robotics

MDPI

Article

Modeling Parallel Robot Kinematics for 3T2R and 3T3R Tasks Using Reciprocal Sets of Euler Angles

Moritz Schappler *, Svenja Tappe and Tobias Ortmaier

Institute of Mechatronic Systems, Leibniz University Hannover, 30167 Hannover, Germany

* Correspondence: moritz.schappler@imes.uni-hannover.de

Received: 27 June 2019; Accepted: 1 August 2019; Published: 6 August 2019

Abstract: Industrial manipulators and parallel robots are often used for tasks, such as drilling or milling, that require three translational, but only two rotational degrees of freedom ("3T2R"). While kinematic models for specific mechanisms for these tasks exist, a general kinematic model for parallel robots is still missing. This paper presents the definition of the rotational component of kinematic constraints equations for parallel robots based on two reciprocal sets of Euler angles for the end-effector orientation and the orientation residual. The method allows completely removing the redundant coordinate in 3T2R tasks and to solve the inverse kinematics for general serial and parallel robots with the gradient descent algorithm. The functional redundancy of robots with full mobility is exploited using nullspace projection.

Keywords: parallel robot; five-DoF task; 3T2R task; functional redundancy; task redundancy; redundancy resolution; reciprocal Euler angles; inverse kinematics

1. Introduction

Industrial tasks like welding, gluing, milling or drilling represent a major part of the applications of industrial robots, which generally have full mobility, i.e., the operational space of their end-effector has three translational and three rotational ("3T3R") degrees of freedom ("DoF"). Parallel robots like the Stewart platform have especially been proposed for milling tasks regarding their high structural stiffness. The task space of the named applications can be defined by three translational DoF and only two rotations due to a symmetry around the tool axis ("3T2R"). This results in a functional or task redundancy, which is not exploited to full extend yet for *parallel* robots.

1.1. Inverse Kinematics and Resolution of Task Redundancy for Serial-Link Robots

Various general gradient-based methods exist to solve the inverse kinematics for *serial* robots; either by augmenting the joint space [1] or by reducing the task space [2–5]. The different approaches each define a residual vector and a gradient matrix considering the properties of 3T2R tasks, e.g., by adding a virtual joint axis [1], orthogonal decomposition of the task space [2], rotation of the residual into a task frame and removing the corresponding component [3], defining the tool axis by two points for constructing a nullspace [4] or by defining the absolute orientation and the orientation residual with two reciprocal sets of Euler angles [5]. The gradient matrices corresponding to the different residuals are used for an iterative Newton-Raphson algorithm [6,7] by exploiting the functional redundancy with a null space projection of additional performance criteria [6]. Without the definition of a proper gradient, a global optimization has to be performed outside of the inverse kinematics algorithm [8,9].

1.2. Overview of Parallel Robots Structures for 3T2R Tasks

Parallel robots in 3T2R tasks can be ordered in classes according to their kinematic structure into

I mechanisms with full platform mobility (3T3R) that are redundantly controlled to five DoF,
II mechanisms with 3T2R platform mobility enforced with a passive five-DoF constraining leg and five other legs with six DoF each,
III mechanisms with 3T2R platform mobility resulting from the mobility of five actuated legs with five or six DoF each,
IV mechanisms with 3T2R platform mobility and five legs with only five DoF each.

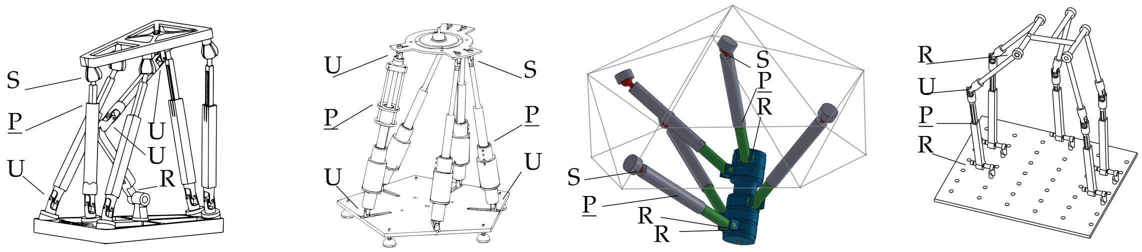
The classes I and II were introduced in [10], where class IV is analyzed regarding leg symmetry and singularities. Class III is mainly influenced by the systematic synthesis of [11] and several existing prototypes and is demarcated against class II by the absence of the passive constraining leg. Class IV can be seen as a subclass of III, but is differentiated in this paper due to its characteristics. Other classifications are provided e.g., by [12], where IV and II are termed “families”.

Examples for the first class are hexapods (6UPS) [13] or the Eclipse [14] machine tool (2PPRS-PPRS). (The joint structure of the parallel robots is denoted by the number of the legs and the order of universal (“U”), prismatic (“P”), spherical (“S”), helical (“H”) and revolute (“R”) joints in the leg chains. Different actuated legs are connected by hyphens (“-”), passive constraint legs are connected by a slash (“/”). Actuated joints are underlined.) Any other parallel robot with full mobility (see e.g., [11,15,16]) may be used as well.

The second class allows for more variety, since the six-DoF mechanism and the five-DoF constraining leg can manifest in different kinematic structures: The UPS structure is used for the six-DoF part of the mechanism by [17] with a focus on kinematic analysis of 5UPS/US, by [18] with a focus on kinetostatic modeling at the example of 5UPS/RUU (see Figure 1a), by [19] with focus on trajectory control of 5UPS/PRPU and by [20] for pose measurement with the passive leg of 5UPS/PRPU. Other possible general base structures are RUS at the 5RUS/US example in [17], PUS, which has been investigated for the control of a redundantly actuated 6PUS/UPU regarding the force control of the redundant leg with inverse dynamics compensation [21] or force optimization [22].

The most-straightforward member of the third class is the 4UPS-UPU of Figure 1b, which is investigated in [23] for a simulation and feasibility study together with a survey on possible architectures for a technical realization of this class. Other possible structures are the 4URS-URU, which is analyzed kinematically in [24] and the 4PSU-PU*U, which has a special parallelogram structure in one leg (termed “U*”) and is presented in [25]. A subclass of III consists of mechanisms [26–28], where the last joint axis of the legs is coaxial with the tool axis and is constructed as rotating ring. It contains the Metrom machine tool (4SPRR-SPR), depicted in Figure 1c, which is analyzed regarding inverse and forward kinematics in [26] or its variants, the redundant 4SPRR-PSPR from [28] or the hybrid 4URHU-URHR with an additional linear actuator at the platform [27]. A structural synthesis based on linear transformations and evolutionary morphology [11] led e.g., to the Isoglide5 mechanisms (3PRRRRR-2PRRRR), which are analyzed and optimized regarding the isotropy of the Jacobian in [29].

The simplest member of class IV, the 5UPU is shown in [30] with the help of screw theory to only have local mobility and no global mobility, since the twist systems of the leg chains have no intersection and the resulting twist system of the platform is empty. Members of class IV have been found by systematic structural synthesis with screw theory, which has been performed in [31] for symmetric 3T2R mechanisms. The resulting 5RPUR of Figure 1d and 5PRUR are analyzed in [10,32].



(**a**) 5UPS/RUU (class II) (**b**) 4UPS-UPU (class III) (**c**) 4SPRR-SPR (Metrom, class III) (**d**) 5RPUR (class IV)

Figure 1. Typical mechanisms of the different classes. Adapted from [18] (**a**), [23] (**b**), [28] (**c**), [10] (**d**).

In a practical application with competing requirements on workspace, stiffness, costs and precision, each of the existing systems has its legitimization. Nevertheless, each of the classes has inherent disadvantages: In tasks like milling with high process forces and requirements on stiffness and precision, robots from class II with one constraining leg have the drawback that the passive leg takes the complete reaction wrench in the blocked rotational degree of freedom, which strongly affects the mechanisms stiffness in this direction [18]. The same is argued by [27] at the example of the Metrom machine tool, but can be extended to all the members of class III, where most leg chains have six DoF and usually only one leg chain has five DoF. This leg chain also has to take the reaction moments in the blocked DoF which affects the overall stiffness. Therefore, members of the classes I and IV can be expected to reach a higher stiffness. Mechanisms of the class IV may further suffer from an increased sensitivity of manufacturing tolerances, which may cause a high pretension of the bearings or even reduce the DoF, since the five DoF of all leg chains have to coincide exactly to allow the platform to also have five DoF. Additionally, only members of class I provide redundancy which allows using the additional DoF for performance optimizations, e.g., to avoid singularities and to compensate the smaller workspace caused by the sixth leg.

Therefore, the remainder of the paper focuses on mechanisms of the first class to allow an optimization of their performance criteria using the degree of task redundancy.

1.3. Inverse Kinematics of Parallel Robots for 3T2R Tasks

The parallel robots with five DoF presented above have a kinematic structure which allows for an analytic model of the inverse kinematics. All references define the end-effector orientation with two consecutive elementary rotations, i.e., define two Euler angles to represent the tool axis orientation in minimal coordinates [10,13,19–22,26,28,32,33], which is called "partial pose" in [13]. The inverse kinematics problem (IKP) is first solved for the first chain, which is called "leading chain" in this paper. Due to the geometry of the leading chain, this solution can be found algebraically. Then the IKP is solved for the other "following" chains with the given orientation from the leading chain and standard methods. For robots of class II, the constraining leg is selected as the leading leg chain and for class III the 3T2R leg is selected.

To the best knowledge of the authors, only one reference [13] for the IKP of functionally redundant parallel robots of the class I is known. The reason presumably is that a solution of the 3T3R IKP for these robots is possible with standard methods, as used in [14] for the 3T3R Eclipse. It is always possible to transfer the 3T2R IKP into 3T3R by adding an arbitrary value for the desired rotation around the tool axis. An optimization of additional performance criteria is possible by varying the redundant rotation angle [8,9]. This approach was chosen in [13] by first defining the IKP with the redundant rotation as a parameter and then performing an optimization of this parameter using analytical computation of the dexterity and interval analysis to ensure a minimum determinant of the inverse Jacobian. The drawback of this method is the need for a cascaded optimization which is more complex than the gradient-based approach presented in this paper.

1.4. Motivation and Summary of the State of the Art

The overview over the literature shows that no general, machine-independent method for the resolution of functional redundancy for 3T3R PKM in 3T2R tasks exists. The works either focus on a general structural synthesis of machines, e.g., via screw theory [31] or linear transformations [29] or the description and improvement of specific, manually selected machines. To choose the best machine for given requirements, a structural synthesis is only the first step. Additionally, a dimensional synthesis should be performed for all possible structures to select the most suitable mechanism. This combined structural and dimensional synthesis [34] is sketched in Figure 2. To be able to perform the dimensional synthesis for all structures, the inverse kinematics has to be implemented in a general form to calculate the performance criteria over a given trajectory for further optimization of the structures and their comparison. For the generation of task redundant parallel robots, the inverse kinematics has to include an optimization of the performance criteria and the restrictions such as joint limits to ensure a comparability of the results.

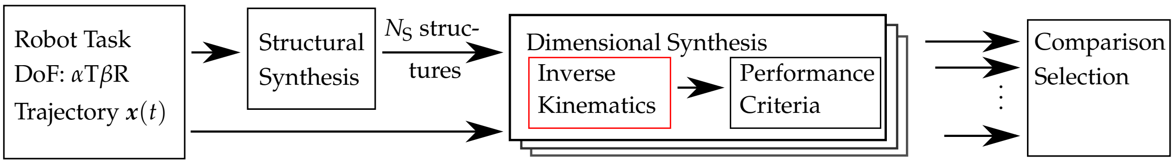


Figure 2. Overview of the procedure for combined structural and dimensional synthesis.

To address this, the contributions of this paper are

- a general kinematics model for parallel robots using the concept of reciprocal Euler angles [5],
- a complete elimination of the redundant operational space coordinate in this formulation for 3T2R tasks allowing a nullspace optimization in the gradient-based inverse kinematics,
- proofs, examples and simulations to show the performance for single serial kinematic leg chains and complete parallel robots.

The remainder of the paper is structured as follows: The description of the inverse kinematics problem and prior definitions are given in Section 2 and the concept of reciprocal Euler angles from [5] is adapted in Section 3 for parallel robots. This leads to the full kinematic constraints of parallel robots in 3T2R tasks, introduced in Section 4 and applied to the differential kinematics in Section 5. The theoretical analysis is followed by examples and simulations in Section 6. The appendix contains proofs and additional details on the mathematical formulation.

2. Inverse Kinematics Problem for Parallel Robots

Before addressing the specific model for parallel robots in 3T2R tasks in the next section, the standard kinematics model of parallel kinematic machines ("PKM") is repeated in the following, corresponding to the state of the art [11,15,35] and serving as a reference to highlight its shortcomings for 3T2R tasks. The regarded parallel robot consists of m legs, which each have the joint coordinates $\boldsymbol{q}_i$. All joints are considered to be single-DoF and additionally to the active joints $\boldsymbol{q}_{i,\mathrm{a}}$ explicitly all passive joints at the base and at the platform $\boldsymbol{q}_{i,\mathrm{p}}$ are included in the coordinates $\boldsymbol{q}_i$ of leg i. The coordinates

