VOLUME 17, N° 3 2023
Journal of Automation, Mobile Robotics and Intelligent Systems

Singularity-Robust Inverse Kinematics for Serial Manipulators

Submitted: 4th December 2022; accepted: 21st April 2023

Ignacy Dulęba

DOI: 10.14313/JAMRIS/3-2023/21

Abstract:
This paper is a practical guideline on how to analyze and evaluate the literature algorithms of singularity-robust inverse kinematics or to construct new ones. Additive, multiplicative, and based on the Singularity Value Decomposition (SVD) methods are examined to retrieve well-conditioning of a matrix to be inverted in the Newton algorithm of inverse kinematics. It is shown that singularity avoidance can be performed in two different, but equivalent, ways: either via properly modified manipulability matrix or not allowing the decrease of the minimal singular value below a given threshold. It is discussed which method can always be used and which can only be used when some pre-conditions are met. Selected methods are compared to with respect to the efficiency of coping with singularities based on a theoretical analysis as well as simulation results. Also, some questions important for mathematically and/or practically oriented roboticians are stated and answered.

Keywords: *Serial manipulator, Forward kinematics, Jacobi matrix, Inverse kinematics, Singularities, Robustness, Evaluation*

1. Introduction

Robustness in robotics has got different meanings. Singularity-robust algorithms (methods) of inverse kinematics are aimed at coping with singular configurations without switching a model that works in a regular case into a much more complicated singular one. As problems with singularities appear only locally, so a slightly modified regular algorithm is able to generate a trajectory passing through a singular configuration. However, an inevitable decrease of a desired end-effector path tracking quality occurs.

Inverse kinematics is probably the most frequently solved task in robotics. The task arises when forward kinematics is defined for serial manipulators [13] or instantaneous kinematics for nonholonomic robots [12] and a configuration or a trajectory is to be found which corresponds to a given point/trajectory in a task-space.

Only for a very few simple robots, the task can be solved analytically. Usually, it is solved using the numerical Newton algorithm [11] which guarantees to obtain a solution when no singular configurations are encountered while generating consecutive configurations. At a singular configuration the algorithm becomes badly conditioned and, temporarily, special measures need to be taken to retrieve its well-conditioning. The simplest, and therefore the most frequently used, method to cope with this problem is to apply a robust version of the Newton algorithm. Around a current singular configuration, the method temporarily modifies a badly conditioned matrix as long as a neighborhood of the singularity is left.

In the robotic literature there are many methods of motion planning through singularities. In [16] Vargas et al. described a singularity approaching as a dynamic estimation of the inverse of the Jacobi matrix, arguing that their method requires less numbers of parameters to fix than other methods and avoids an explicit matrix inversion. Various methods of approximating inverse matrices without their explicit inversion are provided in [3]. Sun et al. [14] used a SVD-based method to optimize a cost function composed of tracking accuracy and velocity dumping terms with the goal to eliminate a terminal error after passing a singularity. The channel algorithm proposed by Duleba in [4] tries to jump through a singular configuration extrapolating behavior of a trajectory when the singular configuration was approached. All the discussed methods are based on the Jacobi, matrix which is just a linear term in the Taylor expansion of kinematics around a current configuration. Thus, the methods can be classified as first-order. Recently, Lloyd et al. [9] argued that second-order methods based also on the Hessian matrix of kinematics also have some advantages from a numerical point of view as an increase of the computational complexity in a single iteration can be compensated by the smaller number of iterations to complete solving the inverse kinematic task. There are also more mathematically advanced methods of motion planning through singularities. The normal form method [15] transforms original kinematics around a singular configuration into its particularly simple, normal form, solves the planning task in the new coordinates and moves back the solution into original coordinates. Unfortunately, this method is computationally demanding. Nevertheless, it allows one to trace a desired path with a high accuracy [15].

Depending on a manipulator redundancy, robust inverse kinematics methods modify either a Jacobi matrix (non-redundant case) or a manipulability matrix (the redundant case). In some practical situations any (sometimes random) modification of a badly conditioned matrix retrieves its well-conditioning.

However, only provably good solutions have got a theoretical value as they work also in the most demanding circumstances which can appear in practice.

Retrieving the well-conditioning of an inverted matrix can be realized in two ways: either an additive one via adding to the matrix a designed matrix, or a multiplicative one via multiplying the Jacobi matrix by a properly designed matrix.

The paper, being a substantially extended version of the conference paper [2], is organized as follows. In Section 2, preliminaries concerning the inverse kinematic task are recalled together with the Newton algorithm used to solve the task. In Section 3, a mathematical framework for designing robust additive or multiplicative matrix inversions was presented. In this section, the approach based on the Singular Value Decomposition (SVD) algorithm was examined. In this section, the geometrical interpretation is given based on how locally singularity-robust methods can impact possible directions of motion in a task-space and why new numeric problems can arise. Section 4 is devoted to the multi-criteria evaluation of the proposed techniques. In the simulation Section 5, based on the evaluation methodology developed, a standard (the literature based) method of singularity-robust inverse kinematics is compared to a method that increases minimal singular values of a Jacobi matrix. In Section 6, auxiliary issues are discussed related to the singularity-robust inverse kinematics yet outside the main scope of the paper. Answers for two questions important for mathematically oriented roboticians are given and two tips for practically oriented designers of algorithms are provided. Section 7 concludes the paper.

