2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Inverse Kinematics without matrix inversion

by Alexandre N. Pechev*

Abstract – This paper presents a new singularity robust and computationally efficient method for solving the inverse kinematics (IK) problem. In this method, the transformation from Cartesian space to joint space is performed in a feedback loop and as a result the new feedback inverse kinematics (FIK) law operates as a filter and does not require matrix manipulations (inversion, singular value decomposition or a computation of a damping factor). While the computational demand is greatly reduced, the performance is comparable to the one delivered by the damped least squares (DLS) law. The new algorithm is capable of escaping and avoiding kinematic singularities and in this respect it outperforms pseudo-inverse based formulations.

1 Introduction

Inverse kinematics is an essential element in any robotic control system and a considerable research has gone in the last decades in identifying a robust and generic solution to this problem. Inverse kinematics can be also linked to other areas, for example spacecraft control with control moment gyros (CMG), animation, protein folding. In attitude control loops of spacecrafts with CMGs, the Jacobian maps gimbal rates to components of torque [1]. In robotics, the Jacobian connects the velocity of the end-effector defined in Cartesian space with the joint velocities. Closed-form solutions to IK is limited to only a certain types of manipulators [2], and most of the proposed techniques resort to numerical methods. The *damped least squares* (DLS) inverse law, developed by Nakamura and Hanafusa in [3] and by Wampler in [4], has been proposed as an efficient and singularity robust solution to the IK problem. This is a pseudo-inverse based method which in the vicinity of the singularity approximates the solution for the expense of some error in the trajectory of the end-effector. The algorithm depends on a damping variable that defines the trade off between solvability and exactness. This variable is computed at every sample and tends to zero when away from singularities to transform the DLS method (or also known as the singularity robust method) to a Moore-Penrose pseudo inverse. It has been presented independently by Wolovich at. al. in [5] and Balestrino and co-workers in [6] that the IK problem can be solved by computing the transpose of the Jacobian instead of its inverse. This method, although numerically efficient, fails to deliver solutions for rank deficient Jacobians. Other methods for solving the inverse kinematics problem resort to different forms of optimisation, for example [7] and [8].

Recent research on singularity avoidance for CMGs has led to the development of a new method for solving the inverse kinematics (IK) problem [9, 10]. The *feedback inverse kinematics* method (or FIK) presented in this paper uses feedback control in the minimisation of the difference between demanded and actual Cartesian velocities. Within the feedback loop, the required joint parameters are derived through the control sensitivity function. The algorithm operates as a filter and does not require matrix manipulations (inversion or singular value decomposition). Singularities are handled without the necessity of a damping factor and this makes it computationally more efficient than pseudo-inverse based methods. Despite the differences, the FIK law compares closely in structure to the DLS IK law and this is demonstrated later in the paper. The dynamic constraints in the manipulator and the trajectory are also systematically linked into the design of the inverse kinematics law and its parameters. It has been also demonstrated that for a special class of singularities, the FIK law outperforms the performance delivered by the DLS method.

2 Problem statement

For a robot manipulator, the end-effector position $\boldsymbol{x} \in \mathbb{R}^m$ is related to the vector of joint variables $\boldsymbol{q} \in \mathbb{R}^n$ through a nonlinear vector-valued function

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q}) \tag{1}$$

The control of the manipulator requires tracking a target trajectory ($\hat{\boldsymbol{x}}$) defined in Cartesian space by manipulating the joint variables $\boldsymbol{q}$. This requires solving the inverse of Eq.1 to get $\boldsymbol{q}$ for a given set $\hat{\boldsymbol{x}}$. From a practical point of view, a more convenient approach is to use the joint velocities $\dot{\boldsymbol{q}}$ and the end-effector Cartesian velocities $\dot{\boldsymbol{x}}$. This requires linearising Eq.1 and computing the Jacobian $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$, $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial \boldsymbol{q}}$. The resultant kinematic representation becomes

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2}$$

*. A. Pechev is with The Surrey Space Centre, University of Surrey, Guildford, GU2 7XH, UK, a.pechev@surrey.ac.uk;

Eq.2 can be used for both position control and velocity control. In the general case, $n \geq m$ and the manipulator is kinetically redundant when $n > m$. Solving Eq.2 for a given set of Cartesian desired variables ($\dot{\boldsymbol{x}}$) requires solving

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}} \tag{3}$$

where $\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ represents the inverse of the Jacobian. For the cases when $m > n$, pseudo-inverse methods can be used to derive $\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ [for better readability, $\boldsymbol{q}$ is dropped from $\boldsymbol{J}(\boldsymbol{q})$]

$$\boldsymbol{J}^{\dagger}(\boldsymbol{q}) = \boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}})^{-1} \tag{4}$$

Since the Jacobean depends nonlinearly on the joint angles, there exists combinations in $\boldsymbol{q}$ at which the Jacobian becomes ill-conditioned with rank $r = \text{rank}(\boldsymbol{J(q)}), r < m$. At these *singular* combination, the pseudo-inverse algorithm in Eq.4 fails to deliver a solution and leads to excessively large joint velocities near the singularities. The *damped least squares* inverse law approximates the solution around the singularities by allowing some error in the end-effector position

$$\boldsymbol{J}^{\dagger}(\boldsymbol{q}) = \boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}} + \lambda \boldsymbol{I})^{-1} \tag{5}$$

The solution depends on a damping factor λ which expresses the trade off between exactness of the solution ($\lambda \approx 0$) and feasibility of the solution ($\lambda \gg 0$). Since the solvability and the stability of Eq.5 depends on λ, there has been a considerable research effort undertaken in identifying different methods for the adaptation of λ [3, 8, 11]. In the subsequent part of the paper, a new method for solving Eq.3 is proposed.

