

Resolution of Functional Redundancy for 3T2R Robot Tasks using Two Sets of Reciprocal Euler Angles

Moritz Schappler, Svenja Tappe, and Tobias Ortmaier

Institute for Mechatronic Systems, Leibniz University Hannover, Germany,
`moritz.schappler@imes.uni-hannover.de`

Abstract. Robotic tasks like welding or drilling with three translational and only two rotational degrees of freedom (“3T2R”) are of high industrial relevance but are rather scarcely addressed in scientific publications. Existing solutions for the resolution of the functional redundancy of robotic manipulators with more than five axes performing these tasks either expand the full kinematic formulation or reduce it in intermediate steps. This paper presents an approach to reduce the kinematic formulation from the start to solve the problem in a simpler way. This is done by using a set of reciprocal Euler angles to describe the end-effector orientation and the orientation error in inverse kinematics.

Keywords: functional redundancy, reciprocal Euler angles, inverse kinematics, robot manipulators, five-DoF task, 3T2R task

1 Introduction

Since the first papers in the 1980s, the fields of inverse kinematics [1] and the resolution of *intrinsic* redundancy [2] in robotics have been extensively elaborated upon. Inverse kinematics for robot manipulators can be solved analytically for some structures and can be approached in general with gradient-based numeric methods at joint position or joint velocity level. Due to the nonlinearity of rotation at position level [1], the latter is usually preferred.

Intrinsic redundancy is defined as a robot joint space dimension higher than the operational space dimension and functional redundancy is defined as an operational space dimension higher than the task space dimension, implying that there are more independent joint coordinates than task coordinates.

The kinematics of *functionally* redundant robots performing tasks with five instead of six degrees of freedom (DoF) has drawn much less attention, even if many industrial relevant tasks only require 5-DoF such as the class of 3T2R-tasks where the tool for the task is axis-symmetric. These tasks comprise amongst others arc welding [3], drilling [4, 5], spray-painting [6], milling [7], laser-cutting and glueing and are transferred to a greater extend to robotic manipulators in the automotive or aviation industry.

Since standard industrial robots have six DoF, one degree of *functional* redundancy exists which allows to improve performance characteristics of the robots as a secondary task while reaching a valid pose to perform one of the aforementioned primary tasks. To be able to incorporate performance optimization into inverse kinematics via the well-established gradient projection method [2], either the joint space can be augmented or the task space can be reduced. Augmenting the joint space is possible by virtually inserting an additional joint for the rotation around the task’s axis of symmetry (“tool axis”) [8]. Reducing the task space can be performed by orthogonal decomposition of the end-effector twist [3], removing the row of the “task frame Jacobian” corresponding to the tool axis [9], the construction of the Jacobian Nullspace based upon geometric properties of the task [10] or constructing a cone or pyramid resulting from the tool axis with an additional range of tolerance for tilt angles [6].

Another possibility for the resolution of the functional redundancy is a cascaded optimization where the inner loop calculates the inverse kinematics with standard methods and the outer loop optimizes the performance index. The optimization can be performed by evaluating the index on a range of rotation angles around the tool axis [4] or by using the incremental change of the rotation around the tool axis directly to adapt the joint angles [5].

The choice of performance indices aims to improve the task execution with measures such as the joint positions quadratic [3] or hyperbolic [4] distances from their limits, singularity avoidance via Frobenius-norm condition number [4], squared condition number [10] or a combination of manipulability and condition number of the Jacobian [11], stiffness [5] or milling chatter stability margins [7].

The characteristic length required to normalize the Jacobian for the performance indices can be either chosen constant regarding the robot geometry [4] or as an additional optimization parameter [10].

The existing methods each have drawbacks that are avoided by using the new method presented in this paper:

- Augmenting the joint space [8] increases the computational cost and can lead to an ill-conditioned Jacobian [11].
- Using a nested optimization [4, 5] does not allow using the well-proven gradient projection method.
- In [9] the tool-axis rotation of the end-effector has to be calculated and canceled out in the nullspace.
- Describing the tool axis with two points requires to define a distance between these points, which has to be chosen e. g. as the normalized Jacobian’s characteristic length to gain a good conditioning of the optimization [10].
- All methods have a high mathematical level of abstraction e. g. using orthogonal decomposition [11], linear matrix inequalities and convex optimization [6], rotation and component-selection of the vector part of the quaternion orientation error [9] or sequential quadratic programming [10].

