Mechanisms and Machine Science 15

Federico Thomas
Alba Pérez Gracia *Editors*

Computational Kinematics

Proceedings of the 6th International Workshop on Computational Kinematics (CK2013)

Springer

Mechanisms and Machine Science

Volume 15

Series Editor

Marco Ceccarelli

For further volumes:
http://www.springer.com/series/8779

Federico Thomas · Alba Pérez Gracia
Editors

Computational Kinematics

Proceedings of the 6th International Workshop on Computational Kinematics (CK2013)

Springer

Editors
Federico Thomas
Universitat Politècnica de Catalunya
Barcelona
Spain

Alba Pérez Gracia
Mechanical Engineering
Idaho State University
Pocatello, ID
USA

ISSN 2211-0984 ISSN 2211-0992 (electronic)
ISBN 978-94-007-7213-7 ISBN 978-94-007-7214-4 (eBook)
DOI 10.1007/978-94-007-7214-4
Springer Dordrecht Heidelberg New York London

Library of Congress Control Number: 2013943006

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book contains the *Proceedings of the 6th International Workshop on Computational Kinematics (CK2013)* sponsored by IFToMM, the International Federation for the Promotion of Mechanism and Machine Science.

This time this workshop comes to Barcelona, taking place from 12 to 15, May 2013. In this occasion, 73 papers have been received. After peer-reviewed evaluation, 44 papers, from 21 different countries, have been accepted for presentation. The community of kinematicians, thus, continues to exhibit its traditional vitality.

The reader will find here a representative sample of the most modern techniques available nowadays for the solution of challenging problems arising in computational kinematics. In light of its contents, this book should be of interest to researchers, graduate students, and practicing engineers working in kinematics or related areas. Especially, roboticists, biomechanicists, machine designers, and computer scientists will find here a useful source of information comprising methods, algorithms, and applications.

In the nineteenth century, Pafnuty Chebyshev, who was a great enthusiast for kinematics, said that Kinematics is more fruitful than Geometry because it adds a fourth dimension of time to the three-dimensional space. The researchers gathering in Barcelona on occasion of CK2013 will continue to make this truth, and to show how their results have an important influence in many different domains that range from Robotics to Proteomics.

We thank the authors for the valuable contributions, as well as the 45 reviewers for having completed their work in a timely manner. Last, but not least, we thank Ana Canales and Victor Vilchez for taking care of all the organizational details, and Josep M. Porta for his help during the painstaking labor of the final typesetting of this book.

Barcelona, Spain Federico Thomas
Pocatello, USA Alba Pérez Gracia

Contents

On the Minimum 2-Norm Positive Tension for Wire-Actuated Parallel Manipulators

Leila Notash

Abstract Utilizing the Moore-Penrose generalized inverse of the Jacobian matrix of wire-actuated parallel robot manipulators, one or more wire tensions could be negative. In this paper, a methodology for calculating positive wire tensions, with minimum 2-norm for tension vector, is presented. A planar parallel manipulator is simulated to illustrate the proposed methodology.

Keywords Cable-driven robots · Positive wire tension · Minimum 2-norm solution

1 Introduction

In wire/cable-actuated parallel robot manipulators, also known as wire/cable-driven parallel manipulators, the motion of mobile platform (end effector) is constrained by wires/cables. Because wires act in tension and cannot exert forces in both directions along their lines of action, i.e., their inputs are unidirectional and irreversible, to fully constrain an m degrees of freedom (DOF) rigid body suspended by wires, in the absence of gravity and external force/moment (wrench), the number of wires should be larger than the DOF of manipulator (Fig. 1), i.e., $n \geq m + 1$.

The manipulator failure could be defined as any event that affects its performance such that the manipulator cannot complete its task as required. Wire-actuated manipulators could fail because of the hardware and/or software failures, including failure of a wire, sensor, actuator, or transmission mechanism; as well as computational failure. These failures could result in the loss of DOF, actuation, motion constraint, and information [1]. From the force point of view, the failure of manipulator occurs if the wire does not provide the required force/torque, e.g., when the actuator force/torque is lost partially or fully or the actuator is saturated. This could also happen when

L. Notash (✉)
Institute of Complex Systems, Queen's University, Kingston, ON, Canada
e-mail: notash@me.queensu.ca

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_1,

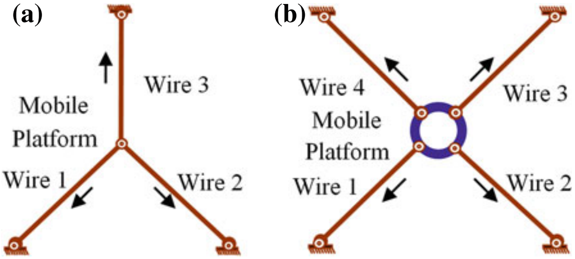


Fig. 1 Planar wire-actuated manipulators **a** 2 DOF; and **b** 3 DOF

the wire is broken or slack (zero tension), wire is jammed (constant length), or its actuating mechanism malfunctions such that a different wire force is provided.

For a given wrench to be applied/resisted by the mobile platform of wire-actuated parallel manipulators, because $n \geq m + 1$, there are infinite solutions for the wire tension vector. The minimum 2-norm solution could result in negative tension for wires, which is not acceptable, and generally the homogeneous solution is used to adjust the tension to positive values if the platform position and orientation (pose) is within the wrench closure workspace, e.g., [2]. Due to space limitation, a review of pertinent literature is not included.

In this paper, formulation of a non-negative wire tension vector for wire-actuated parallel manipulators is investigated when the DOF of manipulator m is one less than the number of wires n, i.e., $n = m + 1$. In Sect. 2, the implementation of the methodology of [3] for achieving minimum 2-norm positive wire tension vector is presented. Simulation results are reported in Sect. 3. The article concludes with Sect. 4.

2 Wrench Recovery for Negative Wire Tension

For the n-wire-actuated parallel manipulators, the $n \times 1$ vector of wire forces $\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \cdots & \tau_n \end{bmatrix}^T$ is related to the $m \times 1$ vector of forces and moments (wrench) $\mathbf{F}$ applied by the platform with the $m \times n$ transposed Jacobian matrix $\mathbf{J}^T$ as

$$\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau} = \begin{bmatrix} \mathbf{J}_1^T \; \mathbf{J}_2^T & \cdots & \mathbf{J}_i^T & \cdots & \mathbf{J}_{n-1}^T \; \mathbf{J}_n^T \end{bmatrix} \boldsymbol{\tau} = \sum_{j=1}^{n} \mathbf{J}_j^T \tau_j \tag{1}$$

where $m \leq 6$ depending on the dimension of task space. Column j of $\mathbf{J}^T$, $\mathbf{J}_j^T$, is a zero pitch screw corresponding to the wrench applied on the platform by the jth wire/actuator. The solution of $\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}$ for the wire tensions is

$$\boldsymbol{\tau} = \boldsymbol{\tau}_p + \boldsymbol{\tau}_h = \mathbf{J}^{\#T} \mathbf{F} + \left(\mathbf{I} - \mathbf{J}^{\#T} \mathbf{J}^T \right) \mathbf{k} = \mathbf{J}^{\#T} \mathbf{F} + \mathbf{N} \boldsymbol{\lambda} \tag{2}$$

where $\mathbf{J}^{\#T}$ is the Moore-Penrose generalized inverse of $\mathbf{J}^T$, $\boldsymbol{\tau}_p = \mathbf{J}^{\#T}\mathbf{F}$ is the minimum 2-norm (particular) solution, and $\boldsymbol{\tau}_h = \left(\mathbf{I} - \mathbf{J}^{\#T}\mathbf{J}^T\right)\mathbf{k}$ and $\boldsymbol{\tau}_h = \mathbf{N}\,\boldsymbol{\lambda}$ represent the homogeneous solution. $\left(\mathbf{I} - \mathbf{J}^{\#T}\mathbf{J}^T\right)\mathbf{k}$ is the projection of an $n \times 1$ vector $\mathbf{k}$ onto the null space of $\mathbf{J}^T$. Columns of the $n \times (n - m)$ matrix $\mathbf{N}$ correspond to the orthonormal basis of the null space of $\mathbf{J}^T$, referred here as the null space vectors, and $\boldsymbol{\lambda}$ is an $(n - m)$-vector. When one or more entries of $\boldsymbol{\tau}_p = \mathbf{J}^{\#T}\mathbf{F}$ are negative the wire tensions could be adjusted by identifying the correctional tension $\boldsymbol{\tau}_h$ that would set all the wire tensions to positive values provided the manipulator pose is in the wrench closure workspace. The adjusted wire tensions should satisfy the tension limits $\tau_{\min} \leq \tau_{pl} + \tau_{hl} = \tau_{pl} + n_l\lambda \leq \tau_{\max}$, for $l = 1, \ldots, n$, where the entries of $\boldsymbol{\tau}_h$ corresponding to negative τ_p must be positive in order to have non-negative tension vector after adjustment.

2.1 Conditions for Non-negative Wire Tension

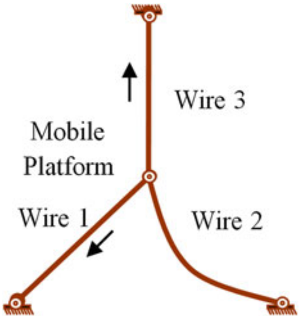
When the platform pose is in the wrench closure workspace of manipulator a non-negative solution to $\boldsymbol{\tau} = \boldsymbol{\tau}_p + \boldsymbol{\tau}_h = \mathbf{J}^{\#T}\mathbf{F} + \mathbf{N}\,\boldsymbol{\lambda}$ exists. The criteria for non-negative wire tension could be defined based on the orthonormal basis of the null space of the $m \times n$ transposed Jacobian matrix $\mathbf{J}^T$ and of the $m \times (n + t)$ augmented transposed Jacobian matrix $\mathbf{J}^T_{aug}$ of $\mathbf{J}^T_{aug}\boldsymbol{\tau}_{aug} = \left[\mathbf{J}^T_1\ \mathbf{J}^T_2\ \cdots\ \mathbf{J}^T_{n-1}\ \mathbf{J}^T_n\ \mathbf{W}^T_1 \cdots \mathbf{W}^T_t\right]\boldsymbol{\tau}_{aug} = \mathbf{0}$. $\mathbf{J}^T_{aug}$ and $\boldsymbol{\tau}_{aug}$ are formed by re-writing Eq. (1) as $\mathbf{J}^T\boldsymbol{\tau} - \mathbf{F} = \mathbf{0}$ and augmenting $\mathbf{J}^T$ and $\boldsymbol{\tau}$ respectively with t wrenches and t number of 1's corresponding to t non-zero components of $\mathbf{F}$, where $t \leq m \leq 6$. $\mathbf{W}^T$ is a wrench with zeros for all entries except for the one that is equal to the negative of corresponding non-zero entry of $\mathbf{F}$.

The orthonormal basis of the null space of the $m \times n$ transposed Jacobian matrix $\mathbf{J}^T$, with full-row rank, is defined by $n - m$ number of n-vectors, i.e., the dimension of the null space vectors of $\mathbf{J}^T$ is $n \times 1$. The sufficient condition for rectifying the negative tension of particular solution to positive tension is the existence of a null space vector $\mathbf{n}$ of $\mathbf{J}^T$ with all positive entries, e.g., refer to [2]. In the presence of external wrench, even if there is no null space vector of $\mathbf{J}^T$ with consistent sign, positive wire tension is feasible if there exist a null space vector $\mathbf{n}_{aug}$ of the augmented Jacobian matrix $\mathbf{J}^T_{aug}$ with non-negative values for the first n entries (corresponding to wires) and positive values for the last t entries (corresponding to non-zero components of external wrench). Detailed discussion on the conditions for positive tension in wires are presented in [3].

2.2 Methodology for Adjusting Negative Wire Tension

When $n = m + 1$ and the pose is in the wrench closure workspace, if the minimum norm solution results in negative tension for wire i, i.e., $\tau_{pi} < 0$, wire i could be

Fig. 2 Three-wire-actuated parallel manipulator with slack wire 2



considered as "failed" and its tension should be set to a non-negative value τ_{ci}. If wire i is left as slack (Fig. 2) $\tau_{ci} = 0$. To increase the wrench capability and stiffness of manipulators, the tension of wire i could be adjusted to a positive value, $\tau_{ci} > 0$. Rewriting Eq. (2) for wire i

$$\tau_{pi} + \tau_{hi} = \tau_{pi} + n_i\lambda = \tau_{ci} \geq 0 \tag{3}$$

or $n_i\lambda = \tau_{ci} + \left|\tau_{pi}\right|$, where n_i corresponds to entry i of the null space vector $\mathbf{n}$ and $\left|\tau_{pi}\right|$ is the magnitude (absolute value) of τ_{pi}. Then, the platform wrench becomes

$$\mathbf{F}_f = \left[\mathbf{J}_1^T\ \mathbf{J}_2^T\ \cdots\ \mathbf{J}_i^T\ \cdots\ \mathbf{J}_{n-1}^T\ \mathbf{J}_n^T\right]\boldsymbol{\tau}_f = \sum_{j=1}^{n}\mathbf{J}_j^T\tau_{pj} - \mathbf{J}_i^T(\tau_{pi} - \tau_{ci}) \tag{4}$$

where $\boldsymbol{\tau}_f = [\tau_{p1}\ \tau_{p2}\ \cdots\ \tau_{ci}\ \cdots\ \tau_{p\,n-1}\ \tau_{p\,n}]^T$ and the change in tension of wire i after adjusting its negative value is $\left|\tau_{pi} - \tau_{ci}\right|$. To provide the platform wrench $\mathbf{F}$, the remaining wires must balance the wrench corresponding to the adjusted negative wire tension. With the "correctional" force provided by the remaining wires $\boldsymbol{\tau}_{corr} = [\tau_{corr\,1}\ \tau_{corr\,2}\ \ldots\ 0\ \ldots\ \tau_{corr\,n-1}\ \tau_{corr\,n}]^T$, the recovered wrench will be

$$\mathbf{F}_r = \mathbf{J}^T\boldsymbol{\tau}_f + \mathbf{J}_f^T\boldsymbol{\tau}_{corr} \tag{5}$$

where column i of $\mathbf{J}_f^T = \left[\mathbf{J}_1^T\ \mathbf{J}_2^T\ \cdots\ \mathbf{0}\ \cdots\ \mathbf{J}_{n-1}^T\ \mathbf{J}_n^T\right]$ and entry i of $\boldsymbol{\tau}_{corr}$ are replaced by zeros. Then, the change in the platform wrench will be $\mathbf{F} - \mathbf{F}_r = \mathbf{J}^T(\boldsymbol{\tau}_p - \boldsymbol{\tau}_f) - \mathbf{J}_f^T\boldsymbol{\tau}_{corr}$. When the minimum 2-norm solution results in negative tension for g wires, after adjusting the negative tensions to positive values, the platform wrench that should be balanced by the remaining wires is $\sum_g \mathbf{J}_i^T(\tau_{pi} - \tau_{ci}) = \mathbf{J}^T(\boldsymbol{\tau}_p - \boldsymbol{\tau}_f)$, where the summation is taken over the wires with negative tension.

To fully compensate for the adjusted negative tensions, i.e., for $\mathbf{F} - \mathbf{F}_r = \mathbf{0}$, the correctional force provided by the remaining wires should be [3]

$$\boldsymbol{\tau}_{corr} = \mathbf{J}_f^{\#T} \sum \mathbf{J}_i^T (\tau_{pi} - \tau_{ci}) = \mathbf{J}_f^{\#T} \mathbf{J}^T (\boldsymbol{\tau}_p - \boldsymbol{\tau}_f) \tag{6}$$

where g columns of $\mathbf{J}^T$, corresponding to the wires with negative tension, are replaced by zeros resulting in $\mathbf{J}_f^T$. Then, the overall wire force will be

$$\boldsymbol{\tau}_{tot} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_{corr} = \mathbf{J}_f^{\#T} \mathbf{J}^T \boldsymbol{\tau}_p + (\mathbf{I} - \mathbf{J}_f^{\#T} \mathbf{J}^T) \boldsymbol{\tau}_f \tag{7}$$

These $\boldsymbol{\tau}_{corr}$ and $\boldsymbol{\tau}_{tot}$ are minimum 2-norm solutions for the chosen τ_{ci} [4]. If $\mathbf{J}_f^T$ has full row-rank, i.e., $\mathbf{F}$ belongs to the range space of $\mathbf{J}_f^T$, $\mathbf{F} \in \Re(\mathbf{J}_f^T)$, $\mathbf{F}_{\Re^\perp} = \left(\mathbf{I} - \mathbf{J}_f^T \mathbf{J}_f^{\#T}\right) \mathbf{F} = \mathbf{0}$, the right-generalized inverse (GI) of $\mathbf{J}_f^T$ is $\mathbf{J}_f^{\#T} = \mathbf{J}_f \left(\mathbf{J}_f^T \mathbf{J}_f\right)^{-1}$ as the vector of wire forces is physically consistent. Otherwise, the weighted left-GI of $\mathbf{J}_f^T$ is used to obtain the platform wrench that best approximates the required wrench in the least-square sense.

When the minimum norm solution results in negative tension for g wires and the pose is in the wrench closure workspace $\tau_{pi} + \tau_{hi} = \tau_{ci} \geq \tau_{\min} \geq 0$ for each of g wires, where $\tau_{\min}$ is the minimum allowable tension. In the following subsections, formulations of τ_{ci} are presented for $n = m + 1$. The implementation of this methodology for adjusting negative wire tensions when the null space basis of $\mathbf{J}^T$ is spanned by two or more vectors, i.e., when $n > m + 1$, is presented in [5].

2.3 Minimum 2-Norm with Negative Wire Tension

Negative tension for one wire. When the pose is in the wrench closure workspace and the minimum norm solution results in negative tension for wire i, $\tau_{pi} < 0$, considering the non-negative null space vector $\mathbf{n}$, $\tau_{pi} + n_i \lambda = \tau_{ci} \geq \tau_{\min}$, there is no condition on τ_{ci} provided the adjusted wire tensions do not exceed the maximum value. The minimum 2-norm solutions for the correctional and overall wire tension vectors are calculated using Eqs. (6) and (7) for the chosen $\tau_{ci} \geq \tau_{\min}$ value.

For the poses that $\mathbf{n}$ has both positive and negative entries, when $\tau_{pi} < 0$ all wire tensions could be adjusted to non-negative values if

$$\frac{\tau_{\min} + \left|\tau_{pi}\right|}{n_i} \leq \frac{\tau_{\min} - \tau_{pj}}{n_j} \qquad \text{for} \quad n_i > 0 \quad \text{and} \quad n_j < 0 \tag{8}$$

with $\tau_{pj} > 0$ for $j \neq i$. That is, when the condition on $\mathbf{n}_{aug}$ is met and $\tau_{pi} < 0$ the wire tensions could be set to positive values using $\tau_{ci} \geq \tau_{\min}$.

Negative tension for two or more wires. When the pose is in the wrench closure workspace and the minimum norm solution results in negative tension for wires i and j the non-negative values for these two wires, i.e., $\tau_{ci} \geq \tau_{\min}$ and $\tau_{cj} \geq \tau_{\min}$, cannot be selected arbitrarily. Using a non-negative null space vector $\mathbf{n}$

$$\begin{aligned} \lambda_i &\geq \tfrac{\tau_{\min}-\tau_{pi}}{n_i} \qquad \text{for} \quad n_i > 0 \\ \lambda_j &\geq \tfrac{\tau_{\min}-\tau_{pj}}{n_j} \qquad \text{for} \quad n_j > 0 \end{aligned} \tag{9}$$

The largest of λ_i and λ_j corresponds to the dominating wire and its τ_c is set to $\tau_{\min}$. The adjusted tension of non-dominating wire, e.g., wire j, is calculated using $\tau_{cj} = \tau_{pj} + n_j\lambda_{dw} > \tau_{\min}$. Equation (9) is also valid when $\mathbf{n}$ has both positive and negative entries with $n_i > 0$ and $n_j > 0$. It should be noted that when n_i and n_j have opposite signs, e.g., $n_i > 0$ and $n_j < 0$, the pose is not in the wrench closure workspace as $\lambda_i > 0$ and $\lambda_j < 0$. When $n_i = n_j$, the dominating wire l_{dw} is the wire corresponding to max $(|\tau_{pi}|, |\tau_{pj}|)$. Then, the minimum 2-norm solutions for the correctional and overall wire tension vectors are calculated using Eqs. (6) and (7) for the chosen τ_c values.

The dominating wire can also be identified using $Q = (\tau_{pi}n_j - \tau_{pj}n_i)/(n_j - n_i)$. For example, when $n_i \neq n_j$, provided $\tau_{pj} + n_j\lambda_{dw} > \tau_{\min}$

$$Q = \frac{\tau_{pi}n_j - \tau_{pj}n_i}{n_j - n_i} \leq \tau_{\min} \quad \Rightarrow \quad l_{dw} = i, \quad \tau_{ci} = \tau_{\min} \tag{10}$$

The procedure could be generalized for the case that the minimum 2-norm solution results in negative tension for g wires to identify the dominating wire as

$$\lambda_{dw} = \max\,(\lambda_i, \lambda_j, \ldots) = \max\left(\frac{\tau_{\min} + |\tau_{pi}|}{n_i}, \frac{\tau_{\min} + |\tau_{pj}|}{n_j}, \ldots\right) \tag{11}$$

Minimum norm positive tension within upper limit. To ensure the adjusted non-negative wire tensions do not exceed the maximum allowable tension, i.e., $\tau_{pl} + n_l\lambda_{dw} \leq \tau_{\max}$, for $l = 1, \ldots, n$, the following conditions should be satisfied.

When the entries of the null space vector $\mathbf{n}$ have consistent signs (non-negative), while adjusting the negative tension of wires, e.g., wire i with $\tau_{pi} + n_i\lambda \geq \tau_{\min}$, the tension of wires with positive particular solution will be increased (will remain unchanged if the corresponding entry of $\mathbf{n}$ is zero). When wire k has the smallest $\lambda_k = (\tau_{\max} - \tau_{pk})/n_k$ among all wires with positive particular solution the adjusted tension of wires will not exceed the limit as long as

$$0 < \frac{\tau_{\min} + |\tau_{pi}|}{n_i} \leq \frac{\tau_{\max} - \tau_{pk}}{n_k} \qquad \text{for} \quad n_i > 0 \quad \text{and} \quad n_k > 0 \tag{12}$$

When more than one wire has negative tension, in Eq. (12) wire i corresponds to the dominating wire of Eq. (11). Hence, the sufficient condition for exceeding the upper limit is $\tau_{pk} \geq \tau_{\max}$, while the necessary condition is $\tau_{pk} > \tau_{\max} - n_k\lambda_{dw}$.

When n_i and n_k have opposite signs, e.g., $n_i > 0$ and $n_k < 0$, for $\tau_{pk} + n_k\lambda_{dw} \leq \tau_{\max}$, the limit on maximum tension is satisfied as long as $\tau_{pk} \leq \tau_{\max} + |n_k|\,\lambda_{dw}$.

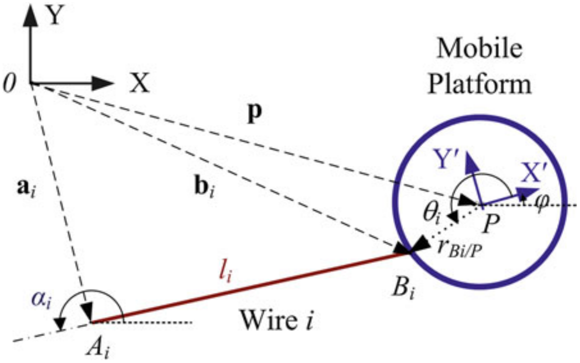


Fig. 3 Parameters of planar wire-actuated parallel manipulators

3 Case Study

In the planar wire-actuated parallel manipulators, the mobile platform is connected to the base by n wires, each wire with a length of l_i and orientation of α_i (Fig. 3). The attachment points of wire i to the base and platform are denoted as points A_i and B_i, respectively. The angular positions of points B_i on the platform are denoted by θ_i. For a 2 DOF translational manipulator with three wires, the coordinates of A_i, $i = 1, \ldots, 3$, are (–2, –1.5), (2, –1.5) and (0, 1.5), respectively, and points B_i coalesce. At the platform pose of $\mathbf{p} = [0.5\ -0.5]^T$ meters, which is in the wrench closure workspace, the wire forces $\boldsymbol{\tau}$ are related to the platform wrench $\mathbf{F}$ using $\mathbf{F} = \mathbf{J}^T\boldsymbol{\tau}$, where

$$\mathbf{J}^T = \begin{bmatrix} \cos\alpha_1 & \cos\alpha_2 & \cos\alpha_3 \\ \sin\alpha_1 & \sin\alpha_2 & \sin\alpha_3 \end{bmatrix} = \begin{bmatrix} -0.929 & 0.832 & -0.243 \\ -0.371 & -0.555 & 0.970 \end{bmatrix} \tag{13}$$

For $\mathbf{F} = [-3.309\ 14.737]^T$ N, the minimum norm vector of wire forces is $\boldsymbol{\tau}_p = \mathbf{J}^{\#T}\mathbf{F} = [-4.236\ -5.699\ 10.310]^T$ with a magnitude of $\|\boldsymbol{\tau}_p\|_2 = 12.519$ and negative tension for wires 1 and 2. Then, the 2×5 $\mathbf{J}^T_{aug}$ is

$$\mathbf{J}^T_{aug} = \begin{bmatrix} \mathbf{J}^T_1\ \mathbf{J}^T_2\ \mathbf{J}^T_3\ \mathbf{W}^T_1 \mathbf{W}^T_2 \end{bmatrix} = \begin{bmatrix} -0.929 & 0.832 & -0.243 & 3.309 & 0 \\ -0.371 & -0.555 & 0.970 & 0 & -14.737 \end{bmatrix} \tag{14}$$

A non-negative null space vector of $\mathbf{J}^T_{aug}$ is $\mathbf{n}_{aug} = [0.413\ 0.576\ 9.981\ 0.703\ 0.625]^T$. The non-negative null space vector of $\mathbf{J}^T$ is $\mathbf{n} = [0.463\ 0.682\ 0.567]^T$, with non-zero entries corresponding to wires 1 and 2. As $\frac{\tau_{p1}n_2 - \tau_{p2}n_1}{n_2 - n_1} = -1.143 < \tau_{\min} = 1$, wire 1 is the dominating wire and its tension is adjusted to $\tau_{c1} = 1$ N and the tension of wire 2 is calculated as $\tau_{c2} = 2.014$ N. These adjusted tensions correspond to $\lambda = \max(\lambda_1 = 11.315, \lambda_2 = 9.829)$. Then

$$\boldsymbol{\tau}_{corr} = \mathbf{J}^{\#T}_f \sum \mathbf{J}^T_i (\tau_{pi} - \tau_{ci}) = \begin{bmatrix} 0\ 0\ 6.414 \end{bmatrix} \tag{15}$$

$$\boldsymbol{\tau}_{tot} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_{corr} = [1.000\ \ 2.014\ \ 16.724]^T \tag{16}$$

which produces the original wrench, and $\|\boldsymbol{\tau}_{corr}\|_2 = 6.414$, $\|\boldsymbol{\tau}_{tot}\|_2 = 16.875$.

At the platform pose of $\mathbf{p} = [2\quad 0]^T$ m, wire 2 is in Y direction, and for $\mathbf{F} = [-17.363\ 2.489]^T$ N, the minimum norm vector of wire forces is $\boldsymbol{\tau}_p = \mathbf{J}^{\#T}\mathbf{F} = [10.490\ \ –0.516\ \ 9.427]^T$ with a magnitude of $\left\|\boldsymbol{\tau}_p\right\|_2 = 14.113$ and negative tension for wire 2, using

$$\mathbf{J}^T = \begin{bmatrix} –0.936 & 0 & –0.800 \\ –0.351 & –1.000 & 0.600 \end{bmatrix} \tag{17}$$

The null space vector of $\mathbf{J}^T$ is $\mathbf{n} = [-0.536\ 0.565\ 0.628]^T$, with a negative entry corresponding to wire 1. Therefore, in the absence of external force this pose is not within the wrench closure workspace. A non-negative null space vector of $\mathbf{J}_{aug}^T$ is $\mathbf{n}_{aug} = [0.680\ 0.074\ 0.725\ 0.070\ 0.049]^T$, hence wire tensions could be adjusted to positive. This is also evident using condition (8)

$$\frac{\tau_{\min} + \left|\tau_{p2}\right|}{n_2} = 2.684 \leq \frac{\tau_{\min} - \tau_{p1}}{n_1} = 17.702 \tag{18}$$

The tension of wire 2 is set to $\tau_{c2} = 1$ N, which corresponds to $\lambda = 2.684$. Then

$$\boldsymbol{\tau}_{corr} = \mathbf{J}_f^{\#T}\mathbf{J}_2^T(\tau_{p2} - \tau_{c2}) = [–1.439\ \ 0\ \ 1.684]^T \tag{19}$$

$$\boldsymbol{\tau}_{tot} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_{corr} = [9.051\ \ 1.000\ \ 11.111]^T \tag{20}$$

which produces the original wrench, and $\|\boldsymbol{\tau}_{corr}\|_2 = 2.215$, $\|\boldsymbol{\tau}_{tot}\|_2 = 14.366$.

4 Conclusion

For wire-actuated parallel manipulators, the minimum 2-norm solution for the vector of wire tensions could result in negative tension for one or more wires. The negative tensions could be adjusted to positive values using the null space vector of the transposed Jacobian matrix and adding the homogeneous solution to the particular solution. In this paper, a methodology for adjusting the negative tension of the minimum 2-norm solution using the generalized inverse of the transposed Jacobian matrix was presented. The method results in minimum 2-norm positive (non-negative) solution for wire tension vector while satisfying the upper limit on tension. The implementation of the methodology was illustrated on a 2 DOF translational manipulator.

References

1. Notash, L., Huang, L.: On the design of fault tolerant parallel manipulators. Manipulators Mech. Mach. Theor. **38**(1), 85–101 (2003)
2. Roberts, R.G., Graham, T., Lippitt, T.: On the inverse kinematics, statics, and fault tolerance of cable-suspended robots. J. Robot. Syst. **15**(10), 581–597 (1998)
3. Notash, L.: Failure recovery for wrench capability of wire-actuated parallel manipulators. Robotica **30**(6), 941–950 (2012)
4. Notash, L.: Designing positive tension for wire-actuated parallel manipulators. In: Proceedings of the 19th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control (RoManSy) (2012)
5. Notash, L.: Designing positive tension for wire-actuated parallel manipulators. Adv. Mech. Robot. Des. Edu. Res. **14**, 251–263 (2013)

Further Analysis of the 2-2 Wire-Driven Parallel Crane

J.-P. Merlet

Abstract The 2-2 wire-driven parallel crane is the most simple planar parallel crane actuated by wires with two wires connected at two different points on the platform. We present original contributions on the kinematics of such robot, namely full inverse kinematics, trajectory, static and singularity analysis in the joint space.

Keywords Cable-driven robots · Kinematics

1 Introduction

The 2-2 wire-driven parallel crane has two coilable wires connected at two different points B_1, B_2 on the platform (Fig. 1). The wires can be coiled by winches fixed to the ground, whose output points are A_1, A_2 . Hence, provided that gravity is included, it is a 2 d.o.f. robot that allows to control the planar motion of the platform center of mass G, that lies in the vertical plane that includes A_1, A_2 (and also includes B_1, B_2). The lengths of the wires will be denoted by ρ_1, ρ_2. We will assume that the distance between B_1, B_2 is smaller than the distance between A_1, A_2 so that both wires cannot be parallel. To the best of the author's knowledge the kinematics of such a robot has been addressed only in [3, 5].

We introduce a reference frame $\mathcal{R} = (A_1, \mathbf{x}, \mathbf{y})$ where $\mathbf{y}$ is the vertical direction pointing upward. In this frame the coordinates of A_2 are $(x_{a_2} > 0, y_{a_2})$ and the coordinates of G are (x_g, y_g). We also define a mobile frame $\mathcal{R}_m = (G, \mathbf{x_m}, \mathbf{y_m})$ and in this frame the coordinates of the B_i are (x_{b_i}, y_{b_i}).

A rotation matrix $\mathbf{R}$ of angle θ is used to get the components in $\mathcal{R}$ of a vector whose components is known in $\mathcal{R}_m$. Especially the coordinates (x_i, y_i) of B_i in $\mathcal{R}$ are obtained as:

J.-P. Merlet (✉)
INRIA,Valbonne, France
e-mail: Jean-Pierre.Merlet@inria.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_2,

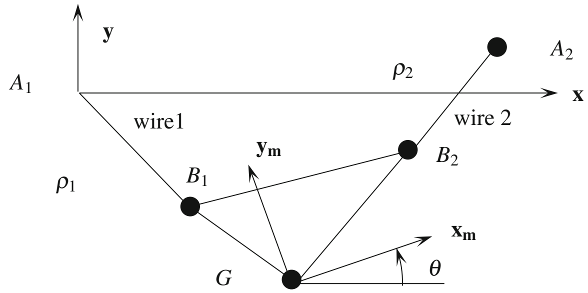


Fig. 1 The 2-2 robot

$$x_i = x_g + \cos\theta x_{b_i} - \sin\theta y_{b_i} \quad y_i = y_g + \sin\theta x_{b_i} + \cos\theta y_{b_i} \tag{1}$$

The length ρ_i of wire i is obtained as

$$\rho_i^2 = (x_i - xa_i)^2 + (y_i - y_i)^2 \tag{2}$$

Let $\mathcal{F} = (0, -mg, 0, 0, 0, 0)^T$ be the force and torque applied on the platform, where m is its mass and let $\tau = (\tau_1, \tau_2)^T$ be vector of tensions in the wires. Static equilibrium is obtained when

$$\mathcal{F} = \mathbf{J^{-T}}\tau \tag{3}$$

where $\mathbf{J^{-T}}$ is a 6×2 matrix whose ith column $\mathbf{J^{-T}}_i$ is

$$\mathbf{J^{-T}}_i = \left(\frac{\mathbf{A_iB_i}}{\rho_i} \quad \frac{\mathbf{GB_i} \times \mathbf{A_iB_i}}{\rho_i}\right)^T \tag{4}$$

This column is the Plücker vector of the line going through A_i, B_i while $\mathcal{F}$ is the vertical line going through G. Eq. (3) indicates that at mechanical equilibrium the lines A_1B_1, A_2B_2 and the vertical line going through G span a linear complex i.e. meet at the same point. Note that the coordinate of $\mathbf{A_iB_i}$ along the z axis of $\mathcal{R}$ is 0 and consequently (3) admits a reduced form: if $\mathcal{F}_r = (0, -mg, 0)^T$ and $\mathbf{J^{-T}}_r$ is a matrix whose ith column is $(\frac{\mathbf{A_iB_i}_x}{\rho_i}, \frac{\mathbf{A_iB_i}_y}{\rho_i}, \frac{\mathbf{GB_i}_x\mathbf{A_iB_i}_y - \mathbf{GB_i}_y\mathbf{A_iB_i}_x}{\rho_i})$, then we have

$$\mathcal{F}_r = \mathbf{J^{-T}}_r\tau \tag{5}$$

Let us define the 3×3 matrix $\mathbf{M}$ whose first and third columns are the first and second columns of $\mathbf{J^{-T}}_r$, while its second column is $\mathcal{F}_r$. If we define $\alpha = (\tau_1, -1, \tau_2)$, then Eq. (5) may be written as $\mathbf{M}\alpha = \mathbf{0}$. As α is not equal to 0 the mechanical equilibrium condition may also be written as:

$$|\mathbf{M}| = 0 \tag{6}$$

Note that Eqs. (5, 6) are only necessary conditions for mechanical equilibrium as we have also to ensure that the tensions in the wires are all positive.

2 Trajectory, Equilibrium Condition and Inverse Kinematics

Let (X_i, Y_i) be the components of the vector $\mathbf{GB_i}$ in $\mathcal{R}$. The equilibrium condition (6) may be written as

$$\begin{aligned}(Y_2 - Y_1)x_g^2 + (X_1 - X_2)x_g y_g + (x_{a_2}(Y1 - Y2) + y_{a_2}(X_2 + Y_2 X_1 - Y_1 X_2)x_g \\ -x_{a_2} X_1 y_g + X_1(y_{a_2} X_2 - x_{a_2} Y_2) = 0 \qquad (7)\end{aligned}$$

Hence if θ is fixed (and consequently so are the X_i, Y_i), then G moves along an hyperbola whose principal axes makes an angle ϕ with the $\mathbf{x}$ axis (such that $\tan(2\phi) = (X_1 - X_2)/(Y_2 - Y_1)$) and whose center admits $x_{a_2} X_1/(X1 - X_2)$ as x coordinate. An analysis of this hyperbola allows one to determine if a given orientation θ is reachable either over the full workspace or at least on part of it. For example the coefficient of y_g in (7) cancels for $x_g = x_g^s = x_{a_2} X_1/(X_1 - X_2)$; consequently if $x_g^s \in [0, x_{a_2}]$, then the workspace is separated into two components and it is not possible to maintain the given orientation over the workspace of the crane. Note that the equilibrium condition (7) is a function of x_g, y_g, θ but we will see in Sect. 3 that it may also be expressed as a function of x_g, y_g, ρ_1, ρ_2.

While the 2-2 is a 2 d.o.f. robot, the platform has still 3 d.o.f. Hence for solving the inverse kinematics it is necessary to specify 2 of this 3 d.o.f while the value of the remaining variable X will be determined by solving the Eq. (7). We examine now the possible different cases:

- x_g, θ are fixed: Eq. (7) is linear in y_g, there is a single possible value for y_g
- y_g, θ are fixed: Eq. (7) is a second order polynomial in x_g, there is up to two possible values for x_g
- x_g, y_g are fixed: using the Weierstrass substitution on (7) leads to a 4th order polynomial in $T = \tan(\theta/2)$ that may have four real roots that furthermore leads to positive tensions in the wires. For example for $x_{a_2} = 20$, $y_{a_2} = 10$, $x_{b_1} = -20$, $x_{b_2} = 20$, $y + b_1 = y_{b_2} = 1$, $x_g = 10$, $y_g = -20$ we get indeed four possible values for θ that all leads to positive wire tensions.

3 Direct Kinematics

The direct kinematics (DK) of the 2-2 robot has been presented in [5]. The solutions can be obtained by solving a 12th order univariate polynomial and [5] provides a geometrical explanation of order of the polynomial that is related to the sextic nature of the coupler curve of the 2-2. But, as mentioned by Carricato [3] if the platform may perform out of the plane motion, then there may be up to 24 solutions: indeed we have to consider possible reflection of the problem or equivalently we have to consider the solutions that are are obtained when reversing the role of A_1, A_2.

We first simply propose here another approach to derive the 12th order polynomial. The static equilibrium condition (7) is a function of $\sin\theta, \cos\theta$, while Eq. (2) are linear functions of these quantities. Solving these two equations in these unknowns and reporting the result in (7) leads to an equilibrium condition $\mathcal{E}_1$ that is function of x_g, y_g, ρ_1, ρ_2 of degree 7 in x_g and 6 in y_g. The constraint $\sin^2\theta + \cos^2\theta - 1 = 0$ leads to another equation $\mathcal{E}_2$. The resultant of $\mathcal{E}_1, \mathcal{E}_2$ in y_g is a 24th order polynomial in x_g that factors out in a 12th order polynomial and a third order polynomial that is raised at the power of 4. This second polynomial provides only solutions with $y_g > 0$ while the first polynomial provides the valid solutions. Stability analysis of these solutions have been addressed in [1, 2].

We have then to consider that obtaining all the solutions from the solving of the univariate polynomial may not be the best method: indeed numerical round-off errors will affect the calculation of the polynomial coefficients while the solving of a 12th order polynomial may be numerically unstable. Instead of using the polynomial, it is possible to transform the problem into an eigenvalue problem, which is numerically more stable, but still the results cannot be guaranteed. Hence we propose to use interval analysis (IA) on a set of DK equations, as this method allows to provide **all** solutions **exactly** (i.e. with an arbitrary accuracy). The efficiency of this method is however dependent upon the set of equations that has to be solved. Hence we may consider several forms of the problem:

1. with 8 equations and 8 unknowns: these unknowns are $x_g, y_g, x_1, y_1, x_2, y_2, \tau_1, \tau_2$. The equations are the five Eqs. (2, 3) and the three geometrical constraints $\mathcal{G}$ defined as $||\mathbf{B_1B_2}||^2 = d_{12}^2$, $||\mathbf{B_1G}||^2 = d_{1G}$, $||\mathbf{B_2G}||^2 = d_{2G}$, where d_{12}, d_{1G}, d_{2G} are the known distances between $(B_1, B_2), (B_1, G), (B_2, G)$
2. with 6 equations and the 6 unknowns $x_g, y_g, x_1, y_1, x_2, y_2$. The constraints are the three equations (2, 6) and the three geometrical constraints $\mathcal{G}$
3. with 5 equations and the 5 unknowns $x_g, y_g, \theta, \tau_1, \tau_2$. The constraints are the 5 Eqs. (2, 3)
4. with 4 equations and the 4 unknowns $x_g, y_g, \sin\theta, \cos\theta$. The constraints are the 3 Eqs. (2, 6) and the constraint $\sin^2\theta + \cos^2\theta = 1$.
5. with 3 equations and the 3 unknowns x_g, y_g, θ. The constraints are the 3 Eqs. (2, 5).
6. with 2 equations and 2 unknowns: these unknowns are either x_g or y_g and θ. The difference between the two Eqs. (2) is linear in x_g, y_g and is used to obtain one of these variables. The constraints are one of (2) and (5)

Note that methods 1 and 2 take reflection into account and hence provide all solutions in a single pass, while the other methods required to be applied twice.

To compare the efficiency of the solving in the various cases we consider a specific robot, called *test robot*, that will be used all over this paper, such that: $A_1 = (0, 0)$ $A_2 = (100, 10)$ $B_1 = (-10, 1)$ $B_2 = (10, 2)$ and we solve the DK for $\rho_1 = 110, \rho_2 = 100$. The solving times in seconds for the above methods are

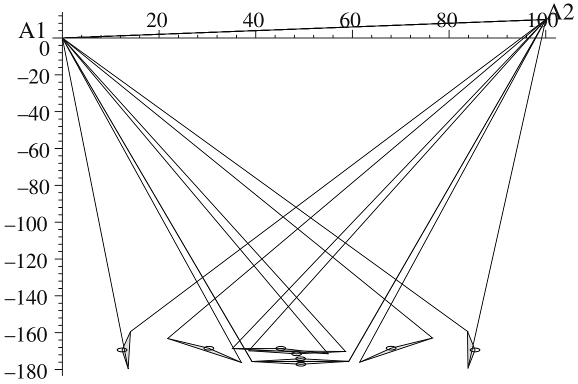


Fig. 2 The eight solutions of the DK obtained for the test robot with $\rho_1 = 180$, $\rho_2 = 190$

0.95 (4), 0.49 (4), 0.8 (2), 0.04 (2), 0.11 (2), 3.66 (6), where the number in parenthesis is the number of solutions. The total number of solutions with positive tensions is 4 as indicated by methods 1 and 2. Methods 3, 4, 5 provide only half of the solutions because reflection is not taken into account and method 6 provides 6 solutions but only 2 with positive tensions. Hence the most efficient methods are 2 and 4 (which requires 2 passes to get all solutions).

We have conducted a study of the number of solutions with positive tensions by selecting randomly 400 wire lengths and solving the DK in each case. We have found that the DK has 2 solutions in 34 % of the cases, 3 in 8.75 %, 4 in 45.5 % 5 in 0.75 %, 6 in 8.5 %, 7 in 0.5 % and 8 in 2 %. Evidently we cannot claim that even with this relatively high number of trials these numbers will always be relevant. Figure 2 shows an example with 8 solutions obtained for $\rho_1 = 180$, $\rho_2 = 190$. The solution (x_g, y_g, θ) obtained for the rotation matrix are (45.19, −168.50, 174.79), (30.31, −168.47, 142.38), (49.34, −177.12, 3.10), (68.01, −168.40, 223.97) with θ in degree. For the reflection we get (48.46, −171.49, 175.96), (12.35, −169.36, 85.78), (49.29, −174.13, 357.43), (85.36, −169.37, 267.40). Note that if out of the plane motion is possible, then we have only 2 stable solutions.

4 Statics in the Joint Space

If the pose parameters are known, then Eq. (3) allows to determine the tensions in the wires. It may however be of interest to study kinematics in the joint space ρ_1, ρ_2. In this section we will address two problems:

1. determining the wire tensions as functions of ρ_1, ρ_2 only
2. determining the region $\mathcal{W}$ in the joint space such that $|\tau_1|, |\tau_2| \leq \tau_{max}$, where τ_{max} is a fixed threshold

For point 1 we note that the two first equations of (5) are linear in x_g, y_g. Solving this linear system and reporting the result in the last equation of (5) and in the 2 Eqs. (2) leads to a system of 3 equations in θ, τ_1, τ_2. This system may be converted into an algebraic system by using the Weierstrass substitution $T = \tan(\theta/2)$. Taking the resultant with respect to T of each pair of equations leads to two polynomials P_1, P_2 in τ_1, τ_2, which have degree (6,6). For a given robot geometry the resultant of P_1, P_2 in τ_2 factor out in two polynomials of degree 12 and 20 in τ_1, only the former one leading to valid values for τ_1. Solving this polynomial and back substituting its roots in P_1, P_2 allow to calculate τ_2, which complete the static analysis in the joint space.

For point 2 checking if, for a given mass of the platform, the value of the wire tensions are lower than the breaking point is clearly of interest. If only joint control is use we are interested in determining the location of the points such that one of the wire tension is equal to its allowed maximum τ_{max} in the joint space ρ_1, ρ_2.

Assume that we set $\tau_1 = \tau_{max}$; Eqs. (2, 3) become a system of 5 equations in the unknowns x_g, y_g, θ, τ_2. The first equation of (3) is used to determine τ_2, while the difference between the 2 Eq. (2) is linear in x_g and is used to determine x_g. The second equation of (3) is linear in y_g: after solving it remains 2 equations (the third equation of (3) and one of the Eq. (2), that are only function of θ. If we define $T = \tan(\theta/2)$ the first equation may be written as the product of 3 polynomials $A \times B(T, \rho_1, \rho_2) \times C(T, \rho_1, \rho_2)$ of degree 2, 2, 4 in T, the polynomial B having only terms in ρ_1^2 or ρ_2^2, while C is a 3rd order polynomial in ρ_1 and includes only terms with ρ_2^2. The second equation may be written as $A \times D(T, \rho_1, \rho_2)$ where D is a sixth order polynomial in T and in ρ_1, while it includes only ρ_2^4, ρ_2^2 terms. The common term A cancels when both wires are parallel, a case we have excluded. and consequently both equations will cancel either when the resultant of B, D or the resultant of C, D is equal to 0. The resultant of B, D factors out in a polynomial of degree 6 in ρ_1 and includes only ρ_2^4, ρ_2^2 terms. The resultant of C, D is too large to be obtained when the geometrical parameters of the robot are kept symbolic but it may be easily obtained for a given geometry and leads to a polynomial of degree 16 in ρ_1 and 12 in ρ_2 with a total degree 16.

To determine the region $\mathcal{W}$ of the $\rho_1 - \rho_2$ joint space where we have $\tau_1 \leq \tau_{max}$ and $\tau_2 \leq \tau_{max}$ we have to plot the above curves for the two cases $\tau_1 = \tau_{max}$ and $\tau_2 = \tau_{max}$. These curves will be splitted in arcs whose start and end extremities are points such that either $\tau_1 = \tau_2 = \tau_{max}$ or $\tau_1 \leq 0$ or $\tau_2 \leq 0$ (the calculation in the joint space in the two later case being presented in Sect. 5). Classicaly the border of $\mathcal{W}$ will be obtained as a set of such arcs, which are determined by checking the constraints for the mid-point of the arcs, using a method that is similar to the one used for determining the workspace of a parallel robot [4].

We consider the test robot and we choose $\tau_{max} = 2\mathcal{F}$. Figure 3 shows the curves with the following notation: a curve denoted `tij` is such that $\tau_i = \tau_{max}$ and a curve denoted `vi` corresponds to the case where wire i supports all the load.

Consider the case where we have $\rho_1 = 40$. Setting $\tau_1 = \tau_{max}$ leads to 4 solutions $U_j = (\rho_2, \tau_2, x_g, y_g)$ with $U_1 = (42.96, 2.095, 49.7146 - 6.66)$, $U_2 = (83.87, 1.989, 28.548, -8.8)$, $U_3 = (115.52, -1.517, -20.134, -23.017)$, $U_4 = (151.688, -1.4219, -35.554, -34.846)$. These points correspond to the

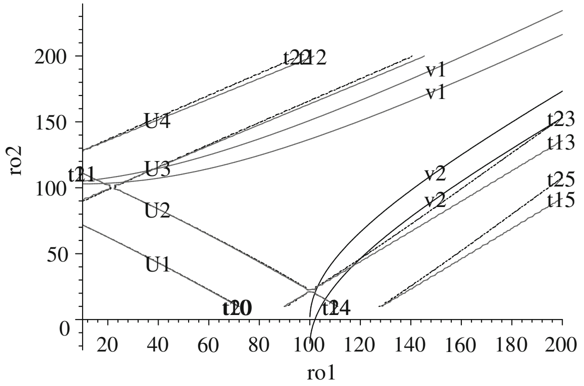


Fig. 3 The curves that appear in the calculation of the region in the $\rho_1 - \rho_2$ space for which $|\tau_1|, |\tau_2| \leq \tau_{max}$

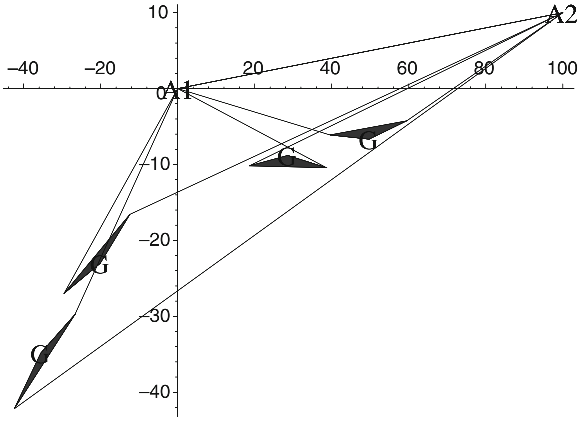


Fig. 4 The poses of the solutions obtained for $\rho_1 = 40$

intersection of the line $\rho_1 = 40$ with the arcs of curve denoted `t10, t13, t13, t12`. It may be noted that for points U_3, U_4 we have $\tau_2 < 0$: this is quite normal as these points are over the line `v1`. We notice also that for point U_1 we have $\tau_2 > 2$. If we now plot the 4 solutions (Fig. 4) we note that the solution U_2 corresponds to the unstable case where G lies over the point B_1, B_2 and, as expected, solutions U_3, U_4 lies on the left side of the y axis which is the limit in which wire 1 supports the whole load while wire 2 is slack. Hence none of these point may belong to the border of $\mathcal{W}$. If we consider the case $\tau_2 = \tau_{max}$ we get also 4 solutions (not represented in the figure) $V_j = (\rho_2, \tau_1, x_g, y_g)$ with $V_1 = (43.22, 1.905, 49.629, -7.192)$, $V_2 = (83.834, 2.01, 28.568, -8.733)$, $V_3 = (116.067, -2.262, -25.997, 14.647)$, $V_4 = (153.893365, -2.337226, -42.997, 25.872)$. For solutions V_3, V_4 we have $\tau_1 < 0$, while for V_2 we have $\tau_1 > \tau_{max}$ and a further analysis shows that this solution is also unstable. Hence the only valid solution is U_1 which is a point of the border of $\mathcal{W}$. A more complete analysis, that cannot be presented here for lack of space, shows that the region $\mathcal{W}$ has as border the curve `v1, v2`, one arc of the curve `t10` and one arc of the curve `t20`.

5 Singularities

Wire-driven parallel robot have the same singularity than parallel robots with rigid legs (provided that the wire tensions are positive). In the 2-2 case this singularity will be obtained if the line A_1B_1, A_2B_2 are colinear. But this situation cannot occur for a crane as the mechanical equilibrium cannot be satisfied (the equilibrium along the $\mathbf{x}$ axis imposes $\tau_1 = -\tau_2$ and therefore the vertical force resulting from the wire tension is 0 and therefore cannot balance the weight of the platform).

But another type of singularity has to be considered namely when a wire tension cancels, as in that case we loose control of one d.o.f. For the 2-2 this happens when G, A_1, B_1 (or G, A_2, B_2) lies on the same vertical line. If we assume that only wire 1 is under tension we get $x_g = 0$, $x_1 = 0$ from which we get $\tan\theta = x_{b_i}/y_{b_i}$ while y_g has an arbitrary negative value. The minimal length of wire 2 may then be computed for a given y_g with equation (2). However it may be of interest to get the singularity condition in the joint space ρ_1, ρ_2. If d_1 is the distance between G and B_1 we have $y_g = -\rho_1 \pm d_1$ Substituting this value in (2) leads to

$$\rho_2^2 \geq \left(\frac{y_{b_1}\, x_{b_2} - x_{b_1}\, y_{b_2}}{d_1} - x_{a_2}\right)^2 + \left(-\rho_1 \pm d_1 + \frac{x_{b_1}\, x_{b_2} + y_{b_1}\, y_{b_2}}{d_1} - y_{a_2}\right)^2 \quad (8)$$

which are the singularity conditions in the joint space. Another singularity condition can be obtained in the same way if wire 2 only is under tension.

6 Conclusion

In spite of its apparent simplicity the kinematics of the 2-2 robot is quite complex. This paper has addressed not so well known kinematics issues that may be of interest for the robot control either in the operational or in the joint space. The next step will then to extend the concepts proposed in this paper to spatial wire robots with 3 to 6 wires.

Acknowledgments This research has received funding from the European Community's Seventh Framework Program under grant agreement NMP2-SL-2011-285404 (CABLEBOT).

References

1. Bosscher, P., Ebert-Uphoff, I.: Wrench-based analysis of cable-driven robots. In: IEEE International Conference on Robotics and Automation, pp. 4950–4955. New Orleans, 28–30 April 2004
2. Carricato, M., Merlet, J.P.: Stability analysis of underconstrained cable-driven parallel robots. IEEE Trans. Robotics **29**(1), 288–296 (2013)
3. Carricato, M., Merlet, J.P.: Geometrico-static analysis of under-constrained cable-driven parallel robot. In: ARK, pp. 309–320. Piran 28 June–1 July 2010
4. Gosselin, C.: Determination of the workspace of 6-dof parallel manipulators. ASME J. Mechan. Des. **112**(3), 331–336 (1990)
5. Michael, N., Fink, J., Kumar, V.: Cooperative manipulation and transportation with aerial robots. In: Robotics: Science and Systems. Seattle, June 2009

Some Rational Vehicle Motions

J. M. Selig

Abstract It is observed that the kinematic equations of many vehicles take the same form. This form is that the body-fixed velocity twist of the vehicle lies in a fixed screw-system of a particular type. The Cayley map can be used to pull-back these equations to the Lie algebra of the group of rigid-body motions. Rational solutions to the equations can be found by the method of undetermined coefficients. Since the Cayley map is a rational map, mapping these rational solutions back to the group gives rational rigid-body motions. A 3-parameter family of rational Frenet-Serret motions is found in this way. Multiplying these motions by a rational roll-motion gives a 4-parameter family of aeroplane motions.

Keywords Rational rigid-body motions · Cayley map · Cars · Aeroplanes.

1 Introduction

Many vehicles are modelled as non-holonomic systems. Usually the kinematic equations for a vehicle's motion have the form

$$G(t)^{-1}\frac{dG(t)}{dt} = S_B, \tag{1}$$

here $G(t)$ lies in the group of rigid-body motions and represents the motion of the vehicle. The body fixed velocity twist S_B is constrained to lie in some fixed screw system. The ubiquity of this type of model can be explained by the fact that for most vehicles the actuators are fixed with respect to the body and provide forces or torques along screws fixed in the body. Many examples exist, a few are listed below.

Bicycles and cars. Perhaps the motivating example and the most intensely studied one. It is of course, a planar problem so the analysis can be done in the group of planar

J. M. Selig (✉)
Faculty of Business, London South Bank University, London SE1 0AA, UK
e-mail: seligjm@lsbu.ac.uk

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_3,

rigid displacements, $SE(2)$. The body-fixed velocity of such a vehicle is restricted to a forward velocity in only one direction and a rotational velocity about a line perpendicular to the plane.

Aeroplanes, roller-coasters, autonomous underwater vehicles. These vehicles can translate forward, in a fixed direction relative to the vehicle, but can also roll, pitch and yaw. Hence the body-fixed velocity twist S_B, of such a vehicle lies in a particular 4-system of screws.

Needle steering. The problem of steering a long flexible needle or cannula with a bevelled tip is now well known in robotics. The motion of the needle tip is a Frenet-Serret motion (see below) but based on a curve with constant curvature. This means that S_B for this system lies in a 2-system of screws. The kinematic equations for this problem can be found in [5], see also [3].

Frenet-Serret motions. Not the motion of a vehicle but it does have that same kinematic equation as given in (1). Here the body-fixed velocity twist S_B, lies in a particular 3-system of screws. This example is often studied in computer aided design but these motions have also been proposed as suitable motions for the end-effector of a manipulator arm. See also Bishop motions [3].

Many workers have studied the path-planning problem for such vehicles. Here a very simple view is taken. The idea is to plan rigid-body motions using simple interpolation schemes. It is well known that polynomial interpolation on the group of rigid motions is far from straightforward due to the non-linear nature of the group. Moreover, another problem arises, not all motions in the group satisfy the kinematic equation (1), for the vehicle. This work addresses the problem of finding simple families of motions satisfying the vehicle's kinematic equations. Such families need to be large enough to be able to interpolate a wide range of possible motions yet simple enough to be easy to work with especially for computers. For these reasons rational motions are preferred.

The following strategy is used to find such families of motions. First the Cayley map is used to pull-back the kinematic equation to $\mathbb{R}^6$. Since the Cayley map is a rational map, rational curves in $\mathbb{R}^6$ will be mapped to rational curves in the group. Then a rational anzatz is substituted into the equation to turn the differential equation into algebraic equations. Solutions of the resulting algebraic equations will produce motions with the desired properties.

2 The 4 × 4 Cayley Map

In classical differential geometry a standard problem is to reconstruct a space-curve from its curvature and torsion functions. The differential equations to be solved for this problem have exactly the same for as given in (1). However, G lies in $SO(3)$, the group of rotations about a point. The classical approach to the problem is to make a substitution which turns the equation into a Riccati equation, see [1] for example.

This approach generalises to other matrix groups. The substitution required is the standard Cayley map. Suppose $G \in \mathcal{G}$, the matrix Lie group. Let S be a square matrix representing an element of the Lie algebra to $\mathcal{G}$. The Cayley map is given by,

$$G = (I + S)(I - S)^{-1}, \tag{2}$$

see [2] for more details. Substituting this into the kinematic equation (1) gives,

$$\frac{dS}{dt} = \frac{1}{2}(I + S)S_B(I - S) = \frac{1}{2}(I + SS_B - S_BS - SS_BS). \tag{3}$$

This is an example of a matrix Riccati equation. If S_B is a known function of time then this equation can be solved numerically to simulate the motion of the vehicle.

Here $G \in SE(3)$,the group of proper rigid-body displacements in 3-dimensions. It is assumed that all that is known is that S_B lies in a known screw-system; a linear subspace of the Lie algebra to $SE(3)$. In the 4×4 representation of the Lie algebra to $SE(3)$ a general twist can be written as,

$$S = \begin{pmatrix} \Omega & \mathbf{v} \\ 0 & 0 \end{pmatrix}, \tag{4}$$

where Ω is the anti-symmetric 3×3 matrix corresponding to the angular velocity vector $\boldsymbol{\omega}$. That is, $\Omega\mathbf{p} = \boldsymbol{\omega} \times \mathbf{p}$ for any vector $\mathbf{p}$. The vector $\mathbf{v}$ is the linear velocity of the origin.

The motion within this system is usually specified by a number of control functions related to the physical actuators. To project out these control functions Eq. (3) can be rearranged to give,

$$2(I + S)^{-1}\frac{dS}{dt}(I - S)^{-1} = S_B. \tag{5}$$

Next we write the equation using the adjoint representation of the group. In this representation twists are represented as 6-dimensional column vectors. These vector are usually written in a partitioned form with the first three components of $\mathbf{s}$ given by $\boldsymbol{\omega}$ and the second three by $\mathbf{v}$. In this representation Eq. (5) becomes,

$$F(\mathbf{s})\frac{d\mathbf{s}}{dt} = \mathbf{s}_B, \tag{6}$$

where $F(\mathbf{s})$ is the matrix.

$$F(\mathbf{s}) = \frac{2}{1 + |\boldsymbol{\omega}|^2}\begin{pmatrix} I - \Omega & 0 \\ V\Omega - V & (1 + |\boldsymbol{\omega}|^2)I - \Omega + \Omega^2 \end{pmatrix}. \tag{7}$$

Here V is the 3×3 anti-symmetric matrix corresponding to $\mathbf{v}$. This result can be found using the methods presented in [2].

Suppose $\mathcal{W}_1, \ldots \mathcal{W}_n$ are a set of linearly independent wrenches forming a basis of the dual space to screw-system that $\mathbf{s}_B$ lies in. That is $\mathcal{W}_i^T\mathbf{s}_B = 0$, $i = 1, \ldots, n$.

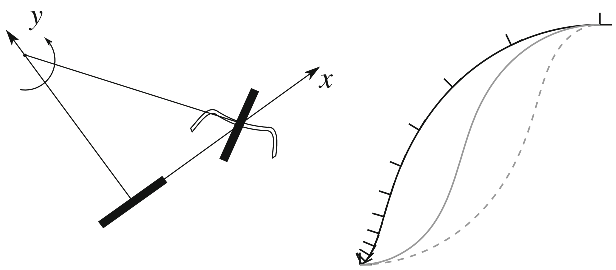


Fig. 1 Bicycle paths. The *left-hand figure* shows the body-fixed frame in the bike. On the *right* three different rational motions between the same positions are illustrated. The motion of the body-fixed frame is shown for the *top path*

Finally the kinematic equations can be written as,

$$\mathcal{W}_i^T F(\mathbf{s})\frac{d\mathbf{s}}{dt} = 0, \qquad i = 1, \ldots, n. \tag{8}$$

To make this clearer a simple example is presented in the next section.

3 Rational Bicycle Motions

This is a planar problem so let,

$$S = \begin{pmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{pmatrix} \qquad \text{or as a column vector} \qquad \mathbf{s} = \begin{pmatrix} \omega \\ v_x \\ v_y \end{pmatrix}.$$

with the coordinates given in Fig. 1 the bicycle can translate in the x-direction or rotate about a vertical line located on the y-axis. In general any combination of these two infinitesimal motion is possible so that,

$$\mathbf{s}_B = \nu \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tag{9}$$

where the arbitrary coefficients ν and μ are related to the forward velocity and turning velocity. The only wrench dual to every possible $\mathbf{s}_B$ is given by $\mathcal{W}^T = (0,\ 0,\ 1)$. This expresses the fact that the bicycle cannot instantaneously translate in the y-direction. In this planar problem the matrix $F(\mathbf{s})$ reduces to,

$$F(\mathbf{s}) = \frac{2}{1+\omega^2}\begin{pmatrix} 1 & 0 & 0 \\ -v_y & 1 & \omega \\ v_x & -\omega & 1 \end{pmatrix}. \tag{10}$$

Hence the kinematic equation for the motion reduces to,

$$v_x\dot{\omega} - \omega\dot{v}_x + \dot{v}_y = 0. \tag{11}$$

We can produce polynomial solutions to this equation by choosing arbitrary polynomials for ω and v_x, substituting these into Eq. (11) and solving for v_y by integration. For example, choosing $\omega = a_1t + a_2t^2$ and $v_x = b_1t + b_2t^2$ gives $v_y = (1/3)(a_1b_2 - a_2b_1)t^3$. The constant terms, including the integration constant, have been chosen as zero, so that the motion begins at the identity when $t = 0$. This gives a 4-parameter family of motions, with a_1, a_2, b_1 and b_2 as the 4-parameters.

The Cayley map can now be used to map this result to the 3×3 representation of the group,

$$G_P(t) = \begin{pmatrix} \frac{(1+a_1t+a_2t^2)(1-a_1t-a_2t^2)}{1+(a_1+a_2t)^2t^2} & \frac{-2(a_1t+a_2t^2)}{1+(a_1+a_2t)^2t^2} & \delta_x \\ \frac{2(a_1t+a_2t^2)}{1+(a_1+a_2t)^2t^2} & \frac{(1+a_1t+a_2t^2)(1-a_1t-a_2t^2)}{1+(a_1+a_2t)^2t^2} & \delta_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{12}$$

where

$$\delta_x = \frac{2\big(3b_1t + 3b_2t^2 - a_1(a_1b_2 - a_2b_1)t^4 - a_2(a_1b_2 - a_2b_1)t^5\big)}{3\big(1 + (a_1 + a_2t)^2t^2\big)}, \tag{13}$$

$$\delta_y = \frac{2\big(3a_1b_1t^2 + (2a_1b_2 + 4a_2b_1)t^3 + 3a_2b_2t^4\big)}{3\big(1 + (a_1 + a_2t)^2t^2\big)}. \tag{14}$$

Examples of this motion are shown in Fig. 1.

4 Rational Frenet-Serret Motions

Frenet-Serret motions are usually defined with respect to a space-curve. Given a regular curve in space its Frenet-Serret motion is the motion of a body rigidly attached to the Frenet frame of the curve. This frame is sometimes known as the normal-tangent-binormal coordinate system. The origin of the coordinate frame fixed in the body to be the point on the curve, the x-axis of the body-fixed frame will be aligned with the tangent to the curve and y-axis will be taken to lie along the curve's normal vector.

The motion of this frame can be written as a curve of group elements $G(t)$ satisfying the famous Frenet-Serret equations, which coincide with Eq. (1). In this case

S_B has the form,

$$S_B = \begin{pmatrix} 0 & -\nu\kappa & 0 & 1 \\ \nu\kappa & 0 & -\nu\tau & 0 \\ 0 & \nu\tau & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{15}$$

where ν, κ and τ are respectively the speed, curvature and torsion of the curve generating the motion. The adjoint representation of this twist is,

$$\mathbf{s}_B = \alpha \begin{pmatrix} \mathbf{k} \\ \mathbf{0} \end{pmatrix} + \beta \begin{pmatrix} \mathbf{i} \\ \mathbf{0} \end{pmatrix} + \gamma \begin{pmatrix} \mathbf{0} \\ \mathbf{i} \end{pmatrix}, \tag{16}$$

where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are the unit vectors in the x, y and z-directions respectively and the coefficients $\alpha = \nu\kappa$, $\beta = \nu\tau$ and $\gamma = \nu$.

The three wrenches dual to any $\mathbf{s}_B$ are clearly given by,

$$\mathcal{W}_1 = \begin{pmatrix} \mathbf{j} \\ \mathbf{0} \end{pmatrix}, \quad \mathcal{W}_2 = \begin{pmatrix} \mathbf{0} \\ \mathbf{j} \end{pmatrix} \quad \text{and} \quad \mathcal{W}_3 = \begin{pmatrix} \mathbf{0} \\ \mathbf{k} \end{pmatrix}.$$

Using the result for $F(\mathbf{s})$ from Eq. (7) above the three equations defining the motion can be given as,

$$0 = \dot{\omega}_y + \omega_x\dot{\omega}_z - \dot{\omega}_x\omega_z, \tag{17}$$

$$\begin{aligned} 0 = {} & (\omega_x\omega_y - \omega_z)\dot{v}_x + (1+\omega_y^2)\dot{v}_y + (\omega_x + \omega_y\omega_z)\dot{v}_z \\ & +(\dot{\omega}_z + \dot{\omega}_x\omega_y - \omega_x\dot{\omega}_y)v_x - (\dot{\omega}_x + \dot{\omega}_y\omega_z - \omega_y\dot{\omega}_z)v_z \end{aligned} \tag{18}$$

$$\begin{aligned} 0 = {} & (\omega_x\omega_z + \omega_y)\dot{v}_x + (\omega_y\omega_z - \omega_x)\dot{v}_y + (1+\omega_z^2)\dot{v}_z \\ & -(\dot{\omega}_y + \omega_x\dot{\omega}_z - \dot{\omega}_x\omega_z)v_x + (\dot{\omega}_x + \dot{\omega}_y\omega_z - \omega_y\dot{\omega}_z)v_y. \end{aligned} \tag{19}$$

Notice that the common denominators in these relations have been cancelled.

Now consider Eq. (17), this has exactly the same form as Eq. (11). Expressions for ω_x and ω_z can be chosen and then the solution for ω_y can be found by integration. Let us set $\omega_x = at$ and $\omega_z = bt$ this gives $\omega_y = 0$. To find possible solutions for v_x, v_y and v_y consider the anzatz,

$$v_x = \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \alpha_4 t^4,\ v_y = \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \beta_4 t^4,\ v_z = \gamma_1 t + \gamma_2 t^2 + \gamma_3 t^3 + \gamma_4 t^4.$$

Substituting the chosen solution for ω_i and the anzatz for the v_j into Eqs. (18) and (19) gives a pair of polynomial equations in t. These equations have degree 4 and 5 in t and the coefficients are linear in the undetermined coefficients α_i, β_j and γ_k. These polynomials must vanish for all values of t hence the coefficients can be set equal to zero. This produces a set of 11 linear equations in the 12 unknowns. The system can be solved easily to give,

$$v_x = \alpha_1 t + \frac{1}{3}\alpha_1(a^2 + 2b^2)t^3, \qquad v_y = \frac{1}{6}\alpha_1(a^2 + b^2)bt^4, \qquad v_z = \frac{1}{3}\alpha_1 abt^3.$$

These results give the entries of S and hence, using the Cayley map, the corresponding 4×4 matrix function of t can be calculated,

$$G_{FS}(t) = \begin{pmatrix} \frac{1+(a^2-b^2)t^2}{1+(a^2+b^2)t^2} & \frac{-2bt}{1+(a^2+b^2)t^2} & \frac{2abt^2}{1+(a^2+b^2)t^2} & \frac{2c}{3}(3t+(a^2-b^2))t^3 \\ \frac{2bt}{1+(a^2+b^2)t^2} & \frac{1-(a^2+b^2)t^2}{1+(a^2+b^2)t^2} & \frac{-2at}{1+(a^2+b^2)t^2} & 2bct^2 \\ \frac{2abt^2}{1+(a^2+b^2)t^2} & \frac{2at}{1+(a^2+b^2)t^2} & \frac{1-(a^2-b^2)t^2}{1+(a^2+b^2)t^2} & \frac{4}{3}abct^3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{20}$$

note that α_1 has been replaced by c in the above for neatness. This is a 3-parameter family of rational Frenet-Serret motions. It includes the rational Frenet-Serret motion found by Wagner and Ravani [4], which is reproduced by setting $a = b = c = 1/\sqrt{2}$ and a change of the coordinate frame.

5 Rational Aeroplane Motions

In [3] it is proved that these general moving frame motions can be factored into the product of a Frenet-Serret motion with a pure rotation about the heading direction, the tangent to the curve. Hence all that is needed to produce a rational family of such motions is to compose a rational rotation with a family of rational Frenet-Serret motions, such as the family derived above.

The required pure rotation is about the x-axis in the body and can be written,

$$G_R(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1-\lambda^2t^2}{1+\lambda^2t^2} & \frac{-2\lambda t}{1+\lambda^2t^2} & 0 \\ 0 & \frac{2\lambda t}{1+\lambda^2t^2} & \frac{1-\lambda^2t^2}{1+\lambda^2t^2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{21}$$

the parameter λ is essentially the rate of rotation. Combining this with the Frenet-Serret motions given in (20) gives a 4-parameter family of rational Aeroplane motions,

$$\begin{aligned} &G_A(t) \\ &= G_{FS}(t)G_R(t) \\ &= \begin{pmatrix} \frac{1+(a^2-b^2)t^2}{1+(a^2+b^2)t^2} & \frac{-2(bt-(2a+\lambda)b\lambda t^3)}{(1+(a^2+b^2)t^2)(1+\lambda^2t^2)} & \frac{2(2+a)bt^2-ab\lambda^2t^4}{(1+(a^2+b^2)t^2)(1+\lambda^2t^2)} & a_x \\ \frac{2bt}{1+(a^2+b^2)t^2} & \frac{1-(a^2+b^2+4a\lambda+\lambda^2)t^2+(a^2+b^2)\lambda^2t^4}{(1+(a^2+b^2)t^2)(1+\lambda^2t^2)} & \frac{-2((a+\lambda)t-(a^2+b^2+a\lambda)\lambda t^3)}{1+(a^2+b^2)t^2} & a_y \\ \frac{2abt^2}{1+(a^2+b^2)t^2} & \frac{2((a+\lambda)t-(a^2-b^2+a\lambda)\lambda t^3)}{(1+(a^2+b^2)t^2)(1+\lambda^2t^2)} & \frac{1-(a^2-b^2+4a\lambda+\lambda^2)t^2+(a^2-b^2)\lambda^2t^4}{(1+(a^2+b^2)t^2)(1+\lambda^2t^2)} & a_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \end{aligned}$$

where,

$$a_x = \frac{2c}{3}(3t + (a^2 - b^2))t^3, \qquad a_y = 2bct^2, \qquad a_z = \frac{4}{3}abct^3.$$

6 Conclusion

In this work a new class of rigid-body motion has been introduced. These vehicle motions include many well known types associated to various types of vehicle as well as some other motions of interest in Computer Aided Design. By unifying these different types of motion it is possible to find methods which can be applied to any of them. For example the Cayley substitution given in Sect. 2, will transform the kinematic equations for any vehicle motion into a Riccarti equation. The rest of the paper gives another method that can be used for any of these vehicle motions. The problem addressed is to find rational motions satisfying the kinematic equations for the vehicle. In Robotics this has applications in path planning, in Computer Aided Design is has applications to motion interpolations for example.

The results given here are only intended to be indicative, the main purpose is to show that very similar methods can be used for rather different vehicles. However, it is not too difficult to see how the solutions presented could be extended to more practical techniques.

Acknowledgments This work was has been much improved following the comment of the anonymous reviewers.

References

1. Eisenhart, L.P.: A Treatise on the Differential Geometry of Curves and Surfaces. Ginn and Co, Boston (1909)
2. Selig, J.M.: Cayley Maps for $SE(3)$. In: The International Federation of Theory of Machines and Mechanisms 12th World Congress, Besançon (2007)
3. Selig, J.M.: Characterisation of Frenet-Serret and Bishop Motions with applications to Needle Steering, Robotica (to appear) **31**(6), 981–992 (2013)

4. Wagner, M., Ravani, B.: Curves with rational Frenet-Serret motion. Comput. Aided Geom. Des. **15**, 79–101 (1997)
5. Webster III, R.J., Kim, J.S., Cowan, N.J., Chirikjian, G.S., Okamura, A.M.: Nonholonomic modeling of needle steering. Int. J. Robot. Res. **25**(5–6), 509–525 (2006)

Topological Representation and Operation of Motion Space Exchange Reconfiguration of Metamorphic Mechanisms

Shujun Li, Hongguang Wang, Jiansheng Dai, Xiaopeng Li, Zhaohui Ren and Shichao Xiu

Abstract The paper presents a topological representation of mechanical chains based on proposed joint-axis matrix based on our previous study, and motion space exchange reconfiguration metamorphic processes of multi-loop mechanisms are operated by the matrix. The matrix operations of metamorphic process are performed by replacing the joint of planar mechanisms with passive mobility joint to form the passive mobility mechanisms firstly, and then changing the orientation of metamorphic joint of the passive mobility planar mechanisms to transform the configuration of the mechanisms from planar to spatial one. The matrix operations of 8-link multi-loop mechanism are illustrated to show the motion space exchange reconfiguration processes.

Keywords Motion space reconfiguration · Matrix metamorphic operation · Augmented adjacency matrix

1 Introduction

Since the metamorphic mechanism was proposed a decade ago [1], interests and attentions are aroused in the field of mechanisms study. Parise et al. [2] proposed a new class of mechanisms called ortho-planar mechanisms. Li et al. [3] presented a method of structural synthesis of metamorphic mechanisms based on the configuration

S. Li (✉) · X. Li · Z. Ren · S. Xiu
College of Mechanical Engineering and Automation, Northeastern University,
Laboratory of Robotics of China110004 Shenyang, China
e-mail: sjli_cn@yahoo.com.cn

H. Wang · S. Li
Key Laboratory of Robotics of China, 110016 Shenyang, China
e-mail: hgwang@sia.cn

J. Dai
King's College London,University of London, London WC2R 2LS, UK
e-mail: jian.dai@kcl.ac.uk

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_4,

transformations. Liu and Yang [4] studied the metamorphic ways, new concept of metamorphic kinematic pair was introduced and two new basic metamorphic ways are presented. Generally, the adjacency matrix was used to study the variable topologies of metamorphic mechanisms via matrix operation to show the configuration changes of the mechanisms [5]. The symbols of joints are added in adjacency matrix to show the changes of type and/or number of joints in the mechanisms [6, 7]. Li et al proposed a topological characteristic matrix of kinematic chains with loop relations, types of joints and orientation of joints [8], and introduced a matrix representation and of motion space exchange metamorphic operations of single-loop kinematic chains [9]. In the paper, we study the motion space exchange reconfiguration operation of multi-loop mechanisms.

2 The Topological Representation of Metamorphic Chains

2.1 Joint–Axis Matrix

The topological information of a joint J_{ij} which connects the ith and jth links can be presented by following four elements **joint-axis matrix** $\mathbf{J}_{ij}$

$$\mathbf{J}_{ij} = \begin{bmatrix} 1 & J_{ij} \\ a_{ij} & 1 \end{bmatrix}_{2X2} \tag{1}$$

where

$$J_{ij} = J_{ji} = \begin{cases} R,\ P,\ \ldots, & or\ by\ corresponding\ number\ assumed \\ 0 & when\ link\ i\ and\ j\ are\ not\ connected \end{cases}$$

The element a_{ij} presents the orientation of axis of turning joint and/or normal of translating joint of J_{ij}

$$a_{ij} = a_{ji} = \begin{cases} 1,\ 2,\ \ldots,\ 8 & \\ 0 & when\ link\ i\ and\ j\ are\ not\ connected \end{cases}$$

The values a_{ij} are assumed as in Table. 1.

Table 1 Proposed axis–orientation–relation and the value

a_{ij}	1	2	3	4	5	6	7	8
Orientation	Parallel z	Parallel x	Parallel y	Intersected on one point	Arbitrary	Parallel x–y plane	Parallel x–z plane	Parallel y–z plane

2.2 Augmented Adjacent Matrix with Axis-Orientation-Relationship of Kinematic Chains

A n–link kinematic chain is assembled by the putting the corresponding elements of Eq. (1) into the general adjacent matrix of n link kinematic chain with types of joints, the general topological form of a n link metamorphic chain with types and orientations of joints can be proposed in the following form.

$$\mathbf{A} = \begin{bmatrix} 1 & J_{12} & \cdots & J_{1i} & \cdots & J_{1j} & \cdots & J_{1,n-1} & J_{1n} \\ a_{12} & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & J_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1i} & \cdots & \cdots & 1 & \cdots & J_{ij} & \cdots & \cdots & J_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{1j} & \cdots & \cdots & a_{ij} & \cdots & 1 & \cdots & \cdots & J_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1,n-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & J_{n-1,n} \\ a_{1n} & a_{2,n} & \cdots & a_{in} & \cdots & a_{jn} & \cdots & a_{n-1,n} & 1 \end{bmatrix} \tag{2}$$

where for $i = 1, 2,..., n, j = 1, 2, ..., n$, the element J_{ij} is the joint of connecting ith and jth links of kinematic chains.

3 Motion Space Change Metamorphic Reconfiguration and Matrix Operation

3.1 Joints Replacing Operation to form the Passive Mobility of the Mechanism

To reconfigure a planar mechanism into a spatial mechanism, it is essential that passive joints have the spatial mobility be added. The operation can be performed as following: first deleting the elements of replaced joint $\mathbf{J}_{ij}$ which is equal to putting the corresponding elements of a negative joint

$$-\mathbf{J}_{ij} = \begin{bmatrix} 1 & -\mathbf{J}_{ij} \\ -\mathbf{a}_{ij} & 1 \end{bmatrix} (for\ i = 1, 2, ...;\ \ j = 1, 2, ...,\ and\ \mathbf{J}_{ij}\ is\ deleted\ joint)$$

into Eq. (2) to delete the replaced joint/joints, and then adding the elements of

$$\mathbf{J}^p_{ij} = \begin{bmatrix} 1 & J^p_{ij} \\ a^p_{ij} & 1 \end{bmatrix} (for\ i = 1, 2, ...;\ \ j = 1, 2, ...,\ and\ \mathbf{J}^p_{ij}\ is\ deleted\ joint)$$

in the corresponding positions of the replaced joint/joints $\mathbf{J}_{ij}$ of the Eq. (4) to form the topological characteristics matrix of kinematic chain with passive joints $\mathbf{A}_p$ as

$$\mathbf{A}_p = \begin{bmatrix} 1 & J_{12} & \cdots & J_{1i} & \cdots & J_{1j} & \cdots & \cdots & J_{1,n-1} & J_{1n} \\ a_{12} & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & J_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1i} & \cdots & \cdots & 1 & \cdots & J_{ij}^{p} & \cdots & \cdots & \cdots & J_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{1j} & \cdots & \cdots & a_{ij}^{p} & \cdots & 1 & \cdots & \cdots & \cdots & J_{jn} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1,n-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & J_{n-1,n} \\ a_{1n} & a_{2,n} & \cdots & a_{in} & \cdots & a_{jn} & \cdots & \cdots & a_{n-1,n} & 1 \end{bmatrix} \quad (3)$$

3.2 Motion Space Change Reconfiguration Operation

The process to change the orientation of a_{ij} of $\mathbf{A}^p$ to form the metamorphic matrix can be operated as deleting the joint elements of corresponding $-a_{ij}$ of orientation changed joint in the $\mathbf{A}^p$ first which is equal to putting the corresponding elements of

$$-\mathbf{J}_{ij}^{o} = \begin{bmatrix} 1 & J_{ij} \\ -a_{ij} & 1 \end{bmatrix} (for\, i = 1, 2, ...;\ j = 1, 2, ..., \ and\, \mathbf{J}_{ij}^{o}\, is\, orientation\, changed\ joint)$$

into Eq. (3) to delete the elements of changed orientations, and then changing the orientation of the metamorphic joints, i.e. adding the elements of

$$\mathbf{J}_{ij}^{c} = \begin{bmatrix} 1 & J_{ij} \\ a_{ij}^{c} & 1 \end{bmatrix} (for\, i = 1, 2, ...;\ j = 1, 2, ..., \ and\, a_{ij}^{c}\, is\, the\, new\, orientation\ of\ the\ joint)$$

in the corresponding positions of the orientation changed joint $\mathbf{J}_{ij}^{o}$ of the Eq. (3) to form the matrix of spatial motion kinematic chain $\mathbf{A}_S$

$$\mathbf{A}_s = \begin{bmatrix} 1 & J_{12} & \cdots & J_{1i} & \cdots & J_{1j} & \cdots & \cdots & J_{1,n-1} & J_{1n} \\ a_{12} & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & J_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1i} & \cdots & \cdots & 1 & \cdots & J_{ij} & J^{p} & \cdots & \cdots & J_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{1j} & \cdots & \cdots & a_{ij}^{c} & \cdots & 1 & \cdots & \cdots & \cdots & J_{jn} \\ \cdots & \cdots & \cdots & a^{p} & \cdots & \cdots & 1 & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1,n-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & J_{n-1,n} \\ a_{1n} & a_{2,n} & \cdots & a_{in} & \cdots & a_{jn} & \cdots & \cdots & a_{n-1,n} & 1 \end{bmatrix} \quad (4)$$

Thus the metamorphic processes for mechanism reconfiguration from planar to motion can be managed to operate only the corresponding metamorphic joint-axis matrix by Eq. (4). It should be noticed that the passive joint $\mathbf{J}_{ij}^{p}$ are not to be chosen as the orientation change joint $\mathbf{J}_{ij}^{c}$ in the metamorphic process generally.

4 Motion Space Change Reconfiguration Approaches

For a multi-loop planar 3-*RRR* parallel mechanism shown in Fig. 1a, the main reconfiguration procedures are as follows.

To form the topological characteristic matrix **A** *of planar 3-**RRR** parallel mechanism*. The topological characteristic matrix **A** of 3-*RRR* planar parallel mechanism can be formed as according to Eq. (3)

$$\mathbf{A} = \begin{bmatrix} 1 & J_{12} & 0 & 0 & 0 & J_{16} & 0 & J_{18} \\ a_{12} & 1 & J_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{23} & 1 & J_{34} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{34} & 1 & J_{45} & 0 & J_{47} & 0 \\ 0 & 0 & 0 & a_{45} & 1 & J_{56} & 0 & 0 \\ a_{16} & 0 & 0 & 0 & a_{56} & 1 & 0 & 0 \\ 0 & 0 & 0 & a_{47} & 0 & 0 & 1 & J_{78} \\ a_{18} & 0 & 0 & 0 & 0 & 0 & a_{78} & 1 \end{bmatrix} = \begin{bmatrix} 1 & R & 0 & 0 & 0 & R & 0 & R \\ 1 & 1 & R & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & R & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & R & 0 & R & 0 \\ 0 & 0 & 0 & 1 & 1 & R & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & R \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

To form the topological characteristic matrix $\mathbf{A}^{P\_3RRS}$ *of planar 3-**RRS** parallel mechanism with passive mobility joints*. If change the three R joints R_{34}, R_{45}, and R_{47} which connected on the moving platform of link 4 into S joints, the mechanism will become the planar 3-*RRS* mechanism with passive mobility joints. According to Eq. (1), the replaced passive joints are

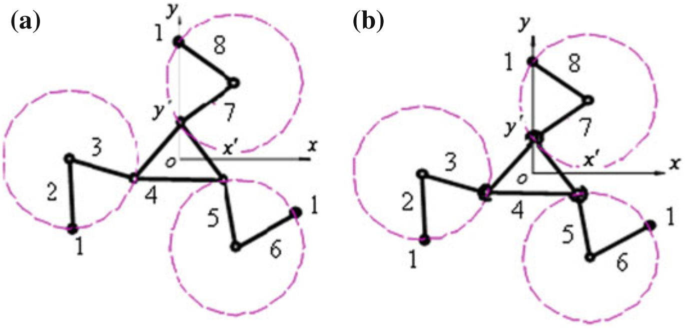


Fig. 1 Planar 3-*RRR* parallel mechanism and 3-*RRS* parallel mechanism

$$\mathbf{J}_{34}^{p}=\begin{bmatrix}1 & J_{34}^{p}\\ a_{34}^{p} & 1\end{bmatrix}=\begin{bmatrix}1 & S\\ 5 & 1\end{bmatrix};\ \mathbf{J}_{45}^{p}=\begin{bmatrix}1 & J_{45}^{p}\\ a_{45}^{p} & 1\end{bmatrix}=\begin{bmatrix}1 & S\\ 5 & 1\end{bmatrix};\ \mathbf{J}_{47}^{p}=\begin{bmatrix}1 & J_{47}^{p}\\ a_{47}^{p} & 1\end{bmatrix}=\begin{bmatrix}1 & S\\ 5 & 1\end{bmatrix}$$

Adding the elements of the replaced passive joints into the corresponding positions of Eq. (3), to form the topological matrix $\mathbf{A}_p$ of this mechanism with passive joints

$$\mathbf{A}_p=\begin{bmatrix}1 & J_{12} & 0 & 0 & 0 & J_{16} & 0 & J_{18}\\ a_{12} & 1 & J_{23} & 0 & 0 & 0 & 0 & 0\\ 0 & a_{23} & 1 & J_{34}^{p} & 0 & 0 & 0 & 0\\ 0 & 0 & a_{34}^{p} & 1 & J_{45}^{p} & 0 & J_{47}^{p} & 0\\ 0 & 0 & 0 & a_{45}^{p} & 1 & J_{56} & 0 & 0\\ a_{16} & 0 & 0 & 0 & a_{56} & 1 & 0 & 0\\ 0 & 0 & 0 & a_{47}^{p} & 0 & 0 & 1 & J_{78}\\ a_{18} & 0 & 0 & 0 & 0 & 0 & a_{78} & 1\end{bmatrix}=\begin{bmatrix}1 & R & 0 & 0 & 0 & R & 0 & R\\ 1 & 1 & R & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 1 & S & 0 & 0 & 0 & 0\\ 0 & 0 & 5 & 1 & S & 0 & S & 0\\ 0 & 0 & 0 & 5 & 1 & R & 0 & 0\\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 0 & 5 & 0 & 0 & 1 & R\\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1\end{bmatrix}$$

The corresponding configuration of $\mathbf{A}_p$ can be formed as shown in Fig. 1b.

To form the orientation change metamorphic matrix $\mathbf{A}_S$ *of metamorphic mechanism based on the* $\mathbf{A}_p$. According to Sect. 3, the orientation changing reconfiguration processes of this mechanism are:

Firstly, turning the link 2 in z with appropriate position, and then changing the orientation of the joint R_{12} 90° to form the orientation change metamorphic matrix $\mathbf{A}_S(R_{12})$. By Eqs. (1) and (4), the orientation changed joint $\mathbf{J}_{12}^{c}$ and the orientation change metamorphic matrix $\mathbf{A}_S(R_{12})$ of the mechanism can be formed as

$$\mathbf{J}_{12}^{c}=\begin{bmatrix}1 & J_{12}\\ a_{12}^{c} & 1\end{bmatrix}=\begin{bmatrix}1 & R\\ 6 & 1\end{bmatrix}$$

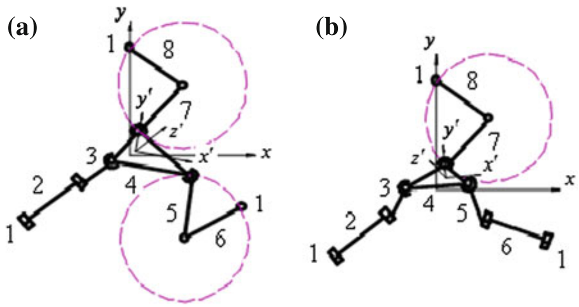
$$\mathbf{A}_S(R_{12})=\begin{bmatrix}1 & J_{12} & 0 & 0 & 0 & J_{16} & 0 & J_{18}\\ a_{12}^{c} & 1 & J_{23} & 0 & 0 & 0 & 0 & 0\\ 0 & a_{23} & 1 & J_{34}^{p} & 0 & 0 & 0 & 0\\ 0 & 0 & a_{34}^{p} & 1 & J_{45}^{p} & 0 & J_{47}^{p} & 0\\ 0 & 0 & 0 & a_{45}^{p} & 1 & J_{56} & 0 & 0\\ a_{16} & 0 & 0 & 0 & a_{56} & 1 & 0 & 0\\ 0 & 0 & 0 & a_{47}^{p} & 0 & 0 & 1 & J_{78}\\ a_{18} & 0 & 0 & 0 & 0 & 0 & a_{78} & 1\end{bmatrix}=\begin{bmatrix}1 & R & 0 & 0 & 0 & R & 0 & R\\ 6 & 1 & R & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 1 & S & 0 & 0 & 0 & 0\\ 0 & 0 & 5 & 1 & S & 0 & S & 0\\ 0 & 0 & 0 & 5 & 1 & R & 0 & 0\\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 0 & 5 & 0 & 0 & 1 & R\\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1\end{bmatrix}$$

The corresponding configuration is shown as in Fig. 2a.

Secondly, turning the link 6 in z with appropriate position, and then changing the orientation of the joint R_{16} 90° to form the orientation change metamorphic matrix $\mathbf{A}_S(R_{12},R_{16})$. By Eqs. (1) and (4), the joint orientation change matrix $\mathbf{J}_{16}^{c}$ and the orientation change metamorphic matrix $\mathbf{A}_S(R_{12},R_{16})$ can be formed as

$$\mathbf{J}_{16}^{c}=\begin{bmatrix}1 & J_{16}\\ a_{16}^{c} & 1\end{bmatrix}=\begin{bmatrix}1 & R\\ 6 & 1\end{bmatrix}$$

Fig. 2 Configuration of the mechanism when the orientation of the joint R_{12} is to be changed 90° and the orientation of the joint R_{12} and R_{16} are to be changed 90°



$$
\mathbf{A}_S(R_{12}, R_{16}) = \begin{bmatrix} 1 & J_{12} & 0 & 0 & 0 & J_{16} & 0 & J_{18} \\ a_{12}^c & 1 & J_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{23} & 1 & J_{34}^p & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{34}^p & 1 & J_{45}^p & 0 & J_{47}^p & 0 \\ 0 & 0 & 0 & a_{45}^p & 1 & J_{56} & 0 & 0 \\ a_{16}^c & 0 & 0 & 0 & a_{56} & 1 & 0 & 0 \\ 0 & 0 & 0 & a_{47}^p & 0 & 0 & 1 & J_{78} \\ a_{18} & 0 & 0 & 0 & 0 & 0 & a_{78} & 1 \end{bmatrix} = \begin{bmatrix} 1 & R & 0 & 0 & 0 & R & 0 & R \\ 6 & 1 & R & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & S & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 & S & 0 & S & 0 \\ 0 & 0 & 0 & 5 & 1 & R & 0 & 0 \\ 6 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 1 & R \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}
$$

The corresponding configuration is shown as in Fig. 2b.

Then, turning the link 8 in z with appropriate position, and then changing the orientation of the joint R_{18} 90° to form the orientation change metamorphic matrix $\mathbf{A}_S(R_{12}, R_{16}, R_{18})$. By Eqs. (1) and (4), the joint orientation change matrix $\mathbf{J}_{18}^c$ and the orientation change metamorphic matrix $\mathbf{A}_S(R_{12},R_{16},R_{18})$, i.e. $\mathbf{A}_S$ can be formed as

$$
\mathbf{J}_{18}^c = \begin{bmatrix} 1 & J_{18} \\ a_{18}^c & 1 \end{bmatrix} = \begin{bmatrix} 1 & R \\ 6 & 1 \end{bmatrix}
$$

$$
\mathbf{A}_S(R_{12}, R_{16}, R_{18}) = \mathbf{A}_S = \begin{bmatrix} 1 & J_{12} & 0 & 0 & 0 & J_{16} & 0 & J_{18} \\ a_{12}^c & 1 & J_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{23} & 1 & J_{34}^p & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{34}^p & 1 & J_{45}^p & 0 & J_{47}^p & 0 \\ 0 & 0 & 0 & a_{45}^p & 1 & J_{56} & 0 & 0 \\ a_{16}^c & 0 & 0 & 0 & a_{56} & 1 & 0 & 0 \\ 0 & 0 & 0 & a_{47}^p & 0 & 0 & 1 & J_{78} \\ a_{18}^c & 0 & 0 & 0 & 0 & 0 & a_{78} & 1 \end{bmatrix} = \begin{bmatrix} 1 & R & 0 & 0 & 0 & R & 0 & R \\ 6 & 1 & R & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & S & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 & S & 0 & S & 0 \\ 0 & 0 & 0 & 5 & 1 & R & 0 & 0 \\ 6 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 1 & R \\ 6 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}
$$

Finally, turning the joints R_{12}, R_{16}, and R_{18} of Fig. 3 with some degrees to form the spatial 3-*RRS* parallel mechanism as shown in Fig. 3 (z direction view), so that the orientation change metamorphic mechanism is formed.

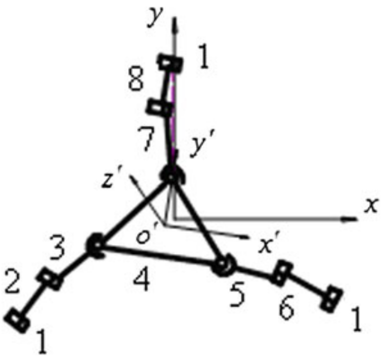


Fig. 3 Configuration of the orientation changed spatial mechanism

5 Conclusions

Based on proposed joint-axis matrix and an augmented adjacent matrix of kinematic chains, the axis orientation change metamorphic processes are presented. The matrix operation metamorphic process can be performed both manually and be executed by computer. The matrix can be used for the topological representation both for metamorphic mechanism and general kinematic chains, and can be extended in the study of structural synthesis of kinematic chains.

Acknowledgments The authors would like to acknowledge the financial support of the National Natural Science Foundation of China (NSFC) (Grant NO. 51175069) and State Key Laboratory of Robotics of China (2012-O16).

References

1. Dai, J.S., Rees John, J.: Mobility in metamorphic mechanisms of foldable/erectable kinds. Trans. ASME: J. Mech. Des. Writ. **121**(3), 375–382 (1999)
2. Parise, J.J., Howell, L.L., Magleby, S.P.: Ortho-planar mechanisms. In: Proceedings 26th Biennial Mechanisms and Robotics Conference Writing (2000)
3. Wang, D., Dai, J.S.: Structure synthesis of metamorphic mechanisms based on the configuration transformations. Chinese J. Mech. Eng. Writing **38**(7), 12–16 (2002)
4. Liu, C.H., Yang, T.L.: Essence and characteristics of metamorphic mechanisms and their metamorphic ways. In: Proceedings 11th World Congress in Mechanism and Machine Science, Writing, pp. 1285–1288 (2004)
5. Dai, J.S., Rees John, J.: Matrix Representation of Topological Changes in Metamorphic Mechanisms. Trans. ASME: J. Mech. Des. Writ. **127**(4), 610–619 (2005)
6. Dai, J.S., Rees John, J.: Theoretical foundation of metamorphic mechanisms and its synthesis. Chinese J. Mech. Eng. Writing **43**(8), 32–42 (2007)
7. Yan, H.S., Kuo, C.H.: Topological Representations and Characteristics of Variable Kinematic Joints. Trans. ASME J. Mech. Des. Writ. **128**(2), 384–391 (2006)

8. Li, S.J., Dai, J.S., Wang, D.: Topological presentation of kinematic chains with loops and orientation of joints axes. Chinese J. Mech. Eng. Writing **45**(6), 34–40 (2009)
9. Li, S.J., Dai, J.S.: A Matrix representation and motion space exchange metamorphic operation. Appl. Mech. Mater. Writ. **308–310**, 2058–2061 (2011)

Classification of Singularities in Kinematics of Mechanisms

Samuli Piipponen, Teijo Arponen and Jukka Tuomela

Abstract In this short article we will discuss methods of finding and classifying singularities of planar mechanisms. The key point is to observe that the configuration spaces of the mechanisms can be understood as *analytic and algebraic varieties*. The set of singular points of an algebraic variety is itself an algebraic variety and of lower dimension than the original one. The singular variety can be computed using the *Jacobian criterion*. Once the singular points are obtained their nature can be investigated by investigating the *localization* of the constraint ideal at *the local ring* at this point. This will tell us if the singularity is an intersection of several motion modes or a singularity of a particular motion mode. The nature of the singularity can be then analyzed further by computing *the tangent cone* at this point.

Keywords Kinematical singularities · Planar mechanisms · Algebraic geometry · Local rings · Tangent cone

1 Introduction

The equations of motion arising from Lagrangian mechanics for multibody systems are usually DAE equations where algebraic equations in our case determine the holonomic constraints. The constraint equations in Lagrangian mechanics in holonomic case are generally of form $g(u) = 0$ and $u : I \mapsto g^{-1}(0) \subset \mathbb{R}^k$ is the trajectory of the system which is the solution of the particular DAE. The set $g^{-1}(0)$ is the analytic or algebraic variety defining the kinematical properties of the system. In this article

S. Piipponen (✉) · T. Arponen · J. Tuomela
Department of Physics and Mathematics, University of Eastern Finland, Kuopio, Finland
e-mail: samuli.piipponen@uef.fi

T. Arponen
e-mail: teijo.arponen@uef.fi

J. Tuomela
e-mail: jukka.tuomela@uef.fi

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_5,

we will introduce methods of algebraic geometry to study the configuration spaces as algebraic varieties and investigate the nature of the possible singularities. The concept of a tangent cone has previously been used for singularity analysis in [1]. More classical methods to study singularities can be found from [2–4] and [5] where also the concept of singularities is little bit wider. One of the earliest works to use algebraic geometry in kinematical analysis was [6] and we have used it previously for example in [7–9].

List of Abbreviations

$\mathbb{A} = \mathbb{K}[x_1, \dots, x_k]$ denotes the *ring of polynomials* with coefficient field $\mathbb{K}$.

$\mathcal{I} = \langle g_1, \dots, g_n \rangle \subset \mathbb{A}$ denotes the *ideal* generated by polynomials $g_i \in \mathbb{A}$.

$\sqrt{\mathcal{I}}$ denotes the *radical* of the ideal $\mathcal{I}$.

$\mathbb{K}(\mathrm{V}(\mathcal{I})) = \{[f] \mid f \in \mathbb{A}\}$ is the *coordinate ring* defined by $\mathcal{I}$.

$\mathrm{V}(\mathcal{I}) \subset \mathbb{K}^k$ is the *algebraic variety* defined by $\mathcal{I}$.

O_p is the *localization of* $\mathbb{A}$ at p. Also called a *local ring* at p.

$\mathcal{O}_{V,p} \subset O_p$ is the *localization of* $\mathrm{V}(\mathcal{I})$ at p.

$\mathsf{I}_l(dg)$ is the ℓ:th *Fitting ideal* of matrix dg generated by its $\ell \times \ell$ minors.

$\Sigma(\mathrm{V}(\mathcal{I}))$ is the *singular variety* of $\mathrm{V}(\mathcal{I})$.

$C_p(\mathrm{V}(\mathcal{I}))$ is the *tangent cone* of $\mathrm{V}(\mathcal{I})$ at p.

2 Preliminary Definitions

In our terminology the dimension of a constraint variety is simply the mobility of the mechanism and the singular points are singular points of the corresponding variety. Let us present shortly the relevant definition and theorems in order to compute our examples. Remember that the *embedding dimension* of an algebraic variety is the minimal number of generators of M_p and $\mathrm{edim}(\mathcal{O}_{V,p}) = \dim_K(M_p/M_p^2)$. Particularly important is that the Krull dimension of an ideal can be easily computed if the elements of Gröbner basis of an ideal are known [10, 11].

Definition 2.1 (Singular and regular points of a variety) Let $\mathcal{I} \subset \mathbb{A}$ be a radical ideal. The local ring $\mathcal{O}_{V,p}$ is a *regular local ring* if

$$\dim_K(\mathcal{O}_{V,p}) = \mathrm{edim}(\mathcal{O}_{V,p}) = \dim(T_p\mathrm{V}(\mathcal{I})). \tag{1}$$

If the point p is not regular it is *singular*.

The last equation in definition 3 with *Krull's principal ideal theorem* [10] gives us actual means to compute the singular points [10, 11].

Theorem 2.1 (Jacobian Criterion) *Let $\mathcal{I} = \langle g_1, \dots g_n \rangle \subset \mathbb{K}[x_1, \dots, x_k]$ be a radical ideal let $\overline{\mathbb{K}}$ be closed extension field of $\mathbb{K}$ and suppose that $\mathrm{V}(\mathcal{I}) \subset \overline{\mathbb{K}}^k$ is equidimensional and dim$(\mathrm{V}(\mathcal{I})) = k - \ell$. Then the singular variety of $\mathrm{V}(\mathcal{I})$ is*

$$\Sigma(\mathrm{V}(\mathcal{I})) = \mathsf{V}\big(\mathcal{I} + \mathsf{I}_l(dg)\big) = \mathsf{V}\big(\mathcal{I}\big) \cap \mathsf{V}\big(\mathsf{I}_l(dg)\big) \subset \overline{\mathbb{K}}^k. \tag{2}$$

In other words if $p \in S(\mathrm{V}(\mathcal{I}))$ then $\mathcal{O}_{V,p}$ is not a regular local ring. Moreover if $1 \in \mathcal{I} + \mathsf{I}_l(dg)$ then the variety $\mathrm{V}(\mathcal{I})$ is naturally smooth since $\mathrm{V}(1) = \emptyset$.

Let us then introduce an other important object in our analysis the *tangent cone* [12]. With Taylor's formula we can expand any polynomial with respect to any point $p \in \mathbb{K}^k$ and present $f \in \mathbb{A}$ by total degree d as a linear combination

$$\begin{aligned} f &= f_{p,0} + \dots + f_{m,j} \\ f_{p,d} &= \sum_{|s|=d} a_s (x-p)^s, \end{aligned}$$

where $s = (s_1, \dots, s_k)$ and $s_1 + \dots + s_k = d$. The polynomial $f_{p,min}$ is the smallest part for which $f_{p,j} \neq 0$ in previous expansion.

Definition 2.2 (Tangent cone) Suppose that $\mathrm{V}(\mathcal{I}) \subset \mathbb{K}^k$ is an affine variety and let $p \in \mathrm{V}(\mathcal{I})$. The *Tangent cone* of $\mathrm{V}(\mathcal{I})$ at p_i, denoted by $C_p(\mathrm{V}(\mathcal{I}))$, is the variety

$$C_p(\mathrm{V}(\mathcal{I})) = \mathrm{V}(f_{p,min} \mid f \in I(\mathrm{V}(\mathcal{I}))), \tag{3}$$

Note that if we make the coordinate transformation of p to origin $C_0(\mathrm{V}(\mathcal{I}))$ is the best approximation of $\mathrm{V}(\mathcal{I})$ at 0 with variety of an homogeneous ideal of same dimension as $\mathrm{V}(\mathcal{I})$. The following theorem allows us also to distinguish between singular and regular points [12].

Theorem 2.2 *Assume that $\mathbb{K}$ is closed and $p \in \mathrm{V}$. Then the following conditions are equivalent*

$$p \in \mathrm{V} \textit{ is regular point of } \mathrm{V} \;\Leftrightarrow\; \dim(C_p(\mathrm{V})) = \dim(T_p\mathrm{V}) \;\Leftrightarrow\; C_p(\mathrm{V}) = T_p\mathrm{V}. \tag{4}$$

The next theorem allows us to recognize certain types of singularities [13].

Theorem 2.3 *Suppose that $\mathcal{I} \subset \mathbb{K}[x_1, \ldots, x_k]$ is an ideal where $\mathbb{K}$ is algebraically closed. Let $p \in \mathrm{V}(\mathcal{I})$ be a singular point of $\mathrm{V}(\mathcal{I})$ and $\mathcal{O}_p$ be the local ring at p. If the prime decomposition of the radical of $\mathcal{O}_{V,p}$ in the local ring is*

$$\sqrt{\mathcal{O}_{V,p}} = \mathcal{I}_1 \cap \ldots \cap \mathcal{I}_r \subset \mathcal{O}_p \tag{5}$$

then the corresponding irreducible varieties $\mathrm{V}(\mathcal{I}_i)$ of prime ideals $\mathcal{I}_i$ represent varieties passing through the singular point p and they intersect at this point. However if the prime decomposition is $\sqrt{\mathcal{O}_{V,p}} = \mathcal{O}_{V,p}$, then $\mathcal{O}_{V,p}$ is an integral domain and the point p is a singularity of an irreducible variety $\mathrm{V}(\mathcal{I})$.

3 Examples

In this section we will apply previous theorems to two relatively easy examples. Let us look at planar N-bar slider crank mechanism and planar closed one loop N-bar mechanism.

In general case it is perhaps shorter to treat the configuration spaces first as analytic varieties when proving necessary conditions for singularities[1].

Theorem 3.1 *Suppose that we have either Planar N-bar slider-crank mechanism or planar closed one loop N-bar mechanism so that the lengths of the bars are $l_1, \ldots, l_n$ as in Fig. 1. Then the necessary condition for existence of kinematical singularities is*

$$f(l_1, \ldots, l_n) = l_1 \pm l_2 \pm \ldots \pm l_n = 0, \quad l_i > 0 \; \forall \; 1 \leq i \leq n. \tag{6}$$

Proof In the case of the N-bar slider-crank mechanism the constraint map is $g{:}\mathbb{R}^n \mapsto \mathbb{R}$

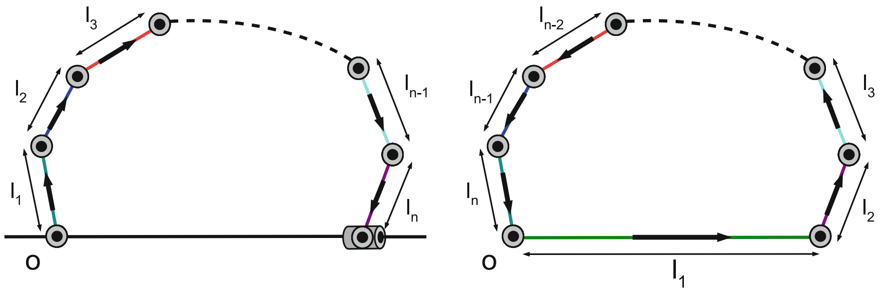


Fig. 1 On the *left*: Planar N-bar slider-crank mechanism. On the *right*: Planar closed one-loop N-bar mechanism

[1] This could have been done by transforming the analytic variety to algebraic variety but let us do that later.

$$g(\theta_1,\ldots,\theta_n) = l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + \ldots + l_n \sin(\theta_n).$$

The Jacobian of g is then the gradient $dg = \nabla g = (l_1 \cos(\theta_1),\ldots,l_n \cos(\theta_n))$. Now if $\theta^* = (\theta_1,\ldots,\theta_n)$ is singular point $\Leftrightarrow$ rank$(dg(\theta^*)) = 0$ and $\cos(\theta_i) = 0 \ \forall\ 1 \le i \le n$ which implies $\theta_i = \pi(1/2+n),\ n \in \mathbb{Z}$. Substituting this to constraint equation $g(\theta^*) = 0$ implies directly

$$g(\theta^*) = l_1 \pm l_2 \pm \ldots \pm l_n = 0.$$

In the case of N-bar planar single closed loop mechanism the constraint map is $\hat{g} := (g_1, g_2) : \mathbb{R}^n \mapsto \mathbb{R}^2$

$$\begin{cases} g_1(\theta_1,\ldots,\theta_n) = l_1 \cos(\theta_1) + \ldots + l_n \cos(\theta_n) \\ g_2(\theta_1,\ldots,\theta_n) = l_1 \sin(\theta_1) + \ldots + l_n \sin(\theta_n). \end{cases}$$

Now the Jacobian of $\hat{g}$ is the $2 \times n$ matrix

$$d\hat{g} = \begin{pmatrix} -l_1 \sin(\theta_1) & \ldots & -l_n \sin(\theta_n) \\ l_1 \cos(\theta_1) & \ldots & l_n \cos(\theta_n) \end{pmatrix}.$$

If $\theta^* = (\theta_1,\ldots,\theta_n)$ is singular point $\Leftrightarrow$ rank$(d\hat{g}(\theta^*)) < 2$ which is equivalent to the fact that all the 2×2 minors of $d\hat{g}$ have to vanish

$$\begin{vmatrix} -l_j \sin(\theta_j) & -l_i \sin(\theta_i) \\ l_j \cos(\theta_j) & l_i \cos(\theta_i) \end{vmatrix} = l_i l_j \sin(\theta_j - \theta_i) = 0 \ \ \forall\ \ 1 \le i, j \le n,\ i \ne j.$$

This is equivalent to $\theta_j = \theta_i + n\pi,\ n \in \mathbb{Z}$ and like in Fig. 1 without loss of generality in kinematical analysis we can choose $\theta_1 = 0$ so that $\theta_j = n\pi,\ n \in \mathbb{Z},\ \forall\ 1 \le j \le n$ and substituting this we get automatically $g_2(\theta^*) = 0$ and the first equation reveals the condition

$$g_1(\theta^*) = l_1 \pm l_2 \pm \ldots \pm l_n = 0.$$

4 Local Analysis of Examples

Let us then investigate the singularities locally first in slider-crank mechanism when $n = 2$ and $n = 3$. The configuration space was the analytic variety $g^{-1}(0)$. With substitutions $c_i = \cos(\theta_i),\ s_i = \sin(\theta_i)$ the general constraint equations take form

$$p_1 = l_1 s_1 + \ldots + l_n s_n = 0,\ \ p_{i+1} = c_i^2 + s_i^2 - 1 = 0,\ \ 1 \le i \le n. \qquad (7)$$

The configuration space is transformed to algebraic variety $\mathrm{V}(\langle p_1,\ldots,p_{n+1}\rangle)$ and the constraint mapping to $p : S^1 \times \ldots \times S^1 \subset \mathbb{R}^{2n} \mapsto \mathbb{R}^{n+1}$. In the case $n = 2$ we set $l_1 = l_2 = 1$ and compute singular variety using theorem (2.1)

$$\Sigma(\mathrm{V}(\mathcal{I})) = \mathrm{V}\big(\mathcal{I} + \mathsf{I}_3(dp)\big) = \{(0, 1, 0, -1), (0, -1, 0, 1)\} = q_1 \cup q_2 \subset (S^1)^2.$$

Let us investigate $q_2 = (0, -1, 0, 1)$. After transformation to origin the constraint ideal defined by takes form

$$\hat{\mathcal{I}} = \langle q_1, q_2, q_3\rangle = \langle b_1 + b_2, a_1^2 + (b_1 - 1)^2 - 1, a_2^2 + (b_2 + 1)^2 - 1\rangle.$$

Now we can compute the tangent cone and get

$$C_0(\mathrm{V}(\hat{\mathcal{I}})) = \mathrm{V}(\langle b_1, b_2, a_1^2 - a_2^2\rangle).$$

Near origin the variety $\mathrm{V}(\hat{\mathcal{I}})$ looks like two lines $s_1 = t(1, 0, 1, 0)$, $s_2 = t(1, 0, -1, 0)$, $t \in \mathbb{R}$ intersecting in the plane $b_1 = b_2 = 0$. Let us then compute the prime decomposition of local ring $\mathcal{O}_{V,0}$. As expected we have $\mathcal{O}_{V,0} = H_1 \cap H_2$. By theorem (2.3) two irreducible varieties/motion modes pass through q_2. In fact it is easy to find out that the configuration space breaks to irreducible components/motion modes $\mathrm{V}(\mathcal{I}) = \mathrm{V}(\mathcal{I}_1) \cup \mathrm{V}(\mathcal{I}_2)$ and $S(\mathrm{V}(\mathcal{I})) = \mathrm{V}(\mathcal{I}_1) \cap \mathrm{V}(\mathcal{I}_2)$.

Let us then do similar analysis for 3-bar slider crank mechanism. The constraint variety defined by (9) is now $\mathrm{V}(\mathcal{I}) = \mathrm{V}(\langle p_1, \ldots, p_4\rangle)$. First we set $l_1 = 2$, $l_2 = l_3 = 1$ and compute the singular variety again by theorem (2.1)

$$\Sigma(\mathrm{V}(\mathcal{I})) = \mathrm{V}(\mathcal{I} + \mathsf{I}_4(dp)) = \{(0, 1, 0, -1, 0, -1), (0, -1, 0, 1, 0, 1)\} = q_1 \cup q_2.$$

Next we investigate q_2 locally and after transformation to origin we have

$$\hat{\mathcal{I}} = \langle b_1 + b_2 + b_3, a_1^2 + (b_1 - 1)^2 - 1, a_2^2 + (b_2 + 1)^2 - 1, a_3^2 + (b_3 + 1)^2 - 1\rangle.$$

When we compute the tangent cone at origin we find

$$C_0(\mathrm{V}(\hat{\mathcal{I}})) = \mathrm{V}(\langle b_1, b_2, b_3, 2a_1^2 - a_2^2 - a_3^2\rangle).$$

so near the origin the variety appears to have a cone type singularity $2a_1^2 - a_2^2 - a_3^2 = 0$ at the hyper plane $b_1 = b_2 = b_3 = 0$. Next we compute the prime decomposition of localization $\mathcal{O}_{V,0}$ of $\mathrm{V}(\mathcal{I})$ at $\mathcal{O}_0$ and get $\sqrt{\mathcal{O}_{V,0}} = \mathcal{O}_{V,0}$. The theorem (2.3) tells us that the singularity is not an intersection of different motion modes/irreducible varieties which in this case agrees with the nature of the nature of the tangent cone. When $n = 2$ the configuration space breaks to two parts and when $n = 3$ such separation does not exist. The phenomena is clearly visible from the plots of configuration spaces in Fig. 2.

Let us then investigate planar closed one loop 4-bar mechanism. Let us fix again first bar to x-axis and without loss of generality choose $l_1 = 1$. With substitutions $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$ the constraint equations take form

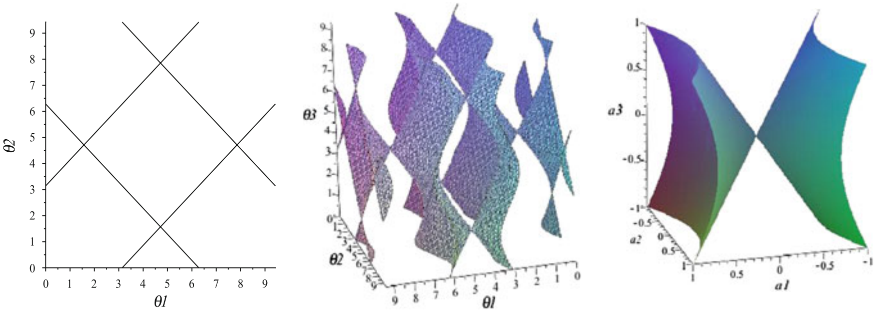


Fig. 2 On the *left*: Configuration space of 2-bar slider crank on (θ_1, θ_2)-space. In the *middle*: Configuration space of 3-bar slider crank on $(\theta_1, \theta_2, \theta_3)$-space. On the *right*: The nonlinear part $2a_1^2 - a_2^2 - a_3^2 = 0$ of the tangent cone $C_0(\mathrm{V}(\mathcal{I}))$ in (a_1, a_2, a_3)-space

$$p_1 = l_2c_2 \ldots + l_ns_n + 1 = 0$$
$$p_2 = l_2c_2 + \ldots + l_nc_n = 0,\ p_{i+1} = c_{i+1}^2 + s_{i+1}^2 - 1 = 0,\ 1 \le i \le n. \quad (8)$$

The configuration space is transformed to algebraic variety $\mathrm{V}(\langle p_1, \ldots, p_{n+1}\rangle)$ and the constraint mapping to $p : S^1 \times \ldots \times S^1 \subset \mathbb{R}^{2n-2} \mapsto \mathbb{R}^{n+1}$. In the case $n = 4$ we set $l_2 = l_3 = l_4 = 1$ and compute again the singular variety

$$\Sigma(\mathrm{V}(\mathcal{I})) = \mathrm{V}\big(\mathcal{I} + \mathrm{I}_5(dp)\big) = \{q_1, q_2, q_3\} \subset (S^1)^3,$$

where $\{q_1, q_2, q_3\} = \{(1, 0, -1, 0, -1, 0), (-1, 0, 1, 0, -1, 0), (-1, 0, -1, 0, 1, 0)\}$.

Now it is also straightforward to check that $\mathrm{V}(\mathcal{I})$ is union of three irreducible varieties/motion modes $\mathrm{V}(\mathcal{I}) = \mathrm{V}(\mathcal{I}_1) \cup \mathrm{V}(\mathcal{I}_2) \cup \mathrm{V}(\mathcal{I}_3)$. Let us still do the local analysis for $\mathrm{V}(\mathcal{I})$ for example at q_1. Let us then move q_1 to origin and denote the transformed constraint ideal as $\hat{\mathcal{I}}$. The tangent cone $C_0(\mathrm{V}(\hat{\mathcal{I}}))$ is

$$C_0(\mathrm{V}(\hat{\mathcal{I}})) = \mathrm{V}(\langle a_2, a_3, a_4, b_2 + b_3 + b_4, b_3b_4\rangle)$$

The singularity looks now again as two lines $s_1 = b_2(1, 0, -1),\ b_2 \in \mathbb{R}$ and $s_2 = b_2(1, -1, 0),\ b_2 \in \mathbb{R}$ intersecting at origin in the hyperplane $a_2 = a_3 = a_4 = 0$. The computation of prime decomposition of $\mathcal{O}_{V,0}$ confirms our previous computation $\mathcal{O}_{V,0} = H_1 \cap H_2$. Two irreducible varieties $\mathrm{V}(\mathcal{I}_1)$ and $\mathrm{V}(\mathcal{I}_3)$ intersect at point q_1 as the theorem (2.3) of suggests.

5 Conclusion

We applied computational algebraic geometry to three simple mechanism examples to find out the possible kinematical singularities using Jacobian criterion and further investigate their nature using concept of localization, local ring and tangent cone. Indeed we can conclude that the singularities in 2-bar slider crank and 4-bar mechanism are *removable singularities* in the sense that they are intersections of smooth assembly modes. However with 3-bar slider crank this is not the case and we can call these *essential singularities*. Although we investigated relatively simple examples the methods generalize to more complicated mechanisms as we will show in the future. For actual computations we have used a well established program Singular [14].

References

1. Müller, A., Rico, J.M.: Mobility and higher order local analysis of the configuration space of single-loops mechanisms. Adv. Rob. Kinematics **4**, 215–224 (2009)
2. Gosselin, C., Angeles, J.: Singularity analysis of closed loop kinematic chains. IEEE J. Robot. Autom. **6**(3), 119–132 (1990)
3. Sefrioui, J., Gosselin, C.: Singularity analysis and representation of planar parallel manipulators. J. Robot. Auton. Syst. **10**, 209–224 (1993)
4. Park, F.-C., Kim, J.-W.: Singularity analysis of closed kinematical chains. J. Mech. Des. **121**(1), 32–38 (1999)
5. Bandyopadhyay, S., Ghosal, A.: Analysis of configuration space singularities of closed-loop mechanisms and parallel manipulators. Mech. Mach. Theory **39**(5), 519–544 (2004)
6. Raghavan, M., Roth, B.: Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and manipulators. J. Mech. Des. **117**(Issue B), 71–79 (1995)
7. Arponen, T., Piipponen, S., Tuomela, J.: Kinematic analysis of Bricard's mechanism. Nonlinear Dyn. **56**(1–2), 85–99 (2009)
8. Arponen, T., Piipponen, S., Tuomela, J.: Analysis of singularities of a benchmark problem. Multibody Sys. Dyn. **19**(3), 227–253 (2008)
9. Piipponen, S.: Singularity analysis of planar linkages. Multibody Sys. Dyn. **22**(3), 223–243 (2009)
10. Kreuzer, M., Robbiano, L.: Computational Commutative Algebra 2. Springer-Verlag, Berlin (2005)
11. Greuel, G.-M., Pfister, G.: A Singular Introduction to Commutative Algebra. Springer-Verlag, Berlin (2002)
12. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms. Springer, New York (2007)
13. Matsumura, H.: Commutative Ring Theory, Cambridge Studies in Advanced Mathematics, volume 8, 2nd edn. Cambridge University Press, Cambridge (1989)
14. Greuel, G.-M., Pfister, G., Schönemann, H.: Singular 3.1.5. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern (2012)

Non-existence of Planar Projective Stewart Gough Platforms with Elliptic Self-Motions

Georg Nawratil

Abstract In this paper, we close the study on the self-motional behavior of non-architecturally singular parallel manipulators of Stewart Gough (SG) type, where the planar platform and the planar base are related by a projectivity κ, by showing that planar projective SG platforms with elliptic self-motions do not exist. The proof of this result demonstrates the power of geometric and computational interaction, but it also points out the limits of symbolic computation.

Keywords Self-motion · Stewart Gough platform · Borel Bricard problem

1 Introduction

The geometry of a planar Stewart Gough (SG) platform is given by the six base anchor points M_i located in the fixed plane π_{M} and by the six platform anchor points m_i of the moving plane π_{m}. If the geometry of the manipulator and the six leg lengths are given, the SG platform is in general rigid, but under particular conditions, it can perform an n-parametric motion ($n > 0$), which is called self-motion. Note that these motions are also solutions of the famous Borel Bricard problem (cf. [1–3]).

It is well known, that planar SG platforms, which are singular in every possible configuration, possess self-motions in each pose (over $\mathbb{C}$). These so-called architecturally singular planar SG platforms were extensively studied in [4–7]. Therefore, we are only interested in self-motions of planar SG platforms, which are not architecturally singular. Moreover, within this paper, we focus on the case, where the base anchor points M_i and the platform anchor points m_i are related by a

Dedicated to Prof. H. Stachel on the occasion of his 70th birthday.

G. Nawratil (✉)
Institute of Discrete Mathematics and Geometry, Vienna University of Technology,
Vienna, Austria
e-mail: nawratil@geometrie.tuwien.ac.at

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_6,

non-singular projectivity κ. For the remainder of this article, we call these manipulators planar projective SG platforms.

2 Self-Motions of Planar Projective SG Platforms

It is well known (cf. [6, 8, 9]), that a planar projective SG platform is architecturally singular, if and only if, one set of anchor points is located on a conic section, which can also be reducible.

The author proved in Lemma 1 of [10] that one can attach a two-parametric set $\mathcal{L}$ of additional legs to a planar projective SG platform without changing the forward kinematics and singularity surface. The platform anchor points m_i and the base anchor points M_i of these additional legs are also related by κ, i.e. $\kappa\colon \mathsf{m}_i \mapsto \mathsf{M}_i$.

Moreover, it was also shown by the author in [10] that non-architecturally singular planar projective SG platforms can either have pure translational self-motions or elliptic self-motions. Under consideration that s denotes the line of intersection of π_{M} and π_{m} in the projective extension of the Euclidean 3-space, the latter type of self-motions can be defined as follows (cf. Definition 1 of [10]):

Definition 2.1 A self-motion of a non-architecturally singular planar projective SG platform is called *elliptic*, if in each pose of this motion s exists with $\mathsf{s} = \mathsf{s}\kappa$ and the projectivity from s onto itself is elliptic.

Note, that an elliptic projectivity of a projectively extended line (line plus its ideal point) onto itself, is a bijective linear mapping, which does not have real fixed points. Therefore, Definition 2.1 implies that neither π_{M} and π_{m} nor two related points of the platform and the base coincide during an elliptic self-motion.

As the geometry of all manipulators with translational self-motions were already determined in [10], we focused on the study of elliptic self-motions in a recent publication [11], where the following results were obtained.

2.1 Results on Elliptic Self-Motions

Until now, it is an open question, whether planar projective SG platforms with elliptic self-motions even exist (cf. later givenConjecture 2.1). In the case of existence, these self-motions have to be one-parametric ones with instantaneously two degrees of freedom in each pose of the self-motion (cf. Theorems 1 and 2 of [11]).

It was also shown in [11], that the angle γ enclosed by the unique pair of ideal points (f, F) with $\mathsf{f}\kappa = \mathsf{F}$ has to remain constant during the self-motion of a planar projective SG platform. By introducing the nomenclature *orthogonal* for elliptic self-motions with $\gamma = \pi/2$, we can give Theorem 3 of [11]:

Theorem 2.1 *There do not exist non-architecturally singular planar projective SG platforms with an orthogonal elliptic self-motion.*

The proof of this theorem was done analytically, but not in the classical way (cf. Section 5.1 of [11]), as this approach resulted in a highly non-linear system of 17 equations in the design parameters, which we were not able to solve explicitly.

Instead, we developed an alternative method (cf. Section 5.2 of [11]), which is based on the algebraic formulation of two geometrically necessary conditions for achieving an elliptic self-motion. These two conditions imply two homogeneous polynomials Υ_1 and Υ_2 of degree 12 in two Euler parameters e_1 and e_2, which remain from the Study parameters, after a performed elimination process. Note, that each of these two polynomials has 1960 terms. A necessary condition for the existence of an orthogonal elliptic self-motion is, that Υ_1 and Υ_2 are fulfilled independently of the Euler parameters e_1 and e_2. Therefore, the coefficients of Υ_1 and Υ_2, with respect to e_1 and e_2, imply a system of 26 equations in the design parameters, which was used to prove Theorem 2.1.

Due to the above cited results, we had good reasons to close the paper [11] with the following conjecture:

Conjecture 2.1 *Non-architecturally singular planar projective SG platforms with an elliptic self-motion do not exist.*

Clearly, the first idea to prove this conjecture, is to do it similarly to Theorem 2.1. Indeed, the problem under consideration has only one more unknown, namely the angle γ, but exactly this additional variable effects enormously the computational complexity: The two corresponding polynomials Υ_1 and Υ_2 of the alternative method can be computed with MAPLE on a high capacity computer (78GB RAM). Each of these two expressions has 8259 terms and is again of degree 12 in e_1 and e_2 (cf. Remark 4 of [11]). We tried hard to solve the resulting system of 26 equations explicitly, but we failed due to its high degree of non-linearity.

Therefore, we have to come up with another idea for proving this conjecture. This new approach, presented in Sect. 4, is a purely geometric one, which is based on some old geometric/kinematic results listed in Sect. 3. Note, that the given proof also finishes the study of planar projective SG platforms with self-motions, whose results are summed up within the conclusions (cf. Sect. 5).

