

Cooperative Control Schemes for Multiple Robot Manipulator Systems

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

Three schemes are developed which are aimed at achieving cooperative control of multiple arm systems manipulating a common object. The first scheme operates wholly on the object task space variables. The second scheme operates on the joint space variables that can be derived via a kinematic inversion from the cooperative task space variables. The third scheme combines the features of the above two by solving the cooperation at the inverse kinematic level and acting the control at the object level. Simulation results are provided for a two-arm planar system to investigate the behavior of the controlled system in the case of inaccurate object modeling.

1. Introduction

Typical requirements of a control algorithm for multiple robot manipulator systems are recognized to be:

- control of the absolute motion of the carried object;
- control of the internal forces acting in the system, e.g. object stretching, shearing, bending.

These goals can be met only if an effective coordination of the arms is accomplished, which in turn demands for a truly cooperative control system to be designed.

One of the most promising approach to task modeling of multiple arm systems seems to be the symmetric formulation proposed in [1] for the case of two arms, and later generalized also to the case of multiple arms [2]. This allows a natural definition of both external and internal task (force and velocity) vectors that can be handled for control purposes. Experimental verification of hybrid position/force control schemes that make use of the above formulation can be found in [3,4].

The problem of choosing the proper variables to be controlled is critically examined. A first scheme is discussed which requires both specification and control of task variables defined in the object space [5]. An alternative two-stage control scheme is proposed which presents the cascade of a kinematic inversion block for

the task variables that characterize the cooperation [6] and a pure joint space controller. Finally, a new scheme is introduced according to which the cooperative task variables can be specified independently of the object space where the control operates. This scheme will be shown to have some advantages over the previous two.

An impedance behavior is adopted for the controlled variables [7], resulting in a computationally cheaper solution that also avoids the use of force sensors [5], as in most previous schemes instead [3,4,8].

The performance of the three schemes is extensively tested out in a number of simulated case studies for a two-arm planar system when inaccurate object modeling occurs.

2. Task Modeling

The symmetric formulation proposed in [1] provides a natural framework for modeling a cooperative task in terms of a suitable set of external and internal variables, forces and velocities respectively.

Without loss of generality, consider a system of two planar manipulators holding a common object. Let assume that the grasp is tight, so that each arm can exert both a force and a moment on the object; then \mathbf{h}_i , $i = 1, 2$ denotes the (3×1) vector of contact forces —two components of forces and one component of moment— exerted by the end-effector at the contact point. Correspondingly, the position variables are identified as \mathbf{x}_i , $i = 1, 2$ —two components of position and the orientation angle— expressing the position of the end-effector at the contact point. All these quantities are intended to be expressed in a common base frame.

Choose a coordinate frame located on the object and consider the virtual sticks pointing from the two contact points to the origin of the object frame. Each stick is assumed to be rigidly attached to the end-effector of the arm. Let then $\mathbf{x}_{i,s}$ denote the location of the stick tip expressed in the base frame. Accordingly, the force at the stick tip $\mathbf{h}_{i,s}$ has to be considered as the contact force.

The use of virtual sticks allows a simple description of force and moment composition. It is supposed that the deformation due to object elasticity is small, so that the locations of the two stick tips can be taken as both coincident with the origin of the object frame. If \mathbf{h}_a denotes the (3×1) vector of absolute (external) forces acting on the object, the relationship between contact and object forces is given by

$$\mathbf{h}_a = \mathbf{W} \begin{pmatrix} \mathbf{h}_{1s} \\ \mathbf{h}_{2s} \end{pmatrix} = (\mathbf{I} \quad \mathbf{I}) \begin{pmatrix} \mathbf{h}_{1s} \\ \mathbf{h}_{2s} \end{pmatrix}, \quad (1)$$

where \mathbf{W} is the (3×6) grasp matrix, and \mathbf{I} denotes the (3×3) identity matrix. As anticipated above, the expression of the grasp matrix is particularly simple, thanks to the use of virtual sticks.

