

Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems

Complexity analysis for calculating the Jacobian matrix of 6DOF reconfigurable machines

Monika Z. Filiposka^{a*}, Ana M. Djuric^b, Waguih ElMaraghy^a

^a University of Windsor, IMSE Department, 401 Sunset Avenue, Windsor, ON, N9B 3P4, Canada

^b Wayne State University, Engineering Technology Division, Fourth St. Detroit, MI 48202, USA

* Corresponding author. Tel.: +1-519-253-3000; +1-519-992-4565. E-mail address: filipos@uwindsor.ca

Abstract

The Jacobian matrix for machinery systems is challenging due to the kinematic structure, the machine behaviour, the machine configurations, and the singularity conditions. In the area of singularity, small velocities in the operational space may result in large velocities in the joint space. Therefore the control processes of machines may sometimes run into difficulties, since the inverse mapping from a Cartesian space to a joint space may cause problems. For solving the singularity of the kinematic structure, the development of the Jacobian matrix is the first priority. All of the different methods used for the Jacobian calculation are complex and require mathematical tools for symbolic calculation. It is important to select the most appropriate method with the minimum complexity for quick and accurate Jacobian matrix calculation. Thus, the authors were motivated to choose different procedures for developing the Jacobian matrix. In this study, the most general reconfigurable machinery kinematic structure, a remodelled n-DOF Global Kinematic Model, CNC-R GKM, is selected for the Jacobian matrix computation analysis. Each joint represents a combination of either rotational and/or translational joints type with any joint positive direction, with total number of 36 possible configurations in one joint. Calculation of the Jacobian matrix for this highly complex reconfigurable kinematic model, gives a comprehensive and unique results. The selected methods for Jacobian calculation are recursive Newton-Euler method and Vector cross multiplication method. The symbolic mathematical tool MAPLE 16 has been used. The comparison between these two calculation procedures has been completed with respect to the calculation complexity, and the results have been validated using the new developed CNC-Robot Global Kinematic Model (CNC-R GKM). The kinematic model and Jacobian matrix calculation used in this study are applicable in reconfigurable robotic and machines in industry, space and health care.

© 2014 The Authors. Published by Elsevier B.V.

Selection and peer-review under responsibility of the International Scientific Committee of “The 47th CIRP Conference on Manufacturing Systems” in the person of the Conference Chair Professor Hoda ElMaraghy.

Keywords: Jacobian matrix; singularities; reconfigurable models; Newton-Euler; Vector method; complexity; robots.

1. Introduction

In today's enormous changing world, we struggle with the need to adapt to the changes quickly. Industries and manufacturers are dealing with the variety and diversity of demand. Customers are more and more insisting on unique products, which will satisfy their needs. For this to be accomplished, industries have to implement and employ reconfigurable systems, which will respond adequate and will

be sustainable in today's inventive world [1].

Modelling a robotic system is a necessary premise to find appropriate control approaches. The kinematic of any mechanical system is primarily important. These systems, like industrial robots and CNC or rapid prototyping machines, usually have a problem with large displacements, and that is why they are experiencing large variations in geometric configuration when they operate even under normal conditions. The Jacobian matrix of any machine or

manipulator plays a key role in solving inverse kinematics problem, but unfortunately, it loses its full rank at singularities. Although there are many algorithms for solving Jacobian matrix, still problems arise when a singular configuration is encountered on the way to a prescribed location in the task space. Newton algorithms are commonly used to solve Jacobian matrix for non-redundant robots. The only effective approach to handle the singularities is to keep a current configuration far away from singular configurations. In recent years operating speeds have been increased, and consequently, there has been an increase in accelerations and inertia forces. In theory, robot manipulators are flexible and can be reprogrammed for new tasks, but each robot's configuration makes it capable for only a limited number of applications.

