A real-time approach for singularity avoidance in Resolved Motion Rate Control of Robotic Manipulators

Giacomo Marani$^1$, Jinhyun Kim$^2$, Junku Yuh$^1$ and Wan Kyun Chung$^2$

$^1$Autonomous Systems Laboratory, University of Hawaii
E-mail: {marani,jyuh}@wilki.eng.hawaii.edu
$^2$Robotics & Bio-Mechatronics Lab., Pohang University of Science & Technology (POSTECH)
E-mail: {pluto,wkchung}@postech.ac.kr

Abstract

In autonomous system, it is important to establish a control scheme that works with stability even near singularity configurations. In this article, we describe an on-line trajectory control scheme that uses the manipulability measure as a distance criteria to avoid manipulator singularities. The proposed approach consists in a method for limiting the minimum value of the distance criteria. The performance is simply affected by the choice of the lower limit. Based on a real-time evaluation of the measure of manipulability, this method does not require a preliminary knowledge of the singular configurations. The proposed algorithm is validated by experimental results.

1 Introduction : Resolved Motion Rate Control

The kinematic output of a generic robotic manipulator is usually represented by a manipulation variable, $r \in \mathbb{R}^n$. A manipulation variable may be, but it is not limited to, the position and orientation of the end-effector. The relationship between $r$ and the joint angles, $q$, is represented by the following equation:

$$ r = f(q) $$  \hspace{1cm} (1)

Invoking small variation, the relationship between $\delta r$ and $\delta q$ is given by:

$$ \delta r = \frac{\partial f}{\partial q} \delta q = J(q) \delta q $$ \hspace{1cm} (2)

where $J(q) \in \mathbb{R}^{n \times n}$ is the Jacobian matrix of the manipulation variable, $r$. In resolved motion rate control [1], we compute $\delta q$ for a given $\delta r$ and $q$ by solving the linear system, Eq. (2). In the general case, this is done by using the pseudo-inverse of the Jacobian matrix as follows [2]:

$$ \delta q = J^+(q) \delta r + (I_n - J^+(q)J(q)) y $$  \hspace{1cm} (3)

where $J^+(q) \in \mathbb{R}^{n \times m}$ is the Moore-Penrose pseudo-inverse of $J(q)$, $y \in \mathbb{R}^n$ is an arbitrary vector and $I_n \in \mathbb{R}^{n \times n}$ indicates an identity matrix.

A singular point is defined as the joint configuration vector $q^*$ where $J(q^*)$ is not of full rank. Its pseudo-inverse, $J^+(q^*)$, is not defined at such configuration. Moreover, in the neighborhood of singular points, even a small change in $\delta r$ requires an enormous change in $\delta q$, which is non-practically feasible in real manipulators and also dangerous for the structure.

The damped least-squares method (Wampler, [3]) is a classical and simple way to overcome this drawback. It consists in adding a regularization term acting in the neighborhood of the singularities:

$$ J^+_{DLS} = J^T(JJ^T + \lambda I)^{-1} $$ \hspace{1cm} (4)

Consequently, the damped least square solution of Eq.(2) is as following:

$$ \delta q_{DLS} = J^+_{DLS} \delta r + [I_n - J^+_{DLS}(q)J(q)] y \hspace{1cm} (5) $$

However, the main disadvantage for this approach is a loss of performance and an increased tracking error [4]. The choice of damping constant must balance the required performance and the error allowed. To overcome those defects, a variable damping factor [5] and numerical filtering of the velocity components are introduced. And, Chiaverini [4] also proposed a modified inverse adding only the damping parameter to the lowest singular values. Their results are shown to be better than those of damped least-squares method, but still have the tuning problem of the damping coefficient.

In autonomous systems, where the human intervention is limited or absent, the possible presence of the above drawback naturally suggests avoiding all the singular configurations. In our works, the real-time singularity free path is generated by a geometric approach. We describe an online trajectory control scheme that uses the manipulability measure as a distance criteria to avoid manipulator singularities. In addition, the performance is predictable upon the lower-limit of the measure of manipulability.
The concept of the manipulability measure is briefly commented in section 2. In section 3, the proposed algorithm for avoiding singularity is derived. Extension to the multiple tasks of redundant resolution is also introduced and experimental verifications are executed in section 4 and 5, respectively. Finally, the contribution of our works are summarized.