3 Related Historical Work

Before we list related historical results, we repeat some elementary facts on regular ruled quadrics, which are the one-sheeted hyperboloid and the hyperbolic paraboloid (see Fig. 1): Both surfaces carry two sets of generators, which are called regulus $\mathcal{R}$ and associated regulus $\mathcal{R}^{\times}$, respectively. Moreover, it should be noted that all lines within one set are skew to each other and that each line of one set is intersected by all lines of the other set. Therefore, a regular ruled quadric is uniquely determined by three pairwise skew generators. For more details, we refer to [12].

In 1873 the following theorem was given by Henrici (cf. [13]):

Theorem 3.1 *If the generators of a hyperboloid Φ of one sheet are constructed of rods, jointed at the points of crossing in a way that at each intersection point one rod is free movable about the other one, then the surface is not rigid, but permits a deformation into a one-parametric set $\mathcal{H}$ of hyperboloids.*

Moreover, Greenhill remarked in 1878 that $\mathcal{H}$ consists of confocal hyperboloids and that the trajectory of a point of Φ is orthogonal to this system of confocal hyperboloids. A proof of Greenhill's statement was given in 1879 by Cayley [14]. In 1899 Schur [15] presented a very elegant proof for Henrici's theorem and Greenhill's addendum, which also showed that these results remain valid if the one-sheeted hyperboloid is replaced by a hyperbolic paraboloid.

Finally, it should be noted that Wiener, who made some very nice models of these deformable one-sheeted hyperboloids and hyperbolic paraboloids (see Fig. 1), also gave a detailed review of this topic in Section 9 of [16].

Beside the cited results on the deformation of regular ruled quadrics, the following theorem is well known to the kinematic community (cf. page 222 of [17]):

Theorem 3.2 *If three points $\mathsf{m}_1, \mathsf{m}_2, \mathsf{m}_3$ of a line g run on spheres, where the centers $\mathsf{M}_1, \mathsf{M}_2, \mathsf{M}_3$ are also located on a line G, then every point m of g has a spherical trajectory, where the center M of this sphere belongs to G and fulfills the relation: $CR(\mathsf{m}_1, \mathsf{m}_2, \mathsf{m}_3, \mathsf{m}) = CR(\mathsf{M}_1, \mathsf{M}_2, \mathsf{M}_3, \mathsf{M})$, where CR denotes the cross-ratio.*

Moreover, it is a well known fact of projective geometry, that the one-parametric set of lines $[\mathsf{m}, \mathsf{M}]$ with m and M of Theorem 3.2 span a regulus $\mathcal{R}$ of a regular ruled quadric, if g and G are skew and $\mathsf{m}_1, \mathsf{m}_2, \mathsf{m}_3$ and $\mathsf{M}_1, \mathsf{M}_2, \mathsf{M}_3$ are pairwise distinct.

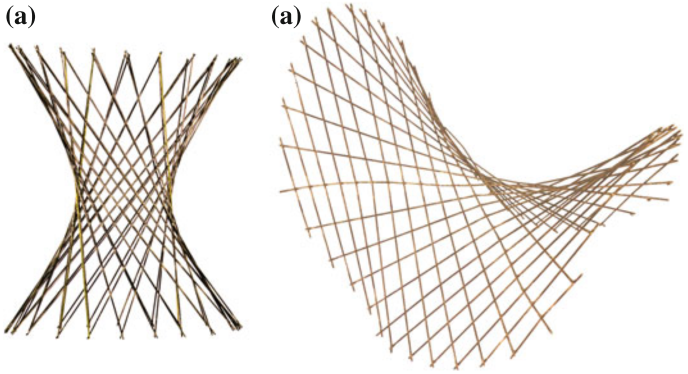


Fig. 1 Wiener's models of a deformable one-sheeted hyperboloid **a** and hyperbolic paraboloid **b** of the collection of mathematical models at the Institute of Discrete Mathematics and Geometry, Vienna University of Technology (see http://www.geometrie.tuwien.ac.at/modelle)

4 Proof of Conjecture 2.1

The proof of this conjecture is done by contradiction. We assume that a non-architecturally singular planar projective SG platform with base anchor points $\mathsf{M}_1, \ldots, \mathsf{M}_6$ and platform anchor points $\mathsf{m}_1, \ldots, \mathsf{m}_6$ exists, which possesses an elliptic self-motion $\mathcal{E}$. Without loss of generality, we can assume that the manipulator is in a pose of $\mathcal{E}$ where π_{m} and π_{M} are not parallel, as this would imply that κ is an affinity. But this affine case was already discussed in Theorem 5 of [10].

4.1 *Definition of a Special Planar Projective SG Platform*

Due to Lemma 1 of [10] and the results of [18], we can replace the original six legs $\overline{\mathsf{m}_i\mathsf{M}_i}$ with $i = 1, \ldots, 6$ by a new set of six legs $\overline{\mathsf{n}_i\mathsf{N}_i}$ without changing the direct kinematics and singularity surface, if $\mathsf{n}_i\kappa = \mathsf{N}_i$ holds and $\mathsf{n}_1, \ldots, \mathsf{n}_6$ are not located on a conic section. Therefore, $\mathsf{n}_1, \ldots, \mathsf{n}_6$ can be selected as follows (cf. Fig. 2):

We chose three lines $g_1, g_2, g_3 \in \pi_{\mathsf{m}}$ in a way that $g_1, g_2, g_3, \mathsf{s}$ are pairwise distinct and that no three of them belong to a pencil of lines. Then we can define n_i as the intersection point of g_i and $\mathsf{s} = \mathsf{s}\kappa$ for $i = 1, 2, 3$. Moreover, the intersection point of g_i and g_j is noted by n_{k+3} with pairwise distinct $i, j, k \in \{1, 2, 3\}$. By applying κ to $\mathsf{n}_1, \ldots, \mathsf{n}_6$, we get the corresponding base anchor points $\mathsf{N}_1, \ldots, \mathsf{N}_6$. It can easily be checked, that the resulting special planar projective SG platform is not architecturally singular. Moreover, we denote $g_i\kappa$ by G_i for $i = 1, 2, 3$.

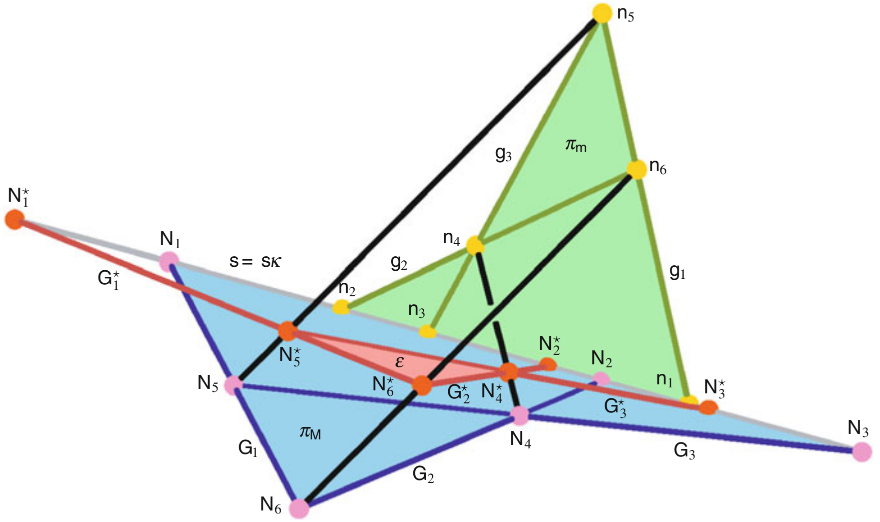


Fig. 2 Sketch and notation of the points, lines and planes used for the proof of Conjecture 2.1

Now we consider the one-parametric set of legs $\overline{\mathsf{nN}}$ with $\mathsf{n} \in g_1$, $\mathsf{N} \in \mathsf{G}_1$ and $\mathsf{n}\kappa = \mathsf{N}$. Due to Lemma 1 of [10], all these legs $\overline{\mathsf{nN}}$ can be added to the manipulator without disturbing the elliptic self-motion $\mathcal{E}$.[1] Moreover, the two lines g_1 and G_1 are skew ($\Leftrightarrow \mathsf{n}_1 \neq \mathsf{N}_1$), as the projectivity of s onto itself is elliptic. As a consequence, the one-parametric set $\mathcal{R}_1$ of lines $[\mathsf{n}, \mathsf{N}]$ is a regulus of a regular ruled quadric Φ_1. Due to the results of Henrici and Schur, we can even add arbitrary lines of the associated regulus $\mathcal{R}_1^{\times}$ to the mechanism without restricting the elliptic self-motion $\mathcal{E}$. Note, that the lines g_1 and G_1 also belong to $\mathcal{R}_1^{\times}$.

Clearly, analogous considerations for g_i, G_i yield the corresponding results for the reguli $\mathcal{R}_i, \mathcal{R}_i^{\times}$ of the regular ruled quadric Φ_i for $i = 2, 3$.

4.2 Planar Intersection of Φ_1

In the general case the planar intersection of Φ_1 is a conic section. But, if we assume that the plane ε contains the line $\mathsf{s} = \mathsf{s}\kappa$, which belongs to the regulus $\mathcal{R}_1$, then the conic degenerates into two distinct lines, which are $\mathsf{s} = \mathsf{s}\kappa$ itself and a line $\mathsf{G}_1^{\star}$ which belongs to the associated regulus $\mathcal{R}_1^{\times}$. Note, that ε is the tangent plane of Φ_1 in the intersection point of $\mathsf{s} = \mathsf{s}\kappa$ and $\mathsf{G}_1^{\star}$, which is denoted by $\mathsf{N}_1^{\star}$.

Clearly, analogous considerations hold for the surfaces Φ_2 and Φ_3, which also yield the points $\mathsf{N}_2^{\star}$ and $\mathsf{N}_3^{\star}$, respectively. Moreover, we introduce the notation $\mathsf{N}_{k+3}^{\star}$ for the intersection point of $\mathsf{G}_i^{\star}$ and $\mathsf{G}_j^{\star}$ with pairwise distinct $i, j, k \in \{1, 2, 3\}$. For these three points $\mathsf{N}_4^{\star}, \mathsf{N}_5^{\star}, \mathsf{N}_6^{\star}$ the following statement holds:

Lemma 4.1 *The points* $\mathsf{N}_4^{\star}, \mathsf{N}_5^{\star}, \mathsf{N}_6^{\star}$ *are pairwise distinct and do not belong to* $\mathsf{s} = \mathsf{s}\kappa$. *Moreover,* $\mathsf{N}_4^{\star}, \mathsf{N}_5^{\star}, \mathsf{N}_6^{\star}$ *are not collinear.*

Proof The point $\mathsf{N}_{k+3}^{\star}$ is located on the line $[\mathsf{n}_{k+3}, \mathsf{N}_{k+3}]$, which belongs to the reguli $\mathcal{R}_i$ and $\mathcal{R}_j$ for pairwise distinct $i, j, k \in \{1, 2, 3\}$. Therefore, these three lines $[\mathsf{n}_4, \mathsf{N}_4]$, $[\mathsf{n}_5, \mathsf{N}_5]$, $[\mathsf{n}_6, \mathsf{N}_6]$ are pairwise skew and not located within the platform π_{m}. As a consequence, the points $\mathsf{N}_4^{\star}, \mathsf{N}_5^{\star}, \mathsf{N}_6^{\star}$ are pairwise distinct and not located on $\mathsf{s} = \mathsf{s}\kappa$.

Now, we prove the second part of this lemma by contradiction. We assume that $\mathsf{N}_4^{\star}, \mathsf{N}_5^{\star}, \mathsf{N}_6^{\star}$ are located on a line L (cf. Fig. 3a). We denote the intersection point of L and $\mathsf{s} = \mathsf{s}\kappa$ by o. It should be noted, that $\mathsf{o} = \mathsf{N}_1^{\star} = \mathsf{N}_2^{\star} = \mathsf{N}_3^{\star}$ holds. Moreover, L belongs to the associated regulus $\mathcal{R}_4^{\times}$ of the regular ruled quadric Φ_4 defined by the regulus $\mathcal{R}_4$, which is spanned by the pairwise skew lines $[\mathsf{n}_4, \mathsf{N}_4]$, $[\mathsf{n}_5, \mathsf{N}_5]$ and $[\mathsf{n}_6, \mathsf{N}_6]$. Now, the unique line of $\mathcal{R}_4$ through o has to be $\mathsf{s} = \mathsf{s}\kappa$, as otherwise this point has to be a fixed point of the projectivity of s onto itself, which contradicts the definition of an elliptic self-motion. Therefore, the intersection of Φ_4 with π_{m} has to consist of $\mathsf{s} = \mathsf{s}\kappa$ and a second line containing the points $\mathsf{n}_4, \mathsf{n}_5, \mathsf{n}_6$, which already contradicts our assumptions of Section 4.1. □

[1] Note, that this can also be concluded as follows: As the cross ratio is invariant under projectivities the relation $CR(\mathsf{n}_1, \mathsf{n}_5, \mathsf{n}_6, \mathsf{n}) = CR(\mathsf{N}_1, \mathsf{N}_5, \mathsf{N}_6, \mathsf{N})$ holds. Then Theorem3.2 yields the results.

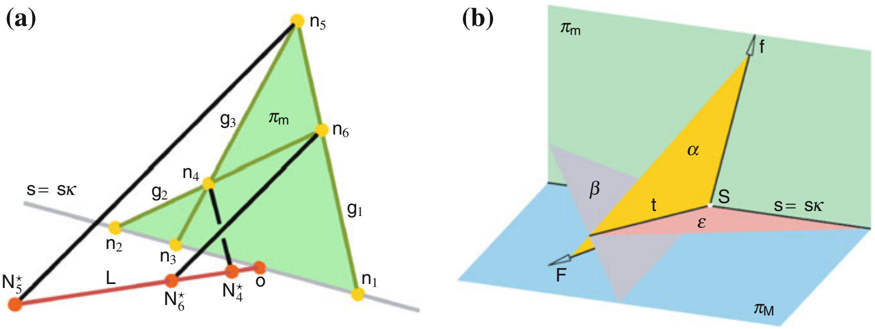


Fig. 3 **a** Sketch for the proof of the second part of Lemma 4.1. **b** Sketch of the construction for the special choice of the plane ε

4.3 Concluding the Proof

In order to verify Conjecture 2.1, we need one more lemma, which is given below:

Lemma 4.2 *There exists a non-singular projectivity $\kappa^\star$ with $\mathsf{n}_i\kappa^\star = \mathsf{N}_i^\star$ for $i = 1, \ldots, 6$. Therefore, the manipulator with platform anchor points $\mathsf{n}_1, \ldots, \mathsf{n}_6$ and base anchor points $\mathsf{N}_1^\star, \ldots, \mathsf{N}_6^\star$ is also a planar projective SG platform with an elliptic self-motion $\mathcal{E}^\star$.*

Proof Due to Lemma 4.1 the points $\mathsf{N}_1^\star, \mathsf{N}_2^\star, \mathsf{N}_4^\star, \mathsf{N}_5^\star$ always form a quadrangle. Therefore, the mapping $\mathsf{n}_i \mapsto \mathsf{N}_i^\star$ for $i = 1, 2, 4, 5$ uniquely defines a regular projectivity $\kappa^\star$. It can easily be seen by the collinearity properties of the anchor points, that also $\mathsf{n}_3\kappa^\star = \mathsf{N}_3^\star$ and $\mathsf{n}_6\kappa^\star = \mathsf{N}_6^\star$ hold.

Moreover, the elliptic self-motion $\mathcal{E}$ of the manipulator with platform anchor points $\mathsf{n}_1, \ldots, \mathsf{n}_6$ and base anchor points $\mathsf{N}_1, \ldots, \mathsf{N}_6$ is transmitted by the motion of the reguli $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ onto the manipulator with platform anchor points $\mathsf{n}_1, \ldots, \mathsf{n}_6$ and base anchor points $\mathsf{N}_1^\star, \ldots, \mathsf{N}_6^\star$. This resulting self-motion denoted by $\mathcal{E}^\star$ has to be elliptic, as a fixed point of the restriction of $\kappa^\star$ on $\mathsf{s} = \mathsf{s}\kappa^\star$ also has to be a fixed point of the restriction of κ on $\mathsf{s} = \mathsf{s}\kappa$. As this would contradict our assumption that $\mathcal{E}$ is an elliptic self-motion, we are done. □

Now the proof of the conjecture can be closed by giving the construction for a special choice of the plane ε (cf. Fig. 3b):

We consider any finite point $\mathsf{S} \in \mathsf{s} = \mathsf{s}\kappa$. This point spans together with the ideal points $\mathsf{f} \in \pi_\mathsf{m}$ and $\mathsf{F} \in \pi_\mathsf{M}$ (cf. Section 2.1) the plane α.[2] Now we intersect α with a plane β, which contains S and is orthogonal to the direction f. We denote the line of intersection by t. Then we chose ε as the plane spanned by $\mathsf{s} = \mathsf{s}\kappa$ and t.

Due to Lemma 4.2, the resulting planar projective SG platform with platform anchor points $\mathsf{n}_1, \ldots, \mathsf{n}_6$ and base anchor points $\mathsf{N}_1^\star, \ldots, \mathsf{N}_6^\star$ possesses an elliptic

[2] Note, that $\mathsf{f} \in \mathsf{s}$ or $\mathsf{F} \in \mathsf{s}\kappa$ cannot hold as this yields $\mathsf{f} = \mathsf{F}$, a contradiction.

self-motion $\mathcal{E}^{\star}$. According to the given construction, $\mathfrak{f}\kappa^{\star}$ equals the ideal point of $\mathfrak{t}$ and therefore $\mathcal{E}^{\star}$ is orthogonal. As planar projective SG platforms with such a self-motion do not exist (cf. Theorem 2.1), we end up with a contradiction. □

5 Conclusion

In this paper, we identified that the method based on the interaction of geometry and symbolic computation, which was used to prove Theorem 2.1, fails for solving the generalized problem formulated in Conjecture 2.1, due to the resulting computational complexity. By pure geometric reasonings, based on some historical results, we were able to verify Conjecture 2.1 by reducing the problem to the already solved one given in Theorem 2.1. This is a prime example for the fact that geometry is essential for solving advanced problems within the field of computational kinematics.

As the proof of Conjecture 2.1 also closes the study of planar projective SG platforms with self-motions, we can give the following main theorem under consideration of the results achieved in [10]:

Theorem 5.1 *A planar projective SG platform, which is not architecturally singular, can only have a self-motion if the projectivity is an affinity* $\mathbf{a}+\mathbf{A}\mathbf{x}$*, where the singular values* s_1 *and* s_2 *of the* 2×2 *transformation matrix* $\mathbf{A}$ *with* $0 < s_1 \leq s_2$ *fulfill the condition* $s_1 \leq 1 \leq s_2$*. All one-parametric self-motions of these manipulators are circular translations. Moreover, the self-motion is a two-dimensional translation, if and only if, the platform and the base are congruent and all legs have equal length.*

Acknowledgments This research is supported by Grant No. I 408-N13 of the Austrian Science Fund FWF within the project "Flexible polyhedra and frameworks in different spaces", an international cooperation between FWF and RFBR, the Russian Foundation for Basic Research.

References

1. Borel, E.: Mémoire sur les déplacements à trajectoires sphériques. Mémoire présenteés par divers savants à l'Académie des Sciences de l'Institut National de France **33**(1), 1–128 (1908)
2. Bricard, R.: Mémoire sur les déplacements à trajectoires sphériques, Journal de École Polytechn. (2), **11**, 1–96 (1906)
3. Husty, M.: E. Borel's and R. Bricard's Papers on displacements with spherical paths and their relevance to self-motions of parallel manipulators, international symposium on history of machines and mechanisms. In: Ceccarelli, M. (ed.), pp. 163–172, Kluwer (2000)
4. Karger, A.: Architecture singular planar parallel manipulators. Mechan. Mach. Theor. **38**(11), 1149–1164 (2003)
5. Nawratil, G.: On the degenerated cases of architecturally singular planar parallel manipulators. J. Geom. Graph. **12**(2), 141–149 (2008)
6. Röschel, O., Mick, S.: Characterisation of architecturally shaky platforms. In: Lenarcic, J., Husty, M.L. (eds.). Advances in Robot Kinematics: Analysis and Control, pp. 465–474. Kluwer (1998)

7. Wohlhart, K.: From higher degrees of shakiness to mobility. Mech. Mach. Theor. **45**(3), 467–476 (2010)
8. Chasles, M.: Sur les six droites qui peuvent étre les directions de six forces en équilibre. Comptes Rendus des Séances de l'Académie des Sci. **52**, 1094–1104 (1861)
9. Karger, A.: Singularities and self-motions of a special type of platforms, In: Lenarcic, J., Thomas, F. (eds.). Advances in Robot Kinematics: Theory and Applications, pp. 155–164. Springer (2002)
10. Nawratil, G.: Self-motions of planar projective Stewart Gough platforms, In: Lenarcic, J. Husty, M. (eds.). Latest Advances in Robot Kinematics, pp. 27–34. Springer (2012)
11. Nawratil, G.: On elliptic self-motions of planar projective Stewart Gough platforms. Trans. Canadian Soc. Mech. Eng. **37**(1) (2013) in press
12. Pottmann, H., Wallner, J.: Computational Line Geometry, Springer, Berlin (2001)
13. Dycks, W.: Katalog Mathematischer Modelle. Wolf, München (1892)
14. Cayley, A.: On the deformation of a model of a hyperboloid. Messenger Math. **8**, 51–52 (1879)
15. Schur, F.: Die Deformation einer geradlinigen Fläche zweiten Grades ohne Änderung der Längen ihrer Geraden. Zeitschrift für Mathematik und Physik **44**, 62–64 (1899)
16. Wiener, H.: Abhandlungen zur Sammlung mathematischer Modelle. Teubner, Leipzig (1907)
17. Kœnigs, G.: Leçons de Cinématique. Hermann, Paris (1897)
18. Borras, J., Thomas, F., Torras, C.: Singularity-invariant leg rearrangements in doubly-planar Stewart-Gough platforms. In: Proceedings of Robotics Science and Systems, Zaragoza, Spain (2010)

Direct Geometrico-Static Problem of Underconstrained Cable-Driven Parallel Robots with Five Cables

Ghasem Abbasnejad and Marco Carricato

Abstract The direct geometrico-static problem of cable-driven parallel robots with five cables is presented. The study provides procedures for the identification of all equilibrium poses of the end-effector when cable lengths are assigned. A least-degree univariate polynomial in the ideal governing the problem is obtained, thus showing that the latter has 140 solutions in the complex field. By a continuation technique, an upper bound on the number of real solutions is estimated. An algorithm based on parameter homotopy continuation is developed for the efficient computation of the whole solution set, including equilibrium poses with slack cables.

Keywords Cable-driven robots · Underconstrained robots · Kinematics · Statics.

1 Introduction

Cable-driven parallel robots (CDPRs) use cables instead of rigid-body legs to control the end-effector (EE) pose. CDPRs are *underconstrained* when the number of cables in tension (namely, *active*) is smaller than the number of degrees of freedom that the EE possesses with respect to the base. In this case, only some freedoms may be controlled, and the EE configuration depends on the applied forces, e.g. gravity [13]. The displacement analysis of these robots requires the simultaneous solution of both loop-closure and equilibrium equations. As a consequence, the *direct geometrico-static problem* (DGP), which aims at finding all equilibrium poses of the EE when cable lengths are assigned, is especially challenging [10, 12].

When the cable lengths of a CDPR with n cables, $n \leq 6$, are assigned as inputs, the number of active cables at the equilibrium is, a priori, unknown. Indeed, the EE may reach an equilibrium pose with one or more cables being slack. Since multiple

G. Abbasnejad · M. Carricato (✉)
Department of Industrial Engineering and Interdepartment Center for Health Sciences and Technologies, University of Bologna, Viale Risorgimento 2, Bologna, Italy
e-mail: marco.carricato@unibo.it

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_7,

stable configurations may exist, possibly characterized by different numbers of taut cables, the robot may switch between them because of inertia forces or external disturbances. Accordingly, the computation of the whole solution set for a given DGP is essential for robust trajectory planning.

The authors have solved so far the DGP of robots with two, three and four active cables [3–5]. These problems were proven to admit 24, 156 and 216 solutions in the complex field, respectively. The DGP of a CDPR suspended by 6 cables admits 40 solutions, since it is equivalent to the forward displacement analysis of the Gough platform [11]. The present contribution shows that the DGP of a CDPR with 5 active cables admits 140 solutions in the complex field, and it also estimates an upper bound on the number of real configurations. Parameter homotopy continuation is used to develop an efficient algorithm to determine the whole solution set. The results reported in the paper complete the authors' study concerning the DGP of underconstrained CDPRs with generic geometry.

2 Geometrico-Static Model

The EE is connected to a fixed base by five cables, which are modeled as inextensible and massless (Fig. 1). The ith cable, $i = 1, \ldots, 5$, is assigned length ρ_i, it exits from the base at point A_i, and it is connected to the EE at point B_i. $\mathcal{A}$ is a fixed Cartesian coordinate frame with origin at A_1, and $\mathcal{B}$ is a Cartesian frame attached to the EE at point G. The EE pose is described by $\mathbf{X}^T = [\mathbf{g}^T; \boldsymbol{\Phi}^T]$, where $\mathbf{g}^T = [x, y, z]^T$ is the position of G in $\mathcal{A}$, and $\boldsymbol{\Phi}^T = [e_1, e_2, e_3]^T$ is the array grouping the Rodrigues parameters parameterizing the EE orientation with respect to $\mathcal{A}$. The EE is acted upon by force $Q\mathcal{L}_e$, where Q is a constant magnitude and $\mathcal{L}_e$ is the normalized Plücker vector of the force line of action, passing through G and parallel to direction $\mathbf{k}$, without loss of generality. The normalized Plücker vector of line $A_i B_i$ is $\mathcal{L}_i/\rho_i$,

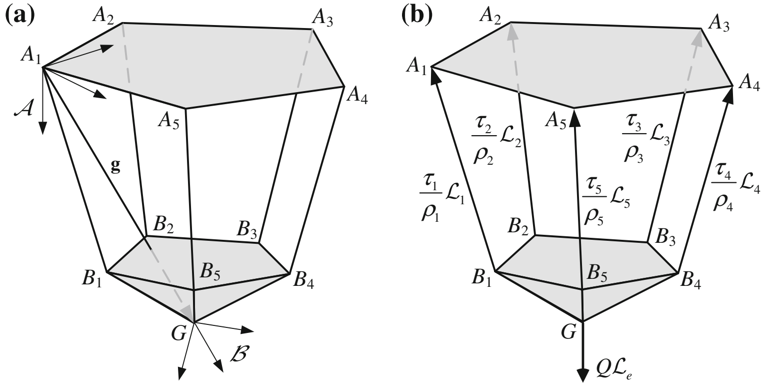

(a)
A_1
A_2
A_3
A_4
A_5
$\mathcal{A}$
$\mathbf{g}$
B_1
B_2
B_3
B_4
B_5
G
$\mathcal{B}$
(b)
A_1
A_2
A_3
A_4
A_5
$\frac{\tau_1}{\rho_1}\mathcal{L}_1$
$\frac{\tau_2}{\rho_2}\mathcal{L}_2$
$\frac{\tau_3}{\rho_3}\mathcal{L}_3$
$\frac{\tau_4}{\rho_4}\mathcal{L}_4$
$\frac{\tau_5}{\rho_5}\mathcal{L}_5$
B_1
B_2
B_3
B_4
B_5
G
$Q\mathcal{L}_e$

Fig. 1 A CDPR with four cables: geometric model (**a**) and static model (**b**)

where $\mathcal{L}_i^T = \left[(A_i - B_i)^T; \{(A_i - A_1) \times (A_i - B_i)\}^T\right]$. If τ_i is the intensity of the cable tensile force, the wrench exerted by the ith cable on the EE is $(\tau_i/\rho_i)\,\mathcal{L}_i$, and static equilibrium may be expressed as

$$\sum_{i=1}^{5} \frac{\tau_i}{\rho_i}\mathcal{L}_i + Q\mathcal{L}_e = \underbrace{\left[\,\mathcal{L}_1\ \mathcal{L}_2\ \mathcal{L}_3\ \mathcal{L}_4\ \mathcal{L}_5\ \mathcal{L}_e\,\right]}_{\mathbf{M}} \begin{bmatrix} (\tau_1/\rho_1) \\ (\tau_2/\rho_2) \\ (\tau_3/\rho_3) \\ (\tau_4/\rho_4) \\ (\tau_5/\rho_5) \\ Q \end{bmatrix} = \mathbf{0}. \tag{1}$$

If *all* cables are active, the following five geometrical constraints must be satisfied:

$$q_i := ||A_i - B_i||^2 - \rho_i^2 = 0, \quad i = 1, \ldots, 5, \tag{2}$$

Equations (1) and (2) amount to 11 scalar relations in 11 variables, namely $\mathbf{X}$ and τ_i, $i = 1, \ldots, 5$. Following Ref. [5], cable tensions may be eliminated from the set of unknowns by observing that Eq. (1) holds only if

$$p := \det \mathbf{M} = 0, \tag{3}$$

which is a purely geometrical condition, since $\mathbf{M}$ only depends on $\mathbf{X}$. Equations (2) and (3) amount to six relations in six pose coordinates. Polynomials $q_1, \ldots, q_5$ in Eq. (2) have degree 4 in $\mathbf{X}$, whereas polynomial p in Eq. (3) has degree 9 in $\mathbf{X}$. The 0-dimensional variety V of the ideal $\langle J \rangle$ generated by the set $J = \{q_1, \ldots, q_5, p\}$ yields the solutions of the DGP at hand.

3 Problem-Solving Algorithm

Like in the studies concerning the DGP of robots suspended by two, three and four cables [3–5], a formal proof about the number of solutions contained in V is provided by implementing an elimination procedure based on Groebner bases and Sylvester dialytic method. Then, a numerical algorithm based on homotopy continuation is presented to compute the solution set in an efficient way.

3.1 The Elimination Approach

In order to ease numeric computation via a computer algebra system, namely the `GroebnerPackage` provided within the software `Maple15`, all geometric parameters of the robot are assigned generic *rational values*. Accordingly, $\langle J \rangle \subset \mathbb{Q}[\mathbf{X}]$, where $\mathbb{Q}[\mathbf{X}]$ is the set of all polynomials in $\mathbf{X}$ with coefficients in $\mathbb{Q}$. All Groebner bases are computed with respect to graded reverse lexicographic monomial orders (grevlex, in brief), which provide the most efficient calculations. In general, a Groebner basis $G[J]$ of $\langle J \rangle$ with respect to grevlex($\mathbf{X}$), with variables ordered so that $z > y > x > e_1 > e_2 > e_3$, may be computed in a fairly expedited way. For instance, for the robot reported in Table 1, `Maple` computes $G[J]$ in roughly 3.3 min, on a PC with a 2.67 GHz Intel Xeon processor and 4 GB of RAM.

Once $G[J]$ is known, the normal set $\mathbf{N}[J]$ of $\langle J \rangle$, i.e. the set of all monomials that are *not* multiples of any leading monomial in $G[J]$, may be easily computed. Since $\mathbf{N}[J]$ comprises 140 monomials, this is also the number of complex roots in V and, thus, the order of the least-degree univariate polynomials of $\langle J \rangle$ [14]. Any one of these polynomials may be computed by the hybrid approach proposed in [3], based on the cooperative use of the FGLM algorithm [9] and a dialytic procedure similar to that presented in [7].

If $\mathbf{X}_l$ is a list of l variables in $\mathbf{X}$ and $\mathbf{X}\backslash\mathbf{X}_l$ is the relative complement of $\mathbf{X}_l$ in $\mathbf{X}$, a monomial order $>_l$ on $\mathbb{Q}[\mathbf{X}]$ is of l-elimination type provided that any monomial involving a variable in $\mathbf{X}_l$ is greater than any monomial in $\mathbb{Q}[\mathbf{X}\backslash\mathbf{X}_l]$. If $G_{>l}[J]$ is a Groebner basis of $\langle J \rangle$ with respect to $>_l$, then $G[J_l] := G_{>l}[J] \cap \mathbb{Q}[\mathbf{X}\backslash\mathbf{X}_l]$ is a basis of the lth elimination ideal $\langle J_l \rangle := \langle J \rangle \cap \mathbb{Q}[\mathbf{X}\backslash\mathbf{X}_l]$ [6]. The FGLM algorithm may be conveniently used to convert $G[J]$ from grevlex ($\mathbf{X}$) to $>_l$, so that $G[J_l]$ may be readily isolated from $G_{>l}[J]$. In particular, the FGLM algorithm may be used to compute the Groebner basis $G[J_3]$ of $\langle J_3 \rangle$, where the latter is the set of polynomials of $\langle J \rangle$ that contain monomials in e_1, e_2 and e_3 only. Eliminating more unknowns by the FGLM algorithm is not convenient, since memory usage and computation time exponentially increase with l. A more efficient alternative emerges by computing a Sylvester-type eliminant matrix from the polynomials of $G[J_3]$. Since $G[J_3]$ comprises 31 polynomials and 31 monomials in e_1 and e_2, if e_3 is assigned the role of 'hidden' variable, the generators of $G[J_3]$ may be set up as

$$\mathbf{T}\,(e_3)\,\mathbf{E} = \mathbf{0}, \tag{4}$$

where $\mathbf{T}(e_3)$ is a 31×31 matrix polynomial in e_3, and $\mathbf{E}$ is a vector grouping the 31 monomials in $G[J_3]$ with variables in $\{e_1, e_2\}$. As expected, letting the determinant of $\mathbf{T}(e_3)$ vanish yields a spurious-root-free polynomial of degree 140 in e_3.

Sylvester dialytic elimination may be applied to the Groebner basis of *any* elimination ideal $\langle J_l \rangle$ of $\langle J \rangle$. However, the smaller l, the higher the order of the eliminant matrix, and the more onerous the expansion of its determinant. Accordingly, the fewer variables are eliminated by the FGLM algorithm, the smaller the computation burden of the FGLM step, but the more demanding the Sylvester elimination. Numerical experimentation seems to indicate that the elimination of x, y and z by the FGLM algorithm, and successively of e_1 and e_2 by a dialytic step, provides the best compromise. The `Maple` implementation of the above procedure is able to compute the univariate polynomial in e_3 for the example reported in Table 1 in roughly 610 min, which is a substantial achievement for a polynomial of order 140.

3.2 Numerical Computation of the Solution Set

The univariate polynomial obtained in Sect. 3.1 is important under a theoretical viewpoint, but it has a too high degree for a practical use. For the numerical calculation of the solution set, homotopy continuation provides a more robust and efficient alternative. In this perspective, the complexity and degree of polynomial p in Eq. (3) are a disadvantage, since they slow down computation and cause stability problems (cf. [4]). For this reason, the formulation of static equilibrium via Eq. (1) and a new parametrization of the EE pose are preferable.

Without loss of generality, unit vectors $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ of the coordinates axes of $\mathcal{B}$ may be chosen so that $\mathbf{u}$ is directed from G to B_1, $\mathbf{v}$ lies in plane GB_1B_2, and $\mathbf{w} = \mathbf{u} \times \mathbf{v}$. If $[\mathbf{u}]_{\mathcal{A}} = [u_1, u_2, u_3]$ and $[\mathbf{v}]_{\mathcal{A}} = [v_1, v_2, v_3]$ are the projections of $\mathbf{u}$ and $\mathbf{v}$ in the fixed frame, the platform pose may be written as a function of the nine variables x, y, z, u_1, u_2, u_3, v_1, v_2, and v_3, which must satisfy the conditions

$$q_6 := \mathbf{u}^T\mathbf{u} - 1 = 0, \quad q_7 := \mathbf{v}^T\mathbf{v} - 1 = 0, \quad q_8 := \mathbf{u}^T\mathbf{v} = 0. \tag{5}$$

Equations (1), (2) and (5) form a system I of 14 scalar equations in 14 variables, i.e.

$$\mathbf{Y} = [x,\ y,\ z,\ u_1,\ u_2,\ u_3,\ v_1,\ v_2,\ v_3,\ \tau_1,\ \tau_2,\ \tau_3,\ \tau_4,\ \tau_5]^T . \tag{6}$$

Though I involves more variables and more equations than J, it comprises simpler lower-order polynomials, which are stabler when homotopy continuation is implemented, thus leading to a faster computation. In particular, polynomials q_1 and q_2 in Eq. (2) and q_6, q_7 and q_8 in Eq. (5) have degree 2 in $\mathbf{Y}$; q_3, q_4 and q_5 in Eq. (2) have degree 4 in $\mathbf{Y}$; and all polynomials in Eq. (1) have degree 3 in $\mathbf{Y}$. On the basis of these degrees, the problem at hand may be cast into the larger family of systems made up by five quadratic, six cubic and three quartic equations on $\mathbf{Y} \in \mathbb{P}^{14}$. By counting solutions at infinity, a general member of this family has a number of isolated roots equal to the minimal multi-homogeneous Bezout number [14]. This is also the number of paths tracked by the homotopy-continuation software used in this paper, i.e. `Bertini` [2]. By searching all possible multi-homogenizations, the minimal Bezout number emerges when $\mathbf{Y}$ is partitioned as $[\{x, y, z, u_1, u_2, u_3\}, \{v_1, v_2, v_3\}, \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}]$, and it is equal to 11, 520.

When the isolated roots of the DGP of a generic robot are known, *parameter-homotopy continuation* [14] may be used to find the solutions for any other DGP of the same kind, in an efficient way. Since the coefficients of the equations in I are continuous functions of the geometric parameters $\mathbf{P}$ of the robot, a continuous path through parameter space determines a continuous evolution of the coefficients and, generally, continuous paths for the solutions as well. Accordingly, if the 140 isolated roots of I are known for a generic $\mathbf{P} = \mathbf{P}_0$, the solutions for any other $\mathbf{P}$ may be found by tracking the homotopy

$$I\left(\mathbf{Y}, (1-t)\,\mathbf{P}_0 + t\mathbf{P}\right) = \mathbf{0}, \tag{7}$$

Table 1 Real equilibrium configurations with nonnegative cable tensions of a five-cable robot with: $\mathbf{a}_1 = [0, 0, 0]$, $\mathbf{a}_2 = [1, 2, -0.75]$, $\mathbf{a}_3 = [3.5, 1, 1]$, $\mathbf{a}_4 = [3.25, -1, 1]$, $\mathbf{a}_5 = [1, -2, -0.5]$, $\mathbf{b}_1 = [-1, 0, -1]$, $\mathbf{b}_2 = [-0.5, 1, -1.25]$, $\mathbf{b}_3 = [0.75, 0.75, -1.25]$, $\mathbf{b}_4 = [0.5, -0.75, -1.25]$, $\mathbf{b}_5 = [-0.25, -0.8, -1.5]$, $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5) = (4.5, 5, 3, 3.75, 4.75)$, $Q = 10$ and $\mathbf{k} = [0, 0, 1]$

No.	(x, y, z)	(e_1, e_2, e_3)	$(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$	$\mathbf{H}_r$
1	1.4589, −1.2145, 2.4744	30.8167, 76.5978, −18.4146	1.73, 0.66, 0.62, 1.72, 6.44,	<
2	1.4441, −1.3000, 2.4723	14.9565, 39.3587, −10.5282	2.85, 0, 0.03, 2.75, 5.75,	<
3	1.4227, −1.2821, 2.4725	19.3442, 45.2411, −12.2058	2.87, 0.10, 0, 2.70, 5.72,	<
4	1.2273, −0.1010, 1.8629	30.4252, −1.1058, −3.3465	0, 5.27, 0.99, 0, 6.37,	<>
5	2.3443, 1.0649, 2.1627	4.5139, 5.2434, 1.4044	0, 6.95, 3.15, 1.97, 0,	<>
6	1.4384, −1.2969, 2.4722	15.5535, 39.7125, −10.6817	2.91, 0, 0, 2.78, 5.71,	<
7	2.4803, 0.5613, 1.8416	53.8079, −123.9777, −16.1938	5.78, 0, 7.44, 0, 1.20,	<>
8	2.1463, 0.2983, 5.3178	−0.1057, 0.0133, −0.0550	3.18, 0, 5.48, 0, 1.65,	>
9	2.2516, 0.3107, 4.0816	−0.0967, −0.5150, 15.2829	4.75, 0, 6.96, 0, 3.96,	<>
10	2.6647, 1.2853, 2.0725	7.4388, 42.2955, 13.4023	0.61, 5.33, 5.77, 0, 0,	<>
11	2.5662, −0.1206, 1.4236	35.9962, −26.9853, 3.5481	0, 0, 7.04, 0, 8.28,	<>
12	1.0000, −0.0768, 1.8522	19.0651, 1.5022, −2.5059	0, 6.27, 0, 0, 6.21,	<>
13	2.6694, 1.3323, 2.0672	5.7677, 19.8499, 6.4974	0, 5.94, 5.60, 0, 0,	<>
14	2.4243, 0.6927, 1.8269	−19.0280, 57.3828, 10.2733	7.05, 0, 7.66, 0, 0,	<>

with t varying from 0 to 1 or, more robustly, along the curve $t = \gamma t' / \left[1 + (\gamma - 1)t'\right]$, with $t' \in [0, 1]$ and $\gamma \in \mathbb{C}$. In this case, only 140 paths need to be tracked, and paths corresponding to solutions at infinity are avoided. By this approach, `Bertini` converges to the solutions of the example reported in Table 1 in roughly 4.28 min (with the default settings). Among these solutions, only two are real, and only one has positive tension in all cables. The latter solution is listed in row 1 of Table 1.

3.3 Maximum Number of Real-Valued Solutions

The DGP of a CDPR suspended by five cables has 140 solutions in the complex field. However, since some roots may remain complex no matter how robot parameters are varied, the maximal number of real solutions may be smaller than 140. Determining a tight bound for this count is a challenging task. By a continuation procedure originally proposed by Dietmaier [8], and recently adapted by the authors to the DGP of underconstrained CDPRs [1], several sets of geometric parameters for which the DGP provides at the most 74 real configurations have been found so far. An example is as follows: $\mathbf{a}_2 = [1.44417, 0, 1.20333]$, $\mathbf{a}_3 = [0.302415, 1.26206, 0.55533]$, $\mathbf{a}_4 = [-0.711127, 0.808726, 0.810451]$, $\mathbf{a}_5 = [0.749568, 0.761578, -0.469085]$, $\mathbf{b}_1 = [2.16169, 0, 0]$, $\mathbf{b}_2 = [-0.125711, 0, 1.32615]$, $\mathbf{b}_3 = [-0.412791, 0.0211425, 0.449869]$, $\mathbf{b}_4 = [-0.16265, -0.468249, -0.399945]$, $\mathbf{b}_5 = [1.59653, 1.31446, 0.96224]$, $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5) = (2.46449, 1.99586, 1.20622, 1.42395, 2.4302)$.

4 Equilibrium Configurations with Unloaded Cables

When cable lengths are assigned as inputs, nothing ensures, a priori, that when the EE reaches its stable equilibrium pose all cables are in tension, since configurations may exist in which the EE is supported by only m cables, with $m \leq 5$ and $5 - m$ cables being slack. Accordingly, the overall solution set emerges by solving the DGP for all possible constraint sets $\{\parallel A_j - B_j \parallel = \rho_j, j \in \mathcal{W}\}$, with $\mathcal{W} \subseteq \{1, 2, 3, 4, 5\}$ and $\text{card}(\mathcal{W}) \leq 5$. Clearly, when the kth cable is slack, the distance $\parallel A_k - B_k \parallel$ cannot be greater than the assigned ρ_k. Hence, for any subset $\mathcal{W}$, only the solutions for which $\parallel A_k - B_k \parallel \leq \rho_k$, for all $k \notin \mathcal{W}$, must be retained. In general, for a robot with five cables, 31 DGPs need to be solved, namely one DGP with five active cables, five DGPs with four active cables, ten DGPs with three active cables, ten DGPs with two active cables, and five DGPs with one active cable.

Table 1 shows the overall results for an exemplifying geometry. Due to space limitations, only the real solutions with nonnegative tension in all cables are reported. These comprise one configuration with five cables in tension (row 1), two configurations with four cables in tension (rows 2–3), seven configurations with three cables in tension (rows 4–10), and four configurations with two cables in tension (rows 11–14). Stability is assessed by means of a reduced Hessian matrix $\mathbf{H}_r$, as proposed in Ref. [5]. Symbols $>$, $<$ and $<>$ denote, respectively, a positive-definite, a negative-definite and an indefinite matrix. A solution is stable when $\mathbf{H}_r$ is positive-definite. For the case at hand, the robot has a single stable configuration, with three cables in tension.

5 Conclusions

This paper studied the direct geometrico-static problem (DGP) of underconstrained cable-driven parallel robot with five cables. The task consists in finding all equilibrium configurations that are compatible with the assigned cable lengths. Since the equations governing the problem comprise both geometrical and static constraints, the DGP is a challenging task.

A least-degree univariate polynomial was numerically obtained by an elimination procedure, thus showing that 140 solutions exist in the complex field. A continuation algorithm was then developed to identify a robot geometry leading to the highest number of real equilibrium configurations. A bound of 74 real configurations was estimated. After the solutions with nonnegative tension in all cables are sifted and stability is considered, the number of feasible configurations decreases remarkably.

For the efficient computation of the whole solution set, including configurations with slack cables, an algorithm based on parameter homotopy continuation was developed. The algorithm is the foundation of the software $\texttt{DGP} - \texttt{Solver}$, which is currently being developed by the authors to automatize the computation of all equilibrium configurations of a CDPR with an arbitrary number of cables.

References

1. Abbasnejad, G., Carricato, M.: Real solutions of the direct geometrico-static problem of under-constrained cable-driven parallel robots with 3 cables: a numerical investigation. Meccanica **47**(7), 1761–1773 (2012)
2. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Bertini: software for numerical algebraic geometry. http://www.nd.edu/~sommese/bertini
3. Carricato, M.: Direct geometrico-static problem of underconstrained cable-driven parallel robots with three cables. ASME J. Mech. Rob. **5**(3), 031008 (2013)
4. Carricato, M., Abbasnejad, G.: Direct geometrico-static analysis of under-constrained cable-driven parallel robots with 4 cables. In: Bruckmann, T., Pott, A. (eds.) Cable-Driven Parallel Robots, pp. 269–285. Springer-Verlag, Berlin Heidelberg (2013)
5. Carricato, M., Merlet, J.P.: Stability analysis of underconstrained cable-driven parallel robots. IEEE Trans. Rob. **29**(1), 288–296 (2013)
6. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms. Springer, New York (2007)
7. Dhingra, A.K., Almadi, A.N., Kohli, D.: A Gröbner–Sylvester hybrid method for closed-form displacement analysis of mechanisms. ASME J. Mech. Des. **122**(4), 431–438 (2000)
8. Dietmaier, P.: The Stewart-Gough platform of general geometry can have 40 real postures. In: Lenarčič, J., Husty, M.L. (eds.) Advances in Robot Kinematics: Analysis and Control, pp. 7–16. Kluwer Academic Publishers, Dordrecht (1998)
9. Faugère, J.C., Gianni, P., Lazard, D., Mora, T.: Efficient computation of zero-dimensional Gröbner bases by change of ordering. J. Symb. Comput. **16**(4), 329–344 (1993)
10. McCarthy, J.M.: 21st century kinematics: synthesis, compliance, and tensegrity. ASME J. Mech. Rob. **3**(2), 020201/1–3 (2011)
11. Merlet, J.P.: Parallel Robots. Springer, Dordrecht (2006)
12. Merlet, J.P.: Wire-driven parallel robot: open issues. In: Padois, V., Bidaud, P., Khatib, O. (eds.) Romansy 19—Robot Design, Dynamics and Control, pp. 3–10. Springer, Vienna (2013)
13. Ming, A., Higuchi, T.: Study on multiple degree-of-freedom positioning mechanism using wires—part 1: concept, design and control. Int. J. Tpn. Soc. Precis. Eng. **28**(2), 131–138 (1994)
14. Sommese, A.J., Wampler, C.W.: The Numerical Solution of Systems of Polynomials Arising in Engineering and Science. World Scientific Publishing, Singapore (2005)

Uniform and Efficient Exploration of State Space Using Kinodynamic Sampling-Based Planners

Rakhi Motwani, Mukesh Motwani and Frederick C. Harris Jr

Abstract Sampling based algorithms such as RRTs have laid down the foundation for solving motion planning queries for systems with high number of degrees of freedom and complex constraints. However, lack of balanced state-space exploration of RRTs calls for further improvement of these algorithms. Factors such as drift, underactuation, system dynamics and constraints, and the lack of an energy/time based distance metric in state space can cause RRT propagation to be uneven. This paper focuses on improving the coverage of the RRT algorithm for physical systems that demonstrate a tendency to restrict the growth of RRT to certain regions of the state space. A localized principal component analysis based approach is proposed to learn the propagation bias of state-space points sampled on a grid during an offline learning phase. To compensate for this bias, expansion of the RRT in real-time is steered in the direction of the least principal component of the propagation of the state-space sample selected for expansion. The algorithm is tested on various systems with high degres of freedom and experimental results indicate improved and uniform state-space coverage.

Keywords Uniform state-space coverage · Local-PCA based RRT

R. Motwani (✉) · M. Motwani · F. C. Harris Jr
University of Nevada, Reno, NV, USA
e-mail: rakhi@cse.unr.edu

M. Motwani
e-mail: mukesh@cse.unr.edu

F. C. Harris Jr
e-mail: fredh@cse.unr.edu

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_8,

1 Introduction

Motion planning algoirthms can be categorized as Exact planning methods, Analytical solutions, Numerical approaches, and Approximate methods [3]. Exact methods [2, 13], Analytical solutions [11] and Approximate methods such as grid-based search techniques [4] do not scale well beyond systems with few degrees of freedom while Numerical approaches [1] converge to locally optimal solutions. Sampling-based methods such as RRT [9], Expansive Spaces [6], and the PDST algorithm [8], also fall under the category of approximate methods and are used for high dimensional systems.

This paper presents a technique to improve one of the most widely used kinodynamic sampling-based planners i.e. Rapidly exploring Random Trees (RRT) [9]. RRT is a popular planner for complex systems with geometric and differential contraints due to its capability of quickly exploring high-dimensional state spaces through sampling. The RRT algorithm makes use of an implicit Voronoi bias to evenly explore the state space. However, this Voronoi bias is no longer available if there is no good distance metric in the state space (that determines proximity of points in state-space in terms of time, or energy) or if drift and other dynamic constraints introduce undesired biases. For example, when using a Euclidean distance metric for systems such as Acrobot and Lunar Lander, the RRT growth is restricted to certain regions of the configuration-space, as illustrated by figures in Table 1. Uniform state space exploration of RRT is desired as it reduces the time to find solution trajectories. Typically, an RRT is grown for several thousand nodes for a system under consideration in order to find a solution path. If balanced coverage can be obtained by a relatively smaller sized tree, then we save computational cost and time.

The objective of the work presented here is to make improvements to the standard RRT algorithm to result in a tree of similar or smaller size that spans the state-space evenly. The focus of this paper is to address the issue of the lack of a good distance metric for physical systems that incorporates the effects of system dynamics and contraints when determining which states are closer. The contributions of this paper are as follows. The proposed approach provides a uniform state-space coverage for an RRT by computing the local exploration bias of the dynamic system at each point in the discretized state-space, and using this information to guide the expansion of the tree out of the biased region into least explored areas of the state-space. Experimental results indicate improvement in RRT's state-space exploration for systems that exhibit a bias in coverage towards a specific direction in the state-space. However, it must be noted that this technique is only effective in situations when RRTs fail to evenly explore the state space. If RRT for systems grow uniformly in the state-space the proposed approach does not contribute towards further improvement of the state-space coverage. The contribution is unique in that no related work has employed localized approaches to obtain balanced RRT exploration.

2 Related Work

The research community has published a variery of algorithms [10] that enhance the performance of standard RRT algorithm by proposing modifications to decrease metric sensitivity, reduce the rate of failed expansion, control the sampling domain, guide tree expansion using local reachability information, bias sampling distributions to search in subspace of complete state space or goal region. There is limited literature [5, 12] that addresses the exploration performance of sampling-based algorithms. Li et al.'s [12] work focused on using principal component analysis(PCA) globally to compensate for the undesirable biases introduced by a physical system's dynamic contraints on the exploration of an RRT algorithm. Their approach is composed of two steps: (i) an offline learning procedure which constructs an RRT for the physical system and executes a PCA on the entire tree to represent the principal directions that the tree has expanded inside the task space; (ii) altering the propagation step for RRT during the online operation by modifying the config-space coordinates of the random state sample in each iteration towards directions in which the variance is lower in the offline generated tree, and choosing the control which takes the system closer to this modified version of the random state sample. As a result, growth of the online tree is promoted towards the least explored direction in the task space. This algorthm has only been tested on a Three-link Acrobot and Car-like systems, and has motivated the authors of this paper to implement this technique on variety of systems to compare it's performance with the proposed algorithm.

Glassman and Tedrake's [5] work is based on control theory to derive an approximation to the exact minimum-time distance pseudometric by adapting the minimum time linear quadratic regulator (LQR) and it's associated cost-to-go function for affine systems. The proposed technique linearizes the system dynamics at the randomly sampled state space point in the RRT framework and defines a cost function based on time and effort which is used as the distance measure. The authors use a finite horizon affine quadratic regulator to compute the optimal cost-to-go functions of linearizations of the physical system for multiple time horizons to locally approximate the optimal distance measure. The proposed affine quadratic regulator-based (AQR) distance metric improves exploration of the state space of double integrator and simple pendulum but proves to be ineffective as the systems' nonlinearity and complexity increases such as the cartpole and torque limited 2-link Acrobot. However, the local-PCA based RRT approach presented in this paper focusses on complex non-linear systems.

3 Approach

The proposed technique is comprised of two steps: (i) an offline step to learn the direction of propagation of state-space points sampled on a grid when system dynamics are simulated at these points, and (ii) the alteration of the propagation step of basic

RRT algorithm during the online construction of the tree. The offline step determines the principal components of the direction of propagation of state-space points on a grid of appropriate resolution, when numerous controls are applied to simulate the system dynamics at that point. These principal components represent the different propagation biases in different parts of the state-space. The online phase utilizes this information during the propagation step of building an RRT to reposition the random sample so as to compensate for the biases and even out the RRT's overall exploration of state-space. Henceforth, the proposed algorithm is referred to as LPCA-RRT.

Algorithm 1 Offline Step - Input: N, M, $\triangle t$

```
for $i = 1$ $to$ $N$ do
    $x \leftarrow$ Sample_Grid_State();
    $u[] \leftarrow$ Sample_Random_Controls($M$);
    $s[] \leftarrow$ Simulate_New_States($x$, $u[]$, $\triangle t$);
    $s'[] \leftarrow$ Transform($x$, $s[]$);
    $lpca_x \leftarrow$ PCA($s'_{subset}$);
end for
Return $lpca\_grid$
```

For the offline learning phase, state space points are sampled on a grid of appropriate resolution. Each sampled point represents a region, i.e. bin, in the state-space. For each sampled state space point, the system dynamics are simulated for a specific timestep for a certain amount of controls (i.e. between 50 and 250 depending on the system) to derive a set of new states. The coordinates of the new states are transformed such that the grid point serves as the new origin for these states. Principal Component Analysis (PCA) [7] is executed on a subset of the state-space coordinates of this set of new states. A subset of the state-space dimensions is considered for computational feasibility and also due to the fact that coverage of configuration space is desired as opposed to good coverage in state-space which comprises of derivates of the configuration space parameters. This PCA is referred to as the local PCA and is stored for each state-space point on the grid.

Algorithm 1 summarizes the offline step of the proposed approach, where N represents the total number of states sampled on the grid. N is determined by the grid bounds, grid resolution, and the dimension of state-space. M denotes the number of controls. M varies from system to system and is experimented with values starting from 50 going up to 50,000 at increments of 100 to determine at what value does the local PCA converge. The algorithm returns local PCAs for points sampled on the entire grid.

The pseudo-code for the online phase is provided in Algorithm 2. The basic RRT algorithm is adapted from [9] where at each iteration a random state x_{rand} is sampled from the state space. For construction of the RRT, the node x_{near} on the tree which is nearest to x_{rand} is selected for expansion. A Euclidean distance metric is used to determine the nearest neighbor along the tree. The coordinates of x_{near} are evaluated to calculate the bin from the offline state-space grid that this node belongs to. The offline generated local PCA is then retrieved for the state x_{near}. Since this local PCA is representative of a bin from the offline state-space grid therefore it is only an approximate representation of the direction of propagation of the tree from x_{near}. The configuration space coordinates of the randomly sampled state-space point at the corresponding iteration of RRT are then modified to position the random sample in the direction of the least significant components of this local PCA. The controls that extend the tree from the selected node closer to the altered random sample state are chosen for propagation of the tree thereby leading the RRT out of the regions, where it would have originally been focussed, into unexplored areas of the state space.

Algorithm 2 Online Step - Input: $x_{init}, N, lpca\_grid$

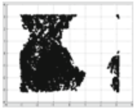
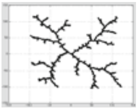
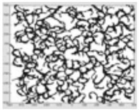
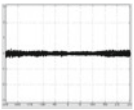
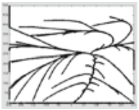
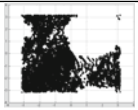
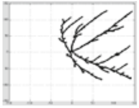
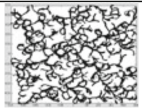
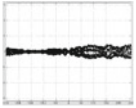
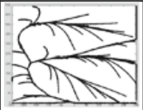
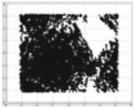
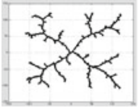
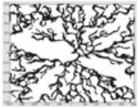
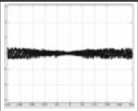
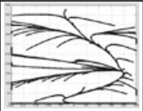
```
$Tree$.init($x_{init}$)
for $i = 1$ $to$ $N$ do
    $x_{rand}$ ← Sample_Random_State();
    $x_{near}$ ← Determine_Nearest_Neighbor($x_{rand}, Tree$);
    $bin_{x_{near}}$ ← Evaluate_Bin($x_{near}, lpca\_grid$);
    $lpca_{x_{near}}$ ← Retrieve_PCA($bin_{x_{near}}, lpca\_grid$);
    $x'_{rand}$ ← Modify($inv(lpca_{x_{near}}), x_{rand}$ );
    $[x_{new}, x_{edge}]$ ← Propagate($x_{near}, x'_{rand}$ );
    $Tree.Vertex\_Add(x_{new})$
    $Tree.Edge\_Add(\ x_{near}, x_{edge})$
end for
Return $Tree$
Modify( $inv(lpca_{x_{near}}), x_{rand}$ )
$x_{adj} \leftarrow inv(lpca_{x_{near}}) * x_{rand};$
for $i = 1$ $to$ $n$ do
    $x'(i)_{adj} \leftarrow \frac{l_1}{l_i} * x(i)_{adj};$
end for
$x'_{rand} \leftarrow lpca_{x_{near}} * x'_{adj};$
Return $x'_{rand}$
```

The algorithm then propagates the selected node x_{near} by applying m random controls to obtain new states. The closest new state to x'_{rand}, denoted by x_{new}, and the corresponding control are selected for the expansion step of the algorithm.

4 Experiments

The algorithm was tested on various systems—Three-link Acrobot, Car-like system, Cart Pole, Hovercraft, and Lunar Lander. The results were compared against the basic RRT, and Li et al.'s algorithm henceforth referred to as GPCA-RRT. Performance of these algorithms was measured in terms of the percentage of bins populated by the generated tree on the discretized subset of state-space and execution time, measured in seconds. Trees were grown for sizes spanning from 1000 to 20,000 nodes

Table 1 Configuration-space coverage plots for trees grown for 20,000 nodes

Plots for →	3-Link Acrobot (θ_1, θ_2) c-space	Hovercraft (x, y) c-space	Car-like System (x, y) c-space	Cart Pole (x, θ) c-space	Lunar Lander (x, y) c-space
RRT					
GPCA					
LPCA					

and the performance results represent an average of ten test runs. The algorithms were implemented in Octave 3.0.5 and executed on the UNR Research Grid. The implementation stores `RRT` in an array and uses linear search for nearest neighbor search, hence recorded processing times are higher. Therefore, the authors would like to emphasize that this is just a proof-of-concept implementation. For `GPCA-RRT`, the experiment used the global PCA of the standard `RRT` of the same size i.e. an `RRT` was grown for N nodes, the global PCA was computed for this `RRT` and was used to generate the `GPCA-RRT` of size N nodes. The figures in Table 1 display a projection of the trees, plotted for various systems, in those configuration space parameters for which the tree exploration was not uniform.

Three-Link Acrobot: The Acrobot was simulated in Passive-Active-Active mode with torque applied at active joints. The angles θ_i are relative to the global reference frame and do not correspond to the angles between consecutive links. As indicated by Table 2, `LPCA-RRT` outperforms both algorithms by 15−20 % in terms of coverage at the expense of spending an average of 2.5 % more in time. Moreover, it was observed

Table 2 Three-link Acrobot Results: $(\theta_1, \theta_2, \theta_3)$ config-space is divided into $50 \times 50 \times 50$ bins to measure coverage

		1000 nodes	3000 nodes	5000 nodes	20000 nodes
RRT	Populated bins	644	1691	2583	8536
	time	28.61	203.41	510.59	28210
GPCA	Populated bins	693	1893	2998	9800
	time	28.70	202.97	510.98	28471
LPCA	Populated bins	722	2015	3170	11285
	time	32.82	209.44	519.30	29329

that the tree generated by LPCA-RRT for 5000 nodes covered the config-space more uniformly than RRT grown for 20,000 nodes.

Car-like System: LPCA-RRT causes the car to move straight with less turns as in the case of RRT or GPCA-RRT. Results indicated that neither GPCA-RRT nor LPCA-RRT provide an improvement in terms of coverage for this system. Results for coverage have not been listed due to space contraints.

Cart Pole: Experiments on Cart Pole system showed that LPCA-RRT resulted in an average improvement of 35 % in coverage with only 1 % increase in time to grow the tree of same size. Space contraints prohibit the authors from sharing the results. From the coverage plots in Table 2, it can be seen that the exploration of θ space (plotted along y-axis) was improved for both LPCA-RRT and GPCA-RRT.

Hovercraft: LPCA-RRT demonstrated a uniform coverage of (x, y) space as compared to RRT. GPCA-RRT tends to skew the growth of RRT towards the upward left direction, which is the principal direction of variance represented by the global PCA computed for the basic RRT algorithm. As per Table 3, coverage of LPCA-RRT improved by an average of 25 % with an average of 1.5 % increase in time cost.

Lunar Lander: The system was simulated in Ascent mode. For RRT and GPCA-RRT, the branches of the tree tend to grow downwards, however LPCA-RRT promotes the growth of the tree sideways. Results from Table 4, show that LPCA-RRT provided an improvement in coverage by 25 % at the cost of 4 % increase in computational time.

Table 3 Hovercraft Results: (x, y, θ) config-space is divided into 50 × 50 × 50 bins to measure coverage in terms of number of populated bins

		1000 nodes	3000 nodes	5000 nodes	20000 nodes
RRT	Populated bins	530	1581	2588	9969
	time	19.68	125.11	316.20	4001
GPCA	Populated bins	477	1113	1772	9234
	time	21.79	127.48	317.59	4261
LPCA	Populated bins	568	1655	2704	9875
	time	53.28	226.26	481.69	4944

Table 4 Lunar lander results: (x, y, θ) config-space is divided into 50 × 50 × 50 bins to measure coverage

		1000 nodes	3000 nodes	5000 nodes	20000 nodes
RRT	Populated bins	361	1089.5	1483	5215
	time	17.249	112.79	313.281	4231
GPCA	Populated bins	323	1104	1773	6501
	time	17.344	113.88	313.289	4247
LPCA	Populated bins	555	1374	1800	6672
	time	23.375	133.54	324.123	4398

5 Conclusion

The proposed technique improves RRT exploration by learning the local effects of constraints in a physical system during an offline phase and then counteracts these effects that inhibit the uniform growth of the RRT during the online operation of RRT by adapting the propagation step. This work executes PCA locally and enables better approximation of the underlying non-linear bias by decomposing the state-space into regions where the bias may be varying. The approach is tested on various systems and experimental results demonstrate that this technique works on systems that exhibit a consistent exploration bias regardless of the size of the tree, and that exploration performance is improved by 15–35 % thereby reducing the cost of finding a solution to specific motion planning queries. One key conclusion is that the algorithm is only effective in scenarios where the standard RRT algorithm fails to uniformly and quickly explore the state-space, as seen in the case of Car-like system. Overall, this technique compensates for the lack of good distance metric in the state-space and can be adopted by various sampling-based planning algorithms.

Acknowledgments The authors would like to thank their colleagues Yanbo Li and Kostas Bekris for providing guidance and assisting with replicting results from [12].

References

1. Betts, J.T.: Survey of numerical methods for trajectory optimization. AIAA J. Guidance, Control Dyn. **21**(2), 193–207 (1998)
2. Canny, J., Rege, A., Reif, J.: An exact algorithm for kinodynamic planning in the plane. Discrete Comput. Geom. **6**, 461–484 (1991)
3. Choset, H.M.: Principles of robot motion: theory, algorithms, and implementation Intelligent robotics and autonomous agents. MIT Press, Cambridge (2005)
4. Donald, B., Xavier, P., Canny, J., Reif, J.: Kinodynamic motion planning. J. ACM **40**(5), 1048–1066 (1993)
5. Glassman, E.L.: A quadratic regulator-based heuristic for rapidly exploring state space. Master's thesis, Massachusetts Institute of Technology, Massachusetts (2010)
6. Hsu, D., Kindel, R., Latombe, J.C.: Rock . Randomized kinodynamic motion planning with moving obstacles (2000)
7. Jolliffe, I.: Principal component analysis, 2nd edn. Springer, New York (2010)
8. Ladd, A.M., Kavraki, L.E.: Motion planning in the presence of drift, underactuation and discrete system changes. In: Robotics: science and systems I, pp. 233–241. MIT Press, Boston (2005)
9. LaValle, S., Kuffner, J.: Rapidly exploring random trees: progress and prospects. In: Proceedings Workshop on Algorithmic Foundations of Robotics WAFR, pp. 293–308 (2001)
10. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
11. Lewis, F.L.: Applied optimal control and estimation: digital design and implementation. Prentice Hall, New Jersey (1992)
12. Li, Y., Bekris, K.E.: Balancing state-space coverage in planning with dynamics. In: IEEE International Conference on Robotics and Automatio (ICRA), pp. 3246–3253 (2010)
13. Schwartz, J.T., Sharir, M.: On the piano movers' problem: Ii. general techniqies for computing topological properties of algebraic manifolds. Commun. Pure Appl. Math. **36**, 345–398 (1983)

Shape Modeling of Continuous-Curvature Continuum Robots

Shaoping Bai

Abstract An essential problem in developing concentric-tube continuum robots is to determine the shape of the robot, which is dependent on robot structure and external load. A comprehensive model that takes into considerations of influencing factors is hence required. In this work, the shape modeling of a type of concentric-tube continuum robot built with a collection of super-elastic NiTiNol tubes is studied. The model, developed on the basis of differential geometry and curved beam theory, is able to determine both the bending deflection and torsional deformation for a continuum robot of continuous curvature. Simulation results for calculating the shape of a continuum robot built with NiTiNol tubes are included.

Keywords Flexible manipulators · Continuous-curvature continuum robots · Shape modeling · NiTiNol tubes

1 Introduction

Continuum robots encompass new principles of robot inspired by the nature. A continuum robot is a robot that is able to deform continuously, similar to their counterparts in nature such as snakes, elephant trunks or octopus arms. The continuum robots are able to move in any direction, both laterally and axially, or even 'turning corner'. Contrary to traditional robots built with rigid links, a continuum robot is constructed with a collection of flexible structures which allow them deform locally to generate desired motion.

The continuum robots can be built with different principles, as seen in some prototypes including multi-sectional pneumatic actuating robot Air-Octor [1], fluid-driven Octpus Arm [2], tendon-driven robots [3], among others. Of these robots,

S. Bai (✉)
Department of Mechanical and Manufacturing Engineering and Centre for Robotics Research,
Aalborg University, Aalborg, Denmark
e-mail: shb@m-tech.aau.dk

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_9,

robots with concentric tubes of NiTiNol (Nickel-Titanium) alloy are more promising for applications where a super mobility in a confined space is critical, such as the robotic minimally invasive surgery, due to the relative simple structure, compact size (diameters can be as small as a few millimeters), and bio compatibility to human tissues. A number of modeling works on continuum robots were reported [4–6]. The models were mainly developed on the basis of Cosserat rod with either energy approach or variational methods [7, 8]. An approach of including torsion was proposed by Dupont et al. in [9]. In most works, shape models were derived by assuming piecewise-constant curvatures.

In this work, the problem of shape modeling for continuum robots built with concentric tubes is addressed, focusing on the geometrically exact model of the robot shape. In this modeling work, both the bending deflection and torsional deformation are considered. As the shape model is derived on the basis of continuous curvatures, a more comprehensive and accurate model for the continuum robot can be obtained, compared with the piecewise constant curvature approach. Simulations were carried out to demonstrate the application of the model.

The paper is organized as follows. Section 2 describes the geometry of a spatial curve and a single tube, upon which the equilibrium condition of a two-tube assembly is established. The shape modeling of continuum robots is presented in Sect. 3. Simulation results are reported in Sect. 4. The work is concluded in Sect. 5.

2 Model of a Single Spatial Tube

A continuum robot can be built with multiple sessions of concentric tubes, as demonstrated in Fig. 1, which can be considered as a combination of several sections serially connected. In each session, two tubes of different diameters are assembled with variable configuration and thus build the desired shape. The shape modeling of continuum robots can thus be built on the shape calculation of a single tube.

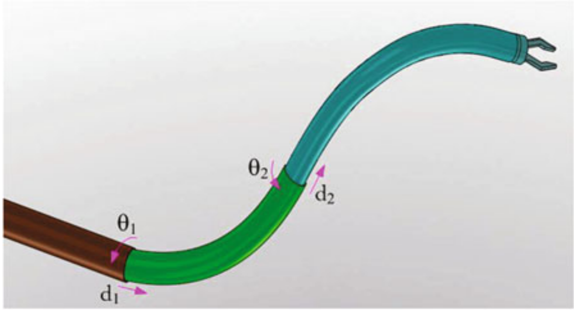


Fig. 1 A continuum robot built with multiple sessions of concentric tubes

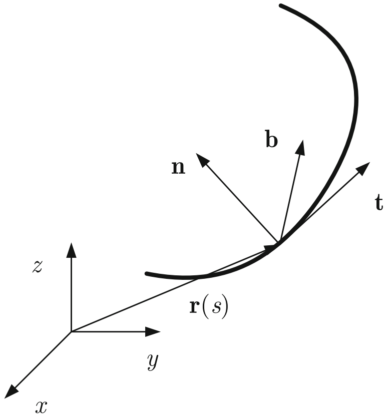


Fig. 2 A spatial curve and its associated Frenet-Serret frame

2.1 Geometry of a Spatial Curve

A space curve $\mathcal{C}$ can be expressed as a function of arc length s, i.e., $\mathcal{C} : \mathbf{r} = \mathbf{r}(s),\ s \in \mathbb{R}$. At any point s, the derivative of the function $\mathbf{r}(s)$ with respect to s is equal to the tangential vector of the curve, i.e. $\mathbf{t} = d\mathbf{r}/ds$, as shown in Fig. 2. Together with the normal vector $\mathbf{n}$ at s, and a third vector $\mathbf{b} = \mathbf{t} \times \mathbf{n}$, the *Frenet-Serret* frame, or *F-S* frame in short, is fully established. The shape function of a space curve can be described by the Frenet-Serret equation [10] in matrix form

$$\mathbf{T}' = \mathbf{T}\widetilde{\boldsymbol{\kappa}}; \quad \widetilde{\boldsymbol{\kappa}} = \begin{bmatrix} 0 & -\tau & \kappa \\ \tau & 0 & 0 \\ -\kappa & 0 & 0 \end{bmatrix} \tag{1}$$

where $\mathbf{T} = [\mathbf{n}, \mathbf{b}, \mathbf{t}]$ and $\boldsymbol{\kappa} = [0, \kappa, \tau]^T$ is a vector of curvatures. Here, and in the balance of the paper, the prime symbol denotes the derivative with respect to arc length s, and $\widetilde{\boldsymbol{\kappa}}$ is the cross-product matrix (a skew-symmetric matrix) of the vector $\boldsymbol{\kappa}$.

The F-S frame is unique, which satisfies the Frenet-Serret equation and describes entirely the curve shape. For a continuum tube, however, the material twist at a cross-section has to be described in order to determine the tube's torsional deformation. In light of this, a frame along the center line has to describe the twist of the tube, in addition to the shape of the curve. In another words, a frame attached to the tube but different from the F-S frame is required. In this case, the vector of curvatures will be transformed due the change of the reference (view) frame, as shown in the following.

Let the F-S frame be noted as $\mathcal{T}$ corresponding to an orientation matrix $\mathbf{T}$. Assume there is another reference frame $\mathcal{F}$ of orientation matrix $\mathbf{F}$. The transformation from $\mathcal{T}$ to $\mathcal{F}$ is described by $\mathbf{F} = \mathbf{RT}$, where $\mathbf{R}$ is a rotation matrix. Let the vector of curvatures be $\boldsymbol{\lambda}$ when the curve is viewed from the reference frame $\mathcal{F}$. The following relationship between $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$ can be found

$$\mathbf{R}^T \boldsymbol{\kappa} + \boldsymbol{\omega} = \boldsymbol{\lambda} \tag{2}$$

where $\boldsymbol{\omega}$ is the relative 'velocity' of $\mathcal{F}$ frame with respect to the $\mathcal{T}$ frame. The alternative reference frame in this case is called an adapted frame.

A simple case is the adapted frame is obtained by rotating the F-S frame about $\mathbf{t}$ for an angle of α. Letting the new frame is noted by three orthogonal unit vectors $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ and $\mathbf{z}$ is parallel to $\mathbf{t}$, Eq. (2) becomes

$$\mathbf{R}_z(\alpha)^T \boldsymbol{\kappa} + \alpha' \mathbf{e}_z = \boldsymbol{\lambda} \tag{3}$$

where $\mathbf{R}_z(\alpha)$ is the matrix of rotation about z-axis. It can be known from Eq. (3) that a curve of curvature of κ and torsional rate τ, when viewed in an adapted frame, has a vector of curvature $\boldsymbol{\kappa} = [\kappa \sin \alpha, \kappa \cos \alpha, \tau + \alpha']^T$.

2.2 Equilibrium Equations

When a tube is subject to forces, an equilibrium condition is established after deformation. Refer to Fig. 3 where a small piece of a spatial rod is shown, the equation of moment equilibrium is derived as

$$d\mathbf{m} + \mathbf{e}_z ds \times \mathbf{f} = 0 \tag{4}$$

where $\mathbf{e}_z = [0, 0, 1]^T$. Noting that $\mathbf{m}$ in this equation is expressed in a moving adapted frame, care has to be given to both the variation of the vector and the frame as well. That means

$$d\mathbf{m} = \mathbf{m}' ds + \boldsymbol{\kappa} \times \mathbf{m} ds \tag{5}$$

Finally, we have

$$\mathbf{m}' + \boldsymbol{\kappa} \times \mathbf{m} + \mathbf{e}_z \times \mathbf{f} = \mathbf{0} \tag{6}$$

The second term in the left-hand side of Eq. (6) accounts for the effect of the tube's torsion. Likewise, the force equilibrium equation is found as

$$\mathbf{f}' + \boldsymbol{\kappa} \times \mathbf{f} = \mathbf{0} \tag{7}$$

Assume the rod deflection is subject to the linear elasticity theory and the axial extension can be ignored, we could concern about only the moment equilibrium equation. In this regard, the constitutive equation for the space rod subject to bending and torsional moments is

$$\mathbf{m}(s) = \mathbf{K}(\boldsymbol{\kappa}(s) - \bar{\boldsymbol{\kappa}}), \quad \text{with } \mathbf{K} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} = \begin{bmatrix} EI & 0 & 0 \\ 0 & EI & 0 \\ 0 & 0 & GJ \end{bmatrix} \tag{8}$$

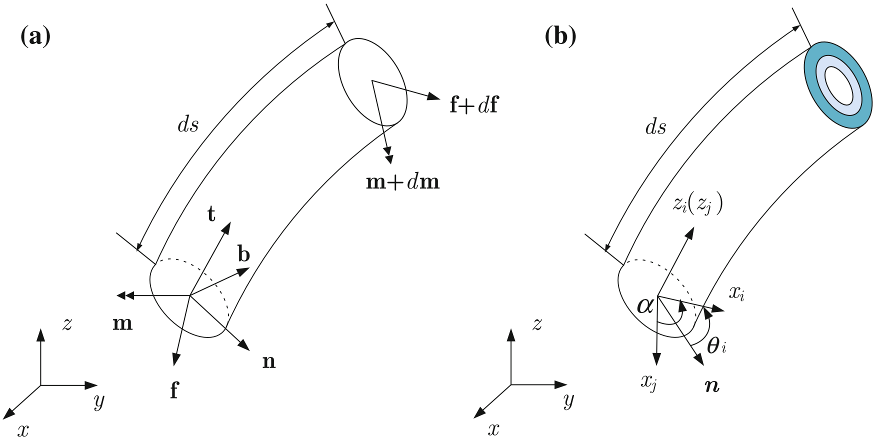


Fig. 3 Equilibrium condition, **a** a small piece of single spatial curved rod, **b** an assembly of two concentric tubes

where $\bar{\boldsymbol{\kappa}} = [\bar{\kappa}_x, \bar{\kappa}_y, \bar{\tau}]^T$ is the vector of initial curvature of the tube. E and G are the material's elastic (Young) and shear modulus, while I and J are the area and polar moment of inertia at a cross section, respectively.

3 Shape Modeling of Assembled Tubes

When two tubes are assembled, they build a constrained mechanical system. To describe the deformation of each tube, local frames are established on each tube, which are able to describe the shape of their centerline and also the tube twist. The two local (adapted) frames are only different in twist angle, as demonstrated in Fig. 3b.

For each segment of the tube assembly, the internal moments are balanced, i.e.,

$$\mathbf{m}_1(s) + \mathbf{R}_z(\alpha)\mathbf{m}_2(s) = 0 \tag{9}$$

with

$$\mathbf{m}_i = \mathbf{K}_i(\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i), \quad i = 1, 2 \tag{10}$$

In this case, $\boldsymbol{\kappa}_i = [\kappa_{ix}, \kappa_{iy}, \tau_i]^T, \bar{\boldsymbol{\kappa}}_i = [\bar{\kappa}_{ix}, \bar{\kappa}_{iy}, \bar{\tau}_i]^T$ for $i = 1, 2$. Moreover, the mechanical constraint implies that two tubes' center lines superpose on each other, with only difference in torsional twist. In other words, Eq. (3) has to be satisfied:

$$\mathbf{R}_z(\alpha)^T\boldsymbol{\kappa}_1(s) + \alpha'\mathbf{e}_z = \boldsymbol{\kappa}_2(s) \tag{11}$$

where α is the relative twist angle, as demonstrated in Fig. 3b, which is defined as

$$\alpha(s) = \theta_2(s) - \theta_1(s) \tag{12}$$

where $\theta_i, i = 1, 2$ is the twist angle of the i-th adapted frame measured with respect to the F-S frame. Both Eqs. (9) and (11) are derived in a local frame. We rewrite Eq. (11) as

$$\boldsymbol{\kappa}_1(s) = \mathbf{R}_z(\alpha)(\boldsymbol{\kappa}_2(s) - \alpha'\mathbf{e}_z) \tag{13}$$

Substituting Eqs. (10) and (13) into (9) yields

$$\mathbf{K}_1(\mathbf{R}_z(\alpha)\boldsymbol{\kappa}_2 - \alpha'\mathbf{R}_z(\alpha)\mathbf{e}_z - \bar{\boldsymbol{\kappa}}_1) + \mathbf{R}_z(\alpha)\mathbf{K}_2(\boldsymbol{\kappa}_2 - \bar{\boldsymbol{\kappa}}_2) = 0 \tag{14}$$

Noting that a special identity $\mathbf{K}_1\mathbf{R}_z(\alpha) = \mathbf{R}_z(\alpha)\mathbf{K}_1$ exists for the tubes, due to the fact that the diagonal stiffness matrix $\mathbf{K}_1$ contains identical bending stiffness for $x-$ and $y-$ directions, we finally obtain

$$\boldsymbol{\kappa}_2 = (\mathbf{K}_1 + \mathbf{K}_2)^{-1}(\mathbf{R}_z^T(\alpha)\mathbf{K}_1\bar{\boldsymbol{\kappa}}_1 + \mathbf{K}_2\bar{\boldsymbol{\kappa}}_2 + \alpha'\mathbf{K}_1\mathbf{e}_z) \tag{15}$$

Obviously, the shape of the tube assembly is fully determined if α and α' can be solved. To this end, we first expand Eq. (15), which yields

$$\kappa_{2x} = \frac{E_1 I_1 \bar{\kappa}_{1y}\sin\alpha + E_1 I_1 \bar{\kappa}_{1x}\cos\alpha + E_2 I_2 \bar{\kappa}_{2x}}{E_1 I_1 + E_2 I_2} \tag{16}$$

$$\kappa_{2y} = \frac{-E_1 I_1 \bar{\kappa}_{1x}\sin\alpha + E_1 I_1 \bar{\kappa}_{1y}\cos\alpha + E_2 I_2 \bar{\kappa}_{2y}}{E_1 I_1 + E_2 I_2} \tag{17}$$

Recall that the third equation of Eq. (6) is about the twist of tubes, which can be rewritten as

$$G_i J_i \theta_i'' + \kappa_{ix} E_i I_i(\kappa_{iy} - \bar{\kappa}_{iy}) - \kappa_{iy} E_i I_i(\kappa_{ix} - \bar{\kappa}_{ix}) = 0, \quad i = 1, 2 \tag{18}$$

that is

$$\theta_i'' = \frac{E_i I_i}{G_i J_i}(\kappa_{ix}\bar{\kappa}_{iy} - \kappa_{iy}\bar{\kappa}_{ix}) \tag{19}$$

Substituting Eqs. (16) and (17) into (19) and finally into equation

$$\alpha''(s) = \theta_2''(s) - \theta_1''(s) \tag{20}$$

yields

$$\alpha''(s) = A\cos\alpha(s) + B\sin\alpha(s) \tag{21}$$

where

Table 1 Physical and geometric parameters of tubes

	Parameter	Value	Description
Physical	E	$5 \times 10^{10} Pa$	Young modulus
	G	$2.3 \times 10^{10} Pa$	Shear modulus
Geometric	$\bar{\kappa}_1, \bar{\kappa}_2$	1/236, 1/294 [1/mm]	Pre-curvatures of outer and inner tubes
	l_1, l_2	200, 200 [mm]	Tube length
	$d_1(d_2), t_1(t_2)$	1.6(1.2), 0.2(0.2) [mm]	Outer (inner) tube's diameter and thickness

$$A = \frac{E_1 I_1 E_2 I_2 (G_1 J_1 + G_2 J_2)}{G_1 J_1 G_2 J_2 \left(E_1 I_1 + E_2 I_2\right)} (\bar{\kappa}_{1x}\bar{\kappa}_{2y} - \bar{\kappa}_{1y}\bar{\kappa}_{2x}) \tag{22}$$

$$B = \frac{E_1 I_1 E_2 I_2 (G_1 J_1 + G_2 J_2)}{G_1 J_1 G_2 J_2 \left(E_1 I_1 + E_2 I_2\right)} (\bar{\kappa}_{1x}\bar{\kappa}_{2x} + \bar{\kappa}_{1y}\bar{\kappa}_{2y}) \tag{23}$$

Differential Eq. (21) can be solved with boundary conditions (BCs). For fixed-free ends, the BCs are

$$\alpha(0) = \theta_2(0) - \theta_1(0), \alpha'(l) = 0 \tag{24}$$

When a load is applied at the free end, the boundary conditions will change. The loaded cases require further detailed formulation and will be discussed in a separate paper.

The differential Eq. (21) is now ready to be solved with a numerical solver. With the curvature and the changing rate of twist angle found, the tube-assembly's shape can be uniquely determined for any given initial configuration at the end $s = 0$.

4 Simulations

The continuum robot in this work is built with concentric tubes of NiTiNol (Nickel-Titanium) alloys (NiTi SE 508). The outer tube has a diameter of 1.6 mm and thickness of 0.2 mm, while the inner tube is 1.32 mm in diameter and 0.225 mm in thickness, as listed in Table 1. The area and polar moments of inertia of the tubes can be calculated with the tube geometric parameters.

The simulation result on torsional deformation was obtained, as shown in Fig. 4a. The input angle refers to the relative twist angle $\alpha(s)$ at $s = 0$. If one tube is fixed and the other is connected a motor, the input angle is equal to the motor's rotation. The measurement of torsional deformation reported in [11] is shown too in Fig. 4a for comparison. It is seen that the two results are very close to each other.

The simulation results on the tube assembly's shape were also obtained. One result is displayed in Fig. 4b, where the pose of the tube-tip is shown together with a frame at the tip to indicate the orientation. While the simulation was carried out for single session of two-tube assembly, it could be extended to complicated configurations where more sessions of tube assemblies connected serially to build a curve of desired shape.

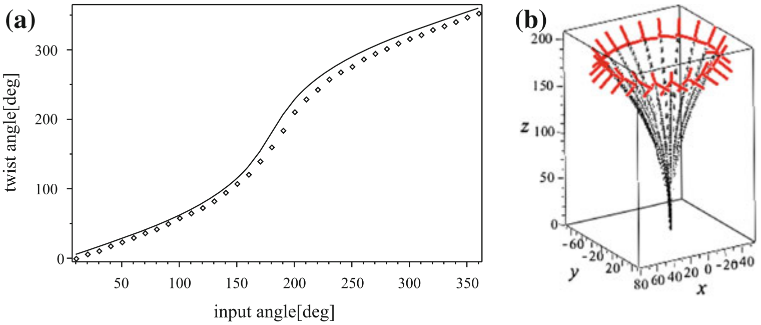


Fig. 4 Simulation results: **a** twist angle simulations (*solid line*) compared with measurements (*dots*), **b** tube centerline changing with outer tube rotation. *Red lines* show the tube orientation

5 Conclusions

In this work, the shape modeling of a type of concentric-tube continuum robot was developed for an assembly of super-elastic NiTiNol tubes. A new model was developed, in which both bending deformation and torsion are considered. The model allows the calculation of the shape a tube assembly as a function of the parameters of tubes and their relative angle. The model, developed for tubes of variable curvatures, can be used in the continuum kinematic design and analysis.

References

1. Blessing, M., Walker, I.D.: Novel continuum robots with variable-length sections. In: Proceedings of the 3rd IFAC Symposium on Mechatronic Systems, Sydney, Australia (2004)
2. Mazzolai, B., Margheri, L., Cianchetti, M., Dario, P., Laschi, C.: Soft-robotic arm inspired by the octopus: II. from artificial requirements to innovative technological solutions. Bioinspiration & Biomimetics **7**(2), 025005 (2012)
3. Li, C.: Design of Continuous Backbone, Cable-driven Robots. Clemson University, Clemson (2000)
4. Dupont, P.E., Lock, J., Butler, E.: Torsional kinematic model for concentric tube robots. In: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp. 2964–2971. (2009)
5. Webster III, R.J., Jones, B.A.: Design and kinematic modeling of constant curvature continuum robots: a review. Int. J. Robot. Res. **29**(13), 1661–1683 (2010)
6. Jones, B.A., Gray, R.L., Turlapati, K.: Three dimensional statics for continuum robotics. In: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2659–2664. (2009)
7. Camarillo, D.B., Milne, C.F., Carlson, C.R., Zinn, M.R., Salisbury, J.K.: Mechanics modeling of tendon-driven continuum manipulators. IEEE Trans. Robot. **24**(6), 1262–1273 (2008)
8. Lang, H., Linn, J., Arnold, M.: Multibody dynamics simulation of geometrically exact Cosserat rods. Multibody Dyn. **25**, 285–312 (2011)
9. Dupont, P.E., Lock, J., Itkowitz, B., Butler, E.: Design and control of concentric-tube robots. IEEE Trans. Robot. **26**(2), 209–225 (2010)

10. Do Carmo, M.P.: Differential geometry of curves and surfaces. Prentice-Hall Inc., Englewood Cliffs (1976)
11. Bai, S., Chuhao Xing, C.: Shape modeling of a concentric-tube continuum robot. In: Proceedings of the 2012 IEEE Inter Conference on Robotics and Biomimetics (ROBIO 2012), pp. 116–121. Guangzhou, China, 11–14 Dec 2012

Dimensional Synthesis of a Spatial Orientation 3-DoF Parallel Manipulator by Characterizing the Configuration Space

M. Urízar, V. Petuya, M. Diez and A. Hernández

Abstract In this paper the authors approach the dimensional synthesis of parallel manipulators focusing on the evaluation of important entities belonging to the configuration space, such as workspace and joint space. In particular, 3-DoF manipulators that can perform non-singular transitions are considered, illustrating the procedure with a case study. The target is to search for designs that achieve the goals of adequate size and shape of the workspace.

Keywords Parallel mechanisms · Configuration space · Dimensional synthesis

1 Introduction

In the design process of parallel manipulators several criteria have been presented in the literature for evaluating which architecture is best. Designers often search for the optimum design parameters such that certain important requirements are achieved. From the kinematic point of view, optimal design methodologies are principally focused on: workspace [2, 6, 8], kinematic performance indices [1, 3], task development [7], etc. Features such as workspace and dexterity can be emphasized as two significant considerations [1, 5], because parallel manipulators have relative smaller workspaces and complex singularities compared to their serial counterparts.

The analysis of the singularity loci, together with the distribution of the Direct Kinematic Problem (DKP) solutions over the workspace, has received a lot of attention during the last years. In this field, the phenomenon of *assembly mode change*, also known as *non-singular solution change*, has been extensively studied [4, 10, 11]. It consists in analyzing how the transitions between different DKP solutions can be made in a safety and controlled way. Manipulators presenting this ability can

M. Urízar (✉) · V. Petuya · M. Diez · A. Hernández
University of the Basque Country, Faculty of Engineering in Bilbao, Mechanical Engineering Department, Bilbao, Spain
e-mail: monica.urizar@ehu.es

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_10,

enlarge their range of motion, as they have access to all the regions associated with the solutions involved in the transition. However, it must be emphasized that, usually, not all the designs of the same manipulator own this ability.

In this paper, the dimensional synthesis of this type of manipulators is approached by characterizing entities of the configuration space, such as workspace and joint space. For three-degree-of-freedom parallel manipulators these entities that can be represented in a three-dimensional space. So as to show the procedure the spatial orientation 3-S$\underline{\text{P}}$S-S parallel manipulator is used as an illustrative example. This manipulator has a broad range of applications, such as: orienting a tool or a workpiece, solar panels, space antennas, camera devices, human wrist prosthesis, haptic devices, etc. The purpose is to analyze several designs, assessing the influence of the design parameters on the resultant workspace and joint space. Then, the aim is to search for the set of possible designs that best satisfy the requirements of size and shape of the operational workspace.

2 Case Study: 3-S$\underline{\text{P}}$S-S Parallel Manipulator

The spatial orientation 3-S$\underline{\text{P}}$S-S parallel manipulator shown in Fig. 1a will be studied. The 3-S$\underline{\text{P}}$S-S manipulator is made up of a moving platform $OB_1B_2B_3$, a base platform $OA_1A_2A_3$, and three extensible limbs denoted by l_i. Both platforms take the form of a tetrahedron, connected one to each other by a fixed spherical joint at point O. The robot has 3-DoF (φ, θ, ψ) defining the orientation of the moving platform.

With respect to the base platform, the fixed spherical joints A_i are located on the principal axes of the fixed frame $\mathcal{F}\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ fulfilling $|\overrightarrow{OA_i}| = R$, for $i = 1, 2, 3$. Besides, the spherical joints B_i are located with respect to the moving frame

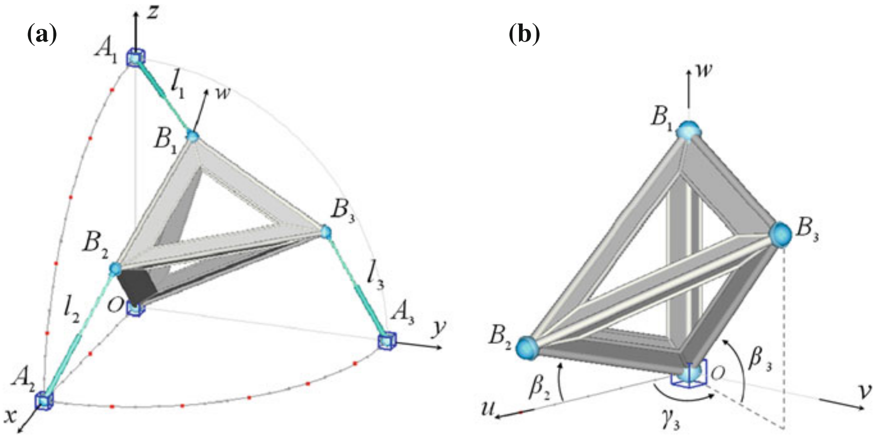


Fig. 1 **a** Spatial 3-S$\underline{\text{P}}$S-S parallel manipulator; **b** Design parameters of the moving platform

$\mathcal{M}\,\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ (see Fig. 1b) such that:

$$\begin{aligned} {}^{\mathcal{M}}\mathbf{b}_1 &= L\,[0, 0, 1]^T \\ {}^{\mathcal{M}}\mathbf{b}_2 &= L\,[b_{2u}, 0, b_{2w}]^T \\ {}^{\mathcal{M}}\mathbf{b}_3 &= L\,[b_{3u}, b_{3v}, b_{3w}]^T \end{aligned} \tag{1}$$

where

$$\begin{aligned} &b_{2u} = c\beta_2; \qquad b_{2v} = 0; \qquad b_{2w} = s\beta_2 \\ &b_{3u} = c\beta_3 c\gamma_3; \qquad b_{3v} = c\beta_3 s\gamma_3; \qquad b_{3w} = s\beta_3 \end{aligned} \tag{2}$$

The transformation from the moving frame $\mathcal{M}$ to the fixed frame $\mathcal{F}$ can be achieved by a 3 × 3 rotation matrix ${}^{\mathcal{F}}_{\mathcal{M}}\mathbf{R}$ defined by the three Euler angles. In this case, the Euler angles (φ, θ, ψ) in their **wvw** version will be used.

The vector ${}^{\mathcal{F}}\mathbf{b}_i$, or simply $\mathbf{b}_i$, expressed with respect to the fixed frame $\mathcal{F}$ is:

$$\mathbf{b}_i = \left[b_{ix}, b_{iy}, b_{iz}\right]^T =^{\mathcal{F}}_{\mathcal{M}} \mathbf{R}^{\mathcal{M}}\mathbf{b}_i \tag{3}$$

On the other hand, the position vector of points A_i with respect to the fixed frame $\mathcal{F}$ is $\mathbf{a}_i = \left[a_{ix}, a_{iy}, a_{iz}\right]^T$.

The loop-closure equation for each limb is $\mathbf{l}_i = \mathbf{b}_i - \mathbf{a}_i$, which results in the following system for $i = 1, 2, 3$:

$$l_i^2 = \mathbf{a}_i^2 + \mathbf{b}_i^2 - 2\mathbf{a}_i^T \mathbf{b}_i \tag{4}$$

- *Inverse Kinematic Problem*: To solve the IKP, the Euler angles are established (φ, θ, ψ) and the length of each limb l_i can be directly obtained from Eq. 4. Only the positive solution yields a physical meaning.
- *Direct Kinematic Problem*: The DKP consists in solving the outputs (φ, θ, ψ) once the three prismatic limb lengths are known. As demonstrated in [9] this manipulator has a maximum of eight solutions to the DKP.

2.1 Velocity Problem

So as to solve the velocity problem the loop-closure equations are differentiated with respect to time, obtaining:

$$\boldsymbol{\omega}_p \times \mathbf{b}_i = \boldsymbol{\omega}_i \times \mathbf{l}_i + \dot{l}_i \cdot \mathbf{s}_i \tag{5}$$

where $\mathbf{s}_i$ is defined as the unit vector directed from A_i to B_i. The moving platform angular velocity is $\boldsymbol{\omega}_p$, and $\boldsymbol{\omega}_i$ corresponds to the angular velocity of each limb l_i.

Dot-premultiplying each term of the system (5) by $\mathbf{l}_i$, the velocity equation expressed in a matrix form is obtained as:

$$\mathbf{J}_{DKP}\left\{\omega_p\right\} = \mathbf{J}_{IKP}\left\{\dot{l_i}\right\} \tag{6}$$

where

$$\mathbf{J}_{DKP} = \begin{bmatrix} (\mathbf{b}_1 \times \mathbf{l}_1)^T \\ (\mathbf{b}_2 \times \mathbf{l}_2)^T \\ (\mathbf{b}_3 \times \mathbf{l}_3)^T \end{bmatrix}; \quad \mathbf{J}_{IKP} = \begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix} \tag{7}$$

The inverse Jacobian matrix, $\mathbf{J}_{IKP}$, is singular only whenever any of the prismatic limbs has zero length, which cannot be achieved in practice. Besides, each limb has only one associated working mode. Hence, we focus on the analysis of the DKP singularity locus in the configuration space.

2.2 DKP Singularity Locus

The DKP singularity locus is obtained by computing the nullity of the determinant of $\mathbf{J}_{DKP}$, which yields:

$$|\mathbf{J}_{DKP}| = -R^3L^3 \cdot s\theta \cdot \xi \tag{8}$$

where

$$\begin{aligned} \xi = {} & c^2\varphi c\theta(b_{2w}(b_{3u}s\psi + b_{3v}c\psi) - b_{3w}(b_{2u}s\psi)) - c^2\varphi s\theta(b_{3v}b_{2u}) \\ & + c^2\psi s\theta(b_{2u}b_{3v}) + s\theta(c\psi s\psi(b_{2u}b_{3u})) - c\theta(b_{2w}(b_{3u}s\psi + b_{3v}c\psi)) \\ & + s\varphi c\varphi(b_{2w}(b_{3u}c\psi - b_{3v}s\psi) - b_{3w}b_{2u}c\psi) \end{aligned} \tag{9}$$

Expression $|\mathbf{J}_{DKP}|$ factorizes into three terms:

- The constant R^3L^3 does not affect the shape of the DKP singularity locus. Parameters R and L define the size of the robot, and the minimum and maximum stroke of the prismatic limbs. For the example under study, without loss of generality, values $R = 1$ and $L = 0.5$ will be assigned.
- The second term corresponds to the function: $s\theta$. So as to avoid the singularity planes $\theta = 0$ and $\theta = \pm\pi$, the interval $\theta \in (0, \pi)$ will be considered.
- Finally, from Eq. 9 yields the expression ξ. This function depends on the output variables (φ, θ, ψ), and on the geometric parameters $(\beta_2, \gamma_3, \beta_3)$. Therefore, the expression ξ will be assessed regarding the dimensional synthesis.

3 Dimensional Synthesis

Parameters $(\beta_2, \gamma_3, \beta_3)$ comprise the design parameters subject of study. Different designs will be analyzed, representing and assessing workspace and joint space entities.

- **Case 1**: Similar Platforms

The first case under study establishes a design of the moving platform such that it is similar to the fixed base. For that, the geometric parameters are: $\beta_2 = \beta_3 = 0$ and $\gamma_3 = 90°$. The expression of the DKP singularity locus, given by Eq. 9, yields:

$$\xi = s\theta(c\psi - c\varphi)(c\psi + c\varphi) \tag{10}$$

It is factorized into the function $s\theta$, and the product of two planes. These planes divide the workspace (φ, θ, ψ) into eight aspects, so $V_{case1} = V_T/8$ being V_T the total volume. Each DKP solution lies inside each aspect, non-singular transitions being not possible. This is corroborated with the non-existence of cusp points inside any section of the joint space (see details in [9], Chap. 10).

- **Case 2**: Joints B_2, B_3 on **uv**-plane

The second case under study locates joints B_2 and B_3 on the **uv**-plane, such that $\beta_2 = \beta_3 = 0$ and γ_3 varies in the interval $(0, 90°)$. The DKP singularity locus yields:

$$\xi = s\theta[b_{3u}s\psi c\psi + b_{3v}(c^2\psi - c^2\varphi)] \tag{11}$$

Yet again, expression ξ factorizes into the function $s\theta$, and a trigonometric expression depending on outputs (φ,ψ) and coordinates b_{3u} and b_{3v}, function of the geometric parameter γ_3. Let us analyze a design included in Case 2, by assigning $\gamma_3 = 30°$. The DKP singularity locus is represented in the workspace in Fig. 2a, the joint space and its cross section for $l_1 = const$ being depicted in Fig. 2b. Contrary to Case 1, only four aspects exist, so that the operational workspace is duplicated $V_{case2} = 2V_{case1} = V_T/4$, because the robot can move between solutions located inside the same aspect. This is in accordance with the existence of cusp points in joint space sections, as shown in Fig. 2b.

Non-singular transitions can be performed between regions in the workspace where different solutions lie, as for example regions 1 and 2 in the workspace section of Fig. 2c. Though the size of the workspace is $V_T/4$ for all designs in Case 2, its shape varies. It is interesting to search for designs that yield a regular workspace, such that the range of motion of the output variables maintains over the entire workspace. As shown in Fig. 2c, the ratio $H = h/r$ can be measured and serves as an indicator of regularity. Its evolution depending on γ_3 is represented in Fig. 2d. It can be observed that small values of γ_3 yield a more regular workspace ($H \approx 1$). The extreme values $\gamma_3 = 0$ (planar moving platform) and $\gamma_3 = 90°$ constitute particular designs. On the one hand, $\gamma_3 = 0$ yields a degeneracy design for which the workspace is formed by

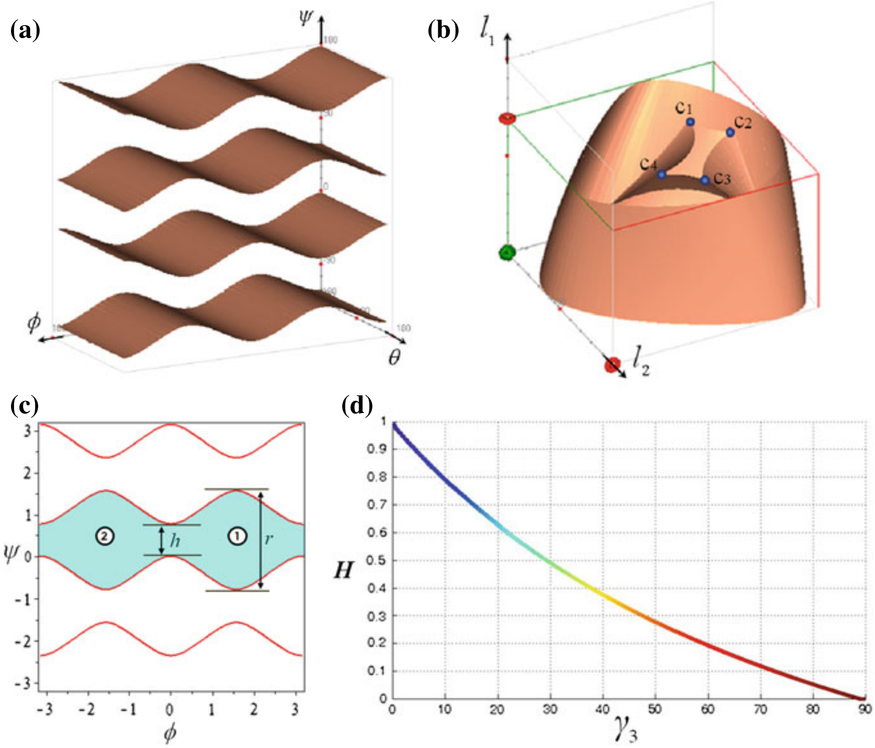


Fig. 2 Case 2: DKP singularity locus in the **a** workspace, **b** joint space and **c** workspace section; **d** Ratio of regularity H

planes ($H = 1$) and only 4 DKP solutions exist. Value $\gamma_3 = 90°$ coincides with Case 1, and verifies $H = 0$, no connection between different regions is possible.

- **Case 3**: General Design

The last case corresponds to a general design of the moving platform. For this case the three dimensional parameters $(\beta_2, \gamma_3, \beta_3)$ can be assigned any value in the range $(0, 90°)$. The singularity locus is given by expression ξ in Eq. (9), which is plotted in the workspace and joint space in Fig. 3 for a specific design: $\beta_2 = 30°$, $\gamma_3 = 60°$ and $\beta_3 = 30°$. Some sections of the joint space are also depicted in Fig. 3b, visualizing the existence of cusp points.

These designs present two aspects, the holes of the singularity surface in the workspace (Fig. 3a) allowing the connection between all solutions having the same sign of $|\mathbf{J}_{DKP}|$. Consequently, the designs of Case 3 exhibit the maximum operational workspace: $V_{case3} = V_T/2$. Nevertheless, the shape that the singularity surface acquires in the workspace, and in the joint space, is much more complex.

In this sense, similarly to Case 2, some indicators that characterize the shape of the operational workspace can be implemented. Then, parameters $(\beta_2, \gamma_3, \beta_3)$ comprise

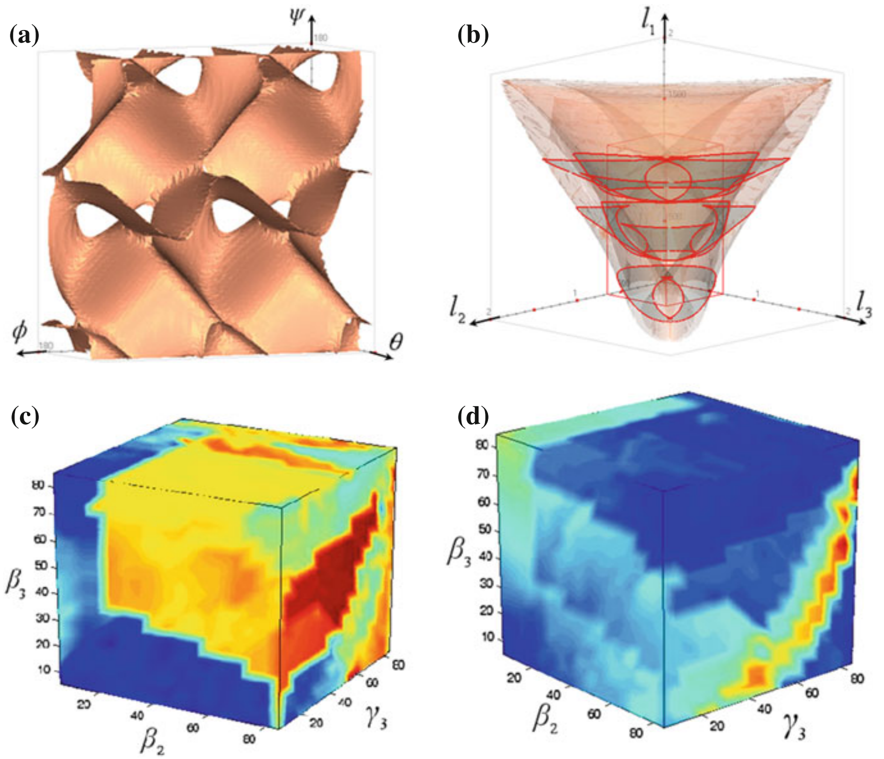


Fig. 3 Case 3: DKP singularity locus in the **a** workspace and **b** joint space; Design parameter space according to indicators (c) R_1 and (d) R_2

the *design parameter space* in which each point represents a possible design, and has an associated value according to the indicator under evaluation. We propose two indicators. The first, R_1, evaluates the regularity, comparing the number of nodes forming the DKP singularity curves among different sections of $\theta_i \in (0, \pi)$. The second indicator, R_2, assesses the quality of the curves in each θ_i section, penalizing the designs for which the curves cover a larger region. The results are displayed in Fig. 3c and d, the blue colored points indicate the geometric parameters corresponding to optimum designs, and the red ones the worst (see details in [9]).

The optimum design parameter space can be computed by intersecting the optimum values of both graphs in Fig. 3c and d. Then, any point belonging to the resultant optimum space constitutes a valid design complying with the established requirements. For example, the following design: $\beta_2 = 15°$, $\gamma_3 = 10°$ and $\beta_3 = 20°$ is an optimum design with regular workspace, maintaining a similar pattern of the singularity curves in different sections of the workspace.

4 Conclusions

Dimensional synthesis of a spatial orientation manipulator has been approached, focusing mainly on the configuration space entities. Analyzing different designs, it has been shown that the ones capable of transitioning between solutions exhibit a larger workspace. Not only the size of the operational workspace but the evaluation of its shape has been also considered, representing the design parameter space according to the different requirements. Then the designer can choose any point belonging to the set of optimum values achieved. The proposed procedure is valid for 3-DoF planar or spatial parallel manipulators that exhibit the transitioning ability.

Acknowledgments The authors wish to acknowledge the financial support received from Ministerio de Economía y Competitividad (Project DPI2011- 22955), the European Union (Project FP7-CIP-ICT-PSP-2009-3) and Basque Government, Dpto. Educ., Univ. e Investig. (Project IT445-10) and UPV/EHU under program UFI 11/29.

References

1. Altuzarra, O., Pinto, C., Sandru, B., Hernández, A.: Optimal dimensioning for parallel manipulators: workspace, dexterity and energy. ASME J. Mech. Des. **133**(4), 041,007-7 (2011)
2. Bonev, I., Rhyu, J.: A geometrical method for computing the constant-orientation workspace of 6- PRRS parallel manipulators. Mech. Mach. Theory **36**, 1–13 (2001)
3. Gosselin, C., Angeles, J.: A global performance index for the kinematic optimization of robotic manipulators. J. Mech. Des. **113**(3), 220–226 (1991)
4. Husty, M.: Non-singular assembly mode change in 3-RPR-parallel manipulators. In: Kecskeméthy, A., Müller, A. (eds.) Computational Kinematics. Springer, Berlin (2009)
5. Liu, X.J., Guan, L., Wang, J.: Kinematics and closed optimal design of a kind of PRRRP parallel manipulator. ASME J. Mech. Des. **129**(5), 558–563 (2007)
6. Merlet, J.P.: Designing a parallel manipulator for a specific workspace. Int. J. Robot. Res. **16**(4), 545–556 (1997)
7. Monsarrat, B., Gosselin, C.: Workspace analysis: and optimal design of a 3-leg 6-DOF parallel platform mechanism. IEEE Trans. Robot. Autom. **19**(6), 954–966 (2003)
8. Ottaviano, E., Ceccarelli, M.: An analytical design for CaPaMan with prescribed position and orientation. In: Proceedings of the ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference, Baltimore (2000)
9. Urízar, M.: Methodology to enlarge the workspace of parallel manipulators by means of non-singular transitions. Ph.D. thesis, University of the Basque Country (UPV/EHU). http://www.ehu.es/compmech/members/monica-urizar/research/ (2012)
10. Urízar, M., Petuya, V., Altuzarra, O., Hernández, A.: Assembly mode changing in the cuspidal analytic 3-R PR. IEEE Trans. Robot. **28**(2), 506–513 (2012)
11. Zein, M., Wenger, P., Chablat, D.: Non-singular assembly mode changing motions for 3-RPR parallel manipulators. Mech. Mach. Theory **43**(4), 480–490 (2008)

Self-Calibration of Redundantly Actuated PKM Exploiting Kinematic Landmarks

Andreas Müller and Maurizio Ruggiu

Abstract A self-calibration method for redundantly actuated parallel manipulators (RA-PKM) is proposed that uses motion reversal points (MRP) of actuators as kinematic calibration landmarks. The basic principle is to restrain a RA-PKM to 1 DOF and detect the MRP of redundant actuators. The difference of measured MRP and those deduced from a kinematic model embodies the calibration error. Therewith a numerical adaptation scheme is introduced. Simulation results for a 3 DOF RA-PKM confirm very high accuracy of the method.

Keywords Calibration · Parallel mechanisms · Redundant actuation · Singularities

1 Introduction

Kinematic calibrationaut]Müller, A. of robotic manipulators is commonly based on theaut]Ruggiu, M. acquisition of redundant measurement data [4, 10, 13, 14]. Actuator readings are compared with external end-effector (EE) measurements. The need for external measurement devices makes the calibration expensive and prohibits simple repetition. Moreover, due to the external measurements, traditionally calibration methods are inherently intrusive. In order to alleviate this intrusion several authors [1, 5, 12, 15, 17] proposed to acquire redundant measurements by locking selected joints of a parallel kinematics machine (PKM). Other schemes aiming at semiautonomous calibration are reported in [16] and [20] where some passive joints are equipped with sensors providing redundant sensor data without application of external devices. A fully autonomous self-calibration method should not require additional

A. Müller (✉)
Institute of Mechatronics, Chemnitz, Germany
e-mail: andreas.mueller@ifm-chemnitz.de

M. Ruggiu
Department of Mechanical Engineering, University of Cagliari, Cagliari, Italy
e-mail: ruggiu@dimeca.unica.it

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_11,

sensors, externally or at passive joints. An import observation is that redundant actuation implies sensor redundancy since it does not increase the DOF of a PKM but gives rise to more encoder readings than necessary to position the PKM. It thus allows for application of the traditional calibration methods as pursued in [18]. Other tailored calibration methods were reported in [2, 19] exploiting the tracking error projected to the null-space of the forward kinematics Jacobian. In this paper a calibration method is proposed that follows a completely different approach. Instead of comparing redundant sensor data the occurrence of kinematic landmarks is compared and exploited for adaptation of the geometric machine parameters. These landmarks are detected inherently by means of actuator measurements. The redundancy required for calibration is thus achieved by the actuation redundancy. The simulations reveal a high accuracy of the method. Throughout the paper configurations where the velocity of one actuator becomes zero, while the EE performs continuous motion, will be called **motion reversal points** (MRP) of that actuator. δ denotes the manipulator's DOF. The vector summarizing the geometric model parameters is denoted $\pi \in \Pi$ where Π is a p-dimensional parameter space manifold.

2 Main Principle: Motion Reversal Points

Undoubtedly kinematic singularities are significant kinematic landmarks intrinsically related to the PKM geometry, and it shall be expected that this can be exploited for the calibration purpose. A singularity-based calibration method was proposed for a non-redundant planar 3-DOF PKM [6–8], and later applied [11] to the calibration of a non-redundant spatial 3$\underline{\text{P}}$RS PKM. The basic idea of that method is to detect active input singularities [3] in the plant by measurements in the actuated joints as well as in a parameterized kinematic model, and to adapt the geometric parameters so that these singularities coincide. Active singularities are characterized by a reversing motion, i.e. a zero velocity, of some actuator coordinates for a continuous EE-motion. This allows to detect them without additional sensors.

The calibration scheme proposed in [6–8, 11] was originally developed for non-redundantly actuated PKM. This poses the apparent problem that the PKM must be steered into a critical configuration in which it is not fully restrained by means of the actuators, and cannot be controlled safely. Moreover in these situations the actuator may mutually interfere. To cope with this problem and ensure passage through the singularity, in [6–8] the PKM was given an initial motion so to passively swing through the input singularity, enabling detection of the reversing motion of one of the actuators. In order to actually detect these singularities the PKM's mobility must be restrained, which is apparent observing that singularities form lower dimensional subvarieties in the configuration space (c-space) that almost sure will not be hit when the PKM can move freely in the c-space. To this end in [6–8] the mobility was reduced to 1 DOF by deactivating the one (backdrivable) actuator for which a MRP (i.e. a zero crossing of its velocity) is to be detected, and locking all remaining $\delta - 1$ actuators. Then only motions of the deactivated (backdrivable) actuator are

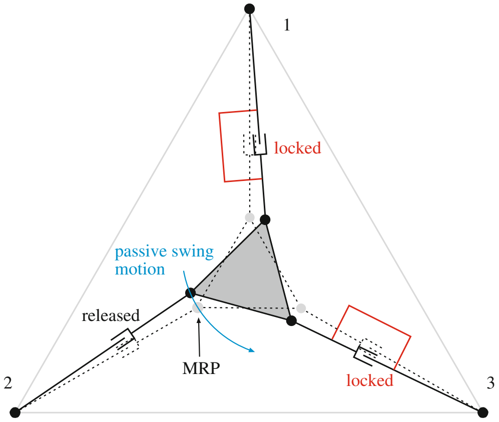


Fig. 1 MRP detection of a 3R$\underline{\text{P}}$R manipulator via a passive swing motion. Actuators 1 and 2 are locked and 3 is free to move passively

possible, and the problem of detecting the input singularities of that particular actuator reduces to a one-dimensional problem. Clearly then the non-redundantly actuated PKM is not controllable anymore since the only one movable actuator is passive. It was proposed to let the PKM passively swing through the anticipated singularity. In Fig. 1 actuators 1 and 3 could be locked and the MRP of the released actuator 2 be observed when the PKM is set in motion. The locked actuators 1 and 3 (with fixed lengths) together with the platform constitute a planar 4-bar linkage. The mount point of actuator 2 moves on the coupler curve of that 4-bar, and a MRP occurs when the point reaches the point on that curve that is closest to the mount point on the base. This configuration is indicated in Fig. 1 by a dashed line. Such passive motion requires external stimulus and precaution. Redundant actuation allows to exploit this principle in a safe and reliable way. The crucial point is that MRP can also be observed in RA-PKM but without meeting input singularities, i.e. without entering critical poses. Input-singularities of a non-redundantly actuated PKM with DOF δ can be eliminated by introducing $m > \delta$ actuators, of which $\rho = m - \delta$ are redundant that can always control the PKM. Consequently, actuation redundancy allows controlling the RA-PKM through points that are input-singularities of the non-redundant PKM, i.e. when controlled by some δ out of the m actuators. Since MRP are significant points, that are inherently related to the mechanism geometry, they can be considered as kinematic **calibration landmarks**. This is the basis for a self-calibration method introduced in [9] where the feasibility was demonstrated by application of an ad hoc minimization algorithm. In this paper the method is completed by a computationally efficient update algorithm. The redundantly actuated 4R$\underline{\text{P}}$R in Fig. 2a (3R$\underline{\text{P}}$R with an added fourth actuated chain is used as example. The joints are mounted at the corners of a rectangle. The RA-PKM still has $\delta = 3$ DOF but is actuated by $m = 4$ prismatic actuators. Also for this RA-PKM joint 2 exhibits MRP, but now, due to the actuation redundancy, the manipulator can safely be controlled through these MRP

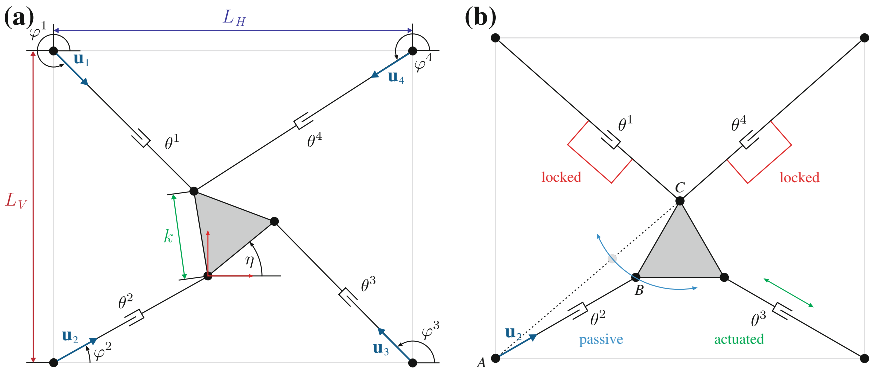


Fig. 2 **a** Planar 3-DOF 4R$\underline{\text{P}}$R RA-PKM. **b** Detection of MRP of actuator 2. Actuators 1 and 4 are locked, and 3 is actuated. The MRP occurs when point B is closest to point A, as indicated

as shown in Fig. 2a. After locking $\delta - 1 = 2$ actuators there are still $m - \delta + 1 = 2$ actuators left of which one is that for which MRP are sought and the other one can be used to drive the (1-DOF) system. Actuators 1 and 4 can be locked and actuator 3 be used to move the manipulator while observing the MRP of actuator 2. A deviation of the geometric parameters from the actual plant geometry is reflected by MRP occurring at different locations. Minimizing this difference is a means for model calibration. This was pursued in [9] where the calibration error was defined as the squared difference of MRP analytically computed from the kinematic model and MRP detected in the plant for N different measurements. For example in Fig. 2b this error would be $e := \sum_{i=1}^{N} \left(\widehat{\theta}_0^{2,i} - \theta_0^{2,i}\right)^2$ where $\theta_0^{2,i}$ is the MRP of actuator joint 2 deduced from the model, and $\widehat{\theta}_0^{2,i}$ is the corresponding MRP detected in the plant for the ith measurement. The critical point when directly minimizing e is that this is a non-linear problem and that it requires an analytic expression for the MRP. Here a MRP-based calibration method is introduced that does not suffer from these restrictions. For RA-PKM with actuation redundancy $\rho = m - \delta$ it consists of three constituent parts: (1) the restriction of the RA-PKM mobility to 1 DOF by locking $\delta - 1$ actuators, (2) detection of MRP, and (3) model adaptation.

3 Selection of Locked Actuators

The RA-PKM is restrained to 1-DOF motions by locking $\delta - 1$ actuators. This leaves a manipulator with $m - \delta + 1$ movable actuators. Out of these one actuator is used to drive the restrained RA-PKM. The remaining $\rho = m - \delta$ actuators are deactivated and presumed to be backdrivable. Since actuators are equipped with encoders the MRP of these ρ joints can be detected by controlling the 1-DOF motion. The choice of locked actuators is not unique. In general there are $C_{\delta-1}^{m} = \binom{m}{\delta-1}$ different possibilities to lock $\delta - 1$ of the m actuators. Most RA-PKM reported in the literature possess simple actuation redundancy, i.e. $m = \delta + 1$, $\rho = 1$, and this assumed in the following. Then $\delta - 1 = m - 2$ actuators must be locked, for which there are $C^m_{m-2} = \frac{(m-1)m}{2}$ different possibilities. Fixing $\delta - 1$ actuators leaves two unlocked. One of these two can be controlled so to detect the MRP of the other one. In this way, for each one of the C^m_{m-2} combinations, the MRP of one deactivated actuator can be detected. For each combination of locked actuators there are two possible actuation schemes. In total there are $2C^m_{m-2}$ different actuation schemes for detecting MRP (generally $(\rho + 1)\, C^m_{\delta-1}$).

Denote $\theta = \left(\theta^1, \ldots, \theta^m\right)$ the vector of actuator coordinates. For each of the C^m_{m-2} possibilities denote with θ_l, $l = 1, \ldots, C^m_{m-2}$ the vector of the $\delta - 1$ locked actuator coordinates. The remaining two **free actuator** coordinates are denoted with θ^j and θ^i, where j refers to the controlled and i to the free passive actuator for which the MRP is sought. The coordinate vector θ is thus partitioned into θ_l, θ^j and θ^i. In Fig. 2b it is $\theta_l = \left(\theta^1, \theta^4\right)$, $\theta^j = \theta^3$, and $\theta^i = \theta^2$.

4 Detection of MRP

For a specified set l of locked actuators, the detection of MRP of θ^i in the model requires an **indicator function**, denoted with F^l_{ij}, such that $F^l_{ij} = 0$ if and only if $\dot{\theta}^i = 0$ for any $\dot{\theta}^j$. The condition $F^l_{ij} = 0$ allows inferring the joint coordinate θ^j_0 of the controlled actuator where MRP of the free passive actuator i occurs in the model for given θ_l and geometry π. An obvious candidate for such an indicator function is the velocity inverse kinematics solution for actuator i (Sect. 6). The MRP of actuator joint i deduced from the model is denoted θ^i_0. In Fig. 2b the indicator function F_{23} returns the velocity of joint 2 when the manipulator is driven by actuator 3 with the remaining joints 1 and 4 locked. The MRP in the plant are detected monitoring the sign of the velocity of the free passive actuator i while performing a smooth 1-DOF motion controlled by actuator j. The detected MRP are only exact up to the measuring accuracy/encoder resolution, and denoted $\widehat{\theta}^i_0$.

5 Calibration Algorithm

The model adaptation exploits the difference of MRP predicted in the model and those measured in the plant. By locking different sets of $\delta - 1$ actuators the MRP can be detected for all m actuators. For a particular set of locked actuators, indexed with l, the remaining two actuator coordinates are interchangeably used to drive the 1-DOF system. This detection is repeated at N different points in the c-space, i.e. for different values of θ_l giving rise to an input data set for the calibration.

The strategy is to detect the MRP of actuator i for a set of locked actuators. Crucial for the calibration algorithm is the indicator function. It is thus desirable to have an indicator function that only depends on θ^i and θ_l, which represent δ non-redundant actuator coordinates. But, depending on the complexity of the kinematics, it may not possible to analytically construct such an indicator function. These cases are distinguished in the following. The PKM pose is locally uniquely determined by δ actuator coordinates, and in particular by the actuator coordinates θ^i and θ_l. Now by definition this unique dependence ceases at the MRP of actuator i. In the 4R$\underline{\text{P}}$R example in Fig. 2b the motion is determined by θ^1, θ^4, and θ^2 except at the MRP of joint 2. Nevertheless, presumed that the closed loop constraints can be expressed in terms of these δ actuator coordinates, it can be assumed that the indicator function attains the form $F^l_{ij}(\theta^i, \theta_l; \pi)$. That is, F does not depend on the controlled actuator coordinate θ^j nor on any coordinates of passive joints. Let $\widehat{\theta^i_0} = \theta^i_0 + \Delta\theta^i_0$ be the measured joint coordinate of the passive actuator i where its MRP occurs. $\Delta\theta^i_0$ represents the deviation from the joint coordinate θ_0 where MRP occurs in the model. Denote with

$$\pi = \pi_0 + \Delta\pi \tag{1}$$

the (unknown) geometric parameters of the plant. Here π_0 is the nominal geometry used as initial value in the model, $\Delta\pi$ is the geometric imperfection to be estimated.

Evaluating the indicator function with the measured MRP values and the nominal model parameters will yield $F^l_{ij}(\widehat{\theta^i_0}, \theta_l, \pi_0) \neq 0$. That is, using the nominal model parameters, the analytic indicator function would not detect the MRP. Moreover, the value of F^l_{ij} does represent a **calibration error**. This error is minimized adapting π_0.

Since the locked actuator coordinates θ_l are input to model and plant the only information about model uncertainties are conveyed by $\widehat{\theta^i_0}$. Hence, treating $\Delta\theta^i_0$ and $\Delta\pi$ as variables, and neglecting measurement errors, leads to the first-order condition

$$F^l_{ij}(\widehat{\theta^i_0}, \theta_l, \pi_0) - \frac{\partial F^l_{ij}}{\partial\theta^i}\Delta\theta^i_0 + \frac{\partial F^l_{ij}}{\partial\pi}\Delta\pi = 0. \tag{2}$$

The measurement is repeated at N_l different locations of MRP of θ^i giving rise to an overdetermined linear system

$$\begin{pmatrix} \frac{\partial F^l_{ij}}{\partial\theta^i}\Delta\theta^i_{0,1} - F^l_{ij}(\widehat{\theta^i_{0,1}}, \widehat{\theta_{l,1}}, \pi_0) \\ \frac{\partial F^l_{ij}}{\partial\theta^i}\Delta\theta^i_{0,2} - F^l_{ij}(\widehat{\theta^i_{0,2}}, \widehat{\theta_{l,2}}, \pi_0) \\ \vdots \\ \frac{\partial F^l_{ij}}{\partial\theta^i}\Delta\theta^i_{0,N_l} - F^l_{ij}(\widehat{\theta^i_{0,N_l}}, \widehat{\theta_{l,N_l}}, \pi_0) \end{pmatrix} = \begin{pmatrix} \frac{\partial F^l_{ij}}{\partial\pi} \\ \frac{\partial F^l_{ij}}{\partial\pi} \\ \vdots \\ \frac{\partial F^l_{ij}}{\partial\pi} \end{pmatrix} \Delta\pi \tag{3}$$

written as $\mathbf{y}_l = \mathbf{M}_l \Delta\pi$, where the partial derivatives of F_{ij}^l are evaluated at $\widehat{\theta}_{0,k}^i$, $\theta_{l,k}$, π_0, $k = 1, \ldots, N_l$. The values of $\Delta\theta_{0,k}^i = \widehat{\theta}_{0,k}^i - \theta_{0,k}^i$ are computed from the difference of measured values and those deduced from the model. For p parameters $\mathbf{M}_l$ is a $N_l \times p$ matrix. This measurement procedure can be carried out for different choices of locked actuators indicated by the subscript l, for which there are $2C_{m-2}^m$ options. It is usually not necessary to exhaust all these combinations, however. Denote with $\mathbf{M}$ the $N \times p$ matrix comprising the $\mathbf{M}_l$ submatrices, and with $\mathbf{y}$ the corresponding left-hand side vector of length N. The system (3) is solved via a pseudoinverse of $\mathbf{M}$ as $\Delta\pi = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{y}$. The final update for the parameter vector is $\pi = \pi_0 + \Delta\pi$. The solution π of the linear approximation (2) may not lead to a vanishing F_{ij}^l. Therefore the adaptation step is repeated. Denoting with π_ν the obtained parameter values at step $\nu = 1, 2, \ldots$ then an improved estimate $\pi_{\nu+1}$ is found by application of the above update step with π_ν as initial parameter. The calibration starts with π_0.

