Proceedings of the 2001 IEEE/RSJ
International Conference on Intelligent Robots and Systems
Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001

The Singular Vector Algorithm for the Computation of Rank-Deficiency Loci of Rectangular Jacobians

E. Dupuis
Canadian Space Agency
Space Technologies
St-Hubert, Canada
erick.dupuis@space.gc.ca

E. Papadopoulos
Nat. Tech. Univ. of Athens
Mechanical Engineering
Athens, Greece
egpapado@central.ntua.gr

V. Hayward
McGill University
Electrical Engineering
Montréal, Canada
hayward@cim.mcgill.ca

Abstract

This paper presents a novel approach to compute the rank-deficiency locus of non-square Jacobian matrices. This algorithm is based on the computation of the singular vectors associated to zero singular values of the Jacobian. Examples are provided to illustrate the application of the algorithm. Results are shown for a four degree-of-freedom and a seven degree-of-freedom manipulator.

1 Introduction

Kinematically redundant manipulators are generally defined as manipulators with more joint degrees-of-freedom than the minimum required to define the pose of their end-effector. This implies that each end-effector pose can be reached using an infinity of manipulator postures. This property is very useful for operations in cluttered environments and has led to the rapid proliferation of redundant manipulators in many areas of application.

For example, in space applications, kinematically redundant manipulators are currently used in routine operations or slated for launch in the short to medium term. The Canadian Space Station Remote Manipulator System (SSRMS) and Special Purpose Dextrous Manipulator (SPDM), the European Robotic Arm (ERA), the Italian SPIDER Arm and the Ranger Teleoperation Shuttle Experiment are examples of kinematically redundant space manipulators.

The resolution of kinematic redundancy for manipulators operating in cluttered environments has traditionally used Jacobian augmentation techniques such as originally proposed by Oh, Orin and Bach [1] and by Bailleul [2]. Constraints are typically added to avoid obstacles or to dictate the posture of the manipulator [3] [4] [5]. Today, the inverse kinematics algorithms of most space-based manipulators use Jacobian augmentation methods to some extent to resolve kinematic redundancy [6] [7].

The constrained kinematic equation of the manipulator takes the following form:

$$\begin{bmatrix} \mathbf{v}_t \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{J}_t(\mathbf{q}) \\ \mathbf{J}_c(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} \qquad (1)$$

where $\mathbf{v}_t$ is the set of translational and angular velocities associated with the task coordinates, $\mathbf{v}_c$ is the set of translational and angular velocities associated with the constraint coordinates and $\dot{\mathbf{q}}$ is the set of joint rates. The concatenation of the task Jacobian matrix, $\mathbf{J}_t(\mathbf{q})$, and the constraint Jacobian matrix, $\mathbf{J}_c(\mathbf{q})$ forms the augmented Jacobian.

A major limitation of augmented Jacobians is the fact that the constraint equations used to augment the task Jacobian can introduce algorithmic rank-deficiencies [8].

The rank-deficiency locus $\mathcal{S}$ of a Jacobian matrix $\mathbf{J}(\mathbf{q})$ is defined as the set of all joint values $\mathbf{q}^*$ such that $\mathbf{J}(\mathbf{q}^*)$ does not have full rank. Analysing the rank-deficiency locus of a given augmented Jacobian and comparing it to that of the task Jacobian, it is possible to determine whether this augmented Jacobian induces algorithmic rank-deficiencies. It is therefore possible to use these analyses to generate sets of task and constraint coordinate pairs that will ensure an appropriate coverage of the configuration space, $\mathcal{Q}$, of the manipulator.

This paper presents a novel algorithm to compute, in symbolic form, the rank-deficiency locus of rectangular Jacobian matrices based on the zero-singular vectors of the Jacobian matrix. The main advantage of computing rank-deficiency loci symbolically is that this provides a global solution over all of $\mathcal{Q}$. Furthermore, the rank-deficiency loci solutions thus computed

0-7803-6612-3/01/$10.00©2001 IEEE

can easily be used for further analyses of the manipulator's kinematics.

Local methods, although less computer-intensive per test, require that the testing be performed everywhere in the configuration space. In practice, such a thing is impossible and the testing would have to be limited to a grid of points in $\mathcal{Q}$. However, the number of points in this test grid increases exponentially with the number of degrees-of-freedom of the manipulator and it is difficult to guarantee that any grid fineness will ever be sufficient to ensure that no singular configurations have been missed.

