Decentralised PD and PID robotic regulators

Y. Bestaoui, DES, MS

Indexing terms: Robotics, Cybernetics and robotics, Numerical analysis, Nonlinear systems

Abstract: The paper presents a new method of robotic manipulator control based on decentralised pole-placement feedback deduced from the computed torque method. An anticipatory action is included in the controller by ensuring subsequent desired joint positions, velocities and accelerations in the Newton-Euler equations of motion. The desired torques and the co-ordination parameters are computed in the first level (discrete) when the torques coming from the decentralised PD or PID regulators are computed in the second level (continuous). The parameters of the decentralised PD and PID regulators are given by simple laws. Decentralised control is appropriate for this control scheme because it is very easy to share calculations among the microprocessors with few interconnections. Then the stability of the regulator is investigated. Finally, the simulation results obtained for the PD and PID regulators are presented.

1 Introduction

Feedback control of robotic devices is usually organised on hierarchical principles. For robot control, hierarchical and decentralised control approach has become a necessity due to natural hierarchy in the control structure and the complexity of decision making in a centralised situation. In fact, the control of the overall system cannot be handled in a single level. Multiprocessor architectures with dedicated joint processors are often used to control industrial robots [1]. This approach ignores the coupling between joints and treats each joint as an independent servo-loop with constant feedback gain. The penalties of applying this static control strategy to an inherently dynamic system are the slow speed and considerable vibration of the robot arm. This approach has been completed by introducing a global feedback for compensation of inertial and gravitational forces [2].

Our primary objective, in the work described here, is to propose a controller that has a simple structure, is easily implemented and is capable of path following for manipulators. This paper presents two levels of a hierarchical structure designed for the co-ordinated control of a multiple-degrees-of-freedom manipulator. This algorithm uses the horizontal division notion, which takes into account the interconnected aspect existing in the robot manipulator system: the coupled degrees of freedom. Each degree of freedom is considered as a subsystem. The first stage is the reference torque and co-ordination parameters computing level. A decentralised control system is designed for each degree of freedom, where all the gains are calculated by simple linear laws. These two levels are derived from the computed torque method.

The following assumptions are made throughout this paper:

(A1) The joint positions and velocities are measurable and the values supplied by respective sensors are exact. No information is required about the joint accelerations

(A2) The actuators dynamics are negligible in comparison with the manipulator dynamics

(A3) The actuators' driving torques are not restricted, the task is correctly planned so that the actuators' torque limits are never reached

(A4) The sampling interval is small enough so that the discretisation effects can be neglected.

Assumptions (A1)–(A4) are very common and do not impose any severe constraints on the following discussion.

2 Mathematical model

The manipulator motion is represented by a mathematical model which is introduced in the following discussion. The manipulators are assumed to be made of rigid links. Any modes of unmodelled dynamics are neglected, such as those due to joint flexibilities; therefore the only uncertainties allowed in the robot model are those due to parameters like the payload, link mass, link dimensions etc. and external disturbances. We treat these uncertainties to compact sets, so that the control problem may be treated in a deterministic setting, instead of a stochastic one.

A priori information needed for control are a set of differential equations describing the dynamic behaviour of a manipulator. Two main approaches are used by most researchers to systematically derive the dynamic model of a manipulator: the Lagrange [6] and the Newton-Euler [7] formulations. The Lagrange method separates the inertial, Coriolis and centrifugal, and gravitational terms of the equations of motion. The Newton-Euler is a general recursive formulation and can handle manipulators with any number of links and any configuration. The major factor that contributes to the overall computational efficiency is the use of closed-form solutions in symbolic form, which reduces the number of required arithmetic operations.

The dynamics of a multilink articulated robot manipulator, excluding the actuator dynamics, gear friction and backlash, can be characterised by a set of nonlinear and coupled second-order differential equation:

$$\Gamma = A(q)\ddot{q} + B(q, \dot{q}) - G(q) \quad (1)$$

where Γ are $n \times 1$ external applied torques for joint actuators, q are $n \times 1$ joint angles, $\dot{q}$ is the $n \times 1$ joint velo-

Paper 6630D (C7, C9), first received 19th January 1988 and in revised form 6th January 1989

The authoress is with the Département de Physique, Institut d'Hydraulique, Université de Tlemcen, 13000 Tlemcen, Algeria

cities, $\ddot{q}$ are $n \times 1$ joint accelerations, $G(q)$ is the gravitational force vector, $B(q, \dot{q})$ the Coriolis and centrifugal force vector and $A(q)$ is an $n \times n$ inertia matrix.

The important aspect of robot manipulator control, trajectory tracking, is presented in the following Section.

3 Control system

Given the equations of motion of a manipulator, the control problem is to find the appropriate torques to servo all the joints of the manipulator in real time, to track a desired trajectory as closely as possible. In fact, if the plant model was perfect, then there would be no need for the feedback gains; we would have a perfect open-loop controller. However, we never have a perfect model of the plant and need the feedback loops to overcome the effects of modelling errors.

3.1 Computed torque technique

Industry generally uses classical techniques like PD or PID controllers, with constant parameters, to operate the robots that it employs. However, the recent literature has been investigating how to increase the speed and tracking accuracy of the manipulators. A number of control schemes have been proposed, such as the inverse technique [8], the computed torque technique [9, 10], table look-up methods [11], linear controllers [12] and nonlinear decoupling controllers [13]. In fact, Khalil [9] proved that the computed torque technique is equivalent to the nonlinear decoupling method.

The computed torque technique sets the basis on which much of the present literature on robotic trajectory control is based. In this method, we compute the necessary torques based on the inertial dynamics of the manipulator. For example, we may use Lagrangian or Newtonian mechanics to model the manipulator dynamics. In the control scheme, preceding the robot dynamics, an inverse model is incorporated. This inverse model calculates the torques needed for the reference trajectory. A great advantage is that the whole system (inverse model + robot) has an almost linear behaviour.

To control the manipulator, the control law [9] is proposed:

$$\Gamma = \Gamma^d + \Delta\Gamma \tag{2}$$

where

$$\Gamma^d = \hat{A}(q^d)\ddot{q}^d + \hat{B}(q^d, \dot{q}^d) + \hat{G}(q^d) \tag{3}$$

and

$$\Delta\Gamma = \hat{A}(q^d)[K_1(q^d - q) + K_2(\dot{q}^d - \dot{q})] \quad \text{(PD)} \tag{4}$$

or

$$\Delta\Gamma = \hat{A}(q^d)\left[K'_1(\dot{q}^d - \dot{q}) + K'_2(q^d - q) + K'_3 \int (q^d - q)\, dt\right] \quad \text{(PID)} \tag{5}$$

where $\hat{A}$ $\hat{B}$ and $\hat{G}$ are estimates of A, B and G, respectively, K_1 and K_2 are $n \times n$ constant diagonal gain matrices with K_{1j} and K_{2j} on the diagonals (idem for K'_1, K'_2 and K'_3). The desired trajectory of the manipulator is given by the trajectory planning system [4, 5, 15, 16].

This control law is chosen because, in the favourable situation of perfect knowledge of parameter values and no disturbances, the jth joint has closed-loop dynamics given by the error equation:

$$\ddot{e}_j + K_{1j}\dot{e}_j + K_{2j}e_j = 0 \quad \text{(PD)} \tag{6}$$

or

$$\ddot{e}_j + K'_{1j}\dot{e}_j + K'_{2j}e_j + K'_{3j}\int e_j\, dt = 0 \quad \text{(PID)} \tag{7}$$

where

$$e_j = q_j - q_j^d, \quad \dot{e}_j = \dot{q}_j - \dot{q}_j^d, \quad \ddot{e}_j = \ddot{q}_j - \ddot{q}_j^d$$

In this ideal situation, the K_{1j}, K_{2j}, K'_{1j}, K'_{2j} and K'_{3j} must be chosen to place the closed-loop poles of each joint so that each degree of freedom may react like a critical damping system, in this case,

$$K_{2j} = -p^2 \quad \text{and} \quad K_{1j} = 2p \quad \text{(PD)} \tag{8}$$

or

$$K'_{2j} = -3p^2, \quad K'_{1j} = 3p \quad \text{and} \quad K'_{3j} = p^3 \quad \text{(PID)} \tag{9}$$

where the pole p is chosen so that the system will be stable.

Servosystems must be critically damped to achieve satisfactory fast positioning. Overshooting may be quite undesirable in some cases; for instance, if the position-controlled motion is only a prelude to a compliant task. The same pole for all degrees of freedom is chosen to avoid the delay due to different poles.

The principal difficulty is obviously related to the computability of the control in real time. The original idea was to compute the driving torques as a function of the desired motion represented in joint variables, velocities and accelerations, and its drawbacks were mainly due to the real-time evaluation of the complete dynamics. Because of the nature of the Newton-Euler formulation and its method of systematically computing the joint torques, the computations are far less numerous than those of the Lagrange method [7]. We shall then use the Newton-Euler equations to compute the nominal torques along the desired trajectory given by the trajectory planning system [15, 16].

3.2 Regulator gains adjustment

In reality, eqns. 2–5 are too difficult to implement in real time, so it is proposed that the PD or PID regulator parameters, given by eqn. 4 or 5, will be approximated by simple laws and that a hierarchical structure will be used. The computations will be divided into two groups: continuous and discrete (Fig. 1). The computing of the desired torques Γ^d and the co-ordination parameters w belong to the discrete level, and the PD or PID regulators torques to the continuous one.