6 Simulation Example

The planar 3 DOF 4R$\underline{\text{P}}$R RA-PKM in Fig. 2a has $m = 4$ of actuators and actuation redundancy $\rho = 1$. The moving platform forms an equilateral triangle with side lengths k. The mount points on the ground are located at the corners of a rectangle with side lengths L_H and L_V. The reference displacements of actuators 1 and 4 for $\theta^1 = \theta^4 = 0$ is denoted with L_1 and L_4. The parameter vector is $\pi = (L_1, L_4, L_H, L_V, k)$. The nominal parameter values are $\pi_0 = (0, 0, 0.5, 0.5, 0.15)$ m.

Numerical results are reported using MRP of actuators 2 and 3 when joints 1 and 4 are locked. That is, only one set of locked actuators is used, for which the MRP of joint 2 is determined when controlled by actuator 3, and the MRP of joint 3 is determined when controlled by actuator 2. The calibration performance is examined for two sets of calibration points in Fig. 3. During calibration actuators θ_1, θ_4 are positioned and locked so that actuator 2 of the model with nominal parameters π_0 exhibits a MRP when point B of the platform coincides with the calibration point (Fig. 3). The first set consists of 4×4 samples on an equidistant grid. The second set comprises 20 points uniformly distributed on a circle. The plant parameters are randomly set to $\pi = \pi_0 + (\Delta L, -\Delta L, -\Delta L, -\Delta L, +\Delta L)$ with a deviation of $\Delta L = 10^{-3}$ m and $\pi_0 = (0, 0, 0.5, 0.5, 0.15)$ m. Simulating encoder precision the joint measurements in the plant are quantized with encoder resolution Δx. Fig. 4a shows the evolution of calibration error for perfect measurement (no measurement errors), and for quantizations $\Delta x = 10^{-4}$ and $\Delta x = 10^{-3}$, when 16 calibration points on the grid in Fig. 2b are used. Shown is the mean value of the absolute deviation of the model from the plant parameters after step i. After 3 steps the calibration converges to the computation accuracy if joint angles are measured perfectly. Otherwise the convergence is bounded by the quantization Δx. For the chosen quantization convergence is observed after one iteration. The final calibration error is below the

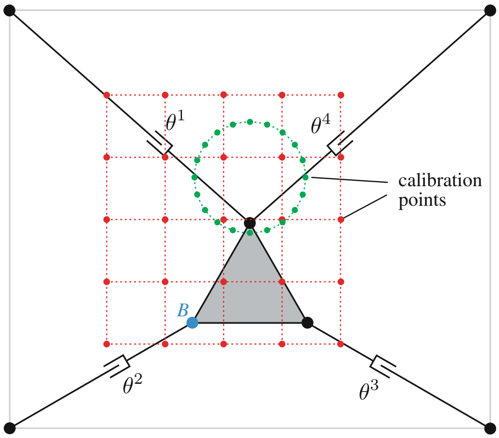


Fig. 3 Calibration points on a grid and on a circle in workspace

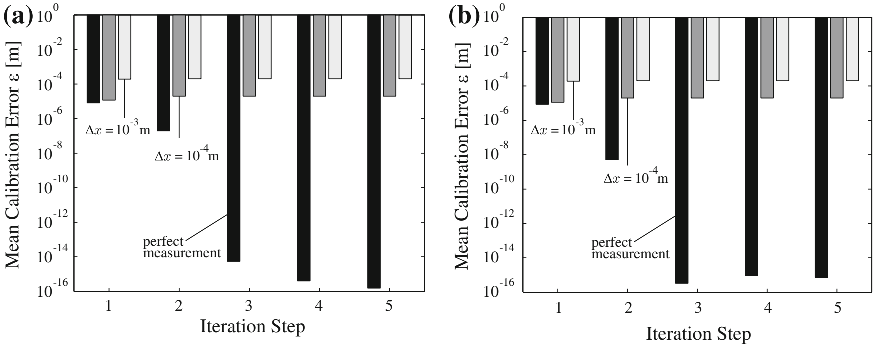


Fig. 4 **a** Calibration error evolution for different measurement accuracies Δx (**a**) using 16 calibration points on a grid in Fig. 3, **b** 20 calibration points on the *circle* in Fig. 3 ($\Delta L = 10^{-3}$)

uncertainty Δx, which can be explained as an averaging effect. Similar results are found for the 20 calibration points on the circle in Fig. 4b.

7 Conclusions and Outlook

A self-calibration method for RA-PKM has been proposed based on motion reversal points. The method does not require any additional sensors. The only (but possibly critical) condition on RA-PKM is that the drives for which MRP are detected must be backdrivable. The general concept is introduced and numerical results are shown for a simple 3 DOF planar RA-PKM. For this example the method shows very good performance and accuracy.

References

1. Bennett, D.J., Hollerbach, J.M.: Autonomous calibration of a single loop closed kinematic chain formed by manipulators with passive endpoint constraints. IEEE T. Rob. Aut. **7**(5), 597–606 (1991)
2. Cong, S., et al.: Kinematic Parameters Auto-Calibration of Redundant Planar 2-Dof Parallel Manipulator. Parallel Manipulators, New Developments. I-Tech Publishing, Vienna (2008)
3. Gosselin, C., Angeles, J.: Singularity analysis of closed loop kinematic chains. IEEE Trans. Rob. Automat. **6**(3), 281–290 (1990)
4. Hollerbach, J.M., Wampler, C. W.: The calibration index and taxonomy for robot kinematic calibration methods. Int. J. Rob. Res. **15**(6), 573–591 (1996)
5. Khalil, W., Besnard, S.: Self-calibration of Stewart-Gough parallel robots without extra sensors. IEEE Trans. Rob. Autom. **15**(6), 1116–1121 (1999)
6. Last, P., Budde, C., Hesselbach, J.: Self-calibration of the HEXA-parallel-structure. Proceedings of the IEEE International Conference Automation Science and Engineering (CASE), Edmonton, Canada, 1–2 Aug 2005
7. Last, P., Schütz, D., Raatz, A., Hesselbach, J.: Singularity based calibration of 3-DOF fully parallel planar manipulators. 12th IFToMM World Congress, France, 18–21 June 2007
8. Last, P., Raatz, A., Hesselbach, J.: Singularity-based calibration—a novel approach for absolute-accuracy enhancement of parallel robots. In: Lazinica A., Kawai H. (eds.) Robot Manipulators New Achievements. InTech Publishing, Vienna (2010)
9. Müller, A., Ruggiu, M.: Self-calibration of redundantly actuated PKM based on motion reversal points. In: Lenarcic J., Husty M. (eds.) Latest Advances in Robot Kinematics (ARK), Springer, Dordrecht (2012)
10. Nahvi, A., Hollerbach, J.M., Hayward V.: Calibration of parallel robot using multiple kinematic closed loops, IEEE ICRA (1994)
11. O‘Brien, S.M., Carretero, J.A., Last, P.: Self calibration of 3-PRS manipulator without redundant sensors. Trans. Can. Soc. Mech. Eng. **31**(4), 483–494 (2007)
12. Ota, H. et. al.: Forward kinematic calibration method for parallel mechanism using pose data measured by a double ball bar system. Int. Conf. Parallel Kin. Mach. 57–62 (2000)
13. Rauf, A., Kim, S.G., Ryu, J.: Complete parameter identification of parallel manipulators with partial pose information using a new measurement device. Robotica **22**(6), 689–695 (2004)
14. Rauf, A., Pervez, A., Ryu, J.: Experimental results on kinematic calibration of parallel manipulators using a partial pose measurement device. IEEE Trans. Rob. Control **22**(2), 379–384 (2006)
15. Ryu, J., Rauf, A.: A new method for fully autonomous calibration of parallel manipulators using a constraint link. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Italy, 8–12 July 2001
16. Wampler, C.W., Hollerbach, J.M., Arai, T.: An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. IEEE Trans. Robot. Autom. **11**(5), 710–724 (1995)
17. Weck, M., Staimer, D.: Accuracy issues of parallel kinematic machine tools: compensation & calibration. International Conference on Parallel Kinematic Machines (2000)
18. Yiu, Y.K., Meng, J., Li, Z.X.: Auto-calibration for a parallel manipulator with sensor redundancy. Proceedings of the IEEE International Conference on Robotics Automation (ICRA), pp. 3660–3665. Taipei (2003)

19. Zhang, Y.X., Cong, S., Shang, W.W., Li, Z.X., Jiang, S.L.: Modeling, identification and control of a redundant planar 2-DOF parallel manipulator. Int. J. Control Aut. Sys. **5**(5), 559–569 (2007)
20. Zhuang, H.: Self-calibration of parallel mechanisms with a case study on Stewart platforms. IEEE Trans. Rob. Autom. **13**(3), 387–397 (1997)

Solving the Forward Kinematics of Cable-Driven Parallel Robots with Neural Networks and Interval Arithmetic

Valentin Schmidt, Bertram Müller and Andreas Pott

Abstract This paper investigates a new approach for solving the forward kinematics of cable-driven parallel robots. This approach combines an interval algorithm with neural networks to provide a fast but accurate initial guess. The neural networks increase the computation speed by a factor of 200 or more, while the interval algorithm provides guaranteed convergence and a definite solution to any chosen degree of accuracy. Iterative techniques are faster still, but the proposed algorithm is considered real-time feasible.

Keywords Cable-driven robots · Neural networks · Interval analysis · Kinematics

1 Introduction

Cable-driven parallel robots, from now on referred to as cable robots, are a class of parallel robots actuated by flexible cables instead of rigid members.

We will differentiate between two types of cable robots. Completely/redundantly restrained cable robots, where the number of cables m exceeds the degrees of freedom n in order to achieve full control; and a second type of cable robot: the suspended cable robots. Suspended cable robots rely on an external wrench to control the robot position and are most often seen as hanging structures.

The forward kinematics—obtaining platform pose from cable lengths—of parallel machines is a difficult problem. However, for the operation and for advanced

V. Schmidt (✉) · B. Müller · A. Pott
Fraunhofer Institute for Manufacturing Engineering and Automation IPA,
70569 Stuttgart, Germany
e-mail: Valentin.Schmidt@ipa.fraunhofer.de

B. Müller
e-mail: Bertram.Mueller@ipa.fraunhofer.de

A. Pott
e-mail: Andreas.Pott@ipa.fraunhofer.de

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_12,

control techniques of cable robots, a numerically stable and fast computation of the forward kinematics is needed. In practice, this is often achieved by using optimization algorithms.

In this paper an algorithm for completely restrained cable robots, based on combining neural networks with interval arithmetic, is introduced and tested. This combined approach offers greater computational speed than algorithms purely based on interval arithmetic, whilst maintaining strict confidence in the obtained solution.

2 Literature Review

Various methods to compute the forward kinematics of parallel manipulators already exist. In some specific cases the problem can be simplified due to specific geometrical traits, resulting in a set of algebraic descriptions which can easily be solved symbolically [3, 12]. Algebraic formulations have also been made for the general case, resulting in high-degree polynomials which are very difficult to solve [7].

Merlet introduced a numerically more stable approach for the general parallel machine using Interval Arithmetic [9].

These approaches stand in contrast to the iterative optimization techniques, which evaluate the inverse Kinematics repeatedly, to gain an increasingly accurate pose with each step. The computation time of such techniques is acceptable and has been successfully implemented for real-time execution on robot systems [10].

Suspended cable robots require additional considerations when evaluating the kinematics. Since the external wrench applied by gravity is an integral part to the robots structure, it needs to be taken into account when evaluating the robot's pose. Recent publications discuss this static and dynamic coupling and propose algebraic solutions [2, 4]. Ghasemi has presented a successful implementation of neural networks to find the solution for a suspended cable robot [5].

2.1 Shortcomings of Existing Methods

Despite having a wide range of computational approaches to choose from, the evaluation of forward kinematics remains a challenge, especially when considering real time constraints on modern machines. Simplifying the problem through the selection of special geometries is not always a feasible step depending on the constraints of the robot considered.

Mathematically exact methods, which are capable of finding all possible solutions and using heuristics to select the right one, are not feasible for production machines due to numerical instabilities or excessive computation time. Interval arithmetic on the other hand yields certain guarantees due to its deterministic nature. Unfortunately, interval arithmetic routines are based on exhaustive search algorithms and typically do not perform in a real-time environment.

Faster iterative optimization algorithms, however, demand a good initial guess for the solution and do not guarantee convergence. This can have unexpected consequences and requires additional sub-routines to manage non-convergence, or even divergence from the solution.

Neural networks are computationally fast algorithms but also lack the capability of providing a guaranteed solution or any indication of confidence. Further, these networks need to be trained a-priori, consuming additional time and effort.

All approaches suffer under kinematic sensitivity [8]. This provides a measure of how numerically sensitive the poses are to changes in the cable lengths and an indication why some solutions are hard to find. Further discussion is out of scope for this paper.

2.2 Benefits of the Combined Approach

Using neural networks to provide a fast initial estimate for the solution and then applying an interval algorithm to the smaller search space, combines the strengths of these two methods. Unlike optimization algorithms interval analysis can easily differentiate between whether an exact solution can be found in the given bound, to a predetermined level of accuracy, or whether a solution is indeterminable. Even detecting singularities is plausible [9]. Neural Networks, once trained, provide a very fast evaluation with minimal computational effort.

It is this combination approach which enables a robust, but still relatively fast computational method.

3 Description of the Combination Approach

The cable-driven parallel robot IPAnema illustrated in Fig. 1, has eight cables and six degrees of freedom, making it redundantly restrained. The geometry is described using two coordinate systems, one for the platform $\mathcal{K}_{\mathcal{P}}$ and one global coordinate system for the base $\mathcal{K}_{\mathcal{O}}$. The vector notation describes the inverse kinematics for cable i as:

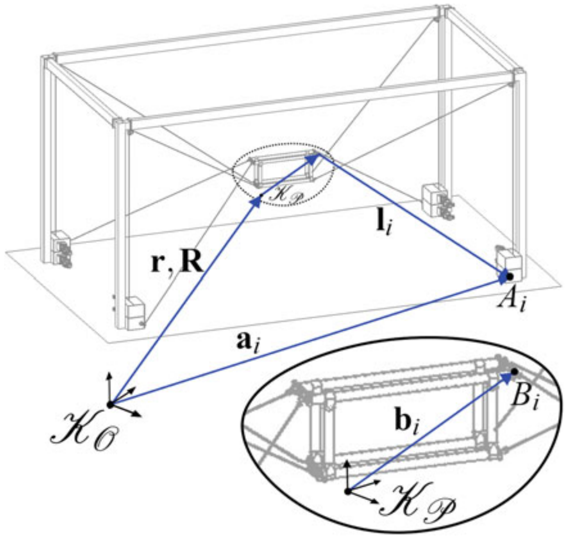
$$\|\mathbf{a}_i - \mathbf{r} - \mathbf{R}\mathbf{b}_i\|_2 = l_i \quad \text{for}\, i = 1, \ldots, m \tag{1}$$

where $\mathbf{a}_i$, $\mathbf{b}_i$, $\mathbf{r}$ are vectors describing attachment points and platform position, l_i is the length of the cable and $\mathbf{R}$ the rotation matrix at a given pose.

The forward kinematics currently implemented in the IPAnema cable robot [10] uses a Levenberg-Marquardt (LM) algorithm to optimize the function

$$\Psi_i\,(\mathbf{l}, \mathbf{r}, \mathbf{R}) = (\|\mathbf{a}_i - \mathbf{r} - \mathbf{R}\mathbf{b}_i\|_2)^2 - l_i^2 \quad \text{for}\, i = 1, \ldots, m \tag{2}$$

to finde the pose $\mathbf{r}$, $\mathbf{R}$ for a set of cable lengths l_i.


$\mathbf{r}, \mathbf{R}$
$\mathcal{K}_{\mathcal{P}}$
$\mathbf{l}_i$
A_i
$\mathbf{a}_i$
$\mathcal{K}_{\mathcal{O}}$
$\mathbf{b}_i$
B_i
$\mathcal{K}_{\mathcal{P}}$

Geometry in [m]

	i	x	y	z
$\mathbf{a}_i$:	1	[−4.3	3.0	4.6]^T
	2	[4.3	3.0	4.5]^T
	3	[4.0	−2.7	4.6]^T
	4	[−4.0	−2.8	4.7]^T
	5	[−4.3	2.9	0.5]^T
	6	[4.3	2.9	0.4]^T
	7	[4.0	−2.8	0.4]^T
	8	[−4.0	−2.8	0.6]^T
$\mathbf{b}_i$:	i	x	y	z
	1	[−0.7	0.2	−0.1]^T
	2	[0.6	0.2	−0.1]^T
	3	[0.6	−0.1	−0.1]^T
	4	[−0.6	−0.1	−0.1]^T
	5	[−0.8	0.2	0.4]^T
	6	[0.8	0.2	0.4]^T
	7	[0.8	−0.0	0.4]^T
	8	[−0.8	−0.0	0.4]^T

Fig. 1 IPAnema Robot with Geometrical Parameters: base vector $\mathbf{a}_i$ and platform vector $\mathbf{b}_i$

For the interval algorithm a different parameterization, based on distance equations, is used. This parameterization focuses on finding the position of f linearly independent reference points in the global coordinate frame and hence the pose of the platform. These reference points on the platform were chosen to be cable attachment points and are chosen so that all other attachment points j are linearly dependent on these reference points k. This relation in the coordinate system $\mathcal{K}_{\mathcal{P}}$ is described by

$$\mathbf{b}_j = \sum_k \mathbf{C}\mathbf{b}_k, \tag{3}$$

where the conversion matrix $\mathbf{C}$ is calculated offline.

The equation sets describing the kinematics, which are solved by the interval algorithm are then as follows. The cable length for each reference point (x_k, y_k, z_k)

$$\left(x_k - A_k^x\right)^2 + \left(y_k - A_k^y\right)^2 + \left(z_k - A_k^z\right)^2 = l_k^2 \quad \text{for } k \in 1, 2, \ldots, f, \tag{4}$$

the remaining cable lengths

$$\left(\sum_{k=1}^{f} \mathbf{C}x_k - A_j^x\right)^2 + \left(\sum_{k=1}^{f} \mathbf{C}x_k - A_j^y\right)^2 + \left(\sum_{k=1}^{f} \mathbf{C}x_k - A_j^z\right)^2 = l_j^2$$
$$\text{for } j \in f + 1, \ldots, n \tag{5}$$

and the distance between reference point pairs

$$\left(x_p - x_q\right)^2 + \left(y_p - y_q\right)^2 + \left(z_p - z_q\right)^2 = \delta_{pq}^2 \quad \text{for } p, q \in 1, 2, \ldots, f, p \neq q. \tag{6}$$

This parameterization is better suited for interval methods as it avoids overestimation of the interval bounds due to each variable being represented only once in each equation. It was found that this formulation of the forward kinematics problem was not beneficial for the iterative methods over an Euler Angle representation used to generate rotation matrix **R** in Eqs. (1) and (2).

3.1 Neural Network

In the combination approach the pose of the platform is initially estimated by a set of neural networks. This greatly reduces the search space for the more time consuming interval algorithm. The neural network sets return a particular solution for the four reference point positions, whose error/uncertainty bounds define the initial search space of the interval algorithm.

The neural network was designed and implemented using the "Stuttgart Neural Network Simulator" (RSNNS) in the R statistical programming language [1]. The design is based on the multilayer perceptron (MLP), also used by Ghasemi [5]. In this case one neural network was used to determine the position (x, y, z) of a given reference point. Each neural network was built with four hidden layers containing 70 neurons each. Since the IPAnema cable robot has eight cables this results in a $8 \times 70 \times 70 \times 70 \times 70 \times 3$ MLP architecture.

The neural network was trained in a supervised approach with standard back-propagation, using default learning parameters of RSNNS. For a random pose the reference point positions and cable lengths are evaluated through the inverse kinematics (1). Then the artificial neural network calculates the reference point positions. The error residual sum of squares in the computed reference point positions is used to determine the weight adjustments of the individual neurons. One epoch repeats this process for every pose in a training set. A test set evaluates the performance of the neural network whose error is minimized over 800 epochs.

The full set of poses consists of 1,00,000 random poses in a workspace of $6 \times 5 \times 4$ m in x, y, z and rotations in the range of ±20° about each axis. 40,000 poses were used as a training set and the remaining 60,000 as a test set. Feature normalization was also applied for the cable lengths in the stated workspace.

Graphs in Fig. 2 show the results of a single network training. The histogram shows the distribution of the final absolute error of the test set. The maximum error determines the bounds of initial search space for the interval algorithm, ±0.02 m. This speeds up the computation time. While interval analysis is a complex iterative procedure, successive operations necessary for each neuron will only be evaluated once.

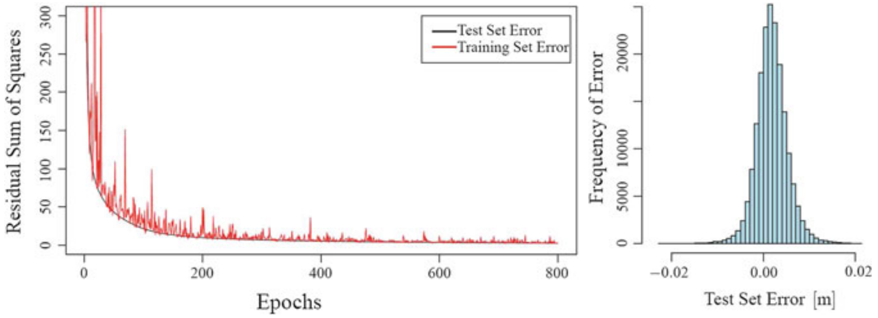


Fig. 2 Single neural network training results

3.2 Interval Algorithm

The interval algorithm based on [9] resembles a bisection method. The workspace is divided into successively smaller boxes which are then evaluated against the parametric Eqs. (4–6) with interval arithmetic. A box is a 12 dimensional hyper-cube of the (x, y, z) values of the four reference points. The starting box was based on the maximum error produced by the neural network test set. Interval arithmetic then determines whether a solution to the parametric equations exists within this box, does not exist, or cannot be determined. This branch and bound algorithm discards boxes containing no solution and continues to divide the rest, until a sufficiently small (specified by the desired accuracy) box without a solution is reached.

This is a deterministic and time consuming method to obtain a solution to the forward kinematics problem. Several techniques to speed up this process by reducing box sizes are implemented. One technique is evaluating for hull (or 2B) consistency as in constraint satisfaction problems as shown in [9]. Here the constraints of a single variable in the parametric equations are shrunk by those defined for the other variables. This can be repeated indefinitely, but tests showed that once the improvement was below 25 % it was more efficient to return to the branch and bound.

Another very effective method to reduce box size was the Interval-Gauss-Seidel method [11], which not only reduces the hyper-volume of a box, but also mitigates effects of solution clustering. Here the variable bounds are also shrunk individually using an iterative technique by applying the formula of a classic linear equation system to this variable.

These methods make the computation more efficient, but not to the same scale as the initial guess by the neural network.

Table 1 Computation time tested on IPAnema geometry

	Interval algorithm	Interval algorithm with neural network	Levenberg-Marquardt optimization [10]
Max. evaluation time [s]	0.764	0.016	N/A
Avg. evaluation time [s]	0.292	0.00121	0.000015
Max. no. of boxes/iterations	173356	303	5
Avg. no. of boxes/iterations	60690	77	4

4 Performance Evaluation

The algorithms for solving the forward kinematics are implemented in C++. This enables the use of SIMD instruction sets on an x64 architecture to provide a significant increase in speed for interval calculations and is discussed in detail by [6]. The processor used for testing was an Intel i7–2600 K with 3.40 GHz.

Table 1 shows the results of a quick comparative benchmark of three algorithms implemented under identical conditions. A random pose list of 1000 poses, with corresponding cable lengths was generated using Eq. (1). Poses were in a $4 \times 4 \times 3$ m box in the workspace and described by a rotation of $-10°$ to $10°$ for each Euler Angle. Each algorithm calculated every pose to an accuracy of 0.00001 m.

The computational speed of the Interval Algorithm is greatly increased through the use of neural networks. The average computation time decreased by more than a factor of 200, the number of boxes even more. This makes the approach more real-time feasible.

Maximum time to solution could not be evaluated for the LM optimization as it was too fast for the timer resolution, so the average was taken from the total time over 1,000 iterations. Keeping in mind that the initial pose estimate is tailored for this cable robot geometry, the optimization algorithm computes extremely fast.

Neural networks do need to be retaught for significant geometry changes, but have no limitations as to the type of geometry to solve. Further, the interval algorithm still can provide certainty of the solution and guarantee convergence, which the optimization cannot.

5 Conclusions

It was shown that combining neural networks with the standard implementation of interval arithmetic to solve the forward kinematics of cable robots provides significant decrease in computation time. This enables a more real-time feasible implementation, while retaining strengths of the interval methods. While the speed does not surpass that of highly optimized LM iterative solutions, it provides a method with guaranteed convergence, and the possibility of finding numerous poses. It is a viable alternative for actual real-time controllers.

Improvements can still be made. The neural network training algorithm could be optimized to provide a more accurate initial guess. Several steps in the interval algorithm could be taken to optimize the switching between the branch and bound, consistency checks, and the Interval-Gauss-Seidel method. This is difficult to optimize for the general case.

The teaching of neural networks is still an issue, as it diminishes the ease of changing geometric configuration. However, slight configuration changes still enable the neural network to converge enough for the interval algorithm to run at acceptable speeds. Further, the only requirement for teaching the networks is a working inverse kinematic implementation. This process can be heavily automated.

Problems of kinematic sensitivity are not addressed by either of these algorithms, but are still subject of ongoing research.

Acknowledgments This work was partially supported by the Fraunhofer-Gesellschaft Internal Programs under Grant No. WISA 823 244. Furthermore, the research leading to these results received founding for the European Community's Seventh Framework Program under Grant agreement number NMP2-SL-2011-285404-CableBot.

References

1. Bergmeir, C., Benítez, J.M.: Neural networks in R using the Stuttgart neural network simulator: RSNNS. J. Stat. Softw. **46**(7), 1–26 (2012). http://www.jstatsoft.org/v46/i07/
2. Berti, A., Merlet, J.P., Carricato, M.: Solving the direct geometrico-static problem of 3–3 cable-driven parallel robots by interval analysis: preliminary results. In: Bruckmann, T., Pott, A. (eds.) Cable-Driven Parallel Robots. vol. 12, pp. 251–268. Springer, Heidelberg (2013)
3. Bosscher, P., Williams II, R.L.: Cable-suspended robotic contour crafting system. Autom. Constr. **17**(1), 45–55 (2007)
4. Carricato, M., Abbasnejad, G.: Direct geometrico-static analysis of under-constrained cable-driven parallel robots with 4 cables. In: Bruckmann, T., Pott, A. (eds.) Cable-Driven Parallel Robots. vol.12, pp. 269–285 (2013)
5. Ghasemi, A., Eghtesad, M., Farid, M.: Neural network solution for forward kinematics problem of cable robots. J. Intell. Rob. Syst. **60**(2), 201–215 (2010)
6. Goualard, F.: Fast and correct SIMD algorithms for interval arithmetic. In: Proceedings of PARA '08, Lecture Notes in Computer Science. Springer, Trondheim (2010)
7. Husty, M.L.: An algorithm for solving the direct kinematic of stewart-gough-type platforms. Mech. Mach. Theory **31**(4), 365–380 (1996)
8. Khalilpour, S., Loloei, A., Taghirad, H., Masouleh, M.: Feasible kinematic sensitivity in cable robots based on interval analysis. In: Bruckmann, T., Pott, A. (eds.) Cable-Driven Parallel Robots, vol. 12, pp. 233–249. Springer, Heidelberg (2013)
9. Merlet, J.P.: Solving the forward kinematics of a gough-type parallel manipulator with interval analysis. Int. J. Robot. Res. **23**(3), 221–235 (2004)
10. Pott, A.: An algorithm for real-time forward kinematics of cable-driven parallel robots. International symposium on advances in robot kinematics, Springer (2010)
11. Shary, S.: Interval Gauss-Seidel method for generalized solution sets to interval linear systems. Reliable Comput. **7**, 141–155 (2001)
12. Won Jeong, J., Hyun Kim, S., Keun Kwak, Y.: Kinematics and workspace analysis of a parallel wire mechanism for measuring a robot pose. Mech. Mach. Theory **34**(6), 825–841 (1999)

Three Types of Parallel 6R Linkages

Zijia Li and Josef Schicho

Abstract In this paper, we consider a special kind of overconstrained 6R closed linkages which we call parallel 6R linkages. These are linkages with the property that they have three pairs of parallel joint-axes. We prove that there are three types of parallel 6R linkage. The first type is new, the other two also appear in a recent classification of linkages with angle equalities. We give constructions for each of the three types.

Keywords Dual quaternions · Overconstrained 6R linkages · Translation property · Angle-symmetric 6R linkages

1 Introduction

Movable closed 6R linkages have been considered by many authors (see [1, 4, 5, 11–13]). In this paper, we study a certain class of such linkages, which we call parallel 6R linkages. By definition, they have three pairs of parallel joint-axes for all possible configurations, or at least for infinitely many configurations (it could be that a certain linkage has two components, where only one of them produces three pairs of parallel joint-axes). Two of the pairs of parallel joint-axes are adjacent, and the third one is a pair of opposite joint-axes. We came across this type of linkages when we investigated 6R linkages with coinciding angles being equal, so called angle-symmetric 6R linkages [10]. Also there, there exist three types of angle-symmetric linkages, and one of the three types consists of parallel linkages. But not all parallel

Z. Li (✉) · J. Schicho
Johann Radon Institute for Computational and Applied Mathematics,
Austrian Academy of Sciences (RICAM), 4040 Linz, Austria
e-mail: zijia.li@oeaw.ac.at

J. Schicho
e-mail: josef.schicho@oeaw.ac.at

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_13,

linkages are angle-symmetric in the sense of [10]. A new type can be constructed by taking three arbitrary lines as axes and applying an arbitrary translation to get the other three rotation axes.[1]

This paper also contains the complete classification of parallel linkages. These parallel linkages would fit into [2, Sect. 3.8], a general investigation of 6H linkages; our case is labelled "get to be examined" there.

Our investigation uses Study's description of Euclidean displacements by dual quaternions (see [7, 8]).

The remaining part of the paper is set up as follows. In Sect. 2, we give the theorems for classifying parallel 6R linkages, defining three types. In Sect. 3, we give a construction for each type.

2 Classification

We recall some notations from [8]. The set of all possible motions of a closed 6R linkage is determined by the position of the six rotation axes in some fixed initial configuration. The choice of the initial configuration among all possible configurations is arbitrary.

The algebra $\mathbb{DH}$ of dual quaternions is the 8-dimensional real vector space generated by $1, \varepsilon, \mathbf{i}, \mathbf{j}, \mathbf{k}, \varepsilon\mathbf{i}, \varepsilon\mathbf{j}, \varepsilon\mathbf{k}$ (see [7, 8]). Following [7, 8], we can represent a rotation by a dual quaternion of the form $\left(\cot\left(\frac{\varphi}{2}\right) - h\right)$, where φ is the rotation angle and h is a dual quaternion such that $h^2 = -1$ depending only on the rotation axis. We use projective representations, which means that two dual quaternions represent the same Euclidean displacement if only if one is a real scalar multiple of the other.

Let L be a 6R linkage given by 6 lines, represented by dual quaternions $h_1, \ldots, h_6$ such that $h_i^2 = -1$ for $i = 1, \ldots, 6$. A configuration (see [7, 8]) is a 6-tuple $(t_1, \ldots, t_6)$, such that the closure condition

$$(t_1 - h_1)(t_2 - h_2)(t_3 - h_3)(t_4 - h_4)(t_5 - h_5)(t_6 - h_6) \in \mathbb{R}\backslash\{0\} \quad (1)$$

holds. The configuration parameters t_i—the cotangents of the rotation angles—may be real numbers or ∞, and in the second case we evaluate the expression $(t_i - h_i)$ to 1, the rotation with angle 0. The set of all configurations of L is denoted by K_L. We say L is movable when K_L is a one-dimensional set. Mostly, we will assume, slightly stronger, that there exists an irreducible one-dimensional set for which none of the t_i is fixed. Such a component is called a non-degenerate component. We also exclude the case $\dim_{\mathbb{C}} K_L \geq 2$. Linkages with mobility ≥ 2 do exist, for instance linkages with all axes parallel have mobility 3, but they are well understood.

[1] Just in the last moment, we learned that a special case of this linkage was discovered in *A. Gfrerrer and P.J. Zsombor-Murray*, Robotrac Mobile 6R Closed Chain, Proc. CSME Forum 2002, see also www.geometrie.tugraz.at/lehre/KinematikRobotik/CrankAxlePerspektive.gif.

If $L = [h_1, h_2, h_3, h_4, h_5, h_6]$ is a 6R linkage with mobility 1, then we say that L is a parallel linkage if the axes h_1, h_6 are parallel and the axes h_3, h_4 are parallel, and the non-adjacent axes h_2, h_5 are parallel for infinitely many configurations in K_L. The parallelity conditions in the initial configuration can be expressed as:

$$\begin{aligned} h_1 &= p_1 + \varepsilon q_1, \quad h_2 = p_2 + \varepsilon q_2, \quad h_3 = p_3 + \varepsilon q_3, \\ h_6 &= -p_1 + \varepsilon q_6, \; h_5 = -p_2 + \varepsilon q_5, \; h_4 = -p_3 + \varepsilon q_4, \end{aligned} \tag{2}$$

where p_i are the primal part of h_i and h_{7-i} for $i = 1, 2, 3$, and q_j are the dual part of h_j for $j = 1, \ldots, 6$.

There is a subset of K_L, denoted by K_{qsym}, defined by the additional restrictions $t_1 = t_6, t_2 = t_5, t_3 = t_4$. For all configurations in $\tau \in K_{qsym}$, the transformed lines h_2^τ and h_5^τ are again parallel. Conversely, if $K_0 \subseteq K_L$ is an irreducible component of dimension 1 that contains the initial configuration ∞^6 and that preserves the parallelity of the second and the fifth axis, then $K_0 \subseteq K_{qsym}$.

Remark 2.1 There exist a 6R linkage L with a one dimensional $K_0 \subseteq K_{qsym}$, but L is not a parallel 6R linkage. A possible construction can be found in [7, 8].

Before the following lemma, we recall the definition of *coupling space* and its dimension in [6, 9]. For a sequence $h_i, h_{i+1}, \ldots, h_j$ of consecutive joints, we define the coupling space $L_{i,i+1,\ldots,j}$ as the linear subspace of $\mathbb{R}^8$ generated by all products $h_{k_1} \cdots h_{k_s}, i \leq k_1 < \cdots < k_s \leq j$. (Here, we view dual quaternions as real vectors of dimension eight.) The empty product is allowed, its value is 1. The *coupling dimension* $l_{i,i+1,\ldots,j}$ is the dimension of $L_{i,i+1,\ldots,j}$.

For a parallel 6R linkage L in (2), we make a special transformation as following:

$$h_1' := P_1 h_1 \overline{P_1},\; h_6' := P_1 h_6 \overline{P_1},\; h_3' := P_2 h_3 \overline{P_2},\; h_4' := P_2 h_4 \overline{P_2},$$

where $\overline{P_i}$ denote the conjugations of P_i for $i = 1, 2$, and P_1 and P_2 are translations such that h_1', h_2', h_3' meet in a common point. This is equivalent to the statement that the dimension of coupling space L_{123}' is 4. Furthermore, we have $(t_1 - h_6)(t_1 - h_1) = (t_1 - h_6')(t_1 - h_1')$ and $(t_3 - h_3)(t_3 - h_4) = (t_3 - h_3')(t_3 - h_4')$, and we get the following.

Lemma 2.1 *Parallel 6R linkage L and its transformed linkage L' as above have the same quasi-angle-symmetric configuration space K_{qsym}.*

Three consecutive rotation axes through the same point can be replaced by a spherical joint. The next lemma follows from the classification of $S3R$ linkages.

Lemma 2.2 *For the transformed parallel linkage L', we have $l_{654}' = 4$ or 6.*

If $l_{654}' = 4$, then the lines h_4', h_5', and h_6' also meet in a common point. There is an unique translation P that maps the common point of h_1', h_2, h_3' to the common point of h_4', h_5, h_6'. So, P maps h_1' to h_6', h_2 to h_5, and h_3' to h_4'. But then, P also maps h_1 to h_6 and h_3 to h_4.

Conversely, assume that for six lines $h_1, \dots, h_6$, there exists a translation taking h_1 to h_6, h_2 to h_5, and h_3 to h_6. Then the linkage $L = [h_1, \dots, h_6]$ is mobile.

If $l'_{654} = 6$, then two cases are possible: either L' is a composition of a spherical linkage $[h'_1, h_2, h'_3, h_7]$ and a Bennett linkage $[h'_6, h_5, h'_4, h_7]$, with a suitable line h_7, or L' is a composition of a spherical linkage $[h'_1, h_2, h'_3, h_7, h_8]$ and a Goldberg 5R linkage $[h'_6, h_5, h'_4, h_7, h_8]$, with suitable lines h_7, h_8 passing through the common point of h'_1, h_2, h'_3. In both cases, we get $t_1 = t_3$, so the linkage L'—therefore also L—is angle-symmetric in the sense of [10]. The first case coincides with the "rank 3" case in [10], and the second case is subsumed by the "rank 4" case in [10].

We have sketched the proof of the following theorem.

Theorem 2.1 *If L is a parallel linkage, then it either has the translation property, or four of the rotation angles are equal.*

3 Constructions

All constructions in this section are given in algebraic terms, using dual quaternions. The examples have been produced by an implementation of the constructions in Maple™.

3.1 Translation Property

Here is a construction of parallel 6R linkage with translation property.

Construction 1 *(Parallel 6R Linkage with Translation Property)*

I. *Choose three rotation axes h_1, h_2, h_3, i.e. dual quaternions such that $h_i^2 = -1$.*
II. *Choose a translation $P = 1 + a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$, with a, b, c in the set of real numbers.*
III. *Set $h_4 = -Ph_3\overline{P}$, $h_5 = -Ph_2\overline{P}$ and $h_6 = -Ph_1\overline{P}$.*
IV. *Our parallel 6R Linkage with translation property is $L = [h_1, h_2, h_3, h_4, h_5, h_6]$.*

□

Example 3.1 A random instance of the above construction is

$$h_1 = \left(\frac{7}{9} - \frac{80}{81}\varepsilon\right)\mathbf{i} - \left(\frac{4}{9} + \frac{34}{81}\varepsilon\right)\mathbf{j} + \left(\frac{4}{9} + \frac{106}{81}\varepsilon\right)\mathbf{k},$$
$$h_2 = \left(\frac{3}{5} + \frac{8}{25}\varepsilon\right)\mathbf{i} - \frac{8}{5}\varepsilon\mathbf{j} - \left(\frac{4}{5} - \frac{6}{25}\varepsilon\right)\mathbf{k},$$
$$h_3 = -\left(\frac{1}{3} - \frac{4}{9}\varepsilon\right)\mathbf{i} - \left(\frac{2}{3} + \frac{4}{9}\varepsilon\right)\mathbf{j} - \left(\frac{2}{3} - \frac{2}{9}\varepsilon\right)\mathbf{k},$$
$$P = 1 - \frac{16}{27}\varepsilon\mathbf{i} - \frac{20}{27}\varepsilon\mathbf{j} + \frac{8}{27}\varepsilon\mathbf{k},$$

$$h_4 = \left(\frac{1}{3} - \frac{148}{81}\varepsilon\right)\mathbf{i} + \left(\frac{2}{3} + \frac{116}{81}\varepsilon\right)\mathbf{j} + \left(\frac{2}{3} - \frac{14}{27}\varepsilon\right)\mathbf{k},$$
$$h_5 = -\left(\frac{3}{5} + \frac{1016}{675}\varepsilon\right)\mathbf{i} + \frac{296}{135}\varepsilon\mathbf{j} + \left(\frac{4}{5} - \frac{254}{225}\varepsilon\right)\mathbf{k},$$
$$h_6 = -\left(\frac{7}{9} - \frac{112}{81}\varepsilon\right)\mathbf{i} + \left(\frac{4}{9} - \frac{46}{81}\varepsilon\right)\mathbf{j} - \left(\frac{4}{9} + \frac{242}{81}\varepsilon\right)\mathbf{k}.$$

Its configuration curve is irreducible of genus 1. Its equations are:

$$-21t_1^2 + 9t_1^2t_2 + 25t_2^2t_1 + 6t_1t_2 - 9t_1 + 6 - 9t_2 - 15t_2^2 = 0,$$
$$-21 + 63t_1 + 5t_2 - 27t_1t_2 - 6t_3 + 72t_3t_2 = 0.$$

Here are the Denavit-Hartenberg parameters [3] of the above linkage. These are the orthogonal distance between two adjacent joint axes a_{ij}, the distance d_i between the two footpoints of the two neighboring axes on the ith axis, and the twist angle between two adjacent joint axes α_{ij}, for $i = 1, \dots, 6$ and $j = i + 1$ (modulo 6). For any parallel linkage with translation property, the parameters fulfill the conditions

$$a_{12} = a_{56},\ a_{23} = a_{45},$$
$$d_1 = d_4 = 0,\ d_2 = d_5,\ d_3^2 + a_{34}^2 = d_6^2 + a_{61}^2,$$
$$\alpha_{34} = \alpha_{61} = 0,\ \alpha_{23} = \alpha_{45},\ \alpha_{56} = \alpha_{12}.$$

In the example, the values are

$$a_{12} = a_{56} = \frac{58\sqrt{5}}{225},\ a_{23} = a_{45} = \frac{2\sqrt{2}}{3},\ a_{34} = \frac{8\sqrt{305}}{81},\ a_{61} = \frac{8\sqrt{5}}{9},$$
$$\alpha_{34} = \alpha_{61} = 0,\ \alpha_{23} = \alpha_{45} = \arccos\left(\frac{1}{3}\right),\ \alpha_{56} = \alpha_{12} = \arccos\left(\frac{1}{9}\right),$$
$$d_1 = d_4 = 0,\ d_2 = d_5 = \frac{11}{25},\ d_3 = \frac{80}{81},\ d_6 = 0.$$

3.2 Parallel 6R Linkage with Angle-Symmetric Property

There are two constructions, corresponding to the two sub cases of angle-symmetric parallel linkages. The first appeared in [10] gives Parallel 6R Linkage with angle-symmetric property (type 1). Here is the second construction.

Construction 2 *(Parallel 6R Linkage with angle-symmetric property, type 2)*

I. *Choose two rotation axes h_1 and h_2, i.e. dual quaternions such that $h_1^2 = h_2^2 = -1$.*

II. *Choose another rotation axis h_6 parallel to h_1; the primal part of h_6 should be the primal part of h_1 times -1.*

III. *Compute two rotation axes m_1 and m_2 such that h_1, h_2, m_1, m_2 form a Bennett 4R linkage. One way to do this is to use the factorization algorithm for motion polynomials [8].*

IV. *Compute two rotation axes m_3 and h_5 such that h_6, m_2, m_3, h_5 form a Bennett 4R linkage, and such that the configuration curve is equal to the one in step III. Again, this can be done by factorizing a motion polynomial.*

V. *Choose a translation $P = 1 + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, where b, c, d are real numbers.*

VI. *Set $h_3 = -Pm_1\overline{P}$, $h_4 = -Pm_3\overline{P}$.*

VII. *Our parallel 6R Linkage is $L = [h_1, h_2, h_3, h_4, h_5, h_6]$.* □

Example 3.2 A random instance of the above construction is

$$
\begin{aligned}
h_1 &= \left(\frac{1}{3} - \frac{4}{9}\varepsilon\right)\mathbf{i} - \left(\frac{2}{3} - \frac{2}{9}\varepsilon\right)\mathbf{j} + \left(\frac{2}{3} + \frac{4}{9}\varepsilon\right)\mathbf{k},\\
h_2 &= -\left(\frac{1}{3} + \frac{8}{9}\varepsilon\right)\mathbf{i} - \left(\frac{2}{3} - \frac{8}{9}\varepsilon\right)\mathbf{j} + \left(\frac{2}{3} + \frac{4}{9}\varepsilon\right)\mathbf{k},\\
h_6 &= -\frac{1}{3}\mathbf{i} + \frac{2}{3}\mathbf{j} - \frac{2}{3}\mathbf{k},\\
a &= \frac{1}{2},\\
m_1 &= \left(\frac{119}{411} + \frac{124340}{168921}\varepsilon\right)\mathbf{i} + \left(\frac{226}{411} - \frac{172130}{168921}\varepsilon\right)\mathbf{j} - \left(\frac{322}{411} + \frac{74860}{168921}\varepsilon\right)\mathbf{k},\\
m_2 &= -\left(\frac{119}{411} - \frac{100888}{168921}\varepsilon\right)\mathbf{i} + \left(\frac{322}{411} - \frac{15560}{168921}\varepsilon\right)\mathbf{j} - \left(\frac{226}{411} + \frac{75292}{168921}\varepsilon\right)\mathbf{k},\\
m_3 &= \left(\frac{11601824}{8614971}\varepsilon - \frac{119}{411}\right)\mathbf{i} - \left(\frac{226}{411} - \frac{13771184}{8614971}\varepsilon\right)\mathbf{j} + \left(\frac{322}{411} + \frac{4651040}{2871657}\varepsilon\right)\mathbf{k},\\
h_5 &= \left(\frac{1}{3} - \frac{344}{459}\varepsilon\right)\mathbf{i} + \left(\frac{2}{3} - \frac{776}{459}\varepsilon\right)\mathbf{j} - \left(\frac{2}{3} + \frac{316}{153}\varepsilon\right)\mathbf{k},\\
P &= 1 - \frac{2}{3}\varepsilon\mathbf{i} - \frac{1}{2}\varepsilon\mathbf{j} + \varepsilon\mathbf{k},\\
h_3 &= \left(\frac{119}{411} + \frac{177770}{168921}\varepsilon\right)\mathbf{i} + \left(\frac{226}{411} - \frac{10388}{18769}\varepsilon\right)\mathbf{j} - \left(\frac{322}{411} - \frac{79}{168921}\varepsilon\right)\mathbf{k},\\
h_4 &= -\left(\frac{119}{411} - \frac{8876894}{8614971}\varepsilon\right)\mathbf{i} - \left(\frac{226}{411} - \frac{9760646}{8614971}\varepsilon\right)\mathbf{j} + \left(\frac{322}{411} + \frac{3377077}{2871657}\varepsilon\right)\mathbf{k}.
\end{aligned}
$$

Here we found that the configuration curve is reducible. It has one non-degenerate component in K_{qsym}, with rational parametrization:

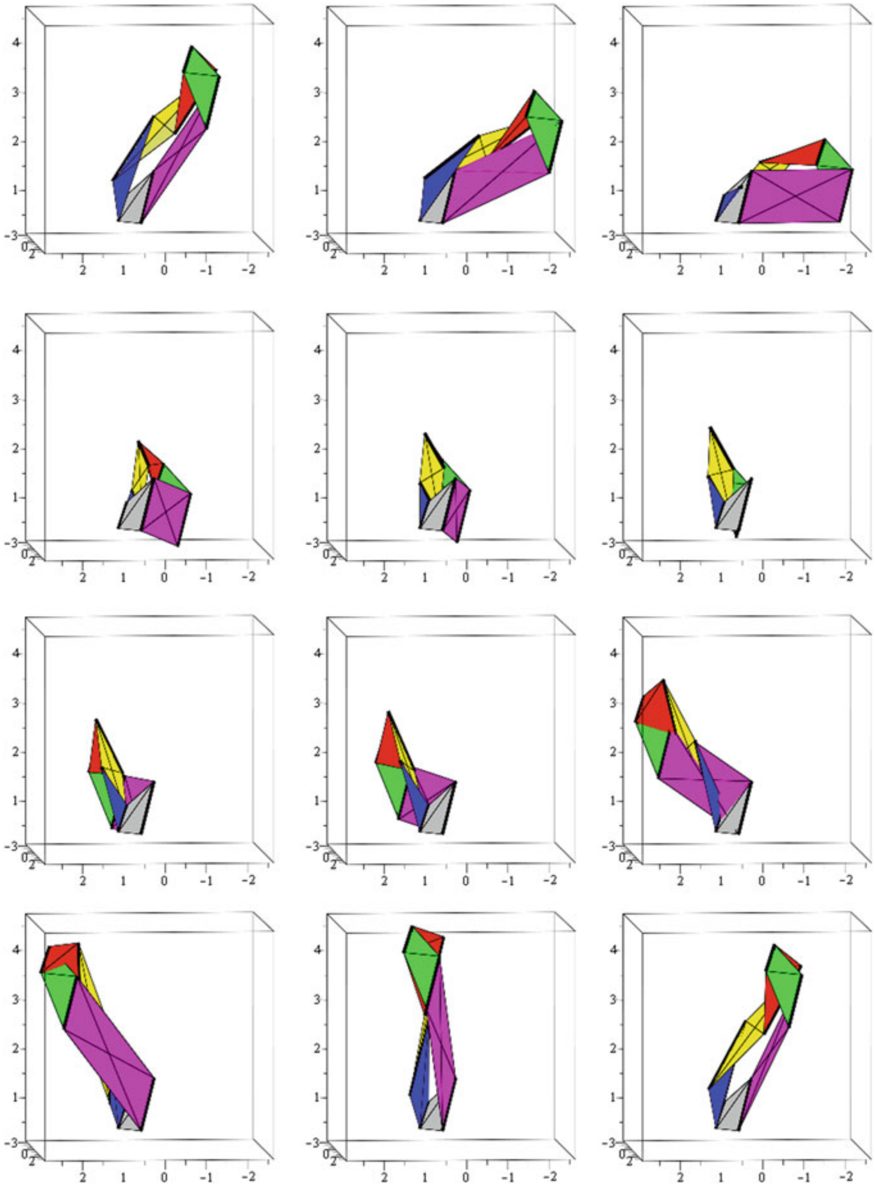


Fig. 1 A parallel angle-symmetric linkage of type 2 (described in Example 3.2). The four colored tetrahedra and the two colored parallelograms represent the six links, and the joints are the common edges of connected tetrahedra/parallelograms. Possible collisions of the links are just shown as overlapping links

$$(t_1, t_2, t_3) = (t, t+1, t)\,.$$

In Fig. 1, we present twelve configuration positions of this linkage produced by Maple. □

Here are the numeric values of the Denavit-Hartenberg parameters.

$$a_{61} = \frac{2}{3},\ a_{12} = \frac{\sqrt{2}}{3},\ a_{23} = \frac{4151\sqrt{34}}{41922},\ a_{34} = \frac{274\sqrt{17}}{459},\ a_{45} = \frac{6617\sqrt{34}}{41992},\ a_{56} = \frac{86\sqrt{2}}{153},$$

$$\alpha_{34} = \alpha_{61} = 0,\ \alpha_{23} = \alpha_{45} = \arccos\left(\frac{135}{137}\right),\ \alpha_{56} = \alpha_{12} = \arccos\left(\frac{7}{9}\right),$$

$$d_1 = d_4 = 0,\ d_2 = d_5 = \frac{923}{1224},\ d_3 = \frac{4795}{1836},\ d_6 = \frac{225}{68}.$$

We do not know the general conditions of the Denavit-Hartenberg parameters of a linkage obtained by the construction.

Acknowledgments The research was supported by the Austrian Science Fund (FWF): W1214-N15, project DK9.

References

1. Baker, J.E.: An analysis of the Bricard linkages. Mech. Mach. Theory **15**(4), 267–286 (1980)
2. Baker, J.E.: Overconstrained six-bars with parallel adjacent joint-axes. Mech. Mach. Theory **38**(2), 103–117 (2003)
3. Denavit, J., Hartenberg, R.S.: A kinematic notation for lower-pair mechanisms based on matrices. Trans. ASME J. Appl. Mech **23**, 215–221 (1955)
4. Dietmaier, P.: Einfach übergeschlossene Mechanismen mit Drehgelenken. Habilitation thesis, Graz University of Technology (1995)
5. Goldberg, M.: New five-bar and six-bar linkages in three dimensions. Trans. ASME **65**, 649–656 (1943)
6. Hegedüs, G., Schicho, J., Schröcker, H.P.: Bond theory and closed 5R linkages. In: Lenarčič, J., Husty, M. (eds.) Latest Advances in Robot Kinematics, pp. 221–228. Springer, Dordrecht (2012)
7. Hegedüs, G., Schicho, J., Schröcker, H.P.: Construction of overconstrained linkages by factorization of rational motions. In: Lenarčič, J., Husty, M. (eds.) Latest Advances in Robot Kinematics, pp. 213–220. Springer, Dordrecht (2012)
8. Hegedüs, G., Schicho, J., Schröcker H.P.: Factorization of rational curves in the study quadric and revolute linkages. ArXiv e-prints (2012)
9. Hegedüs, G., Schicho, J., Schröcker, H.P.: The theory of bonds: a new method for the analysis of linkages. ArXiv e-prints (2012)
10. Li, Z., Schicho, J.: Classification of angle-symmetric 6R linkages. ArXiv e-prints (2013)

11. Sarrus, P.: Note sur la transformation des mouvements rectilignes alternatifs, en mouvements circulaires: et rèciproquement. Comptes Rendus des Séances de l'Académie des Sciences de Paris **36**, 1036–1038 (1853)
12. Waldron, K.J.: Overconstrained linkages. Environ. Plan. B-Plan. Des. **6**, 393–402 (1979)
13. Wohlhart, K.: Merging two general Goldberg 5R linkages to obtain a new 6R space mechanism. Mech. Mach. Theory **26**, 659–668 (1991)

Positioning Two Redundant Arms for Cooperative Manipulation of Objects

Adrià Colomé and Carme Torras

Abstract Bimanual manipulation of objects is receiving a lot of attention nowadays, but there is few literature addressing the design of the arms configuration. In this paper, we propose a way to analyze the relative positioning of two redundant arms, both equipped with spherical wrists, in order to obtain the best common workspace for grasping purposes. Considering the geometry of a robot with a spherical wrist, the Cartesian workspace can be discretized, with an easy representation of the feasible end-effector orientations at each point using bounding cones. After having characterized the workspace for one robot arm, we can evaluate how good each of the discretized poses relate with an identical arm in another position with a quality function that considers orientations. In the end, we obtain a quality value for each relative position of two arms, and we perform an optimization using genetic algorithms to obtain the best workspace for a cooperative task.

Keywords Robot design · Workspace · Bimanual manipulation

1 Introduction

Bimanual manipulationaut]Colomé, A. aut]Torras, C.allows robots to perform more complex tasks than a single-limb robot [1]. However, while a lot of attention is focused on how to manipulate or plan a task, less importance is given to the arms configuration. Usually, a humanoid-like configuration is chosen, to make the robot more human-friendly [2], but when deciding how to use two Barrett WAM (Whole Arm Manipulator) robots, we questioned ourselves if a humanoid-like configuration would be the best, for example, for folding clothes. A first step towards this aim

A. Colomé (✉) · C. Torras
Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
e-mail: acolome@iri.upc.edu

C. Torras
e-mail: torras@iri.upc.edu

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_14,

is to analyze the robot's workspace. In [3], a discretized workspace is used with information about the probability of solving the Inverse Kinematics (IK) with random orientations at each cartesian position, and also manipulability data, an indicator of dexterity and distance to a singularity [4], in order to decide the grasping points for bimanual manipulation. However, this work exploits an existing humanoid robot, which may not have been specifically designed for the task being tackled. Zacharias et al. [5–7] plot the $3D$ cartesian position workspace by initially drawing spheres, whose color varies with the percentage of inverse kinematics solutions found for each point. Moreover, they propose to use different shapes at each point to represent orientations, depending on the feasible end-effector orientations at each position, but their later work also focuses on optimizing manipulation with a given bimanual robot, rather than deciding its arms configuration.

We think that the relative position of the arms may be improved in order to get better grasps. In fact, the WAM robot's workspace is much different from that of a human arm, and so are those of many other commercial arms, thus we investigated a way of deciding the arms relative position, depending on the task, by fully characterizing the workspace of the Barrett WAM arm. Using the fact that the robot has a spherical wrist, we propose to compute its feasible orientations for each cartesian point and pack them in a bounding cone to obtain an easy characterization of robot feasible poses. With these cones, we can evaluate how good two points relate when grasping an object of a certain length. If both robots can reach the position, but their valid end-effectors' orientations for those points are not good enough, the object may not be well manipulated. In Sect. 2 we will explain how we characterize the workspace with information on all feasible orientations and how we store these data. This is later used in Sect. 3 to evaluate a relative positioning of two identical arms. Finally, in Sect. 4 we describe the implementation, and we use a genetic algorithm to search for the best relative positioning of the two arms.

2 Workspace Representation

For a redundant robot, it is well known that the Forward Kinematics (FK) function f is not one-to-one, and given its non-linearity, the workspace may be hard to represent. The Inverse Kinematics (IK), the inverse of the FK, may then be better to characterize the workspace. Note that, for each point in the cartesian space, more than one IK solution may exist [8]. We initially decided to characterize the workspace numerically, as a subset of $\mathbb{R}^3 \times SO(3)$, by discretizing it. To do so, a uniform mesh is set for the cartesian position and/or orientation, and, for each point P_i on the mesh, the existence of a joint solution θ^{P_i} such that $P_i = f(\theta^{P_i})$ is checked. This can be done by sampling the joint space and using the FK function (forward sampling), or by sampling the workspace and using the IK (inverse sampling). Nevertheless, while the forward sampling results in a biased sampling of the workspace, the inverse is able to exhaustively analyze the whole workspace, thus we recommend this option

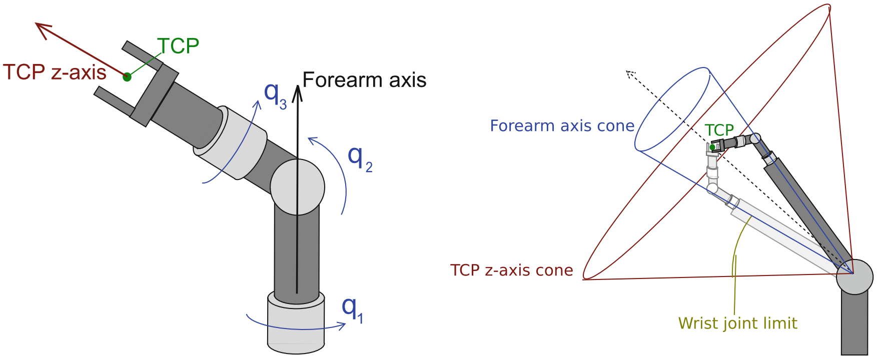


Fig. 1 *Left:* Spherical wrist. θ_2 is the wrist angle. *Right:* Robot scheme showing the bounding cone of all possible forearm axes. This cone is augmented by adding the wrist joint limit to obtain the possible TCP z-axis bounding cone

if a good IK algorithm is available (for the case of the WAM robot, the IK can be obtained either by iterative methods [8] or analytical methods [9, 10]).

To plot the reachable positions of the workspace, and store its data, we used a similar method as in [7]. For each point of the $3D$ mesh representing the workspace, M solutions of the IK of the robot, with different orientations, are obtained. To ensure a good distribution of these orientations, we can use the proposal in [12], where points are arranged in hexagonal patterns to fit on the sphere, or use randomly generated quaternions. If there exists at least one solution, the position is reachable. In addition, for each of these M IK attempts, we can extract additional information, such as manipulability [4] at the obtained pose, percentage of orientations found for a given $3D$ cartesian point, etc.

For the feasible orientations of a robot arm in a cartesian point, several geometrical shapes to represent the valid orientations have been proposed in literature [7]. Among these shapes, cones are probably the best choice, due to their simplicity and easy characterization. In fact, for a robot with a spherical wrist (see Fig. 1a), the Tool Center Point (TCP) stays within a cone whose axis is the rotation axis of the first degree of freedom of the wrist (namely, the forearm axis).

Moreover, discarding the rotation around the TCP z-axis, we propose to collect the set of valid forearm axes at a certain cartesian point $P \in \mathbb{R}^3$, which will be enclosed by a cone, and compute the Bounding Cone (BC) that contains them all with the algoritm proposed in [11]. Also, if the wrist angle has symmetric limits, its aperture can be added to the BC angle, yielding a cone that contains all the TCP z-orientation axis that the robot can reach at the given position (see Fig. 1b).

With this approach, we obtain a mesh for the workspace, encoding all the information gathered when computing the reachable positions such as manipulability, percentage of solutions found, etc. plus the obtained bounding cone containing all the possible z-axis of the TCP. We can see an IK solutions map over the workspace of a WAM robot (see Table 1 in [10] for its dimensional parameters) in Fig. 2.

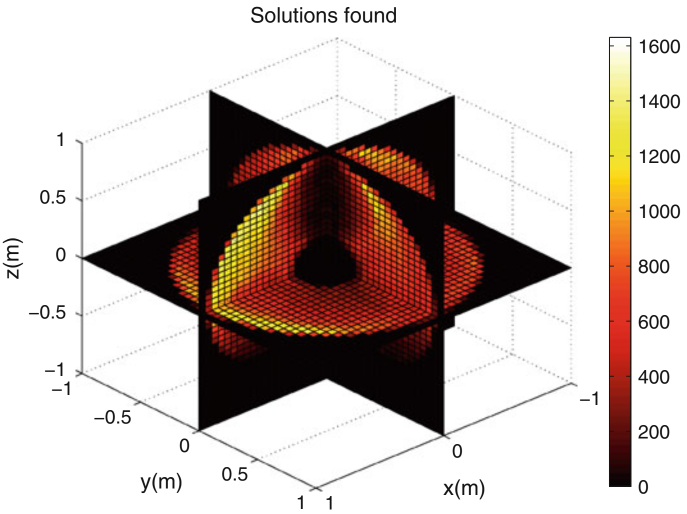


Fig. 2 Solutions found over the workspace of a WAM robot

3 Bimanual Workspace

Multiple-arm cooperative tasks provide the capability of performing tasks that would be impossible or, at least, much more difficult to accomplish with only one arm. Although actuating the arms to simultaneously move an object may be a hard task, the arms relative configuration must be given importance, since depending on the intended use of bimanual robots, some configurations might be better suited than others. Human arms configuration may be the best for the tasks performed by humans along their evolution, with a large workspace in front, and a very reduced workspace at our back, as our attention and visible space stays in front of us.

In [1], a review of bimanual manipulation is done, where the state-of-the-art in cooperative tasks is analyzed. Some examples of existing bimanual robots are shown, such as the *Justin robot* [6], where the arms are placed in a humanoid-like configuration with a tilt of 60 degrees, the DARPA arm robot [13], with two Barrett WAM robot arms with their bases placed perpendicular, or the ARMAR [14, 15] robot, where both z base axes are placed in an aligned humanoid-like configuration.

However, the humanoid configuration may not be the best one for certain tasks. In this section, we provide hints to determine how good is a relative positioning of robot arms, in the form of a parametrized value function to be used with a numerically characterized workspace as that in the previous section.

In this work, we intend to characterize a common workspace between two arms. To this purpose, given two cartesian points $P_1 \in W_1$, $P_2 \in W_2$, we will compute several factors that will lead to a quality function for each pair (P_1, P_2) defined as:.

$$F(P_1, P_2) = DF \cdot SDF \cdot OF \cdot MF \cdot CF,$$

where DF is the Distance Factor, SDF the Solutions Density Factor, OF the Orientation Factor, MF the Manipulability Factor and CF the Conditioning Factor. Multiplication and not addition of factors has been chosen to strongly penalize those positions with very low value on one factor.

Then, the global quality value of a relative position of two arms is:

$$F = V \cdot \sum_{P_1 \in W_1} \sum_{P_2 \in W_2} F(P_1, P_2), \tag{1}$$

V being the total volume of the combined workspace. Evaluating this quality measure (1) we get a mapping $g : \mathbb{R}^3 \times SO(3) \rightarrow \mathbb{R}$, which maps a relative position plus orientation transformation (up to 6 variables) to a real value. This mapping can be used by genetic algorithms to search for its maximum, which would correspond to the best relative positioning.

For each pair of points (P_1, P_2), we have to decide whether to evaluate their relation or not. In order to decide that, we may take a characteristic length L for the object to be manipulated, and then one possible way to evaluate that relation is by using the following Distance Factor (DF):

$$DF = \begin{cases} 1 \text{ if } L - \delta < \|P_2 - P_1\| < L + \delta, |a_1| < \alpha_1, \text{and } |a_2| < \alpha_2 \\ 0 \text{ otherwise} \end{cases}, \tag{2}$$

where δ is a tolerance on the manipulated object length, and $a_1, a_2, \alpha_1, \alpha_2$ are defined with the bounding cones in Fig. 3. If the segment joining P_1 and P_2 does not lie within both orientation cones for P_1 and P_2, their relation may not be evaluated. However, the orientation restriction can be made more permissive, depending on the kind of graspings to perform.

As defined in some previous works [3, 7], the SDF is the ratio of the IK solutions found over the attempted solutions. For each cartesian point of the workspace, we retain the percentage of IK solutions found, given random orientations. The SDF is then defined as the product of the ratios for the two points compared.

Imagine two arms manipulating an object of length L grasped at points $P_1 \in W_1$ and $P_2 \in W_2$. In this situation, a grasp in which the TCP z-axis of each arm is aligned in the direction of the other grasping point is usually preferred. This can be checked using the cones (Z_1, α_1), (Z_2, α_2) obtained for each workspace: we can calculate the angles a_1, a_2 from the vector $P_1 - P_2$ and the cones axes Z_1, Z_2, as in Fig. 3.

Then, we define the Orientation Factor (OF) for the points to compare as

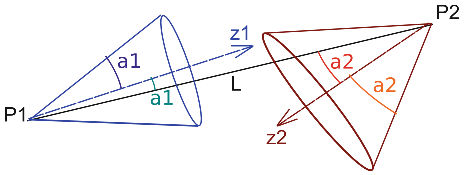


Fig. 3 Distance and Orientation factor variables: α_1, α_2 are the cones angles, and a_1, a_2 the angles between each cone's axis and the line P_1P_2

$$OF = max\left(OF_{min}, 1 + \frac{1}{K}ln\left(\frac{\alpha_1 - |a_1|}{\alpha_1} \cdot \frac{\alpha_2 - |a_2|}{\alpha_2}\right)\right) \tag{3}$$

where K is a tuning parameter, and OF_{min} is the minimum value accepted for the orientation factor.

Thus defined, this factor verifies that $OF_{min} \leq OF \leq 1, \forall a_1, a_2, \alpha_1, \alpha_2$, when satisfying the conditions in (2), it having a value of 1 when both cones' axes are parallel to the vector $P_2 - P_1$ and pointing towards each other, and gradually reducing its value to OF_{min} when the axes point away from each other.

When performing cooperative tasks, or grasping an object with multiple arms, there are approaches to obtain a combined manipulability [16, 17]. However, combined manipulability computation for multiple arms holding an object relies on the arms poses, which are unknown for our workspace representation, as we use the average of many IK computations with different orientations and the redundancy of the robot gives us infinite solutions. So we take the average manipulability of both grasping points as a good approach to evaluate how manipulable is an object. In [3], grasping point candidates are selected based on this manipulability, so the MF is defined as:

$$MF(P_1, P_2) = \overline{m}(P_1)\overline{m}(P_2)\text{ , with } \overline{m}(P_i) = \frac{1}{M}\sum_{j=1}^{M} m(IK(P_i, o_j)), \tag{4}$$

$m(IK(P_i, o_j))$ being the manipulability at the joint position obtained as an IK solution for the robot at position P_i with orientation o_j.

In order to ensure a stable behaviour for the related points, we use the Jacobian Condition Number (CN), which is defined as $\kappa(J) = \sigma_1/\sigma_n$, where σ_1, σ_n are the largest and the smallest singular values of the robot Jacobian matrix. The CN is a measure of the error amplification induced by the Jacobian matrix. Thus, we define

$$CF = \kappa(P_1) \cdot \kappa(P_2), \tag{5}$$

where $\kappa(P_i)$ is the average CN for the solved IK of P_i.

4 Experimentation

As a first application, we searched for optimal relative positionings of two WAM robots. To do so, we used genetic algorithms instead of performing an exhaustive analysis in order to obtain the results faster, using 10 generations of 20 elements each, and a probability of mutation on each variable slightly decreasing after each generation.

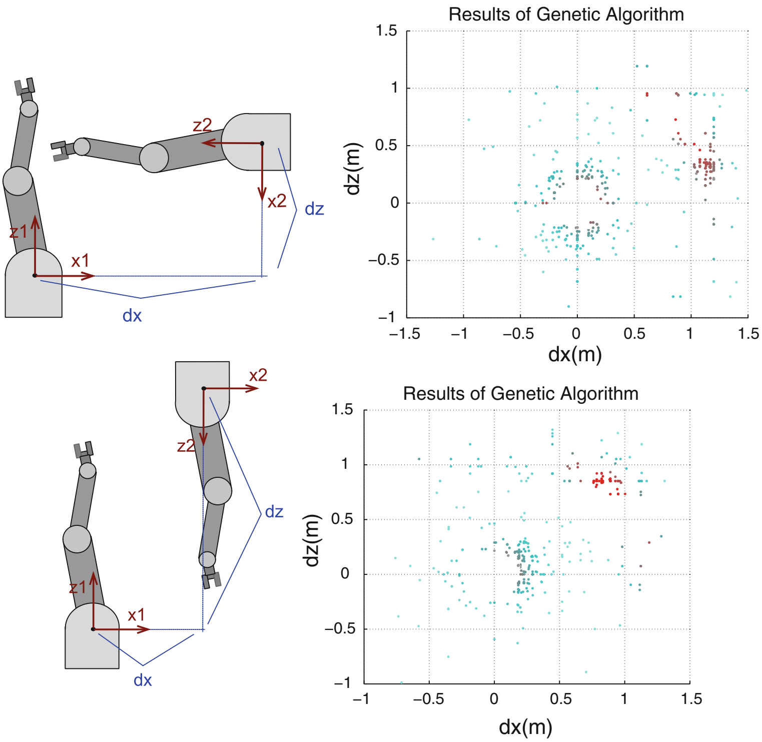


Fig. 4 *Left:* Two experimental settings. *Right:* Results with quality values from *blue* (low value) to *red* (high value)

We considered valid objects for grasping those of a size between 0.3 m and 0.5 m for the DF and a $K = 2$ for the OF. We collected the best half at each generation, paired them, and created 20 new configurations for the next generation.

Several settings were considered for optimizing a two-dimensional relative position between the arms. The first one is similar to that of the DARPA robot, with both z axes perpendicular. We found (see Fig. 4 top) that, for our criterion, the best configurations are those with positive dx and dz, while the DARPA robot has both negative offsets, which yield a lower value of the quality function. The best solution is for $dx = 0.8$ m, $dz = 0.8$ m in which both arms cooperate at a larger distance than the DARPA robot. Other experimentation such as placing both arms with their z-axes parallel and facing each other (see Fig. 4 bottom) also lead to good positionings which might not have been considered when building bimanual robots.

5 Conclusions

In this work, we have presented a novel way to store the workspace information (including orientation, which is often not considered) of a robot with a spherical wrist, in a very compact way, thanks to the efficient bounding cones representation of the end-effector z-axis. This then allows us to evaluate the capability of a dual arm robot to manipulate an object of a certain size, depending on the relative position of both arms. We can compute a global quality measure for a given relative position, in order to quantify how good a dual arm configuration is.

We used this quality measure to obtain better relative configurations with the help of a genetic algorithm. And the results in Sect. 4 seem to indicate that, for the tasks studied, the configuration of current bimanual robots may not be optimal.

The arm configurations of humanoid robots are designed to accomplish a wide repertoire of tasks while obeying diverse design and operational constraints. However, in settings with two independent robot arms, it may be simple and advantageous to tailor their relative configuration to the specificities of each particular task, as shown in the current work. The proposed algorithm can be further used to optimize relative positionings with more than two parameters in order to get more general results.

Acknowledgments This work is partially supported by EU Project IntellAct (FP7-269959), by the CSIC Project CINNOVA (201150E088), and by the Catalan Research Commission through SGR-00155.

References

1. Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D.V., Kragic, D.: Dual arm manipulation—A survey. Rob. Auton. Syst. **60**, 1340–1353 (2012)
2. Hyon, S.-H., Hale, J.G., Cheng, G.: Full-body compliant human-humanoid interaction: balancing in the presence of unknown external forces. IEEE Trans. Rob. **23**(5), 884–898 (2007)
3. Vahrenkamp, N., Przybylski, M., Asfour, T., Dillmann, R.: Bimanual grasp planning. IEEE-RAS Int. Conf. Humanoid Rob. **4**, 493–499 (2011)
4. Yoshikawa, T.: Dynamic manipulability of robot manipulators. IEEE ICRA **2**, 1033–1038 (1985)
5. Zacharias, F., Leidner, D., Schmidt, F., Borst, C., Hirzinger, G.: Exploiting structure in two-armed manipulation tasks for humanoid robots. IEEE/RSJ IROS 5446–5452 (2010)
6. Ott, Ch., Eiberger, O., Friedl, W., Bauml, B., Hillenbrand, U., Borst, ch., Albu-Schaffer, A., Brunner, B., Hirschmoller, H., Kielhofer, S., Konietschke, R., Suppa, M., Wimbock, T., Zacharias, F., Hirzinger. G.: A humanoid two-arm system for dexterous manipulation. IEEE RAS Conf. Humanoid Robots 276–283 (2006)
7. Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure: Representing robot capabilities. IEEE/RSJ IROS 3229–3236 (2007)
8. Colomé, A., Torras, C.: Redundant inverse kinematics: experimental comparative review and two enhancements. IEEE/RSJ IROS 5333–5340 (2012)
9. Shimizu, M., Kakuya, H., Yoon, W.-K., Kitagaki, K., Kosuge, K.: Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution. IEEE Trans. Rob. **24**(5) 1131–1142 (2008)

10. Singh, G.K., Claassens, J.: An analytical solution for the inverse kinematics of a redundant 7-dof manipulator with link offsets. IEEE/RSJ IROS, 2976–2982 (2010)
11. Barequet, G., Elber, G.: Optimal bounding cones of vectors in three dimensions. Inf. Process. Lett. **93**(2), 83–89 (2005)
12. Saff, E., Kuijlaars, A.: Distributing many points on the sphere. Math. Intelligencer **19**(1), 5–11 (1997)
13. Guizzo, E.: DARPA Seeking to revolutionize robotic manipulation. IEEE Spectr. Technol. Sci. News (2010) http://spectrum.ieee.org/automaton/robotics/robotics-software/darpa-arm-program
14. Asfour, T., Berns, K., Dillmann, R.: The humanoid robot ARMAR: Design and control. IEEE/APS Int. Conf. Humanoid Robots 7–8 (2000)
15. Regenstein, K., Azad, P., Schroder, J., Bierbaum, A., Vahrenkamp, N., Dillmann, R.: ARMAR-III: An integrated humanoid platform for sensory-motor control. IEEE/RAS Int. Conf. Humanoid Robots 169–175 (2006)
16. Chiacchio, P., Chiaverini, S., Sciavicco, L., Siciliano, B.:Global task space manipulability ellipsoids for multiple-arm systems. IEEE Trans. Rob. Auton. **7**, 5 (1991)
17. Bicchi, A., Prattichizzo, D.: Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms. IEEE Trans. Rob. Auton. **16**, 4 (2000)

A Sufficient Condition for Parameter Identifiability in Robotic Calibration

Thibault Gayral and David Daney

Abstract Calibration aims at identifying the model parameters of a robot through experimental measures. In this paper, necessary mathematical conditions for calibration are developed, considering the desired accuracy, the sensor inaccuracy of the joint coordinates, and the measurement noise. They enable to define a physically meaningful stop criterion for the identification algorithm and a numerical bound for the observability index O_3, the minimum singular value of the observability matrix. With this bound, observability problems can be safely detected during calibration. Those conditions for calibration are illustrated through a simple example.

Keywords Conditions for calibration · Observability · Least-squares

1 Introduction

Because of manufacturing and assembly errors, kinematic parameters of a robot are only known with uncertainties. In order to reach the desired accuracy of the robot over its workspace, a better knowledge of the model parameters is needed: this is the goal of calibration. Calibration can be decomposed in four important parts: modeling, measurements, identification and implementation [8, 11].

The required qualities of a robot model for calibration are well-known and can be found in [3]. However, necessary conditions also exist for the measurement and the identification steps. Indeed, some common assumptions have to be made: for example that the identification function mostly depends on the variations of the model parameters. Such hypotheses imply conditions both on the measurement device accuracy and on the measurement workspace. However, to the best of our knowledge,

T. Gayral (✉) · D. Daney
INRIA Sophia Antipolis, Sophia-Antipolis cedex, France
e-mail: Thibault.Gayral@inria.fr

D. Daney
e-mail: David.Daney@inria.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_15,

those conditions are not treated in the literature. Deriving those necessary conditions will lead to the calculus of the necessary accuracy of the kinematic parameters and the maximal allowed measurement inaccuracy for calibration. With these values, a numerical scaling of the observability matrix [8] involved in the identification step can be obtained. This enables to define a numerical bound for the observability index O_3 [9], the minimum singular value of the observability matrix. With this new bound, observability problems can be detected during calibration.

2 Overview and Conditions of Calibration

Without loss of generality, we will focus only on the kinematic calibration. Let consider a robot, with either a serial or parallel kinematic structure, with n' actuators controlling n degrees of freedom. The exact pose p of the end-effector is described by the n-vector $\mathbf{x}_p^*$, and $\boldsymbol{\rho}_p^*$ stands for the n'-vector of the exact joint coordinates at pose p. The kinematic model of this robot relates the joint and end-effector coordinates $\boldsymbol{\rho}_p^*$ and $\mathbf{x}_p^*$ through a function $\mathbf{f}^*$ that depends upon the m irreducible exact kinematic parameters [3] described by the vector $\boldsymbol{\xi}^*$:

$$\mathbf{f}^*(\mathbf{x}_p^*, \boldsymbol{\rho}_p^*, \boldsymbol{\xi}^*) = \mathbf{0} \tag{1}$$

However, both the joint coordinates and the kinematic parameters are not known with exactitude because of the accuracy of the actuator sensors, and manufacturing and assembly errors. Let consider measurement errors $\delta\boldsymbol{\rho}_p^*$ and $\delta\boldsymbol{\xi}^*$ between the nominal values $\boldsymbol{\rho}_p$ and $\boldsymbol{\xi}$ and the exact values $\boldsymbol{\rho}_p^*$ and $\boldsymbol{\xi}^*$ such as $\boldsymbol{\rho}_p = \boldsymbol{\rho}_p^* + \delta\boldsymbol{\rho}_p^*$ and $\boldsymbol{\xi} = \boldsymbol{\xi}^* + \delta\boldsymbol{\xi}^*$. These errors lead to a position error $\delta\mathbf{x}_p^*$ of the end-effector. The position error can be estimated by differentiating $\mathbf{f}^*(\mathbf{x}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi})$ with respect to all its parameters with the assumption of no singularity in the robot workspace [6]. Moreover, it is often complicated to obtain the exact model of a robot $\mathbf{f}^*$. Thus the modeling contains also errors $\boldsymbol{\varepsilon}_m$ that can be considered in the position error of the end-effector:

$$\delta\mathbf{x}_p^* = \mathbf{J}_\rho^p(\mathbf{x}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi})\delta\boldsymbol{\rho}_p^* + \mathbf{J}_\xi^p(\mathbf{x}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi})\delta\boldsymbol{\xi}^* \;\; (+\boldsymbol{\varepsilon}_m) \tag{2}$$

where $\mathbf{J}_\rho^p$ is the kinematic Jacobian matrix, and $\mathbf{J}_\xi^p$ is the identification Jacobian matrix of pose p [8]. Note that both $\mathbf{J}_\rho^p$ and $\mathbf{J}_\xi^p$ are functions of all the parameters. For clarity reasons, this dependency will be omitted in the following equations. From (2) can be derived condition **(C1)**:

(C1) Necessary accuracy of the kinematic parameters: Assuming a given desired accuracy $\Delta\mathbf{x}_f$ and accuracy of the joint coordinate sensor $\Delta\boldsymbol{\rho}$, then the necessary accuracy $\delta\boldsymbol{\xi}^{\text{nec}}$ of the kinematic parameters can be derived from (2) under the assumption $\Delta\mathbf{x}_f \gg \boldsymbol{\varepsilon}_m$:

$$\forall \mathbf{x}_p \in \mathcal{W}, \quad \text{abs}(\mathbf{J}_\xi^p)\delta\boldsymbol{\xi}^{\text{nec}} \leq \Delta\mathbf{x}_f - \text{abs}(\mathbf{J}_\rho^p)\Delta\boldsymbol{\rho} \tag{3}$$

where abs(•) stands for a matrix whose terms are the absolute values of the considered matrix, and $\mathcal{W}$ refers to the robot workspace. Note that the assumption $\Delta\mathbf{x}_f \gg \boldsymbol{\varepsilon}_m$ means that the modeling errors can be neglected compared to the desired accuracy. This can be done by considering a model that contains all anticipated sources of error, before decreasing its complexity in the implementation step [1, 7]. □

Thus, (3) gives an estimation of $\delta\boldsymbol{\xi}^{\text{nec}}$ in the worst case, which is when each $\delta\rho = \pm\Delta\rho$ and each $\delta\xi = \pm\delta\xi^{\text{nec}}$: in this case, $\text{abs}(\mathbf{J}_\rho^p\delta\boldsymbol{\rho} + \mathbf{J}_\xi^p\delta\boldsymbol{\xi}) = \text{abs}(\mathbf{J}_\rho^p)\Delta\boldsymbol{\rho} + \text{abs}(\mathbf{J}_\xi^p)\delta\boldsymbol{\xi}^{\text{nec}}$. However, matrices $\mathbf{J}_\xi^p$ and $\mathbf{J}_\rho^p$ are calculated with the non-exact values $\boldsymbol{\xi}$: a better estimation of $\delta\boldsymbol{\xi}^{\text{nec}}$ can be obtained using interval analysis [5]. Another point is that $\mathbf{J}_\xi^p$ is an $n \times m$ matrix, with most of the time $m > n$, which yields to an under-determined system of equations. So, several sets of $\delta\boldsymbol{\xi}^{\text{nec}}$ satisfying (3) can be chosen.

The second step of calibration is the identification process. From the kinematic model, measurements are taken on the robot and the kinematic parameters are estimated so that an objective function depending on both the measurements and the model parameters is minimized. Let consider that all the m kinematic parameters can be identified and that d measurements are taken for N_p different poses of the end-effector. A trivial condition for being able to perform calibration is:

(C2) Sufficient number of measurements: $d.N_p > m$ □

Each d-vector of exact measurements $\mathbf{y}_p^*$, with $p = 1..N_p$, is then compared to an estimation resulting from the kinematic model through the identification function $\mathbf{g}_p$ of pose p, $\boldsymbol{\varepsilon}_p$ being a residual considering the modeling errors:

$$\mathbf{g}_p(\mathbf{y}_p^*, \boldsymbol{\rho}_p^*, \boldsymbol{\xi}^*) = \boldsymbol{\varepsilon}_p \tag{4}$$

Considering the measurement noises $\delta\mathbf{y}_p^*$ and $\delta\boldsymbol{\rho}_p^*$, with $\mathbf{y}_p = \mathbf{y}_p^* + \delta\mathbf{y}_p^*$, and the parameter errors $\delta\boldsymbol{\xi}^*$, differentiating (4) with respect to all its parameters yields to:

$$\mathbf{g}_p(\mathbf{y}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi}) = \boldsymbol{\varepsilon}_p + \mathbf{G}_y^p\delta\mathbf{y}_p^* + \mathbf{G}_\rho^p\delta\boldsymbol{\rho}_p^* + \mathbf{G}_\xi^p\delta\boldsymbol{\xi}^* \tag{5}$$

where matrices $\mathbf{G}_y^p$, $\mathbf{G}_\rho^p$ and $\mathbf{G}_\xi^p$ all depend on the robot pose p. However, since the exact values of $\delta\mathbf{y}_p^*$ and $\delta\boldsymbol{\rho}_p^*$ are not known, the identification process is always performed under the hypothesis that $\mathbf{g}_p$ mostly depends on the variation $\delta\boldsymbol{\xi}^*$ of the kinematic parameters. This hypothesis can be written as in (6) and yields to **(C3)**.

$$\forall p = 1..N_p, \quad \text{abs}(\mathbf{G}_y^p\delta\mathbf{y}_p^* + \mathbf{G}_\rho^p\delta\boldsymbol{\rho}_p^*) \ll \text{abs}(\mathbf{G}_\xi^p\delta\boldsymbol{\xi}^*) \tag{6}$$

(C3) Necessary condition on measurement inaccuracy: Considering the maximal inaccuracy $\Delta\boldsymbol{\rho}$ of the joint coordinates and the necessary accuracy of the kinematic parameters $\delta\boldsymbol{\xi}^{\text{nec}}$, (6) can be rewritten as:

$$\forall p = 1..N_p, \quad \mathrm{abs}(\mathbf{G}_y^p)\Delta \mathbf{y}^{\max} + \mathrm{abs}(\mathbf{G}_\rho^p)\Delta \boldsymbol{\rho} < \mathrm{abs}(\mathbf{G}_\xi^p \delta \boldsymbol{\xi}^{\mathrm{nec}}) \tag{7}$$

with $\Delta \mathbf{y}^{\max}$ the maximal measurement inaccuracy, which enables to choose the appropriate measurement device. □

Condition **(C3)** leads to an estimation of $\Delta \mathbf{y}^{\max}$ in the worst case. If the obtained value of $\Delta \mathbf{y}^{\max}$ is too strong, first another set of $\delta \boldsymbol{\xi}^{\mathrm{nec}}$ can be chosen accordingly to **(C1)**. Then, (7) has to be valid for the N_p measurement configurations only. Thus, the inequality of (7) can also be verified through a proper choice of the observability matrices $\mathbf{G}_\xi^p$. This property is well-known and has already been studied through observability indexes [2, 9]. From (7), we can remark that analyzing $\mathbf{G}_y^p$ and $\mathbf{G}_\rho^p$ can also be of prime interest.

With **(C3)**, we obtain $\mathbf{g}_p(\mathbf{y}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi}+\delta\boldsymbol{\xi}) \approx \mathbf{g}_p(\mathbf{y}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi})+\mathbf{G}_\xi^p \delta\boldsymbol{\xi}$, with $\delta\boldsymbol{\xi} \neq \delta\boldsymbol{\xi}^*$ since $\delta\boldsymbol{\xi}$ considers the measurement noise, sensor inaccuracies and modeling errors. The objective of identification is to find the best set of parameter errors $\delta\boldsymbol{\xi}$ that minimizes functions $\mathbf{g}_p$ for $p = 1..N_p$. Setting $h = \sum_{p=1}^{N_p} \mathbf{g}_p^T \mathbf{g}_p = \mathbf{g}^T \mathbf{g}$ as the objective function of identification, with $\mathbf{g} = [\mathbf{g}_1^T \ldots \mathbf{g}_{N_p}^T]^T$, yields to the following normal Eq. (8) at iteration j:

$$\mathbf{G}_\xi^T \mathbf{G}_\xi \delta\boldsymbol{\xi}^{j+1} = \mathbf{G}_\xi^T \mathbf{g}\left(\mathbf{y}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi} + \sum_{i=1}^{j} \delta\boldsymbol{\xi}^i\right), \quad \text{with } \mathbf{G}_\xi = [\mathbf{G}_\xi^{1\,T} \ldots \mathbf{G}_\xi^{N_p\,T}]^T \tag{8}$$

An estimation of the kinematic parameters is given by $\boldsymbol{\xi}_{\mathrm{est}} = \boldsymbol{\xi} + \sum_{i=1}^{j_{\max}} \delta\boldsymbol{\xi}^i$, $j_{\max}$ being the number of the last iteration. Most of the time, the optimization algorithm is stopped when the residual of h is under a certain threshold. The measurement errors have to be considered in this threshold. However, in practice, this threshold is manually adjusted to obtain the algorithm convergence and has no physical meanings. From the above conditions, the necessary accuracy $\delta\boldsymbol{\xi}^{\mathrm{nec}}$ was derived. This value considers the desired accuracy and can be reached by the identification algorithm considering the measurement noise due to **(C3)**. Thus, the stop criterion of the optimization algorithm can be set as $\mathrm{abs}(\delta\boldsymbol{\xi}^{j_{\max}}) \leq \delta\boldsymbol{\xi}^{\mathrm{nec}}$ which is physically meaningful.

However, results of identification also depend on the scaling of the identification function **g** [4]. Thus, the developed conditions are necessary but not sufficient because of observability issues. They will however be useful to define a physical bound for the observability index O_3 [9].

3 Observability Issues

In practice, even if the previous conditions of calibration are fulfilled, only k model parameters among m can be identified, with $k \leq m$, because some model parameter errors $\delta\xi$ cannot be observed during identification. Such observability problems depend on properties of the observability matrix $\mathbf{G}_\xi$. Three types of identifiability problems can occur [4]: unidentifiable, weakly identifiable or identifiable only in linear combination.

Non-Observability: It occurs when a kinematic parameter is not involved in the identification function $\mathbf{g}$. In this case, its corresponding column of $\mathbf{G}_\xi$ is zero. If the rank of $\mathbf{G}_\xi^T \mathbf{G}_\xi$ is r, then $m - r$ kinematic parameter errors are non-observable. This phenomenon appears when measurements are only partial or when measurement configurations do not involve all the kinematic parameters, and can be observed with a QR-decomposition. A parameter whose variation is non-observable is called non-identifiable.

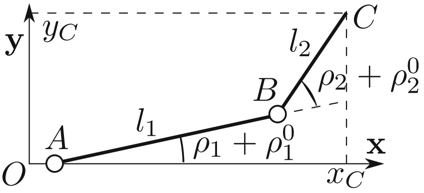
Low-Observability: A model parameter is said to be low-identifiable if its corresponding column of $\mathbf{G}_\xi$ is close to zero: an important variation of its value only has small consequences on the identification function $\mathbf{g}$, compared to the influence of measurement and modeling errors. The identification of such a parameter often leads to an important variation of its initial value and degrades the robustness of calibration [4].

Linked-Observability: The linked-observability occurs when two or more parameter errors $\delta\xi$ appear in the normal equations only as a linear combination whose variation is less than the measurement noise. Their corresponding columns of $\mathbf{G}_\xi$ are linearly related and only the linear combination of those parameters can be observed. The linked-observability often appears for robot having a small workspace, or when measures are not generic enough (with constant orientation for example).

Non-observability is very easy to check. In order to tackle low- and linked-observability problems, observability indexes were proposed [9]. Those indexes are based on properties of the singular values of the observability matrix $\mathbf{G}_\xi$. However, a proper scaling of this matrix is compulsory for a comparison of its singular values [4]. A proper scaling of $\mathbf{G}_\xi$ can be obtained due to the above developed conditions of calibration. Let consider the worst case of observability during identification. This occurs when the parameter variation is minimum, that is $\delta\boldsymbol{\xi}^{\text{nec}}$, and when the identification function $\mathbf{g}$ is highly noisy, that is $\delta\mathbf{g}^{\min} = [\delta\mathbf{g}_1^T \ldots \delta\mathbf{g}_{N_p}^T]^T$ with $\delta\mathbf{g}_p = \text{abs}(\mathbf{G}_y^p)\Delta\mathbf{y}^{\max} + \text{abs}(\mathbf{G}_\rho^p)\Delta\boldsymbol{\rho}$. Let define the scaling vectors $\Delta\boldsymbol{\xi}^i = \text{diag}(\delta\boldsymbol{\xi}^{\text{nec}})^{-1}.\delta\boldsymbol{\xi}^i$ and $\Delta\mathbf{g} = \text{diag}(\delta\mathbf{g}^{\min})^{-1}.\mathbf{g}$, where $\text{diag}(\mathbf{x})$ stands for a matrix whose diagonal is the vector $\mathbf{x}$. Thus, in the worst case, $\Delta\boldsymbol{\xi}^i$ and $\Delta\mathbf{g}$ become vectors whose terms are approximately equal to 1. The normal Eq. (8) can be written as:

$$\mathbf{H}_\xi \Delta\boldsymbol{\xi}^{j+1} = \Delta\mathbf{g} \quad \text{with} \quad \mathbf{H}_\xi = \text{diag}(\delta\mathbf{g}^{\min})^{-1}.\mathbf{G}_\xi.\text{diag}(\delta\boldsymbol{\xi}^{\text{nec}}) \tag{9}$$

In this case, if a singular value σ_L of $\mathbf{H}_\xi$ is under 1, the measurement noise considered in $\Delta\mathbf{g}$ will be amplified, which may lead to a wrong estimation of $\Delta\boldsymbol{\xi}^{j+1}$. So, the condition $\sigma_L(\mathbf{H}_\xi) \geq 1$ is sufficient to prevent observability issues. This condition is related to the observability index O_3 [9]. However, it will be seen in the following example that this condition is sufficient but not necessary for observability.



Nominal values	Real values
x_A=0.1 m	x_A=0 m
y_A=0 m	y_A=0 m
l_1 =1 m	l_1 =0.99 m
l_2 =0.5 m	l_2 =0.45 m
ρ_1^0 =0°	ρ_1^0 =1°
ρ_2^0 =0°	ρ_2^0 =−1°

Fig. 1 The serial planar 2-bar mechanism and its kinematic parameter definition

4 Application Example

The 2-bar serial planar mechanism, presented in Fig. 1, consists of two bars of lengths l_1 and l_2, actuated by two motors at angles ρ_1 and ρ_2 from their initial poses ρ_1^0 and ρ_2^0, respectively. The objective is to calibrate this serial manipulator considering measurements x_C of its end-effector in the measurement frame $(O, \mathbf{x}, \mathbf{y})$. The nominal (initial guess) and real values of the kinematic parameters $\boldsymbol{\xi} = [x_A, y_A, l_1, l_2, \rho_1^0, \rho_2^0]^T$ are given in Fig. 1.

With vectors of the actuator positions $\boldsymbol{\rho}_p = [\rho_1^p, \rho_2^p]^T$ and end-effector locations $\mathbf{x}_p = [x_C^p, y_C^p]^T$ of pose $p \in \mathcal{W}$, the exact kinematic model $\mathbf{f}^*$ can be written, with $\alpha_1^p = \rho_1^p + \rho_1^0$, $\alpha_2^p = \alpha_1^p + \rho_2^p + \rho_2^0$, $s_\alpha = \sin(\alpha)$, and $c_\alpha = \cos(\alpha)$:

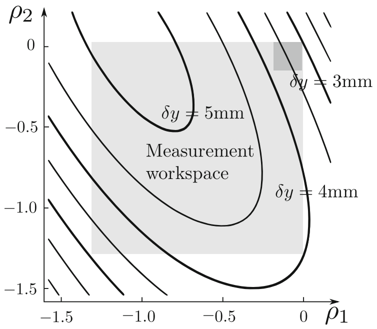
$$\mathbf{f}^*(\mathbf{x}_p, \boldsymbol{\rho}_p, \boldsymbol{\xi}) = \begin{bmatrix} x_A + l_1 c_{\alpha_1^p} + l_2 c_{\alpha_2^p} - x_C^p \\ y_A + l_1 s_{\alpha_1^p} + l_2 s_{\alpha_2^p} - y_C^p \end{bmatrix}, \quad \text{which yields to:} \tag{10}$$

$$\mathbf{J}_\rho^p = \begin{bmatrix} -l_1 s_{\alpha_1^p} - l_2 s_{\alpha_2^p} & -l_2 s_{\alpha_2^p} \\ l_1 c_{\alpha_1^p} + l_2 c_{\alpha_2^p} & l_2 c_{\alpha_2^p} \end{bmatrix}, \; \mathbf{J}_\xi^p = \begin{bmatrix} 1 & 0 & c_{\alpha_1^p} & c_{\alpha_2^p} & -l_1 s_{\alpha_1^p} - l_2 s_{\alpha_2^p} & -l_2 s_{\alpha_2^p} \\ 0 & 1 & s_{\alpha_1^p} & s_{\alpha_2^p} & l_1 c_{\alpha_1^p} + l_2 c_{\alpha_2^p} & l_2 c_{\alpha_2^p} \end{bmatrix}$$

Considering the desired accuracy $\Delta\mathbf{x}_f = [0.01\text{m}, 0.01\text{m}]^T$, the accuracy of the actuator sensors $\Delta\boldsymbol{\rho} = [0.01°, 0.01°]^T$, the allowed motor positions ρ_1 and $\rho_2 = \pm 90°$, and setting that all the distance parameters (respectively the orientation parameters) must have the same necessary accuracy δx^{nec} (respectively $\delta\theta^{\text{nec}}$), we obtain from **(C1)**: $\delta x^{\text{nec}} = 1\text{mm}$ and $\delta\theta^{\text{nec}} = 0.1°$. These values allow to reach $\Delta\mathbf{x}_f$ over the entire robot workspace. The detailed calculus can be found in the Maple® or Mathematica® worksheets of [10].

As for the number of measurements, **(C2)** gives $N_p \geq 6$. However, this is a lower bound and the required number of measurements allowing good calibration results is still an open issue. This number can be minimized through a proper choice of the measurement configurations [2]. The number of measurements is not limited for simulation. Thus, two cases will be studied: $N_p = 250$ and $N_p = 20$ configurations randomly chosen in the measurement workspace.

Since measurements are directly the output x_C of the mechanism, the identification function $\mathbf{g}_p$ of pose p can be chosen as the first row of the kinematic model $\mathbf{f}^*$ of (10).



Case	1	2	3	4
N_p	250	250	20	250
ρ_1 (°)	$[-75, 0]$	$[-75, 0]$	$[-75, 0]$	$[-10, 0]$
ρ_2 (°)	$[-75, 0]$	$[-75, 0]$	$[-75, 0]$	$[-10, 0]$
δy (mm)	3	4	3	3
Δx_C (mm)	10.6	14.3	66.8	436
Δy_C (mm)	7.7	10.1	46.9	457
$\overline{\sigma_L}(\mathbf{H}_\xi)$	0.239	0.179	0.058	0.004

Fig. 2 Choice of the necessary measurement accuracy as a function of the measurement workspace and results of calibration for the 4 cases

This enables the matrices $\mathbf{G}_\rho^p$ and $\mathbf{G}_\xi^p$ to be equal to the first row of $\mathbf{J}_\rho^p$ and $\mathbf{J}_\xi^p$ respectively, and $\mathbf{G}_y^p = [-1]$. With those matrices, the last condition of calibration **(C3)** can be derived. Thus, the accuracy of the measurement device can be chosen accordingly to the measurement workspace as shown in Fig. 2. In this figure are plotted the curves for ρ_1 and $\rho_2 \in [-\frac{\pi}{2}; 0]$ for which (7) is exactly satisfied considering different values of the measurement device accuracy δy. Setting a measurement workspace such as ρ_1 and $\rho_2 \in [-75°; 0]$, represented as a light-gray square in Fig. 2, a necessary accuracy δy^{nec} of 3mm can be chosen from **(C3)**.

Finally, calibration is performed using a least-square algorithm with normal equations of (9). Obviously, y_A is non-identifiable since it does not appear in the identification function $\mathbf{g}_p$, and is removed from the set of identifiable parameters. For 4 different cases presented in Fig. 2, N_p random poses are taken in the measurement workspace. Considering the actuator sensor inaccuracies, the pose x_C^p is calculated and a uniformly distributed measurement noise of $\pm\delta y$ is added. Calibration is repeated 10 000 times and results are presented in Fig. 2 as the maximal obtained inaccuracies Δx_C and Δy_C on the 100 verification poses, randomly taken in the manipulator workspace $\mathcal{W}$.

Case 1 is the ideal case: all conditions are fulfilled and the accuracy $\Delta\mathbf{x}_f$ is reached over $\mathcal{W}$. Case 2 shows that **(C3)**, the maximal measurement inaccuracy, is a necessary condition to reach $\Delta\mathbf{x}_f$. Case 3 confirms the necessity to optimize the configurations of measurement when decreasing their number N_p. Finally, the linked-identifiability issues are addressed in case 4: because of the small measurement workspace and the measurement noise, x_A, l_1 and l_2 appear in $\mathbf{g}_p$ only as a linear combination and cannot be properly identified. The average of the minimum singular values $\overline{\sigma_L}(\mathbf{H}_\xi)$ of the last iteration is also given. Since identifiability is possible for cases 1 and 2, the physically meaningful bound for O_3, $\sigma_L(\mathbf{H}_\xi) \geq 1$, seems to be a too strong condition.

5 Conclusion and Discussions

Necessary mathematical conditions were developed to ensure the quality of calibration with respect to the final accuracy. According to those conditions, a physically meaningful stop criterion for the optimization algorithm can be derived. Those conditions were illustrated on a simple serial example but also stand for parallel manipulators.

The developed conditions also enable the calculus of a lower bound for the minimum singular value of the observability matrix, after a proper scaling. This bound leads to a sufficient condition for observability but seems overestimated since it considers the worst case. More studies are needed to derive the most appropriate formula for this threshold. However, the developed scaled observability matrix must be of prime interest for a proper choice of the configurations of measurement. In the same field, **(C3)** shows that considering the sensitivity of the identification function to measurement noise and sensor inaccuracy through matrices $\mathbf{G}_y^p$ and $\mathbf{G}_\rho^p$ must also be of prime interest.

Acknowledgments This work has been supported by Thales Alenia Space and the Region "Provence-Alpes-Côte-d'Azur".

References

1. Arendt, P., Apley, D.W., Chen, W.: Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. ASME J. Mech. Des. **134**(100), 908 (2012)
2. Daney, D., Papegay, Y., Blaise, M.: Choosing measurement poses for robot calibration with the local convergence method and tabu search. Int. J. Robot. Res. **24**(6), 501–518 (2005)
3. Everett, L.J., Driels, M., Mooring, B.W.: Kinematic modelling for robot calibration. In: Robot. Auto. **4** 183–189 (1987)
4. Hollerbach, J., Khalil, W., Gautier, M.: Handbook of Robotics, Chap. 14. Springer, Berlin (2008)
5. Merlet, J.P., Daney, D.: Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 942–947. Barcelona (2005)
6. Merlet, J.P.: Parallel Robots, 2nd edn. vol. 128. Springer, Dordrecht (2006)
7. Mooring, B.W., Padavala, S.S.: The effect of kinematic model complexity on manipulator accuracy. In: Robotics and Automation, vol. 1. (1989)
8. Mooring, B., Driels, M., Roth, Z.: Fundamentals of Manipulator Calibration. Wiley, New York (1991)
9. Sun, Y., Hollerbach, J.M.: Observability index selection for robot calibration. In: Robotics and Automation, pp. 831–836. (2008)
10. URL www-sop.inria.fr/coprin/developpements/worksheets-ck13/
11. Vischer, P.: Improving the accuracy of parallel robots. Ph.D. thesis, Ecole Polytechnique Fdrale de Lausanne (1996)

An Improved Force Distribution Algorithm for Over-Constrained Cable-Driven Parallel Robots

Andreas Pott

Abstract In this paper we present an improved method to compute force distributions for cable-driven parallel robots. We modify the closed-from solution such that the region where a solution is found is extended almost to the theoretical maximum, i.e. the wrench-feasible workspace. At the same time continuity along trajectories as well as real-time efficiency are maintained. The algorithm's complexity and thus the computational burden scales linearly in the number of redundant cables. Therefore, the algorithm can also be used for highly redundant cable robots. The proposed algorithm is compared to known methods and computational results are presented based on the IPAnema prototype.

Keywords Cable-driven robots · Force distribution · Closed-form · Real-time

1 Introduction

Cable-driven parallel robots are a special kind of parallel manipulators where the rigid struts are replaced by flexible elements. Many cable robots facilitate more cables m than degrees-of-freedom n in order to withstand applied wrenches $\mathbf{w}$ in arbitrary directions. Therefore, these robots are redundantly actuated and static or dynamic balancing of the robots requires a distribution of actuator forces amongst the cables. The force and torque equilibrium for cable robots is usually written in matrix form as follows [13]

A. Pott (✉)
Fraunhofer IPA, Stuttgart, Germany
e-mail: andreas.pott@ipa.fraunhofer.de

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_16,
© Springer Science+Business Media Dordrecht 2014

$$\mathbf{A}^{\mathrm{T}}\mathbf{f} + \mathbf{w} = \mathbf{0} \quad \text{with} \quad 0 < f_{\min} \leq f_i \leq f_{\max}, \quad i \in [1, m], \tag{1}$$

where the matrix $\mathbf{A}^{\mathrm{T}}$ is the pose dependent structure matrix or sometimes also called wrench matrix, $\mathbf{f}$ are the forces in the cables, and $f_{\min}$, $f_{\max}$ represent the minimum and maximum feasible cable forces, respectively. The presented structure equations also covers the dynamic case when using d'Alembert's principle to add the inertial forces to the applied wrench $\mathbf{w}$. Computing force distributions for the cables e.g. for control requires finding solutions to the structure equations above, where the linear system is under-determined for the cable forces $\mathbf{f}$. Therefore, infinitely many solutions are consistent with the structure equations but these solutions are not necessarily in the feasible region given by the bounds $f_{\min}$, $f_{\max}$. Given that such solution exists the problem addressed in this paper is to efficently compute one solution that is continuous along a trajectory of the robot's mobile platform.

Different approaches were proposed in the literature to calculate force distributions, and each approach delivers force distributions with different characteristics while requiring varying computational efforts:

- Gradient-based optimization using a p-norm for $r > 1$ (Verhoeven's method) [13]
- Specialized optimization for p-norm with $p = 4$ [5]
- Constrainted l_1-norm optimization [12]
- Minimizing p-norm with Dykstra method [6]
- Closed-form solution for $p = 2$ [11]
- Linear programming [1, 10]
- Quadratic programming for $r = 2$ [4] and for $r = 0$ [8][1]
- Nonlinear programming [3]
- Barycentric approach [9] and improved implementations [7]
- Kernel method [13, p. 58] for $r = 1$
- Weighted sum of solution space vertices [3]
- Available wrench set [2]

A comparison of some force distribution methods and their properties is given in Table 1. We briefly explain the properties listed in the head of the table. An algorithm is said to be *real-time capable* if the computation time is reasonably short, the worst-case computation time can be strictly bounded, and a real-time implementation was reported in the literature. Some iterative methods were successfully used for computation in real-time although their worst-case computational time was not determined. The *force niveau* may be choosen, e.g. the algorithm may aim at finding minimal (lo), maximal (hi), average (mi), or any solution (any). Furthermore, there might be a parameter (param) that allows to smoothly adjust the niveau of tension between low and high. A couple of authors [11, 13] reported approaches that may fail to find force distributions for special poses of the wrench-feasible workspace. Full *workspace coverage* indicates that for every pose of the wrench-feasible workspace a solution can be found. For some methods it is not known (n.a.) if they cover the full workspace. An algorithm is said to provide *continuity*, if continuous trajectories

[1] Li [8] only deals with the non redundant case $r = 0$, i.e. six cables and six degrees-of-freedom.

Table 1 Comparison of the different methods to compute force distributions

Method	Real-time capable	Force niveau	Workspace coverage	Continuity	Max. redundancy	Computational speed
Linear programming	no	any	yes	no	any	fast
Quadratic programming	yes	hi,lo	n.a.	yes	any	medium
Gradient-based optimization	no	param	no	yes	any	medium
Dykstra	no	any	yes	no	any	slow
Closed-form	yes	any	no	yes	any	fast
Barycentric	yes	mi	yes	yes	$r = 2$	fast
Weighted sum	yes	mi	yes	mostly	any	medium
Kernel method	yes	hi, mi, lo	yes	yes	$r = 1$	fast
Available wrench set	no	hi, mi, lo	yes	no	any	slow

in the pose $\mathbf{r}$, $\mathbf{R}$ as well as in the applied wrench $\mathbf{w}$ produce continuous trajectories in the cable forces $\mathbf{f}$, except for crossing a singularity. Some methods are limited to a certain degree-of-redundancy $r \leq 0$ either because they are specific or because their implementation can hardly be generalized to arbitrary r. The evaluation of the computational speed is problematic because it requires comparable implementations which are not available for all methods reported in the literature. Anyway, it was tried to set up a basic ranking taking into account how complex the underlying numerical method is. For example, linear system solving is considered to be faster than inverting a matrix, which in turn is faster than computing a singular value decomposition. Designing a real-time system might become involved if an advanced numerical algorithm such as advanced optimization or singular value decomposition shall be used. This is due to lack of appropriate real-time capable implementations of the algorithm although the algorithm is part of every state-of-the-art numerical toolbox. The computational speed depends on the degree-of-redundancy in addition to the algorithm's complexity. For this assessment, a low degree-of-redundancy was assumed.

From the table it can be seen that no method is known that is real-time capable, covers the full workspace, delivers continuous solution for control, and works for robot with arbitrary degree-of-redundancy. In this paper, we propose an improved variant of the closed-form method to overcome its shortcomings with respect to workspace coverage while maintaining an acceptable computation time for usage in a real-time controller.

2 Improved Closed-Form Method

Lately, we developed a formula to compute a solution for the force distribution problem in closed-form [11]. The basic idea of the method is to perform a coordinate transformation to the medium feasible cable force $\mathbf{f}_m = \frac{1}{2}(\mathbf{f}_{\min} + \mathbf{f}_{\max})$. This also changes parts of the optimization problem from constrained optimization to pure minimization. The cable forces $\mathbf{f}$ can be computed as [11]

$$\mathbf{f} = \mathbf{f}_m + \mathbf{f}_v = \mathbf{f}_m - \mathbf{A}^{+\mathrm{T}}(\mathbf{w} + \mathbf{A}^{\mathrm{T}}\mathbf{f}_m), \tag{2}$$

where $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{A}^{+\mathrm{T}}$ are the structure matrix and its pseudo-inverse, respectively, and $\mathbf{w}$ is the applied wrench. As discussed in [11] this formula might fail to provide a feasible solution although such a solution exists, if the magnitude of the variable part $\mathbf{f}_v$ of the force distribution is in the range

$$\frac{1}{2} f_m \leq |\mathbf{f}_v|_2 \leq \frac{1}{2}\sqrt{m} f_m. \tag{3}$$

If $|\mathbf{f}_v|_2$ violate the upper limit no solution exists and if it is below the lower limit the distribution is feasible. This undefined case occurs amongst others close to the boarder of the wrench-feasible workspace, for robots with many redundant cables, and for redundant robots in suspended configuration.

In the following, we propose to extend the method such that feasible force distributions are found in almost all cases where the original method fails.[2] The closed-form solution is guaranteed to fulfill the force equilibrium but may violate the force limits. Thus, the following approach is proposed:

1. Equation (2) is used to compute an estimate for the force distribution. If this initial guess already fulfills the cable force conditions we have the sought solution and stop the algorithm.
2. Otherwise, let i be the cable with the largest force over (under) the maximum (minimum) feasible cable force. If one moves from this distribution along the spanning base of the structure matrix kernel one must cross the value where f_i reaches its maximum (minimum) feasible value.
3. Therefore, it is assumed[3] that a feasible force distribution minimizing the 2-norm can only be found if this cable force is fixed to its maximum (minimum) value $f_{\max}$ ($f_{\min}$). Using a constant value for cable force f_i simplifies the force distribution problem as follows

$$\mathbf{A}'^{\mathrm{T}}\mathbf{f}' + \mathbf{w}' = \mathbf{0} \quad \text{with} \quad \mathbf{w}' = f_{\max}\left[\mathbf{A}^{\mathrm{T}}\right]_i + \mathbf{w}, \qquad \left(\mathbf{w}' = f_{\min}\left[\mathbf{A}^{\mathrm{T}}\right]_i + \mathbf{w}\right), \tag{4}$$

[2] In numerical studies some poses were found on the very boarder of the workspace where the presented methods fails to find a solution.

[3] Unfortunately, we have no formal proof that this holds true in general.

where $\mathbf{A}'^{\mathrm{T}}$ and $\mathbf{f}'$ is the structure matrix and the cable forces vector with the i-th column/element dropped, respectively. $[\mathbf{A}^{\mathrm{T}}]_i$ denotes the i-th column of the matrix $\mathbf{A}^{\mathrm{T}}$. Thus, we have reduced the actuator redundancy r by one.

4. Now we compute the solution by recursively reducing the order and computing the closed-form solution by going to step 1 until:

   a. we find a feasible distribution,
   b. the remaining degree-of-redundancy is negative $r < 0$, then no solution exists,
   c. Equation (3) proofs that no feasible solution exists because because the computed force violates the right part of Eq. (3)

Therefore, we find the desired cable force distribution (if it exists) with at most r evaluations of the closed-form formula (2).

3 Simulation Results and Computation Time

For the numerical examples we use the geometrical parameters of the cable robot IPAnema 1 given in Table 2. To compare different algorithms for force distribution a sample trajectory is used which is depicted in Fig. 1. The waypoints of the trajectory are indicated by number 0 to 8 and the following plots with cable forces against time have additional marks above the x-axis indicating the waypoints for better reference. Position was linearly interpolated between the waypoints and the trajectory was choosen such that the robot moves in different regions of the workspace and finally crosses the boarder of the wrench-feasible workspace between waypoint 7 and 8. The force limits were $f_{\min} = 1$ and $f_{\max} = 10$ N. Inertia effects of the platform were neglected.

Figure 2a illustrates the proposed improved algorithms based on closed-form estimation and correction for the remaining cables. From the diagram one can see that the force distribution is continuous along the trajectory and the magnitude of the forces are on a medium level. When approaching the boarder of the workspace (e.g. between $t = 10.0$ s and $t = 17.0$ s) one or two cable forces reach the minimum cable force and remain constant at the limit. It can be seen from the shape of the diagram

Table 2 IPAnema 1 nominal geometric parameters: platform vectors **b** and base vectors **a**

Cable i	Base vector $\mathbf{a}_i$	Platform vector $\mathbf{b}_i$
1	$[-2.0, 1.5, 2.0]^{\mathrm{T}}$	$[-0.06, 0.06, 0.0]^{\mathrm{T}}$
2	$[2.0, 1.5, 2.0]^{\mathrm{T}}$	$[0.06, 0.06, 0.0]^{\mathrm{T}}$
3	$[2.0, -1.5, 2.0]^{\mathrm{T}}$	$[0.06, -0.06, 0.0]^{\mathrm{T}}$
4	$[-2.0, -1.5, 2.0]^{\mathrm{T}}$	$[-0.06, -0.06, 0.0]^{\mathrm{T}}$
5	$[-2.0, 1.5, 0.0]^{\mathrm{T}}$	$[-0.06, 0.06, 0.0]^{\mathrm{T}}$
6	$[2.0, 1.5, 0.0]^{\mathrm{T}}$	$[0.06, 0.06, 0.0]^{\mathrm{T}}$
7	$[2.0, -1.5, 0.0]^{\mathrm{T}}$	$[0.06, -0.06, 0.0]^{\mathrm{T}}$
8	$[-2.0, -1.5, 0.0]^{\mathrm{T}}$	$[-0.06, -0.06, 0.0]^{\mathrm{T}}$

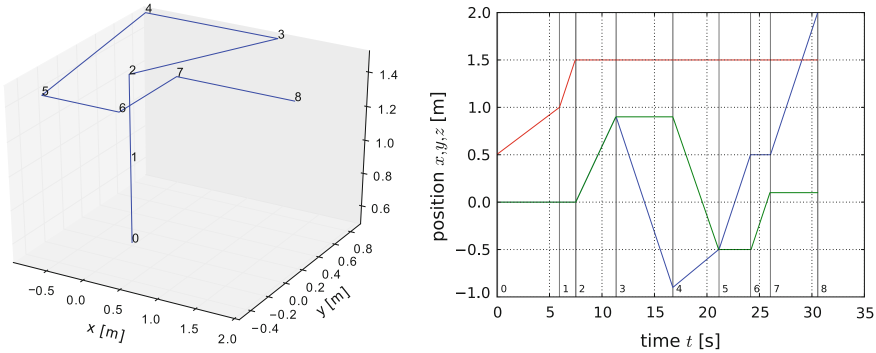


Fig. 1 Test trajectory used for the evaluation

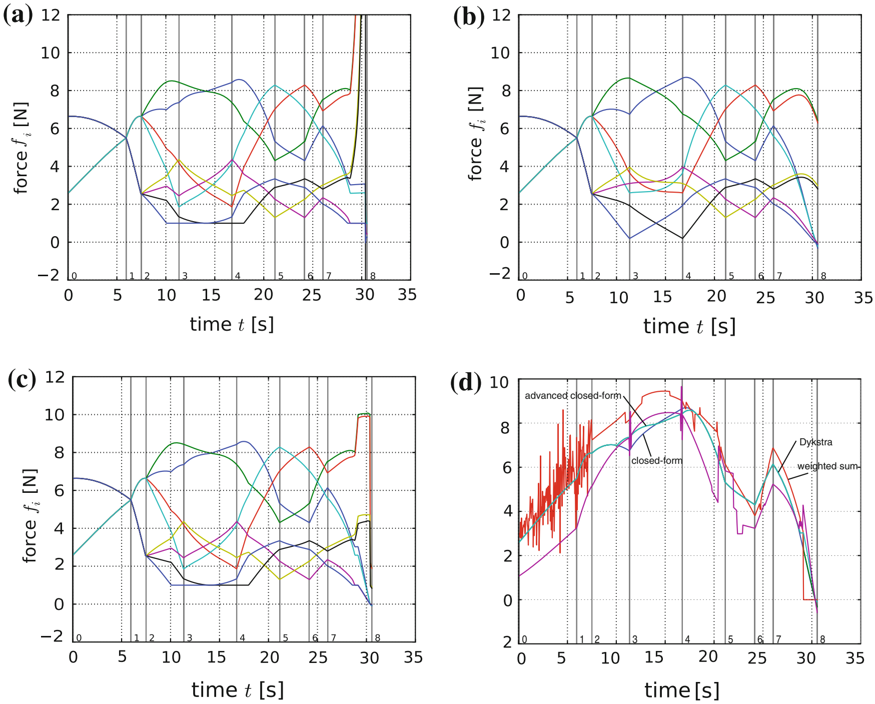


Fig. 2 Comparison of different methods to compute the force distribution. **a** Improved closed-form, **b** Closed-form solution, **c** Dykstra method, **d** Comparison of different methods

that the cable forces quickly increase after leaving the workspace. Anyway, the force distributions remain continuous after crossing the boarder of the workspace.

In Fig. 2b the force distribution is shown for the original closed-form method for comparison. When the platform remains in the inner region of the workspace the results match the force distributions computed with the correction technique.

Table 3 Comparison of computation time on an Intel Core i5-3320M 2.6 GHz,Visual C++ 2010

Algorithm	Calculation time (ms)	Relative time (%)	Evaluations per ms
Closed-form	1,173	100	293
Advanced closed-form	3,359	286	102
Dykstra	71,612	6,103	5
Weighted sum	48,512	4,134	7

Close to the boarder of the workspace the closed-form formula fail to compute force distributions although such distributions exist as it can be seen between waypoint 2 and 5 and also between 7 and 8, where the closed-form solution is not able to prevent some cables from violating the lower force limits.

Cable forces computed with the Dykstra methods are presented in Fig. 2c. It can be observed for the Dykstra method that some cable forces get limited to the minimal values when the boarder of the workspace is approached. After crossing the workspace board between waypoint 7 and 8 force distributions computed with Dykstra show a different behaviour compared to the proposed scheme.

In Fig. 2d the computed forces for the first cable f_1 is compared for different methods. Some methods like the uncorrected weighted sum method do not even provide continuous shapes for the forces which becomes evident between waypoint 0 and 2. Other methods show discrete steps at certain points on the trajectory.

A comparison of the computation time is difficult because the computation time is influenced by the maturity of the implementation as well as the underlying numerical algorithms, the used compiler, the CPU of the real-time system, and the operating system. In the performance test presented here, four algorithms were used for workspace computation with around 344,000 evaluations on an Intel Core i5-3320M. As an estimate some numbers are given in Table 3. The table lists both absolute and relative computation time to allow for comparison amongst the algorithms as well as to present an estimate on the usability in a real-time controller. As expected the closed-form solution works faster than its improved version but the difference is comparibly small. The performance advantages of the presented method over the iterative Dykstra method and over the exhausive search of weighted-sum method can be explained by the more efficient search strategy. Each iteration step of the advanved close-form method is used to fix at least one component in force vector. Both closed-form methods allow for many evaluations based on a controller cycle time of 1 ms and their implementations only requires matrix multiplication and solving of a linear system.

4 Conclusions

In this paper we proposed an improved algorithm to compute force distributions for over-constrained cable-driven parallel robot under real-time requirements. The improved version overcomes a major drawback of the closed-form solution, i.e. that the algorithm failed to find force distributions especially close to the boarder of the workspace. The improved algorithm is still applicable for robots with a large number of cables. Although the computational time of the presented algorithm is now linear in the number of redundant cables it still provides a solution for highly redundant cable robots in reasonable time.

Acknowledgments This work was partially supported by Fraunhofer-Gesellschaft Internal Programs under Grant No. WISA 823 244. Furthermore, the research leading to these results received founding for the European Community's Seventh Framework Program under grant agreement number NMP2-SL-2011-285404-CableBot.

References

1. Borgstrom, P.H., Jordan, B.L., Sukhatme, G.S., Batalin, M.A., Kaiser, W.J.: Rapid computation of optimally safe tension distributions for parallel cable-driven robots. IEEE Trans. Rob. **25**(6), 1271–1281 (2009)
2. Bouchard, S., Moore, B., Gosselin, C.: On the ability of a cable-driven robot to generate a prescribed set of wrenches. J. Mech. Rob. **2**(1), 1–10
3. Bruckmann, T.: Auslegung und Betrieb redundanter paralleler Seilroboter. Universität Duisburg-Essen (2010)
4. Bruckmann, T., Pott, A., Hiller, M.: Calculating force distributions for redundantly actuated tendon-based stewart platforms. In: ARK, pp. 403–412. Springer, Ljubljana (2006)
5. Gosselin, C.: On the determination of the force distribution in overconstrained cable-driven parallel mechanisms. In: Proceedings of the Second International Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, pp. 9–17. Montpellier, France (2008).
6. Hassan, M., Khajepour, A.: Minimum-norm solution for the actuator forces in cable-based parallel manipulators based on convex optimization. In: IEEE International Conference on Robotics and Automation, pp. 1498–1503 (2007)
7. Lamaury, J., Gouttefarde, M.: A tension distribution method with improved computational efficiency. In: Cable-driven parallel robots, pp. 71–85. Springer, Berlin (2013)
8. Li, H., Zhang, X., Yao, R., Sun, J., Pan, G., Zhu, W.: Optimal force distribution based on slack rope model in the incompletely constrained cable-driven parallel mechanism of fast telescope. In: Cable-driven parallel robots, pp. 87–102. Springer, Berlin (2013)
9. Mikelsons, L., Bruckmann, T., Schramm, D., Hiller, M.: A real-time capable force calculation algorithm for redundant tendon-based parallel manipulators. In: ICRA. Pasadena (2008)
10. Oh, S.R., Agrawal, S.K.: Cable suspended planar robots with redundant cables: controllers with positive tensions. IEEE Trans. Rob. (2005)
11. Pott, A., Bruckmann, T., Mikelsons, L.: Closed-form force distribution for parallel wire robots. In: Computational Kinematics, pp. 25–34. Springer, Duisburg (2009)
12. Snyman, J.A., Hay, A.M.: Analysis and optimization of a planar tendon-driven parallel manipulator. In: ARK, pp. 303–312. Kluwer, Sestri Levante (2004)
13. Verhoeven, R.: Analysis of the workspace of tendon-based stewart platforms. Ph.D. thesis, University of Duisburg-Essen, Duisburg (2004)

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

Josep M. Porta, Lluís Ros, Oriol Bohigas, Montserrat Manubens, Carlos Rosales and Léonard Jaillet

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

Keywords Kinematic constraints · Motion analysis and planning · Branch-and-prune · Higher-dimensional continuation

J. M. Porta (✉) · L. Ros · O. Bohigas · M. Manubens · C. Rosales · L. Jaillet
Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
e-mail: porta@iri.upc.edu

L. Ros
e-mail: lros@iri.upc.edu

O. Bohigas
e-mail: obohigas@iri.upc.edu

M. Manubens
e-mail: mmanuben@iri.upc.edu

C. Rosales
e-mail: crosales@iri.upc.edu

L. Jaillet
e-mail: ljaillet@iri.upc.edu

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_17,

1 Introduction

The notion of configuration space (C-space) is fundamental in Robotics. It allows designing motion planning algorithms for broadly-defined classes of robots or mechanisms, without worrying about their particular geometry or multibody structure. In most Robotics textbooks, this notion is introduced for open-chain mechanisms, where the C-space has an explicit global parametrization. In this way, C-spaces are readily understood and algorithms operating on them can be readily defined. In many cases, however, the C-space can have a more intricate structure, and its analysis is by no means trivial. This is the case of parallel manipulators, reconfigurable mechanisms, or robots working under geometric or contact constraints, but similar problems arise when exploring the motions of a protein, or when assembling parts using spatial constraints. Common to these problems is the fact that the feasible configurations are implicitly defined by a nonlinear system of equations

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

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

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

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

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

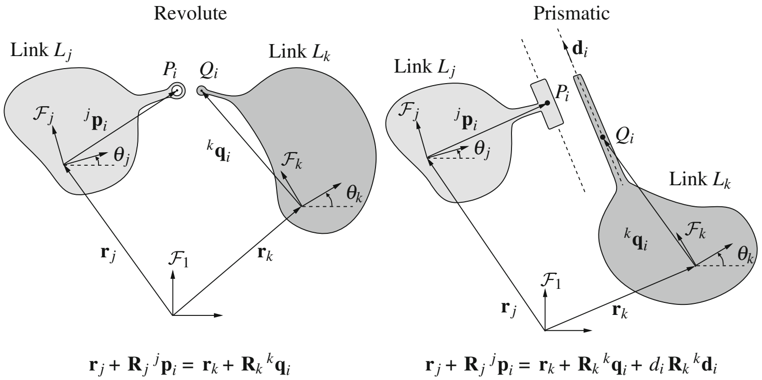


Fig. 1 Assembly constraints for planar pairs. In the equations, $\boldsymbol{r}_j$ is the position of $\mathcal{F}_j$ relative to $\mathcal{F}_1$, $\boldsymbol{R}_j$ is the rotation matrix of angle θ_j, ${}^j\boldsymbol{p}_i$ is the position of P_i in $\mathcal{F}_j$, and ${}^k\boldsymbol{d}_i$ provides $\boldsymbol{d}_i$ in the basis of $\mathcal{F}_k$. $\boldsymbol{R}_j = \boldsymbol{R} \cdot \boldsymbol{R}_k$ in the prismatic joint, where $\boldsymbol{R}$ is a constant rotation matrix

2 Branch-and-Prune Methods

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

Initially, the kinematic equations must be formulated in a form allowing simple pruning operations. In our case we depart from the formulation in Fig. 1, which leads to polynomial equations of a simple form with little manipulation. If J_i is a revolute joint connecting links L_j and L_k, the assembly constraint is equivalent to imposing the coincidence of two points on the axis of the joint, P_i and Q_i, respectively fixed to L_j and L_k. If J_i is prismatic, we further define a unit vector aligned with the joint, $\boldsymbol{d}_i$, and force P_i to lie on the line of L_k defined by Q_i and $\boldsymbol{d}_i$, fixing the orientation of L_j relative to L_k. By assigning a reference frame $\mathcal{F}_j$ to every link L_j, and taking $\mathcal{F}_1$ to be the absolute frame, these conditions can then be written as shown in the bottom of the figure.

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

The algorithm for solving Eq. (1) iteratively removes portions of $\mathcal{B}$ that contain no solution. To this end, we use the linear equations in Eq. (1), and linearly-relaxed versions of the equations $x_k = x_i x_j$ and $x_k = x_i^2$ (Fig. 2a), in order to define a polytope $\mathcal{P}$ bounding the solution set within $\mathcal{B}$. This box is then reduced by computing the smallest axis-aligned orthotope circumscribing $\mathcal{P}$, using the Simplex method.

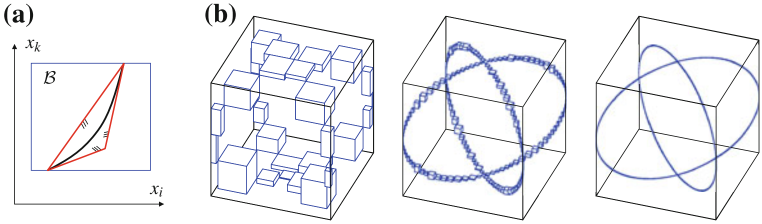


Fig. 2 **a** A linear relaxation is a collection of half-spaces bounding the graph of a given equation within the box $\mathcal{B}$ of interest. The figure shows a relaxation of $x_k = x_i^2$ involving three half-spaces. **b** Progression of the branch-and-prune method on a one-dimensional C-space

The process is repeated until (1) the box is either reduced to an empty set, in which case it contains no solution, or (2) it is sufficiently small, in which case it is returned as a *solution* box, or (3) it cannot be further reduced, in which case it is bisected into two sub-boxes. The whole process is recursively applied to the new sub-boxes, until a fine-enough box approximation of the solution set is finally obtained (Fig. 2b).