2 Existing methods

Very few symbolic methods have been developed for the symbolic computation of rank-deficiency loci of Jacobian matrices and fewer still for rectangular Jacobians. The simplest case is that of square Jacobian matrices. For such matrices, loss of rank implies that the matrix becomes singular and that its determinant is zero. The rank-deficiency locus can be computed as follows:

$$\mathcal{S}_{sq} = \{\mathbf{q}^* \mid \det(\mathbf{J}(\mathbf{q}^*)) = 0\} \qquad (2)$$

For kinematically redundant manipulators, a few algorithms exist. The subdeterminant method for computing rank-deficiency loci takes advantage of the fact that when a rectangular matrix loses rank, all square sub-matrices of the same dimension as the lower dimension of the rectangular matrix also become singular. The rank-deficiency locus of the rectangular matrix is the intersection of the singularity loci of the square submatrices resulting from all possible combinations of columns of $\mathbf{J}$, $\mathcal{S} = \bigcap_i \mathcal{S}_{sq_i}$. Unfortunately this method proves unwieldy as the number of square submatrices increases combinatorially with the number of degrees of freedom of the manipulator and the number of redundant degrees of freedom.

To address the limitations of the subdeterminant method, Nokleby and Podhorodeski have suggested an alternate approach [9] based on screw theory and the principle of virtual power. The algorithm is based on the fact that, at a rank-deficient configuration, there exists a wrench along which the manipulator cannot perform work. This algorithm finds this wrench and the set of joint values leading to the rank-deficient configuration. This approach is more computationally efficient than the subdeterminant method but it is limited to task spaces that can be represented by screws and to rectangular Jacobians with more columns than rows.

3 Singular vector algorithm

The singular vector algorithm for determining rank-deficiency loci of rectangular Jacobian matrices is a generalisation of the algorithm of Nokleby and Podhorodeski [9] but it uses linear algebra instead of screw algebra. The main advantage of the singular vector algorithm is that it can handle rectangular Jacobians of any row and column dimension.

From the definition of rank-deficiency, a rectangular matrix with more columns than rows becomes rank-deficient when its rows are linearly dependent. The existence of a rank deficiency then implies that there exists a set of conditions for which a set of left singular vectors can be found such that the dot product of these singular vectors with all columns of the Jacobian matrix is zero. The singular vector algorithm for computing the rank-deficiency locus of a rectangular Jacobian matrix determines the conditions for which such singular vectors exist. Note that the same reasoning can be applied to rectangular matrices with more rows than columns except that then the columns become linearly dependent.

The methodology will be explained for the case when the Jacobian matrix has more columns than rows $n < m$. This corresponds to kinematically redundant manipulators: there are more joint variables than motion equations to be solved. The rank-deficiency locus then is the set of all values of $\mathbf{q}^*$ such that the rank of the Jacobian matrix is lower than its number of rows. The methodology can easily be generalised to the case when the Jacobian matrix has more rows than columns, which corresponds to an overdetermined system of equations. In this case, the columns of $\mathbf{J}$ are considered instead of its rows and the right singular vectors are used instead of the left singular vectors.

The first step in the computation of the rank-deficiency locus of $\mathbf{J}$ is to extract n columns out of $\mathbf{J}(\mathbf{q})$ to form $\mathbf{J}_{sq}(\mathbf{q})$. The remaining columns of $\mathbf{J}(\mathbf{q})$ are called the redundant columns and form $\mathbf{J}_r(\mathbf{q})$.

$$\mathbf{J}_{sq}(\mathbf{q}) = \begin{bmatrix} \mathbf{s}_1(\mathbf{q}) & \mathbf{s}_2(\mathbf{q}) & \cdots & \mathbf{s}_n(\mathbf{q}) \end{bmatrix} \qquad (3)$$

$$\mathbf{J}_r(\mathbf{q}) = \begin{bmatrix} \mathbf{r}_1(\mathbf{q}) & \mathbf{r}_2(\mathbf{q}) & \cdots & \mathbf{r}_{m-n}(\mathbf{q}) \end{bmatrix} \qquad (4)$$

