

Development of a Control Strategy for Cooperative Manipulators Using a Modular Simulation Package

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 aim of this paper is to investigate the performance of a control strategy for cooperative manipulators holding an object. The cooperation problem is solved by adopting a symmetric object task space description. An impedance behavior is assigned to controlled variables, resulting in a computationally simple control that does not require force sensor measurements. Simulation of the whole controlled system is performed using the LEGO modular simulation package which permits modeling the system as an interconnection of subsystems described by simple equations. Results of two typical case studies are presented.

Introduction

Cooperative manipulators have gained increasing popularity in the robotics research community, due to the potential they offer over conventional single arms. Manipulation tasks that require the use of multiple arms range from handling heavy or non-rigid objects to mating mechanical parts or operating in unstructured environments. The problem of controlling such systems is challenging because of internal forces arising in the closed-kinematic chain created by the arms and the manipulated object. To tackle this problem, control schemes have been proposed that attempt to control the position of the object as well as the internal forces acting on it; typical examples are the master/slave strategy [1,2] and controllers based on an object task space description of the grasp [3,4]. All these methods require the use of force sensors located on the arms to be embedded in the control scheme, but this could constitute a drawback in industrial environments.

In this paper a control scheme that does not require information from force sensors is presented. Based on a symmetric description of the task that decouples the absolute coordinates describing the object position from the relative coordinates characterizing the interaction between the arms [5], an impedance-like control law is derived [6]. In particular, high stiffness is assigned to the positioning part of the task, whilst

compliant behavior is prescribed for holding the object.

In order to simulate a controlled multi-arm system, two alternatives are possible: In one approach, the explicit dynamic equations describing the system are to be solved all together, but they are usually difficult to manage since multiple mechanical parts interact in a complex way. Another approach is based on defining modules describing simple dynamic systems and then giving connection rules between them. To achieve this goal, the simulation tool must be able to solve algebraic loops occurring when a variable is exchanged between modules with no dynamics.

We have followed the latter approach to test the proposed control scheme. The LEGO simulation package developed by the National Electric Power Company (ENEL) was used [7]; among its features is the capability of handling stiff dynamic equations and automatically solving algebraic loops. The results obtained for two planar arms manipulating an object show the effectiveness of the chosen control strategy.

Modeling and Control of a Cooperative Manipulator System

In order to model a system composed of multiple arms manipulating a common object, it is necessary to have a model of each element of the system and a description of the interaction between the arms and the object. Then, there exist several strategies to effectively control the cooperative system according to the chosen task description.

For the purpose of the present work, we focus our attention to the case of a two-arm planar system, although the results can be extended to more complex systems. With reference to Fig. 1, let define the following quantities for each arm ($j = 1, 2$):

- $\mathbf{q}_j$ ($n_j \times 1$) vector of joint variables,
- $\mathbf{B}_j$ ($n_j \times n_j$) inertia matrix,
- $\mathbf{n}_j$ ($n_j \times 1$) vector of Coriolis and centrifugal forces,
- $\mathbf{D}_j$ ($n_j \times n_j$) diagonal matrix of viscous friction constant coefficients,

g_j ($n_j \times 1$) vector of gravitational forces,
 τ_j ($n_j \times 1$) vector of joint driving torques,
 J_j ($3 \times n_j$) Jacobian matrix,
 h_j (3×1) vector of contact forces exerted by the end-effector at the contact point,
 x_j (3×1) vector of end-effector location.

Notice that in the planar case $h_j = (f_j \ m_j)^T$, where f_j is the (2×1) force vector and m_j is the scalar moment. Analogously, $x_j = (p_j \ \varphi_j)^T$, where p_j is the (2×1) position vector and φ_j is the scalar orientation. The task space quantities are intended to be expressed in a common reference frame.

