC||3 = Translate(Vector(-ω3²*(D - C)),G)
 L = G + apC||3
 M = Intersect(Line(H,v4),Line(L,v3))
 apD||3 = Vector(L,M)
 apB||4 = Vector(L,M)
 apD4 = Vector(P,M)
 apDC3 = Vector(G,M)
 α3 = Dot(apD||3,v3)/ID - C I
 α4 = Dot(apB||4,v4)/ID - B I
 P' = (xP',yP')
 ac3 = Translate(ac2,P')
 G' = P' + ac3

```

aD2 = Translate(aD2,P')
M = P'+aD3
aDC3' = Vector(G,M')
n = Rotate(Segment(P',G'), Angle(C,A,G2),P')
n = Rotate(Segment(P',G'), Angle(A,C,G2),G')
G2' = Intersect(m,n)
ac2 = Vector(P',G2')
p = Rotate(Segment(G',M'), Angle(D,C,G3),G')
q = Rotate(Segment(G',M'), Angle(C,D,G3),M)
G3' = Intersect(p,q)
ac3 = Vector(P',G3')
r = Rotate(Segment(P',M'), Angle(D,B,G4),P')
s = Rotate(Segment(P',M'), Angle(B,D,G4),M)
C4' = Intersect(r,s)
aC4 = Vector(P',G4')

***** Commands that build the force-equilibrium diagrams (see Figure 10) *****
rho = Slider(rho_min, rho_max, delta rho)
m2 = rho*A2
J2 = rho*L2
m3 = rho*A3
J3 = rho*L3
m4 = rho*A4
J4 = rho*L4
Q4 = -m4*aG4
VF4 = UnitPerpendicularVector(Q4)
P4 = G4 + VF4*(J4*aG4/abs(Q4))
F4 = Translate(Q4,P4)
a = Line(C,D)
K4 = Intersect(a,Line(P4,F4))
e = Segment(P4,K4)
f1 = Segment(B,K4)
N4 = P4 + F4
N4 = Intersect(Line(P4,f1), Line(R4,a))
F34,4 = Vector(R4,N4)
F14,4 = Vector(N4,F4)
F34,4 = Translate(F34,4,D)
F14,4 = Translate(F14,4,B)
F2,4 = Translate(-F34,4,C)
F12,4 = Translate(-F32,4,A)
Q3 = -m3*aC3
VF3 = UnitPerpendicularVector(Q3)
P3 = G3 + VF3*(J3*aC3/abs(Q3))
F3 = Translate(Q3,P3)
g1 = Line(B,D)
K3 = Intersect(g1,Line(P3,F3))
h1 = Segment(C,K3)

(it applies at P' a vector equal to aD3,4 (=aD3,3))
(it generates point M' located at the point of the arrow representing aD3,3)
(it applies at G' a vector equal to aD3,3)
(it draws a segment parallel to the acceleration difference aC2,A,2 with one ending at P')
(it draws a segment parallel to the acceleration difference aC2,C,2 with one ending at G')
(it generates point G2' that closes the acceleration loop equation aG2,A,2 = aC2,C,2+aC,A,2)
(it generates the vector representing aC2')
(it draws a segment parallel to the acceleration difference aC5,C,3 with one ending at G')
(it draws a segment parallel to the acceleration difference aC5,D,3 with one ending at M)
(it generates point G3' that closes the acceleration loop equation aG3,D,3 = aC5,D,3+aD3,3)
(it generates the vector representing aC3)
(it draws a segment parallel to the acceleration difference aC4,B,4 with one ending at P')
(it draws a segment parallel to the acceleration difference aC4,D,4 with one ending at M)
(it generates point G4' that closes the acceleration loop equation aC4,B,4 = aC4,D,4+aD3,4)
(it generates the vector representing aC4)

(it generates the variable rho, necessary to assign the areal density of the links, ranging from rho_min to rho_max with increment delta rho)
(it computes the mass of link 2)
(it computes the mass moment of inertia about the barycenter of link 2)
(it computes the mass of link 3)
(it computes the mass moment of inertia about the barycenter of link 3)
(it computes the mass of link 4)
(it computes the mass moment of inertia about the barycenter of link 4)
(it computes the resultant of the inertia forces of link 4)
(it generates a unit vector perpendicular to Q4)
(it applies Q4 at P4)
(it generates the line of action of F34,4)
(it generates the intersection of the lines of action of F4 and F34,4)
(it generates a segment lying on the action line of F4)
(it generates a segment lying on the action line of F14,4)
(it generates a point located at point of the arrow representing F4)
(it generates point N4 that graphically solves the force equilibrium equation F4 + F14,4 + F34,4 = 0)
(it generates a vector representing force F34,4)
(it generates a vector representing force F14,4)
(it applies force F34,4 on its line of action)
(it applies force F14,4 on its line of action)
(it applies force F2,4 on its line of action)
(it applies force F12,4 on its line of action)
(it computes the resultant of the inertia forces of link 3)
(it generates a unit vector perpendicular to Q3)
(it generates a point of the central axis of the inertia forces applied to link 3)
(it applies Q3 at P3)
(it generates the line of action of F34,3)
(it generate the intersection of the lines of action of F3 and F34,3)
(it generates a segment lying on the action line of F2,3)

```

```

h4 = Segment(P3, K5)
R3 = P3 + F3
N3 = Intersect(Line(P3,g1), Line(R3,h1))
F23,3 = Vector(R4,N3)
F43,3 = Vector(N3, P3)
F43,3 = Translate(-F43,3,D)
F32,3 = Translate(-F23,3,C)
F14,3 = Translate(-F34,3,B)
F12,3 = Translate(-F32,3,A)
Q2 = -n2*aC2
v12 = UnitPerpendicularVector(Q2)
P2 = C5 + v12*(I2*o2/abs(Q2))
F2 = Translate(Q2, P2)
F12,2 = Translate(-F2,A)
F14 = Translate(F14,4 + F14,3,B)
F34 = Translate(F34,4 + F34,3,D)
F12 = Translate(F12,4 + F12,3 + F12,2,A)
M12 = -(x(P2) - x(A))*y(F2) + (y(P2) - y(A))*x(F2) - (x(A) - x(C))*y(F32) + (y(C) - y(A))*x(F32)

```

- Shaper Mechanism's Kinematic Analysis (Figure 12)

***** Commands that build the dynamic Sketch (see Figure 12). *****

```

T = Slider(Tmin, Tmax, ΔT)
t = Slider(0, T, 0.01)
ω2 = 2*π/T
θ21 = ω2*t*180°/π
A = (xA,yA)
B = A+(0, -a1)
f = Segment(A,B)
a2 = Slider(a2,min, a2,max, Δa2)
u1 = UnitVector(Segment(A, Rotate(B,90°,A)))
u1 = UnitPerpendicularVector(u1)
C = Rotate(A+ a2*u1, θ21,r,A)
g = Segment(A,C)
a4 = Slider(a4,min, a4,max, Δa4)
D = B+a4*UnitVector(Segment(B,C))
h = Segment(B,D)
Polygon(C + 0.5*a6*UnitPerpendicularVector(h) - UnitVector(h)) C - 0.5*a6*(UnitVector(h) + UnitPerpendicularVector(h))/4
BB = B+(a1 + b1)*v1
i = Segment(BB - (a6 - 2)*u1, BB + (a6 + 2)*u1)
E = Intersect(Circle(D,a5),Line(BB,u1))
j = Segment(D,E)
Polygon(E + 0.5*a7*(v1 - u1),E - 0.5*a7*(u1 + v1),4)

```

(it generates a segment lying on the action line of F3)
(it generates a point located at point of the arrow representing F3)
(it generates point N3 that graphically solves the force equilibrium equation $F_3 + F_{23,3} + F_{43,3} = 0$)
(it generates a vector representing force $F_{23,3}$)
(it generates a vector representing force $F_{43,3}$)
(it applies force $F_{43,3}$ on its line of action)
(it applies force $F_{32,3}$ on its line of action)
(it applies force $F_{14,3}$ on its line of action)
(it applies force $F_{12,3}$ on its line of action)
(it computes the inertia forces of link 2)
(it generates a unit vector perpendicular to Q_2)
(it generates a point of the central axis of the inertia forces applied to link 2)
(it applies Q_2 at P_2)
(it applies force $F_{12,2}$ on its line of action)
(it applies the resultant F_{14} on its line of action)
(it applies the resultant F_{34} on its line of action)
(it applies the resultant F_{12} on its line of action)
(it computes the generalized torque the actuator must apply)

(it generates the variable T)
(it generates the variable t (time), necessary to give a reference time to the linkage animation, ranging from 0 s to T s with increment 0.01 s)
(it generates the dependent variable ω_2 , necessary to assign the signed magnitude of the crank angular velocity)
(it generates the crank angle θ_{21} in degrees)
(it generates point A and its coordinates as parameters of the object)
(it generates point B and the "Slider" of frame geometry's parameter a_1)
(it generates the segment AB representing the first part of link 1, which is the frame)
(it generates the variable a_2 (necessary to assign the crank length) ranging from $a_{2,min}$ to $a_{2,max}$ with increment Δa_2)
(it generates unit vector u_1)
(it generates a unit vector perpendicular to u_1)
(it generates the movable ending of the crank)
(it generates the segment AC representing link 2, that is, the crank)
(it generates the variable a_4 (necessary to assign the length of segment BD representing link 4) ranging from $a_{4,min}$ to $a_{4,max}$ with increment Δa_4)
(it generates point D, which is the center of the R-pair joining links 4 and 5)
(it generates the segment BD representing link 4)
(it generates a square centered at C with two sides parallel to segment BD and the associated variable a_6 , which plays the role of side length)
(it generates point BB lying on the horizontal dash-dot line of Figure 12 and the associated frame geometry parameter b_1)
(it generates the dash-dot segment of Figure 12, which plays the role of joint axis, and the associated parameter a_6 , which assigns its half-length)
(it generates point E and the parameter a_5 that is the length of segment DE)
(it generates the segment DE representing link 5)
(it generates a square centered at E with two horizontal sides)

(it generates the black horizontal line of Figure 12, which is the second part of link 1 (the frame))

(it generates point O and its coordinates as parameters of the object)
 (it generates the vector representing velocity $v_{C,2}$ and applies it at point O)
 (it generates point F at the point of the arrow representing $v_{C,2}$)
 (it generates point G that graphically solves the velocity loop equation $v_{C,3} = v_{C,4} + v_{C,34}$)
 (it generates the vector representing velocity $v_{C,4}$)
 (it generates the vector representing relative velocity $v_{C,34}$)
 (it generates the vector representing velocity $v_{D,4}$ ($=v_{D,5}$)) at point O)
 (it generates point H at the point of the arrow representing $v_{D,4}$ ($=v_{D,5}$))
 (it generates point L that graphically solves the velocity loop equation $v_{E,5} = v_{D,5} + v_{ED,5}$)
 (it generates the vector representing velocity $v_{E,5}$ ($=v_{E,6}$))
 (it generates the vector representing velocity $v_{ED,5}$)
 (it computes the signed magnitude, ω_4 , of the angular velocity of link 4)
 (it computes the signed magnitude, ω_2 , of the angular velocity of link 3)
 (it computes the signed magnitude, ω_5 , of the angular velocity of link 5)
 (it computes the Coriolis acceleration)
 (it generates point P and its coordinates as parameters of the object)
 (it generates the variable α_2 , necessary to assign the signed magnitude of crank 5's angular acceleration, ranging from $\alpha_{2,min}$ to $\alpha_{2,max}$ with increment $\Delta\alpha_2$)

(it generates the vector representing the acceleration $a_{C,2}$ ($=a_{C,3}$) and applies it to P)
 (it generates point M located at the point of the arrow representing $a_{C,2}$)
 (it generates the arrow representing Coriolis acceleration a_{cor} and applies it at P)
 (it generates point N located at the point of the arrow representing a_{cor})
 (it generates the vector representing the component $a_{CA||A}$ ($=a_{C||A}$) of acceleration difference $a_{CA,4}$ and applies it to N)
 (it generates point Q located at the point of the arrow representing $a_{C||A}$)
 (it generates point M, which geometrically solves the acceleration loop equation $a_{C,3} = a_{C,4} + a_{C,34} + a_{cor}$)
 (it generates the vector representing $a_{CA\perp A}$ ($=a_{C\perp A}$))
 (it generates the vector representing the relative acceleration $a_{C,34}$)
 (it generates the vector representing $a_{C,4}$ ($=a_{CA,4}$))
 (it generates the vector representing acceleration $a_{D,4}$ ($=a_{D,5}$) and applies it at point N)
 (it generates point S located at the point of the arrow representing $a_{D,4}$)
 (it generates the vector representing the component $a_{ED||5}$ of acceleration difference and applies it to S)
 (it generates point U located at the point of the arrow representing $a_{ED||5}$)
 (it generates point V, which geometrically solves the acceleration loop equation $a_{E,5} = a_{D,5} + a_{ED||5} + a_{ED\perp,5}$)
 (it generates the vector representing $a_{ED\perp,5}$)
 (it generates the vector representing $a_{E,5}$ ($=a_{E,6}$))
 (it computes the signed magnitude, α_4 , of link-4's angular acceleration)
 (it computes the signed magnitude, α_5 , of link-3's angular acceleration)
 (it computes the signed magnitude, α_5 , of link-5's angular acceleration)

**** Commands that build the velocity/acceleration vector diagrams (see Figure 12) ****

```

i' = Translate(i, Vector(-(0.5*ap + 0.01)*v1,))

**** Commands that build the velocity/acceleration vector diagrams (see Figure 12) ****

O = (x0,y0)
vC2 = Translate( $\omega_2$ *Perpendicular Vector(Segment(A,C)),O)
F = O + vC2
G = Intersect(Line(O, Vector(Unit Perpendicular Vector(h))) , Line(F, Vector( Unit Vector(h))))
vC4 = Vector(O,G)
vC34 = Vector(G,F)
vD4 = Translate(Vector(vC4*abs(D - B) / abs(C - B)),O)

H = O + vD4
L = Intersect(Line(O,i),Line(H,Vector(Perpendicular Vector(Segment(D,E)))) )
vE5 = Vector(O,L)
vED5 = Vector(H,L)
 $\omega_4 = \text{Dot}(v_{C,4}, \text{UnitPerpendicular Vector(Segment(B,C))}) / \text{abs}(C - B)$ 
 $\omega_3 = \omega_4$ 
 $\omega_5 = \text{Dot}(v_{D,5}, \text{UnitPerpendicular Vector(Segment(D,E))}) / \text{abs}(E - D)$ 
acor = 2* $\omega_4$ *UnitPerpendicular Vector(vC,34)
P = (xp,yp)
 $\alpha_2 = \text{Slider}(\alpha_{2,min}, \alpha_{2,max}, \Delta\alpha_2)$ 

aC,2 = Translate( $\alpha_2$ *Perpendicular Vector(Vector(A,C)) -  $\omega_2$ *Vector(A,C),P)

M = P + aC,2
acor = Translate(acor,P)

N = P + acor
aC||,4 = Translate(Vector(- $\omega_4$ * $\omega_2$ (C - B)),N)

Q = N + aC||,4
R = Intersect(Line(Q, Vector(Perpendicular Vector(aC||,4))),Line(M,Segment(B,D)))
aC\perp,4 = Vector(Q,R)
aC,34 = Vector(R,M)
aC,4 = Vector(N,R)
aD,4 = Translate(Vector(aC,4*abs(D - B) / abs(C - B)),N)

S = N + aD,4
aED||,5 = Translate(Vector(- $\omega_5$ * $\omega_2$ (E - D)),S)

U = S + aED||,5
V = Intersect(Line(U, Vector(Perpendicular Vector(Segment(D,E)))) , Line(N,u1))
aED\perp,5 = Vector(U,V)
aE,5 = Vector(S,V)
 $\alpha_4 = \text{Dot}(a_{C\perp,4}, \text{UnitPerpendicular Vector(Segment(B,C))}) / |C - B|$ 
 $\alpha_5 = \alpha_4$ 
 $\alpha_5 = \text{Dot}(a_{ED\perp,5}, \text{UnitPerpendicular Vector(Segment(D,E))}) / |E - D|$ 

```

References

1. Mirth, J.A. Parametric Modeling—A New Paradigm for Mechanisms Education? In Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2012), Chicago, IL, USA, 12–15 August 2012. Paper No.: DETC2012-70175.
2. Waldron, K.J.; Kinzel, G.L.; Agrawal, S.K. *Kinematics, Dynamics, and Design of Machinery*, 3rd ed.; John Wiley & Sons Inc.: Chichester, UK, 2016.
3. Erdman, A.G. Computer-Aided Mechanism Design: Now and the Future. *J. Mech. Des.* **1995**, *117*, 93–100. [CrossRef]
4. Kinzel, E.C.; Schmiedeler, J.P.; Pennock, G.R. Kinematic Synthesis for Finitely Separated Positions Using Geometric Constraint Programming. *J. Mech. Des.* **2006**, *128*, 1070–1079. [CrossRef]
5. Kinzel, E.C.; Schmiedeler, J.P.; Pennock, G.R. Function Generation with Finitely Separated Precision Points Using Geometric Constraint Programming. *J. Mech. Des.* **2007**, *129*, 1185–1190. [CrossRef]
6. Mirth, J.A. The Application of Geometric Constraint Programming to the Design of Motion Generating Six-Bar Linkages. In Proceedings of the ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, USA, 12–15 August 2012; Volume 4: 36th Mechanisms and Robotics Conference, Parts A and B. pp. 1503–1511. [CrossRef]
7. Mirth, J.A. A GCP Design Approach for Infinitesimally and Multiply Separated Positions Based on the Use of Velocity and Acceleration Vector Diagrams. In Proceedings of the ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Buffalo, NY, USA, 17–20 August 2014; Volume 5B. [CrossRef]
8. Schmiedeler, J.P.; Clark, B.C.; Kinzel, E.C.; Pennock, G.R. Kinematic Synthesis for Infinitesimally and Multiply Separated Positions Using Geometric Constraint Programming. *J. Mech. Des.* **2014**, *136*, 034503. [CrossRef]
9. Mirth, J.A. Mixed Exact and Approximate Position Design of Planar Linkages via Geometric Constraint Programming (GCP) Techniques. In Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Boston, MA, USA, 2–5 August 2015; Volume 5B. [CrossRef]
10. Mirth, J.A. The Application of Geometric Constraint Programming to the Design of Stephenson III Dwell Linkages. In Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Charlotte, NC, USA, 21–24 August 2016; Volume 5B. [CrossRef]
11. Zimmerman, R.A., II. Planar Linkage Synthesis for Mixed Motion, Path, and Function Generation Using Poles and Rotation Angles. *J. Mechanisms Robotics* **2018**, *10*, 025004. [CrossRef]
12. Alfattani, R.; Lusk, C. Shape-Morphing Using Bistable Triangles with Dwell-Enhanced Stability. *J. Mechanisms Robotics* **2020**, *12*, 051003. [CrossRef]
13. Tong, Y.; Myszka, D.H.; Murray, A.P. Four-Bar Linkage Synthesis for a Combination of Motion and Path-Point Generation. In Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, OR, USA, 4–7 August 2013; Volume 6A. [CrossRef]
14. Ting, K.; Chan, C.L. Reversible Geometric Constraint Programming on Kinematic Analysis and Synthesis of Planar Linkages. In Proceedings of the ASME 2022 International Mechanical Engineering Congress and Exposition, Columbus, OH, USA, 30 October–3 November 2022; Volume 4. [CrossRef]
15. Zimmerman, R.A., II. *Planar Linkage Synthesis: A Modern CAD Based Approach*; Zimmerman Publishing: Novi, MI, USA, 2023; ISBN 979-8218173586.
16. Simionescu, P.A. *Computer-Aided Graphing and Simulation Tools for AutoCAD Users*; CRC Press: New York, NY, USA, 2015; ISBN 978-1-4822-5292-7.
17. Söylemez, E. *Kinematic Synthesis of Mechanisms: Using Excel and GeoGebra*; Springer: Cham, Switzerland, 2023; ISBN 978-3-031-30955-7.
18. Venema, G.A. *Exploring Advanced Euclidean Geometry with GeoGebra*; Mathematical Association of America, Inc.: Washington, DC, USA, 2013; ISBN 978-1-61444-111-3.
19. Assur, L.V. Research of a planar linkages with lower pairs on the basis of their structure and classification. Part 1. *Bull. Petrograd Polytech. Inst.* **1913**, *20*, 329–386. (In Russian)
20. Assur, L.V. Research of a planar linkages with lower pairs on the basis of their structure and classification. Part 2. *Bull. Petrograd Polytech. Inst.* **1914**, *21*, 187–283. (In Russian)
21. Assur, L.V. *Investigation of Plane, Hinged Mechanisms with Lower Pairs, from the Point of View of Their Structure and Classification*; Artboleviskii, I.I., Ed.; Academy of Sciences: Moscow, Russia, 1952.
22. Baranov, G.G. *Curso de la Teoria de Mecanismos y Maquinas*; Editorial Mir: Moscow, Russia, 1979.
23. Galletti, C.U. On the Position Analysis of Assur's Groups of High Class. *Meccanica* **1979**, *14*, 6–10. [CrossRef]
24. Galletti, C.U. A Note on Modular Approaches to Planar Linkage Kinematic Analysis. *Mech. Mach. Theory* **1986**, *21*, 385–391. [CrossRef]
25. Rojas, N.; Thomas, F. On closed-form solutions to the position analysis of Baranov trusses. *Mech. Mach. Theory* **2012**, *50*, 179–196. [CrossRef]
26. Hahn, E.; Shai, O. The Unique Engineering Properties of Assur Groups/Graphs, Assur Kinematic Chains, Baranov Trusses and Parallel Robots. In Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2016), Charlotte, NC, USA, 21–24 August 2016. DETC2016-59135.

27. Sun, Y.; Ge, W.; Zheng, J.; Dong, D. Solving the Kinematics of the Planar Mechanism Using Data Structures of Assur Groups. *J. Mech. Robot.* **2016**, *8*, 061002. [CrossRef]
28. Huang, P.; Ding, H. Structural Synthesis of Baranov Trusses with Up to 13 Links. *J. Mech. Des.* **2019**, *141*, 072301. [CrossRef]
29. Uicker, J.J., Jr.; Pennock, G.R.; Shigley, J.E. *Theory of Machines and Mechanisms*, 6th ed.; Cambridge University Press: Cambridge, UK, 2023; ISBN 9781009303644.
30. Di Gregorio, R. A geometric and analytic technique for studying single-DOF planar mechanisms' dynamics. *Mech. Mach. Theory* **2021**, *168*, 104609. [CrossRef]
31. Di Gregorio, R. A geometric and analytic technique for studying multi-DOF planar mechanisms' dynamics. *Mech. Mach. Theory* **2022**, *176*, 104975. [CrossRef]
32. Di Gregorio, R. The Role of Instant Centers in Kinematics and Dynamics of Planar Mechanisms: Review of LaMaViP's Contributions. *Machines* **2022**, *10*, 732. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Reconfiguration Analysis and Characteristics of a Novel 8-Link Variable-DOF Planar Mechanism with Five Motion Modes

Xianwen Kong ^{1,*} and Jieyu Wang ²¹ School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK² School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China; jywang2021@shu.edu.cn

* Correspondence: x.kong@hw.ac.uk

Abstract: Variable-DOF (or kinematotropic) mechanisms are a class of reconfigurable mechanisms that have varying degrees of freedom (DOF) in different motion modes and can be reconfigured without disassembly. However, the number of proposed variable-DOF multi-loop planar mechanisms is currently limited. This paper introduces a new 8-link variable-DOF planar mechanism that has five motion modes. Firstly, the 8-link variable-DOF planar mechanism is described. Then, reconfiguration analysis of the mechanism is performed using a hybrid approach that combines elimination and computer algebraic geometry methods. The analysis reveals that the 8-link mechanism has one 2-DOF motion mode and four 1-DOF motion modes. It can switch among three motion modes at four transition configurations and between two motion modes at the remaining four transition configurations. The paper also highlights the geometric characteristics of the mechanism in different motion modes. In contrast to variable-DOF planar mechanisms presented in the literature, the proposed 8-link mechanism has two inactive joints in one of its 1-DOF motion modes. Moreover, both closed-loop 4R kinematic sub-chains of the mechanism must appear as either a pair of parallelograms or a pair of anti-parallelograms in the same motion mode. As a by-product of this research, a method for factoring trigonometric functions in two angles is also proposed.

Keywords: variable-DOF mechanism; reconfigurable mechanism; reconfiguration analysis; motion mode; factorization of two-angle trigonometric function

Citation: Kong, X.; Wang, J. Reconfiguration Analysis and Characteristics of a Novel 8-Link Variable-DOF Planar Mechanism with Five Motion Modes. *Machines* **2023**, *11*, 529. <https://doi.org/10.3390/machines11050529>

Academic Editor: Raffaele Di Gregorio

Received: 11 April 2023

Revised: 1 May 2023

Accepted: 1 May 2023

Published: 4 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the current research focuses in mechanisms and robotics is reconfigurable mechanisms and robots [1,2], which could help meet the needs of robots and manufacturing systems that can rapidly adapt to changes in environment and production.

Variable-DOF (or kinematotropic) mechanisms [3–15] are a class of reconfigurable mechanisms that have varying degrees of freedom (DOF) in different motion modes and can be reconfigured without disassembly. Considerable progress has been made in the type synthesis and reconfiguration analysis of variable-DOF mechanisms, including single-loop spatial mechanisms [5,7,12], parallel mechanisms [11,13–20], multi-mode mobile parallel mechanisms [21], and multi-loop mechanisms [4,5,22–29]. It should be noted that variable-DOF mechanisms are composed of conventional kinematic joints and do not involve variable kinematic joints [30], reconfigurable kinematic joints [31] or metamorphic kinematic joints [32].

Apart from the construction methods [7,23,27], most approaches for the type synthesis of variable-DOF mechanisms are based on different mathematical methods ranging from displacement group theory [5,11], intersection of surfaces [10,20], factorization of polynomials [12,33], and primary decomposition of ideals [14,17–19] to the comprehensive Gröbner basis of parametric polynomial equations [19,34]. Through the construction methods in [7,23,27], a number of variable-DOF mechanisms have been constructed from

existing overconstrained mechanisms. No overconstrained mechanisms are required in advance if using methods in [5,10–12,33]; however, only several variable-DOF mechanisms have been obtained by using these methods. Using the methods in [14,17,18], one can detect whether a multi-DOF overconstrained parallel mechanism is a variable-DOF parallel mechanism. Using the methods in [19,34], one can investigate the impact of link parameters of multi-DOF overconstrained parallel mechanism on the number and types of motion modes and identify different variable-DOF parallel mechanisms of the same topological structure. Variable-DOF mechanisms were obtained by using multi-mode single-loop kinematic chains as building blocks in [16,23]. With further development and application of the above methods, more and more variable-DOF mechanisms are expected to be revealed.

Methods for the reconfiguration analysis of variable-DOF mechanisms mainly include the elimination approaches [22], algebraic geometry methods [24,35,36], numerical algebraic geometry methods [37], branch-and-prune methods [38], singular value decomposition approaches [39–42], and the higher-order kinematics based approaches [43–46]. The first five methods can be used to identify all the motion modes of a variable-DOF mechanism as long as the link parameters of the mechanism are given, whereas a singular or transition configuration of the variable-DOF mechanism must be given in advance if the last method, which is more computationally efficient, is used for the reconfiguration analysis.

It is noted that there are no variable-DOF planar single-loop mechanisms composed of R (revolute) and P (prismatic) joints, and the number of variable-DOF multi-loop planar mechanisms is still very limited. The only four variable-DOF multi-loop planar mechanisms proposed so far are the 12-link Wunderlich mechanism in [3], the 10-link Kovalev mechanism in [4], the 8-link variable-DOF planar mechanism in [5], and the 8-link variable-DOF planar mechanism in [47].

One difference between these four variable-DOF multi-loop planar mechanisms lies in the number of their inactive joints. An inactive joint in a variable-DOF mechanism under a specified motion mode is a kinematic joint that loses its DOF due to intrinsic constraints within the mechanism. The 8-link variable-DOF planar mechanism proposed in [5] has four inactive joints in its 3-DOF motion mode and no inactive joint in its 1-DOF motion mode. In the 3-DOF motion mode, this 8-link mechanism degenerates to a planar serial 3R mechanism. The 8-link variable-DOF planar mechanism presented in [47] has four inactive joints in four of its 1-DOF motion modes and no inactive joint in its 2-DOF motion mode. In these four 1-DOF motion modes, a closed-loop 4R kinematic sub-chain of the 8-link mechanism degenerates (or loses its DOF).

One question arising from the above observations is the following: Are there variable-DOF 8-link planar mechanisms which have neither a serial mechanism motion mode nor a motion mode with a degenerated closed-loop 4R sub-kinematic chain? This paper will answer the above question by presenting a novel 8-link variable-DOF planar mechanism. As will be shown later, this 8-link mechanism has two inactive joints in one of its 1-DOF motion modes and no inactive joints in the other motion modes.

This paper is organized as follows. In Section 2, a geometric description of a novel variable-DOF 8-link planar mechanism is given. A set of kinematic equations is set up in Section 3 with the variables selected to better reflect the geometric characteristics of the mechanism in different motion modes. The motion modes and transition configurations of the variable-DOF 8-link planar mechanism are identified using a hybrid approach that combines elimination and computer algebraic geometry methods in Sections 4 and 5. The reconfiguration of the variable-DOF 8-link planar mechanism is detailed in Section 6. Finally, conclusions are drawn.

2. Geometric Description of a Novel 8-Link Variable-Dof Planar Mechanism

The 8-link variable-DOF planar mechanism [47] constructed using two parallelograms is composed of four binary links and four triangular ternary links and has four inactive joints in four of its 1-DOF motion modes and no inactive joint in its 2-DOF motion mode. Recently, it was revealed in [48] that in addition to the 1-DOF motion mode, a 3-RR planar

parallelogram may have up to two structure modes if the two ternary links are triangular or no structure mode if the two ternary links are collinear. The four inactive joints in a motion mode of the 8-link variable-DOF planar mechanism correspond to the structure mode of a 3-RR planar parallelogram. Using two parallelograms with no structure mode, we can construct an 8-link variable-DOF planar mechanism that does not have 1-DOF motion mode with four inactive joints. Alternatively, by simply replacing each triangular ternary link in the 8-link variable-DOF planar mechanism in [47] with a collinear ternary link, the 1-DOF motion modes with four inactive joints of the original 8-link mechanism will be eliminated. This would lead to a novel 8-link variable-DOF planar mechanism (Figure 1).

The novel 8-link variable-DOF planar mechanism is composed of four identical binary links, $A_{11}B_{11}$, $A_{12}B_{12}$, $A_{21}B_{21}$, and $A_{22}B_{22}$, and four identical collinear ternary links, $AA_{11}A_{12}$, $BB_{11}B_{12}$, $AA_{21}A_{22}$, and $BB_{21}B_{22}$, connected by 10 R joints. Link $A_{22}B_{22}$ is the frame. The link parameters of the 8-link variable-DOF planar mechanism are

$AA_{11} = BB_{11} = AA_{21} = BB_{21} = a_1$, $AA_{12} = BB_{12} = AA_{22} = BB_{22} = a_2$, and $A_{11}B_{11} = A_{12}B_{12} = A_{21}B_{21} = A_{22}B_{22} = L_1$.

The link parameters of an example 8-link variable-DOF planar mechanism are $a_1 = 45$, $a_2 = 75$, and $L_1 = 25$. Here, link lengths are represented by dimensionless numbers that indicate their relative length, subject to the condition that $L_1 < (a_2 - a_1)$ in order to avoid link interference.

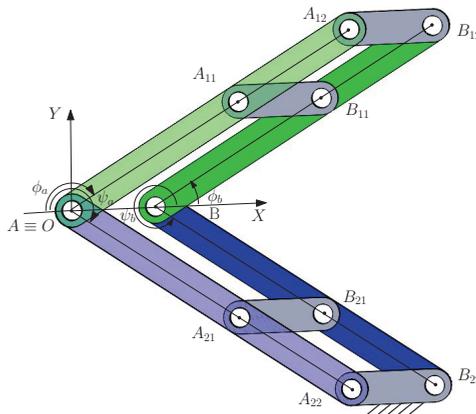


Figure 1. A novel 8-link variable-DOF planar mechanism.

3. Kinematic Equations

To facilitate the identification of the geometric characteristics of the 8-link mechanism, the coordinate system $O-XY$ is set up such that O coincides with R joint center A , and R joint center B is located on the positive X -axis. Let ϕ_a (ψ_a) denote the angle between the negative direction of the X -axis and the link AA_{11} (AA_{21}) measured clockwise and ϕ_b (ψ_b) the angle between the positive direction of the X -axis and the link BB_{11} (BB_{21}) measured anti-clockwise. An auxiliary variable, $L = AB$ ($L > 0$), is introduced to simplify the reconfiguration analysis.

The loop closure equations of loops $ABB_{11}A_{11}A$, $ABB_{12}A_{12}A$, $ABB_{21}A_{21}A$, and $ABB_{22}A_{22}A$ written in vector form are

$$\begin{cases} (\vec{AB} + \vec{BB}_{11} - \vec{AA}_{11}) \cdot (\vec{AB} + \vec{BB}_{11} - \vec{AA}_{11}) = L_1^2 \\ (\vec{AB} + \vec{BB}_{12} - \vec{AA}_{12}) \cdot (\vec{AB} + \vec{BB}_{12} - \vec{AA}_{12}) = L_1^2 \\ (\vec{AB} + \vec{BB}_{21} - \vec{AA}_{21}) \cdot (\vec{AB} + \vec{BB}_{21} - \vec{AA}_{21}) = L_1^2 \\ (\vec{AB} + \vec{BB}_{22} - \vec{AA}_{22}) \cdot (\vec{AB} + \vec{BB}_{22} - \vec{AA}_{22}) = L_1^2 \end{cases}$$

Rewriting the above equation in complex number form, we have

$$\begin{cases} (L + a_1 e^{i\phi_b} - a_1 e^{i(\pi-\phi_a)})(L + a_1 e^{-i\phi_b} - a_1 e^{-i(\pi-\phi_a)}) - L_1^2 = 0 \\ (L + a_2 e^{i\phi_b} - a_2 e^{i(\pi-\phi_a)})(L + a_2 e^{-i\phi_b} - a_2 e^{-i(\pi-\phi_a)}) - L_1^2 = 0 \\ (L + a_1 e^{i\psi_b} - a_1 e^{i(\pi-\psi_a)})(L + a_1 e^{-i\psi_b} - a_1 e^{-i(\pi-\psi_a)}) - L_1^2 = 0 \\ (L + a_2 e^{i\psi_b} - a_2 e^{i(\pi-\psi_a)})(L + a_2 e^{-i\psi_b} - a_2 e^{-i(\pi-\psi_a)}) - L_1^2 = 0 \end{cases}$$

Simplifying the above equation, we obtain

$$\begin{cases} a_1 C(\phi_b + \phi_a) + L(C\phi_b + C\phi_a) + (L^2 - L_1^2)/(2a_1) + a_1 = 0 \\ a_2 C(\phi_b + \phi_a) + L(C\phi_b + C\phi_a) + (L^2 - L_1^2)/(2a_2) + a_2 = 0 \\ a_1 C(\psi_b + \psi_a) + L(C\psi_b + C\psi_a) + (L^2 - L_1^2)/(2a_1) + a_1 = 0 \\ a_2 C(\psi_b + \psi_a) + L(C\psi_b + C\psi_a) + (L^2 - L_1^2)/(2a_2) + a_2 = 0 \end{cases} \tag{1}$$

where S^* and C^* denote \sin^* and \cos^* , respectively.

4. Motion Mode Analysis of an 8-Link Variable-Dof Planar Mechanism Using a Hybrid Approach

In this section, we will reveal all the motion modes of the novel 8-link variable-DOF mechanism (Figure 1) by using resultant elimination, which has been extensively used in the kinematic analysis of mechanisms [49,50], and the primary decomposition of ideals from computer algebraic geometry [35], which has been used in the reconfiguration analysis of multi-mode mechanisms [17,24,36].

Eliminating ϕ_b from the first and second equations of Equation (1) and ψ_b from the third and fourth equations of Equation (1), Equation (1) is reduced to the following set of two equations in three variables ϕ_a , ψ_a , and L (see Appendix A for details)

$$\begin{cases} (1 - C^2\phi_a)[1 - (k_2 - C\phi_a)^2] - [k_1 - C\phi_a(k_2 - C\phi_a)]^2 = 0 \\ (1 - C^2\psi_a)[1 - (k_2 - C\psi_a)^2] - [k_1 - C\psi_a(k_2 - C\psi_a)]^2 = 0 \end{cases} \tag{2}$$

where $k_1 = (L^2 - L_1^2)/(2a_1a_2) - 1$ and $k_2 = -(a_1 + a_2)(L^2 - L_1^2)/(2La_1a_2)$.

For simplicity reasons and without loss of generality, we will investigate the reconfiguration analysis of the 8-link variable-DOF mechanism via the example mechanism given in Section 2. Substituting the link parameters of the example 8-link mechanism into Equation (2), we obtain

$$\begin{cases} (L - 25)(L + 25)f_1 = 0 \\ (L - 25)(L + 25)f_2 = 0 \end{cases} \tag{3}$$

where $f_1 = L^4 + 240L^3C\phi_a + 6750C(2\phi_a)L^2 + 7025L^2 - 150,000C\phi_aL - 9,000,000$ and $f_2 = L^4 + 240L^3C\psi_a + 6750C(2\psi_a)L^2 + 7025L^2 - 150,000C\psi_aL - 9,000,000$.

Since $L > 0$, Equation (3) leads to two cases:

Case A

$$L - 25 = 0 \tag{4}$$

Equation (4) represents a 2-DOF motion mode, motion mode 1 (Figure 2a), of the 8-link mechanism. In motion mode 1, both closed-loop 4R kinematic sub-chains, $A_{11}B_{11}B_{12}A_{12}$ and $A_{21}B_{21}B_{22}A_{22}$, are parallelograms. Throughout the remainder of this paper, including Figure 2, line AB will be kept in a horizontal position by releasing the frame. This is to ensure that the geometric characteristics of the 8-link mechanism are clearly illustrated.

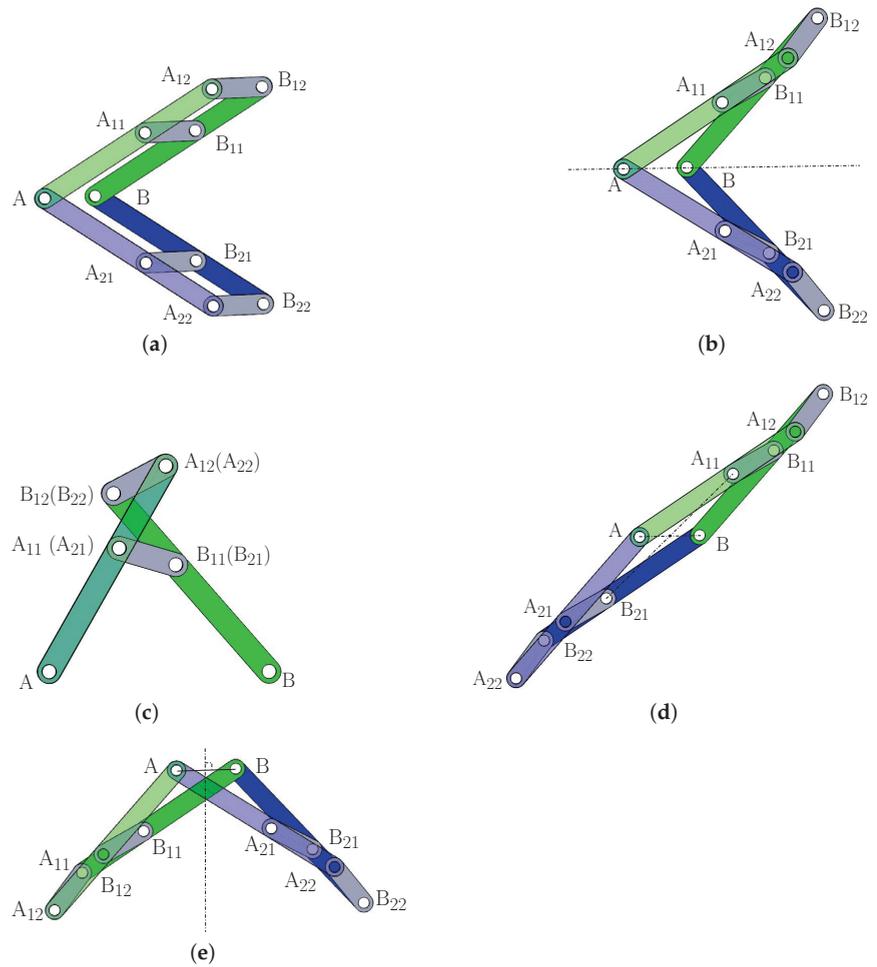


Figure 2. The 8-link variable-DOF planar mechanism in (a) 2-DOF motion mode 1; (b) 1-DOF motion mode 2: Kite motion mode; (c) 1-DOF motion mode 3: Planar 4R mechanism mode; (d) 1-DOF motion mode 4: Parallelogram motion mode; and (e) 1-DOF motion mode 5: Isosceles trapezium motion mode.

Case B

$$\begin{cases} L^4 + 240L^3C\phi_a + 6750C(2\phi_a)L^2 + 7025L^2 - 150,000C\phi_aL - 9,000,000 = 0 \\ L^4 + 240L^3C\psi_a + 6750C(2\psi_a)L^2 + 7025L^2 - 150,000C\psi_aL - 9,000,000 = 0 \end{cases} \quad (5)$$

In the following, we will identify the motion modes associated with Equation (5) by eliminating L using resultant first and then using the primary decomposition of ideals from the algebraic geometry [35] to find the positive-dimensional solutions to the resulted equations.

Eliminating L using resultants from Equation (5), we obtain

$$36,905,625,000,000,000,000,000g(C\phi_a - C\psi_a)^4 = 0 \quad (6)$$

where $g = 72,900C\phi_a^4 - 5940C\phi_a^3C\psi_a - 215,280C\phi_a^2C\psi_a^2 - 5940C\phi_aC\psi_a^3 + 72,900C\psi_a^4 + 63,661C\phi_a^2 - 12,122C\phi_aC\psi_a + 63,661C\psi_a^2 - 57,600$.

From Equation (6), we obtain the following two sub-cases

$$C\phi_a - C\psi_a = 0 \quad (7)$$

and

$$g = 0 \quad (8)$$

Equation (7) has two solutions:

$$\psi_a = -\phi_a \quad (9)$$

and

$$\psi_a = \phi_a \quad (10)$$

Equation (9) represents a 1-DOF motion mode, motion mode 2 (Figure 2b), of the 8-link mechanism. In motion mode 2, both closed-loop 4R sub-kinematic chains, $A_{11}B_{11}B_{12}A_{12}$ and $A_{21}B_{21}B_{22}A_{22}$, are anti-parallelograms, and the 8-link mechanism is symmetric about line AB . Motion mode 2 is called the kite motion mode because, in this motion mode, the 8-link mechanism takes the shape of a kite.

Equation (10) represents a 1-DOF motion mode, motion mode 3 (Figure 2c), of the 8-link mechanism. In motion mode 3, both closed-loop 4R kinematic sub-chains, $A_{11}B_{11}B_{12}A_{12}$ and $A_{21}B_{21}B_{22}A_{22}$, coincide, and the 8-link mechanism has two inactive joints A and B . Motion mode 3 is called the planar 4R mechanism mode because, in this motion mode, the 8-link mechanism degenerates to a planar 4R mechanism.

Using the primary decomposition of ideals from computer algebraic geometry, Equation (8) can be rewritten as (See Appendix B for details)

$$g_1 g_2 = 0 \quad (11)$$

where $g_1 = -281 + 135C(2\phi_a) - 306C(\phi_a - \psi_a) + 295C(\phi_a + \psi_a) + 135C(2\psi_a)$ and $g_2 = -281 + 135C(2\phi_a) + 295C(\phi_a - \psi_a) - 306C(\phi_a + \psi_a) + 135C(2\psi_a)$.

Equation (11) has two solutions:

$$g_1 = 0 \quad (12)$$

and

$$g_2 = 0 \quad (13)$$

Equation (12) represents a 1-DOF motion mode, motion mode 4 (Figure 2d), of the 8-link mechanism. In motion mode 4, both closed-loop 4R sub-kinematic chains, $A_{11}B_{11}B_{12}A_{12}$ and $A_{21}B_{21}B_{22}A_{22}$, are anti-parallelograms, and the 8-link mechanism is rotational symmetric. Motion mode 4 is called the parallelogram motion mode because, in this motion mode, the 8-link mechanism is in the shape of a parallelogram.

Equation (13) represents a 1-DOF motion mode, motion mode 5 (Figure 2e), of the 8-link mechanism. In motion mode 5, both closed-loop 4R kinematic sub-chains, $A_{11}B_{11}B_{12}A_{12}$ and $A_{21}B_{21}B_{22}A_{22}$, are anti-parallelograms, and the 8-link mechanism is symmetric about the perpendicular bisector of AB . Motion mode 5 is called the isosceles trapezium motion mode because, in this motion mode, the 8-link mechanism is in the shape of an isosceles trapezium.

It can be observed that if $(\phi_a^*, \psi_a^*, L^*)$ is a set of solution to Equation (5), then $(\pi \pm \phi_a^*, \pi \pm \psi_a^*, -L^*)$ are also solutions to Equation (5). Since $L > 0$, the $\phi_a - \psi_a$ curve for motion mode 4 (or 5) (see Figure 3) is only one half of the curve obtained using Equation (12) (or Equation (13)) that lies outside of the region enclosed by lines $\psi_a - \phi_a = \pm\pi$ (or

$\phi_a + \psi_a = \pm\pi$). In other words, in motion mode 4, we have $\psi_a - \phi_a > \pi$ or $\psi_a - \phi_a < -\pi$. In motion mode 5, we have $\psi_a + \phi_a > \pi$ or $\psi_a + \phi_a < -\pi$.

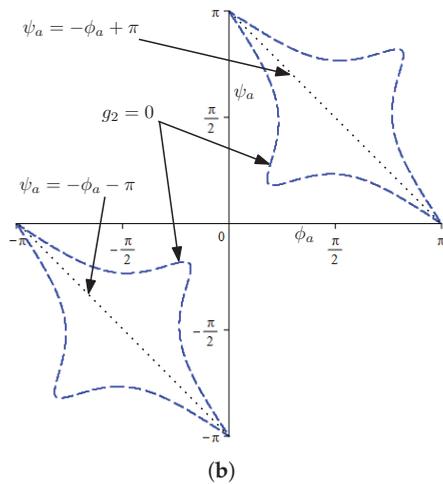
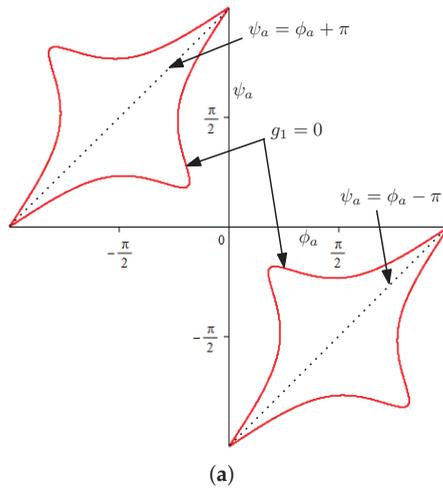


Figure 3. Kinematic analysis of the 8-link variable-DOF planar mechanism in: (a) motion mode 4; (b) motion mode 5.

In summary, the 8-link variable-DOF mechanism has one 2-DOF motion mode and four 1-DOF motion modes (Figure 2). Table 1 summarizes all the five motion modes of the 8-link mechanism and their geometric characteristics. Figure 4 shows the $\phi_a - \psi_a$ curves for the four 1-DOF motion modes 2 (Equation (9)), 3 (Equation (10)), 4 (Equation (12)) and 5 (Equation (13)).

Unlike the two 8-link variable-DOF planar mechanisms in [5,47] which have four inactive joints in some of their motion modes, this 8-link variable-DOF planar mechanism has two inactive joints A and B in one of the 1-DOF motion modes (motion mode 3).

Figure 2 shows that both closed-loop 4R kinematic sub-chains of this novel 8-link variable-DOF planar mechanism must appear as parallelograms in its 2-DOF motion mode or anti-parallelograms in any of its 1-DOF motion mode, whereas both the 8-link variable-

DOF planar mechanism [47] and the 12-link Wunderlich mechanism have a motion mode with an odd number of 4R sub-kinematic chains appearing as anti-parallellograms.

Table 1. Five motion modes of the 8-link variable-DOF mechanism.