The rank-deficiency (singularity) locus of the square sub-Jacobian is computed symbolically by equating its determinant to zero and solving for $\mathbf{q}$:

$$\mathcal{S} = \mathcal{S}_{sq} = \{\mathbf{q}^* \mid \det(\mathbf{J}_{sq}(\mathbf{q}^*)) = 0\} \qquad (5)$$

The rank-deficiency locus of the square sub-Jacobian is then refined iteratively by substituting each condition $\mathbf{q}_i^* \in \mathcal{S}_{sq}$ and finding the conditions that can further reduce the rank of $\mathbf{J}_{sq}(\mathbf{q}_i^*)$. This can be done simply by triangularising the rank-deficient matrix $\mathbf{J}_{sq}(\mathbf{q}_i^*)$ using Gaussian elimination. The result of the Gaussian elimination is a triangular matrix $\mathbf{J}_{\triangle}(\mathbf{q}_i^*)$ whose last row is entirely composed of zeroes.

An upper-triangular submatrix, $\mathbf{J}_{\triangle sub}(\mathbf{q}_i^*)$, is then extracted out of $\mathbf{J}_{\triangle}(\mathbf{q}_i^*)$ by removing its last row. The Singular Vector Algorithm is then applied recursively to $\mathbf{J}_{\triangle sub}(\mathbf{q}_i^*)$ to determine conditions that reduce its rank even more. The recursion stops when $\mathbf{J}_{\triangle}(\mathbf{q}_i^*)$ has only one row left or when $\mathbf{J}_{\triangle sub}(\mathbf{q}_i^*)$ cannot be made rank-deficient. All the conditions for which $\mathbf{J}_{sq}(\mathbf{q}_i^*)$ has reduced rank are recorded in $\mathcal{S}_{sq}$ as additional solution branches.

For each individual branch of the solution $\mathbf{q}_i^* \in \mathcal{S}_{sq}$, the rank-deficiency conditions are substituted back in $\mathbf{J}_{sq}(\mathbf{q})$ and the left singular vectors associated to the zero singular values of the singular square sub-Jacobian are computed as:

$$\mathbf{u}_i^*(\mathbf{q}) = \begin{bmatrix} u_{i1}(\mathbf{q}) & u_{i2}(\mathbf{q}) & \dots & u_{in}(\mathbf{q}) \end{bmatrix}^T \quad (6)$$

such that

$$[\mathbf{u}_i^*(\mathbf{q})]^T \, \mathbf{J}_{sq}(\mathbf{q}^*) = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix} \quad (7)$$

and

$$\mathbf{u}_i^*(\mathbf{q}) \cdot \mathbf{u}_j^*(\mathbf{q}) = 0 \quad \text{for } i \neq j \quad (8)$$

The vectors $\mathbf{u}_i^*(\mathbf{q})$ span the nullspace of $[\mathbf{J}_{sq}(\mathbf{q}_i^*)]^T$. They are then arranged in a matrix of singular vectors as follows:

$$\mathbf{U}^*(\mathbf{q}) = \begin{bmatrix} \mathbf{u}_1^*(\mathbf{q}) & \mathbf{u}_2^*(\mathbf{q}) & \dots & \mathbf{u}_k^*(\mathbf{q}) \end{bmatrix} \quad (9)$$

where k corresponds to the number of zero singular values of the matrix $\mathbf{J}_{sq}(\mathbf{q}_i^*)$. The singularity conditions $\mathbf{q}_i^*$ are then substituted in $\mathbf{J}_r(\mathbf{q})$ and a new matrix is generated by multiplying the matrix of singular vectors, $\mathbf{U}^*(\mathbf{q})$, with the redundant columns of the Jacobian as follows:

$$\mathbf{J}^\dagger(\mathbf{q}) = [\mathbf{U}^*(\mathbf{q})]^T \mathbf{J}_r(\mathbf{q}_i^*) \quad (10)$$

The rank-deficiency locus $\mathcal{S}$ is refined by repeating the algorithm recursively to find the conditions under which $\mathbf{J}^\dagger(\mathbf{q})$ also loses rank. A tree of solution branches is thus formed each solution branch of the singularity locus of $\mathbf{J}_{sq}(\mathbf{q})$ leading to potentially many sub-branches being rank-deficiency loci of $\mathbf{J}^\dagger(\mathbf{q})$. The recursion continues until one of three conditions is met.

