Symbolic Jacobian Inversion For Redundant Manipulators

Carlos L. Luck and Sukhan Lee

CENG Technical Report 91-30

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213)740-7230

October 1991

Symbolic Jacobian Inversion for Redundant Manipulators

CARLOS L. LÜCK[1] SUKHAN LEE[1,2]

Department of Electrical Engineering - Systems[1]
University of Southern California
Los Angeles, CA 90089-0781

Jet Propulsion Laboratory[2]
California Institute of Technology
Pasadena, CA 91109

October 15, 1991

Abstract

This paper formalizes an efficient method for obtaining the general form of the inverse Jacobian mapping for redundant manipulators, referred to here as the reduced Jacobian method. The method allows for a symbolic form of inverse Jacobian to be derived, providing high accuracy and low computational cost. The analytic derivation of singular configurations is achieved in the process, providing stability of solution and a better insight into the kinematic characteristics of a given manipulator. The approach is based on parameterizing selected joints to define a reduced Jacobian, and on obtaining a particular inverse Jacobian solution as well as the homogeneous solution in the Jacobian null space. An efficient procedure for the avoidance of algorithmic singularities is also discussed. The reduced Jacobian method also provides an efficient computation scheme for the pseudo-inverse matrix and the null space projector operator. Additional computational complexity reduction is obtained when the method is applied to manipulators with a redundant spherical wrist, as shown by a case study based on the 8–DoF AAI arm. Given the formalized method for obtaining a fast, general and stable solution of inverse Jacobian for redundant manipulators, the user can apply criteria for redundancy resolution in terms of a convenient and independent higher level algorithm.

1 Introduction

A robotic manipulator is called redundant when it contains more degrees of freedom (DoF) than required for the task execution. The basic reason for introducing extra joints may be summarized in the following statement:

> "The remarkable dexterity and versatility that the human arm exhibits in performing various tasks can be attributed largely to the kinematic redundancy of the arm, which provides the capability of reconfiguring the arm without affecting the hand position." [29]

The use of redundant manipulators for the purpose of enhancing performance introduces a fundamental problem. Given the end-effector configuration, velocity or acceleration, there are infinite corresponding joint configurations, velocities or accelerations. The problem of selecting one out of the many solutions is referred to as redundancy resolution.

Most researchers have been exploring redundancy resolution in the velocity domain for two main reasons: (1) the lack of a general solution for the inverse kinematics problem in the configuration domain, and (2) the convenience of dealing with the linear system based on the Jacobian matrix. The Jacobian is a matrix of partial derivatives, representing the transformation of velocities from the joint space to the task space:

$$\dot{\boldsymbol{x}} = J\dot{\boldsymbol{\theta}} \tag{1}$$

The mapping from the task space (where the task is given) to the joint space (where the robot actuators are being driven) requires some form of inverse transformation. The use of pseudo-inverses for this purpose was first suggested in [2]:

$$\dot{\boldsymbol{\theta}} = J^{+}\dot{\boldsymbol{x}} + (I - J^{+}J)\boldsymbol{\phi} \tag{2}$$

where $J^{+} = J^{T}(JJ^{T})^{-1}$ provides the minimum Euclidean norm solution and $(I - J^{+}J)$ represents the null space projector operator. This is a general inverse mapping in the sense that any particular $\dot{\boldsymbol{\theta}}$ satisfying equation (1) can be represented by equation (2) for some vector $\boldsymbol{\phi}$.

With the exception of a few contributions (e.g. [4]), very little had been done in this area until recently, when problems with non-redundant manipulators triggered the need for an increasing number of joints. We began realizing that the need for real time control algorithms often prohibited the implementation of computationally demanding schemes such as the pseudo-inverse-based general solution (2). The increasing computational power of new technologies and architectures is not likely to alleviate this burden in the years to come because of the correspondingly increasing demand for more complex algorithms. For instance, global optimization methods require the extensive use of pseudo-inverses [13, 19, 22, 30]. In addition, numerical schemes tend to increase the size of the unstable neighborhood of singularities due to error propagation, since the pseudo-inverse solution is extremely sensitive to small numerical perturbations [32].

Several alternatives to the numerical pseudo-inverse have been proposed in the literature. The use of the Jacobian transpose with iterative convergence is described in [9, 14, 25], where the inversion is avoided at the expense of the need for iterations and difficulties in handling the null space.

[11, 12] propose the incorporation of functional constraints as additional rows to the original Jacobian matrix, defining the extended Jacobian – a square matrix and therefore invertible. The constraints are based on a performance index to be maximized (or minimized). In this case, the redundancy resolution is incorporated in the Jacobian inversion process. This scheme introduces undesirable algorithmic singularities and the inversion is essentially numeric as well. Furthermore, the manipulator has to track the exact local optimum configuration at each sampling time, which may not be feasible in the neighborhood of algorithmic singularities due to mechanical velocity limits of the joints and actuators [23]. The combination of both approaches was described in [29] with the use of the transpose of an extended Jacobian.

The pseudo-inverse approach presents yet another inconvenient property. The minimum norm solution produces non-cyclic behavior, i.e. closed paths in the task space do not produce in general closed paths in the joint space [6, 18]. The addition of a null space term based on optimizing a performance criterion overcomes this difficulty, producing quasi-cyclic behavior. It is not exactly cyclic because the null space term gives us the direction to correct local deviation from the optimal path and, therefore, the system follows closely the optimum trajectory but a small local deviation is always present [17].

In light of the above difficulties, some researchers propose approaching the inverse kinematics problem in the configuration domain [17, 18], but the need for some form of Jacobian inverse mapping has not been totally eliminated, unless a symbolic form solution for the inverse kinematics in the configuration domain is readily available [40].

Given that most of the current research in redundant resolution is based on the availability of the pseudo-inverse and the null space projector operator [16, 35], it would be desirable to compute the pseudo-inverse-based general solution using a computationally efficient algorithm. Several schemes have been recently proposed to produce computational efficiency [24, 34, 38], assuming informally the existence of a particular inverse solution based on what we call a reduced Jacobian.

In this paper we will formalize and discuss in detail the reduced Jacobian concept, describing the algorithmic singularities introduced by the method and how to avoid them. In addition, the complete and general solution is analyzed, allowing us to compute the pseudo-inverse matrix and the null space projector efficiently. We extend the RPJ decomposition concept [27] to describe the framework under which the symbolic inversion of a reduced Jacobian can be successfully derived. The 8-DoF AAI Arm is used as a case study, showing the efficiency of the proposed scheme.

The paper is organized as follows. In the next section, we present the notational convention and a summary of the RPJ decomposition method. In section 3, we formalize the reduced Jacobian approach, discussing algorithmic singularities and the general solution. Symbolic inversion and singularities are discussed in section 4 under the redundant spherical wrist constraint. The procedure which allows the efficient computation of the pseudo-inverse matrix and the direct application of several redundancy resolution schemes is discussed in section 5. We present in section 6 the application of the method to the 8–DoF AAI arm. Section 7 is a summary and conclusion. Two appendices are incorporated, giving details on the RPJ decomposition and the detailed singularity analysis of the AAI arm. The contribution of this paper, as outlined in the previous paragraph, is blended with contributions of several researchers, which are properly noted in the manuscript. We have chosen this format in order to provide a smooth and comprehensive evolution of the concepts discussed, friendly to readers who are not familiar with all the contributions in this particular area.

2 Preliminaries

In order to ease the understanding of the concepts being reviewed or presented in this article, we initially define the notation being used throughout the paper, followed by a summary of the Jacobian decomposition method referred to here as RPJ decomposition.

2.1 Notational Convention

1. **Vectors**

   Vector is a well known entity, but we are used to thinking of it in terms of its coordinates. Naturally, we need to represent it numerically, but its existence is independent of the representation. The abstract concept of coordinate-free vectors is used here in order to explore properties of vector algebra which are not coordinate dependent. For instance, the position vector of a point in space with respect to another does not depend on a particular representation frame. Its coordinates may vary depending on the representation frame, but the vector itself is invariant.

   We assign three attributes to a vector in the task space:

   - The description point, termed *the vector of* ...
   - The reference point, termed *with respect to* ...
   - The representation frame, termed *represented on* ...

   The reference point has traditionally been the origin of the representation frame, but we decouple those attributes in this paper. The following convention is adopted (unless otherwise stated):

   (a) A vector is written as a boldface lower case character.

   (b) The reference is noted as a right superscript.

   (c) The representation frame is noted as a left superscript.

   (d) The description and additional characteristics (sub-vectors, decompositions) are noted as right subscripts.

   When the representation frame is missing in the notation, we are considering a coordinate-free vector. When the reference is missing, we assume it to be the origin of the representation frame:

   $$ {}^{i}\boldsymbol{v}_{j} \triangleq {}^{i}\boldsymbol{v}_{j}^{i}. \tag{3} $$

   For instance, ${}^{i}\boldsymbol{p}_{j}^{k}$ stands for the position vector of j with respect to k, represented on frame i. The indices are usually numbers of the corresponding frames. The reference point is taken by default as the origin of the corresponding frame.

   The rules for the assignment of indices are extended to matrices.

2. **Matrices**

Upper case characters refer to matrices. A homogeneous transformation is indicated as ${}^iT_j$, referring to the transformation from frame j to i, i.e., the configuration of frame j represented on frame i. Furthermore,

$$ {}^iT_j \triangleq \begin{bmatrix} {}^i\boldsymbol{n}_j & {}^i\boldsymbol{o}_j & {}^i\boldsymbol{a}_j & {}^i\boldsymbol{p}_j \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4} $$

where $\boldsymbol{n}_j, \boldsymbol{o}_j, \boldsymbol{a}_j$, known as normal, orientation and approaching vectors, are the unit vectors of frame j. They are combined in a matrix called a rotation matrix:

$$ {}^iR_j \triangleq \begin{bmatrix} {}^i\boldsymbol{n}_j & {}^i\boldsymbol{o}_j & {}^i\boldsymbol{a}_j \end{bmatrix}. \tag{5} $$

3. **Space Variables**

Although homogeneous transformations are convenient for representing configurations in the task space, they are clearly redundant, since there are 12 scalars for a 6–DOF task space. We can use an alternative representation, as a 6-dimensional vector:

$$ {}^i\boldsymbol{x}_j \triangleq \begin{bmatrix} {}^i\boldsymbol{\psi}_j \\ {}^i\boldsymbol{p}_j \end{bmatrix}, \tag{6} $$

where $\boldsymbol{\psi}$ is the angular position vector. The transformation between the rotation matrix and the angular position vector is not explored in this paper. The interested reader is referred to [28]. Furthermore, the task velocity vector is given by:

$$ {}^i\dot{\boldsymbol{x}}_j = \begin{bmatrix} {}^i\boldsymbol{\omega}_j \\ {}^i\boldsymbol{v}_j \end{bmatrix}, \tag{7} $$

where $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{v} = \dot{\boldsymbol{p}}$ is the linear velocity.

Analogously, we define a vector for the configuration in the joint space. The dimension of a joint space is n, the number of joints:

$$ \boldsymbol{\theta} \triangleq \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}. \tag{8} $$

In this paper, only revolute joints are considered. The derivative of $\boldsymbol{\theta}$ is the joint velocity vector:

$$ \dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}. \tag{9} $$

2.2 RPJ Decomposition

Conceiving Jacobian as a vectorial matrix, we may select a convenient representation coordinate frame c and a convenient velocity reference point r in order to reduce the complexity of the Jacobian symbolic form. It may be interpreted as a generalization of the arm decomposition method.

