CONF-960448--4

SAND95-2196C

Trajectory Generation For Two Robots Cooperating To Perform A Task

Christopher L. Lewis
Sandia National Labs
PO Box 5800, Mail Stop 0951
Albuquerque, NM 87185-0951

Abstract

This paper formulates an algorithm for trajectory generation for two robots cooperating to perform an assembly task. Treating the two robots as a single redundant system, this paper derives two Jacobian matrices which relate the joint rates of the entire system to the relative motion of the grippers with respect to one another. The advantage of this formulation over existing methods is that a variety of secondary criteria can be conveniently satisfied using motion in the null-space of the relative Jacobian. This paper presents methods for generating dual-arm joint trajectories which perform assembly tasks while at the same time avoiding obstacles and joint limits, and also maintaining certain constraints on the absolute position and orientation of the end-effectors.

1 Introduction

The Jacobian matrix has gained wide use for generating trajectories of a prescribed geometry relative to a fixed coordinate system for redundant and non-redundant manipulators [1, 2, 3, 4]. For redundant manipulators, the concepts of linear algebra have provided a means of satisfying secondary criterion via motion in the null-space of the Jacobian matrix [5, 6]. These criteria include: singularity avoidance, manipulability optimization, and obstacle avoidance, to name a few. Specifying a cooperative assembly task for the two robots as a trajectory in the world coordinate frame reduces the kinematic control problem to that of controlling each robot individually [5, 7, 8, 9]. This technique, while common, does not make use of the inherent flexibility of the combined system, i.e. it ignores a large degree of redundancy. Using this method, secondary criteria may still be satisfied by utilizing off-line path planning techniques which in effect choose a better specification for the task in absolute coordinates. If, however, the robots are treated as a single system and joint trajectories are generated using a Jacobian which relates the velocity of one gripper relative to the other gripper's coordinate frame, then the assembly can be described in a part relative frame which is much more natural and easily derived from the CAD model of the assembly, [10]. Another advantage to treating the dual-arm system as a single redundant kinematic system is that the joint values may be optimized using null-space projection techniques and if done on line the optimization can incorporate sensory data [11].

This paper defines two related techniques for generating the joint trajectories for two robots to execute tasks in part relative space. In the first technique, a relative Jacobian $J_R \in R^{6\times N}$, is defined relating the joint velocities to the relative motion between the grippers. In the second, a related Jacobian matrix, $R_{RD} \in R^{4\times N}$, is defined that relates the joint velocities to the rate of change in the distance and the relative angular velocity between the end-effectors. From these two Jacobians and using null-projection techniques, algorithms are developed which can satisfy the primary assembly task, as well as secondary goals such as collision, obstacle and joint limit avoidance and world frame orientation constraints.

2 Relative Jacobians

The Jacobian for a single robot may be computed from the Denavit-Hartenberg parameters [4, 12]. In this section, a similar procedure is employed to form the relative Jacobian, J_R and the relative distance Jacobian J_{RD}. Then in the next section these Jacobians are used to define algorithms to generate joint trajectories for dual-arm assembly tasks.

As with computing the Jacobian for a single robot, the inverse of the tool transformation is constructed incrementally using the link transformations, $A_i(a, d, \alpha, \theta)$. Then, from each intermediate transformation, a single column of the Jacobian is computed.

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

The tool transformation considered here, T_R, relates vectors in the second gripper's frame to the first gripper's coordinate frame. Its inverse can be given in terms of successive transformations, as:

$$T_R^{-1} = Tl_2^{-1} A2_{n_2}^{-1} \ldots B2^{-1} B1 \ldots A1_{n_1} Tl_1 \quad (1)$$

where, Aj_i is the ith link transformation of the jth robot, $B1$ and $B2$ are the transformations to the bases of manipulators 1 and 2 respectively. Similarly Tl_1 and Tl_2 are the tool transformations.

