Robotica (2019) volume 37, pp. 1190–1201.
doi:10.1017/S0263574718000462

Higher-order Taylor approximation of finite motions in mechanisms

J. J. de Jong†*, A. Müller‡ and J. L. Herder§

†*University of Twente, Enschede, The Netherlands*
‡*Johannes Kepler University, Linz, Austria. E-mail: a.mueller@jku.at*
§*Delft University of Technology, Delft, The Netherlands. E-mail: j.l.herder@tudelft.nl*

(Accepted May 5, 2018)

SUMMARY

Higher-order derivatives of kinematic mappings give insight into the motion characteristics of complex mechanisms. Screw theory and its associated Lie group theory have been used to find these derivatives of loop closure equations up to an arbitrary order. In this paper, this is extended to the higher-order derivatives of the solution to these loop closure equations to provide an approximation of the finite motion of serial and parallel mechanisms. This recursive algorithm, consisting solely of matrix operations, relies on a simplified representation of the higher-order derivatives of open chains. The method is applied to a serial, a multi-DOF parallel, and an overconstrained mechanism. In all cases, adequate approximation is obtained over a large portion of the workspace.

KEYWORDS: Higher-order kinematics; Taylor approximation; Screw theory; 5-bar mechanism; Bennet linkage.

1. Introduction

Screw theory is frequently used to analyze the instantaneous motion of spatial kinematics. This theory gives the instantaneous kinematic relations between the spatial angular and linear velocities of bodies (twists) and constraint forces and moments (constraint wrenches) acting on a mechanism. This differential analysis is only available in the pose of inspection, and in general does not give a description of the finite motion of a mechanism. On the other hand, closed-form solutions to the geometric closure equations are not always available or are intricate to obtain for more complex mechanisms. This hinders the use of algebraic methods for expressing the finite motion of mechanisms. For synthesis and analysis purposes, attempts have been made to extend the infinitesimal screw analysis using higher-order derivatives. Bartkowiak and Woernle[1] used the higher-order derivatives of screws to find the conditions for overconstrained single loop linkages to have a single degree of freedom (DOF). Their numerical method yields an estimated maximum number of derivatives required to guarantee finite local mobility. Wohlhart[2,3] coined the term "order of shakiness." It defines to which order an arbitrary input still satisfies the higher-order derivatives of the loop closure equations. In ref. [4], several mechanisms are discussed that do not possess a finite mobility but still exhibit a higher-order local differential mobility which in practice leads to an unexpected large range of motion. Derivatives up to an arbitrary order of loop closure equations can be found by taking Lie brackets of instantaneous screw axes, which can be expressed as matrix multiplications of twists.[5,6] This paves the way for algorithmic differentiation-free derivatives of the loop closure equations.[7]

However, higher-order derivatives and approximations of finite motion in closed loop mechanisms were not yet reported. This involves finding the higher-order derivatives of the solution to implicit closure equations. These solutions can be an inverse kinematic (IK) model, forward kinematic (FK) model, or other types of mappings, relating the dependent and independent coordinates in the kinematic loop.

* Corresponding author. E-mail: j.j.dejong@utwente.nl

These higher-order derivatives and approximations of finite motion can be used for analysis of the admissible motion of mechanisms. This can be extended to enhanced numerical methods for the simulation of the kinematics. Furthermore, the derivatives of the design criteria can be algebraically expressed as a function of geometric properties such as link lengths, potentially aiding the synthesis of specific kinematics, such as straight line mechanisms. Another possible application is the description of the derivatives of the dynamics of mechanisms, particularly to determine the conditions for dynamic balance in arbitrary mechanisms. A system is dynamically balanced when all shaking forces and moments vanish for all admissible motions.[8] This implies that the angular and linear momentum should be constant (for practical purposes usually zero) and all of their derivatives should remain zero throughout motion. These higher-order derivatives of momentum should provide the geometric and dynamic conditions for dynamic balancing. For path planning, a sufficient smooth function in actuator coordinates is required, i.e., with a sufficient number of derivatives. Therefore, the higher derivatives of kinematic mappings between the end-effector and actuators are needed.

Unfortunately, processing these higher-order multivariate derivatives requires elaborate bookkeeping, as can be seen in the implementation of the higher-order chain rule, the Faa di Bruno's rule.[9] This renders it arduous to find the solution to the implicitly formulated higher-order constraints.

In this paper, a simplified representation of the higher-order derivatives of the screw systems is presented, which directly follows from the product of exponentials of Brocket.[10] With Vetter's method for managing matrix derivatives[11] this enables us to obtain a recursive, differentiation-free algorithm for higher-order derivatives of the solution to the closure equations. Using the resulting higher-order derivatives of the Jacobians, a Taylor approximation of the open and closed loop kinematics is performed. The method is exemplified with an approximation of the finite motion of three mechanisms: (1) a serial 6 DOF manipulator, (2) a parallel 5-bar mechanism, (3) and an overconstrained but mobile Bennet linkage. A preliminary version of this work is presented in ref. [12].

Before we introduce the higher-order derivatives of the loop closure solution, the screw algebra is revisited and applied to an open chain in Section 2.1. Based on this, a simplified representation of the higher-order derivatives of an open chain is presented in Section 2.2. Subsequently, the loop closure equations and the matrix derivatives are revisited in Sections 2.3–2.4. Using these rules, the algorithm for determining the higher-order derivatives of the loop closure and its Taylor expansion are presented in Section 2.5 and its implementation is shown in three examples in Sections 3.1–3.3.

