

Task-Oriented Kinematic Control of Two Cooperative 6-DOF Manipulators

Pasquale Chiacchio Stefano Chiaverini Bruno Siciliano

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125 Napoli, Italy

Abstract—*The kinematic control problem for a system of two cooperative 6-dof manipulators is addressed in this paper. First, a kinematic model of the system is obtained which allows a clear description of the task in terms of absolute and relative variables. Then, a suitable inverse kinematics scheme is designed to compute the joint variable trajectories corresponding to a given trajectory for the above cooperative task variables. The effectiveness of the technique is demonstrated by two case studies for two cooperative PUMA 560 robot manipulators; one regards a typical coordinated motion for the system, the other is aimed at illustrating the redundancy resolution features of the scheme.*

1. Introduction

An effective control strategy for a system of two robots holding a common object must properly take into account the mechanical constraints imposed by the contact between each arm and the object. It can be recognized that both the absolute motion of the object and the internal forces acting between the two end-effectors should be controlled. Also, the presence of kinematic redundancy into the system should be properly handled.

Several works addressed the problem of coordinated control of a two-arm robot system, e.g. [1], which designed control systems that improve on the performance of previous schemes of master/slave fashion [2], or else of leader-follower type [3]. Further schemes were developed which regarded the cooperative system from a global point of view, i.e. operating with object space coordinates, and allowed control of absolute object motion and of internal forces [4-6].

A drawback of the above schemes is that the interpretation of cooperative task space variables is not always straightforward; from the user's point of view, it would be preferable to define variables that are directly related to the assigned cooperative task. In this context, a two-stage kinematic control strategy can be devised which presents the cascade of a kinematic inversion block for the task variables characterizing the cooperation and a conventional control in the joint space. One advantage of a two-stage control strategy is also the capability of solving kinematic redundancy. This approach was developed for two planar arms [7,8] and is here generalized to the case of two cooperative 6-dof manipulators.

The contribution of this work is to obtain a kinematic model of the cooperative system which allows a clear description of the task in terms of absolute and relative components. In particular, an effective definition of meaningful (absolute and relative) orientation

variables is given which overcomes the major difficulty in previous cooperative task descriptions.

Further, a closed-loop inverse kinematics scheme [9,10] is developed to compute the joint variable trajectories corresponding to a given trajectory for the above cooperative task variables. The scheme uses the Jacobians relative to the task variables which, by virtue of the above kinematic description, can be expressed in terms of the end-effector Jacobians for the single arms.

A system composed by two cooperative PUMA 560 robot manipulators is considered to work out two case studies. In the first one, a trajectory is assigned to the task variables realizing a coordinated motion for the system. In the second one, the system is made kinematically redundant by considering a reduced number of task variables, and the features of the scheme to perform redundancy resolution are demonstrated.

2. Task description

The choice of a functional task description for multiple arm systems manipulating a common object is crucial to the development of effective control systems that achieve true cooperation between the arms.

Consider a system of two spatial manipulators. The location of each end-effector can be described by six task space variables, three for position and three for a minimal representation of orientation. This indicates that, if the manipulators have to cooperate, a number of twelve task space variables should be defined to fully describe the kinematics of the system.

The trivial choice would be to consider the two end-effector locations. However, in this way the system is regarded as composed by two independent manipulators and management of coordination is left to the user.

Another possibility would be to adopt as task variables those derived from a kineto-static object level description of the system which is based on the so-called grasp matrix mapping end-effector forces into external object forces [4]. One drawback of this description regards the representation of orientation. In fact, the kinematic variables are originated from the composition of linear and angular velocities of the two end-effectors; a problem then arises in view of the non-integrability of angular velocities. Further, even if a minimal representation of orientation is found, this would not have a clear meaning to the user.

In the following an effective description is established which unambiguously characterizes the cooperative task as well as allows the user to give an imme-

diate specification of the task in terms of meaningful variables.

It can be recognized that the typical task of a cooperative two-arm system is to manipulate a common object. Therefore, it is natural to define not only a set of variables specifying the absolute motion of the object but also a set of variables describing the relative location between the end-effectors which in turn characterizes the object grasp. Obviously, both the absolute and relative variables include position and orientation.