2 Measure of Manipulability

The first step in avoiding singularities is to locate themselves in the joint space. Yoshikawa [6] proposed a continuous measure that evaluates the kinematic quality of robot mechanism:

\[
\text{mom} = \sqrt{\det \left[ JJ^T \right]} \quad (6)
\]

\( \text{mom} \) takes a continuous non-negative scalar value and becomes equal to zero only when the Jacobian matrix is not of full rank. As a matter of fact, the singular value decomposition of the Jacobian matrix is:

\[
J(q) = U \Sigma V^T \quad (7)
\]

where \( U \) and \( V \) are orthogonal matrices and \( \Sigma \) is a \( m \times n \) matrix whose diagonal elements are the ordered singular values of \( J \):

\[
\Sigma = \begin{cases} 
\left[ \text{diag}(\sigma_1, \cdots, \sigma_m) | 0 \right], & (m \leq n) \\
\left[ \text{diag}(\sigma_1, \cdots, \sigma_n) \right] & (m > n)
\end{cases} \quad (8)
\]

Substituting Eq. (7) into Eq. (6) results in:

\[
\text{mom} = \sqrt{\det \left( U \Sigma V^T V \Sigma^T U^T \right)} = \sqrt{\det(\Sigma \Sigma^T)} = \prod_{i=1}^{n} |\sigma_i| \quad (9)
\]

being \( U \) and \( V \) orthogonal matrices.

Equation (9) shows that \( \text{mom} \) is exactly the product of the singular values of \( J \) and can be regarded as a distance from singularity.

In addition, for the future use, it is necessary to find the derivative of Eq. (6) with respect to the joint configuration vector, \( q \). It is simply calculated as following: [7]

\[
\frac{\partial \text{mom}(q)}{\partial q_k} = \text{mom}(q) \cdot \text{trace} \left\{ \frac{\partial J}{\partial q_k} J^+ \right\} \quad (10)
\]

Equation (10) shows that we can express the derivative of measure of manipulability with respect to some already known quantities, as \( \text{mom}(q) \) itself and the pseudo-inverse of Jacobian matrix, \( J^+ \).

In appendix A, the fast calculation of \( \text{mom} \) is described.

The small variation of the measure of manipulability, \( \delta \text{mom}(q) \), is given by:

\[
\delta q = J^+(q) \delta r \quad (11)
\]

The small variation of the measure of manipulability, \( \delta \text{mom}(q) \), is given by:

\[
\delta \text{mom}(q) = \frac{\partial \text{mom}(q)}{\partial q} \delta q = \frac{\partial \text{mom}(q)}{\partial q} J^+ \delta r. \quad (12)
\]

In order to have \( \delta \text{mom}(q) = 0 \), Eq. (12) implies that the given task must be orthogonal to the vector

\[
\frac{\partial \text{mom}(q)}{\partial q} J^+ \quad (13)
\]
or, equivalently, that $\delta r$ must be lie on the surface defined by:

$$\{ x \in \mathbb{R}^m : \left( \frac{\partial mom(q) J^+}{\partial q} \right) \cdot x = 0 \}$$

Let $n_m$ be the unitary vector orthogonal to the surface Eq. (14):

$$n_m = \frac{\left( \frac{\partial mom(q) J^+}{\partial q} \right)^T} {||\frac{\partial mom(q) J^+}{\partial q}||}$$

Consequently, the projection of the given task on the surface is:

$$\delta r_p = \delta r - (\delta r \cdot n_m) n_m$$

(16)