2. Method

2.1. Concepts and notation

In the notation of screw theory, as used in this paper, a reference frame (ψ_i) is associated to each rigid body i. Points in space ($\boldsymbol{a}$) can be expressed with respect to this reference frame (denoted with superscript $\boldsymbol{a}^i$). In the homogeneous representation, the $\boldsymbol{a}^i$-vector is appended with a 1. A change of reference frame from frame i to j follows from the homogeneous transformation matrix ($\boldsymbol{H}_i^j$) that consists of a rotation matrix ($\boldsymbol{R}$) and a translation vector ($\boldsymbol{o}$).

$$\boldsymbol{a}^j = \boldsymbol{H}_i^j \boldsymbol{a}^i \qquad\qquad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{o} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{1}$$

The time derivative of the transformation matrix is given by the twist ($\boldsymbol{t}_i^{k,j}$), i.e., the generalized velocity of body i with respect to body j expressed in frame k. For clarity reasons, the subscript and second superscript are omitted when unambiguous. The twist is a vector containing the angular ($\boldsymbol{\omega}$) and translational ($\boldsymbol{v}$) velocity ($\boldsymbol{t}^\top = [\boldsymbol{\omega}^\top, \boldsymbol{v}^\top]$). The $[\boldsymbol{\omega}\times]$ denotes the skew symmetric matrix of $\boldsymbol{\omega}$.

$$\dot{\boldsymbol{H}}_i^j = [\boldsymbol{t}_i^{j,j}\times]\boldsymbol{H}_i^j \qquad\qquad [\boldsymbol{t}\times] = \begin{bmatrix} [\boldsymbol{\omega}\times] & \boldsymbol{v} \\ \boldsymbol{0} & 0 \end{bmatrix} \tag{2}$$

https://doi.org/10.1017/S0263574718000462 Published online by Cambridge University Press

The twist's "frame of expression" changes with the adjoint transformation matrices here denoted with $Ad(\boldsymbol{H})$.

$$\boldsymbol{t}^j = Ad(\boldsymbol{H}_i^j)\boldsymbol{t}^i \qquad Ad(\boldsymbol{H}) = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0} \\ [\boldsymbol{o}\times]\boldsymbol{R} & \boldsymbol{R} \end{bmatrix} \tag{3}$$

The time derivative of adjoint transformation matrix is given in terms of instantaneous transformation matrix $ad(\boldsymbol{t})$.

$$\frac{d}{dt}\left(Ad(\boldsymbol{H}_i^j)\right) = ad(\boldsymbol{t}_i^{j,j})Ad(\boldsymbol{H}_i^j) \qquad ad(\boldsymbol{t}) = \begin{bmatrix} [\boldsymbol{\omega}\times] & \boldsymbol{0} \\ [\boldsymbol{v}\times] & [\boldsymbol{\omega}\times] \end{bmatrix} \tag{4}$$

This matrix itself can be expressed in another reference frame according to a nested transform

$$ad(\boldsymbol{t}^j) = ad\left(Ad(\boldsymbol{H}_i^j)\boldsymbol{t}^i\right) = Ad(\boldsymbol{H}_i^j)ad(\boldsymbol{t}^i)Ad(\boldsymbol{H}_j^i) \tag{5}$$

Using these twists and their exponentials, a concise formulation for the FK mapping of an open chain is available in the form of Brockett's product of exponentials[10]

$$\boldsymbol{H}_n^0 = \prod_{i=0}^{n} \boldsymbol{H}_i^{i-1}(q_i) = \prod_{i=0}^{n} \exp([q_i\boldsymbol{s}_i^0\times])\boldsymbol{H}_n^0(0) \tag{6}$$

Here, the instantaneous screw vector $\boldsymbol{s}_i^0$, specifies the amount of twist of body i generated by the velocity of joint i expressed in global frame. This screw vector is therefore a purely geometric entity. As this screw vector is defined according to the ordering of the chain – always with respect to the previous body – the second superscript is omitted. This also means that the instantaneous screw vectors of lower kinematic pairs are constant when expressed in the connecting frames, i.e., $d/dt(\boldsymbol{s}_i^{i-1}) = d/dt(\boldsymbol{s}_i^i) = \boldsymbol{0}$.

2.2. *Derivatives of twist systems (open chain)*

For an open chain, the higher-order partial derivatives can be found using the transformations of the previous section. A chain of transformations can be decomposed into a part which is constant and part which is varying with respect to this particular derivative. The nested transform (5) of the twist gives a concise formulation of the derivative of a chain, provided that $i \leq n$

$$\frac{\partial}{\partial q_i}\left(Ad(\boldsymbol{H}_n^0)\right) = Ad(\boldsymbol{H}_{i-1}^0)\frac{\partial}{\partial q_i}\left(Ad(\boldsymbol{H}_i^{i-1})\right)Ad(\boldsymbol{H}_n^i) \tag{7}$$

$$= Ad(\boldsymbol{H}_{i-1}^0)ad(\boldsymbol{s}_i^{i-1})Ad(\boldsymbol{H}_i^{i-1})Ad(\boldsymbol{H}_n^i) \tag{8}$$

$$= ad(\boldsymbol{s}_i^0)Ad(\boldsymbol{H}_n^0) \tag{9}$$

For the second order, such a concise representation also exists. For the consecutive derivative with respect to joint j, there exist two possibilities, it is either after body i in the chain (case 1) or before i in the chain (case 2), provided that $j \leq n$.

1. Case 1. ($i \leq j$) In the case that joint j is higher in the chain than i, the twist is unaffected ($\partial/\partial q_j(ad(\boldsymbol{s}_i^0)) = 0$). Therefore, the second partial derivative becomes

$$\frac{\partial}{\partial q_j}\frac{\partial}{\partial q_i}\left(Ad(\boldsymbol{H}_n^0)\right) = ad(\boldsymbol{s}_i^0)ad(\boldsymbol{s}_j^0)Ad(\boldsymbol{H}_n^0) \tag{10}$$

2. Case 2. ($i \geq j$) In the case that j is below i in the chain, we use the nested transform property to split the chain into a dependent and independent part. It may be verified that $\partial/\partial q_j(ad(\boldsymbol{s}_i^j)Ad(\boldsymbol{H}_n^j)) = 0$.

https://doi.org/10.1017/S0263574718000462 Published online by Cambridge University Press

Therefore,

$$\frac{\partial}{\partial q_j}\frac{\partial}{\partial q_i}\left(Ad\left(\boldsymbol{H}_n^0\right)\right) = \frac{\partial}{\partial q_j}\left(Ad\left(\boldsymbol{H}_j^0\right)ad\left(\boldsymbol{s}_i^j\right)Ad\left(\boldsymbol{H}_n^j\right)\right) \tag{11}$$

$$0 = \frac{\partial}{\partial q_j}\left(Ad\left(\boldsymbol{H}_j^0\right)\right)ad\left(\boldsymbol{s}_i^j\right)Ad\left(\boldsymbol{H}_n^j\right) \tag{12}$$

Using (9), a matrix chain can be found and collected again using the nested transform

$$\frac{\partial}{\partial q_j}\frac{\partial}{\partial q_i}\left(Ad\left(\boldsymbol{H}_n^0\right)\right) = ad\left(\boldsymbol{s}_j^0\right)ad\left(\boldsymbol{s}_i^0\right)Ad\left(\boldsymbol{H}_n^0\right) \tag{13}$$

This leaves us with an expression similar to (10), with the difference that the sequence of multiplication is swapped. This also follows from the symmetry (commutativity) property of mixed partial derivatives.