1. The rank-deficiency locus of $\mathbf{J}^\dagger(\mathbf{q})$ is the empty set: In this case, the set of solution branches of rank-deficiency loci being investigated are not part of the rank-deficiency locus of the overall Jacobian matrix.
2. The number of singular vectors k in $\mathbf{U}^*(\mathbf{q})$ is larger than the number of columns of $\mathbf{J}_r(\mathbf{q})$: In this case, the set of solution branches followed up to this point is obviously part of the rank-deficiency locus of the overall Jacobian matrix because the number of redundant columns is insufficient to cancel entirely the nullspace of $\mathbf{J}_{sq}(\mathbf{q})$.
3. The last redundant column of the matrix $\mathbf{J}(\mathbf{q})$ has been used in $\mathbf{J}_{sq}(\mathbf{q})$: this means that there are no more possible refinements of the rank-deficiency locus $\mathcal{S}$ for the particular set of solutions branches that has been followed.

In each of these cases, the algorithm updates the rank deficiency locus of $\mathbf{J}(\mathbf{q})$ accordingly. If a solution was found, then the intersection of the set of rank-deficiency loci $\{\mathbf{q}^*\} \in \mathcal{S}_{sq}$ of the terminal branch and that of all of its parents is added to the rank-deficiency locus $\mathcal{S}$ of the overall Jacobian. Otherwise, the branch is simply ignored. The algorithm then backtracks in the solution tree until it encounters a branch of the rank-deficiency locus that has not yet been investigated.

After all branches of the solution tree have been investigated, $\mathcal{S}$ then contains the entire rank-deficiency locus of the rectangular Jacobian.

This algorithm is very computationally efficient since it is applied to matrices of rapidly decreasing dimension. It uses only once a square submatrix $\mathbf{J}_{sq}(\mathbf{q})$ whose dimension is equal to the smallest dimension of $\mathbf{J}$. The dimension of the matrices at the next recursion decreases to the dimension of the nullspace of $\mathbf{J}_{sq}(\mathbf{q}^*)$.

Furthermore, the algebraic complexity of the determinant equation of $\mathbf{J}_{sq}$ can be minimised amongst all possible combinations of columns of $\mathbf{J}$ at the cost of computing the determinant equations of all square submatrices of $\mathbf{J}$. The cost of this operation is combinatorial in the number of columns and rows of $\mathbf{J}$ but it only involves additions, multiplications and algebraic simplifications. In most cases, this step is well worth the computational expense since it takes less time to perform than attempting to solve the determinant equation of an arbitrarily selected $\mathbf{J}_{sq}$.

Since it computes rank-deficiency loci in symbolic form, there will undoubtedly be a limit to the complexity of the kinematic equations beyond which the computation of the rank-deficiency locus will not be practically feasible. However, different techniques can be used to simplify the computation of the rank-deficiency locus of the Jacobian and thus push this limit. For example, Waldron [10] has shown that the selection of an appropriate reference frame to express the kinematic equations can greatly simplify the cost of computing the Jacobian.

4 Sample cases

The singular vector algorithm was implemented using the Maple symbolic computation package. The following examples illustrate the application of the algorithm to simple, yet non-trivial, sample cases. Both examples use only revolute joints but the algorithm can handle any joint type as long as the kinematic equations can be expressed in differential form as in Equation (1).

4.1 4R spherical-shoulder manipulator

Let us first consider the case of a 4R spherical shoulder manipulator with four revolute joints arranged in a cluster of three joints at the shoulder in a roll-yaw-pitch configuration followed by an elbow pitch joint as shown on Figure 1. The two links are of identical length L.

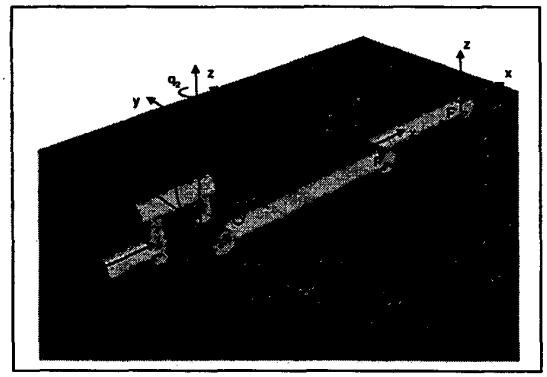

Figure 1: 4R spherical-shoulder manipulator.

