Punjab University Journal of Mathematics
(ISSN 1016-2526)
Vol. 52(3)(2020) pp. 111-134

Fine Tuning of the Fixed Point Iteration-Based Matrix Inversion-Free Adaptive Inverse Kinematics Using Abstract Rotations

Hamza Khan
Doctoral School of Applied Informatics and Applied Mathematics,
Óbuda University Budapest, Hungary,
Email: hamza.khan@phd.uni-obuda.hu
József K. Tar
University Research and Innovation Center, Antal Bejczy Center for Intelligent Robotics
(ABC iRob) Óbuda University Budapest, Hungary,
Email: tar.jozsef@nik.uni-obuda.hu

Received: 23 April, 2019 / Accepted: 12 Feburary, 2020 / Published online: 10 March, 2020

Abstract. The *Inverse Kinematic Task of Robots* of redundant open kinematic chain normally does not have closed form analytical solution. The generally viable approach applies the *Differential Inverse Kinematics* in which the derivative of the nominal trajectory of the robot as well as the derivative of its internal generalized coordinates according to some scalar variable (that may be e.g. the time) are related to each other by the Jacobian of the arm due to the chain rule of differentiation. The traditional solutions compute some *generalized inverse* of this Jacobian that exists only in the non-singular positions, and does not behave well in the vicinity of the singularities where normally complementary tricks (practically the modification of the inverse kinematic task by replacing it with a solvable "deformed" version) are applied to obtain some "solution". These modifications may degrade the precision of the solution in the nonsingular points. The idea of replacing the matrix inversion with *Fixed Point Iteration (FPI)* in solving the inverse kinematic task was suggested in 2016 on the basis of the assumption that the kinematic parameters of the robot are precisely known. It was shown that this approach automatically yielded well behaving solutions in, and in the vicinity of the singularities without the use of any "complementary deformation". In 2017 it was realized that in the possession of an *approximate parameter set of the kinematic model* an adaptive inverse kinematic task solution can be developed on this basis if the pose and location of the last segment of the robot as well as the generalized coordinates can be measured. This approach used counter-rotations to guarantee the convergence of the fixed point iteration. Later it cropped up that similar abstract rotations can be applied in the realization of the fixed point iterations, too. The so elaborated solution can be combined with the inclusion of free parameters that can be used a) for making a trade-off between the precision requirements for the tracked position and/or pose, and b) parameters that affect the "distribution" of the ambiguous solution between the rotations of the redundant generalized coordinates. The operation of the approach is exemplified by the use of a redundant 8 Degree of Freedom robot arm via simulations made in Julia ver. 1.0.3.

Key Words: Redundant Robot Arm, Fixed Point Iteration, Jacobian, Banach Space, Banach's Fixed Point Theorem, Differential Inverse Kinematic Task, Generalized Inverses, Lie Groups.

1. Introduction

The solution of the inversed kinematic task of robots arise in the practical industrial applications. Generally the user navigates by using the Cartesian coordinates fixed to the workshop, and the required motion of the robot is formulated by the use of such coordinates. For instance, in [33] the real-time control of a 5 axles machining tool is considered in which the first 4 joints are used for positioning the workpiece, and the fifth one is applied for moving the machining tool (the cutter). With the assumption that the available kinematic model of the equipment is precise, the Authors developed a non-redundant problem having a size 5×5 Jacobian the determinant of which was computable in closed analytical form, and the kinematic singularities of the construction were determined by the use of this determinant. On this basis, via generalizing the Moore-Penrose pseudoinverse [31, 35] by using a positive definite symmetric weighting matrix instead the identity matrix the Authors developed a real-time controller that feeds back the Cartesian tracking error and investigated it by the use of a Matlab-Simulink application. In [32] a similar problem was investigated for a 6 degree of freedom welding robot that cooperated with a rotary positioner the rotational angle of which served as the 7^{th} axis of this system. The shape of the workpiece and the angle of the positioner's axis determined a complex 3D curve for the end-effector of the welding robot. The problem was solved by the use of a similar pseudoinverse and a control program as the problem in [33].

In solving kinematic problems in robotics often an augmented Jacobian is introduced for obstacle avoidance (e.g. [15, 4]), and the Moore-Penrose pseudoinverse [31, 35] is used for the redundant robot arms for the disambiguation of the otherwise normally ambiguous possible solutions. The ambiguity of the solution can be utilized for taking into consideration other points of view than simply solving the inverse kinematic task. For instance some elements of the null space of the Jacobian can be added later to the so obtained solution (e.g. [30, 36, 41]) that can make the problem of the continuity of the solution arise.

The controllers of robots normally are programmed on the basis of using the *generalized coordinates* of the robot arm, $q \in \mathbb{R}^n$, $n \in \mathbb{N}$, that physically mean either rotations around, or shifts in the direction of unit vectors that can be defined as *kinematic constants* in the *"home position"* of the robot. This definition may contain arbitrary elements. For instance, the use of the *Denavit-Hartenberg Conventions* introduced in 1955 [9] is a reasonable possibility, though its is not compulsory. The so called *Forward Kinematic Task* means the calculation of the position of the endpoint of the robot arm and the *rotational pose* of the last segment as the function of q. Assuming *rigid links*, the possible operations with the rigid bodies are determined by the *Special Euclidean Group* of three dimensions that can be conveniently and lucidly represented by the use of the Lie Group ([28]) of the Homogeneous Matrices and their generators in the case of even redundant robot arms in which $n > 6$. To obtain "smarter" robot arms, the use of 7 *Degree of Freedom (DOF)* constructions became popular in our days (e.g. [45]).

The *Inverse Kinematic Task* means the calculation of the q joint coordinates when the position of the endpoint of the robot arm –the "Tool Center Point (TCP)– and the *rotational pose* of the last segment as the function of a scalar variable (that physically may be the time) are given. (This task can be further completed by adding requirements that are valid for the motion of other links, too.) The solution of this task mathematically is difficult due to its nonlinearities, and ambiguities. Closed form analytical solutions are available only for special constructions (e.g. the PUMA robot [26], the Delta robot [6, 42]). However, the formulation of the *Differential Inverse Kinematic Task* is very simple even for strongly redundant open kinematic chains (e.g. [44]) and leads to the "inversion" of a normally non-quadratic Jacobian. The appropriate "generalized inverses" generally suffer from the presence of kinematic singularities. To evade this problem in 2016 an alternative, FPI-based approach was suggested in [8] that behaved nicely in, and in the vicinity of the singularities.

However, this approach assumed that we have precise kinematic model of the robot arm. In the practice this assumption has limitations even if the links really behave as rigid bodies. If the robot links are long enough, even little manufacturing error in the orientation of the rotary and prismatic axles in the home position can cause considerable error in the position of the endpoint and pose of the last link. In 2017 an *Adaptive Inverse Kinematic Approach* was suggested in [19] that, instead trying to solve the precise identification of the exact kinematic parameters, iteratively corrected their effects in computing the solution of the inverse kinematics for a given prescribed trajectory. In the present paper this method is further developed by the introduction of novel parameters that can weight the significance of the orientation precision versus the precision of the location of the endpoint, and by the use of the ambiguity of the possible solutions influence the "distribution" of the motion task between the redundant joint coordinates.

2. Details of the Fixed Point Iteration-Based Problem Formulation

Let $x \in \mathbb{R}^m$, $m \in \mathbb{N}$ the pose and location data of the last link of the robot arm. (In more general cases this array can be augmented by similar data of other links if necessary.) Let $q \in \mathbb{R}^n$, $n \in \mathbb{N}$ denote the *generalized coordinates* of the robot that directly can be rotated by the robot's controller. If only the location and pose of the last link is considered in x we have only 6 *independent data*, though it may be more convenient to use a *redundant representation* using an x that contains the 9 elements of the rotational matrix and the further 3 components of the Cartesian coordinates of the location with respect to the workshop's frame of reference, i.e. $m = 12$. In this case the *robot itself is redundant* if $n > 6$. (In our numerical investigations $n = 8$.) A *nominal trajectory* that has to be tracked can be prescribed by a scalar variable $s \in \mathbb{R}$ as

$$x^N(s) = f(q(s)) \ , \ x^N(s_{ini}) = f(q_{ini}) \ , \tag{2. 1}$$

in which q_{ini} is ambiguous in the case of a redundant robot arm. Due to the strong nonlinearities in $f(q)$ (2. 1) is differentiated as

$$\frac{\mathrm{d}x^N(s)}{\mathrm{d}s} = \frac{\partial f(q)}{\partial q} \frac{\mathrm{d}q(s)}{\mathrm{d}s} \tag{2. 2}$$

with the *Jacobian* $J(q) \stackrel{def}{=} \frac{\partial f}{\partial q}$, and some *generalized inverse* of $J(q)$ is considered if this inverse exists at the given $q(s)$. A "generalized inverse" means a kind of "disambiguation" applied for the ambiguous solution of the problem. For instance, the *Moore-Penrose Pseudoinverse* [31, 35] minimizes the sum $\sum_i \left(\frac{\mathrm{d}q_i}{\mathrm{d}s}\right)^2$ *under the constraint* that (2. 2) must be valid. It leads to the solution

$$\frac{\mathrm{d}q}{\mathrm{d}s} = J^T \left(JJ^T\right)^{-1} \frac{\mathrm{d}x}{\mathrm{d}s} \tag{2. 3}$$

that exists only if JJ^T is invertible, and can be reliably used only if it is *well conditioned*. In the kinematic singularities and in their vicinity these conditions are not met.