2. Inverse Kinematic Task and its Solution

Forward kinematics assigns to a configuration $\boldsymbol{q} = (q_1, \dots, q_n)^T$ from a configuration space $\mathbb{Q}$ a point $\boldsymbol{k}(\boldsymbol{q})$ in a task space $\mathbb{X}$

$$\boldsymbol{k} : \mathbb{Q} \ni \boldsymbol{q} \rightarrow \boldsymbol{k}(\boldsymbol{q}) \in \mathbb{X}, \quad \dim Q = n, \dim X = m. \tag{1}$$

A redundancy is described by the number $r = n - m$ and for redundant/non-redundant manipulators it takes the value of $r > 0$, $r = 0$, respectively. A basic (point) inverse kinematic task is defined as follows: for a given point $\boldsymbol{x}_f$ in a task-space find such a configuration $\boldsymbol{q}^\star$, that $\boldsymbol{k}(\boldsymbol{q}^\star) = \boldsymbol{x}_f$. An analytic solution of the inverse kinematic task is possible only for a very few, simple manipulators. Therefore instead of solving the task in positional spaces $(\boldsymbol{q}, \boldsymbol{x})$ the search is moved into velocity spaces $(\dot{\boldsymbol{q}}, \dot{\boldsymbol{x}})$ related with the Jacobi matrix

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{k}(\boldsymbol{q})}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \sum_{z=1}^{n} \boldsymbol{J}_{\star,z}(\boldsymbol{q})\dot{q}_z, \tag{2}$$

where $\boldsymbol{J}_{\star,z}(\boldsymbol{q})$ is the z-th column of the Jacobi matrix and $\dot{q}_z := dq_z/dt$. Then, the Newton algorithm is applied given by the iterative scheme [11]

$$\boldsymbol{q}_{i+1} = \boldsymbol{q}_i + \xi_i \cdot \boldsymbol{J}(\boldsymbol{q}_i)^{\$}(\boldsymbol{x}_f - \boldsymbol{k}(\boldsymbol{q}_i)), \tag{3}$$

where $\boldsymbol{q}_i$ is a configuration in the i-th iteration and ξ_i is a positive coefficient that impacts the speed of convergence of the algorithm. An initial configuration $\boldsymbol{q}_0$ is known and selected by a user. Depending on the redundancy, the matrix $\boldsymbol{J}^{\$}$ is expressed as

$$\boldsymbol{J}^{\$} := \begin{cases} \boldsymbol{J}^{-1} & \text{for non-redundant,} \\ \boldsymbol{J}^{\#} = \boldsymbol{J}^T(\boldsymbol{M})^{-1} & \text{for redundant} \end{cases} \tag{4}$$

manipulators, where the Moore-Penrose pseudo-inverse matrix $\boldsymbol{J}^{\#}$ is defined using a symmetric, positive semi-definite manipulability matrix [13]

$$\boldsymbol{M}(\boldsymbol{q}) := \boldsymbol{J}(\boldsymbol{q})\,\boldsymbol{J}(\boldsymbol{q})^T. \tag{5}$$

The algorithm (3) stops its run successfully when matrices (4) are of the full-rank in all iterations until a goal point $\boldsymbol{x}_f$ is reached with a prescribed accuracy δ, i.e.,

$$\|\boldsymbol{x}_f - \boldsymbol{k}(\boldsymbol{q}_i)\| < \delta.$$

The main point of interest in this paper is to analyze the Newton algorithm at and around singular configurations where the Jacobi matrix $\boldsymbol{J}$ drops its maximal possible rank and $\boldsymbol{J}^{\$}$ becomes ill-conditioned. The rank decrease is described by a corank defined as follows:

$$\operatorname{corank}(\boldsymbol{J}(\boldsymbol{q})) := c(\boldsymbol{q}) = m - \operatorname{rank}(\boldsymbol{J}(\boldsymbol{q})). \tag{6}$$

3. Robust Matrix Inversions

3.1. Additive Robust Inverse

Let $\boldsymbol{A}, \boldsymbol{M}$ denote symmetric, square matrices. $\boldsymbol{A}$ is a designed positively-definite matrix, $\boldsymbol{A} > 0$, while $\boldsymbol{M}$ is a positive semi-definite, manipulability matrix (5) $\boldsymbol{M} \geq 0$, which at singular configurations is degenerated, i.e. $\operatorname{rank}(\boldsymbol{J}) < m$, or, alternatively, $\det(\boldsymbol{M}) = 0$.

The key inequality used in designing robust inverse matrices is the following:

$$\det(\boldsymbol{A} + \boldsymbol{M}) > \det(\boldsymbol{A}) + \det(\boldsymbol{M}), \tag{7}$$

and it holds if only $\boldsymbol{M} \neq \boldsymbol{0}$.

Proof (based on the Grossmann's idea [6]): to simplify notations $\boldsymbol{B} := \boldsymbol{A}^{-1/2}$ is defined. Using properties of determinants and symmetric matrices one gets

$$\begin{aligned}
&\det(\boldsymbol{A} + \boldsymbol{M}) = \det(\boldsymbol{A}(\boldsymbol{I} + \boldsymbol{A}^{-1}\boldsymbol{M})) = \\
&\det(\boldsymbol{A}) \det(\boldsymbol{B}\,\boldsymbol{I}\,\boldsymbol{B}^{-1} + \boldsymbol{B}\,\boldsymbol{B}\,\boldsymbol{M}\,\boldsymbol{B}\,\boldsymbol{B}^{-1}) = \\
&\det(\boldsymbol{A}) \det(\boldsymbol{B}(\boldsymbol{I} + \boldsymbol{B}\,\boldsymbol{M}\,\boldsymbol{B})\boldsymbol{B}^{-1}) = \\
&\det(\boldsymbol{A}) \det(\boldsymbol{B}) \det(\boldsymbol{I} + \boldsymbol{B}\,\boldsymbol{M}\,\boldsymbol{B}) \det(\boldsymbol{B}^{-1}) = \\
&\det(\boldsymbol{A}) \det(\boldsymbol{I} + \boldsymbol{B}\,\boldsymbol{M}\,\boldsymbol{B}) > \\
&\det(\boldsymbol{A})(\det(\boldsymbol{I}) + \det(\boldsymbol{B}\,\boldsymbol{M}\,\boldsymbol{B})) = \\
&= \det(\boldsymbol{A}) + \det(\boldsymbol{M}) > 0.
\end{aligned} \tag{8}$$