3 Feedback Inverse Kinematics

For the kinematic representation in Eq.2 and a given desired Cartesian velocity $\dot{\boldsymbol{x}} \in \mathbb{R}^m$, the following vector of error variables $\boldsymbol{e} \in \mathbb{R}^m$ is constructed

$$\boldsymbol{e} = \dot{\boldsymbol{x}} - \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{6}$$

to represents the discrepancy between the current and the desired Cartesian velocities. Let $\boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})$ represent the transpose of the Jacobian, then the main result of this paper can be formally established.

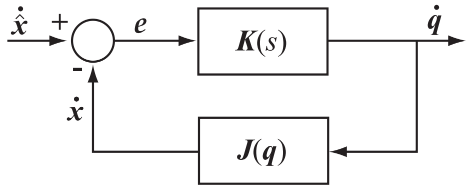

Figure 1. Feedback-based inverse kinematics

Theorem 1. *With the view to minimise the error in Eq.6, a negative feedback loop is constructed as in Fig.1. If $\boldsymbol{K(s)} \in \mathbb{R}^{n\times m}$ represent a full transfer-function matrix that acts as a control law, then there exist an adaptive form of $\boldsymbol{K(s)}$ that minimises this error to an arbitrary small value, leading to the following solution to the inverse kinematics problem*

$$\dot{\boldsymbol{q}} = \boldsymbol{K(s)}(\boldsymbol{J}\boldsymbol{K(s)} + \boldsymbol{I})^{-1}\dot{\boldsymbol{x}} \tag{7}$$

Furthermore the optimal $\boldsymbol{K(s)}$ has the following adaptive form

$$\boldsymbol{K(s)} = \boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})\mathcal{P}(s\mathbf{I} - \mathcal{A}_{\mathcal{K}})^{-1}\mathcal{B}_{\mathcal{K}} \tag{8}$$

where $\mathcal{P} \in \mathbb{R}^{m\times m}$, $\mathcal{P} > 0$ is a symmetric gain matrix, $\mathcal{A}_{\mathcal{K}} \in \mathbb{R}^{m\times m}$, $\mathcal{A}_{\mathcal{K}} < 0$ is a diagonal matrix and $\mathcal{B}_{\mathcal{K}} \in \mathbb{R}^{m\times m}$, $\mathcal{B}_{\mathcal{K}} > 0$ is a diagonal matrix. The specific numerical values of $\mathcal{P}, \mathcal{A}$ and $\mathcal{B}_{\mathcal{K}}$ are linked to the constraints in the manipulator and the desired trajectory.

Proof.

The construction of Eq.7 follows directly from the definition of the feedback loop in Fig.1; The joint rates are related to the desired Cartesian velocity through the control-sensitivity function in Eq.7. To derive the control law in Eq.8, a two step approach is proposed.

A) In the firsdt stage of design, the Jacobian is assumed constant, time invariant, transfer matrix that maps joint velocities to Cartesian velocities, i.e. $\boldsymbol{J(q)} = \boldsymbol{J_o} = \text{constant}$. *An optimal $\mathcal{H}_\infty$ control law is then designed that solves the following sensitivity-minimisation problem*

$$\min_{K(s)} \left\| \begin{pmatrix} \boldsymbol{w_1}(s)(\boldsymbol{J_o}\boldsymbol{K(s)} + \boldsymbol{I})^{-1} \\ \boldsymbol{w_2}(\boldsymbol{J_o}\boldsymbol{K(s)} + \boldsymbol{I})^{-1} \end{pmatrix} \right\|_{\infty} \tag{9}$$

$\boldsymbol{w_1}(s) \in \mathbb{R}^{m\times m}$ determines the bandwidth of the inverse kinematics law in Eq.7 and is derived from the dynamic constraints of the manipulator and the reference trajectory $\dot{\boldsymbol{x}}$. $\boldsymbol{w_2} \in \mathbb{R}^{n\times n}$ determines the constraints in the actuators in terms of the upper bound on the velocities $\dot{\boldsymbol{q}}$. Solving Eq.9 for $\boldsymbol{K(s)}$ then is a trivial task and requires using a standard $\mathcal{H}_\infty$ optimisation algorithm, for example [12].

B) For the adaptation of $\boldsymbol{K(s)}$, the controller derived above is first represented in its state-space form

$$\boldsymbol{K(s)}\colon \begin{array}{ll} \dot{z} & = \mathcal{A}_{\mathcal{K}}z + \mathcal{B}_{\mathcal{K}}\boldsymbol{e} \\ \dot{\boldsymbol{q}} & = \mathcal{C}_{\mathcal{K}}z \end{array} \tag{10}$$

where $z \in \mathbb{R}^{m\times 1}$ represents the state of the controller, $\mathcal{A}_{\mathcal{K}} \in \mathbb{R}^{m\times m}$, $\mathcal{A}_{\mathcal{K}} < 0$ is a diagonal matrix, $\mathcal{B}_{\mathcal{K}} \in \mathbb{R}^{m\times m}$, $\mathcal{B}_{\mathcal{K}}$ is a diagonal matrix and $\mathbf{e}$ is the error in Eq.6. From the the solution the the $\mathcal{H}_\infty$ problem in Eq.9, it can be shown that the optimal, in ∞-norm sense, state-feedback gain is