A plausible approach to deal with the problem of singularities is the modification of the problem with a small parameter $\mu > 0$ and considering the deformed solution $\widehat{J^{-1}} = J^T \left(JJ^T + \mu I\right)^{-1}$ because this inverse always exists, and the $\left|\frac{\mathrm{d}q_i}{\mathrm{d}s}\right|$ components are limited in this manner (e.g. [27, 5]). Since JJ^T is symmetric and positive semidefinite, it has nonnegative real eigenvalues. The method's drawback consists in distorting the solutions in the cases in which they exactly exist. The distortions can be roughly estimated as quantities depending on the ratio of μ and the smallest positive eigenvalue of JJ^T.

An alternative possibility is the application of the *Singular Value Decomposition (SVD)* of J as $J = U^T \Sigma V$, in which Σ has a diagonal form $\Sigma = \langle \sigma_1, \ldots, \sigma_k, 0, \ldots, 0 \rangle$ with the *singular values* $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$, U and V are orthogonal matrices of appropriate sizes. If Σ would be quadratic and invertible, the inverse would be $J^{-1} = V^T \Sigma^{-1} U$. The practical compromise consists in replacing the too small singular values in Σ with zeros, and using the "deformed inverse" as $\widehat{J^{-1}} = V^T \left\langle \sigma_1^{-1}, \ldots, \sigma_\ell^{-1}, 0, \ldots, 0 \right\rangle^T U$ in which $\ell \leq k$. Dropping the reciprocals of the too small singular values prevents the occurrence of too big values in the solution $\left|\frac{\mathrm{d}q_i}{\mathrm{d}s}\right|$ (e.g. [29]). Though from 1965 efficient algorithms are available for the execution of SVD [16], this algorithm is relatively complicated. Furthermore, for its reliable use precise information on $J(q)$ is needed in the given point.

Further alternative method was suggested in [44] that evaded the minimization of any cost function as in the Moore-Penrose pseudoinverse by the use of the *Gram-Schmidt Algorithm* that originally was invented by Laplace in 1820, and later was reinvented by Gram in 1883, and independently by Schmidt in 1907 [25, 17, 39].

Instead of the first order linearization in (2. 2) a complicated second order solution was suggested in 1991 [37]. Another complicated approach suggested in 1993 was the complex extension of the real q coordinates in solving (2. 2) that had to cope with the problem of the physical interpretation of the approximate solution.

Further alternative possibility is the use of *iterative approximation of the solution* instead using finite number of steps algorithms. Such methods have early examples in the 17th century (e.g. [46, 34, 18, 10]). For instance the *Newton – Raphson Algorithm* can quickly converge to the minimum of a positive definite quadratic expression based on the assumption that this minimum is exactly 0. However, for reliable convergence precise knowledge on the gradient of the scalar expression is needed. The essence of any "adaptive approach" is the assumption that only an imprecise, approximate mathematical model is available that can be used for "zero order calculations" that have to be amended and made more precise on the basis of actual measurements/observations.

The 1st step in this direction was made in 2016 when a fixed point iteration-based solution was suggested in [8] according to the scheme given in Fig. 1. In the *"Delay"* boxes a single step of the iteration can be understood, the beginning of the iteration $q_{i+1}(0)$ is the solution of the inverse kinematic task of the solution for $x_i^N = f(q_{i+1}(0))$ that is inherited from the previous cycle. The iterative sequence was generated by the *function of adaptive deformation* as

$$q_{i+1}(n+1) = G\left(q_{i+1}(n), f\left(q_{i+1}(n)\right), x_{i+1}^N\right) \quad . \tag{2. 4}$$

The function G is so constructed that the solution of the task, i.e. $q_{i+1}(\star)$ for which $f\left(q_{i+1}(\star)\right) = x_{i+1}^N$, is its *fixed point*, i.e. $G\left(q_{i+1}(\star), f\left(q_{i+1}(\star)\right), x_{i+1}^N\right) = q_{i+1}(\star)$. From Banach's fixed point theorem [2] it is well known that if $\mathcal{B}$ is a *complete, linear, normed metric space* (Banach Space), and $\Psi : \mathcal{B} \mapsto \mathcal{B}$ is a *contractive map*, i.e. $\exists 0 \leq K < 1$ so that $\forall a,\, b \in \mathcal{B}$: $\|\Psi(b) - \Psi(a)\| \leq K\|b-a\|$, the sequence generated from an arbitrary initial point $x_0 \in \mathcal{B}$ as $\{x_0, x_1 = \Psi(x_0), \dots, x_{n+1} = \Psi(x_n), \dots\}$ converges to the unique fixed point of this function $x_\star$, for which $\Psi(x_\star) = x_\star$. Therefore, for the convergence the function G in (2. 4) must be made contractive. This approach also had antecedents, for instance the Picard – Lindelöf theorem on the existence and uniqueness of the solution of certain ordinary differential equations was proved on the basis of similar considerations (e.g. [1]) that were later summarized and synthesized by Banach in 1922.

Under the assumption that during one digital control step only one step of the numerical iteration can be done for a slowly moving fixed point, the scheme in Fig. 1 was suggested for the use of adaptive dynamic control of single Degree of Freedom (DoF) dynamical systems in [43]. For the dynamic control of multiple DoF systems Dineva suggested a function in (2. 5). Its convergence properties were investigated in [13, 14, 12].

$$q(n+1) = \left[F\left(A\left\|f(q(n)) - x^N\right\| + x_\star\right) - x_\star\right] \frac{f(q(n)) - x^N}{\|f(q(n)) - x^N\|} + q(n) \quad , \tag{2. 5}$$

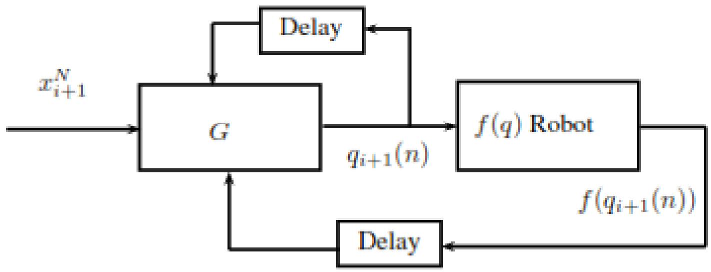

FIGURE 1. The scheme of the fixed point iteration-based solution

where $A \in \mathbb{R}$ is a *real adaptive parameter*, the differentiable function $F : \mathbb{R} \mapsto \mathbb{R}$ has a fixed point $x_\star = F(x_\star)$, and $\|\cdot\|$ is the Frobenius norm. Evidently, if $q_\star$ is the solution of the problem, i.e. $f(q_\star) = x^N$, for $q(n) = q_\star$ (2. 5) provides $q(n+1) = q_\star$, so the solution is a fixed point. Regarding the condition of convergence in the vicinity of the fixed point Dineva applied 1st order Taylor series approximation of $F(x)$ around $x_\star$, and $f(q)$ around $q_\star$ and arrived at the conclusion that the iterative sequence has the property that

$$q(n+1) - q_\star \approx \left[I + F'(x_\star) A \left. \frac{\partial f}{\partial q} \right|_{q_\star} \right] (q(n) - q_\star) \quad . \tag{2. 6}$$

By the use of the *Jordan canonical form* (e.g. [11]) –this term was coined by Shilov in 1977 in [40]– of the matrix $[\cdot]$ in (2. 6), she arrived at the conclusion that (2. 5) can be convergent nearby the fixed point if the real part of each eigenvalue of $\frac{\partial f}{\partial q}$ is simultaneously either positive or negative. (In these cases a small negative or positive adaptive parameter A can be chosen.)

In [8] it was shown that this convergence property is not satisfied even in the case of the Jacobian of the simplest 2 DoF arm. To tackle this problem, in the scheme of Fig. 1 instead of (2. 1) its modification in (2. 7) was considered

$$J^T(q) x^N(s) = J^T(q) f(q(s)) \ , \ x^N(s_{ini}) = f(q_{ini}) \ , \tag{2. 7}$$

in which in the 1st order Taylor series approximation of $f(q)$ in the vicinity of $q_\star$ the matrix $J^T(q)J(q_\star)$ occurs that is very close to a positive symmetric semidefinite matrix if $J(q)$ is precisely known. In contrast to the matrix inversion-based solutions that are apt to provide huge $\left|\frac{\mathrm{d}q}{\mathrm{d}s}\right|$ values near the singularity, our iterative method showed the pleasant "stagnation" of the critical coordinates.

