Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Overcoming Kinematic Singularities with the Filtered Inverse Approach

Lucas V. Vargas * , **Antonio C. Leite** * , **Ramon R. Costa** *

* *Department of Electrical Engineering*
COPPE, Federal University of Rio de Janeiro
Rio de Janeiro, RJ 21941-972 Brazil
(e-mail: lucasvar@coep.ufrj.br, toni@coep.ufrj.br, ramon@coep.ufrj.br)

Abstract: In this work, we propose a solution to the inverse kinematics problem based on the differential kinematics and on a recently proposed algorithm which estimates the inverse of the Jacobian matrix dynamically. The output of the algorithm can be interpreted as a filtered inverse (FI) of the Jacobian matrix. An interesting property of the FI algorithm is its ability to cope with kinematic singularities. The update law of the estimator is driven by error signals that consider both the left and the right inverse matrices, thus enabling trajectory tracking and minimization of the control effort simultaneously. This paper shows that the FI algorithm can be applied to a Jacobian matrix augmented with additional constraints, which allows for setting the priority or weight to different control objectives, such as obstacle avoidance. Simulation results are presented to illustrate the performance and feasibility of the proposed solution.

Keywords: Inverse kinematic problem, Matrix inversion, Robot kinematics, Singularities.

1. INTRODUCTION

In the past years, considerable research efforts have been directed towards the development of a generic and robust solution to the inverse kinematics (IK) problem, commonly found in several applications, such as 3D computer graphics animation (Grochow et al., 2004), protein backbone structure determination (Kolodny et al., 2005) and control of robotic manipulators (Chiaverini et al., 2008). The main difficulty in dealing with the IK problem is the presence of *kinematic singularities*, that is, those configurations at which the Jacobian matrix of the system is rank-deficient. At a singularity, the mobility of the structure is reduced and infinite solutions to the IK problem may exist (Siciliano et al., 2009). Most of the techniques discussed in the literature to tackle the IK problem are based on numerical methods or optimization (Maciejewski and Klein, 1989). The performance of these solutions is evaluated according to the stability, computational cost and robustness to singularities (Wampler, 1986; Tchon and Muszynski, 1998; Nenchev et al., 2000). The methods using the pseudo-inverse Jacobian may fail to find an acceptable solution in the neighborhood of a singularity since they can generate large steady-state error. In addition, the methods based on the transposed Jacobian, although numerically efficient in the presence of singularities, do not guarantee asymptotic stability of the tracking error (Siciliano et al., 2009).

Other methods for solving the IK problem use optimization techniques, such as the *cyclic coordinate descent* (CCD) method, which is based on nonlinear programming (Wang and Chen, 1991). The well-known *damped least square* (DLS) algorithm finds the inverse kinematics based on the pseudo-inverse Jacobian and uses a damping factor to avoid singularities, setting a trade-off between accuracy and feasibility of the solution (Nakamura and Hanafusa, 1986). The *feedback inverse kinematics* (FIK) method does not require a damping factor and employs a feedback loop to minimize the difference between the actual and desired velocities in the operational space (Pechev, 2008).

In this paper, we consider the method recently described in Vargas et al. (2013) which is based on an algorithm that dynamically estimates the inverse of the Jacobian matrix. In the particular case, where the matrix to be inverted is constant, the output from the algorithm can be viewed as its filtered inverse (FI) and, for this reason, it was called *FI algorithm*. The FI algorithm has some notable features and one is its ability to cope with kinematic singularities. It is shown that the only situation to be avoided is a Jacobian matrix converging to a singularity slower than any exponential, which is not a restrictive condition for application purposes. Another feature is related to the update law of the estimator, which is driven by error signals that consider both the left and the right inverse matrices, allowing for trajectory tracking and minimization of the control effort simultaneously.

This paper is a follow-up of our recent work (Vargas et al., 2013) and here we show that the FI algorithm can also be applied to a Jacobian matrix augmented with additional constraints, thus enabling the weighting of different control objectives by adjusting the design gains. Simulation results comprising a case study for the trajectory tracking problem of robot manipulators in the neighborhood of singularities are shown to illustrate the performance and feasibility of the filtered inverse approach.

* This work was supported in part by the Brazilian Funding Agencies: CAPES, CNPq and FAPERJ.

2. FILTERED INVERSE APPROACH

Consider a first-order SISO system described by

$$\dot{y} = k(t)\, u\,, \tag{1}$$

where $u \in \mathbb{R}$ is the scalar input of the plant (assumed to be bounded) or the control variable of the system, $y \in \mathbb{R}$ is the plant output and $k(t)$ is a scalar function. Considering that the control goal is to track a desired reference trajectory $r(t)$, a control law $u(t)$ which linearizes the system (1) and guarantees the asymptotic stability of the tracking error $e := r - y$ is given by

$$u(t) = k^{-1}(t)\,[\,\dot{r} + \lambda\, e\,]\,, \quad \lambda > 0\,. \tag{2}$$

Now, consider that the control law $u(t)$ uses a scalar function $\theta(t)$ dynamically updated, instead of the inverse computed instantaneously as $1/k(t)$, such that $k\theta \to 1$. To derive a suitable dynamics for $\theta(t)$, we introduce the error signal

$$S = k\theta - 1\,, \tag{3}$$

and consider the positive function $2V(S) = S^2$ which has time-derivative given by $\dot{V}(S) = S\dot{S} = S\big[\,\dot{k}\theta + k\dot{\theta}\,\big]$. In view of the last equation, we select the update law

$$\dot{\theta} = -\beta S k\,, \tag{4}$$

where $\beta > 0$ is the update gain. As a result, we get

$$\dot{V}(S) = S\dot{k}\theta - \beta S^2 k^2\,. \tag{5}$$

Notice that the update law (4) assures that the second term in (5) is non-positive. The first term, however, remains with undefined sign and depends on the $\dot{k}$. Here, we begin analyzing the simplest case where $\dot{k} \equiv 0$ and, thus, $\dot{V}(S) \le 0$. The equation (4) can be rewritten as