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

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

3 Continuation Methods

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

$$\boldsymbol{x}_j^i = \boldsymbol{x}_i + \boldsymbol{\Phi}_i \boldsymbol{u}, \tag{2}$$

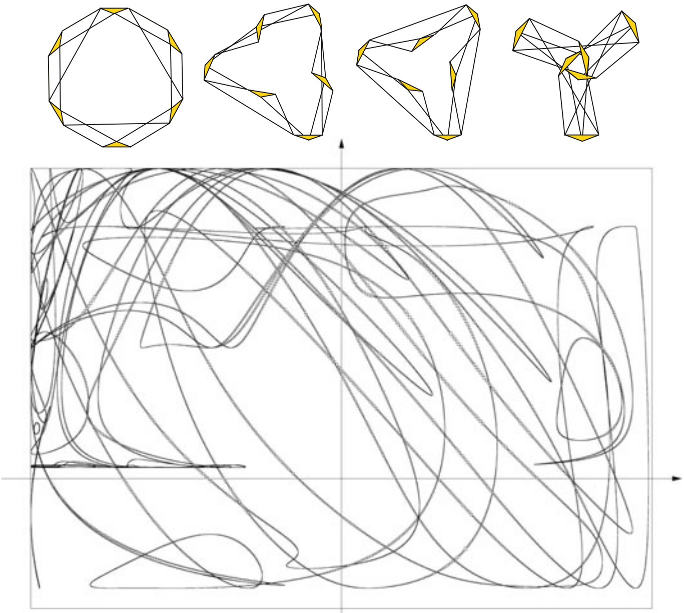


Fig. 3 *Top* The rigid *wheel* linkage on the left involves 10 loops. For the shown bar lengths, the tools of the CUIK suite determine that it can be assembled into 19 different configurations, including the four ones shown herein. *Bottom* The C-space of the wheel linkage with one bar removed, as computed by the mentioned tools. The C-space involves one isolated point and 15 one-dimensional components, which are here projected onto the sines of two link angles. A resultant polynomial describing such curve is expected to be of a very high degree

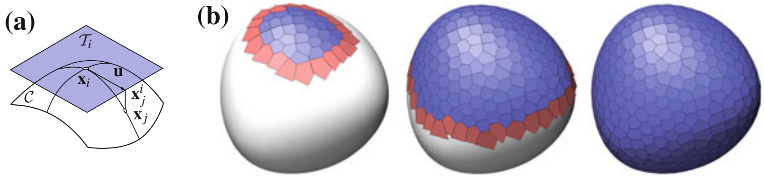


Fig. 4 **a** A chart is used to obtain new C-space points by projecting points from the tangent space. **b** Progress of the atlas construction method on a manifold. *Red polygons* represent the charts to be extended in subsequent iterations

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

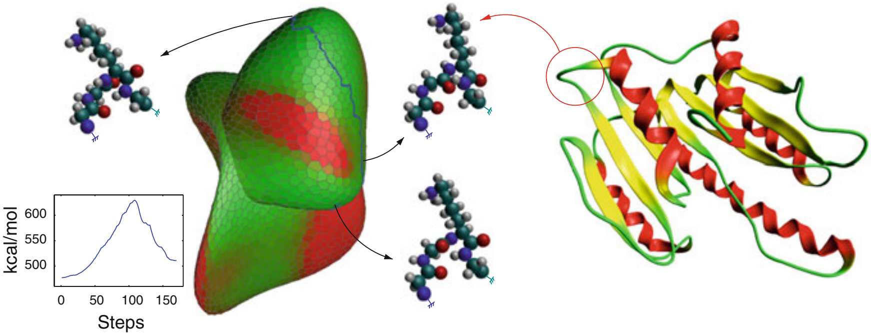


Fig. 5 A low-cost path (*in blue*) computed in the conformational space of a loop of the FTSJ protein of Escherichia Coli (in ribbon diagram on the *right*). The cost is the potential energy of each conformation. The *insets* show the initial conformation, the transition state (i.e., the conformation with the highest potential energy along the path), and the final conformation. Only the atoms in the loop are shown in such conformations. The plot shows the energy profile along the transition path

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

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

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

4 Conclusions

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

Acknowledgments The branch-and-prune methods in the CUIK suite have evolved from years of collaboration with F. Thomas, whose guidance and inspiring comments have shaped the final result. We would also like to express our gratitude to M. Henderson and D. Zlatanov for their feedback and support.

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under project DPI2010-18449, and by Juan de la Cierva and CSIC JAE-Doc fellowships respectively supporting M. Manubens and L. Jaillet.

References

1. Bohigas, O., Henderson, M.E., Ros, L., Manubens, M., Porta, J.M.: Planning singularity-free paths on closed-chain manipulators. IEEE Trans. Rob. **29**(4), 888–898 (2013)
2. Bohigas, O., Manubens, M., Ros, L.: Singularities of non-redundant manipulators: a short account and a method for their computation in the planar case. Mech. Mach. Theory **68**, 1–17 (2013)
3. Bohigas, O., Manubens, M., Ros, L.: A complete method for workspace boundary determination on general structure manipulators. IEEE Trans. Rob. **28**(5), 993–1006 (2012)
4. Bohigas, O., Manubens, M., Ros, L.: Navigating the wrench-feasible C-space of cable-driven hexapods. In: Bruckmann, T., Pott A. (eds.) Cable-Driven Parallel Robots, Mechanisms and Machine Science, vol. 12, pp. 53–68. Springer, Berlin (2013)
5. Bohigas, O., Zlatanov, D., Ros, L., Manubens, M., Porta, J.M.: Numerical computation of manipulator singularities. In: IEEE International Conference on Robotics and Automation, pp. 1351–1358 (2012)
6. Henderson, M.E.: Multiple parameter continuation: computing implicitly defined k-manifolds. Int. J. Bifurcat. Chaos **12**(3), 451–476 (2002)
7. Jaillet, L., Cortés, J., Siméon, T.: Sampling-based path planning on configuration-space costmaps. IEEE Trans. Rob. **26**(4), 635–646 (2010)
8. Jaillet, L., Porta, J.M.: Asymptotically-optimal path planning on manifolds. In: Robotics: Science and Systems, pp. 145–152 (2012)
9. Jaillet, L., Porta, J.M.: Path planning under kinematic constraints by rapidly exploring manifolds. IEEE Trans. Rob. **29**(1), 105–117 (2013)
10. LAAS-CNRS: move 3D. http://www.openrobots.org/wiki/move3d
11. Lavalle, S.: The motion strategy library. http://msl.cs.uiuc.edu/msl
12. Merlet, J.P.: The Alias software. ftp://ftp-sop.inria.fr/coprin/ALIAS

13. Porta, J.M.: CuikSLAM: A Kinematic-based approach to SLAM. In: IEEE International Conference on Robotics and Automation, pp. 2425–2431 (2005)
14. Porta, J.M., Jaillet, L.: Exploring the energy landscapes of flexible molecular loops using higher-dimensional continuation. J. Comput. Chem. **34**(3), 234–244 (2013)
15. Porta, J.M., Jaillet, L., Bohigas, O.: Randomized path planning on manifolds based on higher-dimensional continuation. Int. J. Rob. Res. **31**(2), 201–215 (2012)
16. Porta, J.M., Ros, L., Thomas, F.: A linear relaxation technique for the position analysis of multiloop linkages. IEEE Trans. Rob. **25**, 225–239 (2009)
17. Porta, J.M., Ros, L., Thomas, F., Corcho, F., Cantó, J., Pérez, J.J.: Complete maps of molecular-loop conformational spaces. J. Comput. Chem. **28**(13), 2170–2189 (2007)
18. Rojas, N., Thomas, F.: Closed-form solution to the position analysis of Watt-Baranov trusses using the bilateration method. J. Mech. Rob. **3**(3), 1–10 (2011)
19. Rosales, C., Porta, J.M., Ros, L.: Grasp optimization under specific contact constraints. IEEE Trans. Rob. **29**(3), 746–757 (2013)
20. Rosales, C., Ros, L., Porta, J.M., Suárez, R.: Synthesizing grasp configurations with specified contact regions. Int. J. Rob. Res. **30**(4), 431–443 (2011)
21. Saha, M.: The motion planning kit. http://ai.stanford.edu/~mitul/mpk
22. Sucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. IEEE Rob. Autom. Mag. **19**(4), 72–82 (2012)
23. Vahrenkamp, N.: Simox. http://simox.sourceforge.net
24. Verschelde, J.: The PHC software. http://homepages.math.uic.edu/~jan/PHCpack/phcpack.html
25. Zlatanov, D.S.: Generalized singularity analysis of mechanisms. Ph.D. thesis, University of Toronto (1998)

Spherical Parallel Mechanism with Variable Target Point

Yukio Takeda, Tsuyoshi Ikeda and Daisuke Matsuura

Abstract This paper proposes a position-orientation decoupled parallel mechanism with five degrees of freedom, in which rotational motion of the output link around two axes is controlled by two inputs while translational motion of the target point, the center of rotation of the output link, is controlled by the other three inputs. This mechanism is composed of three connecting chains; one for controlling the position of the target point and two for generating rotational output motion. Conditions of kinematic structures of these chains are discussed and a concrete mechanism is shown. Inverse displacement analysis and Jacobian analysis of this mechanism are carried out to confirm its decoupled feature without encountering the singular point.

Keywords Kinematics · Spherical parallel mechanism · Structural synthesis · Position-orientation decoupled mechanism · Displacement analysis · Singularities

1 Introduction

There are a lot of operations that require precise rotational output motion around two or three axes while the position of the rotation center being changed in a three dimensional space. As such examples, minimum invasive surgery under laparoscope and manufacturing of femoral head of prostheses done by robots are illustrated in Fig. 1. We consider a robot mechanism for such applications. In both cases, the actuators should be remotely located from the operation area in order that they would be protected from the working environment. From the safety point of view, the moving part of the mechanism should be as light as possible. Parallel mechanism is

Y. Takeda (✉) · T. Ikeda · D. Matsuura
Department of Mechanical Sciences and Engineering, Tokyo Institute of Technology, 2-12-1,
O-okayama, Meguro-ku, Tokyo 152-8552, Japan

T. Ikeda
e-mail: takeda@mech.titech.ac.jp

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_18,

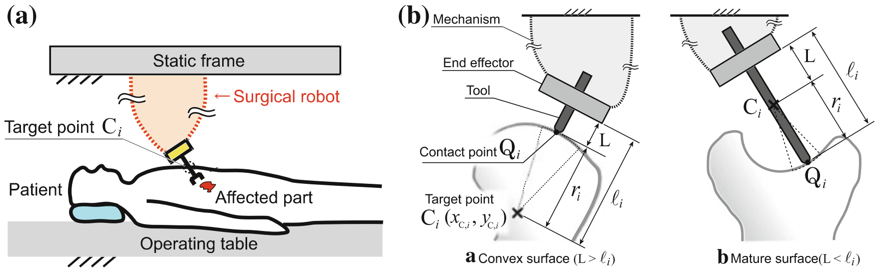


Fig. 1 Target applications. **a** Minimum invasive surgery. **b** Manufacturing of prostheses

considered to be one of the most appropriate candidates for such robot mechanisms, because all the actuators can be located on or close to the base and accurate motion can be achieved even under heavy load condition.

There are a lot of articles regarding kinematic analysis and synthesis of pure-rotational/spherical parallel mechanisms [1, 3, 12, 15]. However, there are a few articles [11, 20] in which design method or proposition of kinematic structure enabling variable position of the center of output rotation in such mechanisms have been discussed. Taking into consideration the simplicity of the control system, position-orientation decoupled parallel mechanism is considered to be one of the optimal choices. In literature, several position-orientation decoupled parallel mechanisms with six degrees of freedom (DOF) having symmetrical structure have been proposed [4, 5, 10, 13, 16, 18]. However, there are a few articles in which mechanisms with asymmetrical structure have been considered [2, 19]. Asymmetrical three-DOF rotational-translational parallel mechanisms [14] and parallel mechanisms generating three-DOF finite translation and two-DOF infinite rotation [9] have been figured out. In the present paper, we propose an asymmetrical position-orientation decoupled parallel mechanism with five DOF. In our previous work, an asymmetrical rotational parallel mechanism has been developed that can perform rotational output motion around two axes and fine translational motion to compensate for position error of the target point, the center of rotation of the output link. However, the actuators for changing the target point's position were located on the output link [11].

The present paper is organized as follows. In Sect. 2, a basic structure of SPMVTP (Spherical Parallel Mechanism with Variable Target Point), one of the asymmetrical position-orientation decoupled parallel mechanisms, is proposed. In Sect. 3, a kinematic structure of SPMVTP is clarified in which translational and rotational output motions are fully decoupled, and its inverse displacement analysis is discussed. In Sect. 4, a numerical example is shown and the effectiveness of the mechanism is discussed.

2 Basic Structure of SPMVTP

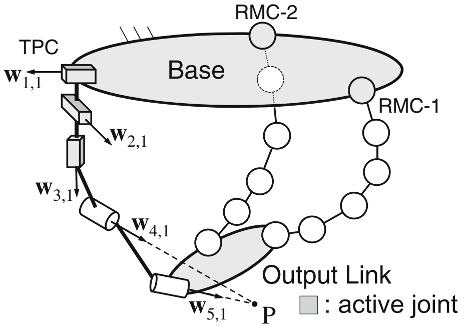
2.1 Mechanism Configuration

Figure 2 shows the basic structure of SPMVTP which is composed of a target point controlling chain (TPC) and two rotational motion generating chains (RMC). In the figure, quadrangular prism and cylinder represent prismatic and revolute joints, respectively. A circle represents a joint with a single DOF of arbitrary type. Types of these joints represented by circles and their axis directions determine the kinematic structure of SPMVTP. The unit vector $\mathbf{w}_{j,i}$ gives the direction either of the rotational motion if $\mathrm{J}_{j,i}$ is a revolute joint or of the translational motion if $\mathrm{J}_{j,i}$ is a prismatic joint, where $\mathrm{J}_{j,i}$ denotes the j-th joint of the i-th chain.

TPC is composed of two passive revolute joints and a translational mechanism with three DOF. In the figure, a serial chain with three active prismatic joints ($\mathrm{J}_{1,1} \sim \mathrm{J}_{3,1}$) is shown as the translational mechanism. The two revolute joints meet at a point P, which is the center of the rotation of the output link, called hereafter “target point”. The translational mechanism determines the position of the target point.

RMC is a serial chain with five passive joints ($\mathrm{J}_{2,i} \sim \mathrm{J}_{6,i}$; $i = 2, 3$) and an active joint ($\mathrm{J}_{1,i}$). Rotational motion around an axis and full translational motion of the output link is constrained by fixing all active joints in TPC. To completely constrain the motion of the output link when all active joints of RMC as well as TPC are fixed, a rotational motion around an axis should be constrained by fixing the active joint of each RMC. In addition to this, the total three constraints with respect to the rotational motion should be linearly independent in order to avoid singularity. Kinematic structures satisfying these conditions are candidates for SPMVTP, in which an output motion composed of three-DOF translational motion and two-DOF rotational motion can be controlled by the five active joints.

Fig. 2 Basic composition of SPMVTP



2.2 Velocity Relationship

For SPMVTP which satisfies with the conditions mentioned in the previous subsection, relationship between the input velocity and output velocity is derived. Input velocity is denoted as $[\dot{\boldsymbol{\theta}}^T \quad \dot{\mathbf{q}}^T]$ ($\dot{\boldsymbol{\theta}} = [\dot{\theta}_{1,2} \quad \dot{\theta}_{1,3} \quad 0]^T, \dot{\mathbf{q}} = [\dot{q}_{1,1} \quad \dot{q}_{2,1} \quad \dot{q}_{3,1}]^T$) where the velocity of $\mathbf{J}_{j,i}$ is denoted by $\dot{\theta}_{j,i}$ (revolute joint) or by $\dot{q}_{j,i}$ (prismatic joint) of RMCs ($i = 2, 3$) and of TPC ($i = 1$), respectively. Output velocity is denoted as $[\boldsymbol{\omega}_{\mathrm{P}}^{\mathrm{T}} \mathbf{v}_{\mathrm{P}}^{\mathrm{T}}]^{\mathrm{T}}$ where $\boldsymbol{\omega}_{\mathrm{P}}$ and $\mathbf{v}_{\mathrm{P}}$ represent the angular velocity and the velocity at P of the output link, respectively. Relationship between the input velocity and output velocity is written as

$$J_{\mathrm{T}} \begin{bmatrix} \boldsymbol{\omega}_P \\ \mathbf{v}_P \end{bmatrix} = \begin{bmatrix} J_A & J_B \\ 0 & J_C \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_P \\ \mathbf{v}_P \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{1}$$

where

$$J_A = \begin{bmatrix} \dfrac{\hat{\mathbf{m}}_2^T}{\hat{\mathbf{m}}_2^T \hat{\boldsymbol{\omega}}_{1,2} + \hat{\mathbf{f}}_2^T \hat{\mathbf{v}}_{1,2}} \\ \dfrac{\hat{\mathbf{m}}_3^T}{\hat{\mathbf{m}}_3^{\mathrm{T}} \hat{\boldsymbol{\omega}}_{1,3} + \hat{\mathbf{f}}_3^{\mathrm{T}} \hat{\mathbf{v}}_{1,3}} \\ (\mathbf{w}_{4,1} \times \mathbf{w}_{5,1})^{\mathrm{T}} \end{bmatrix}, J_{\mathrm{B}} = \begin{bmatrix} \dfrac{\hat{\mathbf{f}}_2^{\mathrm{T}}}{\hat{\mathbf{m}}_2^{\mathrm{T}} \hat{\boldsymbol{\omega}}_{1,2} + \hat{\mathbf{f}}_2^{\mathrm{T}} \hat{\mathbf{v}}_{1,2}} \\ \dfrac{\hat{\mathbf{f}}_3^{\mathrm{T}}}{\hat{\mathbf{m}}_3^{\mathrm{T}} \hat{\boldsymbol{\omega}}_{1,3} + \hat{\mathbf{f}}_3^{\mathrm{T}} \hat{\mathbf{v}}_{1,3}} \\ \mathbf{0}_3 \end{bmatrix}, J_{\mathrm{C}} = \begin{bmatrix} \mathbf{w}_{1,1}^{\mathrm{T}} \\ \mathbf{w}_{2,1}^{\mathrm{T}} \\ \mathbf{w}_{3,1}^{\mathrm{T}} \end{bmatrix}, \tag{2}$$

respectively. $\mathbf{S}_{1,i} = [\hat{\boldsymbol{\omega}}_{1,i}^{\mathrm{T}} \hat{\mathbf{v}}_{1,i}^{\mathrm{T}}]^{\mathrm{T}} (i = 2, 3)$ and $\mathbf{S}_{\mathrm{R}A,i} = [\hat{\mathbf{f}}_i^{\mathrm{T}} \hat{\mathbf{m}}_i^{\mathrm{T}}]^{\mathrm{T}} (i = 2, 3)$ represent the joint screw of the active joint and the constraint wrench imposed by fixing the active joint of RMC, respectively.

Equations for forward velocity calculation is obtained as

$$\left.\begin{aligned} \boldsymbol{\omega}_{\mathrm{P}} &= J_{\mathrm{A}}^{-1}(\dot{\boldsymbol{\theta}} - J_{\mathrm{B}} J_{\mathrm{C}}^{-1} \dot{\mathbf{q}}) \\ \mathbf{v}_{\mathrm{P}} &= J_{\mathrm{C}}^{-1} \dot{\mathbf{q}} \end{aligned}\right\} . \tag{3}$$

It is known from the equation that input motion of RMC generates pure rotational motion while input motion of TPC generates rotational motion coupled with translational motion of the output link. This means that rotational output motion is decoupled from translational motion in SPMVTP regardless of the kinematic structure of RMC.

3 Structure of Fully Decoupled SPMVTP

3.1 Kinematic Structure of RMC

Let us consider a case of $J_{\mathrm{B}} = 0$ in Eq. (1). In such a case, the velocity equation becomes

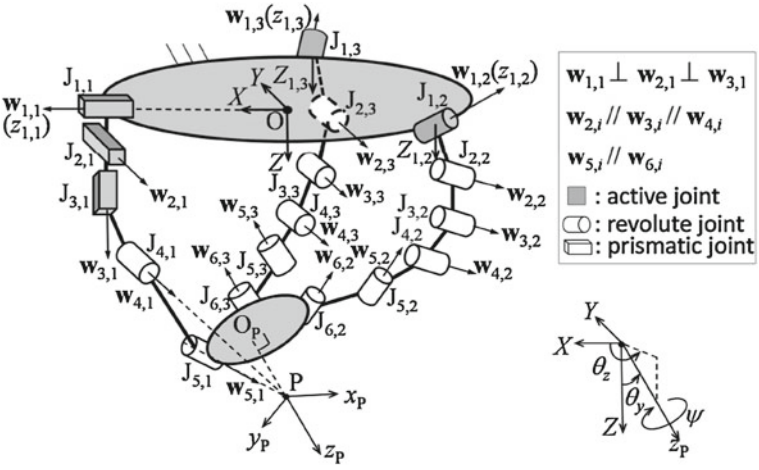


Fig. 3 An example of fully decoupled SPMVTP

$$\left.\begin{aligned}\boldsymbol{\omega}_{\rm P} &= J_{\rm A}^{-1}\dot{\boldsymbol{\theta}}\\ \mathbf{v}_{\rm P} &= J_{\rm C}^{-1}\dot{\mathbf{q}}\end{aligned}\right\} . \tag{4}$$

Mechanisms satisfying this condition are called *fully decoupled SPMVTP*. In this section, its kinematic structure is investigated. Singularity of fully decoupled SPMVTP can be investigated by the determinants of sub-matrices $J_{\rm A}$ and $J_{\rm C}$. If the kinematic structure of TPC shown in Fig. 2 is employed, det $J_{\rm C}$ is always one. In what follows, the singularity defined as det $J_{\rm A} = 0$ will be considered.

From Eq. (1), constraint screw of RMC with its active joint fixed for fully decoupled SPMVTP should be in the form of $\mathbf{S}_{\rm RA,i} = [\mathbf{0}^{\rm T}\ \hat{\mathbf{m}}_{\rm i}^{\rm T}]^{\rm T}({\rm i} = 2, 3)$. This means that thanks to the constraint imposed by the two RMCs with their inputs being fixed, the orientation of the output link does not change by the input motion of TPC for changing the position of target point.

Starting from the kinematic structures for translational parallel mechanism (TPM) with three serial connecting chains, we figured out kinematic structures of RMC for fully decoupled SPMVTP. The structural conditions for the connecting chain of TPM are summarized in [7, 8, 17]. Kinematic structures of RMC for fully decoupled SPMVTP can be obtained by adding a revolute joint at the base to the kinematic chains for TPM so that the conditions for connecting chain of TPM are satisfied even when the added revolute joint is arbitrarily positioned.

An example of fully decoupled SPMVTP is shown in Fig. 3 in which six revolute joints are used as RMC. Kinematic structure and dimensions of RMC should be determined so that the following conditions are satisfied.

1. The axes of the revolute joints $\mathbf{w}_{2,\rm i}$ to $\mathbf{w}_{4,\rm i}$ should be parallel.
2. The axes of the revolute joints $\mathbf{w}_{5,\rm i}$ and $\mathbf{w}_{6,\rm i}$ should be parallel while the axes of joints $\mathbf{w}_{4,\rm i}$ and $\mathbf{w}_{5,\rm i}$ should not be parallel.

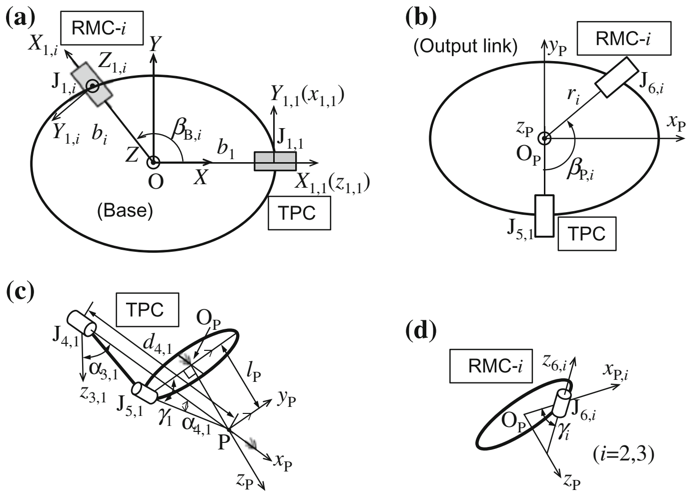


Fig. 4 Definition of kinematic constants. **a** Location of joints on base. **b** Location of joints on output link. **c** Location of revolute joints of TPC. **d** Location of revolute joints of RMC

3. The axes of the revolute joints $\mathbf{w}_{1,\mathrm{i}}$ and $\mathbf{w}_{2,\mathrm{i}}$ should not be parallel to avoid architectural singularity of RMC [6]
4. Rank of the Jacobian matrix with respect to RMC as a serial chain should be 6.
5. Rank of the sub-matrix J_A should be 3.

The orientation angles $(\theta_y, \theta_z, \psi)$ shown in Fig. 3 are used to represent the orientation of the output link. The kinematic constants are defined as shown in Fig. 4. Other kinematic constants are defined using DH parameters (a, d, α and θ are link length, offset length, twist angle, and rotation angle, respectively).

3.2 Inverse Displacement Analysis of SPMVTP

Pose (position of P and orientation) of the output link is represented by a 4 × 4 transformation matrix T_P as

$$T_P = \begin{bmatrix} 1 & 0\ 0\ 0 \\ \mathbf{p} & R \end{bmatrix} \tag{5}$$

where $\mathbf{p}$ and $R = [\mathbf{e_x}\, \mathbf{e_y}\, \mathbf{e_z}]^T$ are the position vector of P and the 3 × 3 rotation matrix representing the orientation, respectively. Transformation matrix can be written as a function of kinematic constants and joint variables of each chain. Those with respect to TPC and RMCs are denoted as $T_{P,TPC}$ and $T_{P,RMC,i}$, respectively. Inverse displacement analysis of SPMVTP is defined as a problem to solve the following equation with respect to $\theta_{1,i}$ and $q_{j,1} (i = 2, 3;\ j = 1, 2, 3)$ for a given pose

$(X_\mathrm{P}, Y_\mathrm{P}, Z_\mathrm{P}, \theta_y, \theta_z)$.

$$T_\mathrm{P} = T_\mathrm{P,TPC} = T_{\mathrm{P,RMC},i} (i = 2, 3) \tag{6}$$

However, since SPMVTP is a mechanism with five DOF, T_P is not fully defined by the given output pose. Then, the following procedure for the inverse displace-ment analysis of SPMVTP has been developed.

1. Since $\mathbf{e_z}$ is a function of θ_y and θ_z, passive joint displacements of TPC $\theta_{4,1}$ and $\theta_{5,1}$ can be obtained by solving $\mathbf{e_z} = \mathbf{e}_{\mathbf{z},\mathrm{TPC}}$. There are two solutions.
2. Input displacements $q_{j,1} (j = 1, 2, 3)$ can be uniquely obtained from

$$\mathbf{p} = \sum_{j=1}^{3} q_{j,1} \mathbf{w}_{j,1}. \tag{7}$$

3. T_P can be fully determined by $T_\mathrm{P,TPC} \rightarrow T_\mathrm{P}$ using the results of steps 1 and 2.
4. Solutions of $\theta_{1,i} (i = 2, 3)$ can be obtained by solving $T_\mathrm{P} = T_{\mathrm{P,RMC},i}$. There are eight real solutions at maximum for each RMC.

4 Numerical Example

For a mechanism with kinematic constants shown in Table 1, inverse displacement analysis has been done following the procedure described in the previous section. Results for the following two cases of the output motion are shown in Fig. 5.

1. $(X_\mathrm{P}, Y_\mathrm{P}, Z_\mathrm{P}) = (0, 0, 130)$[mm], $\theta_y = 30°$, $\theta_z = [0 : 360]°$
2. $X_\mathrm{P} = [-20 : 20]$[mm], $(Y_\mathrm{P}, Z_\mathrm{P}) = (0, 130)$[mm], $\theta_y = 30°$, $\theta_z = 0$

Table 1 Kinematic constants of mechanism

Symbols	Values	Symbols	Values	Symbols	Values
l_P	6.84 mm	$\beta_{\mathrm{B},3}$	135°	$d_{1,i}$	50 mm
b_1	60 mm	$\beta_{\mathrm{P},2}$	45°	$a_{2,i}$	50 mm
$\alpha_{3,1}$	90°	$\beta_{\mathrm{P},3}$	225°	$a_{3,i}$	60 mm
$d_{4,1}$	60 mm	b_i	110 mm	$a_{4,i}$	10 mm
$\alpha_{4,1}$	90°	$\alpha_{0,i}$	45°	$a_{5,i}$	40 mm
γ_1	20°	$\theta_{0,i}$	270°	γ_i	30°
$\beta_{\mathrm{B},2}$	135°	$a_{1,i}$	0 mm	r_i	35 mm

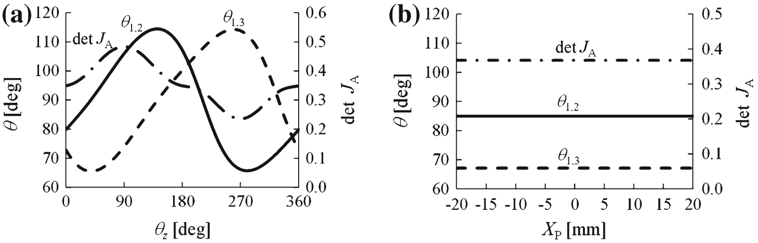


Fig. 5 Input displacement of RMC and det J_A **a** case-I, **b** case-II

In these figures, determinant of the sub-matrix J_A is also shown. From the figures, it is known that the proposed mechanism can achieve a fully-decoupled output motion without encountering singularity.

5 Conclusions

In the present paper, an asymmetrical five-DOF fully-decoupled parallel mechanism has been proposed and its kinematic study has been carried out. Our conclusions are summarized as follows.

1. A kinematic structure of spherical parallel mechanism with variable target point, which is composed of a target point controlling chain and two rotational motion generating chains, has been figured out.
2. A procedure for inverse displacement analysis of the mechanism has been clarified.
3. Effectiveness of the mechanism has been confirmed through a numerical example of inverse displacement analysis with a check of singularity.

References

1. Fang, Y., Tsai, L.W.: Structure synthesis of a class of 3-dof rotational parallel manipulators. IEEE Trans. Rob. Autom. **20**(1), 117 (2004)
2. Geng, Z.J., Haynes, L.: A "3-2-1" kinematic configuration of a stewart platform and its application to six degrees of freedom pose measurements. Robot. Comput. Integr. Manuf. **11**(1), 23 (1994)
3. Gosselin, C.M., Angeles, J.: The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator. Trans. ASME J. Mech. Transm. Autom. Design **111**(2), 202 (1989)
4. Innocenti, C., Parenti-Castelli, V.: Direct kinematics of the 6–4 fully parallel manipulator with position and orientation uncoupled. In: European Robotics and Intelligent Systems Conference, pp. 3 (1991)

5. Jin, Y., Chen, I.M., Yang, G.: Kinematic design of a 6-dof parallel manipulator with decoupled translation and rotation. IEEE Trans. Rob. **22**(3), 545 (2006)
6. Karger, A.: Classification of robot-manipulators with only singular configurations. Mech. Mach. Theory **30**(5), 727 (1995)
7. Kim, D., Chung, W.K.: Kinematic condition analysis of three-dof pure translational parallel manipulators. Trans. ASME J. Mech. Design **125**(2), 323 (2003)
8. Kong, X., Gosselin, C.M.: Type synthesis of 3-dof translational parallel manipulators based on screw theory. Trans. ASME J. Mech. Design **126**(1), 83 (2004)
9. Lee, C.C., Herve, J.M.: Parallel mechanisms generating 3-DoF finite translation and (2 or 1)-DoF infinitesimal rotation. Mech. Mach. Theory **51**(1), 185 (2012)
10. Mianovski, K.: Dexterous fully parallel manipulator with six degrees of freedom. In: Proceedings of RoManSy 12, pp. 253 (1998)
11. Okamura, J., Hanagasaki, S., Takeda, Y.: Kinematic synthesis of two-dof rotational parallel mechanism with compensation for position error. In: Proceedings of 2nd IFToMM International Symposium on Robotics and Mechatronics (2011)
12. Ouerfelli, M., Kumar, V.: Optimization of a spherical five-bar parallel drive linkage. Trans. ASME J. Mech. Design **116**, 166 (1994)
13. Patarinski, S., P., Uchiyama, M.: Analysis and design of position/orientation decoupled parallel manipulators. In: Proceedings of RoManSy 10, 219(1995)
14. Refaat, S., Herve, J.M., Nahavandi, S., Trinh, H.: Asymmetrical three-DOFs rotational-translational parallel-kinematics mechanisms based on Lie group theory. Eur. J. Mech. A/Solids **25**(3), 550 (2006)
15. Takeda, Y., Funabashi, H., Sasaki, Y.: Development of a spherical in-parallel actuated mechanism with three degrees of freedom with large working space and high motion transmissibility (evaluation of motion transmissibility and analysis of working space). JSME Int. J. Series C **39**(3), 541(1996)
16. Takeda, Y., Kamiyama, K., Maki, Y., Higuchi, M., Sugimoto, K.: Development of position-orientation decoupled spatial in-parallel actuated mechanisms with six degrees of freedom. J. Rob. Mechatron. **17**(1), 59 (2005)
17. Tanabe, M., Takeda, Y.: Kinematic design of a translational parallel manipulator with fine adjustment of platform orientation. Adv. Mech. Eng. 485358 (2010)
18. Wohlhart, K.: Displacement analysis of the general spherical stewart platform. Mech. Mach. Theory **29**(4), 581(1994)
19. Zlatanov, D., Dai, M.Q., Fenton, R.G., Benhabib, B.: Mechanical design and kinematic analysis of a three-legged six degree-of-freedom parallel manipulator. In: Proceedings of 22nd Biennial Mechanisms Conference. DE-45, 529(1992)
20. Zlatanov, D., Gosselin, C.M.: A family of new parallel architectures with four degrees of freedom. In: Proceedings of 2nd Workshop on Computational Kinematics, 57 (2001)

A Blend of Delassus Four-Bar Linkages

Chung-Ching Lee and Jacques M. Hervé

Abstract In 1922, Delassus found out three four-bar linkages with four parallel helical H pairs whose one-degree-of-freedom mobility is conditioned by equalities of bar lengths. Our paper establishes that these linkages are not independent. Two isosceles triangle HHHPs are hybridized to obtain the Delassus 4H rhomboid (kite) linkage. Using an auxiliary chain whose mobility is explained by a group of Schoenflies motions and a Delassus 4H parallelogram, the Delassus 4H crossed parallelogram is newly derived from this rhomboid. It is further verified that the isosceles triangle and the Delassus parallelogram are two basic linkages and that the rhomboid and the crossed parallelogram stem from them. Finally, a blend of Delassus 4-bar linkages is proposed and can be used as a basic building block (BBK) for deployable structures. Two examples of deployable linkages with four and six BBKs are introduced.

Keywords Delassus linkage · Isosceles triangle · Rhomboid · Parallelogram · Schoenflies motion · Deployable linkage.

1 Introduction

Seeking after all the four-bar linkages implementing lower pairs, Delassus [1] discriminated two main categories of mobility, namely ordinary (ordinaire in French) and singular mobility. In four-bar linkages, the Degree of Freedom (DoF) may be not modified if a link is cut in two parts which are welded after a relative motion of the parts. When any cut-and-weld operation does not alter the mobility, all the relative motions are subsets of only one subgroup of the rigid-body displacement group

C.-C. Lee (✉)
National Kaohsiung University of Applied Sciences, Kaohsiung 80782, Taiwan
e-mail: cclee@cc.kuas.edu.tw

J. M. Hervé
Ecole Centrale des Arts et Manufactures, Paris, France
e-mail: jacques.herve07@orange.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_19,

and the linkages are “trivial” (or banal) chains in the classification by displacement subgroups [2]. In exceptional chains, the effect of a cut-and-weld operation depends on the choice of the link. In “singular” linkages of Delassus, the mobility is affected by a cut-and-welding of any one of its four links. The singular chains of Delassus are qualified as paradoxical. The paradoxical mobility is subject to geometric conditions that require the use of the Euclidean metric [3]. In trivial and exceptional chains, only concepts of 3D affine geometry are sufficient to stipulate the geometric constraints so the Euclidean metric is not necessarily employed.

Page 304 in Delassus’ paper [1] summarizes the findings that there exist only five singular four-bar chains with one-DoF finite mobility. They are

1. The rhomboid (deltoid or kite) with parallel screw pairs at its four vertices; the pitches of the screws located on the diagonal axis are equal to each other and are equal to the arithmetic mean of the remaining two. If $h_i (i = 1, 2, 3, 4)$ denote the pitches, we have: $h_1 = h_3 = (h_2 + h_4)/2$.
2. The parallelogram with four parallel screws; the sums of pitches of diagonally opposite screws are equal: $h_1 + h_3 = h_2 + h_4$.
3. The crossed parallelogram with four parallel screws; the pitches of alternate screws are equal: $h_1 = h_3$ and $h_2 = h_4$.
4. Two parallel screws with equal pitches and two rectilinear sliding pairs bilaterally symmetric with respect to the plane containing the screw axes.
5. The Bennett linkage.

The first four Delassus linkages implement helical H pairs (or screws) with parallel axes. Hereinafter, they are called Delassus four-bar linkages. The relative motions between the links are 1-DoF motions included in the 4-D subgroup of Schoenflies motions whose axis direction is parallel to the screw axes. The 1-DoF mobility is subject to geometric conditions that imply scalar equation tying the link lengths and the pitches of screw pairs. The last one is the only movable four-revolute chain with non-parallel and non-intersecting revolute (R) joint axes. It was first discovered by G. T. Bennett in 1903 [4] and E. Borel [5] also found it independently only one year later. This linkage and the fourth one are beyond the scope of our article and we focus just on the first three chains.

It is worth mentioning that recently a new paradoxical four-bar linkage was disclosed in [6]. It is an isosceles triangle of structural type HHHP. The H pairs have parallel axes; the two HH links have equal lengths and the prismatic P pair is parallel to the triangle side with a variable length. The three H pitches satisfy the equality $h_1 + h_3 = 2h_2$. In this HHHP chain, a relative motion is a helical Cardan motion as explained in [7].

We notice that the 1-DoF mobility in the 4H rhomboid and the 4H crossed parallelogram is conditioned by two equalities of H pitches whereas the 4H parallelogram and the HHHP isosceles triangle have to satisfy only one pitch equality. In what follows, we will show that the two 4H linkages with two pitch equalities can be derived from the isosceles triangle and the parallelogram with one pitch equality.

Firstly, we hybridize two isosceles triangle HHHP linkages to obtain the rhomboid (kite) 4H linkage. Secondly, in the rhomboid 4H loop, we add two links to form a second loop, which is a special Delassus parallelogram. The third loop appearing in the combination is the general crossed parallelogram of Delassus. In a similar way, we can construct a second crossed parallelogram. Finally, we obtain a blend of four-bars linkages with paradoxical 1-DoF mobility. Actually, the blend of two rhomboids produces a basic building block, which can be used as a module to construct complex deployable structures.

2 Isosceles Triangle HHHP Linkage and Its Helical Cardan Motion

In an isosceles triangle HHHP linkage, the three pitches are tied by one scalar equality, namely $h_1 + h_3 = 2h_2$. Hence, each of the three pitches is a function of two independent numbers. It is convenient to adopt the numbers p and q to verify $h_1 = q + p$, $h_2 = q + p/2$ and $h_3 = q \Leftrightarrow h_1 = q + p, h_1 - h_3 = p$ and $2h_2 = 2q + p$. Using the numbers p and q provides a concise classification of the types of isosceles triangles. When $p = 0$, the three H pitches are equal and the 1-DoF mobility of the HHHP linkage is not conditioned by the equality of the two HH link lengths; in other words, the triangle is not necessarily isosceles. Therefore, the paradoxical mobility which is assumed in the paper implies $p \neq 0$ and q can have any value.

When $p = q = 0$, the H pairs are revolute R pairs and the corresponding isosceles triangle RRRP linkage is planar. As it is well known, if one body of the P pair is assumed to be fixed, then one RR link rotates around a fixed axis and the motion of the other RR link is Cardanic. When $p \neq 0, \forall q$, the isosceles triangle generates a noteworthy kind of motion named helical Cardan motion, which is presented for the first time in [6, 7]. This new helical Cardan motion can be regarded as a spatial generalization of the Cardanic movement on a plane, shown in Fig. 1.

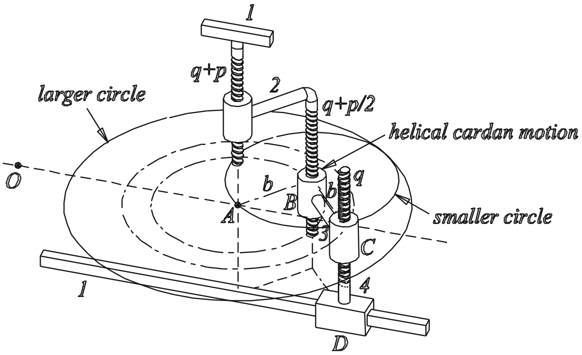


Fig. 1 Isosceles triangle HHHp linkage and 1-DoF helical Cardan motion

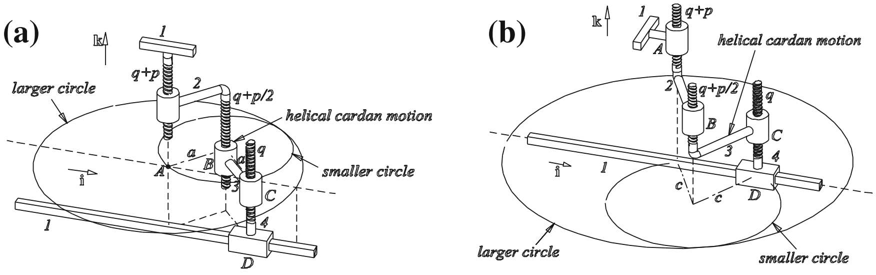


Fig. 2 Two *HHH*P chains with the same pitches **a** with length a, **b** with length c

3 Formation of Delassus Rhomboid 4H Linkage

From two HHHP chains of the helical Cardan motion, we geometrically derive the rhomboid (deltoid or kite) 4H Delassus chain. The two isosceles triangles are not generally congruent but the homologous H pairs have equal pitches. Using self-explanatory figures, we can identify two distinct HHHP chains in Fig. 2. The two chains have the same screw pitch $q + p$ in the H pair of links 1 and 2 and the same screw pitch q in the H pair of links 3 and 4. The pitch of the screw pair between links 2 and 3 is assumed to be $q + p/2$. In one chain, the bars 2 and 3 have the same length a; in the other chain, each length of bars 2 and 3 is c. These two distinct HHHP linkages can be merged in such a way that the resulting mechanism is the rhombus HHHH linkage. In the first step of the synthesis, two HHHP chains share a common P pair, as shown in Fig. 3. Moreover, the two axes of the Hs with the pitch $q + p$ coincide and the other two H pairs with the pitch q are also coaxial. In Fig. 3, actually, two coaxial Hs with the same pitch are equivalent to one H once we ignore the internal helical self-motion of the intermediate body. Consequently, the links 2 and 2 as well as the links 3 and 3 can be connected by only one helical pair with the same pitch. Furthermore, removing the P pair leads to a rhomboid HHHH chain, as shown in Fig. 4. When p is equal to zero, all pitches are equal to q and the HHHH chain is movable whatsoever bar lengths are; the mobility of the chain can be derived using only the closure of the product in the pseudoplanar motion group [8]. When p is not zero, the movability is subject to metric conditions and is qualified as paradoxical. The special case with $q = 0$ and $p \neq 0$ gives an HHHR paradoxical chain.

4 New Derivation of Crossed Parallelogram and a Blend of Delassus 4-Bar Linkages

In a Delassus 4H rhomboid, we add two bars, which are jointed by a cylindrical C pair whose axis is parallel to H axes. Each of the two H pairs with the same pitch

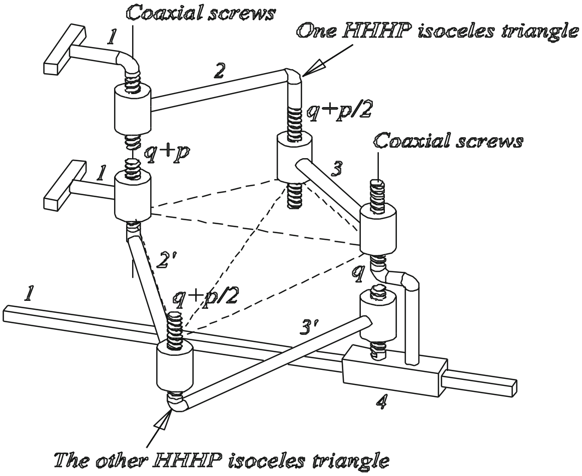
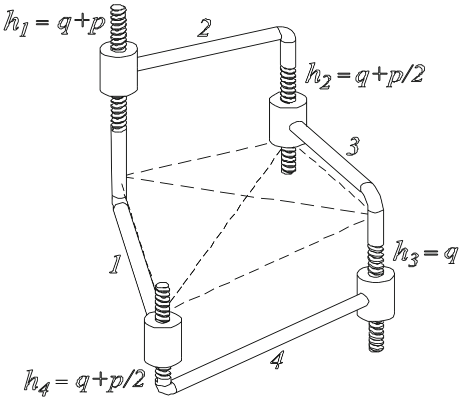


Fig. 3 Two *HHH*P chains

Fig. 4 Rhomboid *HHHH* chains



$q + p/2$ in the rhomboid forms a double H of pitch $q + p/2$ with each of the two added bars. With the added bars, as shown in Fig. 5, the resulting chain includes three closed loops. One loop is the original rhomboid and the other two have an HHHC structural type. Any HHHC chain with four parallel axes is a 1-DoF trivial chain for the Schoenflies motion group [9] whose axis direction is parallel to those of H axes. Consequently, the whole chain is movable with one DoF. Moreover, we assume also that, in one of the HHHC subchains, the two CH bars are parallel to two adjacent bars of the rhomboid. That way, an HHHC subchain forms a parallelogram. Then, the C pair can be replaced by an H pair with the pitch of the coinciding Delassus 4H parallelogram [10]. That pitch is computed by using the pitch equality of a Delassus parallelogram. In fact, we have two choices for constructing an HHHC parallelogram. In one choice, the H pitches of the HHHC parallelogram are $q + p/2$,

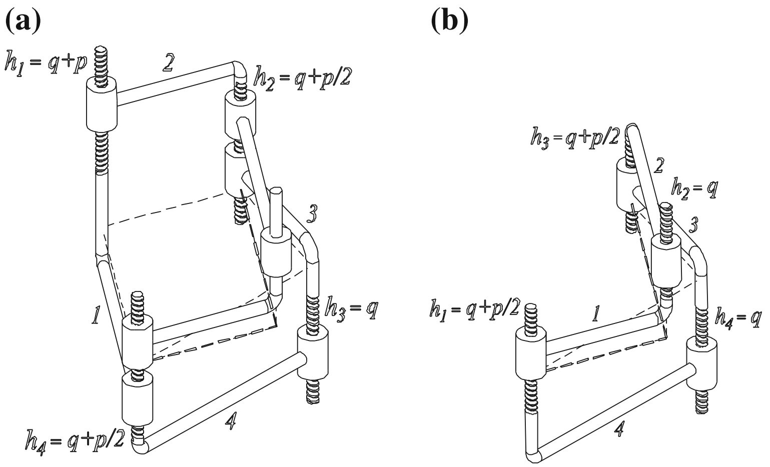


Fig. 5 Forming HHHC subchain. **a** Multiloop chain and **b** derived crossed parallelogram

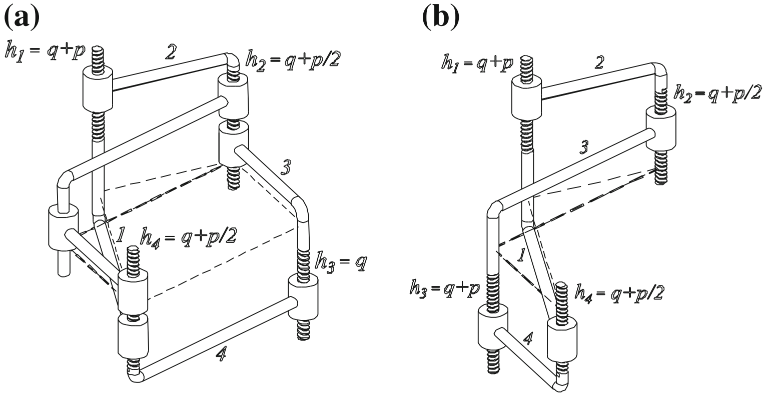


Fig. 6 Second crossed parallelograms. **a** Multiloop chain and **b** derived crossed parallelogram

$q + p$ and $q + p/2$ and the C pair in Fig. 5a can be replaced by an H with the pitch q. A 4H crossed parallelogram appears as being a subchain of the whole chain and consequently moves also with one DoF. In two opposite Hs, the H pitches are equal, which is expressed by two independent equalities, namely $h_2 = h_4 \Leftrightarrow q = q$ and $h_1 = h_3 \Leftrightarrow q + p/2 = q + p/2$ with the notations of Fig. 5b. In the other choice, as shown in Fig. 6a, the H pitches of the HHHC parallelogram are $q + p/2$, q and $q + p/2$ and the C pair can be replaced by an H with the pitch $q + p$. A crossed parallelogram appears as being a subchain which moves also with one DoF. In two opposite Hs, the H pitches are equal, which is expressed by two independent equalities, namely $h_1 = h_3 \Leftrightarrow q + p = q + p$ and $h_2 = h_4 \Leftrightarrow q + p/2 = q + p/2$, Fig. 6b. The crossed parallelogram of Delassus is derived from two 1-DoF chains,

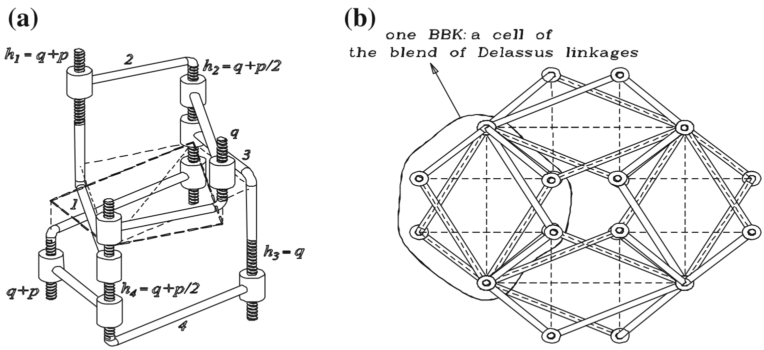


Fig. 7 A blend of Delassus 4-bar linkages and its deployable structure. **a** A blend of linkages and **b** the four-cell deployment

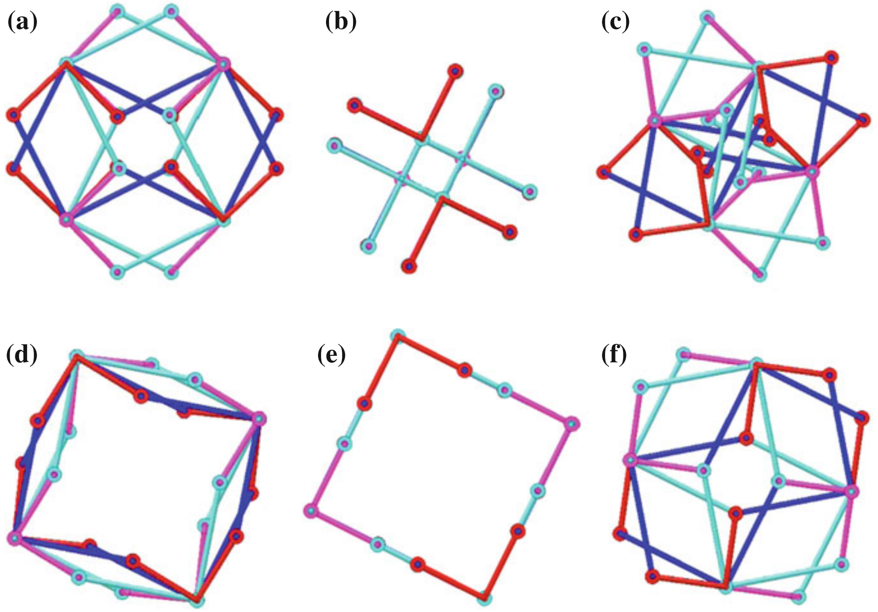


Fig. 8 Different configurations of four-cell deployment

the Delassus 4H kite and the Delassus 4H parallelogram. Hence, it is a combination of two HHHP isosceles triangles together with a Delassus parallelogram. In other words, the HHHP isosceles triangle and the Delassus 4H parallelogram imply the existence of the 4H kite and the 4H crossed parallelogram. Furthermore, we have synthesized a blend of four-bar linkages, in which each loop has a paradoxical 1-DoF mobility as shown in Fig. 7a (isosceles triangles HHHP loops are not depicted). This multiloop linkage can be regarded as a basic building-block (BBK) to construct

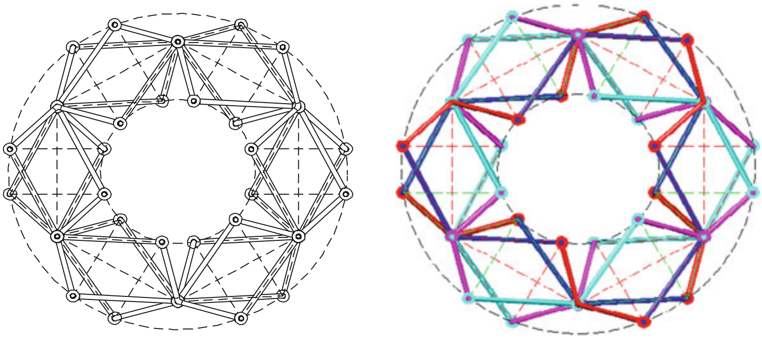


Fig. 9 A deployable ring structure

complex deployable structures. Figure 7b shows this kind of deployment by utilizing four BBKs and its distinct postures are displayed in Fig. 8. Figure 9 illustrates the deployable ring structure composed of six BBKs.

5 Conclusions

The HHHP isosceles triangle and the Delassus 4H parallelogram are basic linkages. Both linkages imply the existence of the 4H kite and the 4H crossed parallelogram. It also shows that two Delassus four-bar linkages and the HHHP triangle are closely related. A blend of Delassus 4-bar linkages can further serve as a module in the construction of different-shaped deployable structures with screw joints. In these deployable structures, the pitches of all screw pairs can be reduced to being zeros. In practical application, deployment devices with revolute pairs are easier to manufacture than the ones with screw pairs but the former are suffering from the singularity when three revolute axes are coplanar. The latter can avoid the singular poses by the choice of their pitches. Hence, deployable devices with screw pairs may be more attractive for potential application.

Acknowledgments The authors are very thankful to the National Science Council for supporting this research under grants NSC 101-2221-E-151-017.

References

1. Delassus, E.: Les chains articulées fermées et déformables á quatre membres. Bull. Sci. Math. **46**(2), 283–304 (1922)
2. Hervé, J.M.: Analyse structurelle des mécanismes par groupe des déplacements. Mech. Mach. Theory **13**(4), 437–450 (1978)

3. Lee, C.-C., Hervé, J.M.: Synthesize new 5-bar paradoxical chains via the elliptic cylinder. Mech. Mach. Theory **46**(6), 784–793 (2011)
4. Bennett, G.T.: A new mechanism. Engineering **76**, 777–778 (1903)
5. Borel, E.: Mémoires des savants étrangers: Mémoire sur les déplacements á trajectories sphériques, Comptes Rendus, Tome.XXXIII, No.1, p. 56 (1904)
6. Hervé, J.M.: A new four-bar linkage completing Delassus findings. Trans. Can. Soc. Mech. Eng. **35**(1), 57–62 (2011)
7. Lee, C.-C., Hervé, J.M.: On the helical Cardan motion and related paradoxical chains. Mech. Mach. Theory **52**, 94–105 (2012)
8. Lee, C-C., Hervé, J.M.: Cartesian parallel manipulators with pseudoplanar limbs. ASME Trans. J. Mech. Design **129**(12), 1256–1264 (2007)
9. Lee, C.-C., Hervé, J.M.: Type synthesis of primitive Schoenflies-motion generators. Mech. Mach. Theory **44**(10), 1980–1997 (2009)
10. Lee, C.-C., Hervé, J.M.: On the Delassus parallelogram. In: Advances in Robot Kinematics, pp. 439–449, Springer Science, ISBN 978-1-4020-8599-4 (2008)

On the Symmetric Molecular Conjectures

Josep M. Porta, Lluis Ros, Bernd Schulze, Adnan Sljoka and Walter Whiteley

Abstract A molecular linkage consists of a set of rigid bodies pairwise connected by revolute hinges where all hinge lines of each body are concurrent. It is an important problem in biochemistry, as well as in robotics, to efficiently analyze the motions of such linkages. The theory of generic rigidity of body-bar frameworks addresses this problem via fast combinatorial algorithms. However, recent work has shown that symmetry (a common feature of many molecular and mechanical structures) can lead to additional motions. These motions typically maintain the original symmetry of the structure throughout the path, and they can often be detected via simple combinatorial counts. In this paper, we outline how these symmetry-based mathematical counts and methods can be used to efficiently predict the motions of symmetric molecular linkages, and we numerically analyze configuration spaces supporting the symmetric Molecular Conjectures formulated herein.

Keywords Rigidity · Flexibility · Symmetry · Molecular linkage · Configuration space

J. M. Porta (✉) · L. Ros
Institut de Robòtica i Informàtica Industrial, UPC-CSIC, Barcelona, Spain
e-mail: porta@iri.upc.edu

L. Ros
e-mail: ros@iri.upc.edu

B. Schulze
Department of Mathematics and Statistics, University of Lancaster, Lancaster, UK
e-mail: b.schulze@lancaster.ac.uk

A. Sljoka · W. Whiteley
Department of Mathematics and Statistics, York University, Toronto, ON, Canada
e-mail: adnanslj@mathstat.yorku.ca

W. Whiteley
e-mail: whiteley@mathstat.yorku.ca

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_20,

1 Introduction

Accurate measurements of flexibility and dynamics of proteins and other molecules can help us interpret the relationship between structure and function, which has significant implications in medicine and drug design [3, 7]. This is an important area of research in computational biology, material science and bioinformatics, and has lately attracted a lot of interest in the robotics community [12].

Over the last 15 years, rigidity-based methods like FIRST, ProFlex, or Kinari [7, 8] have been used to give fast computational predictions of the flexibility of molecular structures, such as DNA, RNA, and proteins. These methods detect rigidity or flexibility in generic molecular linkages using combinatorial (counting) characterizations of rigidity, which can be verified via fast pebble game algorithms [10, 16]. They are based on the mathematical theory of generic rigidity of molecular linkages (Sect. 2), which are built of rigid bodies (atoms) linked in pairs via revolute hinges (rotatable bonds between the atoms) [7, 21].

However, recent work by Schulze et al. [13–15] shows that symmetry can lead to additional flexibility that is not detected by the more general characterizations. The point is crucial because many molecular rings, proteins, or viral capsids are symmetric and symmetry plays a central role in macromolecular stability, assembly, ligand recognition and drug docking, and in phenomena such as allostery [4, 16].

In this paper, we demonstrate how this increased flexibility can still be detected through simple symmetry-adapted counts, and then we formulate the symmetric molecular conjectures of rigidity theory (Sect. 3). To test the power of the counts, and to provide evidence supporting the conjectures, we use the higher-dimensional continuation tools of the CUIK suite [6, 11] on illustrative examples (Sect. 4).

2 Detecting Flexibility in Body-Bar and Molecular Linkages

A 3-dimensional *body-bar framework* consists of a set of rigid bodies in $\mathbb{R}^3$ which are connected by rigid bars (see Fig. 1, left). Each body is free to move continuously in $\mathbb{R}^3$, subject to the constraints that the distance between any pair of points which are connected by a bar remains fixed. A body-bar framework is called *rigid* if *every* such motion also preserves the distance between *all* pairs of points belonging to different bodies. Otherwise the framework is called *flexible* [20, 21].

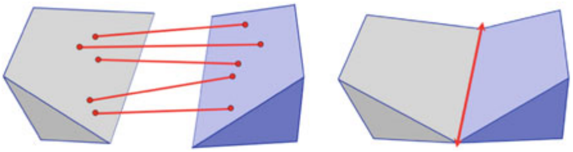


Fig. 1 A body-bar framework and its corresponding body-hinge framework

The underlying combinatorial structure for a body-bar framework is a multigraph $G = (B, E)$, where the vertices in B represent the bodies of the framework and the edges in E represent the rigid bars of the framework [20, 21]. To determine whether a given body-bar framework with underlying multigraph $G = (B, E)$ is rigid is in general a very difficult problem, as it requires solving a system of quadratic equations. It is therefore common to linearize this problem by differentiating the length constraints given by the rigid bars. This leads to a homogeneous system of $|E|$ linear equations in $6|B|$ unknowns, and the corresponding coefficient matrix of this linear system is called the *rigidity matrix* of the framework. The elements in the kernel of this matrix are called the *infinitesimal motions* of the framework, and it is well known that the space of trivial infinitesimal motions of a 3-dimensional body-bar framework (i.e. the space of infinitesimal motions of the framework which correspond to congruent motions of Euclidean 3-space) is of dimension 6. Thus, a body-bar framework possesses a non-trivial infinitesimal motion if and only if the dimension of the kernel of the rigidity matrix is strictly larger than 6.

It is a well known fact that if a body-bar framework is infinitesimally rigid (i.e. it has only has trivial infinitesimal motions), then it is also rigid. The converse is not true in general. However, if the framework is *regular*, that is, if the rigidity matrix of the framework with underlying multigraph G has maximal rank among all realizations of G as a body-bar framework, then the existence of a non-trivial infinitesimal motion also guarantees the existence of a non-trivial continuous motion of the framework [1]. The set of regular realizations of a multigraph G forms a dense open subset of all possible realizations of G as a body-bar framework, and hence 'almost all' realizations of G as a body-bar framework are regular.

Tay showed that for a regular body-bar framework, the rigidity is captured by the following simple combinatorial counts [18]:

Theorem 2.1 *(Tay, 1984) A regular* 3*-dimensional body-bar framework with underlying multigraph $G = (B, E)$ is rigid if and only if there exists a subset E^* of E which satisfies the conditions:*

(1) $|E^*| = 6|B| - 6$*;*
(2) $|E'| \leq 6|B'| - 6$ *for all subgraphs induced by subsets of E^*.*

Tay's counts are related to general Grübler–Kutzbach counting rules, however they are more precise as they can detect topological redundancies. While algorithmically Tay's condition looks like we need to check all possible subsets of bars of the framework (an exponential process!), these counts on a multigraph define independent sets in a matroid, leading to a fast greedy algorithm called the pebble game, with running time of $O(|B||E|)$ [10, 16].

A 3-dimensional *body-hinge framework* consists of a set of rigid bodies which are connected, in pairs, along revolute hinges (i.e., lines in 3-space), and each of the bodies is free to move continuously, subject to the constraints that the contacts along the hinges are preserved (see Fig. 1, right). Since a hinge removes 5 of the 6 relative degrees of freedom between a pair of rigid bodies in 3-space, a body-hinge framework can be modeled as a special case of a body-bar framework by replacing each hinge with 5 independent bars, each intersecting the hinge line.

The counts in Theorem 2.1 (and the corresponding pebble game algorithms) also characterize regular rigid body-hinge frameworks in 3-space [19, 20], where a body-hinge framework with multigraph G is called regular if its rigidity matrix has maximal rank among all body-hinge realizations of G. Moreover, the recent Molecular Theorem [9] confirmed that Tay's counts also characterize regular rigid molecular linkages, where a *molecular linkage* is a body-hinge framework with the special geometry that all hinges of each body are concurrent in a single point. This result solved the more than 20 year old Molecular Conjecture [19].

3 Detecting Symmetry-Preserving Motions in Symmetric Linkages

We say that a structure is symmetric if it possesses a non-trivial point group, i.e., a non-trivial group of symmetry operations (isometries in Euclidean 3-space) which leave the structure unchanged. Many natural structures such us proteins, as well as many human-built structures such as linkages and other mechanical machines are symmetric [4]. As a consequence, there has been a growing interest in the impact of symmetry on the rigidity and flexibility of these structures. In particular, a symmetric analog of the rigidity matrix, called the *orbit rigidity matrix*, has been constructed in [15] to detect hidden symmetry-preserving motions in symmetric frameworks.

For simplicity, we will only consider body-bar frameworks whose symmetry group acts freely on the bodies of the framework (that is, no body of a symmetric framework is unshifted by any non-trivial symmetry operation of the framework). Note that for a framework with a purely rotational point group, this means that no body is centered on a rotational axis. For such frameworks, the symmetry-adapted rigidity analysis leads to some very simple Tay-type counts (see Theorem 3.1).

If we focus only on motions which preserve the full symmetry group S of the original framework, the motion of a single body b uniquely determines the motions of all bodies that lie in the same orbit as b under the group action of S (i.e., all bodies $s(b)$, $s \in S$) as well. So we have 6 degrees of freedom for each orbit of bodies. Similarly, each orbit of bars (i.e., each bar together with all of its symmetric copies) restricts the motion of the combined structure by one. Therefore, the orbit rigidity matrix has $|E_o|$ rows and $6|B_o|$ columns, where $|E_o|$ and $|B_o|$ denote the number of bar orbits and body orbits of the framework, respectively.

The key property of the orbit rigidity matrix is that its kernel is the space of S-symmetric infinitesimal motions of the framework (see [15] for details). In the following, we denote the dimension of the space of trivial S-symmetric infinitesimal motions of the framework by v_S. Note that for any symmetry group S in dimension 3, v_S can be read off from the character tables in [2], for example.

Analogously to the non-symmetric situation, a body-bar (or molecular) framework with symmetry group S and underlying multigraph G is said to be S-regular, if its orbit rigidity matrix has maximal rank among all S-symmetric realizations of G as a body-bar (or molecular) framework. From the results in [14] it follows that if an S-regular body-bar framework has an S-symmetric non-trivial infinitesimal motion, then it also has a non-trivial continuous motion which preserves the symmetry group S throughout the path. This fundamental result leads to the following necessary conditions for an S-regular body-bar (or molecular) framework to be rigid:

Theorem 3.1 *(Schulze, Whiteley, 2010 [14, 15]) If an S-regular body-bar (or molecular) framework in $\mathbb{R}^3$ (with S acting freely on the bodies) has only trivial symmetry-preserving continuous motions, then there exists a subset of representatives for the bar orbits, E_0^*, such that:*

(1) $|E_0^*| = 6|B_0| - v_S$;
(2) $|E_0'| \leq 6|B_0'| - v_S$ *for all subgraphs induced by subsets E_0' of E_0^*.*

It is shown in [17] that there are some further necessary conditions for rigidity, and that this larger set of conditions is also sufficient for an S-regular body-bar framework to have no symmetry-preserving motion.

Very recently, some initial symmetry-extended versions of the pebble game algorithm—based on the orbit counts in Theorem 3.1 and on Tay's original counts—have been developed for the group $\mathcal{C}_2$ in order to test frameworks for (possibly symmetry breaking) flexibility [13]. While these algorithms clearly provide sufficient conditions for the flexibility of a $\mathcal{C}_2$-symmetric structure, an area of ongoing research is whether they also provide necessary conditions for flexibility.

Note that Theorem 3.1 applies to both body-bar and molecular frameworks, because (1) and (2) are necessary, but not sufficient conditions for an S-regular body-bar framework to be rigid, and the rank of the orbit rigidity matrix of an S-regular realization of a multigraph G as a body-bar framework is at least as big as the rank of the orbit rigidity matrix of an S-regular realization of G as a molecular framework. We offer the following two conjectures

Conjecture 3.1 (**Symmetric Molecular Conjecture I)** The orbit rigidity matrix of an S-regular body-bar realization of a multigraph G has the *same* rank as the orbit rigidity matrix of an S-regular molecular realization of G.

In other words, we conjecture that under S-regular conditions, body-bar realizations of G and molecular realizations of G have the same number of S-preserving degrees of freedom.

Conjecture 3.2 (**Symmetric Molecular Conjecture II**). The rank of the entire rigidity matrix—and not just the orbit matrix—of an S-generic body-bar realization of a multigraph G is the same as the rank of the rigidity matrix of an S-generic molecular realization of G, where a structure is called S-generic if the rank of the entire rigidity matrix is maximal, among all S-symmetric realizations.

It would follow from this conjecture that under S-generic conditions, body-bar realizations and molecular realizations of G have exactly the same number of (not necessarily S-preserving) infinitesimal motions.

4 Evidence for the Symmetric Molecular Conjectures

To collect evidence supporting the previous conjectures, we have applied the higher-dimensional continuation methods implemented in the CUIK suite [6, 11] to a variety of symmetric molecular linkages. These methods allow an exhaustive tracing of the C-space component of the linkage that is path-connected to a given configuration. The exploration marches from such configuration in all directions systematically by constructing local charts on the tangent bundle. Neighboring charts are mutually coordinated to keep track of the C-space region explored up to a given point, and branch-switching methods are applied at bifurcation points in order not to leave areas unexplored [5]. In the end, the component topology can be inferred from the neighborhood relationships between charts.

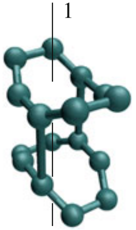
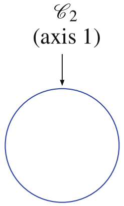
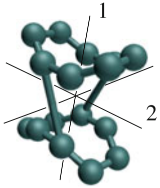
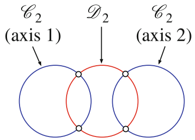
Table 1 provides a selection of the molecular linkages analyzed so far. For each linkage, we show the initial configuration used for continuation, its symmetry group, the number of predicted motions according to the Tay and orbit counts, and the C-space topology inferred from the numerical analysis. We focus on linkages exhibiting the half-turn symmetry group $\mathcal{C}_2$ and the dihedral group $\mathcal{D}_2$ of order 4 (which is generated by two perpendicular half-turn axes), as these are the most common groups for molecular structures such as proteins [4, 13].

Note that each of the linkages in Table 1 is predicted to be rigid with Tay's non-symmetric counts. (In fact, the third linkage even counts to be over-constrained by two.) However, in each case we have $6|B_0| - v_S - |E_0| = 1$, so that we may conclude from Theorem 3.1 that for S-regular realizations, each of these molecular linkages has a symmetry-preserving non-trivial motion (Note that $v_{\mathcal{C}_2} = 2$ in Table 1 since the space of $\mathcal{C}_2$-symmetric trivial motions is generated by a translation along the $\mathcal{C}_2$ axis and a rotation about the $\mathcal{C}_2$ axis.)

From the topology of the configuration space of the first and the third linkage it follows that there are no additional (symmetry-preserving or symmetry-breaking) non-trivial motions for these linkages, which provides evidence for both symmetric Molecular Conjectures.

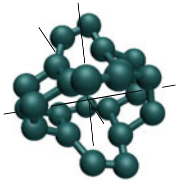
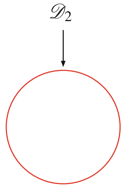
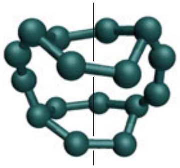
Similarly, the configuration space of the second linkage also supports the first symmetric Molecular Conjecture, since it shows that the linkage has only one $\mathcal{D}_2$-preserving degree of freedom at every $\mathcal{D}_2$-regular point, and only one $\mathcal{C}_2$-preserving degree of freedom at every $\mathcal{C}_2$-regular point. (At the bifurcation points, the linkage is still $\mathcal{D}_2$-regular, and the additional motion only preserves one of the half-turn axes of $\mathcal{D}_2$ so it is not $\mathcal{C}_2$-regular at those points.) This example is also consistent with the Symmetric Molecular Conjecture II, since the bifurcation points are neither $\mathcal{C}_2$- nor $\mathcal{D}_2$-generic, as the rank of the entire rigidity matrix at the bifurcation points is clearly not maximal among the $\mathcal{C}_2$- or $\mathcal{D}_2$-symmetric realizations of the structure.

Table 1 Tay and orbit counts for selected molecules, and the results of the numerical analysis

Molecular structure	Symmetry group	Tay's counts	Orbit counts	C-space analysis
	$\mathcal{C}_2$ $(triv_{\mathcal{C}_\in} = 2)$	$\|B\| = 16$ $\|E\| = 18 \cdot 5 = 90$ $\|E\| = 6\|B\| - 6$ Predicts rigid molecule	$\|B_0\| = 8$ $\|E_0\| = 45$ $\|E_0\| < 6\|B_0\| - 2$ Predicts 1-dim. $\mathcal{C}_2$ symmetric motion	
	$\mathcal{D}_2$ $(triv_{\mathcal{D}_\in} = 0)$	$\|B\| = 16$ $\|E\| = 18 \cdot 5 = 90$ $\|E\| = 6\|B\| - 6$ Predicts rigid molecule	$\mathcal{D}_2 \|B_0\| = 4$ $\|E_0\| = 23$ $\|E_0\| < 6\|B_0\| - 0$ Predicts 1-dim. $\mathcal{D}_2$ symmetric motion $\mathcal{C}_2 : \|B_{0'}\| = 8$ $\|E_{0'}\| = 45$ $\|E_{0'}\| < 6\|B_{0'}\| - 2$	 $\mathscr{C}_2$ (axis 1) $\mathscr{D}_2$ $\mathscr{C}_2$ (axis 2)

(continued)

Table 1 (continued)

Molecular structure	Symmetry group	Tay's counts	Orbit counts	C-space analysis
	$\mathcal{D}_2$ $(triv_{\mathcal{D}_\in} = 2)$	$\lvert B\rvert = 24$ $\lvert E\rvert = 28 \cdot 5 = 140$ $\lvert E\rvert > 6\lvert B\rvert - 6$ Predicts rigid molecule	$\lvert B_0\rvert = 6$ $\lvert E_0\rvert = 35$ $\lvert E_0\rvert < 6\lvert B_0\rvert - 0$ Predicts 1-dim. $\mathcal{D}_2$ symmetric motion	
	$\mathcal{C}_2$ $(triv_{\mathcal{C}_2} = 2)$	$\lvert B\rvert = 16$ $\lvert E\rvert = 18 \cdot 5 = 90$ $\lvert E\rvert = 6\lvert B\rvert - 6$ Predicts rigid molecule	$\lvert B_0\rvert = 8$ $\lvert E_0\rvert = 45$ $\lvert E_0\rvert < 6\lvert B_0\rvert - 2$ Predicts 1-dim. $\mathcal{C}_2$ symmetric motion	2-dimensional C-space with $\mathcal{C}_2$ sub loop. The 6-revolute rings always maintain $\mathcal{C}_2$ symmetry.

Finally, while the orbit counts predict the fourth linkage in Table 1 to have one non-trivial $\mathcal{C}_2$-preserving motion, a more detailed symmetry-based analysis of the subgraphs of the two 6-revolute rings shows that there is in fact an additional symmetry-breaking motion as well. Our computations of the configuration space again confirm that there is indeed a two-dimensional C-space for this molecular linkage. In addition, our computations show that each of the two 6-revolute rings of the structure maintains its half-turn symmetry at all times, which supports our expectation that stressed (over-constrained under symmetry) components of a structure (such as the symmetric 6-rings in the linkage) are more likely to maintain their symmetry than unstressed ones (such as the connecting links between the 6-rings) [13].

5 Conclusions

This paper has described the symmetric molecular conjectures of rigidity theory, giving their context, formulation, and a selection of examples providing supporting evidence for them. The associated combinatorial methods allow to predict hidden motions in symmetric molecules that otherwise seem apparently over-constrained and rigid. The ultimate aim of this work is to move from the theoretical analysis and to apply the symmetry-based methods and pebble game algorithms to actual proteins, to better understand their functions and possible drug targets.

Acknowledgments This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under project DPI2010-18449.

References

1. Asimov, L., Roth, B.: The rigidity of graphs. AMS **245**, 279–289 (1978)
2. Atkins, P., Child, M., Phillips, C.: Tables for Group Theory. Oxford University Press, Oxford (1970)
3. Daniel, R.M., Dunn, R.V., Finney, J.L., Smith, J.C.: The role of dynamics in enzyme activity. Annu. Rev. Biophys. Biomol. Struct. **32**, 69–92 (2003)
4. Goodsell, D., Olson, A.: Structural symmetry and protein function. Annu. Rev. Biophys. Biomol. Struct. **29**, 105–153 (2000)
5. Henderson, M.E.: Multiparameter parallel search branch switching. Int. J. Bifurcat. Chaos Appl. Sci. Eng. **15**(3), 967–974 (2005)
6. Henderson, M.E.: Multiple parameter continuation: computing implicitly defined k-manifolds. Int. J. Bifurcat. Chaos **12**(3), 451–476 (2002)
7. Jacobs, D., Rader, A., Kuhn, L., Thorpe, M.: Protein flexibility predictions using graph theory. Proteins Struct. Funct. Genet. **44**, 150–165 (2001)
8. Jagodzinski, F., Hardy, J., Streinu, I.: Using rigidity analysis to probe mutation-induced structural changes in proteins. J. Bioinform. Comput. Biol. **10**(10), (2012)
9. Katoh, N., Tanigawa, S.: A proof of the molecular conjecture. Discrete Comput. Geom. **45**(4), 647–700 (2011)
10. Lee, A., Streinu, I.: Pebble game algorihms and (k, l)-sparse graphs. In: European Conference on Combinatorics, Graph Theory and Applications, Berlin pp. 181–186 (2005)

11. Porta, J.M., Ros, L., Bohigas, O., Manubens, M., Rosales, C., Jaillet, L.: The CUIK suite: motion analysis of closed-chain multibody systems (Submitted)
12. Porta, J.M., Ros, L., Thomas, F., Corcho, F., Cantó, J., Pérez, J.J.: Complete maps of molecular-loop conformational spaces. J. Comput. Chem. **28**, 2170–2189 (2007)
13. Schulze, B., Sljoka, A., Whiteley, W.: How does symmetry impact the rigidity of proteins? Phil. Trans. R. Soc. A (2013, in press)
14. Schulze, B.: Symmetry as a sufficient condition for a finite flex. SIAM J. Discrete Math. **24**(4), 1291–1312 (2010)
15. Schulze, B., Whiteley, W.: The orbit rigidity matrix of a symmetric framework. Discrete Comput. Geom. **46**(3), 561–598 (2011)
16. Sljoka, A.: Algorithms in rigidity theory with applications to protein flexibility and mechanical linkages. Ph.D. Thesis, York University http://www.math.yorku.ca/~adnanslj/adnanthesis.pdf (2012)
17. Tanigawa, S.: Matroids of gain graphs in applied discrete geometry. arXiv:1207.3601 (2012)
18. Tay, T.S.: Rigidity of multi-graphs, linking rigid bodies in *n*-space. J. Comb. Theory, B **36**, 95–112 (1984)
19. Tay, T.S., Whiteley, W.: Recent advances in generic rigidity of structures. Struct. Topol. **9**, 31–38 (1985)
20. Whiteley, W.: Some matroids from discrete applied geometry. Contemp. Math. **197**, 171–311 (1996)
21. Whiteley, W.: Counting out to the flexibility of molecules. Phys. Biol. **2**, 1–11 (2005)

Stiffness Modeling of Robotic Manipulator with Gravity Compensator

Alexandr Klimchik, Stéphane Caro, Yier Wu, Damien Chablat, Benoit Furet and Anatol Pashkevich

Abstract The paper focuses on the stiffness modeling of robotic manipulators with gravity compensators. The main attention is paid to the development of the stiffness model of a spring-based compensator located between sequential links of a serial structure. The derived model allows us to describe the compensator as an equivalent non-linear virtual spring integrated in the corresponding actuated joint. The obtained results have been efficiently applied to the stiffness modeling of a heavy industrial robot of the Kuka family.

Keywords Stiffness modeling · Gravity compensator · Industrial robot

A. Klimchik (✉) · Y. Wu · A. Pashkevich
Institut de Recherche en Communications et Cybernétique de Nantes, Ecole des Mines de Nantes, Nantes, France
e-mail: alexandr.klimchik@mines-nantes.fr

Y. Wu
e-mail: yier.wu@mines-nantes.fr

A. Pashkevich
e-mail: anatol.pashkevich@mines-nantes.fr

S. Caro · D. Chablat
Institut de Recherche en Communications et Cybernétique de Nantes, National Center for Scientific Research, Nantes, France
e-mail: stephane.caro@irccyn.ec-nantes.fr

D. Chablat
e-mail: damien.chablat@irccyn.ec-nantes.fr

B. Furet
Institut de Recherche en Communications et Cybernétique de Nantes, University of Nantes, Nantes, France
e-mail: benoit.furet@univ-nantes.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_21,

1 Introduction

Recently, in aerospace industry much attention is paid to the high-precision and high-speed machining of large dimensional aircraft components. To satisfy these requirements, industrial robots are more and more used to replace conventional CNC-machines, which are limited with their performances and are suitable for a rather limited workspace. However, some new problems arise here because of essential machining forces caused by processing of modern aeronautic materials that may reduce the quality of technological process. To overcome this difficulty, the robot manufactories attempt to make the manipulators stiffer but quite heavy. The problem of link weights is often solved by using the gravity compensator, which, in its turn, influences on the position accuracy and stiffness properties. The latter motivates enhancement of the manipulator stiffness modeling techniques that are in the focus of this paper. The problem of stiffness modeling for serial manipulators has been studied in the robotic community from different aspects [1–3]. In particular, special attention has been paid to heavy industrial and medical robots with essential deflections of the end-effector [4–6]. Among a number of existing stiffness modeling approaches, the Virtual Joint Modeling (VJM) method looks the most attractive for the robotic applications. Its main idea is to take into account the elastostatic properties of flexible components by presenting them as equivalent virtual springs localized in the actuated or passive joints [7]. Because of its simplicity and efficiency, this approach has been applied to numerous case-studies and progressively enhanced to take into account specific features of robotic manipulators, such as internal and external loadings, closed-loops, etc. [8, 9]. Besides, some authors have extended the VJM approach by using advanced 6 d.o.f. virtual springs describing stiffness properties of the manipulator elements [10]. However, the problem of the stiffness modeling of the manipulators with gravity compensators has not been studied yet sufficiently; there are only limited number of works that addressed this issue. Besides, in previous works, the main attention has been paid to the compensator design [11] and modification of the inverse kinematics algorithms integrated in the robot controller [12]. In contrast to previous works, this paper deals with the stiffness modeling of serial robotic manipulators equipped with a spring-based compensator located between sequential links. It proposes an extension of the VJM-based approach allowing to integrate the elastostatic properties of the gravity compensator (that creates a closed loop) into the conventional stiffness model of the manipulator.

2 Stiffness Model of Gravity Compensator

Let us consider the general type of gravity compensator that incorporates a passive spring attached between two sequential links of the manipulator (Fig. 1). In a such architecture, there is a closed loop that generates an additional torque in the

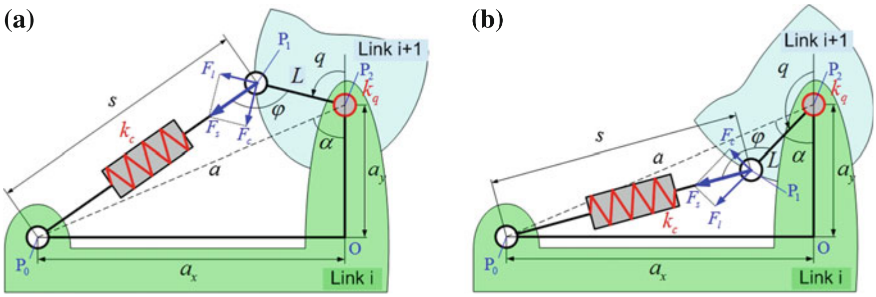


Fig. 1 Mechanics of gravity compensator. **a** Case A: configuration $q < \pi - \alpha$. **b** Case B: configuration $q > \pi - \alpha$

manipulator joint. So, the specificity of such design allows us to limit modifications of the robot stiffness model by adjusting the virtual joint stiffness parameters.

The geometrical model of considered compensator includes three main node points P_0, P_1, P_2. Let us denote corresponding distances as $L = |P_1, P_2|$, $a = |P_0, P_2|$, $s = |P_0, P_1|$. In addition, let us introduce the angles α, φ and the distances a_x and a_y, whose geometrical meaning is clear from Fig. 1. Using these notations, the compensator spring deflection s can be computed as

$$s = \sqrt{a^2 + L^2 + 2 \cdot a \cdot L \cdot \cos(\alpha + q)} \tag{1}$$

and evidently depends on the joint angle q. The considered mechanical design allows us to balance the manipulator weight for any given configuration by adjusting the compensator spring preloading. It can be taken into account by introducing the zero-value of the compensator length s_0 corresponding to the unloaded spring. Under this assumption, the compensator force applied to the node P1 can be expressed as $F_s = K_c \cdot (s - s_0)$, where K_c is the compensator spring stiffness. Further, the angle φ between the compensator links P0P1 and P1P2 (see Fig. 1) can be found from the expression $\sin\varphi = a/s \ \cdot \sin(\alpha + q)$, which allows us to compute the compensator torque M_c applied to the second joint as

$$M_c = K_c \cdot (1 - s_0/s) \cdot a \cdot L \ \cdot \sin(\alpha + q) \tag{2}$$

After differentiation of the latter expression with respect to q, the equivalent stiffness of the second joint (comprising both the manipulator and compensator stiffnesses) can be expressed as:

$$K_\theta = K_\theta^0 + K_c \cdot \ a\,L \ \cdot \ \eta_q \tag{3}$$

where the coefficient

$$\eta_q = \cos(\alpha + q) - \frac{s_0}{s} \cdot \left(\frac{a\,L}{s^2}\sin^2(\alpha + q) + \cos(\alpha + q)\right) \tag{4}$$

highly depends on the value of joint variable q and the initial preloading in the compensator spring described by s_0. Hence, using expression (3), it is possible to extend the classical stiffness model of the serial manipulator by modifying the virtual spring parameters in accordance with the compensator properties. While in the paper this approach has been used for the particular compensator type, similar idea can be evidently applied to other compensators. It is also worth mentioning that the geometrical and elastostatic parameters of gravity compensators (α, a, L and K_c, s_0 for the presented case) usually are not included in datasheets. For this reason, these parameters should be identified from the calibration experiments (see Sect. 4).

3 Extension of the VJM-Based Approach

Stiffness of a serial robot highly depends on its configuration and is defined by the Cartesian stiffness matrix. Using the VJM-based approach adopted in this paper, the manipulator can be presented as the sequence of rigid links separated by the actuators and virtual flexible joints incorporating all elastostatic properties of flexible elements [6]. In accordance with these assumptions, the elastostatic model of the serial robot can be presented as shown in Fig. 2.

For this serial robot, the end-effector location $\mathbf{t}$ can be defined by the vector function $\mathbf{g}(\mathbf{q}, \boldsymbol{\theta})$, where $\mathbf{q}$, $\boldsymbol{\theta}$ denote the vectors of the actuator and virtual joint coordinates respectively. It can be proved that the static equilibrium equations can be written as $\mathbf{J}_\theta^{\mathrm{T}} \cdot \mathbf{F} = \mathbf{K}_\theta \cdot \boldsymbol{\theta}$, where $\mathbf{J}_\theta = \partial \mathbf{g}(\mathbf{q}, \boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ is the Jacobian matrix, the matrix $\mathbf{K}_\theta = diag\left(\mathbf{K}_{\theta_1}, \ldots, \mathbf{K}_{\theta_{\mathrm{n}}}\right)$ aggregates stiffnesses of all virtual springs and $\mathbf{F}$ is the external loading applied to the robot end-effector. In order to find the desired stiffness matrix $\mathbf{K}_{\mathrm{C}}$, the force-deflection relation should be linearized in the neighborhood of the current configuration $\mathbf{q}$ and presented in the form $\mathbf{F} = \mathbf{K}_{\mathrm{C}} \cdot \Delta\mathbf{t}$, where $\Delta\mathbf{t}$ is the end-effector deflection caused by the external loading $\mathbf{F}$. After relevant transformations, one can get the following expression for the desired stiffness matrix [4]

$$\mathbf{K}_{\mathrm{C}} = \left(\mathbf{J}_\theta \cdot \mathbf{K}_\theta^{-1} \cdot \mathbf{J}_\theta^{\mathrm{T}}\right)^{-1} \tag{5}$$

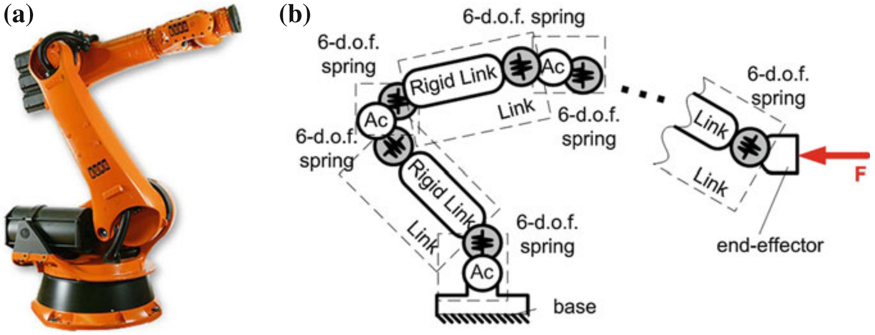


Fig. 2 Serial industrial robot (**a**) and its VJM-based stiffness model (**b**)

which depends on the manipulator geometry and its elastostatic properties. This classical expression has been originally derived for the case of manipulators with-out gravity compensators, where the matrix $\mathbf{K}_\theta$ is constant. To integrate into this model the gravity compensator, it is proposed to replace the classical joint stiffness matrix $\mathbf{K}_\theta^0$ (that takes into account elastostatic properties of serial manipulator without gravity compensator) by the sum of two matrices

$$\mathbf{K}_\theta = \mathbf{K}_\theta^0 + \mathbf{K}_\theta^{GC}(\mathbf{q}) \tag{6}$$

where the second term $\mathbf{K}_\theta^{GC}(\mathbf{q})$ is configuration-dependent and takes into account the elastostatic properties of the gravity compensator (by transforming them in the virtual joint space). In practice, the matrix $\mathbf{K}_\theta^{GC}(\mathbf{q})$ may include either one or several non-zero elements (depending on the number of compensators). These elements ale located on the matrix diagonal and are computed using (3). Hence, by modification of the manipulator joint stiffness matrix in accordance with expression (6), it is possible to integrate the compensator parameters into the classical VJM-based model of the serial manipulator. It should be mentioned that this idea can be also applied for other types of compensators.

4 Application Example

Let us illustrate the efficiency of the developed stiffness modeling technique by applying it to the compliance error compensation in the robotic-based machining performed by the industrial robot KUKA KR-270. This robot is equipped with the spring-based gravity compensator located between the first and second links (which leads to the influence on the second actuated joint). In accordance with the considered specifications, the technological process should be performed in the square area of the size 2,000 mm located at the height 500 mm over the floor level (see Fig. 3 for more details). For comparison purpose, it is assumed that machining force is constant throughout the working area and it is equal to $\mathbf{F} = (0, 360N, 560N, 0, 0, 0)^T$, which corresponds to a typical milling process.

To obtain the desired stiffness model, special calibration experiments have been conducted and dedicated identification procedures have been applied. These yield the desired geometrical and elastostatic parameters of the robot with gravity compensator presented in Table 1. From these data, the classical joint stiffness matrix $\mathbf{K}_\theta^0$ has been constructed straightforwardly (from parameters $k_1, \ldots k_6$). To integrate the gravity compensator properties, the matrix $\mathbf{K}_\theta^{GC}(\mathbf{q})$ has been computed using expression (3) and values k_c, L, a_x and a_y for any robot configuration $\mathbf{q}$. Relevant computations have been done throughout the required working area, where the end-effector deflections has been evaluated for given machining force $\mathbf{F}$. The computational results are summarized in Fig. 4, where the end-effector compliance errors are presented. These results show that the compliance errors are not negligible here and vary from

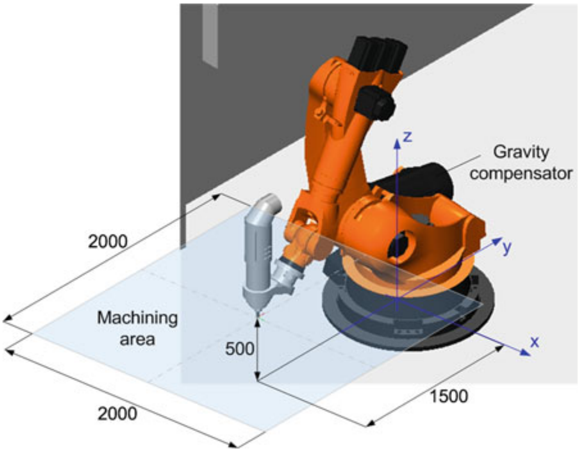


Fig. 3 Industrial robot Kuka KR270 with required machining area

Table 1 Elastostatic and geometrical parameters of robot with gravity compensator

Parameters	Units	Values	CIs
k_1	$[rad \times m/N]$	3.774×10^{-6}	–
k_2	$[rad \times m/N]$	0.302×10^{-6}	0.004×10^{-6}
k_3	$[rad \times m/N]$	0.406×10^{-6}	0.008×10^{-6}
k_4	$[rad \times m/N]$	3.002×10^{-6}	0.115×10^{-6}
k_5	$[rad \times m/N]$	3.303×10^{-6}	0.162×10^{-6}
k_6	$[rad \times m/N]$	2.365×10^{-6}	0.095×10^{-6}
k_c	$[rad \times m/N]$	0.144×10^{-6}	0.031×10^{-6}
s_0	$[mm]$	458.00	27.0
L	$[mm]$	184.72	0.06
a_x	$[mm]$	685.93	0.70
a_y	$[mm]$	120.30	0.69

0.34 to 3.5 mm. So, to achieve the desired precision, it is reasonable to apply the error compensation technique, which is based on the reliable stiffness model.

For comparison purposes, two alternative stiffness models have been examined. The first one is based on the classical assumptions and takes into account the stiffness properties of the actuated joints only (without gravity compensator). The second model integrates the gravity compensator in accordance with the approach proposed in this paper. Both the models have been used for the compliance error compensation for the described above manufacturing task. Relevant results are presented in Fig. 5, which shows the difference in the compliance error compensation while applying the classical and the proposed approaches. As follows from the figure, ignoring the gravity compensator influence may lead to the position error over/under compensation of the order of 0.07 mm, which is not admissible for the manufacturing processes

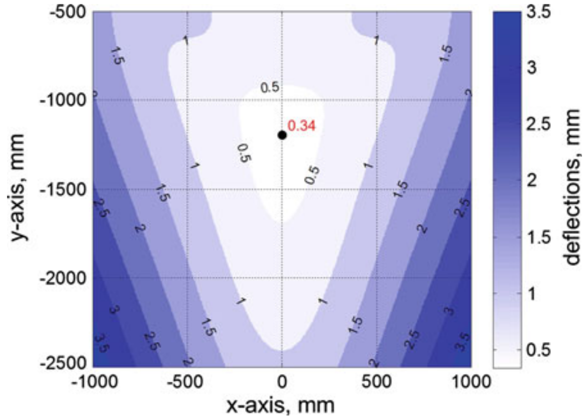


Fig. 4 The end-effector deflections caused by machining forces throughout working area

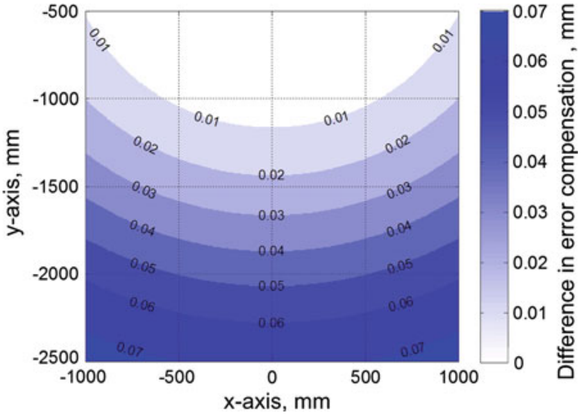


Fig. 5 Difference in stiffness error compensation between two strategies

employed in the aerospace applications studied here. Hence, to ensure the high precision for the robotic-based machining, the compliance error compensation technique must rely on the stiffness model, which takes into account the impact of the gravity compensator, in accordance with the approach developed in this paper.

5 Conclusions

The paper presents a new approach for the stiffness modeling of robots with the spring-based gravity compensators, which are located between the manipulator sequential links. Using this approach, the compensator has been replaced by an equivalent non-linear virtual spring integrated in the corresponding actuated joint.

This methodology allowed us to extend the VJM-based modeling technique for the case of manipulators with closed-loops induced by the gravity compensators via using configuration-dependent joint stiffness matrix. Efficiency of the developed approach and its industrial value have been confirmed by an application ex-ample, which deals with robotic-based milling of large-dimensional parts for aerospace industry. Feature work will deal with integration of this modeling approach into the robotic software CAD system.

Acknowledgments The work presented in this paper was partially funded by the ANR, France (Project ANR-2010-SEGI-003-02-COROUSSO) and Project ANR ROBOTEX. The authors also thank Fabien Truchet, Guillaume Gallot, Joachim Marais and Sbastien Garnier for their great help with the experiments.

References

1. Company, O., Pierrot, F., Fauroux, J.-C.: A method for modeling analytical stiffness of a lower mobility parallel manipulator. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2005), pp. 3232–3237 (2005)
2. Alici, G., Shirinzadeh, B.: Enhanced stiffness modeling, identification and characterization for robot manipulators. Proc. IEEE Trans. Robot. **21**, 554–564 (2005)
3. Tyapin, I., Hovland, G.: Kinematic and elastostatic design optimization of the 3-DOF Gantry-Tau parallel kinamatic manipulator. Model. Ident. Control **30**, 39–56 (2009)
4. Yi, B.-J., Freeman, R.A.: Geometric analysis antagonistic stiffness redundantly actuated parallel mechanism. J. Rob. Syst. **10**(5), 581–603 (1993)
5. Meggiolaro, M.A., Dubowsky, S., Mavroidis, C.: Geometric and elastic error calibration of a high accuracy patient positioning system. Mech. Mach. Theory **40**, 415–427 (2005)
6. Kövecses, J., Angeles, J.: The stiffness matrix in elastically articulated rigid-body systems. Multibody Sys. Dyn. **18**(2), 169–184 (2007)
7. Salisbury, J.: Active stiffness control of a manipulator in Cartesian coordinates. In: Proceedings of the 19th IEEE Conference on Decision and, Control, pp. 87–97 (1980)
8. Klimchik, A., Bondarenko, D., Pashkevich, A., Briot, S., Furet, B.: Compensation of tool deflection in robotic-based Milling. In: Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2012), pp. 113–122 (2012)
9. Quennouelle, C., Gosselin, C.M.: Instantaneous kinemato-static model of planar compliant parallel mechanisms. Proceedings of ASME International Design Engineering Technical Conferences, In (2008)
10. Pashkevich, A., Klimchik, A., Chablat, D., Wenger, Ph.: Accuracy improvement for stiffness modeling of parallel manipulators. In: Proceedings of 42nd CIRP Conference on Manufacturing Systems, CD-proceedings (p. 8), Grenoble (2009)
11. Takesue, N., Ikematsu, T., Murayama, H., Fujimoto, H.: Design and prototype of variable gravity compensation mechanism (VGCM). J. Robot. Mech. **23**(2), 249–257 (2011)
12. De Luca, A., Flacco, F.: A PD-type regulator with exact gravity cancellation for robots with flexible joints. In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, pp. 317–323.

Computational Algorithm for Determining the Generic Mobility of Floating Planar and Spherical Linkages

Offer Shai and Andreas Müller

Abstract It is well-known that structural mobility criteria, such as the Chebychev-Kutzbach–Grbler (CKG) formula, fail to correctly determine the mobility of mechanisms with special geometries. Even more, any known structural mobility criteria also fail to determine the generic (i.e. topological) mobility since they are prone to topological redundancies. A computational algorithm is proposed in this paper, which always finds the correct generic mobility of any planar and spherical mechanism. Its foundation is a novel representation of constraints by means of a constraint graph. The algorithm builds on the 'pebble game', originally developed within combinatorial rigidity theory for checking the rigidity of graphs. An extension of Laman's theorem is introduced that enables application of the algorithm to any planar or spherical mechanism with higher and lower holonomic kinematic pairs and multiple joints. The novel algorithm further yields the redundantly constrained sub-linkages of a mechanism. In addition this algorithm naturally leads to a decomposition of a mechanism into Assur graphs, however this is beyond the scope of this paper.

Keywords Mobility · Topological redundancy · Pebble game · Assur graphs

1 Introduction

The mobility being the essential property of a mechanism has been a major matter of interest in mechanism theory. The approaches can be broadly classified as those that deal with the mobility of a given mechanism, with a particular geometry, and those

O. Shai
Mechanical Engineering School, Tel-Aviv University, Tel-Aviv, Israel

A. Müller (✉)
Institute of Mechatronics, Chemnitz, Germany
e-mail: andreas.mueller@ifm.chemnitz.de

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_22,

that aim on the generic mobility of a class of mechanisms with certain topology [1]. Methods of the first class attempt an explicit solution of the constraint equations or the approximation of the solution variety [13–15], possibly using tools from numerical algebraic geometry [16, 17]. Instead of considering a particular geometry, the second class approaches the problem from a structural point of view. These attempts have a long tradition and only need topological information about the existence of links and joints. The CKG formula is a well-known topological mobility criterion. It is assumed that they generally yield the generic mobility [6], i.e. the mobility of almost all realizations of a particular topology. Although they are independent of any geometric data all such methods are sensitive to topological redundancy since these criteria only take into account the existence of joints and links but not their particular arrangement.

The identification of topological redundancies requires graph-theoretic considerations of the constraints and appropriate algorithms. Such an algorithm is presented in this paper. The basis for this algorithm is a graph representation of the constraints inherited from rigidity theory. This differs from the topological graph [3] often used in that it does not merely represent the arrangement of links and joints, but rather the system of constraints imposed to the links. This is presented in Sect. 2, where the two established types (body-bar, bar-joint) are recalled and are mentioned briefly in the paper, and a novel type of constraint graph is introduced. The mathematical theorem underlying the proposed computational algorithm is given in Sect. 3, and the actual computational algorithm is introduced in Sect. 4. The algorithm is proved to converge to the unique generic mobility [9]. In order to motivate the application of this algorithm, an engineering interpretation of the steps and output of the algorithm is given. The application of the method is shown in Sect. 5 for a simple example, and further interpretations of the output are discussed. The paper concludes with a brief outline of future work in Sect. 6.

The algorithm used in this paper, called pebble game, was developed in 1997 [2] for checking whether a set of points subject to geometric constraints form a rigid structure. The use of this algorithm was also extended to check whether a graph consisting of rigid bodies is rigid or mobile as reported in [10]. In engineering, pebble game was applied to check the mobility of planar mechanisms consisting of only binary links and limited to lower kinematic pairs [8]. It was proved that pebble game can decompose any mechanism with only binary links to Assur graphs in 2d and 3d [7]. The algorithm reported in this paper overcomes this limitation and is applicable to any type of planar mechanisms with holonomic higher and lower kinematic pairs and multiple joints.

2 Constraint Graphs

The kinematic functionality of a mechanism is dictated by the geometric and topological constraints imposed on its bodies. The topological graph already relates bodies and joints but it does not explicitly represent the imposed constraints. To this end a *constraint graph* G is introduced. In the following δ denotes the generic mobility,

$G(E, V)$ the constraint graph (undirected or directed), $e(G) = | E |$ the number, and $v(G) = | V |$ the number of vertices of G.

The idea behind constraint graphs is to represent a mechanism as an abstract relation of objects representing certain degrees of freedom (DOFs). These objects constitute vertices of the constraint graph, and are chosen so as to uniquely represent the mechanisms *configuration*. They can stand for rigid bodies or points. The constraints between them are represented by edges. In this sense the graph represents a system of abstract constraint relations that possibly have different physical meanings (e.g. rotation or translation constraints).

There are several types of constraint graphs, such as Bar-Joint graph and Body-Bar graph, but the most general constraint graph, developed by the authors, applies to any type of planar mechanism is the mixed constraint graph below.

In this paper we introduce a new type of graph, termed mixed constraint graph $G = (V_B \cup v_J, E)$. In this graph a vertex v can represent a rigid body, $v \in V_B$, as well as points, $v \in V_J$. That is, for a planar mechanism, each vertex of the mixed constraint graph embodies an object that can move in the plane, and its physical meaning follows from that of the body-bar and bar joint-graph. If a vertex represents a body then it possesses three DOFs. If it represents a point (i.e. the location of a joint) then it has two DOFs. Note, this type of constraint graph can also deal with multiple joints, a revolute joint connecting m bodies thus stands for $m - 1$ revolute joints with collinear axes. For example, in Fig. 1 a, joint J_1 is a multiple revolute joint while the other two joints, J_2 and J_3, are binary joints, i.e., connect between two bodies/links.

3 Rigidity and Mobility of Mixed Graphs

One of the main problems in checking the correct generic mobility of a mechanical system is to identify whether there is no sub-system having over-determinacy, redundant elements. A mathematical criterion for checking such non-existence of

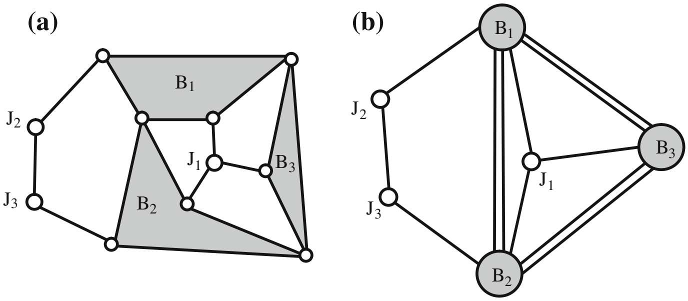


Fig. 1 A linkage **a** whose mixed constraint graph **b** does not satisfy the mixed Laman's theorem

over-determinacy was established and proved in 1970 [4] for bar-joint graph, while in 1991 a mathematical criterion for body-bar graphs was reported [12]. These theorems give rise to the following theorem for mixed constraint graphs:

Planar Mixed Laman's theorem (Shai and Müller, 2013): A floating planar mixed constraint graph $G = (v_B \cup v_J, E)$ with $e(G) = 3v_B(G) + 2v_J(G) - 3$ is determined if and only if $e(G') \leq 3v_B(G') + 2v_J(G') - 3$ for every subgraph G' of G, where $v_B(G)$ and $v_J(G) = \mid V_J \mid$ is the number of vertices corresponding to bodies and points/joints, respectively.

Corollary: A floating planar mixed constraint graph $G = (V_B \cup V_J, E)$ is non-redundant if and only if $e(G') \leq 3v_B(G') + 2v_J(G') - 3$ for every subgraph G'. If this condition is satisfied, the linkage has generic mobility $\delta(G) = 3v_B(G) + 2v_J(G) - e(G) \geq 3$.

For example, the floating system in Fig. 1a is not determined since the corresponding mixed graph in Fig. 1b does not satisfy the Mixed Laman's theorem. To prove that, let us choose the sub-graph spanned by the vertices: $V = \{B_1, B_2, B_3, J_1\}$ having 9 edges which is greater than $3 \cdot 3 + 2 \cdot 13 = 8$, thus mixed Lamans theorem is not satisfied.

4 Pebble Game: A Computational Algorithm

Pebble game is a very efficient algorithm to check if a graph satisfies the mixed Laman's theorem and thus to check if there exists an overdetermined sub-graph. The pebble game is of polynomial order in the number of vertices, $O(\mid V \mid)^2$ and the required memory also grows quadratically, i.e. with $O(\mid V \mid)^2$ [5].

The main concept of the algorithm is to assign 'pebbles' to any physical object in the kinematic model (bodies, points) representing certain DOFs, and to remove them in course of the algorithm. The number of pebbles remaining after running the pebble game is equal to the generic mobility of the linkage. Aiming on the generic, i.e. topological, mobility the method operates exclusively upon the constraint graph, i.e. the topology, and a generic rather than a specific geometry is assumed. Redundancies due to special geometries are thus excluded.

The pebble game starts with an unconstrained system, in the sense that the number of pebbles assigned to a vertex is equal to the DOF as if its members were not subject to any constraint. Denote with $k(v)$ the DOF of the object represented by vertex v. For planar constraints graphs $k(v) = 2$ represents a point and $k(v) = 3$ a body. The algorithm is initialized by assigning $k(v) = 2{,}3$ pebbles to each vertex v. That is, initially there are no constraints between the elements of a linkage, i.e. each element has $k(v)$ DOFs to move in the plane.

Each edge of G represents one constraint. Initially all constraints are inactive, i.e. all objects/vertices are unconstrained. An inactive constraint is represented by an undirected edge (constraint graph G is initially undirected). During the pebble game the constraints are successively activated by directing the edges. This indicates that the DOFs of one vertex are depending on the DOFs of other vertices. In the algorithm this is achieved by removing a pebble from one of its end-vertices. An undirected edge is termed admissible if the total of free pebbles next to its end vertices is at least four. Only admissible edges can be directed and can thus become active constraints.

Input to the Pebble Game algorithm: The algorithm starts from the topological graph, i.e. an undirected graph as described in Sect. 2. Each vertex v represents a physical object that has *k(v)* DOF.

The Pebble Game algorithm:

1. INITIALIZATION: Assign *k(v)* pebbles to each vertex v of the undirected graph, thus all edges are admissible. This is equivalent to regarding all mechanical objects, corresponding to the vertices, as unconstrained, i.e. each having *k(v)* DOFs.
2. WHILE there exist admissible edges DO the following **Orientation Move (Vertex—Edge move)**:
   Let (u, v) be an admissible edge, i.e., the total sum of pebbles next to the two end vertices is at least 4. Remove one pebble from one of its end vertices, let it be vertex u, and replace the edge by a directed edge $\langle u, v\rangle$, i.e., u becomes the tail and v the head vertex of $\langle u, v\rangle$.
   END WHILE
   After this loop there are no admissible edges left. This move corresponds to activating the constraint corresponding to the pebble removed from the tail vertex. The direction of the edge introduces a causality in the sense that one DOF of the tail vertex u is assumed to be dependent on one DOF of the head vertex v. Note that this is an abstract assignment, i.e. it is not said that a certain DOF of u is made dependent on a certain DOF of v.
3. WHILE there are free pebbles left DO the following **Reorientation move (Vertex—Vertex Move)**:
   Choose an undirected edge (u, v) and make it admissible by bringing free pebbles to its end vertices by applying the following steps ($peb(v)$ denotes the number of pebbles at vertex v):
   Suppose $peb(v) < 2$, if v stands for a point, or $peb(v) < 3$, if v stands for a body. Then search for a vertex, say z, with free pebbles, i.e., $peb(z) > 0$, for which there is a non-directed path from v to z. Then redirect all edges within the path from v to z so to form a directed path, and move one pebble from vertex z to v. Finally set $peb(z) := peb(z) - 1$ and $peb(v) := peb(v) + 1$.
   END WHILE

5 Example of Applying Mixed Pebble Game

In Fig. 2 we apply the mixed pebble game to the mixed graph representing the linkage in Fig. 1a. Initially, all the bodies and joints have three and two pebbles, respectively, as shown in Fig. 2a. The orientation move is first applied and all the admissible edges are directed. For example, the two edges (B_1, B_3) and (B_1, J_1) are admissible, thus

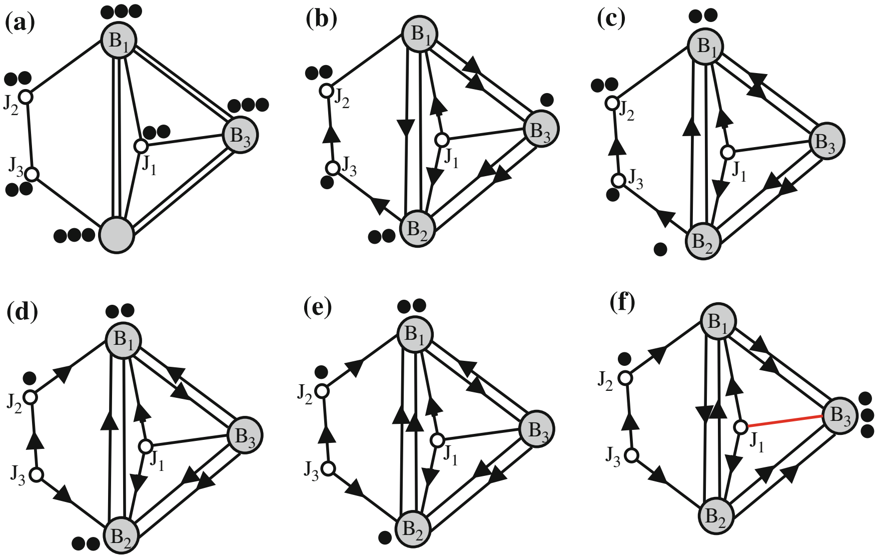


Fig. 2 Example of applying mixed Pebble game on mixed constraint graph

can be oriented, since there are 6 and 5 pebbles respectively next to the two end vertices. Fig. 2b shows all edges that could be directed by applying the orientation move. Since there are no more admissible edges the reorientation move is being applied next. For example, in Fig. 2c edge (J_2, B_1) becomes admissible by moving one pebble from vertex B_3 and one from B_2 thus it can be oriented as shown in Fig. 2d.

Applying reorientation move on edge $\langle B_2, J_3 \rangle$ brings a free pebble to vertex B_2 thus edge $\langle B_2, B_1 \rangle$ is now directed as shown in Fig. 2e.

Now we are left with four free pebbles and one edge, (J_1, B_3) unoriented. It is possible to move 3 pebbles next to any two end vertices, thus we move them to the end vertices of edge (J_1, B_3). For the sake of consistency, we move them to vertex B3 as shown in Fig. 2f.

In Fig. 2f there are no edges that can be made admissible by applying the reorientation move and the algorithm terminates. The output of the algorithm allows for the following interpretations:

Result 1: The most obvious result is the generic mobility of the associated linkage. Since the algorithm terminates with 4 free pebbles the planar linkage generically possesses 4 DOFs.

Result 2: Besides the generic mobility the particular location of the pebbles indicates which links can be moved independently, hence can be used as control inputs. As we deal with floating planar linkages there are always 3 DOFs that correspond to the relocation of the linkage as a whole. In this example there are 4 free pebbles. Each of the pebbles represents one DOF that can be independently controlled.

The specific allocation of pebbles in Fig. 2f, together with the original mechanism in Fig. 1a, allows for an apparent interpretation: the 3 DOFs of B_3 describe the location and orientation of the linkage in the plane, and the one pebble at J_2 is a translation DOF of the location point of J_2 that controls the internal shape.

Notice that

1. The pebble at J_2 is not the joint angle but one component of the location vector.
2. There is no specific assignment of coordinates to the DOFs so that ANY generalized coordinate can be used to represent the DOF of J_2. The pebble game algorithm operates on an abstract level and does not need specific selection of coordinates.
3. The particular allocation of pebbles is not unique and can be controlled in course of the algorithm. Also the algorithm's result can be changed by application of the reorientation move (which does not change the number of free pebbles). For instance, in Fig. 2f a free pebble is now assigned to vertex J_2. With a reorientation of $\langle J_2, J_3 \rangle$, this pebble can be moved to J_3. Now the one independent DOF is assigned to J_3.

6 Conclusions and Outlook

The paper introduces an efficient computational algorithm for determining the correct generic/topological mobility for any planar or spherical mechanism with higher and lower kinematic pairs, including multiple joints. The paper introduces a mixed constraint graph, which is a more general constraint graph than other graphs introduced in the literature, such as body-bar and bar-joint graphs. One of the salient conclusions of this paper is that, by using the mixed constraint graph, it is possible to represent any planar mechanism, and consequently to invoke the corresponding mixed pebble game algorithm. The latter is the main contribution of the paper: it determines the correct generic mobility of the mechanism modeled by a mixed constraint graph. The planar mixed Laman's theorem, which is an extension of the well-known Laman's theorem for bar-joint graphs, is given as a mathematical foundation of the algorithm. As mentioned in the paper, the novel mixed pebble game always converges to the correct generic mobility. Moreover it is discussed that this computational algorithm allows for decomposing any mechanism into its building blocks, namely Assur graphs. The reported algorithm applies to floating linkages, i.e. linkages that are not fixed to a ground. In a forthcoming publication, the mixed pebble game will be amended to include mechanisms (grounded mixed constraint graphs), which requires another type of constraint graphs. To this end the algorithm needs to be qualified so as to be able to treat immobile ground vertices.

References

1. Gogu, G.: Mobility of mechanism: a critical review. Mech. Mach. Theory **40**, 1068–1097 (2005)
2. Jacobs, D. J., Hendrickson, B.: An algorithm for two-dimensional rigidity percolation: the pebble game. J. Comput. Phys. **137**, 346–365 (1997)
3. Jain, A.: Graph theoretic foundations of multibody dynamics, Part I: structural properties. Multibody Syst. Dyn. **26**, 307–333 (2011)
4. Laman, G.: On graphs and rigidity of plane skeletal structures. J. Eng. Math. **4**, 331–340 (1970)
5. Lee, A., Streinu, I.: Pebble game algorithms and sparse graphs. Discret. Math. **308**(8), 1425–1437 (2008)
6. Müller, A.: Generic mobility of rigid body mechanisms. Mech. Mach. Theory **44**(6), 1240–1255 (2009)
7. Shai, O: Rigidity of 2d and 3d Pinned Frameworks and the Pebble Game, London. Mathematical Society Workshop: Rigidity of Frameworks and Applications, July 12–15, Lancaster University, England, (2010)
8. Shai, O., Sljoka, A., Whiteley, W.: Directed graphs, decompositions, and spatial rigidity. Discrete Appl. Math. Accepted 2012
9. Shai, O., Müller, A.: A novel combinatorial algorithm for determining the generic/topological mobility of planar and spherical mechanisms, In: ASME Design Engineering Technical Conferences, Portland, 12–15 Aug 2013
10. Sljoka, A.: Counting for Rigidity, Flexibility and extensions via the pebble game algorithm, Masters Thesis, York University 2006
11. Sljoka, A., Shai, O., Whiteley,W.: Checking mobility and decomposition of linkages via Pebble Game Algorithm, ASME Design Engineering Technical Conferences, In: Washington, USA, 28–31 Aug 2011
12. Tay, T.S.: Henneberg's method for bar and body frameworks. Struct. Topology **17**, 53–58 (1991)
13. Waldron, K.J.: The constraint analysis of mechanisms. J. Mech. **1**, 101114 (1966)
14. Waldron, K.J.: A study of overconstrained linkage geometry by solution of closure equations Part I. Method of study. Mech. Mach. Theory **8**, 95–104 (1973)
15. Waldron, K.J.: A study of overconstrained linkage geometry by solution of closure equations part II. Four-bar linkages with lower pairs other than screw joints. Mech. Mach. Theory **8**, 23–324 (1973)
16. Wampler, C., Larson, B.T., Erdman, A.G.: A new mobility formula for spatial mechanisms, In: Proceeding ASME International Design Engineering Technical Conferences (IDETC), **47**, Las Vegas, Sept (2007)
17. Wampler, C.W., Hauenstein, J.D., Sommese, A.J.: Mechanism mobility and a local dimension test. Mech. Mach. Theory **46**, 1193–1206 (2011)

Determination of the Safe Working Zone of a Parallel Manipulator

Rangaprasad Arun Srivatsan and Sandipan Bandyopadhyay

Abstract This paper formalises the concept of *safe working zone* (SWZ) of a parallel manipulator, which is a subspace of the workspace that is free of singularities as well as issues of joint limits and link interference. It presents further a generic scheme to identify such a space, and specialises the same for the case of a convex SWZ around a chosen point of interest. The theoretical developments are illustrated via an application on a three-degree-of-freedom spatial parallel manipulator, namely, MaPaMan-I.

Keywords Parallel mechanisms · Workspace · Singularities · Link interference · Joint limits

1 Introduction

Parallel manipulators (PMs) offer better load-carrying capacity and accuracy than their serial counterparts. Still, they are not as popular as the latter in the industries. This may be attributed mainly to the complicated kinematics of PMs, which in turn lead to small workspace volumes, rendered even smaller by the existence of *gain-type* singularities inside the workspace. In addition, joint limits and link interference further reduce the usable workspace.

Researchers have attempted to alleviate these problems in different ways. Some have attempted to design the robot such that the singularities are excluded [9, 13]. Others have tried to find regions inside the workspace that are free of singularities [3, 8]. The latter approach requires algebraic operations on the analytical description

R. A. Srivatsan · S. Bandyopadhyay (✉)
Department of Engineering Design, Indian Institute of Technology Madras, Madras, India
e-mail: rarunsrivatsan@gmail.com

S. Bandyopadhyay
e-mail: sandipan@iitm.ac.in

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_23,

of the singular manifold, which is very difficult in general, and may not be possible for all manipulators. This has motivated the development of some numerical schemes to find singularity-free zones inside the workspace [2, 6]. In [4, 5, 11], additional kinematic constraints, i.e., joint limits and link interference, have been considered.

The ultimate objective of all of these, and similar works, is to *identify a subset of the workspace, in which the manipulator can move freely*. This would render the task of path planning trivial, so long as the manipulator stayed inside the said space—which is very attractive from the point of view of applications. In this paper, such a space has been defined as the *safe working zone* (SWZ) of a PM. The criteria for the determination of the same, while considering the singularities, kinematic constraints etc., have been laid down clearly, which in turn have been used to develop a computational framework to compute the SWZ. The theoretical development is then illustrated by means of application to a newly introduced PM, namely, MaPaMan-I [12]. The scheme can be applied to any other PM, or even a serial manipulator.

The rest of the paper is organised in the following manner: in Sect. 2, the concept of SWZ is formalised. Various boundary functions specific to MaPaMan-1 are presented in Sect. 3, followed by the numerical results in Sect. 4. Finally the conclusions are presented in Sect. 5.

2 Definition and Structure of the SWZ

Various terms, such as *practical/desired/specific* workspace have been used in literature to designate subsets of the workspace, which are either free of singularities, kinematic constraints, or both (see, e.g., [4, 10]). Such confusion necessitates the formalisation of the definition of the SWZ.

Definition 2.1 The SWZ of a manipulator (denoted by $\mathcal{W}$) is defined as the subset of the workspace of the manipulator satisfying the following criteria:

1. $\mathcal{W}$ is contained *inside* the workspace, i.e., it is free of loss-type singularities.
2. $\mathcal{W}$ does not contain or touch the singular manifold, i.e., $\mathcal{W}$ is free of gain-type singularities as well.
3. At no point of $\mathcal{W}$ there is an interference between the links, even when the actual physical dimensions of the links are considered.
4. At no point of $\mathcal{W}$ does any joint violate a physical limit on its range of motion. Once again, the actual physical dimensions are to be considered.
5. $\mathcal{W}$ is a *connected* set, containing a given point of interest, '$\boldsymbol{o}$'.

The requirements 1–4 each define a subset of the workspace, which is bounded by the *zero level-set* of a corresponding function:

- The workspace (denoted by $\mathcal{W}_1$), is bounded by the loss-type singularity condition, given by $S_1 = 0$.
- The region $\mathcal{W}_2$ containing $\boldsymbol{o}$ and free of gain-type singularities, is bounded by the set of points defining the singular manifold which satisfy $S_2 = 0$.

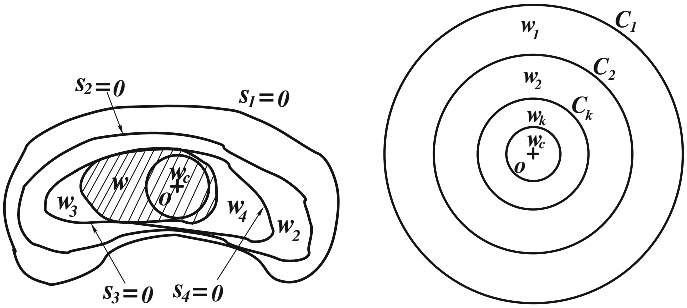


Fig. 1 Definition and structure of $\mathcal{W}$ and $\mathcal{W}_c$. *Left*: definition of $\mathcal{W}$, $\mathcal{W}_c$. *Right*: structural hierarchy of $\mathcal{W}_i$

- The region that includes $\boldsymbol{o}$ and is free of link interference is denoted by $\mathcal{W}_3$, and is bounded by the set satisfying $S_3 = 0$.
- The set of points satisfying $S_4 = 0$ bounds $\mathcal{W}_4$, the space containing $\boldsymbol{o}$ that is free of joint-limit violations.

As seen in Fig. 1-left, $\mathcal{W} = \bigcup_{i=1}^{4} \mathcal{W}_i$.[1] Note that in Fig. 1-right, $\mathcal{W}_k$ could be $\mathcal{W}_3$ or $\mathcal{W}_4$, or both. Physical considerations impose the following hierarchy: $\mathcal{W} \subset \mathcal{W}_3 \bigcup \mathcal{W}_4 \subset \mathcal{W}_2 \subset \mathcal{W}_1$. Moreover, it is generally preferred to identify a *convex set* $\mathcal{W}_c \subset \mathcal{W}$. These observations motivate a scheme for the computation of the final result, namely, $\mathcal{W}_c$. The steps are described below.

1. Compute $\mathcal{W}_1$. Find its *largest* convex subset, $\mathcal{W}_{c_1}$, centred at $\boldsymbol{o}$. The region can be in the form of convex polyhedra, super-ellipsoids, ellipsoids, etc. Without any loss of generality, and for the ease of computation, in this work *circles* have been used in the 2-dimensional subsets of the workspace (see Sect. 4). As shown in Fig. 1-right, the circle C_1 bounds $\mathcal{W}_{c_1}$.
2. In a similar manner, find C_2, which bounds $\mathcal{W}_{c_2}$. Obviously, $\mathcal{W}_{c_2} \subset \mathcal{W}_{c_1}$.
3. Compute the corresponding entities, namely, $\mathcal{W}_{c_3}$, C_3, $\mathcal{W}_{c_4}$, C_4 accordingly. Finally, find $\mathcal{W}_c = \mathcal{W}_{c_3} \bigcup \mathcal{W}_{c_4} \subset \mathcal{W}_{c_2}$.

The above steps are obvious, as it is useless to consider points *outside* $\mathcal{W}_{c_1}$ while computing $\mathcal{W}_{c_2}$, and so on. However, the implication of the hierarchy is very significant in the actual implementation of the scheme. Due to the lack/complexity of analytical results, most often, the set boundaries mentioned above need to be computed through numerical searches (see, e.g., [2, 4, 7, 10, 11]), and hence the progressively diminishing domain for the search algorithm helps reducing the computational requirements for a given desired level of resolution of the results obtained.

[1] Note that not all manipulators have all the four requirements. In all serial manipulators, $\mathcal{W}_1 = \mathcal{W}_2$, and as explained in Sect. 3, for MaPaMan-I, $\mathcal{W}_3 = \mathcal{W}_4$.

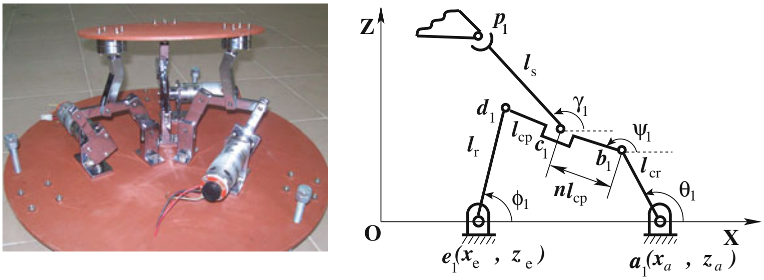


Fig. 2 Prototype (*left*) and kinematic details of a leg (*right*) of MaPaMan-I

3 Formulation of the SWZ of MaPaMan-I

The generic theoretical framework described in Sect. 2 is illustrated in this section by an application to the newly developed MaPaMan-I. It so happens that for the physical dimensions of the present prototype described in [12], the joints have limits on their motions, but there is no other form of link interference. Thus, the computation of $\mathcal{W}_3$ (or, $\mathcal{W}_{c_3}$) is not required.

MaPaMan-Ihas 3-degrees-of-freedom similar to 3-R$\underline{\text{P}}$Sand its task-space can be parametrised in terms of roll (α), pitch (β) and heave (z_c) [12]. The coordinates of the end-effector $\boldsymbol{p}_i$ ($i = 1, 2, 3$) are obtained from the task-space coordinates $\boldsymbol{x} = (\alpha, \beta, z_c)^T$. The input joint angles are: $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^T$, and the passive joint angles are: φ, ψ and γ (see Fig. 2).