However, if only an approximate model and an approximate Jacobian $\check{J}^T(q)$ is available, it cannot be taken for granted that $\check{J}^T(q)J(q_\star)$ will guarantee the convergence criterion set by Dineva in the modified problem $\check{J}^T(q)x^N(s) = \check{J}^T(q)f(q(s))$. Observing the fact that Dineva's above condition is rather satisfactory than necessary, since it works for arbitrary direction of the arrays $q(n+1)-q_{\star}$, though the iteration yields subsequent arrays the directions of which are "inherited", we tried to apply the modification of (2. 7) as

$$\sigma x^N(s) = \sigma f(q(s)) \ , \ x^N(s_{ini}) = f(q_{ini}) \ , \tag{2. 8}$$

in which was $\sigma = \pm 1$, depending on the direction of the previous step of the iteration. Though for simple 2 DoF systems this idea was able to work [20], for higher DoF systems and redundant robot arms no successful simulations were obtained. (This approach was very attractive since it promised the possibility of evading the burden of computing the Jacobian especially in the novel adaptive receding horizon controllers in which the use of Lagrange's General Reduced Gradient Method he invented for use in Classical Mechanics about 1811 [24]) was replaced by fixed point iteration (see [21, 22], and [23]). Therefore, we recapitulate the a more general case where the further modified problem in (2. 9) was considered in which in each step $n+1 \in \mathbb{N}$ of the iteration the multiple dimensional rotation matrix $\mathcal{N}$ "rotates back" the vector into the direction of the previous step, n [19]:

$$\mathcal{N}(n+1)\check{J}^T(q)x^N(s) = \mathcal{N}(n+1)\check{J}^T(q)f(q(s)) \ , \ x^N(s_{ini}) = f(q_{ini}) \ . \tag{2. 9}$$

It is easy to construct such an orthogonal transformation by the generalization of the Rodrigues formula published in 1840 [38]. Let us recapitulate the argumentation of [19]:

> "Consider the vectors $a, b \in \mathbb{R}^n$. At first remove the component parallel to b from a with parameter λ in the form: $a^{Mod} = a + \lambda b$ so that a^{Mod} must be orthogonal to b, that means for the scalar product that $b^T a^{Mod} = b^T a + \lambda b^T b = 0$. This leads to $\lambda = \frac{-b^T a}{b^T b}$. Then consider the *pairwisely orthogonal* unit vectors $e_a = \frac{a^{Mod}}{\|a^{Mod}\|}$, and $e_b = \frac{b}{\|b\|}$. The *skew symmetric matrix* $G \stackrel{def}{=} e_a e_b^T - e_b e_a^T$ generates rotations that mix the components of the vectors only in the two dimensional hyperplane spanned by these unit vectors. With a parameter $\xi \in \mathbb{R}$ these rotations have the form $\mathcal{O} = \exp(\xi G) \stackrel{def}{=} \sum_{s=0}^{\infty} \frac{\xi^s G^s}{s!}$. This matrix can be expressed in closed analytical form in similar manner ... Consider the various powers of G by taking into account that $e_a^T e_b = 0$, $e_a^T e_a = 1$, and $e_b^T e_b = 1$:

$$\begin{array}{c} G^2 = \left(e_a e_b^T - e_b e_a^T\right)\left(e_a e_b^T - e_b e_a^T\right) = -e_a e_a^T - e_b e_b^T \ , \\ G^3 = -\left(e_a e_a^T + e_b e_b^T\right)\left(e_a e_b^T - e_b e_a^T\right) = \\ = -\left(e_a e_b^T - e_b e_a^T\right) = -G \ , G^4 = -G^2 \\ G^5 = -G^3 = G \ , \text{etc.} \end{array} \tag{2. 10}$$

> By selecting the *even* and the *odd powers* of G it is obtained that

$$\mathcal{O} = I + \sin\xi G + (1-\cos\xi)G^2 \tag{2. 11}$$

The angle of rotation can be calculated by the scalar product of the appropriate vectors. In [19] good convergence was found with combination of this method with Dineva's iteration: the direction of the contraction was maintained though the appropriate transformation did not guarantee convergence for arbitrary directions.

In [7], for the purposes of adaptive dynamic control, a novel task transformation was suggested that used abstract rotations constructed as in (4. 17). Assume, that we wish to transform the array $b \in \mathbb{R}^n$ into the array $a \in \mathbb{R}^n$, ($\|a\| \neq \|b\|$). A possible solution is augmenting the dimension n of the vectors to $n+1$ by adding to them a new orthogonal dimension $a \mapsto A \in \mathbb{R}^{n+1}$, $b \mapsto B \in \mathbb{R}^{n+1}$ so that $\|A\| = \|B\| = R_a$ is a common absolute value. Then, according to Fig. 2, the rotation that rotates the array B into A can be constructed. As a consequence, the projection of the rotated vector in the original space will behave accordingly, i.e. b will be moved into a with simultaneous rotation and shrink or dilatation.

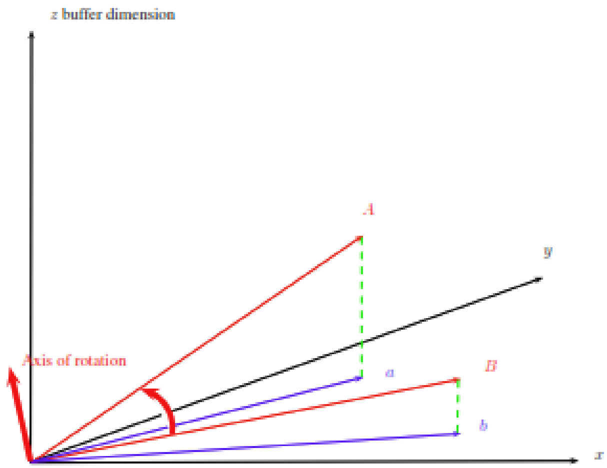

FIGURE 2. Schematic visualization of 2D rotations with a complementary buffer dimension (cited from [7])

The use of this scheme for adaptive control is quite simple: in the iteration in the previous step we observed that we need a rotation that transforms vector B into A, and this rotation has to be applied for a new vector $C \in \mathbb{R}^{n+1}$ where C is the augmented version of vector $c \in \mathbb{R}^n$. The angle of this new rotation can $\lambda_a \in \mathbb{R}$ times of the original rotation. It is very easy to understand the operation of this method. Similarly to the concept of the *increasing* $\mathbb{R} \mapsto \mathbb{R}$ functions, for the $\mathbb{R}^n \mapsto \mathbb{R}^n$ maps the concept of *"Approximately Direction Conserving Functions"* can be introduced for which $\forall x \neq 0 \; x^T f(x) > 0$. Geometrically this means that the angle determined by the vectors x and $f(x)$ is *acute*. In similar manner, it can be said that a function is *"Approximately Locally Direction Keeping"* at x if $\forall$ small $\Delta x \neq 0 \; \Delta x^T \Delta f = \Delta x^T [f(x + \Delta x) - f(x)] > 0$. Its geometric meaning is that the angle determined by the displacements Δx and Δf is *acute*, i.e. *a small modification in the argument of the function causes a displacement in the function value approximately in the same direction.* The 1st order Taylor series approximation of

$f(x)$ for that yields $\Delta x^T \Delta f \approx \Delta x^T \left[f(x) + \frac{\partial f}{\partial x}\Big|_x \Delta x - f(x) \right] = \Delta x^T \frac{\partial f}{\partial x}\Big|_x \Delta x > 0$. Because in this expression only the *symmetric part* of the matrix gives contribution, this condition means that $\frac{1}{2}\left[\frac{\partial f}{\partial x} + \left(\frac{\partial f}{\partial x}\right)^T\right]$ *must have positive eigenvalues.* For the control of such systems in the practice it is easy to develop iterative methods using observation-based learning. For instance, driving a new car by the simultaneous use of the accelerator/brake pedals and the steering wheel means a similar task. Though the numerical values may be quite different, the various cars *qualitatively behave in similar manner*, and on this basis their precise steering can be learned iteratively.