Presuming that the task coordinates of this manipulator describe the Cartesian position of its end-effector, then its task Jacobian is as follows:

$$\mathbf{J}_T = \begin{bmatrix} Ls_2s_4 & 0 & Ls_4 & 0 \\ J_{21} & J_{22} & 0 & 0 \\ -Ls_2c_4 & 0 & -L(c_4+1) & -L \end{bmatrix} \quad (11)$$

where $J_{21} = Lc_2(s_3 + s_{34})$, $J_{22} = -L(c_3 + c_{34})$, $c_i = \cos(q_i)$, $s_i = \sin(q_i)$, $c_{ij} = \cos(q_i + q_j)$ and $s_{ij} = \sin(q_i + q_j)$. Note that to reduce the cost of computing the Jacobian, it is expressed in a reference frame attached to the third joint.

Selecting the last three columns of $\mathbf{J}_T$ to build, $\mathbf{J}_{sq}$, we obtain a matrix whose determinant equation is:

$$L^3 s_4(c_3 - s_3s_4 - c_3c_4) = 0 \quad (12)$$

and whose rank-deficiency locus is:

$$\mathcal{S}_{sq} = \begin{cases} q_4 = 0, \pi \\ q_4 = \pi - 2q_3 \end{cases} \quad (13)$$

Substituting $q_4 = 0$ into $\mathbf{J}_{sq}(\mathbf{q})$ and performing Gaussian elimination, we obtain:

$$\mathbf{J}_\Delta(\mathbf{q}) = \begin{bmatrix} -2L\cos(q_3) & 0 & 0 \\ 0 & -2L & -L \\ 0 & 0 & 0 \end{bmatrix} \quad (14)$$

The rectangular matrix $\mathbf{J}_{\Delta sub}(\mathbf{q})$ is built by extracting the first two rows of $\mathbf{J}_\Delta(\mathbf{q})$. Applying the Singular Vector algorithm recursively on $\mathbf{J}_{\Delta sub}(\mathbf{q})$, it is found that the condition $q_3 = \pm\frac{\pi}{2}$ further reduces the rank of $\mathbf{J}_{sq}(\mathbf{q})$. Its rank-deficiency locus is then updated as:

$$\mathcal{S}_{sq} = \begin{cases} q_4 = 0, \pi \\ q_4 = \pi - 2q_3 \\ q_4 = 0; q_3 = \pm\frac{\pi}{2} \end{cases} \quad (15)$$

Conducting the same exercise for each branch of $\mathcal{S}_{sq}$, it is found that Equation (15) includes all solution branches that further reduce the rank of $\mathbf{J}_{sq}(\mathbf{q})$.

Next, for each solution branch in $\mathcal{S}_{sq}$, the singular vectors associated to the zero singular values of $\mathbf{J}_{sq}(\mathbf{q}^*)$ are found. Let us select the branch $q_4 = \pi - 2q_3$ and substitute back into $\mathbf{J}_{sq}(\mathbf{q})$. We obtain the following matrix:

$$\mathbf{J}_{sq}(\mathbf{q}^*) = \begin{bmatrix} 0 & 2L\sin(q_3)\cos(q_3) & 0 \\ 0 & 0 & 0 \\ 0 & -2L\sin^2(q_3) & -L \end{bmatrix} \quad (16)$$

whose zero left singular vector is $\mathbf{u} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$. Taking the product of this singular vector with the redundant column of $\mathbf{J}_T(\mathbf{q})$, we obtain a one by one

matrix $\mathbf{J}^{\dagger}(\mathbf{q}) = [\ 2L\cos(q_2)\sin(q_3)\ ]$ whose rank-deficiency locus is:

$$\mathcal{S}_{sq} = \begin{cases} q_3 = 0, \pi \\ q_2 = \pm\frac{\pi}{2} \end{cases} \tag{17}$$

Since the last redundant column of $\mathbf{J}_T(\mathbf{q})$ has been processed, the intersection of these rank-deficiency conditions with their parent solution branch are rank-deficiency loci of $\mathbf{J}_T(\mathbf{q})$. The overall rank-deficiency locus is then updated as:

$$\mathcal{S}_T = \begin{cases} q_4 = \pi, & q_3 = 0; \pi \\ q_2 = \pm\frac{\pi}{2}; & q_4 = \pi - 2q_3 \end{cases} \tag{18}$$