Consider a system of two cooperative manipulators. For each manipulator ($i = 1, 2$), let p_i^b be the (3×1) vector denoting the end-effector position. Let also R_i^b be the (3×3) rotation matrix expressing the end-effector orientation. Both quantities are expressed in a common base frame (superscript "b").

The end-effector linear velocity is directly given as the time derivative of the position vector, that is $\dot{p}_i^b$. The end-effector angular velocity is given by the (3×1) vector ω_i^b , which is related to the time derivative of the rotation matrix R_i^b through the relationship

$$\dot{R}_i^b = S(\omega_i^b)R_i^b, \quad (1)$$

where $S(\cdot)$ is the (3×3) skew-symmetric operator performing the vector cross product.

The absolute position of the cooperative system can be defined as the origin of a suitable frame attached to the object (denoted by subscript "a"), which has to be expressed as a function of the positions of the two end-effectors. One simple choice is

$$p_a^b = \frac{1}{2}(p_1^b + p_2^b), \quad (2)$$

whose time derivative gives the absolute linear velocity

$$\dot{p}_a^b = \frac{1}{2}(\dot{p}_1^b + \dot{p}_2^b). \quad (3)$$

In order to define the absolute orientation of the system, consider the matrix operator $R_k(\vartheta)$ expressing the rotation by the angle ϑ about the axis aligned with the unit vector $k = (k_x \ k_y \ k_z)^T$. Note also that the following property holds:

$$\dot{R}_k(\lambda\vartheta) = \lambda \dot{R}_k(\vartheta), \quad (4)$$

where λ is a constant. The rotation matrix giving the absolute orientation is then defined as

$$R_a^b = R_1^b R_{k_{12}}^1(\vartheta_{12}/2), \quad (5)$$

where k_{12}^1 and ϑ_{12} are respectively the unit vector and the angle that realize the rotation described by R_2^1 . Therefore, the above choice corresponds to make a rotation about the axis k_{12}^1 by an angle which is half the angle needed to align R_2^b with R_1^b . By differentiating (5) and using (1,4), it can be shown that the absolute angular velocity is given by

$$\omega_a^b = \frac{1}{2}(\omega_1^b + \omega_2^b). \quad (6)$$

The relative position between the two end-effectors

can be defined as

$$p_r^b = p_2^b - p_1^b, \quad (7)$$

whose time derivative is

$$\dot{p}_r^b = \dot{p}_2^b - \dot{p}_1^b. \quad (8)$$

Nevertheless, notice that for the user it is more convenient to assign the relative position in the object frame, i.e. p_r^a . If this is the case, the relative position in the base frame can be computed as

$$p_r^b = R_a^b p_r^a \quad (9)$$

with R_a^b as in (5). Further, the relative linear velocity in the base frame is given by

$$\dot{p}_r^b = R_a^b \dot{p}_r^a + S(\omega_a^b) p_r^b \quad (10)$$

with ω_a^b as in (6).

The relative orientation between the two end-effectors can be defined in the end-effector frame of either manipulator—say the first one—according to the above choice for the relative position, i.e.

$$R_r^1 = R_2^1. \quad (11)$$

The resulting relative angular velocity is

$$\omega_r^1 = \omega_2^b - \omega_1^b \quad (12)$$

which is notefully related to quantities expressed in the base frame.

3. Closed-loop inverse kinematics scheme

The above task space description constitutes the basis for a kinematic control scheme. This requires the solution of an inverse kinematics problem, that is finding the joint variable trajectories corresponding to given trajectories for the absolute and relative task variables. These trajectories will then constitute the reference inputs to some joint space control scheme.

Any algorithmic solution to the inverse kinematics problem is based on the computation of the Jacobian relative to the task variables of interest. Since these variables have been expressed as a function of the position and orientation of the two end-effectors, the sought Jacobian can be related to the Jacobians of the single manipulators.