Following the above notation, the dynamic model for each arm can be written in the well-known form

$$B_j(q_j)\ddot{q}_j + n_j(q_j, \dot{q}_j) + D_j\dot{q}_j + g_j(q_j) = \tau_j - J_j^T(q_j)h_j. \quad (1)$$

We assume that the two arms hold a purely elastic object. The interaction between the arms and the object can be described by the simple linear model

$$h_1 = -h_2 = K_o(\bar{x} - (x_2(q_2) - x_1(q_1))) \quad (2)$$

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

The above equations (1-2) provide a functional model of the physical system considered. In order to devise a control strategy for the two-arm system, a description of the cooperative task is needed. Among the different techniques proposed in the literature, we adopt the symmetric description in [5].

Let us 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 (Fig. 1). We suppose that the grasp is tight, and each stick is assumed to be rigidly attached to the end-effector of the arm. Let then x_{j_s} denote the location of the stick tip expressed in the base frame; this can be computed as a function of the joint variables and the given grasp geometry (stick length and angle formed by the stick with the last link). As a consequence, a modified stick Jacobian J_{j_s} is to be derived. Accordingly, we have to consider the force at the stick tip h_{j_s} 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 h_o denotes the (3×1) vector of equivalent forces

acting on the object, the relationship between contact and object forces is given by

$$h_o = W \begin{pmatrix} h_{1s} \\ h_{2s} \end{pmatrix} = (I \ I) \begin{pmatrix} h_{1s} \\ h_{2s} \end{pmatrix} \quad (3)$$

where W is the (3×6) grasp matrix, and I denotes the (3×3) identity matrix.

For given object forces, equation (3) can be inverted by using a (weighted) pseudo-inverse of W . Further, observing that matrix W possesses a three-dimensional null space, a solution to (3) is given by—the minimum norm pseudoinverse is adopted for W —

$$\begin{pmatrix} h_{1s} \\ h_{2s} \end{pmatrix} = W^\dagger h_o + Vh_i = \begin{pmatrix} \frac{1}{2}I \\ \frac{1}{2}I \end{pmatrix} h_o + \begin{pmatrix} I \\ -I \end{pmatrix} h_i \quad (4)$$

where V is a (6×3) matrix spanning the null space of W , and h_i is a (3×1) vector of internal forces. Equation (4) can be compressed into

$$\begin{pmatrix} h_{1s} \\ h_{2s} \end{pmatrix} = U \begin{pmatrix} h_o \\ h_i \end{pmatrix} \quad (5)$$

with $U = (W^\dagger \ V)$.

On the basis of the above formulation, we can develop a scheme to control both object and internal forces. Our goal is to design a controller that allows the execution of a task specified in terms of absolute object location and relative location of the sticks. The first design requirement is to avoid 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. The second design requirement is to obtain a computationally cheap control law; this rules out the use of full nonlinear compensation and dynamic decoupling schemes, e.g. [4,8,9]. Nevertheless, we do compensate for gravitational forces since this yields a significant improvement on the steady-state performance of the system at low computational expense [10]. The first requirement is met by resorting to the impedance control concept [6]. Specifically, let x_o denote the location of the origin of the object frame; the kinematic constraint imposed by the closed-chain of the arms and object gives $x_{1s} = x_{2s}$, and then x_o can be computed either as x_{1s} or as x_{2s} . Further, let x_i denote the relative location between the tips of the two sticks, i.e. $x_i = x_{2s} - x_{1s}$. The following PD control laws are designed:

$$h_o = K_{Po}(\bar{x}_o - x_o) + K_{Do}(\dot{\bar{x}}_o - \dot{x}_o) \quad (6)$$

$$h_i = K_{Pi}(\bar{x}_i - x_i) + K_{Di}(\dot{\bar{x}}_i - \dot{x}_i) \quad (7)$$

where $\bar{x}_o$ and $\bar{x}_i$ denote reference values for x_o and x_i , respectively.

