In: Theoretical Approaches in Non-Linear Dynamical Systems, Ed: Awrejcewicz, J. and Kazimierczak, M. and Mrozowski, J., pp 405-417. Dec 2-5, 2019, Lodz, POLAND. ISBN 978-83-66287-29-7.

Alternative inverse kinematic calculation methods in velocity and acceleration level

Dóra Patkó, Ambrus Zelei

Abstract: Inverse kinematics calculation of manipulators is a common building block in most of the robotic control processes. However, the numerical implementation of the inverse kinematics calculation has several alternatives yielding certain advantages and disadvantages.

This paper compares two approaches. In the classical one, the joint position increment is stepwise calculated based on the local velocity vector of the desired trajectory. In contrast, the joint position increment is obtained from the error between the desired and the realized trajectory in some alternative methods. The two approaches are also distinguished on the acceleration level. Our analytical and numerical studies show the benefits and drawbacks of these inverse kinematics methods.

1. Introduction

Kinematic calculation of robotic manipulators is a fundamental task in robot control, involving many issues such as numerical efficiency [8, 10]. In case of *direct kinematics*, the position and orientation of the end-effector is determined based on the joint coordinates. If the joint space and the task space (or workspace), where the end-effector is moving, have different dimensions, there is no kinematic redundancy [6, 7, 5] (joint space is higher dimension than the task space for redundant manipulators). In other words, degrees of freedom (DoFs) of the manipulator exactly equals the necessary DoFs for performing the task. We consider non-redundant cases only.

In practice, the desired position of the end-effector (tool-center-point) is prescribed and we try to find the corresponding joint coordinate values. This mapping from the task space to the joint space is called *inverse kinematics*. Many methods exist in the literature to solve the inverse kinematic problem with different approaches. These methods can be categorized as geometric level approaches, velocity level approaches and acceleration level approaches. Acceleration level resolution methods of the inverse kinematic problem may improve the performance of redundant robots [4, 9] comparing to velocity level methods. There are alternatives for the position error compensation too [2, 1]. In this paper, our focus is on the accuracy and efficiency of different combination of approaches used for inverse kinematics.

1.1. The idea of auxiliary input in motion control

For the sake of easy formalization of the inverse kinematic alternatives, we apply the idea of the auxiliary input that is used in inverse dynamics control of robots [10, 11]. The idea is summarized briefly in this subsection based on the literature. To this end, let us consider the general form of the equation of motion of a robot given as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{H}(\mathbf{q})\mathbf{u}\,. \tag{1}$$

By chosing the control input according to

$$\mathbf{u} = \mathbf{H}^{-1}(\mathbf{q})\left[\mathbf{M}(\mathbf{q})\tilde{\mathbf{v}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\right]\,, \tag{2}$$

the closed loop system assumes the following linear form: $\ddot{\mathbf{q}} = \tilde{\mathbf{v}}$. The term $\tilde{\mathbf{v}}$ in (2) is an auxiliary input, which is typically chosen as $\tilde{\mathbf{v}} = \ddot{\mathbf{q}}^{\mathrm{d}} - \tilde{D}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^{\mathrm{d}}) - \tilde{P}(\mathbf{q} - \mathbf{q}^{\mathrm{d}})$. This choice makes the error dynamics stable, when a desired trajectory $\mathbf{q}^{\mathrm{d}}(t)$ is tracked. This can be proven by substituting (2) into (1):

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} = \mathbf{H}\mathbf{H}^{-1}\left[\mathbf{M}\left(\ddot{\mathbf{q}}^{\mathrm{d}} - \tilde{D}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^{\mathrm{d}}) - \tilde{P}(\mathbf{q} - \mathbf{q}^{\mathrm{d}})\right) + \mathbf{C}\right]\,, \tag{3}$$

which leads to the stable error dynamics

$$\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^{\mathrm{d}} + \tilde{D}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^{\mathrm{d}}) + \tilde{P}(\mathbf{q} - \mathbf{q}^{\mathrm{d}}) = 0\,. \tag{4}$$

All in all, we can say that the auxiliary input $\tilde{\mathbf{v}}$ can be chosen arbitrarily depending on the control goal. The idea of the auxiliary input will be used in the subsequent sections.

2. Inverse kinematics calculation alternatives

Inverse kinematics is a fundamental process, when the task of the robot is transferred to the joint space from the workspace [6, 5, 10]. In case of a simple position trajectory tracking, the desired time history $\mathbf{r}^{\mathrm{d}}(t) \in \mathbb{R}^n$ of the tool-center-point (TCP) position (or end-effector position) is given in the workspace. The goal of the inverse kinematics calculation is to find the joint variables in $\mathbf{q} \in \mathbb{R}^n$ which satisfy the geometric equation for the position error $\mathbf{e}$

$$\mathbf{e}(\mathbf{q}, t) = \mathbf{0} \quad \text{with} \tag{5}$$

$$\mathbf{e}(\mathbf{q}, t) = \mathbf{r}(\mathbf{q}) - \mathbf{r}^{\mathrm{d}}(t). \tag{6}$$

The term $\mathbf{r}(\mathbf{q})$ in (6), which expresses the actual TCP position, depends on the joint coordinates and the geometry of the manipulator. The desired position $\mathbf{r}^{\mathrm{d}}(t)$ is purely time dependent. In the geometric level approach, (5) is solved directly with a proper non-linear root searching algorithm. Geometric solution is not typical in practice; the inverse kinematics is solved in velocity [12] or acceleration level [4, 9] as it is detailed in the followings.