3.1 *Condition for Loss-Type Singularity (S_1)*

Following [14], kinematic constraints are first framed to relate the task-space coordinates to the input coordinates. The length of the strut, l_s, is fixed, hence the *loop-closure* constraints can be cast as: $f_i(\theta_i, \psi_i, \boldsymbol{x}) \overset{\triangle}{=} \|\boldsymbol{b}_i - \boldsymbol{p}_i\| = l_s$, $i = 1, 2, 3$. Likewise, the loop-closure equations for the four-bars (see Fig. 2), upon elimination of the passive variable φ_i, become [12]:

$$\begin{aligned} g_i(\theta_i, \psi_i) = {} & l_0^2 + l_{cr}^2 + l_{cp}^2 - l_r^2 + 2l_0 l_{cr} \cos\theta_i + 2l_0 l_{cp} \cos\psi_i \\ & + 2l_{cr} l_{cp} \cos\theta_i \cos\psi_i + 2l_{cr} l_{cp} \sin\theta_i \sin\psi_i . \end{aligned}$$

From each pair of f_i and g_i, the passive variable ψ_i is eliminated to obtain $h_i(\theta_i, \boldsymbol{x}) = 0$, $i = 1, 2, 3$. The condition for loss type singularity is given by $S_1 = 0$, where $S_1 = \det\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}}\right)$, and $\boldsymbol{h} = (h_1(\theta_1, \boldsymbol{x}), h_2(\theta_2, \boldsymbol{x}), h_3(\theta_3, \boldsymbol{x}))^T$.

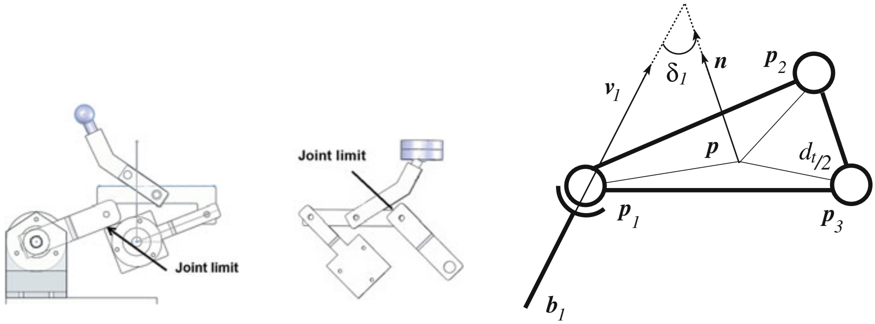


Fig. 3 Joint limits imposed by the physical dimensions of MaPaMan-I. *Left*: joint limit at the crank. *Center*: joint limit at strut-coupler joint. *Right*: motion limit at the spherical joint

3.2 Condition for Gain-Type Singularity (S_2)

Following [12], the loop-closure constraints are cast in the form $\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\gamma}) = \mathbf{0}$, which upon time-differentiation yield $\dot{\boldsymbol{\eta}}(\boldsymbol{q}) = \boldsymbol{J}_{\eta\theta}\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\eta\psi}\dot{\boldsymbol{\psi}} + \boldsymbol{J}_{\eta\gamma}\dot{\boldsymbol{\gamma}}$, where $\boldsymbol{J}_{\eta\theta} = \frac{\partial\eta}{\partial\boldsymbol{\theta}}$, $\boldsymbol{J}_{\eta\psi} = \frac{\partial\eta}{\partial\boldsymbol{\psi}}$, and $\boldsymbol{J}_{\eta\gamma} = \frac{\partial\eta}{\partial\boldsymbol{\gamma}}$. Considering the four-bar alone, $\dot{\boldsymbol{\psi}} = \boldsymbol{J}_{\psi\theta}\dot{\boldsymbol{\theta}}$, where $\boldsymbol{J}_{\psi\theta}$is always well-defined, since by design, the said four-bars satisfy Grashoff's condition. Therefore, $(\boldsymbol{J}_{\eta\theta} + \boldsymbol{J}_{\eta\psi}\boldsymbol{J}_{\psi\theta})\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\eta\gamma}\dot{\boldsymbol{\gamma}} = \mathbf{0}$. The gain-type singularity occurs when the *passive velocity*, $\dot{\boldsymbol{\gamma}}$, cannot be found uniquely for a given $\dot{\boldsymbol{\theta}}$ [1]. This leads $S_2 = 0$, where $S_2 = \det(\boldsymbol{J}_{\eta\gamma})$.

3.3 Condition for the Violation of Joint Limits (S_4)

The issue of the joints reaching their limits in the range of motion is observed at the cranks, the strut-coupler rotary joints, and the spherical joints attached to the end-effector. The following are various limiting conditions for the same:

- The crank is designed such that it is always above the base of the manipulator. From practical considerations, a restriction is imposed upon the maximum and the minimum angle of rotation of the crank denoted by θ_{min}, and θ_{max}, respectively (see Fig. 3-left). Thus, $0 < \theta_i < \theta_{max}$, $i = 1, 2, 3$. Hence the conditions defining the boundary of unacceptable points are $s_{1i} = 0$ and $s_{2i} = 0$, where $s_{1i} = \theta_i - \theta_{min}$, $s_{2i} = \theta_{max} - \theta_i$.
- The angle made by the strut relative to the coupler is limited by the physical joint limits as shown in Fig. 3-center, denoted by γ_{min} and γ_{max}, where $\gamma_{min} < (\pi + \gamma_i - \psi_i) < \gamma_{max}$, $i = 1, 2, 3$. Thus the conditions defining the boundary of the unacceptable sets are: $s_{3i} = 0$, and $s_{4i} = 0$; where $s_{3i} = (\pi + \gamma_i - \psi_i) - \gamma_{min}$, $s_{4i} = \gamma_{max} - (\pi + \gamma_i - \psi_i)$.

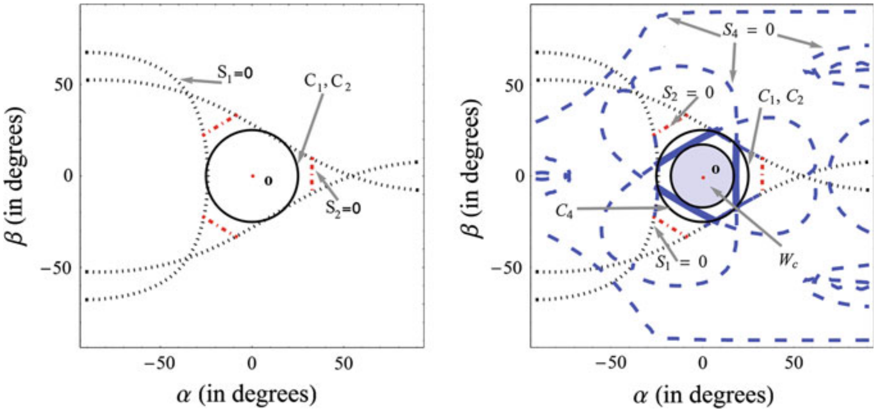


Fig. 4 Zero level-sets of S_i, and C_i in the α–β slice of the workspace at $z_c = 135\,\text{mm}$

- The spherical joints have restricted motions due to the physical dimensions of their constituent mechanical components. This can be modelled as a limit imposed on the angle δ_i, such as $0 < \delta_i < \delta_{max}$ (see Fig. 3-right). The angle δ_i is computed by first finding a vector along the direction of strut (v_1) and then measuring the angle between it and the normal ($\boldsymbol{n}$) to the end-effector: $v_1 = (\boldsymbol{b}_1 - \boldsymbol{p}_1)/l_s$, $\boldsymbol{n} = (\boldsymbol{p}_1 - \boldsymbol{p}_2) \times (\boldsymbol{p}_3 - \boldsymbol{p}_1)/\frac{\sqrt{3}d_t^2}{2}$ and $\delta_1 = \arccos(\boldsymbol{n} \cdot v_1)$. Similarly, δ_2 and δ_3 are computed. Therefore the functions defining the boundary of the desirable set are given by $s_{5i} = 0$, where $s_{5i} = \delta_{max} - \delta_i$. The function S_4 is obtained from the product of individual functions: $S_4 = \prod s_{ij}$, where $i = 1, \ldots, 5$, and $j = 1, 2, 3$.

4 Numerical Results

This section describes the results of the application of the above formulation to the MaPaMan-Iprototype, whose dimensions are given in Table 1 of [12]. The joint limits used are: $\theta_{min} = 25°$, $\theta_{max} = 90°$, $\gamma_{min} = 0°$, $\gamma_{max} = 120°$, $\delta_{max} = 60°$. The task-space of MaPaMan-Iis parametrised by (α, β, z_c). The functions S_i are not available solely in terms of these variables. Therefore instead of direct computation of the zero level-sets, these are found by using a numerical scheme similar to those used in [2, 4]. Since it is computationally demanding to search for solutions of $S_i = 0$ in a 3-dimensional space, 2-dimensional slices in roll and pitch are considered instead, and the solutions are evaluated in these slices for a sequence of heave values.[2] Figure 4 shows the zero level-sets of S_1, S_2 in the α–β plane obtained by slicing the workspace at $z_c = 135\,\text{mm}$. Note that the points satisfying $S_2 = 0$ fall outside $\mathcal{W}_{c_1}$, and therefore in this particular case, $C_2 = C_1$.

[2] Zero level-sets of S_1, S_2, S_4 have been computed using ContourPlot in Mathematica.

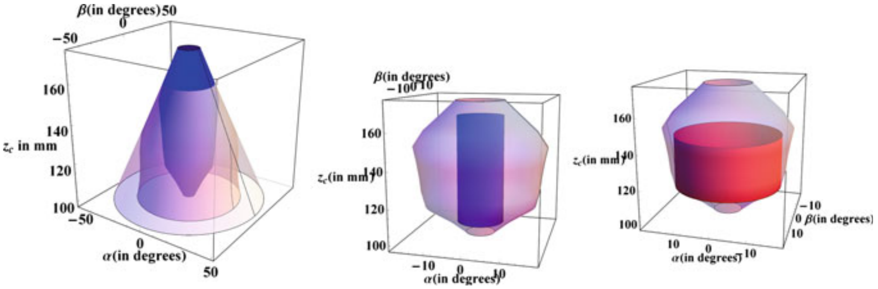


Fig. 5 *Left*: stack of C_1, C_2, C_4 in MaPaMan-I, *Middle*: $\mathcal{W}_c$ as a cylinder for $z_c \in (98.4, 166.5)$ mm, *Right*: $\mathcal{W}_c$ as a cylinder for $z_c \in (111.3, 145.2)$ mm

The zero level-set of $S_4 = 0$ is shown in the entire scan range in Fig. 4. However, only the parts of it appearing inside $\mathcal{W}_{c_2}$ (marked by thicker lines in Fig. 4) are considered for the computation of $\mathcal{W}_{c_4}$. As noted earlier, in this case, $\mathcal{W}_c = \mathcal{W}_{c_4}$. Naturally, the stack of C_4 obtained for all the slices when put together yields a subset of $\mathcal{W}$, that is convex in each slice. This does not necessarily imply that the stack delimits a convex region as a whole. However, one can easily fit a desired convex shape to obtain $\mathcal{W}_c$ in the $(\alpha - \beta - z_c)$ space. Due to the nature of the degree-of-freedomof the manipulator, a cylinder is chosen as the convex shape to be fit inside $\mathcal{W}$. Figure 5 shows the stack of C_1, C_2 and C_4 together for the manipulator under consideration. Note how they follow the hierarchy described in Sect. 2. As the stack of C_4 has been obtained for the entire range of heave, $\mathcal{W}_c$ can be obtained by fitting a convex shape to the stack for any desired subset of the complete range of heave. Note that the radius of the cylinder in the former case is 6.59° while it is 15.62° in the latter (see Fig. 5). Thus, based on the intended application, a convex shape of interest can be fit into $\mathcal{W}$ to obtain $\mathcal{W}_c$ desired.

5 Conclusion

In this paper, the concept of a safe working zone of a parallel manipulator has been formalised, and a generic framework has been presented for its computation. It has been shown that considering a convex subset of the same can lead to a hierarchy in the subsets leading to the final result, reducing the computational requirements significantly in the process. The formulation has been demonstrated by means of an application on a newly developed parallel manipulator, namely MaPaMan-I, whose workspace and singularities were not reported previously. A single computational tool, namely a contour-plotter in a 2-dimensional space, has been used to compute the SWZ and its convex subsets. The generic and simple nature of the scheme presented in this paper can help in identifying the SWZ in similar manipulators, as well as be used in the design of manipulators for a desired SWZ.

References

1. Bandyopadhyay, S., Ghosal, A.: Analytical determination of principal twists in serial, parallel and hybrid manipulators using dual number algebra. Mech. Mach. Theory **39**(12), 1289–1305 (2004)
2. Bohigas, O., Zlatanov, D., Ros, L., Manubens, M., Porta, J.: Numerical computation of manipulator singularities. In: Proceedings of the 2012 ICRA, IEEE Computer Society Press St. Paul, USA, 1351–1358 (2012)
3. Bonev, I.A., Gosselin, C.M.: Singularity loci of planar parallel manipulators with revolute joints. In: Proceedings of the 2nd Workshop on Computational Kinematics, Seoul, pp. 291–299 May 2001
4. Cao, Y., Huang, Z., Zhang, Q., Zhou, H., Ding: Orientation-workspace analysis of the Stewart–Gough manipulator. In: Proceedings of the ASME DETC 2005/MECH-84556, California, Sept 2005
5. Chablat, D., Wenger, P.: Moveability and collision analysis for fully-parallel manipulators. 12th CISM-IFTOMM Symposium, pp. 1–8 (1998)
6. Huag, E., Luh, C.M., Adkins, F.A., Wang, J.Y.: Numerical algorithms for mapping boundaries of manipulator workspaces. J. Mech. Des. **118**(2), 228–234 (1996)
7. Hudgens, J., Arai, T.: Planning link-interference-free trajectories for a parallel link manipulator. In: Proceedings of the IECON '93, Hawaii, vol. **3**, pp. 1506–1511, Nov 1993
8. Jiang, Q., Gosselin, C.M.: The maximal singularity-free workspace of the Gough–Stewart platform for a given orientation. J. Mech. Des. **130**(11), 112–303 (2008)
9. Liu, X.J., Wang, J.S., Gao, F.: On the optimum design of planar 3-dof parallel manipulators with respect to the workspace. In: Proceedings of the 2000 ICRA, San Francisco, vol. **4**, pp. 4122–4127 April 2000
10. Merlet, J.P.: Designing a parallel manipulator for a specific workspace. Int. J. Robot. Res. **16**(4), 545–556 (1997)
11. Pernkopf, F., Husty, M.L.: Workspace analysis of Stewart-Gough-type parallel manipulators. J. Mech. Eng. Sci. **220**(7), 1019–1032 (2006)
12. Srivatsan, R.A., Bandyopadhyay, S.: On the position kinematic analysis of MaPaMan: A reconfigurable three-degrees-of-freedom spatial parallel manipulator. Mech. Mach. Theory **62**(4), 150–165 (2013)
13. Yang, Y., O'Brien, J.: A geometric approach for the design of singularity-free parallel robots. In: Proceedings of the 2009 ICRA, Kobe, pp. 1801–1806 (2009)
14. Zlatanov, D., Bonev, I.A., Gosselin, C.M.: Constraint singularities of parallel mechanisms. In: Proceedings of the 2002 ICRA, Washington DC, pp. 496–502 (2002)

Certified Calibration of a Cable-Driven Robot Using Interval Contractor Programming

Julien Alexandre dit Sandretto, Gilles Trombettoni, David Daney and Gilles Chabert

Abstract In this paper, an interval based approach is proposed to rigorously identify the model parameters of a parallel cable-driven robot. The studied manipulator follows a parallel architecture having 8 cables to control the 6 DOFs of its mobile platform. This robot is complex to model, mainly due to the cable behavior. To simplify it, some hypotheses on cable properties (no mass and no elasticity) are done.An interval approach can take into account the maximal error between this model and the real one. This allows us to work with a simplified although guaranteed interval model. In addition, a specific interval operator makes it possible to manage outliers. A complete experiment validates our method for robot parameter certified identification and leads to interesting observations.

Keywords Cable-driven robots · Calibration · Interval analysis

1 Introduction

Due to tolerances in manufacturing or assembly, the geometry of the actual manipulator does not correspond to the desired design and its theoretical kinematic model. Consequently, the performances of the manipulator, such as its accuracy, decrease.

J. A. dit Sandretto (✉) · D. Daney
Coprin INRIA, Valbonne, France
e-mail: Julien.Alexandre_dit_Sandretto@inria.fr

D. Daney
e-mail: David.Daney@inria.fr

G. Trombettoni
LIRMM, University Montpellier 2, Montpellier, France
e-mail: Gilles.Trombettoni@lirmm.fr

G. Chabert
LINA, Ecole des Mines de Nantes, Nantes, France
e-mail: Gilles.Chabert@mines-nantes.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_24,

This problem could be bypassed by improving the kinematic model, i.e., by finding the actual values of its parameters. These parameters, defining the geometry of the robot (frame and platform) and actuator parts, are provided by a kinematic calibration procedure. Calibration consists in identifying model parameters through redundant information on the state of the robot provided by measurements or constraints [13]. Cable-driven robots have several interesting properties like reduced mass of moving parts (i.e., the cables), ease of reconfiguration and, more importantly, a potentially very large workspace. Nevertheless, the cost of these advantages is a complex kinematic and dynamic behavior due to the flexibility, mass and elasticity of the cables. This kind of complex numerical model, can (and often needs to) be simplified to be used in a command process or a calibration for instance. For example, a kinematic calibration with joint sensors has been reported in [3], whereas a self-calibration procedure for a planar robot is introduced in [2]. Several studies have been carried on cable-driven robot kinematics. Many of them use an approximate model for cables which considers them without mass and elasticity. The error added by this approximation has been bounded for a whole workspace and a range of usual forces in [12]. In robotics, interval analysis is used to manipulate bounded uncertainties [9, 10], or to consider the whole workspace of a robot. A recent approach called contractor programming was proposed by Chabert and Jaulin to solve different interval problems [4]. A contractor is a procedure that reduces the search space, considering generally a set of constraints to satisfy. In the face of outliers, i.e., measures that are completely wrong, a solution which satisfies all the constraints may not exist. An original interval approach for calibration was proposed by one the authors in [5]. However, outliers (along with model errors) led their method to an empty solution. This paper proposes an innovative calibration approach based on certified interval methods. The approach allows the calibration process to manage uncertainties in cable model and in measures, even in presence of outliers thanks to a recent interval operator called q-intersection. This intersection can relax some constraints to escape from outliers.

2 Cable-Driven Robot

This study is part of the CoGiRo (Conception of Giant Robot) project. The goal of this national project is to create a traveling crane with a large workspace and $n = 6$ DOFs. A crane or cable-driven robot is a mechanism that controls $m = 8$ cables whose length or tension provides a movement of the mobile with 6 DOFs w.r.t. a reference.

Cable-driven robot architecture:

In the sketch presented in Fig. 1, the mobile (linked to the frame Ω_C) is connected to the base (linked to the frame Ω_O) by $m = 8$ cables ($m > n$ to be fully controllable). The ith cable connects the point A_i in the base (coordinate a_i in Ω_O) to the point B_i on the mobile (coordinate b_i in Ω_C). The pose of the mobile (defined by the position P and the orientation R of Ω_C expressed in Ω_O) is directly controlled by the length

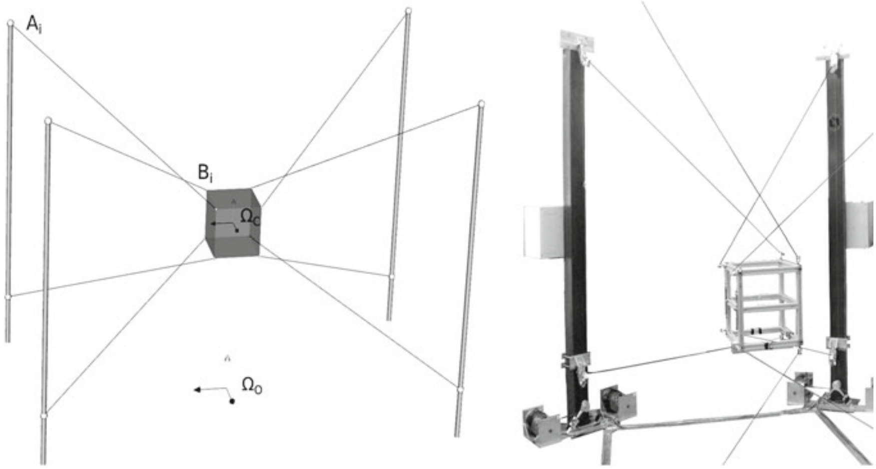


Fig. 1 A cable-driven robot sketch and a `ReelAx8` picture

and the tension of each cable. The prototype, named *ReelAx8* and shown in Fig. 1, was built by the TECNALIA company (`www.tecnalia.com`) in collaboration with the LIRMM laboratory (`www.lirmm.fr`). Eight cables, wound on winches, are attached by spherical joints to the eight corners of a cube shaped platform of about 40 centimeters large. Four pairs of winches are fixed on posts up to three meters high arranged at the four corners of a 3 m by 4 m rectangle.

Interval based kinematic model:

Cable-driven robots take advantage of the use of cables, allowing large workspace, light actuators compared to the possible load mass, and low cost. However, cable-driven robots suffer from the complex kinematics and dynamics of cables. A well-known realistic model often used for the kinematics of cables is proposed by Irvine in [7]. In Irvine's model, the length of a cable depends on its tension and is described by a non linear system of equations. In order to use this realistic model, we need more sensors to measure the tension in cables, but the force sensors are not accurate enough. The hypothesis considering non-elastic and mass-less cables is very useful to simplify control, modeling, calibration, etc. Moreover, it is often realistic and adds a negligible error to the robot accuracy. The majority of papers about cable-driven robots uses this hypothesis and replaces the distances between points A_i and B_i, noted $d(A_i, B_i)$, by the length L_i, $i = 1 \dots m$, of unwound cables. The error between the real length of cable and the distance $d(A_i, B_i)$ has been quantified (and bounded with certification) in [12].

This method evaluates the interval cable length error $[\Delta L] = [\Delta L_{min}, \Delta L_{max}]$ between the realistic Irvine model L_{irvine} (which depends on the tension in cables and considers mass and elasticity of cables) and the strongly simplified model $L = d(A, B)$ over the whole robot workspace and for the whole range of possible tensions in cables. An interval $[x] = [\underline{x}, \overline{x}]$ defines the set of reals x s.t. $\underline{x} \leq x \leq \overline{x}$. For a given pose we can write: $\Delta L_{min} < L - L_{irvine} < \Delta L_{max}$

We can thus define an interval based kinematic model which deals with this modeling error:

$$||P + Rb_i - a_i||_2 - (L_i + [\varDelta L]) \supset 0, \quad i = [1...m] \tag{1}$$

The simplified model approximates the realistic cable by adding the error $[\varDelta L]$. Thus we can also write $L_{irvine} \in L + [\varDelta L]$. This model is therefore correct because it contains the real one. In other terms, *our interval model overestimates the real model in a guaranteed way*. It is relevant for calibration because the intersection principle can reduce this overestimation.

Calibration of a Cable-driven Robot:
The goal of calibration is to enhance the robot performances by the model knowledge improvement. This improvement consists in identifying the model parameters through redundant information on the state of the robot provided by measurement. In case of calibration, N_C measurements of P and R are achieved, for example, with a laser tracker, and are given with tolerances. Therefore $[P]$ and $[R]$ include these tolerances, i.e., the intervals $[P]$ and $[R]$ contain all the possible values for the position and orientation. The coordinate of the point B_i expressed in $\varOmega_O$ at the kth pose is then: $[b_i^k] = [P_k] + [R_k][b_i]$. The coordinates b_i which define the geometry of the platform are also given with manufacturing or measuring tolerances. In this paper, we focus on the A_i coordinate identification. In practice, in the case of big cable-driven robots, the parameters b_i, which describe the platform geometry, are well known, are easy to measure and do not change, contrarily to the a_i which change at each new reconfiguration and are not easy to be measured without a laser due to the large dimension of the robot (4 m in our case). $[a_i]$ denote the coordinates of points A_i. These points are unknown, but we can bound them, even not accurately. The aim of the calibration process described in this paper is to reduce these uncertainties. The identification scheme consists of handling m independent and over-constrained systems of N_C equations:

$$0 \in ||[b_i^k] - [a_i]||_2 - ([L_i^k] + [\varDelta L]), \quad k = [1...N_C], i = [1...m] \tag{2}$$

Due to the independence of these m systems, each a_i can be identified separately. The nature of these calibration equations, where the only variable is a_i, leads to an analogy with a thick sphere intersection problem (N_C thick spheres of center B_i^k, of diameter $[L_i^k] + [\varDelta L]$ intersected in A_i). In the sake of clarity of pictures that follow, the problem will be illustrated by a circle (ring) intersection in a plane.

This calibration model takes into account all the *bounded* uncertainties on modeling, measurements and parameters. Nevertheless it could happen that a measurement is completely wrong, i.e., beyond the estimated tolerances. Several circumstances can lead to a mismeasurement, such as a sensor break down or bad environmental conditions. Such a measure is called an outlier. If it is the cth measure, for example, it means that:

$$0 \notin ||[b_i^c] - [a_i]||_2 - ([L_i^c] + [\varDelta L]), \quad i = [1...m] \tag{3}$$

In the presence of outliers, a classic constraint solver will fail to find a solution. In this paper, we propose to use a recent interval operator called q-intersection which can manage these outliers.

3 Contractor Programming and q-Intersection Contractor

The calibration process suffers from two problems: the acquisition of measures and the model used for identification. The interval approach offers a rigorous representation of the uncertainties on measurements and a powerful ability to rigorously enclose the solutions. Significant theoretical and algorithmic progress have been brought to interval methods that are now mature for application [6, 9]. A good introduction to the interval arithmetic can be found in these last two references. An interval solver is based on a branch & bound. A simple solver evaluates all the functions involved in the system of constraints handled and checks if their images contain 0. To improve this approach, contractors are used.

A *contraction procedure* or *contractor* accepts as input a *box*, i.e., a vector of intervals, and outputs a contracted box (while respecting certain properties detailed in [4]). A given contractor can accept any other parameter to specify it behavior, such as a system of constraints. A contractor program includes calls to several contractors applied to different boxes. The contractor program used in our calibration applies the following scheme to the system of equations modeling our measures:

$$[FixP(QInter(q = 75\%, hc4r(F_1, box), ..., hc4r(F_{N_C}, box))$$

where $hc4r$ [1], $QInter$ [8] and $FixP$ are three contractors.

- $hc4r$ can contract a box w.r.t. a single constraint such that no solution of the constraint is lost in the box. Using a tree representation of the constraint for accelerating the contraction, this contractor isolates every occurrence x_i in the expression and performs a natural evaluation of the corresponding function to contract $[x_i]$. Applied to our ring intersection problem, $hc4r$ intersects the studied box with a ring corresponding to a measure.
- $QInter$ can forget bad measures (outliers), provided that the number of good measures exceeds q. This contractor returns the smallest box including all the points that belong to at least q of the boxes in argument. Applied to our ring intersection problem, $QInter$ returns the box enclosing all the points in the studied box satisfying at least q imprecise measures.
- $FixP$ (*FixPoint*) calls the contractor in argument (here $QInter$) until the box reaches a quasi fixed point in terms of size.[1]

[1] In our tool, the process stops if the maximal reduction on every dimension does not exceed 1%.

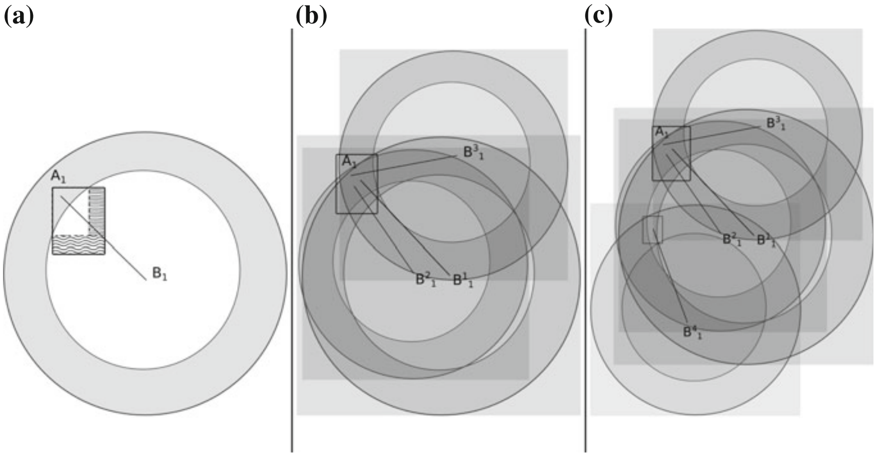


Fig. 2 **a** A pose contracting the box around A_1, **b** A *circle* intersection, **c** An empty *circle* intersection due to an outlier

4 Interval Calibration Approach

Our tool is dedicated to calibration and uses contractor programming to find the m boxes that are guaranteed to contain the points A_i, $[i = 1 \dots m]$. The method is explained in details in the following.

Contractor for each pose:

For each pose k (measure of position, orientation and cable length), the $hc4r$ contractor reduces the size of the box including the point A_i using the calibration constraints $F_{i,k} = ||[b_i^k] - [a_i]||_2 - [L_i^k] + [\Delta L]$. The resulting box of this contraction process is smaller than the initial box, but it is always an overestimation of the coordinate a_i^k (obtained with the kth pose). This is illustrated in the plane (for the sake of clarity) in Fig. 2a.

Calibration with q-intersection:

Our calibration procedure is based on the intersection principle. The interval paradigm ensures that every pose a_i^k contains the point A_i, depending on the corresponding measurement. Therefore the different boxes resulting from the different poses a_i^k are intersected, thus offering a better estimation of the point A_i. In Fig. 2b, we present the intersection of 3 rings corresponding to poses (B_i^k represents the point B_i in the kth pose). Each rectangle in grey represents a contracted box corresponding to one pose, and the final box results from the intersection of all the individual boxes.

A problem can appear in this intersection process. In practice, due to the measure acquisition process, a value may be completely wrong (e.g., B_1^4 in Fig. 2c), even if one considered the maximal possible error allowed by the measurement device. These outliers generally lead to an empty solution for a classical intersection method, as shown in Fig. 2c. In order to avoid this phenomenon (trap), our calibration approach

Table 1 Parameters b_i (in meter) estimated with ±1 mm tolerance

b_i	x	y	z
b_1	[−0.112,−0.110]	[0.345,0.347]	[−0.017,−0.015]
b_2	[0.241,0.243]	[0.337,0.339]	[−0.028,−0.026]
b_3	[−0.171,−0.169]	[0.339,0.341]	[−0.431,−0.429]
b_4	[0.298,0.300]	[0.329,0.331]	[−0.446,−0.444]
b_5	[−0.179,−0.177]	[−0.031,−0.029]	[−0.429,−0.427]
b_6	[0.288,0.290]	[−0.041,−0.039]	[−0.437,−0.435]
b_7	[−0.118,−0.116]	[−0.027,−0.025]	[−0.004,−0.002]
b_8	[0.233,0.235]	[−0.034,−0.032]	[−0.015,−0.013]

uses a q-intersection contractor. This method, presented in Sect. 3, can reject some poses considered as outliers (e.g., B_1^4), even if we do not know in advance which ones are outliers. This works in the calibration of our robot provided that the parameter q has been set to more than 75%. If the q-intersection result is nevertheless empty, additional measurements have to be achieved to obtain a correct calibration result. In a very unsettled environment, or with non accurate tools, it is also possible to choose a smaller value q, although the calibration process has no sense when q falls below 50%.

Method experimentally applied:

After a complete simulation which validated the approach, we have performed a validation with measures coming from experiment. The cable model error computed with the method presented in [12] is $[\Delta L] = [-0.6, 3.7]$ millimeters. The calibration is performed with 42 measures obtained with a laser tracker. The tolerances of this tool is ±1 mm for the position (on x, y and z) and ±10 mRad for rotation (around each axis). An estimation on a_i is given with a tolerance of ±1 cm. The calibration results are reported in Tables 1, 2. The solution given by a least square method (described in [11]) falls outside the certified interval solution obtained with our method. For example, a classical method computes $a_1 = (2.2046, 0.5964, -2.7184)$. This means that the least square solution does not satisfy the calibration equations.

Table 2 Parameters a_i (in meter) identified with interval method and cable model error

a_i	x	y	z
a_1	[2.214,2.214]	[0.604,0.606]	[−2.728,−2.727]
a_2	[2.199,2.219]	[0.516,0.536]	[−0.367,−0.347]
a_3	[0.512,0.532]	[2.135,2.155]	[−2.711,−2.691]
a_4	[0.536,0.556]	[2.193,2.205]	[−0.425,−0.405]
a_5	[−1.955,−1.955]	[−0.418,−0.415]	[−2.726,−2.725]
a_6	[−2.063,−2.043]	[−0.375,−0.355]	[−0.438,−0.418]
a_7	[−0.310,−0.309]	[−2.020,−2.020]	[−2.727,−2.727]
a_8	[−0.392,−0.375]	[−2.045,−2.043]	[−0.403,−0.398]

Using a standard interval approach, the solution of the calibration is empty due to the presence of outliers. We can also observe that $[a_{3,6}]$ are not contracted by our method. This means that the measures are not well chosen for calibrating these parameters (identifiability problem). The measures are numerous but they do not cover all the robot workspace. Remember that in this approach, the more different correct poses we have, the smaller the final box will be. The stopping criterion is given by the time spent to make the measures, but the choice of the poses is very important [5].

5 Discussion and Conclusion

This paper presents the first attempt to achieve a certified calibration of a cable-driven robot using an innovative contractor programming approach. Our interval method is robust to measurement error and outliers. Also, it permits to capture an error due to the modeling. We can use a highly simplified model for the calibration that is guaranteed to contain the realistic one (even if the latter cannot be characterized). We have employed a q-intersection contractor to obtain an outlier-resistant calibration. The obtained solution is not a point A_i, like with the least square method, but a box. This box is an overestimated enclosure of the point to be identified. We could verify that this solution is close to the one obtained by the (non certified) least square method. To summarize, this contractor programming approach holds the potential to become a key feature in certified calibration.

Acknowledgments This work was supported in part by the French National Research Agency (ANR) under grant 2009-SEGI-018 (CoGiRo ANR project) and in part by the Région Languedoc-Roussillon under grant 115217.

References

1. Benhamou, F., Goualard, F., Granvilliers, L., Puget, J.F.: Revising hull and box consistency. In: Proceedings of ICLP, pp. 230–244 (1999)
2. Borgstrom, P., Jordan, B., Borgstrom, B., Stealey, M., Sukhatme, G., Batalin, M., Kaiser, W.: Nims-pl: a cable-driven robot with self-calibration capabilities. IEEE Trans. Robotics **25**(5), 1005–1015 (2009)
3. Bouchard, S.: Geometrie des robots paralleles entraines par des cables. Ph.D. thesis, Universite Laval, Quebec (2008)
4. Chabert, G., Jaulin, L.: Contractor programming. Artif. Intell. **173**, 1079–1100 (2009)
5. Daney, D., Andreff, N., Chabert, G., Papegay, Y.: Interval method for calibration of parallel robots: vision based experiments. Mech. Mach. Theor **41**, 929–944 (2006)
6. Hansen, E.R.: Global Optimization Using Interval Analysis. Marcel Dekker Inc., New York (2003)
7. Irvine, H.M., Caughey, T.K.: The linear theory of free vibrations of a suspended cable. Proc. Royal Soc. London, Ser. A, Math. Phys. Sci. **341**(1626), 299–315 (1974)
8. Jaulin, L.: Robust set-membership state estimation; application to underwater robotics. Automatica **45**(1), 202–206 (2009)

9. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis. Springer, London (2001)
10. Moore, R.E.: Interval Analysis. Prentice-Hall, Englewood Cliffs (1966)
11. Sandretto, J.A.D., Daney, D., Gouttefarde, M.: Calibration of a fully-constrained parallel cable-driven robot. In: RoManSy, Paris (2012)
12. Sandretto, J.A.D., Trombettoni, G., Daney, D.: Confirmation of hypothesis on cable properties for cable-driven robots. In: Eucomes, Santander (2012)
13. Wampler, C., Arai, T.: Calibration of robots having kinematic closed loops using nonlinear least-squares estimation. In: Proceedings of the IFToMM-jc International Symposium Theory Machine Mechanics pp. 153–158 (1992)

Framework Comparison Between a Multifingered Hand and a Parallel Manipulator

Júlia Borràs and Aaron M. Dollar

Abstract In this paper we apply the kineto-static mathematical models commonly used for robotic hands and for parallel manipulators to an example of hand-plus-object (parallel manipulator) with three fingers (legs), each with two phalanges (links). The obtained analytical matrix expressions that define the velocity and static equations in both frameworks are shown to be equivalent. This equivalence clarifies the role of the grasp matrix versus the parallel manipulator Jacobian. Potential knowledge transfer between both fields is discussed in the last section.

Keywords Parallel mechanisms · Multifingered robotic hands · Screw theory.

1 Introduction

A hand manipulating an object held in the fingertips has the same kinematic structure as a parallel manipulator where the platform is the object and the legs are the fingers. Despite this fact has been acknowledged by many authors [6, 10], few works discuss connections between the mathematical frameworks of both systems [4]. A hand-plus-object system is a highly redundant hybrid parallel manipulator, where the only passive joints are the contact attachments. However, the hand-plus-object system has to hold an extra condition: the fingertip force has to be directed towards the object and inside the friction cone [12]. This condition does not modify the kineto-static mathematical model, because it is treated as a constraint when solving the static equations.

J. Borràs (✉) · A. M. Dollar
Department of Mechanical Engineering and Materials Science, Yale University,
New Haven, CT, USA
e-mail: julia.borrassol@yale.edu

A. M. Dollar
e-mail: aaron.dollar@yale.edu

F. Thomas and A. Pèrez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_25,

This paper reviews the mathematical frameworks involved for modeling the hand-plus-object system of a hand with three fingers and two phalanges per finger, and its kinematically equivalent parallel manipulator. As expected, we show how the derived static equations match. We believe that our comparison helps to clarify the role of the grasp matrix versus the role of the parallel manipulator Jacobian matrix. As far as the authors know, there has not been any publication proving that both frameworks are analytically equivalent. The results obtained in this paper are for a particular example. A general complete proof of such equivalence is left as future work.

Section 2 introduces the studied example and its notation. Section 2.1 details the steps to obtain the matrices for hands and Sect. 2.2 for parallel robots. The obtained matrices are compared in Sect. 2.3. Finally, Sect. 3 discusses advantages of the proven equivalence, and proposes future work based on transfer of knowledge between both fields.

2 The 3-UR Hand and Its Equivalent Parallel Manipulator

This paper analyzes the three-fingered hand depicted in Fig. 1. Its architecture is similar to other robotic hands such as the Barrett hand [16] or the JPL hand [13]. The hand consists of three equal fingers with two phalanges each and three rotational joints each (2 in finger flexion, and one base rotation). For each finger i, $\mathbf{z}_{i1} = (0, 0, 1)^T$ and $\mathbf{z}_{i2} = \mathbf{z}_{i3} = (\sin(\theta_{i1}), -\cos(\theta_{i1}), 0)^T$ are the axis of rotation of the first, second and third joints, respectively, with rotation angles θ_{i1}, θ_{i2} and θ_{i3}, respectively (see Fig. 1).

To complete the hand-plus-object system, we need to define the contact model. The two most common contact models are called hard and soft fingers. The first one assumes a point contact with friction with a small contact patch. Kinematically, it is equivalent to a spherical joint. The second model assumes a larger contact patch and thus, the finger can also transmit a moment about the contact normal. This is equivalent to a universal joint. Therefore, the system hand-plus-object using the hard-finger (soft-finger) model is kinematically equivalent to a 3-URS (3-URU) parallel manipulator (where U stands for universal joint, R revolute, and S spherical). In this work, we use the hard-finger model. Then, the mobility of the manipulator, computed using the Grübler-Kutzbach criterion, is 6, that means the object (platform) can be moved in 6 degrees of freedom (DoF). Other more complex models, such as the rolling contact, are left as future work [15].

Hands need to actuate all the joints to keep the fingers rigid when they work without contact, and thus, the resulting manipulator will have the 9 finger joints

$$\Theta = (\theta_{11}, \theta_{12}, \theta_{13}, \theta_{21}, \theta_{22}, \theta_{23}, \theta_{31}, \theta_{32}, \theta_{33}) \tag{1}$$

actuated. The rest of the joints are left free to move (passive). They are defined considering the spherical joints as the intersection of three revolute joints. We define

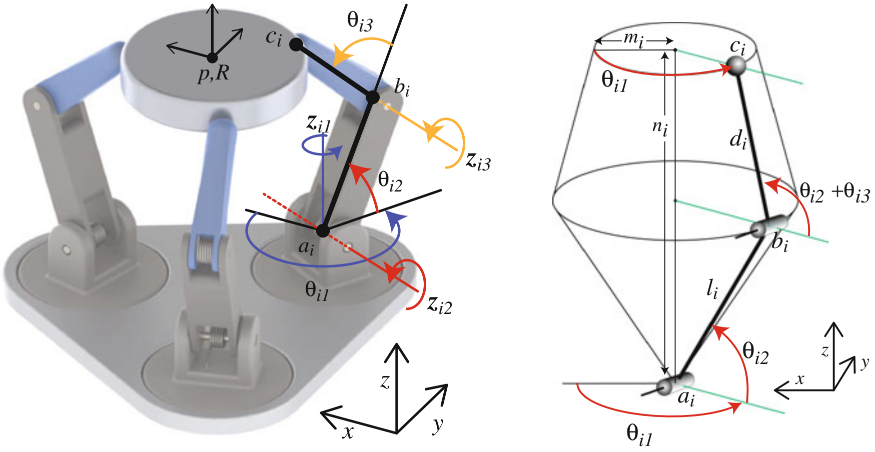


Fig. 1 A three-fingered hand with its corresponding notation. The center points of the palm joints are equally distributed around a circumference of radius r_p, and the contact points on the object around a circumference of radius r_o. By geometric construction, the coordinates of the fingertip can be described using the magnitudes $n_i = l_i \sin(\theta_{i2}) + d_i \sin(\theta_{i2} + \theta_{i3})$ and $m_i = l_i \cos(\theta_{i2}) + d_i \cos(\theta_{i2} + \theta_{i3})$, where l_i and d_i are the lengths of the proximal and distal links of the ith finger, respectively

their axis of rotation as $\mathbf{z}_{i4} = (1, 0, 0)$, $\mathbf{z}_{i5} = (0, 1, 0)$ and $\mathbf{z}_{i6} = (0, 0, 1)$, with angles θ_{i4}, θ_{i5} and θ_{i6}, respectively. Then, we can state that the manipulator has three degrees of *actuation redundancy* (9 actuated joints versus 6 DoF of mobility). As the output twist that defines the velocity and angular velocity of the object (platform) is also 6 dimensional, we can say that the manipulator does not have *kinematic redundancy* [19].

The position and orientation of the object (platform) with respect to the palm (base) reference frame are given by a position vector $\mathbf{p} \in \mathbb{R}^3$ located at the center of mass of the platform (object) and a rotation matrix $\mathbf{R} \in SO(3)$. If $\widetilde{\mathbf{a}}_i$ and $\widetilde{\mathbf{c}}_i$ are the local coordinates of the palm (base) and object (platform) attachments in their local reference frames, their coordinates with respect to the palm (base) fixed reference frame are $\mathbf{a}_i = \widetilde{\mathbf{a}}_i$ and $\mathbf{c}_i = \mathbf{p} + \mathbf{R}\widetilde{\mathbf{c}}_i$. Assuming contact, the coordinates of the contact points must be the same as the coordinates of the fingertips, which can be obtained by geometric construction as

$$\mathbf{c}_i = \mathbf{a}_i + n_i (0, 0, 1)^T + m_i (\cos(\theta_{i1}), \ \sin(\theta_1), \ 0)^T,$$

(where n_i and m_i are defined in Fig. 1-(right)). The loop equations are obtained equating the two obtained coordinates of the contact points $\mathbf{c}_i$. Solving them for Θ or for $\{\mathbf{p}, \mathbf{R}\}$ gives the the inverse and forward kinematic solutions, respectively.

The next two sections describe how to obtain the velocity equations using the grasping framework [12] and the parallel manipulators frameworks [9, 17].

The equations are listed in Table 1, for the described hand (first column of the table) and the equivalent parallel manipulator (second column of the table).

The velocity of the object (platform) is described using screw theory in both frameworks. We define a screw as $\$ = (\mathbf{u}, \mathbf{q} \times \mathbf{u})$ for a given vector $\mathbf{u}$ and a position vector $\mathbf{q}$. Two screws are reciprocal when its reciprocal product is zero, *i.e.*,

$$(\mathbf{u}_1, \mathbf{q}_1 \times \mathbf{u}_1) \circ (\mathbf{u}_2, \mathbf{q}_2 \times \mathbf{u}_2) = (\mathbf{q}_1 \times \mathbf{u}_1, \mathbf{u}_1) \cdot (\mathbf{u}_2, \mathbf{q}_2 \times \mathbf{u}_2) = 0,$$

where $\cdot$ stands for the usual dot product and $\circ$ the reciprocal product [3, 18]. The twist $\mathbf{T} = (\mathbf{v}, \Omega)$ defines the linear and angular velocity of the object (platform).

2.1 The Grasp Matrix and the Hand Jacobian

The total grasp and hand Jacobian matrices are defined stacking together the matrices of each finger as shown in Table 1-row f. To define each finger matrix, first we need to define a set of reference frames, $\{C_i\} = \{\mathbf{c}_i, \mathbf{R}_i\}$, located at each of the contact points and with rotation matrix $\mathbf{R}_i = (\mathbf{n}_i \mathbf{t}_i \mathbf{o}_i)$, with $\mathbf{n}_i$ normal to the plane tangent to the object at the contact point, and directed toward the object. The remaining two vectors are chosen orthonormal to the first one (Table 1-row a). For our case, we define these vectors as

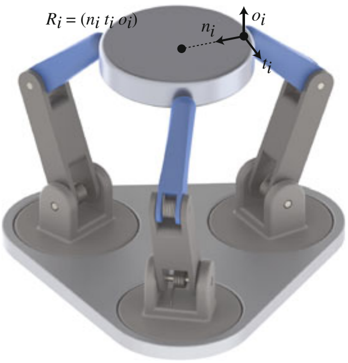
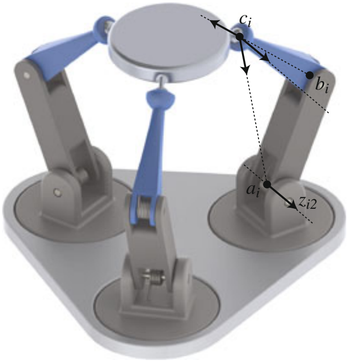
$$\begin{aligned}
\mathbf{n}_i =& (n_{ix}, n_{iy}, n_{iz}) = \frac{\mathbf{p} - \mathbf{c}_i}{r_o}, \\
\mathbf{t}_i =& \left(\frac{n_{iy}}{\sqrt{n_{ix}^2 + n_{iy}^2}}, -\frac{n_{ix}}{\sqrt{n_{ix}^2 + n_{iy}^2}}, 0 \right), \\
\mathbf{o}_i =& n_i \times t_i
\end{aligned} \tag{2}$$

The grasp matrix for the finger i is a change of coordinates of the twist of the object $\mathbf{T}$, from the fixed reference frame to $\{C_i\}$. Let $\mathbf{T}_{fi}$ be the twist at the fingertip i with respect to the reference $\{C_i\}$. Then, $\mathbf{T}_{fi} = \mathbf{G}_i^T \mathbf{T}$ where $\mathbf{G}_i^T = \mathbf{H}_i \overline{\mathbf{R}}_i \mathbf{P}_i$ (see explicit expression in Table 1-row d). The matrix $\mathbf{P}_i$ translates the twist from $\mathbf{p}$ to $\mathbf{c}_i$. The matrix $\overline{\mathbf{R}}_i$ rotates the twist to match $\{C_i\}$ and $\mathbf{H}_i$ is the contact model matrix, that sets to zero the three coordinates corresponding to the angular velocity (see [12] for detailed definition of this matrix).

The hand Jacobian matrix $\mathbf{J}_H$ is defined by the joint twists, whose expressions for each finger i are

$$\begin{aligned}
\$_{i1} =& ((\mathbf{a}_i - \mathbf{c}_i) \times \mathbf{z}_{i1}, \mathbf{z}_{i1})^T \\
\$_{i2} =& ((\mathbf{a}_i - \mathbf{c}_i) \times \mathbf{z}_{i2}, \mathbf{z}_{i2})^T \\
\$_{i3} =& ((\mathbf{b}_i - \mathbf{c}_i) \times \mathbf{z}_{i3}, \mathbf{z}_{i3})^T .
\end{aligned} \tag{3}$$

Table 1 Summary of static and velocity equations. τ and $\dot{\Theta}$ are the vector of joint torques and velocities, respectively. $\mathbf{W}$ and $\mathbf{T}$ are the external wrench and twist acting on the object (platform). λ_f is a 1×9 vector containing the three fingertip forces, and $\mathcal{F}$ represents the friction cone

	Grasping	Parallel manipulators
a		
b	$\mathbf{G}^T\mathbf{T} = \mathbf{J}_H\dot{\Theta}$ $\mathbf{J}_H^T\lambda_f = \tau$	$\mathbf{J}_\mathbf{p}\mathbf{T} = \mathbf{J}_\Theta\dot{\Theta}$
c	$-\mathbf{G}\lambda_f = \mathbf{W}$ $\lambda_f \in \mathcal{F}$	$\mathbf{W} = -\mathbf{J}_\mathbf{p}^T\mathbf{J}_\Theta^{-T}\tau$
d	$\mathbf{G}_i^T = \mathbf{H}_i\begin{pmatrix}\mathbf{R}_i & 0\\ 0 & \mathbf{R}_i\end{pmatrix}\begin{pmatrix}\mathbf{I}_3 & 0\\ (\mathbf{c}_i - \mathbf{p})^\times & \mathbf{I}_3\end{pmatrix}$ where $\mathbf{v}^\times$ is the cross-product matrix	$\mathbf{J}_{pi} = \begin{pmatrix}\mathbf{z}_{i2}^T & (\mathbf{c}_i \times \mathbf{z}_{i2})^T\\ (\mathbf{c}_i - \mathbf{b}_i)^T & (\mathbf{c}_i \times (\mathbf{c}_i - \mathbf{b}_i))^T\\ (\mathbf{c}_i - \mathbf{a}_i)^T & (\mathbf{c}_i \times (\mathbf{c}_i - \mathbf{a}_i))^T\end{pmatrix}$
e	$\mathbf{J}_{Hi} = \mathbf{H}_i\overline{\mathbf{R}}_i(\$_{i1}\$_{i2}\$_{i3})$ with $\$_{ij}$ defined in (3)	$\mathbf{J}_{\Theta i} = \begin{pmatrix}-m_i & 0 & 0\\ 0 & l_i d_i Sin(\theta_{i3}) & 0\\ 0 & 0 & -l_i d_i Sin(\theta_{i3})\end{pmatrix}$ with m_i defined in Fig. 1
f	$\mathbf{G}^T = \begin{pmatrix}\mathbf{G}_1^T\\ \mathbf{G}_2^T\\ \mathbf{G}_3^T\end{pmatrix}, \mathbf{J}_H = \begin{pmatrix}\mathbf{J}_{H1} & 0 & 0\\ 0 & \mathbf{J}_{H2} & 0\\ 0 & 0 & \mathbf{J}_{H3}\end{pmatrix}$	$\mathbf{J}_\mathbf{p} = \begin{pmatrix}\mathbf{J}_{p1}\\ \mathbf{J}_{p2}\\ \mathbf{J}_{p3}\end{pmatrix}, \mathbf{J}_\Theta = \begin{pmatrix}\mathbf{J}_{\Theta 1} & 0 & 0\\ 0 & \mathbf{J}_{\Theta 2} & 0\\ 0 & 0 & \mathbf{J}_{\Theta 3}\end{pmatrix}$

Note that the angular components are computed about the center of the reference $\{C_i\}$. Then, the ith fingertip twist is expressed as $\mathbf{T}_{fi} = \mathbf{J}_{Hi}\dot{\Theta}$, where $\mathbf{J}_{Hi}$ is detailed in Table 1-row e. As before, the matrix $\overline{\mathbf{R}}_i$ is used to write the twist with respect to $\{C_i\}$ and $\mathbf{H}_i$ to select only the transmitted components.

Finally, rows b and c show the velocity and the static equilibrium equations using the complete matrices.

2.2 The Jacobian Matrix of the Parallel Manipulator

Here we follow the steps proposed in [9] or Chapter 5.6 in [17] to obtain the Jacobian matrix for the parallel manipulator shown in Table 1, row a, second column.

Let $\mathbf{T}$ be the twist of the platform, as in the previous section. The theorem in [9] states that it can be written as the sum of the joint twists of each leg, that is,

$$\mathbf{T} = \sum_{j=1}^{6} \dot{\theta}_{ij} \$_{ij}, \text{ for } i = 1, 2, 3. \tag{4}$$

Here, the joint twists angular components are computed about the center of the base fixed reference frame, namely, $\$_{ij} = (\mathbf{r} \times \mathbf{z}_{ij}, \mathbf{z}_{ij})$, for $j = 1, \ldots, 6$, where $\mathbf{r}$ takes the value of the corresponding joint center. The first three joint twists are equivalent to the twists defined in (3). The remaining three correspond to the passive joints.

The passive joint twists $\$_{ij}$, for $j = 4, 5, 6$, can be eliminated from the system (4) computing their reciprocal screws, named as ${}^r\$_{ik}$, for $k = 1, 2, 3$. It is important to note that any set of three linearly independent screws through the contact point are reciprocal to the spherical joint system $\{\$_{i4}, \$_{i5}, \$_{i6}\}$ [3, 18]. After multiplying the reciprocal system at both sides of each equation in (4), we can rewrite the system as $\mathbf{J_p T} = \mathbf{J}_\Theta \dot{\Theta}$, where the rows of the matrix $\mathbf{J_p}$ are the reciprocal screws and the matrix $\mathbf{J}_\Theta$ only depends on the active joint angles, $\mathbf{J}_\Theta = ({}^r\$_{ik} \circ \$_{ij})$, for $j = 1, \ldots, 3$ and $k = 1, \ldots, 3$. That is, it is formed by all the products of the reciprocal screws with the actuated joint screws.

The most convenient choice of the reciprocal screws is to define each one to be reciprocal to all the passive joint twists plus two of the active. This leads to a diagonal matrix $\mathbf{J}_{\Theta i}$ (Table 1-row e). The explicit expressions of the reciprocal screws for each leg i are the rows of the matrix $\mathbf{J}_{pi}$ in Table 1-row d.

We can obtain the ith fingertip wrench, written with respect to the fixed reference frame, by multiplying each set of three columns in $\mathbf{J_p}^T \mathbf{J}_\Theta^{-T}$ by the corresponding three joint torques τ. When the matrix $\mathbf{J}_\Theta$ is not square, we can use the pseudo-inverse.

2.3 Comparison of Frameworks

We computed all the equations using Wolfram Mathematica 9. We can see that the matrices in the rows d, e and f between the two columns of Table 1 are obviously different. However, the analytical expression of the products $\mathbf{J}_\Theta^{-1}\mathbf{J_p}$ and $\mathbf{J}_H^{-1}\mathbf{G}^T$ are the same, except for the angular components. In the grasping framework, the angular velocities (moments) components of the twists (wrenches) are computed with respect to the center of the object (platform), while in the parallel manipulators framework they are computed with respect to the fixed reference frame center. Thus, we can say that they are equivalent $\mathbf{J}_\Theta^{-1}\mathbf{J_p} \equiv \mathbf{J}_H^{-1}\mathbf{G}^T$.

In Sect. 2.2 we state that the reciprocal screws can be chosen arbitrarily, provided that they are independent and through the contact point $\mathbf{c}_i$. Then, let us define them using the vectors of the fingertip frame (see Eq. (2) and figure in Table 1, row a, column 1). In other words, we use $\mathbf{n}_i$, $\mathbf{t}_i$ and $\mathbf{o}_i$ to define the screws in the matrix $\mathbf{J}_{pi}$. Then, all the matrices in both frameworks coincide, that is, $\mathbf{J}_H = \mathbf{J}_\Theta$ and $\mathbf{J}_\mathbf{p} \equiv \mathbf{G}^T$, where the second equivalence is not analytically identical only because the moments and angular velocities are computed with respect to different centers.

Note that the particular choice of the reciprocal screws will shape the final form of the matrices in Table 1. Analogously, in the grasping framework, this choice is made when defining the vectors of the rotation matrix of the reference frames $\{C_i\}$.

In the grasping context, the choice of the vectors $\{\mathbf{n}_i, \mathbf{t}_i, \mathbf{o}_i\}$ is convenient to obtain the expression of the fingertip forces λ_f directly projected to the axes of the friction cones. This facilitates the evaluation of the friction cone conditions. In the parallel manipulator context, the choice is done so that the resulting matrix $\mathbf{J}_\Theta$ is as diagonal as possible. This allows the interpretation of the rows of the complete Jacobian $\mathbf{J}_\Theta^{-1}\mathbf{J}_\mathbf{p}$ in terms of line Plücker coordinates [7]. This is useful to find geometrical interpretation of singularities. Recently, in [2] they have used this technique to hand fingers, and the reciprocal system is chosen to facilitate the single value decomposition of the resulting finger Jacobian matrix.

We can also observe that the steps shown in Sects. 2.1 and 2.2 can be generalized to any type of hand (manipulator), but the resulting matrices will be tall, wide or square depending on the relationship between the mobility, the number of actuated and passive joints and the dimension of the output twist [19]. It remains to proof that the results are always equivalent.

3 Discussion and Future Work

The grasping literature commonly uses the manipulability index to state the quality of the grasp, and it is either based only on the hand Jacobian [13] or on the multiplication of both matrices $\mathbf{J}_H^{-1}\mathbf{G}^T$ [14]. While this can detect singularities, the literature of parallel robots has extensively studied and classified them in much more detail [5, 19, 20].

Among parallel robot designers, it is well known that a smart design has to take into account the singularities inside the workspace [1, 8]. As far as the authors know, this is not done when designing hands. In part, this may be because the actuation redundancy reduces the dimensionality of the singularity locus. However, simplified hands that use underactuated fingers can reduce the degree of actuation redundancy down to 0 or even lower. In particular, we are studying how underactuation with pulling cables can be modeled with similar Jacobian matrices where these kind of singularities need to be taken into account. This type of hands are becoming very popular not only for effective grasps, but also to perform dexterous manipulation [11]. For these hands, singularities may be an issue that researchers will have to take into account in the process of hand design.

We believe that the study of convenient choices of the reciprocal system can lead to useful tools to design hands with increased workspaces. For instance, it can be useful to compute an analytical expression of the hyper-surface of singularities using only task space variables. Analyzing such surface can help to plot independent components inside a workspace, that cannot be crossed without loosing control.

This work has shown how the grasp matrix plays the same role as the Jacobian of reciprocal screws for the analyzed example. Such equivalence allows for transfer of knowledge from parallel manipulators to robotic hands. Extending this work to more general cases is part of a future work that will help to fully understand the parallelisms between these two types of manipulators.

Acknowledgments This work was supported in part by National Science Foundation grant IIS-0952856.

References

1. Bohigas, O., Manubens, M., Ros, L.: A complete method for workspace boundary determination on general structure manipulators. IEEE Trans. Robot. **28**(5), 993–1006 (2012)
2. Cui, L., Dai, J.S.: Reciprocity-based singular value decomposition for inverse kinematic analysis of the metamorphic multifingered hand. J. Mech. Robot. **4**(3), 034,502 (2012)
3. Dai, J.S., Jones, J.: Interrelationship between screw systems and corresponding reciprocal systems and applications. Mech. Mach. Theory **36**, 743–761 (2001)
4. Ebert-Uphoff, I., Voglewede, P.A.: On the connections between cable-driven robots, parallel manipulators and grasping. In: IEEE International Conference on Robotics and Automation, pp. 4521–4526 (2004)
5. Gosselin, C., Angeles, J.: Singularity analysis of closed-loop kinematic chains. IEEE Trans. Robot. Autom. **6**(3), 281–290 (1990)
6. Kerr, J., Roth, B.: Analysis of multifingered hands. Int. J. Robot. Res. **4**(3), 3–17 (1986)
7. Merlet, J.-P.: Singular configurations of parallel manipulators and Grassmann geometry. Int. J. Robot. Res **8**(5), 45–56 (1989)
8. Merlet, J.-P.: Parallel robots, second edition edn. Springer, New York (2006)
9. Mohamed, M., Duffy, J.: A direct determination of the instantaneous kinematics of fully parallel robot manipulators. J. Mech. Trans. Autom. Des. **107**, 226–229 (1985)
10. Montana, D.J.: The kinematics of multi-fingered manipulation. IEEE Trans. Robot. Autom. **11**(4), 491–503 (1995)
11. Odhner, L.U., Dollar, A.M.: Dexterous manipulation with underactuated elastic hands. In: IEEE International Conference on Robotics and Automation, pp. 5254–5260 (2011)
12. Prattichizzo, D., Trinkle, J.C.: Grasping. In: Siciliano, B., Khatib, O., (eds.) Springer handbook of robotics, chap. 28, pp. 671–700. Springer, New york (2008)
13. Salisbury, J.K., Craig, J.J.: Articulated hands: force control and kinematic issues. Int. J. Robot. Res. **1**(1), 4–17 (1982)
14. Shimoga, K.B.: Robot grasp synthesis algorithms: a survey. Int. J. Robot. Res. **15**(3), 230–266 (1996)
15. Staffetti, E., Thomas, F.: Analysis of rigid body interactions for compliant motion tasks using the Grassmann-Cayley algebra. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2325–2332 (2000)
16. Townsend, W.: The BarrettHand grasper—programmably flexible part handling and assembly. Indus. Robot. Int. J. **27**, 181–188 (2000)

17. Tsai, L.W.: Robot analysis. The mechanics of serial and parallel manipulators. Wiley, New York (1999)
18. Zhao, J., Li, B., Yang, X., Yu, H.: Geometrical method to determine the reciprocal screws and applications to parallel manipulators. Robotica **27**(06), 929 (2009)
19. Zlatanov, D.: Generalized singularity analysis of mechanisms. Ph.D. thesis (1998)
20. Zlatanov, D., Bonev, I., Gosselin, C.M.: Constraint singularities of parallel mechanisms. In: International Conference on Robotics and Automation, pp. 496–502 (2002)

A Novel Mechanism with Redundant Elastic Constraints for an Actual Revolute Joint

Delun Wang, Zhi Wang, Huimin Dong and Shudong Yu

Abstract A novel spatial mechanism with redundant elastic constraints is presented in this paper to establish a comprehensive model for simulating kinematic characteristics of an actual revolute joint with flexibility and geometric errors. The rigid cam profiles are specified to represent the geometrical errors. Elastic springs are used to simulate the deformations of joint components and their surfaces for the actual machine parts. The proposed RE mechanism, consisting of suspended cams and multiple followers with springs, yields a total of 32 basic equations for displacement analysis. The numerical results obtained using the proposed approach were compared with the experimental data for an example revolute joint; good agreement was achieved for joint kinematic characteristics. The proposed approach provides a new application of the theory of mechanism in comprehensive performance analysis of a complex mechanical system having many components with machining errors.

Keywords Elastic constraint · Error · Mechanism · Revolute joint

1 Introduction

In a kinematical analysis of mechanism, the links are often assumed to be rigid, the joints are assumed to have ideal geometries and maintain rigid under loads. Take a revolute joint as an example. The shaft in the revolute joint is a rigid and perfect cylinder, which can rotate only about its ideal axis of rotation regardless

D. Wang (✉) · Z. Wang · H. Dong
School of Mechanical Engineering, Dalian University of Technology, Dalian, China
e-mail: dlunwang@dlut.edu.cn

S. Yu
Department of Mechanical and Industrial Engineering, Ryerson University,
Toronto, ON, Canada

S. Yu
Dalian University of Technology, Dalian, China
e-mail: shudong.yu@gmail.com

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_26,

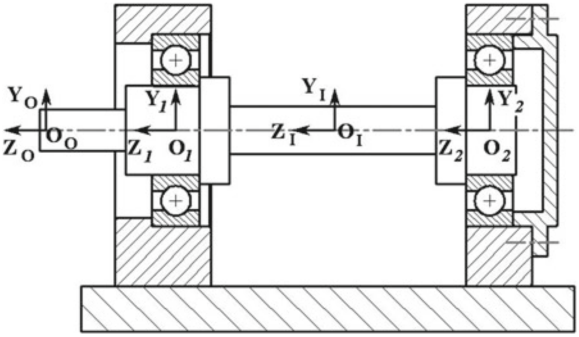


Fig. 1 A typical construction of actual revolute joint

of the applied loads. In reality, all components deform under loads, the shapes of these components have geometrical errors due to machining. Therefore, the shaft in an actual revolute joint has one dominating rotational DOF and five additional DOF's due to joint machining errors and component flexibility. We believe that the kinematic characteristics of a joint in machine design can be obtained by studying the kinematics of a corresponding spatial mechanism with appropriate constraints when considering machining errors and joint flexibility [1, 2].

In this paper, our scope is focused on the analysis of the complex movement of the shaft in a revolute joint with an aim to addressing the effects of machining errors and flexibility (or simply revolute precision) on performance of an actual revolute joint. The revolute precision is a key noteworthy performance of machine tools and various precision instruments. The kinematical analysis and synthesis of an actual revolute joint lays a solid theoretical basis for machine precision design [3–5].

2 An Actual Revolute Joint with Elastic Constraint and Error

An actual revolute joint consists of a rotating shaft, two or more bearings, and a housing unit. The shaft can rotate freely about its axis, as shown in Fig. 1.

Geometrical errors: Every component is manufactured with geometrical errors. Because of the presence of geometrical errors, the shaft no longer has a perfectly cylindrical surface. Similarly, the actual bearing components such as the outer and inner races also deviate from their ideal nominal surfaces. These components with various types of geometrical errors are assembled together to form an actual revolute joint to accommodate the rotational motion of the shaft.

Elasticity and deformations: Every component in an actual revolute joint is made of isotropic or anisotropic materials with finite elasticity. Under loads, these components deform. The load-induced component deformations magnify the

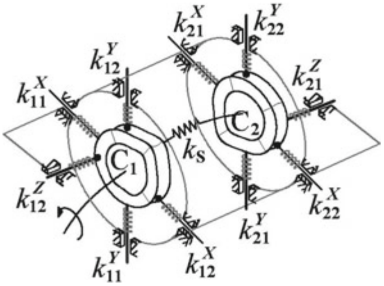


Fig. 2 The RE mechanism with redundant elastic constraints for an actual revolute joint

machining-induced geometrical errors. The actual geometrical shapes of the shaft and bearings also vary with loads.

Constraints and motion: The shaft in an actual revolute joint, supported and constrained by bearings, rotates about the Z-axis freely. The additional five elastic DOF's, two rotations about the X-axis and the Y-axis, and three translations along the X, Y, and Z directions, are always present when geometrical errors and component flexibility are taken into consideration. As a result, a geometrical element of the shaft in an actual revolute joint has six DOF's.

3 The RE Mechanism for an Actual Revolute Joint

Geometry and motion equivalence: Each component has its own geometrical errors, which affects the positions and kinematical characteristics of the shaft. In the motionless conjunction planes (MLCP), the components are fixed together. The components deform with assembled loads, their shapes and positions will also change. While in the motional conjunction surfaces (MCS), the components can only move relatively. The kinematic characteristics (positions and loci) of the motional components will change due to the geometrical errors. This means that the geometrical errors of one component may be transferred into those of another component by the MLCP or the MCS. The shaft in an actual revolute joint is regarded as cams with specified rigid profiles and suspended by the translating followers as shown in Fig. 2. The shaft motion is equivalent to that of the cam having a profile equivalent to the geometrical dimensions and their error characteristics.

Elasticity equivalence: The surfaces of components for an actual revolute joint experience deform under loads. In general, the displacements of the components due to elastic deformations may be transferred to the surface of the shaft by the MLCP or the MCS. As a result, the shaft will have additional elastic displacements, which are determined by the component flexibility and loads. The elasticity of the constraint is equivalent to several springs attached to the followers of the mechanism [6].

RE mechanism: A mechanism, with redundant degree of freedom and elastic constraints, can be defined as an RE mechanism, where R stands for the redundant degree of freedom and E denotes elastic constraints. In an RE mechanism, both rigid links and elastic components (springs) are simultaneously present; the geometrical characteristics and physical characteristics can be extensively described. Particularly, the redundant DOF's coexist with elastic constraints.

In Fig. 2, the RE mechanism has a total of 12 links, a shaft with double cams, 10 translational followers and a base frame. Each follower forms a 1-DOF prismatic pair with the base frame, and a 5-DOF pair with the cam. The number of degrees of freedom can be calculated by the Gruebler's equation [7]

$$F = 6(n - g - 1) + \sum_{i=1}^{g} f_i \tag{1}$$

where $n(= 12)$ is the number of links; $g(= 20)$ is the total number of kinematic pairs; $f_1(= 1 \times 10)$; f_2 $(= 5 \times 10)$. The number of DOF's is $F = 6$. These 6 DOF's are the rotation of the cam about its axis and the five elastic DOF's mentioned above. In addition, the RE mechanism has another 10 DOF's associated with the 10 translational followers, in which five of them correspond to the elastic constraint of the cam, and the other five are the elastic motion of the cam with redundant DOF's. It is enough for a rigid cam mechanism to be constrained with 5-DOF motion. We can see the significance of the elastic motions with redundant DOF's.

The RE mechanism in Fig. 2 is a simplest mechanism with redundant DOF's and elastic constraints for the actual revolute joint with the two bearings. It will keep the same DOF's if an additional follower is introduced since it is accompanied by a 1-DOF pair and a 5-DOF pair, which counterbalances the six DOF's of the introduced follower. This performance provides convenience for displacement analysis of the actual revolute joint with construction error and over-constraint constructions.

Discussion: In mechanism, the constraints and DOF's are closely related. The DOF is the number of independent parameters required to define configurations of the mechanism, with respect to the base frame. For an actual revolute joint, a point of the base link (component) experiences elastic displacements under loads. Contrary to the rigid mechanisms, the constraining link or the follower in an RE mechanism has the elastic motion of springs. Therefore, the DOF in an RE mechanism has a different meaning from that in mechanism with rigid links only.

4 The Basic Equations of an RE Mechanism

For displacement analysis, the basic equations of an RE mechanism have to be determined, including the geometrical equations, the equations of equilibrium, and the equations of physical properties. The geometrical equations can be given as following according to the construction of the RE mechanism, shown in Fig. 3.

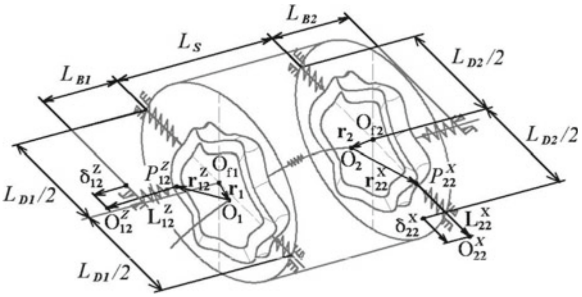


Fig. 3 The displacements of the RE mechanism

$$|\mathbf{r_i} + \mathbf{r^V_{ij}} + (\mathbf{L^V_{ij}} - \boldsymbol{\delta}^{\mathbf{V}}_{\mathbf{ij}})| = L_{Di}/2 \quad i, j = 1, 2;\ V = X, Y \tag{2}$$

$$|\mathbf{r_i} + \mathbf{r^Z_{ij}} + (\mathbf{L^Z_{ij}} - \boldsymbol{\delta}^{\mathbf{Z}}_{\mathbf{ij}})| = L_{Bi} \quad i = 1, j = 2 \ \ or \ \ i = 2, j = 1 \tag{3}$$

The equations of equilibrium for an RE mechanism may firstly be written as, according to the notation in Fig. 4,

$$\sum_{i=1}^{2}\sum_{j=1}^{2}(\mathbf{F^X_{ij}} + \mathbf{F^Y_{ij}} + \mathbf{F^Z_{ij}}) + \mathbf{F_I} + \mathbf{F_O} = \mathbf{0} \tag{4}$$

$$\sum_{i=1}^{2}\sum_{j=1}^{2}(\mathbf{F^X_{ij}P^{Xf}_{ij}} + \mathbf{F^Y_{ij}P^{Yf}_{ij}} + \mathbf{F^Z_{ij}P^{Zf}_{ij}}) + \mathbf{F_I r_I} + \mathbf{F_O r_O} + \mathbf{M_I} + \mathbf{M_O} = \mathbf{0} \tag{5}$$

The equations of equilibrium for the followers are $|\mathbf{F^{VK}_{ij}}| = |\mathbf{F^V_{ij}}| \cos\alpha^V_{ij}$.

The equations of physical properties in an RE mechanism can be written from the elasticity of the follower springs as equation (6). And the physical equations of shaft connected to two cams as equation (7)

$$\mathbf{F^{VK}_{ij}} = k^V_{ij}\boldsymbol{\delta}^{\mathbf{V}}_{\mathbf{ij}} \quad i, j = 1, 2;\ V = X, Y, Z \tag{6}$$

$$\mathbf{K_S}[\Delta^X_{12}, \Delta^Y_{12}, \Delta^Z_{12}, \theta^X_{12}, \theta^Y_{12}, \theta^Z_{12}]^T = \mathbf{P} \tag{7}$$

There are six parameters to be determined in Equation (7), which describes the displacements of the shaft ($\Delta^X_{12}, \Delta^Y_{12}, \Delta^Z_{12}, \theta^X_{12}, \theta^Y_{12}, \theta^Z_{12}$), or the relative differences between the two cams.

Equations (2) to (7), totaling 32 equations with 32 variables, are called the basic equations of an RE mechanism. From the basic equations, the positions and postures of two cams with rotation angle θ_z can now be located at any instant. As a result, any point of the shaft will produce a spatial curve in the based frame, which is the actual non-circular trajectory. The precision of an revolute joint, just the error between the actual trajectory and the ideal curve, can be evaluated. The variations of precision with the working conditions of an actual revolute joint can also be exposed.

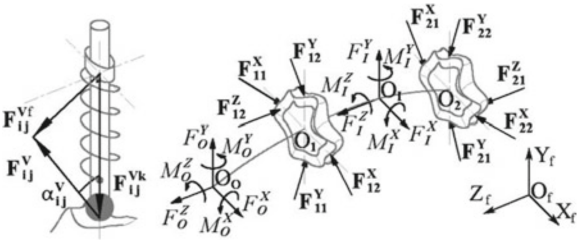


Fig. 4 The forces and moments in the RE mechanism

Table 1 The construction parameters of the actual revolute joint

Bore (d)	60.0 mm	Ball diameter (D_b)	22.22 mm
Outside diameter (D)	130.0 mm	Distance between O_1 and O_O	130.0 mm
Inner ring width (B)	31.0 mm	Distance between O_1 and O_I	105.0 mm
Outer ring width (C)	31.0 mm	Distance between O_1 and O_2	160.0 mm

5 Case Study

An RE mechanism can be constructed for the actual revolute joint shown in Fig. 1. The basic parameters of the revolute joint are given in Table 1.

Parameters of elastic constraints: In this mechanism, there are 10 follower springs. The deformations of the springs are intended to be equivalent to the elastic displacement of the shaft in the revolute joint, which are caused by the combined elastic deformation of the bearings, bearing blocks, flanges and the shaft. For convenience, we calculate the stiffness of the bearing and the supporting structure individually as the rolling bearing is a standard part.

The stiffness of bearings can be calculated by the approximate formula [8]. The stiffness in the radial direction is $k_{ij}^{BV} = 1190282\delta_r^{1/2}$ $(i, j = 1, 2;\ \mathrm{V} = \mathrm{X}, \mathrm{Y})$, and in the axial direction, the stiffness is $k_{ij}^{BV} = 46568\delta_r^{1/2}$ $(i, j = 1, 2;\ \mathrm{V} = \mathrm{Z})$.

The stiffness of the supporting structures, including the bearing block, flange and shaft, is calculated by using the finite element method (FEM) in this paper. The results are given in Table 2. The stiffness of the shaft connected to the two cams is calculated by FEM according to equation (7).

Table 2 Stiffness of the supporting structures ($\times 10^5$N/mm)

k_{11}^{SX}	k_{12}^{SX}	k_{11}^{SY}	k_{12}^{SY}	k_{11}^{SZ}	k_{12}^{SZ}
8.50	8.50	6.06	10.1	unconstrained	2.33
k_{21}^{SX}	k_{22}^{SX}	k_{21}^{SY}	k_{22}^{SY}	k_{21}^{SZ}	k_{22}^{SZ}
5.64	5.64	4.45	7.27	1.19	unconstrained

Geometrical feature and cam profiles: The geometric features and the kinematic characteristics of the cams are similar to those of the bearing inner rings. Ideally, the cam profile is the ideal profile of the inner race ($r_0(\theta, z)$). In fact, the geometric features and kinematic characteristics of the bearing inner rings are related to the geometric errors and loads.

Each cam follower of the RE mechanism corresponds to a specified cam profile. With the MLCP and the MCS, the geometrical errors are transferred into positional errors ($\delta(\theta, z)$) of the moving component. The ideal profile of the inner race ($r_0(\theta, z)$) is designated as the initial profile of the cams, and then, the positional errors ($\delta(\theta, z)$) and the elastic coefficient ($\varepsilon(\theta, z)$) are taken into account. The profiles equation of the cams can be written as

$$r = S(\theta, z) = r_0(\theta, z) + \delta(\theta, z) + \varepsilon(\theta, z) \tag{8}$$

The actual profiles of the bearing components are measured respectively. Based on the elastic deformation, the geometrical errors may change the geometrical shape of the outer and inner races by the interference fitting between the shaft and inner ring, which can be calculated by using a contact model and converted into the positional errors ($\delta(\theta, z)$)of the kinematic model in the RE mechanism.

Experiment: To compare the results of displacement analysis of the RE mechanism with that of experiments, the test equipment is set up for the actual revolute joint, shown in Fig. 5a. The displacements of the reference point of the shaft, which is located at the cross-section of the outer section, are measured in the X-direction.

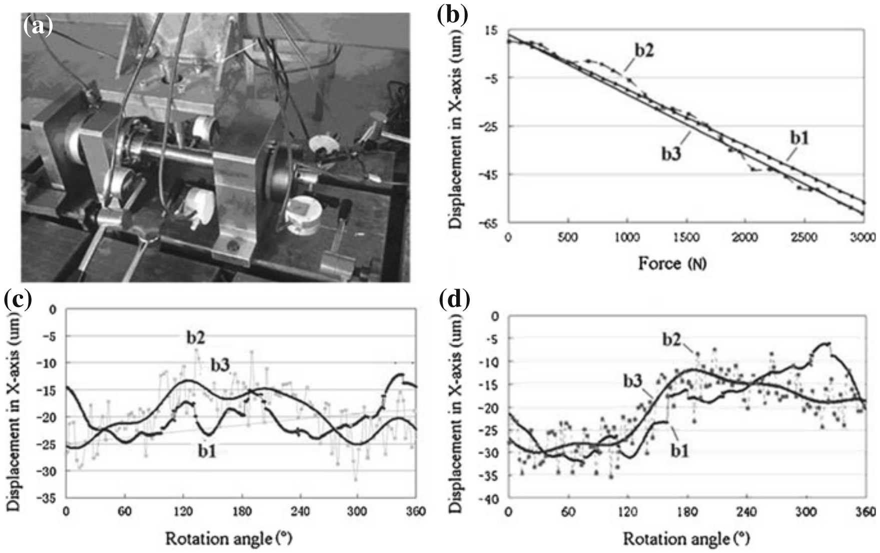


Fig. 5 Experiment of an actual revolute joint

The curve b1 is the numerical results computed by the model of RE mechanism, the curve b2 is the measuring results of the experiment and the curve b3 is the fitting curve of the curve b2, which are shown as followings.

(1) Fig. 5b shows the variations of the displacements with the driving force from 0 to 3000 N for a fixed rotation angle of 320°.
(2) Fig. 5c shows the variations of the displacements with the rotation angle in a revolution for an input load F = 1100 N.
(3) Fig. 5d shows the variations of the displacements with the rotation angle in a revolution while the loads vary sinusoidally as $1100 + 400 \times \sin(\theta)$(N).

From the simulation results obtained using the model of the RE mechanism and the experimental data, it can be seen that the precision of an actual revolute joint depends on geometric errors, elasticity of components and loads. Through the RE mechanism, a generic relationship among various measurable properties and loads for an actual revolute joint is established, The precision of an actual revolute joint and its load-dependent properties can be seen as the solution of the basic equations of a corresponding RE mechanism, this opens up a new application of the theory of mechanism in design of machine components.

6 Conclusions

A novel concept of an equivalent RE mechanism consisting of two spatial cams and 10 followers with springs is presented in the paper to simulate the effects of geometrical errors and elasticity of components in an actual revolute joint. The proposed mechanism is a new spatial mechanism with redundant degrees of freedom and elastic constraints.

The displacement of the RE mechanism can easily simulate precision and properties of an actual revolute joint by taking into account of component geometric errors and their flexibility. The numerical results for the example revolute joint correlate well with the measurements.

Acknowledgments The authors acknowledge with appreciation the financial support from the National Natural Science Foundation of China (51275067).

References

1. Serkow, N.A.: Primary deviations of mechanism links with sliding and revolute pairs. J. Mach. Manufac. Reliab. **40**(2), 107–112 (2011)
2. Joscowicz, L., Sacks, E., Srinivasan, V.: Kinematic tolerance analysis. Comput.-Aided Des. **29**(2), 147–157 (1997)
3. Sadler, J.P., Yang, Zhijia: A comprehensive study of modal characteristics of a cylindrical manipulator with both link and joint flexibility. Mech. Mach. Theory **32**(8), 941–956 (1997)

4. Frisoli, A., Solazzi, M., Pellegrinetti, D., Bergamasco, M.: A new screw theory method for the estimation of position accuracy in spatial parallel manipulators with revolute joint clearances. Mech. Mach. Theory **46**(12), 1929–1949 (2011)
5. Brutti, C., Coglitore, G., Valentini, P.P.: Modeling 3D revolute joint with clearance and contact stiffness. Nonlinear Dyn. **66**(4), 531–548 (2011)
6. Delun, W., Zhi, W., Huimin, D., Shudong, Y., a novel precision-characterization model of a revolute joint based on the concept of spatial mechanisms with redundant elastic constraints. Chinese J. Mech. Eng. (Submitted)
7. Hunt, K.H.: Kinematic Geometry of Mechanisms. Oxford University Press, Oxford
8. Harris, T.A.: Rolling Bearing Analysis, 3rd edn. John Wiley and Sons Inc., New York (1990)

Obtaining the Maximal Singularity-Free Workspace of 6-U$\underline{\text{P}}$S Parallel Mechanisms via Convex Optimization

Amirhossein Karimi, Mehdi Tale Masouleh and Philippe Cardou

Abstract This paper explores the maximal volume inscribed ellipsoid in the singularity free constant-orientation workspace of two classes of 6-U$\underline{\text{P}}$S parallel mechanisms, namely, *quadratic* and *quasi-quadratic* Gough-Stewart platforms. It is of paramount importance to obtain the optimum singularity-free ellipsoid by taking into account the stroke of actuators. Convex optimization is applied as the fundamental optimization tool of this paper. For this purpose, a matrix modeling for the kinematic properties of Gough-Stewart platform is proposed. The main contribution of this paper consists in improving an existing method in a such a way that it leads to a global optimum rather than a suboptimal solution. The proposed algorithm could be regarded as one of the most reliable, in terms of obtaining the global extremum, and propitious approaches, in terms of computational time in comparison with other approaches proposed in the literature for obtaining the singularity-free workspace which make it suitable for real-time applications.

Keywords Parallel mechanisms · Stewart Gough platform · Singularity-free workspace · Convex optimization.

1 Introduction

Despite many controversial deterrents, parallel mechanisms (PMs) [1] are now widely used in different industrial contexts, such as parallel kinematic machines and pick-and-place applications. The workspace and singularities of PMs have been

A. Karimi · M. T. Masouleh
The Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran
e-mail: ah.karimi@ut.ac.ir

M. T. Masouleh
e-mail: m.t.masouleh@ut.ac.ir

P. Cardou (✉)
Department of Mechanical Engineering, Laval University, Quebec City, QC, Canada
e-mail: pcardou@gmc.ulaval.ca

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_27,

extensively studied in precious literature, due to their importance in the kinetostatic performance of the mechanism [2, 3]. To the best knowledge of the authors, few of these studies focused on analyzing the singularity-free workspace of Gough-Stewart platform, which is a definite asset in practice for path planning and control [4–6].

This paper aims at obtaining the singularity-free ellipsoid of a class of 6-DOF PMs, known as Gough-Stewart platforms, for which the singularity loci expression, for a prescribed orientation, is a second degree algebraic polynomial. These mechanisms are referred to as *quadratic* Gough-Stewart PMs. By the same token, the *quasi-quadratic* Gough-Stewart PM is defined as a mechanism for which the singularity loci expression, for a prescribed orientation, is of degree two upon fixing one translational DOF.

Convex optimization, the framework of this paper, can be regarded as a robust and reliable approach which is becoming the state-of-the-art in different disciplines due to its remarkable performance in terms of (1) computational time, (2) guaranteeing the optimality of the obtained solution and (3) providing analytical formulation of the problem.

The remainder of the paper is organized as follows. First the mechanism under study in this paper, the 6-U$\underline{\text{P}}$S PM, is broadly reviewed. We then touch briefly upon some preliminary definitions about convex optimization. Two classes of 6-U$\underline{\text{P}}$S PMs are introduced, namely, quadratic and quasi-quadratic PMs. For each case, a generalized algorithm is proposed in which the optimal ellipsoid is found within the singularity-free workspace. As case studies, the singularity-free workspaces of two given architectures of 6-U$\underline{\text{P}}$S PMs are obtained. Finally, the paper concludes by providing some remarks and describing related ongoing work.

2 Architecture Review and Kinematic Modeling

As depicted in Fig. 1(left) Gough-Stewart platform is composed of six identical limbs of the U$\underline{\text{P}}$S type, which connect the base to the moving platform. Here and throughout this paper, R, P and S stand respectively for a revolute, a prismatic and a spherical joint while the underlined joint is the actuated. The platform generates 6-DOFs by adjusting the lengths ρ_i, $i = 1, \ldots, 6$ of its prismatic joints. The pose (position and orientation) of the mobile platform is described by the two coordinate systems shown in Fig. 1(left), namely, O_{xyz} and $O_{x'y'z'}$ for the fixed and the mobile platforms, respectively. Point A_i and P_i, $i = 1, \ldots, 6$, stand respectively for the coordinate of the U and S joints with respect to the fixed frame, O_{xyz}.

The position vector of the operation point of the mobile platform with respect to the fixed frame is represented by $\mathbf{p} = [x, y, z]^T$. This operation point is chosen to be point P_1. The position vectors of point P_i in the fixed and mobile frames are denoted by $\mathbf{p}_i$ and $\mathbf{p}'_i$, respectively. The position vector of point A_i attached to the base is $\mathbf{a}_i$ with respect to the fixed frame. The rotation of the mobile platform is represented by the proper orthogonal matrix $\mathbf{Q}$ obtained from $[\phi, \theta, \psi]$, i.e., Roll-Pitch-Yaw angles respectively.

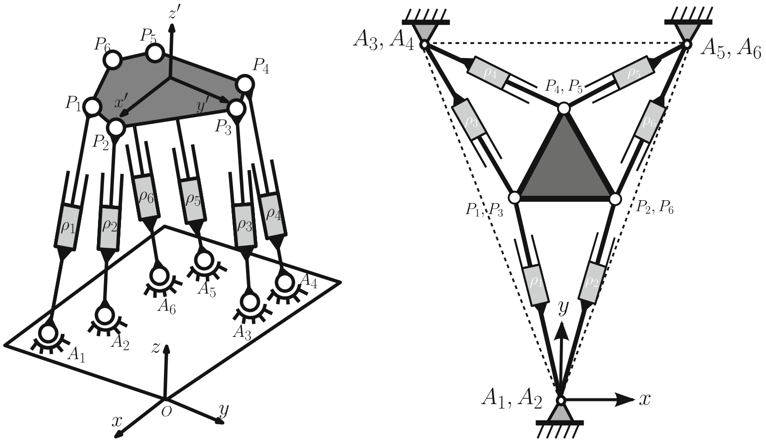


Fig. 1 *Left*: A 6-DOF Gough-Stewart PM. *Right*: a MSSM (*top view*)

2.1 Workspace Analysis

In this paper, the constant-orientation workspace is considered and the procedure to obtain the workspace of the Gough-Stewart platform is according to that proposed in [2]. The constant-orientation workspace is the set of possible Cartesian positions of the end-effector operation point for a prescribed orientation [6]. The kinematic equations corresponding to the ith limb can be expressed as:

$$\rho_i^2 = (\mathbf{p}_i - \mathbf{a}_i)^T (\mathbf{p}_i - \mathbf{a}_i) \tag{1}$$

where $\mathbf{p}_i = \mathbf{p} + \mathbf{Q}\mathbf{p}'_i$ is the position vector of each point P_i in the fixed frame. The stroke of the actuator is represent by the interval $[\rho_{\min}, \rho_{\max}]$. Substituting $\rho_{\min}$ and $\rho_{\max}$ in Eq. (1) leads to the vertex spaces of the corresponding leg, i.e., the workspace of a limb for a given orientation, with respect to the fixed frame [2]. By assuming

$$\mathbf{c}_i = \mathbf{a}_i - \mathbf{Q}\mathbf{p}'_i \tag{2}$$

as the coordinates of the center point of each sphere given in Eq. (1), one can represent the workspace, or more precisely the spheres, in a matrix formulation as follows:

$$\mathbf{x}^{\mathbf{T}}\mathbf{A}_i\mathbf{x} + 2\mathbf{b}_i^T\mathbf{x} + c_i = 0, \quad \text{for } i = 1, ..., 12 \tag{3}$$

$$\mathbf{A}_i = \mathbf{I}_{3\times 3}, \quad \mathbf{b}_i = -\mathbf{c}_i, \quad c_i = \mathbf{c}_i^T\mathbf{c}_i - \rho_i^2$$

where $\mathbf{I}_{3\times 3}$ stands for the 3×3 identity matrix.

2.2 Singularity Analysis

Singularity usually refers to configurations in which the mechanism fails to preserve its innate rigidity and consequently the mobile platform gains or loses some DOFs. The first-order kinematic equation of a general PM can be expressed as:

$$\mathbf{B}\dot{\boldsymbol{\theta}} = \mathbf{A}\mathbf{x} \tag{4}$$

where $\mathbf{x}$ and $\dot{\boldsymbol{\theta}}$ are the infinitesimal motion of the output and the input vector, respectively. Matrices $\mathbf{A}$ and $\mathbf{B}$ denote the so-called Jacobian matrices, the regularity of which is related to the singularity configurations of the mechanism. From the classification proposed in [3], each singularity configuration falls into one of three types. In this paper, we consider only Type II, which is known as the direct kinematic singularities, but which we refer to as *singularity* for the sake of brevity. Thus, the mechanism undergoes a singularity when $\mathbf{A}$ becomes rank deficient, i.e., when $\det(\mathbf{A}) = 0$.

The study carried out in [5] reveals that the quadratic Gough-Stewart platform corresponds to a design with similar base and platform. In this case, the moving platform and the base differ only by a scale factor while a quasi-quadratic Gough-Stewart platform is the one with planar base. In this architecture, all points A_i lie on a plane and the singularity equation in this case is a polynomial of degree three [5]. Furthermore, by inspection, a Gough-Stewart PM with a planar base is quadratic upon fixing z, the axis perpendicular to the fixed base. Hence we say that it is *quasi-quadratic*.

3 Review on Convex Optimization and General Mathematical Framework

A convex optimization problem can be expressed as the minimization of a convex objective function subject to inequality constraints which are all convex functions. The main feature of convex programming is that any locally optimal point is also globally optimal [7, 8]. In what follows, a preamble class of convex optimization problems is briefly introduced, which corresponds to the problem at hand in this paper.

3.1 Maximum Volume Ellipsoid Inscribed in the Intersection of Second Order Surfaces

In this case, the problem consists in obtaining $\mathcal{E}$

$$\mathcal{E} = \{\mathbf{x} \quad | \quad \mathbf{x}^{\mathbf{T}}\mathbf{P}\mathbf{x} + 2\mathbf{q}^{\mathbf{T}}\mathbf{x} + r \leq 1\} \tag{5}$$

as the maximum volume ellipsoid which satisfies all the constraints $\mathcal{C}_i$ defined as

$$\mathcal{C}_i = \{\mathbf{x} \mid \mathbf{x}^T\mathbf{A}_i\mathbf{x} + 2\mathbf{b}_i^T\mathbf{x} + c_i < 0\}, \quad i = 1, \ldots, m \tag{6}$$

where $\mathbf{A}_i$ is a $n \times n$ matrix, $\mathbf{P} \in \mathbf{S}_{++}^n$ ($\mathbf{S}_{++}^n$ represents the set of $n \times n$ symmetric positive definite matrices) and $\mathbf{q}$ is a n-dimensional vector. Since the algorithm proposed in [9] to solve such a problem leads to a suboptimal solution, we introduce here an extension to the latter approach. We propose a judicious iterative procedure in order to converge to the optimal solution. The results obtained in [9] reveal that this problem can be formulated as the following convex optimization problem:

$$\begin{aligned} \max \quad & 1 - \mathrm{tr}(\mathbf{SP}) - 2\mathbf{q}^{\mathbf{T}}\overline{\mathbf{x}} - r \\ \text{s.t.} \quad & \lambda_1, ..., \lambda_m \geq 0, \\ & \begin{bmatrix} \mathbf{P} - \lambda_i\mathbf{A}_i & \mathbf{q} - \lambda_i\mathbf{b}_i \\ (\mathbf{q} - \lambda_i\mathbf{b}_i)^T & r - 1 - \lambda_i c_i \end{bmatrix} \succeq 0, \quad i = 1, ..., m, \\ & \begin{bmatrix} \mathbf{P} & \mathbf{q} \\ \mathbf{q}^T & r \end{bmatrix} \succeq 0 \end{aligned} \tag{7}$$

which is convex in variables $\mathbf{P}$, $\mathbf{q}$, r, and $\lambda_1, ..., \lambda_m$. The objective function in Eq. (7) is the greatest lower bound on Prob$(\mathbf{X} \in \bigcap_1^m \mathcal{C}_i)$, where $\mathbf{X}$ is a random variable on $\mathbf{R}^n$, and prob$(\cdot)$ is the probability function. There is no information about the distribution of $\mathbf{X}$ except that the first and second moments $\overline{\mathbf{x}} = \mathrm{E}(\mathbf{X})$ and $\mathbf{S} = \mathrm{E}(\mathbf{X}\mathbf{X}^{\mathbf{T}})$. Therefore, Eq. (7) is referred to as a lower bound SDP (SemiDefinite Programming) [9]. The obtained ellipsoid is the locally maximum volume ellipsoid tangent to the boundaries of $\mathcal{C}_i$ at some points, which contains point $\overline{\mathbf{x}}$. Notice that it should be regarded as a suboptimal solution since it is generated by an initial guess.

In what follows, an improved approach referred to as improved lower bound SDP is proposed in order to circumvent the latter problem. The main part of this algorithm consists in defining the initial information of the probability distribution, i.e., vector $\overline{\mathbf{x}}_0$ and the matrix $\mathbf{S}_0$, the so-called initial guess. The challenge consists in defining the first guess in order to launch properly the improved approach to find the optimal ellipsoid, i.e., to find $\mathbf{P}_{\mathrm{opt}}$, $\mathbf{q}_{\mathrm{opt}}$ and r_{opt}. To do so, first, an arbitrary point satisfying all the constraints of the problem should be defined for $\overline{\mathbf{x}}_0$ and $\mathbf{S}_0$. To ease the selection of the first guess, the second moment matrix can be computed as follows from the first moment vector:

$$\mathbf{S}_0 = \overline{\mathbf{x}}_0\, \overline{\mathbf{x}}_0^T + w\mathbf{I}_{3\times3} \tag{8}$$

For a small w, upon applying the above equation into Eq. (7), one can compute the optimal ellipsoid around the mean vector $\overline{\mathbf{x}}_0$, but this is not the final and optimal solution to the problem. In order to find the optimal ellipsoid, an improved algorithm should be considered in which for each iteration the center of the expanded ellipsoid is computed as mean vector $\overline{\mathbf{x}}_i$ and the second moment matrix is computed from Eq. (8) for the new mean vector $\overline{\mathbf{x}}_i$. This means that after the first guess $\overline{\mathbf{x}}_0$, for the rest of the algorithm, the results obtained for $\mathbf{P}$, $\mathbf{q}$ and r are considered as the initial guess for pursuing the iteration.

This iterative procedure stops and returns $\mathbf{P}_{\text{opt}}$, $\mathbf{q}_{\text{opt}}$ and r_{opt} as optimal solutions when the distance between the centers of two consecutive expanded ellipsoids is smaller than a given value, ε, which means that there is not a significant change in the center of the obtained ellipsoid by pursuing the procedure and the optimum ellipsoid is attained:

$$||\overline{\mathbf{x}}_i - \overline{\mathbf{x}}_{i-1}||_2 \leq \varepsilon \tag{9}$$

4 Convex Modeling of Singularity-Free Workspace

In this section, the aforementioned improved lower bound SDP algorithm is applied in order to investigate the singularity-free workspaces of two Gough-Stewart platforms. The optimization problem presented in Eq. (7) can be solved by resorting to the CVX package, a convex optimization package implemented in MATLAB by Grant and Boyd (2011) [10].

4.1 Case Study I: Similar Base and Platform

As the first case study, the Gough-Stewart PM with geometric parameters given in Table 1 is considered. Its base and moving platform being similar, thus this Gough-Stewart PM is quadratic.

Table 1 Geometric parameters of the PMs under study

i	1	2	3	4	5	6
x_{a_i}	0	1	2	1	−1	−1
y_{a_i}	0	0	1	2	2	1
z_{a_i}	0	0	0	2	1	0
x'_{p_i}	0	0.5	1	0.5	−0.5	−0.5
y'_{p_i}	0	0	0.5	1	1	0.5
z'_{p_i}	0	0	0	1	0.5	0
$(\rho_{\min})_i$	1	1	1	1	1	1
$(\rho_{\max})_i$	4	4	4	4	4	4
x_{a_i}	0	0	$\frac{1}{\sqrt[4]{3}}$	$\frac{1}{\sqrt[4]{3}}$	$-\frac{1}{\sqrt[4]{3}}$	$-\frac{1}{\sqrt[4]{3}}$
y_{a_i}	0	0	$\sqrt[4]{3}$	$\sqrt[4]{3}$	$\sqrt[4]{3}$	$\sqrt[4]{3}$
z_{a_i}	0	0	0	0	0	0
x'_{p_i}	$-\frac{\sqrt[4]{27}}{5}$	$\frac{\sqrt[4]{27}}{5}$	$\frac{\sqrt[4]{27}}{5}$	0	0	$-\frac{\sqrt[4]{27}}{5}$
y'_{p_i}	0	0	0	$\frac{3\sqrt[4]{3}}{5}$	$\frac{3\sqrt[4]{3}}{5}$	0
z'_{p_i}	0	0	0	0	0	0
$(\rho_{\min})_i$	1	1	1	1	1	1
$(\rho_{\max})_i$	4	4	4	4	4	4

The problem can be formulated as follows. The matrix formulation of the 12 spheres, $\mathcal{C}_i$, six inner and six outer spheres, is obtained from Eq. (3). The singularity equation can be reformulated readily as Eq. (6). The optimization problem given in Eq. (7) is then solved to obtain $\mathbf{P}_{\text{opt}}$, $\mathbf{q}_{\text{opt}}$ and r_{opt}. Finally, we implement the improved lower bound SDP method introduced in Sect. 3, in order to obtain the maximal singularity-free ellipsoid.

It should be noted that the optimal solution obtained from the improved lower bound SDP can be changed according to the choice of the initial guess $\overline{\mathbf{x}_0}$. Thus, selection of different initial guesses in each feasible subregion of the singularity-free workspace results in the same optimal solution. However, the optimal solution varies based on the selection of $\bar{\mathbf{x}}_0$ within other feasible subregions of singularity-free workspace. Figure 2 depicts the results obtained for $\varphi = \theta = \psi = \frac{\pi}{4}$. Two initial guesses are considered, $\bar{\mathbf{x}}_0 = [2, 2, 0]^T$ and $\omega = 0.1$, and $\bar{\mathbf{x}}_0 = [-1, -1, 1]^T$ and $\omega = 0.1$, which lead to two distinct optimal solutions as depicted in Fig. 2. Figure 2(right) represents the result for $z = 0.3$ including the iterative procedure, i.e., the dashed ellipses, to find the optimal solution, the solid ellipse. For this case, the computational time obtained by a PC equipped with an Intel(R) Core(TM) i5-2430M CPU @ 2.40 GHz, and 4 GB RAM is 0.6 s.

4.2 Case Study II: MSSM

As represented in Fig. 1(right), the Minimal Simplified Symmetric Manipulator (MSSM) is an architecture for which the base and the moving platform are isosceles triangles. In this case, a MSSM with an equilateral triangle base of unit area is considered. The moving platform is also an equilateral triangle with area of $\frac{9}{25}$. Table 1 represents the geometric parameters of this mechanism. Since in this architecture,

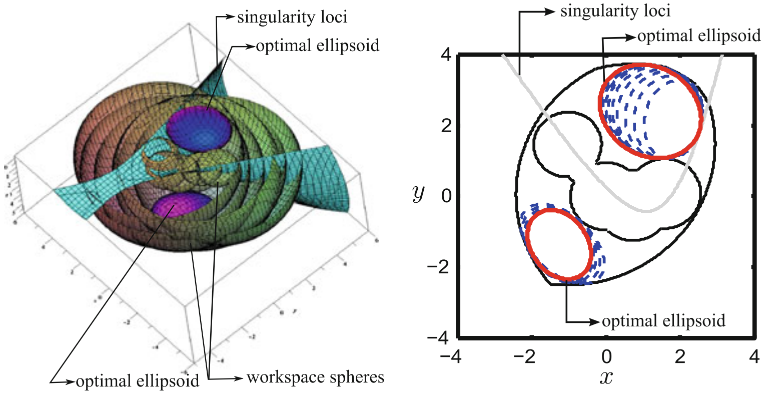


Fig. 2 The maximum-volume singularity-free ellipsoid in the workspace of 6-U$\underline{\text{P}}$S for $\varphi = \theta = \psi = \frac{\pi}{4}$. *Left*: the 3D view. *Right*: the cross-section for $z = 0.3$

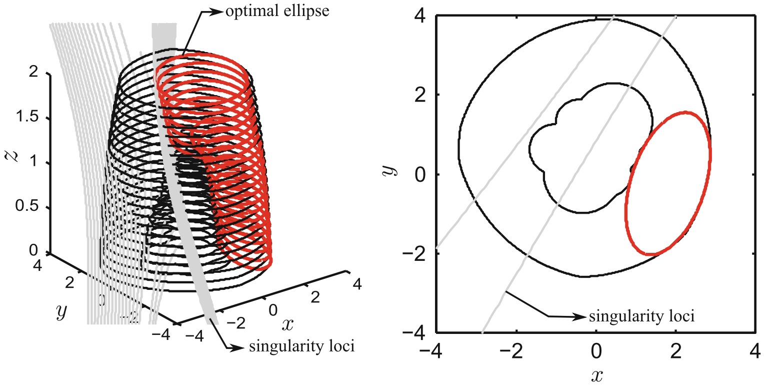


Fig. 3 The maximum-area singularity-free ellipses in the workspace of MSSM for $\varphi = \frac{\pi}{6}, \theta = \frac{\pi}{4}, \psi = 0$. *Left*: the isometric view for $0 \leq z \leq 2$. *Right*: the cross-section for $z = 0.5$

the base of the mechanism is coplanar, the singularity equation is of quasi-quadratic type, i.e., the singularity equation becomes quadratic upon fixing z. Following the same reasoning explained for the previous case, the maximum area ellipse for each section, i.e., $z \in [0, 2]$ in this case, can be obtained via the improved lower bound SDP approach. By starting from $\bar{\mathbf{x}}_0 = [1, 0]^T$ and $\omega = 0.1$, after a computational time of 7.1 s, the optimal ellipse found as depicted in Fig. 3 for $\varphi = \frac{\pi}{6}, \theta = \frac{\pi}{4}$, and $\psi = 0$. Figure 3(left) represents the general result with $\Delta z = 0.1$ as the increment value for each cross section while Fig. 3(right) depicts the result for a given cross section, $z = 0.5$.

5 Conclusion

This paper investigates the singularity-free workspace of two classes of 6-DOF parallel mechanisms referred to as quadratic and quasi-quadratic Gough-Stewart platforms. An extension to an existing approach was presented in order to converge to the optimal solution from an initial guess. In terms of computational time, the proposed algorithms provides some outstanding results with respect to others reported in the literature. Ongoing work include obtaining the appropriate design parameters for a prescribed singularity-free ellipsoid, namely, performing the dimensional synthesis of Gough-Stewart platforms.

References

1. Kong, X., Gosselin, C.: Type Synthesis of Parallel Mechanisms, vol. 33. Springer, Heidelberg (2007)
2. Gosselin, C.: Determination of the workspace of 6-DOF parallel manipulators. ASME J. Mech. Des. **112**(3), 331–336 (1990)
3. Gosselin, C., Angeles, J.: Singularity analysis of closed-loop kinematic chains. IEEE Trans. Robotics Autom. **6**(3), 281–290 (1990)
4. Li, H., Gosselin, C.M., Richard, M.J.: Determination of the maximal singularity-free zones in the six-dimensional workspace of the general Gough-Stewart platform. Mech. Mach. Theory **42**(4), 497–511 (2007)
5. Jiang, Q.: Singularity-free workspace analysis and geometric optimization of parallel mechanisms. Ph.D. thesis, Laval University, Quebec, Oct 2008
6. Merlet, J.P.: Parallel Robots. Springer, Dordrecht (2006)
7. Rao, S.S.: Engineering Optimization: Theory and Practice. Wiley, Hoboken (2009)
8. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
9. Vandenberghe, L., Boyd, S., Comanor, K.: Generalized chebyshev bounds via semidefinite programming. SIAM Rev. **49**(1), 52–64 (2007)
10. Grant, M., Boyd. S.: CVX Users Guide for CVX Version 1.21. April 2011

Type Synthesis of Two DOF Hybrid Translational Manipulators

Latifah Nurahmi, Stéphane Caro and Sébastien Briot

Abstract This paper introduces a new methodology for the type synthesis of two degrees of freedom hybrid translational manipulators with identical legs. The type synthesis method is based upon screw theory. Three types of two degrees of freedom hybrid translational manipulators with two identical legs are identified based upon their wrench decomposition. Each leg of the manipulators is composed of a proximal module and a distal module mounted in series. The assembly conditions and the validity of the actuation scheme are also defined. Finally, some novel two degrees of freedom hybrid translational manipulators are synthesized with the proposed procedure.

Keywords Type synthesis · Parallel mechanisms · Hybrid legs · Screw theory

1 Introduction

At the conceptual design stage of manipulator architectures, the idea is to construct several design alternatives by following a systematic approach. However, the information at this stage is usually qualitative and not quantitative, which makes the design process quite difficult and challenging.

A manipulator is a mechanical system that aims at *manipulating* objects. For simple task such as pick-and-place operations, the two degrees of freedom (*dof*) parallel manipulators may be sufficient. Several two-*dof* translational parallel manipulators (*TPM*) are composed of a planar architecture that yields their stiffness quite low along

L. Nurahmi · S. Caro (✉) · S. Briot
Institut de Recherche en Communications et Cybernétique de Nantes, Nantes, France
e-mail: caro@irccyn.ec-nantes.fr

L. Nurahmi
e-mail: nurahmi@irccyn.ec-nantes.fr

S. Briot
e-mail: briot@irccyn.ec-nantes.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_28,

the normal to the plane of motion [3]. Moreover, those manipulators are usually not composed of identical legs.

In order to increase the stiffness properties of the two-*dof TPM*, some researchers have proposed a new manipulator architecture named the Par2 [2]. This architecture has the particularity to be spatial instead of planar and thus is stiffer along the normal to the plane of motion. Two legs amongst the four legs of the manipulator are linked to each other with a rigid belt in order to constrain the rotation of the moving platform. As a consequence, it leads to a robot with a poor accuracy. To avoid the design problems of the Par2, a new robot with spatial architecture and two legs has been proposed: the IRSBot-2 [4]. Each leg of the IRSBot-2 is hybrid, i.e. it is composed of a proximal module (PM) and a distal module (DM) mounted in series, each module containing two kinematic chains. This mechanism has exhibited interesting stiffness properties. Therefore, it is of interest to focus on the type synthesis of two-*dof TPM* by considering architectures with hybrid legs.

The subject of the paper is about the type synthesis of of two-*dof* hybrid translational manipulators with identical legs. Each leg is composed of a proximal module and a distal module mounted in series. Those modules contain one kinematic chain or two kinematic chains mounted in parallel. These kinematic chains are called sub legs. This research work has been carried out in the framework of the French National Project[1] that aims to develop some fast and accurate robots with a large operational workspace.

In this paper, the general approach for the type synthesis of manipulators is developed based on [5]. The screw systems used in this paper are adopted from [5]. The approach is decomposed into five steps: (i) Classification of the leg-constraint wrench system; (ii) Decomposition of the constraint wrench system of a proximal and distal module; (iii) Type synthesis of the sub legs; (iv) Assembly of the sub legs and legs; (v) Selection of the actuated joints.

2 Two DOF Hybrid Translational Manipulators with Two Identical Legs

The general approach for the type synthesis of two *dof* hybrid translational manipulators with two identical legs are presented using the following procedure.

2.1 Step 1: Classification of the Leg-Constraint Wrench System

The moving platform of two-*dof* hybrid translational manipulators is intended to perform a two translational motion in plane $(\mathbf{x}O\mathbf{z})$. In general configuration, the twist

[1] This work has been partially funded by the French French National Project ANR 2011-BS3-006-01-ARROW (http://arrow.irccyn.ec-nantes.fr/)

system of the moving-platform amounts to a $2\xi_\infty-$system.[2] Therefore, the overall constraint wrench system $\mathcal{W}^c$ is a $1\zeta_0 - 3\zeta_\infty-$system,[3] containing one zero-pitch wrench along **y**-axis and three independent infinite-pitch wrenches. Such manipulators can be obtained by a combination of any leg-constraint wrench system of order $c^i (1 \le c^i \le 4)$, decomposed as follows:

- $c^i = 4 \rightarrow 1\zeta_0 - 3\zeta_\infty-$system
- $c^i = 3 \rightarrow 1\zeta_0 - 2\zeta_\infty-$system, $3\zeta_\infty-$system
- $c^i = 2 \rightarrow 1\zeta_0 - 1\zeta_\infty-$system, $2\zeta_\infty-$system
- $c^i = 1 \rightarrow 1\zeta_0-$system, $1\zeta_\infty-$system

However, only three types of two-*dof* hybrid translational manipulators can be properly assembled with two identical legs, namely:

- **Type 1**: $c^i = 4 \rightarrow 1\zeta_0 - 3\zeta_\infty-$system
- **Type 2**: $c^i = 3 \rightarrow 1\zeta_0 - 2\zeta_\infty-$system
- **Type 3**: $c^i = 2 \rightarrow 1\zeta_0 - 1\zeta_\infty-$system

Type 3 is included in this paper, since the combination of two parallel legs with $1\zeta_0-1\zeta_\infty-$system can generate a manipulator with overall constraint wrench system $1\zeta_0 - 3\zeta_\infty-$system from [1].

The overall constraint wrench system of one leg is determined by the intersection of constraint wrench systems associated with the proximal and distal modules as they are mounted in series. Therefore, the constraint wrench systems associated with the proximal and distal modules are decomposed thereafter. Note that these constraint wrench decompositions are interchangeable between proximal and distal modules.

2.1.1 Type 1: $c^i = 4 \rightarrow 1\zeta_0 - 3\zeta_\infty-$System

The overall constraint wrench system of one leg for Type 1, is $c^i = 4$, $1\zeta_0 - 3\zeta_\infty-$system. Thus, the feasible constraint wrenches are: *Proximal module:*

1. $2\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
2. $1\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$

Distal module:

1. $2\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
2. $1\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$

At least one ζ_0 from the constraint wrench systems described above is along **y**-axis.

[2] ξ_∞ denotes an infinite-pitch twist, namely, a pure translation.

[3] ζ_0 and ζ_∞ denote a zero-pitch wrench (a pure force) and an infinite-pitch wrench (a pure moment), respectively.

2.1.2 Type 2: $c^i = 3 \rightarrow 1\zeta_0 - 2\zeta_\infty-$System

The overall constraint wrench system of one leg for Type 2, is $c^i = 3$, $1\zeta_0 - 2\zeta_\infty-$system. Hence, the possible constraint wrenches are:
Proximal module:

1. $2\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
2. $1\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
3. $2\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2})$
4. $1\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2})$

Distal module:

1. $2\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2})$
2. $1\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2})$

At least one ζ_0 from the constraint wrench systems described above is along **y**-axis.

2.1.3 Type 3: $c^i = 2 \rightarrow 1\zeta_0 - 1\zeta_\infty-$System

The overall constraint wrench system of one leg for Type 3, is $c^i = 2$, $1\zeta_0 - 1\zeta_\infty-$system. Hence, the possible constraint wrenches are:
Proximal module:

1. $2\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
2. $1\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
3. $2\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2})$
4. $1\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2})$
5. $2\zeta_0 - 1\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{01}, \zeta_\infty)$
6. $1\zeta_0 - 1\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_\infty)$

Distal module:

1. $2\zeta_0 - 1\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{01}, \zeta_\infty)$
2. $1\zeta_0 - 1\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_\infty)$

At least one ζ_0 from the constraint wrench systems described above is along **y**-axis.

2.2 Step 2: Decomposition of the Constraint Wrench for Proximal and Distal Modules

The next step for type synthesis of two *dof* hybrid translational manipulators is the decomposition of the constraint wrench for proximal and distal modules. It is noteworthy that Types 1, 2 and 3 have the following constraint wrench systems for proximal and distal modules:

1. $2\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
2. $1\zeta_0 - 3\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2}, \zeta_{\infty 3})$
3. $2\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{02}, \zeta_{\infty 1}, \zeta_{\infty 2})$
4. $1\zeta_0 - 2\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_{\infty 1}, \zeta_{\infty 2})$
5. $2\zeta_0 - 1\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_{01}, \zeta_{01}, \zeta_\infty)$
6. $1\zeta_0 - 1\zeta_\infty-$system, $\mathcal{W} = \text{span}(\zeta_0, \zeta_\infty)$

Such module constraint wrench systems emerge from the vector sum of the subleg-constraint wrench systems, due to the in-parallel arrangement of two identical sublegs. All potential identical subleg combinations are presented in Table 1.

2.3 Step 3: Type Synthesis of Sublegs

Once the different combinations of subleg-constraint wrench systems have been achieved (Table 1), the kinematic chains that instantiate proximal and distal modules can be determined. The concept of virtual chain [5] is used to generate subleg types. A virtual chain is a serial kinematic chain associated with the motion pattern of the proximal and distal modules. The number of joints in the subleg is defined based on the following mobility criterion [5]:

$$f = F + (6 - c) \tag{1}$$

where f is the number of 1-*dof* joints, F is the mobility of a single-loop kinematic chain, and c is the order of the wrench system.

2.3.1 Step 3a: Type Synthesis of Single-Loop Kinematic Chains that Involve a Virtual Chain and have a Specified Subleg-Constraint Wrench System

Type synthesis of single-loop kinematic chains is illustrated for modules with $2\zeta_0 - 3\zeta_\infty-$system and $1\zeta_0 - 3\zeta_\infty-$system only due to space limitation.

Module with $2\zeta_0 - 3\zeta_\infty-$System

This module has 1-*dof* and can be represented by a P[4]-virtual chain. Let us consider a subleg whose constraint wrench system is $2\zeta_0 - 3\zeta_\infty-$system (from Table 1).

[4] P stands for a prismatic joint and R stands for a revolute joint.

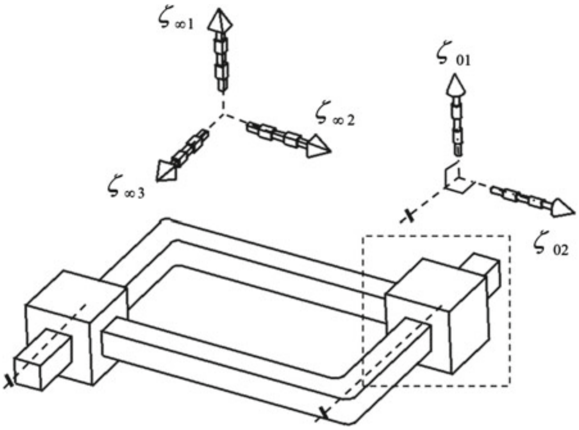


Fig. 1 P subleg with a P-virtual chain

The number of joints that involves a P-virtual chain and has a $2\zeta_0 - 3\zeta_\infty-$system is defined as follows:

$$c^i = 5, \quad f = F + (6 - c) = 1 + (6 - 5) = 2 \text{ joints} \tag{2}$$

This single-loop kinematic chain can only be formed by one P joint, whose direction is perpendicular to the axes of $2\zeta_0$ as depicted in Fig. 1.

Module with $1\zeta_0 - 3\zeta_\infty-$System

This module has 2-*dof* and can be represented by a PP-virtual chain. Let us consider a subleg whose constraint wrench system is $1\zeta_0 - 2\zeta_\infty-$system (from Table 1). The number of joints that involves a PP-virtual chain and has a $1\zeta_0 - 2\zeta_\infty-$system is:

$$c^i = 3, \quad f = F + (6 - c) = 2 + (6 - 3) = 5 \text{ joints} \tag{3}$$

Such a single-loop kinematic chain contains two cases: perpendicular case and general case. For perpendicular case, all R joint axes are parallel to the axis of ζ_0 and all directions of P joints are perpendicular to the axis of ζ_0. In the general case, all R joint axes are perpendicular to the directions of ζ_∞ and coplanar with the axis of ζ_0. One example of RRR subleg is given in Fig. 2 for perpendicular case.

2.3.2 Step 3b: Generation of Types of Sublegs

The types of sublegs are derived by removing the P-virtual chain and PP-virtual chain from Figs. 1 and 2. Those types of sub legs are given in Table 1. Nonetheless, several

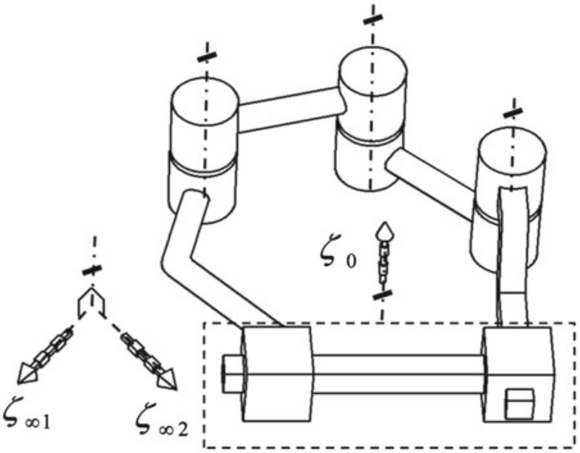


Fig. 2 RRR subleg with PP-virtual chain

types of sub legs that contain a non-invariant sub leg-wrench system (kinematic chains with varying wrench system) [5] are still kept as they can produce two-*dof* translational motions. All these types of sub legs can be assembled to generate either proximal modules or distal modules.

2.4 Step 4: Assembly of Sublegs and Legs

The assembly process of two-*dof* hybrid translational manipulators is performed for Types 1, 2 and 3. Each leg of manipulator is realized by mounting a proximal module and a distal module in series. Then, each leg becomes a hybrid manipulator and is attached to the base at one end and to the moving-platform at the other end. Therefore, the assembly process consists of two steps as explained thereafter.

2.4.1 Step 4a: Assembly of Sublegs → Proximal and a Distal Modules

The proximal and distal modules can be obtained by assembling some sublegs from Table 1. Nevertheless, the following conditions should be respected:

1. The overall wrench system of a module should constitute the desired wrench system, as explained in Step 1.
2. At least one translational twist generated by the module should lie in plane ($\mathbf{x}O\mathbf{z}$).

Table 1 Type of sublegs for proximal and distal modules free of inactive joint

c	Wrench system of PM and DM	Subleg wrench system	Type	Note
5	$2\zeta_0 - 3\zeta_\infty$	$2\zeta_0 - 3\zeta_\infty$	Permutation P	
		$2\zeta_0 - 2\zeta_\infty$	Permutation RR	Non-invariant
		$1\zeta_0 - 2\zeta_\infty$	Permutation RRR	
			Permutation PRR	
4	$1\zeta_0 - 3\zeta_\infty$	$1\zeta_0 - 3\zeta_\infty$	Permutation PP	
		$1\zeta_0 - 2\zeta_\infty$	Permutation RRR	
		$1\zeta_0 - 1\zeta_\infty$	Permutation PRR	
			Permutation RRŔŔ	Non-invariant
	$2\zeta_0 - 2\zeta_\infty$	$2\zeta_0 - 2\zeta_\infty$	Permutation PR	
			Permutation RR	Non-invariant
		$1\zeta_0 - 2\zeta_\infty$	Permutation RRR	
			Permutation PRR	
		$2\zeta_0 - 1\zeta_\infty$	Permutation RRŔ	Non-invariant
			Permutation RŔŔ	Non-invariant
		$1\zeta_0 - 1\zeta_\infty$	Permutation RRŔŔ	Non-invariant
		$2\zeta_0$	Permutation RRŔŘ	Non-invariant
			Permutation PRŔŘ	
3	$1\zeta_0 - 2\zeta_\infty$	$1\zeta_0 - 2\zeta_\infty$	Permutation RRR	
			Permutation PRR	
			Permutation PPR	
		$1\zeta_0 - 1\zeta_\infty$	Permutation RRŔŔ	Non-invariant
	$2\zeta_0 - 1\zeta_\infty$	$2\zeta_0 - 1\zeta_\infty$	Permutation PRŔ	
		$1\zeta_0 - 1\zeta_\infty$	Permutation RRRŔ	
			Permutation PRRŔ	
			Permutation PRŔŔ	
		$2\zeta_0$	Permutation RRŔŘ	
			Permutation PRŔŘ	
2	$1\zeta_0 - 1\zeta_\infty$	$1\zeta_0 - 1\zeta_\infty$	Permutation RRRŔ	
			Permutation PRRŔ	
			Permutation PRŔŔ	
			Permutation PPRŔ	
		$1\zeta_0$	Permutation RRRŔŘ	
			Permutation RRŔŔŘ	
			Permutation RŔŔŘŘ	
			Permutation RRŔŘŘ	
			Permutation PRRŔŘ	
			Permutation PRŔŔŘ	
			Permutation PRŔŘŘ	
			Permutation PPRŔŘ	

2.4.2 Step 4b: Assembly of Legs → Two-dof Hybrid Manipulator

The legs of the two-*dof* hybrid translational manipulators are synthesized by mounting in series the proximal and distal modules derived in Step. 4a. However, the following conditions should be fulfilled:

1. The wrench system of the leg should of Type 1, Type 2, or Type 3 namely, it should be a $1\zeta_0 - 3\zeta_\infty -$system, a $1\zeta_0 - 2\zeta_\infty -$system, or a$1\zeta_0 - 1\zeta_\infty -$system.
2. The linear combination of the wrench systems associated with the legs should be a $1\zeta_0 - 3\zeta_\infty -$system.

Figure 3 illustrates a novel two-*dof* hybrid translational manipulators with identical legs. This is a Type 1 mechanism that has been synthesized with the proposed approach. Each leg has a $1\zeta_0 - 3\zeta_\infty -$wrench system. Both proximal and distal modules have a $2\zeta_0 - 3\zeta_\infty -$wrench system and are composed of two RRR legs, known as *Sarrus Linkage*. This novel mechanism is named Q-Sarrus, Q standing for Quadruple.

Figure 4 depicts another mechanism synthesized with the proposed type-synthesis approach. This mechanism is named IRSBot-2 [4] and is of Type 2. Its proximal modules have a $2\zeta_0 - 3\zeta_\infty -$system and are made up of a Π joint. Its distal modules have a $2\zeta_0 - 2\zeta_\infty -$system and are composed of 2-UU kinematic chains.

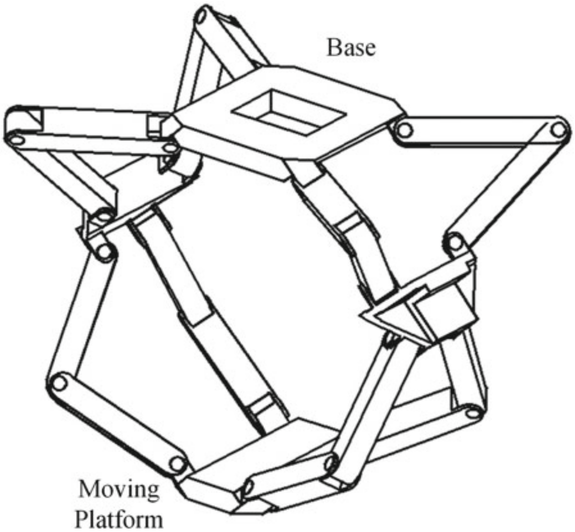


Fig. 3 Q-Sarrus

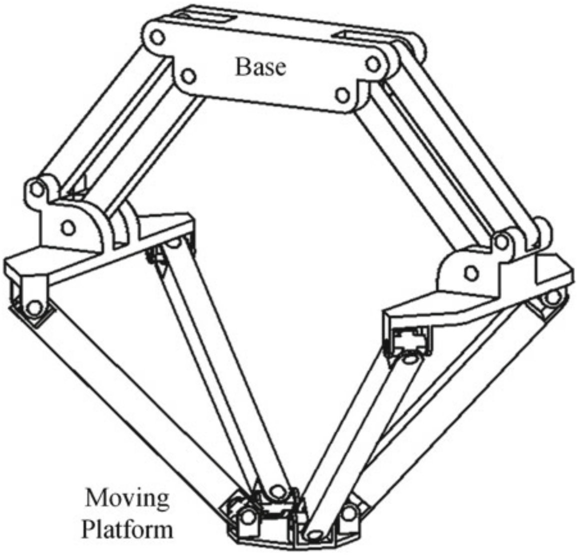


Fig. 4 IRSBot-2

2.5 Step 5: Selection of the Actuated Joints

Let assume that the condition of constraint wrench system is satisfied, namely, the assembly of legs applies a $1\zeta_0 - 3\zeta_\infty -$system on the moving-platfom. In a general configuration, a set of constraint wrench system, $\mathcal{W}^c$, together with an actuation wrench system, $\mathcal{W}^a$, constitute a 6-system. Ultimately, the selection of actuated joints for two-*dof* hybrid translational manipulators can be made in such a way that a basis of the actuation wrench system $\mathcal{W}^a$ contains at least two actuation forces.

3 Conclusion

A general approach has been introduced in this paper for the type synthesis of two-*dof* hybrid translational manipulators with identical legs. The proposed approach is based on the screw theory and the method described in [5]. Two types of two-*dof* hybrid translational manipulators have been highlighted with regard to the wrench system associated with their legs. Moreover, many novel two-*dof* hybrid translational manipulators have been obtained and two of them have been illustrated, namely, the Q-Sarrus and the IRSBot-2. The comparison of the synthesized manipulators with regard to their complexity and intrinsic stiffness is part of the future work.

References

1. Carricato, M.: Persistent screw systems of dimension four. In: Springer (ed.) The 13th International Symposium on Advances in Robot Kinematics, pp. 147–156 (2012)
2. Company, O., Pierrot, F., Krut, S., Baradat, C., Nabat, V.: Par2: a spatial mechanism for fast planar two-degree-of-freedom pick-and-place applications. Meccanica **46**(1), 239–248 (2011)
3. Gao, F., Li, W., Zhao, X., Jin, Z., Zhao, H.: New kinematic structures for 2-, 3-, 4-, and 5-DOF parallel manipulator designs. Mech. Mach. Theory **37**(11), 1395–1411 (2002)
4. Germain, C., Caro, S., Briot, S., Wenger, P.: Singularity-free design of the translational parallel manipulator irsbot-2. Mech. Mach. Theory **64**, 262–285 (2013)
5. Kong, X., Gosselin, C.: Type Synthesis of Parallel Mechanism. Springer, Germany (2007)

Robust Dynamic Control of an Arm of a Humanoid Using Super Twisting Algorithm and Conformal Geometric Algebra

O. Carbajal-Espinosa, L. González-Jiménez, A. Loukianov and E. Bayro-Corrochano

Abstract The pose tracking problem for the 5 DOF (degrees of freedom) arm of a humanoid robot is studied. The kinematic and dynamic models of the manipulator are obtained using the conformal geometric algebra framework. Then, using the obtained models, the well known super-twisting algorithm, is used to design a controller in terms of the conformal geometric algebra for the pose tracking problem. Simulation shows the performance of the proposed controller with the conformal models for the tracking an object.