To compute the relative Jacobian, the following matrices U_i, representing the transformations from the second robot's gripper frame to each link's frame, are formed:

$$\begin{array}{rcl} U_{n_1+n_2} & = & Tl_2^{-1} A2_{n_2}^{-1} \\ U_{n_1+n_2-1} & = & U_{n_1+n_2} A2_{n_2-1}^{-1} \\ \vdots & = & \vdots \\ U_{n_1+1} & = & U_{n_1+2} A2_1^{-1} \\ U_1 & = & U_{n_1+1} B2^{-1} B1 \\ U_2 & = & U_1 A1_1 \\ \vdots & = & \vdots \\ U_{n_1} & = & U_{n_1-1} A1_{n_1-1}. \end{array} \quad (2)$$

For notational convenience, the columns of the U_i matrices are delineated as:

$$U_i = \left[\begin{matrix} n_i & o_i & a_i & p_i \\ 0 & 0 & 0 & 1 \end{matrix}\right]. \quad (3)$$

The columns of the relative Jacobian, J_R, are composed of two 3-vectors, v_i, and ω_i, the relative linear velocity and the relative angular velocity of the second gripper respectively. In the second end-effector's frame these vectors are given by:

$$\omega_i = \begin{cases} 0 & \text{joint } i \text{ prismatic} \\ -a_i & i \leq n_1 \\ a_i & i > n_1 \end{cases} \quad (4)$$

and

$$v_i = \begin{cases} a_i & \text{joint } i \text{ prismatic, } i > n_1 \\ -a_i & \text{joint } i \text{ prismatic, } i \leq n_1 \\ \omega_i \times -p_i & \text{joint } i \text{ rotational} \end{cases}. \quad (5)$$

The relative Jacobian, J_R, is obtained by multiplying the v_i's and ω_i's by the upper 3×3 rotational submatrix, Rot_R of T_R as follows:

$$J_R = \begin{bmatrix} Rot_R & 0 \\ 0 & Rot_R \end{bmatrix} \begin{bmatrix} v_1 & \cdots & v_{n_1+n_2} \\ \omega_1 & \cdots & \omega_{n1+n2} \end{bmatrix}. \quad (6)$$

The relative distance Jacobian matrix, $J_{RD} \in R^{4 \times N}$, is derived from J_R by projecting the linear portion onto the normalized distance vector and copying the rotational portion as follows:

$$J_{RD} = \begin{bmatrix} \hat{p}_R^T & 0_{1\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} J_R \quad (7)$$

where $\hat{p}_R$ is the normalized position vector of T_R.

Another Jacobian matrix that will be used later is the upper $3 \times N$ sub-matrix of the relative Jacobian. The relative position Jacobian is given by:

$$J_{RP} = [I_{3\times3} \quad 0_{3\times3}] J_R \quad (8)$$

and relates the joint rates to the relative linear velocity of the second gripper with respect to the first gripper's frame.

The projections onto the null-space of J_R, J_{RD}, and J_{RP} will be denoted P_{JR}, P_{JRD}, and P_{JRP} respectively. These matrices are computed efficiently from the singular value decompositions of their respective matrices [13].

3 Assembly Via Relative Motion

The task of screwing a nut onto a bolt may be described as the alignment of the nut and bolt's central axis followed by linear motion along this axis until they contact. The nut is then rotated around the central axis while continuing the linear motion at a rate equal to the thread pitch. Descriptions of assembly tasks such as this do not include any reference to the absolute position or orientation of either part. The Jacobians defined above are ideally suited for generating joint trajectories from part-relative assembly descriptions such as this. The algorithms described here will initiate as soon as the individual parts are grasped by the robots and will proceed in a two phase process. The first phase of motion is to bring the parts into an approach position. This phase will use the relative distance Jacobian to solve for the motion. The second phase is the mating phase, where the two parts are brought into contact with one another in an assembly. This phase will use the relative Jacobian to solve for the motion.

The kinematic equations governing the mating phase are given by:

$$\dot{\theta}_R = J_R^+ \begin{bmatrix} v_R \\ \omega_R \end{bmatrix} + \dot{\theta}_H \quad (9)$$

where J_R^+ is the pseudo-inverse of the relative Jacobian and v_R and ω_R are the desired relative linear and angular velocities of the part held in the second robot's gripper with respect to the part held in the first robot's gripper and $\dot{\theta}_H$ is joint motion in the null-space of J_R. The next section will discuss θ_H. Part mating tasks are typically described by a sequence of relative transformations and the error is used in (9) to iteratively find a solution.