For given object forces, Eq. (1) can be inverted by using the pseudoinverse of \mathbf{W} , that is simple to compute as well. A solution to (1) is given by

$$\begin{pmatrix} \mathbf{h}_{1s} \\ \mathbf{h}_{2s} \end{pmatrix} = \mathbf{W}^\dagger \mathbf{h}_a + \mathbf{V} \mathbf{h}_r + \begin{pmatrix} \frac{1}{2} \mathbf{I} \\ \frac{1}{2} \mathbf{I} \end{pmatrix} \mathbf{h}_a + \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \mathbf{h}_r, \quad (2)$$

where \mathbf{V} is a (6×3) matrix spanning the null space of \mathbf{W} , and \mathbf{h}_r is a (3×1) vector of relative (internal) forces. Eq. (2) can be compacted into

$$\mathbf{h}_s = \begin{pmatrix} \mathbf{h}_{1s} \\ \mathbf{h}_{2s} \end{pmatrix} = \mathbf{U} \begin{pmatrix} \mathbf{h}_a \\ \mathbf{h}_r \end{pmatrix} = \mathbf{U} \mathbf{h}_o, \quad (3)$$

with $\mathbf{U} = (\mathbf{W}^\dagger \quad \mathbf{V})$, which gives the relationship between object forces \mathbf{h}_o and stick forces \mathbf{h}_s .

Direct application of the principle of virtual work yields the relationship between the (6×1) vector of object velocities \mathbf{v}_o and the (6×1) vector of stick velocities \mathbf{v}_s in the form

$$\mathbf{v}_o = \begin{pmatrix} \mathbf{v}_a \\ \mathbf{v}_r \end{pmatrix} = \mathbf{U}^T \begin{pmatrix} \mathbf{v}_{1s} \\ \mathbf{v}_{2s} \end{pmatrix} = \mathbf{U}^T \mathbf{v}_s, \quad (4)$$

with obvious meaning of the components of \mathbf{v}_o and \mathbf{v}_s . Further, assume that the two arms hold a purely elastic object. Then, the interaction between the arms and the object can be described by the simple linear model

$$\mathbf{h}_1 = -\mathbf{h}_2 = \mathbf{K}_o (\tilde{\mathbf{x}} - (\mathbf{x}_2(q_2) - \mathbf{x}_1(q_1))), \quad (5)$$

where \mathbf{K}_o is the (3×3) diagonal matrix of object constant spring coefficients, and $\tilde{\mathbf{x}}$ is a (3×1) vector characterizing the object at rest; thus, contact forces arise only from object deformation.

3. Cooperative Control Schemes

Below are illustrated three different control schemes that can be devised according to which space the variables are specified and which space they are controlled in. More specifically, let denote by *task space* the space where the variables that describe the cooperative task are specified; this space does not necessarily coincide with the *object space* that indicates the space where the absolute and relative variables of the above formulation are defined. Finally, the *joint space* is the space where control actions are actuated in terms of joint driving torques. This distinction will allow to reveal the distinctive features of each of the schemes that follow.

It should be remarked that all the schemes can adopt any kind of controller for the relevant variables; in other words, what matters here is the space in which control actions are designed rather than the particular control algorithm used.

In the present work, an impedance behavior is adopted for the controlled variables, following the guidelines of [5]. This choice is motivated by the desire of avoiding measurements of contact forces; force sensors must be frequently retuned, suffer from low signal-to-noise ratios, require significant preprocessing of the raw output data, and then their use in industrial environments may be not desirable. As a further advantage, a computationally cheap control law is obtained, as opposed to the decoupling schemes that make use of full nonlinear compensation, e.g. [8]. Nevertheless, it is convenient to compensate for gravitational forces since this yields a significant improvement on the steady-state performance of the system at low computational expense.

3.1. Control in object space

In this case, the task space is chosen as coincident with the object space. This implies that the variables to be specified are directly those defined in the object space [5].