A consecutive application of (10) and (13) gives us the geometrical higher-order partial derivatives of the adjoint transformation matrix.

$D_{\boldsymbol{q}}^{\boldsymbol{\alpha}} = \frac{\partial^k}{\partial q_1^{\alpha_1}\ldots\partial q_n^{\alpha_n}}$ denotes the mixed partial derivative with respect to the elements of $\boldsymbol{q}$. Vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ comprises the order of derivatives corresponding to $\boldsymbol{q}$, running from the base to the end-effector. $k = \alpha_1 + \cdots + \alpha_n = |\boldsymbol{\alpha}|$ is the total order. The mixed partial derivative of the adjoint transformation matrix is

$$D_{\boldsymbol{q}}^{\boldsymbol{\alpha}}\left(Ad\left(\boldsymbol{H}_n\right)\right) = \prod_{i=1}^{n} ad\left(\boldsymbol{s}_i\right)^{\alpha_i}Ad\left(\boldsymbol{H}_n\right) \tag{14}$$

and similarly, for the higher partial derivatives of the instantaneous screw vectors

$$D_{\boldsymbol{q}}^{\boldsymbol{\alpha}}\left(\boldsymbol{s}_n\right) = \prod_{i=1}^{n-1} ad\left(\boldsymbol{s}_i\right)^{\alpha_i}\boldsymbol{s}_n \tag{15}$$

These results (14) and (15) are similar to that of,[6] with the difference that the index ranges to distinguish between the sequence of derivatives are taken into account by the ordering of $\boldsymbol{\alpha}$. From the commutative property of mixed partial derivatives, it follows that for any sequence of differentiation the same results are obtained.

2.3. Loop closure equations

The open-loop derivatives (14) and (15) can be used to obtain for the derivatives of loop closure equations, as a closed loop can be seen as a connection of open loops, e.g., a simple loop can be seen as an open chain of with the last link is fixed to the base. The loop closure equation (f) states how the members of the loop are constrained. It can be written in terms of locally validly chosen independent ($\boldsymbol{u}$) and dependent coordinates ($\boldsymbol{v}$), also termed input and output, respectively. The total set of coordinates, we call $\boldsymbol{r}^\top = \left[\boldsymbol{v}^\top \ \boldsymbol{u}^\top\right]$

$$f(\boldsymbol{v}, \boldsymbol{u}) = \boldsymbol{0} \tag{16}$$

The solution to this problem is denoted by c, which can be the inverse, forward, or any other kinematic model giving an exact relation between independent and dependent coordinates

$$\boldsymbol{v} = c(\boldsymbol{u}) \tag{17}$$

The solution (c) to the loop closure equation is usually not available for complex mechanisms. Therefore, we are looking for a Taylor expansion using higher-order derivatives of the constraint

https://doi.org/10.1017/S0263574718000462 Published online by Cambridge University Press

formulation and the open-loop derivatives of Section 2.2. We start with the first-order time derivative of the closure equation. This reads

$$\mathbf{0} = \frac{d}{dt}(f) = D_{\boldsymbol{u}}(f)\,\dot{\boldsymbol{u}} + D_{\boldsymbol{v}}(f)\,\dot{\boldsymbol{v}} = \boldsymbol{U}\dot{\boldsymbol{u}} + \boldsymbol{V}\dot{\boldsymbol{v}} \tag{18}$$

Here, $D_{\boldsymbol{u}}(f)$ and $D_{\boldsymbol{v}}(f)$ denote the matrix collection of all the first-order partial derivatives (Jacobians) of the constraint equations with respect to $\boldsymbol{u}$ and $\boldsymbol{v}$ while assuming independence of $\boldsymbol{u}$ and $\boldsymbol{v}$. This gives rise to the Jacobians $\boldsymbol{C}$ and $\boldsymbol{K}$, respectively, linking $\dot{\boldsymbol{v}}$ and $\dot{\boldsymbol{r}}$ to $\dot{\boldsymbol{u}}$

$$\dot{\boldsymbol{v}} = -\boldsymbol{V}^{-1}\boldsymbol{U}\dot{\boldsymbol{u}} = \boldsymbol{C}\dot{\boldsymbol{u}} = D_{\boldsymbol{u}}(c)\,\dot{\boldsymbol{u}} \qquad\qquad \dot{\boldsymbol{r}} = \boldsymbol{K}\dot{\boldsymbol{u}} = \begin{bmatrix}\boldsymbol{C}\\ \boldsymbol{I}\end{bmatrix}\dot{\boldsymbol{u}} \tag{19}$$

We already have seen that closure equations can be written as a function of transformation matrices of the open chain. Therefore, the higher-order partial derivatives of the open-loop equivalent ($D_{\boldsymbol{r}}^{\boldsymbol{\alpha}}(f)$) are available. Now, we are looking for a method of writing the higher-order derivatives of the constraint Jacobian $\boldsymbol{C}_k = D_{\boldsymbol{u}}^k(c)$.

2.4. *Multivariate matrix derivatives using Kronecker product*

The higher-order partial derivatives of matrices can be managed with the use of the Kronecker product.[11] Refer to Appendix 5 for definition and properties of the Kronecker product as used in this paper. Different from,[11] the higher-order derivatives of matrices are organized here as the concatenation of the derivatives of the columns $\boldsymbol{A} = [\boldsymbol{a}_1 \ldots \boldsymbol{a}_m]$

$$D_{\boldsymbol{x}}(\boldsymbol{A}) = \begin{bmatrix}D_{\boldsymbol{x}}(\boldsymbol{a}_1) & \ldots & D_{\boldsymbol{x}}(\boldsymbol{a}_m)\end{bmatrix} \tag{20}$$

$$D_{\boldsymbol{x}}^2(\boldsymbol{A}) = D_{\boldsymbol{x}}\left(D_{\boldsymbol{x}}(\boldsymbol{A})\right) = \begin{bmatrix}D_{\boldsymbol{x}}^2(\boldsymbol{a}_1) & \ldots & D_{\boldsymbol{x}}^2(\boldsymbol{a}_m)\end{bmatrix} \tag{21}$$