$$\dot{\theta} = -\beta(k\theta - 1)k = -\beta k^2 \theta + \beta k\,. \tag{6}$$

Using the differential operator s, we can explicit θ as

$$\theta = \frac{\beta k}{s + \beta k^2} = \frac{1}{\tau s + 1}\,[1/k]\,, \tag{7}$$

where $\tau = 1/\beta k^2$. Thus, θ can be interpreted as the output of a linear filter where the input is the true inverse $1/k$, that is, θ converges exponentially to $1/k$. Observe that the smaller is k, the larger is τ and consequently the slower is the filter. The same is observed for the gain β: the smaller is the update gain, the larger is the time constant of the filter. Motivated by (7), we refer to the signal $\theta(t)$ as the *filtered inverse* of function $k(t)$.

2.1 Properties of the filtered inverse

An important property of the algorithm (4) is that for $k \equiv 0$ it gives $\dot{\theta} \equiv 0$ and, as a consequence, $\theta(t) \equiv \theta(0)$. Fig. 1 shows the $\theta \times k$ plane with the locus of $S = 0$ and some trajectories for different initial conditions $\theta(0)$. In view of this figure, we can state that for any constant k the output $\theta(t)$ is bounded.

Consider now the case of a time-varying function $k(t)$. It is clear from Fig. 1 that the only possibility of $\theta(t)$ to grow without bound is in the region delimited by $-1 < S < 0$ (see the hatched region in the first quadrant). The closer k is to zero, the larger may be θ. However, it can be easily verified that in this region $0 < |k| < 1$ and $0 < |\dot{\theta}| < \beta$. This means that θ cannot escape in finite time.

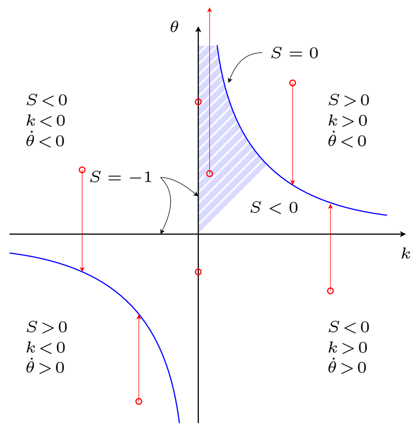

Fig. 1. Trajectories for some initial conditions in $\theta{\times}k$ plane.

Notice that for $\dot{k} \neq 0$, we have that

$$\frac{\partial\theta}{\partial k} = \frac{d\theta}{dt}\frac{dt}{dk} = -\frac{\beta k S}{\dot{k}}\,. \tag{8}$$

If k is a signal exponentially converging to zero with $\dot{k} = -\alpha\, k$, $\alpha > 0$, we can rewrite (8) as

$$\frac{\partial\theta}{\partial k} = \frac{\beta}{\alpha}\, S\,. \tag{9}$$

Therefore, in the hatched region, we have that $-\beta/\alpha < \partial\theta/\partial k < 0$ and consequently, the function $\theta(t)$ is bounded. This shows that, in fact, the filtered inverse $\theta(t)$ can eventually grow unbounded. However, the signal $k(t)$ should be converging to zero slower than any exponential. Consequently, the rate of increase of $\theta(t)$ should also be slower than any exponential. The main properties of the algorithm (4) can be summarized as follows:

(P1) $k \equiv 0 \;\Rightarrow\; \theta(t) \equiv \theta(0)$.
(P2) k constant $\;\Rightarrow\; \theta \in \mathcal{L}_\infty$.
(P3) $k \in \mathcal{L}_\infty \;\Rightarrow\; \dot{\theta} \in \mathcal{L}_\infty$.
(P4) $\theta(t)$ has no finite escape time.

2.2 Filtered inverse of matrices

The algorithm (4) can be easily generalized to deal with matrices. Consider the following first-order MIMO system described by

$$\dot{y} = K(t)\, u\,, \tag{10}$$

where $u \in \mathbb{R}^n$ is the plant input or the control variable of the system, $y \in \mathbb{R}^m$ is the plant output and $K \in \mathbb{R}^{m\times n}$ denotes a nonlinear matrix function. Notice that, in the context of differential kinematics, the matrix K is known as the *Jacobian matrix*.

Here, we consider that the control goal is to track a desired reference trajectory $r(t)$. A control law $u(t)$ which linearizes the system and guarantees the asymptotic stability of the tracking error $e := r - y$ is given by:

$$u(t) = K^{\dagger}(t)\,[\,\dot{r} + \Lambda\, e\,]\,, \quad \Lambda = \Lambda^{\mathsf{T}} > 0\,, \tag{11}$$

where $K^{\dagger}(t) \in \mathbb{R}^{n\times m}$ is the pseudo-inverse of the matrix $K(t)$. Again, consider that the control law $u(t)$ uses a dynamically updated matrix $\Theta(t)$ rather than the pseudo-inverse matrix $K^{\dagger}(t)$. To establish a suitable dynamics for $\Theta(t)$, we introduce an error signal $S_r \in \mathbb{R}^{m\times m}$ based on the right inverse

$$S_r = K\Theta - I_m\,, \tag{12}$$

and an error signal $S_\ell \in \mathbb{R}^{n\times n}$ based on the left inverse

$$S_\ell = \Theta K - I_n \,. \tag{13}$$

Consider initially the error (12) and the positive function $2\,V_r = \mathrm{tr}(S_r^\mathsf{T} S_r)$, where $\mathrm{tr}(\cdot)$ is the trace function. The time-derivative of V_r is given by $2\,\dot V_r = 2\mathrm{tr}(S_r^\mathsf{T} \dot K \Theta) + 2\mathrm{tr}(S_r^\mathsf{T} K \dot\Theta)$. In view of the above equation, we choose the following update law

$$\dot\Theta = -\Gamma K^\mathsf{T} S_r \,, \tag{14}$$

where $\Gamma = \Gamma^\mathsf{T} > 0$ is the update gain matrix. As a result, we have that

$$\dot V_r = \mathrm{tr}(S_r^\mathsf{T} \dot K \Theta) \underbrace{- \mathrm{tr}(S_r^\mathsf{T} K \Gamma K^\mathsf{T} S_r)}_{\le 0} \,. \tag{15}$$

