Theoretical Concept of Inverse Kinematic Models to Determine Valid Work Areas Using Target Coordinates from NC-Programs

A Model Comparison to Extend a System-in-Progress as Systems Engineering Task

Jens Weber(✉)

Heinz Nixdorf Institute, Fuerstenallee 11, Paderborn, Germany
jens.weber@hni.upb.de

Abstract. In order to determine valid production results in the area of NC-based tooling machine, complex simulation tools are used. The challenge is the calculation of the material removal, tool paths as well as valid setup positions of workpiece and periphery which leads mostly to high computational time. The descent of the computational effort includes a high portion of systems engineering. This contribution shows a theoretical concept to substitute complex simulation models by calculation models which observes the inverse kinematic behavior of the machine in combination with a NC-parser which estimates valid workpiece positions. The contribution compares model approaches of inverse kinematic problems considering 5 axis tooling machine to determine valid setup positions and minimize theoretical calculation effort.

Keywords: Inverse kinematic · NC-program · NC-command · Quaternion · Translation vector · Rotation matrix

1 Introduction

In order to realize a simulation-based optimization process to determine valid setup position for workpieces during tooling operations, the idea arose to combine meta-heuristics as optimization component and a simulation model of a tooling machine as evaluation component. The simulation model, which is a CAD-based machine model which is controlled by a real control unit leads to high evaluation effort when each potential solution candidates includes a single simulation. The problem of this approach is to shrink the evaluation process in order to reduce the number of single simulation runs of production processes.

The NC-program offers NC-commands which contains the target coordinates to reach the target geometry of the raw material (workpiece). A given software program, so called NC-parser, identifies the tool paths and calculate the trajectories of the program cycles and return the production time depending on zero-point of workpiece-position coordinates. The combination of the NC-parser as fitness component and PSO algorithm as optimization component is a rapid combination to reach a huge number of

J.J. Jung and P. Kim (Eds.): BDTA 2016, LNICST 194, pp. 91–101, 2017.
DOI: 10.1007/978-3-319-58967-1_11

potential solution candidates. This allows a rapid pre-processing method without using complex simulations. However the system requires a further component which examine unintentional collisions between workpiece, workpiece clamps or machine periphery which is normally given by the complex simulation model. For this problem, there are created several solution concepts and the theoretical model concept of this contribution constitutes a further one to focus the given problem: Development and comparison of inverse kinematic approaches to estimate valid workpiece positions in the machine area. Section two gives an overview about the related work of the basic research area and research project. Section three contains an overview about the system and shows potential inverse kinematic models. Section four offers a statement about the calculation effort of the model approaches and discuss the potential of the models using several role model scenarios to review the approaches. Section five closes the contribution with a conclusion and an outlook.

2 Research Project and Related Work of the Research Project

In order to improve the work preparation process, the research project "*InVorMa*" arose, which is supported by the German Ministry of Education and Research. The goal of the project is to develop a service platform in order to optimize work preparation processes and the identification of optimized and valid setup of production parameters in the area of cutting machining processes. The project contains subprojects to present several solutions to guarantee an optimal job-schedule, suitable machine selection depending on workplace volume as well as setup optimization of production parameters and the distribution to machine instances and computer resources.

In this area, there arose several approaches to decrease the simulation effort, computing time as well as developing an automatically experimental design to identify the best setup position coordinates of workpiece on the machine table which leads to minimal production time by simultaneous collision free positioning of workpieces.

In order to provide more efficiency optimization component, the basis particle swarm Optimization algorithm (PSO) is tested as asynchronous extension to handle stochastic node failures and asynchronous particle evaluation processes in order to shrink the total process time [1]. The contributions [2] illustrates the zero point optimization for workpieces using the metaheuristic PSO as optimization component and the NC-Parser as given software program estimating tool paths and production durations. This contribution acts as proof of concept that the PSO algorithm is correctly configured in order to search the workpiece-zero-point position using machine geometry (3- and 5-axis tooling machines) and real NC-programs in order to minimize tool paths [2]. In order to concern the problem of high simulation effort as well as testing several cluster algorithms in combination with the PSO-NC-parser-concept, the contributions [3–5] offers a conceptual solution.

