An Algorithm for Trajectory Generation in Redundant Manipulators with Joint Transmission Accommodation

Kristopher Wehage[1]** and Bahram Ravani[2]

[1] Jet Propulsion Laboratory, Pasadena CA 91109, USA,
kristopher.t.wehage@jpl.nasa.gov
[2] University of California – Davis, Davis, CA 91516,
bravani@ucdavis.edu

Abstract. Trajectory generation for manipulators involves generating incremental updates of joint variables to achieve a desired end–effector motion. The Jacobian matrix maps incremental joint motion to incremental end–effector motion in a linear fashion and is typically used in manipulator trajectory generation algorithms. In the case of a redundant manipulator, the Jacobian matrix is not square or invertible and therefore algorithms based on pseudoinverses and their variations are commonly used for trajectory generation. These methods either are computationally not efficient or do not utilize all the joints in motion generation and therefore do not completely exploit the redundancy of the manipulator. The method presented in this paper is a simple method that maximizes transmission of all joint variables onto a desired end–effector motion trajectory. The method is based on aligning the null–space of an augmented Jacobian matrix with the path of the desired end–effector motion, from which a linear combination of joints that projects fully onto the desired end–effector trajectory is obtained. In this manner, all joints of the redundant manipulator are used to generate the end–effector trajectory accommodating the ability of each joint in terms of its motion transmission.

Keywords: Redundant Manipulators, Trajectory Generation, Differential Kinematics, Transmission Accommodation.

1 Introduction

Redundant manipulators with 7 or 8 Degrees Of Freedom (DOF) are finding increased applications in industrial settings (see, for example [1, 2]) due to their increased dexterity and ability to avoid bifurcations and singular configurations. Kinematic algorithms used for trajectory generation linearly relate the incremental joint motions to the desired end-effector motion from a nearby location

** This work was done as a private venture and not in the author's capacity as an employee of the Jet Propulsion Laboratory, California Institute of Technology.

using the Jacobian matrix. In general the kinematic equations for a single arm serial manipulator can be written as $f(\underline{q}, \Phi_{fe}) = 0$ where $\underline{q}$ is the vector of joint variables and Φ_{fe} represents a parametrization of the end-effector configuration which is typically described as a screw or an element of the special Euclidean group SE(3). The matrix Φ_{fe} indicates an active transformation from an inertial world frame f to the end-effector frame e. In incremental trajectory generation for a manipulator, a linearized form of this equation is obtained using Taylor series expansion and neglecting the higher order terms:

$$f(\underline{q}, \delta\underline{q}, \Phi_{fe}) = f(\underline{\bar{q}}, \Phi_{ef}) + J\delta\underline{q} = 0 \tag{1}$$

In the above equation, the term $f(\underline{\bar{q}}, \Phi_{fe})$ represents the incremental movement of the end-effector from its existing location to a nearby location. If the end-effector nearby location is specified as Φ^*_{fe} (see for example, [3], pp. 164-168), then the position error E can be expressed as:

$$E = \Phi^*_{fe}\Phi^{-1}_{fe} - I \tag{2}$$

The non-zero terms of E when put in screw coordinate can be represented as $\underline{\mathbf{e}}$. The linearized kinematic equations then become:

$$\underline{\mathbf{e}} = J\delta\underline{q} \tag{3}$$

For redundant manipulators the Jacobian matrix J is rank deficient or not a square matrix. A common technique to resolve rank deficiency is to take the generalized inverse or MoorePenrose pseudo-inverse of the Jacobian as:

$$\delta\underline{q} = (J^T J)^{-1} J^T \underline{\mathbf{e}} = J^+ \underline{\mathbf{e}} \tag{4}$$

which effectively finds a non-linear least squares solution [4] with the terms that are minimed being quadratic in joint velocities and therefore proportional to the kinetic energy of the system. A variation of the generalized inverse method is the use of weighted or damped least squares method:

$$\delta\underline{q} = (J^T W J + W)^{-1} J^T W \underline{\mathbf{e}} \tag{5}$$

where W is a diagonal matrix of weighting factors [5–7]. The weights can also be selected based on the inertia of the links [8]. The generalized inverse methods, in general, not only require many additional floating point operations (FLOPS) but may also sometimes fail [3]. Despite these shortcomings such methods and their variations are still widely used in robotic literature for kinematic control or incremental trajectory generation of redundant manipulators (see, for example, [9–11]).