3. NOVELTIES IN THE PRESENT INVESTIGATIONS

In contrast to the solution used earlier, in the present paper we apply the abstract rotations-based adaptive transformation in combination with the rotations $\mathcal{N}$ in (2. 7). The kinematic construction of the 8 DoF redundant robot arm was modified, too, as follows: The open kinematic chain under consideration was described by the product of 8 homogeneous matrices as

$$\begin{pmatrix} r \\ 1 \end{pmatrix} = H^{(1)}(q_1)H^{(2)}(q_2)\cdots H^{(8)}(q_8)\begin{pmatrix} \tilde{r} \\ 1 \end{pmatrix} = H(q_1,\dots,q_8)\begin{pmatrix} \tilde{r} \\ 1 \end{pmatrix} \quad , \qquad (3.\ 12)$$

in which $\tilde{r} \in \mathbb{R}^3$ is vector of the last segment in the *"home position"* with respect to the last local system of coordinates, i.e. $\tilde{r}$ is constant, $H^{(i)}(q_i) \in \mathbb{R}^{4\times4}$ is the homogeneous matrix of the i^{th} segment, the upper left block of $H^{(i)}$ of size $\mathbb{R}^{3\times3}$, $O(e^{(i)}, q_i)$ is a rotational matrix that rotates around the unit vector $e^{(i)}$ with angle q_i (it is expressed by the use of the Rodrigues formula [38]), and its 4^{th} column is a shift parameter in the form $(r^{(i)T}, 1)^T \in \mathbb{R}^4$. Since the homogeneous matrices form a Lie group, $H(q_1,\dots,q_8)$ is a homogeneous matrix, too. Its upper left block of size $\mathbb{R}^{3\times3}$ is a rotational matrix that describes the "pose" of the last segment, and $r \in \mathbb{R}^3$ is the location of the endpoint with respect to the workshop reference frame.

The unit vectors of the home position of *the approximate ("canonical") model* as well as the shift parameters can be placed in the columns of size 3×8 matrices in which each column belongs to an arm segment (link) as follows:

$$\check{E} \stackrel{def}{=} \begin{pmatrix} 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & 1 & 1 & 0 & 0 & 1 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \\ 1 & 0 & 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \quad , \qquad (3.\ 13)$$

while the shift parameters were

$$\check{R} \stackrel{def}{=} \begin{pmatrix} 0 & \check{L}_1 & \check{L}_1 & \check{L}_1 & \check{L}_3 & \check{L}_3 & \check{L}_1 & \check{L}_3 \\ 0 & 0 & 0 & 0 & 0 & \check{L}_3 & -\check{L}_2 & \check{L}_2 \\ \check{L}_1 & 0 & 0 & \check{L}_2 & \check{L}_3 & 0 & \check{L}_3 & -\check{L}_1 \end{pmatrix} \qquad (3.\ 14)$$

with $\check{L}_1 = 1.0\,[m]$, $\check{L}_2 = 1.5\,[m]$, and $\check{L}_3 = 2.0\,[m]$. The exact unit vectors are the rotated versions of the available *approximate ones* in the columns of (3. 13) in a matrix $\check{E}$. The rotational angles of the units vectors around the workshop axles (φ_1 around X_1, φ_2 around X_2, and φ_3 around X_3) are given in Table 1.

TABLE 1. The rotations of the unit vectors of the rotational axles for the exact model correspond to the rotated version of the approximate ones as $[O = O_1(\varphi_1)O_2(\varphi_2)O_3(\varphi_3)]$, $e^{(i)} = O^{(i)}\breve{e}^{(i)}$

Rotational angle	$\varphi_1\,[rad]$	$\varphi_2\,[rad]$	$\varphi_3\,[rad]$
For $e^{(1)}$:	2×0.10	2×0.09	2×0.08
For $e^{(2)}$:	2×0.02	2×0.02	2×0.0
For $e^{(3)}$:	2×0.02	2×0.03	2×0.07
For $e^{(4)}$:	2×0.08	2×0.06	2×0.04
For $e^{(5)}$:	2×0.05	2×0.01	2×0.07
For $e^{(6)}$:	2×0.03	2×0.01	2×0.06
For $e^{(7)}$:	2×0.06	2×0.06	2×0.08
For $e^{(8)}$:	2×0.04	2×0.01	2×0.10

The counterpart of the approximate matrix $\breve{R}$ in (3. 15) is the *exact one* as

$$R \stackrel{def}{=} \begin{pmatrix} 0 & L_1 & L_1 & L_1 & L_3 & L_3 & L_1 & L_3 \\ 0 & 0 & 0 & 0 & 0 & L_3 & -L_2 & L_2 \\ L_1 & 0 & 0 & L_2 & L_3 & 0 & L_3 & -L_1 \end{pmatrix} \qquad (3.\ 15)$$

with $L_1 = 1.2\,[m]$, $L_2 = 1.8\,[m]$, and $L_3 = 2.2\,[m]$. The *approximate value of the last segment* was the "canonical" $\breve{\tilde{r}} = [2.5,\ 2.5,\ 2.5]\,[m]$ vector of equal components, while the *exact one* was $\tilde{r} = [2.6,\ 2.4,\ 2.3]\,[m]$ that *inevitably causes tracking error in the initial position* that later relaxes. For better relaxation in the first 10 discrete time-point 60 steps of the numerical iteration was applied, and later only 10 steps.

Regarding the problem solution, (2. 9) was further modified in (3. 16) as

$$\mathcal{WN}(n+1)\breve{J}^T(q)\mathcal{F}x^N(s) = \mathcal{WN}(n+1)\breve{J}^T(q)\mathcal{F}f(q(s)) \ , \qquad (3.\ 16)$$

in which $\mathcal{F}$ and $\mathcal{W}$ are *diagonal matrices of positive,* $0 < \mathcal{F}_{ii}, \mathcal{W}_{ii} \leq 1$ *elements*. The role of $\mathcal{F}$ is weighting the relative significance of the rotational pose and the location of the end-point in the solution. (We remind that f has 9 redundant components for the pose, and only 3 ones for the location.) The role of $\mathcal{W}$ is weighting the relative activities of the redundant joint coordinates in the disambiguation of the generally ambiguous solution.

4. SIMULATION RESULTS

For describing the simulations both programming and mathematical details deserve attention. Before presenting the computational results these issues will be briefly considered.

4.1. **On The Programming and Mathematical Details.** In the engineering practice the Matlab Simulink toolboxes are usually utilized when doing integration and differentiation tasks that often arise in solving control and differential inverse kinematic tasks, as e.g. in [33, 32]. While the Simulink package offers the advantage of graphical programming, various numerical integration packages are available to aid the engineers daily work. For the purposes of education certain Matlab packages are available at limited prices for the educational institutions and students. At Óbuda University the University covers the expenses of these license fees for its students, but after graduation the students have to pay these fees themselves. In the same time it worths noting that the solution of certain simpler problems does not need special packages and even graphical programming tools and can be well solved by simple Euler integration. For this purpose free software products are available that works quite efficiently, too.

The "Julia language" for instance is a high level programming language (developed at the MIT, Cambridge, USA) the syntax of which is very similar to that of the Matlab, but it runs almost as fast as a C code or an assembly code (benchmark data are available at [3]). Very advanced graphical options can be "imported" into the Julia by using graphical packages from Python. For our purposes a simple sequential code made in Julia was quite satisfactory. For instance, the abstract rotations quoted in Section 2 are realized by function code as follows:

```
function AdaptiveDeform(realized_prev,deformed_prev,desired_now)
  # Transforms the deformed_prev into the present deformed
  # as output
  # The applied rotation is taken from the rotation
  # rotating the realized_prev into the desired_now (from B to A)
  # The extended vectos
  A=zeros(9)
  A[1:8]=desired_now
  A[9]=sqrt(abs((R^2-(desired_now'*desired_now)[1])))
  B=zeros(9)
  B[1:8]=realized_prev
  B[9]=sqrt(abs((R^2-(realized_prev'*realized_prev)[1])))
  C=zeros(9)
  C[1:8]=deformed_prev
  C[9]=sqrt(abs((R^2-(deformed_prev'*deformed_prev)[1])))
  # The orthogonal vector parts: the part A orthogonal to B
  AortB=A-(B'*B)[1]*B/R^2
  # The orthogonal unit vectors
  norm_AortB=sqrt(abs((AortB'*AortB)[1]))
  ea=AortB/(epsa+norm_AortB)
  eb=B/R
  # The angle of rotation
  sin_fi=min(1.0,norm_AortB/R)
  fi=asin(sin_fi)
  # The generator of the rotation
  Gen=ea*eb'-eb*ea'
  Gen2=Gen*Gen
  # The generalized Rodrigues formula with the "interpolation"
  O=m_eye(9,9)+sin(lambda*fi)*Gen+Gen2*(1-cos(lambda*fi))
```

```
  Transformed=O*C # The transformed vector
  # Here only the corrections are limited
  ki=q_pp_max*tanh.(Transformed[1:8]/q_pp_max)
  return ki,lambda*fi
end
```

