The Singular Vector Method for Computing Rank-Deficiency Loci of Rectangular Jacobian Matrices

Erick Dupuis*

* Space Technologies, Canadian Space Agency, St-Hubert, QC, Canada, J3Y 8Y9

Abstract

This paper presents a novel approach to compute the rank-deficiency locus of non-square Jacobian matrices. This approach is based on the computation of the singular vectors associated to zero singular values of the Jacobian.

1. Introduction

With the advent of the International Space Station, space-based robotic systems are becoming increasingly common and several manipulator systems are slated for launch in the short to medium term future. Most of the space manipulators in development use kinematically redundant designs since they will have to operate in cluttered environments. The inverse kinematics algorithms of all of these systems use Jacobian augmentation methods to resolve the kinematic redundancy. A major limitation of augmented Jacobians is the fact that the constraint equations used to augment the Jacobian can introduce algorithmic singularities [1].

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. Rank-deficiency loci provide a means to analyse whether the constraint equations used to augment the Jacobian of a redundant manipulator do insert algorithmic singularities. This paper presents a novel approach to compute the rank-deficiency locus of rectangular Jacobian matrices.

2. Review of Existing Methods

The simplest method to compute rank-deficiency loci is in the case of square Jacobian matrices. For such matrices, the loss of a rank means that the matrix becomes singular and that its determinant is zero. The rank-deficiency locus can be computed in symbolic form as follows:

$$\mathcal{S}_{sq} = \{\mathbf{q}^* \mid \det(\mathbf{J}(\mathbf{q}^*)) = 0\} \tag{1}$$

For rectangular Jacobian matrices, the determinant method is not applicable. Different methods have been developed to address this problem for kinematically redundant manipulators.

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, an alternate approach [2] based on screw theory and the principle of virtual power has been proposed. The algorithm first extracts a square submatrix of dimension equal to the number of rows of the Jacobian. The determinant equation of this square submatrix is solved to find its singularity locus. These conditions are substituted back into $\mathbf{J}$ and the wrench along which the manipulator cannot do work in this singular configuration is found. The reciprocal product of this wrench is then taken with each of the columns of the Jacobian that were not part of the square submatrix to find the joint values that will lead to zero virtual power. The process is repeated until all columns of the Jacobian have been used. 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 Method

The singular vector method for determining rank-deficiency loci of rectangular Jacobian matrices is a generalisation of the reciprocity-based method [2] using traditional linear algebra techniques instead of screw algebra. The main advantage of the singular vector method 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[1]. The existence of a rank deficiency then implies that there exists a set of conditions for which a set of 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 method for computing the rank-deficiency locus of a rectangular Jacobian matrix determines the conditions for which such singular vectors exist.

[1] 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. 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. The columns of $\mathbf{J}$ are then 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 $\mathbf{J}_r(\mathbf{q})$.

$$\mathbf{J}_{sq}(\mathbf{q}) = \left[\begin{array}{cccc} \mathbf{s}_1(\mathbf{q}) & \mathbf{s}_2(\mathbf{q}) & \dots & \mathbf{s}_n(\mathbf{q}) \end{array} \right] \tag{2}$$

$$\mathbf{J}_r(\mathbf{q}) = \left[\begin{array}{cccc} \mathbf{r}_1(\mathbf{q}) & \mathbf{r}_2(\mathbf{q}) & \dots & \mathbf{r}_{m-n}(\mathbf{q}) \end{array} \right] \tag{3}$$

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

$$\mathcal{S} = \mathcal{S}_{sq} = \{\mathbf{q}^* \mid \det(\mathbf{J}_{sq}(\mathbf{q}^*)) = 0\} \tag{4}$$

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

$$\mathbf{u}_i{}^*(\mathbf{q}) = \left[\begin{array}{cccc} u_{i1}(\mathbf{q}) & u_{i2}(\mathbf{q}) & \dots & u_{in}(\mathbf{q}) \end{array} \right]^T \tag{5}$$

such that

$$[\mathbf{u}_i{}^*(\mathbf{q})]^T \, \mathbf{J}_{sq}(\mathbf{q}^*) = \left[\begin{array}{cccc} 0 & 0 & \dots & 0 \end{array} \right] \tag{6}$$

and

$$\mathbf{u}_i{}^*(\mathbf{q}) \cdot \mathbf{u}_j{}^*(\mathbf{q}) = 0 \quad \text{for } i \neq j \tag{7}$$

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

$$\mathbf{U}^*(\mathbf{q}) = \left[\begin{array}{cccc} \mathbf{u}_1{}^*(\mathbf{q}) & \mathbf{u}_2{}^*(\mathbf{q}) & \dots & \mathbf{u}_k{}^*(\mathbf{q}) \end{array} \right] \tag{8}$$

where k corresponds to the number of zero singular values of the matrix $\mathbf{J}_{sq}(\mathbf{q}^*)$. The singularity conditions $\mathbf{q}^*$ are then substituted in $\mathbf{J}_r(\mathbf{q})$ and a new matrix $\mathbf{J}^\dagger(\mathbf{q})$ is generated as follows:

$$\mathbf{J}^\dagger(\mathbf{q}) = [\mathbf{U}^*(\mathbf{q})]^T \mathbf{J}_r(\mathbf{q}^*) \tag{9}$$

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: 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 $\mathbf{J}_r(\mathbf{q})$: The set of solution branches followed up to this point is obviously part of $\mathcal{S}$ because the number of redundant columns is insufficient to cancel entirely the nullspace of $[\mathbf{J}_{sq}(\mathbf{q})]^T$.

3. The last redundant column of the matrix $\mathbf{J}(\mathbf{q})$ has been used in $\mathbf{J}_{sq}(\mathbf{q})$: 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 locus of the terminal branch and that of all of its parents is added to the 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 (and all combinations of branches at every node) of the solution tree have been investigated, $\mathcal{S}$ then contains the entire rank-deficiency locus of the rectangular Jacobian.

4. Conclusion

This paper has presented a novel algorithm to compute the rank-deficiency locus of rectangular Jacobian matrices. Although this has not been demonstrated in the paper, the algorithm has been used succesfully to compute, in symbolic form, the rank-deficiency locus of complex manipulators such as the Space Station Remote Manipulator System, which has seven degrees-of-freedom. It is of a computational efficiency comparable to the algorithm developed by Nokleby and Podhorodeski but it has the additional advantage that it can be used for the case of task spaces that cannot be described using screws and for Jacobians with more rows than columns.

References

[1] 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)

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