An n-DOF Global Kinematic Model (n-GKM) was previously developed [5] for any combination of either rotational or translational type of joints and it represents n-DOF kinematic structure with either rotational or translational joints. This unified approach, in principle, can be extended to manipulator's architecture by including the design parameters. In this study a remodelled 6-DOF Global Kinematic Model (CNC-R GKM) is selected for the Jacobian matrix computation analysis. In this case, instead of calculating Jacobian for every robot and solving singularities separately, a reconfigurable Jacobian matrix has been developed. This calculation depends on the complexity of the multibody system, and iterative methods will have to be employed in the most difficult cases. The selected methods for Jacobian calculation are recursive Newton-Euler method and Vector method. The emphasis is on the significance of DH parameters for designing and building reconfigurable multibody system for industry. The solutions can be applied to any industrial robot, CNC or rapid prototyping machine, for calculating singular condition, consequently optimizing the same, and granting sustainable solution for the machinery control.

0R_6 - Rotational matrix
$({}^0R_i)^T$ - Rotational transpose matrix
${}^{i-1}P_i$ - Position matrix
Z_{i-1} - Unit vector
J - Jacobian matrix
\dot{X} - End effector velocity
\dot{q} - Joint Velocity
m - number of independent equations
n - number of links, number of joint variables

1.1. Literature survey

The complexity in solving kinematics using direct computation is arising from the non-linearity of forward kinematics. Upon solving the kinematics for a system, the Jacobian matrix is the next priority, for determining singular configurations [2]. The Jacobian matrix calculation based methods are the most truthful, because they are giving the transformation between velocities in the workspace and the kinematic structure. Fast solutions can be obtained, due to the complexity for defining singularity configurations of reconfigurable structures. Different authors are offering unified approaches for Jacobian matrix calculation, but all of the methods differ in complexity and accuracy. In [5, 6] the authors developed the novel n-DOF Global Kinematic Model (GKM) for any combination of joints. The structure belongs to any of the three surfaces or eight sub surfaces. The total number of supported structures is $48(48-1)^{n-1}$; for a 6DOF system, equals to 11,008,560,336 possible kinematic structures. The model is validated with 2DOF RR, RT, TR and TT structures. The same authors developed Reconfigurable Puma-Fanuc kinematic model [7]. The Jacobian matrices are calculated for all eight possible configurations of a 3 DOF kinematic structures and the comparison has been done according to the structures complexity in [4]. The proposed catalogue can be used as a design tool and for the validation of the reconfigurable based solutions. With complex configurations, the difficulty in calculation rises, thus beside the use of Maple Software, in most cases manual simplification is needed. Djuric et al [8] explained Newton-Euler Recursive method for Jacobian matrix calculation, also graphical approach for singularity analysis is included. Validation is conducted on Fanuc family robots, but this work can be extended for CNC-R GKM Jacobian matrix and singularity analysis, further to be incorporated in any robot configuration. In [9] several Newton based algorithms for inverse kinematics are presented and compared. All of them include a Jacobian derivation as compulsory. An approximation to the Jacobian matrix is not recommended by the authors, because of its weight in further recursive computation and trajectory tracking. Duleba and Sasiadek [10] present modified Jacobian method of transversal passing through singularity configurations for both redundant and non-redundant manipulators, but the Jacobian in this case is extended, although it still has square form. The method is not applicable for hyperbolic singularities, only for common quadratic singularities. It enables passing through singularities with smooth velocities, without stoppages or jerking. The method is evaluated on 3DOF manipulators.

Nomenclature

CNC-R GKM - CNC- Robot global kinematic model
D-H - Denavit-Hartenberg parameters
DOF- Degrees of freedom
${}^{i-1}A_i$ - Homogeneous matrices
d_i - Link offset along previous Z to the common normal
θ_i - Joint angle about Z, from old X to new X
a_i - Link length of the common normal
α_i - Twist angle about common normal, from old Z axis to new Z axis
R_i - Joints types control parameters
T_i - Joints types control parameters
K_{Si} - Twist angles sinus control parameter
K_{Ci} - Twist angles cosines control parameter
${}^{i-1}\dot{\theta}_i$ - Angular velocity vector
${}^{i-1}\dot{P}_i$ - Linear velocity vector
${}^i({}^0\omega_i)$ - Angular velocity
${}^i({}^0V_i)$ - Linear velocity