Keywords Kinematics modeling · Dynamic modeling · Super twisting algorithm · Conformal geometric algebra · Humanoid manipulator

1 Introduction

The geometrical relationships between the kinematical chains of a humanoid and an object of interest in the environment, determine the reference for their position and orientation. To follow a target on the task space, a controller must be designed in order to assure a proper torque values on each joint of the manipulator. First, models of the

O. Carbajal-Espinosa (✉), A. Loukianov, E. Bayro-Corrochano
CINVESTAV, Department of Electrical Engineering and Computer Sciences,
Unidad Guadalajara, Mexico
e-mail: ocarbajal@gdl.cinvestav.mx

A. Loukianov
e-mail: louk@gdl.cinvestav.mx

E. Bayro-Corrochano
e-mail: edb@gdl.cinvestav.mx

L. González-Jiménez
ITESO, Department of Electronic, Systems and Informatics, Tlaquepaque, Mexico
e-mail: luisgonzalez@iteso.mx

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_29,

kinematics and dynamics of the robot must be obtained, for this end, the conformal geometric algebra (CGA) framework allows the representation of rigid transformations (rotations, translations, screw motions) and geometric entities (points, lines, spheres, etc), these entities will serve to model the structure of the robot, for example, lines can be considered as links of the manipulator. Moreover, the composition of several rigid transformations acting over a geometric entity can be computed as a sequence of geometric products of consecutive motors (the conformal entity that represents a 3D rigid transformation) [1, 4]. In this work, the conformal geometric algebra (CGA) framework is used to obtain the direct, differential kinematical and forward dynamic models of the arm of a humanoid. The rest of the work is organized as follows. Section 2 presents an introduction to Conformal Geometric Algebra framework, introducing some geometric primitives and rigid transformations in CGA used in this work. Section 3 presents the kinematic and dynamic modeling of serial manipulator. In Sects. 4 and 5 present the control strategy and the simulation results of the proposed method. Finally, some conclusions are given in Sect. 6.

2 Geometric Algebra

We will use G_n to denote the geometric algebra of n-dimensions, which is a graded-linear space. As well as vector-addition and scalar multiplication, we have a non commutative product which is associative and distributive over addition. This is the *geometric* or *Clifford product*. The inner product of two vectors is the standard *scalar* or *dot* product, which produces a scalar. The outer or wedge product of two vectors is a new quantity which we call a *bivector*.

In this paper we will specify the geometric algebra G_n by $G_{p,q,r}$, where p, q and r stand for the number of basis vectors which square to 1, -1 and 0 respectively and fulfill $n = p + q + r$. The entire basis of G_n is defined as the ordered set:

$$\{1\}, \{e_i\}, \{e_i \wedge e_j\}, \{e_i \wedge e_j \wedge e_k\}, \ldots, \{e_1 \wedge e_2 \wedge \ldots \wedge e_n\} \tag{1}$$

where e_i denote the basis vector i, and has the following properties:

$$e_i e_j = \begin{cases} 1 \ for \ i = j \in 1, \cdots, p \\ -1 \ for \ i = j \in p+1, \cdots, p+q \\ 0 \ for \ i = j \in p+q+1, \cdots, p+q+r. \\ e_i \wedge e_j \ for \ i \neq j \end{cases}$$

The *Conformal Geometric Algebra*, $G_{4,1} = G_{4,1,0}$, can be used to treat conformal geometry in a very elegant way, representing the Euclidean vector space $\mathbb{R}^3$ in $\mathbb{R}^{4,1}$, for a more complete treatment, the reader is referred to the texts by [1, 4]. This space has an orthonormal vector basis given by $\{e_i\}$ and $e_{ij} = e_i \wedge e_j$ are bivectorial bases. The unit Euclidean pseudo-scalar $I_e := e_1 \wedge e_2 \wedge e_3$, a pseudo-scalar $I = I_e E$, andthe bivector $E := e_4 \wedge e_5 = e_4 e_5$ are used for computing Euclidean and conformal duals of multivectors. A null can be defined as

$$e_\infty = e_4 + e_5,\ e_0 = \frac{1}{2}(e_4 - e_5) \tag{2}$$

where e_∞ is the point at infinity and e_0 is the origin point. These two null vector satisfies

$$e_\infty^2 = e_0^2 = 0,\ e_\infty \cdot e_0 = 1$$

The *point* is written as

$$x_c = x_e + \frac{1}{2}x_e^2 e_\infty + e_0$$

Given two conformal points x_c and y_c, it can be defined $x_c - y_c = (y_c \wedge x_c) \cdot e_\infty$ and, consequently, the following equality is fulfilled as well:

$$(x_c \wedge y_c + y_c \wedge z_c) \cdot e_\infty = (x_c \wedge z_c) \cdot e_\infty \tag{3}$$

The point will be use to define the position of each joint and the center of mass of each link on a manipulator.

The *Lines* will be used to define the rotation axes and orientation of the manipulator, and can be defined in CGA as a circle passing through the point at infinity. The Outer Product Null Space (OPNS) form of a line is represented as

$$L^* = x_{c_1} \wedge x_{c_2} \wedge e_\infty \tag{4}$$

The standard Inner Product Null Space (IPNS) form of the line can be expressed as

$$L = \mathbf{n} I_e - e_\infty \mathbf{m} I_e \tag{5}$$

where **n** and **m** stand for the line orientation and moment respectively. Given two lines L_a and L_b we can define a third error line L_e as

$$L_\varepsilon = L_a - L_b \tag{6}$$

Rigid transformations can be expressed in conformal geometric algebra by carrying out successive plane reflections.

The *translation* to a vector $a \in \mathbb{R}^3$, of conformal geometric entities can be done by carrying out two reflections in parallel planes π_1 and π_2. That is $T_a Q \widetilde{T_a}$, where

$$T_a = (n + de_\infty)n = 1 + \frac{1}{2}ae_\infty = e^{\frac{a}{2}e_\infty} \tag{7}$$

The *rotation* is the product of two reflections at nonparallel planes which pass through the origin. That is $R_\theta Q \widetilde{R_\theta}$, where

$$R_\theta = cos\left(\frac{\theta}{2}\right) - sin\left(\frac{\theta}{2}\right) l = e^{-\frac{\theta}{2} l} \tag{8}$$

with $l = n_2 \wedge n_1$, and θ twice the angle between the planes π_2 and π_1.

The screw motion *called motor* related to an arbitrary axis L is $M = TR\widetilde{T}$ and it is applied in the same way than a rotor, $M_\theta Q \widetilde{M_\theta}$, where

$$M_\theta = TR\widetilde{T} = cos\left(\frac{\theta}{2}\right) - sin\left(\frac{\theta}{2}\right) L = e^{-\frac{\theta}{2} L} \tag{9}$$

3 Kinematic and Dynamic Modeling of Manipulators

The *direct kinematics* of a manipulator consists in calculating the position and orientation of the end-effector of a serial robot using the values of the joint variables. If the joint variable is a translation, $M_i = T_i = \exp^{-dne_\infty}$ for a prismatic joint and a rotation $M_i = R_i = \exp^{\frac{-\theta L_r}{2}}$ for a revolutive joint. The direct kinematics for a serial robot is a successive multiplication of motors given by

$$Q' = M(q)_1 \ldots M(q)_n Q \widetilde{M(q)}_n \ldots \widetilde{M(q)}_1 = \left(\prod_{i=1}^{n} M(q)_i Q \prod_{i=1}^{n} \widetilde{M(q)}_{i=n-i+1} \right) \tag{10}$$

for a given angular or translation position vector $q = [q_1 \; \ldots \; q_n]^T$.

The *differential kinematics* of the system results from the differentiation of (10) for points and lines, and is given by

$$\dot{x}'_p = J_x \dot{q}, \quad \dot{L}'_p = J_L \dot{q} \tag{11}$$

with $\dot{q} = [\dot{q}_1 \; \ldots \; \dot{q}_n]$, and the Jacobian matrices defined as $J_x = [x'_p \cdot L'_1 \; \ldots \; x'_p \cdot L'_n]$, $J_L = [\alpha_1 \; \ldots \; \alpha n]$, where

$$L'_j = \left(\prod_{i=1}^{j-1} M_i \right) L_j \left(\prod_{i=1}^{j-1} \widetilde{M}_{j-i} \right), \quad \alpha_j = \frac{1}{2}\left(L'_p L'_j - L'_j L'_p \right) \tag{12}$$

and L_i is the axis for the ith joint in the initial position. Please refer to [9] for a more detailed explanation about the differentiation process.

On the other hand, we can write the *dynamic equations* of the system using the Euler-Lagrange equation [6] as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \tag{13}$$

It is possible rewrite this equation in the geometric algebra framework [10], defining the matrix $M(q) = M_v + M_I$, where M_v and M_I are defined as follow. The matrix M_I can be written as the product of two matrix δ and I if we define them as

$$M_I = \delta I = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} I_1 & 0 & \cdots & 0 \\ I_2 & I_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_n & I_n & \cdots & I_n \end{pmatrix}. \tag{14}$$

where I_n are the inertial value of each link. The matrix M_v also can be expressed as the product of two matrices $M_v = V^T m V$, where the matrices V and m are

$$m := \begin{pmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_n \end{pmatrix}, V := \begin{pmatrix} x'_{cm1} \cdot L'_1 & 0 & \cdots & 0 \\ x'_{cm2} \cdot L'_1 & x'_{cm2} \cdot L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x'_{cmn} \cdot L'_1 & x'_{cmn} \cdot L'_2 & \cdots & x'_{cmn} \cdot L'_n \end{pmatrix}. \tag{15}$$

where the values m_i are the mass of each link and x'_{cmi} and L'_i are the center of mass and the rotation axes of each link obtained by the direct kinematic defined in (10) and (12) respectively. Based in the properties of the matrices M and C, it can be written the matrix C without derivatives as $C = V^T m \dot{V}$, where the matrix $\dot{V}$ can be obtained following the next mathematical procedure. Defining $V = XL$ yields

$$V = \begin{pmatrix} x'_{cm1} & 0 & \cdots & 0 \\ 0 & x'_{cm2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x'_{cmn} \end{pmatrix} \begin{pmatrix} L'_1 & 0 & \cdots & 0 \\ L'_1 & L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L'_1 & L'_2 & \cdots & L'_n \end{pmatrix}, \tag{16}$$

then $\dot{V} = \dot{X}L + X\dot{L}$. With $\dot{X} = diag[\dot{x}'_{cm1}, \cdots, \dot{x}'_{cmn}]$ an diagonal matrix, where the $\dot{x}'_{cmi}$ is computed using the differential kinematic (11).

On the other hand

$$\dot{L}' = \begin{pmatrix} \dot{L}'_1 & 0 & \cdots & 0 \\ \dot{L}'_1 & \dot{L}'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \dot{L}'_1 & \dot{L}'_2 & \cdots & \dot{L}'_n \end{pmatrix}. \tag{17}$$

where $\dot{L}'_i$ is computed using (12) and can be expressed in a matrix form as:

$$\begin{pmatrix} \dot{L}'_1 \\ \dot{L}'_2 \\ \vdots \\ \dot{L}'_n \end{pmatrix} = \frac{1}{2} \left[\begin{pmatrix} L'_1 L'_1 & 0 & \cdots & 0 \\ L'_2 L'_1 & L'_2 L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L'_n L'_1 & L'_n L'_2 & \cdots & L'_n L'_n \end{pmatrix} - \begin{pmatrix} L'_1 L'_1 & 0 & \cdots & 0 \\ L'_1 L'_2 & L'_2 L'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L'_1 L'_n & L'_2 L'_n & \cdots & L'_n L'_n \end{pmatrix} \right] \dot{q}. \tag{18}$$

Finally the vector g is expressed as the product of the three matrices as $G = V^T ma$, where V have been defined and the vector $F = ma$ is a force component, where $m = diag[m_1, m_2, \cdots, m_n]$ is a $n \times n$ matrix of masses and $a = [ge_2, ge_2, \cdots]^T$ is a $n \times 1$ vector.

4 Dynamic Control

Now using (11) and (13) we will define the output tracking problem for the position x'_{p1} and the orientation L'_{p1} of the manipulator of the humanoid robot. A state-space model can be obtained using the next state variables: $x_1 = Pose\ of\ the\ manipulator$, $x_2 = q$ and $x_3 = \dot{q}$. Using the state variables, (11) and (13) the state-space model for the pose of the manipulator can be defined as:

$$\begin{aligned} \dot{x}_1 &= Jx_3 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -M^{-1}Cx_3 - M^{-1}G + M^{-1}\tau \end{aligned} \tag{19}$$

where J is the Jacobian matrix, $-M^{-1}Cx_3 - M^{-1}G$ will be expressed by $f(x_2, x_3)$ and $y = x_1$ is the output of the system.

Now, we define a reference signal $x_{ref1}(t)$ for the arm and we will define an error signal. Omitting the parenthesis of the reference, the tracking error is given by $\varepsilon_1 = x_1 - x_{ref1}$. Assuming that we know the derivative $\dot{x}_{ref1}$ and the second derivative $\ddot{x}_{ref1}$, the derivative of the error gives

$$\dot{\varepsilon}_1 = Jx_3 - \dot{x}_{ref1} \tag{20}$$

From (20) we will define a virtual control variable, which ensure that the variable ε_1 will tends to zero. This new variable is obtained as in block control like

$$x_{ref3} = J^{-1}\left(-k_1\varepsilon_1 + \dot{x}_{ref1}\right) \tag{21}$$

Now we will define an error variable for the second block as $\varepsilon_2 = x_3 - x_{ref3}$. Immediately, the error signal ε_2 is differentiated and gives

$$\dot{\varepsilon}_2 = f(x_2, x_3) + M^{-1}\tau - \dot{x}_{ref3} \tag{22}$$

the signal $\dot{x}_{ref3}$ its divided in two terms, one known part (δ_J)

$$\delta_J = J^{-1}(-k_1(Jx_3 - \dot{x}_{ref1}) + \ddot{x}_{ref1}) \tag{23}$$

and an unknown part Δ_J. On this way the derivative of ε_s can be rewritten as

$$\dot{\varepsilon}_2 = f(x_2, x_3) + M^{-1}\tau - \delta_J - \Delta_J \tag{24}$$

Now using (24), we are able to design a control law for τ, to tackle the pose tracking problem. For this end we will use a **Super-Twisting Algorithm** applied to the robotic manipulator. We will design our control law τ as

$$\tau = M(u_0 + u_1) \tag{25}$$

where the term u_0 is used to reject the nominal part of the Eq. (24) and the known signals, and is defined as $u_0 = -f(x_2, x_3) + \delta_J$, this term ensure the sliding mode occurrence from initial instance [7]. Now we use the sliding surface $s = \varepsilon_2$. The derivative of s using u and the control term u_0 yields $\dot{s} = u_1 - \Delta_J$.

To induce an sliding mode in $\dot{s}$ we design the second part u_1 of the control law using the super-twisting [2, 5] algorithm as

$$\begin{aligned} u_1 =& -\sigma N sign(s) + \mu \\ \dot{\mu} =& \Sigma sign(s) \end{aligned} \tag{26}$$

where $N = |s|^\rho$ and σ, ρ, Σ are design parameters. The stability proof and the system convergence, is demonstrated in [2], for more detail please refer to the cited reference.

5 Simulation Results

The initial values of the rotation axes, the values of x_i and the center of mass x_{cmi}, for $i = 1 \ldots 5$, in centimeters, used in the simulation are

$$\begin{aligned} &L_1 = e_{23} + e_\infty (x_1 \cdot e_{23}), && x_1 = 25e_2, && x_{cm1} = -8.91e_1 + 24.92e_2 \\ &L_2 = e_{31} + e_\infty (x_2 \cdot e_{31}), && x_2 = -15.3e_1 + 25e_2, && x_{cm2} = -15.23e_1 + 25e_2 + 0.1e_3 \\ &L_3 = e_{23} + e_\infty (x_3 \cdot e_{23}), && x_3 = -15.3e_1 + 25e_2, && x_{cm3} = -21.8e_1 + 25.1e_2 \\ &L_4 = e_{31} + e_\infty (x_4 \cdot e_{31}), && x_4 = -30.5e_1 + 25e_2, && x_{cm4} = -31.08e_1 + 25.23e_2 + 0.79e_3 \\ &L_5 = e_{32} + e_\infty (x_5 \cdot e_{32}), && x_5 = -30.5e_1 + 25e_2, && x_{cm5} = -41.89e_1 + 25.22e_2 \end{aligned} \tag{27}$$

The euclidean component of the references signals are given by $L_{ref} = [0, 1, -1]^T$ and $x_{ref} = [7cos\,(2t)\,, 24, 14 + 5sin\,(2t)]^T$, where x_{ref} and L_{ref} are the references for the position and orientation, respectively. The mass of the links was $m_1 = 0.45$ kg, $m_2 = 0.05$ kg, $m_3 = 0.34$ kg, $m_4 = 0.1$ kg, $m_5 = 0.11$ kg. The values of the controller was $k_1 = [7,\ 7,\ 5,\ 21,\ 22,\ 24]^T$, $\sigma = 80$, $\rho = 0.5$ and $\Sigma = 6$.

The Fig. 1a,b shows the tracking response of the arm, the Fig. 1c depicts the error signals for the pose of the arm. Finally the Fig. 2 shows a sequence of the humanoid torso using the angular values of the joints obtained with the proposed method in a virtual model developed in CLUCalc [3] and Matlab [8].

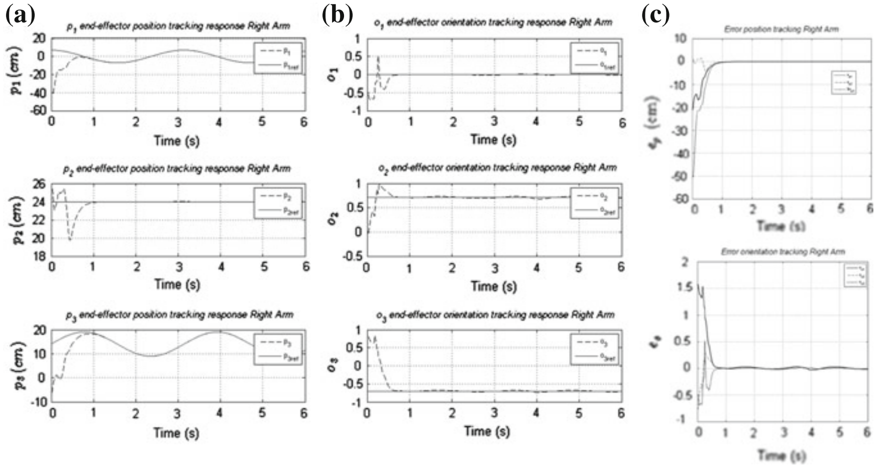


Fig. 1 **a** Position of the end-effector. **b** Orientation of the end-effector. **c** Error signal

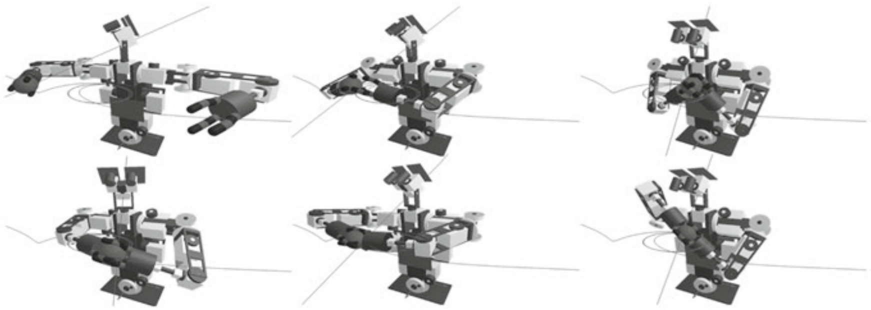


Fig. 2 Image sequence of the simulation results

6 Conclusions

The conformal geometric algebra was used to define, in a simple and compact manner, the kinematic and dynamic models of the arm of a humanoid. Furthermore, a super-twisting controller was designed in this framework which allows to define error variables between complex geometric entities. The proposed control scheme is robust against external disturbances, parameter variations and model uncertainties. Moreover, using a super twisting controller results in chattering-free control signals and finite time convergence of the closed loop system. This work pursues the inclusion of geometric restrictions expressed in CGA into the model of robotic systems, and in the controller design procedure.

References

1. Bayro-Corrochano, E.: Computing, Geometric: For Wavelet Transforms, Robot Vision, Learning, Control and Action. Springer Verlag, London (2010)
2. González, L., Loukianov, A., Bayro-Corrochano, E.: Integral nested super-twisting algorithm for robotic manipulators. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, pp. 3580–3585 (2010)
3. Mathematical Tool to Geometric Algebra http://www.clucalc.info/
4. Li, H., Hestenes, D.F., Rockwood, A.: Generalized homogeneous coordinates for computational geometry. In: Somer, G. (ed.) Geometric Computing with Clifford Algebras, pp. 27–52. Springer-Verlag, Heidelberg (2001)
5. Moreno, J.A., Osorio, M.: A lyapunov approach to second-order sliding mode controllers and observers. In: Proceedings of the 47th IEEE Conference on Decision and Control, Cancún, pp. 2856–2856 (2008)
6. Spong, W.M., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control. John Wiley and Sons, New York (2005)
7. Utkin, V.I., Guldner, J., Shi, J.: Sliding Mode Control in Electromechanical Systems. Taylor and Francis, London (1999)
8. www.mathworks.com/products/matlab/
9. Zamora, J., Bayro-Corrochano, E.: Kinematics and Differential Kinematics of Binocular Heads. In: International Conference of Robotics and Automation (ICRA'06), Orlando (2006)
10. Zamora, J., Bayro-Corrochano, E.: Parallel forward Dynamics: A geometric approach. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, pp. 2377–2382 (2010)

Singularity Locus for the Endpoint Map of Serial Manipulators with Revolute Joints

Ciprian S. Borcea and Ileana Streinu

Abstract We present a theoretical and algorithmic method for describing the singularity locus for the endpoint map of any serial manipulator with revolute joints. As a surface of revolution around the first joint, the singularity locus is determined by its intersection with a fixed plane through the first joint. The resulting plane curve is part of an algebraic curve called the singularity curve. Its degree can be computed from the specialized case of all pairs of consecutive joints coplanar, when the singularity curve is a union of circles, counted with multiplicity two. Knowledge of the degree and a simple iterative procedure for obtaining sample points on the singularity curve lead to the precise equation of the curve.

Keywords Serial manipulator · Revolute joints · Endpoint map singularity

1 Introduction

We consider a serial manipulator with an arbitrary number $n \geq 2$ of revolute joints. The end-effector or hand is abstracted to a single point T on the last link. The n joints, also called joint axes or hinges, are envisaged as full lines and labeled in order $A_1, ..., A_n$. For theoretical purposes, we assume full rotational capability around each joint and allow all geometrical configurations, without regard for possible self-collisions. Thus, the configuration space is parametrized by the n-dimensional torus $(S^1)^n$. The *endpoint map* $e : (S^1)^n \to R^3$ takes a configuration $\theta = (\theta_1, ..., \theta_n) \in (S^1)^n$ to the corresponding position of the endpoint $T(\theta) \in R^3$. When the differential $De(\theta)$ has

C. S. Borcea (✉)
Department of Mathematics, Rider University, Lawrenceville, USA
e-mail: borcea@rider.edu

I. Streinu
Department of Computer Science, Smith College, Northampton, USA
e-mail: istreinu@smith.edu

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_30,

rank strictly less than three, we have a *singular configuration*. The locus of T for all singular configurations is called the *singularity locus* for the endpoint map.

It is fairly well known that the singularity locus is a *surface of revolution*, with the first joint as symmetry axis. When sectioned with a plane passing through this axis, the singularity locus yields a *plane curve* which is part of an *algebraic curve*, called here the *singularity curve*. In this paper we describe a complete and rigorous procedure for obtaining the *equation of the singularity curve*.

The singularity locus is of fundamental importance not only for path planning and avoidance of singular configurations, but also for positional *workspace determination*. The *workspace boundary* is necessarily included in the singularity locus [4, 20].

Previous attempts for describing either the singularity locus or the workspace boundary have usually addressed cases with a very small number of joints often relying on numerical procedures of uncertain accuracy [1–3, 14–16, 18, 19, 21]. The general recursion proposed in [11] seems difficult to work out for larger n and has been explicitly used only in a few instances [12, 17].

The main elements of novelty of the solution presented here reside in the method itself, centered on obtaining an *explicit degree formula* for the singularity curve, the recognition of the algebraic and geometric advantage of using for this purpose the specialized case of manipulators with any two consecutive joints coplanar and the possibility of producing the necessary amount of sample points in the general case, based on the geometric characterization of singular configurations.

More precisely, our determination of the singularity curve is based on the following principles: (i) the *degree* of the curve does not change when the manipulator is continuously altered until any two consecutive joints become coplanar, (ii) for a manipulator with any two consecutive joints coplanar, the singularity curve is made of circles, counted with multiplicity two, (iii) recursion on n yields the general *degree formula*, (iv) sample points on the singularity curve can be produced in arbitrary numbers, (v) with known degree and sufficiently many sample points computed, the equation of the curve is obtained from solving a linear system.

This summary leaves aside some technical details. It will be seen in due course that, for full mathematical rigor, one has to work over the algebraically closed field of *complex numbers*. The 'continuity principle' used in (i), while intuitively persuasive, is actually justified through a more elaborate argument [13]. However, when assuming a certain background in algebraic geometry, these aspects take lesser roles. Thus, the key elements of our approach rely on (a) the fact that manipulators with coplanar pairs of consecutive joints allow the computation of the degree formula, in combination with (b) the possibility of producing sample points based on a simple geometrical characterization of singular configurations.

Part (a) follows from our work on extremal reaches and workspace determination for manipulators with coplanar pairs of consecutive joints, also called *panel-and-hinge chains*: [6, 7]. This special class of manipulators is adequate for computing the degree δ_n of the singularity curve since the latter decomposes into irreducible components of degree two and multiplicity two. The degree formula is obtained in Sect. 3 from a linear recurrence relation which gives:

$$\delta_n = \frac{1}{\sqrt{3}}[(1+\sqrt{3})^{n+1} - (1-\sqrt{3})^{n+1}] - 2^{n+1}$$

with $\delta_2 = 4,\ \delta_3 = 16,\ \delta_4 = 56,\ \delta_5 = 176,\ \delta_6 = 528\ \ldots.$

The *exponential complexity* of the workspace boundary described in [8] is aptly reflected in this degree formula.

The geometric characterization of singular configurations used in (b) is the following **singularity criterion**: *the nR manipulator is in a singular configuration* θ i.e. $rank(De(\theta)) \leq 2$ *when there is a line through T projectively incident with all joint axes* [5, 9, 10]. A line of this type will be called a T-transversal. This criterion and its sibling for the end-to-end (squared) distance from a marked start point S on the first link are already implicated in (a), leading to the important notions of *fold point* and *pivoting* [7, 8]. For (b) the criterion serves in the following way.

We choose an arbitrary line through the terminus point T and the last hinge A_n and then find the two solutions given by the intersection of this line with the hyperboloid generated by rotating A_{n-1} around A_n. Then, with any one of these two solutions in place, we look for the two solutions given by intersecting the line with the hyperboloid generated by rotating A_{n-2} around A_{n-1} and so on. This procedure produces 2^{n-1} singular configurations i.e. 2^{n-1} points of the singularity curve. With sufficiently many sample points determined in this manner all coefficients of the curve can be determined (up to proportionality) by solving the resulting homogeneous linear system. Actually, $\binom{\delta_n+2}{2} - 1$ points imposing independent conditions suffice.

2 Fold Points and Pivoting

In this section we review the argument showing that a manipulator with all pairs of consecutive joints coplanar has a singularity curve made of irreducible components of degree two, which have to be counted with multiplicity two. The key notions are those of *fold point* and *pivoting* at a first fold point introduced in [6–8].

Let $p_{i,i+1} = A_i \cap A_{i+1}$ denote the intersection of a pair of consecutive joint axes. The plane containing this pair of joint axes is called a panel. Our nR manipulator can thereby be conceived as a *panel-and-hinge chain* since one panel is joint to the next by their common joint axis or hinge. The fist panel is taken as a fixed plane through A_1 and the last panel is the plane given by A_n and T. As recalled in the introduction, a singular configuration must allow a T-transversal for all hinges. As long as the T-transversal avoids intersection points $p_{i,i+1}$, consecutive panels must remain in one and the same plane. Thus, singular configurations are either *flat*, with all panels in the same plane, or *non-flat*, with at least one point $p_{f,f+1}$ on the T-transversal such that the three consecutive panels incident to $p_{f,f+1}$ are not coplanar. Such a point $p_{f,f+1}$ is called a **fold point** of the singular configuration.

For the notion of *pivoting* it is useful to review first the case of a 2R manipulator with incident joints, illustrated in Fig. 1. Note that, by definition, all configurations are singular, since $n = 2$ and the rank of $De(\theta)$ cannot be 3. Thus, as a set, the

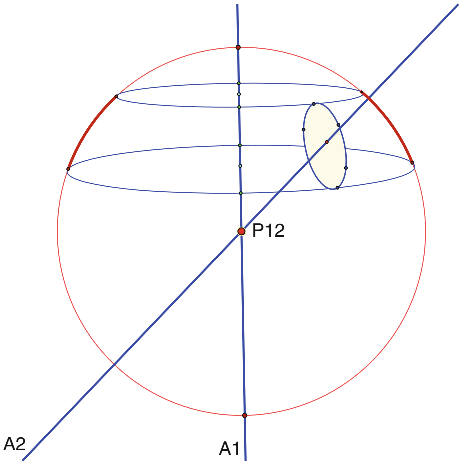


Fig. 1 The locus of T for a manipulator with two intersecting hinges is a *ring-shaped spherical region* (covered twice). In the reference plane through A_1 given by the red grand *circle* it traces two arcs along this circle

singularity locus is given by all possible positions of T. These positions cover a ring-shaped region of a sphere centered at $p_{1,2} = A_1 \cap A_2$. Algebraically, one should remain alert to several aspects: the actual singularity locus is only part of an algebraic surface, but this 'inconvenience' is removed when reformulating the problem over the field of *complex numbers*. Then, 'complex configurations' would map to the complex quadric whose real points are seen as the sphere, and in fact cover it twice. This double covering can be seen on the real scenario over the ring-shaped region and becomes intuitive also when imagining the manipulator with incident joint axes as a limit of manipulators with two skew joint axes. In the latter case, the locus of T is a torus and when the joint axes intersect, the torus degenerates to a 'doubled' spherical region.

In short, all matters algebraic become simpler over the *complex field C* and this must be the adopted setting in general for properly speaking about irreducible components and degree for the singularity curve.

Returning now to arbitrary n and a given configuration, we define **pivoting** at $p_{k,k+1}$ to mean 'locking' all joints except A_k and A_{k+1} and using only these two degrees of freedom. Thus, the nR manipulator becomes a $2R$ manipulator. When we have a singular configuration with first fold point at $p_{f,f+1}$, we take as reference plane the common plane of the first f panels and pivoting at $p_{f,f+1}$, together with the singularity criterion, show that all configurations with T in this reference plane are singular configurations for the nR manipulator. When we start with a *flat configuration*, pivoting at any $p_{k,k+1}$ immediately provides a degree two irreducible component of the complex singularity curve passing through the corresponding T.

Thus, over C, any point of the singularity curve belongs to some degree two irreducible component. In other words: *the complex singularity curve of a manipulator with all pairs of consecutive joints coplanar decomposes into irreducible components of degree two (which must be counted with multiplicity two).*

Before engaging the degree computation, we recall the analogous case of the end-to-end squared distance function for a panel-and-hinge chain [6]. The first panel is fixed and has a marked point S (start). The end-to-end function $f : (S^1)^n \rightarrow R$ gives the squared distance $f(\theta)$ between S and the endpoint $T(\theta) = e(\theta)$. When the differential $df(\theta) = 0$, we have a *critical configuration*. The *geometrical criterion for critical configurations* says that the end-to-end line ST must intersect (projectively) all hinges. Thus, all critical configurations of an end-to-end function are singular configurations for the endpoint map.

3 Counting Circles and the Degree Formula

We consider **critical configurations** for a panel and hinge chain with k hinges and marked S (start) and T (terminus) points. The first panel is fixed. We denote by c_k the number of (real or complex) configurations with line ST intersecting all hinges (i.e. critical configurations).

There are 2^k flat configurations.

For non-flat configurations **we look at the first fold point** $p_{f,f+1}$.

From this point to T we have a chain with $k - (f + 1)$ hinges (and $k - f$ panels). There are $c_{k-(f+1)}$ critical configurations (with line $p_{f,f+1}T$ intersecting all $k - (f + 1)$ hinges.

The first f panels have 2^{f-1} flat configurations.

By *pivoting* at $p_{f,f+1}$ there are **two** (real or complex) alignments of $Sp_{f,f+1}$ with $p_{f,f+1}T$, hence:

$$c_k = 2^k + 2\sum_{f=1}^{k-1} 2^{f-1} c_{k-(f+1)} \tag{1}$$

with $c_0 = 1$. If we put $c_{-1} = 1$, we have:

$$c_k = \sum_{f=1}^{k} 2^f c_{k-(f+1)}, \quad c_{-1} = c_0 = 1 \tag{2}$$

This gives the linear recurrence relation:

$$c_{k+1} = 2(c_k + c_{k-1}), \quad c_0 = c_1 = 1, \; k \geq 1 \tag{3}$$

For the **number of (complex) circles** ν_n traced in the plane of the first panel by the (complex) singularity locus of the endpoint map, we look again at the first fold point $p_{k,k+1}$. (Note that the circle is traced by pivoting at this point.) We have:

$$\nu_n = \sum_{k=1}^{n-1} 2^{k-1} c_{n-(k+1)}, \quad n \geq 2 \tag{4}$$

It follows that:

$$\nu_{n+1} = 2\nu_n + c_{n-1}, \quad n \geq 2 \tag{5}$$

which yields by (3) the linear recurrence relation:

$$\nu_{n+3} = 2(2\nu_{n+2} - \nu_{n+1} - 2\nu_n), \ \ n \geq 2 \tag{6}$$

with $\nu_2 = 1, \ \nu_3 = 4, \ \nu_4 = 14$.

Remark: When passing from counting circles to degrees, the circles have degree two and have to be counted twice since each 'sphere' is covered doubly. Two skew hinges give a torus (for the real picture) i.e. a degree four surface.

From (2) and (4) we obtain:

$$c_n = 2\nu_n + 2^n, \quad n \geq 2 \tag{7}$$

hence

$$\nu_n = 2\nu_{n-1} + c_{n-2} = c_{n-1} + c_{n-2} - 2^{n-1} = \frac{1}{2}c_n - 2^{n-1} \tag{8}$$

The **degree** δ_n of the curve made of ν_n double circles is $4\nu_n$ and we have therefore:

$$\delta_n = 2c_n - 2^{n+1} = \frac{1}{\sqrt{3}}[(1+\sqrt{3})^{n+1} - (1-\sqrt{3})^{n+1}] - 2^{n+1} \tag{9}$$

where the last part of the formula follows from solving the linear recurrence (3). The first few terms of the degree sequence are:

$$\delta_2 = 4, \ \ \delta_3 = 16, \ \ \delta_4 = 56, \ \ \delta_5 = 176, \ \ \delta_6 = 528 \ \ldots$$

4 Generating Sample Points

With the degree δ_n determined from nR manipulators with all pairs of consecutive joints coplanar, we return to the general case when consecutive joints would be skew lines in space. In order to determine the equation of the singularity curve, we need $\binom{\delta_n+2}{2} - 1$ points of the curve which impose independent conditions. As already described in the introduction, we may produce any number of sample points on the curve since we may execute the procedure with arbitrary positions of the T-transversal in the last link. In fact, the needed number of independent sample

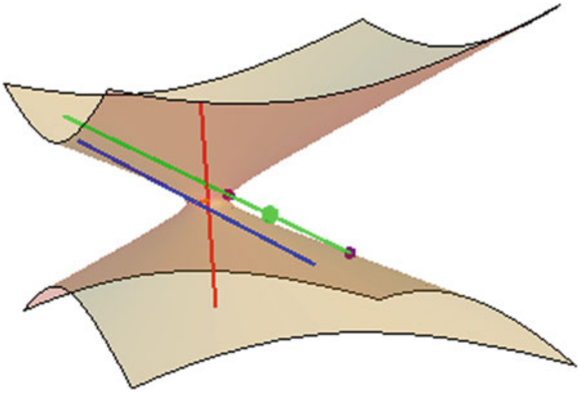


Fig. 2 The main iterative step for generating sample points. The *green line* is the intended T-transversal, with T marked larger in *green*

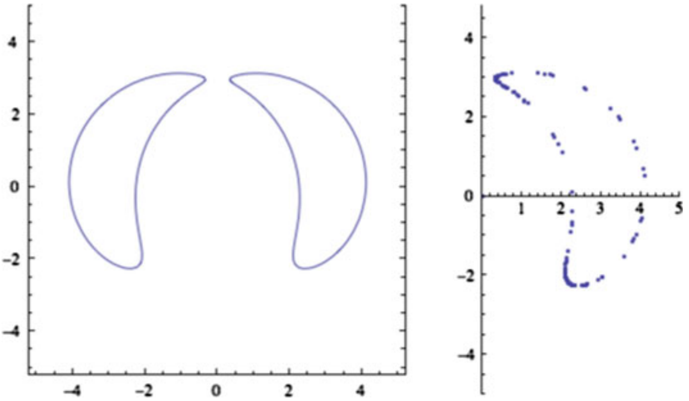


Fig. 3 The singularity curve for a 2R manipulator, obtained after computing all coefficients of the degree four equation from sample points illustrated nearby

points is roughly halved by virtue of the reflection symmetry of the singularity curve in the A_1 point axis.

We illustrate in Fig. 2 the main step of the iterative procedure which starts with a chosen line through T and some point on A_n, our ‘designated’ T-transversal.

After successively placing A_{n-1}, ..., A_{n-k} in contact with this line, we have the situation depicted in the figure, with the designated T-transversal in green, the joint axis A_{n-k} in red and A_{n-k-1} in blue. When all the remaining part of the manipulator is rotated around A_{n-k}, the blue line sweeps the shown hyperboloid and the two specific rotations which position A_{n-k-1} in contact with the designated T-transversal are determined from simple quadratic conditions. Thus, using one or the other rotation we have one more joint on the designated T-transversal. At the final step, A_1 is positioned in contact with the green line, making it a genuine T-transversal.

Figure 3 shows sample points for a 2R manipulator next to a full plot of the (real points of the) singularity *curve*. The degree is $\delta_2 = 4$ in this case and one needs $\binom{6}{2} - 1 = 14$ independent points for the determination (up to a proportionality factor) of the 15 coefficients implicated in the general equation of a plane curve of degree four.

5 Conclusions

We addressed and solved the fundamental problem of obtaining the equation of the singularity curve of a serial manipulator with an arbitrary number n of revolute joints. The key elements of our solution are the *degree formula* (9) , derived from the specialized case of manipulators with any two consecutive joints coplanar and the general possibility of obtaining *sample points* on the curve by an iterative procedure. The full singularity surface for the endpoint map of the manipulator is generated by rotating the singularity curve around the fixed first joint axis. Computational designs for effective implementations of this solution will be detailed in separate publications.

References

1. Abdel-Malek, K., Adkins, F., Yeh, H.J., Haug, E.: On the determination of boundaries to manipulator workspaces. Robot. Comput. Integr. Manuf. **13**(1), 63–72 (1997)
2. Abdel-Malek, K., Yeh, H.J., Khairallah, N.: Workspace, void, and volume determination of the general 5dof manipulator. Mech. Struct. Mach. TEXTIT27, pp. 91–117 (1999)
3. Abdel-Malek, K., Yeh, H.J., Othman, S.: Interior and exterior boundaries to the workspace of mechanical manipulators. Robot. Comput. Integr. Manufact. **16**, 365–376 (2000)
4. Angeles, J.: Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms. Mechanical Engineering Series. Springer Verlag, New York (2007)
5. Borcea, C.S., Streinu, I.: Singularities of hinge structures. Tech. rep. (2008). ArXiv:0812.1373
6. Borcea, C.S., Streinu, I.: Extremal configurations of manipulators with revolute joints. In: Dai, J.S., Zoppi, M., Kong X. (eds.) Reconfigurable Mechanisms and Robots,Proceedings ASME/IFToMM International Conference (ReMAR'09) (2009)
7. Borcea, C.S., Streinu, I.: Exact workspace boundary by extremal reaches. In: Proceedings 27th Symposium on Computational Geometry (SoCG'11), pp. 481–490 (2011)
8. Borcea, C.S., Streinu, I.: Positional workspace boundary for serial manipulators with revolute joints. In: Lenarcic, J., Husty, M.L. (eds.) Latest Advances in Robot Kinematics (ARK'12), pp. 325–332. Springer Verlag, Netherlands (2012)
9. Burdick, J.W.: A classification of 3R regional manipulator singularities and geometries. Mech. Mach. Theory **30**(1), 71–89 (1995)
10. Burdick, J.W.: A recursive method for finding revolute-jointed manipulator singularities. Trans.ASME. J. Mech. Des. **117**, 55–63 (1995)
11. Ceccarelli, M.: A formulation for the workspace boundary of general n-revolute manipulators. Mech. Mach. Theory **31**(5), 637–646 (1996)
12. Ceccarelli, M., Lanni, C.: A multi-objective optimum design of general 3R manipulators for prescribed workspace limits. Mech. Mach. Theory **39**(2), 119–132 (2004)
13. Fulton, W.: Intersection Theory, 2nd edn. Springer Verlag, Berlin (1998)

14. Haug, E., Luh, C.M., Adkins, F., Wang, J.Y.: Numerical algorithms for mapping boundaries of manipulator workspaces. J. Mech. Des. **118**(2), 228–234 (1996)
15. Kohli, D., Spanos, J.: Workspace analysis of mechanical manipulators using polynomial discriminants. J. Mech. Transm. Autom. Des. **107**(2), 209–215 (1985)
16. Kumar, A., Waldron, K.J.: The workspaces of a mechanical manipulator. J. Mech. Des. **103**(3), 665–672 (1981)
17. Ottaviano, E., Husty, M.L., Ceccarelli, M.: Identification of the workspace boundary of a general 3R manipulator. J. Mech. Des. **128**(1), 236–242 (2006)
18. Rastegar, J., Fardanesh, B.: Manipulation workspace analysis using the Monte Carlo method. Mech. Mach. Theory **25**(2), 233–239 (1990)
19. Selfridge, R.G.: The reachable workarea of a manipulator. Mech. Mach. Theory **18**(2), 131–137 (1983)
20. Siciliano, B., Khatib, O. (eds.): Handbook of Robotics. Springer Verlag, Berlin (2008)
21. Zein, M., Wenger, P., Chablat, D.: An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications. Mech. Mach. Theory **41**(8), 971–986 (2006)

Robust Design Synthesis of Spherical Parallel Manipulator for Dexterous Medical Task

Abdelbadia Chaker, Abdelfattah Mlika, Med Amine Laribi, Lotfi Romdhane and Said Zeghloul

Abstract This paper deals with the design synthsesis of a spherical parallel manipulator (SPM) for a dexterous surgery task. A considerable effect of manufacturing errors on the workspace and particularly on the dexterity of the mechanism is noted. Thus, the use of nominal values of the design vector generated by deterministic optimization may be erroneous. The effect of these errors on the mechanism workspace and dexterity is then studied and a Robust approach combining genetic algorithms (GA) and Monte Carlo Simulation (MCS) is presented to lead to an SPM with a low-sensitive dexterity to manufacturing errors. The results are finally discussed through an example.

Keywords Spherical parallel mechanism · Robust design synthesis · Dexterous task · Manufacturing errors · Dexterity · Monte Carlo simulation

A. Chaker, A. Mlika · L. Romdhane (✉)
Laboratoire de Mcanique, Université de Sousse-Ecole Nationale d'Ingénieurs de Sousse, Sousse, Tunisia
e-mail: lotfi.romdhane@gmail.com

A. Mlika
e-mail: abdelfattah.mlika@gmail.com

A. Chaker
e-mail: abdelbadi3@gmail.com

M. A. Laribi · S. Zeghloul
Institut PPRIME, UPR 3346, CNRS, Université de Poitiers—ENSMA, Poitiers, France
e-mail: med.amine.laribi@univ-poitiers.fr

S. Zeghloul
e-mail: said.zeghloul@univ-poitiers.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_31,

1 Introduction

Several studies dealt with the 3-RRR spherical parallel manipulators (SPM) [2, 4, 5]. The platform of the mechanism is moving over a spherical surface around a fixed center of motion. In the bibliography, several studies covered a wide range of characteristics such as workspace [3], kinematic analysis [5] and design parameters (DP) optimization [2]. This characteristic is suitable for our application of minimally invasive surgery since the surgeon is operating with tools through small incisions in the body patient. Indeed, this work is part of a project to design and fabricate a teleoperation system for minimally invasive surgery. An experimental study of this technique based on motion capture was held in previous work [4] to characterize the task workspace. Figure 1 (left) illustrates the expert surgeon operating with tools on a training station (Pelvis Trainer). Figure 1 (right) represents the workspace of the used tools (a clamp and a needle holder) identified by motion capture. Each tool operates in a conical space with a maximum half vertex angle of 26°. However, despite being an over constrained mechanism; few studies were interested in the effect of manufacturing errors on the SPM. A1-Widyan et al. [1] evaluated through a stochastic method the translational displacement of each cylindrical joint in the 3-RCC architecture. In a previous work [4], we were interested in finding the optimal dimensions of a SPM with a given workspace. The effect of the manufacturing errors (ME) on the platform position and dexterity, was also studied. In this paper, the effect of these errors on the dexterity of the mechanism is reviewed and a robust design strategy is proposed. The dexterity of the manipulator resulting from the deterministic optimization showed a high sensitivity to ME, which led us to adopt a new strategy combining genetic algorithms (GA) and a Monte Carlo Simulation (MCS) [7] to synthesize a spherical manipulator with a low-sensitive dexterity to ME. The results of this approach are then presented and the performances of the obtained manipulator are discussed.

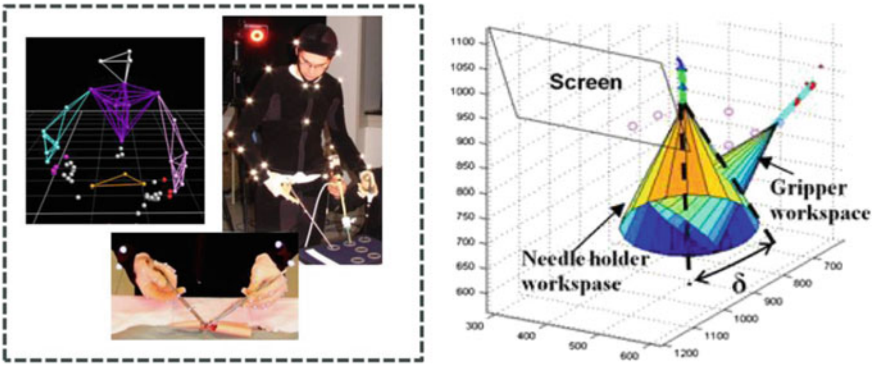


Fig. 1 Experimental study of surgical task. *Left*: reconstruction of the surgeon operating. *Right*: experimental workspace

2 Kinematics of the SPM

Figure 2 (left) presents the 3-RRR architecture of the proposed SPM. Three identical legs A, B and C relate the base to the platform. Each leg of the SPM is made out of two links and three revolute joints. The three actuated revolute joints with the base have orthogonal axes $\mathbf{Z}_{1k}$ *(k = A, B and C)*. All the axes of the joints are intersecting in a single point, the center of motion of the platform.

Figure 2 (right) shows the geometric parameters of one leg. The angles α, β, γ are, respectively, between the first two joint axes, the second and the third one, and between the third axis and the platform axis.

The three legs of the SPM are identical and the actuated joint axes are located along the base frame axes **X**, **Y**and **Z**, respectively. The workspace of the platform is given by the intersection of the three workspaces of three legs, which are considered each as a spherical serial kinematic chain.

The motion of the SPM is generated by only revolute joints. The kinematics of the mechanism can be described by the following relation:

$$\mathbf{Z}_{2k} \cdot \mathbf{Z}_{3k} = cos(\beta) \tag{1}$$

where $\mathbf{Z}_{2k}$ and $\mathbf{Z}_{3k}$ are respectively the axes of the second and the third joint of each leg and detailed as:

$$\mathbf{Z}_{2k} = \mathrm{Rot}(\mathbf{Z}_{1k}, \theta_{1k}).\mathrm{Rot}(\mathbf{X}_{2k}, \alpha).\mathbf{Z}_{1k} \tag{2}$$

$$\mathbf{Z}_{3k} = \mathrm{Rot}(\mathbf{Z}_{1k}, \psi).\mathrm{Rot}(\mathbf{X}, \theta).\mathrm{Rot}(\mathbf{Z}_E, \varphi).\mathrm{Rot}^{-1}(\mathbf{X}_{3k}, -\gamma).\mathbf{Z}_{1k} \tag{3}$$

The $\mathbf{Z}_E$ platform axis is described by the three ZXZ-Euler angles, ψ, θ and φ. θ_{1k}, θ_{2k} and θ_{3k} are, respectively, the revolute joint variables of the leg k ($k = A, B$ and C). The axes $\mathbf{X}_{2k}$ and $\mathbf{X}_{3k}$ are given, respectively, by $\mathbf{X}_{2k} = \mathbf{Z}_{1k} \times \mathbf{Z}_{2k}$ and $\mathbf{X}_{3k} = \mathbf{Z}_{2k} \times \mathbf{Z}_{3k}$.

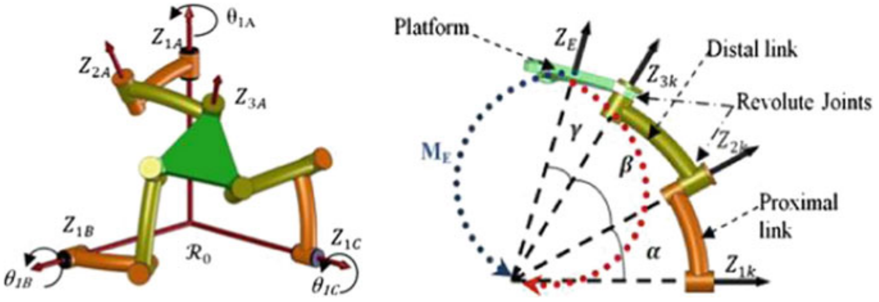


Fig. 2 Architecture and parameters of the SPM. *Left*: SPM architecture. *Right*: one leg parameters

The set of three equations resulting from applying Eq. (1) for the three legs of the mechanism combines the joints parameters $[\theta_{1A}, \theta_{1B}, \theta_{1C}]$ and the platform orientation parameters $[\psi, \theta, \phi]$. Thus, it can be used to detail and solve both the forward and the inverse displacement problems. The inverse kinematics model can be described by the following three equations:

$$A_i cos(\theta_{1k}) + B_i sin(\theta_{1k}) + C_i = \mathbf{0} \tag{4}$$

With $k \in (A, B, C)$ and $i \in (1, 2, 3)$

3 Deterministic Optimization

Chaker et al. [4] presented a detailed approach for the synthesis of an SPM for a surgery application based on GA. Two criteria were minimized. The first one is the workspace that still contains a prescribed workspace. The second one is the the dexterity. The design vector contains the geometric parameters of the mechanism $\mathbf{V} = [\alpha, \beta, \gamma]$ and the optimization problem is expressed as follows:

$$\begin{cases} \text{Minimize } F_1 + F_2 + F_3 \\ F_2 = \sum_{i=0}^{n} \sum_{i=0}^{3} \frac{C_i^2(p_j)}{(A_i^2(p_j)+B_i^2(p_j))} \\ F_3 = \sum_{i=0}^{n} \sum_{i=0}^{3} K(P_j) \\ \text{Subject to } \frac{C_i^2(p_j)}{(A_i^2(p_j)+B_i^2(p_j))} \leq 1 \end{cases} \tag{5}$$

n the number of chosen 'points' on the cone P_j. F_1 is a penalty function that handles the constraints. $F_1 = 0$ means that all the points, defining the desired volume, are contained within the workspace of the SPM. K is the condition number of the Jacobean matrix, which represents the dexterity of the SPM. F_2 represents the sum of the distances of all the points of the manipulator workspace to the cone boundary and F_3 is the sum of the dexterity of the manipulator over its workspace. The desired workspace is represented by a set of orientations P_j within a cone (Fig. 3 (left)). Each point P_j has to be included in the manipulator workspace.

The design vector resulting from this procedure is $\mathbf{V} = [39.4°, 34.1°, 18.2°]$. Figure 3 presents, respectively, the workspace (center) and the dexterity distribution in (ψ, θ) frame (right) of the resulting manipulator. The self-rotation is fixed to 18° and a security angle of 4° was adopted to guarantee that the prescribed workspace can be reached by the end-effector of the mechanism.

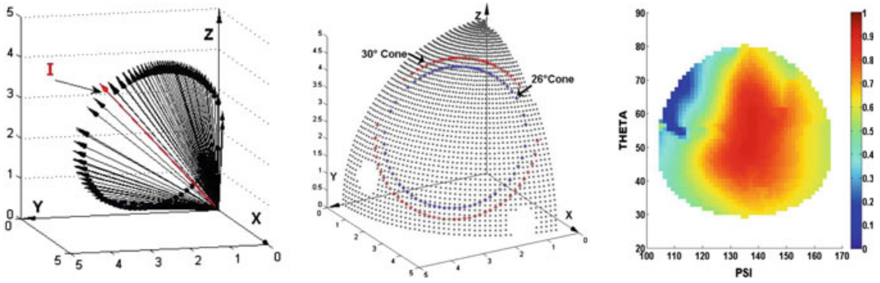


Fig. 3 Architecture and parameters of the SPM. *Left*: prescribed Workspace Model. *Center*: workspace of resulting mechanism. *Right*: dexterity of the resulting mechanism

4 Effects of Manufacturing Errors on the Workspace and the Dexterity

In order to investigate the results of the deterministic optimization, we studied the behavior of the SPM subject to manufacturing errors. For this purpose, we generated a normal distribution with a mean value equal to the nominal design vector resulting from the deterministic optimization. A standard deviation of 5% is imposed. Figure 4a represents the variation of the workspace of the mechanism due to the possible manufacturing errors applied to the design parameters. Three cases are shown: the workspace corresponding to nominal values of the DP, the smallest workspace and the biggest one generated when applying errors on the DP corresponding to the minimal and maximal values of the DP, respectively. We note that the smallest workspace still contains the prescribed workspace. The choice of a security angle is then justified. However, considering the dexterity, the performances of the manipulator are highly variable and it drops to very low values. Figure 4 represents the dexterity of the manipulators with minimal (b), nominal (c) and maximal (d) DP, respectively.

5 Robust Synthesis: Combined GA-MCS

As mentioned before, dexterity showed a high sensitivity toward manufacturing errors. Thus, adopting the results of the deterministic optimization can lead to erroneous results. This issue is treated by the proposed approach for the synthesis of the SPM that combines GA and MCS method [6]. The idea is to take advantages of the GA particularly the wide range of research intervals for DP and multjobiective problem resolution. On the same time, a local evaluation of the behavior of every manipulator toward uncertainty and manufacturing errors is led by the MCS. The optimization problem is then formulated as follows:

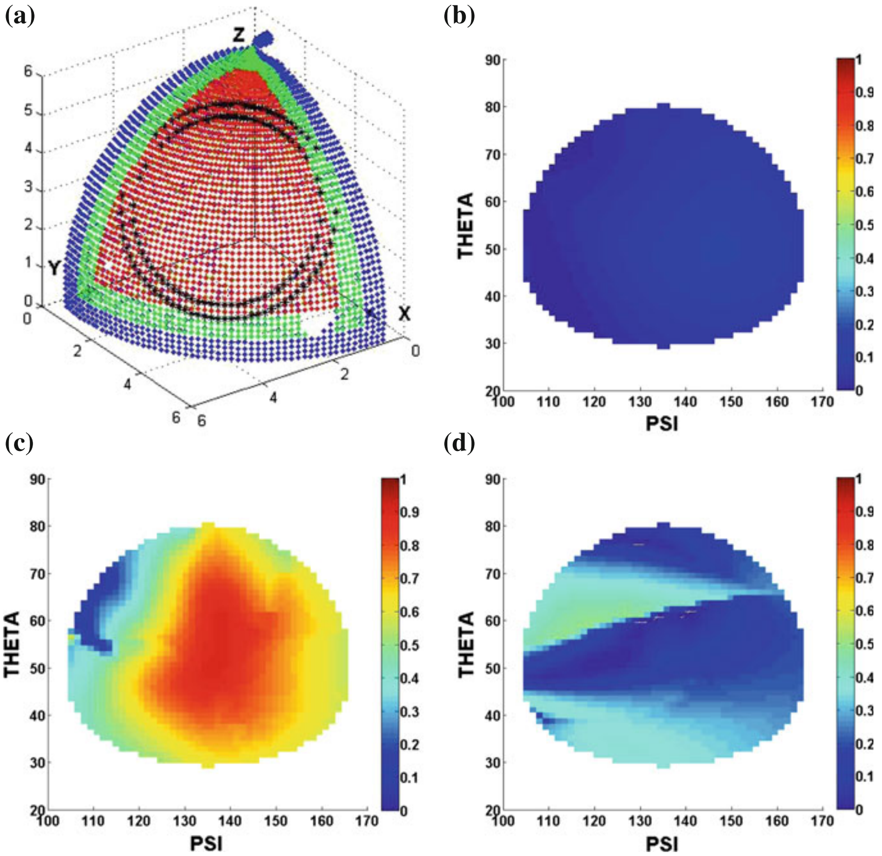


Fig. 4 Manipulator sensitivity to manufacturing error. **a** Workspace variation. **b** Dexterity of manipulator with minimal design parameters. **c** Dexterity of manipulator with nominal design parameters. **d** Dexterity of manipulator with maximal design parameters

$$\begin{cases} \text{Minimize } F_2 \\ \text{Minimize } \bar{F_3} \\ \text{Minimize } \sigma_{F_3} \\ \text{Subject to } \frac{C_i^2(p_j)}{(A_i^2(p_j)+B_i^2(p_j))} \leq 1 \end{cases} \tag{6}$$

Where $\bar{F_3}$ and σ_{F_3} are respectively the mean and the standard deviation of the objective function of the dexterity.

Figure 5 depicts the flow chart of the algorithm. For each iteration of non-dominated solutions, during the evaluation stage, the GA sends a generation of solutions to the MCS [7]. The MCS generates a normal distribution and performs N random simulations for every solution. The first objective function value F_2 is

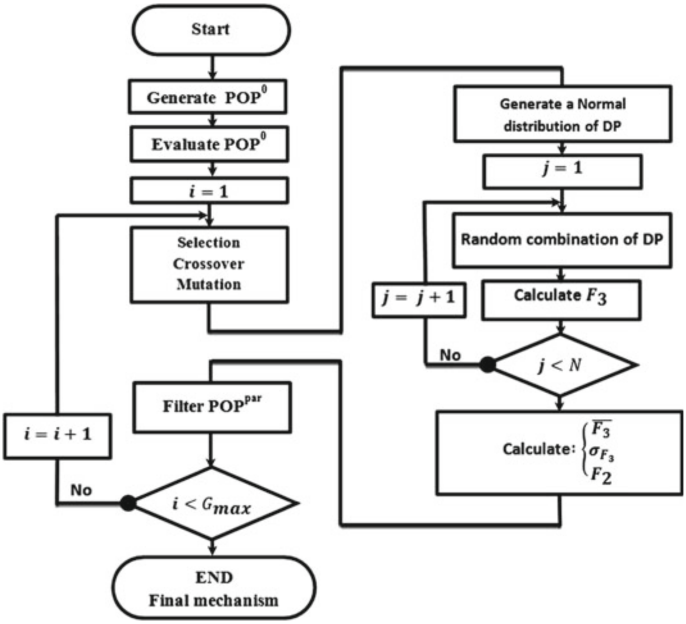


Fig. 5 The GA-MCS flow chart

calculated only for the nominal values of the DP. The output of the MCS is then the value of F_2, the mean value $\bar{F_3}$ and the standard deviation σ_{F_3}. The nondominated solutions undergo the selection, crossover, mutation and reinsertion operations. The MCS number of simulations is $N = 10^3$.

POP^0 is the initial population of the initial design vector (population) to be evaluated and POP^{par} is the Paretian (non-dominated) population.

6 Results and Discussion

The implementation of the Robust algorithm for the synthesis of an SPM led to the following design vector: $\mathbf{V} = [35.2°, 32.9°, 22°]$. Figure 6 shows that the workspace presented in Fig. 6a satisfies the experimental workspace of 26° and the values of the dexterity are kept relatively high with a minimum value of 0.5.

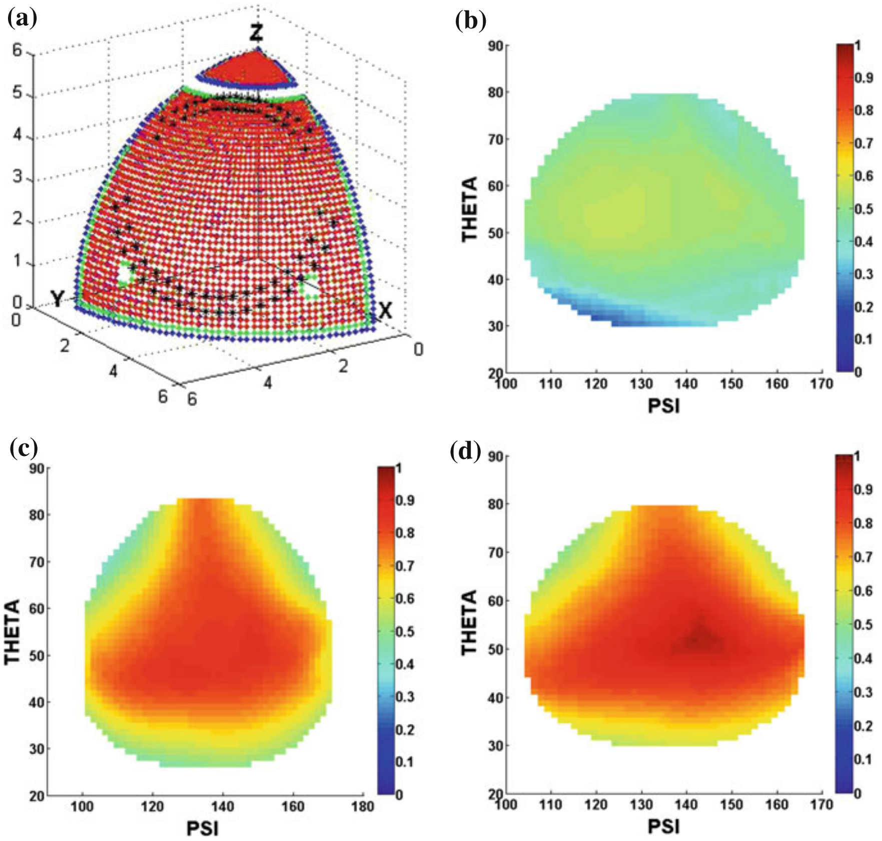


Fig. 6 Performances of mechanism generated by the Robust Synthesis. **a** Workspace variation. **b** Dexterity of manipulator with minimal design parameters. **c** Dexterity of manipulator with nominal design parameters. **d** Dexterity of manipulator with maximal design parameters

7 Conclusion

A multiobjective robust Synthesis strategy for the design of the SPM for dexterous surgery application was presented in this paper. The kinematics of the mechanism was revisited and the effects of manufacturing errors on its workspace and dexterity were studied. This study was based on the results of deterministic optimization elaborated in a previous work. We noted that the dexterity is very sensitive to these errors. Thus, an approach, based on a combined GA-MCS, is proposed to take into account the manufacturing errors in the optimization of the SPM. We are led then to minimize the mean value and the standard deviation of the dexterity. An SPM with low-sensitive dexterity to manufacturing errors is finally presented as a result of this robust algorithm.

References

1. Al-Widyan, K., Ma, X.Q., Angeles, J.: The robust design of parallel spherical robots. Mech. Mach. Theory **46**(3), 335–343 (2011)
2. Bai, S.: Optimum design of spherical parallel manipulators for a prescribed workspace. Mech. Mach. Theory **45**(2), 200–211 (2010)
3. Bulca, F., Angeles, J., Zsombor-Murray, P.: On the workspace determination of spherical serial and platform mechanisms. Mech. Mach. Theory **34**(3), 497–512 (1999)
4. Chaker, A., Mlika, A., Laribi, M.A., Romdhane, L., Zeghloul, S.: Synthesis of spherical parallel manipulator for dexterous medical task. Front. Mech. Eng. **7**(2), 150–162 (2012)
5. Gosselin, C.M., Lavoie, E.: On the kinematic design of spherical three-degree-of-freedom parallel manipulators. Int. J. Robot. Res. **12**(4), 394–402 (1993)
6. Nejlaoui, M., Houidi, A., Affi, Z., Romdhane, L.: Multiobjective robust design optimization of rail vehicle moving in short radius curved tracks based on the safety and comfort criteria. Simul. Model. Pract. Theory **30**, 21–34 (2013)
7. Shinozuka, M.: Monte carlo solution of structural dynamics. Comput. Struct. **2**(5–6), 855–874 (1972)

Self Dual Topology of Parallel Mechanisms with Configurable Platforms

Patrice Lambert and Just L. Herder

Abstract This paper presents first an analysis of the topology of mechanisms via Graph Theory and Screw Theory and next the principle of dual mechanisms in terms of their mobility and overconstraints. Using dual graphs, the graph representations of the mechanisms that are dual to hybrid and Delta mechanisms are revealed. The concept of parallel mechanisms with configurable platforms (PMCPs) is introduced and it is shown that the graph reduction of PMCPs always results in a wheel graph, which has the interesting property of being self-dual. In case of self dual topology, it is then possible to directly convert any method developed for their mobilty analysis into an overconstraint analysis method and vice versa. This self dual topology property can also be exploited to create new PMCPs and is an important aspect in the future development of a type synthesis method that includes PMCPs.

Keywords Parallel mechanisms · Configurable platform · Topology · Mobility · Overconstraints · Duality

1 Introduction

A pure parallel mechanism is formed by two rigid links, called the base and the end-effector, connected in parallel by independent serial chains, called legs. The concept behind parallel mechanisms with configurable platforms (PMCPs) is that the rigid (non-configurable) end-effector is replaced by a closed-loop chain (the configurable platform), see for example Fig. 1. Some of the links of this closed-loop chain are attached to the legs so its configuration can be fully controlled from the motors located near the base. The use of a closed-loop chain instead of a rigid

P. Lambert (✉) · J. L. Herder
Delft University of Technology, Delft, The Netherlands
e-mail: P.Lambert@TUDelft.nl

J. L. Herder
e-mail: J.L.Herder@TUDelft.nl

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_32,

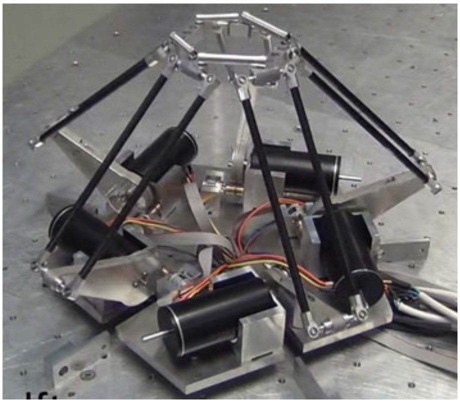


Fig. 1 The 5 DOF PentaG robot with configurable platform. The configurable platform can translate in 3 directions and has 2 internal DOF, providing additional rotation and grasping capabilities. This architecture was used to develop both a haptic device and a pick-and-place robot

platform creates multiple end-effectors that can be used for example to add grasping capabilities. In this case the robot can combine motions and grasping into a structure that provides an inherent high structural stiffness, since no motors are needed at the end-effector location to provide the grasping and all motors can be grounded on the base. Very few PMCPs have been presented in the literature [5, 7]. Mohamed and Gosselin [6] proposed a first generalization of the concept of both planar and spatial PMCPs. In their article, all the mechanisms they proposed were not overconstrained and the case of overconstrained PMCPs was not addressed. PMCPs are ignored in type synthesis methods of mechanisms such as in [2, 3]. In order to describe a topological relation between mobility and overconstraints, Davies [1] introduced the idea of dual mechanisms with the use of dual graphs. In his article, he presents the dual mechanism of a four-bar linkage and a self-dual truss structure.

This article presents a topology analysis of PMCPs via Graph Theory and Screw Theory and shows that unlike pure parallel mechanisms, the topology of PMCPs is represented by a wheel graph, which has the remarkable property of being self-dual. The self dual topology of PMCPs is then exploited to extend the mobility analysis method presented in [4] directly into an overconstraint analysis method and is also used to create new overconstrained PMCPs.

2 Duality Between Parallel Mechanisms and Single Closed-Loops

Closing the mechanical loops of a parallel mechanism introduces dependencies between the joint velocities of the chains and can also produce internal stresses due to overconstraints of the assembly. The mobility and constraints of a serial chain are here expressed via Screw Theory. For each n-twist system S representing the mobility of a serial chain, it is possible to obtain a reciprocal (6–n)-wrench system $S^{\perp}$ representing its constraints. For a parallel mechanism with k legs, in which the mobility of each leg i is represented by the twist system T_i, the twist system T_M

representing the instantaneous mobility between the base and the end-effector is given by the intersection of the mobility of all legs:

$$T_M = \bigcap_{i=1}^{k} T_i \tag{1}$$

Internal stresses in parallel mechanisms are due to redundant constraints between the base and the end-effector. For a parallel mechanism with $i = 1 \dots k$ legs, in which the constraint system of each leg is represented by the wrench system W_i, a set of $(k-1)$ wrench systems W_{Rj} representing the independent overconstraints is obtained by calculating the intersection of the wrench system W_j of leg j with the summation of the constraints of the legs previously connected to the end-effector.

$$W_{Rj} = W_j \bigcap \left(\sum_{i=1}^{j} W_i \right) \quad \text{for } j = 2 \dots k \tag{2}$$

For any mechanism, the solution for the mobility must respect the condition that the sum of the finite twists (describing the joint velocities) of all joints that belong to the same closed loop is zero. In a dual way, the solution for the overconstraints must respect the condition that the sum of the finite wrenches (describing the internal stresses) of all chains that are connected to the same rigid link is zero. The particular way to obtain those solutions depends on the topology of the mechanism, i.e. the particular arrangement of the closed loops that form the mechanism. Graph Theory is the study of pairwise relations between objects and since each joint or each serial chain of joints in a mechanism connects strictly two links, it can be used to analyse the topology of mechanisms. Any mechanism can be represented with a corresponding graph in which a vertex represents a rigid link and an edge represents a joint or a serial chain of joints. The duality between the mobility and overconstraint conditions, which both require that the sum of a set of finite screws is zero, will be investigated more deeply with the use of dual graphs.

The dual graph B of an original graph A is a graph in which each vertex of B corresponds to a loop of A and vice-versa. Edges that are incident to a vertex in the original graph are in the same direction as the corresponding loop of the dual graph and vice versa. If the twist systems of the dual edges are defined as the reciprocal of the twist systems of the original edges, the mechanism corresponding to the dual graph is dual to the original mechanism. The dual mechanism of a pure parallel mechanism is always a single closed-loop and vice versa. Figure 2 shows an example of the graph representation of a pure parallel mechanism with four legs. In [1], Davies used the concept of dual graphs to show that the dual mechanism of a planar four-bar linkage is a parallel mechanism with four parallel SU legs. The principle of dual mechanisms is here extended to some mechanisms that have a topology that is not purely parallel. Following this principle, the dual graphs of a hybrid and a Delta mechanism are shown for the first time in Fig. 3.

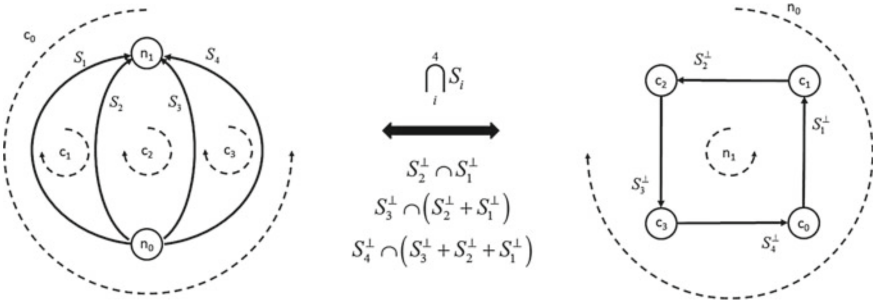

c₀
S_1
n₁
S_4
S_2
S_3
c₁
c₂
c₃
n₀
$\bigcap_{i}^{4} S_i$
$S_2^{\perp} \cap S_1^{\perp}$
$S_3^{\perp} \cap (S_2^{\perp} + S_1^{\perp})$
$S_4^{\perp} \cap (S_3^{\perp} + S_2^{\perp} + S_1^{\perp})$
n₀
$S_2^{\perp}$
c₂
c₁
$S_1^{\perp}$
n₁
$S_3^{\perp}$
c₃
c₀
$S_4^{\perp}$

Fig. 2 Graph representation of a parallel mechanism with four legs and its dual single closed-loop. The screw system over the *big arrow* represents the mobility of the parallel mechanism and the overconstraints of the single closed-loop. The screw systems below the arrow represent the overconstraints of the parallel mechanism and the mobility of the single closed-loop

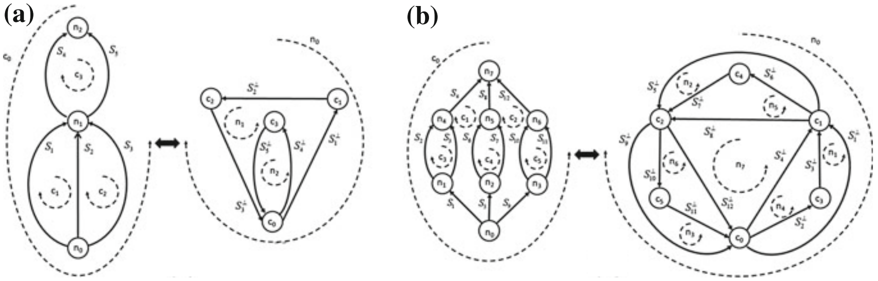


Fig. 3 Graph representation of **a** a hybrid mechanism and **b** a Delta mechanism and their corresponding dual mechanisms. In both cases, the mobility of the dual mechanism corresponds to the overconstraints of the original mechanism and vice versa

3 Self Dual Topology of Parallel Mechanisms with Configurable Platforms

The novel concept behind parallel mechanisms with congurable platforms is that the rigid (non-configurable) end-effector is replaced by a closed-loop chain (the configurable platform). The use of a closed-loop chain instead of a rigid platform allows robots based on such an architecture to have multiple end-effectors on the platform while all the motor are located on the base. The graph reduction of a PMCP is always a wheel graph, in which the center of the wheel represents the base, the spokes of the wheel represent the legs and the rim represents the configurable platform. Wheel graphs have the interesting property of being self-dual. Figure 4 shows the wheel graph of a PMCP with 4 legs and its dual graph. It follows that each PMCP has a dual PMCP for which the mobility of the dual PMCP corresponds to the overconstraints of the original PMCP and vice versa. In particular, the method presented in [4] to calculate the distribution of the mobility of overconstrained PMCPs can be directly used to calculate the distribution of the overconstraints using the dual graph as input.

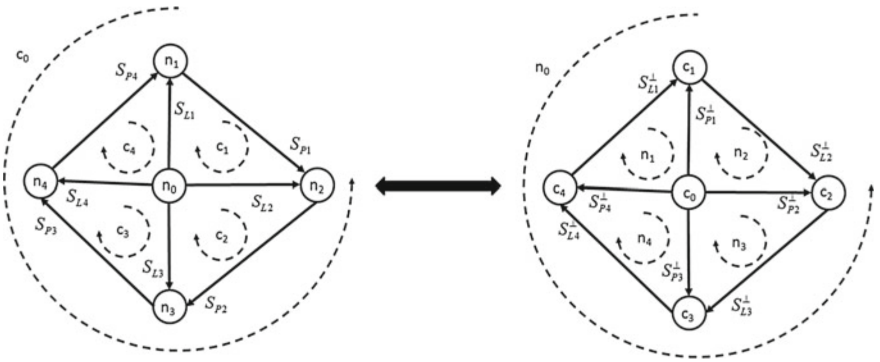


Fig. 4 Self dual wheel graphs of a parallel mechanism with configurable platform and 4 legs

It should be noted that in the dual graph representation of Fig. 4, the edges that originaly represented the platform are now representing the legs and vice versa, and that the reciprocal screw system is used to describe the mobility of each dual edge.

4 Example

This section shows an example of two PMCPs that are dual to each other. The original mechanism M1 is shown in Fig. 5a. It has four legs L1, L2, L3 and L4 and each leg consists of three parallel joints. The configurable platform is a single closed loop with 12 parallel joints. Two adjacent end-effector links n_i are connected by platform limbs P1, P2, P3 or P4. In order to create the dual mechanism, we first need to express the twist system of each leg and each platform limb. Using a reference frame located in the center of the base, the twist systems are

$$T_{L1} = T_{L3} = \left\{ \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \end{bmatrix}, \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \right\} \text{ and } T_{L2} = T_{L4} = \left\{ \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \end{bmatrix}, \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix} \right\} \quad (3)$$

$$T_{Pi} = \left\{ \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix}, \begin{bmatrix} \mathbf{z} \\ \mathbf{0} \end{bmatrix} \right\} \text{ for } i = 1 \ldots 4 \quad (4)$$

Unlike pure parallel mechanisms, PMCPs have multiple end-effectors, represented by the leg attach points n_i. Since the graph representation of such a mechanism is not a series-parallel graph, their global mobility can not be calculated using the traditional rules of addition of twist systems for joints in series and intersection of twist systems for legs in parallel. In [4], a method was proposed to calculate the mobility of mechanisms that have a non-series-parallel graph that is particularly suitable for PMCPs. The solution given by the method forms a matrix in which row i represents the mobility of an end-effector n_i relatively to the base and each column

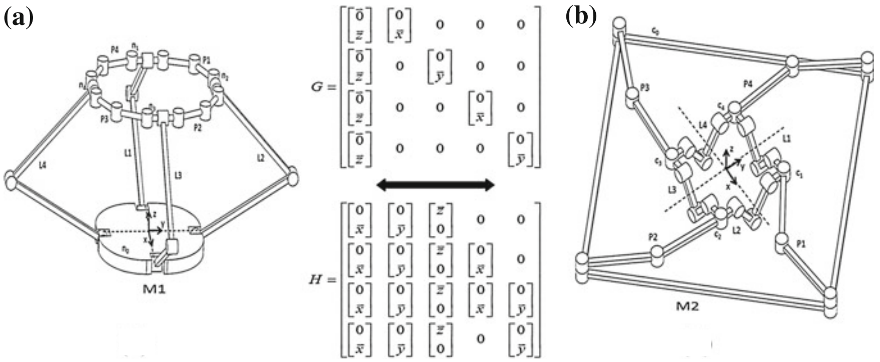


Fig. 5 Dual mobility and overconstraints of two PMCPs. G_L represents the mobility of M1 and the overconstraints of M2. H_P represents the mobility of M2 and the overconstraints of M1. Each line of a matrix of mobility uses twist screws to represent the mobility of a leg end-effector n_i or c_i relatively to the base. Each line of a matrix of overconstraints uses wrench screws to represent the internal stresses in a platform limb between two end-effectors

represents a global mobility of the mechanism. The final leg mobility matrix G of mechanism M1 obtained from this method is

$$G = \begin{bmatrix} \begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix} & \begin{bmatrix}\mathbf{0}\\ \mathbf{x}\end{bmatrix} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix} & \mathbf{0} & \begin{bmatrix}\mathbf{0}\\ \mathbf{y}\end{bmatrix} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix} & \mathbf{0} & \mathbf{0} & \begin{bmatrix}\mathbf{0}\\ \mathbf{x}\end{bmatrix} & \mathbf{0} \\ \begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \begin{bmatrix}\mathbf{0}\\ \mathbf{y}\end{bmatrix} \end{bmatrix} \quad (5)$$

The fact that the matrix G has five columns indicates that the mechanism has 5 DOF. The first column represents a motion where all legs move in the vertical direction. The remaining columns show that each leg can also move independently in the horizontal direction of the plane of their parallel joints. Since all PMCPs have a self dual topology, the overconstraints of mechanism M1 can be obtained by appling the same mobility analysis method to the dual PMCP. The original and dual graphs are shown in Fig. 4. The dual graph gives the information about the way the dual chains must be connected in the dual mechanism. The reciprocal screw system of each serial chain must be calculated in order to obtain the dual serial chains. In this particular mechanism, each leg and each platform chain is formed by three parallel joints. Those chains are known to be self reciprocal and therefore the screw system representing their constraints is the same as the screw system representing their mobility.

$$S_{Li}^{\perp} = S_{Li} \quad \text{and} \quad S_{Pi}^{\perp} = S_{Pi} \qquad \text{for} \quad i = 1 \ldots 4 \quad (6)$$

We can now assemble the dual mechanism by connecting the dual chains according to the edges of the dual graph. The resulting mechanism M2 is shown in Fig. 5b. The mobility of M2 is calculated using the same method that was used to calculate the mobility for M1. The resulting leg matrix of mobility H obtained is

$$H = \begin{bmatrix} \begin{matrix} \begin{bmatrix}\mathbf{0}\\\mathbf{x}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{x}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{x}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{x}\end{bmatrix} \end{matrix} & \begin{matrix} \begin{bmatrix}\mathbf{0}\\\mathbf{y}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{y}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{y}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{y}\end{bmatrix} \end{matrix} & \begin{matrix} \begin{bmatrix}\mathbf{z}\\\mathbf{0}\end{bmatrix}\\ \begin{bmatrix}\mathbf{z}\\\mathbf{0}\end{bmatrix}\\ \begin{bmatrix}\mathbf{z}\\\mathbf{0}\end{bmatrix}\\ \begin{bmatrix}\mathbf{z}\\\mathbf{0}\end{bmatrix} \end{matrix} & \begin{matrix} \begin{bmatrix}\mathbf{0}\\\mathbf{x}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{x}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\-\mathbf{x}\\\mathbf{0}\\-\mathbf{x}\end{bmatrix} \end{matrix} & \begin{matrix} \begin{bmatrix}\mathbf{0}\\\mathbf{y}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\-\mathbf{y}\\\mathbf{0}\\-\mathbf{y}\end{bmatrix}\\ \begin{bmatrix}\mathbf{0}\\\mathbf{y}\end{bmatrix} \end{matrix} \end{bmatrix} \tag{7}$$

This mechanism also has 5 DOF. The first three columns show that the configurable platform can move in the XY plane and rotate around the Z direction. The fourth column shows that legs P1 and P2 can move relatively to leg P3 and P4 in the X direction and the fifth column shows that legs P1 and P4 can move in the Y direction relatively to legs P2 and P3. Since these two mechanisms are dual, the mobility of M1 corresponds to the overconstraints of M2 and vice-versa. The matrix H is the platform matrix of overconstraints of M1. In a platform matrix of overconstraints, each line represents the internal forces transmited by a platform chain and each column represents an independent overconstraint.

The overconstraints of both mechanisms are interpreted as follows: The first three columns of H represent the 3 planar overconstraints of the configurable platform of M1. Column 4 represents internal stresses that occur in the platform limbs when leg L1 and leg L3 of M1 are not perfectly oriented about the axis X. Column 5 represents the internal stresses that occur in the platform limbs when leg L2 and leg L4 are not perfectly oriented around the axis Y. The overconstraints of mechanism M2 are represented by the platform matrix of overconstraints G. The first column represents internal stresses around the Z axis that occur in all the platform limbs if the sum of the angle between the platform limbs is not 360° d. The second column of G shows that the platform limb L1 of M2 must be perfectly oriented around the X axis in order to be connected between leg P4 and leg P1. Columns 3, 4 and 5 of G shows similar overconstraint conditions for platform limbs L2, L3 and L4 of mechanism M2.

5 Conclusion

This paper presented first the principle of dual mechanisms in terms of mobility and overconstraints for some mechanisms that have a topology that is not purely parallel. The graph and screw representations of the mechanisms that are dual to hybrid and Delta mechanisms were revealed. It was explained that the graph reduction of

PMCPs always result in a wheel graph, which has the interesting property of being self-dual. Unlike most other classes of mechanism, it is possible to directly apply any method developed for the mobilty analysis of PMCPs to their overconstraint analysis and vice versa, thanks to their self-dual topology. This self-dual topology property can also be used to generate new PMCPs using the dual mechanism of original PMCPs. PMCPs are promising solutions for robot architecture since they can operate multiple end-effectors while all motors are located at the base. Applications include interaction with humans or environment, in haptic devices and grasping applications, respectively. They are currently ignored by type synthesis methods. A better understanding of the fundamentals of their mobility and overconstraints will help robot designers to consider them as a valid option in their choice of a robot architecture and is an important aspect in the future development of a broader type synthesis method that includes mechanisms with this topology.

Acknowledgments This research is supported by the Dutch Technology Foundation STW (project number 12158).

References

1. Davies, T.H.: Simple examples of dual coupling network. In: 12th IFToMM World Congress, Loughborough University, UK (2007)
2. Gogu, G.: Structural Synthesis of Parallel Robots: Part 1: Methodology, Solid Mechanics and its Applications. Springer, Dordrecht (2007)
3. Kong, X., Gosselin, C.M.: Type synthesis of parallel mechanisms. Springer tracts in. Advanced Robotics, vol. 33, Heidelberg (2007)
4. Lambert, P., Herder, J.L.: Mobility analysis of non series-parallel mechanisms. New Trends in Mechanism and Machine Science, Mechanisms and Machine Science, vol. 7, pp. 63–71. Springer (2012)
5. Lambert, P., Langen, H., Munnig Schmidt, R.: A novel 5 DOF fully parallel robot combining 3T1R motion and grasping. In: 34th Annual IDETC, Parts A and B, vol. 2, pp. 1123–1130. ASME (2010)
6. Mohamed, M., Gosselin, C.: Design and analysis of kinematically redundant parallel manipulators with configurable platforms. IEEE Trans. Rob. **21**(3), 277–287 (2005)
7. Nabat, V., de la O Rodriguez, M., Company, O., Krut, S., Pierrot, F.: Par4: very high speed parallel robot for pick-and-place In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 553–558 (2005)

Four-Dimensional Persistent Screw Systems of the General Type

Marco Carricato

Abstract When a mechanism moves, the twist system S of the end-effector generally varies. In significant special cases, S is a subalgebra of the Lie algebra of the special Euclidean group, and it remains constant. In more general cases, S remains invariant up to a proper isometry, thus preserving its *class*. A mechanism of this kind is said to generate a *persistent screw system* (PSS) of the end-effector. PSSs play an important role in mobility analysis and mechanism design. This paper presents the serial generators of 4-dimensional PSSs with a constant class of the *general* type.

Keywords Screw theory · Mechanism synthesis · Mobility analysis

1 Introduction

Screw systems are the subspaces of the Lie algebra $se(3)$ of the Euclidean group $SE(3)$. Two screw systems are equivalent if one may be moved onto the other by a rigid-body displacement [4]. This equivalence relation divides the space of screw systems into infinitely many *classes*. The latter may be grouped into a finite number of general or special *types* [7, Chap. 12], within which the constituent classes are identified by the values of a small number of parameters. Screw systems belonging to the same class have the same *dimension*, *type* and *shape*, thus differing only in their *pose* in space.

For a mechanism in a configuration $\boldsymbol{\Theta}$, the possible instantaneous motions of the end-effector are given by a screw system $S(\boldsymbol{\Theta}) \subset se(3)$. In general, when $\boldsymbol{\Theta}$ changes, so does $S(\boldsymbol{\Theta})$. An important special case occurs when the mechanism constrains the body to trace out (at least locally) a *subgroup* of $SE(3)$. Then, for any nonsingular $\boldsymbol{\Theta}$, $S(\boldsymbol{\Theta}) = A$, where A is the *algebra* of the subgroup. The mechanism is said to generate

M. Carricato (✉)
Department of Industrial Engineering and Interdepartment Center for Health Sciences
and Technologies, University of Bologna, Bologna, Italy
e-mail: marco.carricato@unibo.it

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_33,

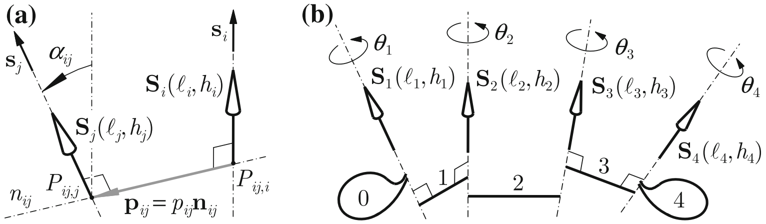

(a)
$\mathbf{s}_j$
α_{ij}
$\mathbf{s}_i$
$\mathbf{S}_i(\ell_i, h_i)$
$\mathbf{S}_j(\ell_j, h_j)$
n_{ij}
$P_{ij,j}$
$P_{ij,i}$
$\mathbf{p}_{ij} = p_{ij}\mathbf{n}_{ij}$
(b)
θ_1
θ_2
θ_3
θ_4
$\mathbf{S}_1(\ell_1, h_1)$
$\mathbf{S}_2(\ell_2, h_2)$
$\mathbf{S}_3(\ell_3, h_3)$
$\mathbf{S}_4(\ell_4, h_4)$
0
1
2
3
4

Fig. 1 Relative pose between two screws (**a**); a chain of four 1-dof lower pairs (**b**)

an *invariant screw system* (ISS) of the end-effector [4–7]. Herein, a more general case is considered, namely a mechanism where $S(\boldsymbol{\Theta})$, although not necessarily constant, has a *constant class* . In other terms, $S(\boldsymbol{\Theta})$ retains its shape while it changes its pose, in effect moving like a rigid body in space. If that is so, and $\dim S(\boldsymbol{\Theta}) = n$, the mechanism is said to generate an n-dimensional *persistent screw system* of the end-effector, or briefly an n-PSS. PSSs were presented by Carricato and Rico Martínez, who showed that PSS generators may be obtained by serially composing generators of ISSs [1–3]. The exhaustive classification of PSS generators is in progress. This paper complements the results in [1], where the generators of 4-PSSs with constant classes of *special* types were described. Here, the serial generators of 4-PSSs with constant classes of the *general* type are presented.

In the following, the locutions 'nG system' and 'nR system', with R being a Roman numeral, denote n-dimensional screw systems, respectively, of the *general* type and of the Rth *special* type, according to [7]. A normalized screw representing a relative twist between two bodies is designated by $\mathbf{S}$. The axis (when it exists), the pitch and the direction of $\mathbf{S}$ are denoted by ℓ, h and $\mathbf{s}$, respectively. When it is useful, ℓ and h accompany $\mathbf{S}$ within parentheses, i.e. $\mathbf{S}(\ell, h)$ (if $h = \infty$, ℓ is replaced by $\mathbf{s}$). Given two screws $\mathbf{S}_i(\ell_i, h_i)$ and $\mathbf{S}_j(\ell_j, h_j)$ (Fig. 1a), n_{ij} is the common normal between ℓ_i and ℓ_j; $P_{ij,i}$ and $P_{ij,j}$ are the feet of n_{ij} on ℓ_i and ℓ_j; $\mathbf{n}_{ij}$ is a unit vector parallel to n_{ij} and directed from $P_{ij,i}$ to $P_{ij,j}$; p_{ij} and α_{ij} are the shortest distance and the relative angle between ℓ_i and ℓ_j, with α_{ij} being measured about $\mathbf{n}_{ij}$ in the interval $(-\pi/2, \pi/2]$; finally, $\mathbf{p}_{ij} = P_{ij,j} - P_{ij,i} = p_{ij}\mathbf{n}_{ij}$.

2 Generators of 4-Dimensional PSSs of the General Type

Any subgroup of $SE(3)$ may be generated (at least locally) by a serial chain composed by 1-dof lower pairs. Hence, any 4-PSS generator emerging by the serial composition of ISSs may be considered 'equivalent' to a serial linkage $\mathcal{S}$ composed by (at least) four 1-dof joints (Fig. 1b). Since joint motions affect neither the pitches nor the relative pose of adjacent joints, the geometry of the chain is completely defined by the joint screws at an arbitrarily-chosen nonsingular reference configuration, i.e. $\mathbf{S}_i = \mathbf{S}_i(\boldsymbol{\Theta})|_{\boldsymbol{\Theta}=\mathbf{0}} = \mathbf{S}_i(\mathbf{0})$, $i = 1\ldots4$. Accordingly, $\mathcal{S}$ may be identified with the array $\langle \mathbf{S}_1, \ldots, \mathbf{S}_4 \rangle$. Link 0 of $\mathcal{S}$ is the predecessor of $\mathbf{S}_1$; link i, with $i = 1\ldots3$, is the body laid between $\mathbf{S}_i$ and $\mathbf{S}_{i+1}$; and link 4 is the successor of $\mathbf{S}_4$.

After a displacement $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_4)$, the ith joint screw, $i > 1$, is moved to $\mathbf{S}_i(\boldsymbol{\Theta}) = \prod_{j=1}^{i-1} \mathbf{D}_j\left(\theta_j\right)\mathbf{S}_i$, where $\mathbf{D}_j$ is the adjoint action of the jth joint displacement. Generally, in the new configuration $S(\boldsymbol{\Theta}) = \text{span}\,[\mathbf{S}_1(\boldsymbol{\Theta}), \ldots, \mathbf{S}_4(\boldsymbol{\Theta})] \neq S(\mathbf{0})$. $\mathcal{S}$ generates a PSS if, for every nonsingular $\boldsymbol{\Theta}$, $S(\boldsymbol{\Theta}) = \mathbf{G}(\boldsymbol{\Theta})S(\mathbf{0})$, where $\mathbf{G}(\boldsymbol{\Theta})$ is the adjoint action of a *proper isometry*. A simpler formulation of this condition emerges by observing that joint motions θ_1 and θ_4 cannot alter the shape of $S(\boldsymbol{\Theta})$, since they do not affect the relative pose of $\mathcal{S}$'s joint screws. Accordingly, when studying the persistent properties of $S(\boldsymbol{\Theta})$, θ_1 and θ_4 may be kept constant, e.g. $\theta_1 = \theta_4 = 0$, and link 2 may be conveniently chosen as the reference frame [2]. As a consequence, $\mathcal{S}$ generates a 4-PSS if and only if, for every nonsingular pair (θ_2, θ_3), there is an adjoint action $\mathbf{G}(\theta_2, \theta_3)$ such that $S(\theta_2, \theta_3) = \mathbf{G}(\theta_2, \theta_3)S(0, 0)$, where $S(\theta_2, \theta_3) = \text{span}\left[\mathbf{D}_2^{-1}(\theta_2)\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{D}_3(\theta_3)\mathbf{S}_4\right]$.

$S(\boldsymbol{\Theta})$ has a constant class of the *general* type if the principal screws $\mathbf{S}_{r1}(\ell_{p1}, -h_{p1})$ and $\mathbf{S}_{r2}(\ell_{p2}, -h_{p2})$ of the cylindroid $S^{\perp}$ reciprocal to S have constant *finite* pitches, and $h_{p1} \neq h_{p2}$ [7] (Fig. 2). If that is so, the principal screw $\mathbf{S}_{pi}(\ell_{pi}, h_{pi})$ of S, with $i = 1, 2$, is collinear with $\mathbf{S}_{ri}$ and has pitch h_{pi}, whereas the principal screws $\mathbf{S}_{p3}$ and $\mathbf{S}_{p4}$ span a cylindrical ISS $\mathcal{C}(\ell_{p3})$ along the nodal line of $S^{\perp}$ (the latter is the line perpendicular to ℓ_{p1} and ℓ_{p2}, passing through their intersection point O). $\mathcal{C}(\ell_{p3})$ and the ∞-pitch screw therein are, respectively, the only available ISS with dimension greater than 1 and the only ∞-pitch screw in S.

A generator $\mathcal{S} = \langle \mathbf{S}_1, \ldots, \mathbf{S}_4 \rangle$ of a $4G$-PSS may be constructed by composing the ISSs available in S. Since $\mathcal{C}(\ell_{p3})$ has dimension 2 and S has dimension 4, no less than three ISSs need to be composed. A *ternary* generator is constructed by composing $\mathcal{C}(\ell_{p3})$ with two 1-dimensional ISSs, i.e. with two single screws. A *quaternary* generator emerges by composing four distinct screws in S. Since the $4G$ system comprises a single ∞-pitch screw, $\mathcal{S}$ cannot include: more than one ∞-pitch screw; more than a pair of adjacent parallel finite-pitch screws; a pair of adjacent parallel finite-pitch screws plus an ∞-pitch screw. Otherwise, configurations in which ∞-pitch screws along more than one direction would appear.

$\mathcal{S}$ may be synthesized by expanding the 3-PSS generators disclosed in [3]. Since the only 3-systems contained in a $4G$ system are the $3G$, $3I$, $3III$, $3VII$ and $3VIII$ systems, the only 3-PSSs that may appear within a $4G$ system are the $3I$- and the $3VIII$-PSS (cf. [3]). For this reason, these two will be the 'building blocks' of the generators described hereafter.

2.1 The Ternary Generator

Let $\mathcal{B} = \langle \mathbf{S}_2(\ell_2, h_2), \mathbf{S}_3(\ell_3, h_3), \mathbf{S}_4(\ell_4, h_4) \rangle$ and let $\mathcal{B}$ form a $3I$-PSS B. For the properties of the $3I$ systems, all screws of B have finite pitch and, if $\mathbf{S}^B_{pi}(\ell^B_{pi}, h^B_{pi})$, $i = 1 \ldots 3$, is the ith principal screw of B, then $h^B_{p1} = h^B_{p2}$ [7]. Since B is persistent, the following conditions also apply [3]:

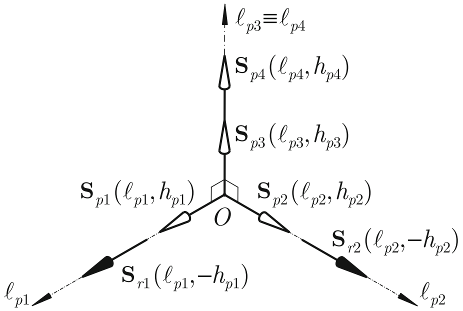

$\ell_{p3}\equiv\ell_{p4}$
$\mathbf{S}_{p4}(\ell_{p4},h_{p4})$
$\mathbf{S}_{p3}(\ell_{p3},h_{p3})$
$\mathbf{S}_{p1}(\ell_{p1},h_{p1})$
$\mathbf{S}_{p2}(\ell_{p2},h_{p2})$
O
$\mathbf{S}_{r2}(\ell_{p2},-h_{p2})$
$\mathbf{S}_{r1}(\ell_{p1},-h_{p1})$
ℓ_{p1}
ℓ_{p2}

Fig. 2 Principal screws of a 4-system of the general type: $h_{p1} \neq \infty \neq h_{p2}$, and $h_{p1} \neq h_{p2}$

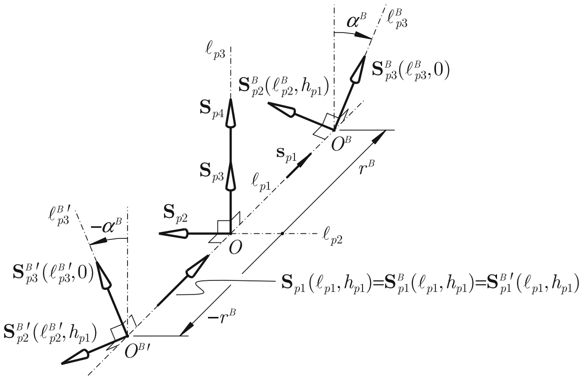

α^B
ℓ^B_{p3}
ℓ_{p3}
$\mathbf{S}^B_{p3}(\ell^B_{p3},0)$
$\mathbf{S}^B_{p2}(\ell^B_{p2},h_{p1})$
$\mathbf{S}_{p4}$
O^B
$\mathbf{s}_{p1}$
r^B
$\mathbf{S}_{p3}$
ℓ_{p1}
$\ell^{B\prime}_{p3}$
$-\alpha^B$
$\mathbf{S}_{p2}$
ℓ_{p2}
O
$\mathbf{S}^{B\prime}_{p3}(\ell^{B\prime}_{p3},0)$
$\mathbf{S}_{p1}(\ell_{p1},h_{p1})=\mathbf{S}^B_{p1}(\ell_{p1},h_{p1})=\mathbf{S}^{B\prime}_{p1}(\ell_{p1},h_{p1})$
$-r^B$
$\mathbf{S}^{B\prime}_{p2}(\ell^{B\prime}_{p2},h_{p1})$
$O^{B\prime}$

Fig. 3 $3I$ systems within a $4G$ system

$$h_3 = h^B_{p3} = 0, \quad \mathbf{S}_3(\ell_3, 0) = \mathbf{S}^B_{p3}(\ell^B_{p3}, 0), \quad h^B_{p1} = h^B_{p2} \neq 0, \quad P_{32,3} \equiv P_{34,3} \equiv O^B, \tag{1}$$

where O^B is the point where $\mathbf{S}^B_{p1}$, $\mathbf{S}^B_{p2}$ and $\mathbf{S}^B_{p3}$ intersect. Furthermore, for $j = 2, 4$,

$$\alpha_{3j} \neq 0, \quad h_j = h^B_{p1}\sin^2\alpha_{3j} \neq 0, \quad p_{3j} = h^B_{p1} \sin\alpha_{3j} \cos\alpha_{3j}. \tag{2}$$

It may be proven that any $3I$ system lying within a $4G$ system must meet the following requirements (where $h^B_{p3} = 0$ is enforced, Fig. 3):

(1) $\mathbf{S}^B_{p1} = \mathbf{S}_{p1}$, i.e. $\ell^B_{p1} \equiv \ell_{p1}$ and $h^B_{p1} = h^B_{p2} = h_{p1}$, with $\mathbf{S}_{p1}$ being the principal screw normal to ℓ_{p3} with the highest pitch in absolute value, i.e. $|h_{p1}| \geq |h_{p2}|$;
(2) the center O^B of B must lie on ℓ_{p1}, namely $O^B - O = r^B\mathbf{s}_{p1}$, with $r^B \in \mathbb{R}$;
(3) $\cos^2\alpha^B = (h_{p2} - h^B_{p3})/(h_{p1} - h^B_{p3}) = h_{p2}/h_{p1}$, with α^B being the angle that ℓ^B_{p3} forms with ℓ_{p3}, evaluated according to the right-hand rule about $\mathbf{s}_{p1}$ and such that $-\pi/2 < \alpha^B \leq \pi/2$;
(4) $r^B = (h_{p1} - h^B_{p3}) \sin\alpha^B \cos\alpha^B = h_{p1} \sin\alpha^B \cos\alpha^B$;

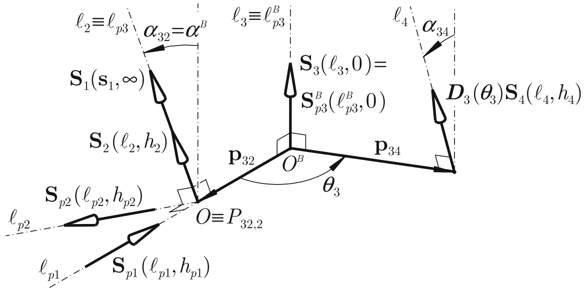

$\ell_2\equiv\ell_{p3}$
$\alpha_{32}=\alpha^B$
$\ell_3\equiv\ell^B_{p3}$
ℓ_4
α_{34}
$\mathbf{S}_1(\mathbf{s}_1,\infty)$
$\mathbf{S}_3(\ell_3,0)=$
$\mathbf{S}^B_{p3}(\ell^B_{p3},0)$
$\boldsymbol{D}_3(\theta_3)\mathbf{S}_4(\ell_4,h_4)$
$\mathbf{S}_2(\ell_2,h_2)$
$\mathbf{p}_{32}$
O^B
$\mathbf{p}_{34}$
θ_3
$\mathbf{S}_{p2}(\ell_{p2},h_{p2})$
ℓ_{p2}
$O\equiv P_{32,2}$
ℓ_{p1}
$\mathbf{S}_{p1}(\ell_{p1},h_{p1})$

Fig. 4 Ternary generator of a 4-PSS with a constant class of the general type

(5) h_{p1} and h_{p2} must have the same sign (when they are different from zero), i.e. $h_{p1} \geq h_{p2} \geq h^B_{p3} = 0$ or $h_{p1} \leq h_{p2} \leq h^B_{p3} = 0$.

For the sake of brevity, the proof of the above statements is not reported. It emerges from statements (1)–(5) that a $4G$ system may comprise only two $3I$ systems, i.e. B and B', such that $h^B_{p3} = 0$ (Fig. 3). B and B' are symmetric under reflection in ℓ_{p3}, and they coalesce when $h_{p2} = 0$ (in which case, $\alpha^B = \pi/2$ and $r^B = 0$).

If $S(\boldsymbol{\Theta})$ has to be a $4G$-PSS, h_{p1} and h_{p2} must remain constant as $\boldsymbol{\Theta}$ varies. Hence, α^B must remain constant too. Two cases need to be distinguished, depending on whether $\mathbf{S}_1$ has infinite or finite pitch. If $h_1 = \infty$, $\mathbf{S}_1$ must be parallel to ℓ_{p3}, as the only ∞-pitch screw of S lies in $\mathcal{C}(\ell_{p3})$. Since $\ell^B_{p3} \equiv \ell_3$, α^B must thus coincide with the angle α_{31} between $\mathbf{s}_1$ and $\mathbf{s}_3$. Since $\cos\alpha_{31} = \mathbf{s}_1 \cdot \mathbf{s}_3 = \cos\alpha_{32}\cos\alpha_{21} - \sin\alpha_{32}\sin\alpha_{21}\cos\theta_2$ and $\sin\alpha_{32} \neq 0$, α_B may be constant only if $\alpha_{21} = 0$, in which case $\alpha^B = \alpha_{31} = \alpha_{32}$. Hence, $\mathbf{S}_1$ and $\mathbf{S}_2$ are parallel, and they span $\mathcal{C}(\ell_{p3})$ (Fig. 4). This condition must be enforced also if h_1 is finite. In fact, $\mathbf{S}_1$ must be reciprocal, for arbitrary values of θ_2, to a screw $\mathbf{S}_{r1}$ of pitch $-h_{p1}$ passing through O^B and lying on a plane perpendicular to ℓ_3. It is not difficult to verify, by direct computation, that this may happen only if $\mathbf{S}_1$ and $\mathbf{S}_2$ are collinear, i.e. if they span $\mathcal{C}(\ell_{p3})$.[1] The 4-system illustrated in Fig. 4 is, thus, persistent. O coincides with $P_{32,2}$, $\mathbf{S}_{p1}$ lies along the common normal between $\mathbf{S}_2$ and $\mathbf{S}_3$, $\mathbf{S}_{p2}$ is orthogonal to ℓ_{p1} and ℓ_{p3}, and the following conditions (deriving from Eqs. (1) and (2) and statements (1)–(3) apply:

$$h_3 = 0, \quad h_4 = h_{p1}\sin^2\alpha_{34}, \quad p_{34} = h_{p1}\sin\alpha_{34}\cos\alpha_{34}, \tag{3a}$$

$$p_{32} = h_{p1}\cos\alpha_{32}\sin\alpha_{32}, \quad h_{p2} = h_{p1}\cos^2\alpha_{23}, \tag{3b}$$

with $h_{p1} \geq h_{p2} \geq 0$ or $h_{p1} \leq h_{p2} \leq 0$. In particular, when $h_{p2} = 0$ and $h_{p1} \neq 0$, Eq. (3b) implies $\alpha_{32} = \pi/2$ and $p_{32} = 0$. When $h_{p1} = 0$, Eq. (3a, b) require

[1] Indeed, $\mathbf{S}_1$ may be reciprocal, for arbitrary values of θ_2, to a screw of pitch $-h_{p1}$ intersecting ℓ_3 at right-angle, even if $p_{32} = 0$, $\alpha_{32} = \pi/2$ and $h_2 = h_{p1}$, with $\mathbf{S}_{r1}$ being, in this case, aligned with $\mathbf{S}_2$. However, requiring $\mathbf{S}_1$ to be also reciprocal to $\mathbf{S}_{r2}$, i.e. a screw both perpendicular to $\mathbf{S}_{r1}$ and intersecting it, straightforwardly leads to the condition that $\mathbf{S}_1$ and $\mathbf{S}_2$ must be collinear. Explicit calculations are not reported for the sake of brevity.

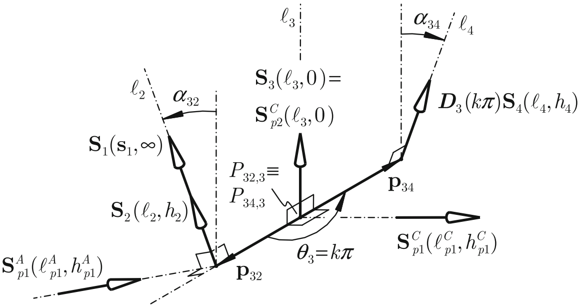

ℓ_3
α_{34}
ℓ_4
ℓ_2
α_{32}
$\mathbf{S}_3(\ell_3,0)=$
$\mathbf{S}^C_{p2}(\ell_3,0)$
$\boldsymbol{D}_3(k\pi)\mathbf{S}_4(\ell_4,h_4)$
$\mathbf{S}_1(\mathbf{s}_1,\infty)$
$P_{32,3}\equiv$
$P_{34,3}$
$\mathbf{p}_{34}$
$\mathbf{S}_2(\ell_2,h_2)$
$\theta_3=k\pi$
$\mathbf{S}^C_{p1}(\ell^C_{p1},h^C_{p1})$
$\mathbf{S}^A_{p1}(\ell^A_{p1},h^A_{p1})$
$\mathbf{p}_{32}$

Fig. 5 Ternary generator of a $4G$-PSS obtained from a $3VIII$-PSS

$h_{p2} = h_4 = p_{32} = p_{34} = 0$, and a ternary generator of a $4I$-PSS is indeed obtained (cf. Fig. 2c in [1]).

The $4G$-PSS in Fig. 4 may also be derived from the $3VIII$-PSS generator shown in Fig. 7a of [3]. Let $\mathcal{A} = \langle \mathbf{S}_1(\mathbf{s}_1, \infty), \mathbf{S}_2(\ell_2, h_2), \mathbf{S}_3(\ell_3, h_3)\rangle$, where $\mathbf{S}_1(\mathbf{s}_1, \infty)$ and $\mathbf{S}_2(\ell_2, h_2)$ form a cylindrical ISS $\mathcal{C}(\ell_2)$, and $\mathbf{S}_3(\ell_3, h_3)$ is a finite-pitch screw nonparallel to ℓ_2. According to [3], $\mathcal{A}$ generates a $3VIII$-PSS A such that

$$h^A_{p2} = h_3 + p_{32} \cot \alpha_{32}, \tag{4}$$

where $\mathbf{S}^A_{p2}(\ell^A_{p2}, h^A_{p2})$ is the principal screw of A perpendicular to ℓ_2 and $\mathbf{n}_{32}$. Since A contains one ∞-pitch screw and $\mathcal{S} = \langle \mathcal{A}, \mathbf{S}_4(\ell_4, h_4)\rangle$ has to generate a $4G$-PSS, $\mathbf{S}_4$ must have finite pitch, and it cannot be parallel to ℓ_3.

If θ_3 is the angle between $\mathbf{n}_{32}$ and $\mathbf{n}_{34}$, whenever $\theta_3 = k\pi$, $k \in \mathbb{N}$, ℓ_2, ℓ_3 and ℓ_4 are perpendicular to the same direction, i.e. $\mathbf{n}_{32} = \pm\mathbf{n}_{34}$ (Fig. 5). In such configurations $\mathcal{C}(\ell_2)$, $\mathbf{S}_3$ and $\mathbf{D}_3(k\pi)\mathbf{S}_4$ cannot be linearly independent, otherwise they would form a 4-system reciprocal to an ∞-pitch screw parallel to $\mathbf{n}_{32}$ (and this could not be a $4G$ system). Thus, these configurations must be singular and, therein, $\mathbf{D}_3(k\pi)\mathbf{S}_4$ must belong, for any k, to A. This implies $P_{32,3} = P_{34,3}$ and $h_3 = 0$. Moreover, since $\mathbf{D}_3(0)\mathbf{S}_4$ and $\mathbf{D}_3(\pi)\mathbf{S}_4$ are symmetric under reflection in ℓ_3, they must form a cylindroid C having one of its principal screws, say $\mathbf{S}^C_{p2}$, on ℓ_3 [7, Chap. 3]. Since C belongs to A and no line in A other than ℓ_2 may contain more than one screw, $\mathbf{S}^C_{p2}$ must necessarily coincide with $\mathbf{S}_3$, so that $h^C_{p2} = h_3 = 0$. By letting $\mathbf{S}^C_{p1}(\ell^C_{p1}, h^C_{p1})$ be the other principal screw of C, it must be [7, Chap. 3]

$$h_4 = h^C_{p2} + (h^C_{p1} - h^C_{p2}) \sin^2 \alpha_{34} = h^C_{p1} \sin^2 \alpha_{34}, \tag{5a}$$

$$p_{34} = (h^C_{p1} - h^C_{p2}) \sin \alpha_{34} \cos \alpha_{34} = h^C_{p1} \sin \alpha_{34} \cos \alpha_{34}. \tag{5b}$$

By requiring $\mathbf{S}^C_{p1}$ to belong to A, one also obtains that

$$h^A_{p2} = h^C_{p1} + p_{32} \cot(\alpha_{32} + \pi/2) = h^C_{p1} - p_{32} \tan \alpha_{32} \tag{6}$$

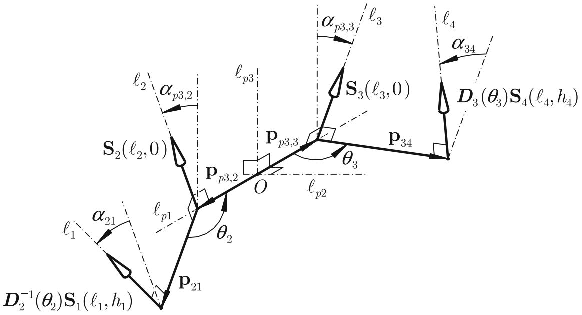


Fig. 6 Quaternary generator of a 4-PSS with a constant class of the general type

and, thus, recalling Eq. (4) and rearranging terms,

$$p_{32} = h_{p1}^{C} \sin\alpha_{32}\cos\alpha_{32}, \quad h_{p2}^{A} = h_{p1}^{C}\cos^2\alpha_{32}. \tag{7}$$

By varying θ_3, $\mathbf{S}_3$ and $\mathbf{D}_3(\theta_3)\mathbf{S}_4$ generate a pencil of cylindroids all congruent to C, i.e. a $3I$-PSS with principal pitches equal to 0 and h_{p1}^{C}. By letting $h_{p1}^{C} = h_{p1}$ and $h_{p2}^{A} = h_{p2}$, Eqs. (5) and (7) coincide with Eq. (3a, b). The generator in Fig. 4 is evidently re-obtained.

2.2 The Quaternary Generator

The arguments developed in Sect. 2.1 provide a clue for obtaining a quaternary generator of $4G$-PSS. It has been seen, in fact, that a $4G$ system such that $h_{p1} > h_{p2} \geq 0$ or $h_{p1} < h_{p2} \leq 0$ comprises only two $3I$ systems with a 0-pitch central principal screw, i.e. B and B' (Fig. 3). According to statements (2)–(4), the poses of the central axes ℓ_{p3}^{B} and $\ell_{p3}^{B\,\prime}$ of B and B' are unambiguously determined by the value of the principal pitches of S, i.e. h_{p1} and h_{p2}. In Fig. 4, the screws $\mathbf{S}_3$ and $\mathbf{D}_3(\theta_3)\mathbf{S}_4$ span (as θ_3 varies) one of these $3I$ systems, say B, with the entire 4-system being generated by composing B with the cylindrical ISS $\mathcal{C}(\ell_{p3})$. If $\mathcal{C}(\ell_{p3})$ is replaced by two screws, i.e. $\mathbf{S}_2(\ell_2, 0)$ and $\mathbf{D}_2^{-1}(\theta_2)\mathbf{S}_1(\ell_1, h_1)$, which span (as θ_2 varies) the other $3I$ system, i.e. B', the same vector subspace is obviously obtained (Fig. 6). The joint screws of the described generator satisfy the conditions

$$h_2 = h_3 = 0, \quad \alpha_{p3,2} = \alpha_{p3,3}, \tag{8a}$$

$$h_{p2} = h_{p1}\cos^2\alpha_{p3,2}, \quad p_{p3,2} = p_{p3,3} = h_{p1}\cos\alpha_{p3,2}\sin\alpha_{p3,2}, \tag{8b}$$

$$h_1 = h_{p1}\sin^2\alpha_{21}, \quad p_{21} = h_{p1}\sin\alpha_{21}\cos\alpha_{21}, \tag{8c}$$

$$h_4 = h_{p1}\sin^2\alpha_{34}, \quad p_{34} = h_{p1}\sin\alpha_{34}\cos\alpha_{34}, \tag{8d}$$

where $p_{p3,i}$ and $\alpha_{p3,i}$ are, respectively, the shortest distance and the relative angle between ℓ_i and ℓ_{p3}, $i = 2, 3$. The quaternary generator in Fig. 6 does not allow $h_{p2} = 0$ and $h_{p1} \neq 0$, since in this case $\alpha_{p3,2} = \alpha_{p3,3} = \pi/2$ and $p_{p3,2} = p_{p3,3} = 0$, and $\mathbf{S}_2$ and $\mathbf{S}_3$ would coincide. Also, this generator may not degenerate into a $4I$-PSS, as for $h_{p1} = 0$ all screws would have zero pitch and pass through O.

3 Conclusions

Two generators of 4-dimensional screw systems with a constant class of the general type (i.e. $4G$-PSSs) were disclosed. It may be proven that no other $4G$-PSS generators exist. Due to space limitations, the proof is omitted, but it will be reported in a future extended version of the contribution.

References

1. Carricato, M.: Persistent screw systems of dimension four. In: Lenarčič, J., Husty, M. (eds.) Latest Advances in Robot Kinematics, pp. 147–156. Springer, Dordrecht (2012)
2. Carricato, M., Rico Martínez, J.M.: Persistent screw systems. In: Lenarčič, J., Stanišià, M.M. (eds.) Advances in Robot Kinematics: Motion in Man and Machine, pp. 185–194. Springer, Dordrecht (2010)
3. Carricato, M., Rico Martínez, J.M.: Persistent screw systems of dimension three. In: Proceedings of 13th World Congress in Mechanism and Machine Science, pp. 1–12. Paper No. A7_430. Guanajuato, Mexico (2011)
4. Gibson, C.G., Hunt, K.H.: Geometry of screw systems–1, screws: genesis and geometry. Mech. Mach. Theory **25**(1), 1–10 (1990)
5. Gibson, C.G., Hunt, K.H.: Geometry of screw systems–2, classification of screw systems. Mech. Mach. Theory **25**(1), 11–27 (1990)
6. Hervé, J.: Lie group of rigid body displacements, a fundamental tool for mechanism design. Mech. Mach. Theory **34**(5), 719–730 (1999)
7. Hunt, K.H.: Kinematic Geometry of Mechanisms. Clarendon, Oxford (1978)

Determination of Maximal Singularity-Free Workspace of Parallel Mechanisms Using Constructive Geometric Approach

Mohammad Hadi Farzaneh Kaloorazi, Mehdi Tale Masouleh, Stéphane Caro and Behnam Mashhadi Gholamali

Abstract This paper proposes a novel approach to obtain the maximal singularity-free regions of planar parallel mechanisms which is based on a constructive geometric reasoning. The proposed approach consists of two algorithms. First, the borders of the singularity-free region corresponding to an arbitrary start point of the moving platform is obtained. Then, the second algorithm aims to find the center of the maximal singularity-free circle which is obtained using the so-called offset curve algorithm. As a case study, the procedure is applied to a 3-$\underline{\text{P}}$RR planar parallel mechanism and results are given in order to graphically illustrate the effectiveness of the proposed algorithm. The proposed approach can be directly applied to obtain the maximal singularity-free circle of similar parallel mechanisms, which is not the case for other approaches proposed in the literature which is limited to a given parallel mechanism, namely, 3-R$\underline{\text{P}}$R. Moreover, as the main feature of the proposed approach, it can be implemented both in a CAD system or in a computer algebra system where non-convex and re-entrant curves can be considered.

Keywords Parallel mechanisms · Singularity-free workspace · Geometric approach · Offset curve algorithm.

S. Caro (✉)
IRCCyN/CNRS, Nantes, France
e-mail: stephane.caro@irccyn.ec-nantes.fr

M. H. F. Kaloorazi · M. T. Masouleh · B. M. Gholamali
Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran
e-mail: mhfarzane@ut.ac.ir

M. T. Masouleh
e-mail: m.t.masouleh@ut.ac.ir

B. M. Gholamali
e-mail: b.mashhadi@ut.ac.ir

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_34,

1 Introduction

Parallel Mechanisms (PMs) are a type of robotic mechanical systems composed of one moving platform and one base connected by at least two serial kinematic chains in parallel [8]. PMs are often erroneously said to be recent developments, have a pedigree far more ancient than that of serial manipulators, which are usually called anthropomorphic [3]. However, the last two decades have witnessed a noticeable rise in the number of publications regarding the kinematic and dynamic analyses of PMs to the end of proposing the most promising design. As it will be seen later on, PMs have their own drawbacks and even a simple one can lead to complicated kinematics analysis. In general, when a PM tends towards structural generality, its geometry and kinematic analysis get more complicated. The latter is the case of this paper where upon applying a simple modification into the kinematic arrangement of a planar 3-Degree-Of-Freedom (DOF) planar PM, the so-called 3-R$\underline{\text{P}}$R PM, and turning it to a 3-$\underline{\text{P}}$RR PM, then the problem of singularity-free workspace becomes a cumbersome task and would be elusive to classical approaches, proposed for the former 3-R$\underline{\text{P}}$R PM [6, 7, 10]. Here and throughout this paper, $\underline{\text{P}}$ stands for an actuated prismatic joint and R stands for a passive revolute joint.

Designing a PM with a singularity-free workspace is a vital condition for further analysis, such as path planning and control problem. In the literature, most of the study propounded on this topic, i.e., singularity-free workspace, are based on either primitive numerical approach or some complicated mathematical approaches where both entail some limits. In [2], Bonev et al. conducted an exhaustive study on the singularity locus of planar 3-DOF PMs by resorting to *screw theory*. In [10], a method based on geometric parameters of the mechanism under study is represented for which a singularity-free circle in the workspace of a 3-R$\underline{\text{P}}$R PM is obtained. In [7], Li et al. redefined the problem as an optimization problem accompanied with a constraint and resorted to Lagrangian multipliers and obtained the maximal singularity-free circle of a 3-R$\underline{\text{P}}$R PM for a prescribed center point. In [6], Jiang and Gosselin proposed some numerical recipes in order to find the singularity-free workspace of planar 3-DOF PMs.

This paper aims at obtaining the Maximal Singularity-Free Circle (MSFC) of 3-DOF planar PMs for a given orientation of the mobile platform. Obtaining the MSFC has eminent effect on reliability and endurance of the workspace of the robot. The circle is chosen because it has the most regular shape and comes in handy in practice. To the best knowledge of authors, in the literature, results of the MSFC were obtained only for a prescribed center point and this assumption bounds the radius of the circle and results into a local optimum solution. In this study, the center point of the MSFC is not prescribed from the outset and subject to be found using the geometrical reasoning proposed in this paper. It should be noted that the MSFC can be readily computed once the center is obtained. The proposed approach for obtaining the center point of the MSFC is based on a novel constructive geometric procedure which is the unique aspect of this work and distinct it with the others reported in the literature [7, 10].

In this paper, a novel geometric algorithm is proposed, called Alg. I, in order to obtain the singularity-free region of PMs which could be applied to non-convex singularity locus. Moreover, offset curve algorithm, Alg. II, is adapted for the geometric purpose of this work. Offset curve algorithm [1, 4] is a geometric constructive tool which has diverse engineering applications and has consequently motivated extensive researches concerning various offset techniques. It plays an important role in numerical controls and CAD/CAM applications [4]. To the best knowledge of the authors, the problem of MSFC has never been investigated upon a geometric standpoint. The proposed algorithm, which is inspired from geometric properties associated to the MSFC, could be implemented either in a computer algebra system or using a CAD system. Almost all the CAD systems have the possibility to make an offset of complex curves. in this paper, due to the simplicity, the details are skipped. Thus, more emphasis is placed on the numerical approach proposed in this paper to make such an offset to a given curve, specifically the singularity-locus curve.

Through this paper, in order to illustrate the proposed approach, as a case study, the procedure of obtaining the MSFC is applied to a 3-PRR planar PM. However, it can be extended to all planar 3-DOF PMs presented in [2]. To the best knowledge of authors, 3-RPR and 6-UPS (SPS) PMs have been widely treated in the literature since they lead respectively to quadratic and cubic polynomial expressions for their singularity locus which simplifies considerably the mathematical challenge. A minor modification in the kinematic arrangement, for instance having a 3-PRR PM, leads to the complexity of the procedure for which methods reported in [7] are no more applicable and fail to provide satisfactory results. One of the problems in such an investigation is the presence of the square roots in the singularity loci expressions. The proposed algorithm is split into two sub-algorithms: (1) a first algorithm to obtain the subregion of interest for the MSFC, called Alg. I, and (2) a second one for obtaining the center point of the MSFC for the foregoing subregion, called Alg. II.

The remainder of this paper is organized as follows. First, the kinematic properties of the PM under study, i.e., the 3-PRR PM, is broadly reviewed. Then, Alg. I toward obtaining the singularity-free region is fully described, by having in mind that, as a case study, it will be applied to the 3-PRR PM. Finally, the offset curve algorithm is introduced to the sake of proposing Alg. II, which is applied into the singularity region obtained from Alg. I.

2 Kinematic Review of 3-PRR Planar Parallel Mechanism

A 3-PRR planar PM consists of three kinematically identical limbs actuated by a prismatic joint fixed at the base and followed by two passive R joints, as depicted in Fig. 1 (left). As it can be observed from Fig. 1 (left), O_{xyz}, with $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ as unit vectors, represents the fixed frame and O_{xyz} stands for the moving frame. The pose (position and orientation) of the mechanism is defined by (x, y, φ) where $\mathbf{p} = [x, y]^T$ and φ represent respectively the Cartesian position and the orientation of the moving frame with respect to the fixed frame. Upon resorting to *screw theory* [2], the Jacobian matrix of the mechanism can be formulated as follows:

$$\mathbf{J} = \left[\begin{array}{ccc|ccc} \mathbf{l}_1 & \mathbf{l}_2 & \mathbf{l}_3 & \mathbf{0} & \mathbf{0} & \hat{\mathbf{k}} \\ \mathbf{r}_1 \times \mathbf{l}_1 & \mathbf{r}_2 \times \mathbf{l}_2 & \mathbf{r}_3 \times \mathbf{l}_3 & \hat{\mathbf{i}} & \hat{\mathbf{j}} & \mathbf{0} \end{array}\right]^T, \tag{1}$$

in which $\mathbf{l}_i$, $i = 1, 2, 3$, is the unit vector along the line connecting point B_i to point C_i and $\mathbf{r}_i$ is the vector connecting the origin of the moving platform to point C_i. Singular configurations of the mechanism occurs when the Jacobian matrix becomes rank deficient [5], i.e., the determinant of the foregoing matrix vanishes, $\det(\mathbf{J}) = 0$. The latter leads to have a polynomial of degree 20 (20 in y and 16 in x) for a constant-orientation of the moving platform [2]. It is worth to be noticed that the latter polynomial corresponds to all the eight working modes of the mechanism and, as reported in [2], it is not possible to find a polynomial expression for a single working mode among the eight one. It should be noted that obtaining such a polynomial is an extremely delicate task and is beyond the scope of this paper. Skipping the latter mathematical manipulations, Fig. 1 (right) depicts the singularity locus of the 3-$\underline{\text{P}}$RR planar PM for $\varphi = \pi/36$.

3 Algorithm to Obtain the Subregion of the Singularity-Free Workspace, Alg. I

As it can be inferred directly from Fig. 1 (right), the singularity locus is such that splits the workspace of the mechanism into different regions which, in this paper, are referred to as *subregion* and called $\mathcal{H}_i,\ i = 1, \cdots, n$. It should be noted that some subregions are not mentioned in Fig. 1 (right) to not overload the figure. This section is devoted to present a new method to the end of obtaining the boundaries of the singularity-free subregion, $\mathcal{H}_i$. It is worth noting that the proposed method could be applied to any kind of complex curve and it does not depend on the convexity of the subregions. Moreover, the main challenge in finding a subregion is the intersection points among different branches of the singularity curve, which are known as *bifurcation* points, called B as indicated in Fig. 2.