$$\mathcal{C}_{\mathcal{K}} = \boldsymbol{J_o^T}\mathcal{P} = \boldsymbol{J_o^T}w^2b\boldsymbol{P} \tag{11}$$

where $\boldsymbol{P} > 0$, $\boldsymbol{P} = \boldsymbol{P}^T$ is the solution to the Riccati equation associated with the state-feedback design of $\boldsymbol{K}(\boldsymbol{s})$ [12], b is the bandwidth of the loop in Fig.1, and w is the upper bound on the joint velocities $\dot{\boldsymbol{q}}$. Eq.11 suggests that $\boldsymbol{K}(\boldsymbol{s})$ can be adapted by taking current values for $\boldsymbol{q}$, computing $\boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})$ and adapting $\mathcal{C}_{\mathcal{K}}$ accordingly, i.e.

$$\mathcal{C}_{\mathcal{K}} = \boldsymbol{J}(\boldsymbol{q})^{\boldsymbol{T}}\mathcal{P} \tag{12}$$

Combining Eq.12 with Eq.10 provides the adaptive control law $\boldsymbol{K}(\boldsymbol{s})$ in Eq.8 that solves the inverse kinematics problem. In transfer-function form, $\boldsymbol{K}(\boldsymbol{s})$ is given in Eq.8. This completes the proof.

3.1 Remarks

Comparison with pseudo-inverse methods: Setting

$$Q = \mathcal{P}(s\mathbf{I} - \mathcal{A}_{\mathcal{K}})^{-1}\mathcal{B}_{\mathcal{K}} \tag{13}$$

allows rewriting the FIK law in Eq.7 to its more compact form [below $\boldsymbol{q}$ is dropped from $\boldsymbol{J}(\boldsymbol{q})$]

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q}(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q} + \boldsymbol{I})^{-1}\dot{\hat{\boldsymbol{x}}} \tag{14}$$

It is of interest to note that the solution in Eq.14 has a similar structure to the DLS law in Eq.5. It can be also compared to the Jacobian transpose method from [5] and [6] with the assumption that $\mathcal{Q}(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q} + \boldsymbol{I})^{-1}$ is simplified by a constant and diagonal matrix. Despite the similarities, important differences need to be acknowledged. Due to the use of the feedback loop in Fig.1, the joint velocities are related to the demanded Cartesian velocities through the control sensitivity function $\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q}(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q} + \boldsymbol{I})^{-1}$ and since the computation is done in a feedback loop, no matrix inversion is required. For the FIK law λ=1 and essentially a damping factor is not used while the new dynamic mapping $\mathcal{Q} \in \mathbb{R}^{m\times m}$ (Eq.13) is introduced. In comparison to the Jacobian transpose methods, the new FIK method provide singularity robustness due to the non-diagonal and dynamic form of $\mathcal{Q}(s)$.

Stability of the IK: From a control point of view, the internal stability for the loop in Fig.1 is guaranteed as long as

$$\det(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q} + \boldsymbol{I}) \neq 0 \tag{15}$$

or alternatively

$$\sigma(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q}) \geq 0 \tag{16}$$

for all combinations in $\boldsymbol{q}$. Since $\mathcal{Q} = \mathcal{Q}^T \geq 0$ by design, this requirement is always satisfied and does not depend on the parameters of the Jacobian $\boldsymbol{q}$. At singularity, $\underline{\sigma}(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q}) = 0$.

Singularity robustness: It is important to establish that for singularity robustness, $\boldsymbol{K}(\boldsymbol{s})$ is designed to possess a full structure. When $\boldsymbol{J}(\boldsymbol{q})$ becomes rank deficient, the error in Eq.6 grows at a particular singular direction. The *escaping* joint velocities are generated through the off-diagonal elements in $\boldsymbol{K}(\boldsymbol{s})$. This is done in an infinity-norm, error-minimisation, sense as per the design of $\boldsymbol{K}(\boldsymbol{s})$. This is quite opposite to what has been chosen in the minimisation for the DLS law in Eq.5 and hence the necessity for λ.

Design for manipulator constraints: If all m-directions in the end-effector are to be weighted equally, $\boldsymbol{w_1}(s)$ is chosen as a diagonal transfer-function matrix. In almost all type of applications, it would be sufficient to set

$$\boldsymbol{w_1}(s) = \frac{b}{s+\alpha}\boldsymbol{I}^{m\times m} \tag{17}$$

where b determines the bandwidth in Fig.1 and α specifies the gain of the sensitivity function at steady-state. With this form of $\boldsymbol{w_1}(s)$, $\mathcal{A}_K$ transforms to

$$\mathcal{A}_{\mathcal{K}} = -\alpha\boldsymbol{I}^{m\times m} \tag{18}$$

$\boldsymbol{w_1}(s)$ allows specifying the tracking accuracy and is linked to the dynamics of the desired trajectory. Contrary to classical pseudo-inverse based IK solutions, this can be specified for each m-direction independently by setting b and α differently for each direction. This can be exploited for prioritisation of operations. For the form in Eq.17, $\mathcal{B}_{\mathcal{K}}$ can be taken to be the identity matrix transforming $\mathcal{C}_{\mathcal{K}} = b\boldsymbol{I}^{m\times m}$. The constraints in the manipulator in terms of the upper bound in the joint velocities ($\|\dot{\boldsymbol{q}}\|_\infty$) can be specified in $\boldsymbol{w_2}$. This is done for each degree of freedom (n). For manipulators with similar characteristics in the actuators it would be sufficient to set

$$\boldsymbol{w_2} = \frac{1}{w}\boldsymbol{I}^{n\times n} \tag{19}$$

where w determines the maximum rate (in m/s or rad/s). Oppositely to the DLS-based methods, the FIK method allows weighting each degree of freedom (n) independently in $\boldsymbol{w_2}$.