Without loss of generality, consider a system of two cooperative 6-dof manipulators. For each manipulator, let q_i indicate the (6×1) vectors of joint variables. The geometric Jacobian $J_i^b(q_i)$ is the (6×6) matrix relating the joint velocity vectors $\dot{q}_i$ to the linear and angular end-effector velocities in the base frame as

$$\begin{pmatrix} \dot{p}_i^b \\ \omega_i^b \end{pmatrix} = J_i^b(q_i) \dot{q}_i \quad i = 1, 2. \quad (13)$$

At this point, combining (3,6) and taking into account (13) yields

$$\begin{pmatrix} \dot{p}_a^b \\ \omega_a^b \end{pmatrix} = J_a^b(q_1, q_2) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}, \quad (14)$$

where the (6×12) absolute Jacobian matrix is defined as

$$J_a^b = \left(\frac{1}{2} J_1^b \quad \frac{1}{2} J_2^b \right). \quad (15)$$

Further, combining (8,12) and taking into account (13) yields

$$\begin{pmatrix} \dot{p}_r^b \\ \omega_r^1 \end{pmatrix} = J_r^b(q_1, q_2) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}, \quad (16)$$

where the (6×12) relative Jacobian matrix is defined as

$$J_r^b = (-J_1^b \ J_2^b). \quad (17)$$

The inverse kinematics algorithm used is the closed-loop scheme based on the computation of the inverse of the manipulator Jacobian [9]. In the general case, the joint velocity solution can be written in the form

$$\dot{q} = J^{-1}(v_d + Ke) \quad (18)$$

where q is the vector of joint variables, J is the Jacobian—assumed to be square and singularity-free—associated to the velocity mapping, v_d is the desired task velocity, K is a suitable diagonal positive gain matrix, and e is the error between the desired and actual task variables.

The closed-loop inverse kinematics scheme based on (18) avoids the typical numerical drift of open-loop resolved-rate schemes. The solution can be made robust with respect to singularities of J by resorting to a damped least-squares inverse of the matrix [11].

Further, if the system possesses redundant degrees of freedom—due either to the presence of additional joint variables or to relaxation of some task variables—the Jacobian becomes a low rectangular matrix. In this case, the solution is modified into [10]

$$\dot{q} = J^\dagger(v_d + Ke) + (I - J^\dagger J)\dot{q}_0 \quad (19)$$

where $J^\dagger$ denotes the pseudoinverse of J , and the operator $(I - J^\dagger J)$ projects the vector of arbitrary joint velocities $\dot{q}_0$ (aimed at exploiting the redundant degrees of freedom) into the null space of J so as not to interfere with the primary end-effector task.

The above algorithms can be keenly applied to solve the inverse kinematics for the cooperative system at issue. In detail, define

$$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}. \quad (20)$$

If the system is non-redundant, solution (18) is applied. The Jacobian is

$$J = \begin{pmatrix} J_a^b \\ J_r^b \end{pmatrix}, \quad (21)$$

where J_a^b , J_r^b are given as in (15,17). The error is

$$e = \begin{pmatrix} e_a \\ e_r \end{pmatrix}. \quad (22)$$

The absolute error has a translation and an orientation component and is given by

$$e_a = \begin{pmatrix} p_{ad}^b - p_a^b \\ \frac{1}{2}(S(n_a^b)n_{ad}^b + S(s_a^b)s_{ad}^b + S(a_a^b)a_{ad}^b) \end{pmatrix}, \quad (23)$$

where p_{ad}^b is the desired absolute position specified by the user in the base frame, p_a^b is the actual absolute position that can be computed as in (2), n_{ad}^b , s_{ad}^b , a_{ad}^b are the column vectors of the rotation matrix R_{ad}^b giving the desired absolute orientation specified by the user in the base frame, and n_a^b , s_a^b , a_a^b are the column vectors of the rotation matrix R_a^b in (5). The relative error is given by

$$e_r = \begin{pmatrix} R_a^b p_{rd}^a - p_r^b \\ \frac{1}{2}(S(n_r^1)n_{rd}^1 + S(s_r^1)s_{rd}^1 + S(a_r^1)a_{rd}^1) \end{pmatrix}. \quad (24)$$

The rotation R_a^b is aimed at expressing the desired relative position p_{rd}^a , assigned by the user in the object frame, in the base frame; in this way, if an error occurs on the object frame orientation this does not affect the specification of the desired relative position between the two end-effectors. Further in (24), p_r^b can be computed as in (7), n_{rd}^1 , s_{rd}^1 , a_{rd}^1 are the column vectors of the rotation matrix R_{rd}^1 giving the desired relative orientation specified by the user in the end-effector frame of the first manipulator, and n_r^1 , s_r^1 , a_r^1 are the column vectors of the rotation matrix R_r^1 in (11). Finally, the desired velocity is