This paper transfers the velocity level approach from [9] to the position level and the orientation error is expressed in Euler angles instead of quaternions:

The absolute orientation of the end-effector and the corresponding orientation error between desired and actual end-effector orientation are expressed with two sets of Euler angles which are mutually reciprocal. Reciprocity for sets of Euler angles is defined in this paper as successive rotations around intrinsic elementary axes of switched order. By this definition, the rotation component belonging to the tool axis in 3T2R tasks is always the same for the absolute orientation and for the orientation error, respectively. The effect of the tool axis can then be eliminated from the kinematic equations. This allows to express the kinematics equations in minimal form which makes it easy to use the gradient projection method to incorporate performance indices in inverse kinematics.

The contributions of this paper are

- a new formulation for the kinematics problem for robot manipulators performing 3T2R-tasks,
- an application of the formulation leading to an efficient solution for inverse kinematics with functional redundancy,
- remarks on the implementation of gradients of and w.r.t. rotation matrices and its performance.

The remainder of the paper is structured as follows: The idea of a kinematic description using sets of reciprocal Euler angles is elaborated upon in Sec. 2. Their application in inverse kinematics of 3T2R tasks and in resolving the functional redundancy is shown in Sec. 3 and 4. Sec. 5 concludes the paper.

2 Using Reciprocal Euler Angles for Robot Kinematics

The core of solving the inverse kinematics problem of robot manipulators for tasks with only two rotational degrees of freedom is the nonlinearity of rotation. Rotation can be described using rotation matrices, Euler angles, rotation angle/axis and its derivations like quaternions or the Rodrigues vector. In contrast to all other notations, consecutive elementary rotations like Euler angles can be reduced to adapt for tasks only requiring two rotatory DoF.

2.1 Kinematics Description

A serial kinematic chain is described with the joint positions \mathbf{q} and the forward kinematics

$$\mathbf{x}(\mathbf{q}) = \mathbf{f}(\mathbf{q}) \quad (1)$$

giving the configuration-dependent position and orientation $\mathbf{x}(\mathbf{q})$ of the actual end-effector (“E”) frame \mathcal{F}_E . In the following, the general, joint-independent, pose \mathbf{x} of the robot end-effector is defined as the *desired* (“D”) end-effector pose in the inverse kinematics problem and will be termed with “ \mathbf{x} ” without further supplements for the sake of compactness of the equations. This general pose

$$\mathbf{x} = [\mathbf{x}_t^T \ \mathbf{x}_r^T]^T \in \mathbb{R}^6 \quad (2)$$

describes the desired robot end-effector frames \mathcal{F}_D position

$$\mathbf{x}_t = {}_{(0)}\mathbf{r}_D \in \mathbb{R}^3 \quad (3)$$

and orientation

$${}^0\mathbf{R}_D(\mathbf{x}_r) = [\mathbf{n}_D \ \mathbf{o}_D \ \mathbf{a}_D] \in \text{SO}(3) \quad (4)$$

with respect to the base frame \mathcal{F}_0 , which is marked with left subscript “(0)” for vectors and left superscript “0” for rotation matrices. The rotation (4) is expressed without loss of generality with a set $\boldsymbol{\beta}$ of *X-Y-Z*-Euler angles

$$\mathbf{x}_r = [\beta_1 \ \beta_2 \ \beta_3]^\top \in \mathbb{R}^3, \quad (5)$$

$$\mathbf{R}(\boldsymbol{\beta}) = \mathbf{R}_x(\beta_1)\mathbf{R}_y(\beta_2)\mathbf{R}_z(\beta_3) \in \text{SO}(3). \quad (6)$$