Performance of the FIK solution: For the feedback loop in Fig.1, the error can be computed from the sensitivity function

$$\boldsymbol{e} = (\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\mathcal{Q} + \boldsymbol{I})^{-1}\dot{\hat{\boldsymbol{x}}} \tag{20}$$

The relationship in Eq.20 suggests that the error is proportional to the size of $\mathcal{Q}$ (Eq.13). Furthermore, considering only steady-state, it follows from Eq.11 that the error is proportional to the square of the maximum rate w of the joint actuator.

Implementation: As identified earlier, the FIK method operates as a filter and requires only multiply and accumulate instructions. For the implementation, Eq.10 is transformed to its discrete equivalent for a given sampling interval τ

$$\begin{aligned} z(k+1) &= \alpha_\tau z(k) + \mathcal{B}_\mathcal{K} \boldsymbol{e} \\ \dot{\boldsymbol{q}} &= \boldsymbol{J}(\boldsymbol{q})^{\boldsymbol{T}} \mathcal{P} z(k) \end{aligned} \tag{21}$$

where $\alpha_\tau = \exp(-\alpha\tau)$. The first equation determines the state of the FIK law and assumes the form of $\mathcal{A}_\mathcal{K}$ in Eq.18. The error $\boldsymbol{e}$ is computed from Eq.6. The second equation uses the Jacobian and the state to compute the joint velocities. If $\mathcal{B}_\mathcal{K}$ is different from the identity matrix, $\mathcal{P}$ can be multiplied by $\mathcal{B}_\mathcal{K}$ a priori to improve on the computational efficiency. Assuming a full form for the Jacobian matrix, the total number of operations is summarised in Table 1. For a 6DOF manipulator it takes only 210 multiplications/additions to compute the IK; Divisions are not required and the singularity robustness is embedded into the algorithm. The number of flops can be further reduced if a care is taken for avoiding multiplications by ones and zeros. In comparisons to the implementation of the DLS algorithm together with the adaptation of its damping factor, results presented in [3, 13, 14] for example, show that the FIK algorithm reduces significantly the computational demand (by a factor of at least five). For applications with multiple end effectors or for augmented Jacobians using joint limits and task-space models, several orders of reduction in the computational demand can be achieved. The FIK filter can also run in parallel with zero overhead for multiprocessing architectures.

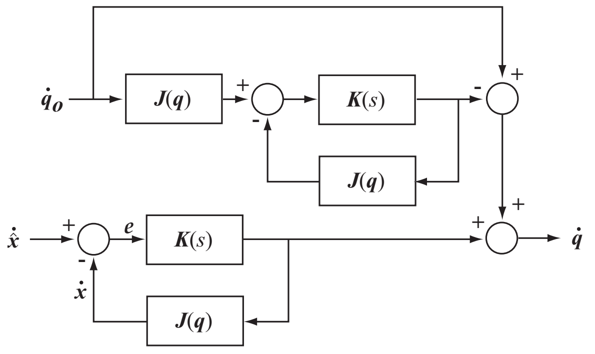

Figure 2. FIK law augmented with a homogeneous term.

Null-motion: For redundant manipulators it is possible to use the $(n-m)$-dimensional null space of $\boldsymbol{J}$ to generate joint velocity vectors $\dot{\boldsymbol{q}}$ that produce null end-effector velocities $\dot{\boldsymbol{x}}$. This is traditionally done by adding a homogeneous term to the pseudo-inverse, i.e.

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger \dot{\boldsymbol{x}} + (\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J}) \dot{\boldsymbol{q}}_{\boldsymbol{o}} \tag{22}$$

where $\dot{\boldsymbol{q}}_{\boldsymbol{o}}$ is an n-dimensional arbitrary joint velocity vector. Similarly, the feedback-based inverse kinematics solution in Fig.1 can be augmented by the projection operator $(\boldsymbol{I} - \boldsymbol{J}^\dagger \boldsymbol{J})$ using the filter form in Fig.2.

	additions	multiplications	total
$(m \times n)$	$2mn + m^2$	$2mn - n + m^2$	$4mn - n + 2m^2$
$(6 \times n)$	$12n{+}36$	11n+36	**23n+72**

Table 1. Computational demand for the FIK law.

4 Numerical examples

Without loosing generality, the presented in this paper new inverse kinematics method is applied to a simple 3-dof manipulator (Fig.3) as the one presented by Nakamura and Hanafusa in [3]. For the analysis the performance is compared with the DLS method from the same paper. We demonstrate in the results section that the new FIK law is not only computationally more efficient but it also outperforms the DLS algorithm in terms of handling singularities. The new inverse kinematic solution has been also applied to manipulators and articulated figures with multiple dof, but for the sake of space these results have not been represented.