Let then \mathbf{x}_a denote the location of the origin of the object frame; the kinematic constraints imposed by the closed-chain of the arms and object lead to computing the absolute object location as

$$\mathbf{x}_a = \frac{1}{2}(\mathbf{x}_{1s} + \mathbf{x}_{2s}), \quad (6)$$

while the relative location between the tips of the two sticks is given by

$$\mathbf{x}_r = \mathbf{x}_{2s} - \mathbf{x}_{1s}. \quad (7)$$

By combining \mathbf{x}_a and \mathbf{x}_r into $\mathbf{x}_o = (\mathbf{x}_a^T \ \mathbf{x}_r^T)^T$, the following PD control laws can be designed:

$$\mathbf{h}_o = \mathbf{K}_P(\bar{\mathbf{x}}_o - \mathbf{x}_o) + \mathbf{K}_D(\dot{\bar{\mathbf{x}}}_o - \dot{\mathbf{x}}_o), \quad (8)$$

where $\bar{\mathbf{x}}_o$ contains reference values for \mathbf{x}_o .

Once \mathbf{h}_o has been synthesized, the stick Jacobians \mathbf{J}_i , have to be computed by expressing \mathbf{x}_i , as a kinematic function of the joint variables \mathbf{q}_i for each arm. Then, the design is completed by computing the joint control torques $\boldsymbol{\tau} = (\boldsymbol{\tau}_1^T \ \boldsymbol{\tau}_2^T)^T$ of the impedance type [7] as

$$\boldsymbol{\tau} = \mathbf{J}_s^T(\mathbf{q})\mathbf{h}_s + \mathbf{g}(\mathbf{q}), \quad (9)$$

where \mathbf{h}_s is given in (3), and a compact notation has been used for $\mathbf{J}_s = \text{diag}(\mathbf{J}_{1s}, \mathbf{J}_{2s})$, $\mathbf{q} = (\mathbf{q}_1^T \ \mathbf{q}_2^T)^T$; further, $\mathbf{g} = (\mathbf{g}_1^T \ \mathbf{g}_2^T)^T$, with \mathbf{g}_i being the vector of gravitational forces for each arm. The resulting block diagram scheme based on Eqs. (8,9) is shown in Fig. 1, where \mathbf{K}_{PD} is the short-hand notation for the PD control laws (8), \mathbf{k}_o denotes the direct kinematic functions that are needed to compute \mathbf{x}_o , and the gravity compensation has not been evidenced.

It can be argued that specification of these variables might be problematic. Regarding \mathbf{x}_a , while the absolute object position does allow a natural task variable specification, less meaningful is the specification of the absolute object orientation when this is computed via (6). The situation is more dramatic for the relative stick location, for which it might be inconvenient to adopt two position variables and one orientation variables.

Another pitfall of this control scheme is that it does not offer the possibility of exploiting eventual kinematic redundancies available in the system; these occur when the total number of joints is greater than the number of task (object) space variables of interest, six in this case.

3.2. Control in joint space

In order to overcome the drawbacks of the above scheme, the radically opposite scheme that can be devised is one that operates completely in the joint space. In other words, a set of meaningful task space variables are specified that describe the cooperative task. Next, the corresponding joint reference variables are computed via a kinematic inversion procedure. Finally, the control is designed completely in the joint space.

For the two-arm system at issue, an effective choice of absolute and relative task variables can be obtained as in [6]. It is anticipated that such variables allow an effective description of the cooperation between the

multiple arms and the object, although they are not directly related to the variables of the above object level formulation. Notice, however, that any set of variables can be selected as long as they lead to a complete description of the cooperative task; the particular choice presented below is related explicitly to the task to execute, but others are feasible for different tasks.