2.1. The application of the idea of the auxiliary input on velocity level

By the time differentiation of (5), we obtain that the equation

$$\dot{\mathbf{e}}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0} \quad \text{with} \tag{7}$$

$$\dot{\mathbf{e}}(\mathbf{q}, \dot{\mathbf{q}}, t) = \dot{\mathbf{r}} - \dot{\mathbf{r}}^{\mathrm{d}} \tag{8}$$

must be satisfied in the velocity level approaches. Based on the system geometry, we know that the linear connection of the TCP position and the joint speed is

$$\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}\,, \tag{9}$$

where the Jacobian is $\mathbf{J}(\mathbf{q}) = \partial \mathbf{r}(\mathbf{q})/\partial \mathbf{q}$. Consequently, we can write that the commanded joint speed is

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\mathbf{v}_{\mathrm{v}}, \tag{10}$$

where $\mathbf{v}_{\mathrm{v}}$ is considered as an auxiliary input of the inverse kinematic control (such as $\tilde{\mathbf{v}}_{\mathrm{v}}$ in (2)). Based on (8), $\mathbf{v}_{\mathrm{v}} = \dot{\mathbf{r}}^{\mathrm{d}}$ is an obvious choice; however it does not guarantee the elimination of position error $\mathbf{e}$. Therefore there are many alternative possibilities.

In the followings, two alternatives are explained for the choice of $\mathbf{v}_{\mathrm{v}}$ with the possibility of the elimination of position errors.

2.1.1. Classical velocity level approach

By chosing the auxiliary input $\mathbf{v}_{\mathrm{v}}$ in (10) in such way that the TCP position error is maintained [12] ($\mathbf{v}_{\mathrm{v}} := \dot{\mathbf{r}}^{\mathrm{d}} - \kappa(\mathbf{r} - \mathbf{r}^{\mathrm{d}})$), we obtain the commanded joint velocity in the form:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\left(\dot{\mathbf{r}}^{\mathrm{d}} - \kappa(\mathbf{r} - \mathbf{r}^{\mathrm{d}})\right)\,. \tag{11}$$

Neglecting the digital effects, and by substituting (11) into (9), the TCP velocity can be expressed as:

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}^{\mathrm{d}} - \kappa(\mathbf{r} - \mathbf{r}^{\mathrm{d}})\,, \tag{12}$$

which leads to a stable error (see (6)) dynamics governed by the following first order differential equation with the solution $\mathbf{e}(t) = \mathbf{B}\mathrm{e}^{-\kappa t}$:

$$\dot{\mathbf{e}} + \kappa\mathbf{e} = \mathbf{0}\,, \tag{13}$$

if $\kappa > 0$. The stability analysis in the presence of digital effect is detailed in Section 3.

2.1.2. An alternative velocity level approach

Again, the question is that how we choose the auxiliary input $\mathbf{v}_\mathrm{v}$ appearing in (10). As it is explained in papers [2, 1], the main idea in this approach is to define the auxiliary input in such way that the position error is eliminated immediately if it is possible.

For the sake of precise explanation, we introduce the actual value of variables in the specific time instances: $\mathbf{q}_i := \mathbf{q}(t_i)$, $\mathbf{r}_i := \mathbf{r}(\mathbf{q}(t_i))$ and $\mathbf{r}_i^\mathrm{d} := \mathbf{r}^\mathrm{d}(t_i)$. We also introduce the timestep $h := t_{i+1} - t_i$.

We consider the error $\boldsymbol{\delta}_i = \mathbf{r}_{i+1}^\mathrm{d} - \mathbf{r}_i$ between the actual TCP position in the current timestep t_i and the desired TCP position in the upcoming timestep t_{i+1}. The goal is to eliminate $\boldsymbol{\delta}_i$. We assume constant TCP velocity $\dot{\mathbf{r}}_i$, with which the desired TCP position of the upcoming timestep $\mathbf{r}_{i+1}^\mathrm{d}$ can be approximately reached (see Fig. 1 left panel):

$$\mathbf{r}_{i+1}^\mathrm{d} = \mathbf{r}_i + h\dot{\mathbf{r}}_i\,. \tag{14}$$

By solving (14) for $\dot{\mathbf{r}}_i$, one can see that the error $\boldsymbol{\delta}_i$ is approximately eliminated if the auxiliary input in the time instant t_i is chosen as $\mathbf{v}_{\mathrm{v},i} = (\mathbf{r}_{i+1}^\mathrm{d} - \mathbf{r}_i)/h$. With this, the commanded joint speed (see (10)) reads:

$$\dot{\mathbf{q}}_i = \mathbf{J}_i^{-1}\,\frac{\mathbf{r}_{i+1}^\mathrm{d} - \mathbf{r}_i}{h}\,. \tag{15}$$

After a single explicit Euler [3] time integration step ($\mathbf{q}_{i+1} = \mathbf{q}_i + h\dot{\mathbf{q}}_i$) we obtain the commanded joint position in the upcoming time instant as:

$$\mathbf{q}_{i+1} = \mathbf{q}_i + \mathbf{J}_i^{-1}(\mathbf{r}_{i+1}^\mathrm{d} - \mathbf{r}_i)\,. \tag{16}$$

The resulting error of the TCP position is depicted in Fig. 1 right panel. The commanded joint coordinates would give the exact TCP positioning of the end-effector in case of linear relation of the joint coordinates and the TCP position. However, if $\mathbf{r}(\mathbf{q})$ is nonlinear, a certain error $\mathbf{e}_{i+1}$ occures. This error is small in case of small timeteps h. The thorough convergence analysis is presented in Section 3.