For the purpose of obtaining a decentralised control in the joint space, the inertial matrix can be redefined as follows

$$\hat{A}(q^d) = A' + \Delta A(q) \tag{10}$$

A' contains only the diagonal terms of $\hat{A}(q)$, and $\Delta A(q)$ accounts for parameter uncertainties along with q-dependent and coupling terms of $\hat{A}(q)$.

The control law becomes then:

$$\Gamma_i = \Gamma_i^d + k_{pi}(q_i^d - q_i) + K_{vi}(\dot{q}_i^d - \dot{q}_i) + w_i \quad \text{(PD)} \tag{11}$$

or

$$\Gamma_i = \Gamma_i^d + k_{pi}(q_i^d - q_i) + K_{vi}(\dot{q}_i^d - \dot{q}_i) + w_i + K'_{Ii}\int (q_i^d - q_i)\, dt \quad \text{(PID)} \tag{12}$$

where

$$K_{pi} = p^2 A_{ii}(q^d(t))$$

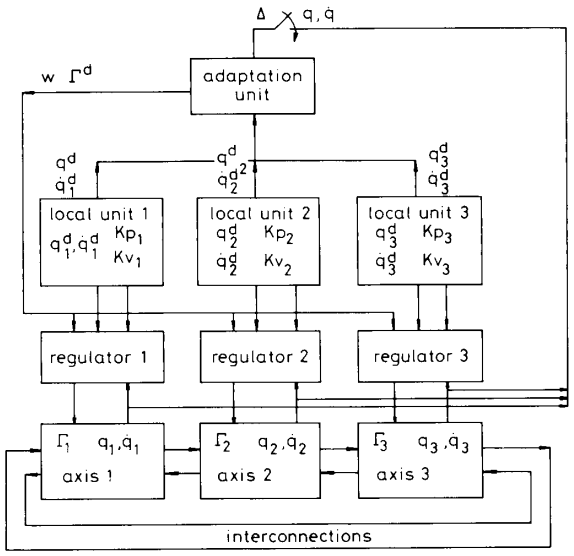

Fig. 1 *Sharing of the computations: continuous and discrete*

and

$$K_{vi} = -2pA_{ii}(q^d(t)); \quad i = 1, \ldots, n \qquad \text{(PD)} \quad (13)$$

or

$$K_{pi} = 3p^2 A_{ii}(q^d(t)), \quad K_{vi} = -3pA_{ii}(q^d(t))$$

and

$$K_{Ii} = -p^3 A_{ii}(q_i^d); \quad i = 1, \ldots, n \qquad \text{(PID)} \quad (14)$$

and

$$w_i = \sum_{j=1, j\neq i}^{n} A_{ij}(p^2 e_j - 2p\dot{e}_j) \qquad \text{(PD)} \quad (15)$$

or

$$w_i = \sum_{j=1, j\neq i}^{n} A_{ij}\left(3p^2 e_j - 3p\dot{e}_j + p^3 \int e_j \, dt\right) \qquad \text{(PID)} \quad (16)$$

The terms A_{ii} $(i = 1, \ldots, n)$ consists of trigonometrical elements, and their evolution is very smooth, so we can attempt to approximate these terms by simple laws. This step is robot-type dependent. Each term of $\hat{A}$, n in all, may be plotted against the normalised time τ. In general, linear interpolation is sufficient:

$$A_{ii} = \alpha_i \tau + \beta_i \qquad (17)$$

$\tau = t/t_f$ is the normalised time with t_f the predicted arrival time and the coefficients α_i and β_i are computed as follows:

Let A_{ii}^0, A_{ii}^f and A_{ii}^m be the values of the inertial matrix diagonal terms evaluated at the initial, final and medium points, respectively, then for

$$0 \leqslant \tau \leqslant 0.5, \alpha_i = 2(A_{ii}^m - A_{ii}^0) \text{ and } \beta_i = A_{ii}^0 \qquad (18)$$

$$0.5 \leqslant \tau \leqslant 1, \alpha_i = 2(A_{ii}^f - A_{ii}^m) \text{ and } \beta_i = 2A_{ii}^m - A_{ii}^f \qquad (19)$$

and the parameters K_{pi} and K_{vi} (or K'_{pi}, K'_{vi} and K'_{Ii}) are calculated via eqns. 13 (or 14). The computation of A^0, A^f, A^m, α and β can be done offline. Online, it will suffice to compute eqns. 17 and 13 or 14. The functions Γ^d and w being very regular, they can be approximated by stepwise functions:

$$w^m = w, \quad \Gamma^{dm} = \Gamma^d, \quad \text{for } m\Delta \leqslant t < (m+1)\Delta$$

Γ^{dm} are the desired torques calculated for q^{dm}, mth sampling of the desired positions q^d, q^m is the mth sampling of the position q and w^m the mth co-ordination parameter.

Thus, the PD regulator presented computes preceding motor torques as

$$\Gamma = \Gamma^{dm} + w^m + K_p(q^d - q) + K_v(\dot{q}^d - \dot{q}) \qquad (20)$$

with

$$w^m = -(\hat{A}(q^{dm}) - A'(q^{dm}))(p^2 e^m - 2p\dot{e}^m) \qquad (21)$$

$$K_p = p^2 M \quad \text{and} \quad K_v = -2pM \qquad (22)$$

$$M = \text{diag}\,(\alpha_1\tau + \beta_1, \alpha_2\tau + \beta_2, \ldots, \alpha_n\tau + \beta_n) \qquad (23)$$

The PID regulator is analogous to the eqns. 20–23, except that we must add the integral term and change the coefficients of the pole p:

$$\Gamma = \Gamma^{dm} + w^m + K_p(q^d - q) + K_v(\dot{q}^d - \dot{q}) + K_I \int (q^d - q)\, dt \qquad (24)$$

$$K_I = -p^3 M, \quad K_p = 3p^2 M \quad \text{and} \quad K_v = -3pM \qquad (25)$$

$$w^m = -(\hat{A}(q^{dm}) - A'(q^{dm}))\left(-p^3 \int e^m \, dt + 3p^2 e^m - 3p\dot{e}^m\right) \qquad (21)$$

4 Multi-microprocessor implementation of the control system

Hierarchical control and decentralised observers are appropriate for this control scheme, because it is very easy to share the calculation among microprocessors with few interconnections. This considerably increases the reliability of such a control system. The most suitable use of microprocessors is for real-time control, as there is a substantial need for robot controllers with enough power to execute servocontrol at high sampling rates. To work, well with a multiprocessor system, a procedure should have a high percentage of tasks which can be executed in parallel, without requiring interprocessor communication.

One possible version of the multiprocessor solution of the observer and control is described here. Fig. 2 shows the block scheme of such a solution. In this case, one

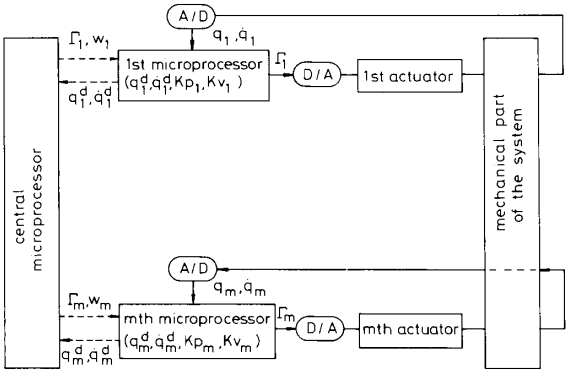

Fig. 2 *Block scheme of a possible version of the multiprocessor solution*

microprocessor is adjoined which realises the local observer and local regulator for the ith joint. Each local microprocessor must be equipped with an analogue/digital convertor: on the basis of these, the system state is reconstructed by means of the local observer on each sampling interval, and the control (being fed to the digital/analogue convertor) is calculated. The task of the ith microprocessor will be to compute the reference trajectories, the gains given by eqns. 13–14, the difference between the desired and actual trajectory e_i, and the motors torques defined by eqn. 20 or 24.

The central microprocessor synchronises the work of all microprocessors and computes the nominal torques using Newton-Euler equations of motion (eqn. 3) and the co-ordination parameters (eqns. 15–16). Its sampling period may be higher than those of the local microprocessors, to have the time necessary to compute the centralised mathematical model of the robot and the co-ordination parameters, using desired positions, velocities and accelerations.

The basic advantage of such a microprocessor structure is that there is relatively little communication among the microprocessors, i.e. a small exchange of information among the microprocessors on one sampling interval. All the links among microprocessors are via the central microprocessor which synchronises the work of all the local microprocessors, determines the sampling time, sends data about the nominal torques and co-ordinating parameters, and receives data about the reference trajectories. Between the local microprocessors themselves, which determine the local observers and regulators, there is no exchange of information.