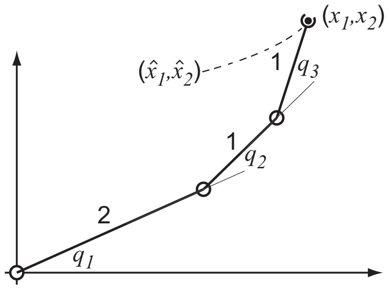

Figure 3. 3-degrees of freedom planar manipulator. $q_1 - q_3$ are the joint variables.

4.1 FIK algorithm design

For the manipulator in Fig.3, $m = 2$, $n = 3$ and the Jacobian is given below

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{pmatrix} -2\sin\bar{q}_1 - \sin\bar{q}_2 - \sin\bar{q}_3, & -\sin\bar{q}_2 - \sin\bar{q}_3, & -\sin\bar{q}_3 \\ 2\cos\bar{q}_1 + \cos\bar{q}_2 + \cos\bar{q}_3, & \cos\bar{q}_2 + \cos\bar{q}_3, & \cos\bar{q}_3 \end{pmatrix} \tag{23}$$

The joint variables are $\boldsymbol{q} = [\bar{q}_1,\ \bar{q}_1,\ \bar{q}_3]^T = [q_1,\ q_1 + q_2,\ q_1 + q_2 + q_3]^T$. For the design, we weight all m-directions and all n-actuators equally. This results in the following performance weights

$$\boldsymbol{w_1}(s) = \frac{400}{s+1}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{w_2} = \frac{1}{5}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{24}$$

$\boldsymbol{w_1}$ specifies a tracking bandwidth for the end-effector of approximately 400 rad/s with an attenuation rate of -52dB at DC. $\boldsymbol{w_2}$ describes the constraint in the manipulator by limiting the joint velocities to 5 rad/s. Setting $(q_1,\ q_2,\ q_3) = (1,\ 2,\ 1)$ rad, allows constructing

$\boldsymbol{J_o}$ from Eq.23

$$\boldsymbol{J_o} = \begin{pmatrix} -1.0673, & 0.6157, & 0.7568 \\ -0.5630, & -1.6436, & -0.6536 \end{pmatrix} \tag{25}$$

Solving the optimisation problem in Eq.9 with the above design inputs gives

$$\mathcal{P} = \begin{pmatrix} 295.28 & 46.96 \\ 46.96 & 225.03 \end{pmatrix} \tag{26}$$

$\mathcal{P}$ above and $\mathcal{A}_{\mathcal{K}} = -\alpha \boldsymbol{I}^{2\times 2} = -\boldsymbol{I}^{2\times 2}$, are used for the construction of $\boldsymbol{K(s)}$ in Eq.8

$$\boldsymbol{K(s)} = \boldsymbol{J^T(q)} \begin{pmatrix} 295.28 & 46.96 \\ 46.96 & 225.03 \end{pmatrix} \begin{pmatrix} \frac{1.66}{s+1} & 0 \\ 0 & \frac{1.66}{s+1} \end{pmatrix} \tag{27}$$

$\boldsymbol{K(s)}$ can be used in the feedback loop in Fig.1 to derive the joint velocities for a given set of required Cartesian velocities. In a filter form, the inverse kinematics transforms to the following set of equations (τ=1 ms, $\alpha_\tau = \exp(-\tau) \approx 1$)

$$\begin{aligned} z(k+1) &= z(k) + \dot{\hat{\boldsymbol{x}}} - \boldsymbol{J(q)}\dot{\boldsymbol{q}}(k) \\ \dot{\boldsymbol{q}}(k+1) &= 1.66\, \boldsymbol{J(q)^T}\mathcal{P} z(k) \end{aligned} \tag{28}$$

4.2 Simulation results

Two sets of simulation studies are considered:

A) a singularity free trajectory with an initial set of joint angles (q_1, q_2, q_3) = ($\frac{\pi}{4}$, $\frac{\pi}{9}$, 0) and a corresponding initial end-effector position $\boldsymbol{x_0} = (x_{10}, x_{20}) =$ (2.26, 3.23); the desired trajectory is set to $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2) = (x_{10} - \frac{1}{11}t, x_{20} - \frac{1}{8}t)$.

B) a trajectory that starts at a singularity and passes through a singularity with an initial set of joint angles $(q_1, q_2, q_3) = (\frac{\pi}{2}, 0, 0)$, corresponding end-effector position $\boldsymbol{x_0} = (x_{10}, x_{20}) = (0, 4.0)$ and a desired trajectory $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2) = (0, x_{20} - \frac{1}{8}t)$.

For the simulation work, the model of the manipulator, the control gains and simulation parameters are taken from the work of Nakamura and Hanafusa in [3]. The dynamics of the joints is represented as a second order system with a natural frequency of 10 rad/s and a damping factor of 0.5. A position feedback is used for the control of the robot. The adaptation law for the damping factor in the DLS algorithm is taken from the same paper

$$\lambda = \begin{cases} \lambda_0(1 - w/w_0)^2 & \text{for}\, w < w_0 \\ 0 & \text{for}\, w \geq w_0 \end{cases} \tag{29}$$