A typical first phase task definition includes the world frame transformation of each of the parts as they are initially grasped and a relative transformation having the parts ready to mate. The task then becomes to move the objects from the initial to their final relative position/orientation. Equation (9) could also be used to generate this type of trajectory, but experience has shown significant drawbacks to this approach. In particular, a straight line in relative coordinates from the initial to the final transformation will often force a collision between the end-effectors. A simple example of this is shown in Fig. 1 where two objects need to be reversed before an assembly can be achieved. The relative distance Jacobian was developed to alleviate this problem. The kinematic equations governing the first phase are given by:

$$\dot{\theta}_{RD} = J_{RD}^{+} \begin{bmatrix} \dot{d} \\ \omega \end{bmatrix} + P_{RD} J_{RP}^{+} v_R + \dot{\theta}_H \qquad (10)$$

where J_{RD}^{+} is the pseudo-inverse of J_{RD}, $\dot{d}$ is the desired rate of change in the distance between the grippers, ω is the desired relative angular velocity, J_{RP}^{+} is the pseudo-inverse of the J_{RP}, v_R is the desired relative linear velocity and $\dot{\theta}_H$ is again joint motion in the null-space of J_R. When moving the system from the initial relative position/orientation to the final one, the first term, $J_{RD}^{+} \begin{bmatrix} \dot{d} \\ \omega \end{bmatrix}$, provides motion to drive the end-effectors to their final separation distance, and orientation. The second term, acting in the null-space of J_{RD} does not change d and provides motion to meet the final relative positioning requirement. If the separation distance criterion is already satisfied this term will move the second gripper relative to the first along the sphere having radius equal to that separation distance. Motion constrained to the sphere is essentially a rotation of the first gripper as seen by an observer in the world frame.

4 Self-Motion to Satisfy Constraints

While equations (9) and (10) generate satisfactory joint trajectories for some simple assembly task, other assembly tasks require that secondary criteria be met. Many assembly operations require constraints on the absolute orientation of the pieces. For instance, a bolt with a washer on it may not be tipped upside down without the washer falling off. Also, whenever two robots are required to be in close proximity, as is true for most assembly operations, collision avoidance is necessary. In previous approaches these type of criteria could only be addressed by judicious placement of the task in world coordinates. In this paper, secondary criteria are easily met using null-space projection techniques commonly used with redundant systems. Motion in the null-space of J_R as seen by an observer in the world frame appears as though the two end-effectors were both gripping a large invisible rigid object. This type of motion is often referred to as self-motion. The remainder of this section will describe a variety of criteria and how they are combined. The technique involves solving a least squares solution to a scaled combination of all the secondary criteria constrained to lie on the tangent plane of the self-motion manifold.

Using the results of [6] the desired self-motion will be given by:

$$\dot{\theta}_H = P_{JR} J_C^{+} \dot{z}_H \qquad (11)$$

where J_C is a composite Jacobian that relates the joint rates to the composite desired beneficial motion, $\dot{z}_H$. The composite Jacobian is formed by augmenting any number of sub-Jacobians. Each sub-Jacobian, Ji, relates the joint rates to a specific beneficial motion, $\dot{z_H i}$. Here, three types of beneficial motion are considered: motion to avoid collisions, motion to avoid joint limits, and motion to control the absolute position and orientation of the end-effectors. The three types of sub-Jacobians which make up the composite Jacobian are the obstacle-avoidance Jacobian, the joint limit Jacobian and the absolute Jacobian denoted Ji_O, Ji_L, and Ji_A, respectively. Similarly, the associated beneficial velocities are denoted $\dot{z}i_O$, $\dot{z}i_L$, and $\dot{z}i_A$. A typical composite Jacobian equation is given by

$$\dot{\theta}_H = P_{RJ} \begin{bmatrix} J1_O \\ J2_O \\ J3_L \\ J4_L \\ J5_A \\ J6_A \end{bmatrix}^{+} \begin{bmatrix} \dot{z}1_O \\ \dot{z}2_O \\ \dot{z}3_L \\ \dot{z}4_L \\ \dot{z}5_A \\ \dot{z}6_A \end{bmatrix} \qquad (12)$$

where the subscripts delineate the type of Jacobian.

To cause a link to avoid an obstacle, it is necessary to solve for a particular set of joint rates which give the link a velocity away from the obstacle. It may be discerned from Fig. 2 that the absolute Cartesian motion of a link is affected only by joints which are closer to the base. Thus the obstacle Jacobian, denoted Ji_O, associated with link s of the second robot