Abele et al [11] is describing a method for calculating Cartesian stiffness based on Jacobian matrix. Even this method allows calculating realistic values for position and orientation in the entire Cartesian workspace, it is only for industrial robots with three degrees of freedom, and it can increase complexity when the workspace is bigger. Kircanski [12] describes simple Stanford manipulator inverse kinematic solution, where Jacobian is decomposed in four sub matrices [3x3], similar like in the Newton-Euler method, which reduces computation complexity 5-6 times. The problem of Jacobian calculation can be managed with designing the manipulators in order to have closed-form of solutions, but that leads to significant limitations in their applications. If no close-form solutions are available, approximations are conducted. Lenarcic and Kosutnik [13] is introducing concept for Jacobian and inverse kinematics approximation, applicable in arc welding robots, and validated on 6DOF. The method has proven accuracy, but in specific cases may introduce unsolvable problems in real-time computation. In [14] control equations based on manipulator Jacobian matrix are presented, which highlights its importance in modelling, kinematics, dynamics and control of robotic multibodies. Also [15] is raising the importance of Jacobian matrix in dynamics, thus unified generation of sparse Jacobian matrix for complex multibody systems is obtained. Another approach is developed with rearranging the Jacobian matrix [16, 17], based on screw theory. The manipulators (Puma type of robots) are always in singular configuration, but still can track a trajectory. Methods like in [18, 19] are offering reduced complexity in computation of the inverse Jacobian matrices. A classification of singularity configurations is offered in [20], where the chosen model for validation is Kuka KR 15/2. After a comparison between Jacobian method and null space approach for trajectory tracking, the results are showing that Jacobian method is simpler. In all reviewed papers the Denavit–Hartenberg (D-H) parameters [3] are used for the kinematic modelling.

2. Development of CNC-R GKM

In serial link manipulators there are series of links, connecting the end-effector to the base, by actuated joints. The homogeneous transformation matrix ${}^{i-1}A_i$, in the n-DOF GKM [5] is giving the relationship between two joints in Cartesian coordinate frame, and i represents the number of joints (1, 2,...6).

$${}^{i-1}A_i = \begin{bmatrix} \cos(R_i\theta_i + T_i\theta_{DHi}) & -K_{Ci} \sin(R_i\theta_i + T_i\theta_{DHi}) & K_{Si} \sin(R_i\theta_i + T_i\theta_{DHi}) & a_i \cos(R_i\theta_i + T_i\theta_{DHi}) \\ \sin(R_i\theta_i + T_i\theta_{DHi}) & K_{Ci} \cos(R_i\theta_i + T_i\theta_{DHi}) & -K_{Si} \cos(R_i\theta_i + T_i\theta_{DHi}) & a_i \sin(R_i\theta_i + T_i\theta_{DHi}) \\ 0 & K_{Si} & K_{Ci} & R_i d_{DHi} + T_i d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$i = 1,2,3,4,5,6$

Because each joint has six different positive directions for rotations or translations, any joint’s vector can be in positive or negative direction in Cartesian space. Thus, this model requires development of its own Jacobian matrix.

Rotational Joints: $R_i = 1, T_i = 0$ (2)

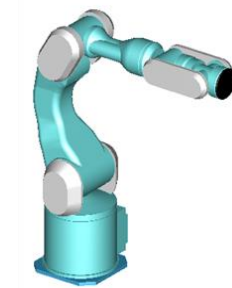
Translational Joints: $R_i = 0, T_i = 1$ (3)

R_i and T_i are used to control the selection of joint type (rotational and/or translational). Their sinus and cosines are defined as the joint’s reconfigurable parameters and expressed in equations (4) and (5).