The partial derivatives of the product rule, the chain rule, Kronecker product, and the inverse matrix derivative are given as follows:

- Product rule of $\boldsymbol{A}(\boldsymbol{x}) \in \mathbb{R}^{n\times m}$ and $\boldsymbol{B}(\boldsymbol{x}) \in \mathbb{R}^{m\times q}$, with $\boldsymbol{x} \in \mathbb{R}^p$, in which $\boldsymbol{I}_p$ is the $p \times p$ identity matrix

$$D_{\boldsymbol{x}}(\boldsymbol{AB}) = \begin{bmatrix}D_{\boldsymbol{x}}(\boldsymbol{a}_1)\boldsymbol{B} & \ldots & D_{\boldsymbol{x}}(\boldsymbol{a}_m)\boldsymbol{B}\end{bmatrix} + \boldsymbol{A}D_{\boldsymbol{x}}(\boldsymbol{B}) \tag{22}$$

$$= D_{\boldsymbol{x}}(\boldsymbol{A})(\boldsymbol{B} \otimes \boldsymbol{I}_p) + \boldsymbol{A}D_{\boldsymbol{x}}(\boldsymbol{B}) \tag{23}$$

- Chain rule of nested variables $\boldsymbol{b}$ and $\boldsymbol{c}$

$$D_{\boldsymbol{c}}(\boldsymbol{A}(\boldsymbol{b}(\boldsymbol{c}))) = D_{\boldsymbol{b}}(\boldsymbol{A})(\boldsymbol{I}_m \otimes D_{\boldsymbol{c}}(\boldsymbol{b})) \tag{24}$$

- Derivatives of the Kronecker product can be given with the use of permutation matrices (refer to Appendix 5)

$$D_{\boldsymbol{x}}(\boldsymbol{A} \otimes \boldsymbol{B}) = (D_{\boldsymbol{x}}(\boldsymbol{A}) \otimes \boldsymbol{B})(\boldsymbol{I}_m \otimes \boldsymbol{P}_{q,p}) + \boldsymbol{A} \otimes D_{\boldsymbol{x}}(\boldsymbol{B}) \tag{25}$$

- Derivative of matrix inversion

$$D_{\boldsymbol{x}}\left(\boldsymbol{A}^{-1}\right) = -\boldsymbol{A}^{-1}D_{\boldsymbol{x}}(\boldsymbol{A})(\boldsymbol{A}^{-1} \otimes \boldsymbol{I}_p) \tag{26}$$

Recursive application of these rules allows the extension of these derivatives to higher orders.

2.5. *Higher-order derivatives of the constraint Jacobians*

Using the rules from the previous section, the second-order derivatives (Hessian) of the solution to the constraint equations are found. This is done by consecutive application of the chain rule, the product rule, and the inverse matrix derivative to the constraint Jacobian (19)

$$\boldsymbol{C}_2 = D_{\boldsymbol{u}}(\boldsymbol{C}_1) = -[D_{\boldsymbol{r}}(\boldsymbol{V}^{-1})(\boldsymbol{U}\otimes\boldsymbol{I}) - \boldsymbol{V}^{-1}D_{\boldsymbol{r}}(\boldsymbol{U})](\boldsymbol{I}\otimes\boldsymbol{K}) \tag{27}$$

$$= -\boldsymbol{V}^{-1}[D_{\boldsymbol{r}}(\boldsymbol{V})(\boldsymbol{C}_1\otimes\boldsymbol{I}) + D_{\boldsymbol{r}}(\boldsymbol{U})](\boldsymbol{I}\otimes\boldsymbol{K}) \tag{28}$$

After reordering and combination of the Kronecker products, we can find a concise formulation of the Hessian matrix

$$\boldsymbol{C}_2 = -\boldsymbol{V}^{-1}\begin{bmatrix} D_{\boldsymbol{r}}(\boldsymbol{V}) & D_{\boldsymbol{r}}(\boldsymbol{U})\end{bmatrix}(\boldsymbol{K}\otimes\boldsymbol{K}) = -\boldsymbol{V}^{-1}\boldsymbol{F}_2\boldsymbol{G}_2 \tag{29}$$

in which $\boldsymbol{F}_2 = D_{\boldsymbol{r}}^2(f)$. A further derivation is applied to show that a similar structure as the Hessian can be found for the third derivative

$$\boldsymbol{C}_3 = D_{\boldsymbol{u}}(\boldsymbol{C}_2) = -\boldsymbol{V}^{-1}\begin{bmatrix} D_{\boldsymbol{r}}(\boldsymbol{V}) & D_{\boldsymbol{r}}(\boldsymbol{F}_2) & \boldsymbol{F}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{C}_2\otimes\boldsymbol{K}\\ \boldsymbol{G}_2\otimes\boldsymbol{K}\\ D_{\boldsymbol{u}}(\boldsymbol{G}_2)\end{bmatrix} = -\boldsymbol{V}^{-1}\boldsymbol{F}_3\boldsymbol{G}_3 \tag{30}$$

For higher orders, this process can be repeated until the desired order is reached, giving us a recursive algorithm

$$\boldsymbol{C}_k = -\boldsymbol{V}^{-1}\begin{bmatrix} D_{\boldsymbol{r}}(\boldsymbol{V}) & D_{\boldsymbol{r}}(\boldsymbol{F}_{k-1}) & \boldsymbol{F}_{k-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{C}_{k-1}\otimes\boldsymbol{K}\\ \boldsymbol{G}_{k-1}\otimes\boldsymbol{K}\\ D_{\boldsymbol{u}}(\boldsymbol{G}_{k-1})\end{bmatrix} = -\boldsymbol{V}^{-1}\boldsymbol{F}_k\boldsymbol{G}_k \tag{31}$$

This algorithm consists of three steps:

1. The higher-order derivatives of $\boldsymbol{V}$ and $\boldsymbol{U}$ are substituted into the proper location of $\boldsymbol{F}_k$. These can be found *a priori* by higher-order screw derivatives of the open-loop equivalent.
2. The $\boldsymbol{G}_k$ matrix is filled with precursory, lower-order results.
3. The combination of the three matrices gives the subsequent partial derivative of the constraint Jacobian.

It is noted that repeating terms occur which could be combined to mitigate the computational burden. The simplification of this recursive formulation is outside the scope of this paper.