The deviation $\boldsymbol{\Phi}$ between the desired end-effector frame \mathcal{F}_D expressed with \mathbf{x} of (2) and the actual robots end-effector frame \mathcal{F}_E expressed with $\mathbf{f}(\mathbf{q})$ of (1) is defined as

$$\boldsymbol{\Phi} = [\boldsymbol{\Phi}_t^\top \ \boldsymbol{\Phi}_r^\top]^\top \in \mathbb{R}^6, \quad (7)$$

which is the residual vector of the inverse kinematics problem. The vector

$$\boldsymbol{\Phi}_t(\mathbf{q}, \mathbf{x}) = -({}_0)\mathbf{r}_D + ({}_0)\mathbf{r}_E(\mathbf{q}) = -\mathbf{x}_t + ({}_0)\mathbf{r}_E(\mathbf{q}) \in \mathbb{R}^3 \quad (8)$$

from the origins of \mathcal{F}_D to \mathcal{F}_E is the translational part and the rotational part

$$\begin{aligned} \boldsymbol{\Phi}_r(\mathbf{q}, \mathbf{x}) &= [\alpha_1 \ \alpha_2 \ \alpha_3]^\top = \boldsymbol{\alpha} \left({}^D\mathbf{R}_E(\mathbf{x}_r, \mathbf{q}) \right) \\ &= \boldsymbol{\alpha} \left({}^0\mathbf{R}_D^\top(\mathbf{x}_r) {}^0\mathbf{R}_E(\mathbf{q}) \right) \in \mathbb{R}^3 \end{aligned} \quad (9)$$

is also chosen as a set of Euler angles $\boldsymbol{\alpha}$ [1], that is calculated from the rotation matrix in (9). 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

$$\mathbf{R}(\boldsymbol{\alpha}^*) := \mathbf{R}_x(\alpha_1^*)\mathbf{R}_y(\alpha_2^*)\mathbf{R}_z(\alpha_3^*) \in \text{SO}(3) \quad (10)$$

the same way as $\boldsymbol{\beta}$ leads to a set of transformations depicted in Fig. 1 (a) where the intermediate steps of the single elementary rotations are omitted since they have no technical meaning. The superscript in $\boldsymbol{\alpha}^*$ in (10) demarcates this specific example and following elaborations on the calculation of $\boldsymbol{\alpha}$.

2.2 Effect of the Reciprocal Euler Angles

Using $\boldsymbol{\Phi}_r = \boldsymbol{\alpha}^*$ as defined in (10), all three components of $\mathbf{x}_r = \boldsymbol{\beta}$ affect the rotation matrix ${}^D\mathbf{R}_E$, which makes it impossible to remove one rotational coordinate from the kinematic description, even if it is not required in the task. To encounter this issue, the Euler angle convention $\boldsymbol{\alpha}$ for the orientation error $\boldsymbol{\Phi}_r$ is now chosen to be

$$\mathbf{R}(\boldsymbol{\alpha}) := \mathbf{R}_z(\alpha_1)\mathbf{R}_y(\alpha_2)\mathbf{R}_x(\alpha_3) \in \text{SO}(3) \quad (11)$$

instead of the definition from (10). This set of *Z-Y-X*-Euler angles $\boldsymbol{\alpha}$ is defined in this paper as being *reciprocal* to the set of *X-Y-Z*-Euler angles of $\boldsymbol{\beta}$ for the absolute orientation.

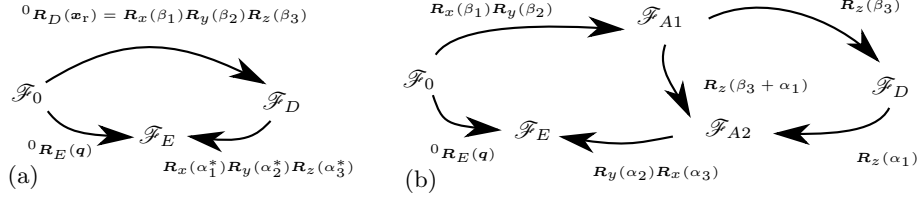


Fig. 1. Overview of the different frames (a) for 6-dof tasks with standard Euler angle notation and (b) for 5-dof tasks with reciprocal Euler angle notation.

The reciprocity refers to the switched order of the elementary axes X , Y and Z in the combination of the sets α and β . One set of Euler angles alone can not be declared as reciprocal without reference to another set of angles. Similar to the six end-effector operational space coordinates \mathbf{x} , the task space of 3T2R tasks is defined to have five coordinates

$$\boldsymbol{\eta} = [\boldsymbol{\eta}_t^T \boldsymbol{\eta}_r^T]^T \in \mathbb{R}^5. \quad (12)$$

The translational part

$$\boldsymbol{\eta}_t = \mathbf{x}_t = {}_{(0)}\mathbf{r}_D \in \mathbb{R}^3 \quad (13)$$

remains unchanged and the rotational part

$$\boldsymbol{\eta}_r = [\beta_1 \ \beta_2]^T = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{=\mathbf{P}_\eta} \mathbf{x}_r \in \mathbb{R}^2 \quad (14)$$

is modified compared to \mathbf{x} . The last rotation β_3 around the z -axis \mathbf{a}_D of ${}^0\mathbf{R}_D$ is excluded from the task space by the selection matrix \mathbf{P}_η , since it corresponds to a rotation around the tool axis in 3T2R tasks and is a DoF of the operational space which can be set arbitrarily (from the kinematic point of view).