The Rodrigues formula for a constant given unit vector of rotary axle $e = [e_1, e_2, e_3]^T$ and a variable rotational angle ξ has the simple analytical form (4. 17):

$$\begin{aligned}
O(\xi, e) &= \exp(\xi G(e)) = I + \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \left(\xi - \frac{\xi^3}{3!} + \frac{\xi^5}{5!} \mp \dots\right) + \\
&+ \begin{bmatrix} e_1^2 - 1 & e_1 e_2 & e_1 e_3 \\ e_2 e_1 & e_2^2 - 1 & e_2 e_3 \\ e_3 e_1 & e_3 e_2 & e_3^2 - 1 \end{bmatrix} \left(\frac{\xi^2}{2!} - \frac{\xi^4}{4!} + \frac{\xi^6}{6!} \mp \dots\right) = \\
&= I + \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \sin\xi + \\
&+ \begin{bmatrix} e_1^2 - 1 & e_1 e_2 & e_1 e_3 \\ e_2 e_1 & e_2^2 - 1 & e_2 e_3 \\ e_3 e_1 & e_3 e_2 & e_3^2 - 1 \end{bmatrix} (1 - \cos\xi) \ .
\end{aligned} \tag{4. 17}$$

The homogeneous matrices, their derivatives according to their rotational angles and inverses in (3. 12) can be constructed in block form built up of the rotational matrices constructed according to (4. 17) and the constant shift components of the home position denoted by L as in (4. 18):

$$\begin{aligned}
H &= \left[\begin{array}{c|c} O(\xi, e) & L \\ 0^T & 1 \end{array}\right], \frac{\mathrm{d}H}{\mathrm{d}\xi} = \left[\begin{array}{c|c} \frac{\mathrm{d}O(\xi,e)}{\mathrm{d}\xi} & 0 \\ 0^T & 0 \end{array}\right], \\
H^{-1} &= \left[\begin{array}{c|c} O^{-1}(\xi, e) & -O^{-1}(\xi, e)L \\ 0^T & 1 \end{array}\right], \\
\frac{\mathrm{d}H}{\mathrm{d}\xi} H^{-1} &= \left[\begin{array}{c|c} \frac{\mathrm{d}O(\xi,e)}{\mathrm{d}\xi} O^{-1}(\xi, e) & -\frac{\mathrm{d}O(\xi,e)}{\mathrm{d}\xi} O^{-1}(\xi, e)L \\ 0^T & 0 \end{array}\right] = \\
&= \left[\begin{array}{c|c} \Omega(\xi, e) & -\Omega(\xi, e)L \\ 0^T & 0 \end{array}\right],
\end{aligned} \tag{4. 18}$$

in which $\Omega = \frac{\mathrm{d}O(\xi,e)}{\mathrm{d}\xi} O^{-1}(\xi, e)$ is a skew-symmetric matrix, i.e. a generator of the rotational matrices (an element of the tangent space of the rotational group at the identity element). Finally the Jacobian of the inverse kinematic task can formulated by finding the coefficients in the linear combination of the actual tangent vectors of the SE(3) Lie group at its identity element that must be identical with the tangent vector determined by the desired motion:

$$\begin{bmatrix} \dot{r}(t) \\ 0 \end{bmatrix} = \left(\dot{\xi}_1 \frac{\mathrm{d}H^{(1)}}{\mathrm{d}\xi_1} H^{(1)^{-1}} + \dot{\xi}_2 H^{(1)} \frac{\mathrm{d}H^{(2)}}{\mathrm{d}\xi_2} H^{(2)^{-1}} H^{(1)^{-1}} + \dots \right.$$
$$\left. + \dot{\xi}_n H^{(1)} H^{(2)} \cdots H^{(n-1)} \frac{\mathrm{d}H^{(n)}}{\mathrm{d}\xi_n} H^{(n)^{-1}} H^{(n-1)^{-1}} \cdots H^{(2)^{-1}} H^{(1)^{-1}} \right) \begin{bmatrix} r(t) \\ 1 \end{bmatrix} , \quad (4.\ 19)$$

in which the $\frac{\mathrm{d}H^{(i)}}{\mathrm{d}\xi_1} H^{(i)^{-1}}$ expressions are the tangents at the identity element, $H^{(1)} \left(\frac{\mathrm{d}H^{(2)}}{\mathrm{d}\xi_2} H^{(2)^{-1}} \right) H^{(1)^{-1}}$ is the 2nd tangent vector transformed by the group element $H^{(1)}$, therefore it is also a tangent at the identity element of the group SE(3), etc. In the Julia program the above matrices can be calculated in closed form, and the elements of the upper 3×3 block, and the $(4,1)$, $(4,2)$, $(4,3)$ elements can be arranged in the 12 rows of the Jacobian having 8 columns. From this point on the traditional matrix operations (e.g. SVD or calculation of the the Moore-Penrose pseudoinverse) can be applied for solving the redundant set of linear equations. In our case, the adaptive function detailed above can be called *within an internal cycle for each discrete point of the trajectory* as

```
r_prev,szog=AdaptiveDeform(W*O_next*transpose(J_Prev_a)*F*f(r_prev),r_prev,
W*O_next*transpose(J_Prev_a)*F*Target)
```

in which the program variable `W` stands for $\mathcal{W}$, and `F` corresponds to $\mathcal{F}$. In the sequel computational results will be revealed.

When it is assumed that the forward kinematic model of the robot is precisely known, therefore on its basis a real-time controller can be developed that directly can use the tracking error in the Cartesian Workshop coordinates as input data.

In the case that is investigated in this paper, the available forward kinematic model is not assumed to be reliably and precisely known, therefore on this basis the feedback errors in a single step cannot be used for control feedback. Instead of that, a grid is created for the nominal motion according to a scalar parameter s that can be the time t itself, or some nonlinear function of the time as $s(t)$. (In the paper the assumption that $s = t$ is used only for the seek of simplicity.) When the robot's joint coordinates are in the grid point n, and we wish to move to the next grid point $n + 1$, on the basis of the available kinematic model we could produce an inappropriate step that could have a great error in the grid point $n + 1$. The internal iteration is used for step-by step decreasing this error by observing the actual motion of the robot that can be quite slow. Following that, the investigation can be continued in the next grid point. As a result, the Cartesian nominal trajectory to be tracked, i.e. $x^N(s)$ is mapped to the *nominal joint coordinates* $q^N(s)$ over the grid. In the next step, a real-time control can be developed by using the $q^N(s(t))$ nominal trajectory, and the $q(t)$ joint coordinates that can be measured by encoders.

4.2. **Initial Tests.** In the first step initial tests were made to check the operation of the algorithm. In these tests one had trivial expectations for the nominal trajectory and its tracking.

To check the operation of the algorithm in the first step the *approximate, canonical model* was used for the generation of the nominal trajectory to be tracked. According to the canonical model X_3^N in the Cartesian coordinates must be constant since the rotation happens around an axis parallel to the vertical one of the workshop frame. Furthermore, the last link's pose suffers rotation around an axis parallel to X_3 of the workshop's frame of reference. In this case $\mathcal{F}$ and $\mathcal{W}$ were the identity matrices, i.e. no any weighting was applied. The results are given in Figs. 3, 4 that correspond to the expectations. The inevitable initial tracking error rapidly decreases and the orientation error is small, too. The solution in the joint coordinates of the robot are given in Fig. 5. The significance of the stabilizing "counter-rotation" and that of the abstract rotations applied in the FPI-based iteration are given in Fig. 6 for $R_a = 100$ "abstract radius" and $\lambda_a = 5\times10^{-4}$ extrapolation parameter. The resolution of the scalar parameter s was 10^{-3}.

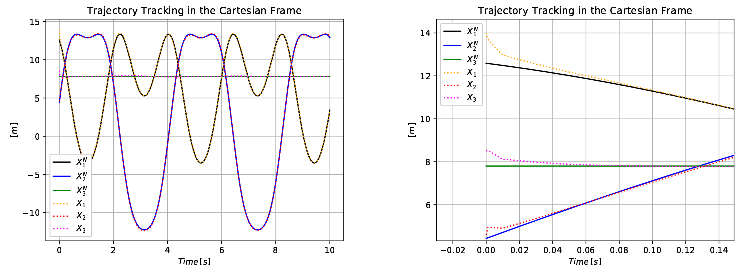

FIGURE 3. Tracking of a nominal trajectory generated by using only q_1 in the canonical approximate model

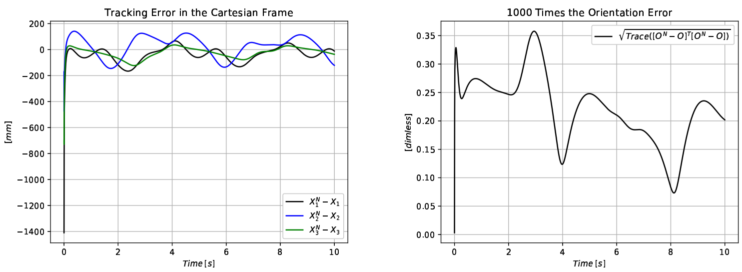

FIGURE 4. The tracking error of the end-point and the orientation for the nominal trajectory generated by using only q_1 in the canonical approximate model

In the next run for $i = 1 : 9$ the $\mathcal{F}_{ii}$ elements were reduced from 1 to 0.5. The fine details of the trajectory tracking in Fig. 7 can be compared with that in Fig. 4. The orientation precision really was degraded, and this effect shows some coupling with the tracking error of the position of the endpoint. Also, in Fig. 8 subtle differences appear in the joint coordinated of solution in comparison with Fig. 5.