A summary of the formulation for RPJ decomposition follows[1]:

$$
\begin{aligned}
{}^{b}\tilde{R}_c &= \begin{bmatrix} {}^{b}R_c & 0 \\ 0 & {}^{b}R_c \end{bmatrix}, \quad \tilde{P}_n^r = \begin{bmatrix} I & 0 \\ -\mathrm{PX}_n^r & I \end{bmatrix}, \\
{}^{c}J_r &= \begin{bmatrix} {}^{c}\boldsymbol{a}_0 & \cdots & {}^{c}\boldsymbol{a}_{i-1} & \cdots & {}^{c}\boldsymbol{a}_{n-1} \\ {}^{c}\boldsymbol{a}_0 \times {}^{c}\boldsymbol{p}_r^0 & \cdots & {}^{c}\boldsymbol{a}_{i-1} \times {}^{c}\boldsymbol{p}_r^{i-1} & \cdots & {}^{c}\boldsymbol{a}_{n-1} \times {}^{c}\boldsymbol{p}_r^{n-1} \end{bmatrix}, \\
{}^{0}J_n &= {}^{0}\tilde{R}_c \, {}^{c}\tilde{P}_n^r \, {}^{c}J_r.
\end{aligned}
\tag{10}
$$

The spherical wrist geometry reduces Jacobian complexity further and becomes an advantage in the task of singularity analysis and Jacobian inversion, as we will discuss in section 4.

The basic motivation for the choice of r and c is the minimization of Jacobian symbolic complexity. $c = 0$ and $r = n$ describe the original Jacobian: the velocity of the end-effector represented on the base (inertial) frame. However, the symbolic complexity is high. Having r and c somewhere in the middle of the serial structure, we can distribute the complexity among the R, P and J matrices. The new Jacobian matrix becomes very sparse, the middle columns trivial and both extreme columns much less complex when compared with the original Jacobian matrix. For a spherical wrist arm, further simplification is provided by the occurrence of a zero sub-matrix.

But what does RPJ decomposition represent? From the mathematical point of view, we have extracted some non-singular components from the original Jacobian, since ${}^{0}\tilde{R}_c$ and ${}^{c}\tilde{P}_n^r$ are intrinsically non-singular. The symbolic expressions of the new Jacobian elements become notably simpler for a wise choice of r and c, and yet no information for its analysis as a system is missing. Further details are found in Appendix A.

The principles behind RPJ decomposition have been applied in recent literature [26, 23]. We have adopted the notation suggested in [27] in order to explore the convenience of the formalism and the abstract concepts, as well as their physical meaning.

[1] PX is a skew-symmetric matrix, representing the cross product operator associated with the vector $\boldsymbol{p}$.

3 Reduced Jacobian

The reduced Jacobian method is based on decomposing the many-to-one Jacobian mapping into two complementary mappings, where one of the mappings is made to be bijective (i.e., one-to-one and onto) by reducing the dimension of the joint space to that of the task space.

Suppose our robot consists of n revolute joints ($n > 6$) and has a $6 \times n$ Jacobian matrix J. By selecting $n - 6$ joints as parameters, we can decompose J into two sub-matrices:

1. **Reduced Jacobian**, J_R. A 6×6 matrix obtained by removing the columns associated with the parameter joints from J.

2. **Parameter Jacobian**, J_P. A $6 \times (n-6)$ matrix composed of the columns associated with the parameter joints, as the complement of the reduced Jacobian.

Consequently, the following equations hold:

$$\begin{aligned} \dot{\boldsymbol{x}}_R &= J_R \dot{\boldsymbol{\theta}}_R, \\ \dot{\boldsymbol{x}}_P &= J_P \dot{\boldsymbol{\theta}}_P, \\ \dot{\boldsymbol{x}} &= \dot{\boldsymbol{x}}_R + \dot{\boldsymbol{x}}_P, \end{aligned} \tag{11}$$

where $\dot{\boldsymbol{\theta}}_R$ and $\dot{\boldsymbol{\theta}}_P$ represent the reduced joint sub-vector (non-parameter joints) and the parameter joint sub-vector respectively.

We introduce the pair column/row shuffle operation into the Jacobian equation in order to isolate the parameter joints:

$$\begin{aligned} J\mathcal{S} &= \left[\begin{array}{c:c} J_R & J_P \end{array} \right] \triangleq J_{\mathcal{S}} \\ \mathcal{S}^{-1}\dot{\boldsymbol{\theta}} &= \left[\begin{array}{c} \dot{\boldsymbol{\theta}}_R \\ \hdashline \dot{\boldsymbol{\theta}}_P \end{array} \right] \triangleq \dot{\boldsymbol{\theta}}_{\mathcal{S}} \\ \dot{\boldsymbol{x}} &= J\mathcal{S}\mathcal{S}^{-1}\dot{\boldsymbol{\theta}} = J_{\mathcal{S}}\dot{\boldsymbol{\theta}}_{\mathcal{S}} \end{aligned} \tag{12}$$

The shuffle operation matrix pair $\{\mathcal{S}, \mathcal{S}^{-1}\}$ may be weighted to account for uneven capabilities among the various joints and actuators [15, 21]. It may also be used to devise a minimum kinetic energy solution [2, 19].

The inverse solution can be represented by the following expression:

$$\dot{\boldsymbol{\theta}}_R = J_R^{-1}\left(\dot{\boldsymbol{x}} - J_P\dot{\boldsymbol{\theta}}_P\right). \tag{13}$$

Equation (13) indicates that a particular inverse Jacobian solution can be obtained from the task velocity $\dot{\boldsymbol{x}}$ by setting $\dot{\boldsymbol{\theta}}_P$ and computing J_R^{-1}. A general inverse Jacobian solution is the sum of a particular solution and the homogeneous solution in the Jacobian null space. The method for obtaining the homogeneous inverse Jacobian solution will be discussed in section 3.2.

Since equation (13) is independent of the representation and the reference point, we can apply RPJ decomposition to simplify the derivation of J_R^{-1} in a symbolic form.

3.1 Singularities

Two types of singularity can be defined in this approach. The first type is referred to as a system singularity. System singularities occur at particular joint configurations, where the arm can not generate motion in a certain task space direction. The system singularity is an intrinsic characteristic of the robot and should in general be avoided in order to maintain dexterity.

The second type is referred to as an algorithmic singularity. The algorithmic singularity is introduced by the choice of parameter joints for building a reduced Jacobian and is given by $\det J_R = 0$. When the reduced Jacobian becomes singular at a particular joint configuration, non-parameter joints alone are not capable of generating velocities in arbitrary directions of the task space. Notice that even though this type of singularity is introduced by the method of solution, it has a simple geometric interpretation, it is avoidable and, as opposed to the extended Jacobian method, it is independent of redundancy resolution criteria.

To avoid situations where no solution exists due to algorithmic singularities, we build a set of reduced Jacobian matrices $J_{R_i}, i = 1, \cdots, \ell$ using different choices for the parameter joints, such that any algorithmic singularity introduced by J_{R_i} can be covered by at least one of the remaining matrices $J_{R_j}, j \neq i$. Such a matrix always exists due to the fact that the Jacobian matrix has full rank [5].

The number ℓ of reduced Jacobian matrices and the choice of parameter joints for each J_{R_i} depend on the mechanical structure of the arm, as well as the given task. Nevertheless, we are providing general criteria for directing the selection, as in the previous paragraph.

We previously suggested that a system singularity should be avoided. But sometimes this is not possible. In this case, the inverse Jacobian in symbolic form is a significant advantage when compared with most numerical methods. When the arm is incapable of generating a certain task velocity component, we are still able to solve the system for the remaining components. This alternative is costly to achieve with numerical methods, as in resorting to singular value decomposition, but the solution is efficient when based on symbolic expressions.

3.2 Null Space

The manipulator redundancy introduces a powerful concept for system resolution and analysis. Redundancy implies that we can change the joint space configuration without changing the end-effector position and orientation through some coordinate motion of the joints. This phenomenon is also called internal motion or self-motion.

Null space is the subspace of the joint velocity space such that its velocities have no effect on the end-effector position and orientation [8].

We can rewrite equation (13) as follows:

$$\dot{\boldsymbol{\theta}}_{\mathcal{S}} = \dot{\boldsymbol{\theta}}_{\text{part}} + \dot{\boldsymbol{\theta}}_N, \tag{14}$$

where

$$\dot{\boldsymbol{\theta}}_{\text{part}} = \begin{bmatrix} J_R^{-1}\dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\theta}}_P = 0 \end{bmatrix} \tag{15}$$

is a particular solution for the joint velocity vector, with the parameter joint velocities set to

zero[2], and

$$\dot{\theta}_N = \begin{bmatrix} -J_R^{-1} J_P \dot{\theta}_P \\ \dot{\theta}_P \end{bmatrix} \tag{16}$$

is a null space joint velocity. Equation (16) has a simple, yet important physical interpretation. It shows that what parameter joints do in terms of end-effector displacement is undone or compensated by the reduced joints, thus preserving the end-effector position and orientation.

We have just outlined a procedure for generating null space velocities. Having an arm composed by n joints, the null space is in general of rank $n-6$. We select one of the parameter joints, set its velocity to a fixed level, say, one unit, and apply (16) to obtain the corresponding null space velocity vector. If we follow this procedure for every parameter joint, the result is a set of $n-6$ linearly independent vectors which span the null space of J [17]. The vectors are not orthonormal, but every null space velocity can be generated as a linear combination of them. In other words, a basis for the null space is obtained.

The task space velocity due to a parameter joint velocity set to one unit is numerically equal to the corresponding column of the Jacobian matrix. Therefore, the null space vector generated by a generic $\dot{\theta}_P$ is:

$$\dot{\theta}_N = N\dot{\theta}_P, \tag{17}$$

where

$$N \triangleq \begin{bmatrix} -J_R^{-1} J_P \\ I_P \end{bmatrix} \tag{18}$$

is a $n \times (n-6)$ matrix with its columns corresponding to a set of basis vectors for the null space. The presence of the identity sub-matrix I_P guarantees that the columns of N are linearly independent.

The matrix N could be symbolically evaluated, but there is little need for that, since only a simple matrix-vector multiplication is required and the elements have already been numerically evaluated for application in the particular solution. The algorithmic singularities associated with the null space generation method are due to the singularities of J_R, and have already been considered.

3.3 General Solution

A compact, yet complete and general formulation for the inverse Jacobian problem is discussed in this section.

The components of equation (14) can be combined into a single matrix transformation:

$$\dot{\theta}_S = \begin{bmatrix} J_R^{-1} & -J_R^{-1} J_P \\ 0 & I_P \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta}_P \end{bmatrix}. \tag{19}$$

Several aspects of equation (19) follow:

[2] In fact, we may set the parameter joint velocities to any finite value. We notice, however, that the resulting joint velocity incorporates only an extra null space shift. Therefore, zero velocities for the parameter joints do not affect the general solution and simplify the computation of the particular solution.

1. The solution is fully symbolic (therefore computationally efficient), provided that J_R^{-1} is available in symbolic form.

2. The solution is derived in the "shuffled" space. Returning to the original joint space is trivial and given by equation (12).

