Trajectory Generation for Cooperating Robots

Christopher L. Lewis   Anthony A. Maciejewski

School of Electrical Engineering
Purdue University,
West Lafayette, Indiana 47907

Abstract

This paper derives a formulation for on-line trajectory generation for two robots cooperating to perform an assembly task. The two robots are treated as a single redundant system. A Jacobian is formulated that relates the joint rates of the entire system to the relative motion of one of the hands with respect to the other. The minimum norm solution of this relative Jacobian equation results in a set of joint rates which perform the cooperative task. In addition to the cooperative task, secondary goals, which include obstacle and joint limit avoidance, are specified using velocities in the null space of the relative Jacobian. This formulation also allows the robots to be controlled in parallel on independent tasks.

I. Introduction

Many researchers have studied the kinematics of two robots cooperating to perform a task [1-3,7,8]. The approaches used can be broadly categorized into two classes. The first is an extension of single robot kinematics in which the joint rate equations for each robot are solved separately. The master-slave paradigm is an example of this type of approach in which the master robot's joint rate equations are solved according to the specification of the task and then the slave robot's joint rate equations are solved to satisfy the constraint equations resulting from the closed kinematic chain [4]. In the second approach, the dual robot system is described by a single set of equations. The task of carrying a rigid object is accomplished by motion in the null space of the Jacobian for the entire system. The redundancy of a system described in this manner may be utilized by optimizing a performance measure along the trajectory [9].

This paper formulates an algorithm for on-line generation of joint trajectories for two robots cooperating to perform an assembly task. The algorithm treats the two robots as a single redundant system and uses the redundancy to avoid obstacles and joint limits as well as to provide for constraints on the absolute orientation and position of the end effectors. Assembly operations are decomposed into two types of tasks, independent and cooperative. Independent tasks are characterized by the control of the absolute position and orientation of the robots to achieve separate but related goals. Collisions are avoided by including constraints on the minimum allowable distance between the robots. Cooperative tasks are characterized by the control of the relative position and orientation between the end effectors. The task of mating two sub-assemblies is a general example of a cooperative task that also requires control of the absolute orientation of the end effectors. In order to perform tasks of this type, two Jacobian equations are required, one relating the joint rates to the relative motion between the end effectors, and the other, a composite Jacobian composed of several Jacobians, relating the joint rates to the constraints on the motion of other parts of the system.

II. Calculating Jacobians

The transformation relating vectors expressed in the second robot's hand coordinate frame to the first robot's hand coordinate frame, denoted $T_{h_1 \leftarrow h_2}$, can be expressed in terms of the homogeneous transformations associated with each link as follows

$$\begin{aligned} T_{h_1 \leftarrow h_2} &= \begin{bmatrix} R_{h_1 \leftarrow h_2} & p \\ 0 & 1 \end{bmatrix} \\ &= B_{n_2} \cdots B_0 A_0 \cdots A_{n_1} \end{aligned} \tag{1}$$

where A_i represents the homogeneous transformation from coordinate frame $i-1$ to i of the first robot which has n_1 joints and B_i is the inverse of the transformation between coordinate frames $i-1$ and i of the second robot which has n_2 joints. During the calculation of (1), the transformation between the ith joint coordinate frame and the coordinate frame of the second robot's end effector, denoted by U_i, can be obtained from

$$\begin{aligned} U_{n_1+n_2} &= B_{n_2} \\ \vdots &= \vdots \\ U_{n_1+1} &= B_{n_2} \ldots B_1 \\ U_1 &= B_{n_2} \ldots B_0 A_0 \\ U_2 &= B_{n_2} \ldots B_0 A_0 A_1 \\ \vdots &= \vdots \\ U_{n_1} &= B_{n_2} \ldots B_0 A_0 A_1 \ldots A_{n_1-1} \end{aligned} \tag{2}$$

CH2872-0/90/0000-0300 © $1.00 1990 IEEE

The columns of the relative Jacobian, which are composed of relative linear and angular velocity vectors due to the ith joint are closely related to the third and fourth columns of the above U matrices. In particular, it is easy to show that the relative linear velocity, v_i, is given by the cross product of the third and fourth column of U_i and that the relative angular velocity, ω_i, is given by the third column of U_i (see Figure 1). The relative Jacobian, denoted J_R, is therefore given by

$$J_R = R_{h_1 \leftarrow h_2} \begin{bmatrix} v_1 & \cdots & v_{n_1+n_2} \\ \omega_1 & \cdots & \omega_{n_1+n_2} \end{bmatrix} \qquad (3)$$

where the upper 3×3 rotation $R_{h_1 \leftarrow h_2}$ from (1), is required in order to express the relative velocity with respect to the first hand's coordinate frame.