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_\mathrm{t}^\mathrm{T} & \boldsymbol{x}_\mathrm{r}^\mathrm{T} \end{bmatrix}^\mathrm{T} \in \mathbb{R}^6 \tag{1}$$

of the end-effector platform describe the position and orientation of the end-effector frame $\mathscr{F}_D$ with respect to the base frame $\mathscr{F}_0$. In the equations, this is marked with left subscript "(0)" for vectors and left superscript "0" for rotation matrices. The platform-related end-effector frame is the desired frame in the inverse kinematics problem and is therefore abbreviated with "D". The position

$$\boldsymbol{x}_\mathrm{t} = {}_{(0)}\boldsymbol{r}_D \in \mathbb{R}^3 \tag{2}$$

is defined as the origin of the platform frame and the rotation matrix

$${}^0\boldsymbol{R}_D(\boldsymbol{x}_\mathrm{r}) = \begin{bmatrix} \boldsymbol{n}_D & \boldsymbol{o}_D & \boldsymbol{a}_D \end{bmatrix} \in SO(3) \tag{3}$$

of the platform frame is expressed with Euler angles

$$\boldsymbol{x}_\mathrm{r} = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 \end{bmatrix}^\mathrm{T} =: \boldsymbol{\beta} \in \mathbb{R}^3 \tag{4}$$

as a minimal representation of the orientation coordinates. The symbol "$\boldsymbol{\beta}$" will be used to denote orientations relative to the base frame throughout this paper. The XYZ convention

$$\boldsymbol{R}(\boldsymbol{\beta}) = \boldsymbol{R}_x(\beta_1)\boldsymbol{R}_y(\beta_2)\boldsymbol{R}_z(\beta_3) \in \mathrm{SO}(3) \tag{5}$$

is used for the Euler angles without loss of generality. The relation between joint coordinates $\boldsymbol{q}$ and platform coordinates $\boldsymbol{x}$ is established with the kinematic constraint equations, for which most commonly the vector loop

$$\boldsymbol{\Phi}_{\mathrm{t},i}(\boldsymbol{q}_i, \boldsymbol{x}) = -{}_{(0)}\boldsymbol{r}_{A_iB_i}(\boldsymbol{x}) + {}_{(0)}\boldsymbol{r}_{A_iB_i}(\boldsymbol{q}_i) \tag{6}$$

between the position of the platform coupling point B_i relative to the base coupling point A_i is used for each leg chain i [15]. The second term ${}_{(0)}\boldsymbol{r}_{A_iB_i}(\boldsymbol{q}_i)$ corresponds to the forward kinematics of the serial leg chain. The vector

$${}_{(0)}\boldsymbol{r}_{A_iB_i}(\boldsymbol{x}) = -{}_{(0)}\boldsymbol{r}_{A_i} + \boldsymbol{x}_{\mathrm{t}} + {}^{0}\boldsymbol{R}_D(\boldsymbol{x}_{\mathrm{r}})\,{}_{(D)}\boldsymbol{r}_{B_i} \tag{7}$$

includes the term ${}^{0}\boldsymbol{R}_D(\boldsymbol{x}_{\mathrm{r}})$ that depends on the full orientation $\boldsymbol{x}_{\mathrm{r}}$ of the end-effector. For the bigger part of existing parallel robots, the passive joint coordinates can be eliminated analytically from Equation (6), e.g., by using the Euclidean distance for U$\underline{\text{P}}$S or R$\underline{\text{P}}$R leg chains or via trigonometry for $\underline{\text{R}}$RR chains. This is termed "minimal kinematics set" in [15] and leads to the scalar constraint equation

$$\Phi_i = \Phi_i(\boldsymbol{q}_{i,\mathrm{a}}, \boldsymbol{x}) \tag{8}$$

for each leg i, which can be assembled to the vector of constraint equations

$$\boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{a}}, \boldsymbol{x}) = \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_m \end{bmatrix}^{\mathrm{T}} \tag{9}$$

for all m legs of the PKM. The differential kinematics of the PKM is calculated with the time derivative

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{a}}, \boldsymbol{x}) = \boldsymbol{\Phi}_{\partial q_{\mathrm{a}}}\dot{\boldsymbol{q}}_{\mathrm{a}} + \boldsymbol{\Phi}_{\partial x}\dot{\boldsymbol{x}} = \boldsymbol{0} \tag{10}$$

where the passive joint coordinates $\boldsymbol{q}_{\mathrm{p}}$ do not occur, since they have been eliminated in a previous step. Following [11] to avoid the term "Jacobian" for the elements of (10), the inverse-kinematics matrix

$$\boldsymbol{\Phi}_{\partial q_{\mathrm{a}}} = \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{q}_{\mathrm{a}}} = \begin{bmatrix} \Phi_{1,\partial q_{1,\mathrm{a}}} & 0 & 0 & 0 \\ 0 & \Phi_{2,\partial q_{2,\mathrm{a}}} & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \Phi_{m,\partial q_{m,\mathrm{a}}} \end{bmatrix} \tag{11}$$

of this model has diagonal form and the direct-kinematics matrix

$$\boldsymbol{\Phi}_{\partial x} = \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{x}} = \begin{bmatrix} (\partial\Phi_1/\partial\boldsymbol{x})^{\mathrm{T}} & (\partial\Phi_2/\partial\boldsymbol{x})^{\mathrm{T}} & \cdots & (\partial\Phi_m/\partial\boldsymbol{x})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{12}$$

is fully populated. This definition of the constraints has the following drawbacks:

1. For parallel robots with arbitrary leg chains like those generated by a structural synthesis [11,31,34], it is generally not possible to analytically eliminate the passive joint coordinates.
2. If more than three joint coordinates per leg influence the coupling point position B_i, the three kinematic constraints per joint in (6) are not sufficient to generate enough equations for the matrix of (11) to become invertible.

3. The velocity-based theory of linear transformations used by [11] allows determining the mobility of arbitrary parallel robots. The linear transformation is generalized in [12] to accuracy and stiffness modeling by means of screw theory resulting in the "generalized Jacobian". Both concepts are similar to (10) and do not provide a direct appliance to solve the IKP, since this requires a formulation of the orientation at position level, not velocity level.
4. Exploiting the reduction of end-effector coordinates for 3T2R tasks is not possible, since all end-effector coordinates are included in (7).

An alternative kinematic model to encounter the combination of these points is presented in the next section, where the concept of reciprocal sets of Euler angles for the inverse kinematics problem for serial-link robots [5] is transferred to the leading leg of parallel robots.

3. Reciprocal Sets of Euler Angles for the Kinematics of a Serial Leg Chain

To take the rotational symmetry around the tool axis in 3T2R tasks into account, a new set of task space coordinates

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_{\mathrm{t}}^{\mathrm{T}} & \boldsymbol{\eta}_{\mathrm{r}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^5 \tag{13}$$

has to be defined. The translational part

$$\boldsymbol{\eta}_{\mathrm{t}} = \boldsymbol{x}_{\mathrm{t}} = {}_{(0)}\boldsymbol{r}_D \in \mathbb{R}^3 \tag{14}$$

remains unchanged relative to the operational space coordinates $\boldsymbol{x}$. The rotational part

$$\boldsymbol{\eta}_{\mathrm{r}} = \begin{bmatrix} \beta_1 & \beta_2 \end{bmatrix}^{\mathrm{T}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{=\boldsymbol{P}_{\eta_{\mathrm{r}}}} \boldsymbol{x}_{\mathrm{r}} \in \mathbb{R}^2 \tag{15}$$

only contains the first two rotational coordinates of $\boldsymbol{x}$. The last operational space coordinate β_3, the rotation around the z-axis $\boldsymbol{a}_D$ of $\mathscr{F}_D$, is excluded from the task space by the selection matrix $\boldsymbol{P}_{\eta_{\mathrm{r}}}$. To be able to set the rotational DoF around the tool axis in 3T2R tasks arbitrarily and use gradient-based inverse kinematics, β_3 has to be eliminated completely from the kinematics Equations (6). To simplify the following elaborations, the platform frame $\mathscr{F}_D$ is still identified as the desired frame of the inverse kinematics problem and the end-effector frame that results from the joint angles of leg i is now termed $\mathscr{F}_E$, which corresponds to the forward kinematics of the leg chain. For a formulation without the tool axis rotation, a different constraint definition

$$\boldsymbol{\Phi}_{\mathrm{t},i}(\boldsymbol{q}_i, \boldsymbol{x}) = -{}_{(0)}\boldsymbol{r}_D + {}_{(0)}\boldsymbol{r}_E(\boldsymbol{q}_i) = -\boldsymbol{x}_{\mathrm{t}} + {}_{(0)}\boldsymbol{r}_E(\boldsymbol{q}_i) \in \mathbb{R}^3 \tag{16}$$

containing the vector loop from the robot base frame $\mathscr{F}_0$ to the platform $\mathscr{F}_D$ and the leg chain end-effector $\mathscr{F}_E$ can be used, where in contrast to (6) only the translational part $\boldsymbol{x}_{\mathrm{t}}$ of the end-effector coordinates appears and not the rotational part $\boldsymbol{x}_{\mathrm{r}}$. The vector loop is depicted in Figure 3 for a planar robot with opened (Figure 3a,b) and closed loops (Figure 3c). The triangle represents the end-effector platform and only one leg chain is drawn in the figure.

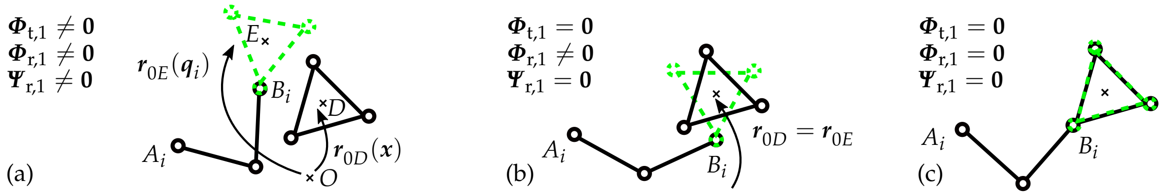

$\boldsymbol{\Phi}_{\mathrm{t},1} \neq \boldsymbol{0}$
$\boldsymbol{\Phi}_{\mathrm{r},1} \neq \boldsymbol{0}$
$\boldsymbol{\Psi}_{\mathrm{r},1} \neq \boldsymbol{0}$
E
$\boldsymbol{r}_{0E}(\boldsymbol{q}_i)$
B_i
D
A_i
$\boldsymbol{r}_{0D}(\boldsymbol{x})$
O
(a)
$\boldsymbol{\Phi}_{\mathrm{t},1} = \boldsymbol{0}$
$\boldsymbol{\Phi}_{\mathrm{r},1} \neq \boldsymbol{0}$
$\boldsymbol{\Psi}_{\mathrm{r},1} = \boldsymbol{0}$
$\boldsymbol{r}_{0D} = \boldsymbol{r}_{0E}$
A_i
B_i
(b)
$\boldsymbol{\Phi}_{\mathrm{t},1} = \boldsymbol{0}$
$\boldsymbol{\Phi}_{\mathrm{r},1} = \boldsymbol{0}$
$\boldsymbol{\Psi}_{\mathrm{r},1} = \boldsymbol{0}$
A_i
B_i
(c)

Figure 3. Different cases for the kinematic constraints of the leading chain for the 3RRR example: (**a**) no constraints complied; (**b**) position and tool axis rotation complied; (**c**) all constraints complied.

As a drawback of (16), all joint angles $\boldsymbol{q}_i$ of the leg i and not only the coordinates of the first joints counted from the base are now included in the vector

$$_{(0)}\boldsymbol{r}_E(\boldsymbol{q}_i) = {}_{(0)}\boldsymbol{r}_{A_i} + {}_{(0)}\boldsymbol{r}_{A_iB_i}(\boldsymbol{q}_i) + {}^0\boldsymbol{R}_{B_i}(\boldsymbol{q}_i)_{(B_i)}\boldsymbol{r}_E. \tag{17}$$

This implies that the platform is now part of the last link of the considered leg chain, as sketched in Figure 3 by the dashed triangle. To account for the increased number of included joints in (17), the full kinematic constraints

$$\boldsymbol{\Phi}_i = \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{t},i}^{\mathrm{T}} & \boldsymbol{\Phi}_{\mathrm{r},i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6, \tag{18}$$

have to be considered, including the rotational part

$$\boldsymbol{\Phi}_{\mathrm{r},i}(\boldsymbol{q}_i, \boldsymbol{x}) = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}^{\mathrm{T}} = \boldsymbol{\alpha}\left({}^D\boldsymbol{R}_E(\boldsymbol{x}_\mathrm{r}, \boldsymbol{q}_i)\right) = \boldsymbol{\alpha}\left({}^0\boldsymbol{R}_D^{\mathrm{T}}(\boldsymbol{x}_\mathrm{r}){}^0\boldsymbol{R}_E(\boldsymbol{q}_i)\right), \tag{19}$$

which is needed to generate enough equations for an invertible matrix in the differential equations. The constraints again contain the deviation between the desired end-effector frame $\mathscr{F}_D$ expressed with $\boldsymbol{x}$ and the actual robots end-effector frame $\mathscr{F}_E$ expressed with $\boldsymbol{q}$. Figure 3b,c show cases, where the translational constraints are met, but the rotational constraints have different values. For 3T3R tasks, only Figure 3c represents a valid solution of the inverse kinematics. For 3T2R tasks, Figure 3b,c represent valid solutions.

The goal of eliminating the tool rotation β_3 from the equations is not achieved yet, since all three components of the platform orientation $\boldsymbol{x}_\mathrm{r}$ affect the rotation matrix ${}^0\boldsymbol{R}_D$. This can be addressed by the selection of the Euler angles: Similar to the definition of the rotational operational space coordinates $\boldsymbol{x}_\mathrm{r}$ in (4), the constraints $\boldsymbol{\Phi}_{\mathrm{r},i}$ are also expressed with a set of Euler angles $\boldsymbol{\alpha}$. In the following, "$\boldsymbol{\alpha}$" will always refer to the rotation error/residual and "$\boldsymbol{\beta}$" to an orientation relative to the base frame. The Euler angle convention of $\boldsymbol{\alpha}$ can be chosen independently of the choice for the orientation representation in $\boldsymbol{\beta}$. The intuitive approach of choosing

$$\boldsymbol{R}(\boldsymbol{\alpha}^*) := \boldsymbol{R}_x(\alpha_1^*)\boldsymbol{R}_y(\alpha_2^*)\boldsymbol{R}_z(\alpha_3^*) \in \mathrm{SO}(3) \tag{20}$$

the same way as $\boldsymbol{\beta}$ leads to a set of transformations depicted in Figure 4a, where the intermediate steps of the single elementary rotations are omitted since they have no technical meaning. The superscript asterisk in (20) demarcates this specific example and the following elaborations on the calculation of $\boldsymbol{\alpha}$.

To be able to remove the redundant coordinate β_3 from the rotational constraints of (19), it is necessary to change the expression of the orientation error $\boldsymbol{\alpha}$ to be reciprocal to the expression of the absolute orientation $\boldsymbol{\beta}$. By using the ZYX Euler angles with

$$\boldsymbol{R}(\boldsymbol{\alpha}) := \boldsymbol{R}_z(\alpha_1)\boldsymbol{R}_y(\alpha_2)\boldsymbol{R}_x(\alpha_3) \in \mathrm{SO}(3), \tag{21}$$

only the error component α_1 is affected by rotations around the tool axis, which is the z-axis of the intermediate frames $\mathscr{F}_{A1}$, $\mathscr{F}_{A2}$ and the platform frame $\mathscr{F}_D$ in Figure 4b, where the frame rotations with the reciprocal set of Euler angles are sketched.