3. The $n \times n$ system has full rank, provided that the manipulator is not in algorithmic singularity, i.e. the reduced Jacobian is invertible.

4. The corresponding parameterized forward Jacobian mapping is given by:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\theta}}_P \end{bmatrix} = \begin{bmatrix} J_R & J_P \\ 0 & I_P \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\theta}}_R \\ \dot{\boldsymbol{\theta}}_P \end{bmatrix}. \tag{20}$$

5. The sub-vector $\dot{\boldsymbol{\theta}}_P$ spans the entire null space of the manipulator, which guarantees the completeness of the solution since for any particular solution there is one (and only one) $\dot{\boldsymbol{\theta}}_P$ such that equation (19) is satisfied.

6. A simplified algorithm can be devised for the selection of the $n-6$ parameters $\dot{\boldsymbol{\theta}}_P$ in order to speed-up the redundancy resolution task and ensure real time implementation.

3.4 Reduced Jacobian Selection

Although any choice for the parameter joints produces the same general solution, provided that the arm is not in algorithmic singularity with respect to that choice, the proximity of a singularity produces numerical instability, affecting the accuracy of the solution.

Given that we have a set of reduced Jacobian matrices to choose from, it is recommended to select the one which is the farthest of all from any singularity. For this purpose, a measure of distance to a singularity is desired. Two measures have been explored in the literature, namely the condition number (the ratio between the largest and the smallest singular value) [33] and the manipulability [10].

The manipulability is in general defined as $w = (\det(JJ^T))^{\frac{1}{2}}$, but since we are considering a square matrix — the reduced Jacobian — the manipulability is simplified to:

$$w_{R_i} = |\det(J_{R_i})|. \tag{21}$$

We adopt the use of the manipulability measure for computational efficiency, since it requires only the evaluation of the determinant of the reduced Jacobian matrices, which are available in symbolic form. The algorithmic singularities associated with a reduced Jacobian are given by $\det(J_R) = 0$. A reduced Jacobian with a small manipulability is expected to be dangerously near a singularity. Notice that the manipulability is used here as a relative measure, since the selection is based on the comparison of determinants.

The reduced Jacobian with the largest determinant in absolute value is not necessarily the farthest from any singularity in the Euclidean sense, nor it leads to the best conditioned matrix, but the objective of preventing numerical instability is accomplished, justifying our choice. Notice that we are considering only a subset of all possible reduced joint combinations, those for which the symbolic inverse is available. As we will discuss shortly, there is no need for the symbolic inversion of all possible reduced Jacobian matrices. We may consider only a suitable set, designed to prevent undesirable algorithmic singularities.

4 Symbolic Inversion

The reduced Jacobian approach is general and can be applied to any redundant manipulator. Symbolic inversion of a reduced Jacobian can be successfully accomplished when the reduced Jacobian matrix is sparse and has symbolic elements with a simple form. However, the effort for the derivation of inverse Jacobian in symbolic form is significantly reduced if the manipulator holds the spherical wrist property.

4.1 The Redundant Spherical Wrist

Spherical wrist is a geometric characteristic common to a class of serial robot arms [7]. A non-redundant spherical wrist can be characterized by the following equivalent properties:

- **Geometry**: the last three joint axis intersect at the same point.
- **D–H parameters** [1]: $a_{n-2} = a_{n-1} = a_n = 0$ and $d_{n-1} = 0$.

We use the term "wrist" because it corresponds to the terminal joints of the serial structure. The term "spherical" comes from the fact that any orientation in the 3-dimensional space can be obtained by moving the wrist joints, without affecting the position of the end-effector. It is assumed that the twist angles are properly designed to provide mechanical means for covering the entire orientation range.

The main property of this class of manipulators is that the origin of frame $n-2$ can be computed in terms of the end-effector task configuration alone:

$$\boldsymbol{p}_{n-2} = \boldsymbol{p}_n - d_n \boldsymbol{a}_n. \tag{22}$$

As a consequence, we can mathematically decouple the derivation of position from orientation of the end-effector.

Next, we extend the definition above to include redundancy. The redundant spherical wrist is thus defined as a set of u terminal intersecting joint axis, for $u > 3$. The corresponding D–H parameters are described in Table I. The origin of the lowest frame located at the wrist intersecting point is $\boldsymbol{p}_{n-u+1}$ and the property given by equation (22) still holds for this point.

Table I: Denavit-Hartenberg parameters for a redundant spherical wrist arm.

link i	length a_i	distance d_i
$n-u+1$	0	d_{n-u+1}
$n-u+2$	0	0
$\vdots$	$\vdots$	$\vdots$
$n-1$	0	0
n	0	d_n

There are equivalent properties in the velocity domain. A wise choice for r and c in the RPJ decomposition produces a considerable simplification of singularity analysis and inverse Jacobian derivation. In fact,

$$c = n - u \tag{23}$$

and

$$r = c + 1 \tag{24}$$

produce a corresponding Jacobian matrix in the following form:

$${}^{c}J_r = \begin{bmatrix} A & B \\ D & 0 \end{bmatrix}, \tag{25}$$

where $B, 0$ (a zero matrix) are $3 \times u$ sub-matrices and A, D are $3 \times (n-u)$ sub-matrices.

The spherical wrist joints define the upper arm and the remaining joints define the lower arm.

4.2 Reduced Jacobian Set

For a simple symbolic inversion, it is desirable to have a reduced Jacobian maintaining the spherical wrist geometry. We can divide the entire Jacobian into two sub-matrices: one consisting of the u columns associated with the spherical wrist, J_U, and the other consisting of the remaining $n-u$ columns, J_L.

By removing $u-3$ columns from J_U, we reduce the upper part to a non-redundant spherical wrist. Therefore, any combination of 3 columns from J_U and 3 columns from J_L is a spherical wrist reduced Jacobian and the symbolic inversion method discussed in Appendix A can be applied.

Regarding algorithmic singularities, suppose that we are able to select two different subsets of parameter joints from J_U, such that the corresponding B matrices are not simultaneously singular. Analogously, for two subsets of parameter joints from J_L, the corresponding D matrices are not simultaneously singular. By doing so, we have guaranteed that 4 reduced Jacobian matrices are never simultaneously (algorithmically) singular when we define them as the combinations of one subset of J_U with one subset of J_L. In this case, the only effort is to invert four (usually sparse) 3×3 matrices symbolically.

4.3 System Singularities

The evaluation of $\det(D)$ for all possible combinations of parameter joints from J_L provides an important result. Since no upper joint is capable of covering positioning singularities of the lower part due to the spherical wrist geometry, every singular condition common to all those combinations is a system singularity. In fact, at least one linear velocity in a particular direction can not be generated. Using the symbolic form, we can explicitly obtain that direction. This method enables us to derive singular configurations of a redundant manipulator analytically; an important property in itself.

Unfortunately, this result can not be fully extended to the upper part, because the loss of rank in the orientation of the end-effector due to the upper part singularity can be potentially covered by the positional null space of the lower part. However, the system singularity in terms of angular velocities has the algorithmic singularity of B as a partial condition. In other words, the singularity of the upper part, together with a condition where the lower part loses its null space capability of covering the loss of rank in the upper part is also a system singularity. Again, using the symbolic form, we can explicitly obtain that condition.

5.1 Redundancy Resolution

The reduced Jacobian method does not embed any algorithm for redundancy resolution. It rather provides an efficient and general solution for the Jacobian inversion of a redundant manipulator, which can be used to emulate the pseudo-inverse-based general solution. Any performance criterion aiming at finding the optimal solution can be implemented as a higher level layer. Among possible criteria we mention manipulability, obstacle avoidance, singularity avoidance, joint limits avoidance, power consumption minimization, fault tolerance [36] and task-priority [16].

We may conceive the performance as a collection of desirable attributes and associate an artificial potential function in the n-dimensional joint configuration space to each attribute. The integration of the attributes form a combined criterion (H) and is accomplished by the superposition of all potential functions [35]. The optimization is obtained by projecting the gradient of the potential function onto the null space. We realize, however, that this method gives us only the direction for optimization. The magnitude depends on the capabilities of the arm and stability considerations [31]. At this point, the advantage of decoupling Jacobian inversion from redundancy resolution becomes apparent. The magnitude of the null space component can be adjusted without the need for reiterating the Jacobian inversion process. In addition, the performance criterion itself can be modified independently and on-line.

For instance, let us suppose that the shuffle matrix $\mathcal{S}$ in equation (12) is weighted by the corresponding joint velocity limits, such that the joint velocity limits in the shuffled space are uniform, represented by a n-dimensional cube centered at the origin. We can define a conservative limit by inscribing a n-dimensional sphere in the cube and the optimal solution will be the intersection of the locus of the optimal solution and the sphere, as shown in figure 2. We can easily explore the property given by equation (32) and devise the following redundancy resolution scheme:

$$\begin{aligned} \boldsymbol{k} &= \hat{N}^T (\nabla_{\mathcal{S}} H) \\ \rho^2 &= \|\dot{\boldsymbol{\theta}}^+\|^2 + \alpha^2 \|\boldsymbol{k}\|^2 \implies \alpha = \left(\frac{\rho^2 - \|\dot{\boldsymbol{\theta}}^+\|^2}{\|\boldsymbol{k}\|^2} \right)^{\frac{1}{2}} \\ \dot{\boldsymbol{\theta}}_{\text{opt}} &= \dot{\boldsymbol{\theta}}^+ + \hat{N}(\alpha \boldsymbol{k}). \end{aligned} \tag{33}$$

The scheme above will produce a constant norm of the uniform joint velocity vector throughout the trajectory. Alternatively, we can intersect the locus of the optimal solution with the n-dimensional cube itself, generating the following redundancy resolution scheme:

$$\begin{aligned} \dot{\boldsymbol{\theta}}_N &= \hat{N} \boldsymbol{k} \\ \alpha &= \min_i \left(\frac{\text{sgn}(\dot{\theta}_{Ni}) \rho - \dot{\theta}_i^+}{\dot{\theta}_{Ni}} \right) \\ \dot{\boldsymbol{\theta}}_{\text{opt}} &= \dot{\boldsymbol{\theta}}^+ + \alpha \, \dot{\boldsymbol{\theta}}_N. \end{aligned} \tag{34}$$

The redundancy resolution procedures described above are representatives of efficient schemes for redundancy resolution which explore properties of the reduced Jacobian method discussed in this paper.