2.6. Higher-order Taylor approximation of closure equation

The Taylor approximation of the loop closure solution can now be written using the partial derivatives of the constraint Jacobians up to the kth order. We assume that at the evaluation point the closure constraint is satisfied, and that the evaluation point is at zero such that the Taylor series becomes a Maclaurin series. The input for the independent variables is given as a power (denoted with $^{\otimes i}$) of Kronecker products[11]

$$\boldsymbol{v}(\boldsymbol{u}) = \boldsymbol{v}(\boldsymbol{0}) + \boldsymbol{C}_1\boldsymbol{u} + \frac{1}{2!}\boldsymbol{C}_2(\boldsymbol{u}\otimes\boldsymbol{u}) + \frac{1}{3!}\boldsymbol{C}_3(\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}) + \cdots \approx \sum_{i=0}^{k}\frac{1}{i!}\boldsymbol{C}_i\boldsymbol{u}^{\otimes i} \tag{32}$$

3. Examples

To demonstrate the performance of this procedure for single and multi-DOF mechanisms, three examples are presented here. In the first example, the Taylor approximation along a trajectory of a serial robot is investigated to assess its performance close to the workspace boundary. In the second example, a multi-DOF approximation of a parallel manipulator is shown. In the third example, the method will be applied to a Bennet linkage to compare two approaches to deal with overconstrained mechanisms.

The computation times for these examples are recorded and reported in Table I. The time reported is for an average over 10 trails with 200 evaluation poses each. These computations were done with Matlab 2014b on a PC with an Intel Core i7 4800MQ running at 2.70 GHz.

https://doi.org/10.1017/S0263574718000462 Published online by Cambridge University Press

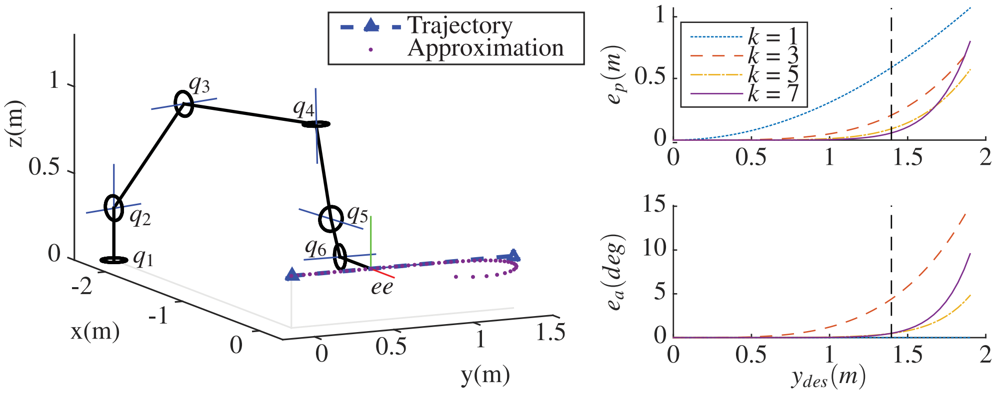

Fig. 1. *Left* The seventh-order Taylor expansion of the IK of a 6 DOF serial manipulator (solid black) over a horizontal trajectory (y_{des}). The evaluation point is [0, 0, 0.3]. For clarity, the robot is shown in isometric view at [0,0.5,0.3]. The trajectory is approximated until close to the end of the workspace at [0, 1.4, 0.3]. *Right* The Position and orientation error (e_p and e_a, respectively) of 4 orders of the Taylor approximation. The vertical line denotes the end of the workspace.

3.1. Approximate solution of the inverse kinematics of a 6 DOF serial manipulator

The IK models of general serial linkages are not readily available. For a 6 DOF serial manipulator, the IK is found by Husty *et al.*[13] by invoking algebraic methods to find and solve a univariate polynomial of order 16. In this first example, we will show the procedure to find the higher-order derivatives and Taylor approximation of a 6 DOF serial manipulator following a straight line, rendering it a single DOF expansion.

This manipulator consists of six bodies with six joints and an end-effector which is to follow a straight line in the y-direction. Therefore, the constraints are written as

$$f : \boldsymbol{H}_6^0(\boldsymbol{q})\boldsymbol{H}_{\text{ee}}^6 = \boldsymbol{H}_{\text{des}}^0(y_{\text{des}})$$

in which $\boldsymbol{H}_{\text{ee}}^6$ describes the location of the end-effector expressed in the sixth body fixed frame, and $\boldsymbol{H}_{\text{des}}^0$ the desired end-effector pose in the global frame. The independent coordinate is the the y-position along the straight line ($\boldsymbol{u} = y_{\text{des}}$). The dependent coordinates are the set of joint angles $\boldsymbol{v} = \boldsymbol{q}$.

Based on these constraint equations, the first- and second-order derivatives of the open-loop Jacobains are given as

$$\begin{aligned} D_{\boldsymbol{u}}(f) &= \boldsymbol{U} = \boldsymbol{s}_{\text{des}} & \qquad D_{\boldsymbol{v}}(f) &= \boldsymbol{V} = \begin{bmatrix} \boldsymbol{s}_1 & \cdots & \boldsymbol{s}_6 \end{bmatrix} \\ D_{\boldsymbol{s}}(\boldsymbol{U}) &= \boldsymbol{0} & D_{\boldsymbol{s}}(\boldsymbol{V}) &= \begin{bmatrix} \boldsymbol{0} & D_{\boldsymbol{r}}(\boldsymbol{s}_2) & \cdots D_{\boldsymbol{r}}(\boldsymbol{s}_6) \end{bmatrix} \end{aligned}$$

in which the product of sequence (15) is used to fill the higher-order Jacobains

$$D_{\boldsymbol{r}}(\boldsymbol{s}_i) = \begin{bmatrix} \boldsymbol{0} & \frac{\partial}{\partial q_1}(\boldsymbol{s}_i) & \cdots & \frac{\partial}{\partial q_{i-1}}(\boldsymbol{s}_i) & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & ad(\boldsymbol{s}_1)\boldsymbol{s}_i & \cdots & ad(\boldsymbol{s}_{i-1})\boldsymbol{s}_i & \boldsymbol{0} \end{bmatrix}$$

This Taylor approximation is made up to the seventh order. The result of this approximation can be seen in Fig. 1. In the initial pose – in which the end-effector is in [0,0,0.3] – the robot lies in the XZ-plane. The desired trajectory is a motion from this initial pose until 2 m in the y-direction, which is beyond the workspace boundary. The workspace ends at 1.4 m.