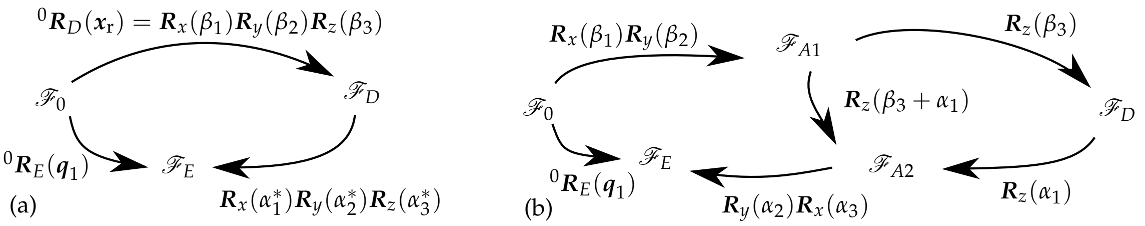

${}^0\boldsymbol{R}_D(\boldsymbol{x}_\mathrm{r}) = \boldsymbol{R}_x(\beta_1)\boldsymbol{R}_y(\beta_2)\boldsymbol{R}_z(\beta_3)$
$\mathscr{F}_0$
$\mathscr{F}_D$
${}^0\boldsymbol{R}_E(\boldsymbol{q}_1)$
$\mathscr{F}_E$
(a)
$\boldsymbol{R}_x(\alpha_1^*)\boldsymbol{R}_y(\alpha_2^*)\boldsymbol{R}_z(\alpha_3^*)$
$\boldsymbol{R}_x(\beta_1)\boldsymbol{R}_y(\beta_2)$
$\mathscr{F}_{A1}$
$\boldsymbol{R}_z(\beta_3)$
$\boldsymbol{R}_z(\beta_3 + \alpha_1)$
$\mathscr{F}_0$
$\mathscr{F}_D$
$\mathscr{F}_E$
$\mathscr{F}_{A2}$
${}^0\boldsymbol{R}_E(\boldsymbol{q}_1)$
(b)
$\boldsymbol{R}_y(\alpha_2)\boldsymbol{R}_x(\alpha_3)$
$\boldsymbol{R}_z(\alpha_1)$

Figure 4. Overview of the different frames (**a**) for six-DoF tasks with standard Euler angle convention and (**b**) for five-DoF tasks with reciprocal Euler angle convention; taken from [5].

The mathematical proof is given in Appendix A.1 and in [5]. The new, reduced rotational part of the kinematic constraints

$$\boldsymbol{\Psi}_{\mathrm{r},i}(\boldsymbol{q}_i, \boldsymbol{\eta}) = \begin{bmatrix} \alpha_2 & \alpha_3 \end{bmatrix}^{\mathrm{T}} = \overbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{=\boldsymbol{P}_{\boldsymbol{\Psi}_{\mathrm{r}}}} \boldsymbol{\Phi}_{\mathrm{r},i}(\boldsymbol{q}_i, \boldsymbol{x}) \in \mathbb{R}^2 \tag{22}$$

does not contain the tool rotation any more. The full kinematic constraints

$$\boldsymbol{\Psi}_i = \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{t},i}^{\mathrm{T}} & \boldsymbol{\Psi}_{\mathrm{r},i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^5 \tag{23}$$

for the reduced coordinates can be used for the inverse kinematics of the leg chain i in 3T2R tasks, where $\boldsymbol{\Psi}_i = \boldsymbol{0}$ leads to a valid position and orientation of the tool axis. The condition $\boldsymbol{\Phi}_i = \boldsymbol{0}$ leads to a valid configuration of leg i of the parallel robot in 3T3R tasks. In the following, "$\boldsymbol{\Phi}$" is always used for 3T3R kinematic descriptions and "$\boldsymbol{\Psi}$" for 3T2R. Note that by omitting the corresponding lines in the operational space coordinates $\boldsymbol{x}$ and the constraint equations $\boldsymbol{\Phi}$, it is also possible to use the 3T3R approach for systems with reduced mobility of 2T1R, 3T0R and 3T1R platform DoF.

4. Full Kinematic Constraints for Parallel Robots Using Reciprocal Sets of Euler Angles

The definition of the full kinematic constraints (16,19,18) of a single leg chain of the parallel robot from the previous chapter can be used to write the kinematic constraints in a general form. The full kinematic constraint equations can only be defined for 3T3R tasks without further adjustments as

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1^{\mathrm{T}} & \boldsymbol{\Phi}_2^{\mathrm{T}} & \cdots & \boldsymbol{\Phi}_m^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{24}$$

The constraints $\boldsymbol{\Psi}_i$ from (23) for the reduced coordinates $\boldsymbol{\eta}$ can only be defined for one leg chain: Figure 5a show an open-loop second leg chain for a given first leg chain from Figure 3.

By also closing the *3T2R* kinematic constraints $\boldsymbol{\Psi}_2$ for the second loop, as depicted in Figure 5b, the tool axis stays arbitrary and the platform pose demanded from the two legs would be different and therefore would not be a valid solution for the complete mechanism, i.e., $\boldsymbol{\Phi}_2 \neq \boldsymbol{0}$. Only if the second leg fulfills the *3T3R* kinematic constraints for all platform coordinates, as shown in Figure 5c, a valid configuration of the mechanism emerges. This approach has already been used for many specific robots systems, as introduced in Section 1. As a generalization, the first leg of the parallel robot is now termed the "leading leg chain" (index "1") and the other legs are termed as "following leg chains" (index "j").

The translational part of the constraints is not coupled by the platform orientation and therefore left unchanged relative to (16) with

$$\boldsymbol{\Phi}_{\mathrm{t},j}(\boldsymbol{q}_j, \boldsymbol{x}) = -_{(0)}\boldsymbol{r}_D + _{(0)}\boldsymbol{r}_E(\boldsymbol{q}_j) = -\boldsymbol{x}_{\mathrm{t}} + _{(0)}\boldsymbol{r}_E(\boldsymbol{q}_j) \in \mathbb{R}^3 \tag{25}$$

for the following legs j.

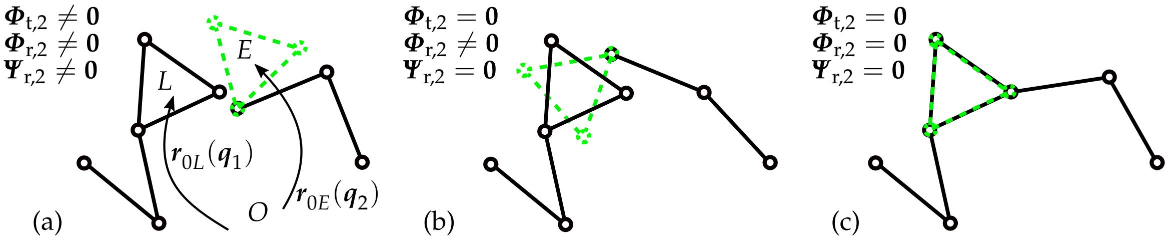

$\boldsymbol{\Phi}_{\mathrm{t},2} \neq \boldsymbol{0}$
$\boldsymbol{\Phi}_{\mathrm{r},2} \neq \boldsymbol{0}$
$\boldsymbol{\Psi}_{\mathrm{r},2} \neq \boldsymbol{0}$
E
L
$r_{0L}(q_1)$
$r_{0E}(q_2)$
O
(a)
$\boldsymbol{\Phi}_{\mathrm{t},2} = \boldsymbol{0}$
$\boldsymbol{\Phi}_{\mathrm{r},2} \neq \boldsymbol{0}$
$\boldsymbol{\Psi}_{\mathrm{r},2} = \boldsymbol{0}$
(b)
$\boldsymbol{\Phi}_{\mathrm{t},2} = \boldsymbol{0}$
$\boldsymbol{\Phi}_{\mathrm{r},2} = \boldsymbol{0}$
$\boldsymbol{\Psi}_{\mathrm{r},2} = \boldsymbol{0}$
(c)

Figure 5. Different cases for the kinematic constraints of the following chain: (**a**) wrong position and orientation; (**b**) correct position and wrong orientation; (**c**) all kinematic constraints are complied.

The orientation for the platform is given with the rotation matrix

$$ {}^{0}\boldsymbol{R}_L(\boldsymbol{q}_1) := {}^{0}\boldsymbol{R}_E(\boldsymbol{q}_1) \tag{26}$$

which gives the reference end-effector frame $\mathscr{F}_L$ resulting from the leading leg 1. The rotational part of the kinematic constraints

$$\boldsymbol{\Phi}_{\mathrm{r},j}(\boldsymbol{q}_j, \boldsymbol{q}_1) = \boldsymbol{\alpha}({}^{0}\boldsymbol{R}_L^{\mathrm{T}}(\boldsymbol{q}_1){}^{0}\boldsymbol{R}_E(\boldsymbol{q}_j)) \tag{27}$$

for the following leg is given by the Euler angle representation of the deviation between the orientation of the platform frame $\mathscr{F}_L$ given by the leading ("L") leg and the frame $\mathscr{F}_E$ given by the respective following leg j. The choice of the Euler angle convention is arbitrary. The full kinematic constraints for the complete parallel robot with m legs for 3T2R tasks

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_1^{\mathrm{T}} & \boldsymbol{\Phi}_2^{\mathrm{T}} & \cdots & \boldsymbol{\Phi}_m^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{28}$$

are assembled from the 3T2R constraints $\boldsymbol{\Psi}_1$ from (23) for the leading leg and the 3T3R constraints $\boldsymbol{\Phi}_j$, $2 \leq j \leq m$ from (25,27) for the following legs. The index "j" is used to distinguish the following legs of the 3T2R case and all legs "i" of the general case in Section 3. The full constraints of (28) lead to a 35-dimensional vector for the kinematic constraints for parallel robots with six legs in 3T2R tasks, which are aggregated as class I in Section 1.2. This formulation can be reduced by combining the mechanism-specific approach for the constraints from (6) or (8) with the principle of leading and following legs of this section. For the 6U$\underline{\text{P}}$S structure this would result in a 10-dimensional constraint vector with five entries for the leading leg and only one entry for each of the five following legs.

5. Differential Kinematics for Parallel Robots

To be able to compute the differential kinematics of the constraints $\boldsymbol{\Phi}$ (24) and $\boldsymbol{\Psi}$ (28) to

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}(\boldsymbol{q},\boldsymbol{x}) = \boldsymbol{\Phi}_{\partial q}\dot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\partial x}\dot{\boldsymbol{x}} = \boldsymbol{0} \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Psi}(\boldsymbol{q},\boldsymbol{\eta}) = \boldsymbol{\Psi}_{\partial q}\dot{\boldsymbol{q}} + \boldsymbol{\Psi}_{\partial \eta}\dot{\boldsymbol{\eta}} = \boldsymbol{0}, \tag{29}$$

the full inverse-kinematics matrices

$$\boldsymbol{\Phi}_{\partial q}(\boldsymbol{q},\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\Phi}_{1,\partial q_1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{2,\partial q_2} & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_{m,\partial q_m} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Psi}_{\partial q}(\boldsymbol{q},\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{\Psi}_{1,\partial q_1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Phi}_{2,\partial q_1} & \boldsymbol{\Phi}_{2,\partial q_2} & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{\Phi}_{m,\partial q_1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_{m,\partial q_m} \end{bmatrix} \tag{30}$$

and the full direct-kinematics matrices

$$\boldsymbol{\Phi}_{\partial x}(\boldsymbol{q},\boldsymbol{x}) = \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{\Phi}_{1,\partial x} \\ \boldsymbol{\Phi}_{2,\partial x} \\ \vdots \\ \boldsymbol{\Phi}_{m,\partial x} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Psi}_{\partial \eta}(\boldsymbol{q},\boldsymbol{\eta}) = \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{\eta}} = \begin{bmatrix} \boldsymbol{\Psi}_{1,\partial \eta} \\ \boldsymbol{\Phi}_{2,\partial \eta} \\ \vdots \\ \boldsymbol{\Phi}_{m,\partial \eta} \end{bmatrix} \tag{31}$$

have to be calculated for the 3T3R and the 3T2R case respectively. The gradient $\boldsymbol{\Phi}_{\partial q}$ has block diagonal form, indicating that the inverse kinematics problem can be solved for each leg independently. The structures of the gradient $\boldsymbol{\Psi}_{\partial q}$ results from the coupling of the leading and following joints in the rotational constraints equation.

The gradient matrices $\boldsymbol{\Phi}_{\partial q}$ and $\boldsymbol{\Phi}_{\partial x}$ contain nested nonlinear functions related to the orientation error. They are calculated with the chain rule and a syntax for stacking matrix columns to avoid differentiating matrices or with respect to matrices, which was introduced in [5]. The product operator

$\overline{\prod}$, the stacking operator $\overline{\boldsymbol{R}}$ and the transpose operator $\boldsymbol{P}_{\mathrm{T}}$ used for the implementation in the next section are explained in Appendix A.4.

5.1. Constraint Gradients for the Leading Leg of the 3T2R and All Legs of the 3T3R Case

The constraint definition $\boldsymbol{\Psi}_1$ for the leading leg of the 3T2R case (28) and $\boldsymbol{\Phi}_i$ with $i = 1, ..., m$ for all legs of the 3T3R case (24) are subject to the same model of (16), (19), (18). In the following, the 3T3R constraints are displayed. The form $\boldsymbol{\Psi}_1$ for the 3T2R case is obtained by removing the corresponding line of the rotational component according to (22) and replacing "$\boldsymbol{\Phi}$" by "$\boldsymbol{\Psi}$" in the following equations. For the analysis, the constraint gradient matrix w. r. t. the joint coordinates has to be divided out to

$$\boldsymbol{\Phi}_{1,\partial q_1} = \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{t},1,\partial q_1}^{\mathrm{T}} & \boldsymbol{\Phi}_{\mathrm{r},1,\partial q_1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{32}$$

where the translational component can be calculated with the geometric Jacobian of the leg chain, as derived in Appendix A.3. The rotational part is written down as a function composition of the three functions $\boldsymbol{\alpha}$ (Euler angles), $\overline{\prod}$ (matrix product) and ${}^0\boldsymbol{R}_E$ (rotation matrix) as

$$\boldsymbol{\Phi}_{\mathrm{r},1,\partial q_1} = \frac{\partial}{\partial \boldsymbol{q}_1}\boldsymbol{\alpha}\left({}^0\boldsymbol{R}_D^{\mathrm{T}}(\boldsymbol{x}){}^0\boldsymbol{R}_E(\boldsymbol{q}_1)\right) = \frac{\partial}{\partial \boldsymbol{q}_1}\boldsymbol{\alpha}\left(\overline{\prod}\left({}^0\overline{\boldsymbol{R}}_D^{\mathrm{T}}(\boldsymbol{x}), {}^0\overline{\boldsymbol{R}}_E(\boldsymbol{q}_1)\right)\right), \tag{33}$$

which is then expanded with the chain rule for differentiation and the stack operators to

$$\boldsymbol{\Phi}_{\mathrm{r},1,\partial q_1} = \underbrace{\frac{\partial \boldsymbol{\alpha}}{\partial \overline{\boldsymbol{R}}}}_{\mathrm{I} \in \mathbb{R}^{3\times 9}} \underbrace{\frac{\partial \overline{\prod}\left({}^0\overline{\boldsymbol{R}}_D^{\mathrm{T}}, {}^0\overline{\boldsymbol{R}}_E\right)}{\partial {}^0\overline{\boldsymbol{R}}_E}}_{\mathrm{II} \in \mathbb{R}^{9\times 9}} \underbrace{\frac{\partial {}^0\overline{\boldsymbol{R}}_E(\boldsymbol{q}_1)}{\partial \boldsymbol{q}_1}}_{\mathrm{III} \in \mathbb{R}^{9\times \dim(\boldsymbol{q}_1)}} \in \mathbb{R}^{3\times \dim(\boldsymbol{q}_1)}. \tag{34}$$