In the chain of transformations (8), the only non-obvious transformation is the first strong inequality occurrence. The matrix $\boldsymbol{B}\,\boldsymbol{M}\,\boldsymbol{B} \neq \boldsymbol{0}$ is symmetric, positive semi-definite with real and non-negative eigenvalues (singular values) collected in the vector $\boldsymbol{S} = (\sigma_1, \dots, \sigma_m)^T$, $\sigma_i \geq 0$, $i = 1, \dots, m$, not all equal to zero (at least $\sigma_1 > 0$). This matrix can be expressed in the form

$$\boldsymbol{B\,M\,B} = \boldsymbol{U}\ \mathrm{diag}(\boldsymbol{S})\ \boldsymbol{U}^T, \tag{9}$$

with a non-singular rotation matrix $\boldsymbol{U} \in \mathbb{SO}(m)$. The identity matrix can be expressed as $\boldsymbol{I} = \boldsymbol{UU}^T$, and eigen-(singular)values of positive semi-definite matrix $\boldsymbol{I} + \boldsymbol{B\,M\,B}$ are equal to $\mathrm{diag}(\boldsymbol{1}) + \boldsymbol{S}$. Finally, using properties of determinants of similar matrices and eigenvalues of matrices one gets

$$\det(\boldsymbol{I} + \boldsymbol{B\,M\,B}) = \prod_{i=1}^{m}(\sigma_i + 1) = 1 + \sum_{i=1}^{m}\sigma_i + \\ + \sum_{i,j=1,i>j}^{m}\sigma_i\sigma_j + \sum_{i,j,k=1,i>j>k}^{m}\sigma_i\sigma_j\sigma_k + \ldots + \prod_{i=1}^{m}\sigma_i = \\ 1 + \prod_{i=1}^{m}\sigma_i + R(\boldsymbol{\sigma}) > 1 + \prod_{i=1}^{m}\sigma_i = \det(\boldsymbol{I}) + \det(\boldsymbol{B\,M\,B}). \tag{10}$$

The inequality (7) guarantees positive-definiteness of the sum of matrices when only a positive-definite matrix is added to the manipulability matrix. This way generalized inverse matrix (4) can be applied. Apparently, the additive disturbance should be small not to cause inaccuracies in tracking a desired path towards the goal point. A very general method to generate the disturbing matrix with desired properties is to take any $(m \times n)$ full-rank matrix $\boldsymbol{B}$ and obtain always positive-definite matrix $\boldsymbol{BB}^T$. The simplest possible choice, and frequently encountered in the robotic literature [11], takes the form

$$\boldsymbol{A} := \lambda\, \boldsymbol{I}_m, \tag{11}$$

with a small positive design parameter λ and the $(m \times m)$ identity matrix $\boldsymbol{I}_m$ [13]. However, the drawback of (11) is the identical handling of all coordinates and a weak relationship with kinematics for which the disturbance is applied. Another natural disturbance definition proposed below does not suffer from the aforementioned drawback (11)

$$\boldsymbol{A} := \lambda \cdot \mathrm{diag}(\langle \boldsymbol{J}_{1,\star}, \boldsymbol{J}_{1,\star}\rangle, \ldots, \langle \boldsymbol{J}_{m,\star}, \boldsymbol{J}_{m,\star}\rangle) \tag{12}$$

where $\langle\cdot,\cdot\rangle$ denotes a dot product and $\boldsymbol{J}_{i,\star}$ is the i-th row of the Jacobi matrix. The expression (12) depends on kinematics and potentially increases items on the main diagonal of the manipulability matrix. Unfortunately, for any model of kinematics there is no guarantee that all rows of the the Jacobi matrix have got a non-zero length. To avoid this extremely rare but still possible case, the always positive kinematic-dependent disturbance function can be defined as

$$\boldsymbol{A} := \lambda \cdot \mathrm{diag}(f_\epsilon(\boldsymbol{J}_{1,\star}), \ldots, f_\epsilon(\boldsymbol{J}_{m,\star})), \tag{13}$$

where

$$f_\epsilon(\boldsymbol{J}_{i,\star}) := \max\{\langle \boldsymbol{J}_{i,\star}, \boldsymbol{J}_{i,\star}\rangle, \epsilon\} \tag{14}$$

with a small, positive design parameter ϵ. The robust inversion (13) shares advantages of (11), (12) while avoiding their drawbacks.

It is worth mentioning that Equation (7) offers more than what is necessary to retrieve the well-conditioning of the inverted matrix

$$\det(\boldsymbol{A} + \boldsymbol{M}) > \det(\boldsymbol{A}) + \det(\boldsymbol{M}) \geq \det(\boldsymbol{A}) > 0. \tag{15}$$

It appears that at singular configurations the badly conditioned matrix $\boldsymbol{M}$ actively supports avoiding singularities because from Equations (8), (10), and (15) it follows that

$$\det(\boldsymbol{A} + \boldsymbol{M}) - \det(\boldsymbol{A}) = \det(\boldsymbol{A})R(\boldsymbol{\sigma}) > 0. \tag{16}$$

3.2. Multiplicative Robust Inverse

In the multiplicative case, the robust inversion of the manipulability matrix is searched for by disturbing factors of the matrix rather than by modifying the matrix itself. The first possibility is to modify the matrix $\boldsymbol{J}$, by adding the $(m \times n)$ matrix $\boldsymbol{\Psi}$ to get the well-conditioned manipulability matrix

$$(\boldsymbol{J} + \boldsymbol{\Psi})(\boldsymbol{J} + \boldsymbol{\Psi})^T = \boldsymbol{JJ}^T + \boldsymbol{J\Psi}^T + (\boldsymbol{J\Psi}^T)^T + \boldsymbol{\Psi\Psi}^T. \tag{17}$$