Similarly, for the error (13) we consider the positive function $2\,V_\ell = \mathrm{tr}(S_\ell^\mathsf{T} S_\ell)$, which has time-derivative along the trajectories of the system given by: $2\,\dot V_\ell = 2\mathrm{tr}(S_\ell^\mathsf{T} \Theta \dot K) + 2\mathrm{tr}(S_\ell^\mathsf{T} \dot\Theta K)$. Now, the update law is selected as

$$\dot\Theta = -\Gamma S_\ell K^\mathsf{T} \,, \tag{16}$$

where $\Gamma = \Gamma^\mathsf{T} > 0$. As as result, we have

$$\dot V_\ell = \mathrm{tr}(S_\ell^\mathsf{T} \Theta \dot K) \underbrace{- \mathrm{tr}(K S_\ell^\mathsf{T} \Gamma S_\ell K^\mathsf{T})}_{\le 0} \,. \tag{17}$$

Remark 1. In the case where the Jacobian matrix $K \in \mathbb{R}^{m\times n}$ is square ($m = n$) and non-singular both update laws (14) and (16) can be used to solve the inverse kinematics problem, since the left and right inverse matrices are equal. However, for the case of a non-square Jacobian matrix K ($m \neq n$), the error matrices S_r and S_ℓ have different dimensions. For a Jacobian matrix with full rank by rows, that is, $\mathrm{rank}(K) = m$, there are infinitely many solutions X such that $KX = I$, while there is no solution Y such that $YK = I$. A preliminary interpretation in terms of the kinematic control approach is that the right inverse obtained from (14) − driven by the error matrix S_r − allows the trajectory tracking in the operational space but, because of the plethora of solutions, the control variable u has components in the null space of K. By the other hand, it can be shown that the left inverse (16) − driven by the error matrix S_ℓ − minimizes the projection of u in the null space of K, resulting in an optimal control variable (Vargas et al., 2013).

Thus, in view of the above remarks, we propose an update law based on both error matrices, S_r and S_ℓ, simultaneously as

$$\dot\Theta = -\Gamma(K^\mathsf{T} S_r + S_\ell K^\mathsf{T}) \,, \tag{18}$$

where $\Gamma = \Gamma^\mathsf{T} > 0$ is the update gain matrix. We can show that the composite update law (18) is easily obtained from the positive function $V_c = V_r + V_\ell$. As a result, the time-derivative of V_c is given by

$$\dot V_c = f(\dot K) \underbrace{- \mathrm{tr}\big((S_r^\mathsf{T} K + K S_\ell^\mathsf{T})\,\Gamma\,(S_\ell K^\mathsf{T} + K^\mathsf{T} S_r)\big)}_{\le 0} \,.$$

For $\dot K \equiv 0$, we have that $\dot V_c \le 0$. In this case, one has that $\dot\Theta = 0$ and $\dot V_c = 0$ when $S_\ell K^\mathsf{T} + K^\mathsf{T} S_r = 0$, that is,

$$\Theta K K^\mathsf{T} + K^\mathsf{T} K \Theta = 2K^\mathsf{T}. \tag{19}$$

Notice that (19) is a *Sylvester equation* and it has a unique solution Θ if and only if K has full rank. The solution converges to the pseudo-inverse of K. If K has reduced rank, the solution is not unique and the output of the algorithm depends on the initial condition $\Theta(0)$. It is easily shown that both conditions $K^\mathsf{T} S_r = 0$ and $S_\ell K^\mathsf{T} = 0$ are verified if (19) holds true. It is noteworthy that the only assumption required to employ this method to solve the IK problem is that the desired trajectory $r(t)$ is not converging (slower than exponentially) to a singularity. The convergence property of the FI algorithm for the update law (18) is analyzed employing the *singular value decomposition* (SVD) approach and can be found in (Vargas et al., 2013).

3. MODIFIED CONTROL LAW

Resorting to Section 2, we consider the scalar gain k and its inverse function k^{-1}. The inverse is an odd function and has the same sign of k. However, the same property is not verified for the function k and its filtered inverse θ. In order to recover the equality of sign, we propose a new way to apply the parameter θ in the control law u in (2) defining

$$\theta_M = \theta^2\, k \,, \tag{20}$$

so that $\mathit{sign}(\theta_M) = \mathit{sign}(k)$. For the multivariable case a similar modification is also proposed and, thus, the matrix Θ_M is given by

$$\Theta_M = \Theta\Theta^\mathsf{T} K^\mathsf{T} \,. \tag{21}$$

It is worth mentioning that $\Theta\Theta^\mathsf{T}$ is a symmetric positive semi-definite matrix.

Now, consider the tracking control problem for a reference trajectory $r(t)$ such that error dynamics is governed by $\dot e = \dot r(t) - K\,u$, where u is the control law and K is the Jacobian matrix. Choosing the Lyapunov function candidate as $2V_e = e^\mathsf{T} \Lambda\, e$, its time-derivative along the trajectories of the system is given by $\dot V_e = e^\mathsf{T} \Lambda(\dot r - K\,u)$. Then, for $u = u_M$ and taking $u_M = \Theta\,\Theta^\mathsf{T} K^\mathsf{T} v$, we obtain

$$\dot V_e = e^\mathsf{T} \Lambda (I - N\,N^\mathsf{T})\,\dot r \underbrace{- e^\mathsf{T} \Lambda^\mathsf{T} N\,N^\mathsf{T} \Lambda\, e}_{\Phi} , \tag{22}$$

where $N = K\Theta$. Notice that the term Φ corresponds to a negative semi-definite term, regardless of the quality of the estimate given by the FI algorithm (18).