The frames \mathcal{F}_{A1} and \mathcal{F}_{A2} result from intermediate elementary rotations, as sketched in Fig. 1 (b). These intermediate frames are the partial frame rotation to the former 3T3R desired frame \mathcal{F}_D

$${}^0\mathbf{R}_{A1} = \mathbf{R}_x(\beta_1)\mathbf{R}_y(\beta_2) = [\mathbf{n}_{A1} \ \mathbf{o}_{A1} \ \mathbf{a}_{A1}] \quad (15)$$

and the partial frame rotation

$${}^0\mathbf{R}_{A2} = {}^0\mathbf{R}_E(\mathbf{q}) (\mathbf{R}_y(\alpha_2)\mathbf{R}_x(\alpha_3))^T = [\mathbf{n}_{A2} \ \mathbf{o}_{A2} \ \mathbf{a}_{A2}] \quad (16)$$

from the actual frame and the x - and y -axis error components. The frames \mathcal{F}_{A1} , \mathcal{F}_{A2} and \mathcal{F}_D all share the same z -axis

$$\mathbf{a}_D = \mathbf{a}_{A1} = \mathbf{a}_{A2} \quad (17)$$

which is also the tool axis, since transformations between these frames are only rotations \mathbf{R}_z around the z -axes from (17).

Since the tool axis rotation β_3 and the orientation error component α_1 are defined around the same axis, β_3 only influences α_1 and not α_2 and α_3 , which can be expressed by

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_1(\mathbf{q}, \beta_1, \beta_2, \beta_3) \\ \alpha_2(\mathbf{q}, \beta_1, \beta_2) \\ \alpha_3(\mathbf{q}, \beta_1, \beta_2) \end{bmatrix} = \begin{bmatrix} \alpha_1(\mathbf{q}, \mathbf{x}) \\ \alpha_2(\mathbf{q}, \boldsymbol{\eta}) \\ \alpha_3(\mathbf{q}, \boldsymbol{\eta}) \end{bmatrix}. \quad (18)$$

This property of the reciprocal sets of Euler angles allows a kinematic description of robots in 3T2R tasks, as elaborated in the next sections. It can be derived by symbolically comparing $\boldsymbol{\alpha}$ of (30) for two different desired rotations $\boldsymbol{\beta}$ which only differ regarding a rotation β_3 around the tool axis.

3 Application on the Inverse Kinematics of 3T2R Tasks

The standard methods, introduced in Sec. 1 for solving the inverse kinematics for 3T2R tasks and exploiting the functional redundancy, struggle with the definition of a Jacobian matrix with appropriate dimensions.

3.1 Jacobian for Gradient-based Inverse Kinematics

To obtain a Jacobian with minimal row dimension, the kinematic condition for the 3T2R problem in the coordinates $\boldsymbol{\eta}$ is now defined as

$$\boldsymbol{\Psi} = [\boldsymbol{\Psi}_t^T \ \boldsymbol{\Psi}_r^T]^T \in \mathbb{R}^5 \quad (19)$$

following the definition for $\boldsymbol{\Phi}$ from (7). The translational part

$$\boldsymbol{\Psi}_t(\mathbf{q}, \boldsymbol{\eta}) = \boldsymbol{\Phi}_t(\mathbf{q}, \mathbf{x}) = -\boldsymbol{\eta}_t + {}_{(0)}\mathbf{r}_E(\mathbf{q}) \in \mathbb{R}^3 \quad (20)$$

remains unchanged to (8). The first component of the rotational part from (9) is omitted by the selection matrix \mathbf{P}_Ψ , since it corresponds to the orientation error α_1 around the tool axis, leaving

$$\boldsymbol{\Psi}_r(\mathbf{q}, \boldsymbol{\eta}) = [\alpha_2 \ \alpha_3]^T = \overbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{=\mathbf{P}_\Psi} \boldsymbol{\Phi}_r(\mathbf{q}, \mathbf{x}) \in \mathbb{R}^2. \quad (21)$$