The general inequality (7) does not work directly for (17) as there are four items instead of two. The first term is badly conditioned at singular configurations while the last term seems to be a good candidate for a positive-definite matrix if only $\boldsymbol{\Psi}$ is non-singular, i.e., $\mathrm{rank}(\boldsymbol{\Psi}) = m$). So, two mid-terms should be eliminated either by zeroing their sum or one of them. The second condition seems to be easier to satisfy but it is stronger than the first one. The condition

$$\boldsymbol{J}\,\boldsymbol{\Psi}^T = \boldsymbol{0}_m \tag{18}$$

means that rows of $\boldsymbol{\Psi}$ are perpendicular to the rows of $\boldsymbol{J}$, so they belong to the null-space of the Jacobi matrix. The dimension of the null space at regular (non-singular) configurations is equal to $r = n - m$ and in typical singular configurations it is increased by $c \in \{1, 2\}$, cf. Equation (6). Thus, the rank of the matrix $\boldsymbol{\Psi}$ is upper-bounded by $c + r$. On the other hand, it was assumed that $\mathrm{rank}(\boldsymbol{\Psi}) = m$. In the vast majority of practical cases, the condition

$$r + c - m \geq 0. \tag{19}$$

cannot be satisfied (example: $n = 4, m = 3, c = 1$). Consequently, this multiplicative method of making the manipulability matrix robust seems to lack generality.

3.3. Robust Inverse Using SVD

Another version of multiplicative modification of the manipulability matrix extensively uses the Singular Value Decomposition algorithm [5] introduced in robotics by Maciejewski and Klein [10]. The SVD decomposes the Jacobi matrix into a product of three matrices

$$\boldsymbol{J} = \boldsymbol{U} \cdot [\boldsymbol{D}, \boldsymbol{0}_{m,r}] \cdot \boldsymbol{V}^T, \tag{20}$$

where $\boldsymbol{0}_{m,r}$ denotes a $(m \times r)$ matrix composed of zeroes only, $\boldsymbol{U} \in \mathbb{SO}(m)$ and $V \in \mathbb{SO}(n)$ are rotation matrices in m and n dimensional spaces, respectively, and a $(m \times m)$ diagonal matrix $\boldsymbol{D} = \operatorname{diag}(d_i)$, $i = 1, \dots, m$, collects non-negative singular values ordered in non-ascending order $d_i \geq d_j$ when $i < j$. A clear advantage of the decomposition (20) is that the pseudo-inverse matrix at regular (i.e., non-singular) configurations can be obtained immediately as follows:

$$\boldsymbol{J}^{\#} = \boldsymbol{V} \cdot \begin{bmatrix} \boldsymbol{D}^{-1} \\ \boldsymbol{0}_{r,m} \end{bmatrix} \cdot \boldsymbol{U}^T, \tag{21}$$

where $\boldsymbol{D}^{-1} = \operatorname{diag}(1/d_i)$, $i = 1, \dots, m$.

From a numerical perspective, the detection of singularities requires dropping any of the singular values below a given design parameter $d_{\min}$, describing a safety margin from singularities. Therefore, a method to avoid singularities should modify those singular values that do not exceed the threshold value. In this method $\boldsymbol{D}^{-1}$ in Equation (21) is replaced with

$$\begin{aligned} \tilde{\boldsymbol{D}}^{-1} &:= \operatorname{diag}(1/\tilde{d}_i), \quad i = 1, \dots, m, \\ \text{where} \quad \tilde{d}_i &:= \max(d_i, d_{\min}). \end{aligned} \tag{22}$$

It is worth noticing that the method (21), (22) of a robust matrix inversion is valid for non-redundant as well as for redundant manipulators. Moreover, it is not computationally demanding, as at regular configurations it is computed using (20), (21), while at singular configurations the modification (22) is minor. In the next section, it will be explained why all very small singular values should be replaced with the same value $d_{\min}$.

3.4. Singularity-Robust Inverse Kinematics — A Geometrical Interpretation

The main idea behind almost all singularity-robust inverse kinematics algorithms is the same: to modify either a Jacobi matrix $\boldsymbol{J}$ or a manipulability matrix $\boldsymbol{M}$ to retrieve invertability of the manipulability matrix. However, the modification may disturb a motion towards a current sub-goal $\boldsymbol{x}_f$ initialized at a current point in a task-space $\boldsymbol{x}_c = \boldsymbol{k}(\boldsymbol{q}_c)$ especially when the modification is not small. In Figure 1, possible motion scenarios are depicted. At regular configurations $\boldsymbol{q}_c$, the motion generated with the Newton algorithm (3) (at least infinitesimally) shifts the point

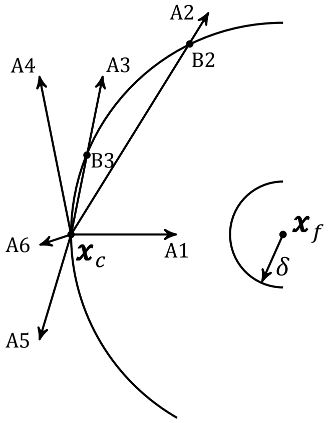

Figure 1. Possible motions in a task-space resulting from modifying either the Jacobi or the manipulability matrix

$\boldsymbol{x}_c$ directly towards the goal (A1) and there are no problems with the convergence of the algorithm even when the Jacobi matrix is not frequently updated as $\boldsymbol{q}_c$ is iterated. When modifications are relatively big (as it happens at or around singular configurations), possible directions of motion can be quite different (A2-A6). Some of them (A3-A6), even infinitesimally, do not guarantee the decrease of a tracking error (so the error is inevitable). Some others (A2-A3) allow the movement of the current point $\boldsymbol{x}_c$ towards the goal but with a carefully selected (optimized) value of the positive coefficient ξ_i, cf. Equation (3), which is not to move further than points (B2-B3). In this case, the number of iterations to obtain a vicinity of the goal point $\boldsymbol{x}_f$ with an assumed accuracy δ increases. Obviously, when δ is small and the absolute value of an angle between a current direction of motion and the vector $\boldsymbol{x}_c\boldsymbol{x}_f$ is close to 90°, but does not exceed it, the number of iterations to converge increases even more.

4. Evaluation of Robust Matrix Inversion Methods

In order to compare methods of coping with singularities via robust matrix inversions, some criteria should be proposed. The most important four factors are listed below:

1) How big is a matrix modification?
2) How far is it allowed to move away from singularities?
3) What is the computational complexity of the method?
4) Can the aforementioned characteristics be computed analytically or numerically only?