5 Stability analysis

The task imposed can be defined as the task of transferring the system state from an initial point to a defined point in the state space, in a finite time interval $[0, t_f]$. It is required that, during the transfer, the system state remains within a bounded region C around the desired trajectory. The system is considered to be practically stable, if, for all initial points, the position q remains in C and, finally, the desired point is attained. The method of proof is as follows:

The control given by eqn. 20 is applied to the actual manipulator given by eqn. 1. The resultant closed-loop equations of motion are rewritten in a form which makes stability analysis possible. The analysis is done, first, for the PD regulator, which is divided into two parts. First, we investigate the effect of the linear approximation of the diagonal inertial matrix elements on the stability, assuming that the computation of the desired torques and the co-ordination of parameters is done continuously. In the second part, we consider the effect of the sample interval Δ on stability. Finally, the stability analysis of the PID regulator is performed.

We deal now with the linear approximation effects. Let C be a compact and connex set in $R^n \times [0, t_f]$, the initial point $(q(0), 0)$, the final point $(q(t_f), t_f)$ and the desired trajectory in the interval $[0, t_f]$ belong to C.

The robot dynamics are described by eqn. 1. The model is considered to be approximative. It is interesting to study the position error, so that eqns. 2 and 3 are equalised:

$$\begin{aligned} A(q)\ddot{q} + B(q, \dot{q}) - G(q) &= \hat{A}(q^{dm})\ddot{q}^{dm} + \hat{B}(q^{dm}, \dot{q}^{dm}) - \hat{G}(q^{dm}) \\ &\quad + [\hat{A}(q^{dm}) - A'(q^{dm})][-p^2 e^m + 2p\dot{e}^m] \\ &\quad + M(-p^2 e + 2p\dot{e}) \end{aligned} \tag{26}$$

We obtain then a differential equation of the following form:

$$\ddot{e} + \mu A_1(q, t)M(0.25\mu e + \dot{e}) + s(q, \dot{q}, t) = 0 \tag{27}$$

where

$$\mu = -2p \tag{28}$$

$$A_1(q, t) = [\hat{A}(q^{dm}) + \Delta A(q)]^{-1} \tag{29}$$

$$\begin{aligned} s(q, \dot{q}, t) = \delta\ddot{q} + A_1(q, t)[\Delta A\ddot{q}^m + \Delta B + \Delta G \\ - (A(q^m) - A'(q^m))(0.25\,\mu e^m + \dot{e})] \end{aligned} \tag{30}$$

$$\left.\begin{aligned} \Delta A(q) &= A(q) - \hat{A}(q^{dm}) \\ \Delta B(q) &= B(q, \dot{q}) - \hat{B}(q^{dm}, \dot{q}^{dm}) \\ \Delta G(q) &= G(q) - \hat{G}(q^{dm}) \end{aligned}\right\} \tag{31}$$

$$\delta q = qd - q^m, \quad \delta\dot{q} = \dot{q}d - \dot{q}^m, \quad \delta\ddot{q} = \ddot{q}d - \ddot{q}^m \tag{32}$$

these differences can come from inaccurate parameter identification (mass, inertia, ...) or from unmodelled dynamics, or truncated calculus.

The robot system is a physical system, the $(n \times n)$ matrix function $A_1(q, t)$ being C^1, A_1 is the inverse of the inertial matrix, and is symmetric positive-definite. The matrix M, being a diagonal matrix, is symmetric. To show that M is positive-definite, we must check if all the elements are positive, i.e.

$$\alpha_i \tau + \beta_i > 0 \quad \text{for } t \in [0, t_f] \text{ and } i = 1, \ldots, n$$

Exprs. 18 and 19 allow us to ensure that these elements are positive, taking into account the fact that the inertial matrices A^0, A^m and A^f are positive-definite. The vector function s is C^1 on an open set included in $R^n \times R \times R^n$ containing $C \times R^n((q, t) \in C, q \in R^n)$.

This differential eqn. 27 can also be stated in the classical form:

$$\frac{dX}{dt} = F(X, t) \tag{33}$$

where

$$X = \begin{pmatrix} e \\ \dot{e} \end{pmatrix} \quad F(X, t) = \begin{pmatrix} \dot{e} \\ -\mu A_1 M(0.25\,\mu e + \dot{e}) - s \end{pmatrix} \tag{34}$$

To facilitate stability in the proof, the following notations are used:

a^m = least eigenvalue of matrix M

a^M = greatest eigenvalue of matrix M

and let the constants σ^m, σ^M/β^m defined as

$$0 < \sigma^m \leqslant \inf_{(q, t) \in c} (a^m(q, t))$$

$$\sigma^M \geqslant \sup_{(q, t) \in c} (a^M(q, t))$$

$$0 < \beta^m \leqslant 1$$

$$b_{max} \geqslant 1$$

We can then state the following theorem:

Theorem: If

$$\mu > \mu_0$$

The initial conditions are such that

$$|e(0)| + |\dot{e}(0)|/\mu < (1 - 1/\alpha)\beta^m(\sigma^m/\sigma^M)^{1/2}/b_{max} \qquad \alpha > 1 \quad (35)$$

The function s is uniformly bounded, then the differential equation solution of $e(t)$ exists and is unique on $[0, t_f]$:

$$\forall t \in [0, t_f], \quad |e(t)| \leqslant \rho, \quad |\dot{e}(t)| \leqslant \rho' \qquad (36)$$

Proof: All the functions involved in the differential equation being C^1, the function $F(X, t)$ is also C^1 on an open set containing $C \times R^n$. In fact, the application of a classical theorem of the differential equation theory allows the conclusion to the local existence and unity of the solution $\{e(t)\}$ when $(e(0), 0) \in C$.

We must show now that, by imposing supplementary conditions on the gain μ [17], the solutions are prolongated until the instant t_f, with

$$\forall t \in [0, t_f], \quad |e(t)| < \rho$$

Let μ_0 be a positive scalar defined as

$$\mu_0 = K/(\sigma^m)^{1/2}/\rho/(1 - 1/\alpha) \qquad (37)$$

The next step is the proof of the unity and boundedness of $e(\tau)(0 \leqslant \tau \leqslant t_f)$ with the boundedness of $|e(t)|$, by a constant independent of τ. For this, let us study the differential equation

$$\dot{Q} = -0.25\ \mu A_1(q, t)MQ + \mu(Q + I)(Q + I) \qquad (38)$$

with the initial condition $Q(0) = 0$

The existence and unity of $Q(t)$ on $[0, \tau_1](\tau_1 \leqslant \tau)$ follows from the existence and unity of $q(t)$ and $\dot{q}(t)$ on $[0, \tau]$.

Let q_i be the ith column of the matrix Q:

$$q_i = Qr_i \qquad (39)$$

where r_i is the unity vector:

$$r_i^T = (0, 0, \ldots, \underset{\substack{\uparrow \\ i\text{th rank}}}{1}, 0, \ldots, 0)$$

If we multiply eqn. 38 by $q_i^T M$ and r_i, we obtain

$$0.5\left(\frac{d}{dt}(q_i^T M q_i)\right) = -0.25\ \mu q_i^T M A_1 M q_i + z_i' \qquad (40)$$

with

$$z_i' = \mu q_i^T M(Q + I)(Q + I)r_i - \tfrac{1}{2}q_i^T \dot{M} q_i \qquad (41)$$

$$\dot{M} = \text{diag}\ (\alpha_1/t_f, \alpha_2/t_f, \ldots, \alpha_n/t_f) \qquad (42)$$

If we define

$$y_i = Uq_i \qquad (43)$$

where U is the square-root matrix of the symmetric positive-definite matrix M:

$$M = U^2 \qquad (44)$$

then

$$U = \text{diag}\ ((\alpha_1\tau + \beta_1)^{1/2}, \ldots, (\alpha_n\tau + \beta_n)^{1/2}) \qquad (45)$$

Eqn. 40 can be rewritten as

$$0.5\frac{d}{dt}|y_i^2| = -0.25\ \mu y_i^T U A_1 U y_i + z_i' \qquad (46)$$

with

$$z_i' = 0.25\ \mu y_i^T U(Q + I)(Q + I)r_i - 0.5 y_i^T U^{-1}\dot{M}U^{-1}y_i \qquad (47)$$

Applying the classical inequalities on matrical norms, we obtain:

$$\begin{aligned} |z_i'| \leqslant (\mu/4)\{&(a^M/a^m)^{1/2}|Q||y_i^2| \\ &+ (a^M)^{1/2}|Q||y_i| + (a^M/a^m)^{1/2}|y_i^2| \\ &+ (a^M)^{1/2}|y_i| + (a^m)^{1/2}|y_i|\} \\ &+ 0.5(a^M)^{1/2}|\dot{M}||y_i^2| \end{aligned} \qquad (48)$$

We wish to prove that there is no instant t_1 such that Q is unbounded. Suppose that there exists an instant t_1 $(0 \leqslant t_1 \leqslant \tau_1)$ and a column q_i of Q such that

$$|y_i(t_1)| = (\sigma^m)^{0.5}\beta^m(\alpha n)^{-1} \qquad (49)$$

$$\forall j, \quad (1 \leqslant j \leqslant n),$$

$$\forall t \in [0, t_1[: |y_j(t)| < (\sigma^m)^{1/2}\beta^m(\alpha n)^{-1}$$

with

$$|Q| \leqslant \sum_{j=1}^{n} |q_j| \quad \text{and} \quad |q_j| \leqslant |U^{-1}||y_j|$$

then, necessarily, at the instant t_1:

$$\begin{aligned} |Q(t_1)| &\leqslant |U^{-1}(t_1)|(\sigma^m)^{1/2}\beta^m\alpha^{-1} \\ &\leqslant (\sigma^m/a^m(t_1))^{1/2}\beta^m\alpha^{-1} \end{aligned} \qquad (50)$$