The absolute position of the object is computed as

$$\mathbf{p}_a = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2), \quad (10)$$

where \mathbf{p}_1 and \mathbf{p}_2 are the position vectors of the end-effectors of the two arms expressed in the common base frame. Let consider a reference frame fixed to the object with origin in \mathbf{p}_a and x -axis aligned with the vector $\mathbf{w} = \mathbf{p}_2 - \mathbf{p}_1$. Then, the absolute orientation of the object can be described as the angle formed by the x -axis of the object frame with the x -axis of the base frame, i.e.

$$\phi_a = \text{atan2}(w_x, w_y). \quad (11)$$

Next, the relative variables must be specified. First, the relative position of the two end-effectors can be described by the vector \mathbf{w} . However, if \mathbf{w} is conveniently expressed in the object frame, only the x_o component of \mathbf{w} is significant; this leads to considering one scalar variable (the signed distance between the two end-effectors) to describe this portion of the task, thus overcoming the drawback of the previous strategy. Then it is

$$p_r = w_x \cos \phi_a + w_y \sin \phi_a. \quad (12)$$

As for the remaining variables, the grasp angle of each arm relative to the x -axis of the object frame can be selected, i.e.

$$\phi_{r1} = \phi_a - \phi_1 \quad (13)$$

$$\phi_{r2} = \pi + \phi_a - \phi_2, \quad (14)$$

where ϕ_1 and ϕ_2 represent the orientation angles of the two end-effectors with respect to the base frame; by the way, each angle is given by the sum of the respective joint coordinates.

Having specified a certain task in terms of the above variables, say a (6×1) vector $\bar{\mathbf{x}}$, Eqs. (10-14) can be used to find the corresponding joint variables $\bar{\mathbf{q}}$; this stage can be performed even off-line. Those variables constitute the references for a simple PD + gravity compensation control of the kind

$$\boldsymbol{\tau} = \mathbf{K}_P(\bar{\mathbf{q}} - \mathbf{q}) + \mathbf{K}_D(\dot{\bar{\mathbf{q}}} - \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}). \quad (15)$$

The resulting control scheme is illustrated in Fig. 2, where this time K_{PD} is the short-hand notation for the PD control laws (15), and the block k^{-1} denotes the inverse kinematic function which is required to transform the chosen task space variables into the corresponding joint space variables.

Compared to the scheme of Fig. 1, there now exists the possibility of exploiting kinematic redundancies at the kinematic inversion level; to this purpose, the relevant Jacobians for absolute and relative variables have to be computed if a Jacobian based algorithm is going to be used to solve for redundancy, e.g. [9]. The weakness of this solution, however, resides in the high sensitivity to imperfect modeling of the cooperation; the feedback control loop, in fact, does not operate at the object level and then cannot effectively contrast any disturbance and/or model inaccuracy occurring between the arms and the held object.

3.3. Control in object space with specification in task space

In order to combine the advantages of the above two schemes, a third new scheme is proposed in the following. The design key is to allow specification of the variables in the task space, like for the second scheme, but to let the control act in the object space, like for the first scheme. This objective is met by retaining the two-stage structure of the scheme in Fig. 2 in principle, i.e. kinematic inversion followed by control, and modifying the control stage in such a way that the object variables of the scheme in Fig. 1 are involved.

Specifically, if $(\bar{q} - q)$ and $(\dot{\bar{q}} - \dot{q})$ indicate joint position and velocity errors, they can be regarded —at first approximation— as generating elastic and damping feedback terms at the stick level of the type $J_s(\bar{q} - q)$ and $J_s(\dot{\bar{q}} - \dot{q})$. These terms can then be transformed into object (absolute and relative) velocities through the relation (4) and then the rest of the scheme follows from that in Fig. 1. In detail, the vector of object forces is selected as

$$h_o = K_P U^T J_s(q)(\bar{q} - q) + K_D U^T J_s(q)(\dot{\bar{q}} - \dot{q}) \quad (16)$$

and the design is completed by computing h_s as in (3) and τ as in (9).

Notice that the PD control actions now operate in the object space and allow different weighting of the force and moment components. The resulting block diagram scheme is illustrated in Fig. 3, where the same short-hand notation as for the other schemes has been used for the block K_{PD} .