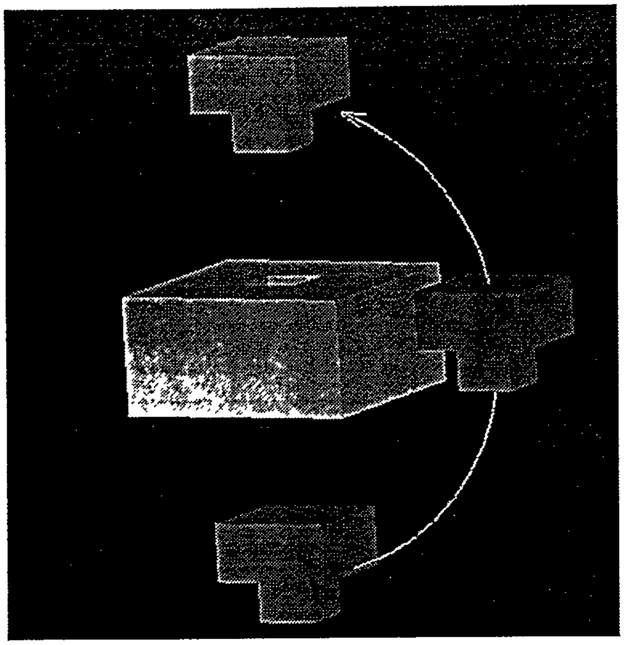

Figure 1: Rotations are often preferable to linear motion in an assembly task.

is given by the $3 \times (n_1 + n_2)$ matrix

$$Ji_O = [\underbrace{0|0|\cdots|0}_{robot1}|\underbrace{v_1|v_2|\cdots|v_s|0|0|\cdots|0}_{robot2}]. \quad (13)$$

The vectors v_i are obtained from the a_i and p_i columns of the homogeneous transforms used to sequentially build the transformation from the world frame to the link frame. In this case the desired velocity is a vector pointing away from the nearest obstacle given by:

$$\dot{z}i_O = \frac{-\hat{n}}{d} \quad (14)$$

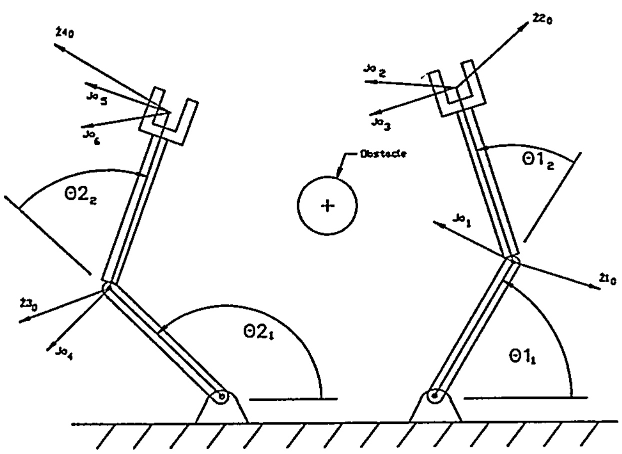

Figure 2: The Obstacle Avoidance Vectors

where, d, is the distance between the obstacle and the link. The vector $\dot{z}i_O$ has a magnitude inversely proportional to the distance. A vector sum blends the effects of more than one obstacle.

The relationship which allows a joint to avoid its mechanical limit is trivial. The only joint which affects the motion of the ith joint is the ith joint. Therefore, a joint limit avoidance Jacobian consist of a single row of all zeros except for the ith element which is a 1. The resulting $1 \times (n_1 + n_2)$ joint limit Jacobian is

$$Ji_L = [0|0|\cdots|0|1|0|\cdots|0]. \quad (15)$$

The desired velocity, $\dot{z}i_L$, may be derived from smooth functions that increase near joint limits [6]. Whenever $\dot{z}i_L$ becomes small, Ji_L may be dropped from the composite Jacobian. This dropping reduces the computation required and also allows other more important criterion to be met more fully.

The individual robot's world frame manipulator Jacobian equation may be used to control the absolute position and orientation of the grippers. A manipulator Jacobian relates the joint rates of the system to the absolute motion of the end-effector. As in the obstacle avoidance Jacobian, none of the joints of the opposite robot affect the absolute motion of the end-effector and their corresponding columns in an absolute Jacobian are zero. The remaining columns of an absolute Jacobian are obtained by inserting the full $6 \times n$ manipulator Jacobian, J_{m2}. For absolute constraints on the second robot the $6 \times (n_1+n_2)$ absolute Jacobian denoted Ji_{A2} is

$$Ji_{A2} = \Big[\underbrace{[0]}_{robot1} | \underbrace{[J_{m2}]}_{robot2}\Big]. \quad (16)$$

The velocities associated with the absolute Jacobian contain both linear velocities v_{m2} and rotational velocities ω_{m2}. In general, absolute constraints are specified by a time varying transform T_d which specifies the desired coordinate frame of the gripper. The velocity in this case is the generalized position and orientation error.