$$v = \begin{pmatrix} v_a \\ v_r \end{pmatrix}. \quad (25)$$

The absolute velocity term is given by

$$v_a = \begin{pmatrix} \dot{p}_{ad}^b \\ \omega_{ad}^b \end{pmatrix}, \quad (26)$$

where $\dot{p}_{ad}^b$ and ω_{ad}^b are respectively the desired absolute linear and angular velocities specified by the user in the base frame. The relative velocity term is given by

$$v_r = \begin{pmatrix} R_a^b \dot{p}_{rd}^a + S(\omega_a^b) R_a^b p_{rd}^a \\ \omega_{rd}^1 \end{pmatrix}, \quad (27)$$

where $\dot{p}_{rd}^a$ is the desired relative linear velocity specified by the user in the object frame and ω_{rd}^1 is the desired relative angular velocity specified by the user in the end-effector frame of the first manipulator. Notice that the expression of the translational part of the relative velocity presents an additional term which is a consequence of having assigned the relative position in the object frame.

On the other hand, if the system is redundant, solution (19) is applied. In this case, the rows in the Jacobian matrix (21) corresponding to the relaxed task variables have to be canceled and J becomes a low-rectangular matrix. The analogous components in the error vector (22) and desired velocity (25) have to be canceled accordingly; the dimension of matrix K is reduced accordingly. Concerning the choice of $\dot{q}_0$ in (19), this is taken as

$$\dot{q}_0 = k_c \nabla_q c(q) \quad (28)$$

where c is a constraint function of the joint variables that is optimized locally and k_c is a signed constant.

4. Case studies

A system of two cooperative PUMA 560 manip-

ulators is considered to develop two case studies. Discrete-time simulations of the inverse kinematics algorithms (18) and (19) have been run in MATLAB at 1 [ms] sampling time. To gain numerical robustness to the occurrence of kinematic singularities, a damped least-squares is used in lieu of the pure (pseudo)inverse of the relevant Jacobian.

A non-redundant system is analyzed in the first case study, i.e. twelve cooperative task variables are considered, and a coordinated motion is specified for the system. The bases of the two manipulators are respectively located at $(0 \ -0.1501 \ 0)^T$ [m] and $(1.4331 \ 0.1501 \ 0)^T$ [m] with reference to a common base frame. The initial joint configurations are $q_1 = (0 \ -2\pi/5 \ 9\pi/10 \ 0 \ 0 \ 0)^T$ [rad] and $q_2 = (\pi \ -2\pi/5 \ 9\pi/10 \ 0 \ 0 \ 0)^T$ [rad]. The initial rotation matrices R_1^b, R_2^b are chosen as identity matrices so as to refer to suitable tool frames aligned with the base frame; of course, the orientation of the two end-effectors may differ but the algorithms will embed proper rotation matrices to pass from the end-effector frames to the user-defined tool frames. With these values, the initial absolute position is $p_a^b = (0.7166 \ 0 \ 0.4310)^T$ [m] and the initial relative position is $p_r^a = (0.1 \ 0 \ 0)^T$ [m]. Also, the initial absolute rotation matrix is $R_a^b = I$ and the initial relative rotation matrix is $R_r^b = I$; this is a direct consequence of the convenient choice made above for R_1^b, R_2^b . The final absolute position is displaced at $(0.05 \ 0 \ 0.05)^T$ [m] from the initial one while the final relative position is displaced at $(-0.02 \ 0 \ 0)^T$ [m] from the initial one. As regards the orientation, a rotation of $\pi/4$ [rad] along the y -axis of the base frame is assigned for the final absolute orientation while a rotation of 0.1 [rad] along the z -axis of the initial R_1^b is assigned for the final relative orientation. The trajectories for the desired variables are generated by using an interpolating polynomial of fifth order with null initial and final velocities and accelerations; the duration time is 1 [s]. The feedback gain matrix in (18) is $K = \text{block diag}\{500I_6, 1000I_6\}$ where I_6 means the (6×6) identity matrix.

The results are plotted in Figs. 1-4. The norm of position and orientation components of both absolute and relative errors (Figs. 1,2) demonstrate the tracking capabilities of the inverse kinematics scheme; the error peaks occurring in the starting phase are due to the effort paid to exit from the wrist singularities of both manipulators. The time history of the joint variables is shown in Figs. 3,4.