QN. 1) The answer for the first question seems to be obvious: a matrix distance between the original manipulability matrix and the modified one should be calculated using a selected matrix norm. However, the first method is based on modifying the manipulability matrix (5) by adding to it either the matrix (11) or (13), while the second method (22) modifies singular values of the Jacobi matrix. The methods based on modification (11) and (22) are comparable because decomposition (20) allows the expression of the manipulability matrix (5) in terms of singular values of $\boldsymbol{J}$ as follows:

$$\boldsymbol{M} = \boldsymbol{U}\,[\boldsymbol{D}, \boldsymbol{0}_{m,r}]\,\boldsymbol{V}^T \boldsymbol{V} \begin{bmatrix} \boldsymbol{D}^T \\ \boldsymbol{0}_{r,m} \end{bmatrix} \boldsymbol{U} = \boldsymbol{U}\boldsymbol{D}^2\boldsymbol{U}^T. \tag{23}$$

Consequently, the modification (11) can be expressed in terms of singular values

$$\boldsymbol{M} + \boldsymbol{A} = \boldsymbol{M} + \lambda \boldsymbol{I} = \boldsymbol{U}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})\boldsymbol{U}^T \tag{24}$$

and the mid-term matrix $\boldsymbol{D}^2 = \operatorname{diag}(d_1^2, \dots, d_m^2)$ is increased by the value of

$$f_1 = m \cdot \lambda. \tag{25}$$

When the modification method based on (22) is used, then the matrix $\boldsymbol{D}^2$ is replaced with $\mathrm{diag}(d_1^2, \ldots, d_{m-c}^2, \underbrace{d_{\min}^2 \ldots, d_{\min}^2}_{c \text{ times}})$, thus the increase is equal to

$$f_2 = \sum_{i=m-c+1}^{m} (d_{\min}^2 - d_i^2) = c \cdot d_{\min}^2 - \sum_{i=m-c+1}^{m} d_i^2, \quad (26)$$

where the corank c was defined in (6) and counts items satisfying the condition $d_i < d_{\min}$. The functions f_1, f_2, respectively, measure a distance between the original matrix $\boldsymbol{M}$ and its modified version, performed either directly (24) or via increasing minimal singular values (22).

QN. 2) Usually, a "distance" to singular configurations is estimated by the value of the manipulability matrix determinant (or its square root — the manipulability index)

$$\det(\boldsymbol{M}(\boldsymbol{q})) = \prod_{i=1}^{m} d_i^2(\boldsymbol{q}). \quad (27)$$

The function (27) is non-negative, has got an analytical gradient with respect to $\boldsymbol{q}$, and attains the zero value at singular configurations. Therefore, it is useful in avoiding singularities by the optimization within a null space of the Jacobi matrix for redundant manipulators [11]. Unfortunately, it does not show how far a current configuration is from a singular one. For this purpose, a better choice is to take the minimal value among singular values, or its value squared, i.e.,

$$g_1(\boldsymbol{q}) := d_m(\boldsymbol{q}), \quad g_2(\boldsymbol{q}) := d_m^2(\boldsymbol{q}) \quad (28)$$

as this value decides the well or badly conditioning aspect of the manipulability matrix. Similar to (28), also in other robotic tasks, a square of a positive value is used instead of the value itself just to avoid the time-consuming square-root operation, like in a task of calculating the distance to obstacles.

The functions (28) have also one drawback as they can be calculated only numerically at a given configuration. Apparently, the minimal singular value of the modified manipulability matrix should be as big as possible. This observation justifies selection (22) to increase all the smallest singular values to the same level. Using the function g_2, the modifications (11), (24), or (22) increase the minimal singular value to

$$h_1(\boldsymbol{q}) := d_m^2(\boldsymbol{q}) + \lambda, \quad h_2 := d_{\min}^2, \quad (29)$$

respectively. Now we are in a position to compare the two modifications assuming that the manipulability matrix increase is the same, $f_1 = f_2$. The better modification should give a greater value of the resulting minimal singular value squared. It will be proven that $h_1 < h_2$, which means that the modification (22) is better than (11). At first, from $f_1 = f_2$ (cf. (25), (26)), λ is calculated and substituted into (29). Then,

$$h_1 < h_2 \Leftrightarrow c \cdot d_{\min}^2 - \sum_{i=m-c+1}^{m} d_i^2 < m\,(d_{\min}^2 - d_m^2). \quad (30)$$

Finally,

$$c \cdot d_{\min}^2 - \sum_{i=m-c+1}^{m} d_i^2 \le c \cdot d_{\min}^2 - c \cdot d_m^2 < m\,(d_{\min}^2 - d_m^2), \quad (31)$$

as $c < m$ and $d_i \le d_{\min}$ for $i = m-c+1, \ldots, m$. From (31), it can be deduced that the difference between h_2 and h_1 increases as c is small (in a typical case $c = 1$) and m is big (usually $m \in \{3, \ldots, 6\}$).

QN. 3) It is commonly agreed that kinematic-like transformations are much simpler than those related to dynamics. Nevertheless, the robust inverse kinematics method based on the SVD decomposition is computationally simpler than the other methods as it couples the detection of singularities with passing through them. In this case, computations performed at the detection stage are effectively used in calculating the robust inversion (21), (22). Moreover, the method works in the same way for redundant as well as for non-redundant manipulators (the only difference is that the sub-matrix $\mathbf{0}_{m,r}$ in (20) disappears in the non-redundant case). The other robust methods do not show this feature and both cases should be considered and implemented separately.

QN. 4) All presented methods of robust inverse kinematics are numerical in nature. For some simple manipulators singular configurations can be calculated explicitly, for more complicated ones only analytical conditions can be formulated to be satisfied at singular configurations. For the PUMA 560 robot [8] shoulder singularities are given by an expression relating components of a configuration vector while elbow and wrist singularities can be calculated explicitly.