For the good task performance, such projection must be done only when approaching to singularities. At that aim, it is necessary to introduce in Eq. (16) a weight as follows:

$$\delta r_p = \delta r - (\delta r \cdot n_m) n_m k(mom, \overline{m})$$

(17)

where $k(mom, \overline{m})$ is a positive, well shaped function of the measure of manipulability, to be equal to zero for values of $mom$ greater than a predefined threshold, $\overline{m}$, and equal to 1 for values of $mom$ smaller than $\overline{m}/2$:

$$k(m, \overline{m}) = \begin{cases} 
0, & \overline{m} \leq m \\
4 \left( \frac{4m^3 - 9m^2 m + 6m - 1}{m^3} \right), & \frac{m}{2} < m < \overline{m} \\
1, & m \leq \overline{m}/2
\end{cases}$$

(18)

Figure 2 shows an example of that function for $\overline{m} = 0.04$. Notice that the first derivative is equal to zero in correspondence of those two points. This allows to progressively lying down the task solution, $\delta r$, on the surface where $mom$ is constant, without introducing instabilities on the controller when closing the loop. However, when $mom$ is already smaller than the value on the surface, Eq. (17) does not guarantee to escape from within the volume enclosed by the same. Moreover, numerical errors may introduce a small derive term driving the task below the surface. In order to avoid the above drawback, a third term has been introduced in Eq. (17):

$$\delta r_p = \delta r - (\delta r \cdot n_m) n_m k(mom, \overline{m}) + k(mom, \overline{m}/2) n_m$$

(19)

It produces a recalling action toward the surface, starting when $\delta r_p$ has no more components along the gradient (that is when $mom < \overline{m}/2$).

Finally, we must ensure to leave from the surface by acting the task correction in Eq. (19) only when the scalar product $\delta r \cdot n_m$ is positive, that is when the measure of manipulability is decreasing:

$$\delta r_p = \delta r - \delta r_{corr}$$

(20)

where:

$$\delta r_{corr} = \frac{1 - \text{sign}(\delta r \cdot n_m)}{2} (\delta r \cdot n_m) n_m k(mom, \overline{m}) + k(mom, \overline{m}/2) n_m$$

(21)

Thus, Eq. (11) becomes:

$$\delta q = J^+(q) (\delta r - \delta r_{corr})$$

(22)

4 Extension to inverse kinematics with multiple tasks

Equation (22) has been derived considering the inverse kinematics of one manipulation variable. Let’s now extend the study to the inverse kinematics taking account of the priority of the subtasks.

The formulation is based upon the Nakamura task-priority based method, with position prior to orientation [8]. The goal is to avoid singular configurations for the first manipulation variable.

Let the first manipulation variable $r_1 \in \mathbb{R}^{m_1}$ be the position of the end-effector. Likewise, let the second manipulation variable $r_2 \in \mathbb{R}^{m_2}$ be the orientation of the end-effector. Thus we have:

$$\delta r_1 = J_1(q) \delta q$$

(23)

$$\delta r_2 = J_2(q) \delta q$$

(24)

where $q \in \mathbb{R}^n$ is the robot configuration vector; $J_1(q) \in \mathbb{R}^{m_1 \times n}$ and $J_2(q) \in \mathbb{R}^{m_2 \times n}$ are the linear and rotational Jacobian of the system, respectively.

The inverse kinematics taking account of the priority of the subtasks is [8]:

$$\delta q = J_1^+ \delta r_1 + J_2^+ (\delta r_2 - J_2 J_1^+ \delta r_1)$$

$$+ (I_n - J_1^+ J_1) (I_n - J_2^+ J_2) z$$

$$\dot{J}_2 = J_2 (I_n - J_1^+ J_1)$$

(25)

(26)
where $J_i^+ \in \mathbb{R}^{n \times 3}$ is the pseudo-inverse of $J_i(q)$, $J_2^+ \in \mathbb{R}^{n \times 3}$ is the pseudo-inverse of $J_2(q)$, $z \in \mathbb{R}^n$ is an arbitrary vector and $I \in \mathbb{R}^{n \times n}$ indicates an identity matrix.