Reconfiguration of a redundant system is analyzed in the second case study. The cooperative system is made redundant by relaxing the absolute position, i.e. nine task variables are considered which have to remain constant. The initial configuration of the system is the same as in the previous case study. The constraint function in (28) is

$$c(q) = \frac{1}{4} u^T (J_{t1}^b (J_{t1}^b)^T + J_{t2}^b (J_{t2}^b)^T) u$$

where the subscript "t" refers to the translational part

of the single arm Jacobians. Minimizing this function is equivalent to maximizing the intersection of the global task space external force manipulability ellipsoid [16] along the task space direction specified by the unit vector u ; in this case, it is $u = (1 \ 0 \ 0)^T$ describing the x -axis of the base frame. The constant in (28) is $k_c = -10$.

The results are plotted in Figs. 5-8. The time history of the function in Fig. 5 demonstrates the effective exploitation of redundant degrees of freedom to minimize the given constraint. This is obtained by means of a reconfiguration of the system; the absolute position (Fig. 6) changes while the errors (Figs. 7,8) show that all the other task variables are practically unaffected.

ACKNOWLEDGMENTS

This work was supported by *Consiglio Nazionale delle Ricerche* under contracts no. 92.01064.PF67 and no. 92.01105.PF67.

REFERENCES

- [1] M. Uchiyama, N. Iwasawa, and K. Hakomori, "Hybrid position/force control for coordination of a two-arm robot," *Proc. 1987 IEEE International Conference on Robotics and Automation*, Raleigh, NC, pp. 1242-1247, 1987.
- [2] E. Nakano, S. Ozaki, T. Ishida, and I. Kato, "Cooperative control of the anthropomorphic manipulator 'MELARM'," *Proc. 4th International Symposium on Industrial Robots*, Tokyo, J, pp. 251-260, 1974.
- [3] 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, n. 3, pp. 60-70, 1987.
- [4] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," *Proc. 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, pp. 350-356, 1988.
- [5] Z. Li, P. Hsu, and S. Sastry, "Dynamic coordination of a multiple robotic system with point contact," *Proc. 1988 American Control Conference*, Atlanta, GA, pp. 505-510, 1988.
- [6] J.T. Wen and K. Kreutz, "Motion and force control for multiple cooperative manipulators," *Proc. 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, pp. 1246-1251, 1989.
- [7] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Dexterous reconfiguration of a two-arm robot system," *Proc. IEE International Conference on Control '91*, Edinburgh, GB, pp. 347-351, March 1991.
- [8] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Cooperative control schemes for multiple robot manipulator systems," *Proc. 1992 IEEE International Conference on Robotics and Automation*, Nice, F, pp. 2218-2223, 1992.
- [9] B. Siciliano, "A closed-loop inverse kinematic scheme for on-line joint-based robot control," *Robotica*, vol. 8, pp. 231-243, 1990.
- [10] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *International Journal of Robotics Research*, vol. 10, no. 4, pp. 410-425, 1991.
- [11] Y. Nakamura and H. Hanafusa, "Inverse kinematic solution with singularity robustness for robot manipulator control," *ASME J. of Dynamic Systems, Measurements, and Control*, vol. 108, pp. 163-171, 1986.
- [12] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Global task space manipulability ellipsoids for multiple arm systems," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 5, pp. 678-685, 1991.

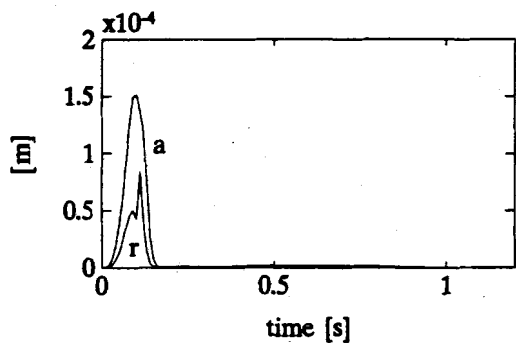


Fig. 1 — Norm of position errors; (a) absolute, (r) relative.

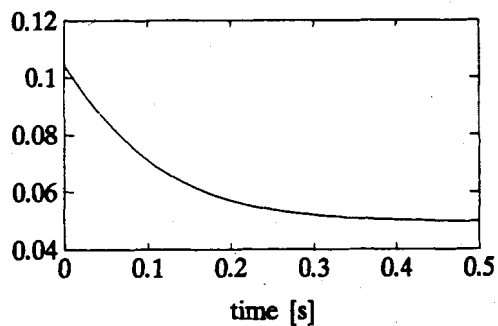


Fig. 5 — Constraint variable.