In the case where K has reduced rank, the use of $u_M(v) = \Theta_M v$ cancels the components of v in the null space of K^T, that is, for $v = v_c + v_n$, where $v_c \in \mathrm{Col}(K)$ and $v_n \in \mathrm{Nul}(K^\mathsf{T})$, we have that $u_M(v) = u_M(v_c)$. It is worth mentioning that V_e is also defined in terms of the weighted error variable $e_w = \Lambda e$. Thus, for the case where the reference trajectory $r(t)$ is out of reach, the gain matrix Λ affects the obtained solution by weighting the different control objectives.

4. AUGMENTED JACOBIAN MATRIX

The FI algorithm provides a solution for the inverse kinematics problem even when unfeasible or unreachable reference trajectories are considered. As we show, the proposed algorithm allows for prioritizing one of the primary control objectives simply by adjusting the gain matrix. In addition, we can include an objective function f in the solution to satisfy an additional constraint to the tracking control problem as

$$\begin{bmatrix} \dot{y} \\ \dot{f} \end{bmatrix} = \underbrace{\begin{bmatrix} K \\ K_f \end{bmatrix}}_{K_+} u\,, \tag{23}$$

where $K_f \in \mathbb{R}^{1\times n}$ is the constraint Jacobian matrix and $K_+ \in \mathbb{R}^{m_e\times n}$ is the *augmented Jacobian matrix*, with $m_e = m+1$. Notice that, considering the secondary control objective, it is possible that $m_e > n$ which justifies the use of the error signal based on the right inverse (12). Similarly to $u_{_M}(v) = \Theta_M\, v$ with $\Theta_M = \Theta\Theta^{\mathsf{T}} K^{\mathsf{T}}$, the control law $u_{_M}(t)$ to the augmented inverse kinematics problem is

$$u_{_M}(t) = \Theta_+ \Theta_+^{\mathsf{T}} K_+^{\mathsf{T}}\, v_+\,, \tag{24}$$

where the matrix Θ_+ is the filtered inverse of the matrix K_+ and v_+ is the augmented Cartesian control signal

$$v_+ = \begin{bmatrix} v \\ v_f \end{bmatrix} = \begin{bmatrix} \dot{r} + \Lambda\, e \\ -f \end{bmatrix}. \tag{25}$$

The natural choice for v_f would be $v_f = -\lambda_f\, f$ where $\lambda_f \geq 0$ is the proportional gain of the objective function. However, we choose to include the weighting given by λ_f directly in the computation of function f. The advantage of this choice is that for the case where there is no constraint or the constraint has low priority, the elements of K_f are equal to zero or have small magnitude, respectively.

5. APPLICATION TO ROBOT CONTROL

To illustrate the applicability of the proposed algorithm, we consider the tracking control problem for a MIMO nonlinear system represented by a robot manipulator.

5.1 Robot kinematics

First, we consider the kinematic modeling of a robot manipulator. The joint spaces variables are related to the operational space variables by means of the following forward and differential mappings

$$p = h(q)\,, \qquad \dot{p} = J_p(q)\, \dot{q}\,, \tag{26}$$

where $h(q)$ is a m-dimensional vector function, nonlinear in general, and $J_p(q) = (\partial h/\partial q) \in \mathbb{R}^{m\times n}$ is the *analytical Jacobian*. Notice that $p, \dot{p} \in \mathbb{R}^m$ denote the position and the linear velocity of the robot end-effector, and $q, \dot{q} \in \mathbb{R}^n$ are the position and velocity of the manipulator joints.

The orientation of the robot end-effector can be described by the *unit-quaternion representation* given by a four-dimensional vector $\phi = \{\eta, \epsilon\} \in \mathbb{H}^1$, where $\eta \in \mathbb{R}$ is the scalar part and $\epsilon \in \mathbb{R}^3$ is the vector part, subject to the unit norm constraint $\eta^2 + \epsilon^{\mathsf{T}}\epsilon = 1$. The so-called *quaternion propagation rule* relates the time-derivative of the unit quaternion $\dot{\phi} \in \mathbb{H}$ with the angular velocity of the robot end-effector $\omega \in \mathbb{R}^3$ as

$$\dot{\phi} = \frac{1}{2}\, E(\phi)\, \omega\,, \quad E(\phi) = \begin{bmatrix} -\epsilon^{\mathsf{T}} \\ \eta I - Q(\epsilon) \end{bmatrix}, \tag{27}$$

where $Q(\cdot) : \mathbb{R}^3 \mapsto \mathrm{so}(3)$ denotes the skew-symmetric matrix operator and $J_r(\phi) = 2\, E^{\mathsf{T}}(\phi) \in \mathbb{R}^{3\times 4}$ is the well-known *representation Jacobian*.

[1] The symbol $\mathbb{H}$ denotes the unit quaternion group satisfying the algebra of quaternion (Siciliano et al., 2009).

The differential kinematics equation provides the relationship between the joint velocities vector and the corresponding linear and angular velocity of the robot end-effector as

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} J_p(q) \\ J_o(q) \end{bmatrix} \dot{q} = J(q)\, \dot{q}\,, \tag{28}$$

where $J(q) \in \mathbb{R}^{m'\times n}$ is the *geometric Jacobian* of the robot manipulator. Notice that, the orientation of the robot end-effector can be given in terms of the manipulator joint angles as $\phi = g(q)$, where $g(\cdot)$ is a vector function, nonlinear in general. Thus, taking into account (27) we have $J_o(q) = J_r(\phi)\, (\partial g(q)/\partial q)$. It is worth mentioning that the kinematic model has the following property very useful for the stability analysis of robot manipulators with revolute joints.