3 Concept of the System and Inverse Kinematic Models

3.1 System Concept

To identify automatically potential setup parameter for correct production processes, the optimization component generates parameters which represent e.g. workpiece-positions, zero-points, and tool changes. This information are summarized as input data to define a simulation job of a potential production and machine scenario. Each of the potential solutions are pre-evaluated by a NC-parser, which offers the resulting cutting time as well as secondary machine time. In a further loop, the results which lead to the minimal production time are clustered by the K-Means algorithm and each cluster is assigned to a virtual tooling machine (computer resource) by the simulation scheduler. Because of the high simulation effort caused by virtual tooling models which use a real control unit powered by named companies, the evaluation time increase depending on computer resources. From academicals perspective that would not be a high challenge to build a scheduling system to offer an adequate solution. However for practical use in a standard production environment in companies, this procedure is unpractical, so that the idea arose to extend the pre-processing optimization containing optimization algorithm (PSO) and NC-parser and offer an inverse kinematic model along the lines of a real used tooling machine, which represents always a kinematic chain. In this way, invalid work areas are determined before the simulation sessions and jobs are organized to save work preparation time in order to decrease the total number of simulation runs with complex machine models. There is now the challenge to identify the most practical method to model the inverse kinematic for a work preparation platform. The developed architecture is shown in Fig. 1.

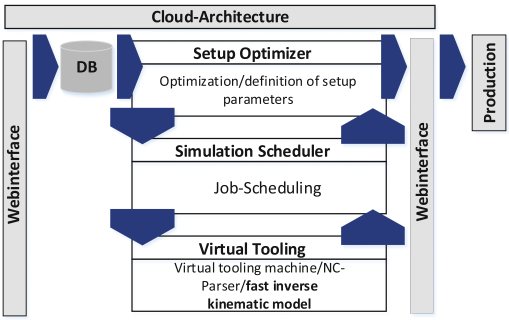

Fig. 1. Schematic system overview

3.2 Concept of the Inverse Kinematic Models and Linked Research Issues

For the solution of the inverse kinematic problems, there are general approaches which are often discussed in literature (see [6, 9–11]): A geometrical solution approach, algebraically solution approach and the numerical solution approach. An often used method especially for robot control is the quaternions algebra. The usefulness of the inverse kinematic model is the calculation of Cartesian coordinate systems in node-coordinates systems [6, 7] which could lead to the problem that the number of potential solution candidates is infinite or no solutions are possible. Figure 2 underlines the most important differences between inverse kinematic and the opposite forward kinematic (see [8, 9]). Figure 2 represents the required in- and output variables.

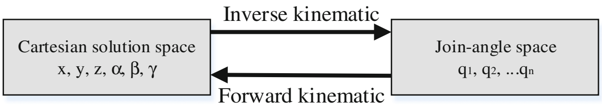

Fig. 2. Schematic overview about the differences between inverse and forward kinematic

The solution space depends on the based technical system and the design of the kinematic structure. For example, the aspired efforts to find a reasonable solution, the inverse kinematic problem has to be comply specific limit values which are given by the physical circumstances of the tooling machine design as kinematic chain. In this contribution, a 5-axis milling machine is the role model for the inverse kinematic model. Because of the kinematic chain of a tooling machine, which is shown in Fig. 3, the inverse kinematic model is built in order to use the combination of the approaches of rotation matrix and translation vector as use case 1. Use case 2 consists of the combination of quaternion and translation vector. In the quaternion algebraic approach translationally movements are feasible because of the screw theory which would lead to a high calculation effort in this context. The Screw Theory means that the turning movements caused by quaternions calculation operations are arranged such a screw thread and after a specific number of turning movements, the translationally distance is feasible (see [11]).