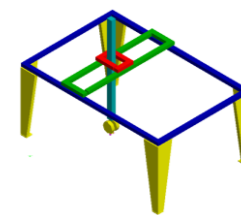
$$K_{Si} = \sin \alpha_i \quad (4)$$

$$K_{Ci} = \cos \alpha_i \quad (5)$$

The need of being able to combine any robot manipulator and any CNC machine D-H parameters (Fig 1) has resulted in development of CNC-R Global Kinematic Model, graphically presented in Fig 2. In the literature, there is no evidence of kinematic structure that unifies different joint types (rotational and translational), and their Jacobian matrix solutions. Combined joint types increase the model complexity, but provide the knowledge of many machines kinematic, which can be used as a design tool for new machine kinematic structure, and much more.



a)



b)

Table a) Robot manipulator D-H parameters

Mitsubishi PA10-6C D-H Parameters				
Joint	Theta	D	A	Alpha
1	0	317	0	-90
2	-90	0	450	0
3	90	0	0	90
4	0	480	0	-90
5	0	0	0	90
6	0	70	0	0

Table b) CNC machine D-H parameters

CNC machine D-H Parameters				
Joint	Theta	D	A	Alpha
1	90	4715.22	0	90
2	90	2714.59	0	90
3	0	0	0	0
4	180	-2664.68	0	-90
5	0	0	0	90
6	0	-281.25	0	0

Fig. 1. Combination of CNC machine b) and robot manipulator a) kinematic structures

The CNC-R GKM has possible kinematic configurations, and each configuration can be modelled with one set of reconfigurable parameters, presented in Table 1 and 2. Compared to the initial 6-DOF GKM, which can have $48(48-1)^{n-1}$ configurations, this kinematic model cannot encompass all 11,008,560,336 structures, but includes 446,071,500 kinematic configurations.

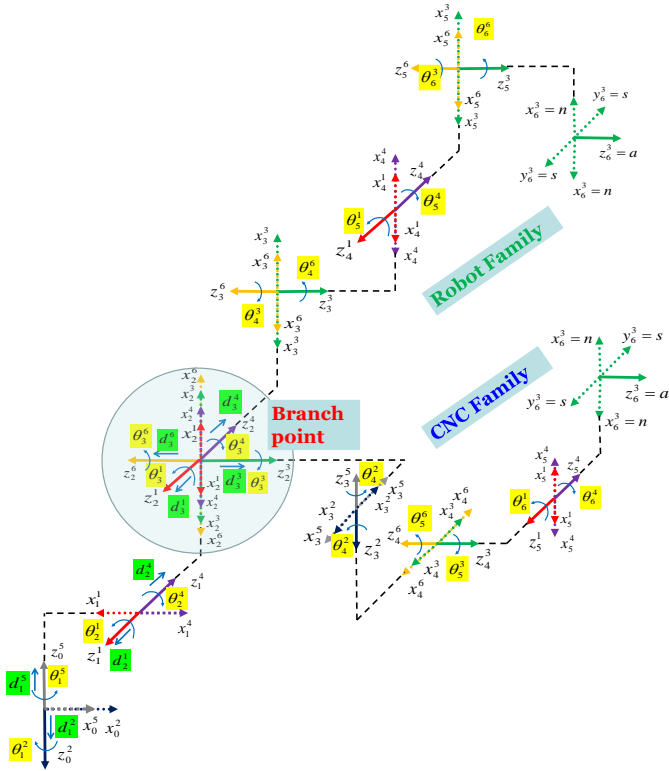


Fig. 2. CNC-R Global Kinematic Model

The link twist angle α_i , in the full reconfigurable kinematic model, usually has five different values, and remains perpendicularity to the joint's coordinate frames; however for the current CNC-R GKM the D-H parameters are presented in Table 1.

Table 1. DH parameters for 6-DOF CNC-R GKM

i	θ_i	d_i	a_i	α_i
1	$R_1\theta_1 + T_1\theta_{DH1}$	$R_1d_{DH1} + T_1d_1$	a_1	$\pm 90^\circ$
2	$R_2\theta_2 + T_2\theta_{DH2}$	$R_2d_{DH2} + T_2d_2$	a_2	$\pm 180^\circ; \pm 90^\circ; 0$
3	$R_3\theta_3 + T_3\theta_{DH3}$	$R_3d_{DH3} + T_3d_3$	a_3	$\pm 180^\circ; \pm 90^\circ; 0$
4	θ_{DH4}	d_{DH4}	a_4	$\pm 90^\circ$
5	θ_{DH5}	d_{DH5}	a_5	$\pm 90^\circ$
6	θ_{DH6}	d_{DH6}	a_6	$\pm 90^\circ; 0^\circ$

The control values for CNC-R reconfigurable Global Kinematic Model parameters are presented in Table 2.