Using expr. 50 substituted into expr. 48, we have

$$\begin{aligned} |z_i'(t_1)| \leqslant \{&\mu(a^M/a^m)^{1/2}[(\beta^m/\alpha)(\sigma^m/a^m)^{1/2} \\ &+ (n + 1) + n(\alpha/\beta^m)(a^m/\sigma^m)^{1/2}] \\ &+ 0.5|\dot{M}|/a^m\}_{t=t_1}|y_i(t_1)|^2 \end{aligned} \qquad (51)$$

and, consequently,

$$\frac{d}{dt}|y_i^2|_{t=t_1} < 0 \qquad (52)$$

y_i being continuously differentiable on $[0, \tau_1[$, this inequality contradicts the assumption of existence of the instant t_1 as $|y_i|$ cannot attain, for the first time, the constant $(\sigma^m)^{0.5}\beta^m(\alpha n)^{-1}$ with a negative derivative in this instant. We have then to establish that

$$\forall t \in [0, \tau_1[, \ i \in [1, n]; |y_i(t)| < (\sigma^m)^{0.5}\beta^m(\alpha n)^{-1} \qquad (53)$$

Let $[0, \tau_2[(\tau_2 \leqslant \tau_1)$, the greatest time interval included in $[0, \tau_1]$ where $|e(t)| \leqslant \rho$. Using eqns. 53 and 43 immediately gives

$$\forall t \in [0, \tau_2[, \quad |Q(t)| < \beta^m/\alpha) \qquad (54)$$

This constant is independant of τ_2 and

$$\forall t \in [0, \tau_2[, \quad |e(t)| < \rho$$

This theorem shows that, if the initial errors $|e(0)|$ and $|\dot{e}(0)|$ are sufficiently low, then $e(t)$ belongs to the neighbourhood of 0 for $t \in [0, t_f]$.

Now, we evaluate the effect of changing the sampling period Δ on the performance of the proposed control law. The gain matrices are a function of the sampling rate of the control system. This algorithm admits a distributed implementation, whereby n processes perform computations and exchange messages with the end goal of minimising the difference between the real and the desired

trajectories. If all n local processors communicate to each other their partial results, at each instance of time, and perform computations synchronously, the distributed algorithm is mathematically equivalent to a single processor algorithm, and its convergence may be studied by conventional means.

From a classical theorem [19], we have

$$\|\Delta A\| \leqslant k\Delta \tag{55}$$

with k defined as the upper bound of the time derivative of $\hat{A}$:

$$\left\| \frac{d\hat{A}}{dt} \right\| \leqslant k \tag{56}$$

The same notations are employed for ΔB and ΔG.

If Δ is beyond a certain threshold ($\Delta' \leqslant \Delta \leqslant t_f$), an oscillation may exist because of the application of a step to a nonlinear system. The oscillation is, then, approximately at the sample frequency. The remedy would then be the diminution of this step, then the diminution of the gain μ. In practice, smaller gains are always preferable, when possible, so as to minimise sensitivity to sensor noise and oscillation occurrence.

The same proof is valid for a proportional-integral derivative regulator. For the PID regulator, eqn. 27 then becomes

$$\ddot{z} + \mu A_1(q, t)M(-\mu^2/27z + \mu/3\dot{z} + \ddot{z}) + s(q, \dot{q}, t) = 0 \tag{57}$$

where

$$\mu = -3p \tag{58}$$

$$s(q, \dot{q}, t) = \delta\ddot{z} + A_1(q, t)[\Delta A\ddot{z}^m + \Delta B + \Delta G - (\hat{A}(q^m) - A'(q^m))(-\mu^2/27z^m + \mu/3\dot{z}^m + \ddot{z})] \tag{59}$$

$$\dot{z} = e, \quad \ddot{z} = \dot{e}, \quad \dddot{z} = \ddot{e} \tag{60}$$

Eqn. 57 can also be stated in the classical form:

$$\frac{dX}{dt} = F(X, t) \tag{61}$$

where

$$X = \begin{pmatrix} z \\ e \\ \dot{e} \end{pmatrix}$$

and

$$F(X, t) = \begin{pmatrix} e \\ \dot{e} \\ -\mu A_1 M(-\mu^2/27z + \mu/3\dot{z} + \ddot{z}) - s \end{pmatrix} \tag{62}$$

The same proof as shown previously (eqns. 30–55) is valid for this new function F. It will be omitted.

This Section has been treating perturbations in the feedback law. It is shown that the situation is robust: steady-state errors are small and stability is maintained, if errors in the feedback law are sufficiently small.

6 Simulation results

Many numerical simulations to the first three joints of the robotic manipulator MA23 and the Stanford arm are illustrated in this Section to test the efficiency of the proposed control scheme. Fig. 3 shows schematic diagrams of these manipulators.

To obtain comprehensive information about the trajectory tracking capabilities of this technique, the algorithm has been evaluated over eight different operational environments. The eight test configurations can be broken down into two blocks:

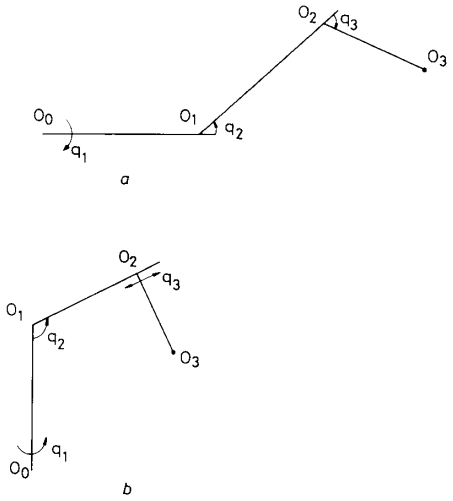

Fig. 3 *Configurations of the studied manipulators*
a MA23 manipulator
b Stanford arm

(*a*) when the nominal mass of the third axis and the load ($m_3 = 4$ kg) is overestimated by 50%, with respect to the assigned mass ($m_3 = 6$ kg)

(*b*) when the nominal mass of the third axis and the load ($m_3 = 4$ kg) is underestimated by 50%, with respect to the assigned mass ($m_3 = 2$ kg).

The nominal values were used in the computation of the nonlinear desired feedback. The assigned values were used in the computational implementation of the robot arm.

In the simulation program, integrations were performed by the 4th-order Runge-Kutta method with a step size of $\delta = 0.0025$ s. The values of the discretisation period Δ, of the nominal torques Γ^d and the coordination parameters w, are (*a*) 0.0025 s, (*b*) 0.01 s (*c*) 0.025 s and (*d*) 2 s.

The initial and final conditions are

$$q_0 = (0, 0, 0) \quad \text{and} \quad q_{goal} = (1, 1, 0.5)$$

the velocity and acceleration constraints are:

$$\dot{q}_{max} = (5, 5, 5) \text{ rad/s (or m/s)}$$

$$\ddot{q}_{max} = (7.5, 7.5, 7.5) \text{ rad/s}^2 \text{ (or m/s}^2\text{)}$$

and the pole chosen was $p = -30$.

The same experiments were done for the two manipulators. The case (*a*) ($\delta = 0.0025$ s and $\Delta = 0.0025$ s) corresponds, in fact, to the computed torque method and the case (*d*) ($\delta = 0.0025$ s and $\Delta = 2$ s) to the classical PD or PID regulator.

Figs. 4–7 show the evolution of the first two diagonal elements of the desired inertial matrix $\hat{A}^d_{ii}$ and their linear approximation, against the time, respectively, of the MA23 manipulator and the Stanford arm. The third curve shows how the real inertial element is related to the load mass. In the nominal cases ($m_3 = 4$ kg), A^d_{11} varies between 0.1 and 1.1 kg m^2 and, in the real case ($m_3 =$ 2 kg), A_{11} varies between 0.1 kg m^2 and 0.65 kg m^2, for

the MA23 manipulator. In the case of the Stanford arm, the third curve visualises the diagonal elements when $m_3 = 6$ kg.

The variations of payload parameters cause variation of the moments of inertia around the axes of the robot joints. The local controller is synthesised for each subsystem, taking into account the moment of inertia of the robot mechanism.

Parts (*a*) of Figs. 8–23 give the error norm between the desired and real positions in Cartesian space, respectively, for the MA23 manipulator and the Stanford arm, and the parts (*b*) of Figs. 8–23 show the joint error against time, for the same manipulators.

The first simulation result is that the error becomes null at the end of the motion, with the PID regulator. There is no overshooting nor oscillation. It is a classical remark that the integration term nullifies the static error. The results of the PID regulator are better than those of the PD regulator, with little difference in the number of operations. It is also shown that the difference between the error of the $\Delta = 0.0025$ s and $\Delta = 0.010$ s cases is really negligible. We can then conclude, from that important result, that the decentralised PID regulators can react continuously, when the desired torques and coordination parameters are computed with a period of 0.01 s.