For higher orders, it can be seen that tracking converges to the desired path until the boundary of the workspace. Beyond this point, the trajectory estimate is no longer adequate and the approximation starts to diverge. In this case, the radius of convergence coincides with the edge of workspace.

3.2. Approximate solution of a 5-bar mechanism's motion

The higher-order derivatives and Taylor expansion technique is applied to approximate the IK solution of a 5-bar mechanism. We choose to describe the 5-bar as a connection of two open chains (a and

https://doi.org/10.1017/S0263574718000462 Published online by Cambridge University Press

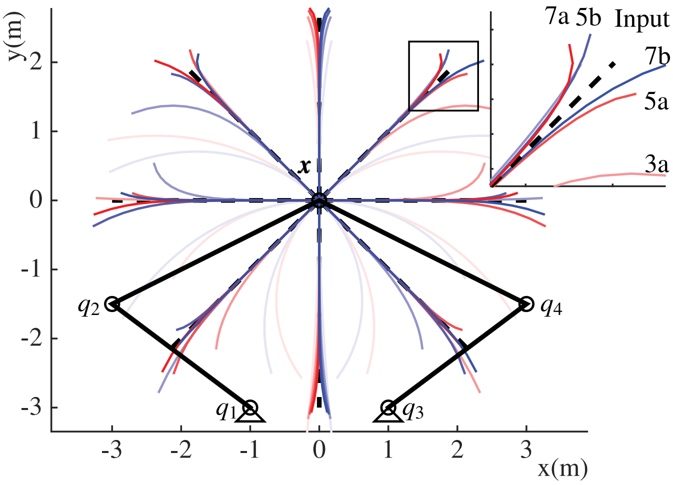

Fig. 2. The Taylor approximation of the IK of a 5-bar (solid gray) around evaluation point at $\boldsymbol{x}^0 = [0, 0]$ up to the seventh order for four different trajectories. It shows the left (a, red) and right (b, blue) estimation of end-effector trajectory (dashed black) for the order 1, 3, 5, and 7. The insert shows convergence for higher-order estimations far from the evaluation point.

b) with joints q_1, q_2, and q_3, q_4, respectively. The interconnection point is the end-effector $\boldsymbol{x}^0$. This interpoint has to satisfy the constraint equation from both sides (*a* and *b*) calculated using the local frame ($\boldsymbol{x}_a^0 = \boldsymbol{H}_2^0(\boldsymbol{q}_{1,2})\boldsymbol{x}^2$ and $\boldsymbol{x}_b^0 = \boldsymbol{H}_4^0(\boldsymbol{q}_{3,4})\boldsymbol{x}^4$). The closure equation can be written as

$$f : \mathbf{0} = \begin{bmatrix} \boldsymbol{x}^0 - \boldsymbol{x}_a^0 \\ \boldsymbol{x}^0 - \boldsymbol{x}_b^0 \end{bmatrix} \tag{33}$$

Using the end-effector coordinates ($\boldsymbol{u} = \boldsymbol{x}^0$) as input and the four joint angles ($\boldsymbol{v} = \left[q_1 \ \dots \ q_4\right]^\top$) as output, the first-order partial derivatives of the closure equation become

$$D_{\boldsymbol{u}}(f) = \boldsymbol{U} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix} \qquad D_{\boldsymbol{v}}(f) = \boldsymbol{V} = \begin{bmatrix} [\boldsymbol{s}_1^0 \times]\boldsymbol{x}_a^0 & [\boldsymbol{s}_2^0 \times]\boldsymbol{x}_a^0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\boldsymbol{s}_3^0 \times]\boldsymbol{x}_b^0 & [\boldsymbol{s}_4^0 \times]\boldsymbol{x}_b^0 \end{bmatrix}$$

The higher-order partial derivatives can be found by using the twist derivatives from Section 2.2 and recursive equations from Section 2.5.

The Taylor approximation, up to the seventh order, is computed for 200 positions of the end-effector ($\boldsymbol{x}^0$) forming four trajectories through the workspace with the aim of finding an approximation of the corresponding joint displacement of the joints ($q_1 \dots q_4$). For evaluation of the quality of the Taylor approximation, the end-effector position approximation from the left ($\boldsymbol{x}_a^0$) and right ($\boldsymbol{x}_b^0$) side are plotted together with input trajectories.

The result of the Taylor approximation (Fig. 2) shows that in a large portion of the workspace around the evaluation point ($\boldsymbol{x}^0 = \mathbf{0}$) the approximation converges to the target trajectory indicating a correct estimation of finite joint displacement. However, further from the evaluation point and closer to workspace boundary the accuracy is less as can be seen in the insert.

3.3. The Bennet linkage: Direct kinematics of an overconstrained linkage

An overconstrained linkage has a redundant set of loop closure constraints. That is, the number of dependent coordinates ($\boldsymbol{v}$) is smaller than the number of constraints. This poses a problem for proposed method – as apparent from Eq. (19) – since it requires inversion of the matrix $\boldsymbol{V}$, which was so far assumed to be square and non-singular. Among others, there are two ways to find the higher derivatives of the loop closure solution: (1) by selection of a subset of constraint conditions to make a square system (2) by replacing the inverse with a left pseudo inverse in Eq. (19). This can be done when the columns of $\boldsymbol{V}$ are independent, otherwise the mechanism is in a singular pose where its instantaneous

DOF increases. Using the left pseudo inverse, the solution is then

$$C_k = -(V^\top V)^{-1} V^\top F_k G_k = -V^+ F_k G_k \tag{34}$$

The corresponding constraint equation for this subset method will be indicated with f_s. A disadvantage of method 1 is that constraint equations can be selected sub-optimally, which may induce parameterization singularities, limiting the Taylor approximation.