The dependence on $\boldsymbol{\eta}$ and not on \mathbf{x} can be explained by using (18) and (14) together with ${}^{A1}\mathbf{R}_{A2} = \mathbf{R}_z(\beta_3 + \alpha_1)$ from Fig. 1 (b) which results to

$$\begin{aligned} \boldsymbol{\Psi}_r(\mathbf{q}, \boldsymbol{\eta}) &= \mathbf{P}_\Psi \boldsymbol{\alpha} ({}^{A2}\mathbf{R}_E(\mathbf{q}, \boldsymbol{\eta}_r, \alpha_1, \beta_3)) \\ &= \mathbf{P}_\Psi \boldsymbol{\alpha} \left({}^0\mathbf{R}_{A1}^T(\boldsymbol{\eta}_r) {}^0\mathbf{R}_E(\mathbf{q}) \right). \end{aligned} \quad (22)$$

The condition $\boldsymbol{\Phi} = \mathbf{0}$ or $\boldsymbol{\Psi} = \mathbf{0}$ leads to a valid configuration of the end-effector position and the complete orientation of the end-effector (using $\boldsymbol{\Phi}$) or the orientation only of the tool-axis (using $\boldsymbol{\Psi}$).

Following [1], the inverse kinematics problem for serial link robots in 6-DoF tasks at the iterative step $k + 1$ can be derived with the linear approximation of the Taylor series expansion of $\Phi(\mathbf{q}, \mathbf{x})$ to

$$\Phi(\mathbf{q}^{k+1}, \mathbf{x}) = \Phi(\mathbf{q}^k, \mathbf{x}) + \left. \frac{\partial \Phi(\mathbf{q}, \mathbf{x})}{\partial \mathbf{q}} \right|_{\mathbf{q}^k} (\mathbf{q}^{k+1} - \mathbf{q}^k) \quad (23)$$

where $\Phi_{\partial \mathbf{q}} = (\partial \Phi / \partial \mathbf{q})$ is called the ‘‘Jacobian matrix corresponding to the residual vector’’ (Φ) in [1] and \mathbf{q}^0 is assumed as given. For 3T2R tasks

$$\Psi(\mathbf{q}^{k+1}, \boldsymbol{\eta}) = \Psi(\mathbf{q}^k, \boldsymbol{\eta}) + \left. \frac{\partial \Psi(\mathbf{q}, \boldsymbol{\eta})}{\partial \mathbf{q}} \right|_{\mathbf{q}^k} (\mathbf{q}^{k+1} - \mathbf{q}^k) \quad (24)$$

can be defined in the same way with the condition

$$\Psi(\mathbf{q}^{k+1}, \boldsymbol{\eta}) = \mathbf{0} \quad (25)$$

for solution of the inverse kinematics in the next step $k + 1$. The increment

$$\Delta \mathbf{q}^k = (\mathbf{q}^{k+1} - \mathbf{q}^k) = \left(\left. \frac{\partial \Psi(\mathbf{q}, \boldsymbol{\eta})}{\partial \mathbf{q}} \right|_{\mathbf{q}^k} \right)^\dagger (\mathbf{0} - \Psi(\mathbf{q}^k, \boldsymbol{\eta})) \quad (26)$$

of the joint angles towards this solution can be used in iterative algorithms like Newton-Raphson together with methods to adapt the step sizes to ensure convergence. Depending on the dimension of the matrix, $(\cdot)^\dagger$ denotes the matrix inverse or the pseudo-inverse.

3.2 Discussion of Singularities

In [9], quaternions were used in favor of Euler angles for the orientation error with the reasons that a singularity-free representation of $\Phi = \mathbf{0}$ is needed. This is the case for Tait-Bryan angles¹ like e. g. the Z - Y - X notation used here for $\boldsymbol{\alpha}$. For control purposes it can be assumed that the components of the orientation error $\boldsymbol{\alpha}$ always stay below $\pm 90^\circ$, which avoids the ‘‘gimbal lock’’ representation singularity of Euler angles. This assumption can be justified by the consideration that active and effective position and orientation tracking will only produce small errors. A singularity-free representation of the absolute orientation $\boldsymbol{\beta}$ has to be ensured at the phase of motion planning, as well as avoiding discontinuities of the trajectory.