The two first partial derivatives from (34) are sparse matrices and can be calculated efficiently as shown in Appendix A.4. The factor "I" contains (A24) with $\overline{\boldsymbol{R}} = {}^D\overline{\boldsymbol{R}}_E(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{q}_1)$ and the factor "II" is (A27) with the contents of ${}^0\overline{\boldsymbol{R}}_D^{\mathrm{T}}(\boldsymbol{x}_{\mathrm{r}})$. The last partial derivative "III" can be derived with computer algebra systems from the analytic expression of the rotation matrix ${}^0\boldsymbol{R}_E(\boldsymbol{q}_1)$ or from the geometric Jacobian, see Appendix A.2. The leading legs constraint gradient matrix w. r t. the platform coordinates can be expanded in the same manner into

$$\boldsymbol{\Phi}_{1,\partial x} = \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{t},1,\partial x_{\mathrm{t}}} & \boldsymbol{\Phi}_{\mathrm{t},1,\partial x_{\mathrm{r}}} \\ \boldsymbol{\Phi}_{\mathrm{r},1,\partial x_{\mathrm{t}}} & \boldsymbol{\Phi}_{\mathrm{r},1,\partial x_{\mathrm{r}}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{\mathrm{r},1,\partial x_{\mathrm{r}}} \end{bmatrix}, \tag{35}$$

where the definitions from (16) and (19) only leave the rotational part

$$\begin{aligned} \boldsymbol{\Phi}_{\mathrm{r},1,\partial x_{\mathrm{r}}} &= \frac{\partial}{\partial \boldsymbol{x}_{\mathrm{r}}}\boldsymbol{\alpha}\left(\left({}^0\boldsymbol{R}_E^{\mathrm{T}}(\boldsymbol{q}_1){}^0\boldsymbol{R}_D(\boldsymbol{x}_{\mathrm{r}})\right)^{\mathrm{T}}\right) = \frac{\partial}{\partial \boldsymbol{x}_{\mathrm{r}}}\overline{\boldsymbol{\alpha}}\left(\boldsymbol{P}_{\mathrm{T}}\overline{\prod}\left({}^0\overline{\boldsymbol{R}}_E^{\mathrm{T}}(\boldsymbol{q}_1), {}^0\overline{\boldsymbol{R}}_D(\boldsymbol{x}_{\mathrm{r}})\right)\right) \\ &= \underbrace{\frac{\partial \overline{\boldsymbol{\alpha}}}{\partial \overline{\boldsymbol{R}}}}_{\mathrm{I} \in \mathbb{R}^{3\times 9}} \underbrace{\boldsymbol{P}_{\mathrm{T}}}_{\mathrm{II} \in \mathbb{R}^{9\times 9}} \underbrace{\frac{\partial \overline{\prod}\left({}^0\overline{\boldsymbol{R}}_E^{\mathrm{T}}, {}^0\overline{\boldsymbol{R}}_D\right)}{\partial {}^0\overline{\boldsymbol{R}}_D}}_{\mathrm{III} \in \mathbb{R}^{9\times 9}} \underbrace{\frac{\partial {}^0\overline{\boldsymbol{R}}_D(\boldsymbol{x}_{\mathrm{r}})}{\partial \boldsymbol{x}_{\mathrm{r}}}}_{\mathrm{IV} \in \mathbb{R}^{9\times 3}} \in \mathbb{R}^{3\times 3}, \end{aligned} \tag{36}$$

where the simplicity of the single expression "I"-"IV" is demonstrated in Appendix A.4. The factors are (A24) with $\overline{\boldsymbol{R}} = {}^D\overline{\boldsymbol{R}}_E(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{q}_1)$ in "I", the permutation matrix for transposition from (A23) in "II", (A27), where the contents of ${}^0\overline{\boldsymbol{R}}_E^{\mathrm{T}}$ are inserted in "III" and (A25) with the elements of $\boldsymbol{x}_{\mathrm{r}}$ for $\boldsymbol{\beta}$ in "IV".

5.2. Constraint Gradients for the Following Leg in the 3T2R Case

As explained regarding (24), the constraints (16) and (19) and their gradients (34) and (36) are used for all legs in the 3T3R case and the leading leg in the 3T2R case. For the following legs in the 3T2R case, the gradients $\boldsymbol{\Phi}_{j,\partial \boldsymbol{q}_1}$, $\boldsymbol{\Phi}_{j,\partial \boldsymbol{q}_j}$ and $\boldsymbol{\Phi}_{j,\partial \boldsymbol{x}}$ with $j = 2, ..., m$ from the right part of (30) and (31) have to be calculated in a similar way. Due to the absence of the platform orientation in (27), (35) simplifies for the following leg to

$$\boldsymbol{\Phi}_{j,\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \tag{37}$$

The gradient w. r. t. the joint coordinates of the following leg contains again the translational part of the legs Jacobian regarding the end-effector platform position in $\boldsymbol{\Phi}_{\mathrm{t},j,\partial \boldsymbol{q}_j}$ and has the rotational part

$$\begin{aligned}\boldsymbol{\Phi}_{\mathrm{r},j,\partial \boldsymbol{q}_j} &= \frac{\partial}{\partial \boldsymbol{q}_j}\boldsymbol{\alpha}\left({}^0\boldsymbol{R}_L^{\mathrm{T}}(\boldsymbol{q}_1){}^0\boldsymbol{R}_E(\boldsymbol{q}_j)\right) = \frac{\partial}{\partial \boldsymbol{q}_j}\boldsymbol{\alpha}\left(\overline{\prod}\left({}^0\overline{\boldsymbol{R}}_L^{\mathrm{T}}(\boldsymbol{q}_1), {}^0\overline{\boldsymbol{R}}_E(\boldsymbol{q}_j)\right)\right) \\ &= \underbrace{\frac{\partial \boldsymbol{\alpha}}{\partial \overline{\boldsymbol{R}}}}_{\mathrm{I}\in\mathbb{R}^{3\times 9}} \underbrace{\frac{\partial \overline{\prod}\left({}^0\overline{\boldsymbol{R}}_D^{\mathrm{T}}, {}^0\overline{\boldsymbol{R}}_E\right)}{\partial {}^0\overline{\boldsymbol{R}}_E}}_{\mathrm{II}\in\mathbb{R}^{9\times 9}} \underbrace{\frac{\partial {}^0\overline{\boldsymbol{R}}_E(\boldsymbol{q}_j)}{\partial \boldsymbol{q}_j}}_{\mathrm{III}\in\mathbb{R}^{9\times \dim(\boldsymbol{q}_j)}} \in \mathbb{R}^{3\times \dim(\boldsymbol{q}_j)},\end{aligned} \tag{38}$$

which is similar to the expression in (34). The factors of Equation (38) are (A24) with $\overline{\boldsymbol{R}} = {}^L\overline{\boldsymbol{R}}_E(\boldsymbol{q}_1, \boldsymbol{q}_j)$ in "I", (A27), where the elements of ${}^0\boldsymbol{R}_L^{\mathrm{T}}(\boldsymbol{q}_1)$ have to be inserted in "II" and the partial derivative of the platform orientation calculated from leg j w. r. t. the legs joint coordinates in "III", similar to term "III" from (34). The gradient w. r. t. the joint coordinates of the leading leg

$$\begin{aligned}\boldsymbol{\Phi}_{\mathrm{r},j,\partial \boldsymbol{q}_1} &= \frac{\partial}{\partial \boldsymbol{q}_1}\boldsymbol{\alpha}\left(\left({}^0\boldsymbol{R}_E^{\mathrm{T}}(\boldsymbol{q}_j){}^0\boldsymbol{R}_L(\boldsymbol{q}_1)\right)^{\mathrm{T}}\right) = \frac{\partial}{\partial \boldsymbol{q}_1}\boldsymbol{\alpha}\left(\boldsymbol{P}_{\mathrm{T}}\overline{\prod}\left({}^0\overline{\boldsymbol{R}}_E^{\mathrm{T}}(\boldsymbol{q}_j){}^0\overline{\boldsymbol{R}}_L(\boldsymbol{q}_1)\right)\right) \\ &= \underbrace{\frac{\partial \overline{\boldsymbol{\alpha}}}{\partial \overline{\boldsymbol{R}}}}_{\mathrm{I}\in\mathbb{R}^{3\times 9}} \underbrace{\boldsymbol{P}_{\mathrm{T}}}_{\mathrm{II}\in\mathbb{R}^{9\times 9}} \underbrace{\frac{\partial \overline{\prod}\left({}^0\overline{\boldsymbol{R}}_E^{\mathrm{T}}, {}^0\overline{\boldsymbol{R}}_L\right)}{\partial {}^0\overline{\boldsymbol{R}}_L}}_{\mathrm{III}\in\mathbb{R}^{9\times 9}} \underbrace{\frac{\partial {}^0\overline{\boldsymbol{R}}_L(\boldsymbol{q}_1)}{\partial \boldsymbol{q}_1}}_{\mathrm{IV}\in\mathbb{R}^{9\times \dim(\boldsymbol{q}_1)}} \in \mathbb{R}^{3\times \dim(\boldsymbol{q}_1)}.\end{aligned} \tag{39}$$

is similar to the expression in (36). The order of the residual expression (27) has to be switched by exploiting the associative property of matrix transposition to avoid differentiating a transposed matrix. The factors of equation (39) are (A24) with $\overline{\boldsymbol{R}} = {}^L\overline{\boldsymbol{R}}_E(\boldsymbol{q}_1, \boldsymbol{q}_j)$ in "I", the permutation matrix for transposition from (A23) in "II", (A27) with ${}^0\overline{\boldsymbol{R}}_E^{\mathrm{T}}(\boldsymbol{q}_j)$ in "III" and the term "III" from (34) in "IV".

5.3. Gradient-Based Solution of the Inverse Kinematics Problem with Redundancy Resolution

The presented kinematic constraints and their gradient matrices can be used to solve the inverse kinematics problem (IKP) of single leg chains and complete parallel robots. Since all active and passive joint angles are involved for the case of parallel robots, solving their IKP results in solving the IKP for all leg chains. As first introduced in [7] for Euler angle residuals in the IKP, the Taylor series expansion of $\boldsymbol{\Phi}(\boldsymbol{q}, \boldsymbol{x})$ leads to the definition of

$$\boldsymbol{\Phi}(\boldsymbol{q}^{k+1}, \boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{q}^k, \boldsymbol{x}) + \frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{\Phi}(\boldsymbol{q}, \boldsymbol{x})\bigg|_{\boldsymbol{q}^k}(\boldsymbol{q}^{k+1} - \boldsymbol{q}^k) \tag{40}$$

in an iterative algorithm at the step $k+1$, which can be used to solve the IKP using a given initial value $\boldsymbol{q}^0$ and the condition

$$\boldsymbol{\Phi}(\boldsymbol{q}^{k+1}, \boldsymbol{x}) = \boldsymbol{0}. \tag{41}$$

Defining the solution of the IKP as the main task ("T"), the stepwise solution for the joint coordinates results to

$$\Delta \boldsymbol{q}^k = \Delta \boldsymbol{q}_{\mathrm{T}}^k = \boldsymbol{q}^{k+1} - \boldsymbol{q}^k = -\boldsymbol{\Phi}_{\partial \boldsymbol{q}}^{\dagger}(\boldsymbol{q}^k, \boldsymbol{x})\boldsymbol{\Phi}(\boldsymbol{q}^k, \boldsymbol{x}). \tag{42}$$

Depending on the dimension, $(\cdot)^{\dagger}$ denotes the matrix inverse or the pseudo-inverse. Again, the Equations (40)–(42) can be written with "$\boldsymbol{\Psi}$" from (28) instead of "$\boldsymbol{\Phi}$" from (24) for the 3T2R case.

In the latter case, the corresponding gradient matrix $\boldsymbol{\Psi}_{\partial q}(\boldsymbol{q}, \boldsymbol{\eta})$ from (30) allows defining of a nullspace in the case of $\dim(\boldsymbol{q}_1) > \dim(\boldsymbol{\eta})$. This redundancy can be exploited by using the nullspace ("N") projection $\Delta\boldsymbol{q}_{\mathrm{N}}$ from [6] additionally to the solution $\Delta\boldsymbol{q}_{\mathrm{T}}$ of the IKP in (42) with the new increment

$$\Delta\boldsymbol{q}^k = \boldsymbol{q}^{k+1} - \boldsymbol{q}^k = \Delta\boldsymbol{q}_{\mathrm{T}}^k + \Delta\boldsymbol{q}_{\mathrm{N}}^k = -\boldsymbol{\Psi}_{\partial q}^{\dagger}\boldsymbol{\Psi} + (\mathbf{1} - \boldsymbol{\Psi}_{\partial q}^{\dagger}\boldsymbol{\Psi}_{\partial q})h_{\partial q} \tag{43}$$

in the iterative algorithm. The optimization of additional performance criteria h requires their gradient $h_{\partial q}$ w. r .t the joint positions. One criterion is the summed $\boldsymbol{W}_1$-weighted quadratic distance

$$h_1(\boldsymbol{q}) = \frac{1}{2}(\boldsymbol{q} - \bar{\boldsymbol{q}})^{\mathrm{T}}\boldsymbol{W}_1(\boldsymbol{q} - \bar{\boldsymbol{q}}), \quad h_{1,\partial q} = \frac{\partial h_1}{\partial \boldsymbol{q}} = \boldsymbol{W}_1(\boldsymbol{q} - \bar{\boldsymbol{q}}) \tag{44}$$

of the joint positions $\boldsymbol{q}$ from their respective reference position $\bar{\boldsymbol{q}}$, e.g., used in [2,36]. Defining $\bar{\boldsymbol{q}}$ to be in the middle of the joint limits and minimizing $h_1(\boldsymbol{q})$ reduces the risk of joints reaching their technical limits, but does not guarantee it, since exceeding the limit for one joint can be compensated by improving other joints. The $\boldsymbol{W}_2$-weighted hyperbolic joint limit distance

$$h_2(\boldsymbol{q}) = \frac{1}{n}\sum_{i=1}^{n} w_{2,i}\frac{(q_{i,\max} - q_{i,\min})}{8}\left(\frac{1}{(q_i - q_{i,\min})^2} + \frac{1}{(q_i - q_{i,\max})^2}\right) \tag{45}$$

from [8] (written element-wise for $n = \dim(\boldsymbol{q})$) circumvents this problem by generating infinitely high values when reaching the limits. In contrast to h_1, the criterion h_2 is only defined for joints within their limits with $q_{i,\min} < q_i < q_{i,\max}$, which is ensured by setting $w_{2,i} = 0$ for joints exceeding their limits and $w_{2,i} = 1$ otherwise. To combine the effect of drawing joint positions to their middle with h_1 of (44) and of strongly rejecting joints directly near their limits with h_2 of (45), their weighted sum

$$h_3(\boldsymbol{q}) = K_{h_1}h_1(\boldsymbol{q}) + K_{h_2}h_2(\boldsymbol{q}) \tag{46}$$

is used in the simulation studies of Section 6. Other criteria not related to the joint limits are for example stiffness [9] or singularity avoidance via Frobenius norm condition number [8] or squared condition number [4]. The method can be used for serial-link robots as well by removing all entries for the following legs from the formulas, as presented in [5]. The platform pose $\boldsymbol{x}/\boldsymbol{\eta}$ corresponds to the desired pose for the serial robots end-effector and the kinematic constraints $\boldsymbol{\Phi}/\boldsymbol{\Psi}$ correspond to the residual of the IKP.