An example of such an overconstrained mechanism is the Bennet linkage.[14] The Bennet linkage consists of a single spatial loop compromising four non-parallel revolute joints which do not intersect in a single point. According to the Chebychev–Grübler–Kutzbach's criterion such a spatial linkage with four bodies and four joints should have a mobility of -2. However, when specific kinematic conditions are satisfied the mechanism is mobile:[1,15]

1. Equality of the opposite link lengths: $l_1 = l_2$, $l_3 = l_4$
2. Equality of the opposite angles between joint axes: $\alpha_1 = \alpha_3$, $\alpha_2 = \alpha_4$
3. The Bennet condition $l_1/\sin(\alpha_1) = l_2/\sin(\alpha_2)$

The joint angles relate according to: $q_1 + q_3 = q_2 + q_4 = 2\pi$ with q_1 and q_2 as

$$\sin(\alpha_1/2 - \alpha_3/2)\tan(q_1/2)\tan(q_2/2) = \sin(\alpha_1/2 + \alpha_3/2) \tag{35}$$

To show the Taylor approximation, we assume no prior knowledge of joint angle relations, and we will write the closure equations in terms of transformation matrices. We do need the kinematic conditions to ensure full mobility. We use $\alpha_1 = 0.6$, $\alpha_2 = 1$, $l_1 = 1$, and $l_2 = 1.5$. The Bennet linkage is treated as two open chains joined together by a body, the first chain consisting of bodies 1 till 3 and a second chain consisting solely of body 4. Both bodies 1 and 4 are hinged to the base. For the loop closure, body 3 is considered rigidly attached to the fourth body. Joint 1 is treated as the independent coordinate ($\boldsymbol{u} = q_1$). There exist three dependent coordinates $\boldsymbol{v} = \begin{bmatrix} q_2 & q_3 & q_4 \end{bmatrix}^\top$. Together with the six constraints this makes the mechanism three times overconstrained. The effect of both strategies will be shown in and around a special configuration of the robot.

With the selection strategy only the angular constraints are taken into account. For the complete constraint also the translational constraints are used

$$f_s : \boldsymbol{R}_0^3(\boldsymbol{q}_{1-3})\boldsymbol{R}_4^0(q_4) = \text{const.} \qquad f_f : \boldsymbol{H}_0^3(\boldsymbol{q}_{1-3})\boldsymbol{H}_4^0(q_4) = \text{const.} \tag{36}$$

For the Bennet linkage, there exist two special configurations when the mechanism is fully collapsed onto a line. In these configurations, all the joint axes ($\boldsymbol{e}_i$) are perpendicular to the same line, as shown in Fig. 3. As f_s is selected to be consisting of the angular constraints, the system is in a parameterization singularity and the corresponding $\boldsymbol{V}_s = \begin{bmatrix} \boldsymbol{e}_2 & \boldsymbol{e}_3 & -\boldsymbol{e}_4 \end{bmatrix}$ matrix is singular. When the full set of constraints is taken into account $\boldsymbol{V}_f = \begin{bmatrix} \boldsymbol{s}_2 & \boldsymbol{s}_3 & -\boldsymbol{s}_4 \end{bmatrix}$ has a rank of 3 and its left pseudo inverse exists.

Also close to this special configuration the Taylor approximation using the selection strategy (f_s) on angular constraints suffers from this parameterization singularity as can be seen in Fig. 3. Here, a Taylor approximation up to the tenth order is made for both strategies. The evaluation pose is close (+0.4 deg of input joint) to the special configuration. It can be seen that the Taylor expansion using the selected constraints follows until roughly 6 deg. With the full set of constraints an approximation up to 30 deg could be obtained. This regression can be explained by numerical round off errors accumulating due to the ill-conditioned matrix inverse of $\boldsymbol{V}_s$.

It should be noted that although the Bennet linkage has no singularities, there is a radius of convergence around its evaluation point. This is due to the fact that the relation between the input and the output angle follows a *arctan* relation (35).

4. Discussion

In all the three examples, it can be seen that the Taylor approximation is confined to a region of convergence. One limiting cause is the existence of the singularities. Additionally – as seen in the Bennet linkage – the underlying closure solution can pose boundaries on the Taylor approximation.

Table I. The mean execution time in [sec] of Taylor expansion for the three examples. For the overconstrained Bennet linkage, the results of the selection (f_s) and full (f_f) method are given.

Order	Serial	5-bar	Bennet (f_s)	Bennet (f_f)
1	0.56×10^{-3}	49×10^{-6}	2.1×10^{-3}	2.1×10^{-3}
2	6.9×10^{-3}	1.5×10^{-3}	3.6×10^{-3}	3.4×10^{-3}
3	11×10^{-3}	3.1×10^{-3}	4.7×10^{-3}	3.4×10^{-3}
4	28×10^{-3}	11×10^{-3}	11×10^{-3}	11×10^{-3}
5	85×10^{-3}	47×10^{-3}	27×10^{-3}	19×10^{-3}
6	0.35	0.47	38×10^{-3}	48×10^{-3}
7	2.6	5.4	81×10^{-3}	85×10^{-3}

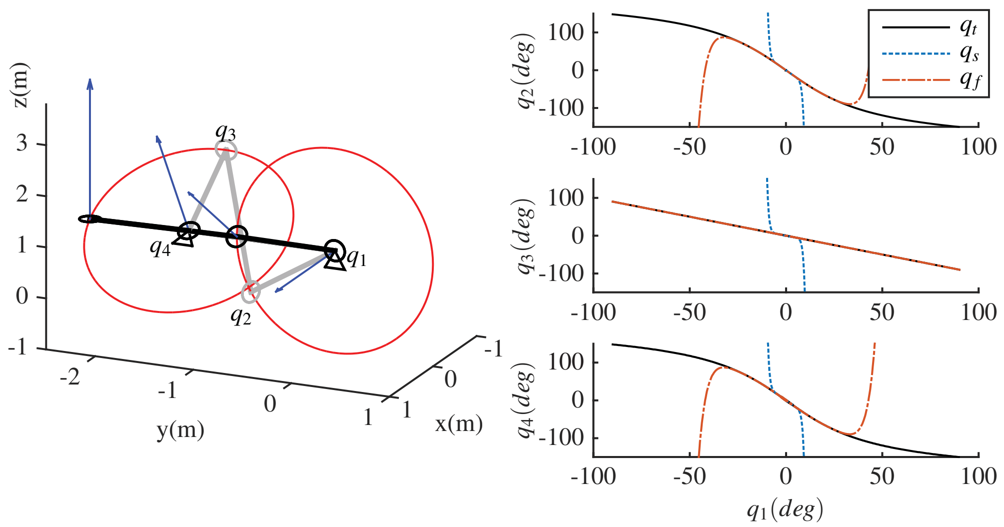

Fig. 3. Comparison between Taylor approximation (tenth order) of the Bennet linkage using two methods of constraining. (1) using a subset of constraints, and (2) the pseudo inverse of the full set. *Left* shows the Bennet linkage as used, in gray the reference pose and in black the evaluation pose, which is close to co-linearity. The blue arrows indicate the joint axes ($\boldsymbol{e}_i$). *Right* shows a comparison between the theoretical angles (q_t), and the approximation using selection (q_s) and pseudo inverse method (q_f) of the passive joint angles (q_2, q_3, q_4) as a function of the input angle q_1.