5 Simulations and Results

The algorithms described were implemented in C. These programs require the D-H parameters for each robot and the trajectory information. Unlike previous methods the trajectories for many assembly tasks are easily defined. Often, the initial placement of the parts in the work-cell, a relative approach configuration and the final assembly configuration are all that is required. With this information, these techniques automatically generate trajectories in joint space which perform the task while avoiding collisions, joint limits and may satisfy world frame requirements.

This technique has proven to be effective in generating dual-robot trajectories for a variety of typical assembly tasks. The mating operation of two parts shown in Fig 3 was successfully performed. Using the relative distance formulation the robots first bring the two parts to a relative approach configuration having their axes aligned but separated enough to avoid collisions. In the approach phase the relative distance formulation was found to be very effective in achieving the goal without causing collisions between the grippers. Then the relative Jacobian formulation was used to mate the two parts.

To demonstrate the concept of relative motion, a sequence of relative task points that form a square was devised and the program was run for the two robots shown in Fig. 3. The resulting path of the second end-effector as viewed by the first gripper's frame and as viewed in the world frame is displayed in Fig. 4 and Fig. 5 respectively. The relative view of the trajectory is a square. The world view is an open curve through space. If closed paths are desired additional self-motion may easily be added in a manner similar to joint limit avoidance.

The algorithms have also proven to be effective in satisfying world frame criteria. Sub-assemblies that have gravitational dependencies, such as a washer on a bolt, require the world frame orientation constraint to be satisfied. In this case the composite Jacobian contains two rows pertaining to the pointing and associated angular velocity to maintain gravity alignment of the part.

6 Summary

A procedure for automatically generating the joint trajectories for two robots to perform assembly operations has been presented. The method treats the two robots as a single redundant system. Two Jacobian equations were defined relating the joint rates of the whole system to the relative motion of the second gripper as seen by the first gripper's coordinate frame. The first Jacobian related the system joint rates to the linear and angular velocity of the second robot's end-effector as observed from the frame of the first robot's end-effector. To implicitly favor orientation changes over linear motion, the second Jacobian related the rate of change in the distance between the end-effectors. In this case, motion to meet relative

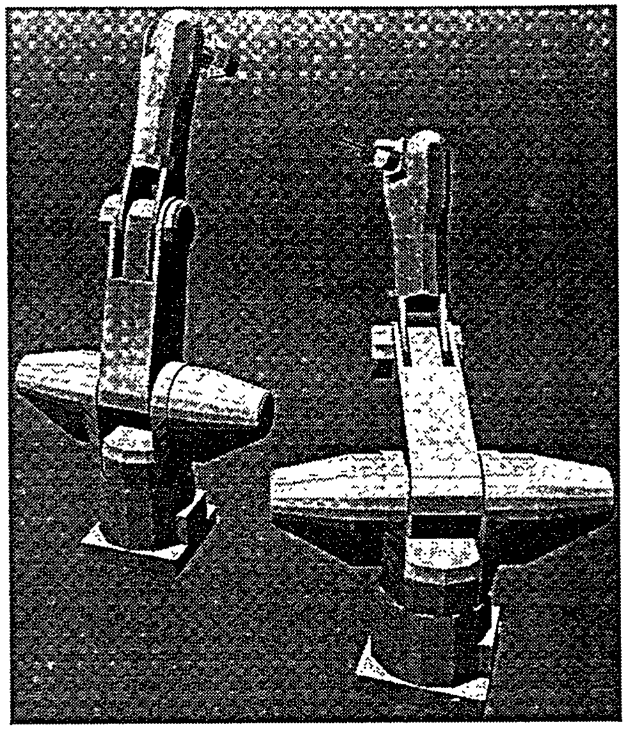

Figure 3: Two robots working as a single 12DOF system may assemble mating parts while at the same time avoiding collisions and joint limits.

