

REDUNDANCY RESOLUTION FOR TWO COOPERATIVE SPATIAL MANIPULATORS WITH A SLIDING CONTACT

P. Chiacchio, S. Chiaverini and B. Siciliano
Federico II University of Naples, Naples, Italy

INTRODUCTION

Cooperative robot manipulator systems are receiving an increasing interest in the research community in view of their potential over ordinary single robot manipulators. The goal is to achieve coordinated motion of the two robots so that a commonly held object can be effectively manipulated.

The task formulation plays an important role in the kinematics description of this kind of systems. Previous cooperative task descriptions [1,2] have the inconvenience that the interpretation of task variables is not always straightforward, especially for what concerns orientation in spatial manipulators.

Following our earlier results for planar manipulators [3], a cooperative task description has recently been proposed for spatial manipulators [4] that allows the user to specify the motion in terms of meaningful absolute and relative variables in a clear fashion.

In this work a cooperative system of two spatial manipulators is considered where one of them is allowed a sliding contact with the object. Parallel research efforts for the case of rolling contacts are reported in [5,6].

Motion coordination is achieved by adopting a closed-loop inverse kinematics scheme [7] which allows computation of joint trajectories corresponding to given task trajectories. Redundancy resolution is performed in view of the extra degrees of freedom available from the sliding contact.

A system of two PUMA 560 robot manipulators is taken to develop numerical case studies aimed at showing the effectiveness of the approach.

REVIEW OF COOPERATIVE TASK FORMULATION

The typical task of a cooperative two-arm system is to manipulate a common object. This demands for a task description in terms of absolute variables describing the motion of the object and relative variables describing the mutual location between the end effectors which in turn characterize the object grasp. Both absolute and relative variables include position and orientation.

In [4] an effective user-oriented task formulation was established which unambiguously defines the cooperative task for spatial manipulators as well as allows the user to give a direct specification of the task in terms of meaningful variables.

The absolute position and orientation are defined by attaching a suitable frame to the object and relating its origin position vector p_a and rotation matrix R_a to the position and orientation of the two manipulators' end effectors, i.e.

$$p_a = \frac{1}{2}(p_1 + p_2) \quad R_a = R_1 R_{k_{12}}^1(\vartheta_{12}/2) \quad (1)$$

where p_1, p_2 are the end-effectors' position vectors, R_1, R_2 are the end-effectors' rotation matrices; these can be conveniently expressed as a function of the joint variable vectors q_1 and q_2 . Also in (1) k_{12}^1 and ϑ_{12} are respectively the unit vector and the angle realizing the rotation described by R_2^1 ; then the actual rotation is by half the angle needed to align R_2 with R_1 . The relative position and orientation are defined as

$$p_r = p_2 - p_1 \quad R_r^1 = R_2^1. \quad (2)$$

All the above quantities are referred to a common base frame.

The differential kinematics corresponding to (1),(2) are found to be

$$\begin{aligned} \dot{p}_a &= \frac{1}{2}(\dot{p}_1 + \dot{p}_2) & \dot{p}_r &= \dot{p}_2 - \dot{p}_1 \\ \omega_a &= \frac{1}{2}(\omega_1 + \omega_2) & \omega_r^1 &= \omega_2 - \omega_1 \end{aligned} \quad (3)$$

which simply express the linear and angular velocities in the cooperative task space as a function of the differential kinematics of the single manipulators.

SLIDING CONTACT

The foregoing task formulation can be directly applied to specify tasks for a tightly grasped object by assigning suitable values to the absolute and relative variables.

To embed the possibility of handling a *sliding contact* in either of the two manipulators, a *virtual end effector* can be introduced by adding an adequate number of fictitious joints at the actual end effector. The sliding contact is realized if the orientation of the actual and virtual end effectors coincide while their positions differ according to the geometry of the sliding surface.

It can be recognized that a sliding contact requires at most *two degrees of freedom*. In the case of a planar surface, *two virtual prismatic joints* are added at the end effector with their axes realizing two degrees of freedom along the surface; of course, the two axes must not be aligned. The result is an augmented kinematics expressing the location of the virtual end effector.

In the case of a curved surface, sliding contact still requires at most two degrees of freedom. However, three virtual prismatic joints have to be introduced at the end effector with a geometric constraint so as to realize two independent degrees of freedom along the surface.