Property 1 (Dixon, 2007): $J(q)$ is bounded for all possible values of $q(t)$, that is, it depends on $q(t)$ as arguments of limited trigonometric functions and $\|J(q)\|_\infty \leq c_1$, where $c_1 \in \mathbb{R}$ is a known positive constant.

5.2 Kinematic control

Consider the kinematic control problem for a n-DoF robot manipulator. In this framework, the robot motion can be simply described by

$$\dot{q}_i = u_i\,, \quad (i = 1, \cdots, n)\,, \tag{29}$$

where q_i and $\dot{q}_i$ are the angular position and the angular velocity of the i-th joint respectively, and u_i is the velocity control signal applied to the i-th joint motor drive. This approach can be applied to most commercial robots with high gear ratios or when the task speed is slow. Then, from (28) and considering the kinematic control approach (29), we obtain the following control system

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q)\, u\,. \tag{30}$$

For the case where the Jacobian matrix is *non-square* $(m < n)$, the velocity control signal $u \in \mathbb{R}^n$ is given by

$$u(t) = J^{\dagger}(q)\, v\,, \tag{31}$$

where $J^{\dagger} = J^{\mathsf{T}} (J\, J^{\mathsf{T}})^{-1}$ is the right pseudo-inverse of J and $v \in \mathbb{R}^m$ is a Cartesian control signal to be designed.

The control signal (31) locally minimizes the norm of the velocities of the joints, provided that (A1) the robot kinematics is *known* and (A2) $v(t)$ does not lead the robot to *singular configurations*. The failure of this last condition is a fairly open-problem in robotics area, and will be discussed in what follows. Notice that, in the case where the Jacobian matrix is square $(m = n)$ and non-singular, the velocity control signal is given by $u(t) = J^{-1}(q)\, v$.

Remark 2. Due to the presence of $n - m$ redundant degrees of freedom, the solution (31) can be modified by introducing a term belonging to the null space of J, *i.e.*

$$u(t) = J^{\dagger}\, v + \left(I - J^{\dagger}\, J\right) \dot{q}_0\,, \tag{32}$$

where $\dot{q}_0$ is a vector of arbitrary joints velocities that can be specified to satisfy an additional constraint with a secondary priority. The solution obtained locally minimizes the norm of the joints velocities and allows the generation of internal movements to reconfigure the manipulator structure without changing the pose of the robot end-effector (Siciliano et al., 2009).

5.3 Position and orientation control

Consider the position and orientation control problem for a n-DoF robot manipulator. The control goal is to track a time-varying desired pose $x_d(t) = [\, p_d(t) \;\; \phi_d(t) \,]^{\mathsf{T}}$, assumed to be bounded, from the current pose of the robot end-effector $x = [\, p \;\; \phi \,]^{\mathsf{T}}$, that is

$$\lim_{t\to\infty} \begin{bmatrix} p \\ \phi \end{bmatrix} = \begin{bmatrix} p_d(t) \\ \phi_d(t) \end{bmatrix} . \qquad (33)$$

The *position error* is simply defined as $e_p = p_d(t) - p$, but the *orientation error* should be defined in terms of the algebra of rotation groups, instead of the vector algebra. Considering the unit-quaternion representation, it is common to choose the orientation error e_o as the vector part of the error quaternion

$$e_o := \Delta\epsilon = \eta(q)\,\epsilon_d - \eta_d\,\epsilon(q) - Q(\epsilon_d)\,\epsilon(q)\,, \qquad (34)$$

where the pair $\{\eta_d\,, \epsilon_d\}$ denotes respectively the scalar and the vector parts of the desired quaternion. The error quaternion is defined as $\Delta\phi = \{\Delta\eta, \Delta\epsilon\} = \phi_d * \phi^{-1}$, where the symbol "$*$" denotes the quaternion product operator. Notice that $\Delta\phi = \{1, 0^{\mathsf{T}}\}$ if and only if ϕ and ϕ_d are coincident.

At this point, we can calculate the velocity control signal u based on a Jacobian pseudo-inverse solution

$$u = J^{\dagger}(q) \begin{bmatrix} v_p \\ v_o \end{bmatrix} = J^{\dagger}(q) \begin{bmatrix} \dot{p}_d + \Lambda_p\, e_p \\ \omega_d + \Lambda_o\, e_o \end{bmatrix} , \qquad (35)$$

where $\Lambda_p = \Lambda_p^{\mathsf{T}} > 0$ and $\Lambda_o = \Lambda_o^{\mathsf{T}} > 0$ are the position and orientation gain matrices respectively, and $\omega_d \in \mathbb{R}^3$ is the desired angular velocity for the robot end-effector. The stability and convergence analysis of the kinematic control approach (35) applied to a non-redundant robot manipulator can be found in (Leite et al., 2009). Fig. 2 shows the block diagram of the inverse kinematics algorithm applied to the tracking control problem considering only the position of the robot end-effector. The complete block diagram including the orientation control problem is intuitive and will be omitted here.

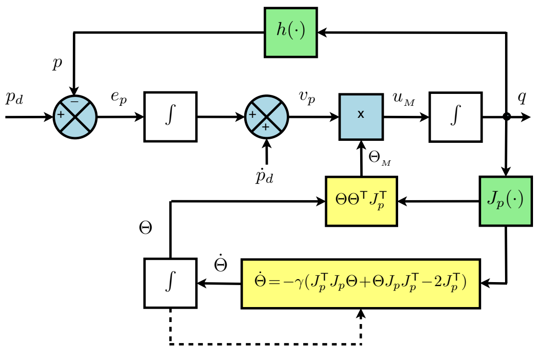

Fig. 2. Block diagram of the kinematic control algorithm.