In the next run $\mathcal{F} = I$ was restored and the last to diagonal elements in $\mathcal{W}$ were decreased to 0.01 to reduce the motion of the last two redundant joint coordinates q_7 and q_8. According to Fig. 9 the tracking precision remained good, and in Fig. 10 it can be seen that q_7 and q_8 were really "blocked".

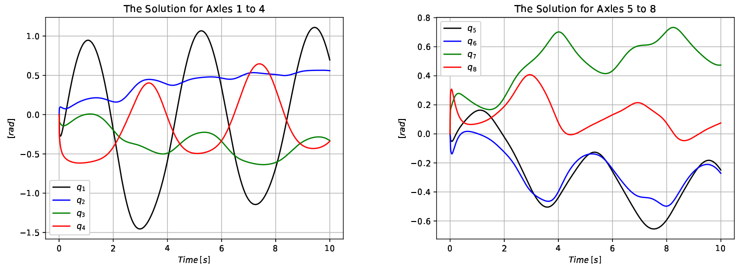

FIGURE 5. The solution in the space of the joint coordinates for the nominal trajectory generated by using only q_1 in the canonical approximate model

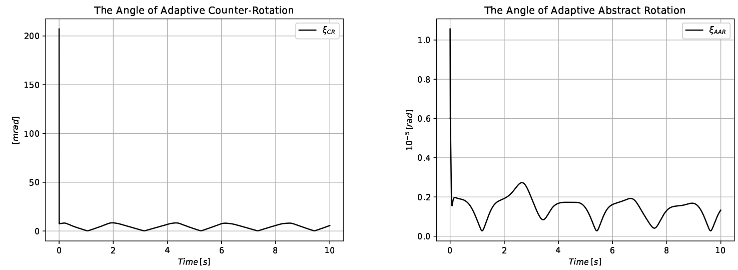

FIGURE 6. The angle of the “stabilizing rotation” $\mathcal{N}$ and the “abstract rotations” of the FPI-based algorithm for the nominal trajectory generated by using only q_1 in the canonical approximate model

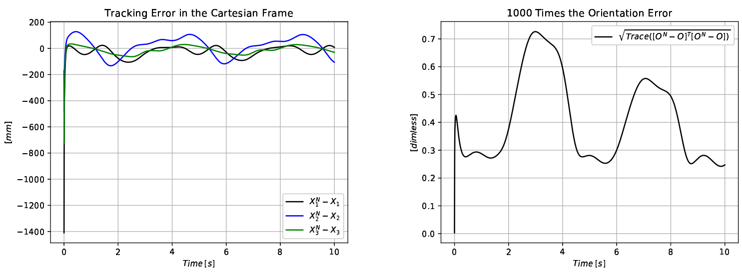

FIGURE 7. The tracking error of the end-point and the orientation for the nominal trajectory generated by using only q_1 in the canonical approximate model with reduced precision of the orientation

4.3. **Results for Nontrivial Trajectories.** In this test the *approximate, canonical* model was used for the generation of a nontrivial trajectory by moving only the generalized coordinates q_7 and q_8 simultaneously (Fig. 11). In this case variation of X_1^N, X_2^N, X_3^N were

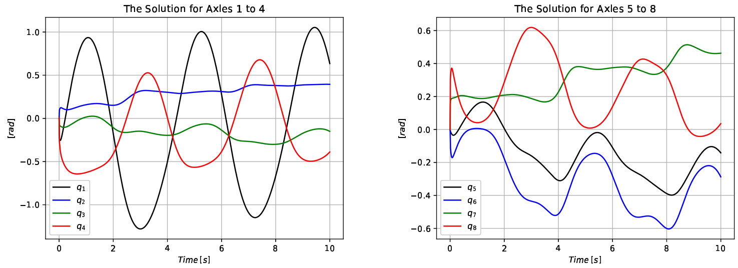

FIGURE 8. The solution in the space of the joint coordinates for the nominal trajectory generated by using only q_1 in the canonical approximate model with reduced precision of the orientation

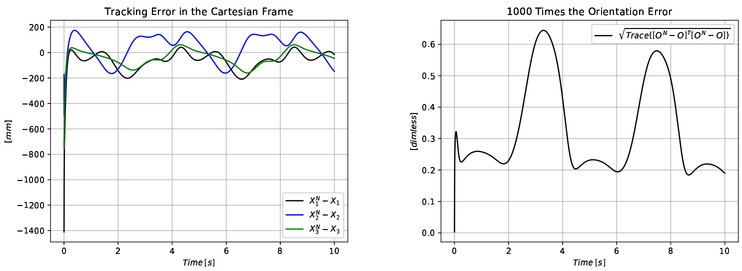

FIGURE 9. The tracking error of the end-point and the orientation for the nominal trajectory generated by using only q_1 in the canonical approximate model with reduced motion of q_7 and q_8

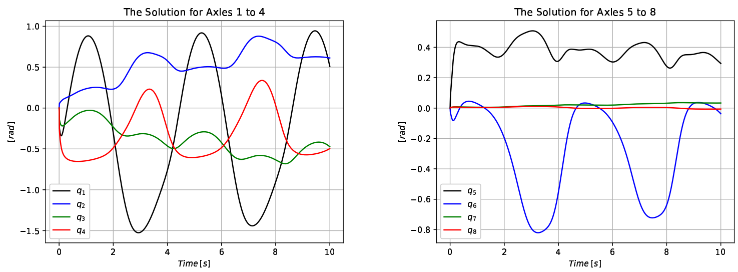

FIGURE 10. The solution in the space of the joint coordinates for the nominal trajectory generated by using only q_1 in the canonical approximate model with reduced motion of q_7 and q_8

expected, and a complicated modification in the orientation of the last link was caused due to the “general” orientation of the two last links in the home position.

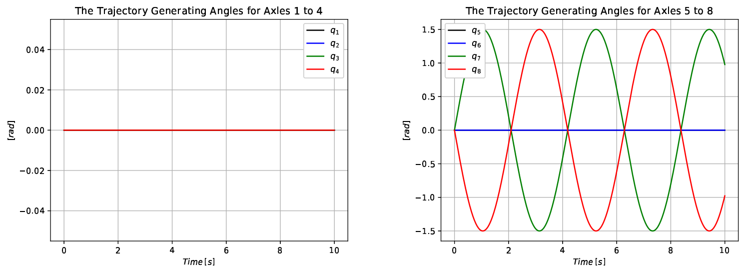

FIGURE 11. The joint coordinates for the generation of the nominal trajectory by using only q_7 and q_8 in the canonical approximate model

The variables $\mathcal{F}$ and $\mathcal{W}$ were restored to the identity matrix. Figures 12–15 reveal acceptable solutions.

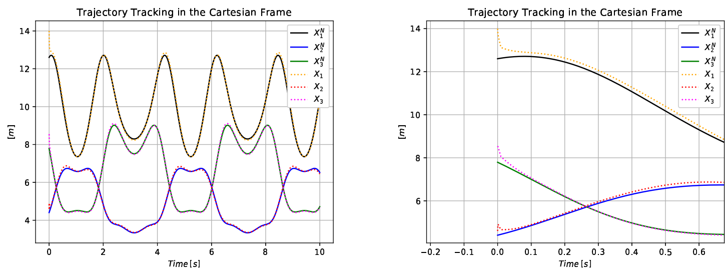

FIGURE 12. Tracking of a nominal trajectory generated by using only q_7 and q_8 in the canonical approximate model

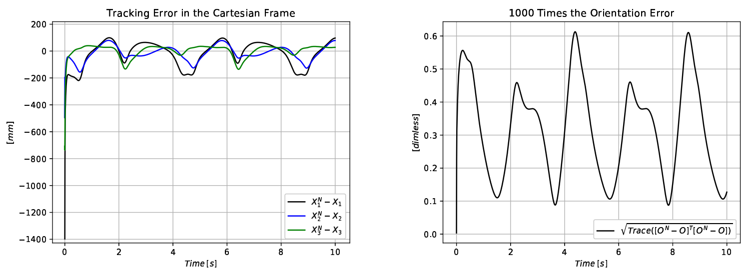

FIGURE 13. The tracking error of the end-point and the orientation for the nominal trajectory generated by using only q_7 and q_8 in the canonical approximate model