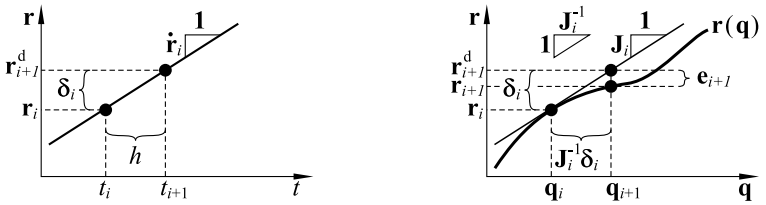

Figure 1. Left panel: sketch of the TCP velocity estimation. Right panel: sketch of the joint coordinate values and the TCP position after a single integration step.

2.2. The application of the idea of the auxiliary input on acceleration level

By the double time differentiation of (5), we obtain that the equation [4, 9]:

$$\ddot{\mathbf{e}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \mathbf{0} \quad \text{with} \tag{17}$$

$$\ddot{\mathbf{e}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \ddot{\mathbf{r}} - \ddot{\mathbf{r}}^{\mathrm{d}} \tag{18}$$

has to be satisfied in the acceleration level approaches. Similarly to (9), we know that the TCP acceleration can be expressed as

$$\ddot{\mathbf{r}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}\,. \tag{19}$$

From (19), we express the commanded joint acceleration as

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{v}_{\mathrm{a}} - \dot{\mathbf{J}}\dot{\mathbf{q}}), \tag{20}$$

where $\mathbf{v}_{\mathrm{a}}$ is again an auxiliary input of the inverse kinematic control. Based on (18), $\mathbf{v}_{\mathrm{a}} = \ddot{\mathbf{r}}^{\mathrm{d}}$ is an obvious choice. Since this choice does not guarantee the decaying of the position error $\mathbf{e}$, there are several alternatives. Two alternatives, which are capable of eliminating the position errors, are explained for the choice of $\mathbf{v}_{\mathrm{a}}$ in the followings.

2.2.1. Classical acceleration level approach

By chosing the auxiliary input $\mathbf{v}_{\mathrm{a}}$ in (20) in such way that the TCP position error is eliminated [4] ($\mathbf{v}_{\mathrm{a}} := \ddot{\mathbf{r}}^{\mathrm{d}} - \kappa_D(\dot{\mathbf{r}} - \dot{\mathbf{r}}^{\mathrm{d}}) - \kappa_P(\mathbf{r} - \mathbf{r}^{\mathrm{d}})$), we obtain the commanded joint acceleration

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}\left(\ddot{\mathbf{r}}^{\mathrm{d}} - \kappa_D(\dot{\mathbf{r}} - \dot{\mathbf{r}}^{\mathrm{d}}) - \kappa_P(\mathbf{r} - \mathbf{r}^{\mathrm{d}}) - \dot{\mathbf{J}}\dot{\mathbf{q}}\right). \tag{21}$$

Neglecting the digital effects, and by substituting (21) into (19), the TCP acceleration in the workspace can be expressed as:

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}^{\mathrm{d}} - \kappa_D(\dot{\mathbf{r}} - \dot{\mathbf{r}}^{\mathrm{d}}) - \kappa_P(\mathbf{r} - \mathbf{r}^{\mathrm{d}})\,, \tag{22}$$

which leads to a stable error (see (6)) dynamics with $\kappa_P > 0$ and $\kappa_D > 0$ governed by the following second order differential equation with the stable solution:

$$\ddot{\mathbf{e}} + \kappa_D\dot{\mathbf{e}} + \kappa_P\mathbf{e} = \mathbf{0}\,. \tag{23}$$

The error dynamics is assymptotically stable, if $\kappa_P > 0$ and $\kappa_D > 0$. The detailed stability analysis, in the presence of digital effect, is detailed in Section 3.