3.3 Remarks on Differentiation and Rotation Matrices

The Jacobians $\Phi_{\partial \mathbf{q}}$ of (23) or $\Psi_{\partial \mathbf{q}}$ of (24) consist of nested non-linear functions and do not use the geometric Jacobian of the serial link manipulator for the rotational part, which is easy to calculate.

¹ Tait-Bryan angles are referred to as A - B - C -Euler angles with axes $A \neq C$ as opposed to ‘‘proper Euler angles’’ with $A = C$. The elementary axes are $A, B, C \in \{X, Y, Z\}$.

However, $\Phi_{\partial q}$ can be implemented efficiently by exploiting the chain rule and sparsity of the matrices with partial derivatives as shown in the following.

The column operator $\overline{\mathbf{R}}$ for rotation matrices \mathbf{R} to stack the coordinate systems unit vectors $\mathbf{n}, \mathbf{o}, \mathbf{a} \in \mathbb{R}^3$ vertically instead of horizontally is defined as

$$\overline{\mathbf{R}}(\mathbf{R}) = \begin{bmatrix} \mathbf{n} \\ \mathbf{o} \\ \mathbf{a} \end{bmatrix} \in \mathbb{R}^9 \quad \text{with} \quad \mathbf{R} = [\mathbf{n} \ \mathbf{o} \ \mathbf{a}] = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \in \text{SO}(3) \quad (27)$$

to avoid differentiating matrices or w.r.t. matrices. Matrix multiplication is then expressed with the matrix product operator $\overline{\Pi}$ such that

$${}^1\overline{\mathbf{R}}_3 = \overline{\Pi}({}^1\overline{\mathbf{R}}_2, {}^2\overline{\mathbf{R}}_3) = \overline{\mathbf{R}}({}^1\mathbf{R}_3) \quad \text{with} \quad {}^1\mathbf{R}_3 = {}^1\mathbf{R}_2 {}^2\mathbf{R}_3. \quad (28)$$

The transposition operator \mathbf{P}_T is a 9×9 permutation matrix such that

$${}^2\overline{\mathbf{R}}_1 = \mathbf{P}_T {}^1\overline{\mathbf{R}}_2 \in \mathbb{R}^9 \quad \text{with} \quad {}^2\mathbf{R}_1 = {}^1\mathbf{R}_2^T \in \text{SO}(3) \quad \text{and} \quad {}^1\overline{\mathbf{R}}_2 = \overline{\mathbf{R}}({}^1\mathbf{R}_2). \quad (29)$$

The Euler angles can be calculated from the general rotation matrix \mathbf{R} of (27) in the same way as before² using these operators with the notation

$$\alpha(\overline{\mathbf{R}}) = \alpha(\mathbf{R}) = \begin{bmatrix} \arctan2(n_y, n_x) \\ \arctan2(-n_z, \sqrt{a_z^2 + o_z^2}) \\ \arctan2(o_z, a_z) \end{bmatrix} \quad (30)$$

at the Z - Y - X example. Finally, applying this to the rotational part of the residual vector Jacobian and using the chain rule for differentiation yields

$$\begin{aligned} \frac{\partial}{\partial \mathbf{q}} \Phi_r &= \frac{\partial}{\partial \mathbf{q}} \alpha \left({}^0\mathbf{R}_D^T(\mathbf{x}) {}^0\mathbf{R}_E(\mathbf{q}) \right) \\ &= \frac{\partial}{\partial \mathbf{q}} \alpha \left(\overline{\Pi} \left({}^0\overline{\mathbf{R}}_D^T(\mathbf{x}), {}^0\overline{\mathbf{R}}_E(\mathbf{q}) \right) \right) \\ &= \underbrace{\left(\frac{\partial \alpha}{\partial \overline{\mathbf{R}}} \right)}_{\text{I} \in \mathbb{R}^{3 \times 9}} \underbrace{\left(\frac{\partial \overline{\Pi} \left({}^0\overline{\mathbf{R}}_D^T, {}^0\overline{\mathbf{R}}_E \right)}{\partial {}^0\overline{\mathbf{R}}_E} \right)}_{\text{II} \in \mathbb{R}^{9 \times 9}} \underbrace{\left(\frac{\partial {}^0\overline{\mathbf{R}}_E(\mathbf{q})}{\partial \mathbf{q}} \right)}_{\text{III} \in \mathbb{R}^{9 \times \dim(\mathbf{q})}}. \end{aligned} \quad (31)$$