In the practical implementation, it has proven to be useful to extend the basic principle of (43) to

$$\Delta\boldsymbol{q}^k = K_{\mathrm{Lim}}(\boldsymbol{q}^k)K_{\mathrm{Rel}}(\boldsymbol{q}^k)(K_{\mathrm{T}}\Delta\boldsymbol{q}_{\mathrm{T}}^k + K_{\mathrm{N}}\Delta\boldsymbol{q}_{\mathrm{N}}^k), \tag{47}$$

where the constant damping coefficients K_{T} for $\Delta\boldsymbol{q}_{\mathrm{T}}^k$ and K_{N} for $\Delta\boldsymbol{q}_{\mathrm{N}}^k$ were introduced to avoid overshooting of the solution for the prize of slower convergence. The damping term K_{N} has to be chosen according to the optimization criterion. Further damping was introduced for the 3T2R case with task redundancy to reduce a $\Delta\boldsymbol{q}^k$ that would lead to overshoot over the joint limits with $K_{\mathrm{Lim}}(\boldsymbol{q}^k)$. The value $K_{\mathrm{Lim}} = 1$ is set if no limits would be violated by the increment $\Delta\boldsymbol{q}^k$. For the 3T3R case, $K_{\mathrm{Lim}} := 1$ is set permanently, since slowing down when approaching the limits does not change the direction of the increment and violating the limits is inevitable. The maximum step size for one iteration $\Delta\boldsymbol{q}^k$ was ensured with $K_{\mathrm{Rel}}(\boldsymbol{q}^k)$ to stay below 5% of the joint limit range to prevent leaving the validity of the first-order approximation of (40). For smaller increments, $K_{\mathrm{Rel}} = 1$ holds. The damping terms are always applied to the full vector and not to single elements and therefore only change the norm and not the direction of $\Delta\boldsymbol{q}^k$.

5.4. Differential Kinematics for the Parallel Robot and Its Applications

The reasoning so far only considered the inverse kinematics of the parallel robot. The kinematic definitions can also be used in the differential kinematics (29) to establish the connection between joint and platform velocity. This was already presented in general form in [15] and also corresponds to the theory of linear transformation which is the base of the works of Gogu on structural synthesis [11]. The derivation in this paper is based on the position *and* orientation, but comes to the same result as the already-existing velocity-based approach. Furthermore, using $\boldsymbol{\Psi}/\boldsymbol{\eta}$ as elaborated before allows for the first time defining differential kinematics specifically for 3T2R tasks in a general form. The differential equation of (29) is expanded to

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{a}}, \boldsymbol{q}_{\mathrm{p}}, \boldsymbol{x}) = \boldsymbol{\Phi}_{\partial q_{\mathrm{a}}}\dot{\boldsymbol{q}}_{\mathrm{a}} + \boldsymbol{\Phi}_{\partial q_{\mathrm{p}}}\dot{\boldsymbol{q}}_{\mathrm{p}} + \boldsymbol{\Phi}_{\partial x}\dot{\boldsymbol{x}} = \boldsymbol{0} \tag{48}$$

to distinguish active (“a”) and passive (“p”) joints. The latter also contain the coordinates of the platform-connecting joints. Reordering the equation leads to the full inverse differential kinematics

$$\boldsymbol{\Phi}_{\partial x}\dot{\boldsymbol{x}} = -\begin{bmatrix}\boldsymbol{\Phi}_{\partial q_{\mathrm{a}}} & \boldsymbol{\Phi}_{\partial q_{\mathrm{p}}}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{q}}_{\mathrm{a}} \\ \dot{\boldsymbol{q}}_{\mathrm{p}}\end{bmatrix} = \boldsymbol{\Phi}_{\partial \mathrm{ap}}\begin{bmatrix}\dot{\boldsymbol{q}}_{\mathrm{a}} \\ \dot{\boldsymbol{q}}_{\mathrm{p}}\end{bmatrix}, \qquad \begin{bmatrix}\dot{\boldsymbol{q}}_{\mathrm{a}} \\ \dot{\boldsymbol{q}}_{\mathrm{p}}\end{bmatrix} = -\boldsymbol{\Phi}_{\partial \mathrm{ap}}^{-1}\boldsymbol{\Phi}_{\partial x}\dot{\boldsymbol{x}}, \tag{49}$$

which has been addressed in the previous sections, and the full direct differential kinematics

$$\boldsymbol{\Phi}_{\partial q_{\mathrm{a}}}\dot{\boldsymbol{q}}_{\mathrm{a}} = -\begin{bmatrix}\boldsymbol{\Phi}_{\partial x} & \boldsymbol{\Phi}_{\partial q_{\mathrm{p}}}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{x}} \\ \dot{\boldsymbol{q}}_{\mathrm{p}}\end{bmatrix} = \boldsymbol{\Phi}_{\partial x\mathrm{p}}\begin{bmatrix}\dot{\boldsymbol{x}} \\ \dot{\boldsymbol{q}}_{\mathrm{p}}\end{bmatrix}, \qquad \begin{bmatrix}\dot{\boldsymbol{x}} \\ \dot{\boldsymbol{q}}_{\mathrm{p}}\end{bmatrix} = -\boldsymbol{\Phi}_{\partial x\mathrm{p}}^{-1}\boldsymbol{\Phi}_{\partial q_{\mathrm{a}}}\dot{\boldsymbol{q}}_{\mathrm{a}}. \tag{50}$$

By only selecting the first rows for $\dot{\boldsymbol{q}}_{\mathrm{a}}$ in (49) and for $\dot{\boldsymbol{x}}$ in (50), the well-known analytic Jacobian of the parallel robot (called “Euler angles Jacobian matrix” in [15] and “design Jacobian” in [11]), relating actuator velocities $\dot{\boldsymbol{q}}_{\mathrm{a}}$ and platform velocities $\dot{\boldsymbol{x}}$, can be obtained from both equations. For the case of task or kinematic redundancy, the pseudo-inverse can be used for $\boldsymbol{\Phi}_{\partial \mathrm{ap}}$ in (49) together with a nullspace optimization as shown in Section 5.3. This allows exploiting the functional redundancy in 3T2R tasks also for trajectories $\boldsymbol{\eta}(t)$, $\dot{\boldsymbol{\eta}}(t)$ and not only for single poses $\boldsymbol{\eta}$. When solving the IKP for a trajectory, the IK on position level of Section 5.3 is only needed to correct linearization errors. The case of task redundancy does not affect (50), since the full platform velocity $\dot{\boldsymbol{x}}$ is obtained from given actuator velocities $\dot{\boldsymbol{q}}_{\mathrm{a}}$.

6. Results

To evaluate the inverse kinematics (IK) algorithm presented in the previous Section 5.3, first the solution of the IKP is shown for the trajectory of a serial-link industrial robot in Section 6.1 and for the trajectory of a parallel robot in Section 6.2. The results are generalized by the statistical analysis of random point-to-point movements of arbitrary serial link chains in Section 6.3.

6.1. Resolution of Functional Redundancy of a Serial-Link Six-DoF Robot in 3T2R tasks

The first evaluation of the IK algorithm from Section 5.3 is performed with simulations at the basic example of a six-DoF industrial robot with a 500 mm × 800 mm rectangular trajectory centered at 1200 mm/−200 mm/200 mm with a constant orientation of the z-axis pointing into the ground plane. The manipulator Fanuc M-710 iC/50 was taken from the example of [36] with the tabulated kinematics parameters and a sketch of the trajectory in Figure 6. Deviations in the parameters relative to [36] result from the use of the *modified* Denavit–Hartenberg notation from [35] for the joint transformations and the use of only positive axis alignment parameters α_i for consistency with the results of the structural synthesis from [34].

i	α_i R_x	d_i T_x	θ_i R_z	r_i T_z	$q_{i,\min}$	$q_{i,\max}$
1	0	0	q_1	0	−180°	180°
2	90°	150 mm	q_2	0	30°	165°
3	0	870 mm	q_3	0	−132°	230°
4	90°	170 mm	q_4	1016 mm	−360°	360°
5	90°	0	q_5	0	−125°	125°
6	90°	0	q_6	-175 mm	−360°	360°

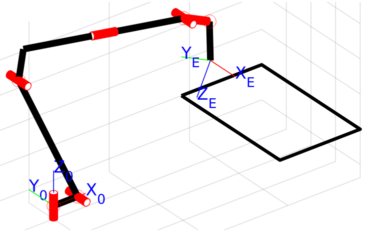


Figure 6. Left: Table with the kinematic parameters of the industrial manipulator Fanuc M-710 iC/50. **Right**: sketch of the robot scenario.

The IKP is solved with two settings: setting the tool axis rotation to different constant values β_3 with the 3T3R algorithm and solving the IKP only for the desired pointing direction with the 3T2R algorithm. The algorithm from (43) was used in the extended version of (47) for both cases with different settings caused by their nature. For the 3T3R case, $K_\mathrm{T} = 0.7$ and $K_\mathrm{N} = 0.7$ were set. The terms K_N and K_Lim have no effect, since no nullspace movement is possible. For the 3T2R case, with $K_{h_1} = 0$ and $K_{h_2} = 1$ only the hyperbolic limit rejection criterion from (45) was used. The first criterion was not used, since in the trajectory example the limits are not even temporarily exceeded by principle. All IKP algorithms had the same initial value from Figure 6.

The results of the inverse kinematics for different settings are given in Figure 7, where the representative joint coordinates q_1 and q_5, the redundant coordinate of the end-effector orientation β_3, as well as the optimization criterion (45) are depicted over time for the trajectory from Figure 6. The positions are normalized to the joint limits from −1 to 1. The first three lines in Figure 7 represent IKP solutions with a given constant end-effector orientation β_3 of −150°, −15° and 45° and the 3T3R algorithm. The 3T2R algorithm without nullspace optimization is plotted with dotted lines for each first sample of the 3T3R cases as initial value with the same colors. Using these initial values for a 3T2R IK with optimization leads to strong nullspace movements at the beginning, quickly converging to a local minimum. Therefore, the 3T2R case with optimization, plotted as the green line with triangle markers, is shown only for the initial condition from Figure 6. It can be observed that the optimization of the criterion leads to the best solution of the IKP. The lines for the criterion for $\beta_3 = -150°$ and $\beta_3 = -15°$ partly exceed the limits of the plot, indicating that the limit is violated, which can also be seen at the plot for q_5. This exposes the need for keeping the solution always within the limits by the measures described. The 3T2R IK without optimization with dotted lines tends to lower changes in the joint positions than the 3T3R IK, since this corresponds to the solution of the matrix pseudo-inverse in (43).

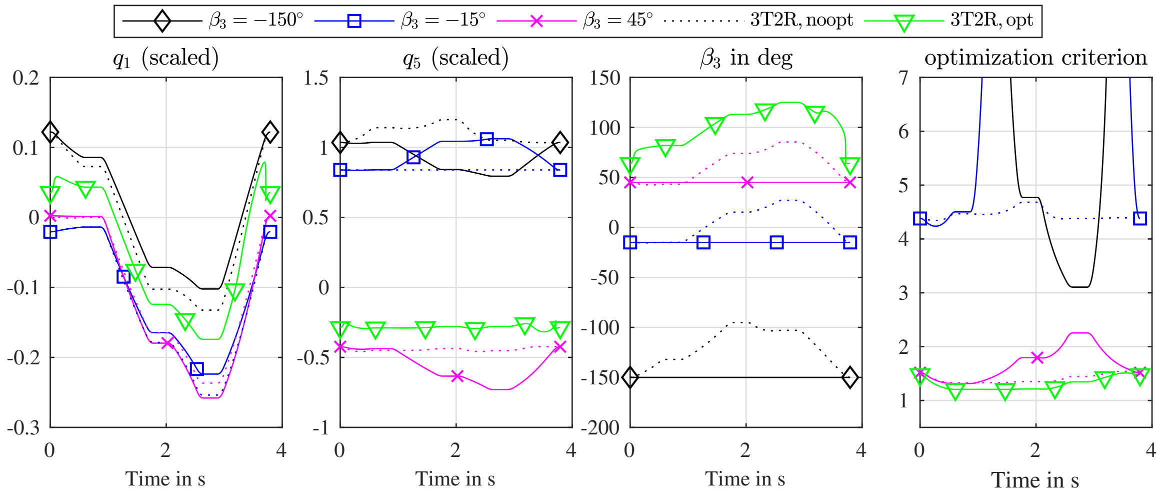


Figure 7. Results of the inverse kinematics with different settings for the trajectory of Figure 6.

6.2. Resolution of Functional Redundancy of a Parallel Robot in 3T2R Tasks

As elaborated in Section 4 and 5, the solution of the IKP for 3T2R and 3T3R tasks is necessary to solve the problem for parallel robots. Therefore, the trajectory evaluation for a 6UPS parallel robot in this section is preceded by the trajectory example for a serial link chain in the previous section. This robot belongs to the first class presented in Section 1.2, which is primarily addressed in this paper.

The robot has a Gough structure [15] with symmetric alignment of the universal joint base couplings on a circle with radius $\|\boldsymbol{r}_{0A_i}\| = 1\,\mathrm{m}$ and the spherical joint platform couplings on a circle with radius $\|\boldsymbol{r}_{B_iE}\| = 0.4\,\mathrm{m}$. The initial pose was set to a center position $\boldsymbol{x}_\mathrm{t}^\mathrm{T} = [0, 0, 0.5\,\mathrm{m}]$ and the initial orientation $\boldsymbol{x}_\mathrm{r}$ was set to zero, meaning an alignment of base and platform frame. The joint positions for each leg were defined to have the initial values $\boldsymbol{q}_i^\mathrm{T} = [30°, -30°, 0.583\,\mathrm{m}, 0°, 30°, 60°]$ for the given initial platform pose to avoid switching $\pm\pi$ within the trajectory and to avoid gimbal lock singularities. The joint limits were set around the resulting zero position to $\pm 0.5\,\mathrm{m}$ for the prismatic joint and $\pm 60°$ for all single revolute joints representing the universal and spherical joints. The values are higher than typical values for real robots to emphasize the effect of the nullspace movement in a bigger simulated workspace of the robot. The settings for the IK solver are similar to those in Section 6.1, since both cases regard solving the IKP for a trajectory.