Let’s consider $z = 0$ in Eq. (25) (null motion absent):

$$\delta q = J_i^+ \delta r_1 + J_2^+ (\delta r_2 - J_2 J_i^+ \delta r_1)$$  \hspace{1cm} (27)

The measure of manipulability $mom_1$ of the first manipulation variable $\delta r_1$ is given by:

$$mom_1 = \sqrt{\det \left[ J_i J_i^+ \right]}$$  \hspace{1cm} (28)

The differential $\delta mom_1(q)$ of the measure of manipulability is given by:

$$\delta mom_1(q) = \frac{\partial mom_1(q)}{\partial q} \delta q$$

$$= \frac{\partial mom_1(q)}{\partial q} \left( \left( J_i^+ - J_2^+ J_2 J_i^+ \right) \delta r_1 + J_2^+ \delta r_2 \right)$$  \hspace{1cm} (29)

In order to have $\delta mom_1(q) = 0$, Eq. (29) implies:

$$\left( \frac{\partial mom_1(q)}{\partial q} \left( J_i^+ - J_2^+ J_2 J_i^+ \right) \right) \cdot \delta r_1 + \left( \frac{\partial mom_1(q)}{\partial q} J_2^+ \right) \cdot \delta r_2 = 0$$  \hspace{1cm} (30)

or

$$\left( \frac{\partial mom_1(q)}{\partial q} \left( J_i^+ - J_2^+ J_2 J_i^+ \right) \right) \cdot \delta r_1 = 0$$  \hspace{1cm} (31)

But, for arbitrary $\delta r_1$ and $\delta r_2$, Eq. (30) is not appropriate. Eq. (31) means that $\delta r_1$ and $\delta r_2$ must be respectively orthogonal to the vectors:

$$n_{1m} = \left( \frac{\partial mom_1(q)}{\partial q} \left( J_i^+ - J_2^+ J_2 J_i^+ \right) \right)^T \left| \frac{\partial mom_1(q)}{\partial q} \left( J_i^+ - J_2^+ J_2 J_i^+ \right) \right|$$  \hspace{1cm} (32)

$$n_{2m} = \left( \frac{\partial mom_1(q)}{\partial q} J_2^+ \right)^T \left| \frac{\partial mom_1(q)}{\partial q} J_2^+ \right|$$

Consequently, the projections of the given tasks are:

$$\delta r_{1p} = \delta r_1 - (\delta r_1 \cdot n_{1m}) n_{1m}$$  \hspace{1cm} (33)

$$\delta r_{2p} = \delta r_2 - (\delta r_2 \cdot n_{2m}) n_{2m}$$  \hspace{1cm} (34)

Finally, Eq. (25) becomes:

$$\delta q = J_i^+ (\delta r_1 - \delta r_{1corr})$$

$$+ J_2^+ (\delta r_2 - \delta r_{2corr} - J_2 J_i^+ (\delta r - \delta r_{1corr}))$$  \hspace{1cm} (35)

5 Application example

The proposed approach has been experimented in the control of the underwater manipulator Maris7080 (Fig. 3), under the SAUVIM project. SAUVIM is a semi-autonomous underwater vehicle equipped with a 7 degrees of freedom electro-mechanical robotic manipulator. The arm, designed for underwater applications at high depths, is internally compensated with appropriate oil. Each degree of freedom is actuated by a brushless motor with reduction unit (spur gear). The accuracy of the angles measure is assured by two resolvers (respectively before and after the gear). A force/torque sensor set between the last degree of freedom of the wrist (G7) and the gripper, allows the acquisition of force and torque data acting on the gripper. Equation (35) has been used in order to avoid singularities for the first manipulation variable (position). This, of course, does not guarantee a singularity-free path for the second manipulation variable (orientation) and the manipulation may fall into a configuration where one or more singular values of the Jacobian $J_2$ are zero. However, a more complete and generalized approach is under investi-
In the current implementation, in order to pseudo-invert $J_2(q)$, see Eq. (26), we used the weighted singularity-robust inverse [5] as follow:

$$J_2^+ = W J_2^T (J_2 W J_2^T + \lambda I_n)^{-1}$$

(37)

The weight $W \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal elements $w_{i,i}$ are given by:

$$w_{i,i} = (1 - k(mom_2, m_2)|n_{3i}|)^2$$

(38)

being $n_{3i}$ the $i$-th component of the unitary vector

$$n_3 = \left( \frac{\delta mom_2}{\delta q} \right)^T$$

(39)

where:

$$mom_2 = \sqrt{\det \begin{bmatrix} J_2 J_2^T \end{bmatrix}}$$

(40)

and $k(mom_2, m_2)$ defined as in Eq. (18). This allows to progressively attenuate, when approaching to a small values of $mom_2$, the components of $\delta q$ responsible of the greatest change in the rate of measure of manipulability.