It is an interesting question to see how the trajectory generated by the motion of only q_7 and q_8 is tracked if the motion of the last two axles is "reduced" by using $\mathcal{W}_{7,7} = \mathcal{W}_{8,8} =$

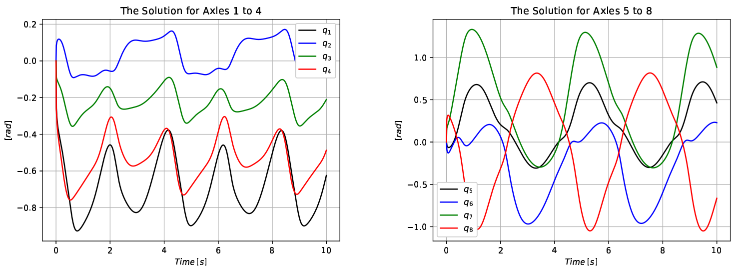

FIGURE 14. The solution in the space of the joint coordinates for the nominal trajectory generated by using only q_7 and q_8 in the canonical approximate model

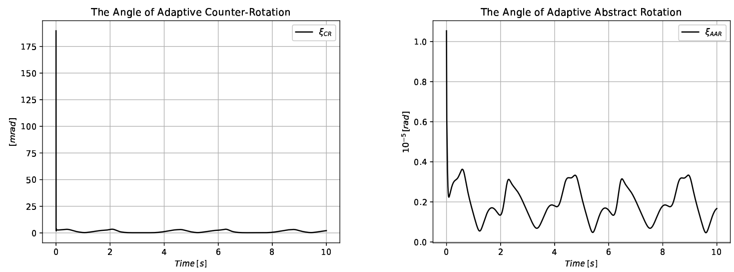

FIGURE 15. The angle of the "stabilizing rotation" $\mathcal{N}$ and the "abstract rotations" of the FPI-based algorithm for the nominal trajectory generated by using only q_7 andq_8 in the canonical approximate model

0.01. Figure 16 reveals degraded precision while in Fig. 17 the reduction of the motion of the last two links can be tracked. According to Fig. 18 it can be sated that the algorithm remained stable and convergent. The numerical conditions are characterized by Fig. 19 describing the minimum and the maximum of the real part of the eigenvalues of $\check{J}(q)^T J(q)$. Regarding the minimum, the numerical value that seems to be randomly scattered around 0 with the order of magnitude 10^{-14} practically means zero, i.e. the satisfactory condition of the convergence set by Dineva was not guranteed. It can be seen, too, that the maximum also varied within a wide range. This testifies that the suggested algorithm was able to successfully tackle a numerically "delicate", nontrivial problem.

5. CONCLUSION

In this paper a Fixed Point Iteration-based, matrix inversion-free algorithm was further refined and investigated for the adaptive numerical solution of the differential inverse kinematic task of redundant robots when the available kinematic model suffers from imprecisions. Such effects are important whenever the robot arm consists of long links, and the precision of manufacturing of the components is limited.

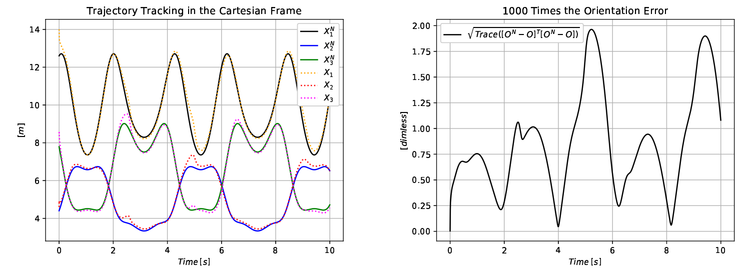

FIGURE 16. Tracking of a nominal trajectory generated by using only q_7 and q_8 in the canonical approximate model when the motion of the generating axles is reduced

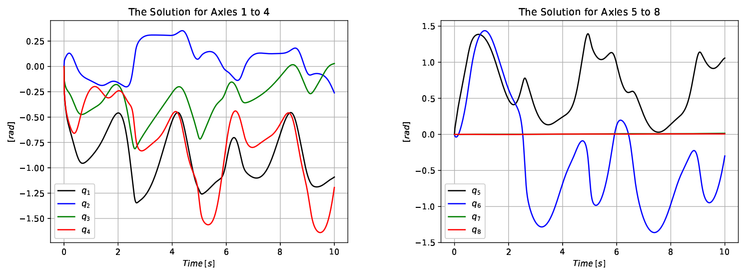

FIGURE 17. The solution in the space of the joint coordinates for the nominal trajectory generated by using only q_7 and q_8 in the canonical approximate model when the motion of the generating axles is reduced

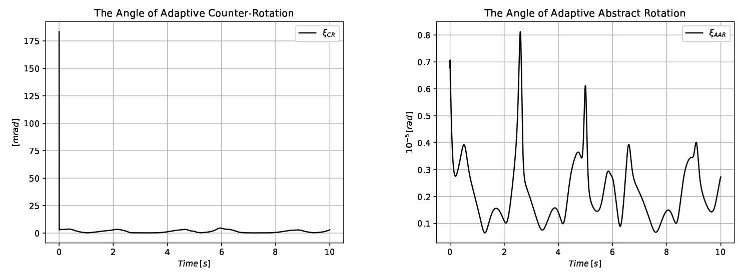

FIGURE 18. The angle of the “stabilizing rotation” $\mathcal{N}$ and the “abstract rotations” of the FPI-based algorithm for the nominal trajectory generated by using only q_7 and q_8 in the canonical approximate model when the motion of the generating axles is reduced

Based on the assumption that the position and the pose of the last link is precisely measurable, the effects of the modeling errors can be compensated adaptively. The main

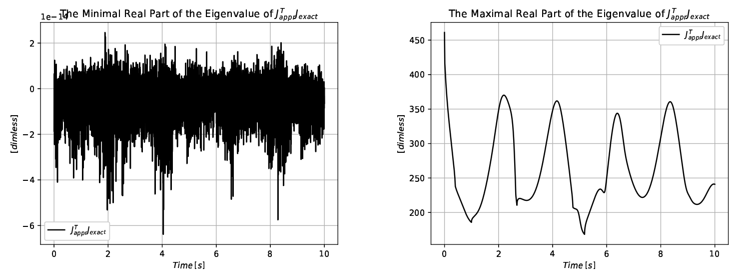

FIGURE 19. The minimal and maximal real part of the eigenvalues $\check{J}(q)^T J(q)$ for the nominal trajectory generated by using only q_7 and q_8 in the canonical approximate model when the motion of the generating axles is reduced

advantage in comparison with the matrix inversion-based solutions is that the suggested algorithm does not require the experimental investigation of the behavior of the robot arm for each independent direction during the measurements. So less hectic motion of the robot arm is required. The new algorithm also can be utilized in the process of more precise identification of the kinematic parameters. It can be used for a precise enough model, too.

According to the earlier simulations, the *"non plus ultra"* of the expectations, i.e. getting rid of the burden of computing the approximate Jacobian was viable only in the case of very low degree of freedom problems. So for a multiple degree of freedom case the calculation of at least the approximate Jacobian was found to be necessary.

The algorithm used abstract multiple dimensional rotations in two different phases of the calculations: in the calculation of the "counter-rotations" with the aim of guaranteeing the convergence of the algorithm, and in the fixed point iterations providing the solutions. It further was "colored" by the inclusion of parameters that affect the distribution of the ambiguous solution over the redundant axles, and influence a compromise between requiring higher or lower precision of the location and the pose of the last link.

As an application example, a redundant, 8 degree of freedom open kinematic chain was considered, in which the "counter-rotations" were realized in a 12 dimensional space, while the adaptive rotations were made in a 9 dimensional real space.

Regarding further research, we should like to return to the application of the same method in the adaptive solution of the optimal controllers in which the fixed point iteration-based solution could substitute Lagrange's "General Reduced Gradient Algorithm" by using a rough estimation for the Jacobian.

6. ACKNOWLEDGEMENT

The authors are highly thankful of the Doctoral School of Applied Informatics and Applied Mathematics Óbuda University Budapest, Hungary on providing a luminous environment for research.

REFERENCES

[1] V. ARNOLD, *Ordinary Differential Equations*, The MIT Press, 1975.

[2] S. BANACH, *Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales (About the Operations in the Abstract Sets and Their Application to Integral Equations)*, Fund. Math., **3** (1922)133–181.

[3] J. BEZANSON, A. EDELMAN, S. KARPINSKI, AND V. SHAH, *Julia*, `https://julialang.org`, (Last time visited: (2019).