The time evolution of platform pose and optimization criteria is depicted in Figure 8. The reference trajectory can be seen at the platform position in Figure 8a and the platform orientation expressed in XYZ Euler angles (β_1-β_3) relative to the base frame in Figure 8b. The IKP is solved using two different methods: Only solving the IKP for the legs separately, called “ser. IK” in Figure 8 and solving the IKP for all legs together, called “par. IK” in Figure 8. Both methods perform an optimization with only h_2 of (45), as justified in Section 6.1. The first approach only performs this optimization according to Section 3 for the first leg using the 3T2R method and then solves the IKP for all other legs with the 3T3R method. The second approach uses the optimization for all legs together according to the 3T2R method from Section 4. This results in improved values for the performance criteria depicted for h_1 in Figure 8c and for h_2 in a logarithmic scale in Figure 8d. Since the first approach does not regard the limits of the following legs, the optimization criterion gives high values indicating many joint limit violations. The second approach only shows peaks at $t = 1.5\,\mathrm{s}$ in Figure 8d that result from a joint position getting near to the limit, but not exceeding it. For the practical implementation, the computation time is only weakly influenced by the selection of the method, since calculating the (pseudo-)inverse for six 5×6 and 6×6 matrices or one 35×36 matrix does not present a challenge for current computing hardware. Therefore, the “par. IK” method should be preferred.

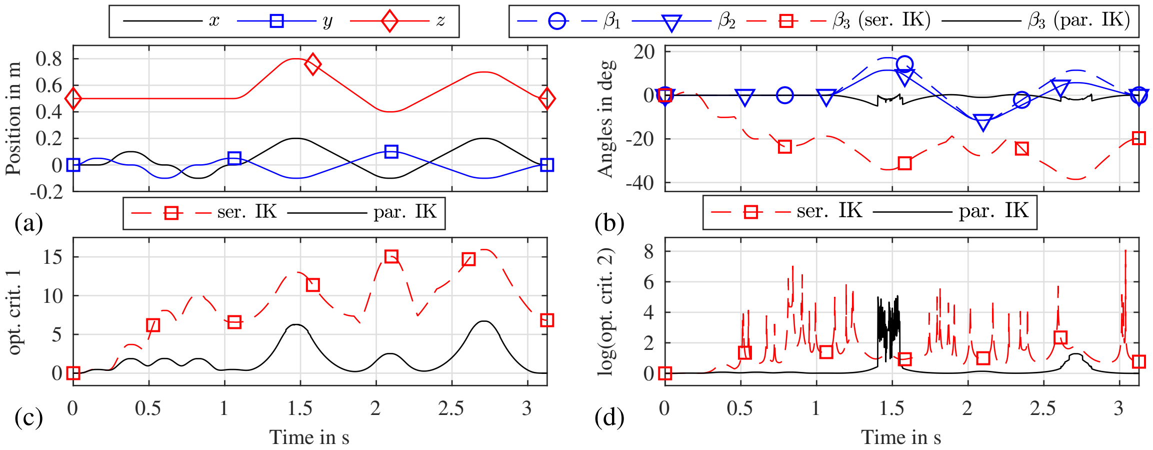


Figure 8. Results of the inverse kinematics of a 6UPS robot in a 3T2R task. (**a**) platform positions, (**b**) platform orientation in Euler angles, (**c**,**d**) optimization criteria $h_1(\boldsymbol{q})$ and $h_2(\boldsymbol{q})$.

6.3. Statistic Results for the Inverse Kinematics of Serial Link Chains

To emphasize the generality of the presented approach, the inverse kinematics is solved for a set of 309 serial kinematic chains with six joints. This set of six-DoF kinematics is generated by permutations of their Denavit–Hartenberg parameters and is reduced with the isomorphism detection of [34] to a minimal set, representing all possible six-DoF serial kinematics with full mobility. The approach is similar to the results of the evolutionary morphology of parallel robot leg chains of [11]. In contrast to the trajectory evaluations in the previous sections focusing on nullspace movement, the inverse kinematics is solved in this section for arbitrary reachable poses of the serial chain in its individual workspace. Therefore, different settings proved to be necessary, since for point-to-point movements, intermediate steps may be outside of the joint limits. In the trajectory case, the initial value for the IKP of the continuous trajectory is always very close to the desired pose of the next trajectory sample. Preventing the algorithm completely from leaving the allowed joint positions reduces the IK success rate. Therefore, the damping term for limit violation was not used in this evaluation, resulting in a constant $K_{\mathrm{Lim}}(\boldsymbol{q}) = 1$ in (47). To reach again an allowed configuration when approaching the goal pose from intermediate steps with limit violations, the combined criterion $h_3(\boldsymbol{q})$ from (46) with $K_{h1} = 0.99$ and $K_{h2} = 0.01$ was used. Further empirically determined values for all different serial chains were the damping coefficients $K_{\mathrm{T}} = 0.6$ and $K_{\mathrm{N}} = 0.01$ in (47). Since these values provide good results for all serial chains with random geometric parameters and for random configurations, they can be regarded as a good choice generally.

To create a general evaluation case, the poses for testing the IK algorithm were generated by the forward kinematics of 50 different joint configurations of the chains uniformly distributed between the joint limits of $\pm\pi$ for revolute joints and $\pm 0.5\,\mathrm{m}$ for prismatic joints. Additionally, the Denavit–Hartenberg parameters were set to 50 different sets of uniformly distributed parameters between 0 and 1 meters or radians resulting in 2500 combinations for each of the 309 chains in total. The initial value $\boldsymbol{q}^0$ for the solution of the IKP of (47) was set to random values from a uniform distribution within the joint limits. The inverse kinematics was calculated for the full pose with the 3T3R algorithm and only using the pointing direction together with the resolution of functional redundancy in the 3T2R algorithm. A maximum of 15 tries with random initial values was allowed to search for a solution of the IKP within the limits. After that, five more tries were allowed to find a solution violating the limits, but presenting a solution of the IKP to be able to distinguish the two cases, which allows further reasoning on the functionality and possible improvements. A success of the IK is defined as a solution within the joint limits.

The aggregated results are presented as histograms in Figure 9 for different settings of the algorithm. The histograms show that for the worst case in 3T2R (3T3R) tasks, the success rate is 87% (6%), marked by the position of the first bars in Figure 9a,c. These results can be vastly improved by setting the initial guess $\boldsymbol{q}^0$ within 20% (w. r. t. the joint limit range) around the pose, from which the desired end-effector pose has been calculated. This improves the worst success rate of all kinematic chains to 98% for 3T2R (Figure 9b) and 95% for 3T3R tasks (Figure 9d).

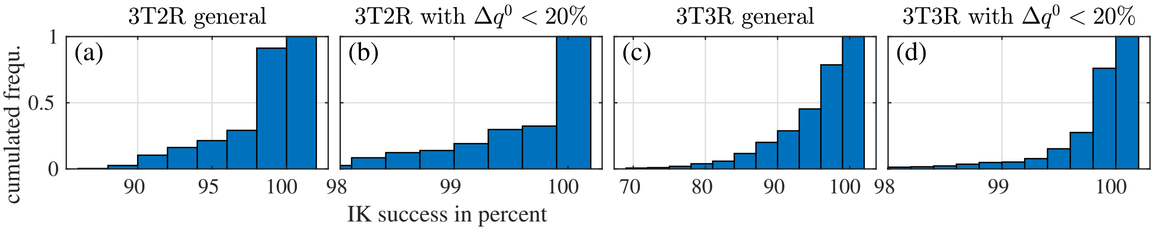


Figure 9. Histograms with cumulated frequency of the IK success for all kinematic chains with different settings: 3T2R tasks (**a**,**b**) vs. 3T3R tasks (**c**,**d**) and arbitrary initial value (**a**,**c**) vs. initial near goal pose.

A detailed investigation on the success rates of all possible serial chains is performed in Figure 10.

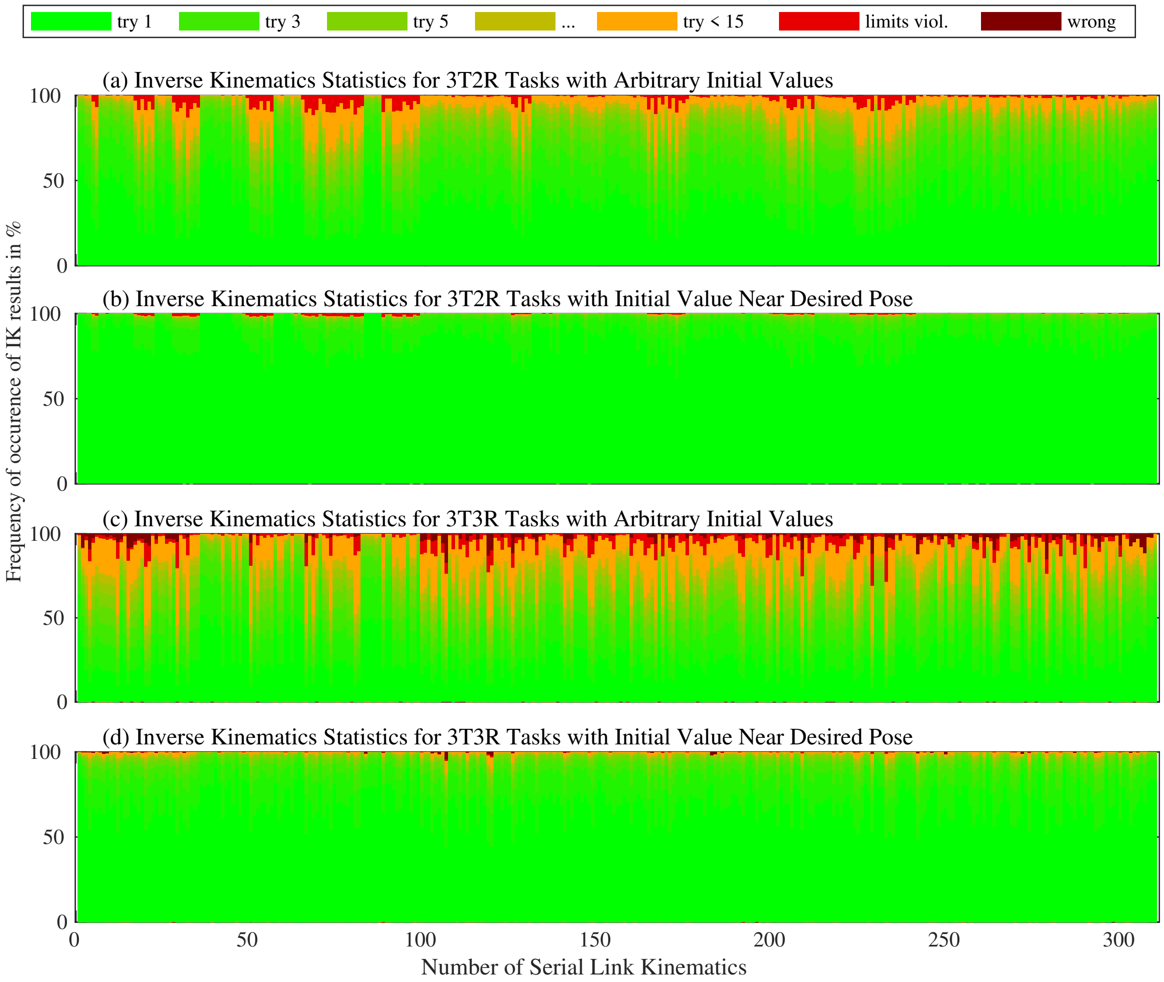


Figure 10. Detailed Statistics of the success of the inverse kinematics algorithm for 3T2R tasks (**a**,**b**) and 3T3R tasks (**c**,**d**). The Success of the IK solver is shown in different shades of green for increasing numbers of required tries. Different initial values $\boldsymbol{q}^0$ are distinguished in (**a**,**c**) and (**b**,**d**).

The 309 serial kinematics are sorted according to their number of revolute joints and are listed on the horizontal axis of the figure: They contain three revolute joints up to no. 98, four R joints up to no. 240, and five R joints up to no. 301. The eight structures from 302 to 309 with six R joints differ in the parallelism of their joint axes. The first 240 kinematic chains with more than one prismatic joint can be seen as a rather academic example and are listed for the sake of completeness. The most prominent chains are the UPS chain from Section 6.2 at no. 266 and the six-DoF industrial robot from Section 6.1 at no. 309. Each bar represents the stacked relative frequency of the IK result state in percent for one kinematic chain. The result state is defined as the number of tries or the success. All bars add up to 100%, which corresponds to the 2500 configurations per chain. Beginning at the bottom, the number of tries necessary for the solution of the IKP is marked with colors from green to orange. Only cases with a violation of the limits (bright red) or wrong position (dark red) correspond to a failure of the algorithm, which has been addressed in the analysis of Figure 9, representing an aggregated form of Figure 10. The subfigures a–d of Figure 10 correspond to the ones in Figure 9. It can be observed that the quality of the results is clustered according to the kinematic groups. Structures with at most one P joint show a considerably better performance of the algorithm with a worst success rate of 97.16% for five R joints and 99.36% for six R joints for the 3T2R case (a), which can be seen at the very small red top parts of the bars in the corresponding range of the diagram. The worse performance of the 3T3R algorithm, mostly caused by limit violations, can be explained by joints changing their configuration, i.e., from “elbow up” to “elbow down”, which causes limit violations but does not affect the 3T3R IK.

7. Discussion

A general kinematics model for parallel robots was introduced to solve the inverse kinematics problem for any kind of parallel robot in tasks with one redundant rotational DoF (3T2R). The prize of the generality of the approach is the increased size of the inverse-kinematics matrices, which is 10 × 11 for robots with a simple UPS structure instead of 6 × 6 for the non-redundant kinematics and grow up to 35 × 36 for general task redundant parallel robots with full mobility. This makes symbolic calculations of the kinematic matrices impossible, allowing only studies on mobility, singularities and other properties of the Jacobian matrix based on numeric calculations. The application of the proposed method can therefore be seen mainly in finding optimal trajectories for task redundant parallel robots in milling or drilling scenarios regarding stiffness, dexterity or joint limits. Due to the performance of the method demonstrated at exemplary cases, an online implementation is possible but has to be proved in future works to converge in real-time conditions for specific machines. The generality of the approach allows using it in a combined structural and dimensional synthesis sketched in Figure 2, extending the purely structural synthesis of parallel robot kinematics from [11,31] to a dimensional synthesis of all structures as shown in [34] for serial robots.

Author Contributions: conceptualization, M.S. and S.T.; methodology, M.S.; software, M.S.; validation, M.S.; formal analysis, M.S.; writing—original draft preparation, M.S.; writing—review and editing, M.S., S.T., T.O.; supervision, S.T. and T.O.; project administration, T.O.; funding acquisition, S.T. and T.O.

Funding: The financial support from the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) under grant number OR 196/33-1 is gracefully acknowledged.