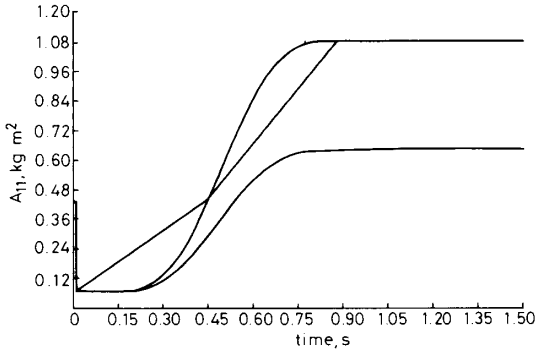

Fig. 4 *Evolution of the MA23 manipulator first diagonal element of the desired inertial matrix, of its linear approximation and of the first diagonal element of the real inertial matrix, when the third axis and load mass is 2 kg*

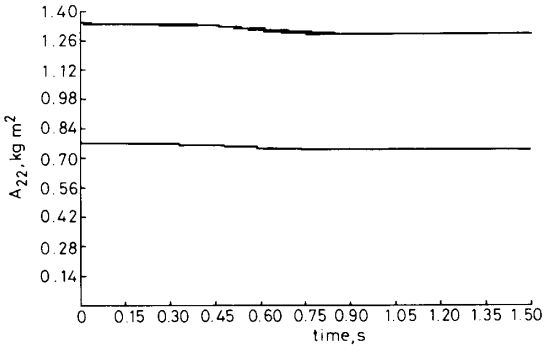

Fig. 5 *Evolution of the MA23 manipulator second diagonal element of the desired inertial matrix, of its linear approximation and of the second diagonal element of the real inertial matrix, when the third axis and load mass is 2 kg*

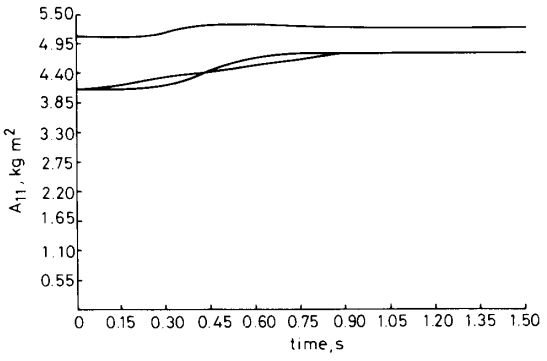

Fig. 6 *Evolution of the Stanford arm first diagonal element of the desired inertial matrix, of its linear approximation and of the first diagonal element of the real inertial matrix, when the third axis and load mass is 6 kg*

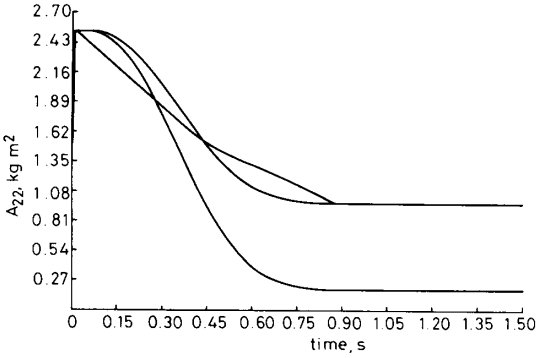

Fig. 7 *Evolution of the Stanford arm second diagonal element of the desired inertial matrix, of its linear approximation and of the second diagonal element of the real inertial matrix, when the third axis and load mass is 6 kg*

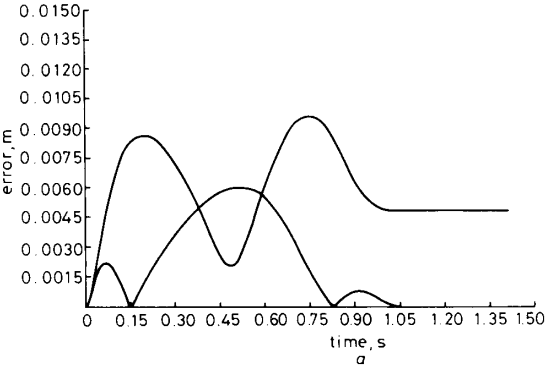

Fig. 8 *MA23 manipulator, pole = −30, $\Delta = 0.0025$ s, $m_3 = 6$ kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

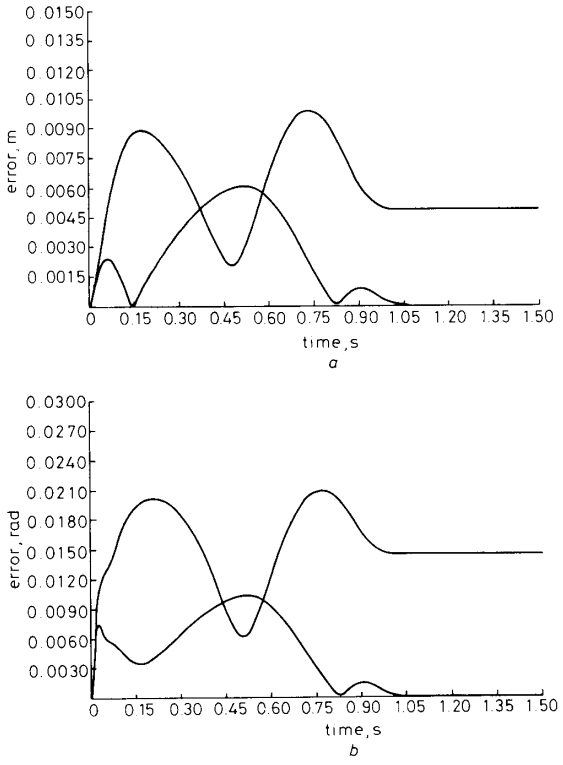

Fig. 9 *MA23 manipulator, pole = −30, Δ = 0.01 s, m_3 = 6 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

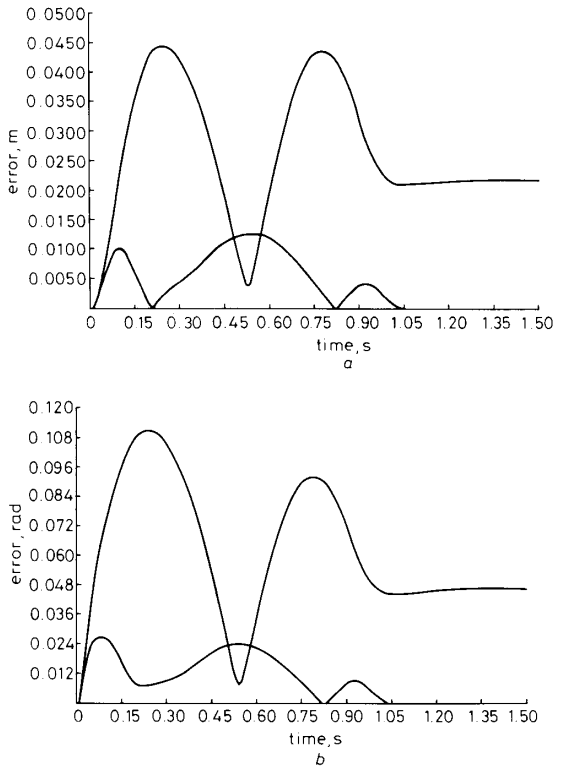

Fig. 11 *MA23 manipulator, pole = −30, Δ = 2 s, m_3 = 6 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

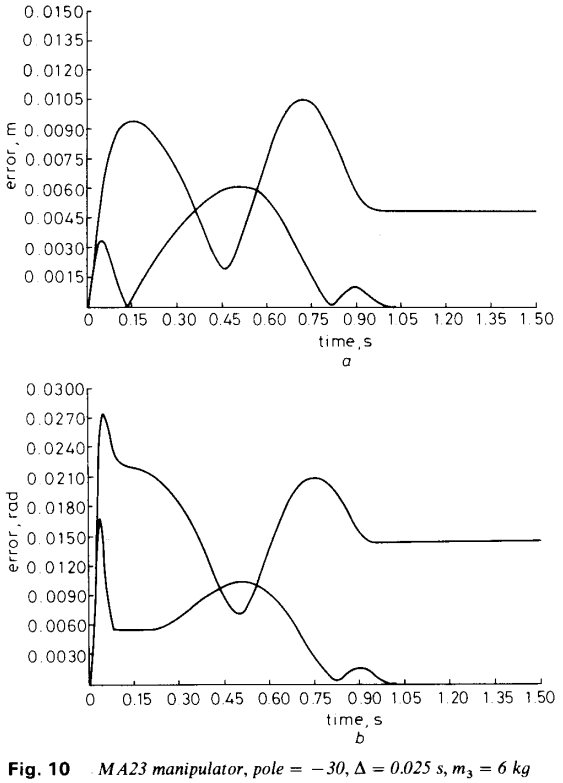

Fig. 10 *MA23 manipulator, pole = −30, Δ = 0.025 s, m_3 = 6 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