The first two partial derivatives ‘‘I’’ and ‘‘II’’ from (31) are sparse matrices of low complexity where the elements of ${}^D\mathbf{R}_E$ and ${}^0\mathbf{R}_D$ have to be inserted. The last partial derivative ‘‘III’’ can be derived efficiently with computer algebra systems.

The translational part $\Phi_{t,\partial q} = \partial \Phi_t / \partial \mathbf{q} = \Psi_{t,\partial q}$ is enclosed in the geometric Jacobian of the manipulator and is not considered at this point to focus on the rotational aspects. The gradient $\Psi_{r,\partial q} = \partial \Psi_r / \partial \mathbf{q} = \mathbf{P}_\Psi \Phi_{r,\partial q}$ is obtained from the results of (31) with the selection matrix \mathbf{P}_Ψ from (21).

² Utilizing the sign-aware operator $\arctan2(y, x)$ instead of $\arctan(y/x)$ allows angles to be in $(-\pi, +\pi]$, removes ambiguities and provides global differentiability.

To account for exchangeable end-effectors or tools, a distinguished frame rotation ${}^0\mathbf{R}_N(\mathbf{q})$ to the last robot link frame and a constant frame rotation ${}^N\mathbf{R}_E$ to the end-effector frame can be used. The properties of the column-operator allow then to substitute the last term “III” in (31) with

$$\begin{aligned} \frac{\partial {}^0\overline{\mathbf{R}}_E(\mathbf{q})}{\partial \mathbf{q}} &= \frac{\partial}{\partial \mathbf{q}} \overline{\Pi}({}^0\overline{\mathbf{R}}_N(\mathbf{q}), {}^N\overline{\mathbf{R}}_E) \\ &= \left(\frac{\partial}{\partial {}^0\overline{\mathbf{R}}_N} \overline{\Pi}({}^0\overline{\mathbf{R}}_N, {}^N\overline{\mathbf{R}}_E) \right) \left(\frac{\partial}{\partial \mathbf{q}} {}^0\overline{\mathbf{R}}_N(\mathbf{q}) \right). \end{aligned} \quad (32)$$

4 Resolving Functional Redundancy for 3T2R Tasks

The inverse kinematics formalism (26), described in the previous section 3, does not take into account the functional redundancy yet.

The typical scenario for 3T2R tasks in industry is a serial link robot with $\dim(\mathbf{q}) > 5$. Most commonly a classical industrial robot with $\dim(\mathbf{q}) = 6$ will be used. Since $\dim(\boldsymbol{\Psi}) = 5$, at least one DoF is free for optimization of additional criteria.

For the sake of simplicity, as an additional criterion the summed \mathbf{W} -weighted quadratic distances

$$h(\mathbf{q}) = \frac{1}{2}(\mathbf{q} - \bar{\mathbf{q}})^T \mathbf{W}(\mathbf{q} - \bar{\mathbf{q}}) \quad (33)$$

of the joint positions \mathbf{q} from their respective reference position $\bar{\mathbf{q}}$ will be used. Minimizing $h(\mathbf{q})$ avoids the risk of joints reaching their technical limits. The gradient

$$h_{\partial \mathbf{q}} = \frac{\partial h}{\partial \mathbf{q}} = \mathbf{W}(\mathbf{q} - \bar{\mathbf{q}}) \quad (34)$$

can be used to include this additional minimization into the solution of the inverse kinematics. The gradient $h_{\partial \mathbf{q}}$ is projected into the nullspace of $\boldsymbol{\Psi}_{\partial \mathbf{q}}$ with

$$\begin{aligned} \Delta \mathbf{q} &= \Delta \mathbf{q}_T + \Delta \mathbf{q}_N \\ &= \boldsymbol{\Psi}_{\partial \mathbf{q}}^\dagger (-\boldsymbol{\Psi}) + (\mathbf{1} - \boldsymbol{\Psi}_{\partial \mathbf{q}}^\dagger \boldsymbol{\Psi}_{\partial \mathbf{q}}) h_{\partial \mathbf{q}} \end{aligned} \quad (35)$$

where the nullspace incremental motion $\Delta \mathbf{q}_N$ does not affect the task achievement ensured via $\Delta \mathbf{q}_T$ [2].