2.2.2. An alternative acceleration level approach

The goal is the elimination of the position error by chosing the auxiliary input appearing in (20) properly. Similarly, as in Section 2.1.2, we consider the error $\boldsymbol{\delta}_i = \mathbf{r}^{\mathrm{d}}_{i+1} - \mathbf{r}_i$ between the actual and the desired TCP position. We assume constant TCP acceleration $\ddot{\mathbf{r}}_i$, with which the desired TCP position of the upcoming timestep $\mathbf{r}^{\mathrm{d}}_{i+1}$ can be reached (see Fig. 2):

$$\mathbf{r}^{\mathrm{d}}_{i+1} = \mathbf{r}_i + h\dot{\mathbf{r}}_i + \frac{1}{2}h^2\ddot{\mathbf{r}}_i\,. \tag{24}$$

By solving (24) for $\ddot{\mathbf{r}}_i$, one can see that the error $\boldsymbol{\delta}_i$ is approximately eliminated if the auxiliary input is chosen as $\mathbf{v}_{\mathrm{a},i} = 2(\mathbf{r}^{\mathrm{d}}_{i+1} - \mathbf{r}_i)/h^2 - 2\dot{\mathbf{r}}_i/h$. Knowing that $\dot{\mathbf{r}}_i = \mathbf{J}_i\dot{\mathbf{q}}_i$, and using the joint speed estimation $\dot{\mathbf{q}}_i \approx \mathbf{J}^{-1}(\mathbf{r}^{\mathrm{d}}_{i+1} - \mathbf{r}_i)/h$, the commanded joint acceleration (see (20)) reads:

$$\begin{aligned}\ddot{\mathbf{q}}_i &= \frac{2\,\mathbf{J}_i^{-1}(\mathbf{r}^{\mathrm{d}}_{i+1} - \mathbf{r}_i)}{h^2} - \frac{2\dot{\mathbf{q}}_i}{h} - \mathbf{J}_i^{-1}\dot{\mathbf{J}}_i\dot{\mathbf{q}}_i = \\ &= \frac{2\,\mathbf{J}_i^{-1}(\mathbf{r}^{\mathrm{d}}_{i+1} - \mathbf{r}_i)}{h^2} - \frac{2\dot{\mathbf{q}}_i}{h} - \mathbf{J}_i^{-1}\dot{\mathbf{J}}_i\mathbf{J}^{-1}(\mathbf{r}^{\mathrm{d}}_{i+1} - \mathbf{r}_i)\,.\end{aligned} \tag{25}$$

The time integration is performed in the following two steps (see Adams-Moulton family of numerical integrators in [3]):

$$\dot{\mathbf{q}}_{i+1} = \dot{\mathbf{q}}_i + h\ddot{\mathbf{q}}_i\,, \tag{26}$$

$$\mathbf{q}_{i+1} = \mathbf{q}_i + h(\dot{\mathbf{q}}_{i+1} + \dot{\mathbf{q}}_i)/2\,. \tag{27}$$

After time integration, we obtain the commanded joint position in the upcoming time instant. Again, the nonlinearity of $\mathbf{r}(\mathbf{q})$ causes small errors in the TCP position. The smaller the timestep, the smaller the position error. The convergence analysis is detailed in Section 3.

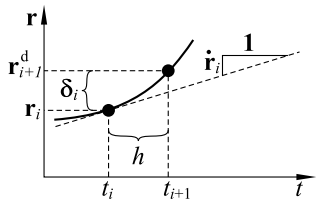

Figure 2. Sketch of the TCP acceleration estimation.

3. Case study examples with convergence analysis

The case study examples are shown in Fig. 3. For all case examples, the mapping of the commanded joint coordinates from the time instant t_i to t_{i+1} arises in the form:

$$\mathbf{q}_{i+1} = \mathbf{A}\mathbf{q}_i + \mathbf{b} \tag{28}$$

after linearizaton around an arbitrarily chosen point of the desired trajectory. Depending on the integration scheme and the formula for the auxiliary input $\mathbf{v}_{\mathrm{v}}$ or $\mathbf{v}_{\mathrm{a}}$, the coefficient matrix $\mathbf{A}$ and the vector $\mathbf{b}$ might change. The eigen-values of $\mathbf{A}$ defines the convergence of the inverse kinematic method, which are summarized in Section 3.4.

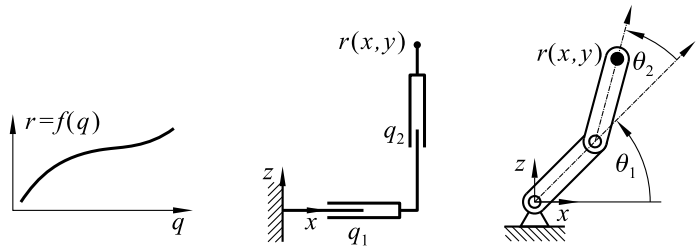

Figure 3. Case study examples: single DoF problem (in the left panel), Cartesian PP robotic arm (in the middle panel), RR robotic arm (in the right panel).

3.1. Case example: single DoF nonlinear system

The simplest (kinematically non-redundant) case example for testing the inverse kinematic methods is depicted in Fig. 3 left. A nonlinear connection of the single joint coordinate q and the TCP position r is given by $r = f(q)$.

3.1.1. Analytical study of the linearized mapping (single DoF example)

For the single DoF example, the mapping matrix $\mathbf{A}$ appearing in (28) is obtained for all four inverse kinematic approaches.