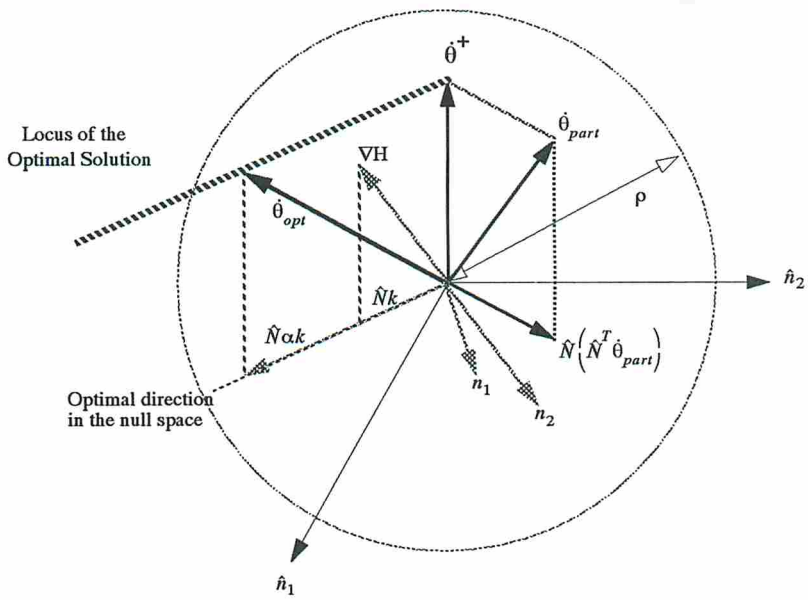

Figure 2: Velocity-based solution optimization using the reduced Jacobian approach

Figure 2 shows the geometric interpretation of the contribution of several elements in the optimization procedure (33) based on the reduced Jacobian method.

$$\theta = [\ 0\ 135\ 180\ 135\ 90\ -90\ 180\ ]^T$$

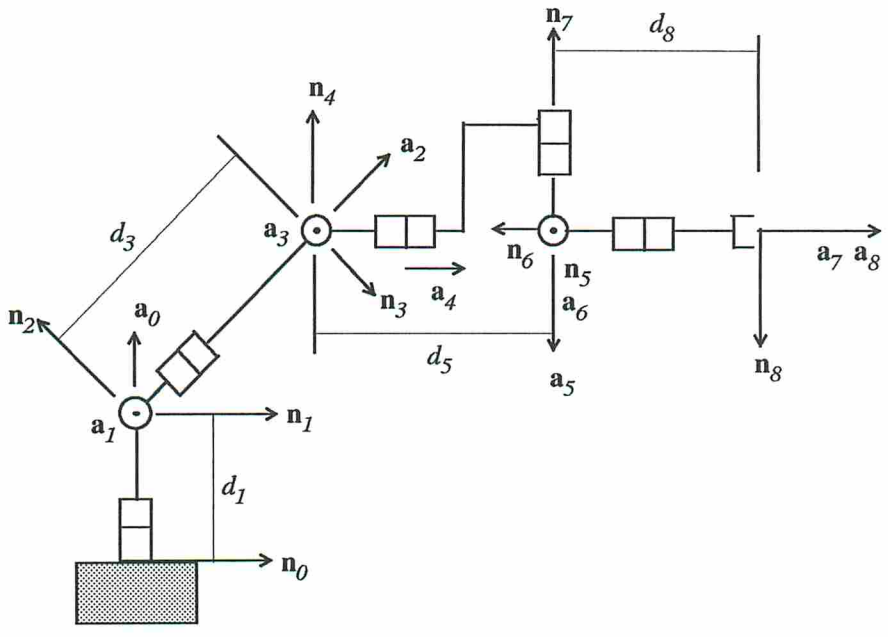

Figure 3: The AAI arm structure.

6 Case Study: the AAI Arm

The reduced Jacobian method discussed in the previous sections has been applied to a representative of the redundant spherical wrist arm class: the 8–DoF AAI arm.

We could present a simplified and academic example, such as a planar robot, to facilitate comprehension. However, it is known that complexity grows exponentially with the dimension of the system. Therefore, in order to show the power of the method, we present a more realistic application.

6.1 Arm Structure and Forward Jacobian

The AAI arm is a redundant manipulator, assembled as a set of 8 revolute joints serially connected. It is fully described in geometric terms by the *Denavit-Hartenberg* parameters [1] (classical notation) shown in Table II.

A closer look in Table II or the arm structure itself (Figure 3) reveals that the last 4 joint axes are intersecting and therefore the AAI arm contains a redundant spherical wrist, with $u = 4$.

The symbolic form of the optimal RPJ decomposition, according to the choice for the rep-

Table II: Denavit-Hartenberg parameters for the AAI Arm.

link i	length a_i	distance d_i	twist angle α_i	joint angle θ_i
1	0	d_1	90^o	θ_1
2	0	0	90^o	θ_2
3	0	d_3	90^o	θ_3
4	0	0	90^o	θ_4
5	0	d_5	-90^o	θ_5
6	0	0	90^o	θ_6
7	0	0	90^o	θ_7
8	0	d_8	0^o	θ_8

resentation frame and reference point given by equations (23) and (24), is the following:

$$
{}^0R_4 = \begin{bmatrix} (C_1C_2C_3 + S_1S_3)C_4 + C_1S_2S_4 & C_1C_2S_3 - S_1C_3 & (C_1C_2C_3 + S_1S_3)S_4 - C_1S_2C_4 \\ (S_1C_2C_3 - C_1S_3)C_4 + S_1S_2S_4 & S_1C_2S_3 + C_1C_3 & (S_1C_2C_3 - C_1S_3)S_4 - S_1S_2C_4 \\ S_2C_3C_4 - C_2S_4 & S_2S_3 & S_2C_3S_4 + C_2C_4 \end{bmatrix},
$$

$$
{}^4\boldsymbol{p}_8^5 = \begin{bmatrix} d_8(C_5C_6S_7 + S_5C_7) \\ d_8(S_5C_6S_7 - C_5C_7) \\ -d_8S_6S_7 \end{bmatrix}, \tag{35}
$$

$$
{}^4J_5 = \begin{bmatrix} C_4C_3S_2 - S_4C_2 & C_4S_3 & S_4 & 0 & 0 & -S_5 & C_5S_6 & C_5C_6S_7 + S_5C_7 \\ S_3S_2 & -C_3 & 0 & 1 & 0 & C_5 & S_5S_6 & S_5C_6S_7 - C_5C_7 \\ S_4C_3S_2 + C_4C_2 & S_4S_3 & -C_4 & 0 & 1 & 0 & C_6 & -S_6S_7 \\ -(d_3C_4 - d_5)S_3S_2 & (d_3C_4 - d_5)C_3 & 0 & d_5 & 0 & 0 & 0 & 0 \\ (d_3 - d_5C_4)C_3S_2 + d_5S_4C_2 & (d_3 - d_5C_4)S_3 & -d_5 & 0 & 0 & 0 & 0 & 0 \\ -d_3S_4S_3S_2 & d_3S_4C_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
$$

The symbolic solution for the elements of ${}^0J_8$ is not required. ${}^0J_8$ can be numerically evaluated for every joint space configuration by:

$$
{}^0J_8 = {}^0\tilde{R}_4 \; {}^4\tilde{P}_8^5 \; {}^4J_5. \tag{36}
$$

Nevertheless, if one wishes to derive ${}^0J_8$ in symbolic form, equation (36) can be symbolically evaluated as well.

For the sake of comparison, we show a single element of ${}^0J_8$ in symbolic form:

$$
\begin{aligned}
\tfrac{\partial x_4}{\partial \theta_1} = \; & d_8((C_7(C_5(C_1C_3 + C_2S_1S_3) - (C_4(C_2C_3S_1 - C_1S_3) + C_1S_2S_4)S_5)) \\
& -(C_6(C_5(C_4(C_2C_3S_1 - C_1S_3) + S_1S_2S_4) + (C_1C_3 + C_2S_1S_3)S_5) \\
& -(-C_4S_1S_2 + (C_2C_3S_1 - C_1S_3)S_4)S_6)S_7) \\
& +d_5(C_4S_1S_2 - (C_2C_3S_1 - C_1S_3)S_4) - d_3S_1S_2 \, ,
\end{aligned} \tag{37}
$$

while the corresponding term in ${}^4J_5$ is simply $-(d_3C_4 - d_5)S_3S_2$.

6.2 Reduced Jacobian Set

A reduced Jacobian for the AAI arm requires that 2 joints be parameterized. In order to maintain the spherical wrist geometry, we must parameterize one joint of the lower part and one joint of the upper part for every reduced Jacobian. Furthermore, in order to avoid algorithmic singularities, we need at least 2 subsets from the lower part and 2 subsets from the upper part.

In a previous work on the AAI arm, inverse kinematics was derived using the same concept of parameterizing joints [40] . At that time, a reasonable choice was to select either 1 or 3 for the lower part and either 5 or 6 for the spherical (upper) part. For consistency, we will preserve that selection.

Based on that selection, we have the following reduced Jacobian Set:

$$J_{15} = \begin{bmatrix} A_1 & B_5 \\ D_1 & 0 \end{bmatrix}, \quad J_{16} = \begin{bmatrix} A_1 & B_6 \\ D_1 & 0 \end{bmatrix},$$
$$J_{35} = \begin{bmatrix} A_3 & B_5 \\ D_3 & 0 \end{bmatrix}, \quad J_{36} = \begin{bmatrix} A_3 & B_6 \\ D_3 & 0 \end{bmatrix}. \tag{38}$$

The indices refer to the corresponding parameter joints.

Furthermore:

$$\begin{aligned} \det(D_1) &= d_5^2 d_3 S_4 C_3, \\ \det(D_3) &= d_5 d_3 S_4 \left[(d_3 - d_5 C_4) S_2 + d_5 S_4 C_3 C_2 \right], \\ \det(B_5) &= S_7, \\ \det(B_6) &= -S_6 C_7. \end{aligned} \tag{39}$$

6.3 Symbolic Inversion

Given all sub-matrices of the selected Reduced Jacobian Set in symbolic form, we derive the required inversions. Skipping the algebraic manipulation, the results are as follows:

$$D_1^{-1} = \begin{bmatrix} 0 & 0 & \frac{1}{d_3 S_4 C_3} \\ 0 & -\frac{1}{d_5} & \frac{(d_3 - d_5 C_4) S_3}{d_3 d_5 S_4 C_3} \\ \frac{1}{d_5} & 0 & -\frac{d_3 C_4 - d_5}{d_3 d_5 S_4} \end{bmatrix} \tag{40}$$

$$D_3^{-1} = \begin{bmatrix} 0 & \frac{C_3}{Q} & -\frac{(d_3 - d_5 C_4) S_3}{d_3 S_4 Q} \\ 0 & \frac{S_3 S_2}{Q} & \frac{(d_3 - d_5 C_4) C_3 S_2 + d_5 S_4 C_2}{d_3 S_4 Q} \\ \frac{1}{d_5} & 0 & -\frac{d_3 C_4 - d_5}{d_3 d_5 S_4} \end{bmatrix}, \quad Q = (d_3 - d_5 C_4) S_2 + d_5 S_4 C_3 C_2 \tag{41}$$

$$B_5^{-1} = \begin{bmatrix} \frac{C_5C_6C_7 - S_5S_7}{S_7} & \frac{S_5C_6C_7 + C_5S_7}{S_7} & -\frac{S_6C_7}{S_7} \\ C_5S_6 & S_5S_6 & C_6 \\ \frac{C_5C_6}{S_7} & \frac{S_5C_6}{S_7} & -\frac{S_6}{S_7} \end{bmatrix} \tag{42}$$

$$B_6^{-1} = \begin{bmatrix} -\frac{C_5C_6C_7 - S_5S_7}{S_6C_7} & -\frac{S_5C_6C_7 + C_5S_7}{S_6C_7} & 1 \\ -\frac{S_5C_6S_7 - C_5C_7}{S_6C_7} & \frac{C_5C_6S_7 + S_5C_7}{S_6C_7} & 0 \\ \frac{S_5}{C_7} & -\frac{C_5}{C_7} & 0 \end{bmatrix}. \tag{43}$$

The following system singularities are obtained based on the symbolic form of the reduced Jacobian matrices:

1. $S_4 = 0$
2. $S_2 = C_3 = 0$
3. $C_5 = S_6 = S_7 = 0$
4. $S_2 = S_6 = S_7 = 0$

Refer to Appendix B for a thorough singularity analysis.