Table 2. CNC-R GKM reconfigurable parameters

Control Values					
Joint	Sine	Cosine	Joint	Sine	Cosine
1	$K_{S1} = \pm 1$	$K_{C1} = 0$	4	$K_{S4} = \pm 1$	$K_{C4} = 0$
2	K_{S2}	K_{C2}	5	$K_{S5} = \pm 1$	$K_{C5} = 0$
3	K_{S3}	K_{C3}	6	$K_{S6} = \pm 1$	K_{C6}

Depending on the purpose of the system, some unnecessary postures should not be encountered, thus last tree joints are considered as rotational only, but first three either translational or rotational.

α_i has five possible values only at second and third joint. This condition produce joint 3 the most complex joint in the model, which is named Branch Point. From joint 3 two branches are formed, one for group of robots named Robot Family Branch, and one for group of CNC machines named CNC Family Branch. See Fig 2. This issue results in the increase of the complexity in the system. It is extremely difficult to simplify or decouple the results in order to get optimal solutions.

3. CNC-R GKM Jacobian Matrix

Differential kinematics describes the analytical relationship between the joint motion and the end-effector motion in terms of velocities, through the manipulator Jacobian matrix $J_{(m \times n)}$. This Jacobian matrix J , also determines the relationship between end-effector velocities \dot{X} and joint velocities \dot{q} (equation 6).

$$J = \frac{\partial f}{\partial q}; \quad \dot{X} = \frac{dx}{dt}; \quad \dot{q} = \frac{dq}{dt} \quad (6)$$

Results derived from Table 1. and Table 2. are important for the development of the model's Jacobian matrix, in particular for its simplification. Besides nonlinearity in the position and orientation equations, the relationship between the joint velocity and the distal end is linear (equations 7).

$$\dot{X} = J_{(m \times n)} \dot{q} \quad (7)$$

Equation (7) can be interpreted as a linear mapping from an m -dimensional vector space X , to an n -dimensional vector space q . In this case n is the number of joints, and equals to 6, also m equals to 6, and it represents the dimension of the end-effector vector X , which determines the size of the matrix, $J_{(6 \times 6)}$.

3.1. Newton-Euler recursive method

The recursive Newton-Euler (RNE) formulation can be written in terms of the Denavit-Hartenberg parameter [21], which performs efficient algorithm computations. This method requires angular and linear velocity vectors computation. In the CNC-R GKM these vectors are defined like in equations (8) for angular and (9) for linear velocity. The first tree joints can be either rotational either translational, but the last three are rotational, so the linear velocity vector for joints 4, 5 and 6 will be zero vector matrix.

$${}^{i-1}\dot{\theta}_i = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_i \end{bmatrix} \quad i = 1,2,3,4,5,6 \quad (8)$$

$${}^{i-1}\dot{P}_i = \begin{bmatrix} 0 \\ 0 \\ \dot{d}_i \end{bmatrix} \quad i = 1,2,3 \quad {}^{i-1}\dot{P}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad i = 4,5,6 \quad (9)$$

The original RNE equations for linear and angular velocities with respect to the matching joint needs to be adapted for this reconfigurable CNC-R GKM kinematic model. Angular and linear vector components need to be

added in the calculations, because of the complexity, also for accuracy, depending on the type of joints (equations 10, 11).

$${}^i({}^0\omega_i) = {}^iR_{i-1} [{}^{i-1}({}^0\omega_{i-1}) + R_i({}^{i-1}\dot{\theta}_i)] \quad i = 1, 2, \dots, 6 \quad (10)$$

$${}^i({}^0V_i) = {}^iR_{i-1} [{}^{i-1}({}^0V_{i-1}) + R_i \{ i({}^0\omega_i) \times {}^iR_{i-1} {}^{i-1}P_i \} + R_i \{ {}^iR_{i-1} [{}^{i-1}\dot{P}_i + {}^{i-1}({}^0\omega_{i-1}) \times {}^{i-1}P_i] \}] \quad i = 1, 2, \dots, 6 \quad (11)$$