In the **classical velocity level** approach (abbreviated as cv), the numerical integration scheme was the second order Adams-Bashforth scheme [3]:

$$q_{i+1} = q_i + (3\dot{q}_{i+1} - \dot{q}_i)h/2\,. \tag{29}$$

This integration scheme is applied on the commanded joint velocity defined in (11) with which, the mapping matrix is obtained in the form

$$\mathbf{A}_{\mathrm{cv}} = \begin{bmatrix} (2-3\kappa h)/2 & -h/2 \\ -\kappa & 0 \end{bmatrix}. \tag{30}$$

In the **alternative velocity level** approach (abbreviated as av), the numerical integration is already incorporated in (16), for which the mapping matrix is:

$$\mathbf{A}_{\mathrm{av}} = \begin{bmatrix} 0 \end{bmatrix}. \tag{31}$$

In the **classical acceleration level** approach (abbreviated as ca), the numerical integration scheme was the second order Adams-Bashforth scheme combined with the second order Adams-Moulton scheme [3]:

$$\dot{q}_{i+1} = \dot{q}_i + (3\ddot{q}_{i+1} - \ddot{q}_i)h/2\,, \tag{32}$$

$$q_{i+1} = q_i + (\dot{q}_{i+1} + \dot{q}_i)h/2\,. \tag{33}$$

This integration scheme is applied on the commanded joint acceleration defined in (21) with which, the mapping matrix reads

$$\mathbf{A}_{\mathrm{ca}} = \begin{bmatrix} (4 - 3\kappa_P h^2)/4 & (4h - 3\kappa_D h^2)/4 & -h^2/4 \\ -3/2\,\kappa_P h & (2 - 3\kappa_D h)/2 & -h/2 \\ -\kappa_P & -\kappa_D & 0 \end{bmatrix}. \tag{34}$$

In the **alternative acceleration level** approach (abbreviated as aa), the commanded acceleration is defined by (25), and the numerical integration scheme is defined in (26) and (27), with which the mapping matrix is:

$$\mathbf{A}_{\mathrm{aa}} = \begin{bmatrix} 0 & 0 \\ -2/h & -1 \end{bmatrix}. \tag{35}$$

3.1.2. Simulations (single DoF example)

In the test simulations (see Fig. 4) and also in the above analytical calculations, the desired trajectory $r^{\mathrm{d}}(t)$ and the joint space $\mapsto$ workspace mapping function $f(q)$ were:

$$r^{\mathrm{d}}(t) = a_0 + a_1 \sin \omega t\,, \tag{36}$$

$$f(q) = c + \sqrt{q} \tag{37}$$

with the parameter values $a_0 = 1.2\,\mathrm{m}$, $a_1 = 0.6\,\mathrm{m}$, $\omega = 0.28\,\mathrm{rad/s}$ and $c = 0.2\,\mathrm{m}$. The simulation was performed with an initial error as it is shown in Fig 4. The control parameters and the timestep were $\kappa = 5.5\,1/\mathrm{s}$, $\kappa_P = 110\,1/\mathrm{s}^2$, $\kappa_D = 6.5\,1/\mathrm{s}$ and $h = 0.12\,\mathrm{s}$ respectively.

3.2. Case example: two DoF linear system

A two dimensional problem is introduced for test purposes. The example PP manipulator, which is illustrated in Fig. 3 middle, consists of two perpendicular prismatic drive. The TCP position $\mathbf{r} = [x, y]^{\mathrm{T}}$ is given by the mapping $\mathbf{r}(\mathbf{q}) = \mathbf{f}(q_1, q_2)$:

$$\mathbf{r}(\mathbf{q}) = \begin{bmatrix} l + q_1 \\ l + q_2 \end{bmatrix}, \tag{38}$$

where l is a geometric parameter.

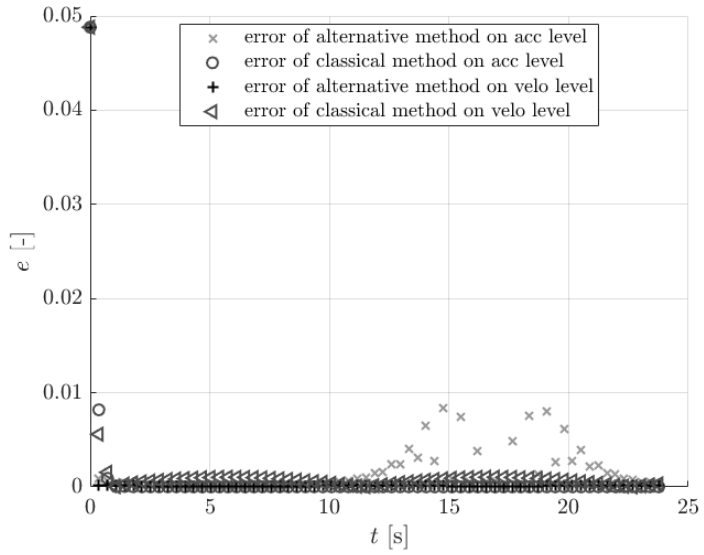

Figure 4. Position error for the 1 DoF case study example with stable parameters. Each inverse kinematics approach is reported here.