6. SIMULATION RESULTS

In this section, we present simulation results obtained for some cases of interest. In case studies I and II, the performance of the proposed algorithm is measured from the tracking of trajectories that have internal/boundary singularities and compared with the DLS algorithm using a damping factor $\delta = 300$ and a singularity neighborhood defined by the minimum manipulability measure $\omega_0 = 10^3$ (Vargas et al., 2013). In case study III, the orientation control problem is also considered and the performance is analyzed for points where the desired position and orientation cannot be achieved simultaneously, that is, the desired posture is unreachable. Lastly, in case study IV, we consider the presence of an obstacle in the workspace of a redundant robot manipulator.

6.1 Case Study I - Internal singularities

Here, we consider only the position tracking problem for a 6-DoF Zebra Zero manipulator (IM Inc.) without actuation in the last 3 joints, configuring a 3-DoF non-redundant case. The trajectory is in the vicinity of internal singularities (or joint space singularities), that is, singular configurations belonging to the robot workspace (Fig. 3).

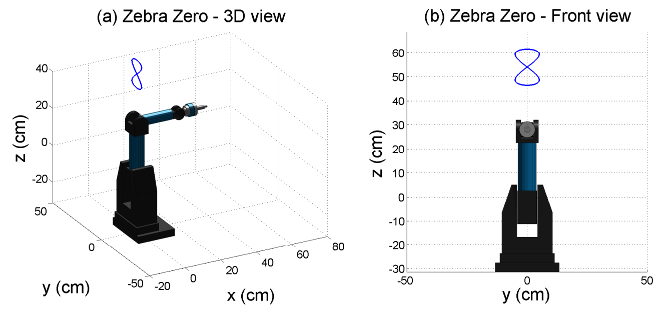

Fig. 3. Reference trajectory with internal singularities.

The length of the robot links are $\ell_1 = 27.94\,cm$ and $\ell_2 = 39.36\,cm$. The initial conditions and the control parameters are: $q(0) = [\,0 \;\; \pi/2 \;\; -\pi\,]^{\mathsf{T}}\,rad$, $\Theta(0) = 0_{3\times3}$, $\Lambda = 2\,\mathrm{I}$ and $\Gamma = \mathrm{I}$. The trajectory tracking for the FI and DLS algorithms as well as the manipulability measure are depicted in Fig. 4(a) and (b) respectively. The norm of the tracking errors is shown in Fig. 4(c) and (d), where it can be observed that the proposed algorithm allows for the tracking of the reference trajectory with small error, even in singular configurations.

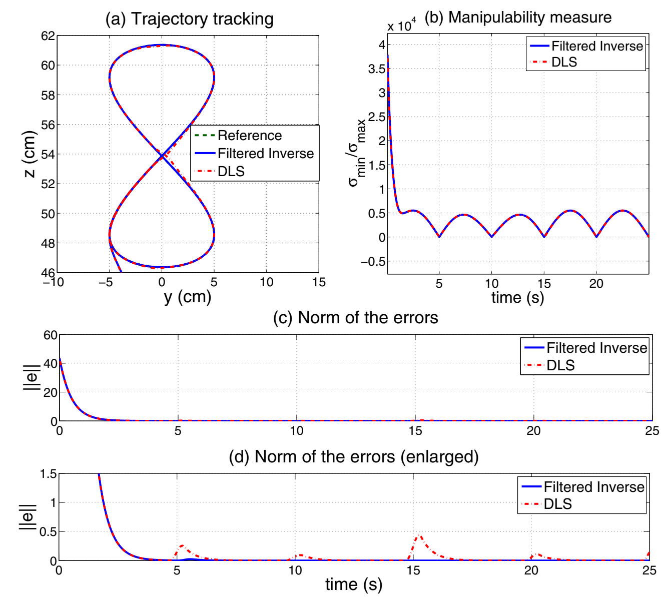

Fig. 4. Trajectory tracking in yz plane, manipulability measure and norm of the position errors.

6.2 Case Study II - Boundary singularities

Now, we consider an extreme case, in which part of the reference trajectory does not belong to the workspace of the manipulator. This type of singularity is known as boundary singularity. The reference trajectory is depicted in Fig. 5 and the trajectories tracking for two different update gains $\Gamma = \gamma\, \mathrm{I}$ are shown in Fig. 6. The update gain Γ is directly related to the performance of the proposed algorithm and for high gains ($\gamma = 25$) the trajectory deviations in the transition points are small, as shown in Fig. 6(b) and (c).

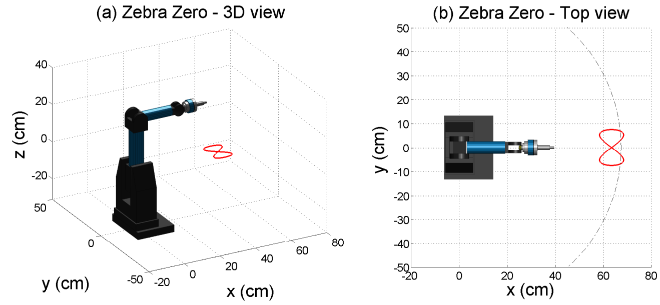

Fig. 5. Reference trajectory with boundary singularities.

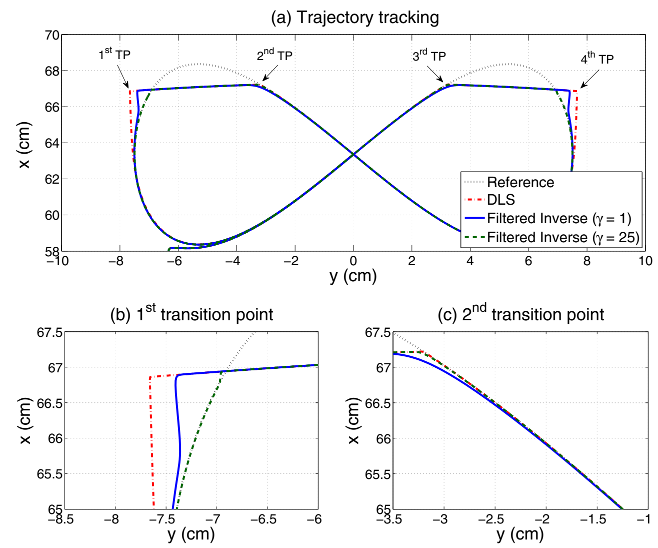

Fig. 6. Trajectories in xy plane and the transition points.