5. Simulation Results

Simulations were performed on the non-redundant manipulator, Figure 2, equipped with three rotational joints and described by the forward kinematics

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_1 A \\ s_1 A \\ a_0 + a_2 s_2 + a_3 s_{23} \end{bmatrix}, \quad (32)$$

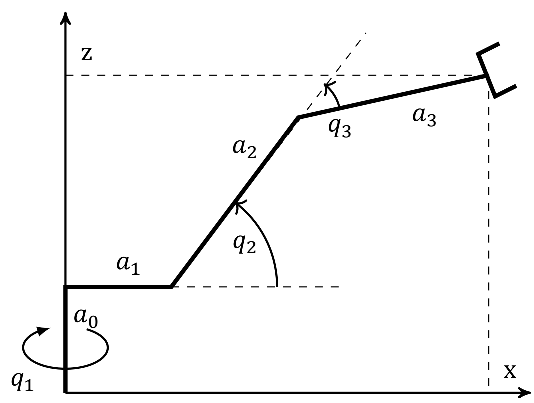

Figure 2. The three degrees of freedom manipulator

where

$$A = a_1 + a_2 c_2 + a_3 c_{23} \tag{33}$$

and a_0, a_1, a_3, a_3 are length parameters. In (32), (33) the standard robotic convention is used to denote trigonometric functions and $s_{23} = \sin(q_2 + q_3)$, $c_2 = \cos(q_2)$, etc. Singular configurations are derived from the Jacobi matrix

$$\boldsymbol{J}(\boldsymbol{q})) = \begin{bmatrix} -s_1 A & -c_1(a_2 s_2 + a_3 s_{23}) & -a_3 c_1 s_{23} \\ c_1 A & -s_1(a_2 s_2 + a_3 s_{23}) & -a_3 s_1 s_{23} \\ 0 & a_2 c_2 + a_3 c_{23} & a_3 c_{23} \end{bmatrix}, \tag{34}$$

and they satisfy the condition

$$\det(\boldsymbol{J}(\boldsymbol{q})) = -a_2\, a_3\, A \cdot s_3 = 0. \tag{35}$$

It appears that singular configurations arise for $q_3 = 0$ at the boundary of the task space but also along the z-axis when $A = 0$ if only the condition

$$a_2 + a_3 \geq a_1 \tag{36}$$

is met.

The two following singularity-robust algorithms of motion planning (3) are compared to:

Algorithm 1: To the manipulability matrix $\boldsymbol{M}$ in (4), the term given in Equation (24) is added,

Algorithm 2: SVD-based method (22) with equal minimal singular values.

In order to make the algorithms comparable, the total increase of the manipulability matrix was fixed and the λ coefficient in Algorithm 1 was calculated from the condition $f_1 = f_2$, cf. (25), (26) based on $d_{\min}$ parameter given.

The other data for a simulation are the following:

- accuracy of reaching sub-goals along a given path $\delta = 10^{-9}$,
- unit-less length parameters: $a_0 = 0, a_1 = 0.3, a_2 = 0.2, a_3 = 0.3$,
- detection of singularities $d_m < 10^{-3}$,
- the minimal singular value after modification $d_{\min} = 10^{-2}$.

Finally, the path to follow

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ (0.6 + 0.2\,t)\sqrt{(a_2 + a_3)^2 - a_1^2} \end{bmatrix} \tag{37}$$

is given on the (time) interval $t \in [0, 2]$ and all its points are realized with singular configurations as $A = 0$, so the path can be considered as hard to follow. The range of admissible z-coordinates for $x = y = 0$

$$\left[a_0 - \sqrt{(a_2 + a_3)^2 - a_1^2}, a_0 + \sqrt{(a_2 + a_3)^2 - a_1^2}\right].$$

Trajectories generated with both robust inverse algorithms were almost the same: $q_1 = 300°$ took almost the constant value (up to the numerical noise) while the remaining components of the configuration vector are depicted in Figure 3. The only difference was observed at $t = 2$ where the SVD-based Algorithm 2 obtained the final path point with the prescribed accuracy while Algorithm 1 cannot do that and stopped at the distance $5.6 \cdot 10^{-6}$ from the target. All other points of the path were obtained with the prescribed accuracy δ, thus both robust inverse kinematic algorithms worked quite well. The point for $t = 2$ is located at the boundary of the task-space and it is obtained in the corank 2 configuration (not only $A = 0$ but also $s_3 = 0$, cf. Equation (35)) and only one column of matrix $\boldsymbol{J}$ in (34) is independent.

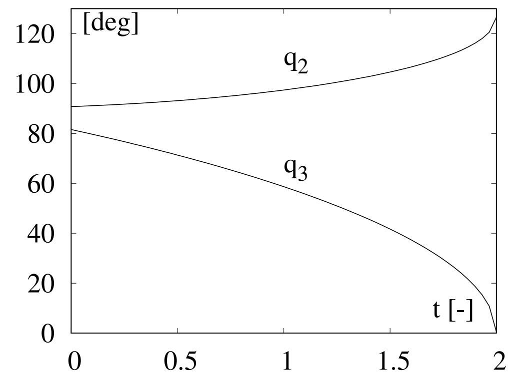

Figure 3. Coordinates q_2 and q_3 corresponding to the path followed

6. Auxiliary Issues

In this section, two mathematical problems are addressed and two practical tips are given. In Subsection 6.1 it is shown that a minimal singular value (the function g_1) used as a distance estimation of a current configuration (via singular values of its Jacobi matrix) from a singularity is not a norm in a mathematical sense. In Subsection 6.2, it is pointed out that for non-redundant manipulators reliable singularity-robust inverse kinematics cannot be performed directly on the Jacobi matrix (but still it can be done using its manipulability matrix or operations on its singular values). The next two subsections provide practically-oriented hints. In Subsection 6.3, advantages of a normalization of forward kinematics are discussed while in Subsection 6.4 it is shown how to minimize disadvantages of existence of inadmissible motion directions within a task-space.