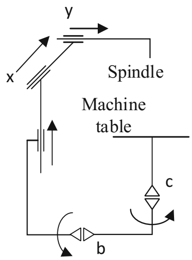

Fig. 3. Schematic kinematic model of a tooling machine

Research Issues which are investigated using the use cases:

1. *Is the combination of several model approaches usable to design the behavior of the inverse kinematic for the role model tooling machine?*
2. *Are the input and output data set required for the inverse kinematic model clear manageable for system engineering?*

3.3 Model for Use Case 1: Combination of Rotation Matrix and Translation Vectors

The degree of freedom of the model for the use cases (see Fig. 3) is assumed as f = 5. The toolpaths in the direction of x, y and z represents the coordinate system KS_0. The support coordinate systems KS_1 and KS_2 are placed including the point of origin in the center of the machine table (see Fig. 3). Let the y_{KS_1}-axis be on one line with the machine rotational b-axis and the z_{KS_1} is one direction with the machine rotational c-axis. KS_1 and KS_2 have the same point of origin and are mutually able to convert using the Euler angles α and γ. The remaining Euler angle β corresponds the angle q_2 as rotation about the b-axis. The rotation about the c-axis by angle q_1 corresponds the Euler angle γ so that the axis of direction in z is the z-axis and the axis in direction of y is defined as y-axis. For that the rotation order is important: It is mandatory to define q_2 at first, followed by q_1. It follows the equals:

$$q_1 = \gamma \tag{1}$$

$$q_2 = \beta \tag{2}$$

Let the coordinate system KS_3 the workpiece coordinate system. In order to determine the join angle q_3, q_4 and q_5 for the given coordinates x_{KS_3}, y_{KS_3} and z_{KS_3} (as target point given by the NC-program), these coordinates have to be converted from the workpiece coordinate system KS_3 to the machine coordinate system KS_0. At first there is the transformation to KS_2, than to KS_1 and finally to KS_0. The required equals are defined as:

$$x_{KS_2} = (\cos\alpha * x_{KS_3} + \sin\alpha * y_{KS_3}) + {}^3V_x^2 \tag{3}$$

$$y_{KS_2} = (-\sin\alpha * x_{KS_3} + \cos\alpha * y_{KS_3}) + {}^3V_y^2 \tag{4}$$

$$z_{KS_2} = z_{KS_3} + {}^3V_x^2 \tag{5}$$

α describes the angel of the x_{KS_2} and x_{KS_3}-axis rotating about the z-axis when their points of origins are overlapped as well as the linked orientation of the workpiece on the machine table. ${}^3V_x^2$ represents the displacement vector which has its direction from the point of origin of KS_3 to the point of origin of KS_2.

The next step is to transform the coordinates in the coordinate system KS_2 to KS_1. For that, it is notable to include the rotation of the coordinates when the machine table is turning, e.g. during the production caused by the NC-commands. For the transformation from the KS_2-system to the KS_1-coordinates, the Euler angles are used to build a transformation matrix. At first, the rotation about the z-axis occurs using angle γ followed by the rotation about the b-axis using angle β:

$$^2Rot_1(\beta,\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \tag{6}$$

$$^2Rot_1(\beta,\gamma) = \begin{pmatrix} \cos\beta\cos\gamma & -\sin\gamma & \cos\gamma\sin\beta \\ \sin\gamma\cos\beta & \cos\gamma & \sin\gamma\sin\beta \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \tag{7}$$

It follows for the coordination transformation for KS_1:

$$x_{KS_1} = (\cos\beta * \cos\gamma) * x_{KS_2} + (\sin\gamma * \cos\beta) * y_{KS_2} - \sin\beta * z_{KS_2} \tag{8}$$

$$y_{KS_1} = -\sin\gamma * x_{KS_2} + \cos\gamma * y_{KS_2} \tag{9}$$

$$z_{KS_1} = (\sin\beta * \cos\gamma) * x_{KS_2} + (\sin\beta * \sin\gamma) * y_{KS_2} + \cos\beta * z_{KS_2} \tag{10}$$

With usage of the displacement vector $^1V_0$ the coordinates x_{KS_1}, y_{KS_1} and z_{KS_1} are able to transform in the coordination system KS_0:

$$q_3 = x_{KS_0} = x_{KS_1} + {}^1V_{0x} \tag{11}$$

$$q_4 = y_{KS_0} = y_{KS_1} + {}^1V_{0y} \tag{12}$$

$$q_5 = z_{KS_0} = z_{KS_1} + {}^1V_{0z} \tag{13}$$

In order to identify the coordinates x_{KS_3}, y_{KS_3} and z_{KS_3} for the coordination system KS_3, the coordinates x_{KS_1}, y_{KS_1} and z_{KS_1} are identified at fist followed by the transformation to the system KS_2. For this, the transformation matrix (7) can be inverted.