In this paper, we present a method that at each configuration of the manipulator numerically determines how well each joint is aligned with a direction orthogonal to the motion of the end-effector from its current position to the desired nearby configuration. The approach aligns the tangent vector in the constraint manifold of the manipulator with the particular path of interest and results in a method that utilizes all the joints maximizing their motion transmission abilities.

2 Incremental Trajectory Generation

In incremental or kinematic trajectory generation for robot manipulators, the motion path of the end-effector is first parameterized in terms of successive incremental orientations and positions. There are a number of approaches for this process in the literature including using methods from Computer Aided Geometric Design such as quaternion based spherical interpolation (SLERP) [12], smooth invariant interpolation of rotations [13], or cubic Hermite splines [14] to name a few techniques. The differential kinematic equations are then used to map each successive motion path of the end-effector to the corresponding joint variables, thereby incrementally updating the joint coordinates to move the robot through its desired trajectory.

The differential kinematic equations map joint velocities to the end-effector velocity using the Jacobian matrix. As stated earlier the Jacobian matrix for a redundant manipulator is not full rank. It can be permuted and block-partitioned into coefficients associated with its primary and secondary constraint equations and dependent and independent joint coordinates [15]. The method presented in this paper does not require an analyst to select dependent and independent joint variables or primary and secondary constraint equations but rather automatically obtains the optimal split for each of its incremental configurations throughout the trajectory generation process.

Applying LU factorization with complete row and column pivoting permutes and partitions the Jacobian matrix J into the following form:

$$P_R J P_C^T = \begin{bmatrix} J^{pd} & J^{pi} \\ J^{sd} & J^{si} \end{bmatrix} \tag{6}$$

and factors the largest nonsingular submatrix J^{pd}, into L and U. Here P_R and P_C^T are row and column permuation matrices, the superscripts p and s indicate terms associated with primary and secondary constraint equations, and the subscripts i and d indicate terms associated with independent and dependent variables. Applying LU decomposition with complete pivoting to the differential kinematic equations yields:

$$\begin{bmatrix} J^{pd} & J^{pi} \\ J^{sd} & J^{si} \end{bmatrix} \begin{Bmatrix} \delta \underline{q}^d \\ \delta \underline{q}^i \end{Bmatrix} = \begin{Bmatrix} -\underline{\mathbf{e}}^p \\ -\underline{\mathbf{e}}^s \end{Bmatrix} \tag{7}$$

The independent joint coordinates $\underline{q}^i$ may be set to any arbitrary value and a change in dependent joint coordinates $\underline{q}^d$ can be found from the row-space as:

$$\delta \underline{q}^d = -U^{-1} L^{-1} (\underline{\mathbf{e}}^p + J^{pi} \delta \underline{q}^i) \tag{8}$$

where L and U are the LU factors of J^{pd}. The incremental changes in the joint variables, $\delta \underline{q}^i$, exert the smallest over-all effect on a change in incremental configuration of the manipulator and therefore can be used in accommodation of best transmission advantage in resolving redundancy as discussed in the next section. Uicker et al. [3] suggest a simple way of solving Eq. (8) to resolve redundancy.

The process involves setting $\delta \underline{q}^i$ equal to zero and solving the reduced system of equations:

$$\delta \underline{q}^d = -U^{-1} L^{-1} \underline{\mathbf{e}}^p \tag{9}$$

Unfortunately this approach does not fully exploit the redundancy of the manipulator because setting $\delta \underline{q}^i = \underline{0}$ makes the manipulator nonredundant in terms of its independent variable. The approach presented in this paper re-establishes the distribution of independent and dependent joint variables to achieve the best transmission accommodation at each incremental configuration.