Consider a system of two cooperative 6-degree-of-freedom manipulators. For each manipulator, let q_i indicate the (6×1) vectors of joint variables. The geometric Jacobian $J_i(q_i)$ 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

$$\begin{bmatrix} \dot{p}_i \\ \omega_i \end{bmatrix} = J_i(q_i)\dot{q}_i \quad i = 1, 2. \quad (4)$$

Let q_S denote the (2×1) vector of the additional joint variables. Without loss of generality, manipulator 2 is assumed to make the sliding contact; this implies that p_2 becomes a function of both q_2 and q_S . Hence the virtual end effector still makes a tight grasp with the object and the foregoing task formulation can be retained.

The task space differential kinematics associated to (3) become:

$$\begin{bmatrix} \dot{p}_a \\ \omega_a \end{bmatrix} = J_a \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_S \end{bmatrix} \quad (5)$$

for the *absolute* part where

$$J_a(q_1, q_2, q_S) = \left[\frac{1}{2} J_1(q_1) \quad \frac{1}{2} J_2(q_2, q_S) \right], \quad (6)$$

and

$$\begin{bmatrix} \dot{p}_r \\ \omega_r \end{bmatrix} = J_r \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_S \end{bmatrix} \quad (7)$$

for the *relative* part where

$$J_r(q_1, q_2, q_S) = \left[-J_1(q_1) \quad J_2(q_2, q_S) \right]. \quad (8)$$

INVERSE KINEMATICS WITH REDUNDANCY RESOLUTION

The task formulation of the previous section constitutes the basis for a kinematic control problem, that is finding the joint variable trajectories corresponding to given trajectories for the absolute and relative task variables; these trajectories will be the reference inputs to some joint space control scheme.

Define $q = [q_1^T \quad q_2^T \quad q_S^T]^T$ and $J = [J_a^T \quad J_r^T]^T$. The following closed-loop *inverse kinematics* scheme with *redundancy resolution*, originally proposed for single arms [7], can be adopted:

$$\dot{q} = J^\dagger(q)(v_d + Ke) + (I - J^\dagger J)k_c \left(\frac{\partial c(q)}{\partial q} \right)^T \quad (9)$$

where K is a positive definite diagonal matrix, $e = [e_a^T \quad e_r^T]^T$ is the task space error, $v_d = [v_{ad}^T \quad v_{rd}^T]^T$ is the desired feedforward velocity, c is a constraint function of the joint variables that is optimized locally in the null space of J and k_c is a signed constant.

In detail, the absolute error is

$$e_a = \begin{bmatrix} p_{ad} - p_a \\ \frac{1}{2}(\mathbf{n}_a \times \mathbf{n}_{ad} + \mathbf{s}_a \times \mathbf{s}_{ad} + \mathbf{a}_a \times \mathbf{a}_{ad}) \end{bmatrix} \quad (10)$$

where p_{ad} is the desired absolute position specified by the user in the base frame, p_a is the actual absolute position that can be computed as in (1), \mathbf{n}_{ad} , \mathbf{s}_{ad} , \mathbf{a}_{ad} are the column vectors of the rotation matrix R_{ad} giving the desired absolute orientation specified by

the user in the base frame, and $\mathbf{n}_a, \mathbf{s}_a, \mathbf{a}_a$ are the column vectors of the rotation matrix \mathbf{R}_a in (1). The relative error is given by

$$\mathbf{e}_r = \left[\begin{array}{c} \mathbf{R}_a \mathbf{p}_{rd}^a - \mathbf{p}_r \\ \frac{1}{2}(\mathbf{n}_r^1 \times \mathbf{n}_{rd}^1 + \mathbf{s}_r^1 \times \mathbf{s}_{rd}^1 + \mathbf{a}_r^1 \times \mathbf{a}_{rd}^1) \end{array} \right] \quad (11)$$

The rotation \mathbf{R}_a is aimed at expressing the desired relative position \mathbf{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 (11), \mathbf{p}_r can be computed as in (2), $\mathbf{n}_{rd}^1, \mathbf{s}_{rd}^1, \mathbf{a}_{rd}^1$ are the column vectors of the rotation matrix \mathbf{R}_{rd}^1 giving the desired relative orientation specified by the user in the end-effector frame of the first manipulator, and $\mathbf{n}_r^1, \mathbf{s}_r^1, \mathbf{a}_r^1$ are the column vectors of the rotation matrix \mathbf{R}_r^1 in (2). The absolute velocity term is given by

$$\mathbf{v}_{ad} = [\dot{\mathbf{p}}_{ad}^T \quad \boldsymbol{\omega}_{ad}^T]^T \quad (12)$$