[4] P. CHIACCHIO, S. CHIAVERINI, L. SCIAVICCO, AND B. SICILIANO, *Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy*, International Journal of Robotics Research, **10** (1991)410–425.

[5] S. CHIAVERINI, O. EGELAND, AND R. KANESTROM, *Achieving user-defined accuracy with damped least squares inverse kinematics*, 1991 IEEE International Conference on Robotics and Automation, (1991)672.

[6] R. CLAVEL, *Delta, a fast robot with parallel geometry*, In 18th Int. Symp. on Industrial Robot, (1988)91 – 100.

[7] B. CSANÁDI, P. GALAMBOS, J. TAR, G. GYÖRÖK, AND A. SERESTER, *A novel, abstract rotation-based fixed point transformation in adaptive control*, In the Proc. of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC2018),(2018)2577–2582.

[8] B. CSANÁDI, J. TAR, AND J. BITÓ, *Matrix inversion-free quasi-differential approach in solving the inverse kinematic task*, In Proc. of the 17^{th} IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2016),(2016)61–66.

[9] J. DENAVIT AND R. HARTENBERG, *A kinematic notation for lower-pair mechanisms based on matrices*, Trans ASME J. Appl. Mech, **23** (1955) 215–221.

[10] P. DEUFLHARD, *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms, Springer Series in Computational Mathematics, 35*, Springer, Berlin, (2004).

[11] J. DIEUDONNÉ, *Oeuvres de Camille Jordan I-IV*, Gauthier-Villars, Paris, 1961-1964.

[12] A. DINEVA, *Non-conventional Data Representation and Control (PhD Thesis, Supervisors: A.R. Várkonyi-Kóczy & J.K. Tar)*, Óbuda University, Budapest, Hungary, (2016).

[13] A. DINEVA, J. TAR, AND A. VÁRKONYI-KÓCZY, *Novel generation of Fixed Point Transformation for the adaptive control of a nonlinear neuron model*, In proc. of the IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong (SMC 2015), (2015)987–992.

[14] A. DINEVA, J. TAR, A. VÁRKONYI-KÓCZY, AND V. PIURI, *Generalization of a Sigmoid Generated Fixed Point Transformation from SISO to MIMO systems*, In Proc. of the IEEE 19^{th} International Conference on Intelligent Engineering Systems, Bratislava, Slovakia (INES 2015),(2015)135–140.

[15] O. EGELAND, *Task-space tracking with redundant manipulators*, IEEE Journal on Robotics & Automation, **3** (1987),471–475.

[16] G. GOLUB AND W. KAHAN, *Calculating the singular values and pseudoinverse of a matrix*, SIAM Journal on Numerical Analysis, **2** (1965)205–224.

[17] J. GRAM, *Über die Entwickelung reeler Funktionen in Reihen mittelst der Methode der kleinsten Quadrate*, Journal für die reine und angewandte Mathematik, **94** (1883)71–73.

[18] C. KELLEY, *Solving Nonlinear Equations with Newton's Method, no 1 in Fundamentals of Algorithms*, SIAM, (2003).

[19] H. KHAN, A. GALÁNTAI, AND J. TAR, *Adaptive solution of the inverse kinematic task by fixed point transformation*, In Proc. of the 15th IEEE International Symposium on Applied Machine Intelligence and Informatics, Herl'any, Slovakia, (2017)247–252.

[20] H. KHAN AND J. TAR, *Fixed point iteration-based problem solution without the calculation of the Jacobian*, In Proc. of the IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI 2019), Herl'any, Slovakia, (2019), 187–192.

[21] H. KHAN, J. TAR, I. RUDAS, AND G. EIGNER, *Adaptive model predictive control based on fixed point iteration*, WSEAS TRANSACTIONS on SYSTEMS and CONTROL, **12**(2017)347–354.

[22] H. KHAN, J. TAR, I. RUDAS, AND G. EIGNER, *Iterative solution in Adaptive Model Predictive Control by using Fixed-Point Transformation method*, International Journal of Mathematical Models and Methods in Applied Sciences, **12** (2018)7–15.

[23] H. KHAN AND J. K. TAR, *On replacing Lagrange's "Reduced Gradient Algorithm" by simplified fixed point iteration in adaptive model predictive control*, In Proc. of the 23^{rd} IEEE International Symposium on Intelligent Engineering Systems (INES 2019), Gödöllő, (2019)201–206.

[24] J. LAGRANGE, J. BINET, AND J. GARNIER, *Mécanique analytique (Eds. J.P.M. Binet and J.G. Garnier)*, Ve Courcier, Paris, (1811).

[25] P.-S. LAPLACE, *Théorie Analytique des Probabilités, in: Oeuvres completes de Laplace / publiés sous les auspices de l'Académie des sciences, par MM. les secrétaires perpétuel*, Gauthier-Villars, Paris, (1886).

[26] C. LEE AND M. ZIEGLER, *RSD-TR-1-83 A geometric approach is solving the inverse kinematics of PUMA robots*, The University of Michigan, Ann Arbor, Michigan (1983) 48109-1109.

[27] K. LEVENBERG, *A method for the solution of certain non-linear problems in least squares*, Quarterly of Applied Mathematics, **2** (1944) 164–168.

[28] M. LIE, *On a Class of Geometric Transformations (PhD Thesis)*, University of Christiania, Norway, 1871.

[29] A. MACIEJEWSKI AND C. KLEIN, *The singular value decomposition: computation and application to robotics*, The International Journal of Robotics Research, **8** (1989) 63–79.

[30] N. MANSARD, O. KHATIB, AND A. KHEDDAR, *A unified approach to integrate unilateral constraints in the stack of tasks*, IEEE Transactions on Robotics, **25** (2009)670–685.

[31] E. MOORE, *On the reciprocal of the general algebraic matrix*, Bulletin of the American Mathematical Society, **26** (1920) 394–395.

[32] C. A. MY, D. X. BIEN, B. H. TUNG, L. C. HIEU, N. V. CONG, AND T. V. HIEU, *Inverse kinematic control algorithm for a welding robot – positioner system to trace a 3d complex curve*, In Proc. of the 2019 International Conference on Advanced Technologies for Communications (ATC), (2019) 319–323.

[33] C. A. MY AND E. L. BOHEZ, *New algorithm to minimise kinematic tool path errors around 5-axis machining singular points*, International Journal of Production Research, **54** (2016) 5965–5975.

[34] J. ORTEGA AND W. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, SIAM, 2000.

[35] R. PENROSE, *A generalized inverse for matrices*, Proceedings of the Cambridge Philosophical Society, **51** (1955), 406–413.

[36] T. PETRIC AND L. ZLAJPAH, *Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem*, Robotics & Autonomous Systems, **61** (2013)948–959.

[37] E. POHL, *A new method of robotic rate control near singularities*, 1991 IEEE International Conference on Robotics and Automation, Pisa, Italy, (1991) 405.

[38] O. RODRIGUES, *Des lois géometriques qui regissent les déplacements d' un systéme solide dans l' espace, et de la variation des coordonnées provenant de ces déplacement considérées indépendent des causes qui peuvent les produire*, J. Math. Pures Appl., **5** (1840)380–440.

[39] E. SCHMIDT, *Zur Theorie der linearen und nichtlinearen Integralgleichungen. I. Teil: Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener*, Matematische Annalen, **63** (1907), 442.

[40] G. SHILOV, *Linear Algebra (Dover Books on Mathematics)*, Dover Publications, (1977)

[41] E. SIMETTI AND G. CASALINO, *A novel practical technique to integrate inequality control objectives and task transitions in priority based control*, Journal of Intelligent & Robotic Systems, **84** (2016)1–26.

[42] J. SOMLÓ, G. VARGA, M. ZENKL, AND B. MIKÓ, *The "Phantom" Delta Robot a new device for parallel robot investigations*, Acta Polytechnica Hungarica, **15** (2018)143–160.

[43] J. TAR, J. BITÓ, L. NÁDAI, AND J. TENREIRO MACHADO, *Robust Fixed Point Transformations in adaptive control using local basin of attraction*, Acta Polytechnica Hungarica, **6** (2009)21–37

[44] J. TAR, L. NÁDAI, I. FELDE, AND I. RUDAS, *Cost Function-free Optimization in Inverse Kinematics of Open Kinematic Chains, In Proc. of the 24^{th} International Workshop on Robotics in Alpe Adria Danube Region (RAAD 2015), May 27-29 2015, Bucharest, Romania, In T. Borangiu (ed.) Advances in Robot Design and Intelligent Control (within the series "Advances in Systems and Computing" 371) (2015)137-145.*

[45] J. WANG, Y. LI, AND X. ZHAO, *Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm*, International Journal of Advanced Robotic Systems, **7** (2010), 1–10.

[46] T. J. YPMA, Historical development of the Newton-Raphson method, SIAM Review, 37 (1995), 531–551