Algorithm 1 The pseudo-code of the algorithm to obtain the subregion of the singularity-free workspace, Alg. I.

```
Input: $\det(\mathbf{J}) = 0$, $P_0$ as the starting point of the moving platform and $\varepsilon$ as the desired accuracy
Output: The corresponding singularity-free subregion, called $\mathscr{H}_1$, consists of $P_i$, $i = 1, \cdots, n$
$i \leftarrow 1$;
$P_i = $ fminsearch($|\det(\mathbf{J})|, P_0$) % use "Nelder-Mead" to find a point on the singularity locus
while $|P_i - P_{i-1}| < \varepsilon$ do
    $i \leftarrow i + 1$
    $K_i = $ circle($P_i, \varepsilon$) % create a clockwise circle with $P_i$ and $\varepsilon$ as center and radius
    $S = $ solve($\det(\mathbf{J}), K_1$) % save the intersection points of the circle and the singularity locus
    $P_i = $ order($S$, clockwise)(1) % save the first item of S with respect to trigonometry order
end while
```

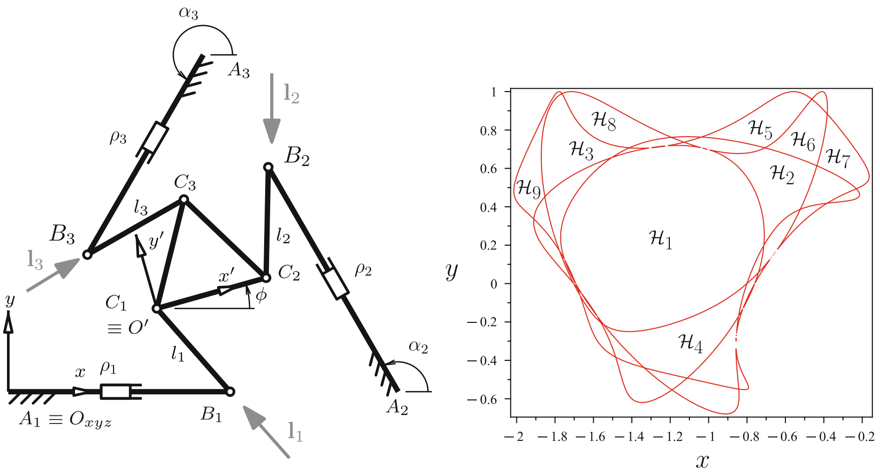


Fig. 1 *Left*: 3-PRR planar PM [2]. *Right*: Singularity locus of 3-PRR planar PM for $\varphi = \pi/36$ and subregions $\mathcal{H}_i$

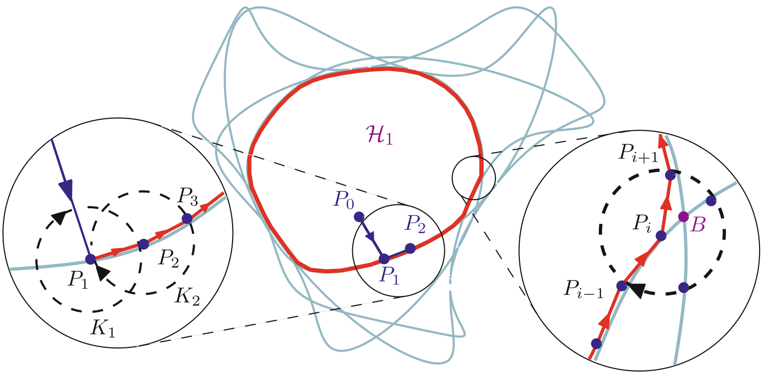


Fig. 2 Result of applying the proposed algorithm to the end of obtaining the singularity-free region of $\mathcal{H}_1$. P_0 is the starting point and the *red polygon* represents the singularity-free of $\mathcal{H}_1$

Algorithm 1 represents the concept of Alg. I which the reasoning is fully described in what follows. The first step to obtain the boundaries of a subregion is to specify which subregion among $\mathcal{H}_i$, $i = 1, \cdots, n$, is of concern. This can be done by specifying an arbitrary point, P_0, lying inside the desired subregion. In practice, this point is the position of the moving platform in the reference configuration of the mechanism. Therefore, the workspace of the moving platform should be bounded within the subregion of the start point, i.e., P_0.

The algorithm starts by finding a point on the singularity locus which lies on the boundary of the desired subregion, called point P_1. The latter can be done readily by using an unconstrained optimization approach for $|\det(\mathbf{J})| = 0$, as the objective function, i.e., using direct pattern search, namely Nelder-Mead (simplex) method with ε as simplex parameter [9]. From P_1, the algorithm starts to search through the boundary of the subregion other points constituting the subregion. To do so, a line is passed from P_0 to P_1 and creates a trigonometry circle, K_1, with P_1 and ε as center point and radius, respectively. The value of ε stands for the computation accuracy. The trigonometric circle covers angle between $\varphi = [0, 2\pi)$ and can be either clockwise or counter-clockwise. The line corresponds to $\varphi = 0$. By changing the angle t from 0 to 2π, the first intersection point of K_1 and singularity locus will be saved and called P_2, as depicted in Fig. 2. In practice, this can be done by considering the discrete circle. By the same token, a trigonometric circle, called K_2, will be created with P_2 and ε as center point and radius, respectively, with the same direction as the previous circle. The same procedure pursues for new points $P_i, i = 1, \dots n$, and at each step the first intersection point will be added to a list of points, called $\mathcal{C}_0$. The stopping criterion of the algorithm is that, the last obtained point, P_n, be close enough to the first member of $\mathcal{C}_0$, i.e., P_1. In other words, $\| P_n - P_1 \| < \varepsilon$. Finally, $\mathcal{C}_0$ is a closed polygon which represents the singularity-free region corresponding to the reference configuration of the mechanism.

The main feature of this algorithm is its ability to deal properly with the multi-sectional areas caused by intersection among the singularity curve. This type of areas are represented in Fig. 2, and, as it can be observed, due to the reasoning of the algorithm, these multi-sectional areas have no significant effect on the procedure and will be automatically circumvented. More precisely, Alg. I is able to detect the correct region when approaching a bifurcation point, B. In Fig. 2, Alg. I is applied in order to find $\mathcal{C}_0$ as the singularity-free region. Point P_0 is the position of the moving platform in its reference configuration. By resorting to Nelder-Mead (simplex) method, $\varepsilon = 0.5$, a point close to the singularity locus is obtained, P_1. Pursuing Alg. I, more points $P_i, i = 1, \cdots, n$, are obtained. **It took 3 s to compute $\mathcal{C}_0$, with a 2 GHz processor**. In fact, $\mathcal{C}_0$ will be used in the next section as the singularity-free region in order to obtain the MSFC. As it will be more apparent in the upcoming section, errors due to the iterative approximation of singularity-free region $\mathcal{C}_0$ tend to zero upon applying offset curve algorithm.

4 Obtaining the MSFC Using Offset Curve Algorithm, Alg. II

The whole concept of the offset curve algorithm, is based on two geometric properties of MSFC for which it should be (a) tangent to the intersection points between the MSFC and the boundaries of the polygon $\mathcal{C}_0$ and (b) its center point should be equidistant to all the intersection points. For a closed-planar polygon $\mathcal{C}_j(t)$, its offset polygons can be written mathematically as follows [4]:

$$\mathcal{C}_{j+1}(t) \longleftarrow \mathcal{C}_j(t) \pm d\, \mathbf{n}(t), \quad j = 1,\ 2,\ \dots m \tag{2}$$

where d is the offset distance and $\mathbf{n}(t)$ is the normal vector at point t on the polygon $\mathcal{C}_j(t)$. In the problem addressed in this paper, “−" is considered as ±, because it is

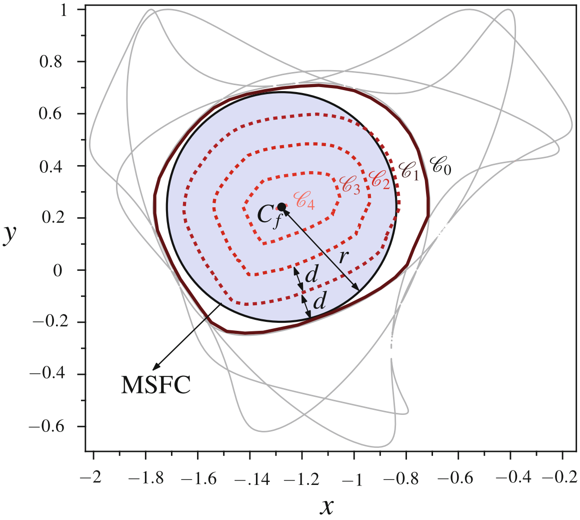


Fig. 3 Using Alg. I, singularity-free region $\mathcal{H}_1$ is obtained, called $\mathcal{C}_0$. Then, by applying four times Alg. II (offset concept) into $\mathcal{C}_0$, the area of the last polygon, $\mathcal{C}_4$, is less than ε. Therefore, it can be estimated by a point, C_f as the center of MSFC with radius as $r = 4\ d$

desired to decrease the area of $\mathcal{C}_i$ to a point. Having in mind the two latter properties of the MSFC, the algorithm is organized as follows. The first step consists in obtaining the tangent of each point t on the perimeter of the polygon $\mathcal{C}_0$. Then each point on the perimeter is moved forward by a given value, d, in the direction of the line perpendicular to its tangent which yields a new polygon, $\mathcal{C}_1$. By the same token, one can obtain $\mathcal{C}_2$, $\mathcal{C}_3$, $\ldots$ and $\mathcal{C}_m$. The latter procedure should be persuaded in such a way that for a given m, the area of $\mathcal{C}_m$ reduces to approximately a point for which the algorithm stops. The latter point, C_f, represents the center of the MSFC. The radius of the circle is simply computed as $r = m\ d$, where m stands for the number of applied offset and can be chosen arbitrarily.

It should be noted that a special situations may arise, which consists in the cases for which the curve contains some necks. In such cases, upon pursuing the offset curve algorithm the curve will be separated and split into different curves and the algorithm should apply the offset approach for each subregion and obtain the corresponding MSFC [1, 4]. The MSFC is the biggest one for all the subregions.

The offset curve algorithm is available in Matlab by using the command `bufferm` and almost all CAD software have the capability of executing such a curve offset. Figure 3 represents the result of the MSFC for a 3-$\underline{P}$RR PM for a constant-orientation of the moving platform singularity locus. Using Alg. I, the singularity-free subregion of the mechanism for a prescribed orientation is obtained, $\mathcal{C}_0$. Then by applying Alg. II, the corresponding MSFC is obtained. In Fig. 3, $\mathcal{C}_i$, $i = 1, \cdots, 4$, are new offset polygons in which, each of them is generated by offsetting its preceding one by d as the normal distance. The latter is continued until converging to a point, C_f, being the center of the MSFC. Finally, by having the center of MSFC the corresponding radius can be readily obtained.

5 Conclusion

This paper proposed a new geometric constructive approach to the end of obtaining the singularity-free region and the maximal singularity-free circle of 3-DOF planar parallel mechanisms. The procedure consisted of two algorithms, which are mainly based on geometrical reasoning of the problem. First, using a new method the boundaries of the singularity-free region corresponding to the starting point of the moving platform was obtained. Then using the so-called offset curve algorithm, the center of the maximal singularity-free circle in the corresponding region was computed. Special conditions were taken into account and proved the robustness of the algorithm. As a case study, the 3-$\underline{\text{P}}$RR planar parallel mechanism was considered. Ongoing works consist in extending the algorithm to higher DOF PMs and taking into account the workspace boundaries as additional constraints to the problem, which is a definite asset in practice.

References

1. Aichholzer, O., Aigner, W., Aurenhammer, F., Hackl, T., Jüttler, B., Pilgerstorfer, E., Rabl, M.: Divide-and-conquer for voronoi diagrams revisited. Comput. Geom. **43**(8), 688–699 (2010)
2. Bonev, I.A.: Geometric analysis of parallel mechanisms. Ph.D. thesis, Laval University, Quebec, QC, Canada (2002)
3. Davidson, J., Hunt, K., Pennock, G.: Robots and Screw Theory: Applications of Kinematics and Statics to Robotics. Oxford University Press, New York (2004)
4. Elber, G., Lee, I., Kim, M.: Comparing offset curve approximation methods. IEEE Comput. Graphics Appl. **17**(3), 62–71 (1997)
5. Gosselin, C.: Determination of the workspace of 6-DOF parallel manipulators. ASME J. Mech. Des. **112**(3), 331–336 (1990)
6. Jiang, L.: Singularity-free workspace analysis and geometric optimization of parallel mechanisms. Ph.D. thesis, Laval University, Quebec (2008)
7. Li, H., Gosselin, C., Richard, M.: Determination of maximal singularity-free zones in the workspace of planar three-degree-of-freedom parallel mechanisms. Mech. mach. theory **41**(10), 1157–1167 (2006)
8. Merlet, J.P.: Parallel Robots. Springer, Dordrecht (2006)
9. Rao, S.: Engineering optimization: theory and practice. Wiley, New York (2009)
10. Yang, Y., O'Brien, J.: A case study of planar 3-RPR parallel robot singularity free workspace design. In: International Conference on Mechatronics and Automation (ICMA), pp. 1834–1838. IEEE (2007)

Kinematic Design of Two Elementary 3DOF Parallel Manipulators with Configurable Platforms

Antonius G.L. Hoevenaars, Patrice Lambert and Just L. Herder

Abstract Parallel Manipulators with Configurable Platforms (PMCPs) have platforms with internal degrees of freedom and form a class of manipulators that is not covered by existing type synthesis methods. Because the minimum number of legs for a PMCP is three, fully parallel 3DOF PMCPs may be considered an elementary subset of PMCPs. To support the extension of type synthesis methods to PMCPs, this paper presents the first kinematic designs of manipulators from this subset. A structured design method has led to the kinematic design of two spatial manipulators that are both capable of independently performing one translation, one rotation and one internal platform motion.

Keywords Parallel mechanisms · Configurable platform · 3DOF · Grasping motion · Spatial

1 Introduction

Robotic manipulation sometimes requires additional degrees of freedom (DOF) such as grasping on top of the rigid end-effector motion. Multiple solutions have been proposed to achieve this additional motion. One example is to combine two separate mechanisms [4] and another is to attach a gripper mechanism in series to the end-effector, as is the case in the commercial omega.7 by Force Dimension. The first

A. G. L. Hoevenaars (✉), P. Lambert · J. L. Herder
Department of Precision and Microsystems Engineering, Delft University of Technology,
Delft, The Netherlands
e-mail: A.G.L.Hoevenaars@TUDelft.nl

P. Lambert
e-mail: P.Lambert@TUDelft.nl

J. L. Herder
e-mail: J.L.Herder@TUDelft.nl

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_35,

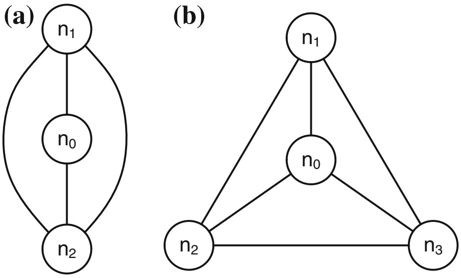


Fig. 1 **a** Using graph theory a PMCP mechanism with two legs can be represented as a series-parallel mechanism with a base n_0 and two link nodes n_1 and n_2, **b** A PMCP with three legs cannot be represented as a series-parallel mechanism, but yields a so-called wheel grap

solution increases the complexity of the system while the latter adds the inertia of an additional motor to the end-effector. Because low inertia at the end-effector is one of the distinguishing features of parallel manipulators, additional inertia especially affects the performance of parallel manipulators.

In the past decade it has been recognised that additional DOF can also be added to the end-effector of a parallel manipulator without compromising its parallel structure. This is achieved by replacing the rigid end-effector with an additional closed loop. Following the 4DOF planar manipulator with grasping motion by Yi et al. [10], Mohamed and Gosselin generalised the analysis of this new class of manipulators called Parallel Manipulators with Configurable Platforms (PMCP) [8]. Other examples of such PMCPs are the Par4 by Nabat et al. [9] and a 5DOF design by Lambert et al. [6].

An illustrative method to discuss kinematic structures is graph theory [1], which represents every mechanism as a series of joints (lines) and rigid bodies (nodes). Figure 1 illustrates how in graph theory a PMCP with two legs is kinematically equivalent to a *series-parallel architecture*, while a PMCP with three legs is not; in fact it belongs to a different category labelled *non-series-parallel architectures* [5]. PMCPs with three legs (serial chains) may therefore be regarderd as the most basic subset of PMCP designs. In this paper only *fully parallel manipulators* are considered, for which the number of legs is strictly equal to the number of DOF of the end-effector [7]. Thus, if only the joints located at the base are actuated, three legs allow 3DOF. Consequently, it is argued in this paper that fully parallel 3DOF PMCPs represent an elementary subset of PMCP designs.

Interestingly, PMCPs discussed in the literature all have a minimum of 4DOF. They have not been developed using a type synthesis method such as the one introduced by Kong and Gosselin [3] or Gogu [2], since existing methods do not cover PMCPs. Because fully parallel 3DOF PMCPs are argued to form an elementary subset of PMCP designs, examples from this subset may provide interesting input for the future development of a type synthesis method that does cover PMCPs.

The goal of this paper is to verify the existence of fully parallel 3DOF PMCPs and present the first architectures from this elementary subset. The structure of the paper is as follows. First the design method is discussed that leads to the two kinematic architectures presented in this paper. Next, the inverse Jacobian is derived for one of the two kinematic designs and four singular configurations are identified.

2 Design Method

Because no type synthesis method exists for PMCPs, the method in this paper relies on the structured combination of a four-bar mechanism (the platform) with a set of pre-defined legs. Furthermore, two restricting conditions are posed on the designs. The first condition is that the resulting 3DOF PMCPs shall be fully parallel. The number of legs is therefore strictly limited to three. Secondly, the axis associated with each DOF shall coincide with an axis of either the inertial reference frame XYZ or the platform reference frame $X^*Y^*Z^*$. This condition facilitates a straightforward description of the resulting mobilities.

The design method applied in this paper consists of four steps. First, the building blocks are defined: a planar four-bar mechanism and three identical legs. A four-bar mechanism with links of equal length is used, which is known to have three overconstraints and one internal DOF. Thus, the total number of platform DOF is seven. The internal DOF is expressed as the distance P_g between one of the joints and the platform reference frame origin. On the premise that a fully parallel 3DOF PMCP requires each of the three legs to have a minimum of three DOF, a minimal leg consists of two links and three joints and describes planar motion. One of the end joints is connected to an actuator at the base. In this paper the choice was made to use rotating actuators but this choice does not impact the DOF of the individual legs. The described building blocks are shown in Fig. 2a.

The second step is to constrain the motion of the platform reference frame origin to a plane, which is achieved through the connection of two legs to opposite joints of the four-bar mechanism. These legs are connected such that the resulting plane of motion of the platform reference frame origin is perpendicular to either X^* or Y^*. This is to ensure that the remaining DOF are all alligned with an axis of either the inertial reference frame or the platform reference frame. The plane of motion of the platform reference frame origin is here defined as the XZ-plane, as shown in Fig. 2b. The mechanism now has four DOF.

In the third step an additional DOF is constrained using the third leg. To constrain the platform in another DOF, the third leg is oriented in either of the planes perpendicular to the first two legs. Connecting the third leg in this orientation to one of the two remaining platform joints adds two additional constraints (and one overconstraint) to the platform. The state of the two kinematic designs after this step is shown in Fig. 2c and d.

By constraining five of the original seven DOF, both mechanisms shown in Fig. 2c and d have two DOF remaining. The final step is therefore to relieve one of the constrained DOF by introducing an additional joint. For the mechanism shown in Fig. 2d this also requires a change in the orientation of the joint connecting the third leg to the platform. The two resulting kinematic designs are shown in Fig. 2e and f.

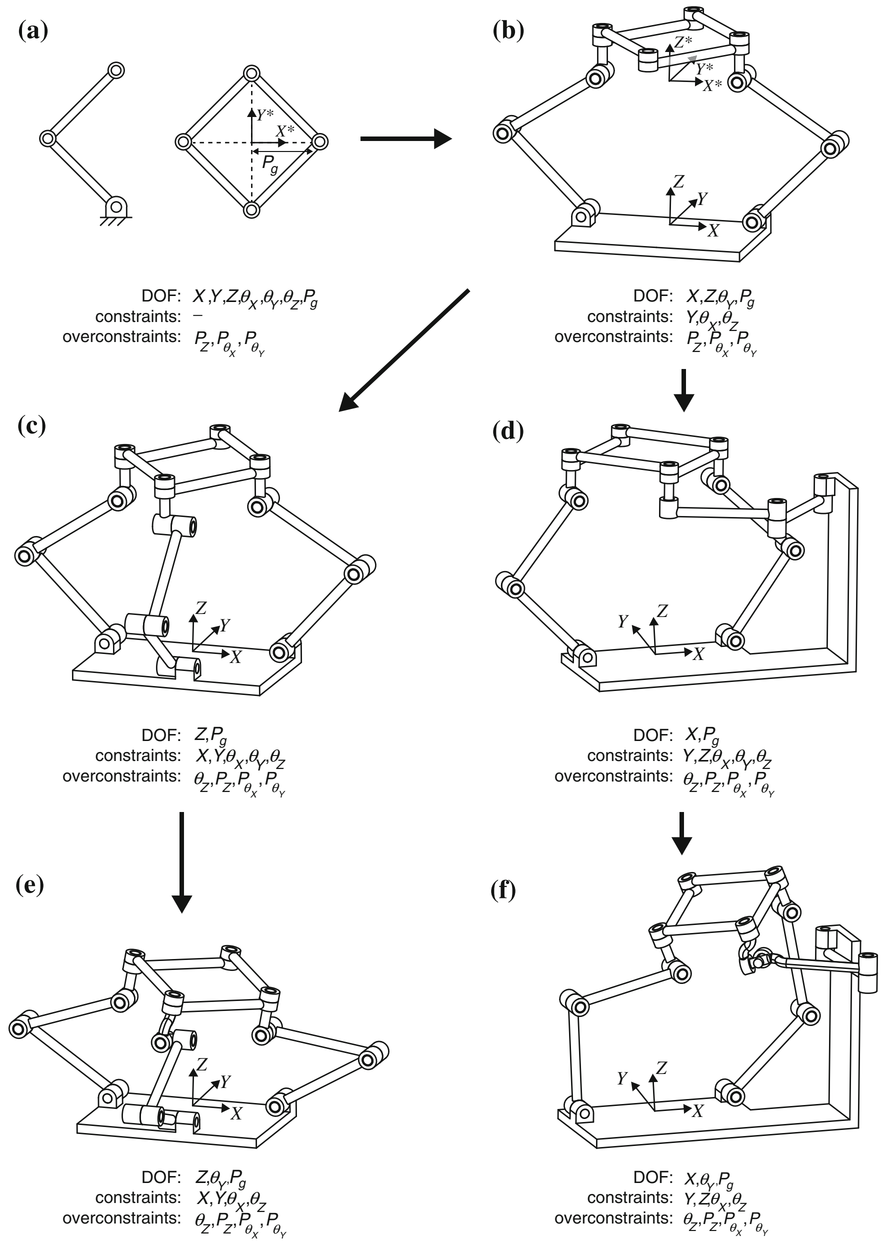

(a)
Y^*
X^*
P_g
DOF: $X,Y,Z,\theta_X,\theta_Y,\theta_Z,P_g$
constraints: –
overconstraints: $P_Z,P_{\theta_X},P_{\theta_Y}$
(b)
Z^*
Y^*
X^*
Z
Y
X
DOF: X,Z,θ_Y,P_g
constraints: Y,θ_X,θ_Z
overconstraints: $P_Z,P_{\theta_X},P_{\theta_Y}$
(c)
Z
Y
X
DOF: Z,P_g
constraints: $X,Y,\theta_X,\theta_Y,\theta_Z$
overconstraints: $\theta_Z,P_Z,P_{\theta_X},P_{\theta_Y}$
(d)
Y
Z
X
DOF: X,P_g
constraints: $Y,Z,\theta_X,\theta_Y,\theta_Z$
overconstraints: $\theta_Z,P_Z,P_{\theta_X},P_{\theta_Y}$
(e)
Z
Y
X
DOF: Z,θ_Y,P_g
constraints: X,Y,θ_X,θ_Z
overconstraints: $\theta_Z,P_Z,P_{\theta_X},P_{\theta_Y}$
(f)
Y
Z
X
DOF: X,θ_Y,P_g
constraints: $Y,Z\theta_X,\theta_Z$
overconstraints: $\theta_Z,P_Z,P_{\theta_X},P_{\theta_Y}$

Fig. 2 **a** The minimum building blocks for a fully parallel 3DOF PMCP, **b** the mechanism after connection of the first two legs, **c**, **d** the two possibilities for connecting the third leg, **e**, **f** the two resulting fully parallel 3DOF PMCPs

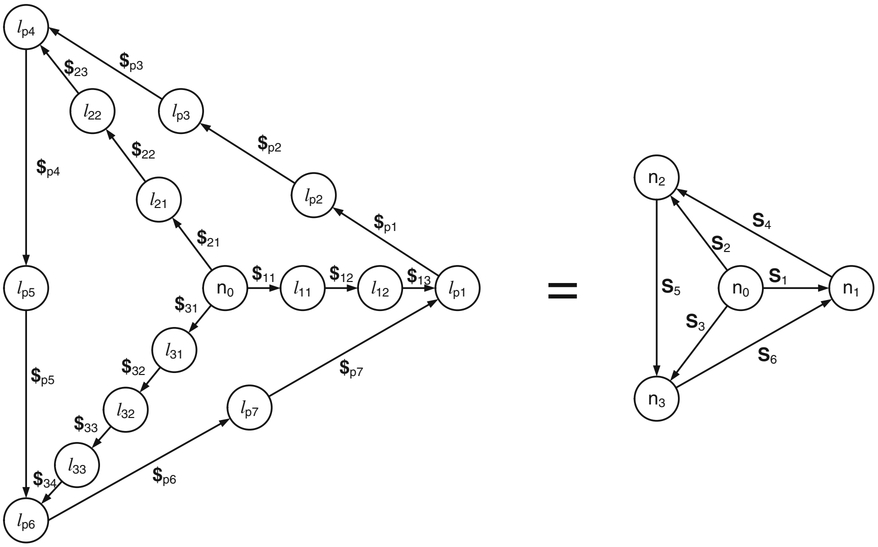


Fig. 3 Representation of the mechanisms shown in Fig. 2e and f using graph theory where l_{il} stands for the lth link of the leg i and $\$_{ij}$ for the screw associated with joint j of leg i, while in the reduced graph after serial reduction n_i and $\mathbf{S}_i$ are respectively the ith link node and screw system

This section has described the kinematic design of two fully parallel 3DOF PMCPs, both of which have four overconstraints. In graph theory notation, both architectures are represented by the graph in Fig. 3 which is equivalent to the one shown in Fig. 1b after serial reductions [5].

3 Derivation of Inverse Jacobian

In this section the inverse Jacobian is derived for the manipulator introduced in Fig. 2e, which is also shown in Fig. 4 including the notations that are used in this section. For the purpose of easy analysis, the end-effector forces acting in the direction of the manipulator DOF are here expressed as forces acting on two specific end-effector points (see Fig. 4). However, in practice the complete configurable platform may act as end-effector, for example when grasping a deformable object.

Before the inverse Jacobian is derived, it is first confirmed that the mobility M and overconstraints R_C presented in Fig. 2e are consistent with the Chebychev–Grübler–Kutzbach criterion. It was observed that both resulting designs have four overconstraints, $R_C = 4$. The links and joints can be counted easily using the graph theory representation in Fig. 3, which counts $n = 15$ links l_{ij} (of which link l_{33} has zero length) and $m = 17$ joints with an associated screw $\$_{ij}$. All joints have one DOF, so $f_m = 1$ for all m joints. Because in the original Chebychev–Grübler–Kutzbach

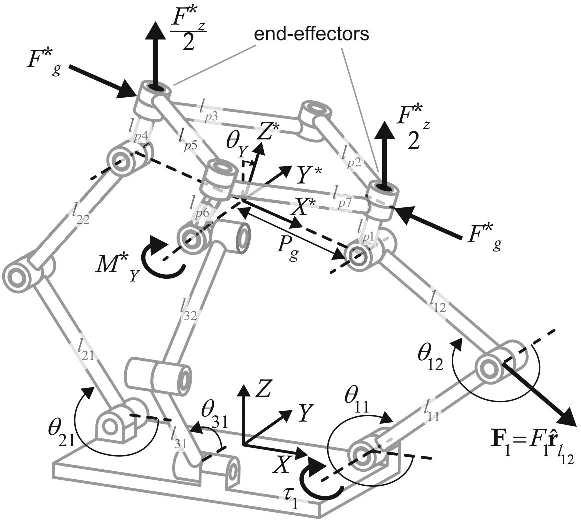


Fig. 4 3DOF PMCP that can independently translate in Z, rotate around Y^* and perform grasping by means of the local platform DOF P_g

criterion any overconstraints are included in the resulting mobility, the criterion is often rewritten to

$$M = 6(n - m - 1) + \sum_{i=1}^{m} (f_m) + R_C \tag{1}$$

which equals $M = 3$ if the above numbers are used. This is consistent with the expected mobility. The remainder of this section deals with the inverse Jacobian derivation for the mechanism shown in Fig. 4. More precisely, the transpose of the inverse Jacobian (J^{-T}) is derived, mapping the actuator torques on the end-effector forces according to

$$\mathbf{F}^* = J^{-T}\bar{\tau} \tag{2}$$

Although linear motors can also be used, the situation is here considered for rotary actuators located at the base. The torques τ_i applied by these actuators are transferred to the platform via forces $\mathbf{F}_i$ directed along links l_{i2}. Because all three legs are equal, $l_{11} = l_{21} = l_{31} = l_1$ and $\mathbf{F}_i$ can expressed using

$$\mathbf{F}_i = F_i\widehat{\mathbf{r}}_{l_{i2}} = \frac{\tau_i}{l_1 \sin(\theta_{i2})}\widehat{\mathbf{r}}_{l_{i2}} \tag{3}$$

These forces are transferred to the platform and can be expressed in terms of forces acting in the direction of the platform DOF. In this paper the internal platform DOF is considered a grasping motion with variable P_g [m], which is acted on by a force F^*_g. The other forces are M^*_Y around Y^* and F^*_Z in the Z-direction as indicated in

Fig. 4. With $l_{p2} = l_{p3} = l_{p5} = l_{p7} = l_p$, the expression of these forces in terms of forces $\mathbf{F}_i$ is

$$\begin{bmatrix} F^*{}_g \\ F^*{}_Z \\ M^*{}_Y \end{bmatrix} = \begin{bmatrix} -\cos(\theta_Y) & 0 & \sin(\theta_Y) \\ 0 & 0 & 1 \\ -P_g \sin(\theta_Y) & 0 & -P_g \cos(\theta_Y) \end{bmatrix} \mathbf{F}_1 + \begin{bmatrix} \cos(\theta_Y) & 0 & -\sin(\theta_Y) \\ 0 & 0 & 1 \\ P_g \sin(\theta_Y) & 0 & P_g \cos(\theta_Y) \end{bmatrix} \mathbf{F}_2 + \begin{bmatrix} 0 & -P_g \sqrt{\left(l_p{}^2 - P_g{}^2\right)} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{F}_3 \tag{4}$$

By combining Eqs. 3 and 4 the platform forces can be directly expressed as a function of the actuator torques. For the manipulator shown in Fig. 4 this results in

$$\begin{bmatrix} F^*{}_g \\ F^*{}_Z \\ M^*{}_Y \end{bmatrix} = \frac{1}{l_1} \begin{bmatrix} \frac{-\cos(\theta_Y - \theta_{11} - \theta_{12})}{\sin(\theta_{12})} & \frac{\cos(\theta_Y - \theta_{21} - \theta_{22})}{\sin(\theta_{22})} & \frac{-P_g \cos(\theta_{31} + \theta_{32})}{\sin(\theta_{32}) \sqrt{\left(l_p{}^2 - P_g{}^2\right)}} \\ \frac{-\sin(\theta_{11} + \theta_{12})}{\sin(\theta_{12})} & \frac{-\sin(\theta_{21} + \theta_{22})}{\sin(\theta_{22})} & \frac{\sin(\theta_{31} + \theta_{32})}{\sin(\theta_{32})} \\ \frac{-P_g \sin(\theta_Y - \theta_{11} - \theta_{12})}{\sin(\theta_{12})} & \frac{P_g \sin(\theta_Y - \theta_{21} - \theta_{22})}{\sin(\theta_{22})} & 0 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{5}$$

which is the expression of Eq. 2 for the manipulator shown in Fig. 4. Because of the power conservation principle the matrix J^{-T} in Eq. 5 can also be used in the velocity relation $\dot{\mathbf{q}} = J^{-1}\dot{\mathbf{x}}$ between the actuator velocities $\dot{\mathbf{q}}$ and the platform velocities $\dot{\mathbf{x}}$. Finally, the matrix J^{-T} can be analysed to reveal some of the characteristics of the developed manipulator, because in singular configurations the rank of J^{-T} reduces. For the manipulator presented in Fig. 4 singularities occur if

- the distance between the end-effectors is zero, $P_g = 0$
- the distance between the end-effectors is maximal, $P_g = l_p$
- one of the legs is completely extended or folded, $\theta_{i2} = \{0, \pi, ..\}$
- for both leg one and leg two, link l_{i2} is in line with the platform, $\theta_Y - \theta_{i1} - \theta_{i2} = \{0, \pi, ..\}$ for $i = \{1, 2\}$

4 Conclusion

This paper has presented the first kinematic designs of fully parallel 3DOF PMCPs, which were identified as an elementary subset of PMCP designs. The resulting mechanisms are spatial manipulators that can be independently controlled in one rotation, one translation and one internal platform motion. For one of the introduced 3DOF PMCPs the inverse Jacobian was derived, which has also allowed the identification of four singular configurations. Since existing type synthesis methods do not cover PMCPs, this paper has applied a structured, but not yet formalised, design method. Because the presented manipulators are considered to be part of an elementary subset of PMCP designs, they may prove to be useful input for the development of a type synthesis method that does cover PMCPs.

Acknowledgments This research was financially supported by the Dutch Technology Foundation STW (project number 12158).

References

1. Earl, C.F., Rooney, J.: Some kinematic structures for robot manipulator designs. J. Mech. Transm. Autom. Des. **105**(1), 15 (1983). doi:10.1115/1.3267337
2. Gogu, G.: Structural Synthesis of Parallel Robots, Part 1: Methodology. Springer, Dordrecht (2008)
3. Kong, X., Gosselin, C.: Type Synthesis of Parallel Mechanisms. Springer, Berlin (2007)
4. Kumar, V.: Instantaneous kinematics of parallel-chain robotic mechanisms. J. Mech. Des. **114**(3), 349 (1992). doi:10.1115/1.2926560
5. Lambert, P., Herder, J.L.: Mobility analysis of non series-parallel mechanisms. In: Viadero, F., Ceccarelli, M. (eds.) New Trends in Mechanism and Machine Science, Mechanisms and Machine Science, vol. 7, pp. 63–71. Springer, Dordrecht (2013). doi:10.1007/978-94-007-4902-3
6. Lambert, P., Langen, H., Munnig Schmidt, R.: A novel 5 DOF fully parallel robot combining 3T1R motion and grasping. In: ASME 34th Annual Mechanisms and Robotics Conference, Parts A and B, vol 2, pp. 1123–1130, (2010). doi:10.1115/DETC2010-28676
7. Merlet, J.P.: Parallel robots, 2nd edn. Solid Mechanics and its Application. Springer, New York (2006)
8. Mohamed, M., Gosselin, C.: Design and analysis of kinematically redundant parallel manipulators with configurable platforms. IEEE Trans. Rob. |textbf21(3), 277–287 (2005). doi:10.1109/TRO.2004.837234
9. Nabat, V., de la O Rodriguez, M., Company, O., Krut, S., Pierrot, F.: Par4: very high speed parallel robot for pick-and-place. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 553–558. IEEE (2005). doi:10.1109/IROS.2005.1545143
10. Yi, B.J., Hong, Y.S., Oh, S.R.: Design of a parallel-type gripper mechanism. Int. J. Rob. Res. **21**(7), 661–676 (2002). doi:10.1177/027836402322023240

Single Exponential Motion and Its Kinematic Generators

Guanfeng Liu, Yuanqin Wu and Xin Chen

Abstract Both constant velocity (CV) joints and zero-torsion parallel kinematic machines (PKMs) possess special geometries in their subchains. They are studied as two different subjects in the past literature. In this paper we provide an alternative analysis method based on the symmetric product on $SE(3)$ (the Special Euclidean group). Under this theoretical framework CV joints and zero-torsion mechanisms are unified into *single exponential motion generators* (SEMG). The properties of single exponential motion are studied and sufficient conditions are derived for the arrangement of joint screws of a serial chain so that the motion pattern of the resulting mechanism is indeed a single exponential motion generator.

Keywords Constant velocity transmission · Zero torsion · Symmetric product · Single exponential motion generator

1 Introduction

Constant velocity (CV) joints have found applications in a variety of domains, ranging from car drive chains to rotation transmissions in DELTA parallel robot. They have received great research interests from the robotics and mechanism community since 1970s. Hunt [7] developed a general theory for analysis and synthesis of CV joints using screw theory . He found that CV couplings could be realized by kinematic chains with special geometry that their joint axes form a symmetric arrangement with

G. Liu (✉) · X. Chen · Y. Wu
Guangdong University of Technology, Guangdong, China
e-mail: guanfeng.liu@innovationteam.cn

X. Chen
e-mail: chenx@gdut.edu.cn

Y. Wu
Hong Kong University of Science and Technology, Hong Kong, China
e-mail: troy.wu@gmail.com

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_36,

respect to a plane. Carricato [4] examined the computational detail of three different types of CV couplings: (i) $\mathcal{U}-\mathcal{U}$; (ii) $\mathcal{R}-\mathcal{S}-\mathcal{R}$ where $\mathcal{R}$ stands for revolute joint, and $\mathcal{S}$ for spherical or ball joint; (iii) $\mathcal{R}-\mathcal{P}L-\mathcal{R}$ where $\mathcal{P}L$ denotes planar gliding joint. He also showed the important roles of CV couplings in the construction of close-chain orientational manipulators with simple and diagonal velocity Jacobian, for which he coined the term *decoupled and homokinemtic transmission*.

Recently, not only the spatial structure of the CV joints but also their motion pattern,[1] usually described by the set of motions of the output shaft with respect to the input shaft, received the attention of robotics researchers. Bonev [3] proposed a modified Euler angle parametrization, the tilt and torsion angle, for studying a class of CV joints with PKM structures. He noticed that the torsion angles for these mechanisms are always zero (hence the name zero-torsion mechanisms), and showed that their forward kinematics map as well as their singularity loci have closed form [1, 3]. Zero-torsion property seems a more general concept than CV coupling although the latter necessarily implies the former. Besides the CV joints, there are different examples exhibiting zero-torsionness. The first example comes from the study of humuan eye movement. Donders (1848) first noticed that human eyes only have 2 DoFs because its orientation is uniquely determined by the line of sight [6]. This 2-DoF motion is zero-torsion because its instantaneous velocity satisfies the Listing's law,[2] [6]. Another example is human shoulder, whose motion pattern is not simply a ball-in-socket joint. Rosheim [9] noticed that human should should be modeled, instead of a ball-in-socket joint, as an omni-wrist, which employs a parallel kinematic structure with four identical $\mathcal{U}-\mathcal{U}$ subchains, where $\mathcal{U}$ stands for universal joints. This omni-wrist uses CV couplings, and are therefore zero-torsion.

The goal of this paper is to extend the theory about CV joints and zero-torsion PKMs with the purpose to put them in a unified theoretical framework, and develop tools for analyzing high-dimensional counterparts. First we found that the symmetric arrangement of joint screws of a serial chain implies a symmetric product of screw motions in its forward kinematics. Then we show that except at singularities symmetric products of screw motions for twists in some special classes of subspaces of the Lie algebra $se(3)$ could be turned into a single exponential on $SE(3)$. Finally we show the sufficient conditions for a serial chain being a single exponential motion generator.

2 Exponential Map, POE, and Zero-Torsion Mechanisms

It is well known that the Special Euclidean group $SE(3)$ is a 6-dimensional Lie group. It could be used to describe the relative position and orientation of the end-effector of a robot with respective to a fixed world frame. The tangent space $T_eSE(3)$ of $SE(3)$ at the identity element e consists of the set of feasible twists of the end-effector.

[1] Sometimes motion pattern is also called motion type.

[2] The Listing's law about human eye movement, also called *the half-angle law*, states that the instantaneous velocity plane tilts exactly one half of that of the line of sight.

$T_eSE(3)$ satisfies the conditions of being a Lie algebra, and is often denoted as $se(3)$. The exponential map:

$$\exp : se(3) \mapsto SE(3) : \hat{\xi} \mapsto e^{\hat{\xi}} \tag{1}$$

is a surjective map and gives a screw motion on $SE(3)$ [8]. The forward kinematics map of a serial articulated chain of lower pairs is given by the product of exponentials (POE) formula [8]

$$g_{wt} = e^{\hat{\xi}_1\theta_1} \cdot e^{\hat{\xi}_2\theta_2} \cdots e^{\hat{\xi}_k\theta_k} \tag{2}$$

where $\hat{\xi}_i \in se(3)$ is the twist for joint i, and θ_i the corresponding joint angle. In other words the kinematic map (2) is a cascaded composition of screw motions. The definition for a zero-torsion mechanism is originally based on the formulation of the tilt-and-torsion parametrization of robot orientation R [2],

$$R = e^{\hat{\omega}\theta} \cdot e^{\hat{n}\alpha}. \tag{3}$$

where $e^{\hat{\omega}\theta}$ denotes a rotation of θ about the axis $\omega \in \mathbb{R}^3$, again in terms of the exponential on the rotation group $SO(3)$. ω lies in a plane with the normal vector n. θ and α are referred to as the tilt and torsion angles respectively. The set of rotations is zero-torsion if $\alpha \equiv 0$, i.e., the orientation set is described by a single exponential. For a 2-DoF orientational serial manipulator with two perpendicular joint axes, its torsion angle is obviously not always zero, but θ_2 as seen from its forward kinematic map

$$R_{wt} = e^{\hat{\omega}_1\theta_1} \cdot e^{\hat{\omega}_2\theta_2}. \tag{4}$$

In fact we could make the same conclusion as long as the two axes ω_1 and ω_2 are not parallel.[3] Although zero-torsionness is a concept originally defined for 2-DoF orientational mechanisms, the idea of using single exponential formulation (3) could be generalized and applied to manipulators with combined translational and rotational motion.

3 Single Exponential Motion and Its Kinematic Generators

Consider the set of motions defined by

$$e^{\Omega} \subset SE(3) \tag{5}$$

[3] According to the Baker-Cambell-Hausdoff formula, we have

$$e^{\hat{\omega}_1\theta_1} e^{\hat{\omega}_2\theta_2} = e^{\hat{\omega}_1\theta_1 + \hat{\omega}_2\theta_2 + \frac{1}{2}[\hat{\omega}_1, \hat{\omega}_2]\theta_1\theta_2 + O(\theta_1^2, \theta_2^2)},$$

which is not a single exponential of a twist in the plane $\{\hat{\omega}_1, \hat{\omega}_2\}$, but a twist in the three-dimensional Lie algebra $\{\hat{\omega}_1, \hat{\omega}_2, [\hat{\omega}_1, \hat{\omega}_2]\}$, which is the Lie algebra $so(3)$ of the rotation group $SO(3)$.

where Ω is a linear subspace of $se(3)$. It is trivial when Ω is a Lie subalgebra of $se(3)$ as e^{Ω} will be simply a Lie subgroup in this case. In other cases e^{Ω} may or may not be a submanifold depending on whether Ω satisfies some conditions.

3.1 Lie Triple System

Definition 3.1 *A linear vector subspace Ω of $se(3)$ (not necessarily a Lie subalgebra) is said to be a Lie triple system (LTS) if it is closed under the double Lie bracket* $[,[,]]$:[4]

$$\forall \hat{u}, \hat{v}, \hat{w} \in \Omega, [\hat{u}, [\hat{v}, \hat{w}]] \in \Omega \tag{6}$$

The Lie subalgebras of $se(3)$ are trivial examples of LTS. But there are nontrivial examples. Let $\{\hat{e}_1, \cdots, \hat{e}_6\}$ be the canonical basis of $se(3)$.

Example 3.1 (The Instantaneous 2R Motion Type) The subspace $\Omega_{2R} := \{\hat{e}_4, \hat{e}_5\}$ is a linear combination of the instantaneous rotations about the x and y axis. This subspace is a LTS, but not a Lie subalgebra of $se(3)$.

Example 3.2 (The Instantaneous 1T2R Motion Type) Consider adding an instantaneous translational DoF $\hat{e}_3$ into Ω_{2R} so a new subspace Ω_{1T2R} forms

$$\Omega_{1T2R} := \{\hat{e}_3, \hat{e}_4, \hat{e}_5\}.$$

Ω_{1T2R} is a LTS by verifying that its basis indeed satisfies (6)

$$\begin{aligned}
&[\hat{e}_3, [\hat{e}_4, \hat{e}_5]] = 0, [\hat{e}_4, [\hat{e}_3, \hat{e}_4]] = \hat{e}_3 \\
&[\hat{e}_4, [\hat{e}_3, \hat{e}_5]] = 0, [\hat{e}_5, [\hat{e}_3, \hat{e}_4]] = 0 \\
&[\hat{e}_5, [\hat{e}_3, \hat{e}_5]] = \hat{e}_3
\end{aligned}$$

We have the following main theorem[5] regarding the manifold property of the motion set (5).

Theorem 3.1 *If $\Omega \subset se(3)$ is a Lie triple system, e^{Ω} is a differential submanifold of $SE(3)$, and is referred to as a single exponential submanifold (SES).*

Example 3.3 (The 2R and 1T2R Motion Pattern). Since Ω_{2R} and Ω_{1T2R} are LTS as proved in the previous examples, both $e^{\Omega_{2R}}$ and $e^{\Omega_{1T2R}}$ are SES. The former is exactly the set of motions generated by a 2-DoF orientational manipulator with zero torsion angles [3].

[4] $[,[,]]$ could be replaced by $[[,],]$ based on the Jacobian identity on any Lie algebra.

[5] Its proof can be found in [5] (Theorem 7.2, Chap. 4)

3.2 Symmetric Products

Consider a SES e^{Ω} for some LTS $\Omega \subset se(3)$ of dimension $k < 6$. We have the following main result about SES

Proposition 3.1 (symmetric product) *Let $\hat{u}_1, \hat{u}_2$ be two vectors in a LTS Ω, then*

$$e^{\hat{u}_1\theta_1} e^{\hat{u}_2\theta_2} e^{\hat{u}_1\theta_1} \in e^{\Omega}, \quad \forall \theta_1, \theta_2 \in \mathbb{R}. \tag{7}$$

$e^{\hat{u}_1\theta_1} e^{\hat{u}_2\theta_2} e^{\hat{u}_1\theta_1}$ is called a symmetric product of $e^{\hat{u}_2\theta_2}$ by $e^{\hat{u}_1\theta_1}$.

Readers are referred to [5] (Chap. 4 and its exercises) for the proof. Proposition 3.1 only states that symmetric products of screw motions in a SES remain on it. It is not known if the resulting composition of screw motions indeed generates the desired SES.

Proposition 3.2 (Single Exponential Motion Generator) *Let $\hat{u}_1, \cdots, \hat{u}_k$ be the basis of the LTS Ω, then the k-layer symmetric products*

$$e^{\hat{u}_1\theta_1} \ldots e^{\hat{u}_{k-1}\theta_{k-1}} e^{\hat{u}_k\theta_k} e^{\hat{u}_{k-1}\theta_{k-1}} \ldots e^{\hat{u}_1\theta_1} \in e^{\Omega}, \quad \forall \theta_1, \cdots, \theta_k \in \mathbb{R}. \tag{8}$$

generates the SES e^{Ω} if the Jacobian of (8) is non-singular.

The proof of this proposition could be deduced using the implicit function theorem. Eq. (8) is a k-layer symmetric product. In fact symmetric products with more than k layers also generate the same SES as long as the set of independent twists forms a basis of Ω and the Jacobian is non-singular.

3.3 Single Exponential Motion Generators

SES are special subsets of $SE(3)$ which, to the best of our knowledge, have not been sufficiently studied by robotics researchers. It is of natural interest to find their kinematic generators, i.e., mechanisms whose task space matches the given SES. Hence the name single exponential motion generator (SEMG) follows. The 2-DoF orientational PKM with zero torsion angles in [3] and the omni-wrist [9] are example SEMG of $e^{\Omega_{2R}}$. Both of them employ CV couplings in their kinematic structure. Here we use the method of symmetric products to derive the sufficient conditions for a serial chain being a SEMG.

Now consider a serial chain with a generic forward kinematic map (2). According to Proposition 3.2, this chain generates some SES of dimension k if it has the form of multi-layer symmetric products and moreover its Jacobian is non-singular and has rank k. The twist coordinate ξ_i describes the spatial location of the joint axis i at a given configuration. It is given by a rigid displacement (a cascading of rigid motions generated by all previous joints) of the corresponding initial twist ζ_i.

$$\begin{aligned}\xi_1 &= Ad_{g_0}\zeta_1\\ \xi_2 &= Ad_{g_0 g_1}\zeta_2\\ \vdots &= \quad \vdots\\ \xi_k &= Ad_{g_0 g_1 \cdots g_{k-1}}\zeta_k\end{aligned}$$

where $g_i = e^{\hat{\zeta}_i \alpha_i}$, $\alpha_i \in \mathbb{R}$, $i = 1, \cdots, k$, and Ad_g denotes the adjoint map of an element $g \in SE(3)$. Without loss of generality g_0 could be chosen to be the identity $e \in SE(3)$. Substituting them into (2) yields

$$g_{wt} = g_0 e^{\hat{\zeta}_1\theta_1} g_1 \cdots g_{k-2} e^{\hat{\zeta}_{k-1}\theta_{k-1}} g_{k-1} e^{\hat{\zeta}_k\theta_k} g_{k-1}^{-1} \cdots g_0^{-1}.$$

Lemma 3.1 *Equation (2) is a symmetric product if and only if*

$$g_0 = g_{k-1}^{-1} g_{k-2}^{-1} \cdots g_0^{-1} \tag{9}$$
$$\zeta_i = \zeta_{k+1-i}, \quad i = 1, \cdots, k \tag{10}$$
$$\theta_i = \theta_{k+1-i} \tag{11}$$
$$g_i = g_{k-i}, \quad i = 1, \cdots, k, \ \square. \tag{12}$$

Employing Eq. (9)–(12), the twists in (2) are calculated as

$$\xi_1 = Ad_{g_0}\zeta_1 \tag{13}$$
$$\xi_2 = Ad_{g_0 g_1}\zeta_2 \tag{14}$$
$$\vdots = \vdots \tag{15}$$
$$\xi_m = Ad_{g_0 g_1 \cdots g_{m-1}}\zeta_m \tag{16}$$
$$\xi_{m+1} = Ad_{g_0 g_1 \cdots g_m}\zeta_{m+1} \tag{17}$$
$$\xi_{m+2} = Ad_{g_0^{-1} g_1^{-1} \cdots g_{k-m-2}^{-1}}\zeta_{k-m-1} \tag{18}$$
$$\vdots = \vdots \tag{19}$$
$$\xi_{k-1} = Ad_{g_0^{-1} g_1^{-1}}\zeta_2 \tag{20}$$
$$\xi_k = Ad_{g_0^{-1}}\zeta_1 \tag{21}$$

where $m = \lfloor k/2 \rfloor$ denotes the greatest integer less than or equal to $k/2$. The set of consecutive twists $\{\xi_1, \cdots, \xi_k\}$ forms a special arrangement because there is a kind of symmetry between pairs of twists, (ξ_i, ξ_{k+1-i}) $(i = 1, \cdots, k)$. Moreover the instantaneous velocity space of the chain is calculated as $W := \{W_1, W_2, \cdots\}$, where

$$\begin{aligned} W_1 &= \xi_1 + \xi_k = (Ad_{g_0} + Ad_{g_0^{-1}})\zeta_1 \\ W_2 &= \xi_2 + \xi_{k-1} = (Ad_{g_0 g_1} + Ad_{g_0^{-1} g_1^{-1}})\zeta_2 \\ \vdots &= \qquad\qquad\qquad \vdots \end{aligned}$$

The dimension of W is either m (if k is even) or $m+1$ (if k is odd). ξ_i and ξ_{k+1-i} $(i = 1, \cdots, m)$ are symmetric with respect to the hyperplane $W \subset se(3)$, and $\xi_{m+1} \in W$ in case that m is odd. This result could be considered as a generalization of Hunt's theory about CV couplings where the joint axes are required to be symmetric about a plane in $\mathbb{R}^3$, while here the symmetry is with respect to a hyperplane in $se(3)$. In this paper we call the latter symmetry as mirror symmetry.

Example 3.4 (Mirror symmetry of $e^{\Omega_{1T2R}}$ generators) According to Lemma 3.1, a serial-chain generator of $e^{\Omega_{1T2R}} = \{e_3, e_4, e_5\}$ may consist of five joints given by:

$$\begin{aligned} \xi_1 &= Ad_{g_0}\zeta_1, \\ \xi_2 &= Ad_{g_0 g_1}\zeta_2, \\ \xi_3 &= \zeta_3, \\ \xi_4 &= Ad_{g_0^{-1} g_1^{-1}}\zeta_2, \\ \xi_5 &= Ad_{g_0^{-1}}\zeta_1 \\ g_0 &= e^{\hat{\eta}_0 \alpha_0}, g_1 = e^{\hat{\eta}_1 \alpha_1}, \eta_i, \zeta_j \in \Omega_{1T2R}. \end{aligned}$$

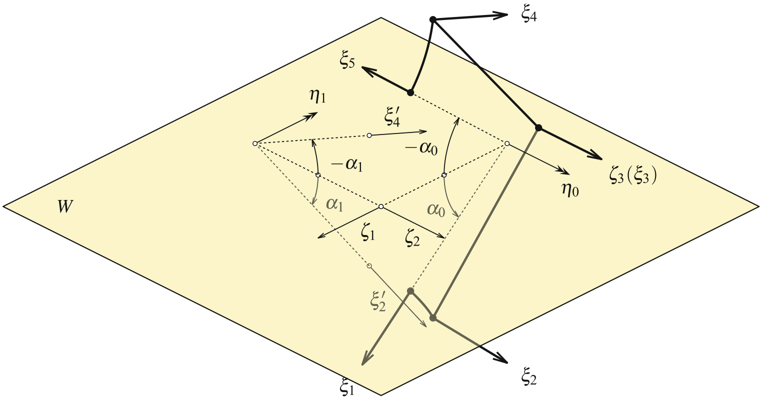


Fig. 1 Mirror symmetry of a $e^{\Omega_{1T2R}}$ subchain: $\xi_2' = Ad_{g_1}\zeta_2$; $\xi_4' = Ad_{g_1^{-1}}\zeta_2$; $\xi_2 = Ad_{g_0}\xi_2' = Ad_{g_0 g_1}\zeta_2$; $\xi_4 = Ad_{g_0^{-1}}\xi_4' = Ad_{g_0^{-1} g_1^{-1}}\zeta_2$; $\xi_1 = Ad_{g_0}\zeta_1$; $\xi_5 = Ad_{g_0^{-1}}\zeta_1$; $\xi_3 = \zeta_3$ $(g_0 = e^{\hat{\eta}_0 \alpha_0}$; $g_1 = e^{\hat{\eta}_1 \alpha_1})$

Notice that we could simply let the initial set of twists be $\zeta_1 = e_4$, $\zeta_2 = e_5$, and $\zeta_3 = e_3$, and then applying rigid displacements g_0 and g_1 yields a new set of mirror symmetric twists $\{\xi_i\}$, as shown in Fig. 1.

Finally we have the following theorem for SEMG

Theorem 3.2 *Let e^{Ω} be an m-dimensional SES. A serial chain consisting of $k = 2m$ or $2m + 1$ joints is a SEMG of e^{Ω} if there exists a configuration $(\theta_1, \cdots, \theta_k)$ at which the set of screws $\{\xi_1, \cdots, \xi_k\}$ is mirror symmetric about $\Omega \subset se(3)$ and the space W spanned by $\{\xi_1, \cdots, \xi_k\}$ satisfies $W = \Omega$, and moreover $\theta_i = \theta_{k+1-i}$, $i = 1, \cdots, m$ is kept valid by imposing suitable constraints (usually by forming closed-loops).*

4 Conclusion

In this paper we generalize the previous results about CV coupling and zero-torsion mechanisms and unify them into single exponential motion generators. We develop the tool of symmetric product and Lie Triple System for analyzing the properties of single exponential submanifolds, and use them to derive the sufficient conditions for single exponential motion generators. Examples are worked out to verify the developed theories.

Acknowledgments This research is supported by Talents Introduction Startup Funds of High Education of Guangdong Province (2050205) and supported by 1000 Young Investigator Plan of the Chinese Government.

References

1. Bonev, I.A.: Direct kinematics of zero-torsion parallel mechanisms. In: Proceedings – IEEE International Conference on Robotics and Automation, 3851–3856 (2008)
2. Bonev, I.A., Ryu, J.: New approach to orientation workspace analysis of 6-dof parallel manipulators. Mech. Mach. Theory **36**(1), 15–28 (2001)
3. Bonev, I.A., Zlatanov, D., Gosselin, C.M.: Advantages of the modified euler angles in the design and control of pkms. In: 2002 Parallel Kinematic Machines International Conference, 171–188 (2002)
4. Carricato, M.: Decoupled and homokinetic transmission of rotational motion via constant-velocity joints in closed-chain orientational manipulators. J. Mech. Robot. **1**(4), 1–14 (2009)
5. Helgason, S.: Differential Geometry, Lie Groups, and Symmetric Spaces, vol. 80. Academic press, New York (1978)
6. Henn, V.: Three-dimensional Kinematics of Eye, Head and Limb Movements, Chap. History of Three-Dimensional Eye Movement Research, pp. 3–14. Harwood Academic Publishers, Australia (1997)
7. Hunt, K.: Constant-velocity shaft couplings: a general theory. ASME J. Eng. Ind **95**(2), 455–464 (1973)
8. Li, Z., Murray, R., Sastry, S.: A Mathematical Introduction to Robotic Manipulation. CRC Press Boca Raton, Florida, USA (1994)
9. Rosheim, M.E.: Leonardo's Lost Robots. Springer, Heidelberg (2006)

Foundations for the Approximate Synthesis of RCCC Motion Generators

Jorge Angeles

Abstract The approximate synthesis of RCCC linkages for motion generation, a.k.a. rigid-body guidance, is the subject of this paper. A formulation is proposed here based on dual algebra, thereby leading to a *dual, constrained, nonlinear least-square problem*. The dual normality conditions necessary to obtain a feasible least-square approximation are established, following which an algorithm for the solution of the problem is proposed.

Keywords Spatial Burmester problem · Approximate rigid-body guidance · Dual normality conditions · Dual · Constrained · Nonlinear least squares

1 Introduction

The general problem of linkage synthesis consists in finding the dimensions of a linkage of a given topology—number of links, number of joints, types of joints, and number of kinematic loops—for a designated task. In this paper the task of interest is rigid-body guidance,as defined by Ludwig Burmester (1840–1927) [1] for the planar case, for which reason the problem is also named after Burmester. It is known that the planar and spherical Burmester problems allow for the synthesis of a four-bar linkage to guide their coupler link through up to five prescribed poses. For the spatial case, the four-bar linkage becomes of the RCCC type, where R stands for revolute, C for cylindrical joint. The linkage is usually synthesized via its two defining *dyads*, RC and CC; then, of the multiple solutions obtained for each dyad, one RCCC linkage is assembled upon coupling the dyads by means of the *coupler link*. Now, the number of parameters that determine a dyad as well as the number of constraints that each dyad type must satisfy are different for each of the two foregoing dyads. The maximum number of coupler poses that each dyad can visit *exactly* is five for the CC dyad, three

J. Angeles (✉)
McGill University, Montreal, Canada
e-mail: angeles@cim.mcgill.ca

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_37,

for the RC (or CR) dyad [2]. Apparently, the maximum number of poses that a RCCC linkage can visit exactly is three, which is rather limited. However, if a condition is imposed that leads to a coupling of the two dyads, e.g., robustness to variations in the selection of the intermediate poses, as reported in [3], then a maximum of four poses can be visited with the RCCC linkage. The number of poses that can be met exactly is thus still limited, whence the motivation behind this paper.

In practice it is seldom required that intermediate poses be visited exactly. For example, if the linkage under design is to be used to deploy and retract an aircraft landing gear, only the deployed and the retracted poses of the wheel are to be met exactly, the intermediate ones being free to deviate from a prescribed trajectory, in pose space, as long as the deviations are within reasonable, prescribed limits and the various moving links do not collide with the fuselage or between themselves. It is thus apparent that the intermediate poses can be visited approximately, thereby allowing for *approximate synthesis*, the subject of the paper.

Approximate linkage synthesis is a classical subject, treated in some books [4–6], that has been approached as a problem of least squares.

2 Problem Formulation

The linkage under synthesis bears the generic geometry depicted in Fig. 1, where link 1 is fixed, link 3 is the coupler, to which a frame $\mathcal{F}\ \{X,\ Y,\ Z\}$ has been attached with origin at point R, while links 2 and 4 are coupled to link 1 by means of a R

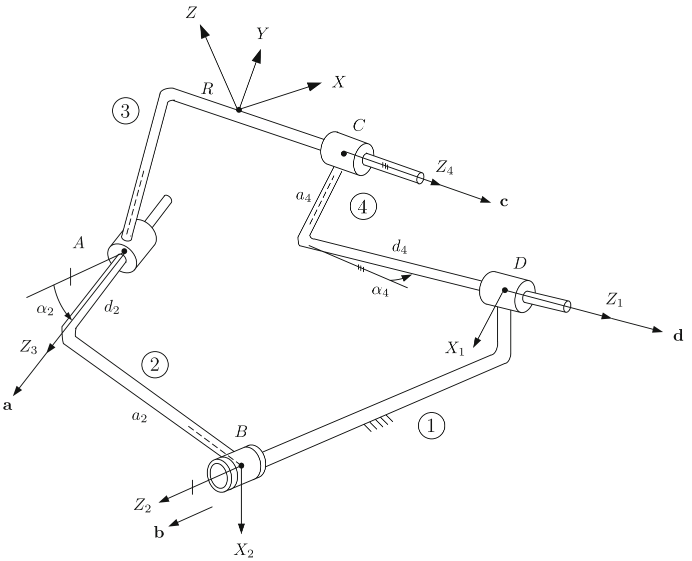


Fig. 1 A generic RCCC linkage

and a C joint, respectively, to link 3 by means of C joints. Moreover, links 2 and 4 are the RC and the CC dyads, respectively. Thus, axes Z_1 and Z_2 play the role of the center points, while axes Z_3 and Z_4 those of the circle points of the planar motion generator [7]. Apparently, axes Z_1 and Z_2 are grounded, and hence, remain stationary during the linkage motion, their counterparts Z_3 and Z_4 becoming lines of corresponding hyperboloids of revolution of axes Z_1 and Z_2, respectively. As a matter of fact, axes Z_3 and Z_4 become generators of the hyperboloids, which, in the general case, of arbitrary—not axially symmetric—single-sheet hyperboloids, are *reguli* of these surfaces. For these reasons, Z_1 and Z_2 will be termed the axis lines, or $\mathcal{A}$-lines, Z_3 and Z_4 the regulus lines, or $\mathcal{R}$-lines of the dyads under synthesis. For simplicity of notation, axes $Z_1, \ldots, Z_4$ will be denoted henceforth $\mathcal{B}, \mathcal{A}, \mathcal{C}, \mathcal{D}$, respectively. Moreover, $\mathcal{A}_0$ and $\mathcal{C}_0$ denote axes $\mathcal{A}$ and $\mathcal{C}$ at the reference pose of $\mathcal{F}$, with $\mathcal{A}_j$ and $\mathcal{C}_j$ denoting the location of $\mathcal{A}$ and $\mathcal{C}$ when $\mathcal{F}$ finds itself at its jth pose, for $j = 1, \ldots, m$.

Moreover, the motion under study is described by a set of poses $\mathcal{P} = \{\,\mathbf{r}_j,\ \mathbf{Q}_j\,\}_1^m$, where $\mathbf{r}_j$ is the position vector of R_j and $\mathbf{Q}_j$ is the orthogonal matrix that rotates frame $\mathcal{F}$ from its reference pose with origin at R_0 and orientation $\mathbf{Q}_0 \equiv \mathbf{1}$ to its jth pose. The purpose of linkage synthesis for motion generation in the case at hand consists in finding lines $\mathcal{A}_0$, $\mathcal{B}$, $\mathcal{C}_0$ and $\mathcal{D}$ that define completely the RCCC linkage, so that $\mathcal{F}$ will visit the set $\mathcal{P}$ with a *minimum error*. A word of caution is in order: vectors $\mathbf{r}_j$ having units of length and matrix $\mathbf{Q}_j$ being nondimensional, the error in missing a prescribed pose cannot be defined. In planar-linkage synthesis, the error is measured indirectly, in terms of the deviations of the various locations of the circle points from lying in a circle with center at one center point. In the same vein, the error in this case is measured as the distance of one $\mathcal{R}$-line from its corresponding axially symmetric hyperboloid. The error will be measured by mimicking exactly what is done in planar linkage synthesis: first find the circle that best fits a set of m center-point locations in the least-square sense; then, find the minimum distance of the putative center point in question to the circle, which is measured along the line that joins the putative circle point with the "center" point. In the spatial case under study, the distance from the $\mathcal{R}$-line to its corresponding $\mathcal{A}$-line consists of two items, the length of the segment of the common normal between $\mathcal{A}$ and $\mathcal{R}$ and the angle between the two lines, which can best be described by means of *dual algebra* [8]: Let $\mathbf{a}, \ldots, \mathbf{d}$ denote the unit vectors parallel to axes $\mathcal{A}, \ldots, \mathcal{D}$, respectively, the *moments*[1] of axes $\mathcal{A}, \ldots, \mathcal{D}$ being denoted by $\mathbf{a}_o, \ldots, \mathbf{d}_o$.

The problem can now be stated as: *Given the set $\mathcal{P}$ of $m > 4$ poses that the coupler link of a RCCC linkage is to visit, find lines $\mathcal{A}_0$, $\mathcal{B}$, $\mathcal{C}_0$ and $\mathcal{D}$ that define the RCCC linkage that carries its coupler link through the set $\mathcal{P}$ with a minimum error in the least-square sense.*

In order to formulate the problem, dual algebra [9, 10] is invoked. A dual unit vector $\hat{\mathbf{l}} = \mathbf{l} + \varepsilon\,\mathbf{l}_o$ represents a line $\mathcal{L}$ of direction given by the unit vector $\mathbf{l}$ and of

[1] The moment of a line in a given coordinate frame is defined as the cross product of the position vector of *any* point of the line times the unit vector parallel to the line. The mechanical interpretation of this concept is the moment of a unit force whose line of action is the line at stake.

moment $\mathbf{l}_o$ with respect to the origin. In this vein, the lines defining the RCCC linkage of Fig. 1 are represented by the dual unit vectors $\hat{\mathbf{a}}$, $\hat{\mathbf{b}}$, $\hat{\mathbf{c}}$ and $\hat{\mathbf{d}}$. Correspondingly, $\hat{\mathbf{a}}_0$ and $\hat{\mathbf{c}}_0$ represent lines $\mathcal{A}_0$ and $\mathcal{C}_0$, respectively, i.e., the reference locations of $\mathcal{A}$ and $\mathcal{C}$, with a similar notation for $\mathcal{A}_j$ and $\mathcal{C}_j$, for $j = 1, \ldots, m > 4$. Therefore,

$$\hat{\mathbf{a}}_j = \hat{\mathbf{Q}}_j \hat{\mathbf{a}}_0, \quad \hat{\mathbf{c}}_j = \hat{\mathbf{Q}}_j \hat{\mathbf{c}}_0, \quad j = 1, \ldots, m \tag{1}$$

with $\hat{\mathbf{Q}}_j = \mathbf{Q}_j + \varepsilon\, \mathbf{Q}_{oj}$ denoting the dual orthogonal matrix that carries $\mathcal{F}$ from its reference pose to its jth pose. In this notation, $\mathbf{Q}_j$ denotes a rotation matrix, while $\mathbf{Q}_{oj} \equiv \mathbf{D}_j \mathbf{Q}_j$, and $\mathbf{D}_j$ denotes the cross-product matrix (CPM) of vector $\mathbf{d}_j$ that represents the translation of point R. The *cross product matrix* of a 3-dimensional vector $\mathbf{u}$ is defined as $\mathbf{U} = \text{CPM}(\mathbf{u}) \equiv \partial(\mathbf{u} \times \mathbf{v})/\partial \mathbf{v}$, for any 3-dimensional vector $\mathbf{v}$.

The angle between two dual unit vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$, representing lines $\mathcal{U}$ and $\mathcal{V}$, respectively, is denoted as $\hat{\theta}$. This angle occurs in the dot and the cross products of the two given vectors, in the form:

$$\hat{\mathbf{u}}^T \hat{\mathbf{v}} = \cos \hat{\theta}, \quad \hat{\mathbf{u}} \times \hat{\mathbf{v}} = \hat{\mathbf{w}} \sin \hat{\theta} \tag{2}$$

where $\hat{\mathbf{w}}$ is the dual unit vector normal to both $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$, i.e., a line that is normal to the two lines represented by $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ and intersecting the two lines. Moreover,

$$\cos \hat{\theta} = \cos \theta - \varepsilon\, d \sin \theta, \quad \sin \hat{\theta} = \sin \theta + \varepsilon\, d \cos \theta \tag{3}$$

with d denoting the distance between $\mathcal{U}$ and $\mathcal{V}$. The dual angle $\hat{\theta}$ thus represents the *dual distance* between the two given lines. Representing the rigid-body condition that links 2 and 4 must obey at every prescribed pose is now straightforward: the dual distance between lines $\mathcal{A}_j$ and $\mathcal{B}$ as well as that between $\mathcal{C}_j$ and $\mathcal{D}$ must remain equal to that between their reference counterparts $\mathcal{A}_0$ and $\mathcal{B}$ and, correspondingly, $\mathcal{C}_0$ and $\mathcal{D}$, i.e., in light of Eq. (1),

$$\hat{\mathbf{b}}^T \hat{\mathbf{Q}}_j \hat{\mathbf{a}}_j = \hat{\mathbf{b}}^T \hat{\mathbf{a}}_0, \quad \hat{\mathbf{d}}^T \hat{\mathbf{Q}}_j \hat{\mathbf{c}}_j = \hat{\mathbf{d}}^T \hat{\mathbf{c}}_0, \quad j = 1, \ldots, m$$

or, in homogeneous form,

$$\hat{\varphi}_j \equiv \hat{\mathbf{b}}^T (\hat{\mathbf{Q}}_j - \mathbf{1}) \hat{\mathbf{a}}_0 = 0, \quad \hat{\varphi}_{j+m} \equiv \hat{\mathbf{d}}^T (\hat{\mathbf{Q}}_j - \mathbf{1}) \hat{\mathbf{c}}_0 = 0, \quad j = 1, \ldots, m \tag{4}$$

which represent $2m$ *dual synthesis equations* in the four dual unknown vectors $\hat{\mathbf{a}}_0$, $\hat{\mathbf{b}}$, $\hat{\mathbf{c}}_0$ and $\hat{\mathbf{d}}$. As each equation involves two real equations, one for its primal, one for its dual part, the total number of real equations is $4m$. Likewise, each dual unknown vector entails two 3-dimensional dual vectors, the total number of unknowns is 24. However, each dual unit vector must obey the unit-vector constraints:

$$\hat{h}_1 \equiv \|\hat{\mathbf{a}}_0\|^2 - 1 = 0, \ \hat{h}_2 \equiv \|\hat{\mathbf{b}}\|^2 - 1 = 0, \ \hat{h}_3 \equiv \|\hat{\mathbf{c}}_0\|^2 - 1 = 0, \ \hat{h}_4 \equiv \|\hat{\mathbf{d}}\|^2 - 1 = 0 \tag{5}$$

Again, each of the foregoing equations represents two real equations, one for its primal, one for its dual parts, thereby obtaining a total of eight real constraints. Moreover, the primal part refers to the normality of the primal unit vector, the dual part to the *Klein condition*.[2] The foregoing conditions apply to dyads of the CC type. The RC dyad must obey one more condition: *the sliding of the joint coupling link 2 with link 1 must vanish*, which can be enforced by stating that all common normals $\mathcal{N}_j$ to $\mathcal{A}_j$ and $\mathcal{B}$ must intersect $\mathcal{N}_0$, the counterpart normal to $\mathcal{A}_0$ and $\mathcal{B}$. Let $\hat{\mathbf{n}}_0$ represent $\mathcal{N}_0$, $\hat{\mathbf{n}}_j$ representing $\mathcal{N}_j$. The intersection condition can be expressed via the dual unit vectors representing the lines of interest. Indeed, from the expansion of $\cos\hat{\theta}$ in Eq. (3), it is apparent that the intersection condition is that the dual part of $\hat{\mathbf{n}}_j^T\hat{\mathbf{n}}_0$, represented as $\mathrm{du}(\hat{\mathbf{n}}_j^T\hat{\mathbf{n}}_0)$, vanish, whence m additional constraints are obtained, namely,

$$h_{4+j} \equiv \mathrm{du}(\hat{\mathbf{n}}_j^T\hat{\mathbf{n}}_0) = 0, \quad j = 1, \ldots, m \tag{6}$$

Further, the 12-dimensional dual vector of unknowns $\hat{\mathbf{x}}$ is introduced:

$$\hat{\mathbf{x}} = \left[\, \hat{\mathbf{a}}_0^T \;\hat{\mathbf{b}}^T \;\hat{\mathbf{c}}_0^T \;\hat{\mathbf{d}}^T \,\right]^T \tag{7}$$

together with the $2m$-dimensional dual vector $\hat{\varphi}(\hat{\mathbf{x}})$ of synthesis equations, whose components are $\hat{\varphi}_j$ and $\hat{\varphi}_{j+m}$, as defined in Eq. (4), and the $(4+m)$-dimensional dual vector of constraints $\hat{\mathbf{h}}(\hat{\mathbf{x}})$, whose components are defined in Eqs. (5) and (6). Notice that, contrary to the exact synthesis case, here the synthesis equations need not be exactly satisfied; a reasonable approximation to those equations suffices. However, the $4 + m$ constraints must be met exactly—up to roundoff error, of course. The optimization problem is now stated as one of *constrained nonlinear least squares*:

$$\hat{f}(\hat{\mathbf{x}}) \equiv \frac{1}{2}\hat{\varphi}(\hat{\mathbf{x}})^T\mathbf{W}\hat{\varphi}(\hat{\mathbf{x}}) \;\rightarrow\; \min_{\hat{\mathbf{x}}} \tag{8a}$$

subject to

$$\hat{\mathbf{h}}(\hat{\mathbf{x}}) = \mathbf{0}_{4+m} \tag{8b}$$

In the problem statement (8a), $\mathbf{W}$ is a symmetric, positive-definite weighting matrix that is introduced to allow for assigning different relevance to different poses. For example, the mth pose may be given much higher relevance than its intermediate counterparts. A better approach would be to raise the mth pose to the category of constraints, so that it would be met exactly. However, the total number of constraints should be smaller than 12, the number of unknowns; else, the problem would be overconstrained and no solution would be possible, i.e., $m < 8$.

[2] This condition states that the primal and the dual parts of a dual unit vector must be mutually orthogonal.

3 The Dual Normality Conditions

The *first-order normality conditions* (FONC) for a *constrained nonlinear programming problem*, the class to which problem (8a and 8b) belongs, are well known in the case of problems defined by vectors over the real field [11]. In our case, all vectors are defined over the ring of dual numbers.[3] Paraphrasing those normality conditions, we have, for the case at hand:

$$\nabla \hat{f} + \hat{\mathbf{J}}^T \hat{\lambda} = \mathbf{0}_{2m} \tag{9}$$

where $\hat{\lambda}$ is the $(4 + m)$-dimensional vector of dual Lagrange multipliers that are needed to take the constraints into account, and $\hat{\mathbf{J}}$ is the $2m \times (4 + m)$ Jacobian matrix of the constraints, i.e., the gradient of $\hat{\mathbf{h}}$. Moreover, by virtue of the form of the objective function $\hat{f}$, $\nabla \hat{f}$ takes the form

$$\nabla \hat{f} = \hat{\Phi}^T \mathbf{W} \hat{\varphi} \tag{10}$$

with $\hat{\Phi}$ defined as $\nabla \hat{\varphi}$. Now, if the expression for the derivative of a dual function $\hat{f}(\hat{x})$ with respect to its dual argument is recalled, with $\hat{f}$ and $\hat{x}$ given by $\hat{f} = f + \varepsilon f_o$ and $\hat{x} = x + \varepsilon x_o$, namely [12],

$$\frac{\mathrm{d}\hat{f}}{\mathrm{d}\hat{x}} = \frac{\mathrm{d}f}{\mathrm{d}x} + \varepsilon \frac{\mathrm{d}f_o}{\mathrm{d}x} = \frac{\mathrm{d}\hat{f}}{\mathrm{d}x} \tag{11}$$

then $\hat{\Phi}$ and $\hat{\mathbf{J}}$ become $\hat{\Phi} = \partial \hat{\varphi} / \partial \mathbf{x}$ and $\hat{\mathbf{J}} = \nabla \hat{\mathbf{h}} = \partial \hat{\mathbf{h}} / \partial \mathbf{x}$, i.e., only the derivatives w.r.t. the primal part of the dual argument come into play in the foregoing gradients.

What condition (9) states is that, at a *stationary point* of problem (8a), $\nabla \hat{f}$ need not vanish, but must lie in the range of $\hat{\mathbf{J}}^T$, i.e., the overdetermined system (9) of $2m$ linear equations in the $4 + m$ (<12) unknowns, the number of dual Lagrange multipliers in $\hat{\lambda}$, must admit an exact solution. The FONC can be stated in two alternative forms:

$$[\mathbf{1} - \hat{\mathbf{J}}^T (\hat{\mathbf{J}}\hat{\mathbf{J}}^T)^{-1} \hat{\mathbf{J}}] \hat{\Phi} \mathbf{W} \hat{\varphi} = \mathbf{0}_{4+m}, \quad \hat{\mathbf{L}}^T \hat{\Phi}^T \mathbf{W} \hat{\varphi} = \mathbf{0}_{4+m} \tag{12}$$

with $\mathbf{0}_{4+m}$ denoting the $(4 + m)$-dimensional zero vector.

The matrix inside the brackets in the first of the foregoing equations can be readily identified as a projector that maps n-dimensional vectors onto the null space of $\hat{\mathbf{J}}$. Matrix $\hat{\mathbf{L}}$ in the second equation is a $12 \times (n - 4 - m)$ orthogonal complement of $\hat{\mathbf{J}}$, i.e, $\hat{\mathbf{J}}\hat{\mathbf{L}} = \mathbf{O}$, with $\mathbf{O}$ denoting the $(4 + m) \times (n - 4 - m)$ zero matrix.

[3] While vector spaces must be defined over a field, in our context we need to define them over the set of dual numbers, that do not form a field, but rather a ring. This difference does not pose any technical problem to the developments in the balance of the paper.

These conditions are necessary for a value of $\hat{\mathbf{x}}$ to be a *stationary point* of the problem under study. For this point to be a minimum, the *second-order normality condition* must be satisfied. In nonlinear-programming problems, this condition is that the *reduced Hessian* of the problem under study be positive-definite at a stationary feasible point, i.e., at a point that satisfies both the FONC, Eq. (12), and the constraints, Eq. (8b). In our case, such a point, designated by $\hat{\mathbf{x}}^*$, is assumed to have been found, vector $\hat{\varphi}(\hat{\mathbf{x}}^*)$ being represented by $\hat{\varphi}^*$. The reduced Hessian matrix takes the form

$$\hat{\mathbf{H}}_r = \hat{\mathbf{L}}^T\left[\hat{\mathit{\Phi}}^T\mathbf{W}\hat{\mathit{\Phi}} + \frac{\partial(\hat{\mathit{\Phi}}^T\mathbf{W}\hat{\varphi}^*)}{\partial\hat{\mathbf{x}}} + \frac{\partial(\hat{\mathbf{J}}^T\hat{\lambda})}{\partial\hat{\mathbf{x}}}\right]\hat{\mathbf{L}} \tag{13}$$

In our case, the second-order normality condition for a minimum is that the primal part of $\hat{\mathbf{H}}_r$ be positive-definite.

4 The Dual Orthogonal-Decomposition Algorithm

The orthogonal-decomposition algorithm (ODA) was developed by the author and his team to solve equality-constrained problems in mathematical programming [13]. When applied to the RCCC approximate-synthesis problem, the algorithm takes the form described below: it is assumed that a feasible approximation to the optimum has been obtained at the kth iteration, $\hat{\mathbf{x}}^k$, an increment $\mathit{\Delta}\hat{\mathbf{x}}^k$ being sought that will yield an improved approximation $\hat{\mathbf{x}}^{k+1}$. The strategy consists in decomposing the increment in two parts, namely,

$$\mathit{\Delta}\hat{\mathbf{x}}^k = \mathit{\Delta}\hat{\mathbf{v}}^k + \hat{\mathbf{L}}_k\mathit{\Delta}\hat{\mathbf{u}}^k \tag{14}$$

with $\hat{\mathbf{L}}_k$ denoting the orthogonal complement $\hat{\mathbf{L}}$ evaluated at $\hat{\mathbf{x}}^k$. Moreover, $\mathit{\Delta}\hat{\mathbf{v}}^k$ is the minimum-norm solution of the underdetermined linear system of dual equations

$$\hat{\mathbf{J}}_k\mathit{\Delta}\hat{\mathbf{v}}^k = -\hat{\mathbf{h}}^k \tag{15}$$

in which $\hat{\mathbf{J}}_k$ and $\hat{\mathbf{h}}^k$ denote the Jacobian $\hat{\mathbf{J}}$ and vector $\hat{\mathbf{h}}$ evaluated at $\hat{\mathbf{x}}^k$. The minimum-norm solution of Eq. (15) can be expressed in terms of the dual right Moore-Penrose generalized inverse [14], namely,

$$\mathit{\Delta}\hat{\mathbf{v}}^k = -\hat{\mathbf{J}}_k^{\dagger}\hat{\mathbf{h}}^k, \quad \hat{\mathbf{J}}_k^{\dagger} \equiv \hat{\mathbf{J}}_k^T(\hat{\mathbf{J}}_k\hat{\mathbf{J}}_k^T)^{-1} \tag{16}$$

where $\hat{\mathbf{J}}_k^{\dagger}$ is to be calculated with the dual QR-decomposition of $\hat{\mathbf{J}}_k^T$. The QR-decomposition for real matrices is well documented in the literature on numerical analysis [15]. With $\mathit{\Delta}\hat{\mathbf{v}}^k$ computed, $\mathit{\Delta}\hat{\mathbf{u}}^k$ is computed as the least-square approximation of an overdetermined system of linear equations:

$$\mathbf{V}\hat{\mathit{\Phi}}_k\hat{\mathbf{L}}_k\mathit{\Delta}\hat{\mathbf{u}}^k = -\mathbf{V}(\hat{\varphi}^k + \hat{\mathit{\Phi}}_k\mathit{\Delta}\hat{\mathbf{V}}^k), \quad \mathbf{W} \equiv \mathbf{V}^T\mathbf{V} \tag{17}$$

whence the solution $\mathit{\Delta}\hat{\mathbf{u}}^k$ is computed with the left Moore-Penrose generalized inverse of the product $\mathbf{V}\hat{\mathit{\Phi}}_k\hat{\mathbf{L}}_k$:

$$\mathit{\Delta}\hat{\mathbf{u}}^k = -(\mathbf{V}\hat{\mathit{\Phi}}_k\hat{\mathbf{L}}_k)^I(\hat{\varphi}^k + \hat{\mathit{\Phi}}_k\mathit{\Delta}\hat{\mathbf{v}}^k), \quad (\mathbf{V}\hat{\mathit{\Phi}}_k\hat{\mathbf{L}}_k)^I \equiv (\hat{\mathbf{L}}^T\hat{\mathit{\Phi}}^T\mathbf{W}\hat{\mathit{\Phi}}\hat{\mathbf{L}})^{-1}\hat{\mathbf{L}}^T\hat{\mathit{\Phi}}^T\mathbf{V} \tag{18}$$

thereby completing the $(k+1)$st iteration. The procedure stops when the FONC, Eq. (12), are verified to a prescribed tolerance.

5 Conclusions

The foundations for the approximate synthesis of RCCC linkages for motion generation were laid down. It was shown that, by virtue of the normality conditions that the dual unit vectors that represent the linkage four joint axes must observe, the number of prescribed poses of the coupler link is limited to being smaller than eight.

References

1. Burmester, L.: Lehrbuch der Kinematik. Arthur Felix Verlag, Leipzig (1888)
2. Chen, P., Roth, B.: Design equations for the finitely and infinitesimally separated position synthesis of binary links and combined link chains. ASME J. Eng. Ind. **91**, 209–219 (1969)
3. Al-Widyan, K.M., Angeles, J.: The kinematic synthesis of a robust rccc mechanism for pick-and-place operations. In: Proceedings of ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE, Chicago, IL, 12–15 Aug 2012, Paper No. DETC2012-70878
4. Dudiţá, Fl., Diaconescu, D., Gogu Gr.: Mecanisme Articulate. Technicá Publishers, Bucharest (1989)
5. Luck, K., Modler, K.-H.: Getriebetechnik. Analyse-Synthese-Optimierung. Akademie Publishers, Berlin (1990)
6. Kimbrell, J.T.: Kinematics Analysis and Synthesis. McGraw-Hill, Inc., New York (1991)
7. McCarthy, J.M., Soh, G.S.: Geometric Design of Linkages. Springer, New York (2011)
8. Fischer, I.S.: Dual-Number Methods in Kinematics, Statistics and Dynamics. CRC Press, Boca Raton (1999)
9. Pradeel, A.K., Yoder, P.Y., Mukundan, R.: On the use of dual matrix exponentials in robot kinematics. Int. J. Robot. Res. **8**(5), 57–66 (1989)
10. Angeles, J.: The application of dual algebra to kinematic analysis. In: Angeles, J., Zakhariev, E. (eds.) Computational Methods in Mechanical Systems, vol. 161, pp. 3–31. Springer, Heidelberg (1998)
11. Luenberger, D.G.: Linear and Nonlinear Programming, 2nd edn. Addison-Wesley Publishing Company, Reading (1984)
12. Kotel'nikov, A.P.: Screw Calculus and Some of Its Applications to Geometry and Mechanics (in Russian), Annals of The Imperial University of Kazan, 2006 edition by KomKniga, Moscow (1895)
13. Teng, C.P., Angeles, J.: A sequential-quadratic-programming algorithm using orthogonal decomposition with Gerschgorin stabilization. ASME J. Mech. Design **123**, 501–509 (2001)
14. Angeles, J.: The dual generalized inverses and their applications in kinematic synthesis. In: Lenarčič, J., Husty, M. (eds.) Latest Advances in Robot Kinematics, pp. 1–10. Springer, Dordrecht (2012)
15. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, Baltimore (1983)

The 3-RPS Manipulator Can Have Non-Singular Assembly-Mode Changes

Manfred Husty, Josef Schadlbauer, Stéphane Caro and Philippe Wenger

Abstract Recently a complete kinematic description of the 3-RPS parallel manipulator was obtained using algebraic constraint equations. It turned out that the workspace decomposes into two components describing two kinematically different operation modes and that self-motions of this manipulator in both operation are possible. In this paper for the first time it is shown that this manipulator has the property of non singular assembly mode change.

Keywords 3-RPS-manipulator · Singularities · Assembly mode change

1 Introduction

Non-singular assembly mode change has been discussed a lot for parallel manipulators since Innocenti and Parenti-Castelli [1] showed examples of such a behavior. Especially planar 3-RPR parallel manipulators were extensively investigated with respect to non-singular assembly mode change (see e.g. [2] and [3]). In [4] it was shown, that every generic planar 3-RPR has two aspects, meaning that the singularity surface divides the workspace into two parts and therefore non-singular assembly

M. Husty · J. Schadlbauer
Institute for Basic Sciences in Engineering, Unit for Geometry and CAD,
University of Innsbruck, Innsbruck, Austria
e-mail: manfred.husty@uibk.ac.at

J. Schadlbauer
e-mail: josef.schadlbauer@uibk.ac.at

S. Caro · P. Wenger (✉)
Institut de Recherche en Communications et Cybernétique de Nantes,
Nantes, France
e-mail: stephane.caro@irccyn.ec-nantes.fr

P. Wenger
e-mail: philippe.wenger@irccyn.ec-nantes.fr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_38,

mode change is always possible. To the best of the knowledge of the authors singularity free assembly mode change has never been shown explicitly for spatial lower mobility parallel manipulators. It is the motivation to demonstrate this behavior for such a manipulator. One of the best investigated designs of lower mobility parallel manipulators is the 3-RPS manipulator introduced by Hunt [5]. This manipulator is simple enough to make this task feasible.

A 3-RPS manipulator is a three degree of freedom (DOF) parallel manipulator. It consists of an equilateral triangular fixed platform and a similar moving platform connected by three identical RPS legs. The first joint (R-joint) is connected to the base and the last joint (S-joint) is connected to the moving platform (see Fig. 1). The legs are extensible, changing lengths via prismatic joints (P-joints), thereby moving the platform with three highly coupled DOFs. In the past few years the 3-RPS obtained a lot of attention in the kinematics community, see e.g. [6].

In [7] an overview of existing results up to the year 2008 can be found. Local analysis, mostly using screw theory was performed in most of the existing investigations especially in [8] and [9]. More recently in [10], using an algebraic description of the manipulator, together with *Study*'s kinematic mapping, a complete characterization of the forward kinematics, the operation modes, the singular poses and the transitions between the operation modes was given. It turned out that the manipulator has two kinematically different operation modes. The first one is characterized by finite π-screws. Axes of these screws are tilted with respect to the base and the translation distance depends on the chosen axis. The second mode has horizontal screw axes with rotation angle and translation distance depending on the chosen axis. Note, that this characterization refers to finite screws and not instantaneous screws. The singularities in both operation modes were derived in the kinematic image space as well as in the joint space. In joint space the singularity surfaces are of degree 24 and it was shown that for input joint combinations fulfilling an eight order polynomial transition from one operation mode to the other is possible. In [11] it was shown, that the manipulator can perform a spherical and "butterfly" self-motion for special leg lengths.

In this paper an example of singularity free assembly mode change will be given. To prove this property a singularity free path will be constructed that moves the manipulator around a cusp in a slice of its workspace.

The paper is organized as follows: In Sect. 2 a description of the architecture of the 3-RPS is given and the set of constraint equations is recalled. Section 3 introduces a method to construct a nonsingular assembly mode change path for this manipulator.

2 Robot Design

With respect to Fig. 1 we consider the 3-RPS parallel manipulator with the following architecture: The base of the 3-RPS consists of an equilateral triangle with vertices $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ and circumradius h_1. The origin of the fixed frame Σ_0 coincides with the circumcenter of the triangle $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$. The yz-plane of Σ_0 is defined

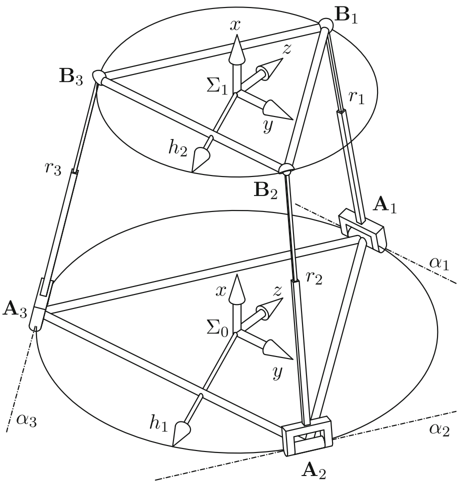


Fig. 1 Design of the 3-RPS parallel robot

by the plane $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$. Finally, $\mathbf{A}_1$ lies on the z-axis of Σ_0. In the platform there is another equilateral triangle with vertices $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ and circumradius h_2. The circumcenter of the triangle $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ lies in the origin of Σ_1, which is the moving frame. Again, the plane defined by $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ coincides with the yz-plane of Σ_1 and $\mathbf{B}_1$ lies on the z-axis of Σ_1.

The two design parameters h_1 and h_2 are taken to be strictly positive numbers. Now each pair of vertices $\mathbf{A}_i$, $\mathbf{B}_i$ $(i = 1, \ldots, 3)$ is connected by a limb, with a rotational joint at $\mathbf{A}_i$ and a spherical joint at $\mathbf{B}_i$. The length of each limb is denoted by r_i and is adjusted via an actuated prismatic joint. The axes α_i of the rotational joints at $\mathbf{A}_i$ are tangent to the circumcircle and therefore lie within the yz-plane of Σ_0. Overall we have five parameters, namely h_1, h_2, r_1, r_2 and r_3. While h_1 and h_2 determine the design of the manipulator, the parameters r_1, r_2 and r_3 are joint parameters, which determine the motion of the robot. We can consider the joint parameters to be like design parameters when they are assigned with specific leg lengths r_i. In some computations the leg lengths r_i will be replaced with their squares which then will be denoted by R_i. Deriving the constraint equations is one essential step in solving the kinematics of a manipulator. To compute these equations which describe the motion capability, the direct kinematics and also the singularities of the manipulator, we use the *Study*-parameterization of the motion group $SE(3)$. The vertices of the base triangle and the platform triangle in Σ_0 resp. Σ_1 are

$$\mathbf{A}_1 = (1, 0, 0, h_1), \quad \mathbf{A}_2 = (1, 0, \sqrt{3}h_1/2, -h_1/2), \quad \mathbf{A}_3 = (1, 0, -\sqrt{3}h_1/2, -h_1/2)$$
$$\mathbf{b}_1 = (1, 0, 0, h_2), \quad \mathbf{b}_2 = (1, 0, \sqrt{3}h_2/2, -h_2/2), \quad \mathbf{b}_3 = (1, 0, -\sqrt{3}h_2/2, -h_2/2)$$

thereby using projective coordinates with the homogenizing coordinate in first place. To avoid confusion coordinates with respect to Σ_0 are written in capital letters and those with respect to Σ_1 are in lower case. To obtain the coordinates $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$

of $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$with respect to Σ_0 a coordinate transformation has to be applied. To describe this coordinate transformation we use *Study's* parameterization of a spatial Euclidean transformation matrix $\mathbf{M} \in SE(3)$ (for detailed information on this approach see [12]).

$$\mathbf{M} = \begin{pmatrix} x_0^2+x_1^2+x_2^2+x_3^2 & \mathbf{0}^\top \\ \mathbf{M}_T & \mathbf{M}_R \end{pmatrix}, \quad \mathbf{M}_T = \begin{pmatrix} 2(-x_0y_1+x_1y_0-x_2y_3+x_3y_2) \\ 2(-x_0y_2+x_1y_3+x_2y_0-x_3y_1) \\ 2(-x_0y_3-x_1y_2+x_2y_1+x_3y_0) \end{pmatrix}$$

$$\mathbf{M}_R = \begin{pmatrix} x_0^2+x_1^2-x_2^2-x_3^2 & 2(x_1x_2-x_0x_3) & 2(x_1x_3+x_0x_2) \\ 2(x_1x_2+x_0x_3) & x_0^2-x_1^2+x_2^2-x_3^2 & 2(x_2x_3-x_0x_1) \\ 2(x_1x_3-x_0x_2) & 2(x_2x_3+x_0x_1) & x_0^2-x_1^2-x_2^2+x_3^2 \end{pmatrix}$$

The vector $\mathbf{M}_T$ represents the translational part and $\mathbf{M}_R$ represents the rotational part of the transformation $\mathbf{M}$. The parameters $x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3$ which appear in the matrix $\mathbf{M}$ are called *Study*-parameters of the transformation $\mathbf{M}$. The mapping

$$\begin{gathered} \kappa : SE(3) \to P \in \mathbb{P}^7 \\ \mathbf{M}(x_i, y_i) \mapsto (x_0 : x_1 : x_2 : x_3 : y_0 : y_1 : y_2 : y_3)^T \neq (0:0:0:0:0:0:0:0)^T \end{gathered} \tag{1}$$

is called *kinematic mapping* and maps each Euclidean displacement of $SE(3)$ to a point P on a quadric $S_6^2 \subset \mathbb{P}^7$. In this way, every projective point $(x_0 : x_1 : x_2 : x_3 : y_0 : y_1 : y_2 : y_3) \in \mathbb{P}^7$ represents a spatial Euclidean transformation, if it fulfills the following equation $S_6^2 : x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3 = 0$ and inequality: $x_0^2 + x_1^2 + x_2^2 + x_3^2 \neq 0$ (see [12]).

The coordinates of $\mathbf{b}_i$ with respect to Σ_0 are obtained by:

$$\mathbf{B}_i = \mathbf{M} \cdot \mathbf{b}_i, \quad i = 1, \ldots, 3.$$

Now, as the coordinates of all vertices are given in terms of the transformation parameters $x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3$ and the design constants, we obtain constraint equations by examining the geometry of the manipulator more closely. First of all the limb connecting $\mathbf{A}_i$ and $\mathbf{B}_i$ has to be orthogonal to the corresponding rotational axis α_i. That means, the scalar product of the vector connecting $\mathbf{A}_i\mathbf{B}_i$ and the direction of α_i vanishes. After computing this product, removing the common denominator $(x_0^2 + x_1^2 + x_2^2 + x_3^2)$ and performing some elementary simplifications the following equations are obtained:

$$\begin{aligned} g_1 &: x_0x_1 = 0 \\ g_2 &: h_2x_2^2 - h_2x_3^2 - 2x_0y_3 - 2x_1y_2 + 2x_2y_1 + 2x_3y_0 = 0 \\ g_3 &: 2h_2x_0x_1 + h_2x_2x_3 - x_0y_2 + x_1y_3 + x_2y_0 - x_3y_1 = 0. \end{aligned} \tag{2}$$

This set of equations is augmented by three leg length conditions:

$$
\begin{aligned}
&g_4 : (h_1 - h_2)^2 x_0^2 + (h_1 + h_2)^2 x_1^2 + (h_1 + h_2)^2 x_2^2 + (h_1 - h_2)^2 x_3^2 + 4(h_1 - h_2) x_0 y_3 + 4(h_1 + h_2) x_1 y_2 \\
&\quad - 4(h_1 + h_2) x_2 y_1 - 4(h_1 - h_2) x_3 y_0 + 4(y_0^2 + y_1^2 + y_2^2 + y_3^2) - (x_0^2 + x_1^2 + x_2^2 + x_3^2) R_1 = 0 \\
&\quad g_5 : (h_1 - h_2)^2 x_0^2 + (h_1 + h_2)^2 x_1^2 + (h_1^2 + h_2^2 - h_1 h_2) x_2^2 + (h_1^2 + h_2^2 + h_1 h_2) x_3^2 - 2(h_1 \\
&\qquad - h_2) x_0 y_3 - 2(h_1 + h_2) x_1 y_2 + 2(h_1 + h_2) x_2 y_1 + 2(h_1 - h_2) x_3 y_0 - 2\sqrt{3}(h_1 \\
&\qquad - h_2) x_0 y_2 + 2\sqrt{3}(h_1 + h_2) x_1 y_3 + 2\sqrt{3}(h_1 - h_2) x_2 y_0 - 2\sqrt{3}(h_1 + h_2) x_3 y_1 \\
&\qquad - 2\sqrt{3} h_1 h_2 x_2 x_3 + 4(y_0^2 + y_1^2 + y_2^2 + y_3^2) - (x_0^2 + x_1^2 + x_2^2 + x_3^2) R_2 = 0 \\
&\quad g_6 : (h_1 - h_2)^2 x_0^2 + (h_1 + h_2)^2 x_1^2 + (h_1^2 + h_2^2 - h_1 h_2) x_2^2 + (h_1^2 + h_2^2 + h_1 h_2) x_3^2 - 2(h_1 \\
&\qquad - h_2) x_0 y_3 - 2(h_1 + h_2) x_1 y_2 + 2(h_1 + h_2) x_2 y_1 + 2(h_1 - h_2) x_3 y_0 + 2\sqrt{3}(h_1 \\
&\qquad - h_2) x_0 y_2 - 2\sqrt{3}(h_1 + h_2) x_1 y_3 - 2\sqrt{3}(h_1 - h_2) x_2 y_0 + 2\sqrt{3}(h_1 + h_2) x_3 y_1 \\
&\qquad + 2\sqrt{3} h_1 h_2 x_2 x_3 + 4(y_0^2 + y_1^2 + y_2^2 + y_3^2) - (x_0^2 + x_1^2 + x_2^2 + x_3^2) R_3 = 0.
\end{aligned}
$$

A detailed explanation of how this set of equations is derived is left out for sake of lack of space but can be found in [10]. To complete the system, we add the Study-equation (g_7), because all the solutions have to be within the Study-Quadric and a normalizing condition (g_8).

$$
g_7 : x_0 y_0 + x_1 y_1 + x_2 y_2 + x_3 y_3 = 0, \quad g_8 : x_0^2 + x_1^2 + x_2^2 + x_3^2 = 1 \tag{3}
$$

It is emphasized that R_i in equations g_4, g_5, g_6 denote the squares of the input parameters (leg lengths). The set of equations describing a general 3-RPS manipulator forms the ideal

$$
\mathcal{I} = \langle g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8 \rangle \tag{4}
$$

From the first equation in this set it is obvious, that this ideal consists of two components $\mathcal{K}_1 = \langle x_0, g_2, g_3, g_4, g_5, g_6, g_7, g_8 \rangle$ and $\mathcal{K}_2 = \langle x_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8 \rangle$. It was shown in [10] that these two components can be treated separately to compute the direct kinematics and all singularities of this manipulator. Therefore the same can be done for computing non singular assembly mode change of this manipulator type.

3 Non-Singular Assembly Mode Change

The main idea in [4] to prove the non-singular assembly mode change behavior is the representation of the singularity surface in the three dimensional kinematic image space, where the two aspects of the singularity surface can be visualized and singularity free assembly mode changing paths can be constructed easily. A similar method was used in [13] and [14] to prove the assembly mode changing property for spherical 3-RPR parallel manipulators.

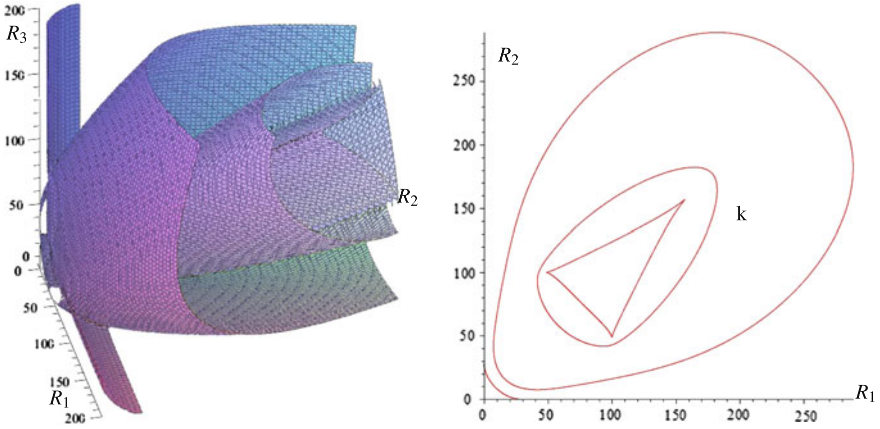


Fig. 2 Singularity surface S in joint space $h_1 = 1, h_2 = 2$ Slice through S at $R_3 = 100$

The same method as in planar and spherical cases cannot be used for a 3-dof spatial manipulator. It is not possible to derive the singularity surface in a 3-dim kinematic image space and construct singularity free paths, because the singularity surface is contained in the 7-dim kinematic image space of spatial displacements and therefore difficult to handle. In the following a new method is presented to overcome these difficulties and to prove that non-singular assembly mode change is also possible in case of a 3-RPS parallel manipulator.

The two governing ideals $\mathcal{K}_1$ and $\mathcal{K}_2$ which describe the motion capabilities of the manipulator, are treated separately. It was already shown in [10] that the singularity surface for each component can be computed in the kinematic image space and in the joint space.

The singularity surface in the joint space, by using the leg lengths $r_i, i = 1 \dots 3$ as coordinates, has degree 24. A closer inspection shows that the variables of the singularity surface in joint space have only even powers. Therefore it makes sense to use the squares of the leg lengths as new coordinates. After the substitution $R_i = r_i^2$ the resulting singularity surface S has only degree 12. A part of S for the parameters $h_1 = 1, h_2 = 2$ is displayed in (Fig. 2).

Next the univariate polynomial of one ideal $\mathcal{K}_i$ in one of the Study parameters is computed. This can be done without specifying the leg length parameters R_i. For this purpose an ordered Groebner basis of the ideal e.g. $\mathcal{K}_1$ is computed and this yields a univariate polynomial F of degree eight in one variable (e.g. x_2) having only even powers

$$F : a_0 x_2^8 + b_0 x_2^6 + c_0 x_2^4 + d_0 x_2^2 + f_0 = 0, \tag{5}$$

where a_0, b_0, c_0, d_0, f_0 are polynomials in the input parameters R_1, R_2, R_3 and the design parameter h_2 (without loss of generality $h_1 = 1$ has been set). In the following we will take x_2 as paradigmatic example for the used Study parameter. Note that

x_2 could be replaced by any other Study parameter if the univariate polynomial had been computed in this other parameter.

Then a slice through the singularity surface is taken by setting one joint parameter constant $R_i = c$. Figure 2 shows an example of such a slice; the intersection curve is denoted k (the chosen joint parameters in the example are $R_1 = R_2 = 120$, $R_3 = 100$). The constant value c is also substituted into F. The result is a polynomial F_p in two joint parameters and one Study parameter. This polynomial can be viewed as level-set in the Study parameter x_2. The graph of this level-set is a surface

$$F_p(R_j, R_k, v) = 0, \;\; j, k \in \{1, 2, 3\}, j, k \neq i$$

of degree 4 in the square of the Study parameter $v := x_2^2$. We display it in the same coordinate system as the slice and extend the intersection curve k also to a level set. This level-set is trivial, because it is only the cylinder S_L above the intersection curve. It is interesting to note, that S_L is tangent to F_p. The curve of tangency is exactly the set of singularities of the manipulator, which belong to values of x_2 on F_p. The cusps on the inner part of k indicate that the surface F_P folds back and a singularity free assembly mode changing path can be constructed. The interior of the inner most part of the curve k is a four solutions region of the direct kinematics and outside of this region and inside of the outermost part of k there are two solutions. Now the methodology used in [3] can be applied. We construct in the plane $R_i = c$ a path around the cusp starting at the point S in the interior of the three cusp curve and ending at the point E which is coincident with S but belonging to another solution of the direct kinematics (Fig. 3, lower picture). This path is projected orthogonally in the direction of the v coordinate onto the surface F_p. And in this projection one can see that the level set folds such that S and E are the same points in the slicing plane but belong to different solutions of the direct kinematics. This projection is the computationally most complicated part, because the path, which consists of three line segments in the plane $R_i = const.$, parameterized by a parameter t must be substituted into the polynomial F. The result is a polynomial of degree four in v and degree 8 in t, which must be solved for v. This yields four v coordinate functions, corresponding to four curves which project onto the given three line segment curve in the plane $R_i = c$. Not all four curves will be real in the considered interval. Figure 4 shows that this algorithm is computationally feasible. One can see how the projected curve b runs on the surface F_p. The red wireframe surface is S_L.

In a last step one has to prove that the curve b does not intersect the singularity curve on F. This can be done numerically and is visualized in a classical two view orthogonal projection. The top view is the plane $R_i = c$ (Fig. 3 lower picture). For the front view we take R_j, v as coordinates (Fig. 3 upper picture). The singularity curve on F_p in the top view is the curve k'. The three segment curve was designed such that it runs around the cusp of k'. We have to show that the two apparent intersection points P_1', P_2' in the top view are no intersection points of the curves k and b in space. It can be computed easily that the two apparent intersection points P_1', P_2' are not on the curve k'' in the front view. Graphically this is also shown in Fig. 3. Note that the top view of the singularity curve is computed as the resultant of S_L and F_p with

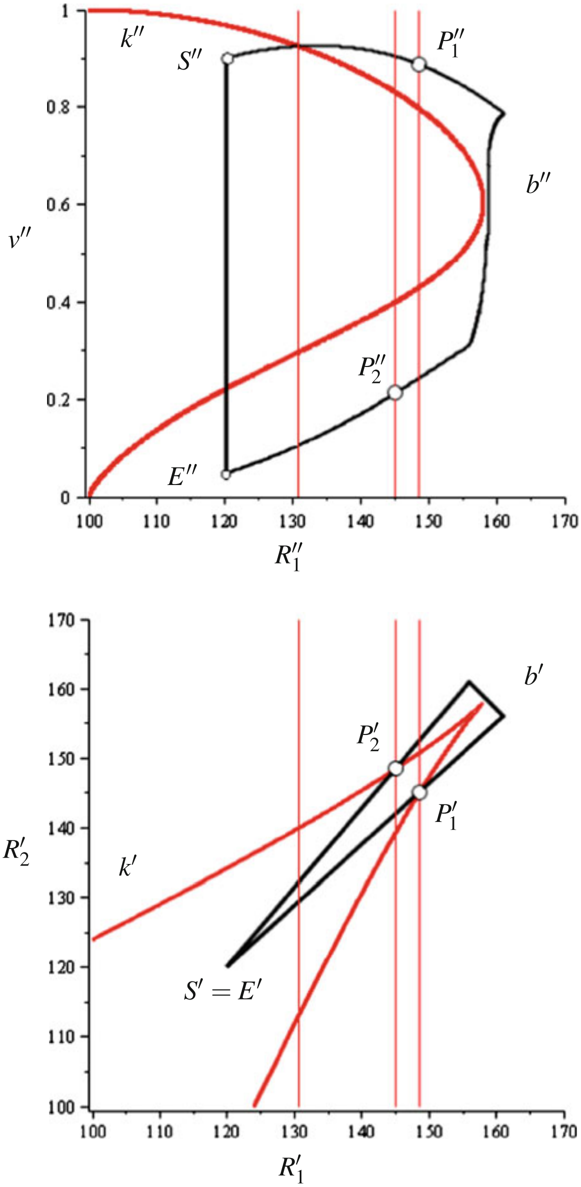


Fig. 3 Singularity curve on F in *front view* and *top view*

respect to the coordinate which is missing in the top view. The same arguments were used to show that the apparent intersection point between the two curves in the front view is no intersection point of the two curves. This proves that the constructed curve connects the two solutions of the direct kinematics without crossing a singularity.

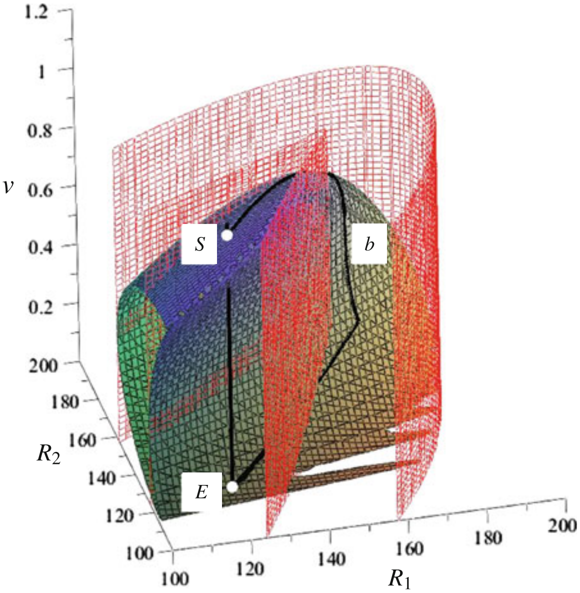


Fig. 4 3-D view of the level-set F singularity levelset S_L and constructed assembly changing curve

4 Conclusion

By constructing a complete example it was shown for the first time that a 3-RPS parallel mechanism allows non-singular assembly mode change. To prove this feature a level set was used and an assembly-mode changing path on the graph of this level set was constructed.

References

1. Innocenti, C., Parenti-Castelli, V.: Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators. J. Mech. Des. **120**, 73–79 (1998)
2. McAree, P.R., Daniel, R.W.: An explanation of never-special assembly changing motions for 33 parallel manipulators. Int. J. Rob. Res. **18**6, 556–574 (1999)
3. Zein, M., Wenger, P., Chablat, D.: Non-singular assembly-mode changing motions for 3-rpr parallel manipulators. Mech. Mach. Theory **43**4, 391–524 (2008)
4. Husty, M.L.: Non-singular assembly mode change in 3-RPR-parallel manipulators. In: Kecskeméthy, A., Müller, A. (eds.) Computational Kinematics, pp. 51–60. Springer, Berlin (2009)
5. Hunt, K.H.: Structural kinematics of in-parallel-actuated robot-arms. Trans. ASME J. Mech. Transm. Autom. Des. **105**, 705–712 (1983)
6. Tsai, L.-W.: Robot Analysis. Wiley, New York (1999)

7. Gallardo, J., Orozco, H., Rico, J., Aguilar, C., Perez, L.: Acceleration analysis of 3-RPS parallel manipulators by means of screw theory. In: Ryu, J.-H. (ed.) Parallel Manipulators, New Developments. I-Tech Education and Publishing, Vienna (2008)
8. Huang, Z., Wang, J., Fang, Y.: Analysis of instantaneous motions of deficient-rank 3-RPS parallel manipulators. Mech. Mach. Theory **37**, 229–240 (2002)
9. Basu, D., Ghosal, A.: Singularity analysis of platform-type multi-loop spatial mechanisms. Mech. Mach. Theory **32**, 375–389 (2002)
10. Schadlbauer, J., Husty, M.-L.: A complete analysis of the 3-RPS manipulator. In: Bandyopadhyay, S., Kumar, G.S., Ramu, P. (eds.) Machines and Mechanisms, pp. 410–419. Narosa Publishing House, New Delhi (2011)
11. Schadlbauer, J., Husty, M., Wenger, P., Caro, S.: Self-motions of 3-RPS manipulators. Front. Mech. Eng. **8**(1), 62–69 (2013)
12. Husty, M.L., Pfurner, M., Schröcker, H.-P., Brunnthaler, K.: Algebraic methods in mechanism analysis and synthesis. Robotica **25**(6), 661–675 (2007)
13. Urizar, M., Husty, M.-L.: Assembly mode change of spherical 3-RPR parallel manipulator. Proceedings of MUSME **2011**, 1–16 (2011)
14. Urizar, M., Husty, M.: Assembly mode change of spherical 3-RPR parallel manipulator. Mech. Based Des. Struct. Mach. **40**, 487–505 (2012)

Exact Workspace Synthesis for RCCR Linkages

Batchimeg Batbold, Yimesker Yihun, James S. Wolper
and Alba Pérez-Gracia

Abstract A tool for the exact kinematic synthesis of a given workspace may be of interest when designing closed linkages. In these cases, finite-position synthesis cannot ensure smoothness of motion between task positions. In order to keep the simplicity of the finite-position synthesis approach, the workspace of relative displacements is described as a set of finite screws forming a screw surface. The screw surface is characterized by a number of screws which are used to generate the whole surface, and in turn to perform the dimensional synthesis. The methodology is here applied to the overconstrained RCCR closed linkage, for which the workspace of finite displacements yields a point-path synthesis problem.

Keywords Workspace synthesis · RCCR linkage

1 Introduction

The dimensional synthesis of parallel robots has focused mainly on optimizing performance indices [8, 12] and reachable workspace sizing [1, 3, 13]; see also [14] for a comprehensive approach.

The use of a prescribed set of positions for the design of parallel robots by synthesizing all supporting legs has been applied in [20] for n-RRS parallel manipulators;

A. Pérez-Gracia (✉)
e-mail: perealba@isu.edu

B. Batbold
Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
e-mail: batchimeg82@gmail.com

Y. Yihun · J. S. Wolper · A. Pérez-Gracia
Idaho State University, Pocatello 83201, USA
e-mail: yihuyime@isu.edu

J. S. Wolper
e-mail: wolpjame@isu.edu

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_39,

also [11] and [16] perform partial kinematic synthesis of a 3-RPS parallel manipulator. In general, the finite-position method does not allow the control of the final trajectory of the parallel system and issues such as circuit defect may appear; in the most extreme cases, it may yield a system with negative mobility, that can be assembled at each task positions but cannot be driven from task position to task position.

The kinematic mapping is used for the synthesis of planar and spherical linkages in order to state design equations and to provide a tool for visualizing the workspace and trajectories of the linkage. See [17] and more recent applications [6, 18] and [21]. For spatial motion, Study's kinematic mapping is used to obtain simplified equations for analysis and synthesis, see [10] and [2].

Considering the workspace of the linkage as a set of finite screws corresponding to finite displacements of the end-effector, and using Parkin's definition for pitch [15], the workspace takes a simple expression in some cases, see [7]. If the expression for the workspace is known, a finite set of positions can define the workspace and synthesize the corresponding linkage [5].

In this paper we apply the technique to the closed, movable RCCR linkage, studied by Waldron [19]. Its finite-screw workspace [4] has a constant orientation, hence the relative motion consists of translations only. The relative translations of the RC chain form a quadric surface, which is related to the finite set of translations used for the synthesis in order to perform exact workspace synthesis. The method yields a maximum of six different solutions; intersecting pairs of real solutions, several RCCR workspaces are generated.

2 The Workspace of Finite Screws

The workspace of relative displacements of an articulated chain can be expressed as a set of screws with a magnitude and a pitch; Parkin's definition of pitch [15] is used. Parkin's pitch appears naturally in the forward kinematics equations when using the Cifford algebra of dual quaternions [4]. When using this pitch, the screws corresponding to finite displacements of some linkages form screw systems; however, the relative workspaces of many linkages are nonlinear screw surfaces.

Given a finite-displacement screw $\mathsf{J} = (1 + \mu\varepsilon)\mathsf{S}$, where μ is the pitch and $\mathsf{S} = (\mathbf{s}, \mathbf{c} \times \mathbf{s})$ are the Plucker coordinates of the screw axis, the exponential of the screw $\frac{\theta}{2}\mathsf{J}$ can be computed using the Clifford algebra product, to yield

$$e^{\frac{\theta}{2}\mathsf{J}} = \left(\cos\frac{\theta}{2} - \frac{d}{2}\sin\frac{\theta}{2}\varepsilon\right) + \left(\sin\frac{\theta}{2} + \frac{d}{2}\cos\frac{\theta}{2}\varepsilon\right)\mathsf{S} = \cos\frac{\hat{\theta}}{2} + \sin\frac{\hat{\theta}}{2}\mathsf{S}. \quad (1)$$

The exponential of the screw defines a unit dual quaternion, corresponding to a relative displacement from an initial position to a final position in terms of a rotation around and a slide along axis S.

For a serial chain with n joints, with joint parameters $\Delta\hat{\Theta} = (\Theta - \Theta_0 + (\mathbf{d} - \mathbf{d}_0)\varepsilon)$ around and along the axis S_i, $i = 1, \dots, k$, the product of exponentials defines the relative workspace from a reference configuration,

$$\hat{D}(\Delta\hat{\Theta}) = \cos\frac{\hat{\psi}}{2} + \sin\frac{\hat{\psi}}{2}\mathsf{S} = e^{\frac{\Delta\hat{\theta}_1}{2}\mathsf{S}_1} e^{\frac{\Delta\hat{\theta}_2}{2}\mathsf{S}_2} \cdots e^{\frac{\Delta\hat{\theta}_k}{2}\mathsf{S}_k}. \tag{2}$$

It is immediate to find the screw axis S, magnitude and pitch from this expression,

$$\sin\frac{\hat{\psi}}{2}\mathsf{S} = \left(\sin\frac{\psi}{2} + \varepsilon\frac{t}{2}\cos\frac{\psi}{2}\right)\mathsf{S} = \sin\frac{\psi}{2}\left(1 + \varepsilon\frac{\frac{t}{2}}{\tan\frac{\psi}{2}}\right)\mathsf{S}, \tag{3}$$

so that the finite-screw relative workspace is a set of screw axes with magnitude $\sin\frac{\psi}{2}$ and Parkin's pitch $\frac{t/2}{\tan\frac{\psi}{2}}$. The value of the magnitude is unique and can be calculated using the scalar part of the forward kinematics, see [9].

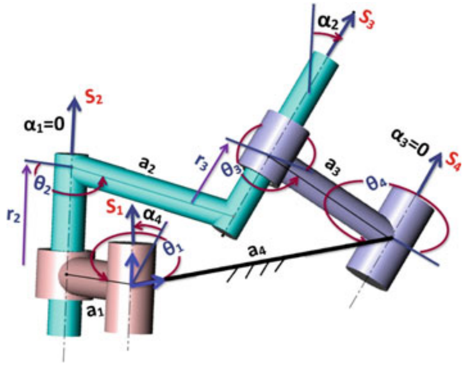
3 The Closed RCCR Linkage

The closed RC-CR linkage is overconstrained and able to move with one degree of freedom [19] when the cylindrical (C) and revolute (R) joints of each pair are parallel, while both pairs are skew one to each other, see Fig. 1.

For arbitrarly-positioned axes, the mobility of this spatial four-bar linkage is, using CKG formula, equal to zero. However it is possible to obtain a one-dof linkage for some special geometry.

The geometric features and the joint variable functions can be derived, for instance, by equating the forward kinematics of both RC serial chains at their end-effector [19]. According to the coordinate frame shown in Fig. 1, and applying the needed condition

Fig. 1 RCCR linkage



of parallel axes, that is, $\alpha_1 = \alpha_3 = 0$, the forward kinematics of both RC chains 1–2 and 4–3 are,

$$[D_{RC1}] = \begin{bmatrix} c(\theta_1+\theta_2) & -s(\theta_1+\theta_2)c\alpha_2 & s(\theta_1+\theta_2)s\alpha_2 & a_2c(\theta_1+\theta_2)+a_1c\theta_1 \\ s(\theta_1+\theta_2) & c(\theta_1+\theta_2)c\alpha_2 & -c(\theta_1+\theta_2)s\alpha_2 & a_2s(\theta_1+\theta_2)+a_1s\theta_1 \\ 0 & s\alpha_2 & c\alpha_2 & r_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

$$[D_{RC2}] = \begin{bmatrix} c(\theta_3+\theta_4) & s(\theta_3+\theta_4) & 0 & -a_3c\theta_4-a_4 \\ -s(\theta_3+\theta_4)c\alpha_4 & c(\theta_3+\theta_4)c\alpha_4 & s\alpha_4 & -r_3s\alpha_4+a_3s\theta_4c\alpha_4 \\ s(\theta_3+\theta_4)s\alpha_4 & -c(\theta_3+\theta_4)s\alpha_4 & c\alpha_4 & -r_3c\alpha_4-a_3s\theta_4s\alpha_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

where s and c stand for the sin and cos functions respectively.

Some geometrical constraints and angular relations are obtained from equating these two transformations,

$$\begin{aligned} &\cos\alpha_2 = \cos\alpha_4 \Longrightarrow \alpha_4 = \pm\alpha_2, \\ &\cos(\theta_1+\theta_2) = \pm 1, \quad \sin(\theta_1+\theta_2) = 0 \Longrightarrow \theta_2 = n*\pi - \theta_1, \\ &\cos(\theta_3+\theta_4) = \pm 1, \quad \sin(\theta_3+\theta_4) = 0 \Longrightarrow \theta_3 = n*\pi - \theta_4, \end{aligned} \quad (6)$$

in which the directions of the fixed joints are parallel to the directions of the moving joints, with coupled rotation angles. We can also derive the following joint variable relations:

$$\begin{aligned} &\theta_4 = \pm\arccos\left(\frac{\pm a_2 - a_4 - a_1\cos\theta_1}{a_3}\right) \\ &r_3 = \frac{a_1\sin\theta_1 - a_3\cos\alpha_4\sin\theta_4}{\sin\alpha_4}, \quad r_2 = \frac{a_1\cos\alpha_4\sin\theta_1 - a_3\sin\theta_4}{\sin\alpha_4} \end{aligned} \quad (7)$$

4 The Workspace of Finite Displacements of the RC-CR Linkage

We denote the RC chain with parallel axes and angles $\theta_2 = -\theta_1$ a *parallel RC chain*. For solving the design problem, it is advantageous to compute the workspace of relative displacements with respect to a reference configuration. The reference configuration can be arbitrarily selected, with $\Delta r_2 = r_2 - r_{20}$ and $\Delta\theta_i = \theta_i - \theta_{i0}$. The workspace of relative displacements a parallel RC chain is

$$\begin{aligned} \hat{D} &= \hat{R}(\Delta\theta_1)\hat{C}(\Delta\theta_2, \Delta r_2) \\ &= 1 + \varepsilon\frac{1}{2}\big(\Delta r_2\mathbf{s}_1 + (\cos\Delta\theta_1 - 1)(\mathbf{c}_2 - \mathbf{c}_1) - \sin\Delta\theta_1(\mathbf{c}_2 - \mathbf{c}_1)\times\mathbf{s}_1\big), \end{aligned} \quad (8)$$

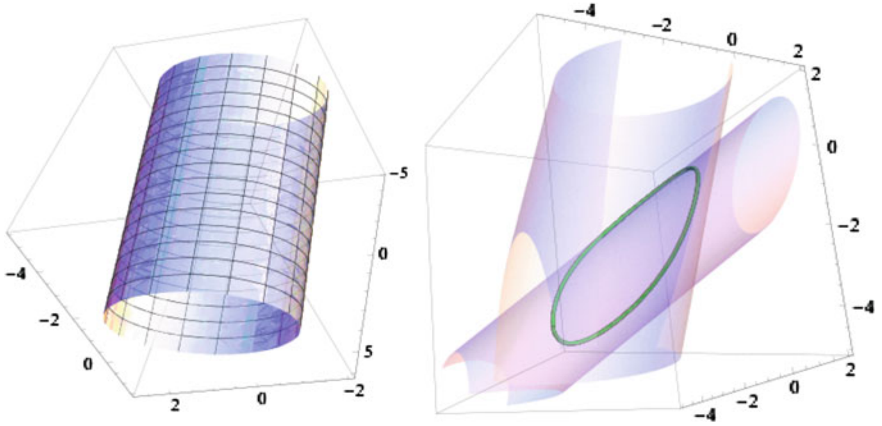


Fig. 2 Workspace of relative displacements for the parallel RC chain, *left*; for the RC-CR linkage, *right*

where $\hat{R}(\Delta\theta_1)$ is a rotation about an axis with Plucker coordinates S_1, and $\hat{C}(\Delta\theta_2, \Delta r_2)$ is a rotation and a translation about and along an axis with Plucker coordinates S_2. Both axes share the same direction $\mathbf{s}_1$ and their rotations are $\Delta\theta_1$ and $\Delta\theta_2 = -\Delta\theta_1$; the points $\mathbf{c}_1$ and $\mathbf{c}_2$ are any points on the axes along a common normal line. Notice that the relative displacements have no change in orientation, so that the chain has a constant-orientation workspace.

The workspace of relative translations for the RCCR linkage is given by the intersection of the workspaces of two parallel RC chains. Figure 2 shows the workspace of a parallel RC chain and the intersection workspace for two chains.

In order to characterize the workspace of the parallel RC chain, we perform implicitization in Eq. (8) to eliminate the joint variables θ_1 and r_2. The elimination yields a quadratic surface of expression

$$\begin{aligned}\mathcal{Q}(x, y, z) :&(s_{1y}^2 + s_{1z}^2)x^2 + (s_{1x}^2 + s_{1z}^2)y^2 + (s_{1x}^2 + s_{1y}^2)z^2 \\ &- 2s_{1x}s_{1y}xy - 2s_{1x}s_{1z}xz - 2s_{1y}s_{1z}yz + c_{21x}x + c_{21y}y + c_{21z}z = 0,\end{aligned} \tag{9}$$

where (x, y, z) is a point of the $\mathbb{R}^3$ space of relative translations, $\mathbf{s}_1 = (s_{1x}, s_{1y}, s_{1z})$ is the direction for both joints, and $\mathbf{c}_{21} = \mathbf{c}_2 - \mathbf{c}_1 = (c_{21x}, c_{21y}, c_{21z})$ is the vector along the common normal between both joints.

This surface is classified as a circular cylinder, with radius $R = \sqrt{\mathbf{c}_{21} \cdot \mathbf{c}_{21}}$ and passing through the origin, which corresponds to the zero relative displacement. The intersection of two such circular cylinders yields a quartic curve which is the workspace of the RCCR linkage.

5 Dimensional Synthesis for the RCCR Linkage

The workspace of the RCCR linkage is a constant-orientation curve, and hence the synthesis problem can be reduced to a point-path synthesis problem. The point-path synthesis problem is stated as follows: given an initial point $\mathbf{P}_1$ (which we will use as reference configuration), relative displacements of the RC-CR chain will move this point to the rest of task points $\mathbf{P}_2, \mathbf{P}_3, \ldots, \mathbf{P}_n$.