Similarly, processing each branch of $\mathcal{S}_{sq}$ and removing rank-deficiency loci that are subsets of others, the overall rank-deficiency locus of $\mathbf{J}_T(\mathbf{q})$ is then found to be:

$$\mathcal{S}_T = \begin{cases} q_4 = 0, \pi \\ q_2 = \pm\frac{\pi}{2}; & q_4 = \pi - 2q_3 \end{cases} \tag{19}$$

The first rank-deficiency loci $q_4 = 0, \pi$ correspond to workspace boundary singularities where the manipulator is either fully stretched or fully folded on itself. The next set of rank-deficiency loci occur when the axes of the first and third joints are aligned and the end-effector is lying on the axis of the second joint. These configurations are shown on Figure 2

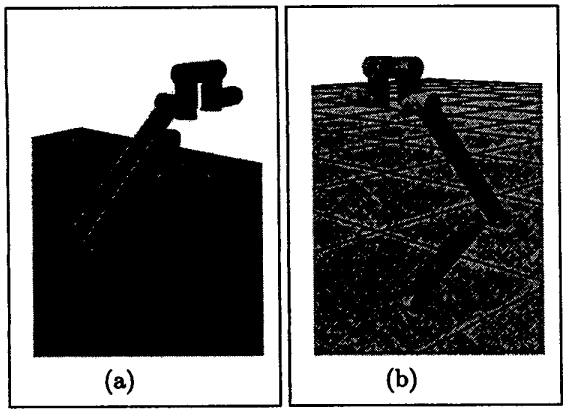

Figure 2: Singular configurations of a 4R spherical-shoulder manipulator: (a) $q_4 = \pi$, (b) $q_2 = \frac{\pi}{2}$ and $q_4 = \pi - 2q_3$.

4.2 7R spherical shoulder and wrist manipulator

As a second example, let us consider adding a spherical wrist to the tip of the manipulator used in the previous example (See Figure 3).

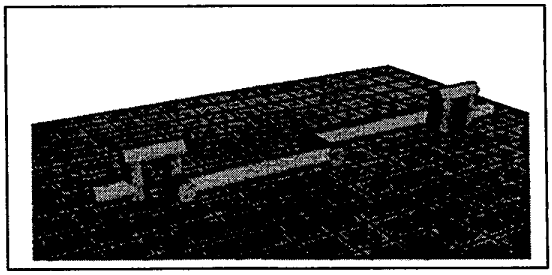

Figure 3: 7R spherical-shoulder and wrist manipulator.

Applying the Singular Vector Algorithm to the 6×7 task Jacobian of this manipulator, it has been found that its rank-deficiency locus is:

$$\mathcal{S}_T = \begin{cases} q_4 = 0, \pi & \\ q_2 = \pm\frac{\pi}{2} & q_6 = \pm\frac{\pi}{2} \\ q_2 = \pm\frac{\pi}{2} & q_4 = \pi - 2q_3 \\ q_6 = \pm\frac{\pi}{2} & q_4 = \pi - 2q_5 \end{cases} \tag{20}$$

The configurations at which the task Jacobian is rank-deficient are shown on Figure 4. The configurations $q_4 = 0$ and $q_4 = \pi$ are workspace boundary rank-deficiencies where the elbow is either fully extended or fully folded. Note that since all joints are assumed to be without offsets, the configuration at $q_4 = \pi$ is not physically achievable. The rank-deficiency locus at $q_2 = \pm\frac{\pi}{2}$ and $q_6 = \pm\frac{\pi}{2}$ represents the case when the axes of five out of seven joints of the manipulator are parallel. Both the wrist and the shoulder joint clusters can effect a self-motion as was already described for the 4R Spherical Shoulder Manipulator: the self-motion manifold is therefore two-dimensional. Since the manipulator has only one more degree-of-freedom than is necessary to completely define the task coordinates, then the task Jacobian is necessarily rank-deficient.

The rank-deficiency locus at $q_2 = \pm\frac{\pi}{2}$ and $q_4 = \pi - 2q_3$ represents the case when the shoulder roll and pitch joints are co-axial and the centre point of the wrist lies on the axis of the shoulder yaw joint. In this configuration, the manipulator cannot move its wrist centre point in a direction perpendicular to the pitch plane. Finally, the rank-deficiency locus where $q_6 = \pm\frac{\pi}{2}$ and $q_4 = \pi - 2q_5$ is the symmetric equivalent of the previous one except that, in this case, it is the centre point of the shoulder joint cluster that is lying on the axis of the wrist yaw joint.