No	DOF	Constraint Equations	Description
1	2	$L = 25$	Both closed-loop 4R sub-kinematic chains are parallelograms (Figure 2a). ϕ_a and ψ_a are independent.
2	1	$\psi_a = -\phi_a$	Both closed-loop 4R kinematic sub-chains are anti-parallellograms. The 8-link mechanism is symmetric about line AB (Figure 2b).
3	1	$\psi_a = \phi_a$	Both closed-loop 4R sub-kinematic chains are anti-parallellograms that coincide with each other (Figure 2c), and the 8-link mechanism has two inactive joints A and B.
4	1	$-281 + 135C(2\phi_a)$ $-306C(\phi_a - \psi_a)$ $+295C(\phi_a + \psi_a)$ $+135C(2\psi_a) = 0$	Both closed-loop 4R kinematic sub-chains are anti-parallellograms. The 8-link mechanism is rotational symmetric (Figure 2d).
5	1	$-281 + 135C(2\phi_a)$ $+295C(\phi_a - \psi_a)$ $-306C(\phi_a + \psi_a)$ $+135C(2\psi_a) = 0$	Two closed-loop 4R sub-kinematic chains are anti-parallellograms. The 8-link mechanism is symmetric about the perpendicular bisector of AB (Figure 2e).

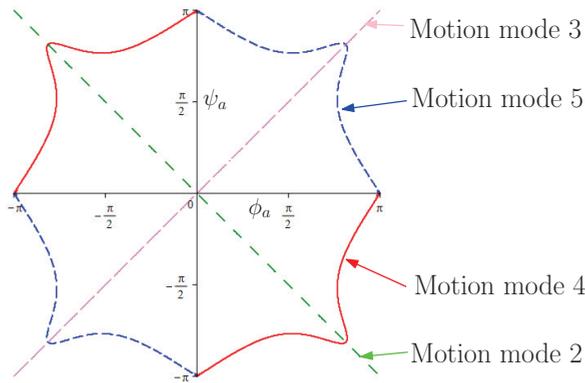


Figure 4. Four 1-DOF motion modes on $\phi_a - \psi_a$ plane.

Although one can solve Equation (5) directly using the primary decomposition of ideals to identify all the motion modes of the 8-link variable-DOF mechanism, it was found hard to obtain the concise equations (Equations (12) and (13)) for motion modes 4 and 5 that were obtained by eliminating L before calculating the primary decomposition of ideals.

It is important to note that the reconfiguration analysis approach described in [47] is not applicable to the 8-link variable-DOF planar mechanism shown in Figure 1. This is because the method in [47] cannot distinguish between motion mode 2 (Figure 2b) and motion mode 4 (Figure 2d), or between motion mode 3 (Figure 2c) and motion mode 5 (Figure 2e), for this mechanism.

5. Transition Configuration Analysis of the 8-Link Variable-Dof Planar Mechanism

The transition configurations between two or more motion modes can be obtained by solving the kinematic equations composed of equations of these motion modes [36].

Let us take the transition configurations between motion modes 2 and 4, $T(2 \wedge 4)$, of the 8-link mechanism as an example.

The set of equations composed of Equations (9) (motion mode 2) and (12) (motion mode 4) is

$$\begin{cases} \psi_a = -\phi_a \\ -281 + 135C(2\phi_a) - 306C(\phi_a - \psi_a) + 295C(\phi_a + \psi_a) + 135C(2\psi_a) = 0 \end{cases} \quad (14)$$

Section 4 shows that in motion mode 4, we have $\psi_a - \phi_a > \pi$ or $\psi_a - \phi_a < -\pi$. Solving Equation (14) under these conditions, we obtain two solutions:

$$\begin{cases} \phi_a = 2.5559(rad) \\ \psi_a = -\phi_a \end{cases} \quad (15)$$

$$\begin{cases} \phi_a = -2.5559(rad) \\ \psi_a = -\phi_a \end{cases} \quad (16)$$

Equations (15) and (16) show that there are two transition configurations between motion modes 2 and 4, $T(2 \wedge 4)_I$ (Figure 5a) and $T(2 \wedge 4)_{II}$ (Figure 5b). In these two transition configurations, links $A_{i1}B_{i1}$ and $B_{i2}A_{i2}$ are parallel to AB. One can readily obtain that the instantaneous DOF of the 8-link variable-DOF mechanism is two in these transition configurations. The details are omitted here since the calculation of instantaneous DOF of a mechanism has been well-documented in the literature.

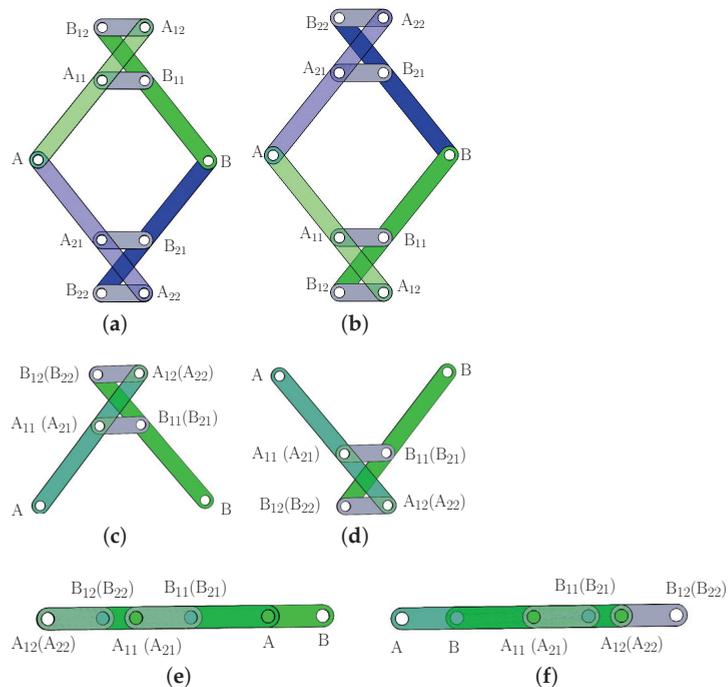


Figure 5. Cont.

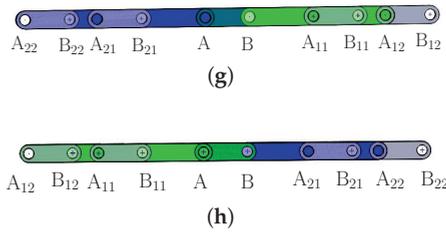


Figure 5. The 8-link variable-DOF planar mechanism in transition configuration: (a) $T(2 \wedge 4)_I$; (b) $T(2 \wedge 4)_{II}$; (c) $T(3 \wedge 5)_I$; (d) $T(3 \wedge 5)_{II}$; (e) $T(1 \wedge 2 \wedge 3)_I$; (f) $T(1 \wedge 2 \wedge 3)_{II}$; (g) $T(1 \wedge 4 \wedge 5)_I$; and (h) $T(1 \wedge 4 \wedge 5)_{II}$.

Following the above steps, we can identify six more transition configurations of the 8-link mechanism, including two transition configurations, $T(3 \wedge 5)_I$ (Figure 5c) and $T(3 \wedge 5)_{II}$ (Figure 5d), in which the mechanism can switch between two motion modes and four transition configurations, $(T(1 \wedge 2 \wedge 3)_I$ (Figure 5e), $T(1 \wedge 2 \wedge 3)_{II}$ (Figure 5f), $T(1 \wedge 4 \wedge 5)_I$ (Figure 5g), and $T(1 \wedge 4 \wedge 5)_{II}$) (Figure 5h), in which the mechanism can switch among three motion modes. It is noted that there are no transition configurations between motion modes 2 and 5 or between motion modes 3 and 4. The geometric characteristics of the 8-link mechanism in all the eight transition configurations are summarized in Table 2. All the transition configurations are singular configurations. The instantaneous DOF of the 8-link variable-DOF in transition configurations can be readily obtained as two in transition configurations $T(2 \wedge 4)_I$, $T(2 \wedge 4)_{II}$, $T(3 \wedge 5)_I$, and $T(3 \wedge 5)_{II}$ and four in transition configurations $T(1 \wedge 2 \wedge 3)_I$, $T(1 \wedge 2 \wedge 3)_{II}$, $T(1 \wedge 4 \wedge 5)_I$, and $T(1 \wedge 4 \wedge 5)_{II}$.

Table 2. Transition configurations of the 8-link mechanism.

No	ϕ_a and ψ_a (rad)	Description	Instantaneous DOF
$T(2 \wedge 4)_I$	$\begin{cases} \phi_a = 2.5559 \\ \psi_a = -\phi_a \end{cases}$	Links $A_{i1}B_{i1}$ and $B_{i2}A_{i2}$ are parallel to AB (Figure 5a).	2
$T(2 \wedge 4)_{II}$	$\begin{cases} \phi_a = -2.5559 \\ \psi_a = -\phi_a \end{cases}$	Links $A_{i1}B_{i1}$ and $B_{i2}A_{i2}$ are parallel to AB (Figure 5b).	
$T(3 \wedge 5)_I$	$\begin{cases} \phi_a = 2.5559 \\ \psi_a = \phi_a \end{cases}$	Links $A_{i1}B_{i1}$ and $B_{i2}A_{i2}$ ($i = 1$ and 2) are parallel to AB (Figure 5c).	
$T(3 \wedge 5)_{II}$	$\begin{cases} \phi_a = -2.5559 \\ \psi_a = \phi_a \end{cases}$	Links $A_{i1}B_{i1}$ and $B_{i2}A_{i2}$ ($i = 1$ and 2) are parallel to AB (Figure 5d).	
$T(1 \wedge 2 \wedge 3)_I$	$\begin{cases} \phi_a = 0 \\ \psi_a = 0 \end{cases}$	All the R joint centers are collinear (Figure 5e).	4
$T(1 \wedge 2 \wedge 3)_{II}$	$\begin{cases} \phi_a = \pi \\ \psi_a = \pi \end{cases}$	All the R joint centers are collinear (Figure 5f).	
$T(1 \wedge 4 \wedge 5)_I$	$\begin{cases} \phi_a = \pi \\ \psi_a = 0 \end{cases}$	All the R joint centers are collinear (Figure 5g).	
$T(1 \wedge 4 \wedge 5)_{II}$	$\begin{cases} \phi_a = 0 \\ \psi_a = \pi \end{cases}$	All the R joint centers are collinear (Figure 5h).	

6. Reconfiguration of the Variable-Dof 8-Link Planar Mechanism

Figure 6 illustrates the reconfiguration of the 8-link planar mechanism among the five motion modes via the eight transition configurations in the $\phi_a - \psi_a$ plane. The curves in the $\phi_a - \psi_a$ plane of the four 1-DOF motion modes, motion modes 2, 3, 4, and 5, are shown in green, pink, red, and blue, respectively. The 2-DOF motion mode, motion mode 1 in which $L = 25$, covers the whole $\phi_a - \psi_a$ plane. However, the 8-link mechanism can only transit among motion modes 1, 2, and 3 at two transition configurations $T(1 \wedge 2 \wedge 3)_I$ and $T(1 \wedge 2 \wedge 3)_{II}$ and among motion modes 1, 4, and 5 at two transition configurations $T(1 \wedge 4 \wedge 5)_I$ and $T(1 \wedge 4 \wedge 5)_{II}$ since we have $L > 25$ in the other configurations in motion modes 2, 3, 4, and 5.

Photos of the LEGO model of this 8-link mechanism at all the transition configurations, configurations with $\phi_a = \pm\pi/2$ and/or $\psi_a = \pm\pi/2$ in 1-DOF motion modes 2, 3, 4, and 5, and a configuration with $\phi_a = \pi/2$ and $\psi_a = -\pi/2$ in 2-DOF motion mode 1 are given. To distinguish the only 2-DOF motion mode from the 1-DOF motion modes, the photos of the sample configuration and the four transition configurations associated with motion mode 1 are framed. In the LEGO model, the links are allocated in six layers, and the axis of R joint A is in a curved shape to allow the mechanism to switch among all the five motion modes through the eight transition configurations without link interference. An animation of the reconfiguration of the 8-link mechanism among the five motion modes can be found in the supplementary materials. In the animation, link AA₂₁A₂₂ is selected as the frame of the mechanism, and all “ \wedge ” have been omitted in the notations for transition configurations for simplicity reasons.

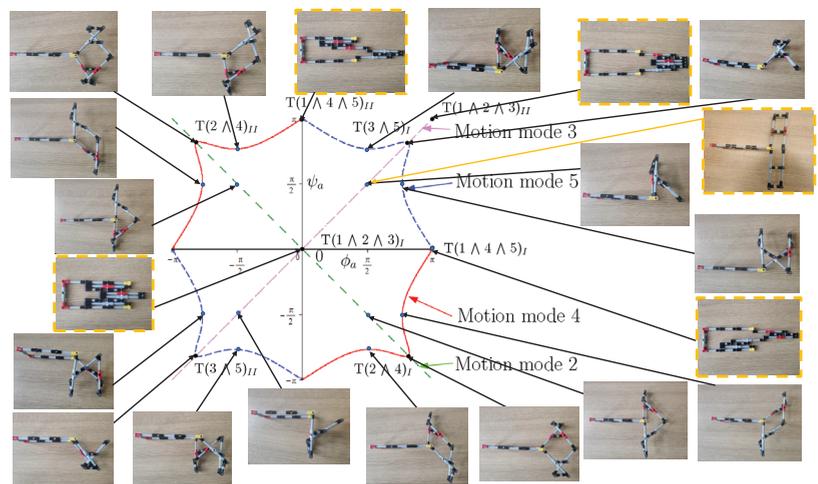


Figure 6. Reconfiguration of the example variable-DOF 8-link planar mechanism among the five motion modes.

The mechanism could be more compact if it is only required to switch among some but not all of its motion modes. For example, if one needs the 8-link mechanism to switch among four motion modes 1, 2, 4 and 5 only (Figure 7), all the links can be located in four layers without encountering link interference as shown in the CAD model of the mechanism in transition configuration $T(1 \wedge 4 \wedge 5)_I$. This four-layer 8-link planar mechanism could be used as a construction unit of new variable-DOF multi-loop mechanisms, which would enrich the types of reconfigurable/multi-mode deployable mechanisms [25–27,42,51,52].

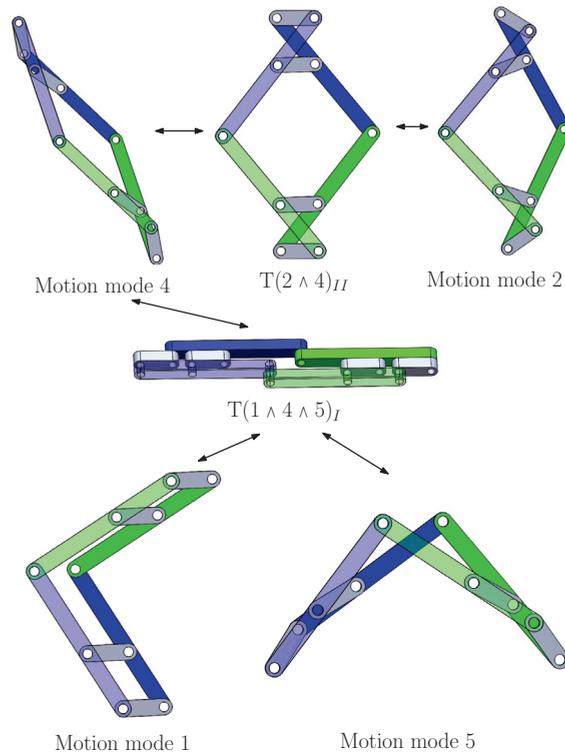


Figure 7. A variable-DOF 8-link planar mechanism in compact design that can transit among four motion modes.

7. Conclusions

A novel 8-link variable-DOF planar mechanism with five motion modes has been proposed. Reconfiguration analysis has shown that the mechanism has one 2-DOF double parallelogram motion mode and four 1-DOF motion modes. In addition, the mechanism can switch among three motion modes at four transition configurations and between two motion modes at four other transition configurations.

In contrast to the two 8-link variable-DOF planar mechanisms in [5,47], which have four inactive joints in some of their motion modes, this novel 8-link variable-DOF planar mechanism has two inactive joints in one of its 1-DOF motion modes. The two closed-loop 4R kinematic sub-chains of the novel mechanism must appear either as a pair of parallelograms in the 2-DOF motion mode or a pair of anti-parallelograms in a 1-DOF motion mode.

The hybrid approach that combines elimination and computer algebraic geometry methods has been found to be more efficient than the algebraic geometry approach without elimination. As a by-product, a method for factoring trigonometric functions in two angles has been proposed.

This work, together with reference [47], provides a starting point for the design and analysis of variable-DOF multi-loop mechanisms constructed using more than two parallelograms, which could be used as reconfigurable/multi-mode deployable mechanisms.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/machines11050529/s1>, Video S1: Reconfiguration of a novel 8-link variable-DOF planar mechanism with five motion modes.

Author Contributions: Conceptualization, X.K. and J.W.; methodology, X.K.; validation, J.W. and X.K.; formal analysis, X.K.; writing—original draft preparation, X.K. and J.W.; writing—review and editing, X.K.; visualization, J.W. and X.K.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Engineering and Physical Sciences Research Council (EPSRC) grant number EP/T024844/1, United Kingdom.

Data Availability Statement: The data presented in this study are available within the paper.

Acknowledgments: The authors would like to thank Mary Kong from the University of Edinburgh for building the LEGO model of the 8-link variable-DOF mechanism shown in Figure 6.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivation of Equation (2)

Solving the set of equations composed of the first and second equations in Equation (1) as a set of linear equations in $C(\phi_b + \phi_a)$ and $C\phi_b + C\phi_a$, we have

$$\begin{cases} C(\phi_a + \phi_b) = k_1 \\ C\phi_a + C\phi_b = k_2 \end{cases} \quad (\text{A1})$$

To eliminate ϕ_b from Equation (A1), rewrite the first equation in Equation (A1) as

$$-S\phi_a S\phi_b = k_1 - C\phi_a C\phi_b$$

Squaring both sides, we have

$$(-S\phi_a S\phi_b)^2 = (k_1 - C\phi_a C\phi_b)^2$$

Eliminating $S\phi_a$ and $S\phi_b$ from the above equation using the trigonometric identities $S^2\phi_a + C^2\phi_a = 1$ and $S^2\phi_b + C^2\phi_b = 1$, we have

$$(1 - C^2\phi_a)(1 - C^2\phi_b) - (k_1 - C\phi_a C\phi_b)^2 = 0 \quad (\text{A2})$$

Solving the second equation in Equation (A1) for $C\phi_b$, we obtain

$$C\phi_b = k_2 - C\phi_a \quad (\text{A3})$$

Substituting Equation (A3) into Equation (A2), we obtain the following equation in L and ϕ_a .

$$(1 - C^2\phi_a)[1 - (k_2 - C\phi_a)^2] - [k_1 - C\phi_a(k_2 - C\phi_a)]^2 = 0 \quad (\text{A4})$$

Similarly, the third and fourth equations in Equation (1) can be reduced to

$$(1 - C^2\psi_a)[1 - (k_2 - C\psi_a)^2] - [k_1 - C\psi_a(k_2 - C\psi_a)]^2 = 0 \quad (\text{A5})$$

Combining Equations (A4) and (A5), we obtain Equation (2).

Appendix B. Derivation of Equation (11)

Equation (11) can be derived from Equation (8) using the primary decomposition of ideals in the following five steps.

Step 1: Convert Equation (8) into a polynomial equation.

Substituting $C\phi_a = ca$ and $C\psi_a = cb$ into Equation (8), we obtain a polynomial equation in ca and cb .

$$f = 0 \quad (\text{A6})$$

where $f = 72,900ca^4 - 5940ca^3cb - 215,280ca^2cb^2 - 5940cacb^3 + 72,900cb^4 + 63,661ca^2 + 12,122cacb + 63,661cb^2 - 57,600$.

Step 2: Calculate the primary decomposition of ideal $\mathcal{J} = \langle f, ca^2 + sa^2 - 1, cb^2 + sb^2 - 1 \rangle$, where $sa = S\phi_a$ and $sb = S\psi_a$. The last two polynomials correspond to the trigonometric identities $S^2\phi_a + C^2\phi_a = 1$ and $S^2\psi_a + C^2\psi_a = 1$.

Calculating the primary decomposition of \mathcal{J} using computer algebra system software, such as MAPLE command *PrimeDecomposition(J, 'removedundant')*, we have

$$\mathcal{J} = \bigcap_{j=1}^2 \mathcal{J}_j \tag{A7}$$

where the irreducible components, \mathcal{J}_1 and \mathcal{J}_2 , of \mathcal{J} are

$\mathcal{J}_1 = \langle ca^2 + sa^2 - 1, cb^2 + sb^2 - 1, -270ca^2 + 11cacb - 270cb^2 + 601sasb + 551, 270ca^2sa + 601ca^2sb - 11cacbsa + 270cb^2sa - 551sa - 601sb, -162,270ca^2sa - 288,301ca^2sb + 6611cacbsa - 2970cacbsb + 72,900cb^2sb + 168,881sa + 212,431sb, 72,900ca^4 - 5940ca^3cb - 215,280ca^2cb^2 - 5940cacb^3 + 72,900cb^4 + 63,661ca^2 + 12,122cacb + 63,661cb^2 - 576,00 \rangle$, and $\mathcal{J}_2 = \langle ca^2 + sa^2 - 1, cb^2 + sb^2 - 1, 270ca^2 - 11cacb + 270cb^2 + 601sasb - 551, 270ca^2sa - 601ca^2sb - 11cacbsa + 270cb^2sa - 551sa + 601sb, 162,270ca^2sa - 288,301ca^2sb - 6611cacbsa - 2970cacbsb + 72,900cb^2sb - 168,881sa + 212,431sb, 72,900ca^4 - 5940ca^3cb - 215,280ca^2cb^2 - 5940cacb^3 + 72,900cb^4 + 63,661ca^2 + 12,122cacb + 63,661cb^2 - 57,600 \rangle$.

Step 3: Calculate the Gröbner basis for each irreducible component.

Using the MAPLE command, *Basis(J1, tdeg(sa, ca, sb, cb))*, we obtain the Gröbner basis of \mathcal{J}_1 as

$\mathcal{J}'_1 = \langle cb^2 + sb^2 - 1, 270ca^2 - 11cacb + 270cb^2 - 601sasb - 551, 11cacb - 270cb^2 + 270sa^2 + 601sasb + 281 \rangle$.

Similarly, the Gröbner basis of \mathcal{J}_2 is

$\mathcal{J}'_2 = \langle cb^2 + sb^2 - 1, 270ca^2 - 11cacb + 270cb^2 + 601sasb - 551, 11cacb - 270cb^2 + 270sa^2 - 601sasb + 281 \rangle$.

Step 4: Convert the polynomials in each of the irreducible components into trigonometrical functions.

Substituting $ca = C\phi_a, sa = S\phi_a, cb = C\psi_a$ and $sb = S\psi_a$ into \mathcal{J}'_1 and simplifying the results, we obtain

$\mathcal{J}'_{11} = \langle C^2\psi_a + S^2\psi_a - 1, 270C^2\phi_a - 11C\phi_aC\psi_a + 270C^2\psi_a + 601S\phi_aS\psi_a - 551, 11C\phi_aC\psi_a - 270C^2\psi_a + 270S^2\phi_a - 601S\phi_aS\psi_a + 281 \rangle$. i.e., $\mathcal{J}'_1 = \langle 0, g_1, -g_1 \rangle$

where $g_1 = -281 + 135C(2\phi_a) + 295C(\phi_a + \psi_a) - 306C(\phi_a - \psi_a) + 135C(2\psi_a)$.

Similarly, we obtain $\mathcal{J}'_2 = \langle 0, g_2, -g_2 \rangle$

where $g_2 = -281 + 135C(2\phi_a) + 295C(\phi_a - \psi_a) - 306C(\phi_a + \psi_a) + 135C(2\psi_a)$.

Step 5: Divide the trigonometrical function in Equation (8) by the product of the trigonometrical functions obtained in Step 4.

Divide g by g_1g_2 , we can readily obtain

$$g / (g_1g_2) = 1$$

i.e.

$$g = g_1g_2 \tag{A8}$$

Substituting Equation (A8) into Equation (8), we obtain Equation (11).

References

1. Xi, F.; Dai, J.S.; Ding, X.; van der Wijk, V. *Proceedings of the 5-th IEEE/IFToMM International Conference on Reconfigurable Mechanisms and Robots, Toronto, ON, Canada, 12–14 August 2021*; Ryerson University: Toronto, ON, Canada, 2021.
2. Herder, J.; van der Wijk, V. *Proceedings of the 2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR 2018)*; IEEE: Piscataway, NJ, USA, 2018.
3. Wohlhart, K. Kinematotropic linkages. In *Recent Advances in Robot Kinematics*; Lenarčič, J., Parenti-Castelli, V., Eds.; Kluwer Academic: Dordrecht, The Netherlands, 1996; pp. 359–368.
4. Kovalev, M.D. Geometric theory of hinge devices. *Russian Acad. Sci. Izv. Math.* **1995**, *44*, 43–68.
5. Galletti, C.; Fanghella, P. Single-loop kinematotropic mechanisms. *Mech. Mach. Theory* **2001**, *36*, 743–761. [CrossRef]
6. Diez-Martínez, C.R. Mobility and Connectivity in Spatial Kinematic Chains. Master's Thesis, Technological Institute of Celaya, Celaya, Mexico, 2005. (In Spanish)
7. Kong, X.; Pflurner, M. Type synthesis and reconfiguration analysis of a class of variable-DOF single-loop mechanisms. *Mech. Mach. Theory* **2015**, *85*, 116–128. [CrossRef]
8. López-Custodio, P.C.; Dai, J.S. Design of a variable-mobility linkage using the Bohemian dome. *ASME J. Mech. Des.* **2019**, *141*, 092303. [CrossRef]
9. Feng, H.; Chen, Y.; Dai, J.S.; Gogu, G. Kinematic study of the general plane-symmetric Bricard linkage and its bifurcation variations. *Mech. Mach. Theory* **2017**, *116*, 89–104. [CrossRef]
10. Lopez-Custodio, P.C.; Rico, J.M.; Cervantes-Sánchez, J.J.; Perez-Soto, G.I. Reconfigurable mechanisms from the intersection of surfaces. *ASME J. Mech. Robot.* **2016**, *8*, 021029. [CrossRef]
11. Fanghella, P.; Galletti, C.; Giannotti, E. Parallel robots that change their group of motion. In *Advances in Robot Kinematics*; Lenarčič, J., Roth, B., Eds.; Springer: Dordrecht, The Netherlands, 2006; pp. 49–56.
12. Liu, K.; Yu, J.; Kong, X. Structure synthesis and reconfiguration analysis of variable-degree-of-freedom single-loop mechanisms with prismatic joints using dual quaternions. *ASME J. Mech. Robot.* **2022**, *14*, 021009. [CrossRef]
13. Kong, X. Type synthesis of variable degree-of-freedom parallel manipulators with both planar and 3T1R operation modes. In *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Chicago, IL, USA, 12–15 August 2012; DETC2012-70621.
14. Coste, M.; Demdah, K.M. Extra modes of operation and self motions in manipulators designed for Schoenflies motion. *ASME J. Mech. Robot.* **2015**, *7*, 041020. [CrossRef]
15. Zeng, Q.; Ehmann, K.F.; Cao, J. Design of general kinematotropic mechanisms. *Robot. Comput. Integr. Manuf.* **2016**, *38*, 67–81. [CrossRef]
16. Zhang, K.; Dai, J.S. Screw-system-variation enabled reconfiguration of the Bennett plano-spherical hybrid linkage and its evolved parallel mechanism. *ASME J. Mech. Des.* **2015**, *137*, 062303. [CrossRef]
17. Nurahmi, L.; Caro, S.; Wenger, P.; Schadlbauer, J.; Husty, M. Reconfiguration analysis of a 4-RUU parallel manipulator. *Mech. Mach. Theory* **2016**, *96*, 269–289. [CrossRef]
18. Nurahmi, L.; Putrayudanto, P.; Wei, G.; Agrawal, S.K. Geometric constraint-based reconfiguration and self-motions of a four-CRU parallel mechanism. *ASME J. Mech. Robot.* **2021**, *13*, 433748391. [CrossRef]
19. Kong, X. Classification of a 3-RER parallel manipulator based on the type and number of operation modes. *ASME J. Mech. Robot.* **2021**, *13*, 021013. [CrossRef]
20. López-Custodio, P.C.; Müller, A.; Dai, J.S. A kinematotropic parallel mechanism reconfiguring between three motion branches of different mobility. In *Advances in Mechanism and Machine Science. IFToMM WC 2019*; Uhl, T., Ed.; Springer: Cham, Switzerland, 2019; pp. 2611–2620.
21. Liu, Y.; Li, Y.; Yao, Y.-A.; Kong, X. Type synthesis of multi-mode mobile parallel mechanisms based on refined virtual chain approach. *Mech. Mach. Theory* **2020**, *152*, 103908. [CrossRef]
22. Qin, Y.; Dai, J.S.; Gogu, G. Multi-furcation in a derivative queer-square mechanism. *Mech. Mach. Theory* **2014**, *81*, 36–53. [CrossRef]
23. Kong, X. Variable degree-of-freedom spatial mechanisms composed of four circular translation joints. *ASME J. Mech. Robot.* **2021**, *13*, 031007. [CrossRef]
24. Arponen, T.; Piipponen, S.; Tuomela, J. Kinematical analysis of Wunderlich mechanism. *Mech. Mach. Theory* **2013**, *70*, 16–31. [CrossRef]
25. Wang, J.; Kong, X. A novel method for constructing multimode deployable polyhedron mechanisms using symmetric spatial compositional units. *ASME J. Mech. Robot.* **2019**, *11*, 020907. [CrossRef]
26. Tian, C.; Zhang, D.; Tang, H.; Wu, C. Structure synthesis of reconfigurable generalized parallel mechanisms with configurable platforms. *Mech. Mach. Theory* **2021**, *160*, 104281. [CrossRef]
27. Liu, R.; Li, R.; Yao, Y.-A.; Ding, X. A reconfigurable deployable spatial 8R-like mechanism consisting of four angulated elements connected by R joints. *Mech. Mach. Theory* **2023**, *179*, 105103. [CrossRef]
28. Laliberté, T.; Gosselin, C. Construction, mobility analysis and synthesis of polyhedra with articulated faces. *ASME J. Mech. Robot.* **2014**, *6*, 011007. [CrossRef]
29. Overvelde, J.; Weaver, J.; Hoberman, C.; Bertoldi, K. Rational design of reconfigurable prismatic architected materials. *Nature* **2017**, *54*, 347–352. [CrossRef] [PubMed]

30. Yan, H.; Kuo, C. Topological representations and characteristics of variable kinematic joints. *ASME J. Mech. Des.* **2006**, *128*, 384–391. [CrossRef]
31. Gan, D.; Dai, J.S.; Caldwell, D.G. Constraint-Based Limb Synthesis and Mobility-Change-Aimed Mechanism Construction. *ASME J. Mech. Des.* **2011**, *133*, 051001. [CrossRef]
32. Jia, P.; Li, D.; Zhang, Y.; Yang, C. A novel reconfigurable parallel mechanism constructed with spatial metamorphic four-link mechanism. *Proc. Inst. Mech. Eng. Part C* **2022**, *236*, 4120–4132. [CrossRef]
33. Li, Z.; Scharler, D.F.; Schröcker, H.-P. Factorization results for left polynomials in some associative real algebras: State of the art, applications, and open questions. *J. Comput. Appl. Math.* **2019**, *349*, 508–522. [CrossRef]
34. Montes, A.; Wibmer, M. Software for discussing parametric polynomial systems: The Gröbner Cover. In *Mathematical Software—ICMS 2014*; Hong, H., Yap, C., Eds.; Springer: Berlin, Germany, 2014; pp. 406–413.
35. Cox, D.A.; Little, J.B.; O’Shea, D. *Ideals, Varieties, and Algorithms*; Springer: New York, NY, USA, 2007.
36. Husty, M.L.; Schröcker, H.-P. Kinematics and algebraic geometry. In *21st Century Kinematics*; McCarthy J.M., Ed.; Springer: London, UK, 2013; pp. 85–123.
37. Wampler, C.; Sommese, A. Numerical algebraic geometry and algebraic kinematics. *Acta Numer.* **2011**, *20*, 469–567. [CrossRef]
38. Shabani, A.; Porta, J.M.; Thomas, F. A branch-and-prune method to solve closure equations in dual quaternions. *Mech. Mach. Theory* **2021**, *164*, 104424. [CrossRef]
39. Pellegrino, S. Structural computations with the singular value decomposition of the equilibrium matrix. *Int. J. Solids Struct.* **1993**, *30*, 3025–3035. [CrossRef]
40. Song, C.Y.; Chen, Y.; Chen, I.-M. A 6R linkage reconfigurable between the line-symmetric Bricard linkage and the Bennett linkage. *Mech. Mach. Theory* **2013**, *70*, 278–292. [CrossRef]
41. Wang, Y.; Zhang, Q.; Zhang, X.; Cai, J.; Jiang, C.; Xu, Y.; Feng, J. Analytical and numerical analysis of mobility and kinematic bifurcation of planar linkages. *Int. J. Non-Linear Mech.* **2022**, *145*, 104110. [CrossRef]
42. Kang, X.; Lei, H.; Li, B. Multiple bifurcated reconfiguration of double-loop metamorphic mechanisms with prismatic joints. *Mech. Mach. Theory* **2022**, *178*, 105081. [CrossRef]
43. Müller, A. Local kinematic analysis of closed-loop linkages—Mobility, singularities, and shakiness. *ASME J. Mech. Robot.* **2016**, *8*, 041013. [CrossRef]
44. Lopez-Custodio, P.C.; Rico, J.M.; Cervantes-Sanchez, J.J.; Perez-Soto, G.I.; Díez-Martínez, C.R. Verification of the higher order kinematic analyses equations. *Eur. J. Mech. A Solids* **2017**, *61*, 198–215. [CrossRef]
45. Lopez-Custodio, P.C.; Rico, J.M.; Cervantes-Sanchez, J.J. Local analysis of helicoid-helicoid intersections in reconfigurable linkages. *ASME J. Mech. Robot.* **2017**, *9*, 031008. [CrossRef]
46. Müller, A. Recursive higher-order constraints for linkages with lower kinematic pairs. *Mech. Mach. Theory* **2016**, *100*, 33–43. [CrossRef]
47. Kong, X. A novel construction method for the type synthesis of variable-DOF mechanisms. In *Proceedings of the 5th IEEE/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, Toronto, ON, Canada, 12–14 August 2021; Xi, F., Dai, J.S., Ding, X., van der Wijk, V., Eds.; Ryerson University: Toronto, ON, Canada, 2021; pp. 31–40.
48. Kong, X. Motion/structure mode analysis and classification of n-RR planar parallelogram mechanisms. *Mech. Mach. Theory* **2022**, *169*, 104623. [CrossRef]
49. Dhingra, A.K.; Almadi, A.N.; Kohli, D. Closed-form displacement analysis of 8, 9 and 10-link mechanisms: Part I: 8-link 1-DOF mechanisms. *Mech. Mach. Theory* **2000**, *35*, 821–850. [CrossRef]
50. Qiao, S.; Liao, Q.; Wei, S.; Su, H.-J. Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions. *Mech. Mach. Theory* **2010**, *45*, 193–199. [CrossRef]
51. Kiper, G.; Gürcü, F.; Korkmaz, K.; Söylemez, E. Kinematic design of a reconfigurable deployable canopy. In *New Trends in Mechanism and Machine Science*; Flores, P., Viadero, F., Eds.; Springer: Cham, Switzerland, 2015; pp. 167–174.
52. Gao, Y.; Yang, F.; Zhang, J. A reconfigurable 6R linkage with six motion modes and three topological structures. *ASME J. Mech. Robot.* **2023**, *15*, 054503. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Kinematic Analysis of a Spatial Cable-Driven Mechanism and Its Equivalent Hybrid Mechanism for Elliptical Trajectory

Li Wu ¹, Xuan Liu ¹, Mingjun Wang ¹, Dengbiao Zhao ² and Lewei Tang ^{3,*}¹ College of Automotive Engineering, Xiangtan Institute of Technology, Xiangtan 410219, China² Department of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China³ College of Mechanical and Electrical Engineering, Hunan Agricultural University, Changsha 410128, China

* Correspondence: tang_lewei@126.com; Tel.: +86-18942508376

Abstract: In this paper, a spatial cable-driven parallel mechanism in a V-shaped cable arrangement is proposed. It is further simplified as a planar hybrid cable-driven parallel mechanism to analyze its kinematics, which consists of two identical active cable chains and a passive cross-slide mechanism. In order to investigate the degrees of freedom (DoFs) of the hybrid mechanism using screw theory, cable chains are represented as rotational–prismatic–rotational (RPR) chains. The motion pairs of all the chains are denoted according to screw theory. Firstly, the number and the types of DoFs of each chain are determined. Then, the number and the types of DoFs for the hybrid mechanism are calculated. Furthermore, the theoretical result is verified using the modified Grübler–Kutzbach (G-K) formula. It shows that the unique DoF of the equivalent mechanism is a rotation with a continuously changing axis, which is consistent with the V-type cable-driven mechanism with elliptical trajectories. Finally, the kinematics analysis of the cross-slider mechanism driven by two cables is carried out. The length, velocity and acceleration of the cables are obtained from numerical calculation in MATLAB, and the results are demonstrated using ADAMS simulation.

Keywords: screw theory; degrees of freedom; kinematic screw; cable-driven mechanism; cross-slide mechanism; kinematic analysis

Citation: Wu, L.; Liu, X.; Wang, M.; Zhao, D.; Tang, L. Kinematic Analysis of a Spatial Cable-Driven Mechanism and Its Equivalent Hybrid Mechanism for Elliptical Trajectory. *Machines* **2023**, *11*, 710. <https://doi.org/10.3390/machines11070710>

Academic Editors: Raffaele Di Gregorio and Dan Zhang

Received: 22 May 2023

Revised: 22 June 2023

Accepted: 3 July 2023

Published: 4 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cable-driven parallel mechanisms are a special type of parallel mechanism driven by flexible cables, which have been widely used in several applications [1–4]. One prominent advantage of cable-driven mechanisms is their capability of fast transportation with high payload over a large-span distance. A mobile cable-driven parallel robot proposed in [5] is capable of changing its geometric architecture to increase the wrench-feasible workspace for specific tasks. To improve the mobility of end-users across a river or other civil structures, a cable-suspended robot with merely prismatic DoFs is studied by Castelli in [6]. The singularity analysis of cable-driven parallel robots in a pairwise cable arrangement is performed in [7]. Cable-suspended manipulators can also be used in industrial environments for pick-place operations using a reconfigurable end-effector [8]. Furthermore, a particular application of cable-driven parallel mechanisms is the astronomical observation of deep space. Six cables are utilized to carry a feed-supporting platform over the range of a radius of 250 m for the purpose of collecting out-of-space signals in China [9,10].

In the present paper, a specific cable-driven parallel mechanism, assembled with V-shaped cable units, is proposed to fulfill the need of achieving large rotation for astronomy observation near the workspace boundaries. It will be described in Section 3 in detail. In particular, in observation tasks, a signal receiver of a large radio telescope has to be positioned around the border of workspace with a large rotational angle. However, it was demonstrated that the rotation angle is limited to less than 40 degrees in the present cable configuration [11]. Due to the cable flexibility, a cable-driven parallel configuration in a

V-type cable arrangement was adopted, as shown in Figure 1. Khakpour [12] analyzed the features of several V-shaped cable-driven planar manipulators with differential actuation. A basic unit of the V-shaped cable-driven parallel mechanism is the connection of both static and moving platforms with one cable via a pulley. As a result, the position of the attachment points of the moving pulleys follows an elliptical trajectory. A V-shaped configuration (i.e., a V-type configuration) was employed to construct a planar cable-driven robot with a parallelogram arrangement [13]. Based on the geometrical feature of an elliptical trajectory, the proposed cable-driven parallel mechanism is capable of realizing large rotation near the workspace boundary accordingly. The contouring control problem with uncertainty is solved for a five-DoF robotic manipulator [14]. Then, the first issue to be solved for the implementation of the proposed cable-driven robot was to determine the DoFs of the mechanism.

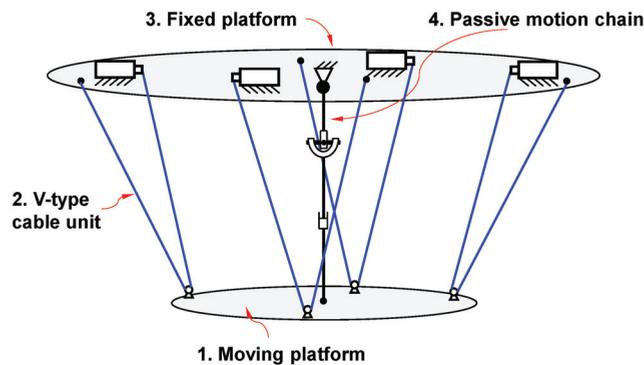


Figure 1. Cable-driven parallel mechanism with four V-shaped cable units.

Screw theory plays an important role in the study of robot configuration and freedom type. During the middle of the 20th century, Dimentberg [15] applied screw theory to first analyze the spatial mechanism. Later, Hunt [16] and Duffy [17] made great contributions to screw theory and its applications. Mohamed [18] exploited screw theory to investigate the kinematics of parallel mechanisms. Referring to [19–22], a breakthrough was made in the generalized method for the DoF analysis of spatial closed-loop mechanisms, especially multi-loop parallel mechanisms using screw theory.

Although screw theory is a versatile method to determine the DoFs of mechanisms, the mechanism has to be composed of only rigid links. To further simplify the kinematic analysis of the spatial cable-driven mechanism, an equivalent planar hybrid mechanism with both rigid links and cables is proposed to implement an elliptical trajectory. The determination of an alternative mechanism is essentially configuration synthesis. Three types of configuration synthesis methods include displacement group theory [23], screw theory [24] and topological theory [25]. Hu proposed the concept of a motion-equivalent parallel mechanism. The study showed that some parallel mechanisms with different structures have the same kinematics and performance. It provides better selection criteria for a specific configuration of parallel mechanisms [26–28]. According to screw theory, the kinematic constraint screws of a parallel mechanism are obtained. Then, from the relationship of the constraint screws and the motion screws, the corresponding parallel mechanism with identical motion but different configurations can be identified. In this process, the constraint screws of the mechanism may have changed, but the constraint space remains the same. The kinematic analysis of the present V-shaped cable-driven parallel mechanism is fundamental work for the astronomical application. It paves the way to the establishment of the dynamic modeling and motion control of the cable-driven parallel mechanism in a V-shaped arrangement.

The main contribution of this paper is to propose a new V-shaped cable-driven parallel mechanism for astronomical observation using elliptical trajectories. An equivalent planar hybrid mechanism is presented by investigating the kinematic constraints. The resulting DoF of the simplified hybrid mechanism was derived as a continuously changing axis along a global axis through screw theory. In addition, the kinematic analysis of the hybrid mechanism was performed and demonstrated using ADAMS simulation. The obtained results of the cable length can be applied in the control of the proposed V-shaped cable-driven parallel mechanism. To the knowledge of the authors, there is no literature on the kinematic analysis of the hybrid mechanism using screw theory, which has the potential to design a more complex cable-driven mechanism in a V-shaped cable configuration.

Except for its potential usage in astronomical observation, there are some practical applications with the requirement of an elliptical trajectory. For instance, elliptical trajectory motion is introduced in the field of pick-place industrial applications [29]. Robotic-assisted elliptical training is being widely employed to help patients retain walking ability [30], where the proposed cable-driven mechanism is suitable due to safety.

The rest of the paper is organized as follows: The preliminary definitions on screw theory are introduced in Section 2. The description of the proposed spatial cable-driven parallel mechanism and the equivalent planar hybrid mechanism are illustrated in Section 3. More importantly, the determination of DoFs for the equivalent mechanism was carried out using screw theory. The obtained results validated using the improved G-K formula are in Section 4. In Section 5, the kinematic analysis of the cable-driven cross-slider mechanism was carried out to further verify the DoF of the mechanism, and ADAMS simulation was performed to compare with the theoretical results. Finally, the conclusions obtained are in Section 6.

2. Preliminary on Screw Theory

Spiral, denoted as $\$$, is also referred to as screw, which is represented by a pair of dual vectors (s, s^0) in space:

$$\$ = (s; s^0) = (l, m, n; p, q, r), \quad s \cdot s^0 \neq 0, \tag{1}$$

where s is the main part, and s^0 is the dual part. When the inner product of a pair of dual parts is zero, i.e., $s \cdot s^0 = 0$, it means that the spiral degenerates into a line vector.

If the reciprocal product of two screws $\$ = (s; s^0)$ and $\$r = (s_r; s_r^0)$ is zero:

$$\$\cdot\$r = s \cdot s_r^0 + s_r \cdot s^0 = 0, \tag{2}$$

then $\$$ and $\$r$ are defined as two reciprocal screws. To calculate DoFs, screws are expressed as motion and constraint screws. For example, $\$ = (s; \mathbf{0}_{1 \times 3})$ denotes a pure rotation motion or a force vector at the origin of the coordinate system, while $\$ = (\mathbf{0}_{1 \times 3}; s^0)$ denotes a pure prismatic motion. As $\$$ represents a motion screw, the corresponding $\$r$ represents its constraint screw.

Each chain of a parallel mechanism imposes several constraints on the moving platform [31,32]. Moreover, synthesis of the constraint screws of each chain determines the DoFs of the moving platform [33]. All the motion pairs in a chain constitute the motion screw system of the chain. The reciprocal screw of the motion screw system constitutes the constraint screw system of the chain. The constraints of a chain imposed on the moving platform are determined accordingly. The constraint screw system of each chain synthesizes the entire constraint screw system of the mechanism, which represents the constrained DoFs of the parallel mechanism. Thus, the DoFs of the proposed mechanism are finally calculated on the basis of screw theory.

3. Equivalent Rigid Parallel Mechanism

The present cable-driven parallel mechanism consists of a moving platform (1), four V-type cable units (2), a fixed platform (3) and a passive motion chain (4), which is depicted in Figure 2. Each V-type cable unit formed by only one cable connects the moving platform and the fixed platform via a pulley attached on the moving platform. Four pulleys are located in a circular array at an angle of 90 degrees, as illustrated in Figure 2. One end of each cable is connected to a servo-motor directly or a winch, and the other end is fixed to the fixed platform at an anchor point, whose shape is described as ‘V’. The cables are retracted or extended such that an elliptical motion of the moving platform is implemented. There are two mounting approaches for four cable units. One approach corresponds to the case that the plane of the cable is always perpendicular to the moving platform, while the other produces a slope with respect to the moving platform. The passive motion chain consists of a hook hinge and a prismatic joint, which is applied to constrain the feasible motion of the proposed mechanism on Plane 1, as shown in Figure 2. Additionally, the angular bisectors of sectors formed by V-type cable units are limited to being on Plane 1 and Plane 2, respectively. Moreover, due to V-type differential cable arrangement and corresponding constraints imposed by the passive chain, the motion trajectory of the moving platform is limited to being elliptical. It is also noted that only one pair of cable units works actively, while the other pair maintains a constant cable length.

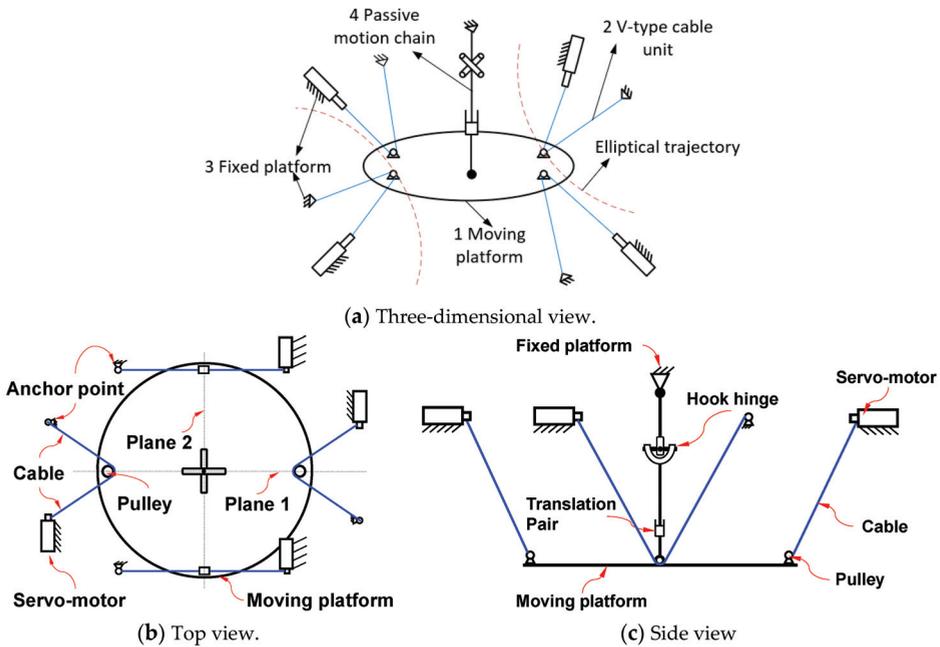


Figure 2. Sketch of cable-driven parallel mechanism for elliptical trajectory.