6.4 A Numerical Example

In order to show how the general solution is obtained and how redundancy can be solved based on a sample criterion, we present a numerical example:

1. Denavit-Hartenberg Parameters:
   $d_1 = 0.30\,\text{m}$, $d_3 = 1.00\,\text{m}$, $d_5 = 0.65\,\text{m}$, $d_8 = 0.20\,\text{m}$.
2. Current Configuration:
   $$\boldsymbol{\theta} = \begin{bmatrix} 90 & 170 & 80 & 45 & 0 & 10 & 10 & 0 \end{bmatrix}^T \text{(deg)}. \tag{44}$$
   The current configuration is near a double interior singularity, given by:
   $$\boldsymbol{\theta}_{\text{sing}} = \begin{bmatrix} 90 & 180 & 90 & 45 & 0 & 0 & 0 & 0 \end{bmatrix}^T \text{(deg)}. \tag{45}$$
3. Our goal is to cross the end-effector task space configuration corresponding to (45) after a short time. One possible path is to deliver the following constant joint velocity vector:
   $$\dot{\boldsymbol{\theta}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 \end{bmatrix}^T \text{(rad/s)}. \tag{46}$$

4. The Jacobian matrix for the current configuration is:

$$ {}^0R_4 = \begin{bmatrix} 0.70 & -0.17 & 0.70 \\ 0.00 & -0.97 & -0.24 \\ 0.72 & 0.17 & -0.68 \end{bmatrix}, \quad {}^4\boldsymbol{p}_8^5 = \begin{bmatrix} 0.03 \\ -0.20 \\ -0.01 \end{bmatrix}, $$

$$ {}^4J_5 = \begin{bmatrix} -0.72 & 0.70 & 0.71 & 0 & 0 & 0 & 0.17 & 0.17 \\ 0.17 & -0.17 & 0 & 1 & 0 & 1 & 0 & -0.98 \\ -0.68 & 0.70 & -0.71 & 0 & 1 & 0 & 0.98 & -0.03 \\ -0.01 & 0.01 & 0 & 0.65 & 0 & 0 & 0 & 0 \\ -0.44 & 0.53 & -0.65 & 0 & 0 & 0 & 0 & 0 \\ -0.12 & 0.12 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{47} $$

5. The task space velocity due to equation (46) is:

$$ {}^4\dot{\boldsymbol{x}}_5^0 = {}^4J_5\dot{\boldsymbol{\theta}} = \begin{bmatrix} 1.23 & -1.17 & -1.00 & 0.01 & -0.12 & 0.12 \end{bmatrix}^T, $$

$$ {}^4\dot{\boldsymbol{x}}_8^0 = {}^4\tilde{P}_8^5\,{}^4\dot{\boldsymbol{x}}_5^0 = \begin{bmatrix} 1.23 & -1.17 & -1.00 & -0.19 & -0.16 & -0.08 \end{bmatrix}^T, \tag{48} $$

$$ {}^0\dot{\boldsymbol{x}}_8 = {}^0\tilde{R}_4\,{}^4\dot{\mathbf{x}}_8^0 = \begin{bmatrix} 0.37 & 1.38 & 1.35 & -0.16 & 0.17 & -0.11 \end{bmatrix}^T. $$

The previous steps correspond to the forward Jacobian task using RPJ decomposition. We used those preliminaries in order to determine an end-effector velocity for which a numerically simple solution moves the arm towards a singularity. Now, using the task space velocity as input, we apply inverse Jacobian according to the reduced Jacobian method.

6. Initially, we evaluate the determinant of the reduced Jacobian matrices from the set (38), producing the following results:

$$ \begin{aligned} &\det(J_{15}) = 13.86 \times 10^{-3}, \quad &&\det(J_{16}) = -13.65 \times 10^{-3}, \\ &\det(J_{35}) = 1.22 \times 10^{-3}, \quad &&\det(J_{36}) = 1.20 \times 10^{-3}. \end{aligned} \tag{49} $$

The results above reveal that J_{15} is the best J_R for the given configuration, according to the manipulability maximization criterion (section 3.4).

7. For simplicity, we consider the joints to be balanced and no weighting for the contribution of each joint is considered. The shuffle matrix for the given reduced Jacobian selection becomes:

$$ \mathcal{S} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{50} $$

and $\mathcal{S}^{-1} = \mathcal{S}^T$.

8. The numerical evaluation of the symbolic inversions (40) and (42) yields:

$$D_1^{-1} = \begin{bmatrix} 0 & 0 & 8.14 \\ 0 & -1.54 & 6.67 \\ 1.54 & 0 & -0.12 \end{bmatrix}, \quad B_5^{-1} = \begin{bmatrix} 5.59 & 1 & -0.98 \\ 0.17 & 0 & 0.98 \\ 5.67 & 0 & -1 \end{bmatrix} \tag{51}$$

9. The particular solution obtained using the efficient form from equation (102) becomes:

$$\dot{\boldsymbol{\theta}}_{\text{part}} = \begin{bmatrix} J_{15}^{-1}{}^4\dot{\boldsymbol{x}}_5^0 \\ \dot{\boldsymbol{\theta}}_{15} = 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}^T. \tag{52}$$

Notice that, as expected, $\dot{\boldsymbol{\theta}}_{\text{part}}$ resulted in the joint space velocity given by (46), since $\dot{\theta}_1 = \dot{\theta}_5 = 0$ as a precondition for the particular solution given by (52). Notice that equation (52) is represented in the shuffled space and, as such, the parameter joints 1 and 5 correspond to coordinates 7 and 8 respectively. For clarity, we will proceed the analysis entirely in the shuffled space:

$$\dot{\boldsymbol{\theta}}_{\mathcal{S}} = \begin{bmatrix} \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_6 & \dot{\theta}_7 & \dot{\theta}_8 & \dot{\theta}_1 & \dot{\theta}_5 \end{bmatrix}^T \tag{53}$$

10. Next, we compute the null space vectors for the given configuration. Considering J_{15}, the application of definition (18) yields:

$$N = \begin{bmatrix} 0.98 & 0 \\ 0.13 & 0 \\ 0 & 0 \\ -0.44 & 0.98 \\ 0.07 & -0.98 \\ -0.44 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{54}$$

11. The general solution according to equation (14) becomes:

$$\dot{\boldsymbol{\theta}}_{\mathcal{S}} = \dot{\boldsymbol{\theta}}_{\text{part}} + N\dot{\boldsymbol{\theta}}_P = \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.98 & 0 \\ 0.13 & 0 \\ 0 & 0 \\ -0.44 & 0.98 \\ 0.07 & -0.98 \\ -0.44 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_5 \end{bmatrix}. \tag{55}$$

12. We now obtain the ortho-normalized basis for the null space applying the Gram-Schmidt procedure with two vectors:

$$\hat{\boldsymbol{n}}_2 = \frac{\boldsymbol{n}_2}{\|\boldsymbol{n}_2\|} \quad , \quad \boldsymbol{n}_1^{\perp} = \boldsymbol{n}_1 - \left(\boldsymbol{n}_1^T \hat{\boldsymbol{n}}_2\right) \hat{\boldsymbol{n}}_2 \quad , \quad \hat{\boldsymbol{n}}_1 = \frac{\boldsymbol{n}_1^{\perp}}{\|\boldsymbol{n}_1^{\perp}\|}, \tag{56}$$

$$\hat{N} = \left[\begin{array}{c:c} \hat{\boldsymbol{n}}_1 & \hat{\boldsymbol{n}}_2 \end{array} \right] = \begin{bmatrix} 0.67 & 0 \\ 0.09 & 0 \\ 0 & 0 \\ -0.14 & 0.49 \\ -0.11 & -0.49 \\ -0.14 & 0.51 \\ 0.68 & 0 \\ 0.17 & 0.51 \end{bmatrix}. \tag{57}$$

13. The minimum norm solution based on equation (31) is given by:

$$\dot{\boldsymbol{\theta}}^{+} = \left(I - \hat{N}\hat{N}^T\right)\dot{\boldsymbol{\theta}}_{\text{part}} = \begin{bmatrix} 0.32 \\ 0.91 \\ 0 \\ -0.86 \\ -0.89 \\ 0.14 \\ -0.69 \\ -0.17 \end{bmatrix}. \tag{58}$$

14. The general solution according to equation (30) becomes:

$$\dot{\boldsymbol{\theta}}_{\mathcal{S}} = \begin{bmatrix} 0.32 \\ 0.91 \\ 0 \\ -0.86 \\ -0.89 \\ 0.14 \\ -0.69 \\ -0.17 \end{bmatrix} + \begin{bmatrix} 0.67 & 0 \\ 0.09 & 0 \\ 0 & 0 \\ -0.14 & 0.49 \\ -0.11 & -0.49 \\ -0.14 & 0.51 \\ 0.68 & 0 \\ 0.17 & 0.51 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}. \tag{59}$$

15. In this stage we go one step further, addressing the redundancy resolution issue. As a sample criterion for redundancy resolution, we consider avoidance of the singularity given by equation (45). Since equation (46) expresses the direction pointing to the singularity, we trivially conclude that the direction which avoids that singularity is the very opposite of (46). Therefore, in the shuffled space we have:

$$\nabla_{\mathcal{S}} H = -\begin{bmatrix} 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}^T. \tag{60}$$

The projection of (60) onto the null space yields:

$$\boldsymbol{k} = \hat{N}^T(\nabla_{\mathcal{S}} H) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \tag{61}$$

Assuming that the maximum allowable velocity is $\rho = 3\,\text{rad/s}$, based on the scheme (33) we have

$$\alpha = \left(\frac{\rho^2 - \|\dot{\boldsymbol{\theta}}^{+}\|^2}{\|\boldsymbol{k}\|^2}\right)^{\frac{1}{2}} = \left(\frac{9 - 2.99}{1}\right)^{\frac{1}{2}} = 2.45 \tag{62}$$

and the optimal solution becomes

$$\dot{\boldsymbol{\theta}}_{\text{opt}} = \dot{\boldsymbol{\theta}}^{+} + \hat{N}(\alpha \boldsymbol{k}) = \begin{bmatrix} -1.32 \\ 0.69 \\ 0 \\ -0.52 \\ -0.62 \\ 0.48 \\ -2.36 \\ -0.59 \end{bmatrix} \text{ (rad/s)}. \tag{63}$$

Based on the scheme (34), with the same maximum allowable velocity of $\rho = 3\,\text{rad/s}$, we have:

$$\dot{\boldsymbol{\theta}}_N = \hat{N}\boldsymbol{k} = \begin{bmatrix} -0.67 & -0.09 & 0 & 0.14 & 0.11 & 0.14 & -0.68 & -0.17 \end{bmatrix}^T \tag{64}$$

$$\alpha = \min_i \left(\frac{\text{sgn}(\dot{\theta}_{Ni})\rho - \dot{\theta}_i^{+}}{\dot{\theta}_{Ni}} \right) = 3.40 \tag{65}$$

$$\dot{\boldsymbol{\theta}}_{\text{opt}} = \dot{\boldsymbol{\theta}}^{+} + \alpha\, \dot{\boldsymbol{\theta}}_N = \begin{bmatrix} -1.96 \\ 0.60 \\ 0 \\ -0.38 \\ -0.52 \\ 0.62 \\ -3.00 \\ -0.75 \end{bmatrix}. \tag{66}$$

6.5 Computational Cost

In this section, we present an evaluation of the computational cost in terms of the number of operations associated with the inverse Jacobian process. The sequence is obtained from the example in the previous section. The cost is customized for the given arm, but a general configuration is assumed.