The composite Jacobian, denoted J_C, represents the set of secondary constraints which consist of obstacle avoidance, joint limit avoidance, and absolute position and orientation requirements. The obstacle avoidance criteria is represented by an obstacle avoidance Jacobian which relates the joint rates to the absolute linear velocity of those links that contain proximity sensors. It is obvious that the absolute Cartesian motion of a link possessing a sensor can only be affected by joints which are closer to the base. Thus the obstacle Jacobian, denoted J_O, for a sensor which is mounted on the sth link of the second robot is given by the $3 \times (n_1 + n_2)$ matrix

$$J_O = \big[\underbrace{0|0|\cdots|0|}_{robot1} \,|\, \underbrace{v_1|v_2|\cdots|v_s|0|0|\cdots|0}_{robot2} \big]. \qquad (4)$$

The Jacobian used to specify the mechanical joint limit constraints is trivial because there is no need to transform from joint space to Cartesian space, i.e. the only joint which affects the motion of the ith joint is the ith joint. Thus the general form for the $1 \times (n_1 + n_2)$ joint limit Jacobian, denoted J_L, is

$$J_L = \big[0|0|\cdots|0|1|0|\cdots|0\big]. \qquad (5)$$

The third type of Jacobian considered is an absolute constraint Jacobian which 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 the absolute Jacobian are zero. The remaining columns of an absolute Jacobian are obtained by inserting the full $6 \times n$ standard Jacobian. For the absolute constraints on the second robot the $6 \times (n_1 + n_2)$ absolute Jacobian denoted $J2_A$ is

$$J2_A = \Big[\underbrace{[0]}_{robot1} \,|\, \underbrace{[J2]}_{robot2} \Big]. \qquad (6)$$

The composite Jacobian is formed by combining the three types of Jacobians described in (4), (5), and (6) to obtain

$$J_C = \begin{bmatrix} [J_{O_1}] \\ \vdots \\ [J_{O_{n_s}}] \\ [J_{L_1}] \\ \vdots \\ [J_{L_{n_l}}] \\ [J1_A] \\ [J2_A] \end{bmatrix} \qquad (7)$$

where each subscript indicates the type of Jacobian and n_s and n_l are the number of sensors and joints with limits respectively. The dimension of J_C is $(3n_s + n_l + 12) \times (n_1 + n_2)$.

III. Specifying Secondary Constraints

The secondary constraints of obstacle avoidance, joint limit avoidance, and absolute end effector locations are described by the equation

$$J_C \dot{\theta} = \dot{z} \qquad (8)$$

where $\dot{z}$ is of the form

$$\dot{z}^T = \big[\, \dot{z}_{O_1} \quad \cdots \quad \dot{z}_{O_{n_s}} \quad \dot{z}_{L_1} \quad \cdots \quad \dot{z}_{L_{n_l}} \quad \dot{z}1_A \quad \dot{z}2_A \big] \qquad (9)$$

and $\dot{z}_O$ is the desired velocity of a link possessing a sensor, $\dot{z}_L$ is the desired velocity of a joint approaching a limit, and $\dot{z}_A$ is the desired absolute velocity of the end effector. The velocity $\dot{z}_O$ is a 3 element vector given by

$$\dot{z}_O = \alpha(d_O)\hat{n} \qquad (10)$$

where $\hat{n}$ is a vector pointing away from the obstacles, d_O is the distance to the obstacles, and α is a scalar function that is inversely proportional to d_O. The velocity $\dot{z}_L$ for the ith joint is given by

$$\dot{z}_L = \frac{\mathrm{sgn}(\theta_{mid_i} - \theta_i)}{d_L} \qquad (11)$$

where θ_{mid_i} is the joint value midway between the two joint limits and d_L is given by

$$d_L = \min(\theta_{max_i} - \theta_i, \theta_i - \theta_{min_i}). \qquad (12)$$

where θ_{max_i} and θ_{min_i} are the maximum and minimum joint limits for the ith joint. The velocity $\dot{z}_A$ is composed of both linear and rotational velocities so that in general

$$\dot{z}_A = \begin{bmatrix} v \\ \omega \end{bmatrix}. \qquad (13)$$

It is assumed that the absolute constraints on the end effectors are specified in world coordinates so that the 3 element linear velocity is given by

$$v = p_c - p_d \qquad (14)$$

where p_d is the desired position for the end effector and p_c is the current position of the end effector. The 3 element rotational velocity is obtained in an analogous manner based on the difference between the desired and current orientations.