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

$$\mathbf{v}_{rd} = \left[\begin{array}{c} \mathbf{R}_a \dot{\mathbf{p}}_{rd}^a + \boldsymbol{\omega}_a \times \mathbf{R}_a \mathbf{p}_{rd}^a \\ \boldsymbol{\omega}_{rd}^1 \end{array} \right] \quad (13)$$

where $\dot{\mathbf{p}}_{rd}^a$ is the desired relative linear velocity specified by the user in the object frame and $\boldsymbol{\omega}_{rd}^1$ 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.

NUMERICAL CASE STUDIES

A system of two cooperative PUMA 560 robot manipulators has been considered to work out numerical case studies. The base of manipulator 1 is located at $(0, -0.1501, 0)$ and the initial configuration places the end effector at $\mathbf{p}_1 = [0.4 \ 0 \ 0.5]^T$; a constant rotation matrix has been used so that $\mathbf{R}_1 = \mathbf{I}$. The base of manipulator 2 is located at $(1, 0.1501, 0)$ and the initial configuration places the end effector at $\mathbf{p}_2 = [0.6 \ 0 \ 0.5]^T$ with $\mathbf{R}_2 = \mathbf{I}$. Manipulator 2 is allowed to make a sliding contact with the object.

Using (1),(2), the initial values for the task variables are computed:

$$\mathbf{p}_a = [0.5 \ 0 \ 0.5]^T \quad \mathbf{R}_a = \mathbf{I} \quad \mathbf{p}_r = [0.2 \ 0 \ 0]^T \quad \mathbf{R}_r^1 = \mathbf{I}.$$

A sketch of the initial configuration is depicted in Fig. 1. For clarity of illustration, the dimensions of the object have been enlarged.

The task is to move the object to the absolute location

$$\mathbf{p}_a = [0.5 \ 0 \ 0.7]^T \quad \mathbf{R}_a = \begin{bmatrix} \cos \pi/4 & -\sin \pi/4 & 0 \\ \sin \pi/4 & \cos \pi/4 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A rectilinear path is assigned from the initial to the final position, whereas an angular motion about a fixed axis in space is assigned to pass from the initial to the final orientation. Smooth trajectories are imposed using 5th-order interpolating polynomials with null initial and final velocities and accelerations, and a time duration of 1 s.

As for the relative variables, it is opportune to define the task with reference to the absolute frame. The task is to keep \mathbf{p}_r^a and \mathbf{R}_r^1 constant; note the simplicity of task specification when it is referred to the absolute frame.

The closed-loop inverse kinematics scheme based on (9), at first without the null space term, has been implemented in MATLAB at 1 ms sampling time; the gain matrix has been chosen as $\mathbf{K} = \text{block diag}\{500\mathbf{I}_6, 1000\mathbf{I}_6\}$. The resulting final system configuration is also shown in Fig. 1. As can be recognized, the object is taken to the desired final position whereas the final orientation of end effector 2 accounts for the sliding contact. The time history of the norm of position and orientation components of both absolute and relative errors (Fig. 2) confirm the good tracking capabilities of the inverse kinematics scheme.

Another numerical case study has been worked out where the kinematic redundancy introduced by the sliding contact is exploited to minimize the constraint $c = 0.5(q_{2,1}(t) - q_{2,1}(0))^2$, i.e. to keep the base revolute joint of manipulator 2 constant. The initial configuration is the same as before (Fig. 1) and the same task has been assigned for the absolute and relative variables. Further, in the discrete-time implementation of (9) it has been chosen $k_C = 3000$. The final configuration in Fig. 3 shows the correct execution of the task while the base revolute joint of manipulator 2 is remarkably kept close to the initial value as shown by the time history of the constraint value.