For the convenience of analysis, a planar hybrid cable-driven mechanism for elliptical trajectory is presented to investigate as a simplified mechanism, as shown in Figure 2. Furthermore, a cross-slide rigid mechanism was exploited to substitute V-shaped cable units to achieve elliptical trajectories. According to the cable configuration, two RPR chains were employed to replace the active cable chains, while the passive cable chain was driven by the cross-slide mechanism in Figure 3.

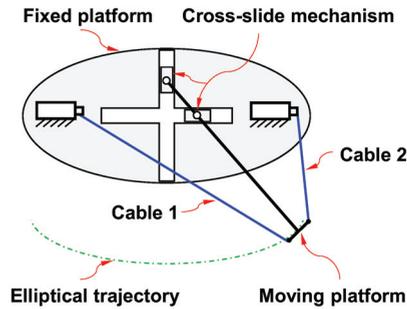


Figure 3. Cross-slide mechanism.

Thus, the simplified planar hybrid cable-driven mechanism is represented by a rigid parallel mechanism of 2RPR-PRRP, as shown in Figure 4, where R denotes rotational pair, and P means prismatic pair.

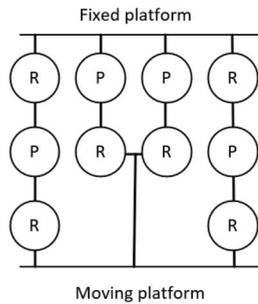


Figure 4. Equivalent mechanism diagram of a rigid parallel mechanism.

As there is a closed loop sub-chain, it is necessary to develop the equivalent motion screw system of the cross-slide mechanism (prismatic–rotational–rotational–prismatic, namely, PRRP), and the specific steps are given as follows:

- (1) List the motion screw system of each chain;
- (2) Find the reverse screw of (1);
- (3) The quadratic anti-screw of (2) is solved to obtain kinematic screws of the mechanism allowed by constraint screws.

The closed-loop chain of the cross-slide mechanism is transformed into two sub-chains, which are regarded as 2PR (Prismatic-Rotational) sub-chains. The motion pair of a horizontal sub-chain was selected as the origin of the coordinate system. x axis is located along the axis of rotational pair, y axis is along the motion direction of prismatic pair, and z axis is defined according to the right-hand rule, as shown in Figure 5.

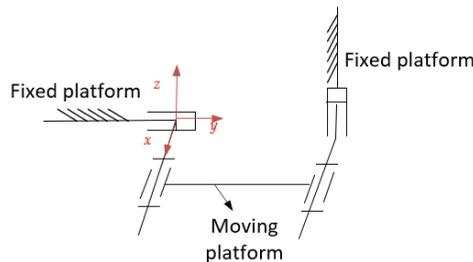


Figure 5. Closed-loop of sub-chain.

Chain 1 is composed of a prismatic pair and a rotational pair. The motion screw system is given as

$$\mathcal{S}_{11} = (0, 0, 0; 0, 1, 0), \mathcal{S}_{12} = (1, 0, 0; 0, 0, 0). \tag{3}$$

Screws \mathcal{S}_{11} and \mathcal{S}_{12} represent the prismatic motion along y axis and the rotation motion around x axis. As two screws are independent, the number of anti-screws is $6 - 2 = 4$. According to the definition of anti-screws, one can obtain

$$\begin{aligned} \mathcal{S}_{11}^r &= (0, 1, 0; 0, 0, 0), \\ \mathcal{S}_{12}^r &= (0, 0, 1; 0, 0, 0), \\ \mathcal{S}_{13}^r &= (0, 0, 0; 1, 0, 0), \\ \mathcal{S}_{14}^r &= (0, 0, 0; 0, 0, 1). \end{aligned} \tag{4}$$

These represent the constraint forces along x, z axis and the constraint moments around y, z axis, respectively.

As for Chain 2, it also includes a rotational pair and a prismatic pair, whose motion screw system $\mathcal{S}_2 = \{\mathcal{S}_{21}, \mathcal{S}_{22}\}$ is

$$\mathcal{S}_{21} = (0, 0, 0; 0, 0, 1), \mathcal{S}_{22} = (1, 0, 0; q_{22}, 0, 0). \tag{5}$$

where q_{22} represents the distance of common normal between the rotational axis and y axis. Regarding \mathcal{S}_2 , the two screws are independent. Similarly, the number of anti-screws is obtained as $6 - 2 = 4$, and one can get the following screws as

$$\begin{aligned} \mathcal{S}_{21}^r &= (0, 1, 0; -q_{22}, 0, 0), \\ \mathcal{S}_{22}^r &= (0, 0, 1; 0, 0, 0), \\ \mathcal{S}_{23}^r &= (0, 0, 0; 1, 0, 0), \\ \mathcal{S}_{24}^r &= (0, 0, 0; 0, 1, 0), \end{aligned} \tag{6}$$

Finally, the quadratic inverse screw of the inverse screw system involving Equations (4)–(6) is solved. The motion screws of the mechanism restrained by the constraint screws are determined. As the rank r of the anti-screw system is 5, the number of the secondary anti-screws of the anti-screw system is $6 - 5 = 1$, and the solution is yielded as

$$\mathcal{S}^{rr} = (1, 0, 0; 0, q_{22}, 0). \tag{7}$$

It means that the moving platform can rotate around the axis parallel to x axis. Thus, the closed-loop sub-chain in Figure 4 can be equivalent to a rotation pair with a continuously changing axis, and the final equivalent parallel mechanism configuration diagram is illustrated in Figure 6.

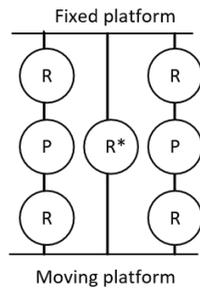


Figure 6. Configuration diagram of the equivalent rigid mechanism with R^* around continuously changing axis.

Next, constraints of each chain of the equivalent rigid parallel mechanism on the moving platform were studied. A new coordinate system was established at the axis of the rotational pair. x axis is along the axis of the rotation pair, z axis is perpendicular to the moving platform, and y axis is determined by the right-hand rule, as shown in Figure 7.

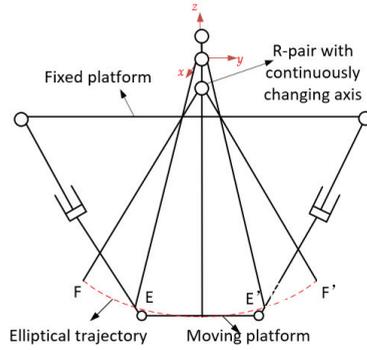


Figure 7. Coordinate system of the equivalent rigid mechanism with R*.

The motion screw system of each chain was calculated. For Chain 1, the motion screw is

$$S_{11} = (1, 0, 0; 0, 0, 0) \tag{8}$$

The constraint inverse screws are

$$\begin{aligned} S_{11}^f &= (1, 0, 0; 0, 0, 0) \\ S_{12}^f &= (0, 1, 0; 0, 0, 0) \\ S_{13}^f &= (0, 0, 1; 0, 0, 0) \\ S_{14}^f &= (0, 0, 0; 0, 1, 0) \\ S_{15}^f &= (0, 0, 0; 0, 0, 1) \end{aligned} \tag{9}$$

Equation (9) shows that Chain 1 imposes constraint force along x , y , z axis and constraint moments around y axis and around z axis on the moving platform.

For Chain 2, in the coordinate system, its motion screws are expressed as

$$\begin{aligned} S_{21} &= (1, 0, 0; 0, q_{21}, r_{21}) \\ S_{22} &= (0, 0, 0; 0, q_{22}, r_{22}) \\ S_{23} &= (1, 0, 0; 0, q_{23}, r_{23}) \end{aligned} \tag{10}$$

Then, the corresponding constraint inverse screws are solved as

$$\begin{aligned} S_{21}^f &= (1, 0, 0; 0, 0, 0) \\ S_{22}^f &= (0, 0, 0; 0, 1, 0) \\ S_{23}^f &= (0, 0, 0; 0, 0, 1) \end{aligned} \tag{11}$$

Equation (11) shows that Chain 2 imposes constraint forces along x , y , z axis on the moving platform. Similarly, for symmetric Chain 3, the anti-screw of its motion screw is:

$$\begin{aligned} S_{31}^f &= (1, 0, 0; 0, 0, 0) \\ S_{32}^f &= (0, 0, 0; 0, 1, 0) \\ S_{33}^f &= (0, 0, 0; 0, 0, 1) \end{aligned} \tag{12}$$

Based on Equations (10)–(12), the reciprocal product of three constraint screws and all motion screws is zero, thus the equivalent rigid parallel mechanism has three common constraints. The linear independent anti-screw system is

$$\begin{aligned} \$1^r &= (1, 0, 0; 0, 0, 0) \\ \$2^r &= (0, 1, 0; 0, 0, 0) \\ \$3^r &= (0, 0, 1; 0, 0, 0) \\ \$4^r &= (0, 0, 0; 0, 1, 0) \\ \$5^r &= (0, 0, 0; 0, 0, 1) \end{aligned} \quad (13)$$

It can be clearly observed that the chains restrict the rotation of the moving platform in y, z directions and the prismatic motion in x, y, z directions. In conclusion, the parallel mechanism has only one rotational DoF with a continuously changing axis, which shows the equivalence of the V-type cable-driven parallel mechanism [34].

4. Verification of DoFs of the Mechanism

The conventional solution formula of the DoFs of the mechanism is given as a G-K formula

$$M = 6(n - g - 1) + \sum_{i=1}^n f_i \quad (14)$$

where M represents the DoFs of the mechanism, n is the number of components of the mechanism (including the moving platform and the fixed platform), g is the number of motion pairs in the mechanism, and f_i is the number of DoFs of the i th motion pair. More importantly, the modified G-K formula was derived by considering the geometric constraints as

$$M = d(n - g - 1) + \sum_{i=1}^n f_i + v - \zeta, \quad (15)$$

where d means the order of the mechanism $d = 6 - \lambda$, v stands for redundant constraint number, λ is common constraint number, and ζ is the number of local DoFs of the mechanism. Thus, the modified G-K formula was used to check the DoFs of the hybrid mechanism.

Case 1: The DoFs of the mechanism including the PRRP-closed loop are analyzed. The value of the parameters are given as follows:

$$\lambda = 3, d = 6 - \lambda = 3, n = 8, g = 10, \sum_{i=1}^n f_i = 10, v = 0, \zeta = 0.$$

Substitute the above values into Equation (15) and yield $M = 1$.

Case 2: The DoFs of the final equivalent mechanism with R^* are analyzed. The parameters are calculated as

$$\lambda = 3, d = 6 - \lambda = 3, n = 6, g = 7, \sum_{i=1}^n f_i = 7, v = 0, \zeta = 0.$$

Take these above values into Equation (15), and $M = 1$ is obtained in accordance with the proposed parallel mechanism design for elliptical trajectories.

5. Kinematic Analysis of Hybrid Mechanism

The kinematic analysis of the hybrid mechanism was divided into position analysis, velocity analysis and acceleration analysis, where position analysis was the basis of velocity analysis and acceleration analysis.

5.1. Position Analysis

Position analysis of the cross-slider mechanism is dependent on the elliptical trajectory of the moving platform to calculate the cable length. For the hybrid mechanism, the connection points between the cable and the moving platform are denoted as A_i and B_i , respectively. The relative coordinate system P-xy and the global coordinate system O-XY

are established. The origin P and O of the coordinate systems are located at the geometric center of the moving and fixed platforms, respectively. The lengths of the moving platform and the fixed platform are denoted as $2e$ and $2f$, respectively. Then, in the fixed coordinate system O-XY, the coordinates of A_i ($i = 1,2$) in the global coordinate system are represented as $A_1 = (-e, 0)$, $A_2 = (e, 0)$. Similarly, in the relative coordinate system P-xy, the coordinates of B_i ($i = 1,2$) on the moving platform are presented as $B_1 = (-f, 0)$, $B_2 = (f, 0)$, as shown in Figure 8.

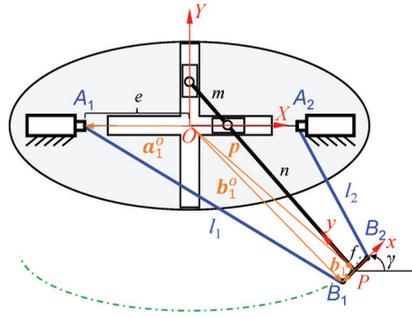


Figure 8. Parameters of the hybrid planar mechanism.

Suppose the rotation matrix from P-xy to O-XY:

$$R = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix}, \tag{16}$$

where γ means the Euler angle of the relative coordinate system in the global coordinate system. The position vector b_i^O in the global coordinate system is obtained by the position vector b_i in the relative coordinate system:

$$b_i^O = Rb_i + p, \quad (i = 1,2), \tag{17}$$

where $p = [x, y]^T$. According to the principle of a closed vector loop, the length vectors of the two driven cables in the global coordinate system are represented as

$$l_i = b_i^O - a_i^O = Rb_i + p - a_i^O, \quad (i = 1,2). \tag{18}$$

The direction of the vector l_i is expressed from A_i to B_i . Accordingly, the cable length l_i is obtained as below:

$$l_i = \|l_i\| = \|Rb_i + p - a_i^O\| = \sqrt{I_i^T I_i}, \quad (i = 1,2). \tag{19}$$

The analytical expressions of two cable lengths are given as

$$\begin{aligned} l_1^2 &= (-f\cos\gamma + x + e)^2 + (-f\sin\gamma + y)^2, \\ l_2^2 &= (f\cos\gamma + x - e)^2 + (f\sin\gamma + y)^2. \end{aligned} \tag{20}$$

where $x = (m + n)\cos\alpha$, $y = n\sin\alpha$, m is the length of the rod between Slider 1 and Slider 2, and n is the length of the rod between Slider 2 and the moving platform. The relation between α and γ is obtained from the tangent equation of the ellipse

$$\frac{n}{(m + n)} \cot\alpha = \tan\gamma. \tag{21}$$

From Equations (20) and (21), one can clearly observe that only one parameter, γ , is independent, which is consistent with the analytical DoF of the mechanism in Section 3.

5.2. Velocity Analysis

In this paper, the attitude change is described as the moving platform driven by two cables to a prescribed position. The attitude of the moving platform is controlled by manipulating the lengths of the two cables. In a summary, the relationship between the velocity of each cable and the attitude change speed of the moving platform was analyzed.

By the derivation with respect to time t on both sides of Equation (19), the velocity of each cable is obtained as

$$l_i \frac{dl_i}{dt} = (\mathbf{R}\mathbf{b}_i + \mathbf{p} - \mathbf{a}_i^O)^T \left(\frac{d\mathbf{R}}{dt} \mathbf{b}_i + \frac{d\mathbf{p}}{dt} \right), \quad (i = 1, 2), \tag{22}$$

where $\frac{d\mathbf{R}}{dt}$, the derivative of the rotation matrix, can be expressed as

$$\frac{d\mathbf{R}}{dt} = \boldsymbol{\omega} \times \mathbf{R} = \mathbf{S}(\boldsymbol{\omega})\mathbf{R} = \begin{bmatrix} 0 & -\dot{\gamma} \\ \dot{\gamma} & 0 \end{bmatrix} \mathbf{R}, \tag{23}$$

where $\boldsymbol{\omega}$ represents the angular velocity of the moving platform, and $\mathbf{S}(\boldsymbol{\omega})$ is an anti-symmetric matrix concerning the angular velocity vector $\boldsymbol{\omega}$. Substitute Equation (23) into Equation (22), then one obtains

$$l_i \frac{dl_i}{dt} = (\mathbf{R}\mathbf{b}_i + \mathbf{p} - \mathbf{a}_i^O)^T (\boldsymbol{\omega} \times \mathbf{R}\mathbf{b}_i + \mathbf{v}), \quad (i = 1, 2), \tag{24}$$

where \mathbf{v} is the linear velocity of the moving platform. Furthermore, Equation (22) can be transformed into

$$\dot{l}_i = \boldsymbol{\rho}_i^T \mathbf{v} + (\mathbf{R}\mathbf{b}_i \times \boldsymbol{\rho}_i)^T \boldsymbol{\omega}, \quad (i = 1, 2). \tag{25}$$

where $\boldsymbol{\rho}_i = \frac{\mathbf{l}_i}{\|\mathbf{l}_i\|}$. Therefore, the relationship between the velocity of the two cables and the generalized speed of the moving platform is derived as

$$\begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\rho}_1^T & (\mathbf{R}\mathbf{b}_1 \times \boldsymbol{\rho}_1)^T \\ \boldsymbol{\rho}_2^T & (\mathbf{R}\mathbf{b}_2 \times \boldsymbol{\rho}_2)^T \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}, \tag{26}$$

where the Jacobian matrix \mathbf{J} is defined as follows:

$$\mathbf{J} = \begin{bmatrix} \boldsymbol{\rho}_1^T & (\mathbf{R}\mathbf{b}_1 \times \boldsymbol{\rho}_1)^T \\ \boldsymbol{\rho}_2^T & (\mathbf{R}\mathbf{b}_2 \times \boldsymbol{\rho}_2)^T \end{bmatrix}. \tag{27}$$

Moreover, the singularity analysis of this mechanism can be obtained by deriving the degeneration configurations of the Jacobian matrix \mathbf{J} . In addition, the stability of the proposed mechanism is able to be evaluated by the Hessian matrix based on the Jacobian matrix \mathbf{J} .

5.3. Acceleration Analysis

The second derivation of both sides of Equation (22) can obtain the velocity change rate of the cable length, i.e., the acceleration of the cable length:

$$l_i \frac{d^2l_i}{dt^2} = (\boldsymbol{\omega} \times \mathbf{R}\mathbf{b}_i + \mathbf{v})^T (\boldsymbol{\omega} \times \mathbf{R}\mathbf{b}_i + \mathbf{v}) + (\mathbf{R}\mathbf{b}_i + \mathbf{p} - \mathbf{a}_i^O)^T \left(\frac{d^2\mathbf{R}}{dt^2} \mathbf{b}_i + \dot{\mathbf{v}} \right), \tag{28}$$

where $\frac{d^2\mathbf{R}}{dt^2}$ is expressed as

$$\frac{d^2\mathbf{R}}{dt^2} = \frac{d\mathbf{S}(\boldsymbol{\omega})\mathbf{R}}{dt} + \mathbf{S}(\boldsymbol{\omega})^2\mathbf{R}. \tag{29}$$

Substituting Equation (29) into Equation (28), one can obtain

$$\ddot{i}_i = \frac{(\boldsymbol{\omega} \times \mathbf{R}\mathbf{b}_i + \mathbf{v})^T (\boldsymbol{\omega} \times \mathbf{R}\mathbf{b}_i + \mathbf{v})}{l_i} + \rho_i^T \left[\left(\frac{d\mathbf{S}(\boldsymbol{\omega})}{dt} \mathbf{R} + \mathbf{S}(\boldsymbol{\omega})^2 \mathbf{R} \right) \mathbf{b}_i + \dot{\mathbf{v}} \right], \quad (i = 1, 2). \tag{30}$$

5.4. Kinematics Simulation

The structural parameters of the hybrid cable-driven parallel mechanism, i.e., the length of the rod between Slider 1 and Slider 2 as $m = 0.25$ m and the length of the rod between Slider 2 and moving platform $n = 0.25$ m, was adopted as a case study in the simulation. The lengths of the moving platform and the static platform are $2e$ and $2f$ ($e = 0.5$ m, $f = 0.05$ m).

The motion function of the moving platform is given as

$$\begin{aligned} x &= l_{op} \cos \alpha, \\ y &= -l_{op} \sin \alpha. \end{aligned} \tag{31}$$

The relationship between the displacement of Slider 2 and angle α is shown in Figure 9.

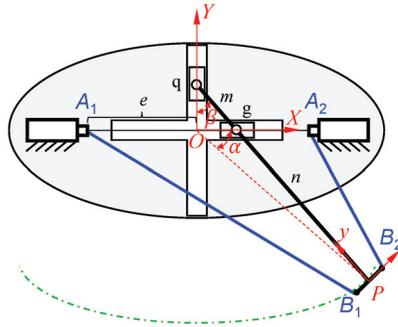


Figure 9. Diagram of relationship between displacement of Slider 2 and angle α .

When Slider 2 is the active joint, let the distance of motion be s . Suppose the initial value of l_{og} is 0.15 m. The expressions of other parameters are

$$\begin{aligned} l_{og} &= 0.15 - s, \\ l_{oq}^2 &= m^2 - l_{og}^2, \\ \sin \beta &= \frac{l_{og}}{m}, \\ l_{op}^2 &= l_{oq}^2 + (m + n)^2 - 2l_{oq}(m + n)\cos \beta, \\ \cos \alpha &= \frac{l_{og}^2 + l_{op}^2 - n^2}{2l_{og}l_{op}}. \end{aligned} \tag{32}$$

After several mathematical manipulations, the following expression is obtained on the basis of Equation (32):

$$\alpha = \arccos \left[2m \left(\frac{m - l_{oq} \cos \beta}{l_{og} l_{op}} \right) \right], \tag{33}$$

where β , l_{og} , l_{oq} and l_{op} are expressed as functions of displacement s .

To verify the present kinematic analysis, a simulation was performed in ADAMS and MATLAB, as depicted in Figure 10. The simulation model of the hybrid cable-driven cross-slide mechanism was created in ADAMS using the aforementioned values of each parameter. The lengths of each cable can be attained during a cycle of an elliptical motion within a period of 20 s.

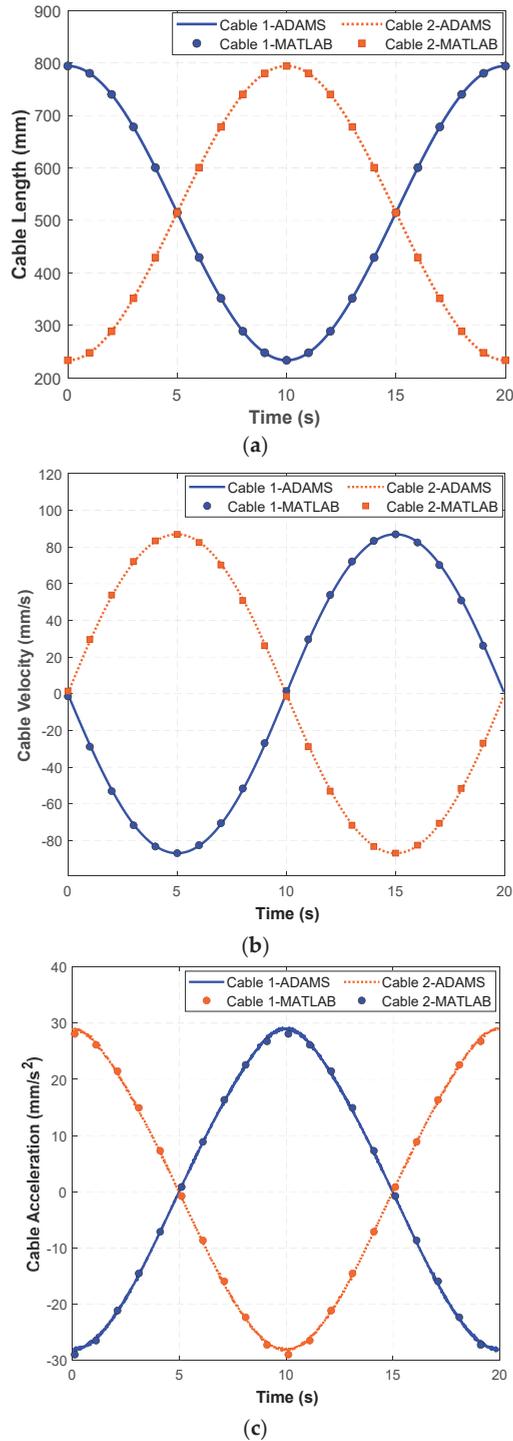


Figure 10. Kinematics simulation in MATLAB and ADAMS. (a) Length variation of Cable 1 and Cable 2 from ADAMS and MATLAB. (b) Velocity variation of Cable 1 and Cable 2 from ADAMS and MATLAB. (c) Acceleration variation of Cable 1 and Cable 2 from ADAMS and MATLAB.

From Figure 10, it can be observed that the fluctuation amplitude and numerical value of the curves in MATLAB and ADAMS are in good agreement with each other. The efficiency of the kinematic analysis of the hybrid mechanism is verified accordingly. The cross points of acceleration are because the initial acceleration of the cable was set to 0. For the elliptical trajectory, the kinematic characteristic curves of two cables are smooth and continuous. The sinusoidal-like function for cable length control was identified from both the theoretical and simulation results. Thus, the proposed cable-driven parallel mechanism is considered to have good kinematic characteristics for implementing a large rotation angle over the border of the workspace. In addition, the limitation of this method was obtained by comparing the V-type cable mechanism and the hybrid mechanism. It is worth noting that two cables of constant length should be kept in tension due to the unidirectional force feature.

6. Conclusions

This paper deals with a spatial cable-driven parallel mechanism in a V-shaped cable arrangement, which is simplified as a planar hybrid cable-driven parallel mechanism with two identical active cable chains and a passive cross-slide mechanism. The DoFs of the planar hybrid cable-driven mechanism were analyzed using screw theory. This paper focuses on an alternative approach from V-type cable sub-chains to the equivalent rigid closed-loop chains. The presented method solves easily the DoFs of the rigid parallel mechanisms using screw theory. It is neither limited to the selection of the coordinate systems nor needs to solve the specific parameters of screws, which leads to an effective approach for solving the DoFs of the cable mechanisms with a V-shaped configuration. The kinematic characteristic curves of two cables for the planar hybrid cable-driven mechanism are smooth and continuous under the elliptical trajectory, which show good kinematic characteristics in terms of velocity and acceleration smoothness. Future work will focus on the singularity analysis and stability analysis of the proposed V-shaped cable-driven parallel mechanism based on the Jacobian matrix.

Author Contributions: Conceptualization, L.W. and L.T.; methodology, D.Z.; software, X.L.; validation, D.Z., M.W. and L.W.; formal analysis, D.Z.; investigation, X.L.; resources, L.T.; data curation, L.T.; writing—original draft preparation, D.Z.; writing—review and editing, L.W.; visualization, X.L.; supervision, L.T.; project administration, L.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the research project from Xiangtan Institute of Technology (Project No. XTLG2022B017) and the research development project from Development and Reform Commission of Hunan Province (Project No. 2112-430000-04-03-549452).

Data Availability Statement: Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Z.; Shao, Z.; You, Z.; Tang, X.; Zi, B.; Yang, G.; Gosselin, C.; Caro, S. State-of-the-art on theories and applications of cable-driven parallel robots. *Front. Mech. Eng.* **2022**, *17*, 37. [CrossRef]
2. Qian, S.; Zi, B.; Xu, Q. A review on cable-driven parallel robots. *Chin. J. Mech. Eng.* **2018**, *31*, 66.
3. Tang, X. An overview of the development for cable-driven parallel manipulator. *Adv. Mech. Eng.* **2014**, *6*, 823028. [CrossRef]
4. Tang, L.; Gouttefarde, M.; Sun, H.; Yin, L.; Zhou, C. Dynamic modelling and vibration suppression of a single-link flexible manipulator with two cables. *Mech. Mach. Theory* **2021**, *162*, 104347. [CrossRef]
5. Rasheed, T.; Long, P.; Caro, S. Wrench-feasible workspace of mobile cable-driven parallel robots. *J. Mech. Robot.* **2020**, *12*, 031009. [CrossRef]
6. Castelli, G.; Ottaviano, E.; Rea, P. A cartesian cable-suspended robot for improving end-users' mobility in an urban environment. *Robot. Comput.-Integr. Manuf.* **2014**, *30*, 335–343. [CrossRef]
7. Tang, L.; Shi, P.; Wu, L.; Wu, X.; Tang, X. Singularity analysis on a special class of cable-suspended parallel mechanisms with pairwise cable arrangement and actuation redundancy. *J. Mech. Des.* **2020**, *142*, 024501. [CrossRef]
8. Barbazza, L.; Oscari, F.; Minto, S.; Rosati, G. Trajectory planning of a suspended cable driven parallel robot with reconfigurable end effector. *Robot. Comput.-Integr. Manuf.* **2017**, *48*, 1–11. [CrossRef]

9. Tang, X.; Zhu, W.; Sun, C.; Yao, R. Similarity model of feed support system for FAST. *Exp. Astron.* **2011**, *29*, 177–187. [CrossRef]
10. Yin, J.; Jiang, P.; Yao, R. An approximately analytical solution method for the cable-driven parallel robot in FAST. *Res. Astron. Astrophys* **2021**, *21*, 46–60. [CrossRef]
11. Yao, R.; Jiang, P.; Sun, J.-H.; Yu, D.-J.; Sun, C. A motion planning algorithm for the feed support system of FAST. *Res. Astron. Astrophys* **2020**, *20*, 68. [CrossRef]
12. Khakpour, H.; Birglen, L.; Tahan, S. Synthesis of differentially driven planar cable parallel manipulators. *IEEE Trans. Robot.* **2014**, *30*, 619–630. [CrossRef]
13. Khodadadi, N.; Hosseini, M.I.; Khalilpour, S.A.; Taghirad, H.D.; Cardou, P. Multiobjective optimization of a cable-driven robot with parallelogram links. In *Cable-Driven Parallel Robots. CableCon 2021*; Gouttefarde, M., Bruckmann, T., Pott, A., Eds.; Springer International Publishing: Cham, Switzerland, 2021; Volume 104, pp. 170–181.
14. Nuchkrua, T.; Chen, S. Precision contouring control of five degree of freedom robot manipulators with uncertainty. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881416682703. [CrossRef]
15. Dimentberg, F. *The Screw Calculus and Its Applications to Mechanics*; Clearinghouse for Federal Scientific & Technical Information: Online, 1968.
16. Hunt, K. *Kinematic Geometry of Mechanisms*; Clarendon Press: Oxford, UK, 1978.
17. Sugimoto, K.; Duffy, J. Application of Linear Algebra to Screw Systems. *Mech. Mach. Theory* **1982**, *17*, 73–83. [CrossRef]
18. Mohamed, M.G.; Duffy, J. A direct determination of the instantaneous kinematics of fully parallel robot manipulators. *J. Mech. Des.* **1985**, *107*, 226–229. [CrossRef]
19. Huang, Z. Modelling formulation of 6-dof Multi-loop Parallel Manipulators, Part 1-Kinematic Influence Matrix. In Proceedings of the 4th IFToMM Conference on Mechanisms and CAD, Bucharest, Romania, 27–30 September 1985; Volume 2, pp. 163–170.
20. Huang, Z.; Liu, J.; Zeng, D. A general methodology for mobility analysis of mechanisms based on constraint screw theory. *Sci. China Ser. E-Technol. Sci.* **2009**, *52*, 1337–1347. [CrossRef]
21. Huang, Z.; Liu, J.; Li, Y. *Mobility of Mechanisms*; Science Press: Beijing, China, 2011.
22. Yang, T.; Shen, H.; Liu, A.; Dai, J. Review of the Formulas for Degrees of Freedom in the Past Ten Years. *J. Mech. Eng.* **2015**, *51*, 69–80. [CrossRef]
23. Hervé, J.M. The mathematical group structure of the set of displacements. *Mech. Mach. Theory* **1994**, *29*, 73–81. [CrossRef]
24. Huang, Z.; Li, Q. General methodology for type synthesis of lower-mobility symmetrical parallel manipulators and several novel manipulators. *Int. J. Robot. Res.* **2002**, *21*, 131–145. [CrossRef]
25. Yang, T.; Liu, A.; Luo, Y.; Hang, L.; Shen, H.; Shi, Z. Basic principles, main characteristics and development tendency of methods for robot mechanism structure synthesis. *J. Mech. Eng.* **2010**, *46*, 1–11. [CrossRef]
26. Hu, B. Kinematically identical manipulators for the Exechon parallel manipulator and their comparison study. *Mech. Mach. Theory* **2016**, *103*, 117–137. [CrossRef]
27. Hu, B.; Huang, Z. Kinematically identical manipulators of a 2-RPU+UPR parallel manipulator and their kinematics analysis. In Proceedings of the 14th IFToMM World Congress, Taipei, Taiwan, China, 25–30 October 2015; pp. 310–319.
28. Hu, B. Theoretical Study of Limited-Dof Parallel Manipulators and Serial-Parallel Manipulators Based on Constrained Force/Torque. Ph.D. Thesis, Yanshan University, Qinhuangdao, China, 2010.
29. Li, W.; Xiao, Y.; Bi, S.; Du, G. Automatic elliptical trajectory planning algorithm for pick and place operation. In Proceedings of the 2013 International Conference on Advanced Mechatronics Systems, Luoyang, China, 19–22 February 2013.
30. Fallah-Tafti, F.; Pfeifer, C.M.; Buster, T.W.; Burnfield, J.M. Effect of motor-assisted elliptical training speed and body weight support on center of pressure movement variability. *Gait Posture* **2020**, *81*, 138–143. [CrossRef] [PubMed]
31. Tang, Z.; Dai, J. Bifurcated configurations and their variations of an 8-bar linkage derived from an 8-kaleidocycle. *Mech. Mach. Theory* **2018**, *121*, 745–754. [CrossRef]
32. Wang, R.; Song, Y.; Dai, J. Reconfigurability of the origami-inspired integrated 8R kinematotropic metamorphic mechanism and its evolved 6R and 4R mechanisms. *Mech. Mach. Theory* **2021**, *161*, 104245. [CrossRef]
33. Shi, C.; Guo, H.; Zhang, S.; Liu, R.; Deng, Z. Configuration synthesis of linear foldable over-constrained deployable unit based on screw theory. *Mech. Mach. Theory* **2021**, *156*, 104163. [CrossRef]
34. Khakpour, H.; Birglen, L.; Tahan, S.-A. Analysis and optimization of a new differentially driven cable parallel robot. *J. Mech. Robot.* **2015**, *7*, 034503. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Parallel Pointing Systems Suitable for Robotic Total Stations: Selection, Dimensional Synthesis, and Accuracy Analysis

Henrique Simas¹, Raffaele Di Gregorio^{2,*}, Roberto Simoni³ and Marco Gatti²

¹ Raul Guenther Lab. of Applied Robotics, Department of Mechanical Engineering, Federal University of Santa Catarina, Florianópolis 88040-900, SC, Brazil; henrique.simas@ufsc.br

² Laboratory of Mechatronics and Virtual Prototyping (LaMaViP), Department of Engineering, University of Ferrara, Via Saragat, 1, 44100 Ferrara, Italy; marco.gatti@unife.it

³ Department of Mobility Engineering, Federal University of Santa Catarina, Joinville 89219-600, SC, Brazil; roberto.simoni@ufsc.br

* Correspondence: raffaele.digregorio@unife.it; Tel.: +39-0532-974828

Abstract: Robotic Total Stations (RTS) are fully automated theodolites with electronic distance measurement (EDM) that include a number of additional tools (e.g., camera, laser rangefinder, onboard computer, and tracking software, etc.) enabling them to work autonomously. The added tools make RTSs able to track mobile targets on civil structures thus opening to the use of RTSs in structural monitoring. Unfortunately, the available RTSs are able to track a target up to a motion rate of 3 Hz. Reducing mobile masses is a viable design strategy for extending this frequency border. Such a strategy is pursued in this study by proposing the use of parallel pointing systems (PPS) as basic mechanical architectures for RTSs. The literature on PPSs is reviewed and the applicable PPS architectures are selected. Successively, the selected architectures are sized according to RTSs' functional requirements, and the positioning precision of the sized mechanisms is evaluated. The result of this study is that there are three PPS architectures suitable for RTSs, whose detailed comparison is also presented.

Keywords: parallel mechanism; pointing system; dimensional synthesis; accuracy analysis; robotic total station

Citation: Simas, H.; Di Gregorio, R.; Simoni, R.; Gatti, M. Parallel Pointing Systems Suitable for Robotic Total Stations: Selection, Dimensional Synthesis, and Accuracy Analysis. *Machines* **2024**, *12*, 54. <https://doi.org/10.3390/machines12010054>

Academic Editor: Dan Zhang

Received: 2 December 2023

Revised: 2 January 2024

Accepted: 6 January 2024

Published: 12 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A theodolite [1] is an instrument that measures two angles, named pan and tilt angles, which locate the orientation of a line pointing at a target. It consists of a telescope connected to a frame (base) by the means of a universal (U) joint, which allows its rotation around a vertical axis (pan rotation) and a horizontal axis (tilt rotation), and a set of graduated circles that make the two angles accurately measurable. The base is levelled before the angle measurement and provides the reference horizontal plane. The telescope is equipped with a crosshair that enables the operator to point at a particular target; the axis of the telescope is the line whose pan and tilt angles are measured. Land surveying, construction, and mapping are the applicative fields of theodolites, where they establish points, lay out boundaries, and determine the heights and positions of objects or other features on the Earth's surface.

Theodolites evolved into total stations, which also feature a laser rangefinder and an onboard computer that calculates the position of the target point with respect to the instrument. The added tools allow the operator (usually a surveyor) to point at a target, record the measured angles and distance, and repeat the two previous operations for many target points without moving the instrument to create a detailed 3D map of the site. Successively, total stations evolved into robotic total stations (RTS) that include a number of additional tools (e.g., actuators for controlling their motion, camera, tracking software, etc.), enabling them to work autonomously by following a loaded measurement program or remote controller.

The added tools [2–6] make RTSs also able to track mobile targets on civil structures, thus making possible their use in structural monitoring [7–14]. Indeed, they have been employed either together with other instruments, as GPS [7,8], or alone to measure the natural frequencies of road bridges [9,10], railway bridges [11,12], and pedestrian bridges [13,14] excited by the transit of people or vehicles, that is, when the oscillation amplitude has the order of centimeters and the oscillation frequency is lower than 1 Hz. In civil structures, the frequency border of 1 Hz is sufficient for many applicative cases; nevertheless, extending this border up to 5 Hz, or better, to 10 Hz, would greatly extend this field of application.

Despite the fact that recently presented RTSs (see Table 1) are equipped with 360° reflector prisms [15] and have doubled their recording rate [16], extending this frequency border over 3 Hz still is a goal to reach [17]. Since RTSs' angle/distance measurements come after the telescope collimation toward the target point and the collimation is a mechanical operation, extending this frequency limitation is intimately related to increasing the dynamic performance of the mechanism that moves the telescope.

Increasing the dynamic performances of any mechanical system unavoidably passes through a reduction in the mobile masses. Table 1 shows that the total mass of commercial RTSs ranges roughly from 5 kg to 9.5 kg and that most of their mass is loaded on mobile parts, even though the telescope, which is the component to move, usually weighs less than 0.5 kg. In short, until now, the automation of total stations has been implemented by simply adding actuators in the two revolute (R) pairs of the U-joint, which connects the telescope to the base, without redesigning the mechanical system for reducing all the mobile masses.

Moving the electrically supplied parts (e.g., actuators, Wi-Fi antenna, and display, etc.) together with their batteries onto the base is the design strategy to implement for greatly reducing the mobile masses. The vast majority of these components (e.g., Wi-Fi antenna, onboard computer, and display) are moveable onto the base without changing the mechanism that moves the telescope; whereas, moving the actuators onto the base needs to change such a mechanism from a serial architecture to a parallel architecture¹. Replacing a serial architecture with a parallel one also brings other advantages. Indeed, parallel architectures, in general, are stiffer and more precise than their serial counterpart. Parallel architectures that orientate a line with respect to their base are named parallel pointing systems (PPS).

This paper reviews the PPS architectures proposed in the literature and selects those that are more suitable for RTSs with reference to RTSs' functional requirements. Successively, it addresses the dimensional synthesis and accuracy analysis of the selected architectures. The result of this study is that there are three PPS types that are more suitable for RTSs. The pros and cons of these three PPS architectures are also discussed.

The paper is organized as follows. Section 2 reviews the literature on PPSs, defines the functional requirements for a PPS to be used in an RTS, and selects the most suitable ones. Section 3 addresses the position analysis and dimensional synthesis of the PPSs selected in the previous section, using the RTS's functional requirements identified in the same section. Finally, Section 4 discusses the results, also evaluating their accuracy, and Section 5 draws conclusions.

¹ Serial architectures are open kinematic chains (in this case, the U joint) that connect the base to the end effector (in this case, the link carrying the telescope). Differently, parallel architectures feature more-than-one kinematic chains (limbs) that simultaneously connect the end effector to the base.

Table 1. Selection and data sheets of commercial robotic total stations (RTS).

								
<i>Manufacturer</i>	Leica	Sokkia	Trimble	Topcon	GeoMax	Carlson	Hilti	Stonex
<i>Model</i>	Leica Nova TS60	iX-1201	S9 HP	MS05AXII	Zoom95	STReAM360	POS180	R180
<i>Angular accuracy</i>	0.5" (0.15 mgon)	1" (0.3 mgon)	0.5" (0.15 mgon)	0.5" (0.15 mgon)	1" (0.3 mgon)	1" (0.3 mgon)	3"	0.5"
<i>Distance accuracy</i>	0.6 mm + 1 ppm	1 mm + 2 ppm	0.8 mm + 1 ppm	0.8 mm + 1 ppm	1 mm + 1.5 ppm	1 mm + 1.5 ppm	--	1 mm + 1 ppm
<i>Maximum pan/rotation speed</i>	180° / s	150° / s	115° / s	85° / s	90° / s	--	90° / s	180° / s
<i>Motorization</i>	Piezo Drives	Direct Drives	MagDrive servo	DC motor drives	Hybrid Drives	Hybrid Drives	--	Tdrive
<i>Weight</i>	7.7 kg	5.8 kg	5.85 kg	6.8 kg	5.3 kg	5.3 kg	--	9.3 kg
<i>Other data</i>	Camera's frame rate: Up to 20 fps	Max auto tracking speed: 20° / s	Tracking rate: 10 Hz	--	--	Tracking speed: 90 km/h at 100 m	--	--

2. Selection of PPS Architectures for RTS

The literature on the type synthesis² of parallel architectures is practically complete. Indeed, type synthesis methods based on graph theory [18–23], screw theory [24–30] group theory [31–38], and evolutionary morphology [39], among others, have been proposed together with a long list of parallel-architecture types that satisfy many motion requirements. In this literature, the most extended lists of PPS architectures are reported in [37], [38] (Chap. 12), [39] (Vol. 4). Over these lists, papers (e.g., [40,41]) and patents (e.g., [42]) on specific PPS architectures have been presented, too.

PPSs are two degrees-of-freedom (DOF) closed-chain mechanisms. All the PPS architectures proposed in the literature exhibit a U-joint that directly connects the end effector to the base. Such a U-joint is either fully passive (i.e., the two R pairs of the U-joint are both non-actuated) or partially active (i.e., one out of the two R pairs of the U-joint is actuated). It is fully passive in PPS architectures (e.g., [40,41]) that have two more limbs connecting the end effector to the base: these two additional limbs contain the two actuators (one per limb). These two-looped architectures pay their higher stiffness with a workspace reduction; they are suitable for applications (e.g., orientating satellite antennas or solar panels, etc.) where the end effector is heavy and the requested orientation workspace is limited.

Differently, the U joint is partially active in PPS architectures (e.g., [42]) that contain only one additional limb, which carries the second actuator. Since the telescope of an RTS is not a heavy object and it needs an ample orientation workspace, these single-looped architectures with a partially active U-joint must be chosen for RTSs. In particular, in the U joint, the R pair adjacent to the base, that is, the one with a vertical axis (see the images in Table 1), which makes the telescope perform the pan rotation, must be actuated. Moreover, in the additional limb, the first joint (i.e., the one adjacent to the base) must be actuated so that the remaining part of the limb can play only the role of a mechanical transmission that controls the motion of the non-actuated R pair of the U joint (i.e., the one that makes the telescope perform the tilt rotation). Figure 1 shows a generic PPS architecture that satisfies these requirements.

According to the Grübler–Kutzbach mobility criterion [43] and Euler’s formula [44], a two-DOF single-looped spatial mechanism, like the PPS architectures selected for RTSs, must satisfy the following condition to be non-overconstrained:

$$\left. \begin{aligned} 2 &= 6(m - 1) - \sum_{i=1,5} (6 - i)c_i \\ m &= \sum_{i=1,5} c_i \end{aligned} \right\} \Rightarrow \sum_{i=1,5} i c_i = 8 \quad (1)$$

where m is the number of links and c_i is the number of joints with i DOFs. Since the U joint yields a term equal to 2 in the summation on the left-hand side of Equation (1), the following relationships hold for the additional limb of a non-overconstrained PPS:

$$\left\{ \begin{aligned} \sum_{i=1,5} i c_{i,limb} &= 6 \\ \sum_{i=1,5} c_{i,limb} &= \sum_{i=1,5} c_i - 1 \\ m_{limb} &= m - 2 = \sum_{i=1,5} c_{i,limb} - 1 \end{aligned} \right. \quad (2)$$

where m_{limb} is the number of links of the additional limb and $c_{i,limb}$ is the number of joints with i DOFs of the additional limb.

Choosing non-overconstrained PPS architectures is preferable since they do not require the imposition of tight tolerances on the unavoidable geometric errors that come out during manufacturing. Nevertheless, reducing the number of links and joints under the values provided by Equation (2) is also interesting when the introduced overconstraint is as “easy” to obtain as, for instance, it is for some planar or spherical kinematic chains. Moreover,

² Type synthesis is the identification of mechanism topologies that match some motion requirements.

PPS architectures that do not contain passive (i.e., non-actuated) prismatic (P) pairs must be preferred.

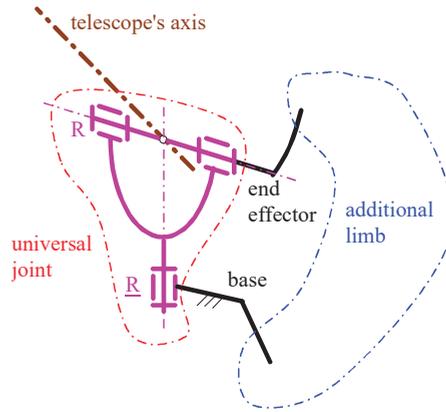


Figure 1. Generic PPS architecture suitable for an RTS.

The above-defined requirements, together with the fact that the pan rotation must be a complete rotation in an RTS (i.e., the additional limb must not limit the pan rotation), lead one to select the PPS architectures shown in Figures 2–4 with five, six, and seven links, respectively. Such architectures are deduced from the three main families of single-looped PPS architectures without passive P pairs reported in [37], [38] (Chap. 12), [39] (Vol. 4) by imposing that the additional limb must not limit the pan rotation and that its number of links and joints must be as small as possible.

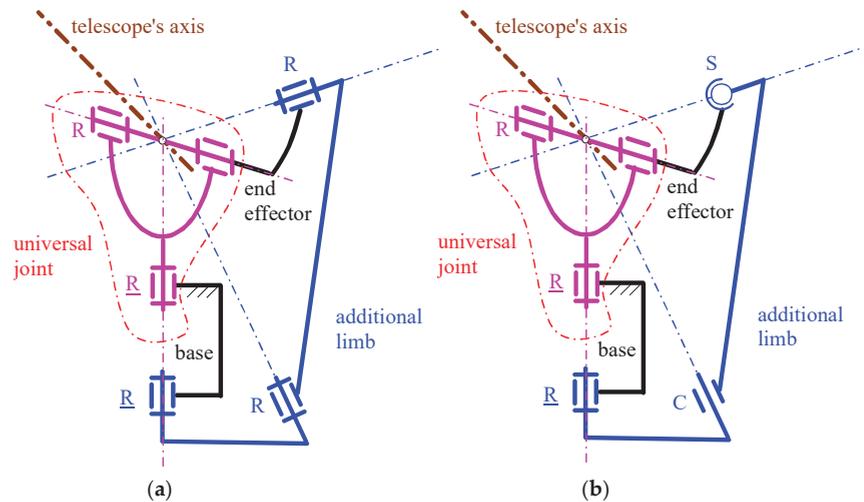


Figure 2. Spherical five-bar PPSs of type (C and S stand for cylindrical and spherical pair, respectively; the underscore indicates the actuated pairs): (a) \underline{RR} -RRR (overconstrained solution), and (b) \underline{RR} -RCs (non-overconstrained solution).