3.2.1. Analytical study of the linearized mapping (two DoF linear example)

The mapping matrices were obtained as it was explained in case of the single DoF example. Although, the size of the matrices are double because of the two DoFs:

$$\mathbf{A}_{\mathrm{cv}} = \begin{bmatrix} (2-3\kappa h)/2 & 0 & -h/2 & 0 \\ 0 & (2-3\kappa h)/2 & 0 & -h/2 \\ -\kappa & 0 & 0 & 0 \\ 0 & -\kappa & 0 & 0 \end{bmatrix}, \tag{39}$$

$$\mathbf{A}_{\mathrm{av}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{40}$$

$$\mathbf{A}_{\mathrm{ca}} = \begin{bmatrix} \frac{4-3\kappa_P h^2}{4} & 0 & \frac{4h-3\kappa_D h^2}{4} & 0 & -\frac{h^2}{4} & 0 \\ 0 & \frac{4-3\kappa_P h^2}{4} & 0 & \frac{4h-3\kappa_D h^2}{4} & 0 & -\frac{h^2}{4} \\ -\frac{3}{2}\kappa_P h & 0 & \frac{2-3\kappa_D h}{2} & 0 & -\frac{h}{2} & 0 \\ 0 & -\frac{3}{2}\kappa_P h & 0 & \frac{2-3\kappa_D h}{2} & 0 & -\frac{h}{2} \\ -\kappa_P & 0 & -\kappa_D & 0 & 0 & 0 \\ 0 & -\kappa_P & 0 & -\kappa_D & 0 & 0 \end{bmatrix}, \tag{41}$$

$$\mathbf{A}_{\mathrm{aa}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2/h & 0 & -1 & 0 \\ 0 & -2/h & 0 & -1 \end{bmatrix}. \tag{42}$$

The eigen-values of the matrices of the two DoF linear example are the same as the single DoF example, but there are twice as many of them because of the eigen-value multiplicity.

3.2.2. Simulations (two DoF linear example)

Firstly, we performed simulation (see Fig. 5 and Fig. 6 left) with parameters which are intentionally out of the stable range; i.e. the norm of the corresponding eigenvalues of $\mathbf{A}$ are larger than 1. The control parameters and the timestep were $\kappa = 17\,1/\mathrm{s}$, $\kappa_P = 400\,1/\mathrm{s}^2$, $\kappa_D = 6\,1/\mathrm{s}$ and $h = 0.0628\,\mathrm{s}$ respectively.

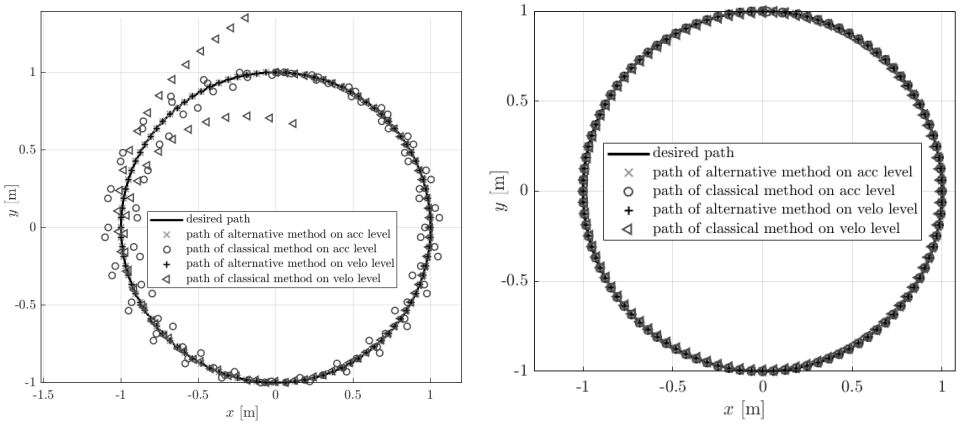

Figure 5. Left: Tracking of a test trajectory with badly tuned unstable parameters. Right: Tracking of a test trajectory with nearly optimal and stable parameters.

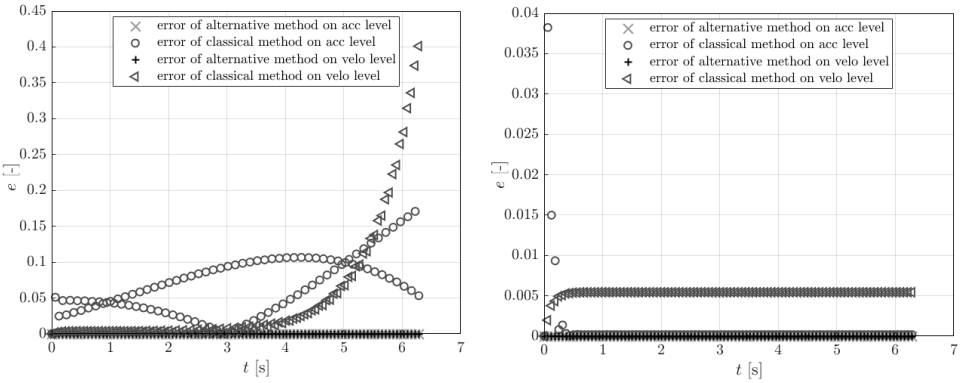

Figure 6. Left: Tracking error with badly tuned unstable parameters. Right: Tracking error with nearly optimal and stable parameters.

Secondly, the simulations were performed with stable parameters (see Fig. 5 and Fig. 6 right), with which the norm of the eigenvalues are smaller (or equal in alternative acceleration method) than 1. The timestep h was the same. The control parameters were $\kappa = 11.5\,1/\mathrm{s}$, $\kappa_P = 365\,1/\mathrm{s}^2$, $\kappa_D = 12.5\,1/\mathrm{s}$ respectively.