Once h_o and h_i have been synthesized, the second design requirement is met by computing the control torques at each arm as

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} J_{1a}^T(q_1) & O \\ O & J_{2a}^T(q_2) \end{pmatrix} \begin{pmatrix} h_{1a} \\ h_{2a} \end{pmatrix} + \begin{pmatrix} g_1(q_1) \\ g_2(q_2) \end{pmatrix} \quad (8)$$

where O denotes a (3×3) matrix of null elements and $(h_{1a}^T, h_{2a}^T)^T$ is given in (5).

Notice that the same impedance concept was used in [11] at the object level, but full dynamic compensation was performed on the basis of contact force measures and feedforward specification of internal forces was input.

Simulation Environment

The model of a controlled cooperative manipulator system is usually difficult to arrange in a set of explicit differential equations that allows simulation of the overall system using standard numerical packages. In fact, the equations are difficult to manage since multiple mechanical parts interact in a complex way.

A viable approach is to describe each mechanical part as a separate subsystem and leave the simulation tool the task of solving the interaction between subsystems. This has the advantage of writing modules constituted of few simple algebraic and/or differential equations describing the individual subsystems. Further, the number of needed different modules is limited because typical large-scale systems can be easily decomposed in small-scale systems of relatively few types. This is just the case of robotics; even a single manipulator is naturally modeled as a kinematic chain of links.

We have adopted LEGO which is a modular simulation package developed at the Automation and Computing Research Center of the National Electric Power Company (ENEL-CRA) [7]. A LEGO module essentially consists of a system of nonlinear algebraic and differential equations of the form

$$\begin{aligned} \dot{z} &= f(z, y, u, a) \\ 0 &= g(z, y, u, a) \end{aligned} \quad (9)$$

where u is the vector of input variables, z is the vector of state variables, y is the vector of output variables, and a is a vector of physical parameters (design data). LEGO's numerical features are:

- discretization of differential equations according to the trapezoidal rule and simultaneous resolution of both discretized and algebraic equations with the Newton-Raphson iterative method,
- use of sparse matrix techniques,

- steady-state computation which allows the selection of the unknowns among the input/output variables and uncertain parameters; obviously, the number of unknowns must be equal to the number of equations,
- inherent resolution of implicit equations.

Previous work on extensive use of LEGO in modeling and simulation of an industrial robot can be found in [11].

To simulate the system described by eqs. (1,2) —two arms and object— and eqs. (5-8) —controller— in LEGO, we have identified the following three elementary modules for the cooperative system.

LINK module

It is well-known that the most efficient method to compute the dynamic equations of a robotic manipulator is based on the Newton-Euler recursive algorithm [12]. The LINK module is naturally devised on the basis of Newton-Euler equations; in fact, forward recursion of velocities $\dot{x}_j^k$ and accelerations $\ddot{x}_j^k$ is performed through the open kinematic chain from the base link to the end effector, while a backward recursion propagates forces h_j^k from the end effector to the base link; also, the module performs forward propagation of link positions x_j^k to allow the computation of elastic contact forces at the end effector. From the schematic representation of the module for link k of arm j in Fig. 2, it can be observed that the actuation input to the module is the joint driving torque τ_j^k while the observed outputs are the joint variables q_j^k and velocities $\dot{q}_j^k$. Finally, the module is provided with the link's kinematic and dynamic parameters.

OBJECT module

The OBJECT module is devised to implement the elastic model of the object described by eq. (2). As shown in Fig. 2, the inputs to the module are the end-effector locations x_1 and x_2 while the outputs are the elastic forces h_1 and h_2 arising from object deformation. The module parameters are K_o and $\bar{x}$.

CONTROLLER module

The CONTROLLER module is devised to implement the control laws designed as in eqs. (5-8). Its inputs are the joint variables q_1, q_2 and velocities $\dot{q}_1, \dot{q}_2$; its outputs are the joint torques τ_1, τ_2 (Fig. 2). Further inputs are the reference values $\bar{x}_o, \bar{x}_i, \bar{x}_i, \bar{x}_i$ that are chosen according to the specified task. The feedback gain matrices K_{Po}, K_{Do} and K_{Pi}, K_{Di} are parameters for the module, together with the parameters characterizing the geometry of the arms and grasp.