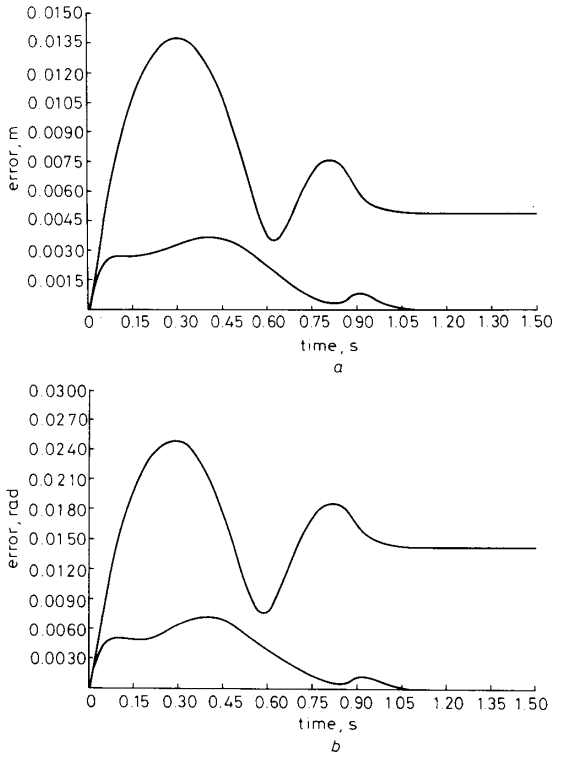

Fig. 12 *MA23 manipulator, pole = −30, Δ = 0.0025 s, m_3 = 2 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

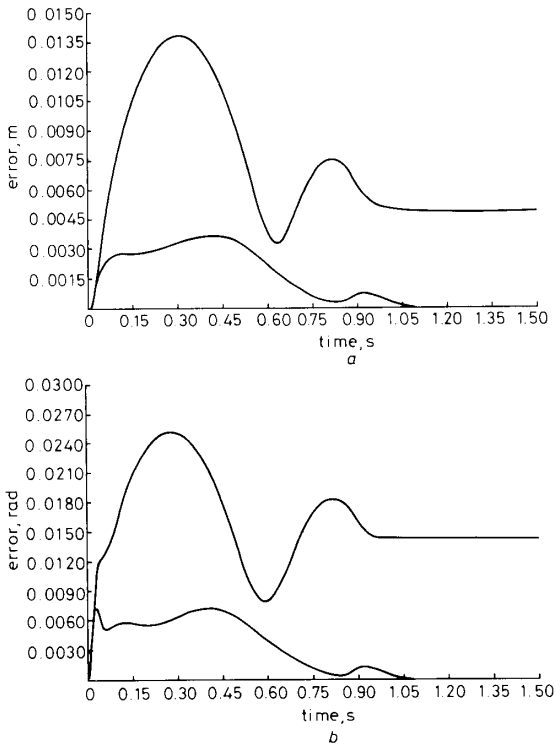

Fig. 13 *MA23 manipulator, pole* $= -30$, $\Delta = 0.01$ *s*, $m_3 = 2$ *kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

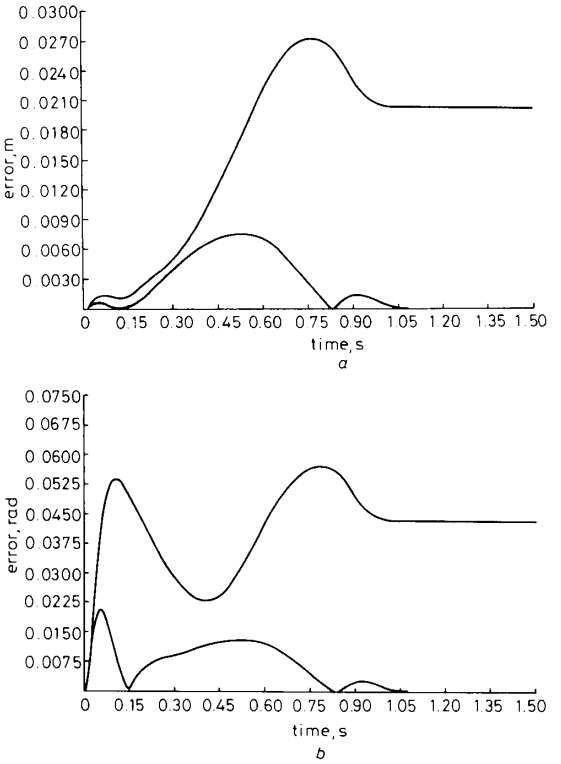

Fig. 15 *MA23 manipulator, pole* $= -30$, $\Delta = 2$ *s*, $m_3 = 2$ *kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

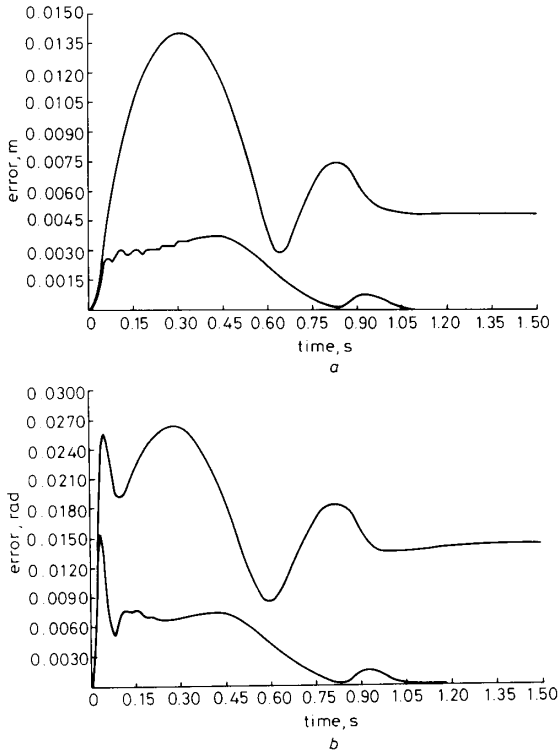

Fig. 14 *MA23 manipulator, pole* $= -30$, $\Delta = 0.025$ *s*, $m_3 = 2$ *kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

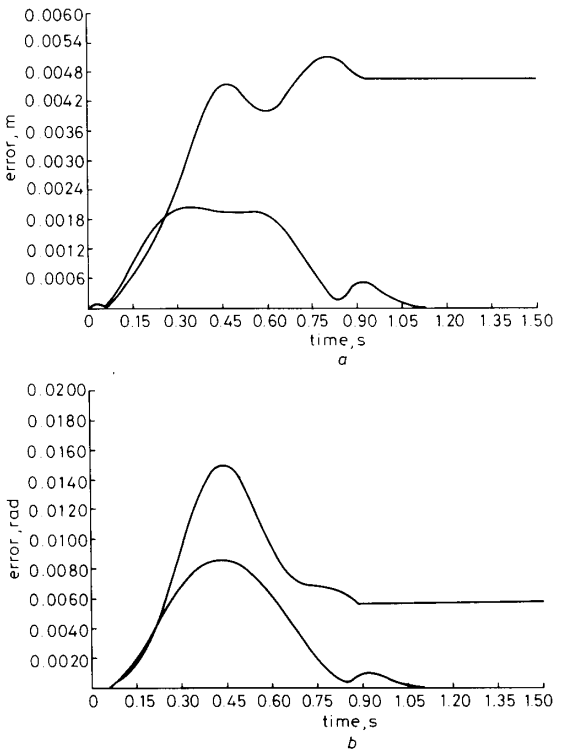

Fig. 16 *Stanford arm, pole* $= -30$, $\Delta = 0.0025$ *s*, $m_3 = 6$ *kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

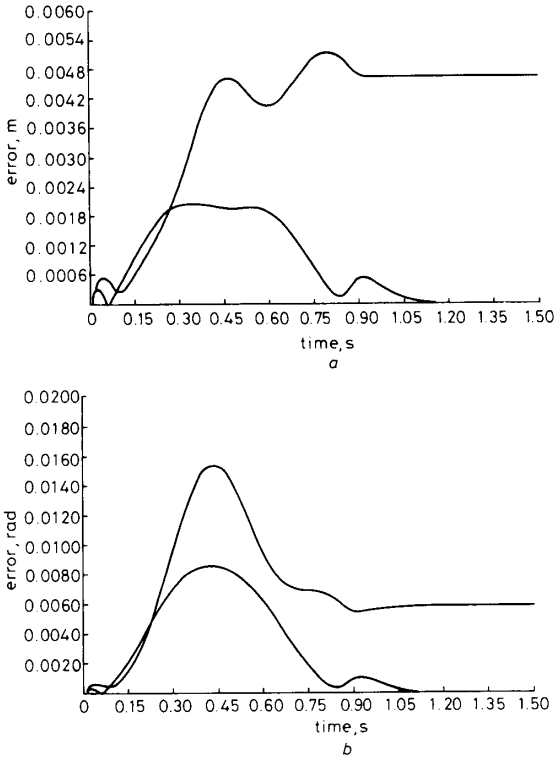

Fig. 17 *Stanford arm pole = −30, Δ = 0.01 s, m_3 = 6 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

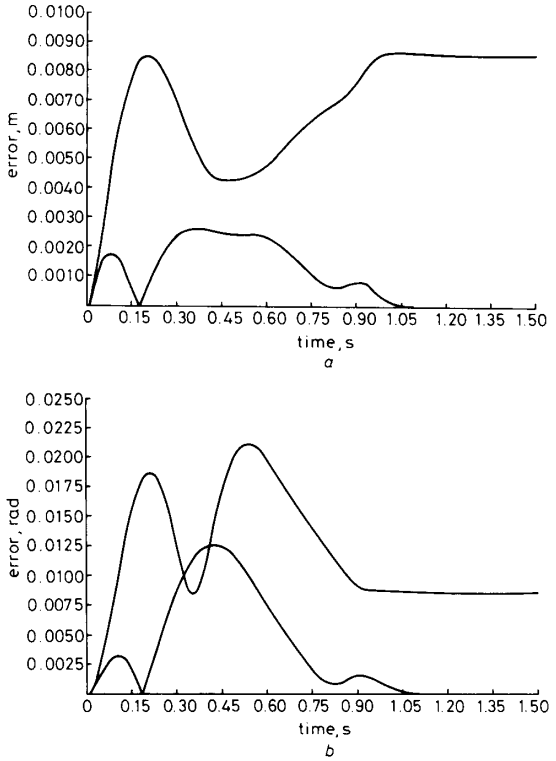

Fig. 19 *Stanford arm, pole = −30, Δ = 2 s, m_3 = 6 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