6.1. Is the Minimal Singular Value a Norm for the Singularity Detection?

The function g_1 defined in (28) and equal to the minimal singular value defines a "distance" of a Jacobi matrix to singularity. However, it is not a distance in a strict mathematical sense as it does not satisfy the triangle inequality. In order to show this, let us take two exemplary matrices

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{A} + \boldsymbol{B} = \boldsymbol{I}_2. \tag{38}$$

Apparently, $g_1(\boldsymbol{A}) = g_1(\boldsymbol{B}) = 0$ but $g_1(\boldsymbol{A} + \boldsymbol{B}) = 1$. It is known [7] that the maximal singular value can serve as a norm of a matrix (so called the spectral norm). However, this norm is not useful in coping with singularities as the smallest singular value of a matrix impacts singularities the most. To emphasize the importance of the minimal singular value in the vicinity of singular configuration let us assume that a small displacement $\Delta\boldsymbol{x}$ ($\|\Delta\boldsymbol{x}\| = \nu$) is to be executed there. Using (2) and rearranging (20), the following formula is obtained

$$\boldsymbol{U}^T\Delta\boldsymbol{x} = [\boldsymbol{D}, \boldsymbol{0}_{m,r}] \cdot \boldsymbol{V}^T\Delta\boldsymbol{q}. \quad (39)$$

In the worst possible case, when $\boldsymbol{U}^T\Delta\boldsymbol{x} = (0, \ldots, 0, \nu)^T$,

$$\nu = \|\boldsymbol{U}^T\Delta\boldsymbol{x}\| = d_m\|\boldsymbol{V}^T\Delta\boldsymbol{q}\| = d_m\|\Delta\boldsymbol{q}\|. \quad (40)$$

Equation (40) describes a well known fact that around a singular configuration velocities at joints can attain very large (infinite) values. But it also prompts how to upper-bound the norm on the velocities by replacing d_m by a design parameter $d_{\min}$

$$\|\Delta\boldsymbol{q}\| \leq \nu/d_{\min}. \quad (41)$$

6.2. Varying the Jacobi Matrix for Non-redundant Manipulators

For redundant manipulators, the classical modification of the manipulability matrix (24) is provable good for redundant manipulators. But the question arises: is it possible to modify the Jacobi matrix itself and avoid the computationally costly modification via the manipulability matrix (in this case non-redundant manipulator is considered as a special case of redundant one)? The answer is no. To prove this statement let us analyze a hypothetical 2-dof manipulator with its Jacobi matrix expressed in a general form

$$\boldsymbol{J} = \begin{bmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{bmatrix} \quad (42)$$

and considered at a singular configuration

$$\det(\boldsymbol{J}) = j_{11}j_{22} - j_{11}j_{21} = 0. \quad (43)$$

The modified Jacobi matrix $\boldsymbol{J} + \lambda\boldsymbol{I}_2$, taking into account (43), has got its determinant equal to

$$\det(\boldsymbol{J}+\lambda\boldsymbol{I}_2) = \lambda(j_{11}+j_{22})+\lambda^2 = (j_{11}+j_{22}+\lambda)\lambda \quad (44)$$

which, in general, may take the value of zero for $\lambda \neq 0$. The same reasoning repeated for higher dimensional configuration spaces reveals that the polynomial (44) will be of the n-th degree with its coefficients depending of items of $\boldsymbol{J}$ and non-zero value not-guaranteed.

6.3. Normalization of Kinematics

The Jacobi matrix (2), (20) transforms velocities from a configuration space into a task-space. In practice, coordinates of vectors from the configuration and the task spaces have got a physical meaning and units as well (angles, positions, and translations). Thus, in some applications there is a practical need to weight their contribution in the Newton algorithm (3).

In order to unify different (in units and ranges) configuration coordinates, a weighted pseudo-inverse matrix [11] can be used

$$\boldsymbol{J}^{\#}_{W} = \boldsymbol{W}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T)^{-1} \quad (45)$$

where $\boldsymbol{W}$ is a symmetric, positive definite weighting matrix. This (right) pseudo-inverse matrix results from maximizing $\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{q}}$ with the fixed value of $\dot{\boldsymbol{q}}^T\boldsymbol{W}\dot{\boldsymbol{q}}$.

While analyzing the decomposition (20), one can notice that the matrix $\boldsymbol{V}^T$ rotates velocities $\dot{\boldsymbol{q}}$, then the rotated velocity is scaled and truncated into m dimensional object by the matrix $[\boldsymbol{D}, \boldsymbol{0}_{m,r}]$ and finally this object is rotated in the m-dimensional space by the matrix $\boldsymbol{U}$ to get $\dot{\boldsymbol{x}}$. In the previous sections singular values were extensively used as a tool for detecting singularities and constructing robust inverses. Here a natural question can be posed about their units and ranges. Assuming that rotation matrices are unitless, then components of $[\boldsymbol{D}, \boldsymbol{0}_{m,r}]$ should have the same units as the Jacobi matrix $\boldsymbol{J}$. Clearly, singular values should depend on geometrical parameters of a manipulator. In order to make them less dependent on the parameters, it is advised to normalize components of forward kinematics (1) by dividing each component by its maximal allowable value. Unfortunately, as components of kinematics depend also on the configuration, the optimization could be a serious numerical task. Fortunately, it is quite simple to get a reasonable approximation of the value by summing up upper bounds of items contributing to the component of kinematics. For an exemplary expression

$$x = a_1 + a_2\cos q_2 + a_3\cos(q_2 + q_3), \quad (46)$$

the value is equal to $a_1 + a_2 + a_3$. The normalization of kinematics has got one more potential advantage. Frequently, to complete the Newton algorithm (3), the vector from a current location to the goal one should drop below a given threshold: $\|\boldsymbol{x}_f - \boldsymbol{k}(\boldsymbol{q}_i)\| < \delta$ with the Euclidean metric $\|\cdot\|$ used. When the normalized kinematics is used, the components of the vector are comparable.