For simplicity, we consider two types of operations:

1. a: addition or subtraction;
2. m: multiplication, division or square root.

Table III summarizes the cost, step by step, given the input ${}^0\dot{\boldsymbol{x}}_8$. It is assumed that forward kinematics has been previously computed in such a way that the rotation matrix ${}^0R_4$ and the approaching and position vectors $\boldsymbol{a}$ and $\boldsymbol{p}$ from equation (10) are intermediate results and stored for the current stage.

The total cost for the parameterized solution (55) becomes:

$$94m + 51a. \tag{67}$$

The minimum norm solution accompanied by an ortho-normalized basis for the null space has the cost given by Table IV.

Table III: Computational cost for the parameterized solution.

task	: cost
${}^4\dot{\boldsymbol{x}}_8^0 = {}^4\tilde{R}_0\, {}^0\dot{\boldsymbol{x}}_8$	: $18m + 12a$
${}^4\dot{\boldsymbol{x}}_5^0 = {}^4\tilde{P}_5^8\, {}^4\dot{\boldsymbol{x}}_8^0$	: $6m + 6a$
${}^4J_5$	: $8m + 2a$
$\det(J_R)$	: 6m + a
D_1^{-1}	: $6m$
B_5^{-1}	: $12m + 2a$
$\dot{\boldsymbol{\theta}}_L = D_1^{-1}\, {}^4\boldsymbol{v}_5^0$	: $5m + 2a$
$\dot{\boldsymbol{\theta}}_U = B_5^{-1}({}^4\boldsymbol{\omega}_5^0 - A_1\dot{\boldsymbol{\theta}}_L)$	: $14m + 12a$
N	: $19m + 14a$

Table IV: Computational cost for the ortho-normalization.

task	: cost
$\hat{N}$	: $29m + 14a$
$\dot{\boldsymbol{\theta}}^+$	: $18m + 14a$

The total cost for the efficient computation of the ortho-normalized solution given by equation (59) becomes:

$$141m + 79a. \tag{68}$$

Finally, the application of the redundancy resolution scheme (33) has the additional cost given by Table V, assuming that the gradient is given.

Table V: Computational cost for the first redundancy resolution scheme.

task	: cost
$\boldsymbol{k} = \hat{N}^T(\nabla_S H)$	: $11m + 9a$
$\alpha = \left(\frac{\rho^2 - \|\dot{\boldsymbol{\theta}}^+\|^2}{\|\boldsymbol{k}\|^2}\right)^{\frac{1}{2}}$	: $11m + 8a$
$\dot{\boldsymbol{\theta}}_{\text{opt}} = \dot{\boldsymbol{\theta}}^+ + \hat{N}(\alpha\boldsymbol{k})$	: $13m + 11a$

Analogously, the application of the redundancy resolution scheme (34) has the additional cost given by Table VI.

The overall cost of obtaining the optimal solution (63) becomes:

$$176m + 107a. \tag{69}$$

Table VI: Computational cost for the second redundancy resolution scheme.

task	: cost
$\dot{\boldsymbol{\theta}}_N = \hat{N}\hat{N}^T(\nabla_S H)$	$: 22m + 13a$
$\alpha = \min_i \left(\frac{\text{sgn}(\dot{\theta}_{Ni}) - \dot{\theta}_i^+}{\dot{\theta}_{Ni}} \right)$	$: 7m + 7a$
$\dot{\boldsymbol{\theta}}_{\text{opt}} = \dot{\boldsymbol{\theta}}^+ + \alpha \dot{\boldsymbol{\theta}}_N$	$: 7m + 7a$

Alternatively, the overall cost of obtaining the optimal solution (66) becomes:

$$177m + 106a. \tag{70}$$

Depending on the factorization, the reduced Jacobian selection, and which intermediate results are stored, the number of operations might increase or decrease by a few units.

The goal of this section is to show the computational efficiency of the method, since numerical algorithms such as the pseudo-inverse-based general solution (2) have their cost estimated in several thousands of operations for an arm of this complexity [17].

As a final remark in this section, we point out that the computational cost for the parameterized solution (55), given by (67), corresponds to only approximately half of the overall computational cost described above. As we mentioned in section 3.3, a redundancy resolution algorithm which determines directly the joint parameter velocities with low computational cost based on some practical criterion can be designed to reduce even further the overall cost. In other words, a compromise between computational speed and the complexity of the redundancy resolution criterion may be reached at this level.

7 Summary and Conclusion

The reduced Jacobian method is based on selecting a full rank square sub-matrix of the redundant Jacobian and deriving its symbolic inverse. The remaining columns are used to generate a basis for the null space, producing the complete solution. The choice of the sub-matrix introduces algorithmic singularities, which are analyzed to produce a method of avoidance. The exploration of linear system properties leads to the efficient computation of the pseudo-inverse matrix and the null space projector operator. Further details on reduced Jacobian inversion and singularity analysis are given under the spherical wrist constraint. The method is applied to the 8–DoF AAI Arm, showing the computational simplicity and efficiency.

Our main objective is to formalize the concept and analyze the singularities of the reduced Jacobian method, providing an efficient alternative to the pseudo-inverse-based approach, without coupling the general inverse Jacobian problem to the redundancy resolution problem.

The matrix reduction concept can in principle be applied to any linear system with more variables than equations. What makes this approach particularly appealing for the manipulator kinematics area is the fact that the inverse solution can be obtained in symbolic form, which results in a significant reduction of the computational cost. In the process, an analytical tool for deriving both algorithmic and system singularities is provided. The numerical error propagation is reduced to a minimum, increasing the solution accuracy and the region of numerical stability.

The method is very effective for redundant manipulators with a spherical wrist. Nevertheless, general manipulators are not excluded, since the symbolic inversion of a 6×6 matrix is usually feasible when some elements are either zero or very simple. The RPJ decomposition is a suitable tool for providing those characteristics.

We estimate that the method can be applied to redundant arms with up to around 10 joints for complete coverage of algorithmic singularities. Beyond that, we might need to let some algorithmic singularities uncovered, since only a few options would remain for the build-up of reduced Jacobian matrices capable of providing simple symbolic inversion. In this case, however, the high degree of redundancy might enable us to ignore some algorithmic singularities which are avoidable by other means and thus have little impact on the overall performance. For practical applications, we may consider that some algorithmic singularities are incorporated in the system and redundant arms with more than 10 joints can thus benefit from the efficiency of the reduced Jacobian method.

Appendix

A The RPJ Decomposition

Jacobian matrix is mathematically defined as a matrix of partial derivatives. In the robotics field, we use it primarily to describe the velocities (both angular and linear) in the task space as a function of the joint velocities:

$$\dot{\boldsymbol{x}} = J\dot{\boldsymbol{\theta}}. \tag{71}$$

A general element of the Jacobian matrix is thus defined as

$$J_{ij} \triangleq \frac{\partial x_i}{\partial \theta_j}. \tag{72}$$

Once the configuration $\boldsymbol{x}(\boldsymbol{\theta})$ is in symbolic form, one could derive the symbolic expression of each element based on equation (72). However, this method is inefficient because of the complexity of each coordinate x_i. An alternative is based on velocity propagation through the joints [3]. Customizing velocity propagation for revolute joints, with the D–H frame assignment rules (classical notation):

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \boldsymbol{a}_{i-1}\dot{\theta}_i \tag{73}$$

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{p}_i^{i-1}, \tag{74}$$

where $\boldsymbol{a}_i$ is the approaching vector of frame i. Applying equations (73) and (74) recursively from the base frame (0) towards the end-effector frame (n), the task space velocity can be numerically evaluated.

The arm decomposition method is basically a combination of the other two. The arm is divided into 2 serial arms; Jacobian is solved for each part, and velocity propagation is applied to combine the contribution of each part.

Both methods have advantages and disadvantages. Jacobian is required for singularity analysis and inverse function evaluation, but is inefficient as originally defined. Velocity propagation is straightforward and can be very efficient if customized for a given arm, but there is no insight on the entire system, preventing the implementation of the inverse transformation and singularity analysis.

A.1 Vectorial Matrix

Let us analyze the Jacobian matrix from a vectorial point of view. The contribution of the i^{th} joint to the end-effector velocity can be derived by freezing all other joints, and is expressed as:

$$\boldsymbol{\omega}_n(\dot{\theta}_i) = \boldsymbol{a}_{i-1}\dot{\theta}_i \tag{75}$$

$$\boldsymbol{v}_n(\dot{\theta}_i) = \boldsymbol{\omega}_n \times \boldsymbol{p}_n^{i-1} = (\boldsymbol{a}_{i-1} \times \boldsymbol{p}_n^{i-1})\dot{\theta}_i. \tag{76}$$

The entire Jacobian matrix can be obtained by superposition:

$$\dot{\boldsymbol{x}}_n^0 = \begin{bmatrix} \boldsymbol{\omega}_n^0 \\ \boldsymbol{v}_n^0 \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{a}_0 & \cdots & \boldsymbol{a}_{i-1} & \cdots & \boldsymbol{a}_{n-1} \\ \boldsymbol{a}_0 \times \boldsymbol{p}_n^0 & \cdots & \boldsymbol{a}_{i-1} \times \boldsymbol{p}_n^{i-1} & \cdots & \boldsymbol{a}_{n-1} \times \boldsymbol{p}_n^{n-1} \end{bmatrix}}_{J} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_i \\ \vdots \\ \dot{\theta}_n \end{bmatrix}. \tag{77}$$

Although the Jacobian matrix based on equation (77) has a more objective form and added physical meaning, there is no improvement in terms of reducing its symbolic complexity. We notice, however, that the Jacobian in equation (77) is a $2 \times n$ vectorial matrix. The elements of J are coordinate-free vectors. As a consequence, we can take advantage of properties which are independent of the representation frame.

A.2 Representation Transformation

Our goal is to represent the end-effector velocity on the inertial frame[4], ${}^0\dot{\boldsymbol{x}}_n$. But suppose we wish to represent J on frame c. Once ${}^cJ$ is given, we have:

$$\begin{aligned} {}^c\dot{\boldsymbol{x}} &= {}^cJ\,\dot{\boldsymbol{\theta}} \\ {}^0\dot{\boldsymbol{x}} &= {}^0\tilde{R}_c\,{}^c\dot{\boldsymbol{x}}, \end{aligned} \tag{78}$$

thus

$${}^0J = {}^0\tilde{R}_c\,{}^cJ, \tag{79}$$

where

$${}^b\tilde{R}_c \triangleq \begin{bmatrix} {}^bR_c & 0 \\ 0 & {}^bR_c \end{bmatrix} \tag{80}$$

is a 6×6 matrix, representing the transformation of a task space velocity vector from frame c to frame b.