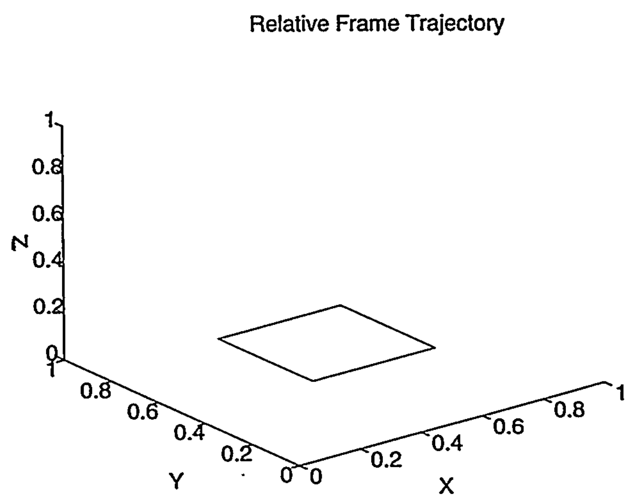

Figure 4: The trajectory of the second gripper is a square when observed in the first gripper's frame.

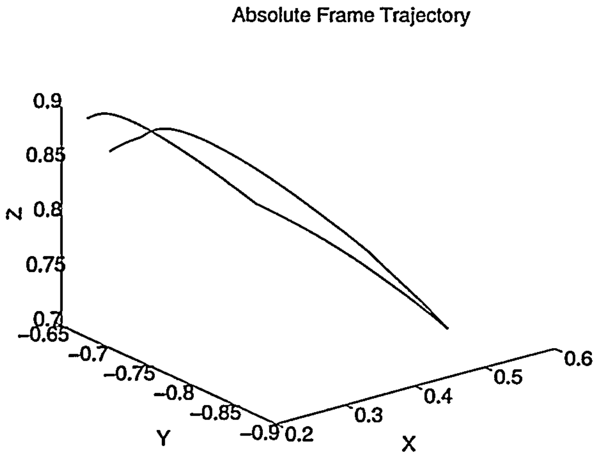

Figure 5: The trajectory of the second gripper is an open curve when observed in the world frame.

positioning requirements was accomplished through null-space projection techniques.

The proposed methods combine the minimum norm solution to the relative Jacobian equations with a least squares solution to a composite Jacobian projected into the null-space of the relative Jacobian. In this way a variety of secondary criteria were met. The algorithm was implemented and the results were presented for two robots performing a typical assembly task. The composite Jacobian provided for collision and joint limit avoidance, and for satisfying absolute orientation constraints. The relative distance Jacobian formulation was shown to be advantageous for the approach phase of the dual-arm assembly process and the relative Jacobian formulation was necessary for exact relative trajectories such as in part mating.

References

[1] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *Journal of Robotic Systems*, vol. 5, no. 6, pp. 527–552, December 1988.

[2] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, September 1986.

[3] D. E. Orin and W. W. Schrader, "Efficient Jacobian determination for robot manipulators," in M. Brady and R. Paul, editors, *Robotics Research: The First International Symposium*, pp. 727–734. MIT Press, 1984.

[4] K. J. Waldron, S. L. Wang, and S. J. Bolin, "A study of the Jacobian matrix of serial manipulators," *Mechanisms, Transmissions, and Automation in Design*, vol. 107, pp. 230–237, June 1985.

[5] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 78–97, February 1989.

[6] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, Fall 1985.

[7] B. Benhabib, G. Zak, and E. Tabarah, "Position control of two-arm manipulators for coordinated point-to-point motion," *Journal of Robotic Systems*, vol. 5, no. 2, pp. 103–124, 1988.

[8] A. Hemami, "Kinematics of two-arm manipulators for coordinated point-to-point motion," *Journal of Robotics and Automation*, vol. RA-2, no. 4, pp. 225–228, 1986.

[9] J. Y. S. Luh and Y. F. Zheng, "Constrained relations between two coordinated industrial robots for motion control," *International Journal of Robotics Research*, vol. 6, no. 3, pp. 60–70, Fall 1987.

[10] R. Zapata, A. Fournier, and P. Dauchez, "True cooperation of robots in multi-arms tasks," *in Proc. 1987 International Conference on Robotics and Automation*, pp. 1255–1260, (Raleigh, NC), March 31-April 3 1987. IEEE.

[11] C. L. Lewis and A. A. Maciejewski, "Trajectory generation for cooperating robots," *in Proc. 1990 International Conference on Systems Engineering*, pp. 300–303, (Pittsburgh, PA), August 9-11 1990. IEEE.

[12] K. J. Waldron, "Geometrically based manipulator rate control algorithms," *Mechanism and Machine Theory*, vol. 17, no. 6, pp. 379–385, 1982.

[13] J. C. Nash, *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation*, A. Hilger, Bristol, 1979.

DISCLAIMER