Rotational matrices in these calculations, ${}^iR_{i-1}$ are defined by the upper left [3x3] sub matrices from the homogenous transformation matrices ${}^{i-1}A_i$ (equation 1), correlated with each joint and equation (12) is presenting rotational transpose matrix.

$${}^iR_{i-1} = ({}^0R_i)^T = \begin{bmatrix} \cos(R_i\theta_i + T_i\theta_{DHi}) & \sin(R_i\theta_i + T_i\theta_{DHi}) & 0 \\ -K_{Gi} \sin(R_i\theta_i + T_i\theta_{DHi}) & K_{Gi} \cos(R_i\theta_i + T_i\theta_{DHi}) & K_{Si} \\ K_{Si} \sin(R_i\theta_i + T_i\theta_{DHi}) & -K_{Si} \cos(R_i\theta_i + T_i\theta_{DHi}) & K_{Gi} \end{bmatrix} \quad (12)$$

The position vectors ${}^{i-1}P_i$, are defined by the upper right [3x1] sub matrices from the homogenous transformation matrices ${}^{i-1}A_i$ (equation 1), correlated with each joint. See equation (13).

$${}^{i-1}P_i = \begin{bmatrix} a_i \cos(R_i\theta_i + T_i\theta_{DHi}) \\ a_i \sin(R_i\theta_i + T_i\theta_{DHi}) \\ R_i d_{DHi} + T_i d_i \end{bmatrix} \quad (13)$$

The Jacobian matrix calculated using the Newton-Euler method is relative to the end-effector frame. In order to get the Jacobian relative to the base frame, this matrix needs to be multiplied with the rotational matrix 0R_6 .

3.2. Vector cross multiplication method

The Vector method [22, 23] offers simpler results, especially if only kinematics computation is required. The method does not require velocities computation (does not involve differentiation too), and it is based on link transformation matrices found in forward (direct) kinematics (equations 14, 15).

$$J_i(q) = \begin{bmatrix} [Z_{i-1} X^{i-1} P_n] \\ Z_{i-1} \\ [Z_{i-1}] \\ 0 \end{bmatrix} \quad i = 1, 2, \dots, 6; n = 6 \quad (14)$$

$$J = \begin{bmatrix} z_0 \times {}^0p_n & z_1 \times ({}^0p_n - {}^0p_1) & z_2 \times ({}^0p_n - {}^0p_2) & \dots & z_{n-1} \times ({}^0p_n - {}^0p_{n-1}) \\ z_0 & z_1 & z_2 & \dots & z_{n-1} \end{bmatrix} \quad (15)$$

Although, Z_{i-1} unit (generating) vectors computation is needed (equation 16), along the motion of the joints expressed in base frame coordinates, also pose matrix P_n , computation is required (equation 17).

$$Z_i = R_i Z_0 \quad (16)$$

$$P_n = R_1 (R_2 (\dots (R_{n-2} (R_{n-1} P_n + P_{n-1}) P_{n-1} + \dots) + P_2) + P_1) \quad i = 1, 2, \dots, 6; n = 6 \quad (17)$$

The Jacobian matrix derived using the Vector cross multiplication method is relative to the base frame.

4. Comparison of the Jacobian matrices derived by the two methods

The Jacobian matrices derived by the use of the both methods are dependant of the rotational and translational components, respectively. Because the CNC-R model is very complex 6DOF kinematic structure, the results are complex, also. Moreover manual decoupling is always necessary, especially in Jacobian matrix calculations in order to simplify the results derived using both, Newton-Euler and Vector methods.

The derived results in equations (18) and (19) are presenting unified solution for 6DOF CNC-R GKM; also, they clearly show the dependence of the joints, from rotational and translational components separately.