It follows the equals:

$$x_{KS_1} = x_{KS_0} - {}^1V_{0x} \tag{14}$$

$$y_{KS_1} = y_{KS_0} - {}^1V_{0x} \tag{15}$$

$$z_{KS_1} = z_{KS_0} - {}^1V_{0x} \tag{16}$$

$$^2Rot_1(\beta,\gamma) = \begin{pmatrix} \cos\beta\cos\gamma & \sin\gamma\cos\beta & -\sin\gamma \\ -\sin\gamma & \cos\gamma & 0 \\ \sin\beta\cos\gamma & \sin\beta\sin\gamma & \cos\beta \end{pmatrix} \tag{17}$$

$$x_{KS_2} = (\cos\beta\cos\gamma) * x_{KS_1} - \sin\gamma * y_{KS_1} + (\sin\beta\cos\gamma) * z_{KS_1} \tag{18}$$

$$y_{KS_2} = (\sin\gamma\cos\beta) * x_{KS_1} + \cos\gamma * y_{KS_1} + (\sin\beta\sin\gamma) * z_{KS_1} \tag{19}$$

$$z_{KS_2} = -\sin\gamma * x_{KS_1} + \cos\beta * z_{KS_1} \tag{20}$$

After the transformation of the coordinates from the coordination system KS_2 to KS_3, it follows:

$$x_{KS_3} = \cos(-\alpha) * \left(x_{KS_2} - {}^3V_{2x}\right) + \sin(-\alpha) * \left(y_{KS_2} - {}^3V_{2y}\right) \tag{21}$$

$$y_{KS_3} = -\sin(-\alpha) * \left(x_{KS_2} - {}^3V_{2x}\right) + \cos(-\alpha) * \left(y_{KS_2} - {}^3V_{2y}\right) \tag{22}$$

$$z_{KS_3} = z_{KS_2} - {}^3V_{2z} \tag{23}$$

The given model presents the theoretical concept to allow transformation between node-coordinates and given target coordinates from the NC-program under the restrictions of finite movements of the machine and the maximum of 5-machine-axis ordered in a kinematic chain that is connected in series.

3.4 Model for Use Case 2: Combination of Translation Vectors and Quaternions

The following model will show the determination of the inverse kinematic problem of the 5-axis-tooling machine using translation vectors and quaternions.

The transformation between the coordination systems KS_3 and KS_2 are performed by displacement vector ${}^3V_2$ and quaternion ${}^3Q_2$. For the definition of the quaternion ${}^3Q_2$, there is a change of sign of the angle α between KS_3 and KS_2:

$${}^3Q_2 = \cos\frac{-\alpha}{2} + \sin\frac{-\alpha}{2} * k \tag{24}$$

$$P_{KS_2} = P_{KS_3} + 2 * {}^3Q_{2xyz} \otimes \left({}^3Q_{2xyz} \otimes P_{KS_3} + {}^3Q_{2w} * P_{KS_3}\right) + \overline{{}^3V_2} \tag{25}$$

The transformation from KS_2 to KS_1 will be performed by the total-quaternion. In order to use equals

$$w = \cos(\varphi/2) \tag{26}$$

$$(x, y, z)^T = \sin(\varphi/2) * \overrightarrow{D} \tag{27}$$

two rotation quaternions for the b- and c-axis are definable:

$${}^2Q_{1.1} = \cos\frac{-\beta}{2} + \sin\frac{-\beta}{2} * k \tag{28}$$

$${}^2Q_{1.2} = \cos\frac{-\gamma}{2} + \sin\frac{-\gamma}{2} * j \tag{29}$$

φ is a rotation angle in the range $\varphi \in [0, \pi]$, D describes the rotation axis with the unit vector $D = (D_x, D_y, D_z)^T$ (see [7, 10]), ${}^3Q_{2xyz}$ and ${}^3Q_{2w}$ describe transformation quaternions. The general nomenclature for quaternions is defined as Q. $P_{KS_{1,2,3}}$ describes a point including given coordinates in a defined coordination system KS.

The b-axis rotates about the y-unit vector and the c-axis rotates about the z-unit vector. For the angles, there are also a change of signs necessary.