It is reported in [5] that their optimization does not work for industrial robots where the tool axis is aligned parallel to the last robot joint axis. This “pointing configuration” can also not be addressed by the nullspace projection from (35), since the nullspace corresponds to the last robot axis and (in the case of a six-DoF robot)

$$\mathbf{1} - \boldsymbol{\Psi}_{\partial \mathbf{q}}^\dagger \boldsymbol{\Psi}_{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{0}_{5 \times 5} & \mathbf{0}_{5 \times 1} \\ \mathbf{0}_{1 \times 5} & 1_{1 \times 1} \end{bmatrix} \quad (36)$$

only projects the gradient $h_{\partial \mathbf{q}}$ of additional criteria onto the last joint. Therefore, the method does only work if the tool is mounted in a different configuration (“side” or “hanging” in [5]), which might be unfavorable for some end-effectors or tasks.

5 Conclusions

This paper presented a novel concept to formulate the inverse kinematics problem of serial kinematic chains using reciprocal sets of Euler angles. This exploits the properties of Euler angles to reduce the number of coordinates required for 3T2R tasks which are of high industrial relevance. Applications to the inverse kinematics of serial robots are given. Future works will include comparative simulative evaluation against state of the art methods, inquiries on singularities and the relation to the geometric Jacobian and the application to parallel robots.

Acknowledgements

The financial support from the Deutsche Forschungsgemeinschaft (DFG) under grant number OR 196/33-1 is gracefully acknowledged.

References

1. Goldenberg, A., Benhabib, B., Fenton, R.: A complete generalized solution to the inverse kinematics of robots. *IEEE Journal on Robotics and Automation* **1**(1), 14–20 (1985). DOI 10.1109/JRA.1985.1086995
2. Yoshikawa, T.: Analysis and control of robot manipulators with redundancy. In: *Robotics research: the first international symposium*, pp. 735–747. MIT press Cambridge, MA (1984)
3. Huo, L., Baron, L.: Kinematic inversion of functionally-redundant serial manipulators: application to arc-welding. *Transactions of the Canadian Society for Mechanical Engineering* **29**(4), 679–690 (2005). DOI 10.1139/tcsme-2005-0045
4. Zhu, W., Qu, W., Cao, L., Yang, D., Ke, Y.: An off-line programming system for robotic drilling in aerospace manufacturing. *The International Journal of Advanced Manufacturing Technology* **68**(9-12), 2535–2545 (2013). DOI 10.1007/s00170-013-4873-5
5. Guo, Y., Dong, H., Ke, Y.: Stiffness-oriented posture optimization in robotic machining applications. *Robotics and Computer-Integrated Manufacturing (RCIM)* **27**(2), 367–376 (2015). DOI 10.1016/j.rcim.2015.02.006
6. From, P.J., Gravdahl, J.T.: A real-time algorithm for determining the optimal paint gun orientation in spray paint applications. *IEEE Transactions on Automation Science and Engineering* **7**(4), 803–816 (2010). DOI 10.1109/TASE.2009.2033567
7. Mousavi, S., Gagnol, V., Bouzgarrou, B.C., Ray, P.: Stability optimization in robotic milling through the control of functional redundancies. *RCIM* **50**, 181–192 (2018). DOI 10.1016/j.rcim.2017.09.004
8. Baron, L.: A joint-limits avoidance strategy for arc-welding robots. In: *Int. Conf. on Integrated Design and Manufacturing in Mech. Eng.*, pp. 16–19 (2000)
9. Žlajpah, L.: On orientation control of functional redundant robots. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2475–2482 (2017). DOI 10.1109/ICRA.2017.7989288
10. Léger, J., Angeles, J.: Off-line programming of six-axis robots for optimum five-dimensional tasks. *Mechanism and Machine Theory* **100**, 155–169 (2016). DOI 10.1016/j.mechmachtheory.2016.01.015
11. Huo, L., Baron, L.: The joint-limits and singularity avoidance in robotic welding. *Industrial Robot: An International Journal* **35**(5), 456–464 (2008). DOI 10.1108/01439910810893626