The PPS of Figure 2a, hereafter named the “spherical PPS”, is obtained from the general scheme of Figure 1 by choosing an additional limb of the \underline{RRR} type (the underscore indicates an actuated joint). The peculiarities of this \underline{RRR} limb are the following: (a.1) the axes of the three R pairs are so oriented that they all pass through the center of the

partially active U joint (i.e., the purple \underline{RR} limb in the figure), and (a.2) the axis of the actuated R-pair, adjacent to the base, coincides with the pan-rotation axis. The resulting single-looped architecture is a particular spherical five-bar linkage of the \underline{RR} - \underline{RRR} type that has the spherical motion center coincident with the U-joint center. Condition (a.2) is the one that allows the telescope to perform a complete pan rotation. This spherical five-bar linkage is overconstrained (i.e., it does not satisfy Equation (1)); nevertheless, its overconstraint can be easily removed by replacing, in the \underline{RRR} limb, the intermediate R-pair with a cylindrical (C) pair and the R-pair adjacent to the end effector with a spherical (S) pair, as shown in Figure 2b. Doing so, the additional limb becomes of the \underline{RCS} type.

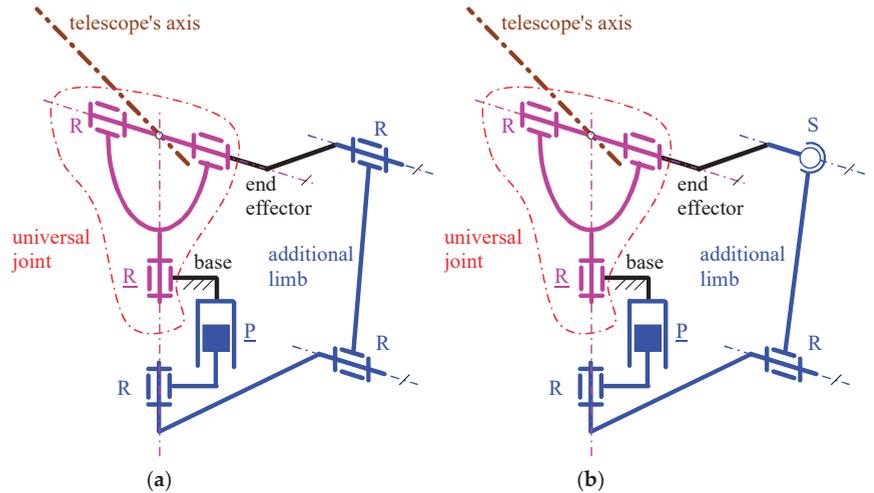


Figure 3. Six-bar PPSs of type (P and S stand for prismatic and spherical pair, respectively; the underscore indicates the actuated pairs): (a) \underline{RR} - \underline{PRRR} (overconstrained solution), and (b) \underline{RR} - \underline{PRRS} (non-overconstrained solution).

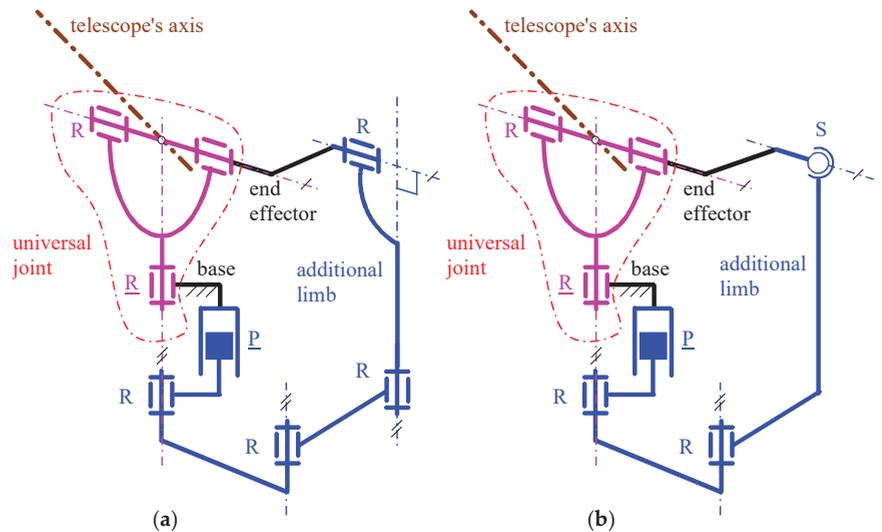


Figure 4. Overconstrained 7-bar PPS of type \underline{RR} - \underline{PRRRR} (a) and its equivalent non-overconstrained 6-bar PPS of type \underline{RR} - \underline{PRRS} (b).

The PPS of Figure 3a, hereafter named the “1st decoupled PPS”, is obtained from the general scheme of Figure 1 by choosing an additional limb of the PRRR type. The peculiarities of this PRRR limb are the following: (b.1) the axis of the first R-pair, adjacent to the actuated P pair, coincides with the pan-rotation axis, (b.2) the sliding direction of the actuated P pair is parallel to the pan-rotation axis, and (b.3) the axes of the second and the third R pairs are both parallel to the tilt-rotation axis. The resulting single-looped architecture is a particular six-bar linkage of the RR-PRRR type that, when the actuated P pair (the actuated R pair) is locked, keeps the tilt (the pan) angle fixed (i.e., the pan and tilt rotations are independently moveable). This six-bar linkage is overconstrained (i.e., it does not satisfy Equation (1)); nevertheless, its overconstraint can be easily removed by replacing, in the PRRR limb, the R-pair adjacent to the end effector with an S pair, as shown in Figure 3b. Doing so, the additional limb becomes of the PRRS type.

The PPS of Figure 4a, hereafter named the “2nd decoupled PPS”, is obtained from the general scheme of Figure 1 by choosing an additional limb of PRRRR type. The peculiarities of this PRRRR limb are the following: (c.1) the axis of the first R-pair, adjacent to the actuated P pair, coincides with the pan-rotation axis, (c.2) the sliding direction of the P pair and the axes of the second and third R pairs are all parallel to the pan-rotation axis, and (c.3) the axis of the fourth R pair, adjacent to the end effector, is parallel to the tilt-rotation axis. The resulting single-looped architecture is a particular seven-bar linkage of the RR-PRRRR type that, when the actuated P pair (the actuated R pair) is locked, keeps the tilt (the pan) angle fixed (i.e., the pan and tilt rotations are independently moveable). This seven-bar linkage is overconstrained (i.e., it does not satisfy Equation (1)); nevertheless, its overconstraint can be easily removed by replacing, in the PRRRR limb, the last two R-pairs with an S pair, as shown in Figure 4b. Doing so, the additional limb becomes of the PRRS type and the resulting linkage has only six links.

After having selected the PPS architectures suitable for RTSs, the dimensional synthesis of these architectures must be completed by imposing that:

- (i) the tilt angle can cover a variation range (at least 90 degrees) suitable to make the telescope axis assume any direction inside the upper hemisphere;
- (ii) the additional limb guarantees a sufficiently good transmission angle at any PPS configuration.

The next section, firstly, will deduce the kinematic relationships necessary to control the motion of the selected PPS architectures; then, it will exploit the deduced relationships to complete the dimensional synthesis of the selected PPS architectures by imposing the above-defined additional design requirements.

3. Kinematic Analysis and Dimensional Synthesis

In order to evaluate and compare the selected PPS architectures, the analytic relationships necessary to solve their kinematic analysis problems, which are involved in their motion control, must be deduced. Moreover, their dimensional synthesis must be completed for the determination of the geometric constants' values that make the PPS satisfy the above-defined design requirements (i) and (ii).

In the following part of this section, these analytic/numeric computations are implemented for each of the three PPS architectures identified in the previous section. In doing so, a variation range of 120° is chosen for the tilt angle when imposing design requirements (i) and (ii). Hereafter, ψ , ζ , and θ_1 denote, respectively, the pan angle, the tilt angle, and the actuated-joint variable of the partially active U joint that directly connects the end effector to the base, which is the rotation angle of the R pair adjacent to the base (see Figure 1). Moreover, without losing generality, no phase difference is assumed between ψ and θ_1 , that is, for all the selected PPSs (see Figure 1), the following relationship holds:

$$\psi = \theta_1 \quad (3)$$

Eventually, the kinematic analyses of the overconstrained mechanisms (i.e., Figures 2a, 3a and 4a) and their non-overconstrained counterparts (i.e., Figures 2b, 3b and 4b) coincide

with one another if no geometric error occurs. As a consequence, since the evaluation of geometric error effects is out of the scope of this work, the kinematic analyses that follow will refer to the nominal geometry and, for the sake of simplicity, all the notations will be defined by using the overconstrained or non-overconstrained architectures according to convenience.

3.1. Spherical PPS

Figure 5a illustrates the adopted notations for the kinematic analysis and dimensional synthesis of the PPSs shown in Figure 2a. With reference to Figure 5a, $Ox_0y_0z_0$ ($Ox_1y_1z_1$) is a Cartesian reference system fixed to the base (to the cross link of the U joint (see Figure 2a) with the origin, O, coincident with the center of the U joint, the z_0 -coordinate axis (z_1 -coordinate axis) coincident with the pan rotation axis, and the y_0 -coordinate axis as a phase reference for measuring the pan rotation (the y_1 -coordinate axis is coincident with the tilt rotation axis). The x_0y_0 -coordinate plane, coincident with the x_1y_1 -coordinate plane, is the horizontal plane of the theodolite, which is also the phase reference for measuring the tilt angle, ζ , whereas the two mutually orthogonal axes of the two R pairs constituting the U joint lie on the y_1z_1 -coordinate plane. Angle α_i , for $i = 0, \dots, 4$, is the constant angle between the axes of the R pairs at the endings of link i . In particular, links 0 and 3 are the base and the cross link of the U joint, respectively; as a consequence, α_0 and α_3 are equal to 0° and 90° , respectively, whereas α_1 , α_2 , and α_4 must be sized by imposing design requirements (i) and (ii). Eventually, angle θ_2 is the joint variable of the first R pair, adjacent to the base, of the additional limb of the RRR type, and angle φ is equal to $(\theta_2 - \theta_1)$.

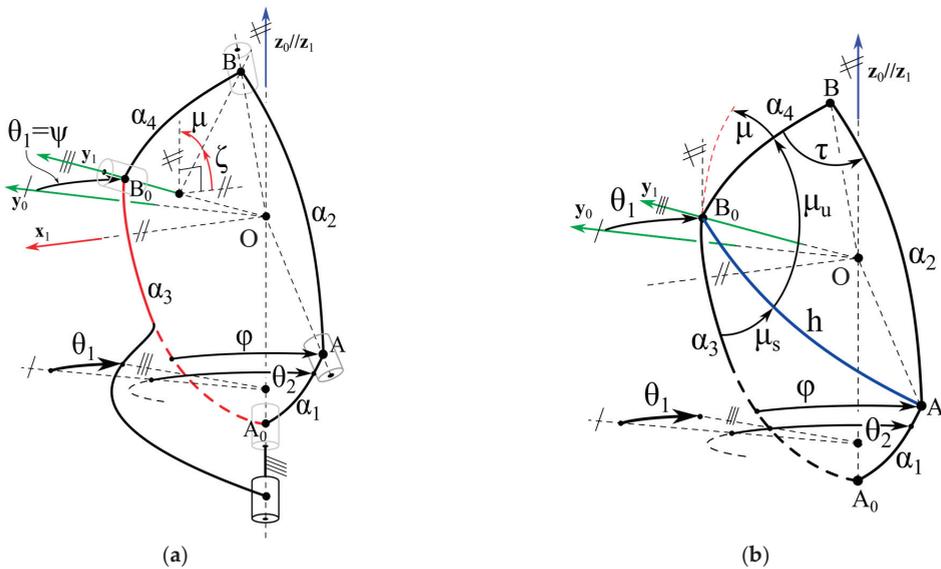


Figure 5. Kinematic scheme of the spherical PPS: (a) notations, and (b) spherical 4-bar obtained by locking the pan rotation, ψ .

When the pan rotation, ψ ($=\theta_1$ (Equation (3))), is locked, the spherical five-bar linkage becomes the spherical four-bar linkage shown in Figure 5b, which has link 1 as an input link with φ ($=(\theta_2 - \theta_1)$) as the input variable and link 4 as the output link with the tilt angle, ζ , as the output variable. Such a spherical four-bar linkage is the same for any value of ψ . As a consequence, the kinematic model of this spherical four-bar linkage, together with Equation (3), provides the kinematic model of this spherical PPS.

With reference to Figure 5b, τ is the transmission angle [45] of this spherical four-bar linkage and h is the convex angle between segments OB_0 and OA . Moreover, the following relationships hold:

$$\zeta + \mu = \frac{\pi}{2}; \mu_s + \mu_u + \mu = \pi; \mu_s + \mu_u = \frac{\pi}{2} + \zeta. \tag{4}$$

The cosine law for spherical triangles [45], when applied to the spherical triangles ΔA_0B_0A and ΔAB_0B of Figure 5b, which share side h , yields:

$$\cos \alpha_2 = \cos h \cos \alpha_4 + \sin h \sin \alpha_4 \cos \mu_u \tag{5a}$$

$$\cos h = \cos \alpha_1 \cos \alpha_3 + \sin \alpha_1 \sin \alpha_3 \cos \varphi \tag{5b}$$

$$\cos \alpha_1 = \cos h \cos \alpha_3 + \sin h \sin \alpha_3 \cos \mu_s \tag{5c}$$

whereas the application of the sine law for spherical triangles [45] to the spherical triangle ΔA_0B_0A gives:

$$\frac{\sin h}{\sin \varphi} = \frac{\sin \alpha_1}{\sin \mu_s} \Rightarrow \sin h \sin \mu_s = \sin \alpha_1 \sin \varphi \tag{6}$$

By taking into account the value of $\alpha_3 (= \pi/2)$, the introduction of Equation (5b) into Equation (5a) transforms it as follows:

$$\cos \alpha_2 = \sin \alpha_1 \cos \varphi \cos \alpha_4 + \sin h \sin \alpha_4 \cos \mu_u \tag{7}$$

which, after having introduced the expressions of μ_u coming from Equation (4) and $\cos \mu_s$ and $\sin \mu_s$ coming from Equations (5c) and (6), respectively, yields

$$\sin \alpha_1 \sin \alpha_4 \sin \varphi \cos \zeta - \cos \alpha_1 \sin \alpha_4 \sin \zeta + \sin \alpha_1 \cos \alpha_4 \cos \varphi - \cos \alpha_2 = 0 \tag{8}$$

Equation (8) is the closure equation of the spherical four-bar linkage of Figure 5b. Over ζ (i.e., the output variable) and $\varphi (= \theta_2 - \theta_1)$, i.e., the input variable), it contains only the geometric constants of the linkage (i.e., α_1, α_2 , and α_4). If the geometric constants are known (as happens when a control algorithm must be implemented), it can be used to compute ζ as a function of φ (direct position analysis (DPA)), or, vice versa, φ as a function of ζ (inverse position analysis (IPA)). Differently, if the function $\zeta = \zeta(\varphi)$ is fully or partly known (i.e., design requirements are assigned), it can be used to compute the linkage's geometric constants α_1, α_2 , and α_4 (dimensional synthesis).

3.1.1. Position Analysis of the Spherical PPS

The computation of ζ for the assigned values of the geometric constants and φ (i.e., the solution of the DPA) is implementable by rewriting Equation (8) as follows:

$$m_1 \sin \zeta + m_2 \cos \zeta + m_3 = 0 \tag{9}$$

where:

$$\begin{cases} m_1 = -\cos \alpha_1 \sin \alpha_4 \\ m_2 = \sin \alpha_1 \sin \alpha_4 \sin \varphi \\ m_3 = \sin \alpha_1 \cos \alpha_4 \cos \varphi - \cos \alpha_2 \end{cases} \tag{10}$$

and then, by solving the quadratic equation obtained from Equation (9) through the half-tangent substitution (i.e., the change of variable $\sin x = 2t/(1+t^2)$ and $\cos x = (1-t^2)/(1+t^2)$ where $t = \tan(x/2)$). Doing so, the following closed-form solution is obtained:

$$\zeta_i = 2 \operatorname{atan2}\left(-m_1 + (-1)^i \sqrt{m_1^2 + m_2^2 - m_3^2}, m_3 - m_2\right) \quad i = 0, 1 \tag{11}$$

Analogously, the computation of φ for the assigned values of the geometric constants and ζ (i.e., the solution of the IPA) is implementable by rewriting Equation (8) as follows:

$$n_1 \sin \varphi + n_2 \cos \varphi + n_3 = 0 \tag{12}$$

where:

$$\begin{cases} n_1 = \sin \alpha_1 \sin \alpha_4 \cos \zeta \\ n_2 = \sin \alpha_1 \cos \alpha_4 \\ n_3 = -\cos \alpha_1 \sin \alpha_4 \sin \zeta - \cos \alpha_2 \end{cases} \tag{13}$$

and then, by solving the quadratic equation obtained from Equation (12) through the half-tangent substitution. Doing so, the following closed-form solution is obtained:

$$\varphi_i = 2 \operatorname{atan2} \left(-n_1 + (-1)^i \sqrt{n_1^2 + n_2^2 - n_3^2}, n_3 - n_2 \right) \Rightarrow \theta_{2,i} = \varphi_i + \psi \quad i = 0, 1 \tag{14}$$

3.1.2. Dimensional Synthesis of the Spherical PPS

Here, design requirement (i) with $\Delta\zeta = \zeta_{\max} - \zeta_{\min} = 120^\circ$ must be imposed by taking into account that a crank-rocker four-bar has to be preferred and that design requirement (ii) must be satisfied, too. In order to have a crank-rocker four-bar, the Grashof rule [45] must be satisfied, that is, the following additional conditions must be imposed:

$$\begin{cases} \alpha_1 = \alpha_{\min} = \min(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \alpha_{\min} + \alpha_{\max} \leq \frac{1}{2} \sum_{i=1,4} \alpha_i \end{cases} \tag{15}$$

where $\alpha_{\max} = \max(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$.

The two extreme positions of link 4 (i.e., the rocker) correspond to the minimum, ζ_{\min} , and the maximum, ζ_{\max} , values of ζ and occur [45] when links 1 and 2 are flattened, that is, when the three segments OA_0 , OA , and OB lie on the same plane. The application of the cosine law to the spherical triangle ΔBA_0B_0 at the occurrence of such a condition leads one to write the two equations:

$$\cos(\alpha_1 + \alpha_2) = \cos \alpha_3 \cos \alpha_4 - \sin \alpha_3 \sin \alpha_4 \sin \left(\zeta_{\min} + \frac{2}{3}\pi \right) \stackrel{\alpha_3 = \frac{\pi}{2}}{\downarrow} = -\sin \alpha_4 \sin \left(\zeta_{\min} + \frac{2}{3}\pi \right) \tag{16a}$$

$$\cos(\alpha_2 - \alpha_1) = \cos \alpha_3 \cos \alpha_4 - \sin \alpha_3 \sin \alpha_4 \sin \zeta_{\min} \stackrel{\alpha_3 = \frac{\pi}{2}}{\downarrow} = -\sin \alpha_4 \sin \zeta_{\min} \tag{16b}$$

which, by expanding the left-hand sides, can be transformed as follows:

$$\left. \begin{cases} \cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2 = \\ = -\sin \alpha_4 \sin \left(\zeta_{\min} + \frac{2}{3}\pi \right) \\ \cos \alpha_1 \cos \alpha_2 + \sin \alpha_1 \sin \alpha_2 = \\ = -\sin \alpha_4 \sin \zeta_{\min} \end{cases} \right\} \Rightarrow \begin{cases} \sin \zeta_{\min} = -\frac{\cos \alpha_1 \cos \alpha_2 + \sin \alpha_1 \sin \alpha_2}{\sin \alpha_4} \\ \cos \zeta_{\min} = \frac{\cos \alpha_1 \cos \alpha_2 \left[\cos \left(\frac{2}{3}\pi \right) - 1 \right] + \sin \alpha_1 \sin \alpha_2 \left[1 + \cos \left(\frac{2}{3}\pi \right) \right]}{\sin \left(\frac{2}{3}\pi \right) \sin \alpha_4} \end{cases} \tag{17}$$

where the trigonometric identity $\sin \left(\zeta_{\min} + \frac{2}{3}\pi \right) = \sin \zeta_{\min} \cos \left(\frac{2}{3}\pi \right) + \cos \zeta_{\min} \sin \left(\frac{2}{3}\pi \right)$ has also been introduced. Eventually, the introduction of Equation (17) into the trigonometric identity $\sin^2 \zeta_{\min} + \cos^2 \zeta_{\min} = 1$ yields:

$$2 \cos^2 \alpha_1 \cos^2 \alpha_2 \left[1 - \cos \left(\frac{2}{3}\pi \right) \right] + 2 \sin^2 \alpha_1 \sin^2 \alpha_2 \left[1 + \cos \left(\frac{2}{3}\pi \right) \right] = \sin^2 \left(\frac{2}{3}\pi \right) \sin^2 \alpha_4 \tag{18}$$

The transmission angle, τ , reaches its minimum, τ_{\min} , and maximum, τ_{\max} , values when links 1 and 3 are flattened [45], that is, when the three segments OA_0 , OA , and OB_0 lie on the same plane. The application of the cosine law to the spherical triangle ΔAB_0B at the occurrence of such a condition leads one to write the two equations:

$$\cos(\alpha_3 + \alpha_1) = \cos \alpha_2 \cos \alpha_4 + \sin \alpha_2 \sin \alpha_4 \cos \tau_{\max} \tag{19a}$$

$$\cos(\alpha_3 - \alpha_1) = \cos \alpha_2 \cos \alpha_4 + \sin \alpha_2 \sin \alpha_4 \cos \tau_{\min} \tag{19b}$$

which, since $\alpha_3 = \frac{\pi}{2}$, become:

$$-\sin \alpha_1 = \cos \alpha_2 \cos \alpha_4 + \sin \alpha_2 \sin \alpha_4 \cos \tau_{\max} \tag{20a}$$

$$\sin \alpha_1 = \cos \alpha_2 \cos \alpha_4 + \sin \alpha_2 \sin \alpha_4 \cos \tau_{\min} \tag{20b}$$

If the values of τ_{\min} and τ_{\max} are assigned so that design requirement (ii) is satisfied, Equations (18) and (20) become a system of three trigonometric equations in three unknowns (i.e., α_1 , α_2 , and α_4), whose solution provides the sought-after sizes that satisfy both the design requirements. If the found solution also satisfies condition (15), it can be accepted; otherwise, the system must be solved again with new assigned values of τ_{\min} and τ_{\max} until all the conditions are satisfied.

The replacement of Equation (20a) with the sum of Equation (20a,b) and the introduction of Equation (20b) into Equation (18) after replacing $\cos^2 \alpha_1$ with the trigonometric identity $\cos^2 \alpha_1 = 1 - \sin^2 \alpha_1$ transform the system to solve as follows:

$$\left[2(\cos \alpha_2 \cos \alpha_4 + \sin \alpha_2 \sin \alpha_4 \cos \tau_{\min})^2 \left[\sin^2 \alpha_2 - \cos^2 \alpha_2 + \cos \left(\frac{2}{3} \pi \right) \right] + 2 \cos^2 \alpha_2 \left[1 - \cos \left(\frac{2}{3} \pi \right) \right] \right] = \sin^2 \left(\frac{2}{3} \pi \right) \sin^2 \alpha_4 \tag{21a}$$

$$\left. \begin{array}{l} 2 \cos \alpha_2 \cos \alpha_4 + q \sin \alpha_2 \sin \alpha_4 = 0 \\ \text{with} \\ q = (\cos \tau_{\max} + \cos \tau_{\min}) \end{array} \right\} \begin{array}{l} \text{if } \alpha_2 \neq \frac{\pi}{2} \\ \Downarrow \\ \Rightarrow \end{array} \tan \alpha_2 = -\frac{2 \cos \alpha_4}{q \sin \alpha_4} \tag{21b}$$

$$\sin \alpha_1 = \cos \alpha_2 (\cos \alpha_4 + \tan \alpha_2 \sin \alpha_4 \cos \tau_{\min}) = \pm \frac{\cos \alpha_4 + \tan \alpha_2 \sin \alpha_4 \cos \tau_{\min}}{\sqrt{1 + \tan^2 \alpha_2}} \tag{21c}$$

where Equation (21c) is Equation (20b) transformed by taking into account the trigonometric identity $\cos x = \pm 1 / \sqrt{1 + \tan^2 x}$. It is worth stressing that the value $\alpha_2 = \pi/2$, excluded in the deduction of Equation (21b), leads the determination of acceptable values of α_4 (i.e., $\alpha_4 \in [0, \pi]$) only if simultaneously $q = (\cos \tau_{\max} + \cos \tau_{\min}) = 0$. In this case, such a condition makes Equation (21b) identically satisfied and transforms Equation (20b) as follows $\sin \alpha_1 = \sin \alpha_4 \cos \tau_{\min}$, whose substitution into Equation (18) yields the following condition on τ_{\min} : $2 \cos^2 \tau_{\min} \left[1 + \cos \left(\frac{2}{3} \pi \right) \right] = \sin^2 \left(\frac{2}{3} \pi \right)$, that is, $\cos \tau_{\min} = -\cos \tau_{\max} = \sqrt{3}/2$. Such formulas make one determine $\tau_{\min} = 30^\circ$, $\tau_{\max} = 150^\circ$, and, as possible values for α_1 and α_4 that also satisfy condition (15), $\alpha_1 = 46.7805^\circ = 0.81647$ rad and $\alpha_4 = 57.29578^\circ = 1$ rad. Eventually, the introduction of $\alpha_2 = \pi/2$ into Equation (17) yields $\sin \zeta_{\min} = -\sin \alpha_1 / \sin \alpha_4 \equiv -\cos \tau_{\min}$, which provides $\zeta_{\min} = -60^\circ$ and $\zeta_{\max} = \zeta_{\min} + 120^\circ = 60^\circ$ for $\tau_{\min} = 30^\circ$.

The introduction of Equation (21b) into Equation (21a,c), where $\cos\alpha_2$ and $\sin\alpha_2$ have been replaced through the trigonometric identities $\cos x = \pm 1/\sqrt{1 + \tan^2 x}$ and $\sin x = \pm \tan x/\sqrt{1 + \tan^2 x}$, transforms system (21) into the final form:

$$\tan \alpha_2 = -\frac{2 \cos \alpha_4}{q \sin \alpha_4} \tag{22a}$$

$$\sin \alpha_1 = \pm \frac{r|\sin \alpha_4| \cos \alpha_4}{\sqrt{q^2 \sin^2 \alpha_4 + 4 \cos^2 \alpha_4}} \tag{22b}$$

$$\frac{2q^2}{q^2 \sin^2 \alpha_4 + 4 \cos^2 \alpha_4} \left\{ \left[1 - \cos\left(\frac{2}{3}\pi\right) \right] + \cos^2 \alpha_4 \left(\frac{r}{q}\right)^2 \left[1 + \cos\left(\frac{2}{3}\pi\right) - \frac{2q^2 \sin^2 \alpha_4}{q^2 \sin^2 \alpha_4 + 4 \cos^2 \alpha_4} \right] \right\} = \sin^2\left(\frac{2}{3}\pi\right) \tag{22c}$$

with $r = (\cos \tau_{\max} - \cos \tau_{\min})$.

The introduction of the trigonometric identity $\sin^2\alpha_4 = 1 - \cos^2\alpha_4$ into Equation (22c), together with the variable change $p = \cos^2\alpha_4$, transforms it into the following quadratic equation:

$$g_2 p^2 + 2g_1 p + g_0 = 0 \tag{23}$$

with:

$$\begin{cases} g_0 = q^4 \left\{ 2 \left[1 - \cos\left(\frac{2}{3}\pi\right) \right] - \sin^2\left(\frac{2}{3}\pi\right) \right\} \\ g_1 = q^2 \left\{ [4 - q^2] \left[1 - \cos\left(\frac{2}{3}\pi\right) - \sin^2\left(\frac{2}{3}\pi\right) \right] + r^2 \left[\cos\left(\frac{2}{3}\pi\right) - 1 \right] \right\} \\ g_2 = 2r^2 \left\{ 4 \left[1 + \cos\left(\frac{2}{3}\pi\right) \right] - \left[\cos\left(\frac{2}{3}\pi\right) - 1 \right] q^2 \right\} - \sin^2\left(\frac{2}{3}\pi\right) [4 - q^2]^2 \end{cases} \tag{24}$$

whose solutions are:

$$p_i = \frac{-g_1 + (-1)^i \sqrt{g_1^2 - g_0 g_2}}{g_2} \Rightarrow (\cos \alpha_4)_{ij} = (-1)^j \sqrt{p_i} \quad i, j = 0, 1 \tag{25}$$

The values of α_4 obtained from Equation (25) as a function of τ_{\min} and τ_{\max} , when back substituted into Equation (22a,b), provide the corresponding values of α_1 and α_2 . Eventually, the back substitution of the so-obtained triplets $(\alpha_1, \alpha_2, \alpha_4)$ into Equation (17) yields the corresponding values of ζ_{\min} . Equation (25) yields, at most, four real values for $\cos\alpha_4$ and as many values of α_4 in the range of $[0, \pi]$ rad. Successively, Equation (22a) (Equation (22b)) associates one value of $\tan\alpha_2$ (two values of $\sin\alpha_1$), which corresponds to two values of α_2 (of α_1) in the range of $[0, \pi]$ rad, to each computed value of α_4 . In conclusion, system (22) can have up to eight values of $(\alpha_1, \alpha_2, \alpha_4)$, with α_1, α_2 , and α_4 belonging to the range of $]0, \pi[$ rad that solve it and are selectable as possible sides of the studied spherical four bar.

This procedure has been implemented by using many values of τ_{\min} and τ_{\max} , but it has not led to the identification of values of $\alpha_1, \alpha_2, \alpha_4$, and ζ_{\min} that correspond to the values of τ_{\min} and τ_{\max} better than the ones found above for the case $\alpha_2 = 90^\circ = \pi/2$ rad (i.e., $\tau_{\min} = 30^\circ, \tau_{\max} = 150^\circ$). As a consequence, those values (i.e., $\tau_{\min} = 30^\circ, \tau_{\max} = 150^\circ, \alpha_2 = 90^\circ = \pi/2$ rad, $\alpha_1 = 46.7805^\circ = 0.81647$ rad, $\alpha_4 = 57.29578^\circ = 1$ rad, and $\zeta_{\min} = -\zeta_{\max} = -60^\circ = -1.0472$ rad) are adopted as the optimal solution of the dimensional synthesis.

3.2. First Decoupled PPS

Figure 6 illustrates the adopted notations for the kinematic analysis and dimensional synthesis of the PPS shown in Figure 3a. With reference to Figure 6, d_2 is the actuated-joint variable of the P pair; τ is the transmission angle; and a_1, a_3 , and a_4 are the distances

between the two ending R-pair axes of links 1, 3, and 4, respectively, and they are the geometric constants of this PPS.

When the pan rotation is locked, the additional limb becomes a single-DOF planar linkage of type PRRR, whose motion plane is the plane perpendicular to the tilt-rotation axis that passes through the pan-rotation axis. In this plane, the adopted notations make it possible to write the following two relationships:

$$d_2 = a_3 \sin(\tau + \zeta) - a_1 \sin \zeta \tag{26a}$$

$$a_1 \cos \zeta = a_4 + a_3 \cos(\tau + \zeta) \tag{26b}$$

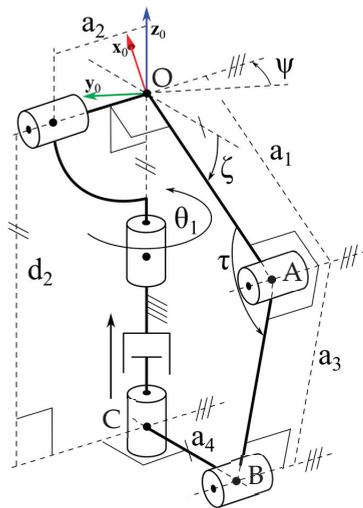


Figure 6. Kinematic scheme of the 1st decoupled PPS: notations.

Equation (26a,b) immediately allow for the determination of the explicit expressions of $\sin(\tau + \zeta)$ and $\cos(\tau + \zeta)$, respectively, whose introduction into the trigonometric identity $\cos^2(\tau + \zeta) + \sin^2(\tau + \zeta) = 1$ yields the closure equation:

$$a_3^2 = (d_2 + a_1 \sin \zeta)^2 + (a_1 \cos \zeta - a_4)^2 \Rightarrow a_3^2 = d_2^2 + a_4^2 + a_1^2 + 2a_1(d_2 \sin \zeta - a_4 \cos \zeta) \tag{27}$$

3.2.1. Position Analysis of the 1st Decoupled PPS

The computation of ζ for assigned values of the geometric constants and d_2 (i.e., the solution of the DPA) is implementable by rewriting Equation (27) as follows:

$$k_1 \sin \zeta + k_2 \cos \zeta + k_3 = 0 \tag{28}$$

where:

$$\begin{cases} k_1 = 2a_1d_2 \\ k_2 = -2a_1a_4 \\ k_3 = d_2^2 + a_4^2 + a_1^2 - a_3^2 \end{cases} \tag{29}$$

and then, by solving the quadratic equation obtained from Equation (28) through the half-tangent substitution. Doing so, the following closed-form solution is obtained:

$$\zeta_i = 2 \operatorname{atan}2 \left(-k_1 + (-1)^i \sqrt{k_1^2 + k_2^2 - k_3^2}, k_3 - k_2 \right) \quad i = 0, 1 \tag{30}$$

Differently, the computation of d_2 for assigned values of the geometric constants and ζ (i.e., the solution of the IPA) is implementable by simply noting that Equation (27) is a quadratic equation in d_2 , whose solution is:

$$d_{2,i} = -a_1 \sin \zeta + (-1)^i \sqrt{a_1^2 \sin^2 \zeta - a_4^2 - a_1^2 + a_3^2 + 2a_1 a_4 \cos \zeta} \quad i = 0, 1 \quad (31)$$

3.2.2. Dimensional Synthesis of the 1st Decoupled PPS

With reference to Figure 6, the transmission angle is computable as a function of the actuated-joint variable, d_2 , as follows:

$$\left. \begin{aligned} \|B - O\|^2 &= [(B - A) + (A - O)] \times [(B - A) + (A - O)] = \\ &= \|B - A\|^2 + \|A - O\|^2 + 2(B - A) \times (A - O) \\ \|B - O\|^2 &= d_2^2 + a_4^2; \|B - A\|^2 = a_3^2; \|A - O\|^2 = a_1^2 \\ (B - A) \times (A - O) &= 2a_1 a_3 \cos(\pi - \tau) = -2a_1 a_3 \cos \tau \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} \cos \tau &= \frac{a_1^2 + a_3^2 - a_4^2 - d_2^2}{2a_1 a_3} = \\ &= \frac{1 + \left(\frac{a_3}{a_1}\right)^2 - \left(\frac{a_4}{a_1}\right)^2 - \left(\frac{d_2}{a_1}\right)^2}{2\left(\frac{a_3}{a_1}\right)} \end{aligned} \right. \quad (32)$$

The same figure reveals that τ_{\max} (ζ_{\min}) and τ_{\min} (ζ_{\max}) correspond to the maximum, $d_{2,\max}$, and minimum, $d_{2,\min}$, values of d_2 , respectively. Accordingly, if $\zeta_{\max} = 60^\circ$ and $\zeta_{\min} = -60^\circ$ (considering that design requirements (i) with $\Delta\zeta = \zeta_{\max} - \zeta_{\min} = 120^\circ$ must be imposed) are chosen, the introduction of $(\tau_{\max}, \zeta_{\min})$, and then, of $(\tau_{\min}, \zeta_{\max})$, into Equation (26b) yields the following two relationships:

$$\left(\frac{a_4}{a_1}\right) + \left(\frac{a_3}{a_1}\right) \cos\left(\tau_{\max} - \frac{\pi}{3}\right) = \cos\left(\frac{\pi}{3}\right) \quad (33a)$$

$$\left(\frac{a_4}{a_1}\right) + \left(\frac{a_3}{a_1}\right) \cos\left(\tau_{\min} + \frac{\pi}{3}\right) = \cos\left(\frac{\pi}{3}\right) \quad (33b)$$

which are compatible if and only if the following relationship holds:

$$\cos\left(\tau_{\max} - \frac{\pi}{3}\right) = \cos\left(\tau_{\min} + \frac{\pi}{3}\right) \Rightarrow \tau_{\max} - \frac{\pi}{3} = \pm\left(\tau_{\min} + \frac{\pi}{3}\right) \Rightarrow \left\{ \begin{aligned} \tau_{\max} - \tau_{\min} &= \frac{2\pi}{3} \\ or \\ \tau_{\max} + \tau_{\min} &= 0 \end{aligned} \right. \quad (34)$$

Since 90° is the best transmission angle and the variation range of τ should be centered on this value, the condition $\tau_{\max} + \tau_{\min} = 0$ is not valid, whereas the other condition can be satisfied by choosing $\tau_{\max} = 150^\circ (= 90^\circ + 60^\circ)$ and $\tau_{\min} = 30^\circ (= 90^\circ - 60^\circ)$, which, when introduced into Equation (33), yield the condition $a_4 = 0.5a_1$ (see Figure 7). Moreover, Equation (26a) yields (see Figure 7):

$$\left. \begin{aligned} d_{2,\max} &= a_3 \sin(\tau_{\max} + \zeta_{\min}) - a_1 \sin \zeta_{\min} \\ d_{2,\min} &= a_3 \sin(\tau_{\min} + \zeta_{\max}) - a_1 \sin \zeta_{\max} \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} \Delta d_2 &= d_{2,\max} - d_{2,\min} = \\ &= a_3[\sin(\tau_{\max} + \zeta_{\min}) - \sin(\tau_{\min} + \zeta_{\max})] + \\ &\quad + a_1(\sin \zeta_{\max} - \sin \zeta_{\min}) \\ 2d_{2,\text{average}} &= d_{2,\max} + d_{2,\min} = \\ &= a_3[\sin(\tau_{\max} + \zeta_{\min}) + \sin(\tau_{\min} + \zeta_{\max})] + \\ &\quad - a_1(\sin \zeta_{\max} + \sin \zeta_{\min}) \end{aligned} \right. \quad (35)$$

System (35) leads one to determine the following explicit formulas:

$$\left\{ \begin{aligned} a_1 &= \frac{2d_{2,\text{average}}[\sin(\tau_{\max} + \zeta_{\min}) - \sin(\tau_{\min} + \zeta_{\max})] - \Delta d_2[\sin(\tau_{\max} + \zeta_{\min}) + \sin(\tau_{\min} + \zeta_{\max})]}{\Delta} \\ a_3 &= -\frac{\Delta d_2(\sin \zeta_{\max} + \sin \zeta_{\min}) + 2d_{2,\text{average}}(\sin \zeta_{\max} - \sin \zeta_{\min})}{\Delta} \end{aligned} \right. \quad (36)$$

with:

$$\Delta = -\{[\sin(\tau_{\max} + \zeta_{\min}) - \sin(\tau_{\min} + \zeta_{\max})](\sin \zeta_{\max} + \sin \zeta_{\min}) + (\sin \zeta_{\max} - \sin \zeta_{\min})[\sin(\tau_{\max} + \zeta_{\min}) + \sin(\tau_{\min} + \zeta_{\max})]\} \quad (37)$$

which, for the above-determined values of τ_{\max} , ζ_{\min} , τ_{\min} , and ζ_{\max} , yields:

$$a_1 = \frac{\Delta d_2}{\sqrt{3}}; a_3 = 2d_{2,\text{average}} = 2d_{2,\text{min}} + \Delta d_2 \quad \Delta d_2 = a_1 \sqrt{3} \quad \Downarrow \quad a_1 \sqrt{3} \left(\frac{2d_{2,\text{min}}}{\Delta d_2} + 1 \right) \quad (38)$$

where the values of $d_{2,\text{min}}$ and Δd_2 depend on how the actuated P-pair is sized (in Figure 7, $d_{2,\text{min}} = d_{20}$ and $\Delta d_2 = d_1 = 2a_1 \sin \zeta_+$). Formula (38), together with the above-deduced relationship $a_4 = 0.5a_1$, provides the sizes of the 1st decoupled PPS for the chosen values of τ_{\max} , ζ_{\min} , τ_{\min} , and ζ_{\max} .

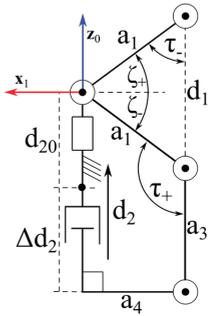


Figure 7. First decoupled PPS sized so that $a_4 = a_1 \cos \zeta_{\pm}$.

3.3. Second Decoupled PPS

Figure 8 illustrates the adopted notations for the kinematic analysis and dimensional synthesis of the PPS shown in Figure 4b. With reference to Figure 8, d_2 is the actuated-joint variable of the P pair and τ is the transmission angle. The lengths a_2 , a_3 , and a_4 are the distances of the S-pair center (point A in Figure 8), respectively, from the plane perpendicular to the pan-rotation axis that passes through point C, and from the axis of the second R pair of the RRS limb. Eventually, a_5 is the distance between the axes of the two R pairs of the RRS limb.

The following relationships hold among the above-defined geometric parameters:

$$d_2 = a_3 - a_2 \sin \zeta \quad (39a)$$

$$\tau = \frac{\pi}{2} - \zeta \quad (39b)$$

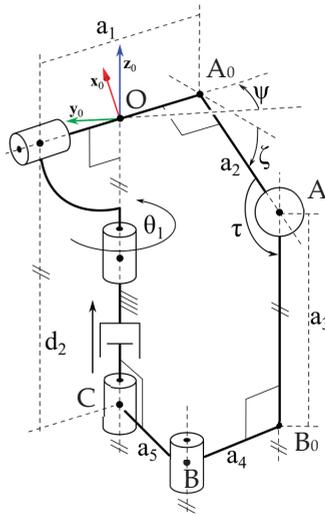


Figure 8. Kinematic scheme of the 2nd decoupled PPS: notations.

3.3.1. Position Analysis of the 2nd Decoupled PPS

Equation (39a) immediately provides the unique IPA solution (i.e., the determination of d_2 for assigned values of the geometric constants and ζ); whereas, for the DPA (i.e., the computation of ζ for assigned values of the geometric constants and d_2), it straightforwardly gives the following solution formula:

$$\zeta = \arcsin\left(\frac{a_3 - d_2}{a_2}\right) \tag{40}$$

which provides a unique solution for $\zeta \in [-90^\circ, 90^\circ]$.

3.3.2. Dimensional Synthesis of the 2nd Decoupled PPS

Equation (39b) states a linear relationship between the transmission angle, τ , and the tilt angle, ζ , that does not depend on the geometric constants of the PPS and gives the best transmission angle (i.e., 90°) for $\zeta = 0$. Therefore, the best choice is $\zeta_{\max} = -\zeta_{\min} = 60^\circ$, which gives $\tau_{\max} = 150^\circ (= 90^\circ + 60^\circ)$ and $\tau_{\min} = 30^\circ (= 90^\circ - 60^\circ)$. Moreover, since the geometric constants are not involved in Equation (39b), they can be freely chosen, provided that the following geometric constraints are satisfied (see Figure 8 and Equation (39a)):

$$\begin{aligned} \zeta_{\max} = -\zeta_{\min} = 60^\circ \\ \Delta d_2 = d_{2,\max} - d_{2,\min} = a_2(\sin \zeta_{\max} - \sin \zeta_{\min}) \quad \downarrow \quad a_2\sqrt{3} \end{aligned} \tag{41a}$$

$$d_{2,\text{average}} = \frac{d_{2,\max} + d_{2,\min}}{2} = a_3 - a_2 \frac{\sin \zeta_{\max} + \sin \zeta_{\min}}{2} \quad \downarrow \quad a_3 \tag{41b}$$

$$\sqrt{\|A_0 - O\|^2 + a_2^2} \leq a_4 + a_5 \tag{41c}$$

where the values of $d_{2,\text{average}}$ and Δd_2 depend on how the actuated P-pair is sized.

4. Discussion and Accuracy Analysis

The above-reported kinematic analyses show that all three PPS architectures have closed-form solutions to their IPA and DPA, even though the formulas of the spherical PPS are slightly more cumbersome than those of the other two PPSs. Since the complexity of the formulas is not a problem when a motion control software uses them, the three selected architectures are equivalent from the motion control point of view. Of course, the fact that the spherical PPS has slightly coupled kinematics, which need the motion of both the actuators to keep the tilt angle fixed when only the pan angle has to change, requires a motion control software slightly more complex than that of the other two PPSs. Indeed, the “complexity” simply reduces to satisfying the condition that φ ($=(\theta_2 - \theta_1)$) be constant (see, Figure 5).

The above-reported dimensional synthesis determined the same optimal values of ζ_{\min} , ζ_{\max} , τ_{\min} , and τ_{\max} (i.e., $\zeta_{\max} = -\zeta_{\min} = 60^\circ$, $\tau_{\min} = 30^\circ$, and $\tau_{\max} = 150^\circ$) for all of the three PPSs. Thus, even though the extreme values of the transmission angle are too far from its optimal value (i.e., $\tau = 90^\circ$), this drawback is common to all of them, and, in practice, is acceptable, since the telescope is light and does not carry heavy loads.

In order to complete the comparison among the selected PPSs, the relationships and sizes determined in the previous section must be used to evaluate the positioning precision (accuracy) of the additional limbs. Such an evaluation consists of estimating how a possible error in the actuated-joint variable of the additional limb affects the tilt angle, and it is implementable as follows.

For the spherical PPS, the differentiation of Equation (8) yields:

$$d\zeta = \frac{\sin \alpha_1 (\sin \alpha_4 \cos \varphi \cos \zeta - \cos \alpha_4 \sin \varphi)}{\sin \alpha_4 (\sin \alpha_1 \sin \varphi \sin \zeta + \cos \alpha_1 \cos \zeta)} d\varphi \quad (42)$$

where, in the coefficient that multiplies $d\varphi$, ζ is related to φ through Equation (11), whereas the values of α_1 and α_4 are the optimal ones determined in the above-reported dimensional synthesis (i.e., $\alpha_1 = 46.7805^\circ = 0.81647$ rad and $\alpha_4 = 57.29578^\circ = 1$ rad). By replacing the differentials $d\zeta$ and $d\varphi$ with the joint-variable errors $\Delta\zeta$ and $\Delta\varphi$, respectively, in Equation (42), and then, by taking the maximum absolute value of the coefficient that multiplies $d\varphi$ in Equation (42), Equation (42) leads to the determination of the following limitation on $|\Delta\zeta|$:

$$|\Delta\zeta| \leq 1.7321 |\Delta\varphi| \quad (43)$$

which proves that, for the spherical PPS, $|\Delta\zeta|$ has the same order of magnitude as $|\Delta\varphi|$.

For the 1st decoupled PPS, the differentiation of Equation (27) yields:

$$d\zeta = -\frac{\left(\frac{d_2}{a_1}\right) + \sin \zeta}{\left(\frac{d_2}{a_1}\right) \cos \zeta + \left(\frac{a_4}{a_1}\right) \sin \zeta} d\left(\frac{d_2}{a_1}\right) \quad (44)$$

where, in the coefficient that multiplies $d(d_2/a_1)$, (d_2/a_1) is related to ζ through Equation (31), whereas the values of a_1 and a_4 are the optimal ones determined in the above-reported dimensional synthesis (i.e., $a_4 = 0.5a_1$, $a_1\sqrt{3} = \Delta d_2$). By replacing the differentials $d\zeta$ and $d(d_2/a_1)$ with the joint-variable errors $\Delta\zeta$ and $\Delta(d_2/a_1)$, respectively, in Equation (44), and then, taking the maximum absolute value of the coefficient that multiplies $d(d_2/a_1)$, for $\zeta \in [-60^\circ, 60^\circ]$, $\Delta d_2 = \sqrt{3}$, and $d_{2,\min} = 0.5a_1$, in Equation (44), Equation (44) leads to the determination of the following limitation on $|\Delta\zeta|$:

$$|\Delta\zeta| \leq 2 \left| \Delta \left(\frac{d_2}{a_1} \right) \right| \quad (45)$$

which proves that, for the 1st decoupled PPS, $|\Delta\zeta|$ has the same order of magnitude as $|\Delta(d_2/a_1)|$.

For the 2nd decoupled PPS, the differentiation of Equation (39a) yields:

$$d\zeta = -\frac{1}{\cos \zeta} d\left(\frac{d_2}{a_2}\right) \quad (46)$$

where, in the coefficient that multiplies $d(d_2/a_2)$, ζ is related to (d_2/a_2) through Equation (39a). By replacing the differentials $d\zeta$ and $d(d_2/a_2)$ with the joint-variable errors $\Delta\zeta$ and $\Delta(d_2/a_2)$, respectively, in Equation (46), and then, taking the maximum absolute value of the coefficient that multiplies $d(d_2/a_2)$, for $\zeta \in [-60^\circ, 60^\circ]$, in Equation (46), Equation (46) leads to the determination of the following limitation on $|\Delta\zeta|$:

$$|\Delta\zeta| \leq 2 \left| \Delta\left(\frac{d_2}{a_2}\right) \right| \quad (47)$$

which proves that the 2nd decoupled PPS has the same accuracy as the 1st decoupled PPS with $|\Delta\zeta|$ that has the same order of magnitude as $|\Delta(d_2/a_2)|$.