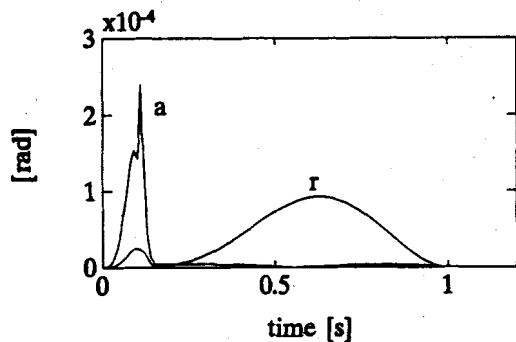


Fig. 2 — Norm of orientation errors; (a) absolute, (r) relative.

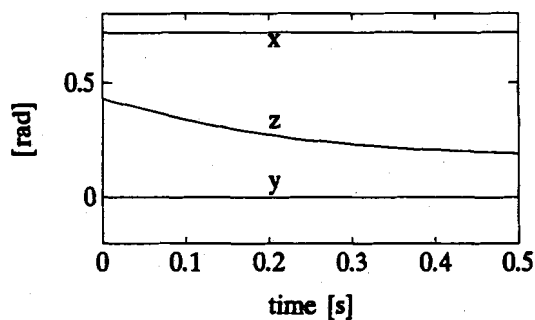


Fig. 6 — Absolute position coordinates.

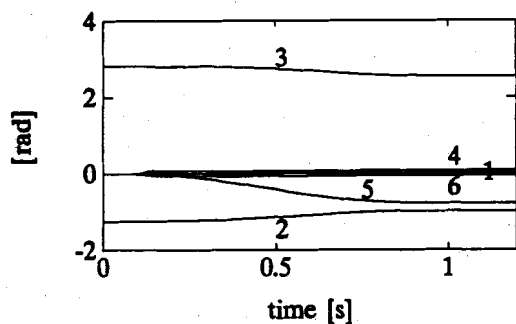


Fig. 3 — Joint variables of manipulator # 1.

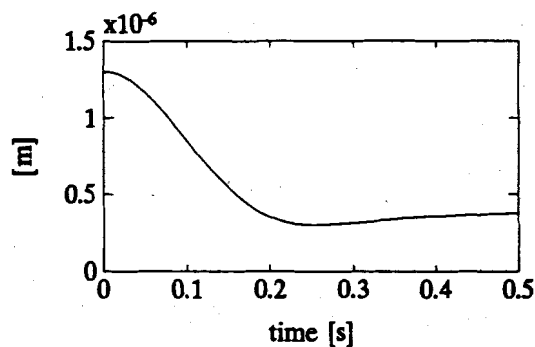


Fig. 7 — Norm of relative position error.

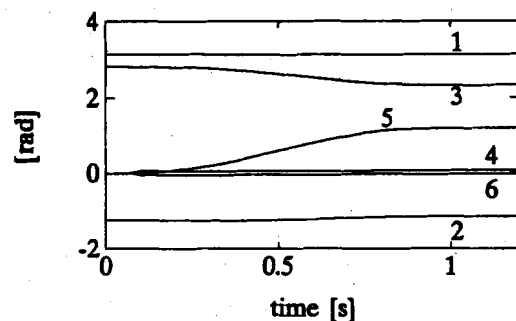


Fig. 4 — Joint variables of manipulator # 2.

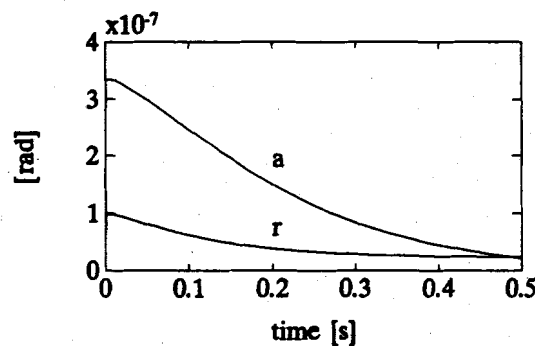


Fig. 8 — Norm of orientation errors; (a) absolute, (r) relative.