Case Studies

We have tested the proposed control strategy for a cooperative system of two equal three-degree-of-freedom planar arms holding a disk-shaped object.

Therefore, three LINK modules are required for each arm. The simulation layout which illustrates the interconnection of modules is sketched in Fig. 3. Notice that the location of the base of each arm must be input to the respective first LINK module.

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

We locate a reference frame at the base of the first arm and an 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 1.4 from the reference frame along the x -axis.

The object spring matrix is $\mathbf{K}_o = 10^4 \mathbf{I}$ 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 $\tilde{\mathbf{x}}$ as used in (2).

The assigned grasp is such that the last link of each arm is aligned with the x -axis of the object frame; then, the location of the stick tips can be computed, as needed in (7,8). The controller PD actions have been tuned to $\mathbf{K}_{Po} = \text{diag}(5000, 5000, 500)$, $\mathbf{K}_{Do} = \text{diag}(700, 700, 70)$, $\mathbf{K}_{Pi} = \mathbf{O}$, $\mathbf{K}_{Di} = \text{diag}(700, 700, 70)$. We have chosen $\mathbf{K}_{Pi} = \mathbf{O}$ because we want to hold the object without deforming it in regard to its rest state.

In the first case study, the initial configuration of the system is $\mathbf{q}_1 = (0 \ \pi/2 \ -\pi/2)^T$, $\mathbf{q}_2 = (\pi \ -\pi/2 \ \pi/2)^T$, giving $\mathbf{x}_o = (0.7 \ 0.3 \ 0)^T$. The task is to reach the object location $\mathbf{x}_o = (0.6 \ 0.2 \ 0)^T$. The results show that the target location is reached satisfactorily (Figs. 4,5) with low values of internal forces (Figs. 6,7).

The second case study is aimed at analyzing the tracking performance of the proposed control scheme. The system is in the same initial configuration as above, and the task trajectory is specified as

$$\mathbf{x}_o = \begin{pmatrix} 0.7 + 0.05 \cos(2\pi t) \\ 0.3 + 0.05 \sin(2\pi t) \\ 0 \end{pmatrix} \quad 0 \leq t \leq 3.$$

The results reveal that appreciable tracking errors occur (Figs. 8,9); this is not surprising since the controller based on the simple laws (8) does not realize full dynamic compensation; however, it can be recognized that internal force effects are contained (Figs. 10,11).

Conclusions

A control strategy has been developed for cooperative manipulators holding an elastic object. The controller is designed in two stages: First, impedance-like behavior is prescribed for both external and internal object forces, remarkably avoiding force sensor measurements. Then, static force distribution with simple gravity compensation is performed to compute joint torques. The resulting control algorithm is computationally inexpensive and has been tested using a modular simulation package.

Two major conclusions can be drawn:

- The simulation of a complex dynamic system, such as a controlled cooperative arm system, is eased by decomposition into simple subsystems to be handled by a modular simulation package.
- The performance of the system under the proposed controller is rewarding when the task requirements are not demanding from a dynamic viewpoint; this is the case, for instance, of routine industrial manipulation tasks of the pick-and-place type.

Acknowledgements

This work was supported partly by *Ministero dell'Università e della Ricerca Scientifica e Tecnologica* under 60% funds and partly by *Consiglio Nazionale delle Ricerche* under contract n. 90.00355.PF67.

References

- [1] E. Nakano, S. Ozaki, T. Ishida, and I. Kato, "Cooperational control of the anthropomorphic manipulator 'MELARM'," *Proc. 4th Int. Symp. on Industrial Robots*, Tokyo, Japan, pp. 251-260, 1974.
- [2] C.D. Kopf and Yabuta, T., "Experimental comparison of master/slave and hybrid two arm position/force control," *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, pp. 1633-1637, 1988.
- [3] M. Uchiyama, N. Iwasawa, and K. Hakomori, "Hybrid position/force control for coordination of a two-arm robot," *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, pp. 1242-1247, 1987.
- [4] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Dynamic force/motion control of cooperative robot systems," *1990 ASME Winter Annual Meeting*, Dallas, TX, Nov. 1990.
- [5] 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.