In sum, the nice feature of the proposed solution is that the kinematic inversion is executed on task space

variables which can be different from the ones which are controlled. In this fashion, the user is offered the following advantages:

- specify the task in terms of a set of convenient, physically understandable variables;
- exploit eventual kinematic redundancies, for instance by reconfiguring the system in a more dexterous configuration for the execution of the given task [6], e.g. on the basis of task space manipulability ellipsoids [10,11];
- rely on the robustness of a control algorithm that operates on the object space variables and then is capable of rejecting disturbance effects occurring between the contact points and the point of interest on the object.

4. Case Studies

The three cooperative control schemes presented in the preceding section have been tested on a system of two equal three-degree-of-freedom planar arms holding a common disk-shaped object. Link parameters in SI units are reported in the table below where ℓ_k is the link length, ℓ_{ck} is the distance of the link center of mass from the joint axis, m_k is the link mass, I_k is the link inertia, and d_k is the joint viscous friction coefficient.

Link	ℓ_k	ℓ_{ck}	m_k	I_k	d_k
1	0.4	0.2	15	0.228	1.5
2	0.3	0.15	10	0.080	1.5
3	0.2	0.1	5	0.019	1.5

The base frame is located at the base of the first arm and the object frame with its x -axis oriented along the line connecting the two contact points. The base of the second arm is located at a distance of 0.6 from the reference frame along the x -axis.

The object spring matrix is $K_o = 10^5 I$ and the vector characterizing the object at rest is $(0.2 \ 0 \ \pi)^T$ expressed in the object frame; then, a proper coordinate transformation is accomplished to obtain \bar{x} as used in (5).

The reference task requires the object absolute position to follow a rectilinear path from $\bar{p}_a(0) = (0.3 \ 0.5)^T$ to $\bar{p}_a(t_f) = (0.2 \ 0.4)^T$ in a time $t_f = 2$ sec. The initial end-effector position vectors are $\bar{p}_1(0) = (0.2 \ 0.5)^T$ and $\bar{p}_2(0) = (0.4 \ 0.5)^T$, respectively. The assigned grasp is such that the last link of each arm is aligned with the x -axis of the object frame, and this must be kept constant along the path. A sketch of the required motion is given in Fig. 4. The trajectories for the reference variables are generated by using an

interpolating polynomial of fifth order with null initial and final velocities and accelerations.

In order to demonstrate the necessity of a control that operates in object space variables, the actual object location is $(0.201 \ 0 \ \pi)^T$ but the control assumes the object to be located at $(0.200 \ 0 \ \pi)^T$. It will be seen that even a 1 mm offset will cause appreciable differences in the performance of the three control schemes. The feedback gains of the PD control actions have been properly tuned so as to get approximately the same kind of response for the three schemes. The resulting trajectories for object absolute location are not reported since all the schemes show satisfactory tracking behavior.

The control scheme in object space (Fig. 1) is tested first. The location of the two stick tips have to be computed. It is easily obtained $\mathbf{x}_{1s} = (0.3 \ 0.5 \ 0)^T$ and $\mathbf{x}_{2s} = (0.3 \ 0.5 \ \pi)^T$. The absolute location of the object (6) moves from $\bar{\mathbf{x}}_a(0) = (0.3 \ 0.5 \ \pi/2)^T$ to $\bar{\mathbf{x}}_a(t_f) = (0.2 \ 0.4 \ \pi/2)^T$, while the relative location between the stick tips (7) is $\bar{\mathbf{x}}_r = (0 \ 0 \ \pi)^T$ to be kept along the entire path. Incidentally, it can be pointed out that a geometrical interpretation of the value of the third component of $\bar{\mathbf{x}}_a$ is not straightforward. The gains in (8) are chosen as $\mathbf{K}_P = \text{diag}(20000, 20000, 2000, 0, 0, 0)$, $\mathbf{K}_D = \text{diag}(15000, 15000, 1500, 5000, 5000, 1250)$. Notice that the choice of zero proportional gains for the relative variables implies that it is desired to hold the object without deforming it in regard to its rest state. A good behavior is observed for the forces and moment (Fig. 5) in spite of inaccurate object modeling.

