

# Engineering Design of a Multirate Nonlinear Controller for Robot Manipulators

J. A. Tenreiro Machado and J. L. Martins de Carvalho

A new multirate nonlinear controller for robot manipulators is described. The algorithm stems from the computed torque method, but has several modifications on its structure, resulting in a more robust controller.

The main philosophy behind those improvements resides on a sound engineering and management of the system resources. Consequently, the use of multirate sampling, memory tables, and predictive action arises naturally from that methodology.

The resulting controller is easy to tune using standard industrial engineering practice; moreover, by alleviating the computational burden its implementation is feasible with present day microprocessors.

ロボットマニピュレータのための、新しいマルチレート非線形コントローラが述べられる。アルゴリズムはcomputed torque methodに由来しているが、構造にいくつかの改良が加えられた結果、よりロバストなコントローラになっている。これらの改良の裏にある主な思想は音響工学とシステム手法のマネジメントにある。

る。マルチレートサンプリング、メモリテーブルと予測行動が利用される。その結果、コントローラは通常の工業技術訓練によって容易に調節できる。また計算の負担が軽減されるため現在のマイクロプロセッサに移植することが可能である。

## I. INTRODUCTION

The demand for faster and more accurate robot manipulators has been increasing in the last years. Usually, these mechanical systems with several links interconnected by rotational or linear joints have complex dynamic interactions, which make difficult the development of efficient controllers. A natural candidate for such a controller would be a classical linear system.<sup>1,2</sup> Nevertheless, the mentioned nonlinear dynamics make it inadequate if good

See List of symbols on page 14.

performances are demanded. Such limitations motivated the emergence of a new generation of nonlinear controllers, capable of meeting the robot dynamic system requirements.<sup>3-6</sup> In spite of these efforts, its industrial application has been delayed due to several problems not yet fully resolved, namely:

- Some controller algorithms are prohibitive with present day commercially available microprocessors.<sup>7</sup>
- There is a lack of knowhow (and the corresponding engineering “intuition”), relating the nonlinear controller system performances to imperfect modelling, computational delays, and discrete system stability requirements.

In this article these problems are addressed for the computed torque nonlinear controller. Some research in this area has already been done by Neuman,<sup>8,9</sup> showing that better robustness is possible if the basic controller structure is modified. The present work stems from those ideas, and attempts to give a more systematic treatment, which results in easy to use standard engineering development tools. In section II some limitations of the computed torque method are analysed. Based on these conclusions, the controller is improved through the introduction of some modifications in its structure and by a more efficient allocation of the computational resources. Finally, in section IV conclusions are drawn and engineering concepts are derived.

## II. ON THE LIMITATIONS OF THE COMPUTED TORQUE CONTROLLER

Robot manipulator dynamics is modelled by a nonlinear matrix differential equation<sup>10,11</sup>

$$\mathbf{T} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}). \quad (1)$$

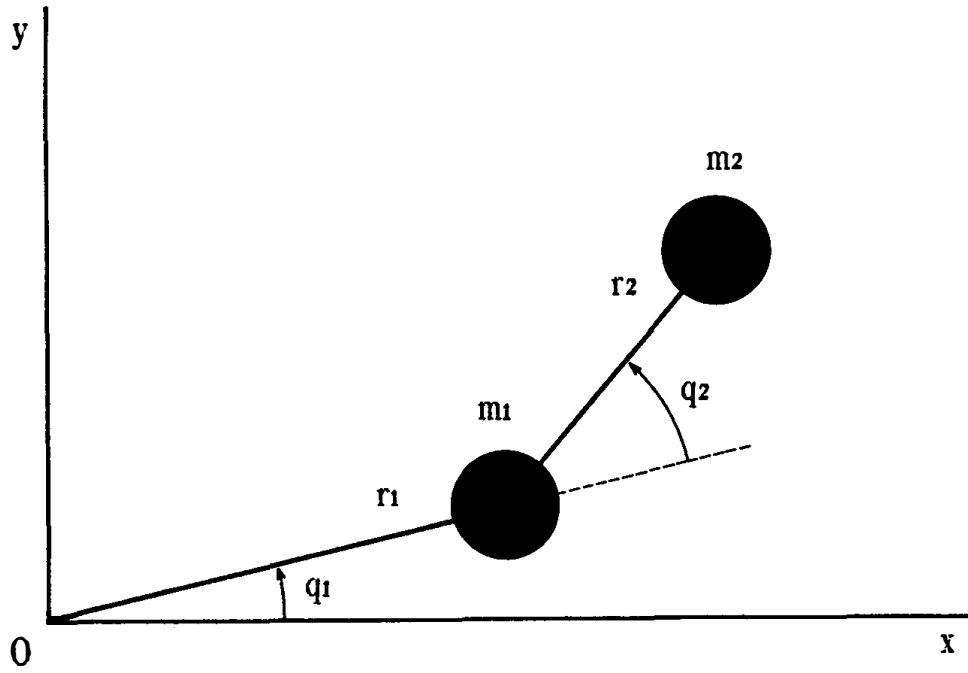
For a six degree of freedom (DOF) manipulator, Eq. (1) is complex and lengthy, and to derive it by hand is a formidable task. Several researchers<sup>12-15</sup> developed computer algorithms for automatic generation of such formulae; nevertheless, for many aspects of robot system research, the simple two or three DOF manipulator models are sufficient, if all the required effects under study are included.