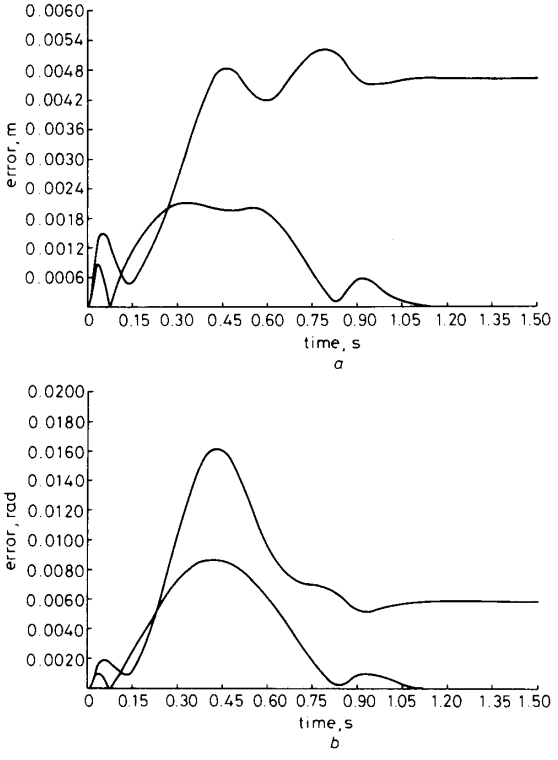

Fig. 18 *Stanford arm, pole = −30, Δ = 0.025 s, m_3 = 6 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

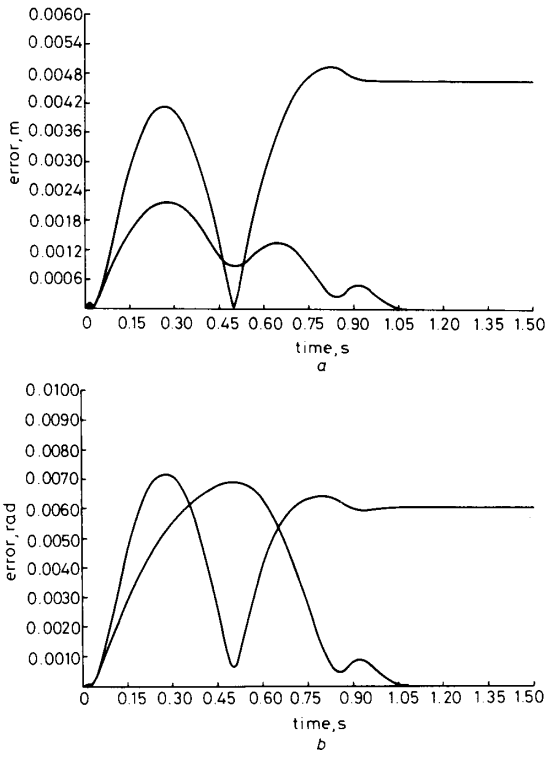

Fig. 20 *Stanford arm, pole = −30, Δ = 0.0025 s, m_3 = 2 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

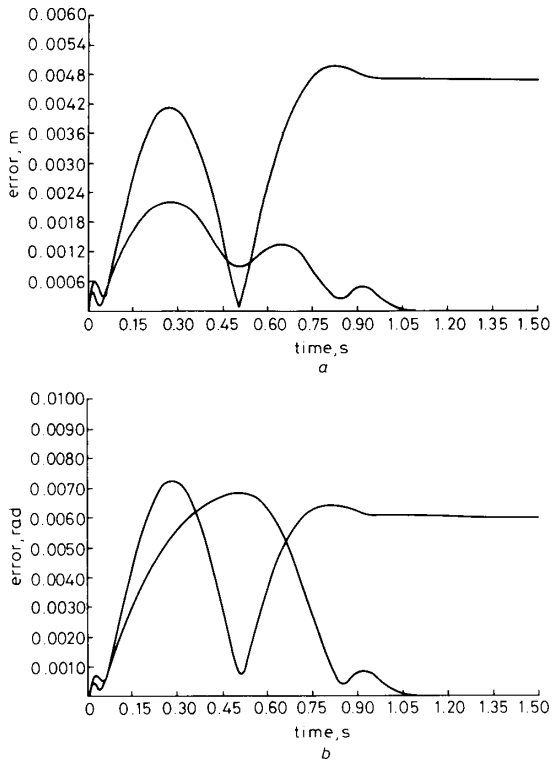

Fig. 21 *Stanford arm, pole = −30, Δ = 0.01 s, m_3 = 2 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

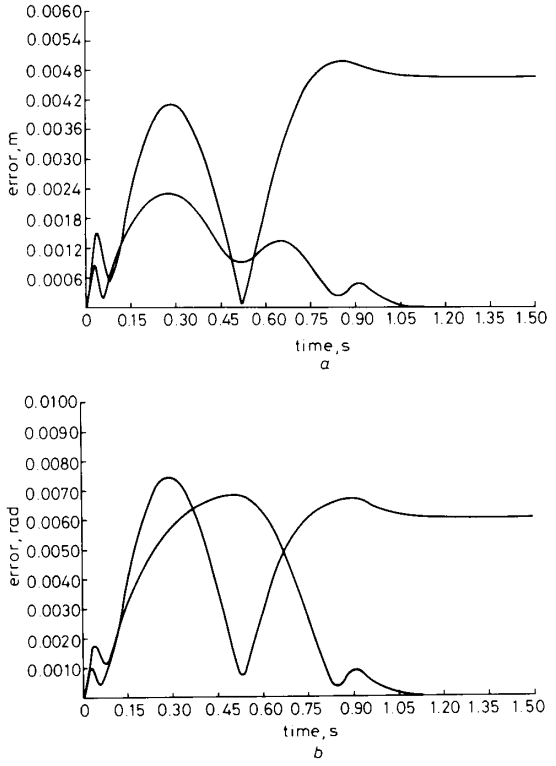

Fig. 22 *Stanford arm, pole = −30, Δ = 0.025 s, m_3 = 2 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

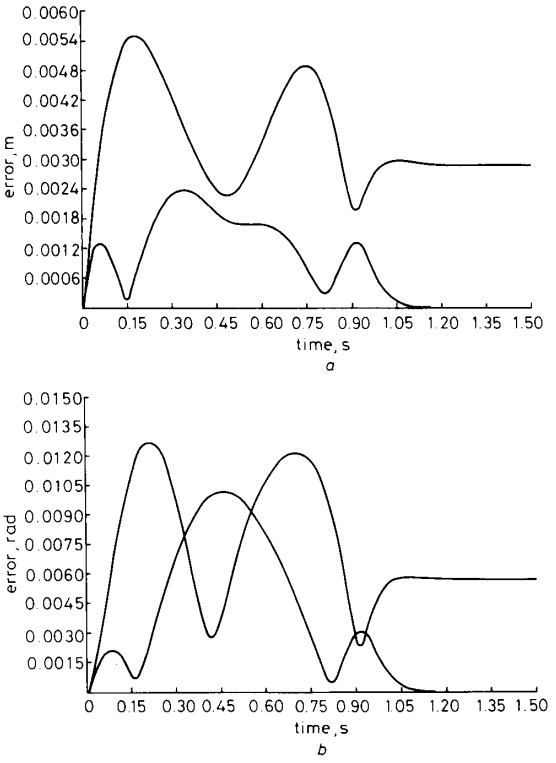

Fig. 23 *Stanford arm, pole = −30, Δ = 2 s, m_3 = 2 kg*
a Cartesian error norm resulting from the PD and PID regulators
b Joint error norm resulting from the PD and PID regulators

The algorithm robustness is showed by Figs. 8–23. The reference trajectory is well tracked and there is no overshooting.

The simulation results have also shown that a suitable selection of the pole, in computing the feedback gains, also plays an important role in reducing position and velocity errors in tracking a desired path. It can be seen from Figs. 24–27 that, when $\Delta > \Delta_{min}$, oscillations may occur: this is the case for the MA23 manipulator and not for the Standford arm. Diminution of μ leads to diminution of the oscillations, but the amplitude of the error diminishes when the gain increases. It is then necessary to perform a compromise between the desired precision and the form of the curve (no oscillation is required), for each studied manipulator. The interpolation of the diagonal

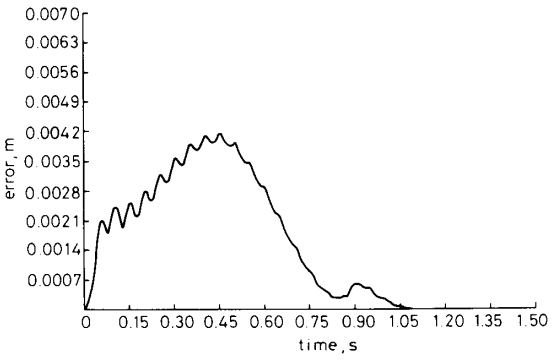

Fig. 24 *MA23 manipulator PID regulator, pole = −35, Δ = 0.025 s, m_3 = 2 kg, Cartesian error norm*

inertial terms is valid for increasing as well as for decreasing terms; for example, for the Stanford arm, the first term is increasing when the second is decreasing.