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

Abbreviations

The following abbreviations are used in this manuscript:

PKM	parallel kinematic machine (parallel robot)
IKP	inverse kinematics problem
DoF	degrees of freedom
xTyR	x translational and y rotational degrees of freedom

Appendix A. Mathematical Symbols for Reciprocal Euler Angles in Inverse Kinematics

The following appendix contains additional detailed information about the kinematic constraint formulation of this paper. Appendix A.1 contains a mathematical proof for the properties of reciprocal Euler angles in inverse kinematics, which is only outlined in equ. 18 of [5]. The relations of the partial derivatives to the geometric and analytic Jacobian are derived in Appendixes A.2 and A.3. The matrix operations for the partial derivatives are replicated from [5] in Appendix A.4 and the contents of the single partial derivatives are given in Appendix A.5 to facilitate the understanding and implementation by the reader.

Appendix A.1. Proof for the Properties of Reciprocal Euler Angles

This section derives the effect of the reciprocity of Euler angles at the example of the kinematics description of Section 3 and the frames of Figure 4b: An end-effector orientation $\boldsymbol{\beta} = \boldsymbol{x}_\mathrm{r}$ gives the rotation matrix

$$ {}^{D}\boldsymbol{R}_{E}(\boldsymbol{\beta},\boldsymbol{q}) = {}^{0}\boldsymbol{R}_{D}^{\mathrm{T}}(\boldsymbol{\beta}){}^{0}\boldsymbol{R}_{E}(\boldsymbol{q}), \tag{A1} $$

which rotates vectors from the actual end-effector frame $\mathscr{F}_E$ to the desired or platform end-effector frame $\mathscr{F}_D$. Using the XYZ Euler angles and exploiting the properties of SO(3) rotation matrices yields

$$ {}^{0}\boldsymbol{R}_{D}^{\mathrm{T}}(\boldsymbol{\beta}) = \boldsymbol{R}_z(-\beta_3)\boldsymbol{R}_y(-\beta_2)\boldsymbol{R}_x(-\beta_1), \tag{A2} $$

as introduced in (5). With an additional rotation $-\delta$ around the z-axis for the desired orientation, the resulting new Euler angles $\boldsymbol{\beta}'$ are

$$\beta'_1 = \beta_1, \quad \beta'_2 = \beta_2, \quad \beta'_3 = \beta_3 - \delta. \tag{A3}$$

The additional rotation corresponds to the tool axis defined in Section 3 and leads to a new residual orientation error expressed as a rotation matrix

$$\begin{aligned} {}^{D}\boldsymbol{R}_E(\boldsymbol{\beta}', \boldsymbol{q}) &= {}^{0}\boldsymbol{R}_D^{\mathrm{T}}(\boldsymbol{\beta}')\,{}^{0}\boldsymbol{R}_E(\boldsymbol{q}) \\ &= \left({}^{0}\boldsymbol{R}_D(\boldsymbol{\beta})\boldsymbol{R}_z(-\delta)\right)^{\mathrm{T}}\,{}^{0}\boldsymbol{R}_E(\boldsymbol{q}) \\ &= \boldsymbol{R}_z(\delta)\,{}^{0}\boldsymbol{R}_D^{\mathrm{T}}(\boldsymbol{\beta})\,{}^{0}\boldsymbol{R}_E(\boldsymbol{q}) \\ &= \boldsymbol{R}_z(\delta)\,{}^{D}\boldsymbol{R}_E(\boldsymbol{\beta}, \boldsymbol{q}). \end{aligned} \tag{A4}$$

The first residual orientation error from (A1) corresponding to $\boldsymbol{\beta}$ is defined as a rotation matrix

$${}^{D}\boldsymbol{R}_E(\boldsymbol{\beta}, \boldsymbol{q}) = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{A5}$$

and as a ZYX Euler angle representation

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \operatorname{arctan2}\left(n_y, n_x\right) \\ \operatorname{arctan2}\left(-n_z, \sqrt{{a_z}^2 + {o_z}^2}\right) \\ \operatorname{arctan2}\left(o_z, a_z\right) \end{bmatrix}. \tag{A6}$$

Using the sign-aware operator $\operatorname{arctan2}(y, x)$ instead of $\arctan(y/x)$ allows angles to be in $(-\pi, +\pi]$, removes ambiguities and provides global differentiability. The second residual corresponding to $\boldsymbol{\beta}'$ only differs regarding the additional rotation δ. Combining (A4) and (A5) leads to

$${}^{D}\boldsymbol{R}_E(\boldsymbol{\beta}', \boldsymbol{q}) = \begin{bmatrix} n'_x & o'_x & a'_x \\ n'_y & o'_y & a'_y \\ n'_z & o'_z & a'_z \end{bmatrix} = \begin{bmatrix} C_\delta\, n_x - S_\delta\, n_y & C_\delta\, o_x - S_\delta\, o_y & C_\delta\, a_x - S_\delta\, a_y \\ C_\delta\, n_y + S_\delta\, n_x & C_\delta\, o_y + S_\delta\, o_x & C_\delta\, a_y + S_\delta\, a_x \\ n_z & o_z & a_z \end{bmatrix},$$

where $C_\delta = \cos(\delta)$ and $S_\delta = \sin(\delta)$. The ZYX Euler angles from this rotation matrix are

$$\boldsymbol{\alpha}' = \begin{bmatrix} \alpha'_1 \\ \alpha'_2 \\ \alpha'_3 \end{bmatrix} = \begin{bmatrix} \operatorname{arctan2}\left(n'_y, n'_x\right) \\ \operatorname{arctan2}\left(-n'_z, \sqrt{{a'_z}^2 + {o'_z}^2}\right) \\ \operatorname{arctan2}\left(o'_z, a'_z\right) \end{bmatrix} = \begin{bmatrix} \operatorname{arctan2}\left((C_\delta\, n_y + S_\delta\, n_x), (C_\delta\, n_x - S_\delta\, n_y)\right) \\ \operatorname{arctan2}\left(-n_z, \sqrt{{a_z}^2 + {o_z}^2}\right) \\ \operatorname{arctan2}\left(o_z, a_z\right) \end{bmatrix}, \tag{A7}$$

where δ only influences the first component α'_1. This allows the conclusion that β_3 only influences α_1 and results in the dependencies

$$\alpha'_1 = \alpha'_1(\boldsymbol{q}, \beta_1, \beta_2, \beta_3) \tag{A8}$$
$$\alpha'_2 = \alpha'_2(\boldsymbol{q}, \beta_1, \beta_2) = \alpha_2 \tag{A9}$$
$$\alpha'_3 = \alpha'_3(\boldsymbol{q}, \beta_1, \beta_2) = \alpha_3 \tag{A10}$$

with the consequences for the kinematic modeling of robots in 3T2R tasks described in Section 3.

Appendix A.2. Relation of the Geometric Jacobian and the Partial Derivative of the Rotation Matrix

To be able to calculate the gradient matrices of Section 5, the very uncommon partial derivative of the end-effector rotation matrix ${}^{0}\boldsymbol{R}_{E}$ with respect to the joint coordinates $\boldsymbol{q}_1$ of the kinematic chain is needed as term "III" in (34) and (38). It can either be derived with computer algebra systems from the analytic expression of the rotation matrix, or by first devising the time derivative

$$ {}^{0}\dot{\overline{\boldsymbol{R}}}_{E}(\boldsymbol{q}_1,\dot{\boldsymbol{q}}_1) = \frac{\mathrm{d}}{\mathrm{d}t}{}^{0}\overline{\boldsymbol{R}}_{E}(\boldsymbol{q}_1) = \frac{\partial}{\partial \boldsymbol{q}_1}{}^{0}\overline{\boldsymbol{R}}_{E}(\boldsymbol{q}_1)\dot{\boldsymbol{q}}_1 \tag{A11}$$

and then performing the derivative w.r.t $\dot{\boldsymbol{q}}_1$. Using the operators of Appendix A.4 on the relation between rotation matrices and angular velocities leads to the formulation

$$ \frac{\partial}{\partial \boldsymbol{q}_1}{}^{0}\overline{\boldsymbol{R}}_{E}(\boldsymbol{q}_1) = \frac{\partial}{\partial \dot{\boldsymbol{q}}_1}{}^{0}\dot{\overline{\boldsymbol{R}}}_{E} = \frac{\partial}{\partial \dot{\boldsymbol{q}}_1}\overline{\prod}\left(\overline{S}(\boldsymbol{\omega}_E), {}^{0}\overline{\boldsymbol{R}}_{E}\right) \quad \text{with} \quad \boldsymbol{\omega}_E = \boldsymbol{J}_{\omega,1}(\boldsymbol{q}_1)\dot{\boldsymbol{q}}_1, \tag{A12}$$

where the end-effector angular velocity $\boldsymbol{\omega}_E$ can be expressed with the rotational part $\boldsymbol{J}_{\omega,1}$ of the geometric Jacobian. It is used as the argument of the cross product matrix operator in stacked form

$$ \overline{S}(\boldsymbol{\omega}) = \begin{pmatrix} 0 & \omega_z & -\omega_y & -\omega_z & 0 & \omega_x & \omega_y & -\omega_x & 0 \end{pmatrix}^{\mathrm{T}} \quad \text{with} \quad \boldsymbol{\omega} = \begin{pmatrix} \omega_x & \omega_y & \omega_z \end{pmatrix}^{\mathrm{T}}. \tag{A13}$$

Applying again the chain rule for differentiation of the function composition of the three functions $\overline{\prod}$ (matrix product), $\boldsymbol{S}$ (cross product matrix) and $\boldsymbol{\omega}_E$ (linear relation of velocities), as practiced in Section 5, yields

$$ \frac{\partial}{\partial \boldsymbol{q}_1}{}^{0}\overline{\boldsymbol{R}}_{E} = \underbrace{\frac{\partial}{\partial \overline{S}}\overline{\prod}\left(\overline{S}, {}^{0}\overline{\boldsymbol{R}}_{E}\right)}_{\mathrm{I}\in\mathbb{R}^{9\times 9}} \underbrace{\frac{\partial}{\partial \boldsymbol{\omega}}\overline{S}(\boldsymbol{\omega})}_{\mathrm{II}\in\mathbb{R}^{9\times 3}} \underbrace{\boldsymbol{J}_{\omega,1}(\boldsymbol{q}_1)}_{\mathrm{III}\in\mathbb{R}^{3\times\dim(\boldsymbol{q}_1)}} \in \mathbb{R}^{9\times\dim(\boldsymbol{q}_1)}. \tag{A14}$$

The first factor is (A28), where the contents of ${}^{0}\overline{\boldsymbol{R}}_{E}$ are inserted. The second factor is a sign matrix which can be derived directly from (A13) and only contains $0/-1/1$. The third factor is the rotational part of the geometric Jacobian. This completes the calculation of the gradient matrix (34) for the Euler angles orientation residual, which was first introduced, but not pursued further in [7].

Appendix A.3. Relation of the Inverse- and Direct-Kinematics Matrices to the Analytic Jacobian

The properties of the gradient matrices from Section 5 can be illustrated further by comparing them to the analytic Jacobian of one kinematic chain, which gives the platform or end-effector velocity

$$ \dot{\boldsymbol{x}} = \boldsymbol{J}_1\dot{\boldsymbol{q}}_1, \qquad \begin{bmatrix} \dot{\boldsymbol{x}}_{\mathrm{t}} \\ \dot{\boldsymbol{x}}_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\mathrm{t},1}(\boldsymbol{q}_1) \\ \boldsymbol{J}_{\mathrm{r},1}(\boldsymbol{q}_1) \end{bmatrix} \dot{\boldsymbol{q}}_1. \tag{A15}$$

Also using the differential form (29) for the first kinematic leg chain of the parallel robot gives

$$ \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}_1(\boldsymbol{q}_1,\boldsymbol{x}) = \boldsymbol{\Phi}_{1,\partial\boldsymbol{q}_1}(\boldsymbol{q}_1,\boldsymbol{x})\dot{\boldsymbol{q}}_1 + \boldsymbol{\Phi}_{1,\partial\boldsymbol{x}}(\boldsymbol{q}_1,\boldsymbol{x})\dot{\boldsymbol{x}} = \boldsymbol{0} \tag{A16}$$

and results reorganized to the form of (A15) with (32) and (35) written component-wise to

$$ \begin{bmatrix} \dot{\boldsymbol{x}}_{\mathrm{t}} \\ \dot{\boldsymbol{x}}_{\mathrm{r}} \end{bmatrix} = -\begin{bmatrix} -\boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{\mathrm{r},1,\partial\boldsymbol{x}_{\mathrm{r}}}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{t},1,\partial\boldsymbol{q}_1} \\ \boldsymbol{\Phi}_{\mathrm{r},1,\partial\boldsymbol{q}_1} \end{bmatrix} \dot{\boldsymbol{q}}_1 = \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{t},1,\partial\boldsymbol{q}_1} \\ -\boldsymbol{\Phi}_{\mathrm{r},1,\partial\boldsymbol{x}_{\mathrm{r}}}^{-1}\boldsymbol{\Phi}_{\mathrm{r},1,\partial\boldsymbol{q}_1} \end{bmatrix} \dot{\boldsymbol{q}}_1. \tag{A17}$$

By equating coefficients of (A15) and (A17) the relations

$$\boldsymbol{\Phi}_{1,\partial q_1}(\boldsymbol{q}_1) = -\boldsymbol{\Phi}_{1,\partial x}(\boldsymbol{q}_1, \boldsymbol{x})\boldsymbol{J}_1(\boldsymbol{q}_1), \tag{A18}$$

$$\boldsymbol{\Phi}_{\mathrm{t},1,\partial q_1}(\boldsymbol{q}_1) = \boldsymbol{J}_{\mathrm{t},1}(\boldsymbol{q}_1) \quad \text{and} \tag{A19}$$

$$\boldsymbol{\Phi}_{\mathrm{r},1,\partial q_1}(\boldsymbol{q}_1, \boldsymbol{x}) = -\boldsymbol{\Phi}_{\mathrm{r},1,\partial x_\mathrm{r}}(\boldsymbol{q}_1, \boldsymbol{x})\boldsymbol{J}_{\mathrm{r},1}(\boldsymbol{q}_1) \tag{A20}$$

can be obtained between the gradient matrices and the analytic Jacobian $\boldsymbol{J}_1$ of the serial leg chain. The dependency on $\boldsymbol{q}_1$ and $\boldsymbol{x}$ has been added to highlight the main requirement, namely the zero equality condition of (A16): (A18) and (A20) only hold if the orientation residual is zero. In the inverse kinematics procedure of Section 5.3 the residual at step k in (40) is unequal to zero, which means that the Equation (A20) does not hold in this case. The translational part is unaffected, as can be seen by the missing argument $\boldsymbol{x}$ in (A19).