IV. Calculating Solutions

If the desired relative velocity required to complete an assembly task is denoted by $\dot{x}_R$ then the solution for the joint velocity that achieves this velocity and simultaneously comes closest to satisfying the secondary constraints specified by (8) is given by [5,6]

$$\dot{\theta} = J_R^+ \dot{x}_R + [J_C(I - J_R^+ J_R)]^+ (\dot{z} - J_C J_R^+ \dot{x}_R). \qquad (15)$$

where $^+$ denotes the pseudoinverse. The first term $J_R^+ \dot{x}_R$ represents the minimum joint velocity required to achieve the primary task. The term $[J_C(I - J_R^+ J_R]^+$ is the projection of the composite Jacobian onto the null space of the relative Jacobian which is required to keep the secondary constraints from affecting the relative motion specified as the primary constraint. The vector $J_C J_R^+ \dot{x}_R$ must be subtracted from the desired secondary constraint velocity $\dot{z}$ in order to account for the velocities induced by the pseudoinverse solution of the primary constraint.

The solution given by (15) gives equal weight to all of the secondary criteria, i.e. it minimizes the quantity $\|\dot{z} - J_C\dot{\theta}\|^2$. In practice a weighted least squares solution is desirable in order to assign a dynamic priority to the secondary constraints. This is performed by specifying a diagonal positive semi-definite matrix Q and then minimizing the quantity $\|Q(\dot{z} - J_C\dot{\theta})\|^2$. In addition to specifying relative priorities, Q provides a convenient method of completely removing any of the secondary constraints by setting the corresponding element to zero.

V. Results

The trajectory generation program was tested for a workcell consisting of two seven link robots, the Cybotech and CESARM manipulators. The robots are specified to start in an awkward initial configuration with one arm over the other (see Figure 2-a) and are then commanded to follow a relative end effector trajectory which ends with the two end effectors facing each other. The joint trajectories to achieve the desired relative trajectory are first calculated using only the minimum norm solution. Next, the secondary criteria of obstacle and joint limit avoidance are added and the resulting motions are compared to the minimum norm case. A quantitative comparison of the three cases is presented in Figures 3 and 4 which graphically present the minimum distance to obstacles and joint limits throughout the trajectory.

The application of only the minimum norm solution rapidly results in a collision between the two arms (see Figure 2-b). When the obstacle avoidance constraint is added to the trajectory generation, however, the manipulators reorient themselves so that there is no collision at the point where the minimum norm solution collides (see Figure 2-c) or throughout the remainder of the trajectory (see Figures 2-d and 3). Unfortunately, the reorientation used to avoid the obstacles results in a violation of the joint limit constraint on the sixth joint of the Cybotech manipulator (see Figure 4). This can be remedied by using the formulation which includes both the obstacle avoidance and the joint limit avoidance constraints. A comparison of Figures 3 and 4 illustrates how the proposed algorithm uses the weighting parameters to allocate the available redundancy to the more critical of the secondary constraints. As a result, the manipulators are able to complete the assigned task without violating any of the constraints.

VI. Summary

The algorithm presented generates joint trajectories for two robots cooperating to perform an assembly task specified by a trajectory expressed in the coordinates of one of the parts. The robots are considered as a single redundant system with secondary criteria such as obstacle and joint limit avoidance, as well as absolute constraints being satisfied while performing the assembly task. The Jacobian relating the joint rates of the entire system to the relative motion of one of the hands with respect to the other is used to generate the relative motion satisfying the cooperative task. Motion satisfying the secondary criteria is generated by a composite Jacobian equation whose solution is constrained to be in the null space of the relative Jacobian.

References

[1] A. Hemami, "Kinematics of two-arm robots," *IEEE J. Robotics Auto.*, vol. RA–2, no. 4, pp. 225–228, 1986.

[2] K. Laroussi, H. Hemami, and R. E. Goddard, "Coordination of two planar robots in lifting," *IEEE J. Robotics Auto.*, vol. RA–4, no. 1, pp. 77–85, 1988.

[3] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *IEEE Trans. Robotics Auto.*, vol. RA–5, no. 1, pp. 78–97, 1989.

[4] J. Y. S. Luh and Y. F. Zheng, "Constrained relations between two coordinated industrial robots for motion control," *Int. J. Robotics Res.*, vol. 6, no. 3, pp. 60–70,1987.

[5] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robotics Res.*, vol. 4, no. 3, pp. 109–117, 1985.

[6] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robotics Res.*, vol. 6, no. 2, pp. 3–15, Summer 1987.

[7] H. Seraji, "Adaptive force and position control of manipulators," *J. Robotic Systems*, vol. 4, no. 4, pp. 52–71, 1987.