Including the quaternion calculation rules for multiply quaternions, the total-quaternion is defined as:

$$
\begin{aligned}
{}^{2}Q_{1} = \cos\frac{-\gamma}{2}\cos\frac{-\beta}{2} + \left(\cos\frac{-\gamma}{2}\sin\frac{-\beta}{2}\right) * j + \left(\sin\frac{-\gamma}{2}\cos\frac{-\beta}{2}\right) \\
* k + (\sin\frac{-\gamma}{2}\sin\frac{-\beta}{2}) * i
\end{aligned}
\tag{30}
$$

For the variables j, i, k is notable that there are elements from the complex numbers: j, i, k $\in \mathbb{C}$.

For the transformation of the coordinates of a point in the coordinate system KS_2 into KS_1, the following equal is necessary:

$$
P_{KS_1} = P_{KS_2} + 2 * {}^{2}Q_{1xyz} \otimes ({}^{2}Q_{1xyz} \otimes P_{KS_2} + {}^{2}Q_{1w} * P_{KS_2}) \tag{31}
$$

For the transformation of the coordinates from KS_1 to KS_0, the equals (11), (12) and (13) in order to determine q_3, q_4, q_5 can be used. For the determination of q_1 and q_2 the eqals (1) and (2) are necessary.

4 Review Using Theoretical Scenarios

After comparison of the approaches from Sects. 3.3 and 3.4, the basic calculation steps are shown in Table 1. The result indicates that the less complex model using rotation matrix and translation vectors are sufficient for the low complex inverse kinematic such as the 5-axis machine model as an inverse kinematic chain. The basic calculation operation number of model 1 amounts $\sim$64% compared to model 2. The model 1 fulfills the requirements for the research issue 1 more than model 2 for less complex kinematic problems. The results can be explained with the fact that the transformation between two coordination systems contain only one transformation matrix which is very operational in practice using displacement vectors and rotation matrix. For parallel inverse kinematic problems, the more complex approach using quaternions in combination with translation vectors would be more profitable to implement, because the number calculation steps using quaternions is less than operating with matrix-multiplication instead [6, 10]. In addition the quaternions are numerically more stable than Euler angles and there appears no singularities [10].

To ensure if the given models follow a logically machine behavior there will be conducted pilot-calculation containing given coordinate systems and target coordinates (points) which are initialized by the machine. The calculation should estimate, that the target coordinates are valid or how far the results determine invalid machine and production behavior. Figure 4 shows an example. The given target coordinates represent the result that P_1 is calculable and successful retractable for the machine mode. P_2 is determined as invalid result, because the coordinate is out of the accessible area of the machine (see Fig. 4).

Table 1. Overview of the basic computing steps to determine results

Approaches	Multiplication operations	Additions operations	Geometrical elementary function operations	Total result of basic operation number
Model use case 1	16	13	16	45
Model use case 2	32	28	10	70

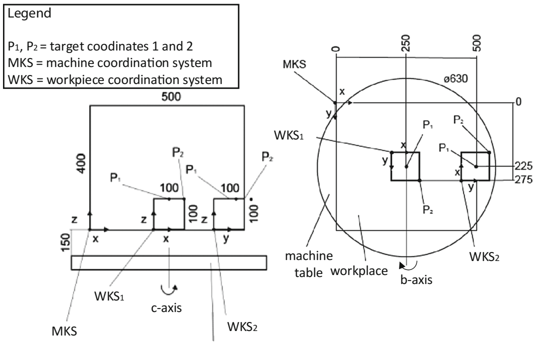

Fig. 4. Calculation scenario with two target coordinates (side and top view)

Under the role model of a real tooling machine[1] the two models offer high potential to determine invalid target positions of the tool paths and production operations, because the mathematical calculation contains a fraction of computing durations compared to the full simulation model, based on graphical presentation of the production process. The usage of inverse kinematic models under restrictions of the real machine is a useful chance to implement a successful pre-processing system in combination with the NC-parser. The pre-processing serves an estimation of near-valid setup positions of workpiece in the work area of the machine. That means, the search of valid positions without a visualization and waiting period until the simulation is finished can be spared and the probability that the results of the model represent valid positions could be increase.