And, the value of $\lambda$ is chosen according to the distance of $J_2$ from its singular configuration:

$$\lambda = k(mom_2, m_2)$$

(41)

and is acting only when $mom_2$ is lower than the constant limit $m_2$.

Figure 4 show our experimental results. In this example, the given task is a circle partially enclosed in the workspace: the manipulator tries to follow the path, giving the highest priority to the position when running far from a singular configuration. Instead, when approaching to the boundary of the workspace, both the first and second manipulation variable tasks are performed with an error necessary to keep the measure of manipulability equal to the lower limit $m_2$/2(Fig. 4(b)).

A comparison with damped least-squares pseudo-inversion (see Fig. 5(a)) shows that our method has a faster error recovery. In our example, the damping constant has been chosen at the limit of the controller stability for a sample time of 15ms (with smaller values we observed a chattering behavior in the controller output, unless to decrease the sample time). As a matter of fact, because the measure of manipulability is never zero, we can use the exact pseudo-inversion minimizing the (projected) task error. The last differs from the original task only when this is necessary for limiting $mom$.

Figure 4: Experimental results with the proposed algorithm

### 6 Conclusions

The proposed method for singularity avoidance allows moving along a singularity-free path for a generic manipulator whose singular configurations are not preliminarily known. This is done without using a global approach and so it is suitable for a real-time implementation. In addition, one of the advantages of the proposed algorithm is that the performances are predictable. It allows to perform "as much as possible" the desired task under the constraint that the distance from a singular configuration must be greater than a lower limit.

### A Real-time computation of measure of manipulability

In order to speed-up the computation of Eq. (6), that is:

$$mom = \sqrt{\det \begin{bmatrix} J J^T \end{bmatrix}}, J \in \mathbb{R}^{m \times n}$$

(42)
we can use some property of the product of the matrix

\[ M = JJ^T \]  \hspace{1cm} (43)

Because \( M \) is a symmetric semi-positive definite matrix, it has a special, more efficient, and triangular decomposition: the Cholesky decomposition [9]. We can factorize the matrix \( M \) of Eq. (43) as follow:

\[ LL^T = M \]  \hspace{1cm} (44)

where \( L \) is a lower triangular matrix. The components of \( L \) are:

\[ L_{i,i} = \sqrt{m_{i,i} - \sum_{k=1}^{i-1} (L_{i,k})^2} \]  \hspace{1cm} (45)

\[ L_{j,i} = \frac{1}{L_{i,i}} \left( m_{i,j} - \sum_{k=1}^{i-1} L_{i,k} L_{j,k} \right), \quad j = i+1, i+2, \ldots, n \]  \hspace{1cm} (46)

Thus we have:

\[ \text{mom} = \sqrt{\text{trace}(L)} \]  \hspace{1cm} (47)

The operation’s count is about \( n^3/6 \) multiplications and subtractions, with also \( n + 1 \) square roots. In our application, it takes about 0.4ms on a 68060 Motorola CPU(40MHz).

A similar approach has been used to pseudo-inverting the Jacobian matrix. In this way, the overall time for computing Eq. (35) is less than 2ms on the above hardware.

All the simulative tests were made with the aid of Robosim [10], a software package for simulation and fast prototyping in robotics applications.

**Acknowledgements**

This research was done under the SAUVIM project at the Autonomous Systems Laboratory (University of Hawaii). It was sponsored in part by the NSF PYI Award (BES91-57896), NSF (BES97-01614, INT9603043), and ONR (N00014-97-1-0961, N00014-00-1-0629).

**References**