- [6] 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.
- [7] R. Cori and S. Spelta, "Il sistema LEGO dedicato alla modellistica assistita da calcolatore per la simulazione dinamica di sistemi complessi," *ENEL-CRA Internal Report*, 804/88, 1988.
- [8] Z. Li, P. Hsu, and S. Sastry, "Dynamic coordination of a multiple robotic system with point contact," *Proc. 1988 Amer. Contr. Conf.*, Atlanta, GA, pp. 505-510, 1988.
- [9] T.J. Tarn, A.K. Bejczy, and X. Yun, "Design of dynamic control of two cooperating robot arms: Closed chain formulation," *Proc. 1987 IEEE Int. Conf. Rob. Autom.*, Raleigh, NC, pp. 7-13, 1987.
- [10] P. Chiacchio, L. Sciacivco, and B. Siciliano, "The potential of model-based control algorithms for improving industrial robot tracking performance," *IEEE Int. Work. on Intelligent Motion Control*, Istanbul, Turkey, pp. 831-836, 1990.

- [11] G. Magnani, P. Bologna, G. Rizzi, and G. Lovati, "Modelling and simulation of an industrial robot," *Proc. 1988 IEEE Int. Conf. on Robotics Automation*, Philadelphia, PA, pp. 1532-1538, 1988.
- [12] J.Y.S. Luh, M.W. Walker, and R.P.C. Paul, "On-line computational scheme for mechanical manipulators," *Trans. ASME J. of Dynamic Systems, Measurement, and Control*, vol. 102, no. 2, pp. 69-76, 1980.