[8] H. Suh and K. G. Shin, "Coordination of dual robot arms using kinematic redundancy," *Proc. 1988 Int. Conf. Robotics Auto.*, pp. 504–509, 1988.

[9] R. Zapata, A. Fournier, and P. Dauchez, "True cooperation of robots in multi-arms tasks," *Proc. 1987 Int. Conf. Robotics Auto.*, pp. 1255–1260, 1987.

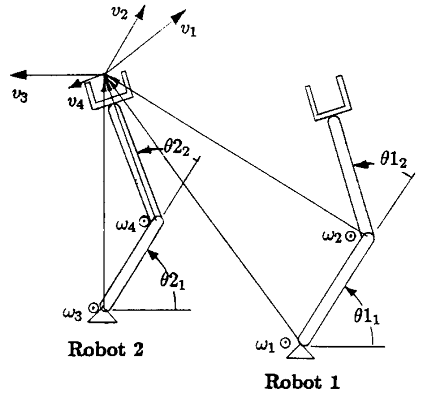

Figure 1: Geometric interpretation of the vectors defining the relative Jacobian

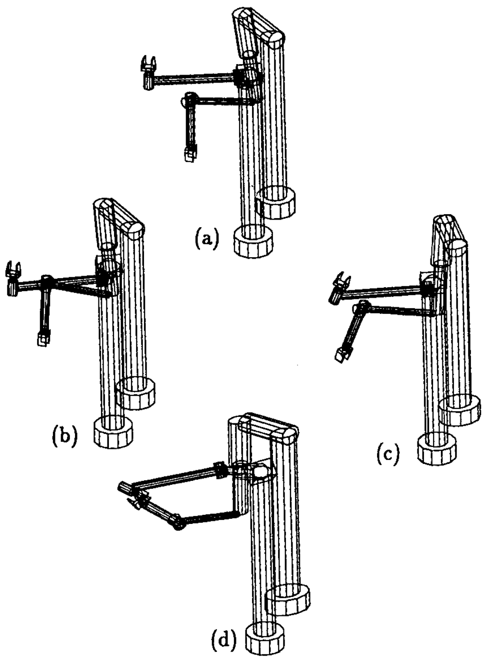

Figure 2: Comparison of the minimum norm path to the obstacle avoidance path: (a) - initial configuration, (b) - minimum norm collision point, (c) - obstacle avoidance at the same time as (b), (d) - final configuration for obstacle avoidance trajectory.

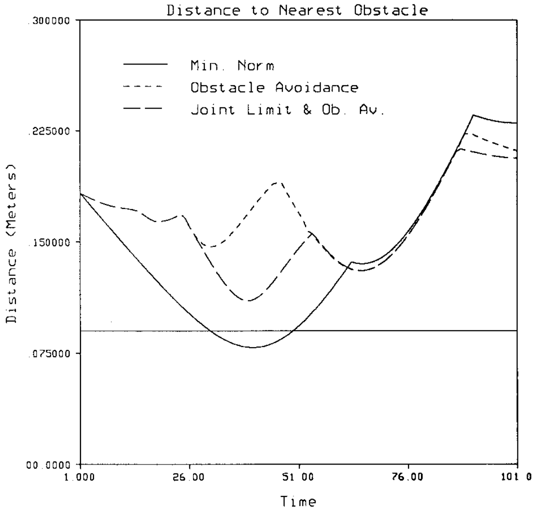

Figure 3: Comparison of the minimum distance to the obstacles for solutions with and without the secondary constraints.

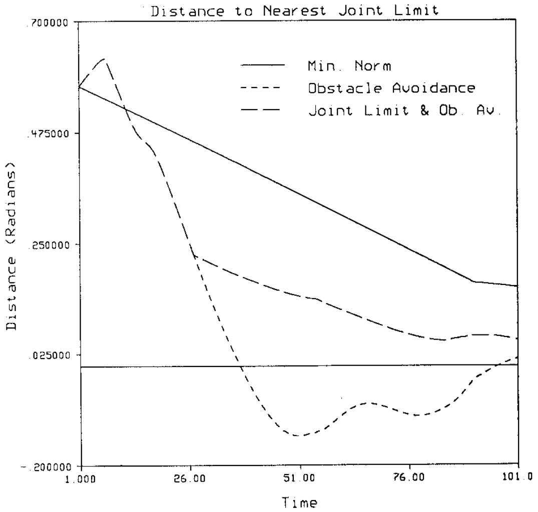

Figure 4: Comparison of the minimum distance to the joint limits for solutions with and without the secondary constraints.