6.3 Case Study III - Position and orientation

Now, we consider the 6-DoF Zebra-Zero manipulator with full actuation. The control goal is to track the time-varying desired position while keeping the orientation of the robot end-effector constant, configuring a 6-DoF non-redundant case. We also consider: $\Theta(0) = 0_{6\times 6}$, $\Gamma = 25\,\mathrm{I}$, $q(0) = [0 \;\; \pi/2 \;\; -\pi \;\; 0 \;\; \pi/2 \;\; 0]^{\mathsf{T}}\, rad$, $\Lambda_p = \lambda_p\, \mathrm{I}$, $\Lambda_o = \lambda_o\, \mathrm{I}$ and the unit-quaternion formulation to describe the orientation of the robot end-effector. The desired orientation ϕ_d is given by the elementary rotation by an angle $\pi\, rad$ about y-axis. Fig. 7 shows the norms of the position and orientation errors, e_p and e_o, for different combinations of proportional gains λ_p and λ_o.

For those points in which the time-varying desired pose $x_d(t)$ cannot be fully achieved, the choice of gains allows

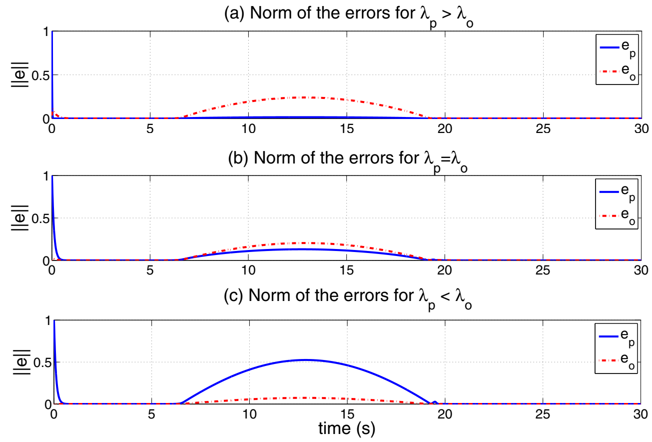

Fig. 7. Norm of the position and orientation errors.

for the weighting of the position and orientation errors in the operational space. In Fig. 7(a) the position gain ($\lambda_p = 100$) is higher than the orientation gain ($\lambda_o = 10$) and, therefore, the position control objective is prioritized. By the other hand, in Fig. 7(c) the orientation gain ($\lambda_o = 100$) is higher than the position gain ($\lambda_p = 10$), prioritizing the orientation control objective. The choice $\lambda_p = \lambda_o$ results in similar errors in the workspace, as shown in Fig. 7(b). Fig. 8(a) and (b) depict the sections of the trajectory tracking with position and orientation priorities, respectively.

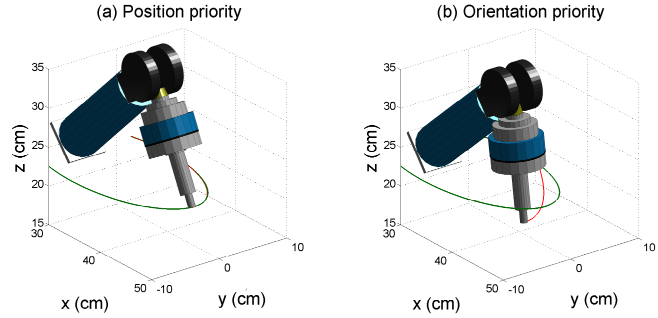

Fig. 8. Trajectory tracking with different priorities.

6.4 Case Study IV - Obstacle avoidance

Here, we consider the position tracking control problem for a 3-R planar robot manipulator ($\ell_1 = 2$ and $\ell_2 = \ell_3 = 1$) in the presence of a single obstacle. The objective function $f(\cdot)$ for obstacle avoidance is inspired by the *Gaussian function* and it is defined in terms of the distance Δp between the points $p_j \in \mathbb{R}^3$, $(j = 1 \cdots n)$, fixed on the robot joints to the centre of the obstacle $\mu \in \mathbb{R}^m$.

The level curves are ellipsoids defined by the matrix $M_f = R_f\, D_f^{-1}\, R_f^{\mathsf{T}}$ with $M_f = M_f^{\mathsf{T}} > 0$, where $R_f \in SO(3)$ is the rotation matrix which defines the axes of the ellipsoids and $D_f \in \mathbb{R}^{3\times 3}$ is the stretching curve matrix which has positive elements in its diagonal, whose roots are proportional to the stretching of the level curve at these axes. The objective function that we consider is given by:

$$f(p_1, \cdots, p_n) := \sum_{j=1}^{n} \alpha_j e^{-(p_j-\mu)^{\mathsf{T}} M_f\, (p_j-\mu)} \tag{36}$$

where $\alpha_j > 0$ is a scalar gain. Applying the coordinates transformation given by the matrix R_f, we can rewrite the objective function as:

$$f(p_1, \cdots, p_n) = \sum_{j=1}^{n} \alpha_j e^{-(\Delta p_j)'^{\mathsf{T}} D_f^{-1} (\Delta p_j)'} \qquad (37)$$

where $(\Delta p_j)' = R_f^{\mathsf{T}} \Delta p_j$ is the representation of the vector $\Delta p_j = (p_j - \mu)$ in the original coordinate frame rotated by the matrix R_f. The parameters of the objective function are defined in the operational space by $\mu = (\mu_x, \mu_y) = (2, 0.75)$, $R_f = \mathrm{I}$ and $D_f = (0.2)^2\,\mathrm{I}$ evaluated at the robot end-effector. The simulation results obtained with the FI algorithm considering $\Theta(0) = 0_{3\times2}$, $\Gamma = 5\,\mathrm{I}$ and $\Lambda = 5\,\mathrm{I}$ are shown in Fig. 9(a)-(c) respectively.