We highlight this argument, using a 2R robot manipulator (Fig. 1) model, without any loss of generality. In fact, the 2R manipulator dynamic model

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} m_1 r_1^2 + m_0(r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1 & m_0(r_2^2 + r_1 r_2 C_2) \\ m_0(r_2^2 + r_1 r_2 C_2) & m_0 r_2^2 + J_2 \end{bmatrix} \quad (2a)$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -2m_0 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2 - m_0 r_1 r_2 S_2 \dot{q}_2^2 \\ m_0 r_1 r_2 S_2 \dot{q}_1^2 \end{bmatrix} \quad (2b)$$

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} g[m_1 r_1 C_1 + m_0(r_2 C_{12} + r_1 C_1)] \\ g m_0 r_2 C_{12} \end{bmatrix} \quad (2c)$$



**Figure 1.** 2R robot manipulator.

already includes all the configuration dependent inertial, Coriolis/centripetal and gravitational torques that may appear in manipulators with more DOF, thus being adequate for our prototype manipulator. This strategy has been used by other researchers<sup>3,16,17</sup> with parameter values similar to those used in the present study (Table I).

The well known computed torque nonlinear controller<sup>4</sup> described by the equations

$$\text{Commanded acceleration} \quad \mathbf{u}(t) = \ddot{\mathbf{q}}_d + \mathbf{K}_1(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_2(\mathbf{q}_d - \mathbf{q}) \quad (3a)$$

$$\text{Computed torque} \quad \mathbf{T}(t) = \mathbf{J}_c(\mathbf{q})\mathbf{u}(t) + \mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}_c(\mathbf{q}) \quad (3b)$$

$$\text{Error} \quad \mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t) \quad (3c)$$

$$\text{Error equation} \quad \ddot{\mathbf{e}} + \mathbf{K}_1\dot{\mathbf{e}} + \mathbf{K}_2\mathbf{e} = \mathbf{W}(t) \quad (3d)$$

$$\begin{aligned} \text{Disturbance vector} \quad \mathbf{W}(t) = & \mathbf{J}_c(\mathbf{q})^{-1}\{[\mathbf{J}(\mathbf{q}) - \mathbf{J}_c(\mathbf{q})]\ddot{\mathbf{q}} + \\ & + [\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}})] \\ & + [\mathbf{G}(\mathbf{q}) - \mathbf{G}_c(\mathbf{q})]\} \end{aligned} \quad (3e)$$

**Table I.** 2R robot manipulator numerical values.

$r_1 = 1 \text{ m}$	$r_2 = 0.8 \text{ m}$	$J_1 = 5 \text{ Kg m}^2$	$J_2 = 5 \text{ Kg m}^2$
$m_1 = 0.5 \text{ Kg}$	$m_2 = 0.25 \text{ Kg}$	$m_L = 6 \text{ Kg}$	

has been proposed as an efficient method to control robot manipulator systems. It is also common knowledge that in general, the expected performance is only guaranteed if we have an exact model, that is

$$\mathbf{J}_c(\mathbf{q}) = \mathbf{J}(\mathbf{q}), \quad \mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}), \quad \mathbf{G}_c(\mathbf{q}) = \mathbf{G}(\mathbf{q}) \quad (4)$$