Equation (18) presents the Jacobian matrix J_{NE} calculated using the Newton-Euler method. Equation (19) presents the Jacobian matrix J_V calculated using the Vector method.

$$J_{NE} = \begin{bmatrix} J_{11} & J_{12} & J_{12} & J_{14} & J_{15} & 0 \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & 0 \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & 0 \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & 0 \\ J_{51} & J_{52} & J_{53} & J_{54} & 0 & J_{56} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & 0 \end{bmatrix} \begin{bmatrix} R_1 \dot{\theta}_1 \\ R_2 \dot{\theta}_2 \\ R_3 \dot{\theta}_3 \\ R_4 \dot{\theta}_4 \\ R_5 \dot{\theta}_5 \\ R_6 \dot{\theta}_6 \end{bmatrix} + \quad (18)$$

$$+ \begin{bmatrix} J_{11} & J_{12} & J_{13} & 0 & 0 & 0 \\ J_{21} & J_{22} & J_{23} & 0 & 0 & 0 \\ J_{31} & J_{32} & J_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_1 d_1 \\ T_2 d_2 \\ T_3 d_3 \\ T_4 d_4 \\ T_5 d_5 \\ T_6 d_6 \end{bmatrix}$$

$$J_V = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ 0 & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \\ 0 & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} \\ 0 & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} \\ J_{61} & 0 & J_{63} & J_{64} & J_{65} & J_{66} \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \end{bmatrix} + \quad (19)$$

$$+ \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ 0 & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \\ 0 & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} \\ 0 & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} \\ J_{61} & 0 & J_{63} & J_{64} & J_{65} & J_{66} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

The 6DOF CNC-R GKM kinematic model joints are represented using generalized coordinates (equation 20).

$$\dot{q}_i = R_i \dot{\theta}_i \text{ or / and } T_i d_i \quad (20)$$

By observing these two equations it is clear that they are different. These two methods are using two different reference frames in the calculation procedure, which generate their

differences. The Jacobian matrix calculated using the Newton-Euler method is relative to the end-effector frame, while the Jacobian matrix derived using the Vector cross multiplication method is relative to the base frame.

The relation between these two Jacobians is expressed in equation (21). 0R_6 is the rotational matrix for the observed system between the base and end-effector frames.

$$J_V = {}^0R_6 J_{NE} \quad (21)$$

Conclusion

With using the two methods, Newton-Euler and Vector method, a unified CNC-R GKM Jacobian matrix is derived. From Maple 16 calculations it is easy to spot that the calculation time is almost 2,5 times greater in Newton-Euler, than in Vector method, disregarding the manual simplification.

If only kinematic solutions are needed, it is concluded that for calculation of Jacobian matrix, the Vector method is less complex. It does not require computation of linear and angular velocities and it directly gives the Jacobian matrix relative to the base frame. In contrary, the Newton-Euler method gives the Jacobian relative to the end-effector frame, and in order to obtain it in base frame coordinates, additional computation of appropriate rotational transpose matrix is required.

However, if the problem requires dynamic computations, linear and angular velocities are mandatory, hence it is recommended using the Newton-Euler method, ignoring the complexity. The authors recommend using Vector method in Maple 16 calculation, and they find it more appropriate for calculating Jacobian matrix of complex reconfigurable multibody machinery systems. See Table 3.

Table 3. Comparison between NE and Vector Method for Jacobian calculation

	Newton-Euler Method	Vector Method
Consumed memory	260.5M	90.67M
Calculation time	56.27s	22.26s
Size of equations	Big	Medium
Manual simplifications	High	Low
Overall complexity	High	Medium
Recommendation for Jacobian calculation	Recommended only if there is need for toques/forces calculation	recommended

The next step is to find singular conditions for this complex model, and to use that result as a design tool for optimal selection of the kinematic structure related to the desired applicant and layout design conditions.

References

[1] ElMaraghy HA. Changeable and reconfigurable manufacturing systems : Springer; 2009.
 [2] Lenarčič J, Husty ML. Latest advances in robot kinematics : Springer; 2012.