3 New Redundancy Resolution Method

The new redundancy resolution method begins with computing the right null-space of an augmented Jacobian matrix J with one column appended to represent the desired instantaneous screw motion of the end-effector. A convenient representation of the row-space of J is found from:

$$L^{-1} U^{-1} \left[J^{pd} \; J^{pi} \right] = \left[I \; B^{di} \right] \tag{10}$$

which allows trivially computing the right null-space of J, N, as:

$$N = \begin{bmatrix} -B^{di} \\ I \end{bmatrix} \tag{11}$$

The LU decomposition with complete pivoting algorithm overwrites the row–space of the matrix J with its LU factors and the residual matrix U_R as:

$$\left[LU \; U^{-1} J^{pi} \right] = \left[LU \; U_R \right] \tag{12}$$

The quantity $-B^{di}$ is computed efficiently using backward substitution as:

$$-B^{di} \leftarrow -L^{-1} U_R \tag{13}$$

QR decomposition [16] is then applied to the matrix N as:

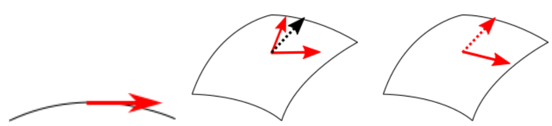
$$N = ZR = \begin{bmatrix} Z^{di} \\ R^{-1} \end{bmatrix} R = \begin{bmatrix} Z^{di} \\ Z^{ii} \end{bmatrix} R \tag{14}$$

In Eq. 14, R is a transformation matrix and Z is an orthonormal basis. The matrix R effectively takes a linear combination of directions along the manipulator's constraint manifold and orthogonalizes and normalizes the columns of Z. The performance of the LU decomposition in the above computations can be enhanced using a heuristic application of complete pivoting as described in [15].

The resulting Z matrix is analogous to a unit orthogonal direction-cosine matrix on the manipulator's constraint manifold where each column corresponds to a direction along the manifold and each row corresponds to unique joint variable. Each of the entries in Z corresponds to a projection of a joint onto one orthogonal direction tangent to the constraint manifold of the manipulator configuration. The matrix Z is partitioned into Z^{di} and Z^{ii} (see Eq. (14)) with each row corresponding, respectively, to a unique dependent or independent joint variable. The null-space basis Z of the Jacobian matrix, therefore, represents two orthogonal tangent vectors on the constraint manifold of the manipulator, which is schematically illustrated in Fig. 3. At this stage neither of the two tangent vectors is aligned with the desired path of the end-effector which is shown by the black dashed line in Fig. 1(b). The necessary information to align the null-space basis of the Jacobian of the manipulator to the desired path is found in the last row of Z. When the tangentspace is aligned with the desired endeffector path, the entry in the first column of the row associated with the path will project fully onto the path. The entry in the corresponding second column will be zero. A 2×2 orthonormal rotation matrix, C, can therefore be applied to rotate Z to align the tangent vectors as:

$$Z' = ZC \tag{15}$$

Note that both Z and Z' matrices lie in the right null-space of the Jacobian matrix J and represent a set of orthonormal basis vectors aligned with the mechanism's constraint manifold.



(a) (b) (c)

(a) a one-dimensional manifold; (b) a two dimensional manifold with an arbitrary set of tangent vectors; (c) a two dimensional manifold with one tangent vector aligned with path direction

Fig. 1. Basis vectors along one and two-dimensional manifolds.

The matrix Z' now has the first column aligned with the target end-effector path. The first column represents a linear combination of the joint velocities that fully project onto the instantaneous screw motion of the path. The second column of Z' represents a tangent vector that is orthogonal to the target path. The linear combination of joint velocities indicated in the second column of Z' imparts no motion along the target path. Once the first column of Z' has been aligned to the target path, the entries in the first column indicate how to proportion joint velocities to impart maximum transmission onto the path. The method is illustrated by an example in the next section.