Therefore, in the case the closure solution is not known beforehand, one cannot discriminate between both causes of bounded convergence, based on a single Taylor approximation.

This method is restricted to lower kinematic pairs by the assumption that the screw vector associated with the joints is constant when expressed in the frames of the connecting bodies. For most practical applications, this is sufficient.

For the calculation of higher-order partial derivatives, the proposed method uses Kronecker products of matrices, which can lead to very large matrices for larger systems and higher orders, as can be seen in Table I. This possibly poses practical limits on applicability of this procedure. Sparse matrices and the aggregation of mixed partial derivatives can be used to mitigate computer memory usage and reduce the number of matrix operations. Also, as seen in the Bennet linkage example, an expansion close to singularity leads to ill-conditioned matrices reducing the numerical accuracy significantly.

The method presented here generates higher-order derivatives of motion by matrix multiplications without the need of taking derivatives analytically. This method allows to investigate the finite motion of open and closed loop linkages without the need to solve the closure equations while maintaining algebraic insight between the geometrical parameters. This opens up a potential of algebraic investigation and synthesis of motion of closed loop linkages.

https://doi.org/10.1017/S0263574718000462 Published online by Cambridge University Press

5. Conclusion

In this paper, a recursive method was presented that gives the higher-order partial derivatives of open and closed loop mechanisms with lower kinematic pairs. This method relies on a combination of a simplified representation of the higher-order twist derivatives, also presented here, and the matrix derivatives of Vetter. This enables the Taylor approximation of a kinematic mapping over a given trajectory and workspace, as exemplified by three examples. The method showed to be applicable to multi-DOF open and closed loop mechanisms and to overconstrained mechanisms, yielding an algebraic expression for the derivatives and the approximation of finite motion over a large portion of the workspace.

Acknowledgments

A. Müller acknowledges that this work has been partially supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics (LCM).

References

1. R. Bartkowiak and C. Woernle, "Necessary and sufficient mobility conditions for single-loop overconstrained nH mechanisms," *Mech. Mach. Theory* **103**, 65–84 (2016).
2. K. Wohlhart, "Degrees of shakiness," *Mech. Mach. Theory* **34**(7), 1103–1126 (1999).
3. K. Wohlhart, "From higher degrees of shakiness to mobility," *Mech. Mach. Theory* **45**, 467–476 (Mar. 2010).
4. C. Chen, "The order of local mobility of mechanisms," *Mech. Mach. Theory* **46**(9), 1251–1264 (2011).
5. J. Rico, J. Gallardo and J. Duffy, "Screw theory and higher order kinematic analysis of open serial and closed chains," *Mech. Mach. Theory* **34**(4), 559–586 (1999).
6. A. Müller, "Higher derivatives of the kinematic mapping and some applications," *Mech. Mach. Theory* **76**, 70–85 (Jun. 2014).
7. A. Müller, "Recursive higher-order constraints for linkages with lower kinematic pairs," *Mech. Mach. Theory* **100**, 33–43 (Jun. 2016).
8. V. Van der Wijk and J. L. Herder, "Synthesis method for linkages with center of mass at invariant link point Pantograph based mechanisms," *Mech. Mach. Theory* **48**, 15–28 (Oct. 2011).
9. T. W. Ma, "Higher chain formula proved by combinatorics," *Electron. J. Comb.* **16**(1), 1–7 (2009).
10. R. W. Brockett, "Robotic manipulators and the product of exponentials formula," *Math. Theory Netw. Syst. SE - 9* **58**, 120–129 (1984).
11. W. J. Vetter, "Matrix calculus operations and Taylor expansions," *Soc. Ind. Appl. Math.* **15**(2), 352–369 (1973).
12. J. J. de Jong, A. Müller, J. van Dijk and J. L. Herder, "Differentiation-Free Taylor Approximation of Finite Motion in Closed Loop Kinematics," **In**: *Computational Kinematics. Mechanisms and Machine Science* (S. Zeghloul, L. Romdhane and M. Laribi, eds.), vol. 50 (2018) pp. 577–584.
13. M. L. Husty, M. Pfurner and H. P. Schröcker, "A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator," *Mech. Mach. Theory* **42**(1), 66–81 (2007).
14. G. Bennet, "A new mechanism," *Engineering* **76**(12), 777–778 (1903).
15. J. E. Baker, "The Bennett, Goldberg and Myard linkages-in perspective," *Mech. Mach. Theory* **14**(4), 239–253 (1979).

Appendix: A selection of Kroncker product identities

Consider the following set of matrices: $\boldsymbol{A} \in \mathbb{R}^{n\times m}, \boldsymbol{B} \in \mathbb{R}^{p\times q}$. The Kronecker product is defined as the collection of the element-wise multiplication of all elements in the respective matrices

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1m}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\boldsymbol{B} & \dots & a_{nm}\boldsymbol{B} \end{bmatrix} \tag{A1}$$

The mixed-product property is used to combine Kronecker products

$$\boldsymbol{AB} \otimes \boldsymbol{CD} = (\boldsymbol{A} \otimes \boldsymbol{C})(\boldsymbol{B} \otimes \boldsymbol{D}) \tag{A2}$$

The sequence of the Kronecker product can be swapped with pre- and post- multiplication of permutation matrices

$$\boldsymbol{A} \otimes \boldsymbol{B} = \boldsymbol{P}_{p,n}\boldsymbol{B} \otimes \boldsymbol{A}\boldsymbol{P}_{m,q} \tag{A3}$$

These permutation matrices are binary, orthogonal, square nm $\times$ nm matrices consisting of $m \times n$ submatrices. These submatrices ($\boldsymbol{P}_{n,m}^{i,j} \in \mathbb{N}^{m\times n}$) have only one 1 on a specific location ($\boldsymbol{P}_{n,m}^{i,j}(j, i) = 1$).