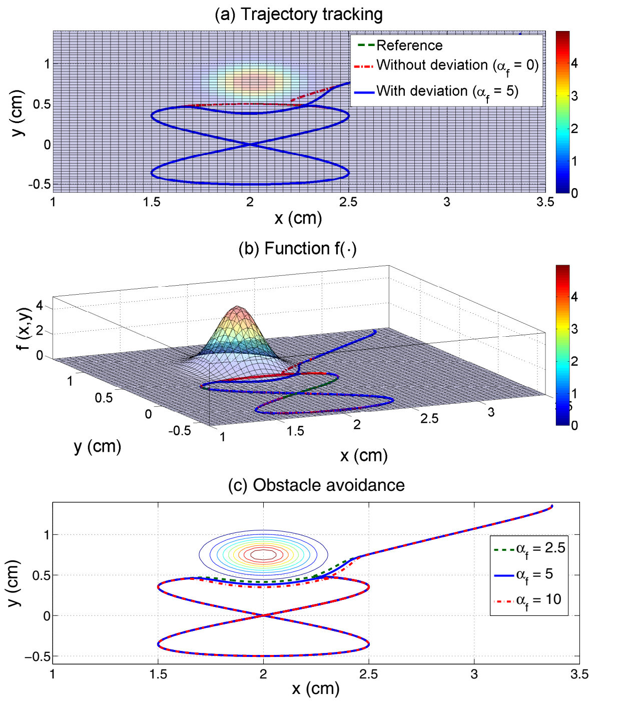

Fig. 9. Trajectory tracking with obstacle avoidance.

The use of positive values for α_f allows the robot end-effector to deviate from the obstacle defined by function f as depicted in Fig. 9(a) and (c), where it can be observed the trajectory tracking for different values of gain α_f and the level curves of the normalized function f over α_f. As expected, the higher is the gain α_f, the greater is the deviation. Besides, the function f shown in Fig. 9(b) can be evaluated in intermediate points along the links of the robot arm, allowing internal reconfiguration and avoiding collision between links and obstacles.

7. CONCLUDING REMARKS

In this work, we present an alternative algorithm that uses the filtered inverse of the Jacobian matrix to tackle the inverse kinematics problem and deal with kinematics singularities simultaneously. A generalization of the filtered inverse approach for non-square matrices is guaranteed by means of a composite update law built up from left and right error matrices.

The combination of the proposed solution with the proper tuning of the gain matrices allows for weighting of one of the primary control objectives (e.g., position or orientation) when the reference trajectory is out of reach. Besides, an augmented Jacobian matrix can be defined to include an additional constraint to the kinematic control problem with secondary priority (e.g., obstacle avoidance).

Compared to other inversion algorithms described in the literature, the main advantage of our proposal is related to the *number of design parameters* to be tuned – only the update gain – and the *computational efficiency*, since it does not require matrix inversion, singular value decomposition or computation of manipulability measures. Some potential topics for future research comprise the application of the proposed solution to closed-chain mechanisms and relaxing the condition of the complete knowledge of the robot kinematics.

REFERENCES

S. Chiaverini, G. Oriolo, and I. D. Walker. Kinematically redundant manipulators. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 245–268. Springer-Verlag Ltd., 1st edition, 2008.

W. E. Dixon. Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics. *IEEE Trans. Aut. Contr.*, 52(3):488–493, 2007.

K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Trans. Graph.*, 23 (3):522–531, 2004.

R. Kolodny, L. Guibas, M. Lewitt, and P. Koehl. Inverse kinematics in biology: The protein loop closure problem. *Int. J. Rob. Res.*, 24(2-3):151–163, 2005.

A. C. Leite, F. Lizarralde, and L. Hsu. Hybrid adaptive vision-force control for robot manipulators interacting with unknown surfaces. *Int. J. Rob. Res.*, 28(7):911–926, 2009.

A. A. Maciejewski and C. A. Klein. The singular value decomposition: Computation and applications to robotics. *Int. J. Rob. Res.*, 8(6):63–79, 1989.

Y. Nakamura and H. Hanafusa. Inverse kinematic solution with singularity robustness for robot manipulator control. *ASME J. Dyn. Syst. Meas. Contr.*, 108(3):163–171, 1986.

D. N. Nenchev, Y. Tsumaki, and M. Uchiyama. Singularity-consistent parameterization of robot motion and control. *Int. J. Rob. Res.*, 19(2):159–182, 2000.

A. N. Pechev. Inverse kinematics without matrix inversion. In *IEEE Int. Conf. Rob. Aut.*, pages 2005–2012, Pasadena, CA, USA, 2008.

B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Inc., 2009.

K. Tchon and R. Muszynski. Singular inverse kinematic problem for robotic manipulators: A normal form approach. *IEEE Trans. Rob. Aut.*, 14(1):93–104, 1998.

L. V. Vargas, A. C. Leite, and R. R. Costa. Kinematic control of robot manipulators using filtered inverse. In *Med. Conf. Contr. Aut.*, pages 27–33, Crete, Greece, 2013.

C. W. Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans. Syst. Man Cybern.*, 16(1):93–101, 1986.

L-C. T. Wang and C. C. Chen. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Trans. Rob. Aut.*, 7(4): 489–499, 1991.