4 Example

This section provides a brief illustration of the method for redundancy resolution by applying it to a 7R serial manipulator depicted in Fig. 2 consisting of seven revolute joints labeled A through G. In order to determine the Jacobian matrix for this manipulator the shape matrices (see [3]) are formed. The numerical values of these matrices are presented in Eq. 16.

$$\Phi^{1h^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.045 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Phi^{2h^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.09 \\ 0 & 0 & 1 & 0.02 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Phi^{3h^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.085 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Phi^{4h^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.125 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$\Phi^{5h^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -0.085 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Phi^{6h^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.125 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Phi^{7h^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.085 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

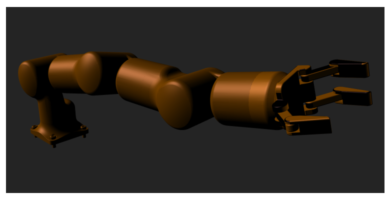


Fig. 2. A 7R redundant manipulator

The augmented Jacobian matrix J for the 7R manipulator is given as:

$$J = \left[\underline{\mathbf{h}}_A \; \underline{\mathbf{h}}_B \; \underline{\mathbf{h}}_C \; \underline{\mathbf{h}}_D \; \underline{\mathbf{h}}_E \; \underline{\mathbf{h}}_F \; \underline{\mathbf{h}}_G \middle| \underline{\mathbf{h}}_{path} \right] \tag{17}$$

where the terms $\underline{\mathbf{h}}_{A-G}$ represent the influence coefficient matrices associated with revolute joints A–G, and the quantity $\underline{\mathbf{h}}_{path}$ represents the influence coefficient matrix associated with the instantaneous screw motion of the end-effector.

Inserting numerical values for a specific configuration yields:

$$J = \left[\begin{array}{ccccccc|c} 0.0000 & -0.0134 & -0.1356 & -0.0256 & 0.1756 & -0.0123 & -0.0936 & 0.2853 \\ 0.0000 & 0.0200 & -0.0000 & 0.1821 & 0.0073 & -0.3956 & -0.0074 & 0.6905 \\ 0.0000 & -0.1343 & 0.0136 & 0.1766 & -0.0106 & -0.1770 & 0.0040 & -0.0611 \\ 0.0000 & 0.9950 & -0.0978 & -0.9791 & 0.0604 & 0.9981 & -0.0584 & 0.2548 \\ 1.0000 & 0.0000 & 0.1986 & -0.1947 & -0.0038 & -0.0040 & 0.2024 & -0.6369 \\ 0.0000 & -0.0998 & -0.9751 & 0.0585 & 0.9981 & -0.0604 & -0.9775 & -0.1671 \end{array} \right] \tag{18}$$

An orthonormalized null–space basis for J is obtained as

$$Z = \begin{array}{r} A \\ B \\ C \\ D \\ E \\ F \\ G \\ path \end{array} \begin{bmatrix} -0.2296 & 0.1972 \\ 0.0255 & -0.0278 \\ -0.3567 & -0.7338 \\ -0.5108 & 0.2346 \\ 0.1345 & -0.5346 \\ -0.5013 & 0.2115 \\ 0.5165 & 0.1783 \\ -0.1491 & 0.0677 \end{bmatrix} \tag{19}$$

As discussed previously, the null-space basis in Eq. (19) is not unique and needs to be rotated to align the tangent vectors.

$$Z' = ZC = Z \begin{bmatrix} -0.9107 & -0.4132 \\ 0.4132 & -0.9107 \end{bmatrix} = \begin{array}{r} A \\ B \\ C \\ D \\ E \\ F \\ G \\ path \end{array} \begin{bmatrix} 0.2906 & -0.0847 \\ -0.0348 & 0.0148 \\ 0.0217 & 0.8156 \\ 0.5621 & -0.0026 \\ -0.3434 & 0.4313 \\ 0.5439 & 0.0146 \\ -0.3967 & -0.3758 \\ 0.1638 & 0. \end{bmatrix} \tag{20}$$