$w = \sqrt{\det[\boldsymbol{J(q)J^T(q)}]}$ is computed at every sample using current $\boldsymbol{q}$, $w_0 = 1.0$ and $\lambda_0 = 0.3$. We stress here that the particular choice for the design parameters have been optimised for both case studies (A) and (B). Other forms for adapting λ in the DLS law based on singular value decomposition have been also implemented but since no a greater improvement on the performance has been achieved, only results based on Eq.29 are included. For case-study (A), the Cartesian position of the end effector is shown in Fig.4(a); the individual trajectories (x_1, x_2) are given in Fig.4(c) for the DLS algorithm and in Fig.4(d) for the FIK algorithm. As evident from the condition number $c = \sigma_{\min}(\boldsymbol{JJ^T})/\sigma_{\max}(\boldsymbol{JJ^T})$ in Fig.4(b), at $t = 26$ sec, the trajectory passes close to the origin and both algorithms manage to avoid the singularity. The FIK law manages to track better the desired trajectory with lower values values for the error in the end-effector. For the DLS law, the performance is determined by the particular choice for λ_0. Both algorithms keep $\dot{\boldsymbol{q}}$ below 0.2 rad/s; Without modifications in the FIK and the DLS algorithms, a case study (B) is now considered to assess the singularity avoidance properties of the algorithms. It is evident from the condition number plot in Fig.6(b) that the DLS algorithm is not capable to escape the initial singular configuration for more than 10 seconds. Thus although the desired trajectory is changing along the required law, the end-effector is locked in its initial configuration. This is evident from the plots in Fig.6(c) and from the error plot in Fig.7(c). We stress here that there was not possible to select a suitable value for λ_0 to improve on this response. Furthermore, the same results were reproduced by other algorithms for adapting λ based on SVD. Contrary to this, the FIK law developed in this paper manages successfully to escape from the singularity (Fig.6(b) dotted line) and to track closely the desired trajectory (Fig.6(a)). The errors are also kept to small values throughout the whole trajectory (Fig.7). The joint rates (Fig.8) are also within the design requirements of 5 rad/s.

5 Conclusions

A novel method for solving the inverse kinematics problem is presented. The solution is approached from a control prospective and the resultant feedback-based inverse kinematics (FIK) law works as a filter, does not require matrix manipulations and a computation of a damping factor. While the performance is closely comparable with well established pseudo-inverse based algorithms such as the damped least squares (DLS), the proposed algorithm outperforms the DLS algorithm in terms of handling singularities. Furthermore, since the computation does not resort to matrix manipulations, the FIK law is a computationally more efficient than any pseudo-inverse based laws. In this paper, simulation results are included from a planar manipulator but the new algorithm has been successfully applied to redundant and non-redundant configurations with a high number of degrees of freedom and multiple end-effectors.

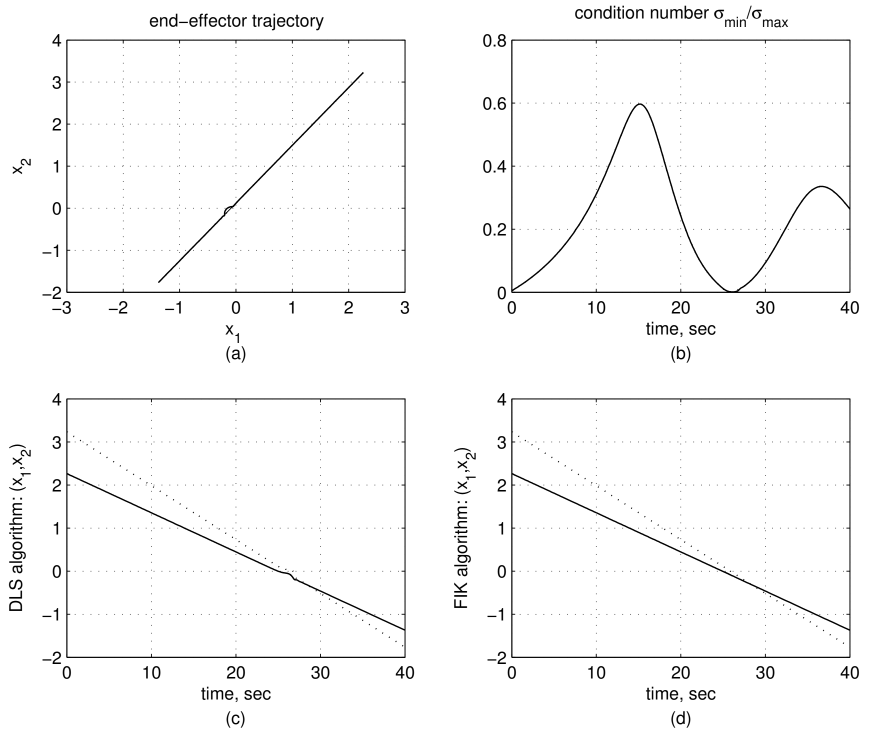

Figure 4. Responses for study (A): (a) (x_1, x_2) path; (b) condition number for DLS and FIK algorithms; (c) time responses of x_1 and x_2 using DLS; (d) time responses of x_1 and x_2 using FIK; dashed lines correspond to the DLS algorithm.