Next, the control scheme in joint space (Fig. 2) is tested. The reference values for the absolute object position in (10) have already been specified above; the absolute object orientation must be kept constant at $\bar{\phi}_a = 0$. As far as the relative variables, these are immediately computed from (12), i.e. $\bar{p}_r = 0.2$, and from (13,14), i.e. $\bar{\phi}_{r1} = \bar{\phi}_{r2} = 0$ to be constant as well. Notice that, with respect to the previous scheme, the specification of task space variables is more direct and easy to understand from the geometry of the system. Then, from the above references the corresponding joint space trajectories are derived via a simple kinematic inversion. The gains in (15) are chosen as $\mathbf{K}_P = \text{diag}(25000, 25000, 25000, 25000, 25000, 25000)$, $\mathbf{K}_D = \text{diag}(500, 500, 500, 500, 500, 500)$. This time, the feedback gains are chosen all equal; in fact, the controller operates in the joint space and it is cumbersome to associate the motion of a particular joint to the motion of the corresponding task space component. The internal forces reported in Fig. 6 reveal that

a bias effect occurs at steady-state due to inaccurate object modeling.

Finally, the control scheme in object space (Fig. 3) with specification in task space is tested. The reference values for the absolute object location are obviously chosen as in the second scheme, and the kinematic inversion takes place yielding the same joint reference variables as for the second scheme. The gains in (16) are chosen as $\mathbf{K}_P = \text{diag}(360000, 360000, 36000, 0, 0, 0)$, $\mathbf{K}_D = \text{diag}(5000, 5000, 500, 2500, 2500, 625)$. As above, no object deformation is desired at rest. The resulting trajectories for internal forces reported in Fig. 7 demonstrate that the effects of inaccurate object modeling are soon recovered.

5. Conclusions

The problem of cooperative control of multiple robot manipulators holding a common elastic object has been addressed in this work. The focus has been pointed to the issue of effective task variable specification and control. This has led to deriving three different schemes which achieve, namely; control in object space, control in joint space, and control in object space with specification in task space. This last scheme seems the most attractive one since it attempts to combine the advantages of the first two schemes. A number of case studies have been developed to illustrate the performance of the three schemes. In particular, it has been shown how the occurrence of imperfect modeling is successfully tackled only by the first and the third schemes. The particular two-arm system analyzed, however, was quite a simple one and the difficulties in programming the reference values for the task space variables in the first scheme could not fully been appreciated in order to 'break the tie' in favor of the third scheme. Future work will be dedicated to exploit eventual kinematic redundancy in the system.

Acknowledgements

This work was supported partly by *Consiglio Nazionale delle Ricerche* under contract n. 91.01921.PF67, and partly by *Commission of the European Communities* under an ES-PRIT Research Grant assigned to the first author.

References

- [1] 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.
- [2] I.D. Walker, R.A. Freeman, and S.I. Marcus, "Multiple cooperating manipulators: The case of kinematically redundant arms," *Proc. NASA Conf. on Space Telerobotics*, Pasadena, CA, pp. 15-24, 1989.

- [3] M. Uchiyama and Y. Nakamura, "Symmetric hybrid position/force control of two cooperating robot manipulators," *Proc. 1988 IEEE Int. Workshop on Intelligent Robots and Systems*, Tokyo, J, pp. 515-520, 1988.
- [4] P. Dauchez, *Descriptions de Tâches en Vue de la Commande Hybride Symétrique d'un Robot Manipulateur a Deux Bras*, Thèse d'Etat, Université des Sciences et Techniques du Languedoc, Montpellier, F, 1990.
- [5] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Development of a control strategy for cooperative manipulators using a modular simulation package," *Proc. 1991 American Control Conference*, Boston, MA, pp. 2515-2520, 1991.
- [6] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Dexterous reconfiguration of a two-arm robot system," *Proc. IEE Int. Conf. on Control '91*, Edinburgh, GB, pp. 347-351, 1991.
- [7] N. Hogan, "Impedance control: An approach to manipulation (part I-III)," *Trans. ASME J. of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1-24, 1985.
- [8] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Dynamic force/motion control of cooperative robot systems," *Proc. ASME Winter Annual Meeting*, Dallas, TX, vol. DSC-26, pp. 121-126, 1990.
- [9] 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," *Int. J. of Robotics Research*, vol. 10, pp. 410-425, 1991.
- [10] 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, pp. 15-24, 1991.
- [11] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Task space dynamic analysis of multiarm system configurations," *Int. J. of Robotics Research*, vol. 10, pp. 708-715, 1991.