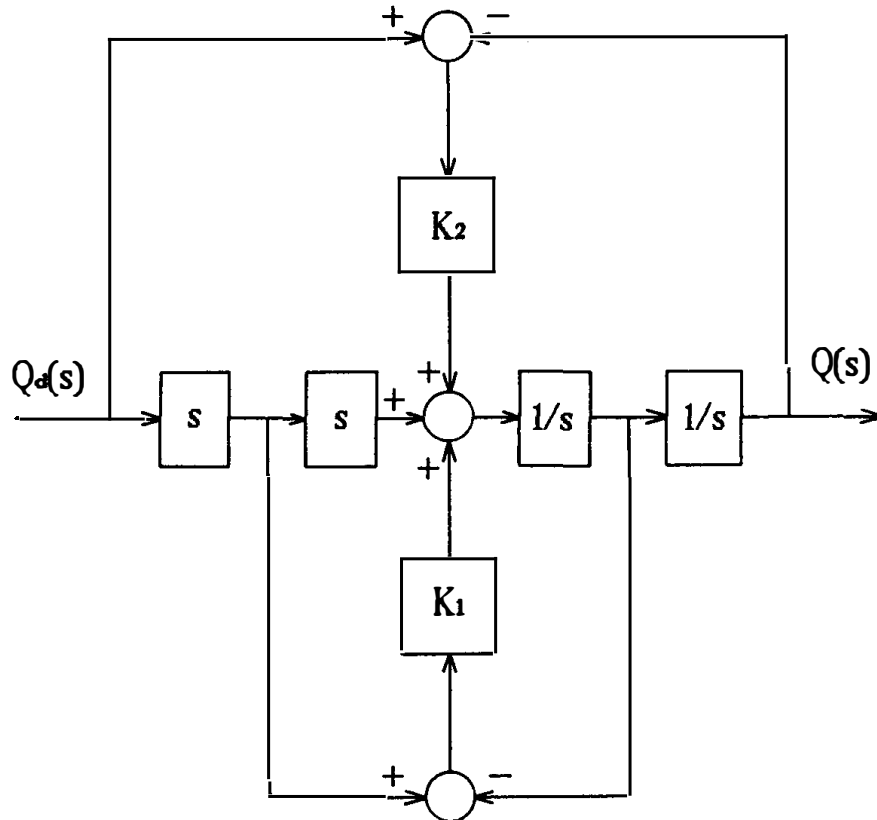
If these conditions are not fulfilled, a complex nonlinear system (3d, 3e) results. Therefore, it is somewhat difficult to analyse the resulting dynamic behaviour.

As shown in Figure 2, with an exact model the system behaves like a linear block with transfer function

$$Q(s)/Q_d(s) = (s^2 + K_1s + K_2)/(s^2 + K_1s + K_2) = 1 \quad (5)$$

Nevertheless, the computer controlled system will have a nonzero controller computing time and nonzero sampling period, as well as imperfect modelling. These factors justify a more elaborate study of their implications. Rearranging Eqs. (3d) and (3e) we get

$$\mathbf{J}_c(\mathbf{q})(\ddot{\mathbf{e}} + \mathbf{K}_1\dot{\mathbf{e}} + \mathbf{K}_2\mathbf{e}) = d\mathbf{T} \quad (6)$$



**Figure 2.** Linear model of the computed torque system.

where

$$d\mathbf{T} = [\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q})] - [\mathbf{J}_c(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}_c(\mathbf{q})] \quad (7)$$

is the model error. From (6) we conclude:

- The right hand side may be viewed as the model error torque and the left hand side as the required controller correcting torque.
- This controller correcting torque is driven by the feedback loop  $\mathbf{J}_c(\mathbf{q})$ ,  $\mathbf{K}_1$ ,  $\mathbf{K}_2$ .
- The more inaccurate the model the higher the requirements imposed to the feedback loop.
- For digital control systems, the finite sampling frequency, does not allow a continuous torque correction. Therefore, the discontinuous correcting torque may result in a system performance degradation.

These conclusions clearly show the dependence of the system dynamic properties upon the correcting feedback loop  $\mathbf{J}_c(\mathbf{q})$ ,  $\mathbf{K}_1$ ,  $\mathbf{K}_2$ . Nevertheless, the system is also required to have good performances in “time independent” situations, that is in steady state. Here, we have

$$\ddot{\mathbf{q}}_d = 0; \quad \ddot{\mathbf{q}} = 0; \quad \dot{\mathbf{q}}_d = 0; \quad \dot{\mathbf{q}} = 0 \quad (8)$$

and the system equations become:

$$\mathbf{e}_{ss} = \mathbf{K}_2^{-1} \mathbf{J}_c(\mathbf{q})^{-1} [\mathbf{G}(\mathbf{q}) - \mathbf{G}_c(\mathbf{q})] \quad (9)$$

As expected, the steady state error  $\mathbf{e}_{ss}$  is determined by the gravitational model error, and is inversely proportional to  $\mathbf{K}_2$ .