A.3 Reference Point Transformation

Suppose we wish to obtain the velocity of a point in frame n other than its origin. This point must be well defined with respect to the end-effector frame for us to compute the contribution of all joints to its motion. Furthermore, suppose that this point is rigidly connected to the end-effector frame. In this case, we can fully describe its velocity in terms of the end-effector velocity. For a point r with those characteristics:

$$\begin{aligned} \boldsymbol{\omega}_r &= \boldsymbol{\omega}_n, \\ \boldsymbol{v}_r &= \boldsymbol{v}_n + \boldsymbol{\omega}_n \times \boldsymbol{p}_r^n. \end{aligned} \tag{81}$$

[4] For some application we may want to represent the end-effector velocity on the end-effector frame as: ${}^n\dot{\boldsymbol{x}}_n^0$.

We represent a cross product in terms of matrix-vector multiplication:

$$\mathrm{PX}\,\boldsymbol{\omega} = \boldsymbol{p} \times \boldsymbol{\omega} \tag{82}$$

where

$$\mathrm{PX} \triangleq \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \tag{83}$$

is a skew-symmetric matrix defined in terms of the position vector $\boldsymbol{p} = [\, p_x \; p_y \; p_z \,]^T$.

The following properties hold:

$$(\mathrm{PX}_n^r)^T = -\mathrm{PX}_n^r = \mathrm{PX}_r^n \tag{84}$$

We rewrite equation (81) in a combined vectorial form:

$$\begin{aligned} \dot{\boldsymbol{x}}_n &= \tilde{P}_n^r \, \dot{\boldsymbol{x}}_r, \\ \dot{\boldsymbol{x}}_r &= J_r \, \dot{\boldsymbol{\theta}}, \end{aligned} \tag{85}$$

therefore

$$J_n = \tilde{P}_n^r \, J_r, \tag{86}$$

where

$$\tilde{P}_n^r \triangleq \begin{bmatrix} I & 0 \\ -\mathrm{PX}_n^r & I \end{bmatrix} \tag{87}$$

represents the reference point transformation, a 6×6 matrix with I and 0 standing for the 3×3 identity and zero matrices, respectively.

A.4 Generalized Jacobian

Given the definition of representation and reference point transformations, we rewrite equation (77), defining the generalized Jacobian:

$${}^cJ_r \triangleq \begin{bmatrix} {}^c\boldsymbol{a}_0 & \cdots & {}^c\boldsymbol{a}_{i-1} & \cdots & {}^c\boldsymbol{a}_{n-1} \\ {}^c\boldsymbol{a}_0 \times {}^c\boldsymbol{p}_r^0 & \cdots & {}^c\boldsymbol{a}_{i-1} \times {}^c\boldsymbol{p}_r^{i-1} & \cdots & {}^c\boldsymbol{a}_{n-1} \times {}^c\boldsymbol{p}_r^{n-1} \end{bmatrix}. \tag{88}$$

The following relations hold:

$$\begin{aligned} {}^c\dot{\boldsymbol{x}}_r^0 &= {}^cJ_r \, \dot{\boldsymbol{\theta}} \\ {}^c\dot{\boldsymbol{x}}_n^0 &= {}^c\tilde{P}_n^r \, {}^c\dot{\boldsymbol{x}}_r^0 \\ {}^0\dot{\boldsymbol{x}}_n &= {}^0\tilde{R}_c \, {}^c\dot{\boldsymbol{x}}_n^0. \end{aligned} \tag{89}$$

Therefore:

$$\begin{aligned} {}^0J_n &= {}^0\tilde{R}_c \, {}^c\tilde{P}_n^r \, {}^cJ_r \\ {}^0\dot{\boldsymbol{x}}_n &= {}^0J_n \, \dot{\boldsymbol{\theta}}. \end{aligned} \tag{90}$$

A.5 RPJ Decomposition vs Arm Decomposition

The generalized Jacobian presented in the previous section has a powerful geometric interpretation.

Suppose we have chosen r as the origin of a certain intermediate link frame. Notice that, according to our definition of reference point transformation, r is rigidly connected to the end-effector frame and, therefore, its velocity is in general a function of all joint velocities. In our case, r is coincident with an intermediate link frame origin only for the current time and configuration. as the joints move, the previously computed r may not be coincident with that origin due to the displacement of the upper joints, and we need to recalculate r for the next Jacobian evaluation. Therefore, although r is rigidly connected to the end-effector as far as the Jacobian matrix is concerned, the description of r is configuration dependent.

Let us define the following terms:

$$\begin{array}{ll} \boldsymbol{\theta}_L & : \text{lower arm joint configuration } [\theta_1\, \theta_2 \cdots \theta_{r-1}]^T \\ J_{rL} & : \text{lower arm Jacobian sub-matrix} \\ \dot{\boldsymbol{x}}^0_{rL} & : \text{velocity of } r \text{ with respect to the inertial frame, due to } \dot{\boldsymbol{\theta}}_L \\ \boldsymbol{\theta}_U & : \text{upper arm joint configuration } [\theta_r\, \theta_{r+1} \cdots \theta_n]^T \\ J_{rU} & : \text{upper arm Jacobian sub-matrix} \\ \dot{\boldsymbol{x}}^0_{rU} & : \text{velocity of } r \text{ with respect to the inertial frame, due to } \dot{\boldsymbol{\theta}}_U \end{array}$$

According to the definitions above:

$$\begin{aligned} \dot{\boldsymbol{x}}^0_{rL} &= J_{rL}\dot{\boldsymbol{\theta}}_L \\ \dot{\boldsymbol{x}}^0_{rU} &= J_{rU}\dot{\boldsymbol{\theta}}_U. \end{aligned} \tag{91}$$

Since the velocity reference point is the same for both systems, the cross product term of the velocity propagation disappears, and

$$\dot{\boldsymbol{x}}^0_r = \dot{\boldsymbol{x}}^0_{rL} + \dot{\boldsymbol{x}}^0_{rU}, \tag{92}$$

which is fully expressed by the catenation of both Jacobian sub-matrices:

$$J_r = \left[\; J_{rL} \;\vdots\; J_{rU} \;\right], \tag{93}$$

with the significant advantage that the description of the sub-matrices in terms of free vectors is decoupled, since the position vectors of J_{rL} and J_{rU} depend solely on the configuration of the lower arm and upper arm respectively.

Furthermore, in order to evaluate (92) numerically, all the elements involved must be represented on the same frame. It is useful to have the representation frame holding the same decoupling property. The goal is to have J_{rL} as a function of only $\boldsymbol{\theta}_L$ and J_{rU} as a function of only $\boldsymbol{\theta}_U$.

The requirement above is achieved, for instance, by choosing r as the representation frame. Therefore, (92) can be rewritten as:

$${}^r\dot{\boldsymbol{x}}^0_r = {}^r\dot{\boldsymbol{x}}^0_{rL}(\boldsymbol{\theta}_L, \dot{\boldsymbol{\theta}}_L) + {}^r\dot{\boldsymbol{x}}^0_{rU}(\boldsymbol{\theta}_U, \dot{\boldsymbol{\theta}}_U), \tag{94}$$

which is the desired property of arm decomposition.

Equation (94) proves that arm decomposition is a special case of RPJ decomposition, governed by appropriate constraints on the choice of r and c.

A.6 Computational Aspects

There is an efficient short-cut in the computation of Jacobian related to forward kinematics. Usually, before the evaluation of Jacobian, forward kinematics is applied for a variety of reasons. There is no significant additional cost if forward kinematics is evaluated so that the vectors of the generalized Jacobian (88) are computed as intermediate results. In this case, the Jacobian evaluation is reduced to a few cross-product operations.

In order to take advantage of RPJ decomposition in the actual computer program, we need to carefully evaluate our goals. The Jacobian matrix is not only used in the robotics field for velocity transformation. We apply it to static force transformation and acceleration transformation as well. Nevertheless, a given Jacobian matrix is needed only a few times, since the configuration changes due to joint motion, and the Jacobian must be re-evaluated for the next iteration. Therefore, the sequence of matrix-vector multiplications (89) is more efficient than the evaluation of the complete Jacobian followed by the transformation (90).

A more powerful advantage, however, is obtained by customizing the reduced Jacobian transformation for a given arm and exploring the fact that coordinate and reference point transformations are both sparse matrices. The decomposition of the Jacobian into the R, P and J matrices provides a compact formalization of the method, simplifying its analysis. For efficiency, however, we implement the algorithm making use of the underlying concepts, computing linear and angular velocities separately. For the case of a spherical wrist manipulator:

$$
\begin{aligned}
{}^{c}\boldsymbol{v}_r^0 &= D\dot{\boldsymbol{\theta}}_L \\
{}^{c}\boldsymbol{\omega}_r^0 &= A\dot{\boldsymbol{\theta}}_L + B\dot{\boldsymbol{\theta}}_U, \\
{}^{c}\boldsymbol{\omega}_n^0 &= {}^{c}\boldsymbol{\omega}_r^0 \\
{}^{c}\boldsymbol{v}_n^0 &= {}^{c}\boldsymbol{v}_r^0 - {}^{c}\boldsymbol{p}_n^r \times {}^{c}\boldsymbol{\omega}_r^0, \\
{}^{0}\boldsymbol{\omega}_n &= {}^{0}R_c\, {}^{c}\boldsymbol{\omega}_r^0 \\
{}^{0}\boldsymbol{v}_n &= {}^{0}R_c\, {}^{c}\boldsymbol{v}_n^0.
\end{aligned}
\tag{95}
$$

A.7 Symbolic Inversion for Spherical Wrist Arms

In case the non-redundant (or reduced) manipulator holds the spherical wrist geometry, we can write the Jacobian matrix in the following form:

$$
{}^{0}J_n = {}^{0}\tilde{R}_c\, {}^{c}\tilde{P}_n^r\, {}^{c}J_r, \tag{96}
$$

where

$$
{}^{c}J_r = \begin{bmatrix} A & B \\ D & 0 \end{bmatrix} \tag{97}
$$

and $A, B, D, 0$ are 3×3 sub-matrices.

The inversion of coordinate and reference point transformations is straightforward and, therefore,

$$
({}^{0}J_n)^{-1} = ({}^{c}J_r)^{-1}({}^{c}\tilde{P}_n^r)^{-1}({}^{0}\tilde{R}_c)^{-1}, \tag{98}
$$

where

$$({}^0\tilde{R}_c)^{-1} = {}^c\tilde{R}_0 = ({}^0\tilde{R}_c)^T \tag{99}$$

$$({}^c\tilde{P}_n^r)^{-1} = {}^c\tilde{P}_r^n = \begin{bmatrix} I & 0 \\ {}^c\mathrm{PX}_n^r & I \end{bmatrix} \tag{100}$$

$$({}^cJ_r)^{-1} = \begin{bmatrix} 0 & D^{-1} \\ B^{-1} & -B^{-1}AD^{-1} \end{bmatrix}. \tag{101}$$

Notice that, given a Jacobian matrix in the form of equation (97), the task of deriving inverse Jacobian is reduced to the symbolic inversion of two 3×3 sub-matrices, B and D. The remaining task is a few steps of sparse matrix-matrix multiplications.

In order to reduce computational cost, we can evaluate equation (101) in a more efficient form:

$$\begin{aligned} \dot{\boldsymbol{\theta}}_L &= D^{-1}\,{}^c\boldsymbol{v}_r^0, \\ \dot{\boldsymbol{\theta}}_U &= B^{-1}({}^c\boldsymbol{\omega}_r^0 - A\dot{\boldsymbol{\theta}}_L). \end{aligned} \tag{102}$$

Equation (97) allows also a straightforward analysis of arm singularity. The expression for singular configurations of (96) is given by

$$\det(B)\det(D) = 0, \tag{103}$$

since $\det(\tilde{R}) = \det(\tilde{P}) = 1$ by design.