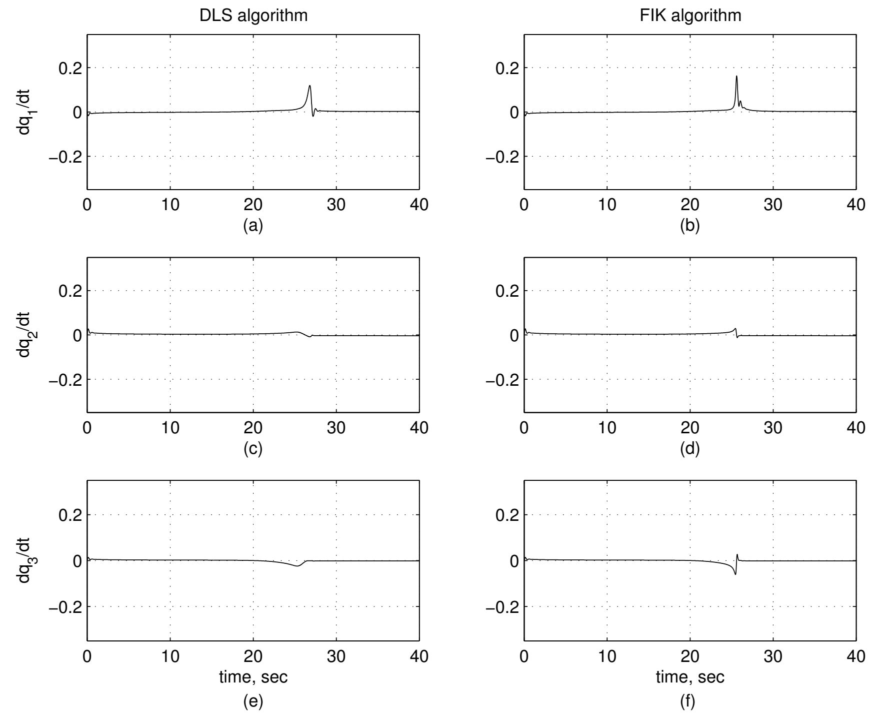

Figure 5. Responses for study (A): (a), (c) and (e) $\dot{\boldsymbol{q}}$ for DLS algorithm; (b), (d) and (f) $\dot{\boldsymbol{q}}$ for FIK algorithm.

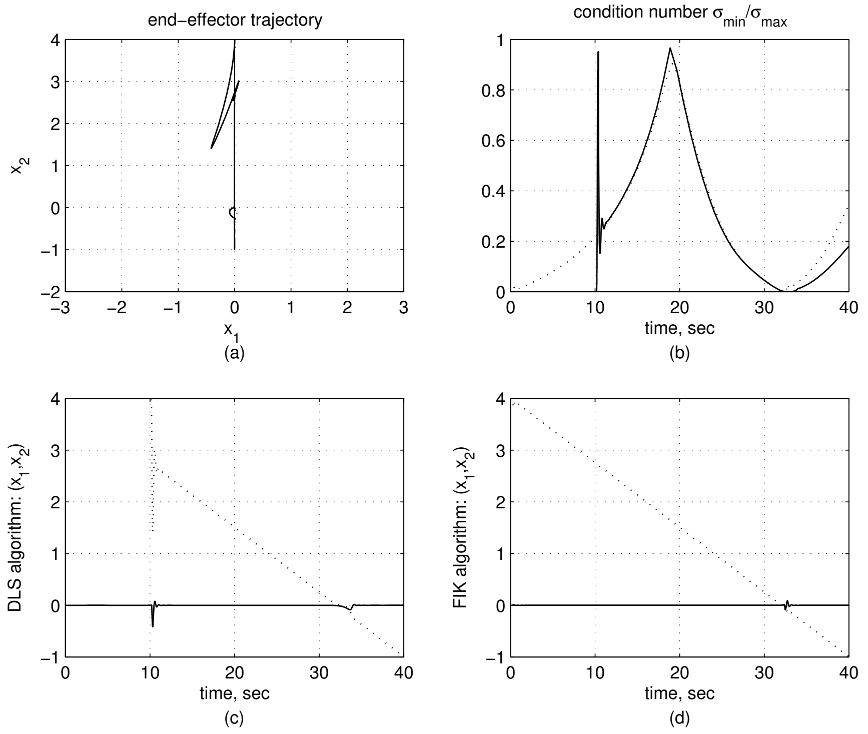

Figure 6. Responses for study (B): (a) (x_1, x_2) path; (b) condition number for DLS and FIK algorithms; (c) time responses of x_1 and x_2 using DLS; (d) time responses of x_1 and x_2 using FIK; dashed lines correspond to the DLS algorithm.

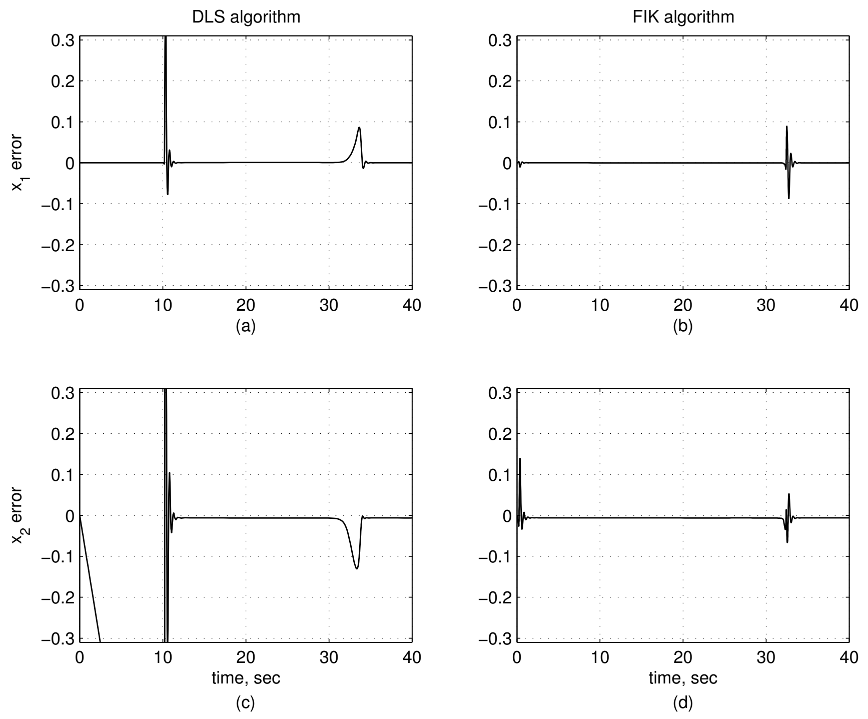

Figure 7. Responses for study (B): (a) and (c) position error for DLS algorithm; (b) and (d) position error for FIK algorithm.