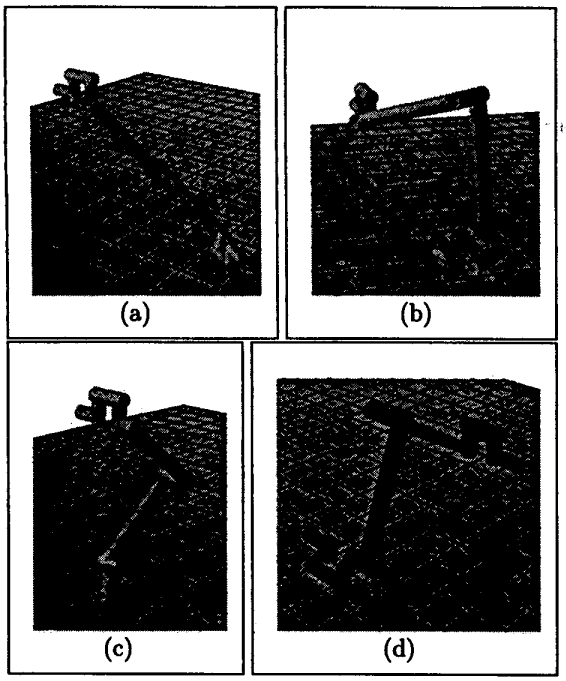

Figure 4: Rank-deficient configurations of the task Jacobian of the 7R spherical-shoulder and wrist manipulator: (a) $q_4 = 0$, (b) $q_6 = \frac{\pi}{2}$, $q_4 = \pi - 2q_5$, (c) $q_2 = \frac{\pi}{2}$, $q_4 = \pi - 2q_3$, (d) $q_2 = q_6 = \frac{\pi}{2}$.

5 Conclusion

This paper has introduced the Singular Vector Algorithm to compute symbolically the rank-deficiency locus of rectangular Jacobian matrices.

This algorithm is computationally very efficient since it is applied to matrices of rapidly decreasing dimension. It uses only once a square submatrix $\mathbf{J}_{sq}(\mathbf{q})$ whose dimension is equal to the smallest dimension of the rectangular Jacobian matrix $\mathbf{J}(\mathbf{q})$. The dimension of the matrices at the next recursion decreases to the dimension of the nullspace of $\mathbf{J}_{sq}(\mathbf{q}^*)$. The algorithm has been used successfully to compute, in symbolic form, the rank-deficiency locus of manipulators with respectively 4 and 7 degrees-of-freedom.

References

[1] S.Y. Oh et.al., "An Inverse Kinematic Solution for Kinematically Redundant Robot Manipulators", *J. of Robotic Systems*, Vol. 1, (1984)

[2] J. Bailleul, "Kinematic Programming Alternatives for Redundant Manipulators", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (1985)

[3] H. Seraji, "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Trans. on Robotics and Automation*, Vol. 5, (1989)

[4] C.L. Boddy and J.D. Taylor, "Whole Arm Reactive Collision Avoidance of Kinematically Redundant Manipulators", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (1993)

[5] T. Tsuji et.al., "Instantaneous Inverse Kinematic Solution for Redundant Manipulators Based on Virtual Arms and Its Application to Winding Control", *JSME Int. J.*, 38(1), (1995)

[6] MacDonald Detwiller Space and Advanced Robotics Ltd, "SSRMS Arm Control Design", (1997)

[7] C.R. Carignan and R.D. Howard, A Partitioned Redundancy Management Scheme for an Eight Joint Revolute Manipulator, *submitted to J. of Robotic Systems*, (to appear)

[8] Z. Wang and K. Kazerounian, "Identification and Resolution of Structural and Algorithmic Singularity in Redundancy Control of Serial Manipulators", *J. of Robotic Systems*, 12(7), 465-478 (1995)

[9] S.B. Nokleby and R.P. Podhorodeski, "Methods for Resolving Velocity Degeneracies of Joint-Redundant Manipulators", *Advances in Robot Kinematics*, 217-226 (2000)

[10] K.J. Waldron et.al., "A Study of the Jacobian Matrix of Serial Manipulators", *Trans. of the ASME*, Vol. 107, pp.230-238 (1985)