3.3. Case example: two DoF nonlinear system

A two dimensional non-redundant problem with geometric nonlinearity is shown in Fig. 3 right. The example system is a planar RR manipulator, consists of two rigid bars and the driven joints. The TCP position $\mathbf{r} = [x, y]^{\mathrm{T}}$ is given by $\mathbf{r}(\mathbf{q}) = \mathbf{f}(q_1, q_2)$:

$$\mathbf{r}(\mathbf{q}) = \begin{bmatrix} l\cos(q_1) + l\cos(q_2) \\ l\sin(q_1) + l\sin(q_2) \end{bmatrix}, \tag{43}$$

where l is a the length of the bars.

After defining a statical point in the workspace, the analytical calculations were performed similarly as in the previous case examples, and the resulting mapping matrices were obtained in exactly the same form (see (39), (40), (41) and (42)). Consequently the eigenvalues and the stable parameter regions are the same too.

Firstly, we performed simulation (see Fig. 7 and Fig. 8) with unstable parameters. The control parameters and the timestep were $\kappa = 5\,1/\mathrm{s}$, $\kappa_P = 61\,1/\mathrm{s}^2$, $\kappa_D = 3.8\,1/\mathrm{s}$ and $h = 0.2\,\mathrm{s}$ respectively.

Secondly, the simulations were performed with stable parameters. The timestep h was the same. The control parameters were $\kappa = 3.25\,1/\mathrm{s}$, $\kappa_P = 40\,1/\mathrm{s}^2$, $\kappa_D = 4\,1/\mathrm{s}$ respectively.

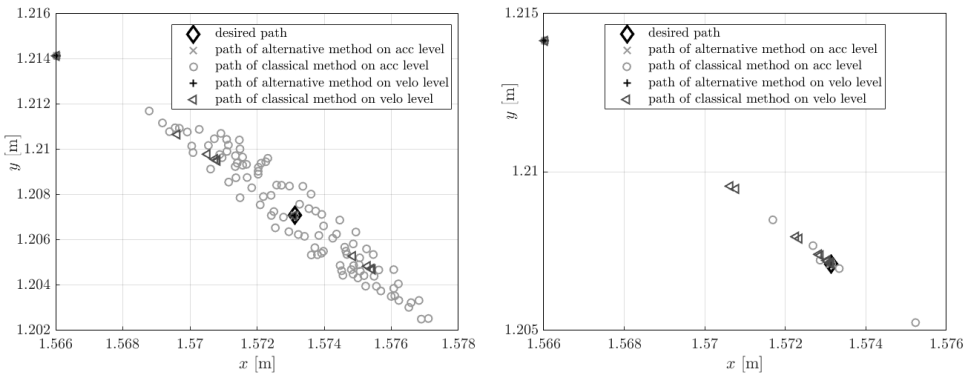

Figure 7. Left: Tracking of a test trajectory with badly tuned parameters. Right: Tracking of a test trajectory with nearly optimal and stable parameters.

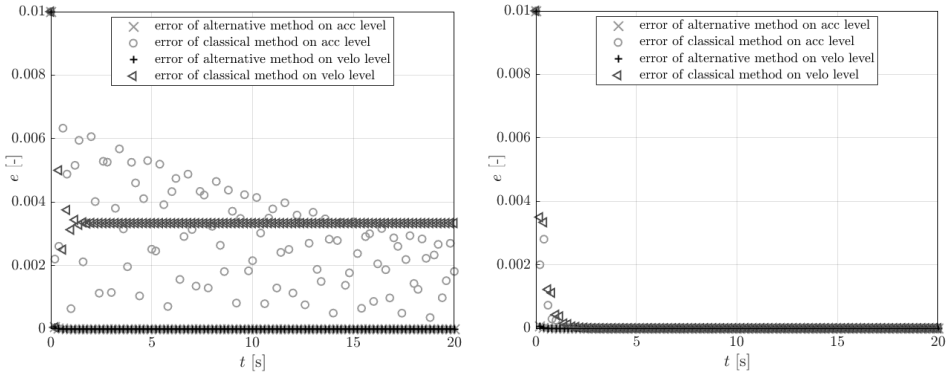

Figure 8. Left: Tracking error with badly tuned parameters. Right: Tracking error with nearly optimal and stable parameters.

3.4. Summary of the case examples

For all case examples, the same stability chart is obtained for the classical velocity and acceleration level approaches. These are reported in Fig. 9. The grey region is stable in the left panel for the classical velocity level approach. On the right panel, for different h values, different regions are plotted in the 2D space of the control parameters κ_P and κ_D.