The comparison of inequalities (45)–(47) shows that the three selected PPSs have essentially the same accuracy, with the spherical PPS being slightly better. In short, the kinematic analysis, the dimensional synthesis, and the accuracy analysis do not make any out of the three PPSs prevail over the remaining two. As a consequence, the fact that the spherical PPS has the minimum number of links and can actuate the tilt angle by using a continuous rotation as the input motion leads one to the conclusion that it should be preferred. Figure 9 shows the 3D CAD model of an RTS actuated through the spherical PPS with the above-determined sizes (i.e., $\alpha_2 = 90^\circ = \pi/2$ rad, $\alpha_1 = 46.7805^\circ = 0.81647$ rad, and $\alpha_4 = 57.29578^\circ = 1$ rad (see Figure 5)), whereas Figures 10 and 11 show the tilt angle, ζ , and the transmission angle, τ , respectively, as a function of $\varphi (= \theta_2 - \theta_1)$ for the same PPS. Eventually, the video “video_RTS.mp4” that shows the motion of the 3D CAD model (only tilt rotation followed by only pan rotation and, then, combined pan and tilt rotations) is downloadable from the supplementary materials that accompany this paper.

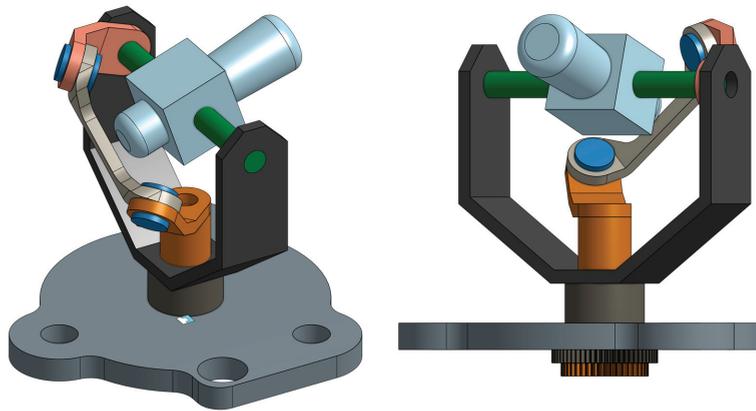


Figure 9. Two views of the 3D CAD model of an RTS actuated by means of a spherical PPS whose sizes are the ones determined in Section 3.

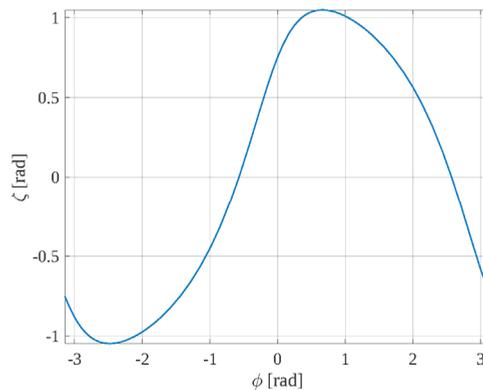


Figure 10. Diagram of the tilt angle, ζ , as a function of ϕ ($=\theta_2 - \theta_1$) for the spherical PPS with $\alpha_2 = 90^\circ = \pi/2$ rad, $\alpha_1 = 46.7805^\circ = 0.81647$ rad, and $\alpha_4 = 57.29578^\circ = 1$ rad (see Figure 5).

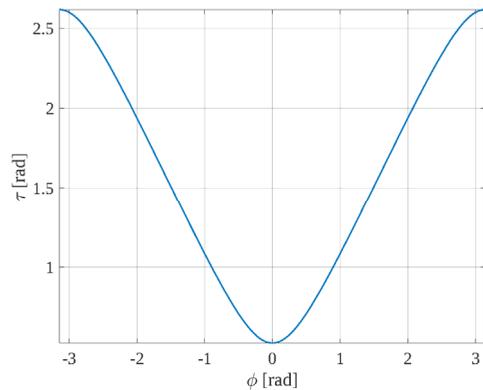


Figure 11. Diagram of the transmission angle, τ , as a function of ϕ ($=\theta_2 - \theta_1$) for the spherical PPS with $\alpha_2 = 90^\circ = \pi/2$ rad, $\alpha_1 = 46.7805^\circ = 0.81647$ rad, and $\alpha_4 = 57.29578^\circ = 1$ rad (see Figure 5).

5. Conclusions

The literature on parallel pointing systems (PPS) has been reviewed to identify PPS architectures suitable for actuating robotic total stations (RTS). This review has led to the selection of three PPS architectures, named “Spherical PPS”, “1st Decoupled PPS”, and “2nd Decoupled PPS”.

The kinematic analysis, the dimensional synthesis, and the accuracy analysis of the selected PPS architectures have been addressed in detail in order to compare them. These studies have yielded the following conclusions: all of them (i) have simple motion control algorithms based on closed-form formulas, (ii) can be sized so that the telescope of the RTS can collimate toward any target on the upper hemisphere, and (iii) have a good accuracy.

Even though the three PPSs have comparable performances, the fact that the “Spherical PPS” has the minimum number of links and can actuate the tilt angle by using a continuous rotation as the input motion makes it prevail over the remaining two.

Supplementary Materials: The file “video_RTS.mp4”, containing a video of the RTS, shown in Figure 9, in motion, can be downloaded at <https://www.mdpi.com/article/10.3390/machines12010054/s1>.

Author Contributions: Conceptualization, H.S., R.D.G., R.S. and M.G.; methodology, H.S. and R.D.G.; software, H.S., R.D.G. and R.S.; validation, H.S., R.D.G., R.S. and M.G.; formal analysis, H.S., R.D.G., R.S. and M.G.; resources, R.S. and M.G.; writing—original draft preparation, H.S., R.D.G., R.S. and M.G.; writing—review and editing, H.S., R.D.G., R.S. and M.G.; supervision, R.D.G. and

M.G.; funding acquisition, R.D.G., H.S. and M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been developed at the Laboratory of Mechatronics and Virtual Prototyping (LaMaViP) of Ferrara Technopole, supported by UNIFE FIRD2023 fund, in partnership with Laboratory of Applied Robotics of Federal University of Santa Catarina, supported by CNPq—*Conselho Nacional de Desenvolvimento Científico e Tecnológico* (National Council for Scientific and Technological Development) Project 307249/2021-2, Brazil.

Data Availability Statement: All the data used in this paper are reported in it.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- Nadolinet, L.; Levin, E.; Akhmedov, D. *Surveying Instruments and Technology*; CRC Press: Boca Raton, FL, USA, 2017; ISBN 978-1-4987-6238-0.
- Kirschner, H.; Stempfhuber, W. The Kinematic Potential of Modern Tracking Total Stations—A State of the Art Report on the Leica TPS 12001. In Proceedings of the 1st International Conference on Machine Control & Guidance, Zurich, Switzerland, 24–26 June 2008; Ingensand, H., Stempfhuber, W., Eds.; Eidgenössische Technische Hochschule (ETH): Zurich, Switzerland, 2008; pp. 51–60. [CrossRef]
- Kleemaier, G.; Maar, H.; Zogg, H.M. Enhanced Automation Performance of Total Stations for Kinematic Applications Using the ATR Plus Technology. In Proceedings of the 5th Int. Conf. on Machine Control and Guidance, Vichy, France, 5–6 October 2016.
- Cook, D. Robotic Total Stations and Remote Data Capture: Challenges in Construction. *Geotech. News* **2006**, *24*, 42–45.
- Kontogianni, V.; Kornarou, S.; Stiros, S. Monitoring with Electronic Total Stations: Performance and Accuracy of Prismatic and Non-Prismatic Reflectors. *Geotech. News* **2007**, *25*, 30–39.
- Ehrhart, M.; Lienhart, W. Object Tracking with Robotic Total Stations: Current Technologies and Improvements Based on Image Data. *J. Appl. Geod.* **2017**, *11*, 131–142. [CrossRef]
- Brownjohn, J.M.W.; Koo, K.Y.; Scullion, A.; List, D. Operational deformations in long-span bridges. *Struct. Infrastruct. Eng.* **2015**, *11*, 556–574. [CrossRef]
- Psimoulis, P.; Peppas, L.; Ince, S.; Meng, X. Combination of GPS and RTS Measurements for the Monitoring of Semi-Static and Dynamic Motion of Pedestrian Bridge. In Proceedings of the 3rd Joint International Symposium on Deformation Monitoring, Vienna, Austria, 30 March–1 April 2016.
- Lienhart, W.; Ehrhart, M.; Grick, M. High frequent total station measurements for the monitoring of bridge vibrations. *J. Appl. Geod.* **2017**, *11*, 1–8. [CrossRef]
- Yu, J.; Zhu, P.; Xu, B.; Meng, X. Experimental Assessment of High Sampling-Rate Robotic Total Station for Monitoring Bridge Dynamic Responses. *Measurement* **2017**, *104*, 60–69. [CrossRef]
- Psimoulis, P.; Stiros, S. Measuring Deflections of a Short-Span Railway Bridge Using a Robotic Total Station. *J. Bridge Eng.* **2013**, *18*, 182–185. [CrossRef]
- Moschas, F.; Stiros, S. Rapid Decay of a Timber Footbridge and Changes in Its Modal Frequencies Derived from Multiannual Lateral Deflection Measurements. *J. Bridge Eng.* **2014**, *19*, 05014005.
- Psimoulis, P.; Stiros, S. A Supervised Learning Computer-Based Algorithm to Derive the Amplitude of Oscillations of Structures Using Noisy GPS and Robotic Theodolites (RTS) Records. *Comput. Struct.* **2012**, *92–93*, 337–348. [CrossRef]
- Charalampous, E.; Psimoulis, P.; Guillaume, S.; Spiridonakos, M.; Klis, R.; Bürki, B.; Rothacher, M.; Chatzi, E.; Luchsinger, R.; Feltrinnet, G. Measuring Sub-mm Structural Displacements Using QDaedalus: A Digital Clip-On Measuring System Developed for Total Stations. *Appl. Geomat.* **2015**, *7*, 91–101. [CrossRef]
- Lackner, S.; Lienhart, W. Impact of Prism Type and Prism Orientation on the Accuracy of Automated Total Station Measurements. In Proceedings of the 3rd Joint International Symposium on Deformation Monitoring (JISDM), Vienna, Austria, 30 March–1 April 2016.
- Frukacz, M.; Presl, R.; Wieser, A.; Favot, D. Pushing the Sensitivity Limits of RTS-Based Continuous Deformation Monitoring of an Alpine Valley. *Appl. Geomat.* **2017**, *9*, 81–92. [CrossRef]
- Gatti, M. Measurement of vibration frequencies of ties in masonry arches by means of a robotic total station. *Earthq. Eng. Eng. Vib.* **2022**, *21*, 489–500. [CrossRef]
- Tischler, C.R.; Samuel, A.E.; Hunt, K.H. Kinematic chains for robot hands: Part 1 orderly number-synthesis. *Mech. Mach. Theory* **1995**, *30*, 1193–1215. [CrossRef]
- Tischler, C.R.; Samuel, A.E.; Hunt, K.H. Kinematic chains for robot hands: Part 2 kinematic constraints, classification, connectivity, and actuation. *Mech. Mach. Theory* **1995**, *30*, 1217–1239. [CrossRef]
- Tsai, L.-W. *Systematic Enumeration of Parallel Manipulators*; Technical Report ISR.; TR 1998-33; University of Maryland: College Park, MD, USA, 1998.

21. Tsai, L.-W. *Mechanism Design: Enumeration of Kinematic Structures According to Function*; Mechanical Engineering Series; CRC Press: Washington, DC, USA, 2001.
22. Alizade, R.; Bayram, C. Structural synthesis of parallel manipulators. *Mech. Mach. Theory* **2004**, *39*, 857–870. [CrossRef]
23. Sunkari, R.P.; Schmidt, L.C. Structural synthesis of planar kinematic chains by adapting a McKay-type algorithm. *Mech. Mach. Theory* **2006**, *41*, 1021–1030. [CrossRef]
24. Fang, Y.; Tsai, L.-W. Structure synthesis of a class of 4-dof and 5-dof parallel manipulators with identical limb structures. *Int. J. Robot. Res.* **2002**, *21*, 799–810. [CrossRef]
25. Huang, Z.; Li, Q. General methodology for type synthesis of symmetrical lower-mobility parallel manipulators and several novel manipulators. *Int. J. Robot. Res.* **2002**, *21*, 131–145. [CrossRef]
26. Huang, Z.; Li, Q. Type synthesis of symmetrical lower-mobility parallel mechanisms using the constraint-synthesis method. *Int. J. Robot. Res.* **2003**, *22*, 59–79. [CrossRef]
27. Fang, Y.; Tsai, L.-W. Structure synthesis of a class of 3-dof rotational parallel manipulators. *IEEE Trans. Robot. Autom.* **2004**, *20*, 117–121. [CrossRef]
28. Kong, X.; Gosselin, C. Type synthesis of 3-dof translational parallel manipulators based on screw theory and virtual joint. In Proceedings of the 15th CISM-IFToMM Symposium on Robot Design, Dynamics and Control (ROMANSY2004), Montreal, QC, Canada, 14–18 June 2004; Volume 126, pp. 83–93.
29. Carricato, M. Fully isotropic four-degrees-of-freedom parallel mechanisms for Schoenflies motion. *Int. J. Robot. Res.* **2005**, *24*, 397–414. [CrossRef]
30. Kong, X.; Gosselin, C. *Type Synthesis of Parallel Mechanisms*; Springer: New York, NY, USA, 2007.
31. Hervé, J.M. Analyse structurelle des mécanismes par groupe des déplacements. *Mech. Mach. Theory* **1978**, *13*, 437–450. [CrossRef]
32. Hervé, J.M.; Sparacino, F. Structural synthesis of parallel robots generating spatial translation. In Proceedings of the 5th International Conference on Advanced Robotics—ICAR, Pisa, Italy, 19–22 June 1991; Volume 1, pp. 808–813.
33. Hervé, J.M. The mathematical group structure of the set of displacements. *Mech. Mach. Theory* **1994**, *29*, 73–81. [CrossRef]
34. Tuttle, E.R. Generation of planar kinematic chains. *Mech. Mach. Theory* **1996**, *31*, 729–748. [CrossRef]
35. Hervé, J.M. The Lie group of rigid body displacements, a fundamental tool for mechanism design. *Mech. Mach. Theory* **1999**, *34*, 719–730. [CrossRef]
36. Angeles, J. The qualitative synthesis of parallel manipulators. *ASME J. Mech. Des.* **2004**, *126*, 617–625. [CrossRef]
37. Hervé, J.M. Uncoupled actuation of pan-tilt wrists. *IEEE Trans. Robot.* **2006**, *22*, 56–64. [CrossRef]
38. Li, Q.; Hervé, J.M.; Ye, W. *Geometric Method for Type Synthesis of Parallel Manipulators*; Springer: Singapore, 2020; ISBN 978-981-13-8754-8. [CrossRef]
39. Gogu, G. *Structural Synthesis of Parallel Robots*; Springer: Dordrecht, The Netherlands, 2008; Volume 1–5.
40. Di Gregorio, R. Analytic determination of workspace and singularities in a parallel pointing system. *J. Robotic Syst.* **2002**, *19*, 37–43. [CrossRef]
41. Carricato, M.; Parenti-Castelli, V. A novel fully decoupled two-degrees-of-freedom parallel wrist. *Int. J. Robot. Res.* **2004**, *23*, 661–667. [CrossRef]
42. Gosselin, C.M.; Caron, F. Two-Degree-of-Freedom Spherical Orienting Device. U.S. Patent 5,966,991, 19 October 1999.
43. Uicker, J.J., Jr.; Pennock, G.R.; Shigley, J.E. *Theory of Machines and Mechanisms*, 6th ed.; CUP: Cambridge, UK, 2023.
44. Trudeau, R.J. *Introduction to Graph Theory*, 2nd ed.; Dover Publications: New York, NY, USA, 1993.
45. Chiang, C.H. *Kinematics of Spherical Mechanisms*; Krieger Publishing Company: Malabar, FL, USA, 2000.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Article

Buffering Performance Analysis of an Ostrich-like Leg Based on a Seven-Link Parallel Mechanism

Daming Nie ¹, Ruilong Du ^{1,*}, Jianguan Tian ¹, Pu Zhang ¹, Fangyan Shen ^{1,*}, Jason Gu ² and Yili Fu ³

- ¹ Interdisciplinary Innovation Research Institute, Zhejiang Lab, Hangzhou 310000, China; niedaming@zhejianglab.com (D.N.); tianjr@zhejianglab.com (J.T.); zhangpu@zhejianglab.com (P.Z.)
- ² Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS B3H 4R2, Canada; jason.gu@dal.ca
- ³ State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150000, China; meylfu@hit.edu.cn
- * Correspondence: duruilong@zju.edu.cn (R.D.); shenfangan@zhejianglab.com (F.S.)

Abstract: As one of the fastest running animals on land, the ostrich's excellent athletic ability benefits from its unique leg structure. Based on the idea of bionics, this paper intends to obtain a kind of robotic leg structure with a similar buffering capacity to that of the ostrich. For this purpose, the structural characteristics of a seven-link parallel mechanism are analyzed firstly, having some specific features similar to ostrich legs, such as the center of mass (COM) located at the root of the leg, a large folding/unfolding ratio, and so on. Then, the kinematic model of the bionic leg is established, and the energy storage of the flexible parts of the leg is investigated. Finally, an impact experiment of the structure onto the ground is carried out to verify the accuracy of the established kinematic model. This paper systematically reveals the nonlinear law of the elasticity of an ostrich-like leg and provides the buffering performance characteristics of the leg in the process of hitting the ground, based on its elastic properties by the kinematic model and the experiment.

Keywords: biomimetic robotics; parallel mechanism; buffering; leaf spring

Citation: Nie, D.; Du, R.; Tian, J.; Zhang, P.; Shen, F.; Gu, J.; Fu, Y. Buffering Performance Analysis of an Ostrich-like Leg Based on a Seven-Link Parallel Mechanism. *Machines* **2022**, *10*, 306. <https://doi.org/10.3390/machines10050306>

Academic Editor: Med Amine Laribi

Received: 23 March 2022

Accepted: 20 April 2022

Published: 25 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The flexible locomotion of humans and pods depends on close cooperation between legs. Compared with terrestrial animals without feet, bipeds move faster, adapt to more complex environments, and have stronger anti-interference abilities [1,2]. At present, many researchers have reported the dynamics of robotic bionic legs. For example, He et al. [3] studied the nonlinear mechanical control of a single-legged jumping robot in the flight stage. By transforming the dynamics into chain form, an exponentially stabilizable control method based on the integral backstepping process was proposed. Komarsofla et al. [4] established a single-leg jumping mechanism performing hopping by transferring linear momentum between the reciprocating mass and its main body. Larin et al. [5] considered the inertia characteristics of single-leg jumping and designed the spatial motion model. The reports presented above all focus on driving mechanisms. As for underactuated robots, Zhang et al. [6] proposed a 3-DOF underactuated leg mechanism and built dynamics formulas to calculate the driving force based on the principle of virtual work. He et al. [7] investigated the locomotion control method of an underactuated jumping robot and proposed the corresponding modeling, motion planning, and control method. In these papers, the elastic properties of the mechanism were rarely analyzed from the perspective of the material characteristics of the elastic parts [8]. In this study, the influence factors of the nonlinear elasticity of the underactuated leg are investigated systematically, including the elasticity of the leaf springs, and the buffering performance is concluded by a kinematic model and experiment.

2. Structural Analysis of Ostrich Leg

As the fastest biped running on land, the ostrich has strong and powerful lower limbs and has the ability for steady, lasting, and high-speed running [9] (Figure 1). The ostrich weighs 100 kg, with a continuous running speed of about 60 km/h, a sprint speed of more than 70 km/h, and can last for about 30 min [10]. Ostriches have such obvious advantages in speed and energy saving, partly because of their unique leg and foot structure [11]. The muscle at the root of the ostrich's thigh accounts for 80% of the weight of the whole lower limb [12], which can reduce the moment of inertia of the ostrich's leg during movement and facilitate the rapid change of its speed. In addition, the length of the ostrich's calf bone is similar to that of the thigh bone, and there are few muscles in the knee and calf. Therefore, the leg has a large folding/unfolding ratio. What is more, the distance between the two legs of an ostrich is four-fifths of its body width; this ratio makes it difficult to fall when the upper body swings from side to side.

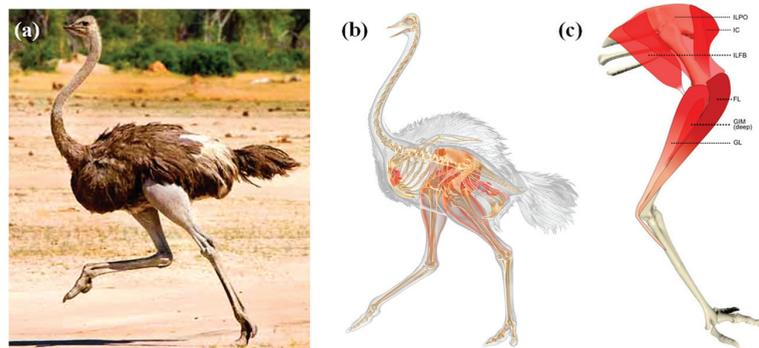


Figure 1. Ostrich and its leg structure: (a) ostrich running, (b) bone and muscle structure, (c) leg structure.

What is more important, the ostrich's legs have good buffering performance because the muscles and tendons on the ostrich's thigh are mixed, and the tendons have good elasticity, which slows down the impact acceleration of the legs when they collide with the ground. This characteristic, in turn, helps the ostrich keep running at a high speed on an uneven road without damaging the legs due to the mild collision with the ground [13].

3. Structural Characteristics of Bionic Legs

The leg is an important part of a robot and plays a very important role in the movement of a legged robot [14]. According to the principle of engineering bionics, the superior performance of the ostrich leg can be applied to the design of a biped robot leg structure. The single leg designed in this experiment is shown in Figure 2. Two motors are set at the hip to simulate the hip and knee joints of an ostrich. The hip motor (M) is mainly used to control the swing of the leg. The knee motor (N) is mainly used to control the expansion and contraction of the leg. In order to imitate the tendon of the thigh, an additional joint is set at point P and constrained with a leaf spring to form the elastic joint, and BQ is also set as a leaf spring to reduce impact acceleration [15]. The center of mass (COM) of the whole leg is located at its root, which is similar to the mass distribution of the ostrich leg. In order to imitate the meniscus buffer structure of the ostrich knee joint, the output link of the driving motor for the toe joint is set as a leaf spring, so that the impact acceleration will be greatly reduced when the leg collides with the ground. In addition, the energy stored in the leaf spring can help the bionic leg take off from the ground during running [16]. In terms of leg length distribution, following the leg size of an ostrich, the length ratio of connecting rod AF to AN is set to 0.75.

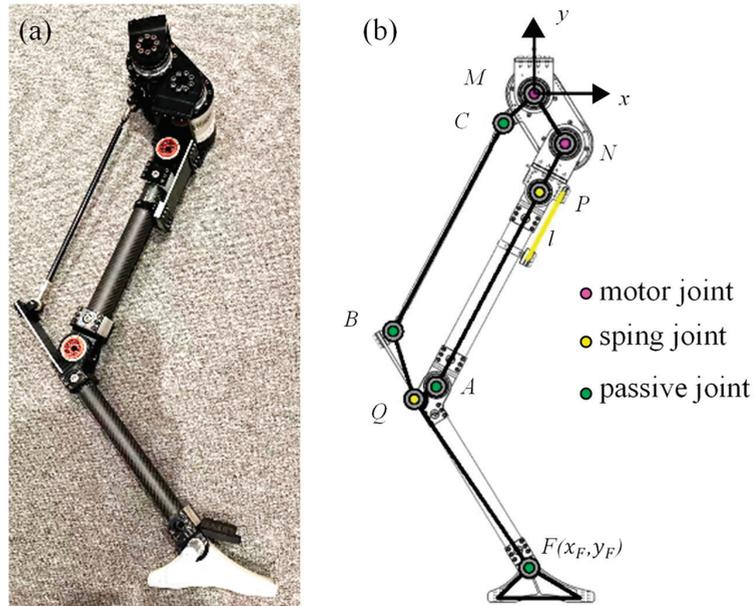


Figure 2. Schematic diagram of the single-leg structure: (a) prototype, (b) schematic diagram of the structure.

Analyzed in essence, the leg belongs to a seven-link parallel structure. The weight of the connecting rod is much lighter than that of the motor, thus it can be ignored in order to facilitate calculation. If the leg stands on the ground, the corresponding static balance relationships are:

$$F_{Mx}^{(MC)} = F_{Mx}^{(MN)} \tag{1}$$

$$F_{Cy}^{(CM)} + F_{Ny}^{(NM)} = G_M + G_N \tag{2}$$

$$F_{My}^{(MN)} \times MN + F_{Mx}^{(MN)} \times MN = 0 \tag{3}$$

$$F_{Bx}^{(BC)} \times BC - F_{By}^{(BC)} \times BC = 0 \tag{4}$$

$$F_{Mx}^{(BC)} \times BQ + F_{My}^{(MC)} \times BQ - K_Q \delta_Q = 0 \tag{5}$$

$$F_{Ax}^{(An)} \times AN - F_{Ay}^{(AN)} \times AN + K_P \delta_P = 0 \tag{6}$$

The letters in the above equations correspond to those in Figure 2b.

The leaf springs of elastic joints are equivalent to cantilever beam under loading, and the relationship between its stiffness and elastic modulus is as follows:

$$F = \frac{3EI}{L^3} Y \tag{7}$$

where F is the bending force, E is the elastic modulus, I is the inertial moment, and L is the length of the cantilever beam.

$$M = FL = K_p \delta = \frac{Ebh^3}{4L} \delta \tag{8}$$

where M is the moment of the cantilever beam, δ is the joint angle, in rad, $b = 50$ mm, $h = 3$ mm considering the influence of connectors at both ends, L is equal to the $0.57l_0$, and l_0 is the length of the leaf spring. Therefore,

$$E = \frac{4LK_p}{bh^3} \tag{9}$$

3.1. Influencing Factors of Equivalent Elastic Modulus of the Ostrich-like Leg

Different from active compliance control [17], this mechanism improves the buffering properties through passive compliance. The material 65 Mn spring-steel is selected for the two leaf springs P and Q, thus the elastic moduli E_p and E_q are both 197 GPa, and the compression force range is from 300 N to 1300 N. This loading range is employed here because the weight of a single bionic leg is 17 kg. Considering the jump gait during rapid movement, the maximum impact force is more than 300 N. As shown in Figure 3, the equivalent elastic modulus (\bar{E}) of the leg, i.e., the change rate of the compressive force to compression, decreases during the process of compression. The data fitting of the equivalent elastic modulus is shown in Formula (10), and the fitting error is 2.679, indicating that the ability of the spring leg to store elastic properties is nonlinear and decreases with the increment of compression.

$$F = 44 + 15x - 0.082x^2 + 1.9 \times 10^{-4}x^3 \tag{10}$$

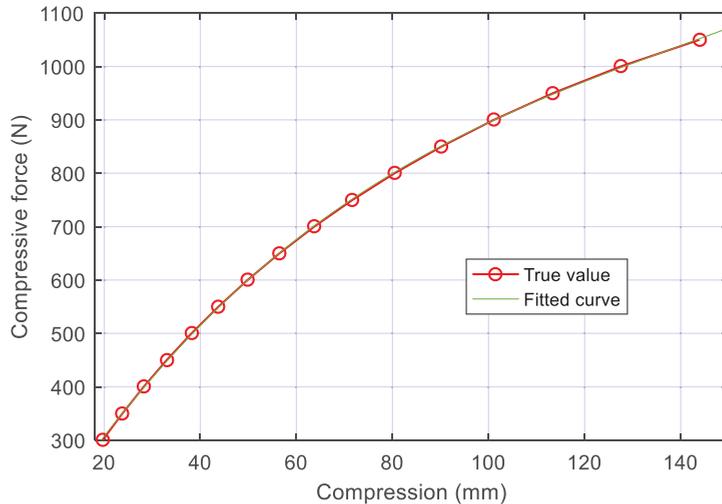


Figure 3. Leg length change under different compressive forces, $E_p = 197$ GPa, $E_q = 197$ GPa, initial length = 800.9 mm.

3.2. Influence of E_p , E_q on the \bar{E} and F_x of the Ostrich-like Leg

The core problem we are concerned about is the influence of the leaf spring as a flexible part on the energy storage and buffering capacity of the structure. Therefore, the effect of the elastic modulus of the P and Q leaf springs on the equivalent elastic modulus (\bar{E}) of the structure is analyzed. When $E_q = 197$ GPa, the initial length equals 800.9 mm and the compressive force F equals 300 N. The change of the equivalent elastic modulus \bar{E} with E_p is shown in Figure 4. The results indicate that the increasing rate of \bar{E} decreases with the increase in E_p . The horizontal swing of the leg first decreases, arriving at 0, and then increases positively, the rate of change decreasing with the increasing E_p .

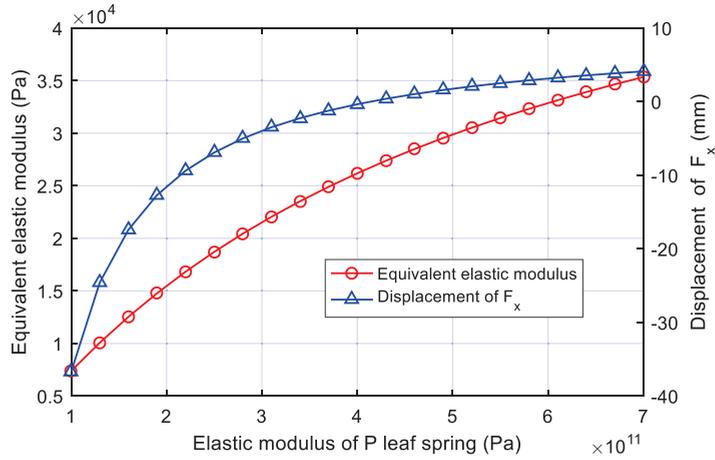


Figure 4. Variation of equivalent elastic modulus \bar{E} and the x -direction displacement of the bionic leg (F_x) with E_p , $E_q = 197$ GPa, initial length 800.9 mm, and the compressive force is 300 N.

While simplifying the structure as a spring, we find that when the compressive force is applied, the F point of the leg (Figure 2) will move in the x -direction (F_x), which indicates that the robot will move in the horizontal plane when squatting. This may cause the change of the robot’s centroid of mass (COM) on the horizontal plane, and it should be corrected by the control algorithm. In order to reduce the adjusting time, it is feasible that the change on the horizontal plane is small during the squatting process of the robot, so the E_p and E_q that makes the F_x close to 0 should be preferred.

In order to analyze the influence of E_q on the equivalent elastic modulus of the spring leg, we set $E_p = 197$ GPa and initial length = 800.9 mm (Figure 5). The value range of E_q is 100–700 GPa and the elastic modulus of few materials exceeds 300 GPa. Here, it is extended to 700 GPa to analyze the trend of change. The results show that both the equivalent elastic modulus and the displacement of the F point of the structure increase with E_q , and the rate of change keeps decreasing.

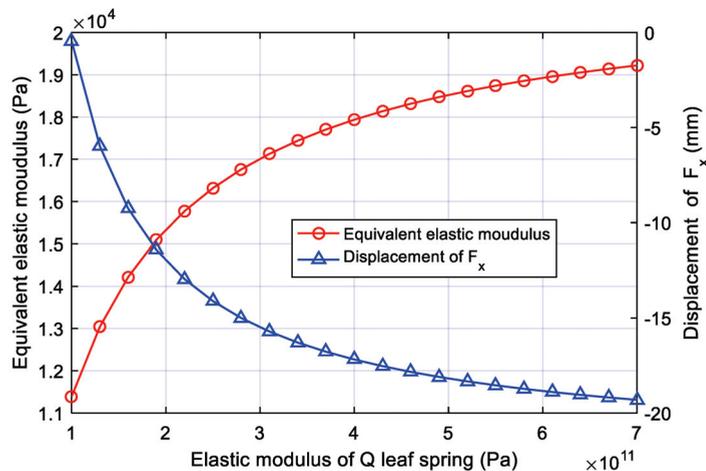


Figure 5. Variation of equivalent elastic modulus and displacement in the x -direction of the bionic leg with E_q , $E_p = 197$ GPa, initial length 800.9 mm, and compressive force 300 N.

When E_p is 197 GPa, the initial length is 800.9 mm, and the compressive force is 300 N, from the perspective of control stability, the greater the elastic modulus of the Q leaf spring, the greater the swinging of the fuselage. Generally, the larger the elastic modulus of the leaf spring, the lower the flexibility of the structure; however, the rule breaks here. The cause is the parallel structure characteristics of bionic legs. If the elastic modulus of the P leaf spring is smaller than that of the Q leaf spring, more deformation is concentrated on the P leaf spring, resulting in an increment in the horizontal displacement of the upper end of the fuselage. If the P and Q elastic moduli are configured in an appropriate proportion, the sloshing on the horizontal plane of the fuselage can be close to 0.

For the underactuated parallel mechanism [18], the swing of the fuselage cannot be fully controlled by the driving motors. Because the flexible parts of the leg will lose their static balance in a short time after being loaded, they inevitably have a process of swinging to consume energy and rebalance. The superposition of this swinging and the active rotation of the motors will reduce the locomotion stability of the robot.

The equivalent elastic modulus can be controlled by the thickness of the leaf spring; specifically, the value of the elastic modulus can be increased by adding the number of leaf springs. Therefore, when the elastic modulus of the leaf spring needs to be increased, the material of the spring does not need to be changed. It is also effective to stack and fix two or more original leaf springs to the connection position of the leg.

Moreover, it should be noted that the influence of the Q leaf spring here is limited. The reason is that the elastic modulus of most materials that can be employed as the leaf springs is below 250 GPa, therefore the displacement of F_x in this range is below 15 mm for the bionic leg with an initial length of 800.9 mm, so it has little influence on the control stability.

In order to verify the universality of the above laws, the variation trend of equivalent elastic modulus \bar{E} with E_p under different E_q is investigated. As shown in Figure 6, the blue curves represent the value of F_x and red curves represent the \bar{E} of the single leg. Among the curves with the same colors (red or blue), the lighter colors represent smaller E_q , and the curves with darker colors correspond to the values on the condition of larger E_q . The results show that the value of \bar{E} increases with E_q under the same E_p condition, and so does the increment rate. With the increase in the value of E_q , the displacement of the F point in the x -direction, F_x , increases, and the rate increases too. The laws presented above are the same as the ones illustrated in Figure 5, indicating that the variation trend is universal.

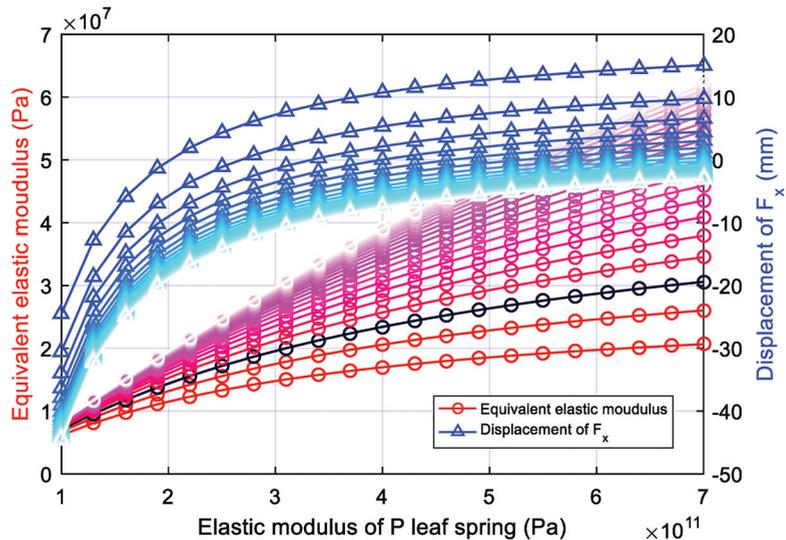


Figure 6. Effect of different E_p and E_q on the equivalent elastic modulus and x -direction displacement (F_x) of the bionic leg.

As can be seen from Figure 6, with appropriate values of E_p and E_q , the horizontal displacement in the x -direction can turn to 0, but this does not mean that the horizontal displacement of the F point of the bionic leg in Figure 2 can remain unchanged in the whole process of locomotion, because the other two influencing factors, namely the initial length and load, will also lead to the change in the displacement of the F point in the x -direction. Therefore the absolute zero horizontal displacement in the process will not be realized, while a deformation range that approximates zero can be selected.

3.3. Influence of Initial Length to \bar{E} , F_x of the Ostrich-like Leg

The initial length is adjusted by the angle at points M and N. The effect of the initial length of the spring leg on \bar{E} , F_x . \bar{E} increases with the initial length, and the variation rate also increases with the initial length. This rule allows us to adjust the length of the leg on the appropriate premise so that the leg can store more energy during impact.

F_x decreases first and then increases with the increase of the initial length. The lowest extreme point is located at about 840 mm (Figure 7). Therefore, the leg length should be set near this value, at which the shaking of the upper body during locomotion turns out to be the least, in theory.

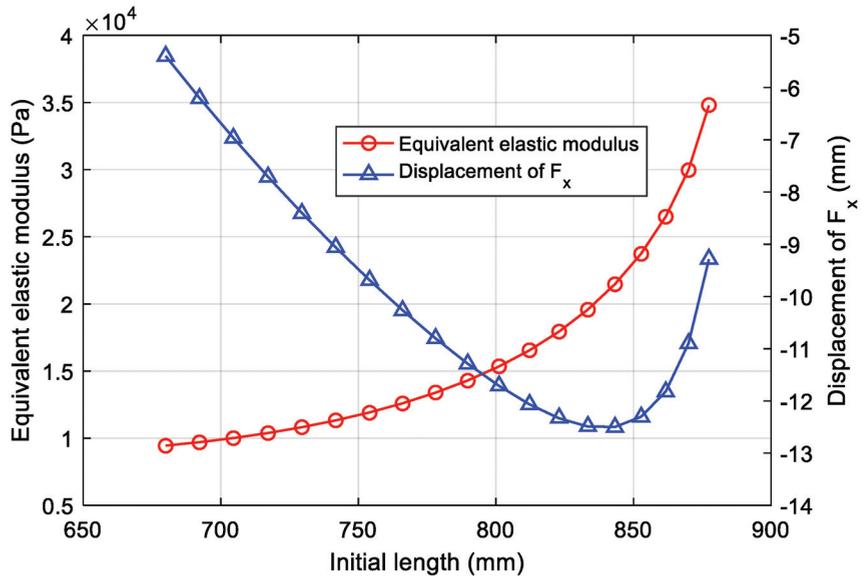


Figure 7. Effect of different initial lengths on the equivalent elastic moduli and the lateral displacement of the bionic leg.

In order to eliminate the interference of randomly selected compressive force on the above conclusions, The variation law of \bar{E} and F_x is obtained under different initial lengths and loads, as shown in Figure 8. \bar{E} increases with the increment of compressive force in a static equilibrium state at the same initial length, and the increment of the elastic modulus increases monotonously. F_x also increases with the compression force, and the increment is the largest when the initial height is about 840 mm, while F_x first decreases and then increases with compressive force when the initial height is about 680 mm.

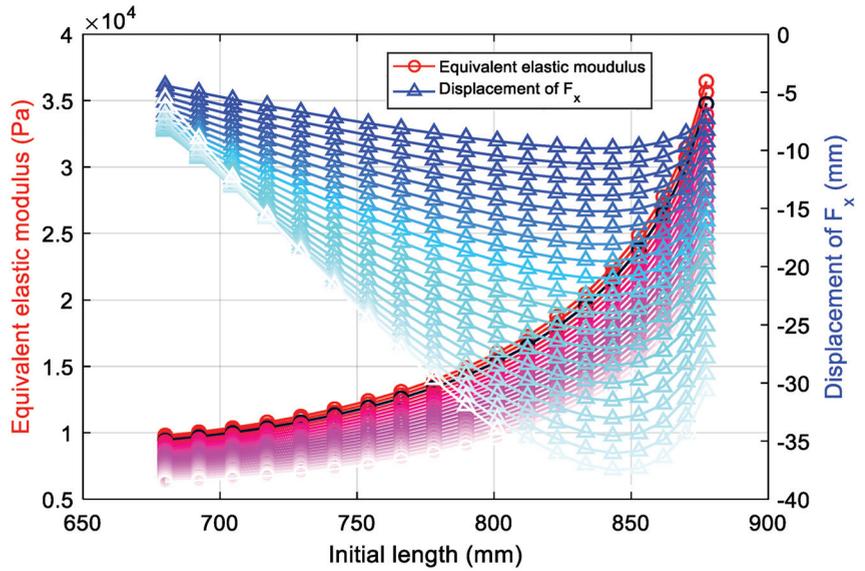


Figure 8. Effect of the \bar{E} and F_x of the bionic leg under different compression forces and initial lengths.

4. Buffering Characteristics of the Bionic Leg

4.1. The Energy Storage of the Leaf Spring

The influence of E_p and E_q on energy storage is reflected in the rotation angle of the leaf spring under compression. The stored energy increases with the increment of the rotation angle. Therefore, the change of energy stored in the leaf spring with compression force is calculated according to Formula (1):

$$k_{(y)} = A_1 + 2A_2y + 3A_3y^2 \tag{11}$$

$$Q = \int k_{(y)} y dy = \frac{1}{2}A_1y^2 + \frac{2}{3}A_2y^3 + \frac{3}{4}A_3y^4 \tag{12}$$

where, $A_1 = 15$, $A_2 = -0.082$, and $A_3 = 1.9 \times 10^{-4}$

$$Q = \int F_{(\delta)} d\delta = \frac{Ebh^3}{4L^2} \cdot \frac{1}{2}\delta^2 = \frac{Ebh^3}{8L^2}\delta^2 = \frac{K}{2L}\delta^2 = \frac{K}{2L}\delta^2 \tag{13}$$

$$E = \frac{4 \times 0.57 \times l_0K}{bh^3} = \frac{4 \times 0.57 \times 130}{50 \times 27}K = 0.22 \times 10^6K \tag{14}$$

$$Q_p = \frac{K_p}{2L}\delta_p^2 = \frac{E_p}{2 \times 102 \times 0.22 \times 10^6}\delta_p^2 \tag{15}$$

$$Q_q = \frac{K_q}{2L}\delta_q^2 = \frac{E_q}{2 \times 110.42 \times 0.22 \times 10^6}\delta_q^2 \tag{16}$$

where the initial length is 800.9 mm and E_p and E_q are both equal to 197 GPa.

The variation law of compressive force to Q_p , Q_q can be calculated and is shown in Figure 9, by calculating δ_p , δ_q with regard to the change of compressive force.

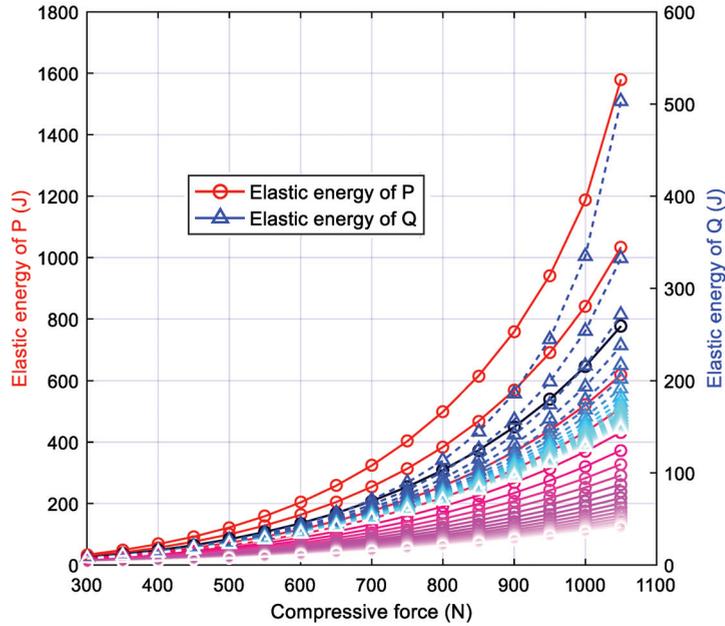


Figure 9. Energy storage of two leaf springs of the bionic leg under different compression forces and E_p .

As shown in Figure 9, the blue curves represent the elastic energy of Q and the red curves represent the elastic energy of P of the single bionic leg. Among the curves with the same colors (red or blue), the lighter colors represent smaller E_p , and the curves with darker colors correspond to the values on the condition of larger E_p . It can be seen from the trend that the energy stored in the P and Q leaf springs increases with the compression force, and the rate is also increasing. Under the same compression force, the stored energy of P and Q increases with the increment of the elastic modulus of the P leaf spring, The increasing rate of P is larger than that of Q.

4.2. Buffering Model Prediction

The buffering performance of the structure is usually evaluated by impact acceleration during the compression of the bionic leg:

$$\frac{1}{2}mv^2 + mgy = \int k_{(y)} y dy \tag{17}$$

where $k_{(y)} = A_1 + 2A_2y + 3A_3y^2$, y is the compression when considering the leg as spring. $E_p = 197$ GPa, $E_q = 197$ GPa, and the initial length is equal to 800.9 mm:

$$\int k_{(y)} y dy = \int (A_1y + 2A_2y^2 + 3A_3y^3) dy = \frac{1}{2}A_1y^2 + \frac{2}{3}A_2y^3 + \frac{3}{4}A_3y^4 \tag{18}$$

where $h = 100$ mm, y is obtained through Newton’s iterative calculation, and the value is 90.14 mm:

$$ma = k_{(y)}y - mg \tag{19}$$

The maximum impact acceleration, a , is equal to 1.57 g.

5. Experimental Process and Analysis

An experiment is designed to test the accuracy of the above kinematic model. The peak value of the acceleration while the leg impacts the ground is measured by the acceleration sensor. The sensor is set at the upper end of the bionic leg, which is buffered by the leaf spring (Figure 10).

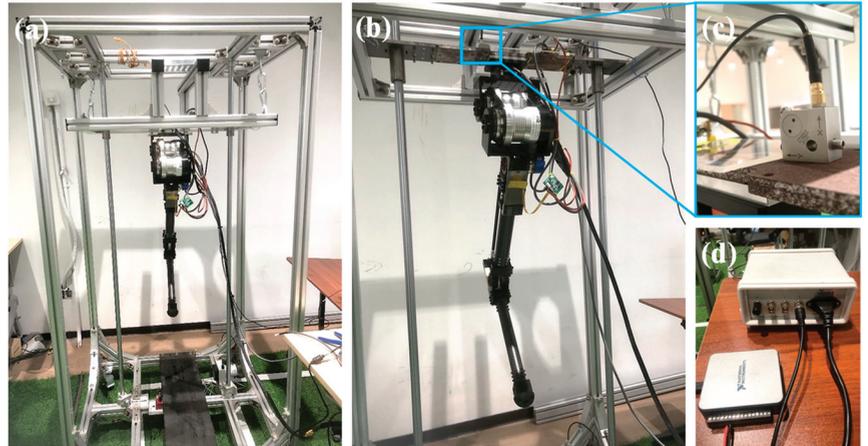


Figure 10. Impact acceleration detection device: (a) the single leg assembled on the test platform, (b) an enlarged view of the single-leg installation, (c) the acceleration sensor, and (d) the power supply and data acquisition system.

Based on the designed model, the single leg was processed and assembled. The maximum torque of the hip and knee driving motor is 60 N·m. The maximum speed is 60 rpm, the rated voltage is 48 V, and Ethernet communication was adopted. The movable end of the bionic leg was mounted on the movable table of the assembled frames. The movable table has one degree of freedom along the vertical direction. The acceleration sensor was fixed on the top surface of a movable table. The impact acceleration measurement system includes a sensor, NI data acquisition system, DC power supply, and a LabVIEW data display terminal. The sensor range is 30 g, the corresponding voltage of 1 g is 0.02 V, and the sampling frequency is set to 10 kHz, which can capture the impact wave.