[3] Denavit J, Hartenberg RS. A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME J. Appl. Mech* 23 1955;23(215):22.
 [4] Chandrasekaran K, Djuric A, ElMaraghy W. Selection Catalogue of Kinematic Configuration for Pick and Place Application. *Enabling Manufacturing Competitiveness and Economic Sustainability*: Springer; 2012. p. 41-46.
 [5] Djuric A, ElMaraghy W. Generalized reconfigurable 6-joint robot modeling. *Transactions of the Canadian Society for Mechanical Engineering* 2006;30(4):533-565.
 [6] Djuric AM, Al Saidi R, ElMaraghy W. Global Kinematic Model generation for n-DOF reconfigurable machinery structure. 2010 IEEE Conference on Automation Science and Engineering (CASE); 2010.
 [7] Djuric AM, ElMaraghy W. A unified reconfigurable robots jacobian. *Proc. of the 2nd Int. Conf. on Changeable, Agile, Reconfigurable and Virtual Production*, 2007;811-823
 [8] Djuric A, Filipovic M, Kevac L, Urbanic J. Singularity Analysis for a 6 DOF Family of Robots. *Enabling Manufacturing Competitiveness and Economic Sustainability*: Springer; 2014. p. 201-206.
 [9] Duleba I, Opalka M. A comparison of jacobian-based methods of inverse kinematics for serial robot manipulators. *Int.J.Appl.Math. Comput.Sci* 2013;23(2):373-382..
 [10] Duleba I, Siasidek JZ. Modified Jacobian method of transversal passing through the smallest deficiency singularities for robot manipulators. *Robotica* 2002;20(04):405-415.
 [11] Abele E, Rothenbücher S, Weigold M. Cartesian compliance model for industrial robots using virtual joints. *Production Engineering* ;2008;2(3):339-343.
 [12] Kircanski MV. Inverse kinematic problem near singularities for simple manipulators: Symbolical damped least-squares solution.. *Proceedings, IEEE International Conference on Robotics and Automation*; 1993.
 [13] Lenarčič, Jadran, and Andreja Košutnik. Approximate calculation of robot inverse kinematics applied to arc welding. *Experimental Robotics I*: Springer; 1990.
 [14] Schwartz EM, Manseur R , Doty KL. Non-commensurate Manipulator Jacobian. *Robotics and Applications IASTED International Conference*; 2003.
 [15] Kecskemethy A. Sparse-matrix generation of jacobians for the object-oriented modeling of multibody systems. *Nonlinear Dynamics* 9 (1-2) ;1996, pp.185-204
 [16] Hu Z, Hairong F, Yuefa F. New solution algorithm for singularity control of serial manipulators. In *TENCON'02. Proceedings IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, vol. 3;2002, pp. 1554-1557.
 [17] Hu Z, Fu Z, Fang H. Study of singularity robust inverse of Jacobian matrix for manipulator. In *Proceedings International Conference on Machine Learning and Cybernetics*, vol. 1; 2002, pp. 406-410
 [18] Meldrum DR, Rodriguez G, Franklin GF. Efficient Control with an Order (n) Recursive Inversion of the Jacobian for an n-Link Serial Manipulator. In *IEEE American Control Conference*;1991, pp. 2039-2044
 [19] Wu C, Kuu-Young Y. An efficient solution of a differential inverse kinematics problem for wrist-partitioned robots. *IEEE Transactions Robotics and Automation*, no. 1;1990, pp 117-123.
 [20] Gracia L, Andres J,Tornero J. Trajectory tracking with a 6R serial industrial robot with ordinary and non-ordinary singularities. *International Journal of Control, Automation and Systems* 7, no. 1;1990, pp 85-96
 [21] Spong M, Vidyasagar M. *Robot Dynamics and Control*. J. Wiley and Son, New York;1989.
 [22] Fu KS, Gonzalez RC, Lee CSG. *Robotics: control, sensing, vision, and intelligence*", McGraw-Hill Inc.;1987, pp.544-547
 [23] Orin D, Schrader W. Efficient Computation of the Jacobian for Robot Manipulators. *The First International Symposium in Robotics Research*: MIT Press, Cambridge, MA;1984, pp. 727-734