Summing up we have that the controller blocks fulfill different tasks:

- For dynamic aspects (transients, stability etc.) the feedback loop  $\mathbf{J}_c(\mathbf{q})$ ,  $\mathbf{K}_1$ ,  $\mathbf{K}_2$ , is of prime importance.
- In steady state, the gravitational model and the constant gain position matrix ( $\mathbf{K}_2$ ) are the main blocks.

Although expressions (6) and (9) are exact, they are hard to use in a controller design stage. In order to enable the use of standard design techniques, we need to make some modifications in the control structure of Figure 2. To accomplish a more accurate Laplace type model, we will introduce:

- A delay block ( $e^{-sT^1}$ ), responsible for the computational time delay  $T^1$ .
- A second delay block ( $e^{-sT^2}$ ), that approximately models the sample and hold block. Usually, this block is modelled as

$$H(s) = (1 - e^{-sh})/(sh); \quad h = \text{sampling period} \quad (10)$$

but an elementary Taylor series expansion reveals that for “low frequencies”  $H(s)$  may be approximated by

$$H(s) \approx e^{-sh/2}; \quad T2 = h/2 \quad (11)$$

- The diagonal inertial matrix coefficient inaccuracy is modelled by a nonunity gain  $K = 1$  present in the direct loop.
- The nondiagonal inertial, Coriolis/centripetal and gravitational model inaccuracy is simulated by an additive perturbing signal  $P(s)$  in the direct loop.
- In order to study more general controller structures, the constant gain matrices  $\mathbf{K}_1$  and  $\mathbf{K}_2$  are replaced by transfer functions  $\mathbf{G}_1(s)$  and  $\mathbf{G}_2(s)$ , respectively.

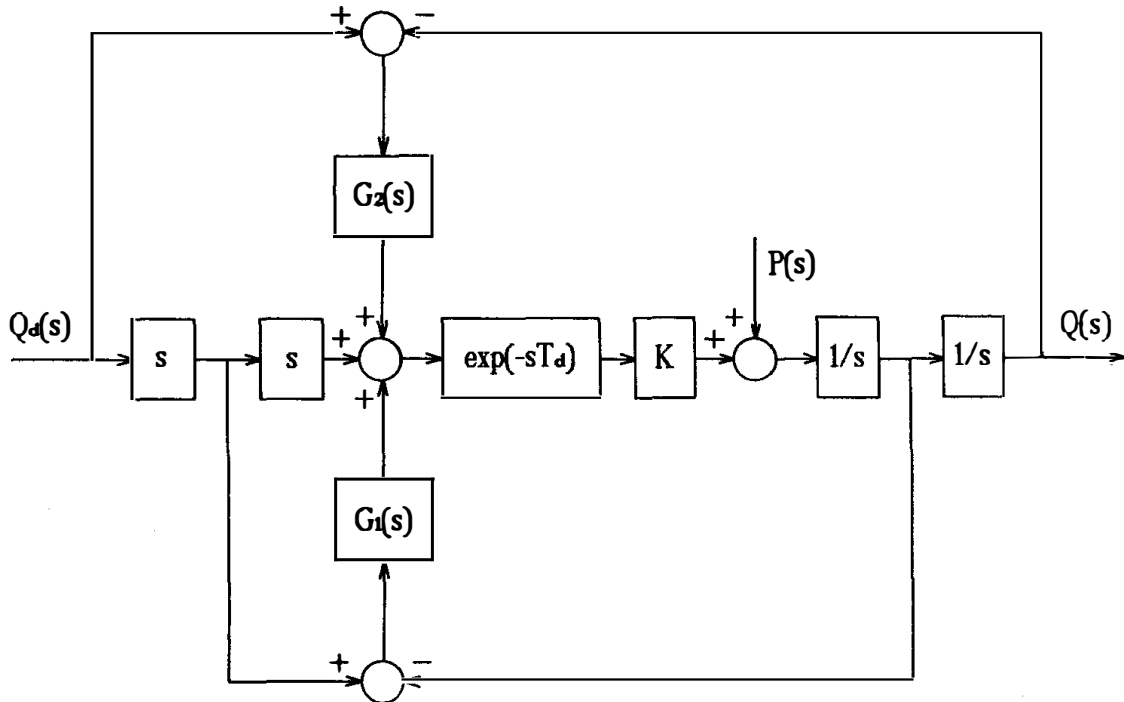
The resulting (decoupled) system block diagram is depicted in Figure 3, with:

$$Q(s)/Q_d(s) = Ke^{-sT_d}\{1 + (1 - e^{-sT_d})[sG_1(s) + G_2(s)]/D(s)\} \quad (12a)$$

$$Q(s)/P(s) = e^{-sT_d}/D(s) \quad (12b)$$

$$D(s) = s^2 + Ke^{-sT_d}[sG_1(s) + G_2(s)] \quad (12c)$$

$$T_d = T1 + T2 \quad (12d)$$



**Figure 3.** Improved linear model of the system.

Although using a simplified model we obtain conclusions similar to those obtainable from (6) and (9): the delay  $T_d = T_1 + T_2$  should be as small as possible, gain  $K$  should be unity, and if a more robust controller is required  $G_1(s)$  and  $G_2(s)$  should be modified.