The motor was powered to maintain its designed behavior after the fixing of the leg. The leg was let down from a height of 10 cm and the impact acceleration was measured at the upper end of the movable table. The data measured by the sensor is illustrated in Figure 11. According to the analysis of the data results, the frequency of the impact wave is about 3 Hz, which is very low due to the buffering of the leaf springs. The measured value of the first wave peak is 0.04 V, indicating that the impact acceleration is about 2 g, which is close to the calculated value. There may be two factors responsible for the prediction error: (1) connection mode, the leaf spring is connected to the fuselage by screws, thus the actual action length of the leaf spring is uncertain; (2) the complex stress state of the leaf spring is simplified as a plane force state in the model, however, it will be subjected to the force along the width direction under actual conditions, which can affect the prediction results of the model to a certain extent.

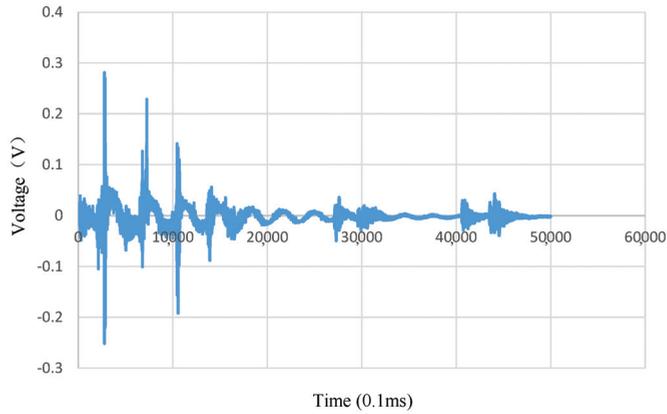


Figure 11. Signal voltage of bionic leg detected by the acceleration sensor.

The subsequent decaying waveform indicates that there are several damped cyclic movements of the bionic leg after the first impact.

The leaf spring is an important flexible part of the mechanism. If the elastic modulus of the leaf spring is extremely large, it would not have an ideal buffering effect, while a too-small elastic modulus would worsen the control accuracy. The mechanical properties of spring steel are most suitable for the requirements of this experiment, however, its determination is high density. Therefore, several fiber-reinforced composites were tried as illustrated in Figure 12, including high-strength carbon fiber-reinforced thermosetting composites (HSCF + TSM), high-strength carbon fiber-reinforced thermoplastic composites (HSCF + TPM), high-strength glass fiber-reinforced thermosetting (HSGF + TSM) and thermoplastic (HSGF + TPM) matrix materials, and medium-strength glass fiber-reinforced thermosetting (MSGF + TSM) and thermoplastic (MSGF + TPM) matrix materials.

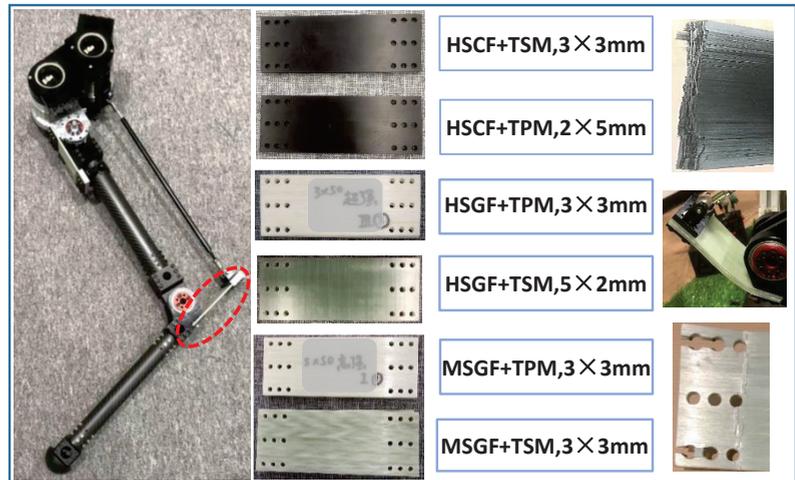


Figure 12. Failure of leaf springs made of different materials.

Various thicknesses of leaf springs were employed for these materials as required because the stiffness and strength of each plate are different. The results indicate that no matter what material of the matrix of the composite is used, the brittle fracture always occurs in carbon fiber reinforced composite, indicating that the high strength and low elongation of carbon fiber make it unsuitable for leaf spring materials.

The toughness of the glass fiber composite plate is much better than that of carbon fiber. Even if the medium-strength glass fiber-reinforced and thermoplastic polymer-matrix (MSGF + TPM) plate breaks at the joint, its property can be further improved by optimizing the structure of the joint. The HSGF + TPM plate is proved to be the most suitable material, better than the spring steel after a comprehensive comparison in strength and weight.

In addition, it is found through experiments that the bending strength of the leaf spring is different even for the same material and total thickness. For example, the bending leaf spring shown in Figure 12 is made of five 2 mm thick stacked plates with a 10 mm total thickness, whose material is HSGF + TPM, and the bionic leg cannot stand on the ground completely after the motor is powered on, which results from over-bending at the Q leaf spring. However, two 5 mm stacked plates with a 10 mm total thickness are qualified for jumping even when made of the same material.

6. Prospect

On the basis of the above exploration, further research on the following aspects is planned:

Firstly, the two legs should be assembled to improve the stability in the process of standing up and squatting and the buffering effect of the leaf spring in boosting gait should be analyzed.

Subsequently, as the leaf spring is the core part of the bionic leg, and its performance directly affects the buffering and jumping ability of the bionic leg, the analysis of the optimal value of the mechanical performance of the leaf spring is necessary for updating the bionic leg.

Finally, further improvements to the structure are needed to reduce the movement of the center of mass on the horizontal plane. The center of mass should be specially investigated and adjusted by controlling the rotation angle of the motors.

7. Conclusions

This paper intends to obtain a kind of robotic leg with a similar buffering capacity to that of the ostrich based on the idea of bionics. The structural and functional characteristics of the ostrich-like leg are investigated and the buffering performance of the seven-link parallel mechanism is analyzed from the perspective of the leaf spring. The feasibility of the kinematic model is verified by experiments. Attempts are made to optimize the material of the leaf spring. The following conclusions could be drawn:

1. For a single leg with an initial length of 800.9 mm and a compressive force of 300 N, the equivalent elastic modulus of the structure (\bar{E}) and the displacement of point F in the x -direction (F_x) also increases with the increase of the elastic moduli of the P and Q leaf springs, and the rate of change decreases with the elastic moduli of the leaf springs.
2. Compared with the ~1000 Hz impact wave frequency of the rigid body, the frequency of this leg is about 3 Hz, indicating that the structure has good buffering performance.
3. The initial height of the bionic leg has a significant impact on the equivalent elastic modulus of the structure and the movement stability of the robot in the horizontal plane. When the initial height is about 850 mm, the stability of the robot is the best.
4. The calculation result of impact acceleration by the kinematic model is close to the measured one, and the error is due to the connection mode and the complex stress form of the leaf spring.
5. Due to the small bending strain of carbon fiber-reinforced leaf spring, it has less ability for energy storage in this structure and is prone to brittle fracture. The material of HSGF + TPM composite has the merits of being lightweight and having good impact resistance. It may be ideal for the spring leaf of a bionic leg.

Author Contributions: D.N.: theme selection and general planning. R.D.: design of the experiment and investigation of research status. J.T.: writing of the paper. P.Z.: implementation and optimization of the experiment. F.S.: formal analysis and check of the paper. J.G.: contact relevant experimental equipment. Y.F.: supervision. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the National Natural Science Foundation of the People's Republic of China, grant number 51905495, and supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LQ22E050024, and China Postdoctoral Science Foundation (No. 2021M692955).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not Applicable.

Acknowledgments: The authors would like to thank Doudou Chen from Shiyanjia Lab (www.shiyanjia.com, accessed on 22 March 2022) for the tests of mechanical properties of the specimens.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wood, B. Four legs good, two legs better. *Nature* **1993**, *363*, 587–588. [CrossRef] [PubMed]
2. Zhang, G.; Ma, S.; Shen, Y. A Motion Planning Approach for Nonprehensile Manipulation and Locomotion Tasks of a Legged Robot. *IEEE Trans. Robot.* **2020**, *36*, 855–874. [CrossRef]
3. He, G.; Geng, Z. Exponentially stabilizing an one-legged hopping robot with non-SLIP model in flight phase. *Mechatronics* **2009**, *19*, 364–374. [CrossRef]
4. Komarsofla, A.K.; Yazdi, E.A.; Eghtesad, M. Dynamic Modeling and Control of a Novel One-Legged Hopping Robot. *Robotica* **2021**, *39*, 1692–1710. [CrossRef]
5. Larin, V.B. A 3D model of one-legged hopping machine. *Int. Appl. Mech.* **2004**, *40*, 583–591. [CrossRef]
6. Zhang, J.Z.; Jin, Z.L.; Zhao, Y.M. Dynamics analysis of leg mechanism of six-legged firefighting robot. *J. Mech. Sci. Technol.* **2018**, *32*, 351–361. [CrossRef]
7. He, G.-P.; Tan, X.-L.; Zhang, X.-H. Modeling, motion planning, and control of one-legged hopping robot actuated by two arms. *Mech. Mach. Theory* **2008**, *43*, 33–49. [CrossRef]
8. Luo, G.; Du, R.; Zhu, S. Design and Dynamic Analysis of a Compliant Leg Configuration towards the Biped Robot's Spring-Like Walking. *J. Intell. Robot. Syst.* **2022**, *104*, 64. [CrossRef]
9. Muysshondt, P.G.G.; Claes, R.; Aerts, P. Quasi-static and dynamic motions of the columellar footplate in ostrich (*Struthio camelus*) measured ex vivo. *Hear. Res.* **2018**, *357*, 10–24. [CrossRef] [PubMed]
10. Schaller, N.U.; Herkner, B.; Villa, R. The intertarsal joint of the ostrich (*Struthio camelus*): Anatomical examination and function of passive structures in locomotion. *J. Anat.* **2009**, *214*, 830–847. [CrossRef] [PubMed]
11. Smith, N.C.; Payne, R.C.; Jespers, K.J. Muscle moment arms of pelvic limb muscles of the ostrich (*Struthio camelus*). *J. Anat.* **2007**, *211*, 313–324. [CrossRef] [PubMed]
12. Ramesh, P.; Sundaresan, S.S.; Shobana, N. Structural studies of hemoglobin from two flightless birds, ostrich and turkey: Insights into their differing oxygen-binding properties. *Acta Crystallogr. D* **2021**, *77*, 690–702. [CrossRef] [PubMed]
13. Smith, N.C.; Wilson, A.M.; Jespers, K.J. Muscle architecture and functional anatomy of the pelvic limb of the ostrich (*Struthio camelus*). *J. Anat.* **2006**, *209*, 765–779. [CrossRef] [PubMed]
14. Du, R.; Song, S.; Yuan, H. Discussion on the Stiffness of the Drive Chain in the Legs of Biped Robots. *Actuators* **2022**, *11*, 79. [CrossRef]
15. Rodino, S.; Curcio, E.M.; Di Bella, A. Design, Simulation, and Preliminary Validation of a Four-Legged Robot. *Machines* **2020**, *8*, 82. [CrossRef]
16. Wang, F.T.; Li, C.B.; Niu, S.H. Design and Analysis of a Spherical Robot with Rolling and Jumping Modes for Deep Space Exploration. *Machines* **2022**, *10*, 126. [CrossRef]
17. Fabrègue, D.; Mouawad, B.; Hutchinson, C.R. Enhanced recovery and recrystallization of metals due to an applied current. *Scr. Mater.* **2014**, *92*, 3–6. [CrossRef]
18. Pappalardo, C.M.; Guida, D. Forward and Inverse Dynamics of a Unicycle-Like Mobile Robot. *Machines* **2019**, *7*, 5. [CrossRef]

Article

Patient-Specific Stent Fabrication Using a Seven-Degree-of-Freedom Additive Manufacturing System

John M. Huss ¹, Malachi Lehman ² and Arthur G. Erdman ^{1,*}

¹ Department of Mechanical Engineering, University of Minnesota, 111 Church Street SE, Minneapolis, MN 55455, USA

² Department of Biomedical Engineering, The Pennsylvania State University, 513 Shortlidge Rd, University Park, PA 16802, USA

* Correspondence: agerdman@umn.edu

Abstract: With advances in additive manufacturing technologies, the creation of medical devices which are tailored to the geometry of a patient's unique anatomy is becoming more feasible. The following paper details the capabilities of a seven-degree-of-freedom fused filament deposition modeling system which enables a wide variety of user-control over previously restricted parameters, such as nozzle angle, print bed rotation, and print bed tilt. The unique capabilities of this system will be showcased through the production of a patient-specific tracheal stent using three different methods: segmented overmolding, transverse rastering, and longitudinal rastering. The resulting opportunities and time savings demonstrated by the prints will provide a case for greater implementation of seven-degree-of-freedom manufacturing technologies.

Keywords: seven-degree-of-freedom; additive manufacturing; overmolding; tracheal stent

Citation: Huss, J.M.; Lehman, M.; Erdman, A.G. Patient-Specific Stent Fabrication Using a Seven-Degree-of-Freedom Additive Manufacturing System. *Machines* **2022**, *10*, 1144. <https://doi.org/10.3390/machines10121144>

Academic Editor: Raffaele Di Gregorio

Received: 21 October 2022

Accepted: 28 November 2022

Published: 1 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Additive Manufacturing for Patient-Specific Medical Devices

Due to advantages such as streamlined preoperative planning, device cost-reduction, and improved clinical outcomes, patient-specific design within the medical devices industry has become increasingly popular [1]. Researchers have found that intraoperative computer technologies for patient-specific device planning are being utilized at a higher rate, while new advantages of patient-specific designs are being demonstrated across a wide variety of medical fields [2]. As planning and design is being improved, there remains a demand for manufacturing techniques which can seamlessly and accurately fabricate patient-specific devices according to medical tolerances. A potential solution exists in additive manufacturing, because it allows for computer-assisted designs to be quickly generated with physical material.

Additive manufacturing techniques such as selective laser sintering (SLS), stereolithography (SLA), and fused deposition modeling (FDM), are currently being used to meet a wide subset of clinical needs. For example, many orthopedics companies are using SLS to create high resolution, small-scale, porous scaffolds, bone plates, and implants [3,4]. Furthermore, SLS can generate fine surface details, overhangs, and ductal structures that might be applied to patient-specific design [5]. The tolerances, specificity, and cost of SLS in many cases make it optimal compared to its manufacturing counterparts [6], and has given rise to greater commercialization of SLS-manufactured medical products in the US [7].

SLA and FDM additive manufacturing techniques provide additional utility due to their cost-effectiveness and rapid realization of geometries. Both SLA and FDM are used in rapid prototyping, construction of anatomical models, and manufacturing of medical devices such as stents, shunting tubes, surgical tools, and other simple static forms [8]. They have become such a universal standard in translating raw ideas into testable prototypes, that some products of SLA and FDM manufacturing have even been developed for clinical use,

including drug-delivery systems, tissue grafts, and tissue scaffolds [9,10]. SLA techniques are usually able to produce higher resolution models compared to FDM, while FDM systems are usually cheaper, easier-to-use, and can access a wider array of rigid and flexible deposition materials [11]. Both tools have been highly beneficial to the patient-specific device design space, particularly due to their cost-effectiveness in producing anatomical molds. Finally, creative modifications to the above techniques have been employed to solve complications associated with radial or highly non-uniform geometries. A six degree-of-freedom FDM printing system has been created wherein a filament deposition nozzle mounted on a serial motor arm allows the user to access more surfaces of a workpiece [12]. Similarly radial FDM systems which are capable of depositing along curved or cylindrical print surfaces have also been formulated [13]. Finally, a seven-axis serial robotic additive manufacturing system has been suggested by the large additive manufacturing technologies company, Stratasys [14]. These technologies represent what may be the future standard in patient-specific device manufacturing, providing more tools to address the unique geometric challenges of human anatomy.

1.2. Limitations of Current Additive Manufacturing Methods

While the current scope of technologies fulfills many specific commercial and clinical needs, several limitations prevent any singular method from expanding beyond its set of established use-cases. SLS manufacturing is limited in large part due to the post-processing of the workpiece, wherein excess powder must be removed from porous or ductal spaces [15], limiting the creation of certain geometries. In addition, some toolpaths of an SLS print are restricted or entirely unfeasible due to the thermal deformation of the part during sintering [16,17]. Even allowable paths may sometimes require scaffold networks to be constructed if there is large separation between two branches of the workpiece in 3D-space. Finally, the segmented nature of the sintering process, wherein material is fused in a series of layers, can produce variations in molecular grain structure which introduce undesirable macro-scale anisotropy to printed structures [18].

SLA printers are similarly limited in the geometries they can create. Because parts are being constructed from a photocured resin bath, there is no solid or dense surrounding material to support larger structures as they are being printed [19]. Therefore, as with SLS, scaffolds must frequently be built to prevent larger prints from shifting or collapsing. The post-processing removal of these scaffolds often blemishes the surface of the part, or entirely limits some hollow structures from being created. SLA also only gives partial user-influence over the global structural properties of a part, limited to part orientation on a print-bed, layer-height, and curing methods [20].

FDM is perhaps the most limited with regard to patient-specific design, with primary drawbacks stemming from the method by which FDM adds material to a workpiece. Strands of filament are heated, extruded, and deposited along a toolpath to build a geometry, giving rise to characteristic layer artifacts created by the orientation of the layered material [21]. These artifacts pose potential problems for devices that must be manufactured with tight surface-finish tolerances and are difficult to mitigate without post-processing [22]. In addition, typical FDM methods cannot realize complex features or enclosed hollow structures without making significant sacrifices to accuracy, surface finish, or strength [23,24]. Similar to SLS and SLA methods, these types of FDM prints require scaffolds to support toolpath layers suspended in 3D space, meaning hollow, branching, or overhanging structures are very difficult to produce.

Fundamentally, these limitations can be tied to a universal constraint among systems. Current methods rely on fixed cartesian or radial coordinate systems to drive the deposition or sintering toolpath, thus limiting user-control over directional mechanical properties and demanding some form of post-processing which reduces the accuracy of the final structure. Without universal influence over the toolpath, the scope in which each method can be applied must be narrowed to a particular subset of geometries and structures. Such limitations inhibit patient-specific design, as they cannot account for the dynamic demands

that each unique person's anatomy might present. This paper will explore the advantages of a new seven-axis fused filament deposition 3D printer. This system will allow for enhanced user-control over the deposition toolpath and layer orientation of prints. These unique advantages will be showcased through a method and workflow for creating a tracheal stent from patient-specific scan data.

1.3. The Seven-Degree-Of-Freedom Additive Manufacturing System

The FDM system constructed for this research is able to reposition parts as they are built in order to manipulate the orientation of deposited material and give access to exposed surfaces of the workpiece. Specifically, the base of such a system, acting as a traditional build plate, can tilt and rotate independently of the filament nozzle, which itself can tilt laterally and transversely, as well as move by way of a gantry in the three traditional X, Y, Z directions. This system, shown in Figure 1 [25] and demonstrated in Videos S1–S4, has been designed and built at the University of Minnesota for a mechanical engineering PhD thesis project by Dr. John Huss. The inverse kinematics and other details of the robotic system can be found in [26].

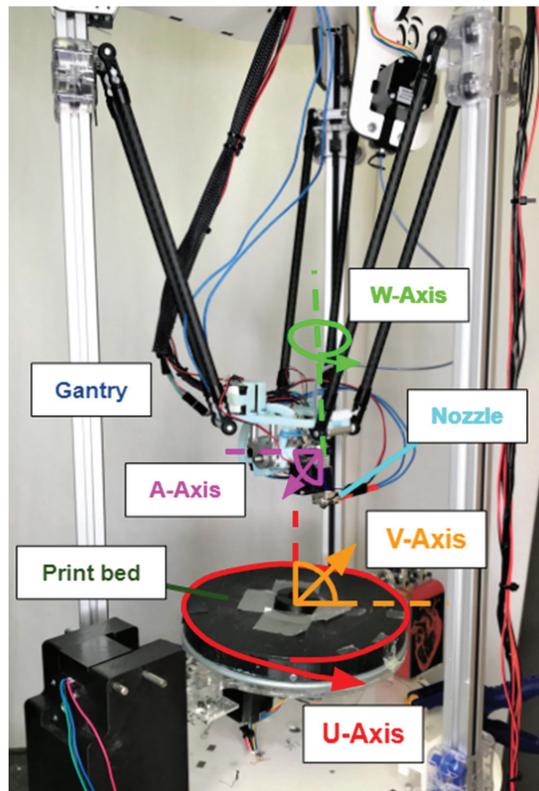


Figure 1. Seven-degree-of-freedom 3D-printing system designed by Dr. John Huss. The build plate rotates along the U-axis and tilts along the V-axis. The filament nozzle moves in the X, Y, and Z directions by way of the gantry, tilts laterally along the W-axis, and transversely along the A-axis.

When additively manufacturing devices for medical applications, one can imagine scenarios where devices must contain widely spaced, ductal, weblike, or highly non-uniform geometries with predictable structural properties. Oropharyngeal airways, heart valves, and medical stents are examples where the devices must take on highly non-uniform geometries with tightly regulated material and structural properties [27], limiting current

additive manufacturing methods from impacting their production in a significant way. These applications are particularly well suited to the seven-axis system because of its ability to fabricate complicated and non-uniform shapes. Parts require little, if any, sacrificial support structure because newly deposited material can be supported by the model itself. This greatly reduces the amount of support material used and discarded as waste and provides more flexibility while printing part features, giving the user increased control over local deposition density or other properties. This self-supporting capability is combined with a five-degree-of-freedom print head to allow for changing layer line directions as well. Finally, the general nature of this printer allows it to print on surfaces that are more complex than two-dimensional build trays and part layers [28]. This leads to the possibility of printing onto existing additively manufactured parts, or more interestingly, objects that are not additively manufactured, such as machined parts, or anatomical structures.

To fully demonstrate and evaluate the benefits of the seven-axis system, a series of methods was devised to manufacture both a patient-specific anatomy model of a tracheal airway, and an airway-supporting stent prototype to adhere to its outer surface. A future tracheal stent device based on the concept introduced here, could potentially be used in patients with a stenotic, or narrowed, tracheal airway [29]. The stent example demonstrates the system's ability to both control material deposition properties and produce geometries with open spaces. The airway anatomy mold will demonstrate the system's ability to follow curvilinear toolpaths along cylindrical surfaces according to patient-specific medical data. Finally, these combined models will demonstrate the system's ability to enable device processing parallelization and streamlining.

The methods were chosen to highlight the unique capabilities afforded by the seven-axis system, rather than the clinical efficacy of the product. The produced stent is meant to be a prototyping example rather than a clinically viable medical device. Therefore, considerations that would be important in a clinical use, such as the biocompatibility of the filament used, the deployment of the stent itself, and mechanical integrity of the stent during medical use, were not showcased. Further directions for the system's evaluation and use will be addressed below.

2. Materials and Methods

2.1. Overview

The methodology introduced in this paper follows a sequence summarized in Figure 2. First a series of mathematical and programming steps was taken to transform a conventional standard triangle language (STL) printing file of a patient trachea airway scan to a stent tool path compatible with the seven-axis printing system. The trachea airway was printed in three different ways: segmented, transversely rastered, and longitudinally rastered, to highlight the system's key advantages. Finally, a patient specific stent was printed onto the surfaces of the various trachea models.

In summary, these steps create a replicable process which allows for an STL file of a patient airway to be transformed into a physically printed stent device formed directly to its anatomy. The individual steps are covered in more detail below.

2.2. Trachea Model Generation

The first step is to transform a magnetic resonance imaging (MRI) scan from a patient into an STL file which captures the geometry of a specific organ. Typically, to obtain a 3-dimensional STL mesh file of a patient's anatomy, a process called segmentation is done. In segmentation, a computed tomography scan or a stack of MRI images are opened in a software which allows the user to manually or automatically identify and separate areas of tissue in each image [30]. For the purposes of this paper, a patient-specific STL model of a trachea was downloaded from an online repository of anatomical 3D files and segmented via the Mimics version 19.0 [31] software package. The tissue model of the trachea was then inverted using a Boolean operation to recreate a model of the tracheal lumen rather than the tracheal tissue itself. A locking base was added to the bottom of the section of trachea to

allow for mating with a matching receptacle located in the center of the seven-axis printer build plate, shown below in Figure 3a,b, respectively. The locking receptacle in Figure 3a was printed directly onto the center of the build plate to ensure it was located in a known position. Locking the trachea model into the printer using this mechanism ensures it does not move while material is added and allows for additional trachea molds to be swapped between prints.

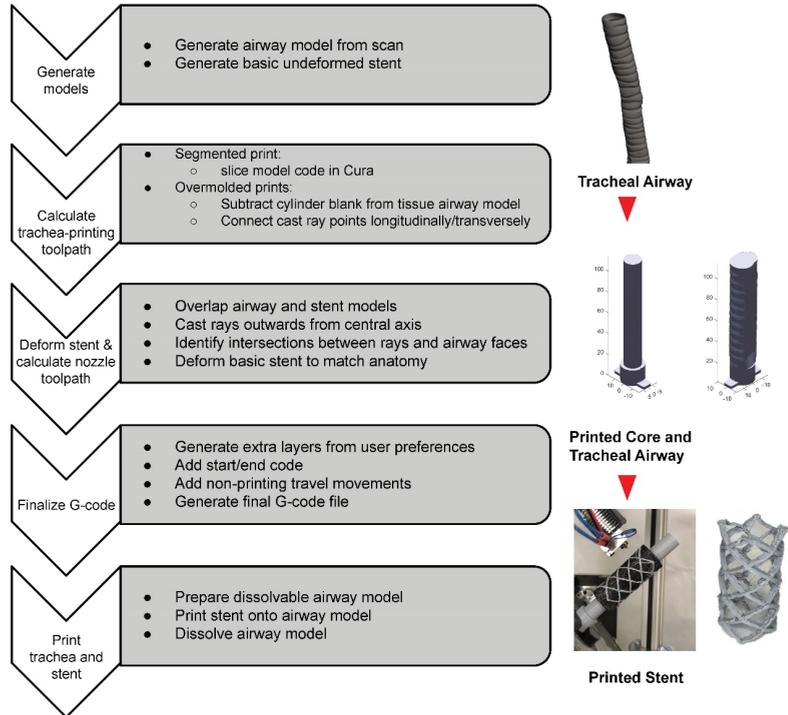


Figure 2. Flowchart describing the patient stent manufacturing process.

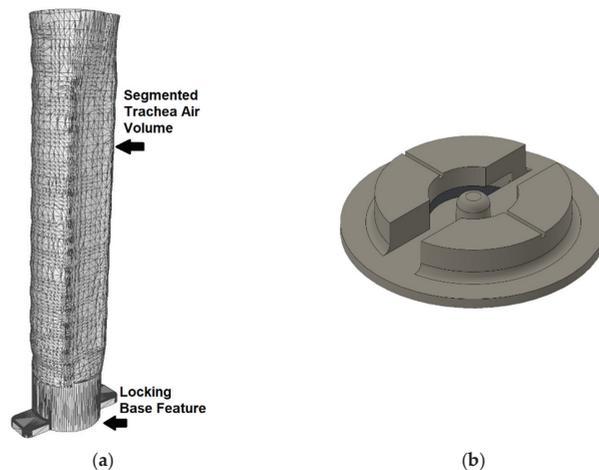


Figure 3. (a) The trachea mesh with added base for registration and locking. (b) The matching 3D-printed locking receptacle.

2.3. Toolpath Generation for the Trachea Airway Print

Three different methods were utilized to convert the STL file into a printed tracheal airway model: traditional FDM, transverse overmolding, and longitudinal overmolding. Beginning with the traditional FDM method, the STL file was imported to the Ultimaker Cura v3.3.0 [32] slicing software which turns the geometry into a conventional three-axis G-code. The G-code was then loaded into a Creality3D CR-10 traditional three-axis printer [33] which executed a segmented, layer-by-layer print of the tracheal airway. This segmented print acts as a baseline, while the transverse and longitudinal overmolding methods demonstrate the seven-axis system's unique capabilities.

The transverse and longitudinal overmolding methods highlight potential process parallelization using the seven-axis system. Given that much of the tracheal airway is made up of filler material, an "anatomy blank" can be created that approximates invariant features such as the inner core of the model. This blank is subtracted from the patient specific details, like the variable trachea tissue surface, which can be added back on later using the seven-axis system, saving time and resources. For a trachea, this blank can be a cylinder with a diameter slightly smaller than the expected minimum diameter of the airway, shown in Figure 4.

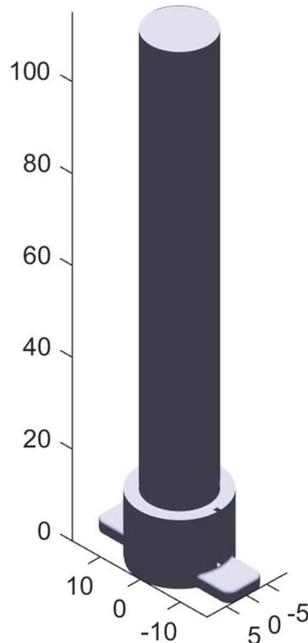


Figure 4. This blank represents the bulk air volume within a patient's trachea. The patient specific model will be added on the surface of the blank. Units in mm.

From this point, the seven-axis system can print on the surface of the blank using the transverse and longitudinal rastering methods. The transverse method adds material by rotating the model underneath the nozzle while slowly maneuvering from the bottom to the top of the model. The longitudinal method deposits material by following along the long axis of the blank, rotating a user-specified amount of 2.5 degrees, and then moving back in the other direction while depositing additional rows. Each of these methods requires modification of the STL file to obtain a toolpath which can be interpreted by the seven-axis system.

This toolpath was created through the following process. First, candidate toolpath points were generated such that they formed an even distribution of points within a hollow

cylindrical volume, with an inner radius equal to the radius of the blank and an outer radius slightly larger than the maximum radius of the patient's airway. The points were chosen based on the parameters in Table 1 and are shown (with many points removed for clarity) in Figure 5.

Table 1. Slicing parameters used in the final toolpathing.

Parameter	Value	Description
Ri	7.25 mm	Radius of the blank
Length	70 mm	Length of section to print
Longitudinal spacing	0.5 mm	Vertical spacing of points
Angular Spacing	2.5°	Angular spacing of points
Layers	30	Max number of layers to generate
Offset	35 mm	Distance from base to start
Retraction	1 mm	Nozzle retraction during movement

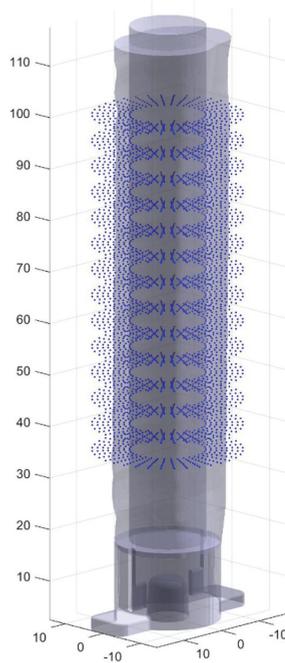


Figure 5. Points are generated as candidates for the final toolpath.

Rays were then cast from the central axis of the blank through the candidate points surrounding the blank. Using a ray/triangle intersection algorithm each point was determined to be inside the final patient specific model and therefore kept as a final toolpath point, or outside the model and rejected. A given point was considered inside the model if its distance from the central Z axis (the norm of its (X,Y) coordinate) was less than the norm of the triangle/ray intersection point of the ray passing through the point. The final selection of points were then connected together to create a toolpath, which is where the two methods diverge. Longitudinal toolpathing connects the points along the long axis of the airway first in a series of vertical lines, and transverse toolpathing connects them around the blank into a series of rings. While ultimately both toolpaths cover the same volume, different axes are primarily utilized. Primarily (X,Y,Z) linear motion is used in the transverse toolpath, while more rotation around the center of the part, defined from here as the U axis of rotation (see Figure 1), is used in the transverse toolpath. The final step was to

connect the often disconnected printing segments (since many candidate points outside of the final model were removed) with nonprinting retracted moves above the surface of the object. The resulting toolpaths are shown below in Figure 6, again with many points removed for clarity.

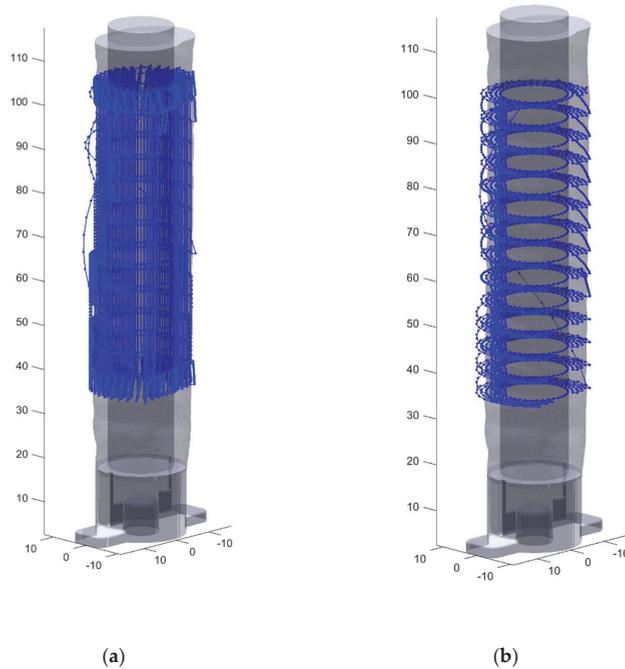


Figure 6. Toolpaths generated for stent print (a) Simplified view of the longitudinal toolpath generation (b) Simplified view of the transverse toolpath generation.

Both of these developmental processes, longitudinal and transverse rastering, result in completely solid layers rather than the traditional shell and infill style typical of FDM, and an unfortunate loss of much of the fine surface detail. Once a toolpath was created, a blank can be loaded into the printer and a file can be run to create the final trachea model out of a dissolvable PVA material. The meshes were compared to see exactly how their accuracy compared to standard FDM printing.

2.4. Stent Model Generation and Transform

The next step is to generate the toolpath for the patient specific stent based on the trachea model. This process was initiated using a basic stent model consisting of a series of eight lines forming a helix pattern. The helix was given a 40 mm Z axis offset from the printer bed to allow the printer nozzle to have adequate access to the side of the trachea model without colliding with the locking base features. A script was created in MATLAB R2018b (MATLAB 9.5) [34] to both generate this basic stent design and import the STL file faces and vertices. The sample stent design is shown below in Figure 7a. After aligning the coordinate systems of the trachea model and the proposed stent design, seen below in Figure 7b, this script determines how to deform the basic stent to match the patient model.

The script accomplishes this by casting hundreds of rays from the central axis outwards through each of the stent struts and through the surface of the trachea model. The hundreds of rays are shown extending through the airway STL in Figure 8a. The rays were programmed to be spaced 0.05 mm apart, so they appear as a helical plane. The trachea STL model created had 25,530 faces and 11,269 vertices. The intersection points of the cast rays and the trachea mesh faces were finally determined using a ray/triangle

intersection algorithm first developed by Moller and Trumbore [35]. The algorithm completes a search by examining all rays in order and checking each ray against all triangles. Shown in Figure 8a, this model contains 25,530 faces, which were cast through by 8000 rays, resulting in 204,240,000 calculations if each ray intersects the last triangular face checked. In practice, this number is substantially lower because once a ray is found to intersect a face, the algorithm moves onto the next ray. Once the intersection points of each ray were determined, the stent radius was deformed to match each of the intersecting points on the airway model, plus a small 0.1 mm offset so that the stent is on top of the surface, shown below in Figure 8b. Extra layers were then added by repeating the patient specific stent toolpath at the specified layer height to give the stent the desired thickness. Using at least three layers at 0.15 mm thickness provided a stent that was strong enough to survive the post processing. Ten layers were used for most testing and images shown.

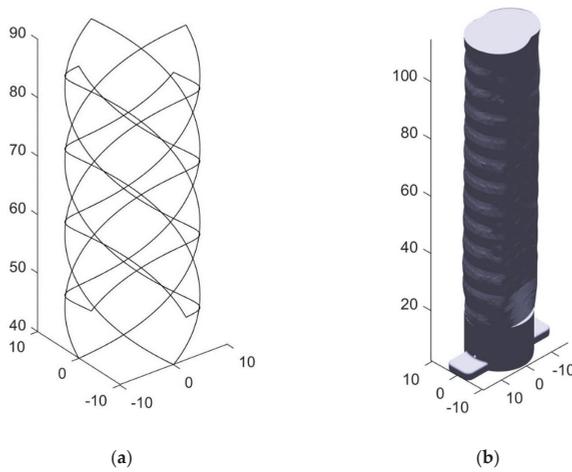


Figure 7. (a) The untransformed stent model composed of eight helical struts, four in each direction (b) The stent is overlaid onto this trachea model.

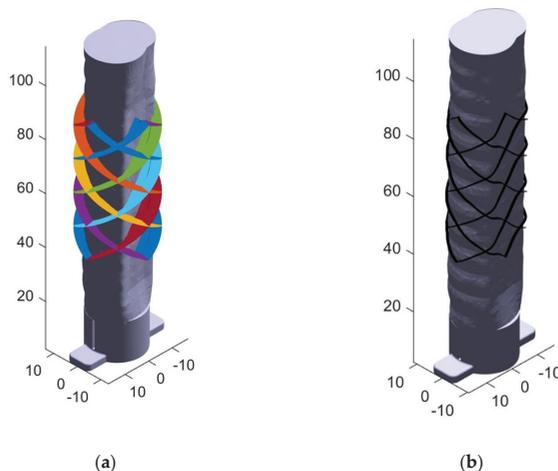


Figure 8. (a) Many rays are cast outwards through the trachea walls from the central axis. Each color represents a different pass of the print head in the final toolpath. (b) The intersection points are used to deform the basic stent struts and extra layers are added as specified by the user.

2.5. Nozzle Angle Calculation

With a tool path generated, the printer nozzle position and angle could be calculated. Referring to Figure 9, the nozzle must tilt 90 degrees sideways relative to the build plate and to access and print on the airway model from the side. The bed also must be tilted on the V axis, that is, the axis which is manipulated according to the reference X axis of the ground frame (see Figure 9), at 45 degrees so that the airway model will be below the extruded plastic. The trachea model will then rotate on the U axis through the center of the build plate and the nozzle will move axially along its length. For the best print quality it is imperative to keep the nozzle as perpendicular to the surface as possible, matching the contours of the airway.

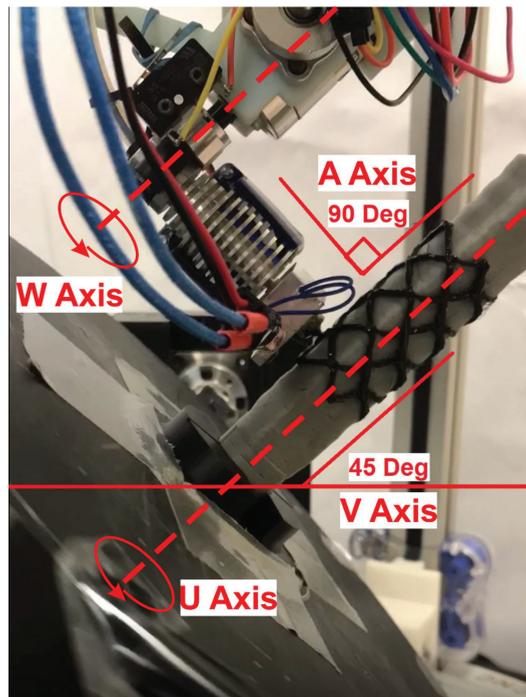


Figure 9. The stent is printed onto the patient's trachea model from the side. The object frame rotates with the U axis, while the V frame is static and tilted 45 degrees.

To communicate the original toolpath to the seven-axis system, a transform was performed using two separate reference frames: the rotating object frame (centered at the U-axis) and the stationary V frame. X , Y , and Z were selected to represent coordinates within the object frame, which rotates and tilts on the print bed with respect to the ground frame, and X_v , Y_v , Z_v were selected to represent coordinates in the stationary V-frame, which only tilts with the V axis of the tilting bed. U was chosen to represent the current angle of the continuous rotation axis through the center of the bed, and V_z is the constant 5 mm vertical offset between the reference frames due to the construction of the printer. Shown in Equation (1), this transform results in a series of coordinates which map the movement of the nozzle system.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(U) & -\sin(U) & 0 & 0 \\ \sin(U) & \cos(U) & 0 & 0 \\ 0 & 0 & 1 & V_z \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{bmatrix} \quad (1)$$

As stated, it is also imperative that the nozzle remain normal to the surface of the airway model to preserve print quality and properly extrude material. To accomplish this, the nozzle compensation angle, W , must be calculated for each point on the stent. Figure 10 below shows how the coordinate system rotates around the U -axis as the printing process progresses and what the W angle represents. The U -axis is a rotational axis that is congruent with the linear Z axis of the object frame. The W axis is measured in reference to the V frame and does not rotate with the object.

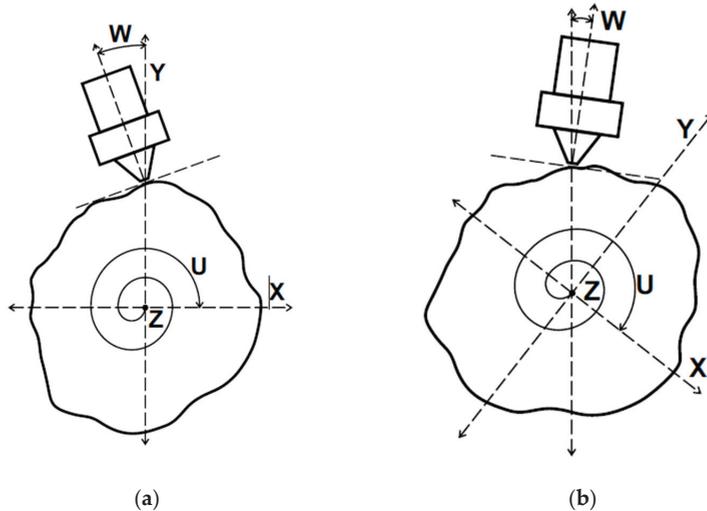


Figure 10. (a) The trachea model viewed end-on during stent printing. (b) The object frame and model rotate while the nozzle follows the surface contours.

The W angle of any given point is a measure of the slope in degrees at that particular point on the airway model. Since this model is made of discrete points, the W angle at any point n was determined by the arctangent of the line between the current point n and the previous point $n - 1$ relative to a flat reference line. The slope equation must also take into account the round surface, and handle angular positions starting under 360 degrees and wrapping around 0 by using the modulo operator. Equation (2) calculates W according to these relations.

$$W_n = \text{rad2deg}(\text{mod}(a \tan 2(Y_n - Y_{n-1}, X_n - X_{n-1}) - a \tan 2(Y_n, X_n), \pi) - \pi/2) \quad (2)$$

The resulting W_n angle produced by the function was then filtered to produce smooth continuous motion of the W axis using a ten-point moving average filter. However, since a filter introduces undesired phase delay into the new W angle data, the data was passed back through the filter in reverse to remove the phase delay using zero-phase digital filtering. Performing this filter operation twice, once forward and once reversed, doubles the effective order of the chosen filter transfer function and squares the magnitude of the input transfer function. The ten-point moving average transfer function was chosen with these effects in mind. The results of the filtering process are shown below in Figure 11. Each line in the plot represents one of the eight supporting struts of the stent. In the figure below the left end of the x -axis is the portion of the stent closest to the printer bed beginning at 40 mm from the surface, and the right side is the highest portion of the stent ending at 90 mm from the bed surface.

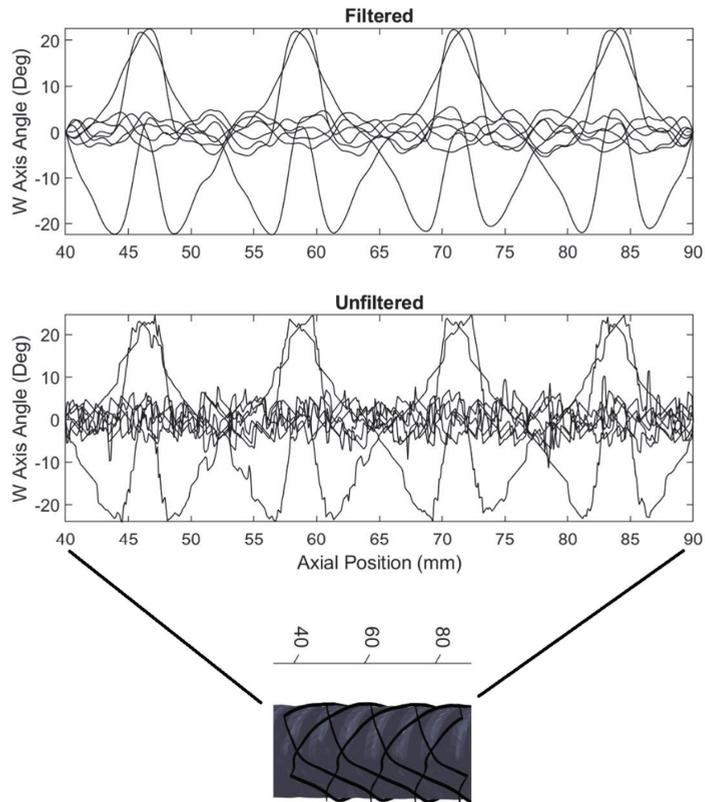


Figure 11. A moving average filter is used to smooth the W axis motion as the stent prints. Each line in the plot represents one of the eight stent struts.

2.6. G-Code Finalization

With the printing movements computed, the final toolpath can be calculated. A set of various commands were created to form the start and end G-code including heating, feeds and speeds settings, homing, and a pause feature that allows time to connect the airway model upon which the stent is printed to the locking receptacle after homing is complete. Non-printing travel commands were added next, between each printing move. These commands retract the nozzle and filament away from the surface, interpolate extra coordinates between the end of one printing segment and the start of another, and then un-retract the nozzle back into place on the model surface at the start of the next printing move. Once all printing moves have been completed, a standard end code is executed, turning off the heaters and fans, and pausing for the completed stent to be removed from the system. The toolpath is shown in several forms in Figure 12.

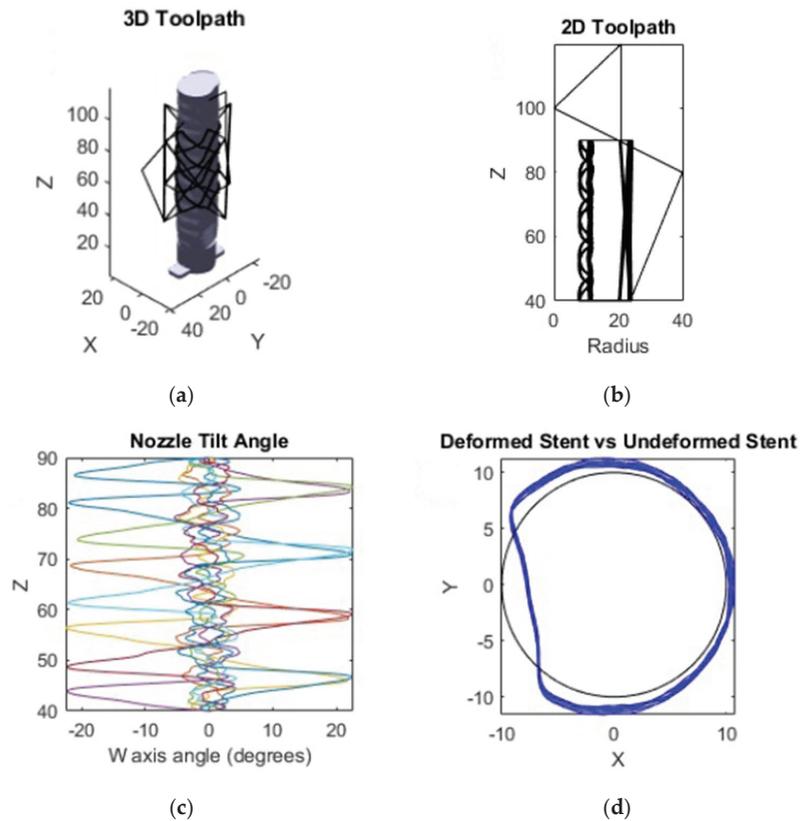


Figure 12. (a) The final toolpath in the object frame. (b) The toolpath in the non-rotating V frame. (c) The W axis compensation angle as the stent prints, with each color denoting a single radial pass of the extruder. (d) The transformed stent vs. untransformed base stent as seen from above.

2.7. Printing of the Stent

The stent printing G-code file was executed in the same way any other G-code file would, except that it additively manufactured an object onto an existing geometrically complex object by following the toolpath generated earlier. In this case, the existing object was a dissolvable PVA printed airway model that can be dissolved after the stent is created. The trachea model rotates underneath the printer nozzle, which is oriented perpendicular to the surface of the airway as seen in Figure 9. Due to the complexity of a seven axis system, there are limitations to the range of positions that are calibrated. Staying closer to the neutral position results in a more accurate print because the printer is calibrated in the neutral position. As much trachea material as possible is removed by hand before letting the remaining PVA material dissolve away completely in warm water. The whole process of removing the inner core takes as little as five minutes with a bit of manual labor as the PVA softens.