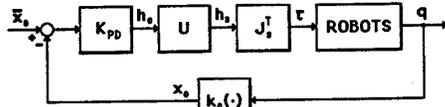


Fig. 1 Block scheme of control in object space.

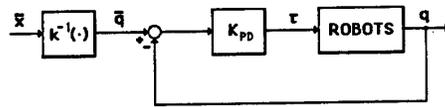


Fig. 2 Block scheme of control in joint space.

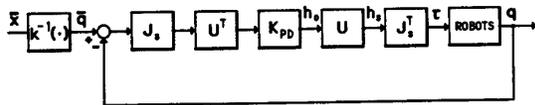


Fig. 3 Block scheme of control in object space with specification in task space.

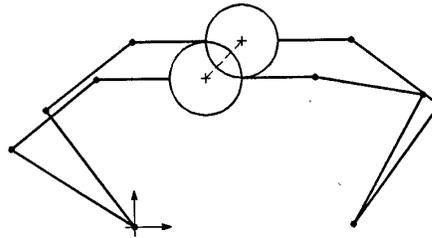


Fig. 4 Required motion for the two-arm system.

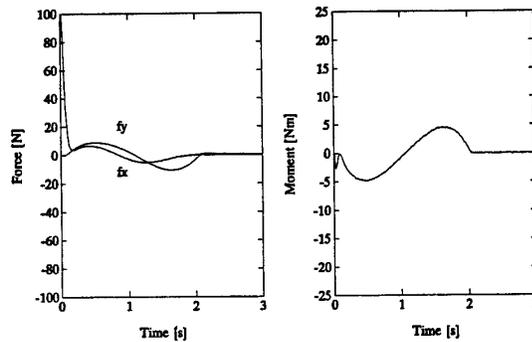


Fig. 5 Internal forces with the scheme of Fig. 1.

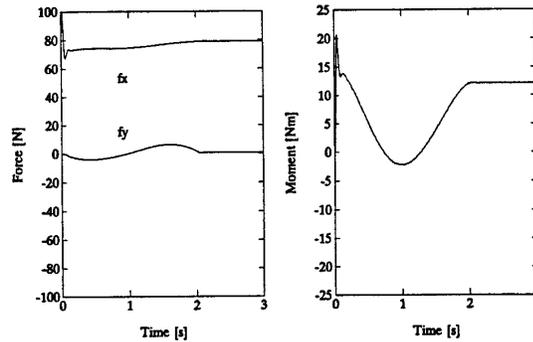


Fig. 6 Internal forces with the scheme of Fig. 2.

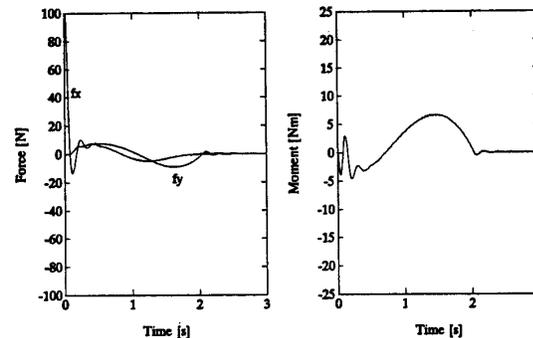


Fig. 7 Internal forces with the scheme of Fig. 3.