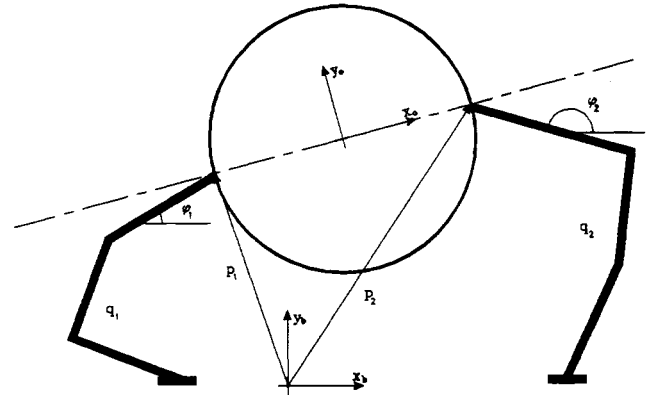


Fig. 1 A cooperative two-arm planar system

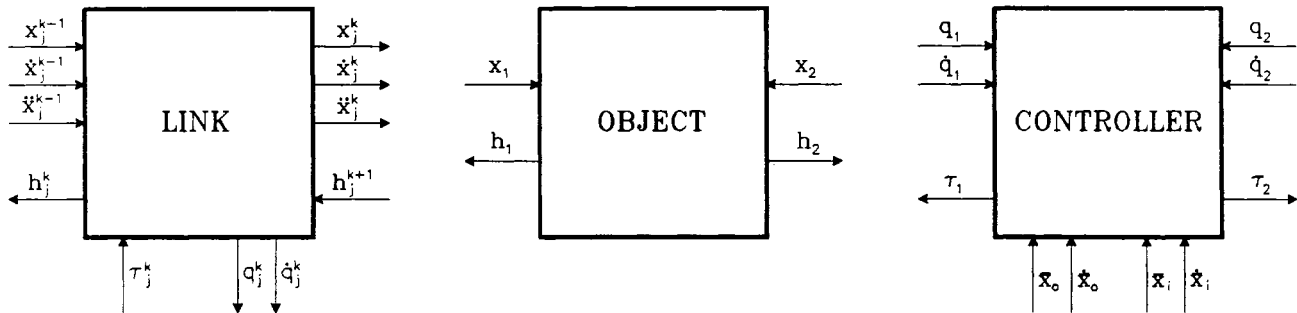


Fig. 2 Elementary modules

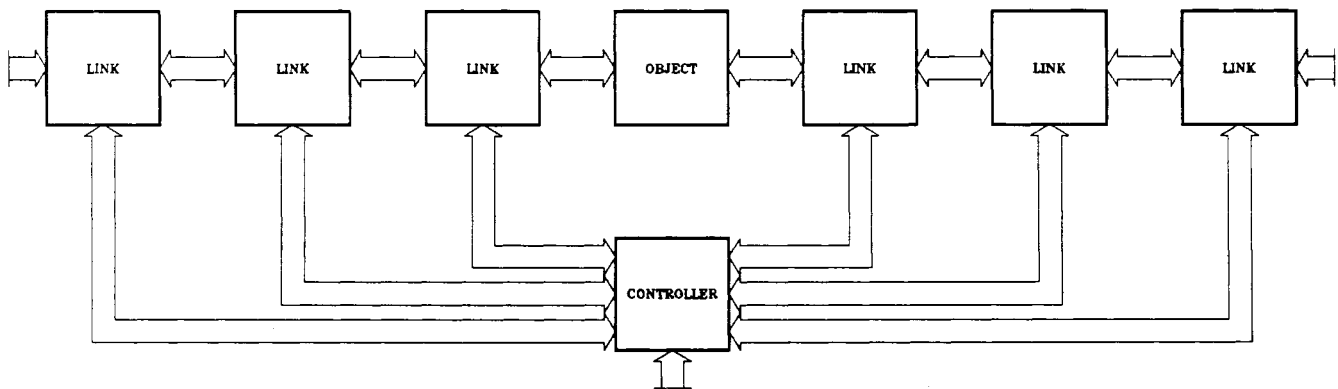


Fig. 3 Simulation layout for a controlled system of two 3-DOF arms holding a common object