Once B and D are symbolically given, with some zero and trivial elements depending on an appropriate selection of c and r, we can derive B^{-1} and D^{-1} in symbolic form with little effort.

B Singularity Analysis for the AAI Arm

The evaluation of (103) for all possible combinations of parameter joints preserving spherical wrist geometry provides the following singularities[5]:

1. lower part

   D_1: $S_4 C_3 = 0$

   D_2: $S_4 S_3 S_2 = 0$

   D_3: $S_4[(d_3 - d_5 C_4)S_2 + d_5 S_4 C_3 C_2] = 0$

   D_4: intrinsically singular

   Therefore, the system singularities associated to the lower part are:

   (a) $S_4 = 0$: workspace boundary, where the elbow (θ_4) is either fully stretched out or folded back.

   (b) $S_2 = C_3 = 0$: interior singularity, in which $\boldsymbol{a}_2 \equiv \boldsymbol{a}_0$ and $\boldsymbol{p}_5^1 \in (\boldsymbol{a}_1, \boldsymbol{a}_2)$plane.

   In addition, we have the following algorithmic singularities introduced by the reduced Jacobian matrices:

   (a) $C_3 = 0$: D_1 becomes singular, but we can still solve the system if $S_2 \neq 0$ using D_2 or D_3.

   (b) $S_2 = 0$: D_3 becomes singular, but we can still solve the system if $C_3 \neq 0$ using D_1 or D_3.

   (c) $(d_3 - d_5 C_4)S_2 = -d_5 S_4 C_3 C_2$: this condition is more complex to analyze, but some practical results apply. In case $d_3 > d_5$, we notice that when $C_3 = 0$, the only way to cause D_3 to become singular is by having $S_2 = 0$, the system singularity.

   It is clear that the set $\{D_1, D_2\}$ is the ideal choice in terms of general avoidance of algorithmic singularities, but when $d_3 > d_5$, the set $\{D_1, D_3\}$ is equivalent. This condition can be verified during the design of the arm and in this case either selection is appropriate.

2. upper part (spherical wrist)

   B_5: $S_7 = 0$

   B_6: $S_6 C_7 = 0$

   B_7: $C_6 S_7 = 0$

   B_8: $S_6 = 0$

   The singularity associated with the upper part is given by $S_6 = S_7 = 0$. In this case, $\boldsymbol{a}_4 \equiv \boldsymbol{a}_6$ and $\boldsymbol{a}_5 \equiv \boldsymbol{a}_7$. The system is not yet singular, since the positional null space of the lower part can potentially cover the loss of rank in the orientation induced by the condition above. However, a closer look in the geometry of the arm reveals that when at least one of the following conditions is verified, the positional null space of the lower part is not capable of covering the loss of rank in the upper part:

[5] It is assumed that the link distances d_3 and d_5 are not zero.

(a) $C_5 = 0$: The rotation axis of the null space defined by the lower part, $\boldsymbol{p}_5^1$, lies on the $(\boldsymbol{a}_4, \boldsymbol{a}_5)$plane.

(b) $S_2 = 0, C_3 \neq 0$: The null space loses its rotational axis, since $\boldsymbol{a}_2 \equiv \boldsymbol{a}_0$ and $\boldsymbol{p}_5^1 \notin (\boldsymbol{a}_1, \boldsymbol{a}_2)$plane. If $C_3 = 0$, the rotational axis is restored, but in this case the system singularity is due to the lower part.

In addition, we have the following algorithmic singularities introduced by the reduced Jacobian matrices:

(a) $S_7 = 0$: only part of the system singularity is verified. We can still solve the system using B_6 or B_8.

(b) $S_6 = 0$: again, we can still solve the system using B_5 or B_7.

(c) $C_6 = 0$: an algorithmic singularity restricted to B_7.

(d) $C_7 = 0$: an algorithmic singularity restricted to B_6.

The ideal choice, $\{B_5, B_8\}$, is again different from ours, but $\{B_5, B_6\}$ is still solvable for all conditions except the system singularity due to the upper part and, therefore, is equally appropriate.

In summary, the following system singularities are derived:

1. $S_4 = 0$
2. $S_2 = C_3 = 0$
3. $C_5 = S_6 = S_7 = 0$
4. $S_2 = S_6 = S_7 = 0$

References

[1] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, pp. 215–221, June 1955.

[2] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Machine Syst:*, Vol. MMS-10, No. 2, pp. 47–53, June 1969.

[3] K. R. Symon, *Mechanics*, 3rd edition, Addison-Wesley, 1971.

[4] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-7, No. 12, pp. 868–871, December 1977.

[5] G. Strang, *Linear Algebra and its Applications*, Academic Press, Second Edition, 1980.

[6] C. A. Klein and C. H. Huang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 3, pp. 245–250, March/April 1983.

[7] R. P. Paul and C. N. Stevenson, "Kinematics of Robot Wrist," *The Int. Journal of Robotics Research*, Vol. 2, No. 1, pp. 31–38, Spring 1983.

[8] C. T. Chen, *Linear System Theory and Design*, Holt, Rinehart and Winston, Inc., 1984.

[9] W. A. Wolovich and H. Elliott, "A Computational Technique for Inverse Kinematics," it Proc. IEEE Conf. on Decision and Control, pp. 1359–1363, 1984.

[10] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The Int. Journal of Robotics Research*, Vol. 4, No. 2, pp. 3–9, Summer 1985.

[11] J. Baillieul, "Kinematic Programming Alternatives for Redundant Manipulators," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 722–728, 1985.

[12] J. Baillieul, "Avoiding Obstacles and Resolving Kinematic Redundancy," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1698–1704, 1986.

[13] Y. Nakamura and H. Hanafusa, "Optimal Redundancy Control of Robot Manipulators," *The Int. Journal of Robotics Research*, Vol. 6, No. 1, pp. 32–42, Spring 1987.

[14] J. J. E. Slotine and D. R. Yoerger, "A Rule-Based Inverse Kinematics Algorithm for Redundant Manipulators," *Int. Journal of Robotics and Automation*, Vol. 2, No. 2, pp. 86–89, 1987.

[15] J. Angeles, M. Habib and C. S. López-Cajún, "Efficient Algorithms for the Kinematic Inversion of Redundant Robot Manipulators," *Int. Journal of Robotics and Automation*, Vol. 2, No. 3, pp. 106–116, 1987.

[16] Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators," *The Int. Journal of Robotics Research*, Vol. 6, No. 2, pp. 3–15, Summer 1987.

[17] P. H. Chang, "A Closed-Form Solution for Inverse Kinematics of Robot Manipulators with Redundancy," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, pp. 393–403, October 1987.

[18] C. W. Wampler, "Inverse Kinematic Functions for Redundant Manipulators," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 610–617, 1987.

[19] K. C. Suh and J. M. Hollerbach, "Local versus Global Torque Optimization of Redundant Manipulators," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 619–624, 1987.

[20] D. R. Baker and C. W. Wampler, "On the Inverse Kinematics of Redundant Manipulators," *The Int. Journal of Robotic Research*, Vol. 7, No. 2, pp. 3–21, March/April 1988.

[21] R. Dubey and J. Y. S. Luh, "Redundant Robot Control Using Task Based Performance Measures," *Journal of Robotic Systems*, Vol. 5, No. 5, pp. 409–432, 1988.

[22] K. Kazerounian and Z. Wang, "Global versus Local Optimization in Redundancy Resolution of Robotic Manipulators," *The Int. Journal of Robotic Research*, Vol. 7, No. 5, pp. 3–12, October 1988.

[23] R. V. Dubey, J. A. Euler and S. M. Babcock, "An Efficient Gradient Projection Optimization Scheme for a Seven-Degree-of-Freedom Redundant Robot with Spherical Wrist", *Proc. IEEE Conf. on Robotics and Automation*, pp. 28–36, April 1988.

[24] C. Chevallereau and W. Khalil, "A New Method for the Solution of the Inverse Kinematics of Redundant Robots", *Proc. IEEE Conf. on Robotics and Automation*, pp. 37–42, April 1988.

[25] H. Das, J. J. E. Slotine and T. B. Sheridan, "Inverse Kinematic Algorithms for Redundant Systems", *Proc. IEEE Conf. on Robotics and Automation*, pp. 43–48, April 1988.

[26] Y. L. Gu, "Dynamics and Control for Redundant Robots," *Proc. IEEE Conf. on Robotics and Automation*, pp. 194–199, April 1988.

[27] A. Fijany and A. K. Bejczy, "Efficient Jacobian Inversion for the Control of Simple Robot Manipulators," *Proc. IEEE Conf. on Robotics and Automation*, pp. 999–1007, April 1988.

[28] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed., Addison-Wesley Publishing Company, Inc., 1989.

[29] H. Seraji, "Configuration Control of Redundant Manipulators: Theory and Implementation," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 4, pp. 472–490, August 1989.

[30] D. P. Martin, J. Baillieul and J. M. Hollerbach, "Resolution of Kinematic Redundancy Using Optimization Techniques," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 4, pp. 529–533, August 1989.

[31] J. A. Euler, R. V. Dubey and S. M. Babcock, "Self Motion Determination Based on Actuator Velocity Bounds for Redundant Manipulators," *Journal of Robotic Systems*, Vol. 6, No. 4, pp. 417–425, 1989.

[32] P. R. Chang and C. S. G. Lee, "Residue Arithmetic VLSI Architecture for Manipulators in the Presence of Redundancies," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 5, pp. 569–582, October 1989.

[33] K. Anderson and J. Angeles, "Kinematic Inversion of Robotic Manipulators in the Presence of Redundancies," *The Int. Journal of Robotics Research*, Vol. 8, No. 6, pp. 80–97, December 1989.

[34] S. Hirose and S. Ma, "Redundancy Decomposition Control for Multi-Joint Manipulator," *Proc. IEEE Conf. on Robotics and Automation*, pp. 119–124, 1989.

[35] K. Cleary and D. Tesar, "Incorporating Multiple Criteria in the Operation of Redundant Manipulators," *Proc. IEEE Conf. on Robotics and Automation*, pp. 618–624, 1990.

[36] A. A. Maciejewski, "Fault Tolerant Properties of Kinematically Redundant Manipulators," *Proc. IEEE Conf. on Robotics and Automation*, pp. 638–642, 1990.

[37] R. V. Mayorga and A. K. C. Wong, "A Singularities Prevention Approach for Redundant Robot Manipulators," *Proc. IEEE Conf. on Robotics and Automation*, pp. 812–817, 1990.

[38] H. Zghal, R. V. Dubey and J. A. Euler, "Efficient Gradient Projection for Manipulators with Multiple Degrees of Redundancy," *Proc. IEEE Conf. on Robotics and Automation*, pp. 1006–1011, 1990.

[39] M. Z. Huang and H. Varma, "Optimal Rate Allocation in Kinematically Redundant Manipulators – The Dual Projection Method," *Proc. IEEE Conf. on Robotics and Automation*, pp. 702–707, 1991.

[40] S. Lee and A. K. Bejczy, "Redundant Arm Kinematic Control Based on Parameterization," *Proc. IEEE Conf. on Robotics and Automation*, pp. 458–465, 1991.