3. Results

With regard to testing the physical products of the process, both the final medical device and the overmolded trachea were evaluated. The stent print details were observed to verify radial orientation of the print layers, while the quality and utility of the trachea airway print surfaces were analyzed through a Hausdorff and time-to-print analysis. A Hausdorff analysis measures the distance from every point of a reference mesh to the closest point on a target mesh [36].

3.1. Stent Print

The final stent is shown below in Figure 13 printed on a standard and overmolded airway model. A stent with the airway model dissolved is also shown. Finally, Figure 14 shows a close up of the layers of the stent oriented along a specific axis.

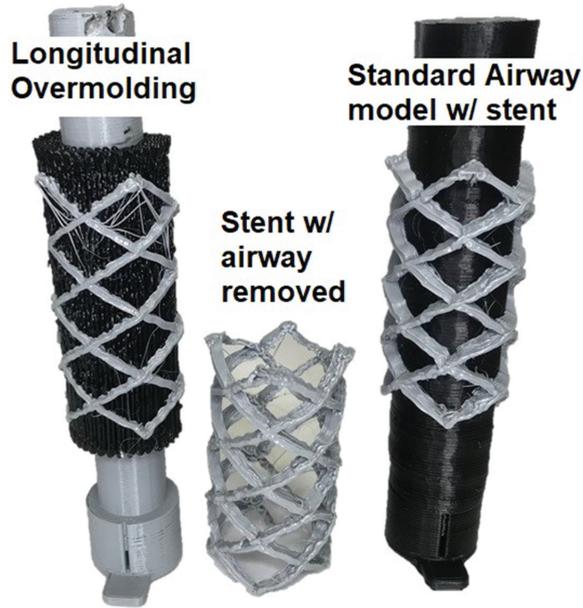


Figure 13. The stent is once again printed onto the patient's trachea model.

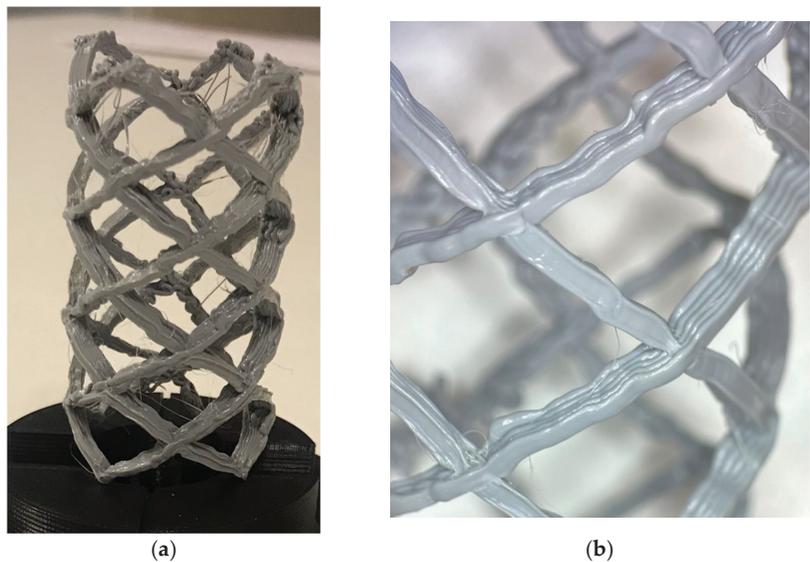


Figure 14. (a) The completed stent model. (b) Close up of the radial layers of the stent.

3.2. Trachea Airway Prints: Mesh Analysis

A 3D scan of the surfaces of the final airway prints is shown in Figure 15. On the overmolded prints, many of the surface features are left appearing rough, such as the repeating pattern of bulges due to the trachea cartilage. A Hausdorff analysis was performed to determine the overall accuracy and repeatability of the printed trachea models in the segmented and overmolding methods. Three of each standard, longitudinal, and transverse trachea models were examined here. All nine models were truncated to be the same height of 70 mm, for a proper comparison.

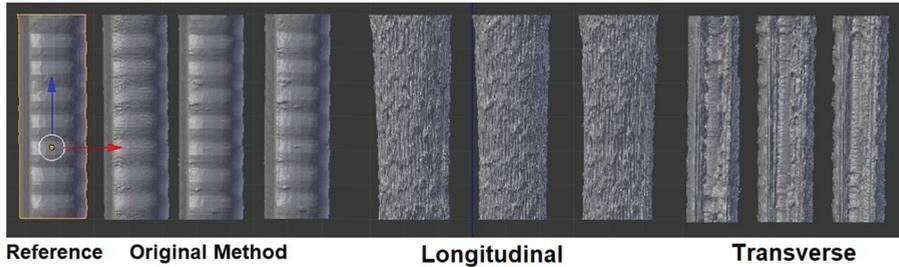


Figure 15. Comparison of the original, longitudinally rastered, and transversely rastered airway models.

In this case, all nine of the scanned meshes in Figure 15 were compared to the reference mesh sliced directly from the original Mimics file. A heatmap showing root mean square (RMS) values for each point from the analysis is also shown below in Figure 16. The higher the value, the less accurate the mesh. Unfortunately, Meshlab 2016 [37], the software package used to generate the heatmaps in Figure 16, does not provide an absolute scale for these analyses. Each model uses a relative scale, where red indicates the lowest value and blue indicates the highest for that specific model.

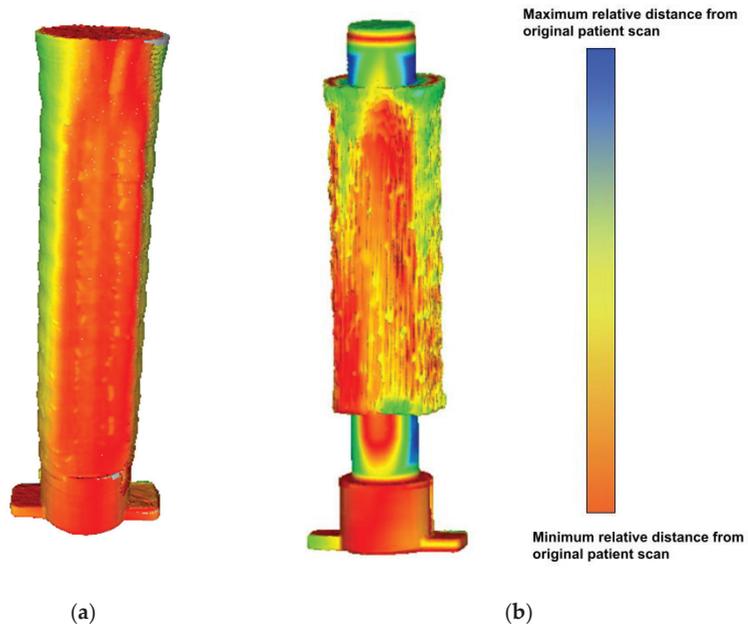


Figure 16. (a) Sample Hausdorff heatmap for the standard airway model. (b) Sample Hausdorff heatmap for the longitudinal airway model.

The RMS Hausdorff values are recorded below in Table 2, along with the averages and standard deviations of the three samples in each group.

Table 2. RMS Hausdorff distances for each printing method.

Airway Hausdorff RMS (mm)	Sample 1	Sample 2	Sample 3	Mean	SD
Standard	0.2997	0.4275	0.3713	0.3662	0.052
Longitudinal	0.8602	0.7921	0.7769	0.8098	0.036
Transverse	0.8272	1.3325	0.8673	1.0090	0.229

3.3. Comparisons for Airway and Stent Printing

The three methods chosen to print the trachea airway and the stent were compared based on the time they contributed to the entire stent fabrication process. Table 3 shows that once segmentation is complete, the printing time of the stent is relatively short.

Table 3. Time savings between standard and alternative methods in hours.

Time (Hours)	Segmenting	Blank	Airway	Stent	Total (Blank Not Included)
Standard	2.00	-	1.25	0.25	3.50
Longitudinal	2.00	0.50	0.50	0.25	2.75
Transverse	2.00	0.50	1.50	0.25	3.75

4. Discussion

4.1. Implications of the Patient-Specific Stent Case Study

The stent print represents an example of a patient-specific medical device prototype which highlights the unique advantages of the seven-axis printer system. The stent's geometry contains properties which could not be reconstructed without the system's ability to manipulate the influence of gravity and print on a rotating surface. For example, in Figure 14, it can be observed that because scaffolding was not required, the surface of the part remains unblemished. In addition, curvilinear print layers form the entirety of the stent body. Compared to other additive manufacturing methods, a primary observation about this system's utility can be made. While many available additive manufacturing technologies are capable of producing various types of medical stents [38,39], the system introduces novel material deposition directional control which can positively impact the strength and stiffness properties of the finished part [26]. For example, giving the user control over the layer orientation, allows for creation of structures with applications in high-stress biological environments. Overall, the particular created device could not be used in clinical practice at this level of development, but indicates that it is highly reasonable that patient-specific devices like a large-scale airway stent could be generated quickly and effectively using the methods outlined.

4.2. Evaluation of Processes Parallelization

Although requiring print quality to be sacrificed in the process, the trachea airway prints represent additional manufacturing capabilities made available by the seven-axis system. The system allows the user to parallelize the printing of a large volume of the anatomy mold with the segmenting of the patient's unique anatomical surface features. A manufacturing pipeline could theoretically be developed wherein patient data is received, segmented, and immediately transferred to a pre-printed anatomy blank, saving time in the translation of medical data into a patient specific device. The parallelization methods will require significant development however, with only the longitudinal overmolding generating moderate time-savings. The overmolded models also have fewer or no remaining trachea cartilage rings or defining features of the original patient anatomy. In addition, they have significantly higher Hausdorff error values than the segmented

models. One may therefore assume these models are poor representations of the original anatomy. Comparing the overmolding methods to each other, the longitudinal models are more accurate than the transverse models despite the fact that the transverse models do clearly at least capture the rings of cartilage. This can be explained by examining the mechanics of the system; the rotation axis through the centerline of the trachea model, U, is less accurate than the (X,Y,Z) axes used in the longitudinal method due to the fact that the nozzle position in Cartesian space is affected by both the angle and the distance from the U axis. Any inherent angular inaccuracies in the rotational axis are magnified as deposition occurs farther from the axis. The rings remain visible in the transverse method because they are aligned with the printing travel direction in this case, despite the less accurate model. Overall, the statement can be explored, that with future optimization, surface finish and overmolded print quality can be improved to the point where it parallels segmented printing. Potential areas of improvement include the mechanical components of the system, improved slicing and filtering algorithms, and filament variation capabilities. Also worthy of note, the standard deviation for the longitudinal models is less than that of the standard models, indicating that while they may not be as accurate to the reference mesh, the alternative process is repeatable and consistent.

4.3. Current Benefits and Future Directions

Indeed a benefit of the technology presented in a paper would be for training of physicians and/or helping them become more familiar with patient-specific anatomy. The above case study could currently be employed as a surgical planning tool, thereby providing physical models which can be handled and examined by physicians and patients prior to actual surgery. In the future, other features could be added to the seven-axis technology to fully realize the proposed patient-specific design scenario. For example, other moving printheads, such as powder direct energy deposition (DED) heads or screw-extrusion based nozzles, could replace the direct extruder shown, to allow for the use of filaments commonly associated with medical products, such as metal filaments or non-filamentous biocompatible compounds. In addition, the resolution of the system could be improved to match the resolutions of other common additive manufacturing methods. While lacking some of these developments, the study demonstrates potential advantages for incorporating additional additive manufacturing technology in the patient-specific design space, and provides an idealized pipeline for translating anatomy data into a patient-specific medical device prototype.

5. Conclusions

Additive manufacturing as a set of technologies has expanded in recent years. Many additive manufacturing methods are being used to great effect in the medical devices field for creating structures that are tailored to specific patient anatomy. However, limitations of this technology still exist. Before now, additive manufacturing systems have provided users with limited control over the local deposition orientation of material onto a workpiece, minimizing user-control over material properties and workpiece geometry. The seven degree-of-freedom additive manufacturing system described here solves a number of these problems, by allowing for the simultaneous control over the orientation of the deposition nozzle and the workpiece in 3D space. With this control, users can specify the local deposition properties of material via user-generated custom toolpaths as well as print on highly nonuniform surfaces in changing coordinate planes. These developments show promise beneficial to the patient-specific device prototyping field due to the suitability in working with highly nonuniform anatomical geometries. With continued improvement to the software and hardware components, seven-axis additive manufacturing systems will likely solve new challenges in patient-specific medical device design.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/machines10121144/s1>. Video S1. Demonstration of the seven-axis system modifying the deposition nozzle angle around a single point. Video S2. Demonstration of the print bed tilting along the seven-axis system's V axis to create a structure with an overhang. Video S3. Demonstration of the print bed tilting along the seven-axis system's V axis to create a structure with angularly offset deposition layers. Video S4. Timelapse of the fabrication of a curved structure with angularly offset deposition layers.

Author Contributions: Conceptualization, J.M.H. and A.G.E.; methodology, J.M.H.; software, J.M.H.; validation, J.M.H. and A.G.E.; formal analysis, J.M.H.; investigation, J.M.H.; resources, J.M.H.; data curation, J.M.H.; writing—original draft preparation, M.L.; writing—review and editing, M.L. and A.G.E.; visualization, J.M.H.; supervision, A.G.E.; project administration, A.G.E.; funding acquisition, A.G.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by Boston Scientific to match the 2017 University of Minnesota MN Robotics, Sensing, and Additive Manufacturing (RSAM) Drive grant. Further information is available in: <https://mndrive.umn.edu/robotics/funding/mndrive-rsam-industrial-partnership-grants>, accessed on 11 November 2022.

Data Availability Statement: The data presented in this study are available in: <https://conservancy.umn.edu/handle/11299/211762>, 11 November 2022.

Acknowledgments: The authors acknowledge the support of the Bakken Medical Devices Center, both financially and the prototyping facilities. The authors also acknowledge the financial support of Boston Scientific Corporation and University of Minnesota MN Drive grant program.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- Hafez, M.A.; Moholkar, K. Patient-specific instruments: Advantages and pitfalls. *SCIOT-J* **2017**, *3*, 66. [CrossRef]
- Gualdrón, C.I.L.; Ibarra, E.R.B.; Bohórquez, A.P.M.; Bohórquez, I.G. Present and future for technologies to develop patient-specific medical devices: A systematic review approach. *Med. Devices* **2019**, *12*, 253–273. [CrossRef] [PubMed]
- Wong, K.C. 3D-printed patient-specific applications in orthopedics. *Orthop. Res. Rev.* **2016**, *8*, 57–66. [CrossRef] [PubMed]
- Kolken, H.M.A.; Lietaert, K.; van der Sloten, T.; Pouran, B.; Meynen, A.; Van Loock, G.; Weinans, H.; Scheys, L.; Zadpoor, A.A. Mechanical performance of auxetic meta-biomaterials. *J. Mech. Behav. Biomed. Mater.* **2020**, *104*, 103658. [CrossRef]
- Gueche, Y.A.; Sanchez-Ballester, N.M.; Cailleaux, S.; Bataille, B.; Souliail, I. Selective Laser Sintering (SLS), a New Chapter in the Production of Solid Oral Forms (SOFs) by 3D Printing. *Pharmaceutics* **2021**, *13*, 1212. [CrossRef]
- Tagliaferri, V.; Trovalusci, F.; Guarino, S.; Venettacci, S. Environmental and Economic Analysis of FDM, SLS and MJF Additive Manufacturing Technologies. *Materials* **2019**, *12*, 4161. [CrossRef] [PubMed]
- Mazzoli, A. Selective laser sintering in biomedical engineering. *Med. Biol. Eng. Comput.* **2013**, *51*, 245–256. [CrossRef]
- Ventola, C.L. Medical Applications for 3D Printing: Current and Projected Uses. *Pharm. Ther.* **2014**, *39*, 704–711.
- Melčová, V.; Svoradová, K.; Menčík, P.; Kontárová, S.; Rampichová, M.; Hedvičáková, V.; Sovková, V.; Příkryl, R.; Vojtová, L. FDM 3D Printed Composites for Bone Tissue Engineering Based on Plasticized Poly(3-hydroxybutyrate)/poly(D,L-lactide) Blends. *Polymers* **2020**, *12*, 2806. [CrossRef]
- Shaqour, B.; Samaro, A.; Verleije, B.; Beyers, K.; Vervaeet, C.; Cos, P. Production of Drug Delivery Systems Using Fused Filament Fabrication: A Systematic Review. *Pharmaceutics* **2020**, *12*, 517. [CrossRef]
- Chen, J.V.; Dang, A.B.C.; Dang, A. Comparing cost and print time estimates for six commercially-available 3D printers obtained through slicing software for clinically relevant anatomical models. *3D Print. Med.* **2021**, *7*, 1. [CrossRef] [PubMed]
- Uno, M.; Sugimoto, K.; Taniguchi, I.; Tanaka, K. Six-Degree-of-Freedom Articulated Robot Mechanism and Assembling and Working Apparatus Using Same. U.S. Patent 5,197,846, 30 March 1990.
- Zhang, H.; Liu, D.; Huang, T.; Qingxi, H.; Lammer, H. 3D Printing Method of Spatial Curved Surface by Continuous Natural Fiber Reinforced Composite. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Qingdao, China, 28–29 December 2019; IOP Publishing: Bristol, UK, 2020; Volume 782, p. 022059.
- Newell, C. Multiple Axis Robotic Additive Manufacturing System and Methods. U.S. Patent 11,198,252, 14 December 2021.
- Hunter, L.W.; Brackett, D.; Brierley, N.; Yang, J.; Attallah, M.M. Assessment of trapped powder removal and inspection strategies for powder bed fusion techniques. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 4521–4532. [CrossRef]
- Paul, R.; Anand, S.; Gerner, F. Effect of Thermal Deformation on Part Errors in Metal Powder Based Additive Manufacturing Processes. *J. Manuf. Sci. Eng.* **2014**, *136*, 031009. [CrossRef]

17. Zaeh, M.; Branner, G. Investigations on residual stresses and deformations in selective laser melting. *Prod. Eng.* **2009**, *4*, 35–45. [CrossRef]
18. Tang, M.; Pistorius, P. Anisotropic Mechanical Behavior of AlSi10Mg Parts Produced by Selective Laser Melting. *JOM* **2017**, *69*, 516–522. [CrossRef]
19. Laa, D. Fine Tune Your Resin 3D Print Supports. *Make* **2020**, *75*.
20. Martín-Montal, J.; Pernas-Sánchez, J.; Varas, D. Experimental Characterization Framework for SLA Additive Manufacturing Materials. *Polymers* **2021**, *13*, 1147. [CrossRef]
21. Tumbleston, J.R.; Shirvanyants, D.; Ermoshkin, N.; Janusziewicz, R.; Johnson, A.R.; Kelly, D.; Chen, K.; Pinschmidt, R.; Rolland, J.P.; Ermoshkin, A.; et al. Continuous liquid interface production of 3D objects. *Science* **2015**, *347*, 1349–1352. [CrossRef]
22. Dizon, J.R.; Gache, C.C.; Cascolan, H.M.S.; Cancino, L.T.; Advíncula, R. Post-Processing of 3D-Printed Polymers. *Technologies* **2021**, *9*, 61. [CrossRef]
23. Lee, C.S.; Kim, S.G.; Kim, H.J.; Ahn, S.H. Measurement of anisotropic compressive strength of rapid prototyping parts. *J. Mater. Process. Technol.* **2007**, *187–188*, 627–630. [CrossRef]
24. Sood, A.K.; Ohdar, R.K.; Mahapatra, S.S. Experimental investigation and empirical modelling of FDM process for compressive strength improvement. *J. Adv. Res.* **2012**, *3*, 81–90. [CrossRef]
25. Huss, J.M. Seven Degree of Freedom Curvilinear Toolpath Generation for FDM 3D Printing with Applications in Patient-Specific Medical Device Prototyping. Ph.D. Thesis, University of Minnesota, Minneapolis, MN, USA, 2019.
26. Huss, J.M.; Erdman, A.G. Gravity Augmented Fused Filament Fabrication Additive Manufacturing. *J. Med. Devices* **2022**, *in press*.
27. FDA.gov. Available online: www.fda.gov/medical-devices/overview-device-regulation/regulatory-controls (accessed on 16 November 2022).
28. Zhao, G.; Ma, G.; Feng, J.; Xiao, W. Nonplanar slicing and path generation methods for robotic additive manufacturing. *Int. J. Adv. Manuf. Technol.* **2018**, *96*, 3149–3159. [CrossRef]
29. Folch, E.; Keyes, C. Airway stents. *Ann. Cardiothorac. Surg.* **2018**, *7*, 273–283. [CrossRef] [PubMed]
30. Sharma, N.; Aggarwal, L.M. Automated medical image segmentation techniques. *J. Med. Phys.* **2010**, *35*, 3–14. [CrossRef]
31. *Mimics*; Materialize NV: Leuven, Belgium, 2018.
32. *Cura*; Ultimaker: Utrecht, The Netherlands, 2018.
33. *Creality CR-10*; Creality: Shenzhen, China, 2019.
34. *MATLAB R2018b*; Mathworks: Natick, MA, USA, 2018.
35. Moller, T.; Trumbore, B. Fast, Minimum Storage Ray-Triangle Intersection. *J. Graph. Tools* **1997**, *2*, 21–28. [CrossRef]
36. Gneiting, T.; Ševčíková, H.; Percival, D.B. Estimators of Fractal Dimension: Assessing the Roughness of Time Series and Spatial Data. *Stat. Sci.* **2012**, *27*, 247–277. [CrossRef]
37. Meshlab; MeshLab 2016, General Public License, September 2016. Available online: <https://www.meshlab.net/> (accessed on 28 November 2022).
38. Cabrera, M.S.; Sanders, B.; Goor, O.J.G.M.; Driessen-Mol, A.; Oomens, C.W.J.; Baaijens, F.P.T. Computationally Designed 3D Printed Self-Expandable Polymer Stents with Biodegradation Capacity for Minimally Invasive Heart Valve Implantation: A Proof-of-Concept Study. *3D Print. Addit. Manuf.* **2017**, *4*, 19–29. [CrossRef]
39. Paunović, N.; Bao, Y.; Coulter, F.B.; Masania, K.; Geks, A.K.; Klein, K.; Rafsanjani, A.; Cadalbert, J.; Kronen, P.W.; Kleger, N.; et al. Digital light 3D printing of customized bioresorbable airway stents with elastomeric properties. *Sci. Adv.* **2021**, *7*, eabe9499. [CrossRef]

Article

The Development of an Automated Multi-Spit Lamb Rotisserie Machine for Improved Productivity

Xun Wei Chia¹, Poh Kiat Ng^{1,*}, Robert Jeyakumar Nathan², Jian Ai Yeow², Way Soong Lim¹
and Yu Jin Ng³

- ¹ Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, Melaka 75450, Malaysia; chiaxunwei@gmail.com (X.W.C.); wslim@mmu.edu.my (W.S.L.)
² Faculty of Business, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, Melaka 75450, Malaysia; robert.jeyakumar@mmu.edu.my (R.J.N.); jayeow@mmu.edu.my (J.A.Y.)
³ Department of Social Science and Humanities, Putrajaya Campus, College of Energy Economics and Social Sciences, Universiti Tenaga Nasional (UNITEN), Jalan IKRAM-UNITEN, Kajang 43000, Malaysia; yujin@uniten.edu.my
* Correspondence: pkng@mmu.edu.my

Abstract: Innovations in food manufacturing support the agenda for sustainable development goal 9 (SDG9) on industry, innovation and infrastructure. Pursuant to this goal, this study aims to develop an automated multi-spit lamb rotisserie machine that potentially improves the lamb-roasting productivity for small and medium enterprises (SMEs). The conceptualisation involved patents, scholarly literature and product reviews of lamb-roasting devices. The design and analysis are performed using Autodesk Inventor 2019. A scaled-down prototype is developed and tested with (1) roasting output, (2) roasting time and (3) temperature stability tests. The data for test (1) are analysed by comparing the means between control and experimental groups. The data for tests (2) and (3) are analysed using the *t*-test and Mann–Whitney U test, respectively. Significant differences are observed in tests (1) and (2), with outcomes being in favour of the proposed invention. The prototype cooks 92.27% faster with 700% more meat than a regular lamb roaster. It also cooks at a stable temperature. The cost analysis indicated that this invention could be sold at USD 278 if mass-produced. The design is structurally simple, inexpensive and easy to manufacture, allowing SMEs that rely on traditional spit-based machines to enhance their ability in producing roast lamb.

Keywords: rotisserie machine; engineering design; productivity; roast lamb; meat; usability; SDG9

Citation: Chia, X.W.; Ng, P.K.; Nathan, R.J.; Yeow, J.A.; Lim, W.S.; Ng, Y.J. The Development of an Automated Multi-Spit Lamb Rotisserie Machine for Improved Productivity. *Machines* **2021**, *9*, 165. <https://doi.org/10.3390/machines9080165>

Academic Editors: Raffaele Di Gregorio and Benoit Eynard

Received: 5 July 2021
Accepted: 9 August 2021
Published: 11 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Innovation- and productivity-driven small and medium enterprises (SMEs) play an important role in propelling the advancement of middle-income countries to high-income nations [1–3]. In Malaysia, SMEs are the drivers of the economy, accounting for 38.9% of the country's GDP in 2019, which is an increase of 0.6% from 2018 [3,4]. For food sector SMEs, there is a need to develop smart, energy-saving and cost-saving technologies in supporting the United Nation's objective for SDG 9 (industry, infrastructure and innovation) [5–7]. One of the growing food SMEs in Malaysia includes the lamb-roasting industry.

Roast lamb, or also known as Kambing Golek in Malay, is a dish where a whole lamb is roasted on a rotisserie machine. This dish has a deep cultural significance in Malaysia and Indonesia, especially during the Eid al-Adha festival where roast lamb is often prepared for the celebration [8].

Malaysia, in particular, often sees a steep increase in the demand for roast lamb around the middle of the year. In a survey, it was found that sheep or goat meat is consumed by 72% of Malaysian consumer respondents. This meat delicacy is normally served at celebration banquets, restaurants and hotels [9].

Seasonal changes in the demand introduced several problems for roast lamb suppliers. One of the problems includes the difficulty in increasing roast lamb productivity. The

roasting process for a single lamb takes about 4–5 h, and while being roasted, the lamb needs to be basted once every 15 min to prevent the loss of moisture in the meat [10]. During seasons of high demand, the quantity of roast lambs that a vendor can produce is limited by the number of roasting machines they own. Moreover, a traditional lamb-roasting machine is only capable of roasting one lamb at a time and takes up an area of approximately 900 mm × 400 mm [11].

The contemporary solution that vendors often consider to fulfil the high demands on roast lambs is to purchase more roasting machines. The average cost for a single-spit lamb rotisserie machine ranges from USD 150 to 350. Hence, this solution might not be ideal as the investment-to-profit ratio is rather low, and the supplier may face problems storing the unused machine during seasons of low demand.

Problem Statement

There have been a few products developed in the market that attempted to solve the productivity issues in meat roasting. In the rotisserie chicken industry, a machine known as the 1425.4SMiE uses four spits with independent motors to increase the capacity of chicken roasting. In the 1425.4SMiE, a total of 20 to 24 chickens can be roasted to completion within 1.5 h [12].

Some existing products use infrared as the heat source to replace conventional heating methods such as charcoal and gas. This type of roasting method has its benefits, such as short preheating duration and evenly distributed heat [13]. Compared to charcoal roasting, which has a preheating duration of 15 min, an infrared burner takes only 3 min to preheat. This duration excludes the time taken to arrange the charcoal on the tray and to start the fire [14].

Nonetheless, the above-mentioned features are not applicable for the mass production of roast lambs for several reasons. The cooking method for a roast lamb is already well established, and having charcoal as the heat source is an important factor due to customer preferences for charcoal roasting over other roasting methods. The long cooking duration allows the charcoal to infuse a smoky aroma into the meat that other types of heat sources are unable to render. This aroma stems from a chemical compound known as guaiacol [15]. Guaiacol is an aroma compound produced when heat is used to break down lignin, the resin responsible for holding strands of cellulose together to form wood [16]. Food produced from charcoal grills often has this smoky aroma that most people associate with roast lamb.

In the case of machines such as the 1425.4SMiE, it is not applicable for mass production in the context of roasting lambs due to the high cost required. A whole lamb weighs about 18 to 22 kg, while a whole chicken weighs around 2.2 to 3.2 kg. Hence, the motor power output required to rotate one whole lamb on a spit would be higher compared to the output required for a few whole chickens on a spit.

The aforementioned limitations in existing solutions assert that there are no studies that investigate inventive solutions applicable to roasting lambs for improved productivity. Hence, the study aims to develop an inventive solution that is capable of solving the inherent productivity issues of existing lamb-roasting machines in the market.

This study adds value to the community of researchers in the area of machine design as it accounts for underlying stages in the mechanical design process, which includes conceptualisation, design, development and usability evaluation [17]. There is also a need for machine designers to advance from developing rigid, single-output and manual machines to flexible, multi-output and automated machines to stay formidably competitive in productivity, efficiency and operating cost [18,19]. As such, with a special emphasis on lamb roasting, this study serves as a beneficial reference for machine design researchers in their quest to improve the productivity of their machines inventively and at a reduced operating cost.

2. Literature Review

The lamb-roasting machine is classified as a rotisserie machine because of the nature of its cooking method. Rotisserie, also known as spit roasting, is a roasting method where the meat is skewered on a spit. A spit refers to a slender solid rod used to lock the food in place while it is being cooked on fire [20]. This cooking method is common for roasting whole animals because it not only cooks the meat evenly in its own juices but also facilitates continuous basting [21]. The rotisserie cooking method is found to be used as early as the Tudor Period (1485–1603). In those days, the spit was operated through manual rotation, usually by a servant of a large community. Later on, mechanical “roasting jacks” or turnspits were invented to cook more efficiently. These devices were first powered by dogs on treadmills and by steam power and clockwork mechanisms later on [22]. Other mechanisms, such as worm gears (for torque and speed transmissions), were also used in the past.

Currently, electric motors are often the preferred actuators for lamb rotisserie machines due to the high reliability of the motors. A conventional lamb-roasting machine is characterised by a few features, namely the electric motor for automatic rotation, stainless steel body, charcoal as the heat source and grilling platforms for auxiliary cooking. There are advantages and disadvantages to each of these features.

Electric motors assist in rotating the spit. As opposed to manual rotations, rotating with electric motors eliminates the need for manpower to rotate the spit, which is considered a one-dimensional task for a worker. Although electric motors are reliable solutions, the motors are not fully utilised in conventional lamb rotisserie machines. For instance, while a 15 W electrical motor used by a lamb rotisserie product may be able to provide a torque of about 6800 Nm, the amount of torque it takes to turn a 70 kg lamb is only about 800 Nm. This condition indicates that only around 12% of the motor’s turning power is utilised to turn a single lamb.

2.1. Contemporary Lamb-Roasting Devices

2.1.1. Existing Product Review

Product 1: Electric Grill Stainless Spit Roaster [23]. The body and frame of this modern lamb rotisserie machine are made of stainless steel. Stainless steel is a common choice for such machines because it prevents surface corrosion, which is important when cooking is involved. The maintenance of stainless steel products is also easy. If utilised properly, products made from stainless steel can be expected to last for many years [24]. This product is designed to make use of charcoal as the heating source. It is also equipped with four lockable wheels for easy manoeuvring. Table 1, which was adapted from [23], shows a summary of its features and specifications.

Table 1. Summary of the features and specifications.

Price	USD 268.59
Features	Height adjustable, automatic rotation, lockable wheels, peripheral accessories
Grill Type	Charcoal grill
Material	Stainless steel
Dimension	400 mm (W) × 1180 mm (L) × 800 mm (H)
Motor specification	28 W, 110 V/50 Hz

Product 2: XL Lamb Rotisserie [25]. This lamb rotisserie device is manufactured by an Austrian company known as the Pig Lamb Rotisserie Shop and is sold on their website at the price of 279 USD/unit. Similar to Product 1, this rotisserie machine uses an electric motor to turn the spit. It includes a 110 V 40 W and 2.5 RPM motor capable of turning

an entire lamb that weighs 57 kg at most. The spit is 1650 mm in length and 33 mm in diameter.

In contrast to the previous device, which has four legs with lockable wheels, this device only has two legs supported by three beams in a tripod formation. This design also does not possess a stainless steel body to hold the charcoal. Instead, the user is expected to set up an area for the charcoal in between the device's legs.

Although this design lacks manoeuvrability and built-in charcoal storage features, the 2-legged design is more segmented and can be taken apart when it is not in use. This aspect makes it portable. Table 2, which was adapted from [25], shows a summary of the device's features and specifications.

Table 2. Summary of the features and specifications.

Price	USD 279
Features	Automatic rotation, high portability, ease of assembly
Grill Type	Depending on user
Material	Stainless steel spit, stainless steel hooks, cast iron stands
Dimension	Height: 250–500 mm; Spit: 1650 mm, 32 mm diameter
Motor specification	220 V, 3 RPM—up to 50 kg

2.1.2. Patent Review

Charcoal Barbeque Rotisserie Grill Cooker [26]: This invention is a charcoal-fired barbeque grill that transforms interchangeably between two cooking modes, namely rotisserie cooking and barbeque grill. It comes in a compact and portable design and has a charcoal basket, which rotates between a lowered horizontal orientation for grilling and a raised vertical orientation for rotisserie cooking. A handle located outside the cooker's frame is used to change the cooking modes.

The rotisserie spit lowers into the container along the sidewalls of the cooker's frame. When the charcoal basket is upright, a drip pan can be placed under the spit. In order to ensure that hot air flows over the roast, a combination of vents and a heat shield located between the vertical charcoal basket and the back wall of the cooker is included. The handle on the lid can be used to pick up the entire cooker.

Charcoal Grill [27]: This invention provides a method of igniting charcoal with improved safety and convenience compared to the traditional methods of charcoal ignition. This design requires only 1 to 2 sheets of newspaper to ignite any number of charcoal briquettes piled above the hole in the bottom of the bowl. Bringing the charcoal to a condition that is ready for cooking requires about 5 to 10 min depending on the amount of charcoal ignited.

This invention uses a few sheets of newspaper to start the fire, which is safer than methods that require the use of flammable fluid. Another advantage of this invention includes the prolonged life of the grill, which is applicable if the user complies with the recommended guidelines of use. These guidelines include using the grill without any layers of gravel, aluminium foil or the like. After the cooking is done, the coals should be burned out, and the ashes should be raked towards the hole at the grill's base. The ashes go into the receptacle for further disposal. Alternatively, water can be sprinkled on the coals to douse the embers. The excess water will flow out of the hole at the grill's base together with the dirt and ashes.

2.2. Cooking Method (Roasting/Rotisserie)

Roasting is a way of cooking that uses dry heat where hot air is used to cover the food, cooking it evenly on all sides [28]. The style of roasting known as rotisserie is commonly used because of the enhanced flavours that stem from the caramelisation and Maillard browning on the surface of the food. Maillard browning happens when the water molecules on the surface of the food are eliminated, and molecular rearrangement occurs, which subsequently produces the Amadori product (1-amino-1-deoxy-2-ketose).

The Amadori product does not contribute to flavour but is an important precursor for the flavour compound [29].

Compared to the faster types of roasting, such as oven roasting, rotisserie cooks the meat at a much slower rate, which allows the meat to retain more of its original moisture, dissolves more of the collagen that makes the meat tough and improves the meat's tenderness.

The tenderness of roast meat is greatly affected by the temperature and time of roasting. This relationship has been studied empirically. In a particular study, around 240 roasts were cooked to calculate the ideal cooking time [30]. Table 3, which was adapted from [30], shows the recommended cooking times for the corresponding lamb roasts.

Table 3. Cooking time (min/kg) for roast lamb in varying doneness degree.

Type of Roast	Doneness Degree	Recommended Time of Cooking (Min/kg)
Leg, whole bone-in (3.2–4.1 kg)	Rare	33–44
	Medium	44–55
	Well done	55–66
Leg, boneless (2.3–3.2 kg)	Rare	55–66
	Medium	66–77
	Well done	77–88
Leg, shank-half (0.9–1.8 kg)	Rare	66–77
	Medium	88–99
	Well done	99–110
Leg, sirloin-half (1.1–2.3 kg)	Rare	55–66
	Medium	77–88
	Well done	99–110
Shoulder, boneless (1.8–2.7 kg)	Rare	66–77
	Medium	77–88
	Well done	88–99
Shoulder, pre-sliced (0.9–2.3 kg)	Rare	77–88
	Medium	88–99
	Well done	99–110
Seven-rib rack (0.7–1.1 kg)	Rare	66–77
	Medium	77–88
	Well done	88–99
Crown-rib, not stuffed (0.9–1.4 kg)	Rare	33–44
	Medium	55–66
	Well done	66–77

The findings suggested that the lamb roasts increased in doneness with time. There was no difference observed between the rare and medium groups of pre-sliced shoulders. This outcome was the same for the medium and well-done groups of the boneless leg, seven-rib rack and crown-rib roasts. An average increase of 15.6 min/kg was required to raise the internal temperature from 60 to 70 °C, while an average of 11.4 min/kg was needed to raise the internal temperature from 70 to 77 °C. In terms of reaching a common internal temperature, the time/kg value required was found to be lesser in larger roasts than in smaller roasts [30].

2.3. Meat Doneness

Doneness is a measure of how thoroughly cooked the piece of meat is based on its internal temperature and colour when cooked. The definition and gradation in relation to its internal temperature vary across different dishes. For steaks, common gradations include rare, medium rare, medium, medium well and well done [31]. Apart from investing in proper cooking equipment, it is important to monitor meat doneness when roasting to

avoid food safety concerns and prevent serious foodborne illnesses such as food poisoning, typhoid, cholera, hepatitis A and dysentery [32–34].

The United States Department of Agriculture (USDA) recommends that for cuts of beef, veal and lamb to be considered safe for consumption, the internal cooking temperature should be at least 145 °F (63 °C) [35]. The same meats should be thoroughly cooked to 160 °F (71 °C) when ground or tenderised by cutting since these processes distribute bacteria throughout the meat. Table 4, which was adapted from [35–37], shows a scale used for meat doneness concerning beef and lamb.

Table 4. The meat doneness scale.

Description	Scale	Temperature Range		USDA Recommendation
		°C	°F	
Very red	Extra rare	46–49	115–125	-
Red centre; soft	Rare	52–55	125–130	-
Warm red centre; firm	Medium rare	55–60	130–140	-
Pink and firm	Medium	60–65	140–150	145 °F and rest for at least 3 min
Small amount of pink in the centre	Medium well	65–69	150–155	-
Gray-brown throughout; firm	Well done	69–71	155–160	160 °F for ground beef
Blackened throughout; hard	Overcooked	>71	>160	-

3. Materials and Method

3.1. Concept Generation

This section includes the generation of new lamb rotisserie machine concepts. These concepts are created according to the features selected from patents, journal articles and existing products. These features are shown in Table 5. The features are deemed useful in potentially improving the productivity of conventional lamb-roasting machines and are adopted into several concepts. Table 6 shows the draft of all the concepts.

Table 5. A list of useful features.

Feature	Description	Comment	Sources
Spit Height Adjustability	Adjustability of the rotisserie spit height (distance between roast lamb and heat source). Can be achieved by pin joints/mechanism.	Important feature for temperature control.	[23,25]
Automatic Spit Rotation	The roast lamb is rotated as it is being roasted. Can be achieved by an electric motor/pulley system.	Very important feature for productivity.	[23,25]
Grilling Platform	A metal grill that is placed on top of a heat source to grill secondary items such as vegetables and smaller pieces of meat.	Good features for productivity. Eliminates the need for another apparatus for grilling.	[23,26,27]
Accessible Skewer	The skewer can be taken out from the stand easily for basting purposes.	Important feature for productivity.	[25]
Multiple Rotisserie Spits	Increased number of roast lamb output.	Very important feature for productivity. Able to increase productivity without changing the cooking method.	[12]
Lockable Wheels	Wheels at the base of the machine for portability.	Average for productivity.	[23,27]

Table 6. Draft of the concepts.

Concept	Draft
A	
B	
C	
D	

Concept A: This concept utilises a central bevel/mitre gear mechanism to rotate an array of lamb rotisserie stations. A pulley is used to rotate a long, central pinion gear, and as the mitre gear mechanism is engaged, the pinion gear drives the rotisserie spits of multiple lamb-roasting stations. This design aims to increase productivity by increasing the number of rotisserie machines without increasing the number of rotisserie spit actuators.

By using a central mitre gear mechanism, multiple rotisserie stations are driven by a motor. The number of rotisserie stations can be increased further by increasing the length of the pinion mitre gear.

Concept B: This concept includes a rack-and-pinion gear mechanism and a roller chain/sprocket mechanism. The rack-and-pinion gear is placed at the centre of the platform and is used to adjust the distance between the rotisserie spits and heat source. Multiple rotisserie spits are located on top of a platform. The platform is installed with a roller chain mechanism that is driven by an electric motor. The rotisserie spits are modified to include a sprocket near the handle. The sprocket interacts with the roller chain mechanism. As the roller chain moves in a linear and horizontal direction, the rotisserie spits are rotated due to the sprocket-roller chain interaction.

Concept C: Concept C aims to solve the productivity issue of the traditional lamb-roasting machine by replacing the charcoal-fired heat source with an infrared burner. Unlike charcoal fire, an infrared burner has a faster heat-up time and does not require maintenance to sustain the temperature of the heat source. An infrared burner is located at the back wall of the cooker's body, and a lid is used to prevent the escape of hot air from the cooker.

Concept D: Concept D intends to increase the productivity of a lamb-roasting station by installing an automatic basting mechanism. Basting is the action of applying marination to the lamb's surface to retain the meat's moisture, and it is done every 15 min. On average, a roast lamb takes about 3–4 h to cook from start to finish, which means that a user of the traditional lamb-roasting device would have to baste the lamb about 6–8 times throughout a single cooking session. The automatic basting mechanism solves this problem of incessant basting, which potentially reduces the workforce, thereby increasing productivity.

3.2. Concept Selection

The concept selection is done through a scoring process. The concepts are rated numerically using scales relative to a benchmark. The following steps are taken to conduct the scoring process.

1. A set of criteria is created.
2. A reference design is identified. In this case, a typical lamb rotisserie machine is chosen as a reference.
3. Weights are designated to each individual criterion.
4. The different concepts are evaluated and assigned ratings.
5. The weighted score and rank of the concepts are determined.

The list of selection criteria includes productivity, cost-effectiveness, ease of storage and portability. The justifications for using each of the selection criteria are as follows:

1. Productivity. The core purpose of this study involves developing an inventive solution to improve the productivity of conventional lamb-roasting solutions. Based on this core purpose, the concept selection should emphasise productivity. Hence, a 50% weightage is allotted to the productivity criterion.
2. Ease of storage. The demand for roast lamb fluctuates all year round. It is somewhat important for the solution to be easily stored or possess space-saving attributes so that it can be kept away easily when not in use. Thus, a weightage of 10% is assigned to this selection criterion.
3. Portability. Lamb rotisserie cooking is mostly done outdoors due to the high temperature and burning of charcoal. In view of this condition, it is somewhat important for the solution to be portable for the user to conveniently move it around (for instance, from indoors to outdoors). Hence, a weight of 10% is allocated to this selection criterion.
4. Cost-effectiveness. Every invention needs to be cost-effective, as it improves manufacturability and start-up costs. Therefore, a weight of 30% is given to this selection criterion.

Table 7 shows the scoring process. The ratings were solely proposed by the main author based on his specific experience and knowledge of various rotisserie machines, with over 4 years of research experience in this area of study. In reference to the main author's superior design sense in this specific area, the co-authors of this study concurred with the ratings and rankings provided by the main author. This process has been used and published in previous studies [38–40]. The scores range from 1 to 5. The description for each numerical value is as follows:

- Score 1: much worse than reference;
- Score 2: worse than reference;
- Score 3: matches reference;
- Score 4: better than reference;
- Score 5: much better than reference.

Table 7. Scoring of the concepts.

Criteria	W (%)	Concept									
		Concept A		Concept B		Concept C		Concept D		Reference	
		R	WS								
Productivity	50	4	2.0	4	2.0	4	2.0	3	1.5	3	0.6
Ease of Storage	10	3	1.8	3	0.3	3	0.3	3	0.3	3	0.3
Portability	10	2	0.2	3	0.3	2	0.2	2	0.2	3	0.6
Cost-Effectiveness	30	4	1.2	4	1.2	3	0.9	1	0.1	3	0.9
Total Score		5.2		5.3		3.4		2.1		2.4	
Rank		2		1		3		5		4	
Continue?		No		Yes		No		No		-	

Notes: W—Weight; R—Rating; WS—Weighted score; Reference—Conventional lamb-roasting machine.

The scoring results show that concept B ranked first among the other concepts, with a total weighted score of 5.3. This design excels in the productivity aspect as it uses multiple spits for the machine to produce multiple roast lambs. It is also more cost-effective than the other concepts as it allows for multiple rotisserie spits to be actuated with only 1 electric motor, making it an inexpensive option with regard to its production. Hence, concept B is selected as the finalised design for further development.

3.3. Material Selection Standards in the Food Industry

In food manufacturing, cleanliness and hygiene are of paramount importance. The most effective materials for food processing equipment include non-corrosive and inert materials [41]. The material chosen for this study needs to have high heat resistance as charcoal burns at temperatures exceeding 1100 °C [42]. In comparison, the melting point of iron is approximately 1200 to 1550 °C. The most common form of corrosion is oxidation, where oxygen reacts with a metal, usually in the presence of water, to produce more non-reactive material such as rust. For ferrous metals, iron and steel, rust ruins the surface quality and structural stability of the equipment. However, for other types of metals, oxidation may be beneficial [43].

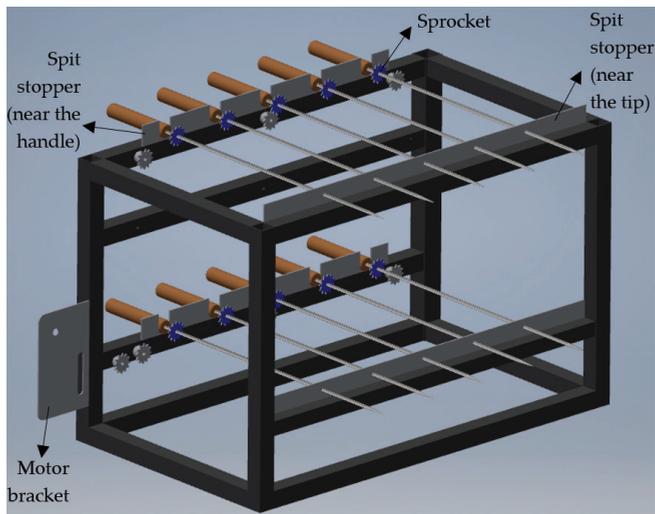
Stainless steel is used widely in food manufacturing. It is characterised by the addition of chromium (at least 10.5% of the total composition). Chromium is highly reactive to oxygen-enriched environments and quickly forms a strong passivated barrier on its outer surface. This barrier is highly resilient and protects the internal structures from further corrosion [44].

All components in contact with food during machine operations have to abide by the above-mentioned constraints. Apart from these constraints, the material selected for the base of the machine also has to be strong enough to withstand the total load on the entire machine. Due to its cost, heat resistance, strength, corrosive resistance, ease of

machining, availability and conventional use in food manufacturing equipment, stainless steel is chosen for further design simulations.

3.4. Design Drawing of Modified Concept

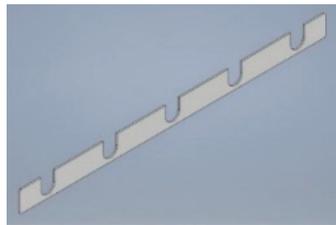
The design drawing and optimisations are presented in the data availability sheet. After careful considerations, the rack and pinion gear mechanism used for the height adjustment of the rotisserie spits (distance from heat source) is eliminated. Instead, a second platform is created to support more rotisserie spits for enhanced productivity. The modified design supports up to 10 rotisserie spits. Figure 1a shows the 3D drawing of the modified design's complete assembly. A groove is made at the side of the motor bracket in Figure 1b so that a sprocket can be installed on top to guide the movement of the roller chain. Therefore, the tension of the roller chain is adjustable for maintenance or emergencies. The space between the rotisserie spits is about 300 mm wide.