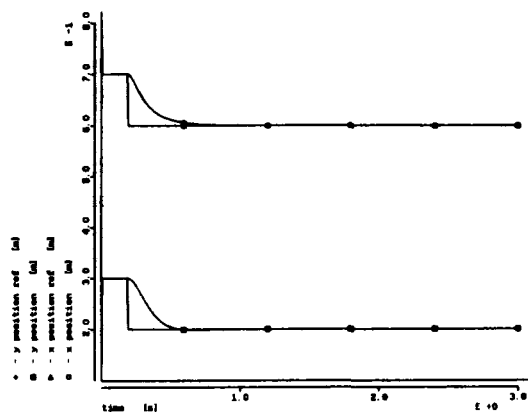


Fig. 4 Object position for the regulation test

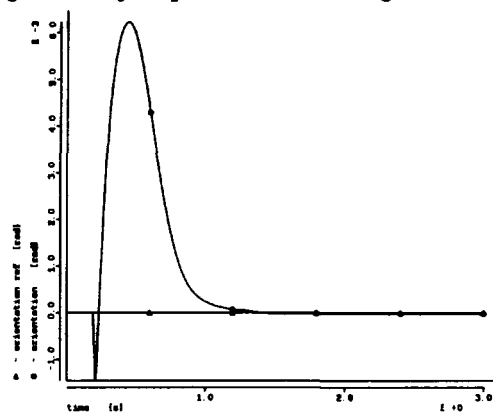


Fig. 5 Object orientation for the regulation test

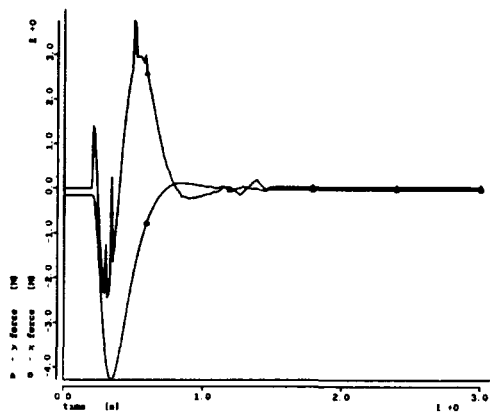


Fig. 6 Internal forces for the regulation test

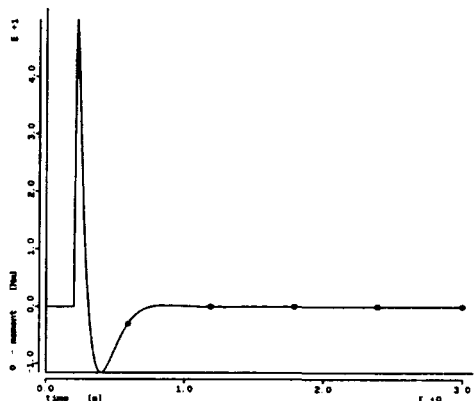


Fig. 7 Internal moment for the regulation test

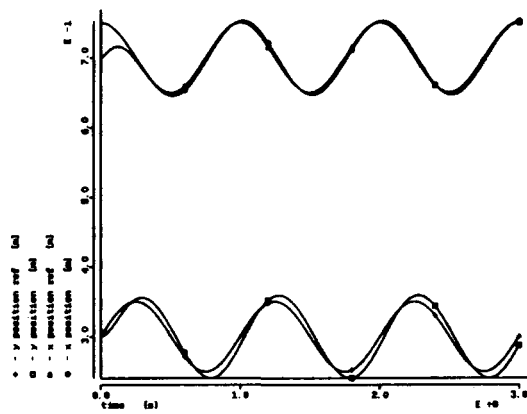


Fig. 8 Object position for the tracking test

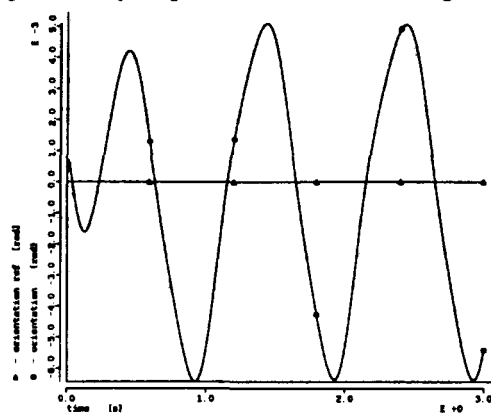


Fig. 9 Object orientation for the tracking test

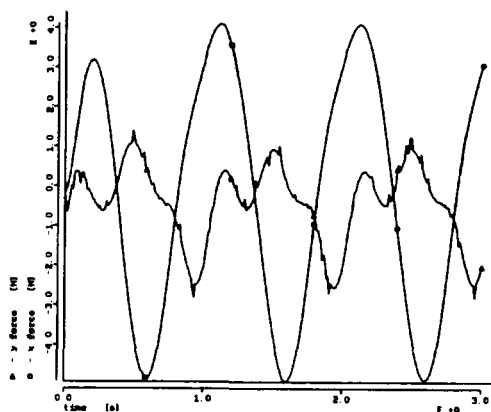


Fig. 10 Internal forces for the tracking test

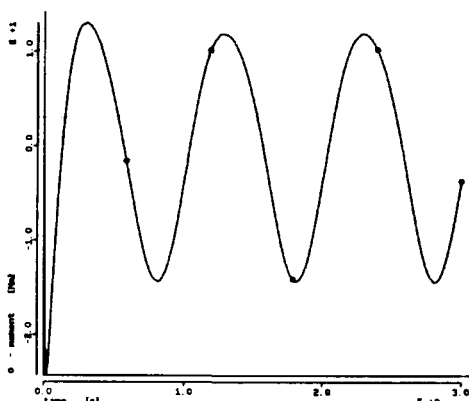


Fig. 11 Internal moment for the tracking test