CONCLUSIONS

An inverse kinematics scheme for a system of two spatial manipulators holding a common object with a sliding contact has been presented. The scheme is based on a cooperative task formulation that allows the user to give a straightforward description of coordinated motions in terms of absolute and relative variables. Two numerical case studies have demonstrated the good performance of the scheme also when the redundant degrees of freedom introduced by the sliding contact are effectively exploited.

ACKNOWLEDGEMENTS

The research reported in this work was supported by *Consiglio Nazionale delle Ricerche* under contract n. 93.00946.PF67.

REFERENCES

- [1] J.Y.S. Luh and Y.F. Zheng, "Constrained relations between two coordinated industrial robots for motion control," *Int. J. of Robotics Research*, vol. 6, n. 3, pp. 60-70, 1987.
- [2] M. Uchiyama, and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots". *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, pp. 350-356, 1988.
- [3] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Kineto-static analysis of cooperative robot manipulators achieving dexterous configurations," *9th CISM-IFTOMM Symp. on Theory and Practice of Robots and Manipulators*, Udine, I, Sep. 1992, in *RoManSy 9*, Lecture Notes in Control and

Information Sciences 187, A. Morecki, G. Bianchi, K. Jaworek (Eds.), Springer-Verlag, Berlin, D, pp. 93-100, 1993.

- [4] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Task-oriented kinematic control of two cooperative 6-dof manipulators," *Proc. 1993 American Control Conf.*, San Francisco, CA, pp. 336-340, 1993.
- [5] P. Chiacchio, S. Chiaverini, and B. Siciliano, "User-oriented task description for cooperative spatial manipulators: One-degree-of-freedom rolling grasp," *Proc. 32nd IEEE Conf. on Decision and Control*, San Antonio, TX, pp. 1126-1127, 1993.
- [6] P. Chiacchio and S. Chiaverini, "User-oriented task description for cooperative spatial manipulators: Rolling grasp," *5th Int. Symp. on Robotics and Manufacturing*, Maui, HI, Aug. 1994.
- [7] P. Chiacchio, S. Chiaverini, L. Sciavico, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task-space augmentation and task-priority strategy", *Int. J. of Robotics Research*, vol. 10, pp. 410-425, 1991.

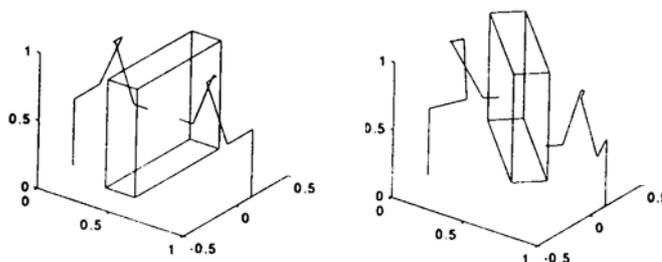


Fig. 1 — Initial (left) and final (right) configurations.

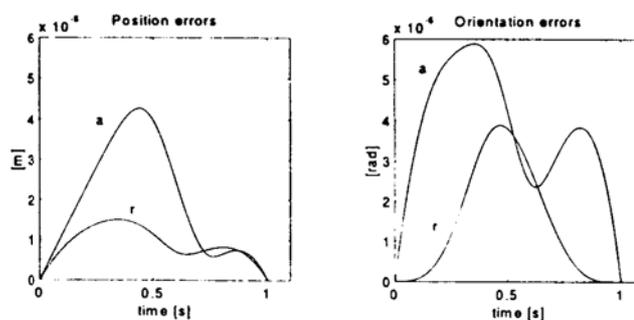


Fig. 2 Time history of norm of position and orientation errors: a—absolute; r—relative.

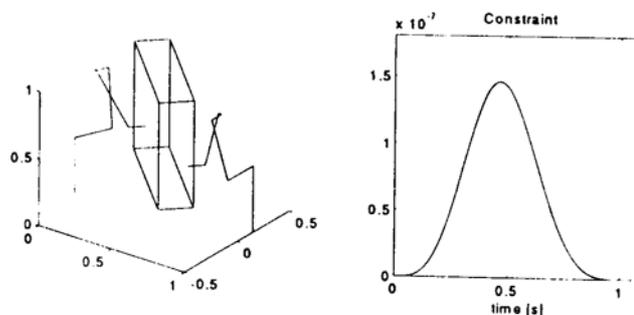


Fig. 3 Final configuration and time history of constraint value in case of redundancy resolution.