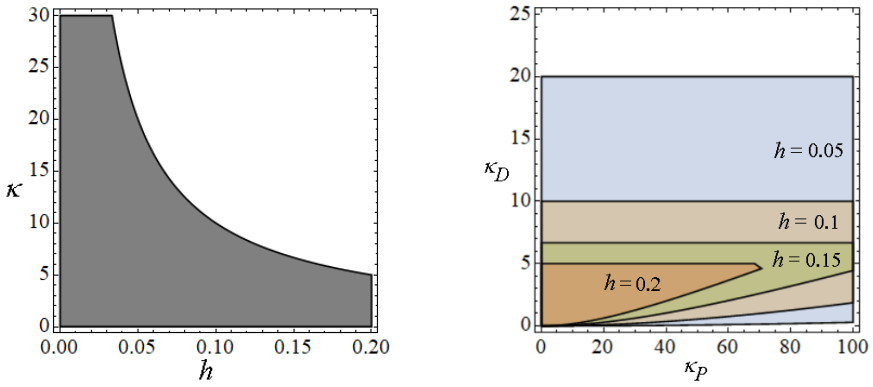

Figure 9. Left panel: stable region of the parameters h[s] and κ[1/s] in case of the classical velocity level methods. Right panel: stable region of the parameters h[s], κ_D[1/s] and κ_P[1/s^2] in case of the classical acceleration level methods.

For all case examples, the eigenvalue(s) for the alternative velocity level approach is/are always 0 regardless any parameter. The set of eigenvalues for the alternative acceleration method always consists of 0 and -1 values.

In simulations, the alternative acceleration level invers kinematic methods behaved more favorably than expected based on the analytical calculations. The linear stability was marginal; however, the trajectory tracking error decayed because of the nonlinear effects.

4. Conclusion

In the literature, the classical velocity and acceleration level approaches are analysed. The alternative velocity level method is available in the literature too. As a novelty, the acceleration level alternative method was tested.

Based on the simulations, we observed that the trajectory tracking error in the velocity level is smaller for the alternative method comparing to the classical one. However, in acceleration level, the classical method performs better, since the alternative method is marginally stable only.

The stable control parameter ranges were defined by the linearization of the numerical inverse kinematics formulae. The simulations showed stable operation when the control parameters and the time step were chosen from the theoretically stable domain.

Acknowledgments

This work was supported by the The Higher Education Excellence Program of the Ministry of Human Capacities in the frame of Biotechnology research area of Budapest University of Technology and Economics (BME FIKP-BIO), by the Hungarian National Research, Development and Innovation Office (Project id.: NKFI-FK18 128636), by the National Research, Development and Innovation Fund (TUDFO/51757/2019-ITM, Thematic Excellence Program) and by the Hungarian-Chinese Bilateral Scientific and Technological cooperation Fund under Grant No. 2018-2.1.14-TET-CN-2018-00008.

References

[1] Burrell, T., Montazeri, A., Monk, S., and Taylor, C. J. Feedback control-based inverse kinematics solvers for a nuclear decommissioning robot. *IFAC-PapersOnLine 49*, 21 (2016), 177–184. doi:10.1016/j.ifacol.2016.10.541.

[2] Buss, S. R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Tech. rep., IEEE Journal of Robotics and Automation, 2004.

[3] Butcher, J. *Numerical Methods for Ordinary Differential Equations*. John Wiley and Sons, Ltd, Hoboken, San Francisco, Weinheim, Queensland, Milton, Singapore, Mississauga, 2008. ISBN 978-0-470-72335-7.

[4] De Luca, A., Oriolo, G., and Siciliano, B. Robot redundancy resolution at the acceleration level. *Laboratory Robotics and Automation* (1992).

[5] From, P. J., and Gravdahl, J. T. General solutions to functional and kinematic redundancy. In *Proceedings of the 46th IEEE Conference on Decision and Control* (New Orleans, LA, USA, 12-14 December 2007), pp. 1–8. ISSN: 0191-2216, doi:10.1109/CDC.2007.4434442.

[6] Hollerbach, J. M., and Suh, K. C. Redundancy resolution of manipulators through torque optimization. *IEEE Journal of Robotics and Automation 3*, 4 (1987), 308–316. doi: 10.1109/JRA.1987.1087111.

[7] Nakamura, Y. *Advanced Robotics: Redundancy and Optimization.* Addison-Wesley, 1991.

[8] Orin, D. E., and Schrader, W. W. Efficient computation of the jacobian for robot manipulators. *The International Journal of Robotics Research 49*, 21 (1984), 66–75. doi:10.1177/027836498400300404.

[9] Reiter, A., Müller, A., and Gattringer, H. On higher-order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators. *IEEE Transactions on Industrial Informatics 14*, 12 (2018), 1681 – 1690. doi:10.1109/TII.2018.2792002.

[10] Siciliano, B., and Khatib, O. *Handbook of Robotics.* Springer, 2008, ISBN: 978-3-540-38219-5.

[11] Spong, M. W., and Vidyasagar, M. *Robot Dynamics and Control.* John Wiley & Sons, 1989.

[12] Wang, J., Li, Y., and Zhao, X. Inverse kinematics and control of a 7-dof redundant manipulator based on the closed-loop algorithm. *International Journal of Advanced Robotic Systems 7*, 4 (2010), 1–10.

Dóra Patkó, B.Sc.: Budapest University of Technology and Economics, Muegyetem rkp. 3. Budapest, H-1111, Hungary (*patkodora@edu.bme.hu*).

Ambrus Zelei, Ph.D.: MTA-BME Research Group on Dynamics of Machines and Vehicles, Muegyetem rkp. 3. Budapest, H-1111, Hungary (*zelei@mm.bme.hu*). The author gave a presentation of this paper during one of the conference sessions.