6.4. Admissible and Inadmissible Directions of Motion at Singular Configurations

At singular configurations, a motion in a task-space along vectors that belong to the corank($\boldsymbol{J}$) dimensional space spanned by vectors perpendicular to independent columns $\boldsymbol{J}_{,\star,z}$, cf. (2), of the Jacobi matrix cannot be realized using any velocities at joints. Generally, the directions can be calculated with a numerical procedure only (using the Gramm-Schmidt ortogonalization algorithm [1]) but for the manipulator (32) it can be done analytically.

At singular configurations localized along the z-axis, the first column in (34) is a zero vector. It can be checked by direct calculations that a vector $(s_1, -c_1, 0)$ is perpendicular to the two remaining columns, so along this direction the infinitesimal motion cannot be performed.

Using the SVD-decomposition (20), it can be easily checked that the space of admissible motions in a task-space is spanned by $(m - c)$ first columns $(\boldsymbol{U}_{\star,1}, \dots, \boldsymbol{U}_{\star,m-c})$ of the matrix $\boldsymbol{U}$ while the remaining c columns span the space of inadmissible motions. Clearly, the higher corank c (6) is, the more restricted admissible motions are. As already mentioned in the introductory section, singularity-robust inverse kinematics will cause inaccuracy in tracking a desired path. So, let us assume that at a singular configuration a desired motion is described by the vector $\Delta\boldsymbol{x}$. The angle between $\Delta\boldsymbol{x}$ and its projection onto the subspace spanned by $(\boldsymbol{U}_{\star,1}, \dots, \boldsymbol{U}_{\star,m-c})$ may serve as a qualitatively good estimator of an expected tracking error. For a designer of a singularity-robust algorithm this observation can be useful while planning a path to be followed very precisely.

7. Conclusion

In this paper various singularity-robust methods of inverse kinematics were discussed and their reliable work has been evidenced even in the most demanding circumstances. The theoretical background of the methods given is aimed to help to understand, analyze, and evaluate any particular method. Some tips for robotics practitioners are provided on how to design singularity-robust methods and/or to properly set their parameters. In this paper, it has also been shown that the SVD-method with a fixed modification of a sum of singular values that sets all the smallest values to the same level is the best within this class of methods as it allows getting as far as possible from singularities and also reducing joint velocities. This method can be advised as the first choice method for computational reasons as computations performed at the singularity detection stage can be used in a regular as well as in a singular case. In this paper, some questions related to facilitate passing through singular configurations have been answered. They concern a transformation of kinematics before planning as well as preparing a path to be followed.

AUTHOR

Ignacy Dulęba – Department of Cybernetics and Robotics, Wroclaw University of Science and Technology, Janiszewski St. 11/17, 50-372 Wroclaw, Poland, e-mail: ignacy.duleba@pwr.edu.pl, www.kcir.pwr.edu.pl/~iwd.

References

[1] W. Cheney, and D. Kincaid, *Linear Algebra: Theory and Applications*, Jones & Bartlett Publ., 2009.

[2] I. Dulęba. "Robust inverse kinematics at singular configurations," A. Mazur and C. Zieliński, eds., *Advances in Robotics*, vol. 197 of *Electronics*, pp. 5–10. Publ. House of the Warsaw Univ. of Technology, 2022 (in Polish).

[3] I. Duleba. "A comparison of jacobian-based methods of inverse kinematics for serial robot manipulators," *Int. Journal of Applied Mathematics and Computer Science*, vol. 23, no. 2, 2013, pp. 373–382.

[4] I. Duleba. "Channel algorithm of transversal passing through singularities for non-redundant robot manipulators," *IEEE Int. Conf. on Robotics and Automation*, vol. 2, 2000, pp. 1302–1307; doi: 10.1109/ROBOT.2000.844778.

[5] G. Golub, and C. Reinsch. "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, 1970, pp. 403–420.

[6] B. Grossmann. "The product of two symmetric, positive semidefinite matrices has non-negative eigenvalues". Mathematics Stack Exchange; https://math.stackexchange.com/q/982822 (version: 2014-10-21).

[7] R. Horn, and C. Johnson, *Matrix analysis*, Cambridge Univ. Press, 2012.

[8] C.-G. Kang. "Online trajectory planning for a PUMA robot," *Int. Journal of Precision Enginnering and Manufacturing*, vol. 8, no. 4, 2007, pp. 16–21.

[9] S. Lloyd, R. A. Irani, and M. Ahmadi. "Fast and robust inverse kinematics of serial robots using Halley's method," *IEEE Transactions on Robotics*, vol. 38, no. 5, 2022, pp. 2768–2780; doi: 10.1109/TRO.2022.3162954.

[10] A. A. Maciejewski, and C. Klein. "The singular value decomposition: Computation and applications to robotics," *Int. Journal of Robotics Research*, vol. 8, 1989, pp. 63–79.

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

[12] A. Ratajczak, J. Ratajczak, and K. Tchoń. "Task-priority motion planning of underactuated systems: an endogenous configuration space approach," *Robotica*, vol. 28, no. 6, 2010, pp. 885–892.

[13] M. Spong, and M. Vidyasagar, *Robot Dynamics and Control*, MIT Press, 1989.

[14] J. Sun, Y. Liu, and C. Ji. "Improved singular robust inverse solutions of redundant serial manipulators," *Int. Journal of Advanced Robotic Systems*, vol. 17, no. 3, 2020, pp. 1–12; doi: 10.1177/1729881420932046.

[15] K. Tchoń, and J. Ratajczak. "Singularities of holonomic and non-holonomic robotic systems: a normal form approach," *Journal of the Franklin Institute*, vol. 358, no. 15, 2021, pp. 7698–7713.

[16] L. V. Vargas, A. C. Leite, and R. R. Costa. "Overcoming kinematic singularities with the filtered inverse approach", *IFAC Proceedings Volumes*, vol. 47, no. 3, 2014, pp. 8496–8502; doi: 10.3182/20140824-6-ZA-1003.01841, 19th IFAC World Congress.