The action of the chain on this point can be calculated using one of the conjugations in the Clifford algebra. If the forward kinematics of relative displacements of Eq. (8) is denoted by $\hat{D} = 1 + \varepsilon \mathbf{d}$, then

$$\mathbf{P}_i = \hat{D}\hat{P}_1\hat{D}^*, \quad i = 2, \ldots, n, \tag{10}$$

where $\hat{P}_1 = (1 + \varepsilon \mathbf{P}_1)$ is the dual quaternion expression of the point $\mathbf{P}_1$, and the conjugation yields

$$\left(1 + \varepsilon \frac{1}{2}\mathbf{d}\right)(1 + \varepsilon \mathbf{P}_1)\left(1 + \varepsilon \frac{1}{2}\mathbf{d}\right) = 1 + \varepsilon\left(\mathbf{P}_1 + \mathbf{d}\right) \tag{11}$$

Notice that this is equivalent to equating the relative translations, $\mathbf{d} = \mathbf{P}_i - \mathbf{P}_1$ for $i = 2, \ldots, n$.

Let us consider the case of the parallel RC chain, in which the values of θ_1 and r_2 are independent. This results in $3(n-1)$ design equations, with the structural variables $\mathbf{s}_1$ and $\mathbf{c}_{21} = \mathbf{c}_2 - \mathbf{c}_1$ and the joint variables r_2 and θ_1 for each point, for a total of $4 + 2(n-1)$ unknowns. Up to $n = 5$ point-positions can be defined in order to do exact point-path synthesis.

The standard finite-position synthesis technique equates the parameterized expression of the translation workspace to the task relative translations. However in this case, the implicit equation for the workspace has a simpler expression as a function of the chain structural parameters.

The four relative translations $\mathbf{P}_i$, $i = 2, \ldots, 5$, are used to shape the circular cylinder, and define the parallel RC chain that creates the motion. The system of design equations consists of six quadratic equations in six unknowns,

$$\begin{aligned} &\mathcal{Q}(\mathbf{P}_i) = 0, \quad i = 2, 3, 4, 5; \\ &\mathbf{s}_1 \cdot \mathbf{s}_1 = 1, \mathbf{s}_1 \cdot \mathbf{c}_{21} = 0, \end{aligned} \tag{12}$$

which are easy enough to be solved using algebraic techniques. There are at most 6 different solutions.

Table 1 Goal points and solution RC chains

Point	Coordinates	
P_1	$(2.31, 3.84, -1.08)$	
P_2	$(0.34, -2.81, 0.89)$	
P_3	$(2.21, -3.47, 0.63)$	
P_4	$(2.18, 3.77, -2.66)$	
P_5	$(-1.22, -1.42, -2.22)$	
Solution	$\mathbf{s}_1$	$\mathbf{c}_2 - \mathbf{c}_1$
1	$(-0.09, -0.04, -0.99)$	$(0.03, 7.38, -0.30)$
2	$(-0.54, 0.42, -0.73)$	$(3.86, 5.19, 0.15)$
3	$(-0.10, 0.99, -0.08)$	$(4.11, 0.51, 1.22)$
4	$(0.54, 0.84, -0.03)$	$(-4.32, 2.85, 1.82)$

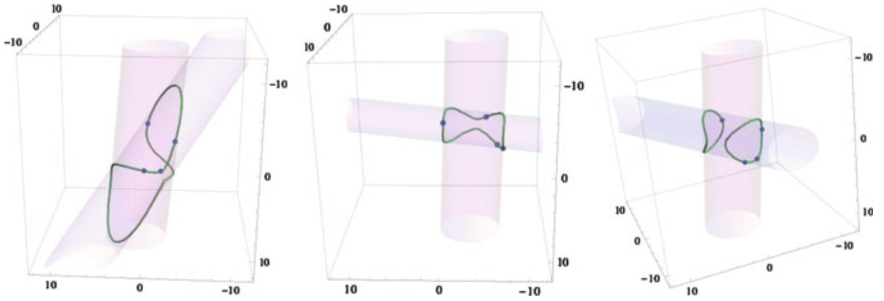


Fig. 3 Three of the six RCCR workspaces with the task points

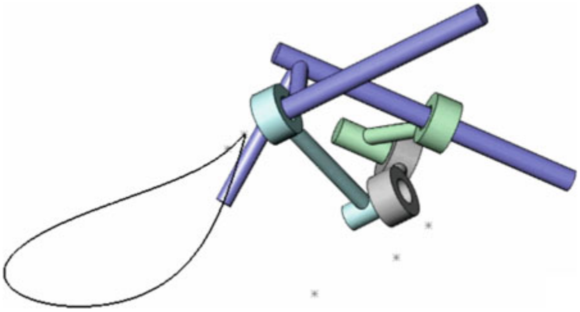


Fig. 4 RCCR linkage created with chains 2 and 3, passing through points on one of its two circuits

6 Example

The task points used in this example are presented in Table 1. The system of Eq. (12) yields four real solutions, presented in Table 1, which can be assembled in pairs in order to create RCCR linkages. The number of different workspaces obtained is six.

The solution workspaces for the parallel RC chains can also be intersected pairwise in order to create workspaces for the RCCR chains. The workspace equations can be used to visually assess the trajectory of the linkage and also to check for circuit defect. Figure 3 shows three of the six possible combinations for this example, and Fig. 4 shows one of the possible RCCR chains.

7 Conclusions

This paper presents an exact-workspace synthesis method for the RCCR linkage, an overconstrained mechanism with mobility one. The implicitization of the algebraic equations of the workspace of relative displacements yields a circular cylinder that can be shaped using a set of finite positions. This simple case, in which the workspace has a constant orientation, is a building block towards a more general methodology for the exact workspace synthesis of spatial linkages.

References

1. Affi, Z., Romdhane, L., Maalej, A.: Dimensional synthesis of a 3-translational-dof in-parallel manipulator for a desired workspace. Eur. J. Mech. A. Solids **23**, 311–324 (2004)
2. Brunnthaler, K.: Synthesis of 4r linkages using kinematic mapping. Ph.D. Thesis, Institute for basic sciences and engineering, University of Innsbruck, Innsbruck (2006)
3. Chablat, D., Wenger, P.: Architecture optimization of a 3-dof parallel mechanism for machining applications, the orthoglide. IEEE Trans. Robot. Autom. **19**(3), 403–410 (2003)
4. Gracia, P.A., McCarthy, J.M.: The kinematic synthesis of spatial serial chains using Clifford algebra exponentials. In: Proceedings of the Institution of Mechanical Engineers, Part C. J. Mech. Eng. Sci. **220**(7), 953–968 (2006)
5. Gracia, A.P.: Synthesis of spatial RPRP closed linkages for a given screw system. ASME J. Mech. Robot. **3**(2), 1–8 (2011)
6. Hayes, M.J.D., Luu, T., Chang, X.W.: Kinematic mapping application to approximate type and dimension synthesis of planar mechanisms. In: Lenarcic, J., Galletti, C. (eds.) On Advances in Robot Kinematics, Kluwer Academic Publisher, Dordrecht, The Netherlands (2004)
7. Huang, C.: The cylindroid associated with finite motions of the Bennett mechanism. In: Proceedings of the ASME Design Engineering Technical Conferences, USA (1996)
8. Huang, T., Li, M., Zhao, X.M., Mei, J., Chetwynd, D.G., Hu, S.J.: Conceptual design and dimensional synthesis for a 3-dof module of the trivariant - a novel 5-dof reconfigurable hybrid robot. IEEE Trans. Robot. **21**(3), 449–456 (2005)
9. Huang, C.: Linear property of the screw surface of the spatial RPRP linkage. ASME J. Mech. Des. **128**, 581–586 (2006)
10. Husty, M.L., Pfurner, M., Schrocker, H.P., Brunnthaler, K.: Algebraic methods in mechanism analysis and synthesis. Robotica **25**, 661–675 (2007)
11. Kim, H.S., Tsai, L.W.: Kinematic synthesis of a spatial 3-RPS parallel manipulator. ASME J. Mech. Des. **125**, 92–97 (2003)
12. Kim, H.S., Tsai, L.W.: Design optimization of a cartesian parallel manipulator. ASME J. Mech. Des. **125**, 43–51 (2003)
13. Kosinska, A., Galicki, M., Kedzior, K.: Design and optimization of parameters of delta-4 parallel manipulator for a given workspace. J. Rob. Syst. **20**(9), 539–548 (2003)
14. Merlet, J.P.: Optimal design of robots. In: Proceedings of Robotics: Science and Systems, USA, June 2005

15. Parkin, I.A.: A third conformation with the screw systems: finite twist displacements of a directed line and point. Mech. Mach. Theory **27**, 177–188 (1992)
16. Rao, N.M., Rao, K.M.: Dimensional synthesis of a 3-RPS parallel manipulator for a prescribed range of motion of spherical joints. Mech. Mach. Theory **44**, 477–486 (2009)
17. Ravani, B., Roth, B.: Motion synthesis using kinematic mappings. ASME J. Mech. Trans. Autom. Des. **105**(3), 460–467 (1983)
18. Schrocker, H.P., Husty, M.L., McCarthy, J.M.: Kinematic mapping based assembly mode evaluation of planar four-bar mechanisms. AMSE J. Mech. Des. **129**, 924–929 (2007)
19. Waldron, K.J.: A study of overconstrained linkage geometry by solution of closure equations - part II- four-bar linkages with lower pair joints other than screw joints. Mech. Mach. Theory **8**, 233–247 (1972)
20. Wolbrecht, E., Su, H.J., Perez, A., McCarthy, J.M.: Geometric design of symmetric 3-RRS constrained parallel platforms. In: ASME (ed.) Proceedings of the 2004 ASME International Mechanical Engineering Congress and Exposition, Anaheim, 13–19 November, 2004
21. Wu, J., Purwar, A., Ge, Q.J.: nteractive dimensional synthesis and motion design of planar 6r single-loop closed chains via constraint manifold modification. ASME J. Mecha. Robot. **2**(3), 1–8 (2010)

A Closed-Form Solution of Spatial Sliders for Rigid-Body Guidance

Chintien Huang, Weiche Huang and Gökhan Kiper

Abstract This paper presents a closed-form solution to the four-position synthesis problem by using a spatial slider, which is a spatial dyad of two perpendicularly intersected cylindrical joints. We utilize the dialytic elimination method to simplify the synthesis equations and to obtain a univariate ninth degree polynomial equation. Among the nine sets of solutions, two of them are infinite, and one is the displacement screw from the first position to the second position. Therefore, we have at most six real solutions that can be used to design spatial sliders for the four-position synthesis problem. A numerical example is provided in to demonstrate the validity of the solution procedure.

Keywords Spatial slider · Cylindrical joint · Rigid-body guidance · Dialytic elimination

1 Introduction

The rigid-body guidance problem is the central problem in kinematic synthesis of linkages. In planar kinematics, given several positions of a body, one can design a crank or a slider to guide the body through the prescribed positions [1, 2]. Spatial counterpart of the planar synthesis problem has been extensively studied too; various spatial dyads have been investigated for rigid-body guidance [5, 7, 8]. This paper focuses on the spatial generalization of the planar slider, and we investigate the

C. Huang (✉) · W. Huang
Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan
e-mail: chuang@mail.ncku.edu.tw

W. Huang
e-mail: tony77113@gmail.com

G. Kiper
Department of Mechanical Engineering, İzmir Institute of Technology, İzmir, Turkey
e-mail: gokhankiper@iyte.edu.tr

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_40,

solution to the synthesis equations of the special cylindrical-cylindrical dyad, with the joint axes perpendicularly intersected, for rigid body guidance.

By using screw geometry, the spatial generalization of the synthesis of the planar crank has led to elegant geometric and algebraic results in computational kinematics [3, 4, 7]. Built upon these results, this paper utilizes a special screw triangle [5] to derive synthesis equations of the spatial slider and seeks to find all possible solutions to the design equations. However, due to the complexity of the equations, analytical solutions cannot be obtained. Instead, we utilize the dialytic elimination method [6] to find a closed-form solution, in which a univariate polynomial equation is obtained. The degree of the polynomial equation is the number of all possible solutions; however, we may need to exclude some extraneous roots.

This paper is organized as follows: first, we derive the synthesis equations and show that the maximum number of design positions is four. Second, we divide the equations into two uncoupled groups and use one of them to solve for the direction vectors of the joints. To further simplify the solution procedure, we superimpose two three-position problems instead of solving a four-position problem. Third, based on the obtained solutions of the direction vectors, we utilize the second group of equations to obtain the position vectors of points on the joint axes. Finally, a numerical example is provided to verify the solution procedure.

2 Design Equations for the Synthesis of Spatial Sliders

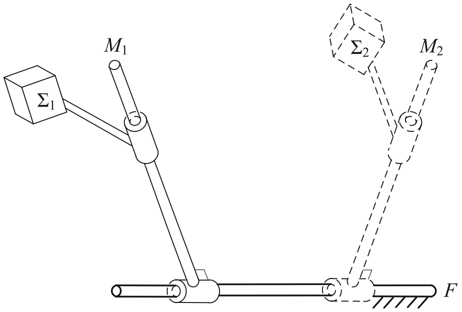
Figure 1 shows a spatial slider guiding a rigid body from one position to another. The spatial slider consists of two perpendicularly intersected cylindrical joints. The ground (fixed) and moving joints are denoted by F and M_1, respectively. Let the displacement screw for displacing the body from position Σ_1 to position Σ_2 be denoted by $\$_{12}$, and the rotation and translation parameters be θ_{12} and d_{12}, respectively. The direction vector of $\$_{12}$ is denoted by $\hat{\mathbf{s}}_{12}$, and the position vector of a point on $\$_{12}$ is denoted by $\mathbf{A}_{12}$. The geometry of the relation among F, M_1, and $\$_{12}$ is a special screw triangle illustrated in Fig. 2. The following equations can be obtained based on the geometric relation of the screw triangle [3]:

$$\hat{\mathbf{F}} \cdot \hat{\mathbf{M}}_1 = 0 \tag{1}$$

$$\hat{\mathbf{F}} \cdot (\hat{\mathbf{s}}_{12} \times \hat{\mathbf{M}}_1) + \tan\frac{\theta_{12}}{2}(\hat{\mathbf{F}} \times \hat{\mathbf{s}}_{12}) \cdot (\hat{\mathbf{s}}_{12} \times \hat{\mathbf{M}}_1) = 0 \tag{2}$$

$$\hat{\mathbf{F}} \cdot (\mathbf{Q}_1 \times \hat{\mathbf{M}}_1) + \hat{\mathbf{M}}_1 \cdot (\mathbf{G} \times \hat{\mathbf{F}}) = 0 \tag{3}$$

Fig. 1 A rigid body guided by a spatial slider



$$[1-(\hat{\mathbf{s}}_{12}\cdot\hat{\mathbf{F}})^2]\cdot\{[\hat{\mathbf{s}}_{12}-(\hat{\mathbf{s}}_{12}\cdot\hat{\mathbf{M}}_1)\hat{\mathbf{M}}_1]\cdot(\mathbf{Q}_1-\mathbf{A}_{12})\}-[1-(\hat{\mathbf{s}}_{12}\cdot\hat{\mathbf{M}}_1)^2]\cdot\{[\hat{\mathbf{s}}_{12} \\ -(\hat{\mathbf{s}}_{12}\cdot\hat{\mathbf{F}})\hat{\mathbf{F}}]\cdot(\mathbf{G}-\mathbf{A}_{12})\}+\frac{d_{12}}{2}[1-(\hat{\mathbf{s}}_{12}\cdot\hat{\mathbf{F}})^2]\cdot[1-(\hat{\mathbf{s}}_{12}\cdot\hat{\mathbf{M}}_1)^2]=0 \tag{4}$$

$$\hat{\mathbf{M}}_1^2=m_{1x}^2+m_{1y}^2+m_{1z}^2=1 \tag{5}$$

$$\hat{\mathbf{F}}^2=f_x^2+f_y^2+f_z^2=1 \tag{6}$$

$$\hat{\mathbf{M}}_1\cdot\mathbf{Q}_1=m_{1x}\cdot q_{1x}+m_{1y}\cdot q_{1y}+m_{1z}\cdot q_{1z}=0 \tag{7}$$

$$\hat{\mathbf{F}}\cdot\mathbf{G}=f_x\cdot g_x+f_y\cdot g_y+f_z\cdot g_z=0 \tag{8}$$

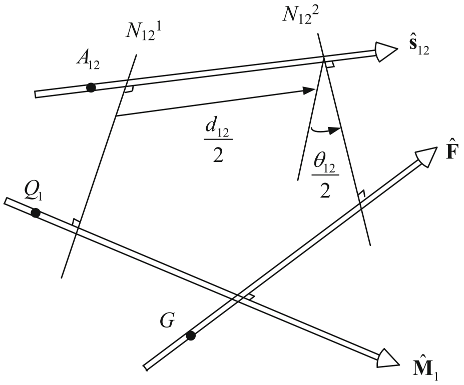
Note that we use the symbol ^ to denote unit vectors or unit screws. The unknowns in the above equations are the direction vector of the moving joint $\hat{\mathbf{M}}_1(m_{1x}, m_{1y}, m_{1z})$, the position vector of a point on the moving axis $\mathbf{Q}_1(q_{1x}, q_{1y}, q_{1z})$, the direction vector of the fixed joint $\hat{\mathbf{F}}(f_x, f_y, f_z)$, and the position vector of a point on the fixed axis $\mathbf{G}(g_x, g_y, g_z)$. Equations (1) and (3) indicate that the joint axes are intersected perpendicularly. Equations (7) and (8) constrain the points on the axes in such a way that the position vectors of the points must be perpendicular to the joint direction vectors.

There are 12 unknowns but only eight equations; therefore, we can specify two more positions by displacement screws $\$_{13}$ (with rotation θ_{13} and translation d_{13}) and $\$_{14}$ (with rotation θ_{14} and translation d_{14}). As a result, we have four more design equations as follows:

$$\hat{\mathbf{F}}\cdot(\hat{\mathbf{s}}_{13}\times\hat{\mathbf{M}}_1)+\tan\frac{\theta_{13}}{2}(\hat{\mathbf{F}}\times\hat{\mathbf{s}}_{13})\cdot(\hat{\mathbf{s}}_{13}\times\hat{\mathbf{M}}_1)=0 \tag{9}$$

$$\hat{\mathbf{F}}\cdot(\hat{\mathbf{s}}_{14}\times\hat{\mathbf{M}}_1)+\tan\frac{\theta_{14}}{2}(\hat{\mathbf{F}}\times\hat{\mathbf{s}}_{14})\cdot(\hat{\mathbf{s}}_{14}\times\hat{\mathbf{M}}_1)=0 \tag{10}$$

Fig. 2 Screw geometry of the spatial slider



$$[1-(\hat{\mathbf{s}}_{13}\cdot\hat{\mathbf{F}})^2]\cdot\{[\hat{\mathbf{s}}_{13}-(\hat{\mathbf{s}}_{13}\cdot\hat{\mathbf{M}}_1)\hat{\mathbf{M}}_1]\cdot(\mathbf{Q}_1-\mathbf{A}_{13})\}-[1-(\hat{\mathbf{s}}_{13}\cdot\hat{\mathbf{M}}_1)^2]\cdot\{[\hat{\mathbf{s}}_{13} \\ -(\hat{\mathbf{s}}_{13}\cdot\hat{\mathbf{F}})\hat{\mathbf{F}}]\cdot(\mathbf{G}-\mathbf{A}_{13})\}+\frac{d_{13}}{2}[1-(\hat{\mathbf{s}}_{13}\cdot\hat{\mathbf{F}})^2]\cdot[1-(\hat{\mathbf{s}}_{13}\cdot\hat{\mathbf{M}}_1)^2]=0 \tag{11}$$

$$[1-(\hat{\mathbf{s}}_{14}\cdot\hat{\mathbf{F}})^2]\cdot\{[\hat{\mathbf{s}}_{14}-(\hat{\mathbf{s}}_{14}\cdot\hat{\mathbf{M}}_1)\hat{\mathbf{M}}_1]\cdot(\mathbf{Q}_1-\mathbf{A}_{14})\}-[1-(\hat{\mathbf{s}}_{14}\cdot\hat{\mathbf{M}}_1)^2]\cdot\{[\hat{\mathbf{s}}_{14} \\ -(\hat{\mathbf{s}}_{14}\cdot\hat{\mathbf{F}})\hat{\mathbf{F}}]\cdot(\mathbf{G}-\mathbf{A}_{14})\}+\frac{d_{14}}{2}[1-(\hat{\mathbf{s}}_{14}\cdot\hat{\mathbf{F}})^2]\cdot[1-(\hat{\mathbf{s}}_{14}\cdot\hat{\mathbf{M}}_1)^2]=0 \tag{12}$$

3 Solutions of Direction Vectors of the Joint Axes

The 12 design equations are too complicated to be solved simultaneously. However, we can deal with only three positions at a time and superimpose two three-position problems. For example, we can first design the spatial slider to guide the body through positions 1, 2, and 3 and then design the spatial slider to guide the body through positions 1, 2, and 4. Notice that the 12 equations can be decoupled by first solving Eqs. (1, 2, 5, 6, 9), and (10) for the direction vectors. We can then utilize the remaining equations to solve for the position vectors.

For positions 1, 2, and 3, substituting Eq. (1) into Eqs. (2) and (9) and rearranging the equations gives:

$$[\hat{\mathbf{s}}_{12}\times\hat{\mathbf{M}}_1+\tan\frac{\theta_{12}}{2}(\hat{\mathbf{s}}_{12}\cdot\hat{\mathbf{M}}_1)\hat{\mathbf{s}}_{12}]\cdot\hat{\mathbf{F}}=0 \tag{13}$$

$$[\hat{\mathbf{s}}_{13}\times\hat{\mathbf{M}}_1+\tan\frac{\theta_{13}}{2}(\hat{\mathbf{s}}_{13}\cdot\hat{\mathbf{M}}_1)\hat{\mathbf{s}}_{13}]\cdot\hat{\mathbf{F}}=0 \tag{14}$$

We can rearrange Eqs. (1, 13), and (14) in a matrix form as follows:

$$\left[\mathbf{K}_m\right]\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

Note that the unit direction vectors of the specified screws $\hat{\mathbf{s}}_{12}(s_{12x}, s_{12y}, s_{12z})$ and $\hat{\mathbf{s}}_{13}(s_{13x}, s_{13y}, s_{13z})$ are known parameters. For the linear system, Eq. (15), to have non-trivial solutions, the determinant of $\mathbf{K}_m$ must be zero:

$$\begin{aligned} \det(\mathbf{K}_m)_{123} &= \hat{\mathbf{M}}_1 \cdot \{[\hat{\mathbf{s}}_{12} \times \hat{\mathbf{M}}_1 + \tan\frac{\theta_{12}}{2}(\hat{\mathbf{s}}_{12} \cdot \hat{\mathbf{M}}_1)\hat{\mathbf{s}}_{12}] \\ &\quad \times [\hat{\mathbf{s}}_{13} \times \hat{\mathbf{M}}_1 + \tan\frac{\theta_{13}}{2}(\hat{\mathbf{s}}_{13} \cdot \hat{\mathbf{M}}_1)\hat{\mathbf{s}}_{13}]\} \\ &= 0 \end{aligned} \tag{16}$$

Expanding the determinant gives:

$$\begin{aligned} a_1 m_{1x} m_{1y}^2 + a_2 m_{1x}^2 m_{1y} + a_3 m_{1x}^3 + a_4 m_{1y}^3 + a_5 m_{1x}^2 m_{1z} + a_6 m_{1y}^2 m_{1z} \\ + a_7 m_{1x} m_{1y} m_{1z} + a_8 m_{1x} m_{1z}^2 + a_9 m_{1y} m_{1z}^2 + a_{10} m_{1z}^3 = 0 \end{aligned} \tag{17}$$

where the coefficients a_i, $i = 1, 2, \ldots$,10, are functions of known parameters.

Similarly, for positions 1, 2, and 4, we obtain the following equation by solving Eqs. (1, 2), and (10):

$$\begin{aligned} \det(\mathbf{K}_m)_{124} &= \hat{\mathbf{M}}_1 \cdot \{[\hat{\mathbf{s}}_{12} \times \hat{\mathbf{M}}_1 + \tan\frac{\theta_{12}}{2}(\hat{\mathbf{s}}_{12} \cdot \hat{\mathbf{M}}_1)\hat{\mathbf{s}}_{12}] \\ &\quad \times [\hat{\mathbf{s}}_{14} \times \hat{\mathbf{M}}_1 + \tan\frac{\theta_{14}}{2}(\hat{\mathbf{s}}_{14} \cdot \hat{\mathbf{M}}_1)\hat{\mathbf{s}}_{14}]\} \\ &= 0 \end{aligned} \tag{18}$$

Expanding the above equation gives

$$\begin{aligned} b_1 m_{1x} m_{1y}^2 + b_2 m_{1x}^2 m_{1y} + b_3 m_{1x}^3 + b_4 m_{1y}^3 + b_5 m_{1x}^2 m_{1z} + b_6 m_{1y}^2 m_{1z} \\ + b_7 m_{1x} m_{1y} m_{1z} + b_8 m_{1x} m_{1z}^2 + b_9 m_{1y} m_{1z}^2 + b_{10} m_{1z}^3 = 0 \end{aligned} \tag{19}$$

where the coefficients b_i, $i = 1, 2, \ldots$,10, are functions of known parameters.

Next, we employ the dialytic elimination method [5] to seek a closed-form solution of $\hat{\mathbf{M}}_1(m_{1x}, m_{1y}, m_{1z})$. For the purpose of employing the dialytic elimination method, we denote Eqs. (17) and (19) with f_1 and f_2. In addition, to simplify the notations for the dialytic elimination process, let $m_{1x} = x$, $m_{1y} = y$, $m_{1z} = z$. We have

$$f_1 : a_1xy^2 + a_2x^2y + a_3x^3 + a_4y^3 + a_5x^2z + a_6y^2z + a_7xyz + a_8xz^2 + a_9yz^2 + a_{10}z^3 = 0 \tag{20}$$

$$f_2 : b_1xy^2 + b_2x^2y + b_3x^3 + b_4y^3 + b_5x^2z + b_6y^2z + b_7xyz + b_8xz^2 + b_9yz^2 + b_{10}z^3 = 0 \tag{21}$$

Note that f_1 and f_2 are homogeneous equations. Letting $y = px$ and $z = qx$ and substituting them into f_1 and f_2 gives:

$$f_1 : a_1p^2 + a_2p + a_3 + a_4p^3 + a_5q + a_6p^2q + a_7pq + a_8q^2 + a_9pq^2 + a_{10}q^3 = 0 \tag{22}$$

$$f_2 : b_1p^2 + b_2p + b_3 + b_4p^3 + b_5q + b_6p^2q + b_7pq + b_8q^2 + b_9pq^2 + b_{10}q^3 = 0 \tag{23}$$

Suppressing the variable p into the coefficients gives:

$$f_1 : \mathrm{A}_0q^3 + \mathrm{A}_1q^2 + \mathrm{A}_2q + \mathrm{A}_3 = 0 \tag{24}$$

$$f_2 : \mathrm{B}_0q^3 + \mathrm{B}_1q^2 + \mathrm{B}_2q + \mathrm{B}_3 = 0 \tag{25}$$

Manipulating f_1 and f_2 in the following manners yields three equations:

$$f_1 \times \mathrm{B}_0 - f_2 \times \mathrm{A}_0 : (\mathrm{A}_1\mathrm{B}_0 - \mathrm{B}_1\mathrm{A}_0)q^2 + (\mathrm{A}_2\mathrm{B}_0 - \mathrm{B}_2\mathrm{A}_0)q + (\mathrm{A}_3\mathrm{B}_0 - \mathrm{B}_3\mathrm{A}_0) = 0 \tag{26}$$

$$f_1 \times \mathrm{B}_3 - f_2 \times \mathrm{A}_3 : (\mathrm{A}_0\mathrm{B}_3 - \mathrm{B}_0\mathrm{A}_3)q^2 + (\mathrm{A}_1\mathrm{B}_3 - \mathrm{B}_1\mathrm{A}_3)q + (\mathrm{A}_2\mathrm{B}_3 - \mathrm{B}_2\mathrm{A}_3) = 0 \tag{27}$$

$$\begin{aligned} f_1 \times (\mathrm{B}_0 \cdot q + \mathrm{B}_1) - f_2 \times (\mathrm{A}_0 \cdot q + \mathrm{A}_1) : (\mathrm{A}_2\mathrm{B}_0 - \mathrm{A}_0\mathrm{B}_2)q^2 \\ + (\mathrm{A}_2\mathrm{B}_1 - \mathrm{A}_1\mathrm{B}_2 \\ + \mathrm{A}_3\mathrm{B}_0 - \mathrm{A}_0\mathrm{B}_3)q + (\mathrm{A}_3\mathrm{B}_1 - \mathrm{A}_1\mathrm{B}_3) = 0 \end{aligned} \tag{28}$$

Rearranging Eqs. (26–28) in a matrix form gives:

$$\begin{bmatrix} (\mathrm{A}_1\mathrm{B}_0 - \mathrm{B}_1\mathrm{A}_0) & (\mathrm{A}_2\mathrm{B}_0 - \mathrm{B}_2\mathrm{A}_0) & (\mathrm{A}_3\mathrm{B}_0 - \mathrm{B}_3\mathrm{A}_0) \\ (\mathrm{A}_0\mathrm{B}_3 - \mathrm{B}_0\mathrm{A}_3) & (\mathrm{A}_1\mathrm{B}_3 - \mathrm{B}_1\mathrm{A}_3) & (\mathrm{A}_2\mathrm{B}_3 - \mathrm{B}_2\mathrm{A}_3) \\ (\mathrm{A}_2\mathrm{B}_0 - \mathrm{A}_0\mathrm{B}_2) & (\mathrm{A}_2\mathrm{B}_1 - \mathrm{A}_1\mathrm{B}_2 + \mathrm{A}_3\mathrm{B}_0 - \mathrm{A}_0\mathrm{B}_3) & (\mathrm{A}_3\mathrm{B}_1 - \mathrm{A}_1\mathrm{B}_3) \end{bmatrix} \begin{bmatrix} q^2 \\ q \\ 1 \end{bmatrix} = 0 \tag{29}$$

Again, to have non-trivial solutions of q, the determinant of the coefficient matrix must be zero, which gives a ninth degree polynomial equation in p. Upon substituting the solution of p into Eq. (29), we obtain a corresponding q. Substituting the values of p and q into Eq. (5) gives:

$$x = m_{1x} = \sqrt{1/(1 + p^2 + q^2)} \tag{30}$$

which leads to the direction vector $\hat{\mathbf{M}}_1 = (m_{1x}, m_{1y}, m_{1z}) = (x, y, z) = (x, px, qx)$.

Among the nine sets of solution of p and q, there are two complex-number solutions satisfying the following constraint:

$$p^2 + q^2 = -1 \tag{31}$$

which leads to two infinite solutions of $\hat{\mathbf{M}}_1(m_{1x}, m_{1y}, m_{1z})$. One of the remaining seven sets of solution is $\hat{\mathbf{s}}_{12}(s_{12x}, s_{12y}, s_{12z})$, which cannot be used for the moving joint. Therefore, we have a total of six finite solutions of $\hat{\mathbf{M}}_1(m_{1x}, m_{1y}, m_{1z})$. Substituting each solution of $\hat{\mathbf{M}}_1(m_{1x}, m_{1y}, m_{1z})$ into Eqs. (1, 2), and (6) gives a corresponding $\hat{\mathbf{F}}(f_x, f_y, f_z)$.

4 Solutions of Position Vectors and Numerical Example

Once the solutions of direction vectors are obtained, we can substitute them into Eqs. (3, 4, 7, 8, 11), and (12) to solve for the position vectors $\mathbf{Q}_1(q_{1x}, q_{1y}, q_{1z})$ and $\mathbf{G}(g_x, g_y, g_z)$. First, we rearrange Eqs. (3, 4, 8), and (11) in to a matrix form as follows:

$$[\mathbf{K}_q] \begin{bmatrix} g_x \\ g_y \\ g_z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{32}$$

where $\mathbf{K}_q$ is of the following form

$$\mathbf{K}_q = \begin{bmatrix} f_x & f_y & f_z & 0 \\ k_{q21} & k_{q22} & k_{q23} & k_{q24} \\ k_{q31} & k_{q32} & k_{q33} & k_{q34} \\ k_{q41} & k_{q42} & k_{q43} & k_{q44} \end{bmatrix} \tag{33}$$

In order for Eq. (32) to have non-trivial solutions, the determinant of $\mathbf{K}_q$ must be zero, which gives the following linear equation in (q_{1x}, q_{1y}, q_{1z}):

$$k_1 q_{1x} + k_2 q_{1y} + k_3 q_{1z} + k_4 = 0 \tag{34}$$

Table 1 Displacement screws for the four specified positions

Screw	Direction vector	Position vector of a point	Translation	Rotation
12	(0.103, 0.737, −0.668)	(0.941, 0.521, −0.419)	1.384	183.616°
13	(−0.208, −0.838, −0.505)	(−0.610, 0.837, 0.123)	1.899	178.431°
14	(−0.832, 0.261, 0.490)	(0.733, 0.689, −0.164)	−1.117	234.494°

where the coefficients $k_i, i = 1, 2, 3, 4$, contain the specified parameters and the direction vectors previously solved. Similarly, we obtain another linear equation by using Eqs. (3, 4, 8), and (12):

$$k_5 q_{1x} + k_6 q_{1y} + k_7 q_{1z} + k_8 = 0 \tag{35}$$

where the coefficients $k_i, i = 5, 6, 7, 8$, contain the specified parameters and the direction vectors previously solved. Now we have three linear equations, Eqs. (7), (34), and (35), allowing us to obtain a unique solution of (q_{1x}, q_{1y}, q_{1z}). Finally, we can use Eq. (32) to obtain the corresponding solution of (g_x, g_y, g_z).

In what follows, we provide a numerical example to verify the solution procedure discussed above. Table 1 gives three displacement screws denoting four positions of a body. Following the discussed solution procedure, we obtain six real solutions, as listed in Table 2. Note that in this case, we have six real solutions, while other specifications may yield only four, two, or zero real solutions.

Table 2 Solution to the synthesis equations

Solution #	$\hat{\mathbf{F}}(f_x, f_y, f_z)$	$\mathbf{G}(g_x, g_y, g_z)$
1	(−0.6917, 0.5429, 0.4762)	(0.2110, 0.1513, 0.1341)
2	(0.3975, 0.5431, 0.7396)	(1.9873, 0.4768, −1.4183)
3	(−0.3927, 0.7106, 0.5838)	(−0.8295, 0.7632, −1.4870)
4	(−0.1186, −0.5320, 0.8384)	(−6.4081, −1.8656, −2.0903)
5	(−0.9773, 0.1780, 0.1149)	(0.2275, 0.9293, 0.4961)
6	(−0.0933, −0.6980, 0.7100)	(7.5762, −1.0020, 0.0105)
	$\hat{\mathbf{M}}_1(m_{1x}, m_{1y}, m_{1z})$	$\mathbf{Q}_1(q_{1x}, q_{1y}, q_{1z})$
1	(−0.1323, 0.5529, −0.8227)	(−1.6848, 1.6638, 1.3892)
2	(−0.8434, −0.1013, 0.5277)	(−0.1331, 0.1786, −0.1784)
3	(−0.1029, −0.7372, 0.6678)	(−1.8024, 0.9896, 0.8147)
4	(0.9900, 0.0011, 0.1407)	(−0.9567, −6.8947, 6.7816)
5	(0.2009, 0.9508, 0.2358)	(−1.3157, 0.1772, 0.4061)
6	(0.9863, −0.1622, −0.0298)	(0.6223, 4.5468, −4.1465)

5 Conclusion

This paper presents the solution of the spatial slider for guiding a rigid body to pass through four positions. By using screw triangle geometry, we obtained 12 equations to solve for 12 unknowns that determine a perpendicularly intersected cylindrical-cylindrical dyad. In order to seek a closed-form solution, we decoupled the equations into two groups and solved the direction vectors using one of the groups. The position vectors of points on the cylindrical joint axes were obtained by solving the second group of design equations.

By using the dialytic elimination method, we obtained a ninth degree polynomial equation that led to nine possible solutions. Among the nine sets of solutions, two of them are infinite, and one is the displacement screw from the first position to the second position. Therefore, we have at most six real solutions that can be used to design spatial sliders for four-position problems. Notice that the solution procedure and result given in this paper are comparable to those in the synthesis of a planar slider.

References

1. Bottema, O., Roth, B.: Theoretical Kinematics. Dover Publications, Inc., New York (1990)
2. Burmester, L.: Gerafführung durch das Kurbelgetriebe. Civilingenieru **22**, 597–606 (1876)
3. Larochelle, P.M.: Circuit and branch rectification of the spatial 4C mechanism. In: Proceedings ASME DETC Mechanism Conference, USA, 10–13 September 2000
4. Murray, A.P., McCarthy, J.M.: Five position synthesis of spatial CC dyad, In: Proceedings ASME DETC Mechanism Conference, Minneapolis, 11–14 Sept 1994
5. Roth, B.: On the screw axes and other special lines associated with spatial displacements of a rigid body. J. Eng. Ind., Trans. ASME B. 102–110 (1967)
6. Tsai, L.W.: Robot Analysis: The Mechanics of Serial and Parallel Manipulator. Wiley, New York (1999)
7. Tsai, L.W., Roth, B.: Design of dyads with helical, cylindrical, spherical, revolute and prismatic joints. Mech. Mach. Theory **7**, 85–102 (1972)
8. Wu, C.H.: Spatial Generalization of the Synthesis of Planar Slider Linkages. M.S. Thesis, Department of Mechanical Engineering, National Cheng Kung University, Taiwan (2011)

A Cartesian Cable-Suspended Robot for Aiding Mobility

Gianni Castelli and Erika Ottaviano

Abstract In this chapter we propose the analysis and simulation of a cable system developed to be used in large-scale handling for applications in urban, civil and naval environments. For the proposed system, which belongs to the Cartesian Cable-Suspended Robots (CCSR), the main issue is that it can provide translational motion of the end-effector being suspended, thus it may be considered well suited for a number of applications including the proposed one. In this chapter we focus our attention on a spatial version of the cable system designed to improve the mobility of end-users in urban environment. Kinetostatics and dynamic simulation are proposed and discussed.

Keywords Cable-suspended robot · Modelling · Simulation.

1 Introduction

Cable driven robots have been developed and tested for several years, but their practical implementation in industrial and naval environments and in civil work sites is still an open issue. Many fully-constrained manipulators were proposed for a number of possible applications, but feasible tasks are often limited due to the increasing number of cables, [1]. Probably, the most interesting solution for the above-mentioned areas of application regards cable-suspended robots, i.e. in a crane-like configuration, but there are still several open issues related to the design and control of these cable robots that limit their practical use. Under-constrained robots rely on gravity force, indeed in a crane-like configuration the moving platform is suspended and operated by cables that are connected to the base. Some reported research on cable-suspended robots are: the 3D cable robot ROBOCRANE [2]; the Sky-Cam [3], a Cargo Transfer

G. Castelli · E. Ottaviano (✉)
Department of Civil and Mechanical Engineering, University of Cassino
and Southern Lazio, Cassino, Italy
e-mail: ottaviano@unicas.it

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_41,

System [4]. Other aspects on design and control can be found in [5]. Applications of cable suspended robots were proposed in [6] for rescue operations, and [7] for industrial applications. In this chapter a cable-based system named as CaSIMo (Cable System for Improving Mobility) is proposed to improve the mobility of end-users in urban environment. The design and implementation of safe and reliable devices that are able to help end-users to provide a better quality of life is becoming of great inter-est in the scientific community. Lot of efforts have been devoted to the development of the concept of Home & Building Automation, but little has been done for the city concept. In fact, there are still several architectural barriers such as stairs, rivers, roads, canals, not only in old European cities but even in modern towns that limit the mobility of people with motor impairment. Thus, in this context we have considered the development of technical solutions able to overcome such architectural barriers, when classical solutions such as bridges cannot be adopted. The manipulator can provide translational motion being suspended [8], thus it can be proposed for a number of applications such as industrial pick and place operations. But, if large-scale applications are involved, a correct modelling of the system is necessary to obtain accurate positioning of the end-effector, as shown in [9, 10]. In addition, cable characteristics should be taken into account since the robot has to lift and transport people in a safe and robust way. The developed models and simulations take into account issues such as cable mass, elasticity and the effect of uncertainties in cable connections, according to previous works proposed by the authors in [9, 10].

2 A Cartesian Cable Suspended Robot (CCSR)

The robot understudy belongs to the class of Cartesian Cable- Suspended Robot (CCSR) [8]. Cable-robots that belong to this class are the C4 [11], the crane in [12] and DeltaBot in [13]. The proposed Cartesian Cable Suspended Robot has eight cables arranged in-parallel by pairs, each pair having the same length. The proposed CCSR has 3 translational DOFs by keeping a constant orientation. Therefore, the position workspace will be considered in this context. In particular, the geometry and force closure allow the robot to maintain a constant orientation. Four pairs of parallel cables are attached to the end-effector and collected by eight spools mounted on the upper base after passing through guide holes on the spools' frames. According to the scheme of Fig. 1, for each pair of cables, a close-chain can be identified, as for example $A_{11}A_{21}B_{11}B_{21}$. Since segments $A_{11}A_{21} = B_{11}B_{21} =$ h, and cables $l_{11} = l_{21}$, hence each pair of segments of this close-chain will be parallel too, if the cables are all in tension. Therefore, if it is true for the four pairs of cables, rotations about X and Y are not allowed. Furthermore, if the top of the frame $A_{11}A_{12}A_{13}A_{14}$ and top of end-effector $B_{11}B_{12}B_{13}B_{14}$ are similar, that is $b_x/b_y = L_xL_y$ in Fig. 1, the end-effector cannot perform rotation about Z axis. A proof for the translational-motion is reported in [8, 14] for the CCSR structure.

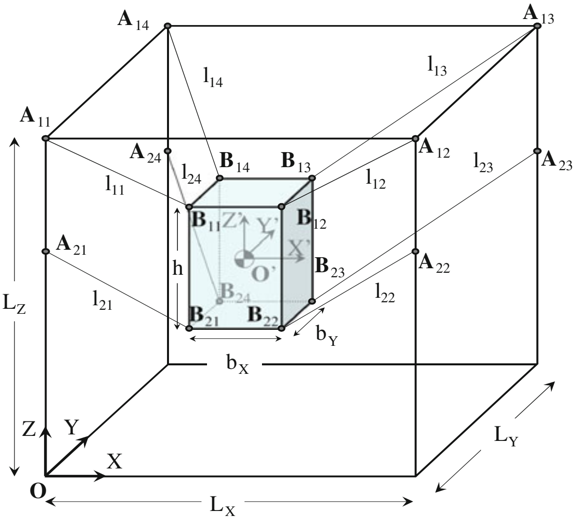


Fig. 1 Scheme for a kinetostatic analysis of a cartesian cable-suspended robot [15]

The general Inverse Kinematics equation can be written as

$$l_{ij} = \|\mathbf{X} + \mathbf{R}\mathbf{b}_{ij} - \mathbf{a}_{ij}\| \tag{1}$$

in which l_{ij} is the length of each cable, $\mathbf{X}$ is the position vector of the reference point O' on the end-effector, $\mathbf{R}$ is the rotation (identity matrix), $\mathbf{B}_{ij}$ (for $i, j = 1, \ldots, n$) is the position vector of the *bij* point attached to the moving platform expressed in the moving reference frame, and $\mathbf{a}_{ij}$ is the position of the exit point of each cable on the fixed frame. For force equilibrium it holds

$$\sum_{i=1}^{4}\sum_{j=1}^{2}\mathbf{F}_{ij} = -\sum_{i=1}^{4}\sum_{j=1}^{2}F_{ij}\widehat{\mathbf{l}}_{ij} = \mathbf{P};\ \sum_{i=1}^{4}\sum_{j=1}^{2}\mathbf{t}_k = -\sum_{i=1}^{4}\sum_{j=1}^{2}F_{ij}\widehat{\mathbf{l}}_{ij} \times \mathbf{R}\mathbf{b}_{ij} = \mathbf{M} \tag{2}$$

$\mathbf{F}_{ij}$ in Eq. 2 is the cable tension that is applied to cable. Moreover, $\mathbf{P}$ and $\mathbf{M}$ are the resultant vector force and torque (wrench $\mathbf{W}$) that are exerted on or by the environment. Substituting the above-mentioned terms into 2 yields to

$$\mathbf{J}^{\mathrm{t}}\mathbf{F} = \mathbf{W} \tag{3}$$

in which $\mathbf{F}$ represents the vector of cable forces, $\mathbf{J}$ is the Jacobian matrix. It has been assumed that there are no external wrenches other than gravity. Equation 3 can be used to evaluate the vector $\mathbf{F}$ for a given trajectory. By using the Moore-Penrose Matrix Inverse we obtain a solution that is the minimum Euclidean norm corresponding to the lowest energetic value for the set of scalar cable forces. In most of reported works, it is assumed that the connection point to the mobile platform coincides with the centre of the pin. In particular in the following this assumption is considered correct for the mobile attachment points, but for the base attachments suitable pulleys are modelled, according to [10, 16]. Cables are represented as lumped mass systems. Elasticity is included by considering that a mass less cable behaves as linear spring and its elasticity coefficient can be evaluated as function of the cable section area A, the Young Modulus E, and the cable length l. The complete model is then obtained by discretizing the cable into a chain of several mass-spring elements. Therefore, each cable will be composed by N lumped masses m_i, and $N+1$ linear springs connecting the adjacent masses with elasticity coefficient k_i. In [9] it has been assumed that all the m_i masses are equal, as well as the k_i coefficients. The sum of the m_i masses equals the total cable mass, and k_i is equal to $(N + 1)\ k$, in which k is the overall cable elasticity. It is important to point out that, given the position and trajectory of the end-effector, cable lengths and forces can vary considerably according to the developed model. Therefore, if accurate simulation is required those aspects must be considered.

3 CaSIMo: Cable System for Improving Mobility

CaSIMo is a robotic system that can be used to improve end-users mobility in urban environment allowing the person on a wheelchair (and eventually companions) to overcome rivers, roads and any other large obstacle in cities with safety and a remarkable simplicity of use. Figure 2 shows an urban installation of the CaSIMo to overpass a canal. The proposed system is based on an innovative design of the CCSR manipulator described in Sect. 2. It is composed by a cage to accommodate passengers and suitable actuation to drive steel cables. The cage is equipped with a push-button panel like lifts. The cage is also equipped with an automated system for opening and closing the doors. The particular arrangement of cables' connection allows to reduce the oscillations along the movement directions and prevents the fall of the cage even if an accidental cable breaking occurs. Current rules used for the design of lifts have been considered for the general definition of the system. The geometry of the CCSR provides two characteristics to the CaSIMo system: 1) safety: if a cable breaking occurs the redundant number of cables prevents accidental falls of the cage; 2) stability: the geometry helps to prevent oscillations in the direction of the movement.The CaSIMo can be built by using commercial type components and therefore the total cost of the device is similar than other cable systems.

Furthermore, thanks to the absence of evident large sized supporting structure, it greatly reduces the environmental and architectural impact. In Fig. 2 main dimension for designing the CaSIMo system are: dimension of the canal to overpass 12.5 m; cage dimension 2.8 m × 1.5 m× 1.5 m, it can hold 3 or 4 people with total mass of 1,000 kg. Steel cables have been chosen with a diameter of 10 mm, breaking load of 83 kN and linear density of 0.49 kg/m.

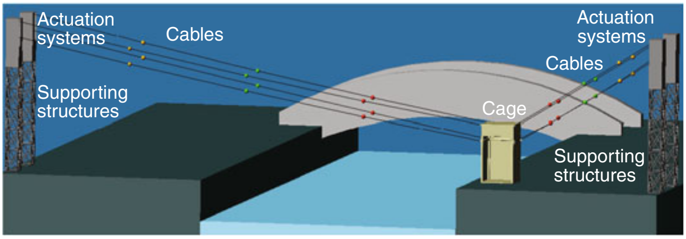


Fig. 2 An example of the CaSIMo in an urban installation to overpass a canal

Table 1 Position of the centers' of rotation for pulleys in the model of Fig. 2

	A_{11}	A_{21}	A_{12}	A_{22}	A_{13}	A_{23}	A_{14}	A_{24}
X(m)	0	0	32	32	32	32	0	0
Y(m)	0	0	0	0	3.5	3.5	3.5	3.5
Z(m)	8	7.2	8	7.2	8	7.2	8	7.2

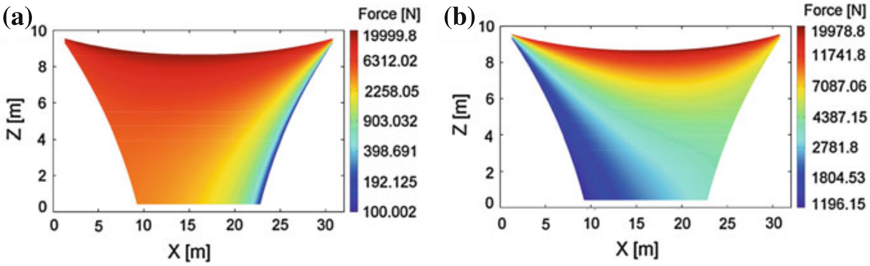


Fig. 3 Cable tension distribution in a workspace cross-section: **a** F_{11}; **b** F_{21}

According to the chosen cables, pulleys with a diameter of 500 mm have been selected. Pulleys centers' of rotations are given in Table 1. We have developed a model for the cable release point (referred as the anchor position or anchor point) that varies as the position and orientation (pose) of the mobile platform is changed. Therefore, for the developed simulations the anchor positions have been calculated for each pose of the cage according to the pulleys' diameter. According to these design data a workspace analysis has been performed. Figure 3 shows cable tension distribution for cables 11 and 21 respectively (see the scheme in Fig. 1) in a cross-section area taken in the plane of motion of the cage Center of Mass (CM).

The cross-section area is $9\,\text{m} \times 26.1\,\text{m} = 230\,\text{m}^2$, the safe area of the CaSIMo that we can obtain maintaining a constant orientation of the cage is $195\,\text{m}^2$, which represents the 84 % of the working area.

A suitable safety factor is chosen being 15 kN the maximum allowed value for the cable tension. If we consider this force limit the cage can be lifted of about 3 m, which is a value more than satisfactory for the application. In particular, cable tension determination takes into account this maximum force value.

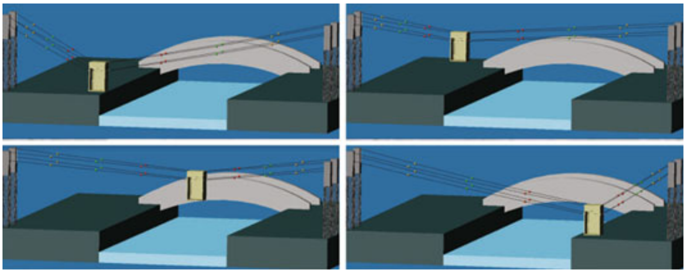


Fig. 4 A sequence of the cage transfer phases with CaSIMo

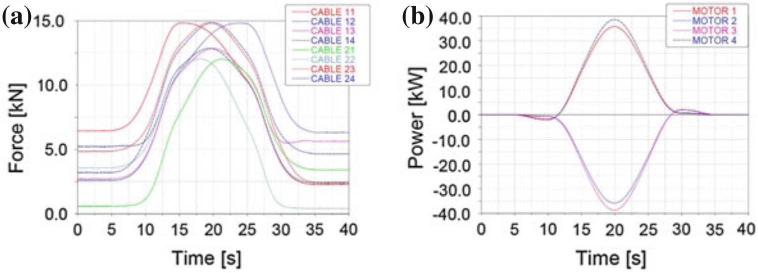


Fig. 5 Simulation numerical results: **a** cables' tensions **b** required power

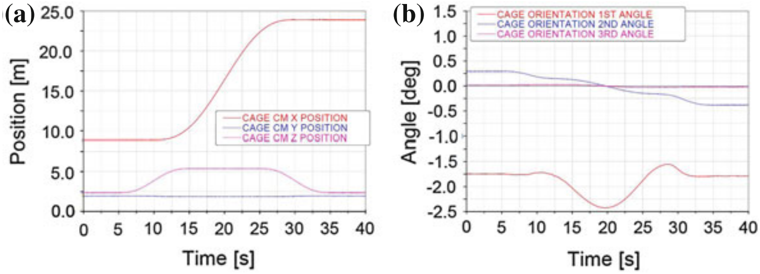


Fig. 6 Simulation numerical results: **a** cage CM position; **b** cage angular orientation

4 Simulation Results

A simulation of the cage transfer is reported in the sequence of Fig. 4. The proposed simulation takes 40 s: 5 s at the beginning and the end are used to reach the static equilibrium of the system. During the first phase the cage is lifted up by 3 m. During the 2nd phase the cage is moved along X direction by 15 m, finally, during the 3rd phase the cages is lifted down. For simulation purposes, only people are not displayed during the simulation given in Fig. 4, but their effect is considered as an additional mass in the cage.

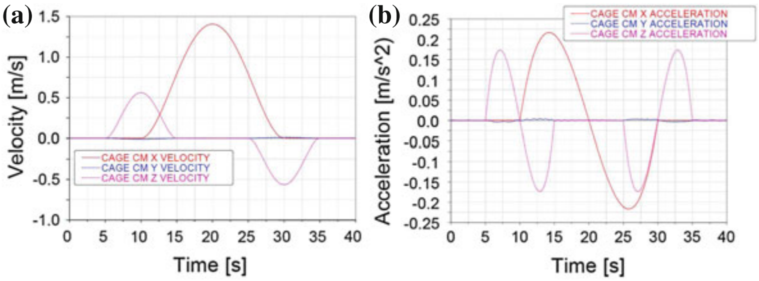


Fig. 7 Simulation numerical results: **a** cage velocity; **b** Cage acceleration

Results are shown in Figs. 5, 6, 7 for the dynamic simulation run under ADAMS environment. In particular, the cables' tensions and required power, velocity and acceleration of the Center of Mass (CM) for the cage are shown. It is worth noting that, although the suspended nature of the cable system, the simulation shows negligible rotations of the cage (the 3rd angle only has a maximum rotation of 2°). Since the system has to transport people, accelerations effects of the cage can be almost neglected, although the simulation time can be considered short according to the task.

5 Conclusion

In this chapter a large dimension cable system has been proposed to be used for mobility and handling applications with high level of safety and stability. In addition, since the cable based manipulator is proposed here to transfer people, cable mass and elasticity, and transmission system are considered. The proposed cable robot belongs to the class of CCSR, its particular design allows translational motion only and then it is well suited for the application. Nevertheless, the proposed model makes this design well suited for large-scale handling applications, such as in naval and civil environments.

References

1. Kawamura, S., Kino, H., Won, C.: High-speed manipulation by using parallel wire-driven robots. Robotica **18**(1), 13–21 (2000)
2. Albus, J., Bostelman, R., Dagalakis, N.: The NIST robocrane. J. Rob. Syst. **10**(5), 709–724 (1992)
3. August Design, http://www.august-design.com, (2013)
4. Tadokoro, S., Verhoeven, R., Hiller, M., Takamori, T.: A portable parallel manipulator for search and rescue at large-scale urban earthquakes and an identification algorithm for the installation in unstructured environments. In: Proceedings of the IEEE/RSJ IROS International Conference

on Intelligent Robots and Systems, pp. 1222–1227. Kyunghee Printing Co, Kyongju, Korea (1999)

5. Yamamoto, M., Noritaka, Y., Akira, M.: Trajectory control of incompletely restrained parallel-wire-suspended mechanism based on inverse dynamics. IEEE Trans. Robot. **20**(5), 840–850 (2004)
6. Merlet, J.-P.: Kinematics of the wire-driven parallel robot MARIONET using linear actuators. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3857–3862. Pasadena, CA (2008)
7. Izard, J.-B., Gouttefarde, M., Michelin, M., Tempier, O., Baradat, C.: Reconfigurable robot for Cable-driven parallel robotic and industrial scenario proofing. In: 1st International Conference on Cable-Driven Parallel Robots, pp. 135–148. Springer, Stuttgart (2012)
8. Castelli, G., Ottaviano, E., Gonzàlez, A.: Analysis and simulation of a new cartesian cable-suspended robot. In: Proceedings of the Institution of Mechanical Engineers Part C: Journal of Mechanical Engineering Science, vol. 224, pp. 1717–1726. (2010) doi:10.1243/09544062JMES1976
9. Ottaviano, E., Castelli, G.: A study on the effects of cable mass and elasticity in cable-based parallel manipulators. In: Proceedings of the 18th CISM-IFToMM Symposium on Robot Design, Dynamics and Control, pp. 149–156. Springer, Udine (2010)
10. Ottaviano E., Castelli G.: Issues on the modelling of cable-based parallel manipulators, Congresso dell'Associazione Italiana di Meccanica Teorica e Applicata (AIMETA), paper MEM-13-0. Bologna Publi and Stampa Edizioni, Conselice, Italy (2011)
11. Bosscher, P., Williams II, R.L.: Cable-suspended robotic contour crafting system. Autom. Constr. **17**(1), 45–55 (2007)
12. Krut, S., Ramdani, N., Gouttefarde, M., Company, O., Pierrot, F.: A parallel cable-driven crane for SCARA-motions. In: Proceedings of the ASME 2008 IDETC/CIE 2008, Paper DETC2008–49969. ASME, Brooklyn, New York, USA (2008)
13. Dekker, R., Khajepour, A., Behzadipour, S.: Design and testing of an ultra-high-speed cable robot. Int. J. Robot. Autom. **21**(1), 25–34 (2006)
14. Voglewede, P.A., Ebert-Uphoff, I.: Application of the antipodal grasp theorem to cable driven robots. IEEE Trans. Robot. **21**(4), 713–718 (2005)
15. Ottaviano, E., Castelli, G., Rea, P.: Macchina automatica cartesiana CCSR (Cartesian cable-suspended robot) basata su sistema a cavi per la movimentazione di carichi, Italian Patent n. FR2009A000025, 02/11/2009
16. Notash, L., McColl, D.: Workspace investigation of wire-actuated parallel manipulators with uncertainties in wire connections. ASME 2010, Annual Mechanisms and Robotics Conference, Paper DETC2010–28228, Montreal, Canada (2010)

Evaluating the Spatial Compliance of Circularly Curved-Beam Flexures

Farid Parvari Rad, Giovanni Berselli, Rocco Vertechy and Vincenzo Parenti Castelli

Abstract In this chapter, the closed-form compliance equations for Circularly Curved-Beam Flexures are derived. Following a general modeling procedure previously described in the literature, each element of the spatial compliance matrix is analytically computed as a function of both hinge dimensions and employed material. The theoretical model is then validated by comparing analytical data with the results obtained through Finite Element Analysis. Finally, a case study is presented concerning the potential application of these types of flexures in the optimal design of compliant robotic fingers.

Keywords Circularly curved-beam flexures · Compliance matrix · Robotic fingers · Finite element analysis

1 Introduction

A flexure hinge is a flexible connector that can provide a limited rotational motion between two parts by means of material deformation. According to [2], these connectors can be used to substitute traditional kinematic pairs (like bearing couplings)

F. P. Rad · V. P. Castelli
Department of Mechanical Engineering, University of Bologna, Bologna, Italy
e-mail: farid.parvarirad2@unibo.it

V. P. Castelli
e-mail: vincenzo.parenti@unibo.it

G. Berselli
Department of Mechanical Engineering, University of Modena and Reggio Emilia, Reggio Emilia, Italy
e-mail: giovanni.berselli@unimore.it

R. Vertechy (✉)
Percro Laboratory, Scuola Superiore Sant'Anna, Pisa, Italy
e-mail: r.vertechy@sssup.it

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_42,

in rigid-body mechanisms, thus obtaining the so-called Lumped Compliant Mechanisms (LCMs), in which compliance is concentrated in relatively small regions connected through rigid links.When compared to their rigid-body counterpart, LCMs are characterized by reduced weight, absence of backlash and friction, part-count reduction, but restricted range of motion.

From a design perspective, the introduction of flexure hinges in serial articulated chains, like anthropomorphic hands and prosthesis, seems promising as it can allow the generation of very slender and light mechanisms that better reproduce biological structures. For instance, Figs. 1 and 2 depict two compliant robotic fingers, previously proposed by Lotti and Vassura [8], that employ either Straight-Beam Flexures (SBF) or Circularly Curved-Beam Flexures (CCBF) as possible substitutes for traditional revolute joints (the corresponding hinge rotation being defined as principal rotation [3]). In this case, regardless of the flexure topology, the use of flexible joints allows one-piece manufacturing and enhanced performance in terms of robustness and safety when interacting with unknown environments or humans (e.g., [4]). Despite the aforementioned advantages, LCMs also introduce new engineering challenges mainly due to possible fatigue failures and undesired spatial motions, which may occur under the action of out-of-plane forces also in LMCs initially conceived as planar mechanisms.

In this scenario, relatively simple models, such as the well-known pseudo-rigid-body model described in [5], can turn very useful for model-based control of robotic systems [1], and for designing LCMs with prescribed load-displacement profiles at one point on their structure [9]. In parallel, the knowledge of the hinge compliance behavior in the 3D space, even in the small displacement range where a 6x6 spatial compliance matrix can be defined, may become extremely valuable for both first attempt sizing the hinge dimensions and for comparison purposes. For instance, a method for comparing the selective compliance of elastic joints with generic morphology has been proposed in [3]. Empirical equations based on Finite Element Analysis (FEA) for various hinge profiles have been reported in e.g., [12], whereas the stiffness matrices concerning several hinge geometries (e.g., circular and elliptical) can be found in [7]. Furthermore, several studies concerned the stiffness analysis of curved beams by means of the Castigliano second theorem [10] or the so-called direct methods [6].

Following a similar approach, the contributions of this chapter are: (a) to report the closed-form compliance equations for CCBF (i.e., a particular case of generic curved beams); (b) to compare CCBF and SBF in terms of selective compliance and maximum achievable principal rotation. In particular, the CCBF and SBF employed for the fabrication of the robotic fingers depicted in Figs. 1 and 2 are considered as a case study, whereas the theoretical CCBF model is derived following the general procedure outlined in [6] and subsequently validated via FEA.

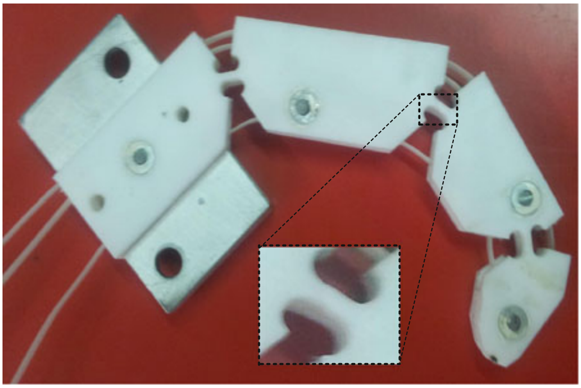


Fig. 1 Mono-piece robotic finger employing straight-beam flexures [8]

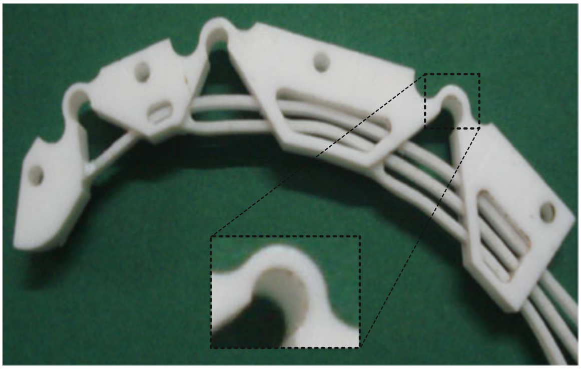


Fig. 2 Mono-piece robotic finger employing circularly curved-beam flexures [8]

2 Closed-Form Compliance Equations

As previously said, the direct analytical method proposed in [6] is used for defining the CCBF flexural behavior. With reference to Fig. 3, let us consider a cantilever curved beam with a uniform cross section and generically loaded on the free end. Node 1 and node 2 are two points located on the beam fixed and free end respectively. The external load, **P**, and the corresponding deformation, **Q**, might be expressed in a predefined global coordinate system via the following column vectors:

$$\mathbf{P} = \left[f_x\ f_y\ f_z\ m_x\ m_y\ m_z\right]^{\mathbf{T}}; \qquad \mathbf{Q} = \left[u\ v\ w\ \alpha\ \varphi\ \psi\right]^{\mathbf{T}} \tag{1}$$

where u, v, w and α, φ, ψ are, respectively, the three displacements of node 2 and the three rotations of the corresponding beam cross section along the x, y and z directions. With reference to Fig. 4, a local coordinate system centered on the centroid

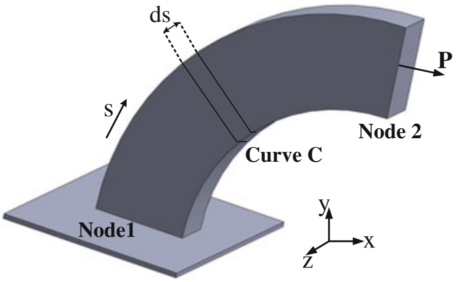


Fig. 3 Cantilever curved beam loaded at the free end

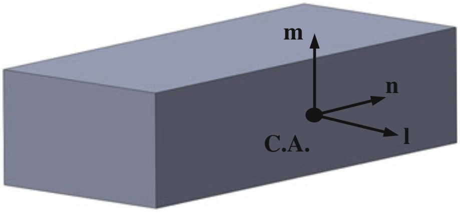


Fig. 4 Cross section of the beam and the local coordinates

of a generic beam cross section can be defined. In particular, these local coordinates are denoted as **l**, **m** and **n**, namely the tangent vector and the principal vectors of the cross section [6]. The relation between local and global coordinates can be written as follows:

$$\begin{bmatrix} \mathbf{l} \\ \mathbf{m} \\ \mathbf{n} \end{bmatrix} = \begin{bmatrix} l_x(s) & l_y(s) & l_z(s) \\ m_x(s) & m_y(s) & m_z(s) \\ n_x(s) & n_y(s) & n_z(s) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{bmatrix} = \mathbf{R}(s) \cdot \begin{bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{bmatrix} \tag{2}$$

where s refers to the coordinate variable along the curve and $\mathbf{R}(s)$ is the rotation matrix that relates global and local coordinate frames. The curve defining the centroid of the beam cross sections, curve C, in the global coordinates can be expressed by:

$$\mathbf{r}(\mathbf{s}) = x(s)\mathbf{i} + y(s)\mathbf{j} + z(s)\mathbf{k} \tag{3}$$

The load **P** acting on the free end is balanced by a load **P**′ acting on the element ds of the curve C. This load **P**′ produces a deformation, **E**, on the same element. The matrices **P**′ and **E**, together with the corresponding analytical relation can be expresses as:

$$\mathbf{P}' = \left[f_l \ f_m \ f_n \ m_l \ m_m \ m_n\right]^T; \quad \mathbf{E} = \left[\varepsilon_{ll} \ \gamma_{lm} \ \gamma_{ln} \ \kappa_{ll} \ \kappa_{lm} \ \kappa_{ln}\right]^T; \quad \mathbf{P}' = \mathbf{K} \cdot \mathbf{E} \tag{4}$$

The matrix **K** in Eq. (4) is the stiffness matrix of the element ds that can be written as:

$$\mathbf{K} = \begin{bmatrix} EA & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta_m GA & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta_n GA & 0 & 0 & 0 \\ 0 & 0 & 0 & GJ & 0 & 0 \\ 0 & 0 & 0 & 0 & EI_m & 0 \\ 0 & 0 & 0 & 0 & 0 & EI_n \end{bmatrix} \tag{5}$$

where A, β_m, β_n, I_m, I_n, J, E and G are, respectively, cross section area, shear coefficients, principal moments of inertia and polar moment of inertia of the beam's cross section, Young's modulus and shear modulus of the employed material. The deformation, $d\mathbf{Q}'$, of the element ds, due to the load $\mathbf{P}'$, is defined by:

$$d\mathbf{Q}' = \left[du' \ dv' \ dw' \ d\alpha' \ d\varphi' \ d\psi'\right]^T = \mathbf{E} \cdot ds \tag{6}$$

where u', v', w' and α', φ', ψ' are respectively displacements and rotations of the element ds in the $\mathbf{l}$, $\mathbf{m}$ and $\mathbf{n}$ directions. The load $\mathbf{P}'$, acting on ds and due to the presence of a load $\mathbf{P}$ on the free end, can be computed via the adjoint transformation matrix $\mathbf{T} \in \mathbb{R}^{6\times6}$ between the global and local coordinates. The following relation holds:

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P} \tag{7}$$

The adjoint matrix $\mathbf{T}$ is a function of s and can be computed from Eqs. (2) and (3), as:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \\ (\widetilde{\mathbf{r}}_s \cdot \mathbf{R})^T & \mathbf{R}^T \end{bmatrix} \tag{8}$$

where $\mathbf{r_s} = \mathbf{r} - \mathbf{r_2}$ is the position vector connecting node 2 to the centroid of the section, $\mathbf{0} \in \mathbb{R}^{3\times3}$ is a null matrix, and $\widetilde{\mathbf{r}}_s$ denotes the cross product matrix of $\mathbf{r}_s$, i.e., the matrix such that $\widetilde{\mathbf{r}}\mathbf{u} = \mathbf{r} \times \mathbf{u}$ for any vector $\mathbf{u}$. In addition, the deformation of the element ds, $d\mathbf{Q}'$, causes a deformation at the free end, $d\mathbf{Q}$, that can be calculated using the following equation:

$$d\mathbf{Q} = \mathbf{T}^T \cdot d\mathbf{Q}' \tag{9}$$

By merging Eqs. (4, 6, 7) and (9), one obtains:

$$d\mathbf{Q} = \mathbf{T}^T \cdot \mathbf{K}^{-1} \cdot \mathbf{T} \cdot \mathbf{P} \cdot ds \tag{10}$$

By integrating Eq. (10), one can find the relation between the load, $\mathbf{P}$, and the deformation, $\mathbf{Q}$, of the free node, as follows:

$$\mathbf{Q} = \mathbf{C} \cdot \mathbf{P} \tag{11}$$

where:

$$\mathbf{C} = \int_C \mathbf{T}^T \cdot \mathbf{K}^{-1} \cdot \mathbf{T} \cdot ds \tag{12}$$

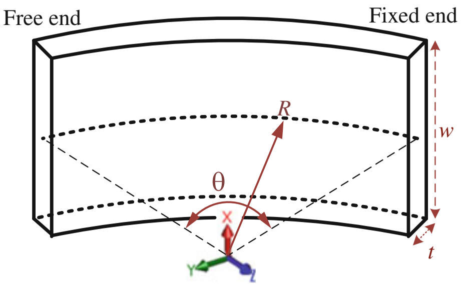


Fig. 5 Cross section properties and geometric parameters of the hinge

Matrix **C** is the compliance matrix for a general cantilever curved beam and it gives the relation between the loads at the free end and the corresponding deformations.

This method is applied to a CCBF as the one depicted in Fig. 5, in order to estimate its compliant behavior under a generalized loading condition. Henceforth, the CCBF compliance matrix is derived in its analytical form and explicitly presented hereafter:

$$\mathbf{C} = \begin{bmatrix} C_{x,f_x} & 0 & 0 & 0 & C_{x,m_y} & C_{x,m_z} \\ 0 & C_{y,f_y} & C_{y,f_z} & C_{y,m_x} & 0 & 0 \\ 0 & C_{z,f_y} & C_{z,f_z} & C_{z,m_x} & 0 & 0 \\ 0 & C_{\theta_x,f_y} & C_{\theta_x,f_z} & C_{\theta_x,m_x} & 0 & 0 \\ C_{\theta_y,f_x} & 0 & 0 & 0 & C_{\theta_y,m_y} & C_{\theta_y,m_z} \\ C_{\theta_z,f_x} & 0 & 0 & 0 & C_{\theta_z,m_y} & C_{\theta_z,m_z} \end{bmatrix} \tag{13}$$

where:

$$C_{x,f_x} = R\left(\frac{\theta}{\beta_n GA} + \frac{R^2(3/2\,\theta - 2\,\sin(\theta) + 1/2\,\cos(\theta)\sin(\theta))}{GJ} + \frac{R^2(-1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta)}{EI_m}\right)$$

$$C_{x,m_y} = C_{\theta_y,f_x} = R\left(\frac{R(\sin(\theta) - 1/2\,\cos(\theta)\sin(\theta) - 1/2\,\theta)}{GJ} - \frac{R(-1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta)}{EI_m}\right)$$

$$C_{x,m_z} = C_{\theta_z,f_x} = R\left(-\frac{R\left(1/2\,(\cos(\theta))^2 - \cos(\theta)\right)}{GJ} + 1/2\,\frac{R(\cos(\theta))^2}{EI_m}\right)$$

$$C_{y,f_y} = R\left(\frac{1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta}{EA} + \frac{-1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta}{\beta_m GA} + \frac{R^2(3/2\,\theta - 2\,\sin(\theta) + 1/2\,\cos(\theta)\sin(\theta))}{EI_n}\right)$$

$$C_{y,f_z} = C_{z,f_y} = R\left(1/2\,\frac{(\cos(\theta))^2}{EA} - 1/2\,\frac{(\cos(\theta))^2}{\beta_m GA} + \frac{R^2\left(1/2\,(\cos(\theta))^2 - \cos(\theta)\right)}{EI_n}\right)$$

$$C_{y,m_x} = C_{\theta_x,f_y} = \frac{R^2(\theta - \sin(\theta))}{EI_n}$$

$$C_{z,f_z} = R\left(\frac{-1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta}{EA} + \frac{1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta}{\beta_m GA} + \frac{R^2(-1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta)}{EI_n}\right)$$

$$C_{z,m_x} = C_{\theta_x,f_z} = -\frac{R^2\cos(\theta)}{Ei_n}$$

$$C_{\theta_x,m_x} = \frac{R\theta}{EI_n}$$

$$C_{\theta_y,m_y} = R\left(\frac{1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta}{GJ} + \frac{-1/2\,\cos(\theta)\sin(\theta) + 1/2\,\theta}{EI_m}\right)$$

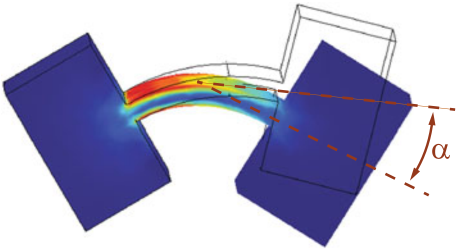


Fig. 6 FEA of the circularly curved-beam flexure hinge

$$C_{\theta_y,m_z} = C_{\theta_z,m_y} = R\left(1/2\,\frac{(\cos(\theta))^2}{GJ} - 1/2\,\frac{(\cos(\theta))^2}{EI_m}\right)$$
$$C_{\theta_z,m_z} = R\left(\frac{-1/2\,\cos(\theta)\sin(\theta)+1/2\,\theta}{GJ} + \frac{1/2\,\cos(\theta)\sin(\theta)+1/2\,\theta}{EI_m}\right)$$
$$A = wt,\ I_m = 1/12\,tw^3,\ I_n = 1/12\,wt^3,\ J = I_m + I_n = 1/12\,wt\left(t^2+w^2\right)$$

In particular, with reference to Fig. 5, R, θ, w, t represent the radius of the hinge centroid with respect to the global coordinates, the angle of the centroid from the free to the fixed end, the hinge width and thickness respectively.

3 Numerical Example and Model Validation

As a case study, the compliant behavior of the CCBF and of the SBF depicted in Figs. 1 and 2 are numerically evaluated. As for the CCBF, the following geometric parameters are considered, namely $R = 30\,\text{mm}$, $t = 1.2\,\text{mm}$, $w = 6\,\text{mm}$ and $\theta = \pi/4$. The flexure hinge connects two rigid links located at a distance $l = 2R\sin(\theta/2)$ and is made of Acrylic Plastic with Young's modulus $E = 3{,}000\,\text{Mpa}$, Poisson's ratio $\nu = 0.33$, shear modulus $G = 1{,}130\,\text{Mpa}$ and the shear deformations being neglected. The principal hinge compliance [3] for the considered application is $C_{\theta_x,m_x} = 12R\theta\big/Ewt^3 = 9\,\text{rad/Nm}$. The method described in Sect. 2 is used for computing the overall CCBF compliance matrix, whereas Finite Element Analysis (FEA) is performed in order to validate the theoretical model. Figure 6 depicts, as an example, the CCBF undeformed and deformed shapes when subject to a flexural moment applied on the hinge free end. Similar FEA simulations are carried out by individually loading the CCBF at the free end for each component of the load **P** (that is individual forces and moments are applied) and obtaining the corresponding deformations (displacements and rotations). The ratio between each load and deformation component simply represents the compliance factors along different axes. The overall numerical results are shown in Table 1, which also depicts the percentage error between analytical and FEA methods. A maximum percentage error of less than 3 % confirms the validity of the proposed modeling technique.

The same procedure is then applied to compute the SBF compliance matrix whose analytical solution is known from the literature [11]. As said, the SBF is designed so

Table 1 Compliance factors for the CCBF flexure hinge and comparison between analytical and FEA results

Compliance factors	C_{x,f_x}	$C_{x,m_y} = C_{\theta_y,f_x}$	$C_{x,m_z} = C_{\theta_z,f_x}$	C_{y,f_y}	$C_{y,f_z} = C_{z,f_y}$	C_{θ_x,m_x}
Analytic	8.001e-5	3.016e-4	−5e-3	1.466e-4	4.483e-4	9.0903
FEA	7.746e-5	3.015e-4	−4.99e-3	1.457e-4	4.457e-4	9.0897
Percentage error	3.3	4.7e-2	6e-2	5.9e-1	5.8e-1	4.9e-4
Compliance factors	C_{z,f_z}	$C_{z,m_x} = C_{\theta_x,f_z}$	$C_{y,m_x} = C_{\theta_x,f_y}$	C_{θ_y,m_y}	$C_{\theta_y,m_z} = C_{\theta_z,m_y}$	C_{θ_z,m_z}
Analytic	1.5e-3	1.017e-1	2.72e-2	8.256e-1	−1.797e-1	4.662e-1
FEA	1.486e-3	1.017e-1	2.72e-2	8.271e-1	−1.803e-1	4.664e-1
Percentage error	3.4e-1	0	0	1.8e-1	3.2e-1	4.5e-2

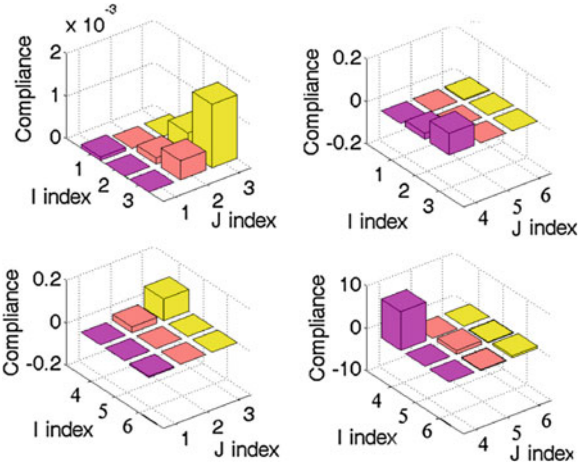


Fig. 7 3D bar representation for the compliance matrix of the CCBF

as to connect the same rigid links of the previous example and to provide the same principal compliance as the CCBF previously modeled. Henceforth, the SBF length is $l = 2R\sin(\theta/2)$, the SBF principal compliance is $\overline{C}_{\theta_x,m_x} = 12l\big/Ew\overline{t}^3 = 9$ rad/Nm, whereas the SBF thickness, $\overline{t}$, is chosen accordingly as $\overline{t} = t\big(2\sin(\theta/2)\big/\theta\big)^{1/3}$.

The numerical values of the compliance matrix entries are depicted in Figs. 7 and 8 respectively. Similarly to [3], this 3D bar graph representation allows a qualitative comparison of the hinge behavior in terms of selective compliance. It can be noticed that, in this particular case, the two solutions behave similarly. However, CCBF outperforms SBF in terms of maximum achievable principal rotation. In fact, these maximum rotations might be respectively computed as $\alpha_{CCBF} = \max(\alpha_1, \alpha_2)$ and

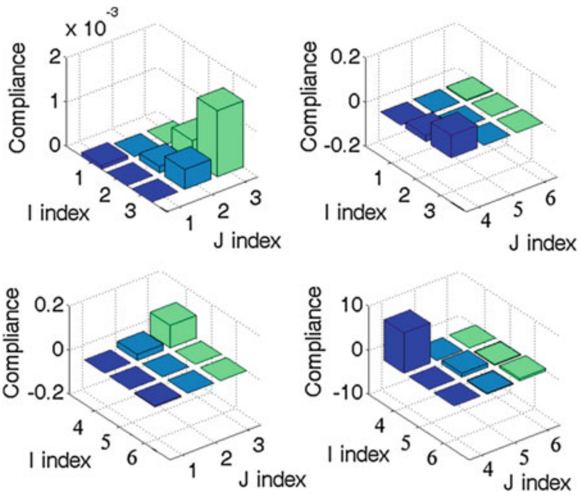


Fig. 8 3D bar representation for the compliance matrix of the SBF

$\alpha_{SBF} = 2lS_Y\big/\bar{t}E$, the term S_Y being the material yield strength [11] and the terms α_1 and α_2 being defined in Eqs. (14) and (15), such that $\alpha_{CCBF}/\alpha_{SBF} > 1$.

$$\alpha_1 = \frac{6R(t+R)S_y\theta\left(-2t+(t+2R)\text{Log}\left[\frac{t+R}{R}\right]\right)}{t^2E\left(-t+(t+R)\text{Log}\left[\frac{t+R}{R}\right]\right)}; \tag{14}$$

$$\alpha_2 = 6E^{-1}RS_y\theta\left(\frac{t+2R}{t^2}+\frac{1}{-t+R\text{Log}\left[\frac{t+R}{R}\right]}\right) \tag{15}$$

4 Conclusions

The closed-form compliance equations for CCBF have been presented and validated via FEA. A comparison has been carried out between CCBF and SBF for possible application in serial articulated chains, like robotic fingers. For this particular case, it is observed that the hinge compliance matrices are very similar, when comparing solutions having the same value of the principal compliance and connecting rigid links located at the same relative distance. Nonetheless, CCBF outperforms SBF in terms of maximum achievable principal rotation. Future work includes a detailed analysis of the CCBF properties as a function of the hinge geometrical parameters and an in-depth investigation of the hinge behavior in the large displacement range.

References

1. Albu-Schäffer, A., Ott, C., Hirzinger, G.: A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. Int. J. Robot. Res. **26**(1), 23–39 (2007)
2. Ananthasuresh, G., Kota, S.: Designing compliant mechanisms. Mech. Eng. **117**, 93–96 (1995)
3. Berselli, G., Vassura, G., Piccinini, M.: Comparative evaluation of the selective compliance in elastic joints for robotic structures, pp. 759–764. IEEE ICRA, International Conference on Robotics and Automation IEEE Shangai, China (2011)
4. Dollar, A., Howe, R.: A robust compliant grasper via shape deposition manufacturing. IEEE/ASME Trans. Mechatron. **11**(2), 154–161 (2006)
5. Howell, L.L.: Compliant Mechanisms. Wiley, New York (2001)
6. Jafari, M., Mahjoob, M.: An exact three-dimensional beam element with nonuniform cross section. ASME J. Appl. Mech. **77**(6), (2010)
7. Lobontiu, N.: Compliant Mechanisms: Design of Flexure Hinges. CRC Press, Boca Raton (2002)
8. Lotti, F., Vassura, G.: A novel approach to mechanical design of articulated fingers for robotic hands. In: IEEE/RSJ IROS International Conference on Intelligent Robots and Systems (2002)
9. Meng, Q., Berselli, G., Vertechy, R., Castelli, V.P.: An improved method for designing flexure-based nonlinear springs, pp. 1–10. ASME IDETC International Design Engineering Technical Conferences, Chicago, USA (2012)
10. Palaninathan, R., Chandrasekharan, P.: Curved beam element stiffness matrix formulation. Comput. & Struct. **21**(4), 663–669 (1985)
11. Timoshenko, S., Goodier, J.: Theory of Elasticity. 3rd edn. McGraw Hill, Higher Education (1970)
12. Yong, Y.K., Lu, T., Handley, D.: Review of circular flexure hinge design equations and derivation of empirical formulations. Precis. Eng. **32**, 63–70 (2008)

Kinematic Analysis of Slider-Cranks Derived from the λ-Mechanism

Erika Ottaviano, Pierluigi Rea and Marco Conte

Abstract In this paper a kinematic analysis is presented for slider-cranks derived from the λ-mechanism. In particular, for this linkage the coupler curves traced by a reference point are Berard curves. By properly choosing the design parameters of the mechanism the coupler curves are represented by quartics, which have been identified and classified.

Keywords Kinematics · Slider-crank · Coupler curve · Singularities

1 Introduction

Planar mechanisms are widely used in industrial environment for automatic machinery in order to give prescribed law of motions. They can be referred as *Function Generators* from the input to the output links, *Rigid Body Guidance* through the study of the rigid coupler motion and *Path Generators* by referring to the coupler curve, as extensively reported in [1–4]. This paper deals with the analysis of slider-cranks derived from the λ-mechanism, which can be used to give suitable coupler curves for application in automatic machinery, providing some constraints, such as the region in which the curve should be contained, or geometrical characteristics of the curve. Given a set of tasks constituting functional requirements, the design process consists in producing a mechanism that will meet all the specifications. Sometimes, for industrial applications, a first requirement deals with the definition of a suitable working area in which the mechanism should produce a given trajectory. This can be due

E. Ottaviano (✉) · P. Rea · M. Conte
Department of Civil and Mechanical Engineering,
University of Cassino and Southern Lazio,
03043 Cassino, Italy
e-mail: ottaviano@unicas.it

P. Rea
e-mail: Rea@unicas.it

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_43,

to physical limits, or actuation constraints. Then a designer should select a shape of the path and other constraints, which can be linked to design parameters of the mechanism and constitute the design problem dealing with precision points, or given trajectory, or rigid body motion. Following this idea we propose design guidelines focusing on a slider-crank derived from the λ-mechanism [5]. In particular, it has been shown that this slider-crank has a practical engineering interest since it can better approximate a straight line than the corresponding four-bar λ-mechanism [5].

2 Position Analysis of the Slider-Crank and Coupler Curves Expression

The first analytical investigation for a coupler curve of a four-bar linkage was undertaken by Prony [6], who analyzed Watt's straight-line motion (1796). Samuel Roberts showed in 1876 that the "three-bar curve" (coupler curve) of the four-bar linkage is an algebraic curve of the sixth order [7]. Cayley gave further properties of the curve. His interest was directed to linkages hypothetically able to generate specific algebraic curves of any order [8]. In general, the more links, the higher is the degree of curve generated. Since a curve can have up to as many intersections with a straight line as the degree of its polynomial expression, it is hypothetically possible to generate (or approximate) any trajectory designing a suitable mechanism. In this contest we focus our attention on slider-cranks that in general have fourth order coupler curves. The equation of the coupler point curve for any slider-crank mechanism may be obtained by analytic geometry being the loci of any point P that belongs to a segment for which a point B is constrained to lie on a circle and another point C is constrained to have a linear trajectory. In the following we restrain the attention on a particular slider-crank for which point P lies on the same line of the coupler link, as shown in the scheme of Fig. 1. It is known as a λ-mechanism [5]. The derivation presented here follows that of Samuel Roberts proposed for a four-bar linkage [8]. The equation can be written in Cartesian coordinates, when the X axis is chosen along the line parallel to the slider, without loss of generality. Let (x, y) (s, e) be, respectively, the coordinates of coupler point P and point C, then

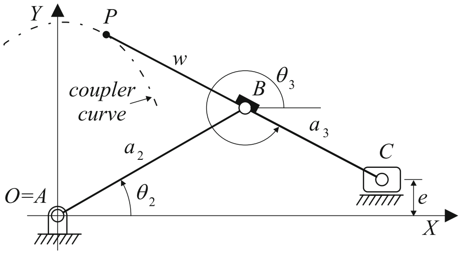
$$\begin{bmatrix} s \\ e \end{bmatrix} = \begin{bmatrix} x + (a_3 + w)cos(\theta_3) \\ y + (a_3 + w)sin(\theta_3) \end{bmatrix} \tag{1}$$

Since B describes a circle (or arc of a circle) centered in O with $\|OB\|^2 = a_2^2$

$$\|OB\|^2 = (x + wcos\theta_3)^2 + (y + wsin\theta_3)^2 \tag{2}$$

Let us take the second equation from 1 and 2. The coupler curve can be obtained by eliminating θ_3 from Eqs. 3 and 4.

Fig. 1 A kinematic scheme of a slider-crank derived from λ-mechanism with design parameters



$$y - e + (w + a_3)sin\theta_3 = 0 \tag{3}$$

$$x^2 + y^2 + w - a_2^2 + (2xw)cos\theta_3 + (2yw)sin\theta_3 = 0 \tag{4}$$

Let us consider the half-tangent substitution in 3 and 4 as

$$cos\frac{\theta_3}{2} = \frac{1-u^2}{1+u^2}, sin\frac{\theta_3}{2} = \frac{2u}{1+u^2}, \tag{5}$$

The coupler curve f of the slider-crank can be described by the algebraic equation of fourth-order in the form

$$\begin{aligned} f = {} & (a_3 - w)^2 y^4 + (a_3 + w)^2 x^4 + 2(a_3^2 + w^2)x^2 y^2 + 4we(a_3 - w)y^3 \\ & + 4we(a_3 - w)x^2 y + [-2(a_3 + w)^2(a_3^2 + w^2) + 4w^2e^2]x^2 \\ & + [-2(a_3^2 - w^2)(a_2^2 - w^2) + 4w^2e^2]y^2 - 4we(a_2^2 - w^2)(a_3 + w) \\ & + (a_3 + w)^2(a_2 - w)^2(a_2 + w)^2 \end{aligned} \tag{6}$$

f is symmetric with respect to the Y axis. This can be proved since Eq. 6 contains only even powers of x.

3 Characterization of the Coupler Curve

In general, a singularity is a point at which an equation, curve, or surface, becomes degenerate. Singularities are often called singular points or geometric singularities [9]. Real geometric singularities of the coupler curve of a slider-crank can be found considering f together with its partial derivatives f_x and f_y with respect to x, and y respectively. The zeros of the set of equations: $f = 0$, $f_x = 0$ and $f_y = 0$ gives the geometric singularities of the coupler curve. They can be identified as

$$x_s = 0;\ \ y_s = \frac{-ew \pm \sqrt{w^4 + (e^2 - a_3^2 - a_2^2)w^2 + a_2^2 a_3^2}}{a_3 - w} \tag{7}$$

Equation 7 can be used to identify geometric singularities of the coupler curve and can be further used to derive kinematic considerations. It is evident from Eq.7 that singularities for this slider-crank may arise on the Y-axis only. Points belonging to these zeros are denoted by C_i, D_i, and A_i, when C_i indicates cusps, D_i double points and A_i acnodes. They can be classified by considering the second partial derivatives of f in the form 9

$$g = f_{xy}^2 - f_{xx} - f_{yy} \tag{8}$$

Functions g, f_x and f_y, can be used to fully characterize real geometric singularities of the coupler curve. They can be related to a classical problem in linkages design known as a "branch defect". In particular, the presence of singularities gives information on the number of branches of the coupler curve, furthermore it allows the determination if two task positions lie on the same branch, as described in [10].

Equations 6–8 lead to a further investigation on the coupler curves characteristics by giving an enumeration of quartic equations representing the trajectory traced by the P reference point. In particular, we focus our attention on quartics studied by Berard in 1820 and then by Ruiz-Castizo in 1889. The coupler curve of a slider-crank derived from a λ-mechanism can be represented by Berard curves [11]. In particular they become "egg shaped" curves if the eccentricity is equal to zero. Furthermore, by properly choosing design parameters, these curves can be identified as well known quartics. In the following we investigate the properties of the coupler curves and their special cases.

–*Quartics of Bernoulli* (1687):

the curve traced by a coupler reference point P of a λ-slider-crank can be represented by Bernoulli quartics if the following conditions are met: $w = a_3$ and $e = 0$. The coupler curves in 6 then become

$$x^4 + x^2y^2 - 2(a_2^2 + a_3^2)x^2 + (a_2 + a_3)^2(a_2 - a_3)^2 = 0 \tag{9}$$

Quartics in 9 can be always represented by two affinely finite branches, as it can be proven they are free of geometric singularities. Examples of coupler curves representing Bernoulli quartics are shown in Fig. 2a. Furthermore if, additionally, $a_3 = a_2$ Eq. 9 degenerates into a circle centered in the (0,0) with radius equal to $2a_2$, as it is shown in 2a.

–*Quartics of Ruiz-Castizo*:

the curve traced by a coupler reference point P of a λ-slider-crank can be represented by Ruiz Castizo quartics if the following conditions are met: $a_3 = a_2 + e$ and $w = a_3$. The coupler curves in 6 can be expressed as 10. Quartics in 10 can be always represented by one connected component, as the existence of geometric singularities can be proven.

$$\begin{aligned} x^4 + x^2y^2 - (4a_2^2 + a_2e + e^2)x^2 + e^2y^2 + (4a_2e^2 + 2e^3)y \\ + e^2(4a_2^2 + a_2e + e^2) = 0 \end{aligned} \tag{10}$$

Examples of coupler curves representing Ruiz-Castizo quartics are shown in Fig. 2b. In particular, geometric singularities are given by

$$x_s = 0; \;\; y_s = -(2a_2 + e) \tag{11}$$

Singular point given in Eq. 11 and Fig. 2b is a cusp, since g in Eq. 8 is equal to zero. In addition, the curve in 10 degenerates if $e = -a_2$. If $e = 0$ Eq. 10 is a circle centered in the origin with radius equal to $2a_2$, as shown in Fig. 2b.
–*Lemniscate*:
the curve traced by a coupler reference point P of a λ-slider-crank can be represented by lemniscate if the following condition is met: $e \geq a_3 - a_2$. It can be always represented by one connected component, as the existence of geometric singularities can be proven. Examples of coupler curves represented by lemniscates are shown in Fig. 3a. Geometric singularities are given by

$$x_s = 0; \;\; y_s = -(2a_2 + w); \;\; y_s = \frac{a_2 w + a_2 a_3 - w^2 - a_3 w}{a_3 - w} \tag{12}$$

If $a_2 = a_3 = e = w$, then the curve in 6 then become

$$x^4 + y^2 w^2 + x^2 y^2 - 3w^2 x^2 = 0 \tag{13}$$

–*Cardano motion*:
the curve traced by a coupler reference point P of a λ-slider-crank can be represented by Cardano motion if the following condition is met: $a_2 = a_3$ and $e = 0$.

$$[x^2 + y^2 - (a_2 + w)^2][(a_2 + w)^2 x^2 + (a_2 - w)^2 y^2 + (a_2^2 - w^2)^2] = 0 \tag{14}$$

In this case the coupler curves are obtained by the union of a circle and an ellipse, as shown in Fig. 3b. Major semi-axis of the ellipse is on Y axis (X axis) if w is greater (less) than a_2. If additionally $w = a_2$ then the coupler curve represented by the ellipse degenerates into a circle. Geometric singularities of the curve are given by $y_s = \pm(a_3 + w)$, $x_s = 0$. They are cusps.

4 Design Guidelines

The boundary curve of a family can be obtained by considering the equation of the family together with the derivative of the family with respect to the parameter. In particular, in the following coupler curves equation in 6 is considered, taking as a parameter of the family the eccentricity and coupler length, respectively, as shown in the numerical examples shown in Figs. 4 and 5. If one considers eccentricity as the family parameter, then the equation of the boundary curve becomes, (when a_3 is different from $-w$, w different from 0)

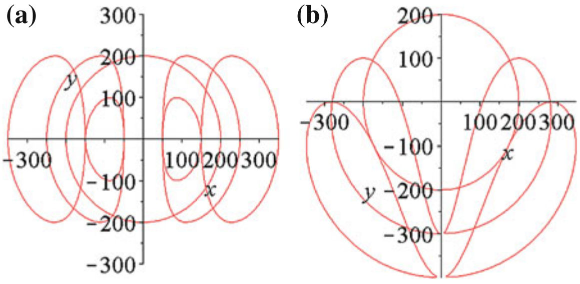


Fig. 2 Examples of Bernoulli quartics in (**a**) and Ruiz-Castizo quartics in (**b**)

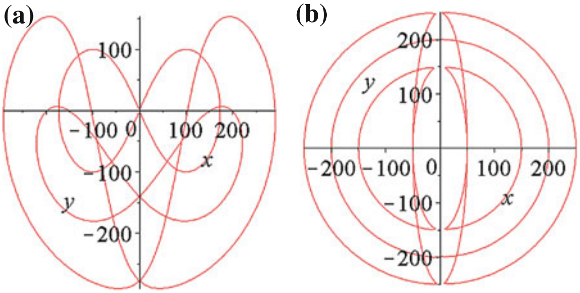


Fig. 3 Examples of lemniscates in (**a**) and Cardano motion curves in (**b**)

$$(w^2 + 2a_2w + a_2^2 - x^2 - y^2)(w^2 - 2a_2w + a_2^2 - x^2 - y^2) = 0 \tag{15}$$

If one considers a_3 as the family-parameter, then the equation of the boundary curve becomes, (when w is different from 0)

$$(w^2 + 2a_2w + a_2^2 - x^2 - y^2)(w^2 - 2a_2w + a_2^2 - x^2 - y^2) = 0 \tag{16}$$

According to the above-mentioned considerations, design guidelines of a slider-crank can be given in terms of shape and characteristics of the coupler curves as follows:

1. define limits for y and x in the plane of motion, say x_{max}, x_{min}, y_{max}, y_{min};
2. evaluate design parameters in Eqs. 15 and 16;
3. the family parameter can be chosen according to other design specifications.

In particular, in this context we can generate any symmetrical egg shaped path giving overall size of the curve in the plane of motion OXY, being characterized for example to have a coupler point with stationary curvature, when a suitable reference point P is chosen along the coupler link BC, as belonging to the cubic of stationary curvature $\mathcal{C}$. In fact, for the crank angle $\theta_2 = 0$, the cubic of stationary curvature degenerates in a φ-curve, as reported in [12]. Moreover, when P is chosen as coincident with the

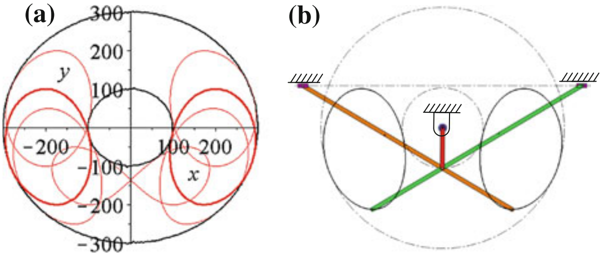


Fig. 4 Boundary curve and family curve of e parameter when $a_2 = 100$, $a_3 = 400$, $w = 200$ in (**a**) and an example of designed slider-crank with $e = -100$ in (**b**)

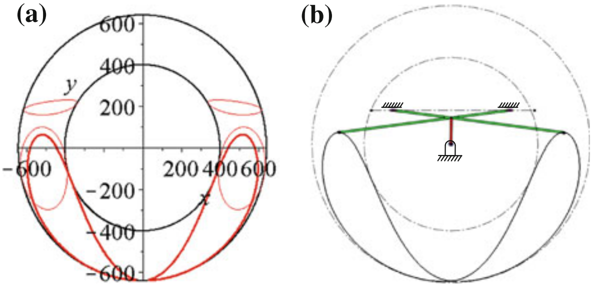


Fig. 5 Boundary curve and family curve of a_3 parameter when $a_2 = 120$, $e = 150$, $w = 270$ in (**a**) and an example of a designed slider-crank with a_3 = 520 in (**b**)

inflection pole J, or in general on the inflection circle, an approximate straight line can be also obtained, as reported in [11].

5 Conclusion

By studyng the kinematics of slider-cranks derived from λ-mechanism giving the coupler curve described by a fourth-order polynomial, this paper provides interesting characteristics processed by the classical slider-crank linkage when parameters of the mechanism are properly chosen. In particular, the coupler curve traced by a reference point of the coupler can be represented by quartics of Bernoulli, quartics of Ruiz-Castizo, lemniscate and producing Cardano motion. Furthermore, in this paper singularities of the coupler curve are investigated.

References

1. McCarthy, J.M., Joskowicz, L.: Kinematic Synthesis. Cambridge University Press, New York (2001)
2. Hall, A.S.: Kinematics and Linkage Design. Waveland Press. Inc., Prospect Heights (1961)
3. Hunt, K.H.: Kinematic Geometry of Mechanisms. Oxford University Press, New York (1990)
4. Shoup, T.E.: Centrodes of the slider-crank mechanism. In: 8th IFToMM World Congress on the Theory of Machines and Mechanisms. vol. 1, pp. 59–62. Prague (1991)
5. Dijksmann, E.A.: The inverted slider-crank used for the design of an approximate straight-line mechanism. Eng. Res. **61**, 129–134 (1995)
6. Prony, G.F.: Nouvelle Architecture Hydraulique, vol. II, Firmin Didot, Paris (1796)
7. Norton, R.L.: Design of Machinery. McGraw-Hill, New York (1999)
8. Hartenberg, R.S., Denavit, J.: Kinematic Synthesis of Linkages. McGraw Hill, New York (1964)
9. Gibson, C.G.: Elementary Geometry of Algebraic Curves. Cambridge University Press, New York (1998)
10. Schrocker, H.-P., Husty, M.L., McCarthy, J.M.: Kinematic mapping based evaluation of assembly modes for planar four-bar synthesis. In: Proceedings of ASME 29th Mechanism and Robotics Conference, Long Beach, DETC2005/MECH-85037 (2005)
11. Figliolini, G., Conte, M., Rea, P.: Algebraic algorithm for the kinematic analysis of crank/rocker mechanisms, ASME J. Mech. Robot. **4**(1), 011003–1 (1964)
12. Dijksmann, E.A.: Motion Geometry of Mechanisms. Cambridge University Press, London (1976)

Improved Muscle Wrapping Algorithms Using Explicit Path-Error Jacobians

Andreas Scholz, Ian Stavness, Michael Sherman, Scott Delp and Andrés Kecskeméthy

Abstract Muscle wrapping computations are an important feature in musculoskeletal simulations. In this paper we present a novel Jacobian-based method for line-based muscle-path computations over multiple general smooth surfaces allowing for second-order Newton-Raphson iterations. The method is based on the analytical determination of infinitesimal displacements along geodesics using Jacobi fields. It does not share the disadvantages of discretized methods in terms of non-smoothness when using surface discretizations, and high computational costs when using discretized spring-mass approaches. The paper focusses on the technical details of the proposed method, while specific biomechanical applications are left for future contributions. An example with three surfaces involving a surface with a general distribution of curvature shows the general applicability of the method.

Keywords Muscle wrapping · Jacobi fields · geodesics

A. Scholz (✉) · A. Kecskeméthy
University of Duisburg-Essen, Essen, Germany
e-mail: scholz.andreas@uni-due.de

A. Kecskeméthy
e-mail: andres.kecskemethy@uni-due.de

I. Stavness
University of Saskatchewan, Saskatoon, Canada
e-mail: stavness@gmail.com

M. Sherman · S. Delp
Stanford University, Santa Clara, USA
e-mail: msherman@stanford.edu

S. Delp
e-mail: delp@stanford.edu

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_44,

1 Introduction

A key task of musculoskeletal simulations is the computation of the transmission of tensional muscle force to joint moments and/or reaction forces. Muscles are commonly modeled as thin strings which take the locally shortest path between their origin and insertion points while wrapping frictionlessly around multiple wrapping surfaces that represent neighboring bones, tissue, and the neglected dimensions of the muscle. State-of-the-art approaches to line-based muscle wrapping can be divided into two main groups: (1) approaches using surface or path discretizations and (2) approaches using explicit smooth surfaces. Discretizing approaches such as [1, 2] yield fast approximate solutions and allow for using realistic bone geometry obtained from MRI or CT. However, they cause nonsmooth path motion at surface edges and hence only C^1 continuous behavior of the path length, which slows down variable-step-size integrators in dynamic simulations driven by muscle forces. Explicit smooth wrapping-surface approaches such as [3–5] provide continuous wrapping, but are limited to simple objects such as single spheres, single cylinders, or a compound of both, which are not always sufficient to represent general bone and surrounding soft tissue surfaces. Elastic approaches such as [4] circumvent this problem, but introduce new difficulties such as an oscillatory behavior of the muscle path. Recently, Stavness et al. 2012 [6] proposed a root-finding approach in which the total path is regarded as a concatenation of straight-line segments between geodesics on the surfaces, allowing to tackle general smooth surfaces while avoiding oscillatory behavior. The path is computed by iterating the positions of the boundary points of the geodesics until all transitions between adjacent segments are collinear. To this end, the Jacobian mapping variations of geodesic boundary points to variations of the path error is required. While this Jacobian can be determined by finite differences, such discretizations are expensive and also do not render smooth transitions between time steps. In this paper, we derive, based on the formulation [6], explicit formulas for the path-error Jacobian using differential-geometric properties of infinitesimal displacements along geodesics based on Jacobi fields. The approach is easy to implement, yields fast convergence and is thus well-suited for muscle wrapping applications.

2 Conditions for a Geodesic over Several Surfaces

We regard a string that is spanned between an origin point O and an insertion point I. The string wraps frictionlessly across a set of n wrapping surfaces $\mathcal{S}^i$ $(i = 1, \ldots, n)$ and minimizes the length with respect to all other neighboring trajectories connecting O and I. The total muscle path results as a concatenation of $n - 1$ straight-line segments between the surfaces, two straight lines to points O and I, and n geodesics on the surfaces. Each geodesic γ^i is uniquely defined by its start point P^i and end point Q^i, and each straight-line segment is defined by the unit direction vector $\underline{e}^i$

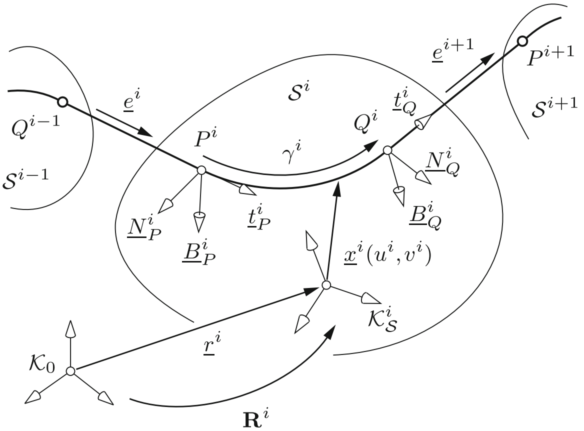


Fig. 1 Global path error components

and its respective boundary points Q^{i-1} and P^i for $1 < i < n + 1$, O and P^1 for $i = 1$ and Q^n and I for $i = n + 1$.

Assume that the section of interest of surface $S^i \in \mathbb{R}^3$ can be parameterized by a nonsingular differentiable function $\underline{x}^i(u^i, v^i) : \mathbb{R}^2 \mapsto \mathbb{R}^3$ with respect to a surface base frame $\mathcal{K}^i_{\mathcal{S}}$ in terms of two surface coordinates $(u^i, v^i) \in \mathbb{R}^2$ such that for the outwards normal it holds (Fig. 1)

$$\underline{N}^i := \frac{\underline{x}^i_u \times \underline{x}^i_v}{\| \underline{x}^i_u \times \underline{x}^i_v \|}, \quad \underline{x}_{(.)} := \frac{\partial \underline{x}}{\partial (.)}. \tag{1}$$

Moreover, let $\underline{t}^i$ be the geodesic's tangent and let $\underline{B}^i = \underline{t}^i \times \underline{N}^i$ be its binormal.

At the solution configuration, all transitions between adjacent segments are collinear (see Fig. 1). If they are not, for each geodesic γ^i, four possible local path errors arise from the orthogonality conditions

$$\underline{\varepsilon}^i(\underline{q}^i, \mathcal{K}^i_{\mathcal{S}}) := \begin{bmatrix} \underline{e}^i \cdot \underline{N}^i_P \\ \underline{e}^i \cdot \underline{B}^i_P \\ \underline{e}^{i+1} \cdot \underline{N}^i_Q \\ \underline{e}^{i+1} \cdot \underline{B}^i_Q \end{bmatrix}, \quad \underline{q}^i = \begin{bmatrix} u^i_P \\ v^i_P \\ u^i_Q \\ v^i_Q \end{bmatrix}. \tag{2}$$

The local path errors can be assembled into the global path-error vector

$$\underline{\varepsilon}(\underline{q}, \mathcal{K}) = \begin{bmatrix} \underline{\varepsilon}^1 \\ \underline{\varepsilon}^2 \\ \vdots \\ \underline{\varepsilon}^n \end{bmatrix} \in \mathbb{R}^{4n \times 1}, \quad \underline{q} = \begin{bmatrix} \underline{q}^1 \\ \underline{q}^2 \\ \vdots \\ \underline{q}^n \end{bmatrix} \in \mathbb{R}^{4n \times 1}, \quad \mathcal{K} = \begin{bmatrix} \mathcal{K}^1_{\mathcal{S}} \\ \mathcal{K}^2_{\mathcal{S}} \\ \vdots \\ \mathcal{K}^n_{\mathcal{S}} \end{bmatrix}. \tag{3}$$

yielding the nonlinear root condition for the muscle path

$$\underline{\varepsilon}(\underline{q}, \mathcal{K}) = \underline{0}. \tag{4}$$

For each time step, the set of reference frames contained in $\mathcal{K}$ is fixed, and Eq. (4) can be solved for the unknown geodesic boundary-point coordinates $\underline{q}$ using a Newton-Raphson method. This requires knowledge of the Jacobian $\mathbf{J}_q := \partial\underline{\varepsilon}/\partial\underline{q} \in \mathbb{R}^{4n\times 4n}$ containing the partial derivatives

$$\mathbf{J}_q = \begin{bmatrix}
\frac{\partial\underline{\varepsilon}^1}{\partial\underline{q}_P^1} & \frac{\partial\underline{\varepsilon}^1}{\partial\underline{q}_Q^1} & \frac{\partial\underline{\varepsilon}^1}{\partial\underline{q}_P^2} & & & & & & & \\
 & \frac{\partial\underline{\varepsilon}^2}{\partial\underline{q}_Q^1} & \frac{\partial\underline{\varepsilon}^2}{\partial\underline{q}_P^2} & \frac{\partial\underline{\varepsilon}^2}{\partial\underline{q}_Q^2} & \frac{\partial\underline{\varepsilon}^2}{\partial\underline{q}_P^3} & & & & & \\
 & & \ddots & \ddots & \ddots & \ddots & & & & \\
 & & & \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_Q^{i-1}} & \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_P^i} & \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_Q^i} & \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_P^{i+1}} & & & \\
 & & & & \ddots & \ddots & \ddots & \ddots & & \\
 & & & & & \frac{\partial\underline{\varepsilon}^{n-1}}{\partial\underline{q}_Q^{n-2}} & \frac{\partial\underline{\varepsilon}^{n-1}}{\partial\underline{q}_P^{n-1}} & \frac{\partial\underline{\varepsilon}^{n-1}}{\partial\underline{q}_Q^{n-1}} & \frac{\partial\underline{\varepsilon}^{n-1}}{\partial\underline{q}_P^n} & \\
 & & & & & & & \frac{\partial\underline{\varepsilon}^n}{\partial\underline{q}_Q^{n-1}} & \frac{\partial\underline{\varepsilon}^n}{\partial\underline{q}_P^n} & \frac{\partial\underline{\varepsilon}^n}{\partial\underline{q}_Q^n}
\end{bmatrix}. \tag{5}$$

A general block-row $\mathbf{J}_q\,(i,:)$ in Eq. (5) comprises four submatrices

$$\mathbf{J}_q\,(i,:) = \begin{bmatrix} \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_Q^{i-1}} & \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_P^i} & \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_Q^i} & \frac{\partial\underline{\varepsilon}^i}{\partial\underline{q}_P^{i+1}} \end{bmatrix} \tag{6}$$

which represent two coupling terms $\partial\underline{\varepsilon}^i/\partial\underline{q}_Q^{i-1}$ if $i > 1$, and $\partial\underline{\varepsilon}^i/\partial\underline{q}_P^{i+1}$ if $i < n$, as well as the local path-error Jacobians $\mathbf{J}_q^i := \partial\underline{\varepsilon}^i/\partial\underline{q}^i$

$$\mathbf{J}_q^i = \begin{bmatrix}
\frac{\partial}{\partial u_P^i}\left(\underline{e}^i\cdot\underline{N}_P^i\right) & \frac{\partial}{\partial v_P^i}\left(\underline{e}^i\cdot\underline{N}_P^i\right) & \frac{\partial}{\partial u_Q^i}\left(\underline{e}^i\cdot\underline{N}_P^i\right) & \frac{\partial}{\partial v_Q^i}\left(\underline{e}^i\cdot\underline{N}_P^i\right) \\
\frac{\partial}{\partial u_P^i}\left(\underline{e}^i\cdot\underline{B}_P^i\right) & \frac{\partial}{\partial v_P^i}\left(\underline{e}^i\cdot\underline{B}_P^i\right) & \frac{\partial}{\partial u_Q^i}\left(\underline{e}^i\cdot\underline{B}_P^i\right) & \frac{\partial}{\partial v_Q^i}\left(\underline{e}^i\cdot\underline{B}_P^i\right) \\
\frac{\partial}{\partial u_P^i}\left(\underline{e}^{i+1}\cdot\underline{N}_Q^i\right) & \frac{\partial}{\partial v_P^i}\left(\underline{e}^{i+1}\cdot\underline{N}_Q^i\right) & \frac{\partial}{\partial u_Q^i}\left(\underline{e}^{i+1}\cdot\underline{N}_Q^i\right) & \frac{\partial}{\partial v_Q^i}\left(\underline{e}^{i+1}\cdot\underline{N}_Q^i\right) \\
\frac{\partial}{\partial u_P^i}\left(\underline{e}^{i+1}\cdot\underline{B}_Q^i\right) & \frac{\partial}{\partial v_P^i}\left(\underline{e}^{i+1}\cdot\underline{B}_Q^i\right) & \frac{\partial}{\partial u_Q^i}\left(\underline{e}^{i+1}\cdot\underline{B}_Q^i\right) & \frac{\partial}{\partial v_Q^i}\left(\underline{e}^{i+1}\cdot\underline{B}_Q^i\right)
\end{bmatrix}. \tag{7}$$

In this Jacobian, the derivatives of the normal vectors $\underline{N}^i_P$, $\underline{N}^i_Q$ as well as of the unit vectors $\underline{e}^i$, $\underline{e}^{i+1}$ with respect to the coordinates u^i_P, v^i_P and u^i_Q, v^i_Q can be determined directly from local surface geometry. These derivations are left out here due to lack of space. On the other hand, the partial derivatives of the geodesic's binormals $\underline{B}^i_P$ and $\underline{B}^i_Q$ require through their definition $\underline{B}^i = \underline{t}^i \times \underline{N}^i$ the partial derivatives of the tangent vectors $\underline{t}^i_P$ and $\underline{t}^i_Q$ with respect to the coordinates of both geodesic boundary points. These derivatives involve cross-over differential mappings over geodesics which can be computed using Jacobi fields, as discussed next.

3 Coupled End-Point Derivatives Across Geodesics

In this section, we review some fundamental concepts of differential geometry (see [7, 8]) and apply them to the given problem. Let the geodesic γ be given in polar form, i.e. assume that the start point P is fixed. Let s be the arc length of γ and let θ be an angular coordinate defining the initial direction of γ (Fig. 2). By the Lemma of Gauss it holds $\mathrm{F}_{\text{polar}} = \underline{x}_\theta \cdot \underline{x}_s = 0 \; \forall \; s \neq 0$.

When the angle θ is varied, a point Q at some constant distance from the pole P will travel an arc length β_Q along a geodesic circle. We define the positive arc direction of such a circle to be oriented along the binormal vector $\underline{B}_Q$ at Q. The partial derivative $a = \partial\beta/\partial\theta$ of the arc length β at any point of the geodesic fulfills the scalar Jacobi equation (see [7, 9])

$$a'' + K\,a = 0\,, \quad a(s=0) = 0\,, \quad a'(s=0) = 1\,, \quad (.)' := \partial(.)/\partial s\,, \qquad (8)$$

where K is the Gaussian curvature. The scalar Jacobi Eq. (8) can be integrated together with the differential equations of the geodesic.

For the computation of the partial derivatives of the tangent vectors with respect to the boundary-point coordinates, a local coordinate-transformation is carried out first. Let $\mathrm{d}s_Q$ and $\mathrm{d}\beta_Q$ be infinitesimal increments along the geodesic γ and the geodesic circle at point Q for the fixed pole P (Fig. 2). Likewise, let $\mathrm{d}s_P$ and $\mathrm{d}\beta_P$

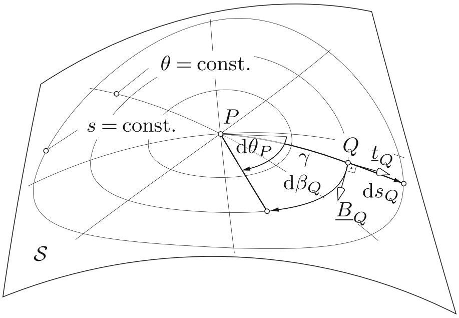


Fig. 2 Geodesic polar coordinates (θ, s) for a fixed pole P

be the infinitesimal increments along the geodesic and the geodesic circle at point P when Q is taken as the fixed pole. Here, $\mathrm{d}s_P$ is oriented along $\underline{t}_P$, i.e. in direction of length shortening, while $\mathrm{d}\beta_P$ is oriented along $\underline{B}_P$. Locally, the transformation of the differentials of β, s and u, v is given by

$$\frac{\partial s}{\partial u} = \underline{x}_u \cdot \underline{t}\,, \quad \frac{\partial s}{\partial v} = \underline{x}_v \cdot \underline{t}\,, \quad \frac{\partial \beta}{\partial u} = \underline{x}_u \cdot \underline{B}\,, \quad \frac{\partial \beta}{\partial v} = \underline{x}_v \cdot \underline{B}\,. \tag{9}$$

The derivatives of $\underline{t}_P$ and $\underline{t}_Q$ with respect to s_P and s_Q can be obtained from the Frenet-Serret formulas (see, e.g. [8])

$$\frac{\partial \underline{t}_P}{\partial s_P} = \kappa_P\, \underline{n}_P\,, \quad \frac{\partial \underline{t}_P}{\partial s_Q} = \underline{0}\,, \quad \frac{\partial \underline{t}_Q}{\partial s_P} = \underline{0}\,, \quad \frac{\partial \underline{t}_Q}{\partial s_Q} = \kappa_Q\, \underline{n}_Q\,, \tag{10}$$

where $\underline{n}_P := \underline{x}''_P/\kappa_P$ and $\underline{n}_Q := \underline{x}''_Q/\kappa_Q$ are the unit normals of the geodesic and κ_P and κ_Q are the curvatures of the geodesic at P and Q.

The concept of the Jacobi field along the geodesic γ allows for the computation of the derivatives of $\underline{t}_P$ and $\underline{t}_Q$ with respect to β_P and β_Q. For an infinitesimal motion $\mathrm{d}\beta_Q$ of the geodesic's end point Q, the tangent vector $\underline{t}_P$ at the start point P rotates about the surface normal $\underline{N}_P$ with an angle $\mathrm{d}\theta_P$. This relation is given by the scalar Jacobi field at Q, yielding

$$\frac{\partial \underline{t}_P}{\partial \beta_Q} = \frac{\partial \underline{t}_P}{\partial \theta_P}\frac{\partial \theta_P}{\partial \beta_Q} = \underline{B}_P\, a_Q^{-1}\,. \tag{11}$$

Here $\partial \underline{t}_P/\partial \theta_P$ is a local derivative, while $\partial \theta_P/\partial \beta_Q$ depends on the geodesic. Note that the latter term becomes singular at conjugate points of P, which are defined by a vanishing Jacobi field $a = 0$. Analogously, symmetry yields

$$\frac{\partial \underline{t}_Q}{\partial \beta_P} = \frac{\partial \underline{t}_Q}{\partial \theta_Q}\frac{\partial \theta_Q}{\partial \beta_P} = -\underline{B}_Q\, \hat{a}_P^{-1}\,, \tag{12}$$

where $\hat{a}$ denotes the "backwards" Jacobi field obtained by integrating Eq. (8) from Q to P.

The other derivatives are obtained similarly using the definition $\underline{t} = \partial \underline{x}/\partial s$

$$\frac{\partial \underline{t}_P}{\partial \beta_P} = \frac{\partial \underline{t}_P}{\partial \theta_Q}\frac{\partial \theta_Q}{\partial \beta_P} = \frac{\partial}{\partial \theta_Q}\left(\frac{\partial \underline{x}_P}{\partial s_P}\right)\hat{a}_P^{-1} \tag{13}$$

and the theorem of Schwarz, yielding

$$\frac{\partial \underline{t}_P}{\partial \beta_P} = \frac{\partial}{\partial s_P}\left(\frac{\partial \underline{x}_P}{\partial \theta_Q}\right)\hat{a}_P^{-1} = \frac{\partial}{\partial s_P}\left(\underline{B}_P\, \hat{a}_P\right)\hat{a}_P^{-1} = -\tau_P\, \underline{N}_P - \underline{B}_P\, \hat{a}'_P\, \hat{a}_P^{-1}\,, \tag{14}$$

where τ is the geodesic's torsion. Likewise, it holds by symmetry

$$\frac{\partial \underline{t}_Q}{\partial \beta_Q} = -\tau_Q \, \underline{N}_Q + \underline{B}_Q \, a'_Q \, a_Q^{-1} \, . \tag{15}$$

4 Results

The formulas described above can be assembled into a modular program for muscle-path computations. Here we show the results of our implementation in Matlab. In this implementation, the muscle path can be spanned over several general surfaces (Fig. 3a), and both the end points as well as all surfaces can perform arbitrary spatial motions. Each time frame comprises two types of iterations: an inner loop and an outer loop. The inner loop carries out geodesic shooting from point P such that point Q is reached. Each iteration step consists of numerically integrating the geodesic equations and Eq. (8) with given initial direction θ. Corrections of the geodesic length and the initial direction are obtained by projecting the difference $\Delta \underline{x}_E = \underline{x}_Q - \underline{x}_E$ from the current trial end point E to the target point Q onto the two polar directions at E

$$\Delta s_E = \Delta \underline{x}_E \cdot \underline{t}_E \tag{16}$$

$$\Delta \theta_P = \Delta \underline{x}_E \cdot \underline{B}_E \, a_E^{-1} \, . \tag{17}$$

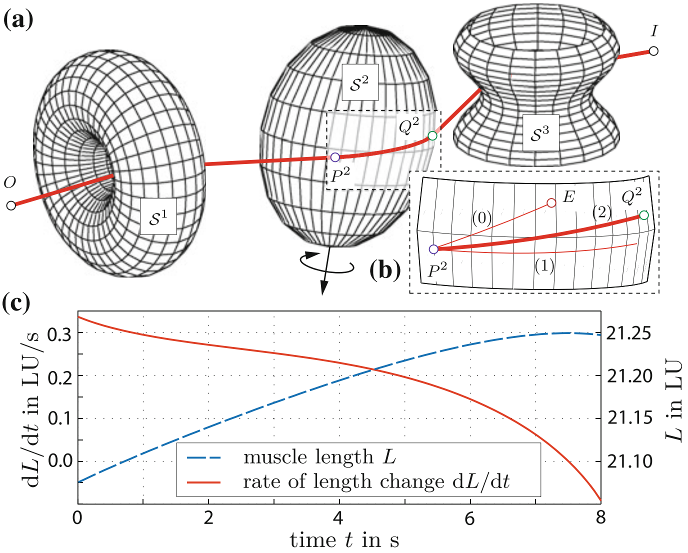


Fig. 3 **a** Sample application. **b** Inner-loop iterations using Eqs. (16, 17). **c** Smooth muscle length and rate of length change

The outer loop carries out the Newton-Raphson iteration for path error $\underline{\varepsilon}^{(k)}$

$$\underline{q}^{(k+1)} = \underline{q}^{(k)} - \left[\mathbf{J}_q^{(k)}\right]^{-1} \underline{\varepsilon}^{(k)} . \tag{18}$$

Figure 3a shows a sample motion with three surfaces, where S^3 represents the case of a general distribution of curvature and where the nonsymmetric ellipsoid $\mathcal{S}^2$ is rotating about a skew axis. Due to the closedness of subsequent time frames, both loops converge in 2-3 iterations per frame. Figure 3b shows one inner loop for $\mathcal{S}^2$. Figure 3c contains the time histories of total muscle length and its rate of length change, showing that both curves are smooth.

5 Conclusions

The presented approach is suitable for efficient smooth muscle-wrapping based on second-order Newton iterations. The path-error Jacobian can be determined explicitly by solving the Jacobi field Eq. (8). Second-order convergence can be achieved for inner loop geodesic shooting iterations using geodesic polar coordinates. The algorithms are operational for an arbitrary number of surfaces which can be parameterized explicitly. Future publications will provide a comparison to existing approaches and the application to specific biomechanics examples, and may involve the generalization of the formulation to multiple-patch as well as to implicit surface parametrizations.

Acknowledgments Adrian Butscher, Leonidas Guibas, Justin Solomon, and Matthew Millard. This work was supported by the German National Academic Foundation, the Natural Sciences and Engineering Research Council of Canada, and NIH grants U54 GM072970 (Simbios) and R24 HD065690 (NCSRR).

References

1. Desailly, E., Sardain, P., Khouri, N., Yepremian, D., Lacouture, P.: The convex wrapping algorithm: A method for identifying muscle paths using the underlying bone mesh. J. Biomech. **43**, 2601–2607
2. Gao, F., Damsgaard, M., Rasmussen, J., Christensen, S.T.: Computational method for muscle-path representation in musculoskeletal models. Biol. Cybern. **87**, 199–210 (2002)
3. Charlton, I.W., Johnson, G.R.: Application of spherical and cylindrical wrapping algorithms in a musculoskeletal model of the upper limb. J. Biomech. **34**, 1209–1216 (2001)
4. Marsden, S.P., Swailes, D.C., Johnson, G.R.: Algorithms for exact multi-object muscle wrapping and application to the deltoid muscle wrapping around the humerus. In: Proceedings of the Institution of Mechanical Engineers. Part H, Journal of Engineering in Medicine, vol. **222**(7), 1081–1095 (2008)
5. Garner, B.A., Pandy, M.G.: The obstacle-set method for representing muscle paths in musculoskeletal simulations. Comput. Methods Biomech. Biomed. Eng. **3**, 1–30 (1998)

6. Stavness, I., Sherman, M., Delp, S.L.: A general approach to muscle wrapping over multiple surfaces, In: Proceedings of the American Society of Biomechanics (2012)
7. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces. 1st edn. Prentice Hall, ohio. Feb 1976
8. Struik, D.J.: Lectures on Classical Differential Geometry, 2nd edn. Dover Publications, New York, April 1988
9. Thielhelm, H., Vais, A., Brandes, D., Wolter, F.-E.: Connecting geodesics on smooth surfaces. Vis. Comput. **28**, 529–539 (2012)

Erratum to: Computational Kinematics

Federico Thomas and Alba Pérez Gracia

Erratum to: F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*, DOI 10.1007/978-94-007-7214-4

In Chap. 1 entitled "On the Minimum 2-Norm Positive Tension for Wire-Actuated Parallel Manipulators", the affiliation of Leila Notash should read as "Queen's University, Kingston, ON, Canada" instead of "Institute of Complex Systems, Queen's University, Kingston, ON, Canada".

In Chap. 17 entitled "An Open-Source Toolbox for Motion Analysis of Closed-Chain Mechanisms", the order of the authors should read as "Lluís Ros, Josep M. Porta, Oriol Bohigas, Montserrat Manubens, Carlos Rosales and Léonard Jaillet" instead of "Josep M. Porta, Lluís Ros, Oriol Bohigas, Montserrat Manubens, Carlos Rosales and Léonard Jaillet".

In Chap. 18 entitled "Spherical Parallel Mechanism with Variable Target Point", the order of the authors should read as "Tsuyoshi Ikeda, Yukio Takeda and Daisuke Matsuura" instead of "Yukio Takeda, Tsuyoshi Ikeda and Daisuke Matsuura".

The online version of the original book can be found under 10.1007/978-94-007-7214-4

F. Thomas
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: fthomas@iri.upc.edu

A. Pérez Gracia
Mechanical Engineering, Idaho State University, Pocatello, ID, USA
e-mail: perealba@isu.edu

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4_45,

Author Index

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4,

Subject Index

F. Thomas and A. Pérez Gracia (eds.), *Computational Kinematics*,
Mechanisms and Machine Science 15, DOI: 10.1007/978-94-007-7214-4,