Appendix A.4. Matrix Operations for Partial Derivatives

To simplify the calculations of the gradient matrices of the residuals in Section 5, operators for matrices are replaced by operators for vectors, to avoid differentiating matrices or w.r.t. matrices which would require multi-dimensional tensors. The column operator $\overline{\boldsymbol{R}}$ for rotation matrices $\boldsymbol{R}$ to stack the coordinate systems unit vectors $\boldsymbol{n}, \boldsymbol{o}, \boldsymbol{a} \in \mathbb{R}^3$ vertically instead of horizontally is defined as

$$\overline{\boldsymbol{R}}(\boldsymbol{R}) = \begin{bmatrix} \boldsymbol{n} \\ \boldsymbol{o} \\ \boldsymbol{a} \end{bmatrix} \in \mathbb{R}^9 \quad \text{with} \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{o} & \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \in \mathrm{SO}(3) \tag{A21}$$

to avoid differentiating matrices or w.r.t. matrices. The special properties of the SO(3) group are not exploited and the operator can be used for $\mathbb{R}^{3\times3}$ as well. Matrix multiplication is expressed with the matrix product operator $\overline{\prod}$ such that

$${}^1\overline{\boldsymbol{R}}_3 = \overline{\prod}\left({}^1\overline{\boldsymbol{R}}_2, {}^2\overline{\boldsymbol{R}}_3\right) = \overline{\boldsymbol{R}}({}^1\boldsymbol{R}_3) \quad \text{with} \quad {}^1\boldsymbol{R}_3 = {}^1\boldsymbol{R}_2{}^2\boldsymbol{R}_3. \tag{A22}$$

The transposition operator $\boldsymbol{P}_\mathrm{T}$ is a 9 × 9 permutation matrix such that

$${}^2\overline{\boldsymbol{R}}_1 = \boldsymbol{P}_\mathrm{T}{}^1\overline{\boldsymbol{R}}_2 = \overline{\boldsymbol{R}}({}^1\boldsymbol{R}_2^\mathrm{T}) = {}^1\overline{\boldsymbol{R}}_2^\mathrm{T} \in \mathbb{R}^9 \;\; \text{with} \;\; {}^2\boldsymbol{R}_1 = {}^1\boldsymbol{R}_2^\mathrm{T} \in \mathrm{SO}(3) \;\; \text{and} \;\; {}^1\overline{\boldsymbol{R}}_2 = \overline{\boldsymbol{R}}({}^1\boldsymbol{R}_2). \tag{A23}$$

Writing ${}^1\overline{\boldsymbol{R}}_2^\mathrm{T}$ instead of $\boldsymbol{P}_\mathrm{T}{}^1\overline{\boldsymbol{R}}_2$ serves for the clarity of the expressions (34,36,38,39) and overloads the transposition operator for $\mathbb{R}^9$ noted with the bar.

Appendix A.5. Contents of the Partial Derivatives

The single expressions derived in Section 5 can be calculated with low computational effort from the definition of the XYZ and ZYX Euler angles from (5), (21) and (A6). With $\overline{\boldsymbol{R}} = [n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z]^\mathrm{T}$ the gradient "I" in (34,36,38,39) for ZYX angles becomes

$$\frac{\partial \boldsymbol{\alpha}(\overline{\boldsymbol{R}})}{\partial \overline{\boldsymbol{R}}} = \begin{bmatrix} -\frac{n_y}{n_x{}^2+n_y{}^2} & \frac{n_x}{n_x{}^2+n_y{}^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{a_z{}^2+o_z{}^2} & 0 & 0 & \frac{n_z\,o_z}{\sqrt{a_z{}^2+o_z{}^2}} & 0 & 0 & \frac{n_z\,a_z}{\sqrt{a_z{}^2+o_z{}^2}} \\ 0 & 0 & 0 & 0 & 0 & \frac{a_z}{a_z{}^2+o_z{}^2} & 0 & 0 & -\frac{o_z}{a_z{}^2+o_z{}^2} \end{bmatrix} \tag{A24}$$

and the reciprocal gradient “IV” in (36) for XYZ angles yields

$$\frac{\partial \overline{\boldsymbol{R}}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{bmatrix} 0 & -S_2\,C_3 & -C_2\,S_3 \\ C_1\,S_2\,C_3 - S_1\,S_3 & S_1\,C_2\,C_3 & -S_1\,S_2\,S_3 + C_1\,C_3 \\ S_1\,S_2\,C_3 + C_1\,S_3 & -C_1\,C_2\,C_3 & C_1\,S_2\,S_3 + S_1\,C_3 \\ 0 & S_2\,S_3 & -C_2\,C_3 \\ -C_1\,S_2\,S_3 - S_1\,C_3 & -S_1\,C_2\,S_3 & -S_1\,S_2\,C_3 - C_1\,S_3 \\ -S_1\,S_2\,S_3 + C_1\,C_3 & C_1\,C_2\,S_3 & C_1\,S_2\,C_3 - S_1\,S_3 \\ 0 & C_2 & 0 \\ -C_1\,C_2 & S_1\,S_2 & 0 \\ -S_1\,C_2 & -C_1\,S_2 & 0 \end{bmatrix} \tag{A25}$$

with $C_i = \cos(\beta_i)$, $S_i = \sin(\beta_i)$. The property

$$\left(\frac{\partial \boldsymbol{\beta}}{\partial \overline{\boldsymbol{R}}}\right)\left(\frac{\partial \overline{\boldsymbol{R}}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right) = \mathbf{1} \in \mathbb{R}^{3\times 3} \tag{A26}$$

can be used to test the implementation, if (A24, A25) are defined for the same Euler angle convention.

The gradient of the matrix product (A22) w.r.t. the second factor used in (34)/II, (36)/III, (38)/II and (39)/III is

$$\frac{\partial}{\partial \overline{\boldsymbol{R}}_2}\overline{\prod}(\overline{\boldsymbol{R}}_1, \overline{\boldsymbol{R}}_2) = \begin{bmatrix} \boldsymbol{R}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_1 \end{bmatrix} \tag{A27}$$

and to complete the enumeration the gradient w.r.t. the first factor used in (A14)/I is

$$\frac{\partial}{\partial \overline{\boldsymbol{R}}_1}\overline{\prod}(\overline{\boldsymbol{R}}_1, \overline{\boldsymbol{R}}_2) = \begin{bmatrix} \mathrm{diag}(n_x) & \mathrm{diag}(o_x) & \mathrm{diag}(a_x) \\ \mathrm{diag}(n_y) & \mathrm{diag}(o_y) & \mathrm{diag}(a_y) \\ \mathrm{diag}(n_z) & \mathrm{diag}(o_z) & \mathrm{diag}(a_z) \end{bmatrix}, \tag{A28}$$

where $n_x, n_y, \ldots$ are the entries of $\boldsymbol{R}_2$ and the diag matrices are 3×3. By transposing the elements of the matrix product (A22), only the first form (A27) had to be used in Section 5.

References

1. Baron, L. A joint-limits avoidance strategy for arc-welding robots. Presented at the International Conference on Integrated Design and Manufacturing in Mechanical Engineering, Montreal, QC, Canada, 16–19 May 2000; note: the paper is not part of the conference proceedings published by Springer. Available online: https://www.researchgate.net/profile/Luc_Baron (accessed on 2 August 2019).
2. Huo, L.; Baron, L. Kinematic inversion of functionally-redundant serial manipulators: Application to arc-welding. *Trans. Can. Soc. Mech. Eng.* **2005**, *29*, 679–690. [CrossRef]
3. Žlajpah, L. On orientation control of functional redundant robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2475–2482. [CrossRef]
4. Léger, J.; Angeles, J. Off-line programming of six-axis robots for optimum five-dimensional tasks. *Mech. Mach. Theory* **2016**, *100*, 155–169. [CrossRef]
5. Schappler, M.; Tappe, S.; Ortmaier, T. Resolution of Functional Redundancy for 3T2R Robot Tasks using Two Sets of Reciprocal Euler Angles. In *Advances in Mechanism and Machine Science, Proceedings of the 15th IFToMM World Congress on Mechanism and Machine Science, Kraków, Poland, 30 June–4 July 2019*; Springer: Berlin, Germany; 2019; Volume 73, pp. 1701–1710. [CrossRef]
6. Yoshikawa, T. Analysis and control of robot manipulators with redundancy. In *Robotics Research: The 1st International Symposium*; MIT Press: Cambridge, MA, USA, 1984; pp. 735–747; ISBN 0262022079.
7. Goldenberg, A.; Benhabib, B.; Fenton, R. A complete generalized solution to the inverse kinematics of robots. *IEEE J. Robot. Autom.* **1985**, *1*, 14–20. [CrossRef]

8. Zhu, W.; Qu, W.; Cao, L.; Yang, D.; Ke, Y. An off-line programming system for robotic drilling in aerospace manufacturing. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 2535–2545. [CrossRef]
9. Guo, Y.; Dong, H.; Ke, Y. Stiffness-oriented posture optimization in robotic machining applications. *Robot. Comput. Integr. Manuf.* **2015**, *27*, 367–376. [CrossRef]
10. Tale-Masouleh, M.; Gosselin, C. Singularity analysis of 5-RPUR parallel mechanisms (3T2R). *Int. J. Adv. Manuf. Technol.* **2011**, *57*, 1107–1121. [CrossRef]
11. Gogu, G. Solid Mechanics and Its Applications. In *Structural Synthesis of Parallel Robots, Part 1: Methodology*; Springer: Berlin, Germany; 2008; Volume 866.
12. Huang, T.; Liu, H.; Chetwynd, D. Generalized Jacobian analysis of lower mobility manipulators. *Mech. Mach. Theory* **2011**, *46*, 831–844. [CrossRef]
13. Merlet, J.P.; Perng, M.W.; Daney, D. Optimal trajectory planning of a 5-axis machine-tool based on a 6-axis parallel manipulator. In *Advances in Robot Kinematics*; Springer: Berlin, Germany, 2000; pp. 315–322.
14. Hong, K.S.; Kim, J.G. Manipulability analysis of a parallel machine tool: application to optimal link length design. *J. Robot. Syst.* **2000**, *17*, 403–415. [CrossRef]
15. Merlet, J.P. Solid mechanics and its applications. In *Parallel Robots*, 2nd ed.; Springer: Berlin, Germany, 2006; Volume 128.
16. Zhang, D. *Parallel Robotic Machine Tools*; Springer: Berlin, Germany, 2009.
17. Wang, J.; Gosselin, C.M. Kinematic analysis and singularity representation of spatial five-degree-of-freedom parallel mechanisms. *J. Robot. Syst.* **1997**, *14*, 851–869. [CrossRef]
18. Zhang, D.; Gosselin, C.M. Kinetostatic modeling of N-DOF parallel mechanisms with a passive constraining leg and prismatic actuators. *J. Mech. Des.* **2001**, *123*, 375–381. [CrossRef]
19. Zheng, K.J.; Gao, J.S.; Zhao, Y.S. Path control algorithms of a novel 5-DOF parallel machine tool. In Proceedins of the IEEE International Conference on Mechatronics and Automation, Niagara Falls, ON, Canada, 29 July–1 August 2005; Volume 3, pp. 1381–1385. [CrossRef]
20. Gao, J.S.; Sun, H.; Zhao, Y.S. The primary calibration research of a measuring limb in 5-UPS/PRPU parallel machine tool. In Proceedings of the IEEE International Conference on Intelligent Mechatronics and Automation, Chengdu, China, 26–31 August 2004; pp. 304–308. [CrossRef]
21. Liu, X.; Xu, Y.; Yao, J.; Xu, J.; Wen, S.; Zhao, Y. Control-faced dynamics with deformation compatibility for a 5-DOF active over-constrained spatial parallel manipulator 6PUS–UPU. *Mechatronics* **2015**, *30*, 107–115. [CrossRef]
22. Wen, S.; Qin, G.; Zhang, B.; Lam, H.K.; Zhao, Y.; Wang, H. The study of model predictive control algorithm based on the force/position control scheme of the 5-DOF redundant actuation parallel robot. *Robot. Auton. Syst.* **2016**, *79*, 12–25. [CrossRef]
23. Mbarek, T.; Nefzi, M.; Corves, B. Prototypische Entwicklung und Konstruktion eines neuartigen Parallelmanipulators mit dem Freiheitsgrad fünf. *VDI-Berichte Nr. 1892* **2005**. Available online: https://www.igmr.rwth-aachen.de/index.php/en/rob-en/rob-pentapod-en (accessed on 20 June 2019). (In German)
24. Schreiber, H.; Gosselin, C. Analyse et conception d'un manipulateur parallèle spatial à cinq degrés de liberté. In French. *Mech. Mach. Theory* **2003**, *38*, 535–548. [CrossRef]
25. Gao, F.; Peng, B.; Zhao, H.; Li, W. A novel 5-DOF fully parallel kinematic machine tool. *Int. J. Adv. Manuf. Technol.* **2006**, *31*, 201. [CrossRef]
26. Bär, G.F.; Weiß, G. Kinematic Analysis of a Pentapod Robot. *J. Geometry Graph.* **2006**, *10*, 173–182.
27. Lin, W.; Li, B.; Yang, X.; Zhang, D. Modelling and control of inverse dynamics for a 5-DOF parallel kinematic polishing machine. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 314. [CrossRef]
28. Alagheband, A.; Mahmoodi, M.; Mills, J.K.; Benhabib, B. Comparative analysis of a redundant pentapod parallel kinematic machine. *J. Mech. Robot.* **2015**, *7*, 034502. [CrossRef]
29. Gogu, G. Fully-isotropic parallel manipulators with five degrees of freedom. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 1141–1146. [CrossRef]
30. Huang, Z.; Li, Q.C. General Methodology for Type Synthesis of Symmetrical Lower-Mobility Parallel Manipulators and Several Novel Manipulators. *Int. J. Robot. Res.* **2002**, *21*, 131–145. [CrossRef]
31. Kong, X.; Gosselin, C.M. Type synthesis of 5-DOF parallel manipulators based on screw theory. *J. Robot. Syst.* **2005**, *22*, 535–547. [CrossRef]

32. Tale-Masouleh, M.; Saadatzi, M.H.; Gosselin, C.; Taghirad, H.D. A geometric constructive approach for the workspace analysis of symmetrical 5-PRUR parallel mechanisms (3T2R). In Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, QC, Canada, 15–18 August 2010; pp. 1335–1344. [CrossRef]
33. Cheng, L.; Wang, H.; Zhao, Y. Analysis and experimental investigation of parallel machine tool with redundant actuation. In *Intelligent Robotics and Applications, Proceedings of the International Conference on Intelligent Robotics and Applications, Wuhan, China, 15–17 October 2008;* Springer: Berin, Germany, 2008; pp. 179–188. [CrossRef]
34. Ramirez, D.; Kotlarski, J.; Ortmaier, T. Automatic generation of a minimal set of serial mechanisms for a combined structural—Geometrical synthesis. In Proceedings of the 14th IFToMM World Congress, Taipeh, Taiwan, 25–30 October 2015; [CrossRef]
35. Briot, S.; Khalil, W. *Dynamics of Parallel Robots*; Springer: Berlin, Germany, 2015.
36. Huo, L.; Baron, L. The joint-limits and singularity avoidance in robotic welding. *Ind. Robot.* **2008**, *35*, 456–464. [CrossRef]