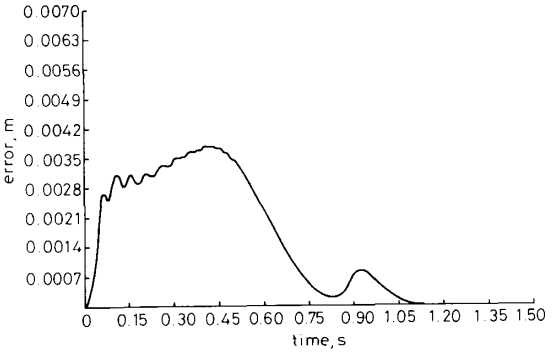

Fig. 25 *MA23 manipulator PID regulator, pole = −30, Δ = 0.025 s, m_3 = 2 kg, Cartesian error norm*

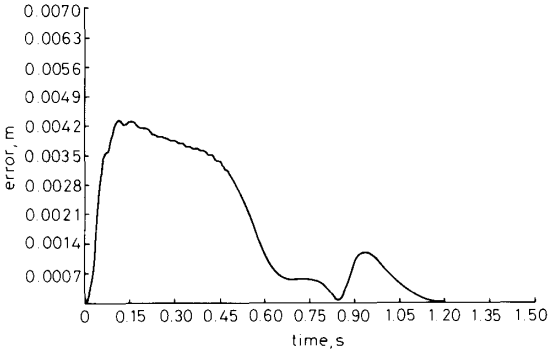

Fig. 26 *MA23 manipulator PID regulator, pole = −25, Δ = 0.025 s, m_3 = 2 kg, Cartesian error norm*

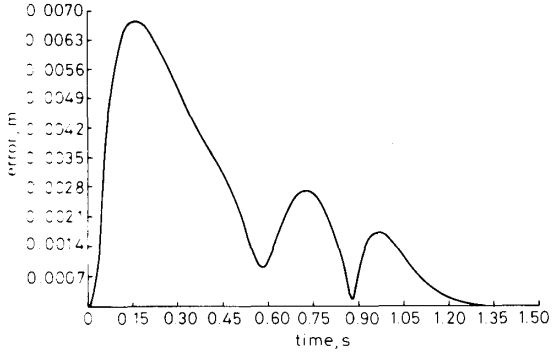

Fig. 27 *MA23 manipulator PID regulator, pole = −20, Δ = 0.025 s, m_3 = 2 kg, Cartesian error norm*

Other simulation results have shown that the form of the curves of the inertial matrix elements is independent of the value of the maximum velocity and acceleration admissible for the actuators.

7 Conclusions

Recent advances in microcomputer technology have intensified interest in distributed computation schemes. Aside from modular expandability, another potential advantage of such schemes is a reduction in computation time for solving the regulation problem due to parallelism of computation. This advantage is of crucial importance in real-time applications, where problem solution time can be an implementation bottleneck.

This paper is devoted to the investigation of a hierarchical robot control system with a gain-scheduling approach. To ensure that the real trajectory is as close as possible to the desired one, a two-stage control synthesis is introduced:

(*a*) a nominal dynamics stage

(*b*) a perturbed dynamics stage.

At the stage of nominal dynamics, the synthesis of control Γ^d is performed, which transfers the system state from the initial point to the desired goal point, in finite time t_f. Synthesis is performed at the base of the complete centralised model without any approximation. Because such programmed control is essentially centralised, at this stage, one part of the coupling is taken into account. The co-ordination parameters also take care of a great part of the remaining coupling. Thus, the coupling at the next stage is reduced. The local microprocessors keep performing computations without having to wait until they receive the messages that have been transmitted from the central microprocessor, computing Γ^d and w. That algorithm can alleviate communication overloads and it is not excessively slowed down, by neither communication delays nor by differences in the time it takes the processors to perform their computations, because the number of operations that must perform the local microprocessors is very small and the functions $\Gamma^d(t)$ and $w(t)$ are very regular.

Then, we examine the performance of the exact model when synthesised control is applied. We shall, therefore, consider the model of state deviation from nominal motion, in terms of the model of the mechanical part of the system. A comparatively simple control, with respect to the computational facilities and feedback information required, is considered. Its implementation in real time is discussed. The key objective is to implement a control law using present-day microprocessors. Its principal advantages are:

(i) it only requires feedback information about the joint positions and velocities, i.e. in compliance with common practice in industrial robots today

(ii) it decouples the system, thus providing a natural way for development of decentralised multiprocessor controllers

(iii) it considers the dynamics of the manipulator, requiring, at the same time, rather restrictive computational facilities calculation of the A_{ii} elements.

Simulation results showed that the algorithm is quite effective. The effects of the nonlinearity, couplings and gravitational terms can be reduced mostly by the model reference compensation. The approximated regulator gains give the robot high tracking capability. The algorithm can be easily implemented online. All the noncompensated dynamic terms and the parameter uncertainties play the role of disturbances to the system.

8 Acknowledgments

The author would like to thank Professor R. Mezencev, Professor W. Khalil and the reviewers for their comments.

9 References

1 LUH, J.Y.S., WALKER, M.W., and PAUL, R.P.: 'On-line computational scheme for mechanical manipulator', *J. Dyn. Syst. Meas. & Control*, 1980, **102,** (2), pp. 69–76

2 VUKOBRATOVIC, M., and STOKIC, D.: 'Scientific fundamentals of robotics 2 — Control of manipulation robots, theory and applications' (Springer-Verlag, 1982)

3 GRAVEL, D.T., and HSIA, T.C.: 'Decentralized adaptive control of robot manipulators', *Proc. 1987 IEEE Intl. Conf. Robotics & Autom.*, 1987, pp. 1230–1235

4 KAHN, M.E., and ROTH, B.E.: 'The near minimum time control of open loop articulated kinematic chains', *J. Dyn. Syst. Meas. & Control*, 1971, **93,** (3), pp. 164–172

5 CRAIG, J.J.: 'Introduction to robotics, mechanics and control' (Addison-Wesley, 1986)

6 MAHIL, S.S.: 'On the application of Lagrange's method to the description of dynamic systems', *IEEE Trans.*, 1982, **SMC-12,** pp. 877–889

7 KHALIL, W.: 'Minimization of the computational cost of the dynamic models of robots'. ATP-CNRS Symposium, Sept. 1986, Paris, France

8 PAUL, R.P.C.: 'Robot manipulator, mathematics, programming and control' (MIT Press, 1981)

9 KHALIL, W., LIEGEOIS, A., and FOURNIER, A.: 'Commande dynamique de robot', *RAIRO Syst. Analysis & Control*, 1979, **13,** pp. 189–201

10 KINNAERT, M., and HANUS, R.: 'A new adaptive control strategy derived from the computed torque methods for robotic manipulators'. Proc. Intl. Symp. Theory Robots, IFAC, Vienna, Austria, Dec. 1986, pp. 257–262

11 RAIBERT, M.H., and HORN, B.K.P.: 'Manipulator control using the configuration space method', *Ind. Robot*, 1978, **5,** pp. 69–73

12 KOIVO, A.J., and GUO, T.H.: 'Adaptive linear controller for robotic manipulator', *IEEE Trans.*, 1983, **AC-28,** pp. 162–171

13 FREUND, E.: 'The structure of decoupled non-linear systems', *Int. J. Control*, 1975, **21,** pp. 443–450

14 TARN, T.J., BEJCZY, A.K., ISIDORI, A., and CHEN, Y.: 'Nonlinear feedback in robot arm control'. Proc. 23rd IEEE Conf. Decision & Control, Las Vegas, NV, USA, Dec. 1984, pp. 736–751

15 BESTAOUI, Y.: 'Adaptive hierarchical control for robotic manipulators', *Robotics*, 1988, **4,** (2), pp. 145–155

16 MEZENCEV, R., SZYMANOWSKI, J., and BESTAOUI, Y.: 'On-line implementation of the reference trajectories for a multi-axis robot'. Proc. 12th IMACS World Congress Sci. Comput., Paris, France, July 1988

17 SAMSON, C.: 'Une approche pour la synthèse et l'analyse de la commande des robots manipulateurs rigides'. Memo 356, IRISA, Rennes France, April 1987, p. 226

18 BERTSEKAS, D.P.: 'Distributed dynamic programming', *IEEE Trans.*, 1982, **AC-27,** (3), pp. 610–616

19 LELONG-FERRAND, J., and ARNAUDIES, J.M.: 'Cours de mathématiques' (Dunod Ed., Paris, France, 1977)

20 GILBERT, E.G., and HA, I.J.: 'An approach to non-linear feedback control with applications to robotics', *IEEE Trans.*, 1984, **SMC-14,** (6), pp. 879–884