Result for research issue 2: As input data, the x-,y- and z-coordinates are given by the NC-program and can be read in automatically. For rotation processes, it is more complicate to read in the specified NC-cycles. For that it is necessary to define rules for the automatically data processing. The defined cycle data from the NC-program could contain the circumstances that there have to recalculate in angle values for the Euler angles or related angle data.

[1] As role model it was used the 5-axis tooling machine "DMU 50" designed by Mori Seiki AG, Germany.

5 Conclusion and Outlook

This contribution introduces two inverse kinematic models to calculate the machine behavior under the restrictions of a real role model machine with the goal to identify valid positions for workpieces. The inverse kinematic models are kept as simple in order to prevent high calculation effort and it is shown that the usage of rotation matrix and translation vector is sufficient for the inverse kinematic as series chain. Quaternions and translation vectors is usable for more complex kinematic model e.g. parallel kinematic models. Research issue 1 will be fit for model 1 and for model 2 in order to address a high complex inverse kinematic problem. Research issue 2 leads to the result that there are extended data processing for NC-cycles required that contains rotation commands. For the future work, the model has to be extended (see Sect. 3.1) for further tests in combination with the PSO-algorithm and the NC-parser especially the evaluation by the tooling machine model (see Sect. 3.1).

Acknowledgments. We thank Silas Lummer for the model preparation and his support to execute the tests and experiments.

References

1. Reisch, R.-E., Weber, J., Laroque, C., Schröder, C.: Asynchronous optimization techniques for distributed computing applications. In: Tolk, A., Padilla, J.J., Jafar, S. (eds.) Spring Simulation Multi Conference 2015 Proceedings, 48th Annual Simulation Symposium, Alexandria, Virginia, vol. 47. No. 2, pp. 49–57. Institute of Electrical and Electronics Engineers, Inc. (2015)
2. Weber, J.: A technical approach of a simulation-based optimization platform for setup-preparation via virtual tooling by testing the optimization of zero point positions in CNC-applications. In: Yilmaz, L., Chan, W.K.V., Moon, I., Roeder, T.M.K., Macal, C., Rossetti, M.D. (eds.) Proceedings of 2015 Winter Simulation Conference, Huntington Beach, CA, USA (2015)
3. Weber, J., Mueß, A., Dangelmaier, W.: A simulation based optimization approach for setting-up CNC machines. In: Doerner, K.F., Ljubic, I., Pflug, G., Tragler, G. (eds.) Operations Research Proceedings 2015. ORP, pp. 445–451. Springer, Cham (2017). doi:10.1007/978-3-319-42902-1_60
4. Mueß, A., Weber, J., Reisch, R.-R., Jurke, B.: Implementation and comparison of cluster-based PSO extensions in hybrid settings with efficient approximation. In: Niggemann, O., Beyerer, J. (eds.) Machine Learning for Cyber Physical Systems – Selected Papers from the international Conference ML4CPS 2015 – Technologien für die intelligente Automation, Lemgo, Germany, vol. 1, pp. 87–93. Springer, Heidelberg (2015)
5. Laroque, C., Weber, J., Reisch, R.-R., Schröder, C.: Ein Verfahren zur simulationsgestützten Optimierung von Einrichtungsparametern an Werkzeugmaschinen in Cloud-Umgebungen. In: Nissen, V., Stelzer, S., Straßburger, S., Firscher, D. (eds.) Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI) 2016, vol 3, Ilmenau, Germany, pp. 1761–1772. Monsenstein und Vannerdat OHG, Universitätsverlag Ilmenau (2016)
6. Weber, W.: Industrieroboter: Methoden der Steuerung und Regelung. Fachbuchverlag, Carl-Hanser-Verlag, Leipzig, Germany (2009)

7. Wenz, M.: Automatische Konfiguration der Bewegungssteuerung von Industrierobotern. Logos Verlag, Berlin, Germany (2008)
8. Siciliano, B., Khatib, O.: Handbook of Robotics. Springer, Heidelberg (2008)
9. Sieger, H.-J., Bocionek, S.: Robotik: Programmierung Intelligenter Roboter. Springer, Heidelberg (1996)
10. Husty, M., Karger, A., Sachs, H., Steinhilper, W.: Kinematik und Robotik. Springer, Heidelberg (1997)
11. Selig, J.M.: Geometric Fundamentale of Robotics, 2nd edn. Springer, Heidelberg (2005)