The first column of Z' represents a linear combination of joint velocities AG that project fully onto the instantaneous screw motion of the path. The combination of quantities in the second column of Z' reconfigure the robot while maintaining the artificial constraint at the end–effector. Therefore, the first column of Z' represents the linear combination of joint variables that maximizes projection onto the desired trajectory with no wasted effort orthogonal to the desired trajectory.

$$\delta \underline{q}_{A-G} = \left\{ Z'_{[A,0]} \; Z'_{[B,0]} \; Z'_{[C,0]} \; Z'_{[D,0]} \; Z'_{[E,0]} \; Z'_{[F,0]} \; Z'_{[G,0]} \right\}^T \frac{\delta q_{path}}{Z'_{[path,0]}} \tag{21}$$

With this information, a first-order update of the joint variables $\delta \underline{q}$ can be applied using Eq. (21). It should be noted that this approach requires approximately 40% fewer floating–point operations than the pseudo-inverse and least squares methods [15, 16].

5 Conclusions and Future Work

The method for redundancy resolution presented in this paper represents a greedy algorithm for obtaining an incremental update of joint variables at any time interval. It is greedy in the sense that it maximizes transmission of joint variables onto the desired trajectory at a given time interval with no consideration of the state of the manipulator at the next time interval(s). It should be noted that it is possible that the optimal projection onto the desired trajectory at one time interval may put the robot in a configuration where one or more of the joints may have a low ability to impart transmission onto the desired trajectory at a later time interval. The work presented here may be further extended by developing optimal robot path tracking algorithms (i.e. following a specified path while minimizing an objective function), which will be investigated in the authors' future work.

6 Acknowledgements

This work was supported by the International Research Training Group (IRTG2057) with the University of Kaiserslautern, Germany.

References

1. E. Guizzo and E. Ackerman. How rethink robotics built its new baxter robot worker. *IEEE Spectrum*, 2012.
2. K. Kreutz-Delgado, M. Long, and H. Seraji. Kinematic analysis of 7-dof manipulators. *The International journal of robotics research*, 11(5):469–481, 1992.
3. J.J. Uicker, B. Ravani, and P.N. Sheth. *Matrix Methods in the Design Analysis of Mechanisms and Multibody Systems*. Cambridge University Press, 2013.
4. D.E. Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 1969.
5. Y. Nakamura and H. Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of dynamic systems, measurement, and control*, 108(3):163–171, 1986.
6. C.W. Wampler and L.J. Leifer. Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 110(1):31–38, 1988.
7. H. Seraji and R. Colbaugh. Improved configuration control for redundant robots. *Journal of Robotic Systems*, 7(6):897–928, 1990.
8. J. Wang, Y. Li, and X. Zhao. Inverse kinematics and control of a 7-dof redundant manipulator based on the closed-loop algorithm. *International Journal of Advanced Robotic Systems*, 7(4):1–9, 2010.
9. F. Branz, A. Antonello, A. Carron, R. Carli, and A. Francesconi. Kinematics and control of redundant robotic arm based on dielectric elastomer actuators. In *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, pages 943023–943023. International Society for Optics and Photonics, 2015.
10. A. Müller. A robust inverse dynamics formulation for redundantly actuated pkm. In *13th World Congress in Mechanism and Machine Science, Guanajuato, Mexico*, pages 19–25, 2011.
11. J.M. Hollerbach and K.C. Suh. Redundancy resolution of manipulators through torque optimization. *Robotics and Automation, IEEE Journal of*, 3(4):308–316, 1987.
12. K. Shoemake. Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics*, volume 19, pages 245–254. ACM, 1985.
13. F.C. Park and B. Ravani. Smooth invariant interpolation of rotations. *ACM Transactions on Graphics (TOG)*, 16(3):277–295, 1997.
14. X.F. Zha and H. Du. Generation and simulation of robot trajectories in a virtual cad-based off-line programming environment. *The International Journal of Advanced Manufacturing Technology*, 17(8):610–624, 2001.
15. K.T. Wehage, R.A. Wehage, and B. Ravani. Generalized coordinate partitioning for complex mechanisms based on kinematic substructuring. *Mechanism and Machine Theory*, 92:464–483, 2015.
16. G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.