(a) The assembly after design modification



(b) Motor bracket



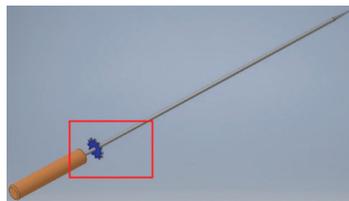
(c) Spit stopper (near the handle)



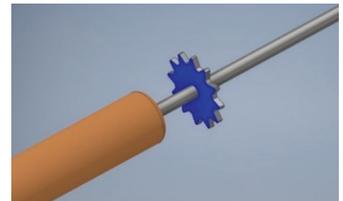
(d) Spit stopper (near the tip)



(e) Sprocket



(f) Spit with sprocket



(g) Spit with sprocket (Enlarged)

Figure 1. A 3D drawing of the complete assembly and parts.

During the roasting process, the rotisserie spits will rest on top of the moving roller chain. In order to prevent the translational motion of the spits, a stopper must be included. Figure 1c,d shows the isometric views of the rotisserie spit stopper placed near the rotisserie spit handle and near the tip of the rotisserie spit. Figure 1e shows the 3D drawing of the sprocket. Figure 1f,g shows the rotisserie spit assembled with the sprocket and an enlarged view of the sprocket on the spit. The sprocket will rest on the moving roller chain and facilitate the rotation of the spit.

3.5. Simulation on Modified Concept

The load on the design is represented by the mass of the rotisserie spits and the lamb. Each spit weighs 2.761 kg, and each lamb weighs 23.6 kg. During the operation, each rotisserie spit rests on the frame at 2 opposing points. Thus, each load component represents half of the load composed of the rotisserie spit and lamb, which is around 129.3 N. The material assigned to the frame is steel.

Figure 2 shows the maximum displacement of the frame, which occurs in the middle of the upper platform's horizontal column as indicated. The maximum displacement is 0.249 mm, which happens at the horizontal bars on which the rotisserie spits are positioned. The minimum displacement is 0 mm, which happens at most of the frame columns where no static force is exerted. The maximum displacement of 0.249 mm is insignificant in relation to the length of the horizontal column, which is 600 mm. In addition, the maximum equivalent strain from the analysis is 0.00006052 mm/mm or 60.52 $\mu\epsilon$, which is reflective of the small displacement obtained.

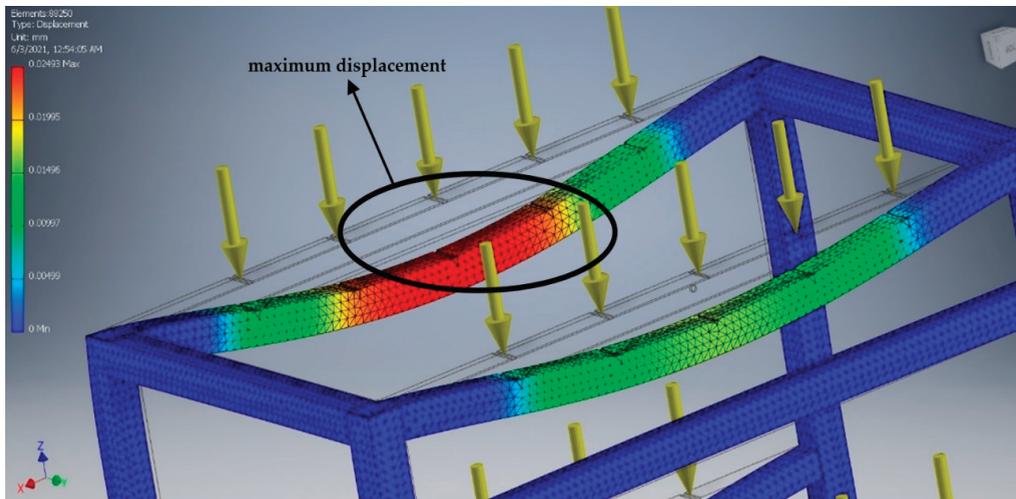


Figure 2. Stress simulation results for the frame (displacement).

Figure 3 shows the von Mises stress analysis. The maximum von Mises stress acting on the frame is 8.16 MPa. Most of the stress concentrates on the edge of the frame, where the surface area is the smallest. The von Mises stress exerted on the frame does not exceed the tensile strength of steel which is approximately 370 MPa.

The safety factor remained as 15 despite optimisations in the design's column dimension (as per the data availability sheet). This design is still considered over-engineered. A change of material might be able to reduce the safety factor to an appropriate level. However, it was established that stainless steel is the preferred material for food production equipment due to its high corrosion resistance, heat resistance and ease of maintenance. This preference is also recommended under the guidelines of the FDA [45]. Considering the preference of using stainless steel coupled with a concern of exceeding the slenderness

ratio, which creates risks in buckling and also the time constraint of completing the project, the researchers decided to proceed with the present design and material.

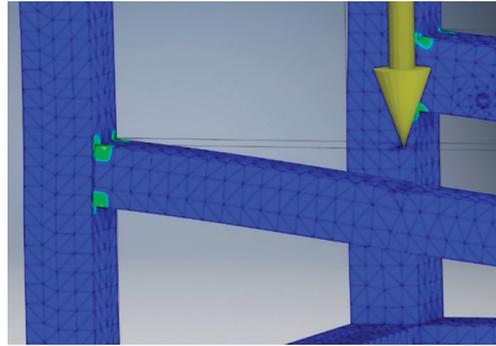


Figure 3. Stress simulation for the frame (von Mises stress).

3.6. Prototype Fabrication

3.6.1. Prototype Scaling

Due to project cost constraints, the prototype design is scaled down for fabrication. Table 8 shows the dimension comparison of the full-size lamb rotisserie machine and the prototype lamb rotisserie machine. The frame of the prototype is scaled down at a ratio of 2.25:1 in height, 4:1 in length, and 3:1 in width.

Table 8. Size comparison between full-size and prototype lamb rotisserie machine.

Dimension	Full Size (mm)	Prototype Size (mm)
Height	900	400
Length	1400	360
Width	1900	600

3.6.2. Parts Procurement

Based on the optimal dimensions and specifications, the researchers identified that the 12SB12 sprocket and the RS25 roller chain are suitable parts to be used in the proposed lamb rotisserie machine. The motor used is the SPG30-120 K 12 V electric motor [46]. This motor is chosen for its low RPM as the prototype requires a slow rotational speed and high torque.

3.6.3. Fabrication of Frame

The metal frame of the prototype is made out of 25 mm rectangular mild steel columns. The mild steel columns were purchased from the subcontractor’s warehouse. Table 9 shows the assortment of mild steel columns required to make up the frame of the prototype. The machinist separated the mild steel columns into 4 pieces of 400 mm columns, 4 pieces of 360 mm columns and 6 pieces of 620 mm columns.

Table 9. Dimension and amount of mild steel columns required.

Dimension of Mild Steel Column (mm)	Amount
400	4
360	4
600	6

3.7. Test Plan

There are 3 main experiments conducted to test the prototype's functionality. The first two experiments focus on the productivity of the prototype, while the remaining experiment investigates the temperature stability or heat distribution when the prototype is in use. Two set-ups will be used in this experiment, one being the experimental set-up and another being the control set-up. The proposed lamb rotisserie machine is used for both set-ups. When the prototype is utilised at its full capacity (all 10 rotisserie spits used), it is a representation of the experimental set-up. When the prototype is utilised at its limited capacity (only 1 rotisserie spit used), it is a representation of the control set-up. In the experiments, the controlled variable is roasting time and the number of cooked lamb meat produced. Figure 4a,b shows the experimental and control set-up, respectively.

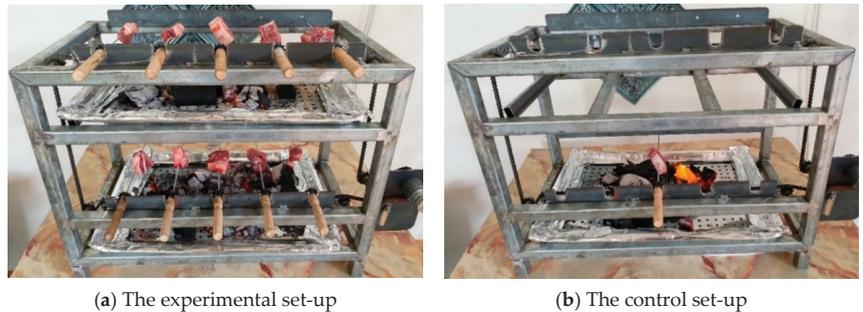


Figure 4. The prototype set-up for the experiments.

In this project, two experiments are conducted to test whether the prototype improves productivity in comparison with a regular lamb-roasting machine. The experiments are (1) the roasting output test and (2) the roasting time test. Minitab 17 is used to run the two-sample *t*-test, Mann–Whitney U test and other basic statistical calculations such as mean and standard deviation.

The main researcher (main author) gave his written informed consent prior to the experiments. All procedures and protocols have been approved by the Research Ethics Committee (REC) from the Technology Transfer Office (TTO) of Multimedia University. The research ethics approval for the project has been granted with the approval number EA0052021 on 26 February 2021, and the approval letter has been endorsed by the TTO Director-cum-REC Secretariat of the university.

3.7.1. Experiment 1: Roasting Output Test

This experiment compares the roasting capabilities (amount of roasted meat produced in a total fixed time of 16 min) between the experimental set-up and the control set-up. The range of temperatures at which the meat was deemed to be successfully cooked is 60–71.11 °C (140–160 °F). The internal temperature of the meat is measured using a meat temperature probe. The procedures used for the experimental set-up are as such:

1. The lamb meat is cut into small chunks with a uniform thickness of 20 mm in the cross-section.
2. Ten pieces of lamb meat are inserted into the ten separate rotisserie spits and set aside.
3. The charcoal is ignited with a charcoal starter and then placed onto the charcoal tray of the prototype.
4. All rotisserie spits are placed onto their respective slots.
5. The timer is started as soon as the motor of the prototype is started.
6. The timer is paused after 16 min. The cooked pieces of meat are taken out, and another batch of raw meat is inserted into the rotisserie spit.

7. The internal temperature of the cooked meat is measured immediately by inserting the meat internal temperature thermometer into the lamb meat for 5 s.
8. The timer is resumed when the new batch of raw meat is placed onto their respective slots.
9. Steps 6 and 7 are repeated after every interval of 16 min until the timer reaches the total time fixed time of 80 min.
10. The number of successfully cooked meat is recorded.

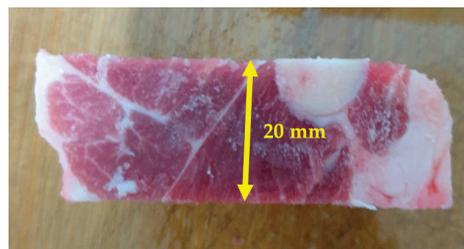
The procedures used in the control set-up are similar to the ones used in the experimental set-up. The only difference for the control set-up is in step 2, whereby instead of using 10 pieces of lamb meat and 10 separate rotisserie spits, the researcher only needs to use 1 piece of lamb meat and 1 rotisserie spit.

3.7.2. Experiment 2: Roasting Time Test

This experiment compares the difference in roasting time per unit when producing successfully cooked meat with the experimental and control set-up. Similar to experiment 1, the researcher only roasts 1 piece of meat for the control set-up. In order to observe the roasting time per unit for the experimental set-up, the researcher cooks 2 pieces of meat simultaneously. The procedures used for the experimental set-up are as such:

1. The lamb meat is cut into small chunks with a uniform thickness of 20 mm in the cross-section.
2. Two pieces of lamb meat are inserted into two separate rotisserie spits and set aside.
3. The charcoal is ignited with a charcoal starter and then placed onto the charcoal tray of the prototype.
4. The rotisserie spit is placed onto their respective slots.
5. The timer is started as soon as the motor of the prototype is started.
6. The internal temperatures of the meat are measured every 4 min by inserting the meat internal temperature probe into the lamb meat for 5 s.
7. A piece of lamb meat is taken out of the prototype as soon as its internal temperature registers at least 60 °C or 140 °F.
8. The timer is stopped once the two pieces of meat have been successfully cooked.

The procedures used in the control set-up are similar to the ones used in the experimental set-up. The only difference for the control set-up is in step 2, whereby instead of using 2 pieces of lamb meat and 2 separate rotisserie spits, the researcher only needs to use 1 piece of lamb meat and 1 rotisserie spit. Figure 5a shows the standardised thickness for the lamb meat, and Figure 5b shows the meat internal temperature thermometer.



(a) Thickness of all lamb meat standardised to 20 mm



(b) Meat internal temperature thermometer

Figure 5. The lamb meat and thermometer.

4. Results and Discussion

4.1. Finalised Prototype

The prototype for the automated multi-spit lamb rotisserie machine possesses two platforms, each accommodating up to five rotisserie spits. All the spits are automatically rotated on a roller chain which is moved by an electric motor. The inventive configuration of the sprocket-roller chain mechanism allows the entire machine to be actuated by only 1 electric motor. Figure 6a–d shows the isometric, front, back and side view of the prototype.

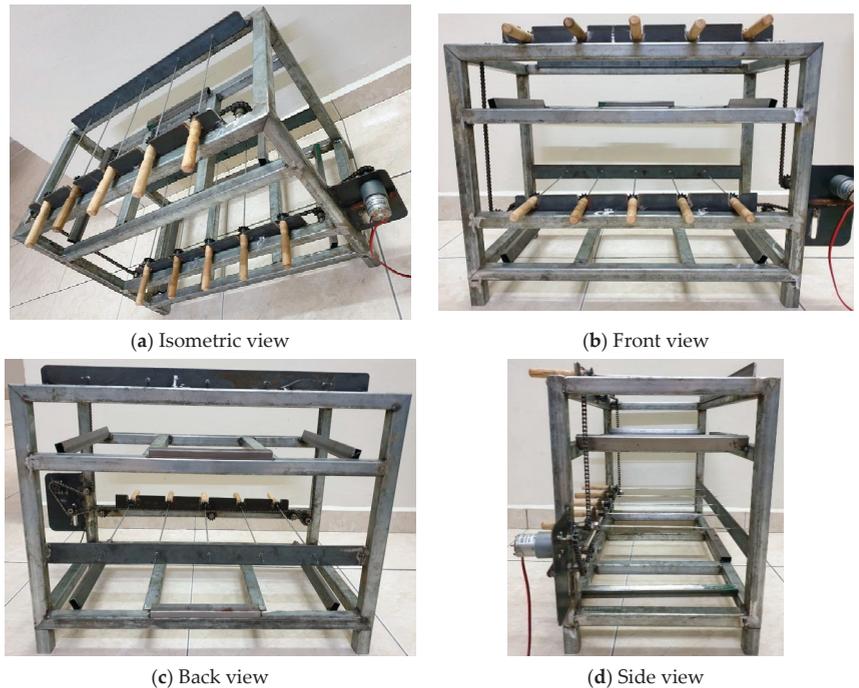


Figure 6. The finalised prototype.

The total weight of the entire rotisserie machine with all parts and components considered is 37.5 kg. This weight is considered moderate when compared to commercial lamb rotisserie machines in the market. Commercial-grade single spit lamb rotisserie devices weigh from 8 to 70 kg [25,47,48].

4.2. Sprocket-Roller Chain Mechanism

In Figure 7a, the circled locations show the freely rotating sprockets. There are 6 freely rotating sprockets installed onto the frame of the prototype. These sprockets are installed at their specific locations to hold the roller chain in place.

Figure 7b–d shows the sprocket configuration that creates a link from the top platform to the bottom platform. These links allow the rotational movement from the bottom platform to be transmitted to the top platform, thus reducing the need for individual electric motors for each platform. During the design phase of this project, the frame dimension and sprocket placement were designed to allow enough clearance for the movement of the roller chain. In Figure 7c, when the motor is turned on, the circled sprocket will rotate clockwise and move the entire roller chain in the direction of the arrows.



(a) Location of sprocket-roller chain engagements



(b) Top-bottom roller chain link, right side



(c) Sprocket configuration on the motor bracket



(d) Top-bottom roller chain link, left side

Figure 7. Sprocket-roller chain mechanism.

Figure 8a shows the rotisserie spit of the prototype. The rod of the rotisserie spit is made out of stainless steel, while the handle is made out of wood. A 25SB12 sprocket is installed on the top of the handle. As shown in Figure 8b, the sprocket on the rotisserie spit will rest on top of the roller chain. As the roller chain moves, the sprocket teeth move along the slots of the roller chain, causing the rotisserie spit to rotate concurrently. Table 10 shows the specification of the prototype.

Table 10. Specifications of the prototype.

Part	Specification
Frame dimensions	600 (L) × 400 (H) × 360 (W) mm
Frame material	Mild steel
Frame weight	10.9 kg
Number of sprockets	10
Sprocket specification	25B12
Roller chain length	10 ft
Motor	Speed: 38 RPM Torque: 0.49 Nm
Spit	295 cmm

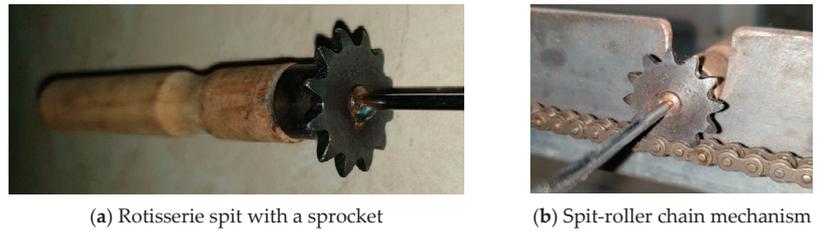


Figure 8. The rotisserie spit used for the spit-roller chain engagement.

4.3. Results and Discussion of Experiments

The experimental set-up includes the use of the automated multi-spit lamb rotisserie machine prototype, while the control set-up includes the use of the same prototype in the capacity of a single-spit rotisserie machine.

Roasting Time Test

This experiment aims to assess the time taken to produce one unit of successfully roasted lamb meat. The time is recorded when the internal temperature of the meat reaches 60 °C. Two pieces of lamb meat are roasted at the same time for each batch of the experimental set-up, whereas only one piece of lamb meat is roasted for each batch of the control set-up. The average time taken to successfully roast the lamb meat for each batch under the experimental set-up can be calculated as such:

$$t_E = \frac{t_A + t_B}{2} \quad (1)$$

where t_A —time taken to roast meat A per batch under the experimental set-up; t_B —time taken to roast meat B per batch under experimental set-up; t_E —average time taken to roast meat A and B per batch under experimental set-up.

The average roasting time for the experimental and control set-up is also compared. There are two pieces of lamb meat roasted for each batch in the experimental set-up ($N_E = 2$) and one piece of lamb meat roasted for each batch in the control set-up ($N_C = 1$). The roasting time per unit for the experimental and control set-up can be calculated using Equations (2) and (3). The data collected will be available in the data availability sheet.

$$r_E = \frac{t_E}{N_E} = \frac{t_E}{2} \quad (2)$$

$$r_C = \frac{t_C}{N_C} = \frac{t_C}{1} = t_C \quad (3)$$

where r_E —roasting time per unit for each batch under experimental set-up (s/unit); N_E —number of meat roasted per batch under experimental set-up; r_C —roasting time per unit for each batch under control set-up (s/unit); t_C —average time taken to roast meat per batch under control set-up; N_C —number of meat roasted per batch under control set-up.

Sample Size Estimation and Normality

Before a two-sample *t*-test is conducted, it is essential to inspect if the sample size is adequate. Using the pooled standard deviation, mean difference and a power value of 80%, the sample size is estimated to be 2 (see Table 11). The sample size of the current dataset is 10, which exceeds the sample size calculated by Minitab 17, indicating that the current sample size is sufficient for this study.

Table 11. Sample size estimation by Minitab.

Parameter	Value
Pooled standard deviation	9.524
Mean difference (s/unit)	476.3
Target power	80%
Sample size	2
Actual power	100%

For the assumption of normality, the Ryan–Joiner normality test proved that the data are not significantly different from a normal distribution ($p > 0.05$). Thus, the dataset is appropriate for further parametric tests. The F-test and Levene’s test also verified that there is a significant difference in the variance of the dataset between the experimental and control group ($p < 0.05$). Hence, the t -test used would apply the assumption of unequal variance.

Two-Sample t -Test

The following hypotheses are used for the roasting time test.

- Null Hypothesis, H_{0a} : There is no significant difference in the roasting time between the control and experimental set-up ($p > 0.05$).
- Alternative Hypothesis, H_{1a} : There is a significant difference in the roasting time between the control and experimental set-up ($p < 0.05$).

Table 12 shows the mean and standard deviation of the roasting time for the experimental and control set-up. The results show that the experimental set-up uses a shorter roasting time than the control set-up. It is important to note that the unit for roasting time is seconds/unit (s/unit), which measures the amount of time it takes to cook one piece of meat to the internal temperature of 60 °C. The experimental set-up uses a mean roasting time of 497.3 s/unit, while the control set-up uses a mean roasting time of 973.6 s/unit.

Table 12. Statistics for the roasting time per unit.

Set-Up	Mean (s/Unit)	StDev
Experimental	497.3	7.64
Control	973.6	20.5

Notes: N = 10; StDev—standard deviation.

According to the results in Table 13, there is a significant difference in the roasting time between the experimental and control group ($t(11) = -66.62$, $p < 0.05$). The results indicate that the automated multi-spit lamb rotisserie machine prototype is significantly more productive than a regular lamb-roasting set-up which only produces one roast lamb at a time. Therefore, H_{1a} is supported.

Table 13. Two-sample t -test results.

Parameter	Value
Estimate for difference	-476.35
95% CI for difference	(-491.59, -461.11)
t -value	-68.79
p -value	0.000
DF	11

Roasting Output Test

In this experiment, both the experimental and the control set-ups are used to roast lamb meat in five cooking sessions. The aim is to roast as much lamb meat as possible within each cooking session which lasts for 16 min. For the experimental set-up, all

10 rotisserie spits are used to cook the pieces of lamb meat. The control set-up only used one rotisserie spit to cook one piece of lamb meat. After roasting for 16 min, all the pieces of lamb meat are removed from the rotisserie spits. The internal temperature for each meat is measured. If the internal temperature of a piece of lamb meat is within the range of 60 °C to 71.11 °C, it is considered successfully cooked. Otherwise, the particular sample is rejected. In other words, only cooked meat that is classified as medium rare, medium well, medium and well done are considered successfully cooked. The ones that are classified as extra rare, rare and overcooked are considered not successfully cooked.

Table 14 shows the summary of data from the roasting output test. The results show that the experimental group is able to produce an average of eight successfully roasted lamb meat while the control group is only able to produce an average of one successfully roasted lamb meat for all five cooking sessions. All in all, the experimental set-up produces an average surplus in roast meat output of 700% as compared to the control set-up (see Table 15).

Table 14. A summary of data from the roasting output test.

Set-Up	Average Roast Meat Output (Unit)					Mean (Unit)
	a	b	c	d	e	
Experimental	7	8	8	9	8	8
Control	1	1	1	1	1	1

Notes: a, b, c, d, e—Cooking sessions.

Table 15. Surplus roast meat output (experimental versus control set-up).

Cooking Session	Roast Meat Output Surplus (%)
1	600
2	700
3	700
4	800
5	700
Mean	700

Productivity issues that are inherent in regular lamb-roasting machines are partly because the regular machine is only able to produce one roast lamb at a time. The roasting time and output tests proved that the present study's automated multi-spit lamb rotisserie machine could significantly increase the amount of roast lamb produced and significantly shorten the overall time taken to produce roast lamb.

The concept used in this study is akin to the design of an existing product known as the 1425.4SMiE Special Market chicken rotisserie machine [12]. This product is able to accommodate multiple whole chickens (up to 24 units) in one cooking session and also aims to improve productivity.

While this product possesses a similar aim, its focus is mainly on producing roast chicken. In addition, the way it produces multiple roasts at a time is also different from the way the prototype of this study functions. For instance, the 1425.4SMiE emphasises roasting multiple whole chickens inserted in a single spit, while the prototype of this study emphasises roasting multiple spits of a whole lamb. The inventive solution used by this study (i.e., using the sprocket and roller chain) also makes the current prototype unique apart from being productive.

4.4. Temperature Stability Test

When the prototype runs at its full capacity (i.e., 10 rotisserie spits running concurrently), it is noticed that at least one unsuccessfully roasted meat would be produced, be it undercooked or overcooked. This issue is possibly due to the uneven heat exposure caused by factors such as the distance of the meat to the heat source and unregulated air flow.

Therefore, it is essential to perform an overall temperature stability test using the temperature dataset collected from the previous experiment. Since the temperature dataset violated the assumption of normality, the Mann–Whitney U test (a non-parametric test) is used to determine if the temperature distribution differs significantly between the experimental and control set-up. The following hypotheses are used.

- Null Hypothesis, H_{0b} : There is no significant difference in the temperature distribution of the meat roasted with the experimental set-up as compared to the control set-up ($p > 0.05$).
- Alternative Hypothesis, H_{1b} : There is a significant difference in the temperature distribution of the meat roasted with the experimental set-up as compared to the control set-up ($p < 0.05$).

In this analysis, the temperature distribution is determined by the internal temperature of the lamb meat after being roasted for 16 min (as per the roasting output test procedures). As shown in Table 16, the Mann–Whitney U test indicated that the temperature of the experimental group (median = 150.1 °F) does not significantly differ from the temperature of the control group (median = 151.5 °F), $U = 1370$, $p = 0.3878$.

Table 16. Results of median calculated from Mann–Whitney U test.

Set-Up	N	Median (°F)	U	p-Value
Experimental	50	150.1	1370	0.3878
Control	5	151.5		

In summary, the null hypothesis, H_0 , is accepted, and the analysis confirms that the overall temperature distribution for the experimental group is considered stable even with a few samples being unsuccessfully roasted for some cooking sessions.

4.5. Temperature Outlier Tests

Although the overall meat temperature is stable with the experimental set-up, the prototype still consistently produced around 1 to 2 pieces of unsuccessfully cooked meat. It is noticed that the unsuccessful outputs come from the rotisserie spits located at the sides of the rotisserie machine. In order to verify this observation, four tests are conducted to compare the roast meat temperature in different rotisserie spit locations.

As shown in Figure 9, each rotisserie spit is assigned a number. The tests aim to identify if the meat from spits 1, 5, 6 and 10 have significantly different temperatures than the meat from spits 3 and 8 after being roasted for 16 min (as per the roasting output test procedures). Spits 3 and 8 are selected as references since there are normally no defects produced in these spit locations. Based on the location of the platforms, spits 1 and 5 are compared with spit 3, while spits 6 and 10 are compared with spit 8. Therefore, the hypotheses can be formulated as such:

- Null Hypothesis, H_{0c} : There is no significant difference in the internal temperature of the meat cooked for spits 1, 5, 6 and 10 as compared to spits 3 and 8 ($p > 0.05$).
- Alternative Hypothesis, H_{1c} : There is a significant difference in the internal temperature of the meat cooked for spits 1, 5, 6 and 10 as compared to spits 3 and 8 ($p < 0.05$).

Table 17 shows the *t*-test results for the temperature outliers. All of the meat cooked in spits 1, 5, 6 and 10 are found to register an internal temperature of less than 60 °C after being roasted for 16 min. The meat cooked in spits 3 and 8 registered internal temperatures of 60 °C and above. The tests show that there is indeed a significant difference in the internal temperature of the meat cooked for spits 1, 5, 6 and 10 as compared to spits 3 and 8 ($p < 0.05$). Hence, H_{1c} is supported.



Figure 9. Rotisserie spit position numbering.

Table 17. *t*-test results for temperature outliers.

Parameters	Tests			
	Test 1–3	Test 3–5	Test 6–8	Test 8–10
Estimate for difference	−13.32	18.26	−13.12	19.14
95% CI for difference	(−24.17, −2.47)	(3.61, 32.91)	(−18.91, −7.33)	(9.82, 28.46)
<i>t</i> -value	−3.41	3.46	−5.23	4.73
<i>p</i> -value	0.027	0.026	0.001	0.001
DF	4	4	8	8

The analysis concludes that the heat exposure for the spits located at the sides of the prototype is not as uniform in comparison to the heat exposure for the spits located in the middle. In terms of productivity, such an uneven distribution of heat is undesirable. This issue may be due to the position of the charcoal fire during the experiments. Apart from controlling the air flow of the surrounding area, one way to mitigate the issue is to periodically refill the platform with charcoal for more consistent burning.

4.6. Brief Cost Analysis

Table 18 shows the price list for the parts that make up the prototype. The cost to produce one automated multi-spit lamb rotisserie machine is 216.30 USD/unit. The high costs of workmanship and material are due to the make-to-order nature of this project. If the product were to be mass-produced in the future, the total cost will decrease. Hence, if mass production were to take place, the material and workmanship costs can be reduced by about 40%. The high cost is also attributed to the price of the 25SB12 sprockets. The price of this particular sprocket is high due to the limited supply from the gear manufacturer. The 25SB12 sprocket is also not frequently manufactured. If this prototype were to be sold commercially, the cost of the parts would also reduce. Hence, if mass production were to take place, the sprocket costs can be reduced by about 50%. However, since this prototype is actually a scaled-down version, the total cost is still estimated at USD 216.30.

There are also other variable costs involved if the plan in the future is to set up a company that manufactures these lamb-roasting machines. The manufacturing of this prototype requires skilled workers for metal cutting and welding. The minimum wage in Malaysia is forecasted to be around 1200 MYR/month (around 286 USD/month) by the end of 2021 [49]. Hence, a total salary of 450 USD/month per skilled worker would be sufficient. The total salary of 2 skilled workers is around 900 USD/month, which is about 5.63 USD/hour (accounting for 40 h a week, for 4 weeks a month). If the company

considers these skilled workers as permanent workers, the labour cost of making a single prototype becomes:

$$\text{Labour cost (USD)} = 216.30 - 23.50 + 5.63 = 198.43 \text{ USD}$$

Table 18. Price list for the product parts.

No.	Parts	Description	Unit	Unit Price (USD)	Total Cost (USD)
1	Mild steel column	400 mm	4	5.50	22
		360 mm	4	4.80	19.20
		620 mm	6	3.60	21.60
2	Sprocket	25SB12	18	3.80	68.40
3	High Torque Brushless Motor	SPG30-120 K, 12 V	1	9.70	9.70
4	Roller Chain	RS25, 10 ft	1	5.00	5.00
5	Workmanship	Professional welding, cutting	1	23.50	23.50
6	Welding cost	Material cost of welding	1	23.50	23.50
7	Charcoal briquettes	-	5	4.30	21.50
8	Charcoal tong	To pick and place hot charcoal	1	1.90	1.90
Total					216.30

The USD 23.50 is subtracted as the original workmanship cost is not required (the workmanship is replaced with permanent skilled workers). By accounting for a 40% profit margin from the labour cost, the sales price of the automated multi-spit lamb rotisserie machine is estimated to be around USD 278. This price is reasonable for owners of small and medium enterprises (SMEs) who wish to ramp up the production of their roast lamb and compete in a larger market.

5. Conclusions

The aim of this study was to develop an automated multi-spit lamb rotisserie machine for improved productivity. In order to achieve this aim, reviews of patents, research articles and existing products in the market were done. The ideas and features extracted from the literature review were then utilised to design concepts that can be used to roast lamb for improved productivity. Apart from the conceptualisation stage, the development accounted for material selection, designing with CAD, stress analysis and prototype fabrication.

The productivity and performance of this solution were also tested with several usability experiments that measured the roasting time, roasting output and temperature stability. The data were analysed using *t*-tests.

5.1. Summary of Findings and Main Outcomes

The proposed lamb rotisserie invention was able to consistently outperform the conventional way of roasting lambs by a significant margin. In the roasting output test, the proposed invention was able to cook 700% more meat than a regular lamb-roasting set-up. In the roasting time test, the proposed invention was able to produce successfully cooked lamb meat 92.27% faster than the regular lamb-roasting set-up. There was also no significant difference between the proposed invention and the regular lamb roaster in terms of cooking temperature stability.

Lastly, the sales price for the proposed invention was estimated to be around USD 278, which is a reasonable price for lamb-roasting SME owners. In summary, the proposed invention successfully demonstrated the capability of enhancing the productivity and

efficiency of producing roast lamb at a decent price with minimal changes to the system's roasting reliability. This achievement benefits not only SMEs in increasing lamb-roasting production capacity but also lamb-roasting machine designers and manufacturers that wish to gain a competitive edge over the existing lamb roaster market competition with this new invention.

5.2. Limitations

This study describes the novelty of the machine to escalate the productivity of roast lamb, which is not influenced in any way the origin of the meat [50], which could affect its texture or appearance when it is roasted. It is also believed that the quality of meat obtained for the experiment is consistent or standard, which is not affected by any external factors such as COVID-19, referring to a study carried out in China [51] or even any unfavourable economic landscape [52].

A limitation of this design is its uneven heat exposure. Through several *t*-tests using the temperature data from the roasting output test, it was found that the undercooked pieces of meat were consistent in number and often positioned close to the sides of the roasting pit. These positions included the rotisserie spits that were placed on slots that were farther away from the centre of the heat source compared to the other rotisserie spits. This uneven heat exposure caused some of the meat to be undercooked. Another limitation includes the lack of safe insulation around the frame of the body. During the roasting process, the metal frame of the prototype can get dangerously hot, exposing the risks of burning to the user.

Although proof of concept has been established in this study, an actual usability test and survey were not done among the relevant SMEs due to cost and time limitations in creating a full-scale prototype. The effects of thermal stress on the reliability of the prototype over time were also not studied in this paper.

5.3. Recommendations for Future Research

For an evenly distributed heat exposure, a thermometer can be embedded within the design to monitor the temperature of the charcoal fire and charcoal briquettes. The position of the charcoal briquettes could also be spread out evenly. In order to mitigate the risk of burning to the user, an insulation layer could be added to all vertical columns of the frame and all horizontal columns of the bottom platform as the user needs to be near these columns during the roasting process.

It would also be of interest to conduct design of experiments with the prototype to investigate various factors that can influence its performance and use more statistical analyses such as ANOVA, regression or paired samples *t*-tests. It is recommended that a full-scale prototype is developed in the future to account for further analyses and actual field testing amongst SMEs. One of the analyses can include the reliability analysis of the structure after being repeatedly subjected to high thermal stress over a period of time. Sensory analysis can also be performed to test the quality of the roast lamb produced by this machine.

Author Contributions: Conceptualization, X.W.C. and P.K.N.; data curation, X.W.C. and P.K.N.; formal analysis, X.W.C. and P.K.N.; investigation, X.W.C., P.K.N., R.J.N., J.A.Y. and Y.J.N.; methodology, X.W.C., P.K.N. and W.S.L.; project administration, X.W.C., P.K.N., R.J.N., W.S.L. and Y.J.N.; supervision, X.W.C. and P.K.N.; validation, X.W.C. and P.K.N.; visualization, X.W.C.; writing—original draft, X.W.C., P.K.N., R.J.N. and J.A.Y.; writing—review and editing, X.W.C., P.K.N., R.J.N., J.A.Y., W.S.L. and Y.J.N. All authors have read and agreed to the published version of the manuscript.

Funding: The article processing charge for this manuscript is partially supported by Multimedia University (Reference number: MMU/RMC/PC/2021/26908).

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki. All procedures and protocols have been approved by the Research Ethics Committee (REC) from the Technology Transfer Office (TTO) of Multimedia University. The research ethics approval for the project has been granted with the approval number EA0052021 on 26 February 2021, and the approval letter has been endorsed by the TTO Director-cum-REC Secretariat of the university.

Informed Consent Statement: The main researcher (main author) gave his written informed consent prior to the experiments.

Data Availability Statement: This project contains the following underlying data: Data Availability Sheet.docx (dataset used for the design drawing, optimisation, modifications, data collections and limitations). The data can be found at Figshare (doi:10.6084/m9.figshare.14994555). Data are available under the terms of the Creative Commons Attribution 4.0 International license (CC-BY 4.0).

Acknowledgments: This study was part of the main author’s final year project for his B.Eng. (Hons) Mechanical degree. The researchers would like to thank the Faculty of Engineering and Technology, Multimedia University (MMU), for allowing this study to be carried out. The researchers would also like to thank Norhidayah Mohamad for providing some useful comments during the main author’s project presentation. Special thanks go to the Research Ethics Committee and Technology Transfer Office members of MMU for granting the researchers with ethical clearance to conduct this study. The researchers are exceptionally grateful to Chiew Fen Ng for her constructive criticism of the manuscript. Finally, the researchers also thank the anonymous reviewers for their valuable comments.

Conflicts of Interest: No potential conflict of interest was reported by the authors.

References

1. Khairadini, F.; Sunarti, T.C.; Anggraeni, E. The role of innovation in the development of melinjo chips agro-industry in Aceh. *IOP Conf. Series Earth Environ. Sci.* **2020**, *472*. [CrossRef]
2. Sasha, F.; Charabsarn, A.; Ahmad, N.H.; Mohaidin, Z. Driving Food and Beverage SMEs Innovation. *Adv. Econ. Bus. Manag. Res.* **2020**, *17*–19. [CrossRef]
3. Wahid, Z.; Daud, M.R.C. Study on productivity improvement of manual operations in soya sauce factory. *IJUM Eng. J.* **2020**, *21*, 202–211. [CrossRef]
4. Department of Statistics Malaysia. Small and Medium Enterprises (SMES) Performance 2019. Available online: <https://bit.ly/3wqmZbx> (accessed on 14 June 2021).
5. Ahmad, S.; Wong, K.Y. Development of weighted triple-bottom line sustainability indicators for the Malaysian food manufacturing industry using the Delphi method. *J. Clean. Prod.* **2019**, *229*, 1167–1182. [CrossRef]
6. Fassio, F.; Tecco, N. Circular Economy for Food: A Systemic Interpretation of 40 Case Histories in the Food System in Their Relationships with SDGs. *Systems* **2019**, *7*, 43. [CrossRef]
7. Issa, I.; Delbrück, S.; Hamm, U. Bioeconomy from experts’ perspectives—Results of a global expert survey. *PLoS ONE* **2019**, *14*, e0215917. [CrossRef] [PubMed]
8. Shapiro, S. A Meal with Meaning. Available online: <https://www.baltimoresun.com/news/bs-xpm-2003-04-16-0304150420-story.html> (accessed on 23 November 2020).
9. Kaur, B. Consumer Preference for Goat Meat in Malaysia: Market Opportunities and Potential. *J. Agribus. Mark.* **2010**, *3*, 40–55.
10. Athan, L.L. How to Roast a Whole Lamb on a Spit. Available online: <https://www.thespruceeats.com/lamb-on-a-spit-1705925> (accessed on 1 January 2021).
11. Robertson, N. Top 10 Best Barbecue Rotisseries in March 2021. Available online: <https://homediy.guide/barbecue-rotisseries/#> (accessed on 24 March 2021).
12. Rotisol. Special Market—The Chicken Rotisserie Machine for Markets and Trucks. Available online: <https://www.rotisol.com/en/rotisserie-special-market/> (accessed on 1 March 2021).
13. Bagheri, H. Application of Infrared Heating for Roasting Nuts. *J. Food Qual.* **2020**, *2020*, 1–10. [CrossRef]
14. Alfaro, D. The Pros and Cons of Infrared Grills: Is Infrared Good or Bad? Available online: <https://www.thespruceeats.com/infrared-good-or-gimmick-3971847> (accessed on 24 March 2021).
15. Simon, M. Charcoal Grilling Tastes Better Than Gas. It’s Just Science. Available online: <https://www.wired.com/2016/07/charcoal-grilling-tastes-better-gas-just-science/> (accessed on 24 March 2021).
16. National Center for Biotechnology Information. PubChem Compound Summary for CID 460, Guaiacol. Available online: <https://pubchem.ncbi.nlm.nih.gov/compound/Guaiacol> (accessed on 24 March 2021).
17. Ullman, D.G. *The Mechanical Design Process, 4th ed*; McGraw-Hill: New York, NY, USA, 2010.
18. Nagi, G.K. Fundamentals of Machine Design and Manufacturing: Design of a Compliant Winding Machine. Master’s Thesis, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, 2014.

19. Nokelainen, P.; Nevalainen, T.; Niemi, K. *Mind or Machine? Opportunities and Limits of Automation*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 13–24. [CrossRef]
20. Vujkov, Z. How to Make a Spit Roast. Available online: <https://ultimatespitroastmotors.com/spit-roast-guide> (accessed on 19 September 2020).
21. Raichlen, S. The Rotisserie Reignites. Available online: <https://www.nytimes.com/2001/07/04/dining/the-rotisserie-reignites.html#:~:text=Rotisserie> (accessed on 25 March 2021).
22. Schinto, J. The Clockwork Roasting Jack, or How Technology Entered the Kitchen. *Gastronomy* **2004**, *4*, 33–40. [CrossRef]
23. Yonntech HappyHome Store. Yonntech Electric Grill Stainless Steel Keba BBQ Spit Roaster Rotisserie Lamb Pig Grilling Outdoor Party Garden Barbecue. Available online: <https://bit.ly/2Px1N3e> (accessed on 3 January 2021).
24. Dewangan, A.K.; Patel, A.D.; Bhadania, A.G. Stainless Steel for Dairy and Food Industry: A Review. *J. Mater. Sci. Eng.* **2015**, *4*. [CrossRef]
25. Pig Lamb Rotisserie Shop. XL Regular Size Pig Lamb Rotisserie. Available online: <https://pig-lamb-rotisserie.com/product/xl-regular-size-pig-lamb-rotisserie/> (accessed on 6 January 2021).
26. Stier, R.J. Charcoal Barbeque Rotisserie Grill Cooker. U.S. Patent US8677891B2, 2 May 2013.
27. Stewart, K.E.; Erwin, T.H. Charcoal Grill. U.S. Patent US3209743A, 5 October 1965.
28. Gavin, J. Roasting 101. Available online: <https://www.jessicagavin.com/roasting/> (accessed on 26 March 2021).
29. Mottram, D. *Meat Flavour*, 1st ed.; Springer: Boston, MA, USA, 1994; pp. 141–143.
30. Griffin, G.L.; Smith, J.W.S.G.C.; Rhee, K.S.; Johnson, H.K. Cooking times, cooking losses and energy used for cooking lamb roasts. *J. Food Qual.* **1985**, *8*, 69–79. [CrossRef]
31. Wharton, M.D. *Internal Color and Tenderness of the Infraspinatus, Longissimus Thoracis, and Semimembranosus Are Affected by Cooking Method and Degree of Doneness*, 1st ed.; Proquest, Umi Dissertation Publishing: Ann Arbor, MI, USA, 2011.
32. Murugesan, M. Covid-19: Ensuring Food Safety during Ramadan. Available online: <https://www.nst.com.my/news/nation/2020/05/590259/covid-19-ensuring-food-safety-during-ramadan> (accessed on 23 July 2021).
33. Gargi, A.; Sengun, I.Y. Marination liquids enriched with probiotics and their inactivation effects against food-borne pathogens inoculated on meat. *Meat Sci.* **2021**, *182*, 108624. [CrossRef] [PubMed]
34. Ooi, Y. Tips on How to Ensure the Food You Buy is Safe. Available online: <https://www.thestar.com.my/lifestyle/health/2021/06/11/tips-on-how-to-ensure-the-food-you-buy-is-safe> (accessed on 23 July 2021).
35. Li, K.; McKeith, A.G.; Shen, C.; McKeith, R. A Comparison Study of Quality Attributes of Ground Beef and Veal Patties and Thermal Inactivation of *Escherichia coli* O157:H7 after Double Pan-Broiling Under Dynamic Conditions. *Foods* **2017**, *7*, 1. [CrossRef] [PubMed]
36. Hunt, M.; King, A.; Barbut, S.; Claus, J.; Cornforth, D.; Hanson, D.; Lindahl, G.; Mancini, R.; Milkowski, A.; Mohan, A.; et al. *AMSA Meat Color Measurement Guidelines*, 1st ed.; American Meat Science Association: Illinois, IL, USA, 2012.
37. Taste of Home Editors. Our Test Kitchen's Guide to Food-Safe Cooking Temperatures. Available online: <https://www.tasteofhome.com/article/food-safe-cooking-temperatures/> (accessed on 27 March 2021).
38. Kang, C.; Ng, P.; Liew, K. The Conceptual Synthesis and Development of a Multifunctional Lawnmower. *Inventions* **2021**, *6*, 38. [CrossRef]
39. Lim, S.; Ng, P. The Design and Development of a Foldable Wheelchair Stretcher. *Inventions* **2021**, *6*, 35. [CrossRef]
40. Tan, D.W.H.; Ng, P.K.; Noor, E.E.M. A TRIZ-driven conceptualisation of finger grip enhancer designs for the elderly. *F1000Research* **2021**, *10*, 392. [CrossRef] [PubMed]
41. Done, B. Stainless Steel in Food Manufacturing: Grade Selection and Care. Available online: <https://www.manufacturing.net/operations/article/13163098/stainless-steel-in-food-manufacturing-grade-selection-and-care> (accessed on 23 April 2021).
42. Munoz, G. How Hot Is a Bonfire? Available online: <https://sciencing.com/hot-bonfire-8770.html> (accessed on 23 April 2021).
43. Helmenstine, A.M. Oxidation Definition and Example in Chemistry. Available online: <https://bit.ly/3gE5TSF> (accessed on 23 April 2021).
44. Vanmeensel, K.; Lietaert, K.; Vrancken, B.; Dadbakhsh, S.; Li, X.; Kruth, J.-P.; Krakhmalev, P.; Yadroitsev, I.; Van Humbeeck, J. Additively manufactured metals for medical applications. *Add. Manufact.* **2018**, *8*, 261–309. [CrossRef]
45. US FDA. *Food Code: US Public Health Service*; US Department of Health and Human Services: College Park, MD, USA, 2017.
46. Autobotic. 12 V 225 RPM 1.3 kgfcm Brushed DC Geared Motor (SPG30-20K). Available online: <https://www.autobotic.com.my/cytron-technologies/dc-geared-motor-spg30-20k> (accessed on 23 February 2021).
47. ebay. Hog Roast Machine BBQ Spit Roaster Rotisserie Grill Roasting Motor Lamb Picnic. Available online: <https://www.ebay.com/itm/Hog-Roast-Machine-BBQ-Spit-Roaster-Rotisserie-Grill-Roasting-Motor-Lamb-Picnic-/174296884757> (accessed on 23 July 2021).
48. Berjaya. Stainless Steel Charcoal Lamb Roaster. Available online: <https://www.berjayasteel.com/products/stainless-steel-charcoal-lamb-roaster> (accessed on 23 July 2021).
49. Trading Economics. Malaysia Minimum Monthly Wages. Available online: <https://tradingeconomics.com/malaysia/minimum-wages> (accessed on 15 June 2021).
50. Miller, R. Drivers of Consumer Liking for Beef, Pork, and Lamb: A Review. *Foods* **2020**, *9*, 428. [CrossRef] [PubMed]

51. Hutchings, S.; Guerrero, L.; Miroso, M.; Bremer, P.; Mather, D.; Pavan, E.; Hicks, T.; Day, L.; Realini, C. The Implications of COVID-19 on Chinese Consumer Preferences for Lamb Meat. *Foods* **2021**, *10*, 1324. [CrossRef] [PubMed]
52. Rabadán, A.; Martínez-Carrasco, L.; Brugarolas, M.; De Vera, C.N.-R.; Sayas-Barberá, E.; Bernabéu, R. Differences in Consumer Preferences for Lamb Meat before and during the Economic Crisis in Spain. Analysis and Perspectives. *Foods* **2020**, *9*, 696. [CrossRef] [PubMed]

Author Index

A

Autiero M. 4

B

-

C

Cera M. 4

Chen G. 59

Chia X.W. 200

Cinti T. 77

Cirelli M. 4

D

Di Gregorio R. 1, 77, 147

Du R. 168

E

Erdman A.G. 181

F

Fu Y. 168

G

Gatti M. 147

Gu J. 168

H

Haug E.J. 37

Huss J.M. 181

I

-

J

-

K

Kong X. 117

L

Lehman M. 181

Li Z. 59

Lim W.S. 200

Liu Y. 59

Liu Xin 59

Liu Xuan 133

M

-

N

Nathan R.J. 200

Ng P.K. 200

Ng Y.J. 200

Nie D. 168

O

-

P

Peidro A. 37

Pennestri E. 4

Q

-

R

-

S

Shen F. 168

Simas H. 147

Simoni R. 147

T

Tang L. 133

Tian J. 168

U

-

V

Valentini P.P. 4

W

Wang J. 117

Wang M. 133

Wu L. 133

X

Xu W. 59

Y

Yeow J.A. 200

Z

Zhao D. 133

Zhang P. 168

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Machines Editorial Office
E-mail: machines@mdpi.com
www.mdpi.com/journal/machines



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Collection Editor. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editor and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Academic Open
Access Publishing

[mdpi.com](https://www.mdpi.com)

ISBN 978-3-7258-2966-8