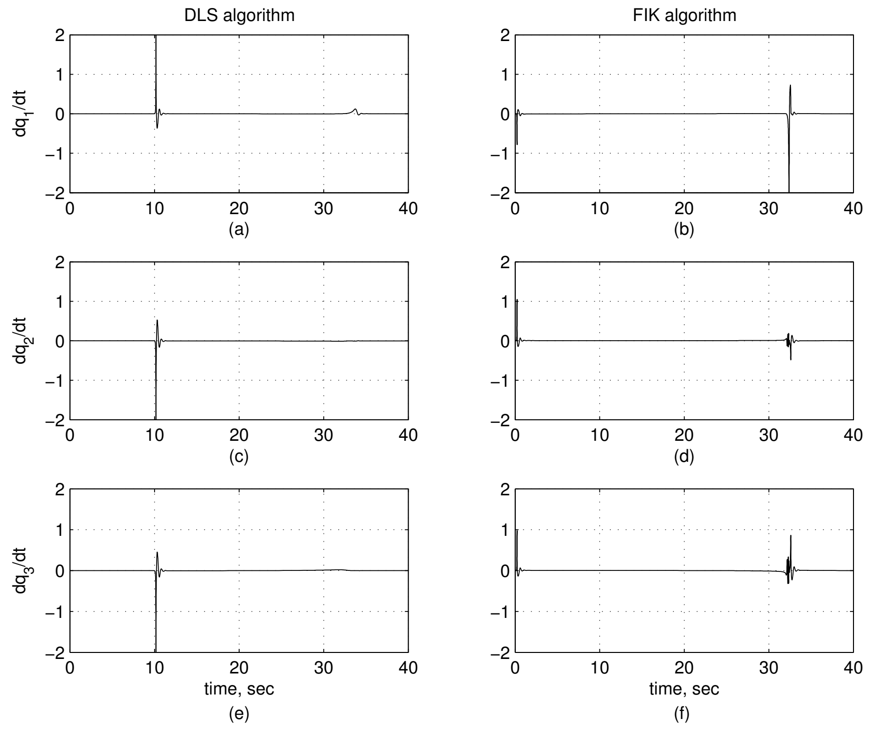

Figure 8. Responses for study (B): (a), (c) and (e) $\dot{\boldsymbol{q}}$ for DLS algorithm; (b), (d) and (f) $\dot{\boldsymbol{q}}$ for FIK algorithm.

Bibliography

[1] N. S. Bedrossian, J. Paradiso, E. V. Bergmann, and D. Rowell. Steering law design for redundant single-gimbal control moment gyroscopes. *Journal of Guidance Control Dynamics*, 13:1083–1089, 1990.

[2] D. L. Pieper. The kinematics of manipulators under computer control. *PhD Thesis, Stanford University*, 1968.

[3] Y. Nakamura and H. Hanafusa. Inverse kinematics solutions with singularity robustness for robot manipulator control. *Trans. ASME J. of Dynamic System, Measures and Control*, 108:163–171, 1986.

[4] C. W. Wampler. Manipulator inverse kinematic solution based on vector formulations and damped least squares methods. *IEEE Trans. on System Man and Cybernetics*, 16(1):93–101, 1986.

[5] W. A. Wolovich and H. Elliott. A computational technique for inverse kinematics. *The 23rd IEEE Conference on Decision and Control*, 23:1359–1363, 1984.

[6] A. Balestrino, G. De Maria, and L. Sciavicco. Robust control of robotic manipulators. *9th IFAC World Congress, Budapest*, 1984.

[7] L-C. T. Wang and C. C. Chen. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Transaction on Robotics and Automation*, 7(4):489–499, 1991.

[8] A. S. Deo and I. D. Walker. Adaptive non-linear least squares for inverse kinematics. *IEEE International Conference on Robotics and Automation*, pages 186–193, 1993.

[9] A. N. Pechev. Feedback-based steering law for control moment gyros. *Journal of Guidance Control Dynamics*, 30(3):848–855, 2007.

[10] A. N. Pechev. Computationally efficient and singuylarity robust method for solving the inverse kinematics problem. *Patent pending*, 2007.

[11] J. R. Sagli I. Spangelo and O. Egeland. Bounds on the largest singular value of the manipulator jacobian. *IEEE Transaction on Robotics and Automation*, 9(1):93–96, 1993.

[12] J.C. Doyle, K. Glover, P. Khargonekar, and B. Francis. State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Transactions on Automatic Control*, 34:831–847, 1989.

[13] S. Chiaverini, B. Siciliano, and O. Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on control systems technology*, 2(2):123–134, August 1994.

[14] A. A. Maciejewski and J. M. Reagin. A parallel algorithm and architecture for the control of kinematically redundant manipulators. *IEEE Transaction on Robotics and Automation*, 10(4):405–414, August 1994.