Based on the results of this analysis, a new controller is now presented.

### III. THE MULTIRATE NONLINEAR CONTROLLER

Equation (12) enables a systematic study of an improved computed torque controller; therefore, the designer may use standard control techniques much as:

- Use of integral action (i.e.,  $G_2(s) = K_2 + K'_2/s$ ) in order to eliminate the position steady state error (12b).
- Use of derivative action (i.e.,  $G_1(s) = K_1 + K'_1s$ ) so that stability is improved.
- Stability analysis using (12c) and conventional criteria like the Nyquist diagram.

Besides the referred standard improvements, a reduction in the time delay constant ( $T_d = T_1 + T_2$ ), may be obtained through the use of different sampling ratios<sup>18</sup> on the controller loops:

- The heavily time dependent  $\mathbf{J}_c(\mathbf{q})$ ,  $\mathbf{G}_1(s)$ ,  $\mathbf{G}_2(s)$  loop is computed at high frequency ( $1/h_F$ ).
- The less critical loop  $\mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}_c(\mathbf{q})$ , is computed at low frequencies ( $1/h_s$ ).

Therefore, the control computer works in a time sharing way. This aspect is of utmost importance, as it corresponds to a more efficient use of the system resources. A uniform sampling frequency would be a “waste” of computing power on the  $\mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}_c(\mathbf{q})$  loop, which would be more profitable at the  $\mathbf{J}_c(\mathbf{q})$ ,  $\mathbf{G}_1(s)$ ,  $\mathbf{G}_2(s)$  loop. Another important observation resides on the computational burden of each block:  $\mathbf{G}_1(s)$  and  $\mathbf{G}_2(s)$  are easily computed,  $\mathbf{J}_c(\mathbf{q})$  and  $\mathbf{G}_c(\mathbf{q})$  represent a medium load,  $\mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}})$  is highly complex.

In the fast computational loop  $\mathbf{J}_c(\mathbf{q})$  is the heavier block. Although efficient computational techniques are available<sup>19–24</sup> possible *alternative* calculation methods, are the use of computer memory<sup>25–34</sup> or of dedicated compiler schemes.<sup>35,36</sup>

Computer memory has already been used for robot manipulator control. In the present case, this method may be viewed as another system management action. In fact, a few hundred kbytes are within reach of today’s micro-computers; therefore, it is imperative to take advantage of these capabilities to achieve new and better controller architectures. Moreover, the total table requirements may be optimized knowing that:

- the memory table can be reduced by half, as inertial terms are symmetrical for positive and negative position values, in rotational joints;
- the number of stored coefficients of  $\mathbf{J}_c(\mathbf{q})$  may be reduced from  $n^2$  to  $n(n+1)/2$  because it is a symmetrical matrix;
- the dimension of the required memory is of order  $n-1$  because the inertial terms are independent of the position of the first link;
- the memory table can be reduced using data compressing techniques such as the CMAC<sup>25,26</sup> associative memory model;
- obviously, one may use interpolations.

Other possible computational technique is the use of a dedicated compiler scheme. With this method the conversion of any mathematical formula to machine code, is made much more efficiently than using standard compilers. The resultant object code is sped-up through Binary Decision Diagrams,<sup>37</sup> and runs not only on conventional microcomputers but, furthermore, points out specialized parallel reduced instruction set computing machines.

Finally, a third improvement can be implemented by a predictive action.<sup>38-42</sup> As good path tracking is expected, Coriolis/centripetal and gravitational terms can be approximately computed in advance, estimating positions and velocities as the planned ones. Computation between two “slow samples” is made by interpolation.

Figure 4 depicts the resulting controller. Although based on an improvement of the computed torque controller it has some similarities with the feedforward dynamics compensation scheme.<sup>43</sup> This is not surprising because, as shown in Ref. 44, both control schemes are strongly related.

The conventional computed torque algorithm is now analysed and compared with the modified controller. Our design tool will be the ideal error equation

$$\ddot{\mathbf{e}} + \mathbf{K}_1 \dot{\mathbf{e}} + \mathbf{K}_2 \mathbf{e} = \mathbf{0} \quad (13a)$$

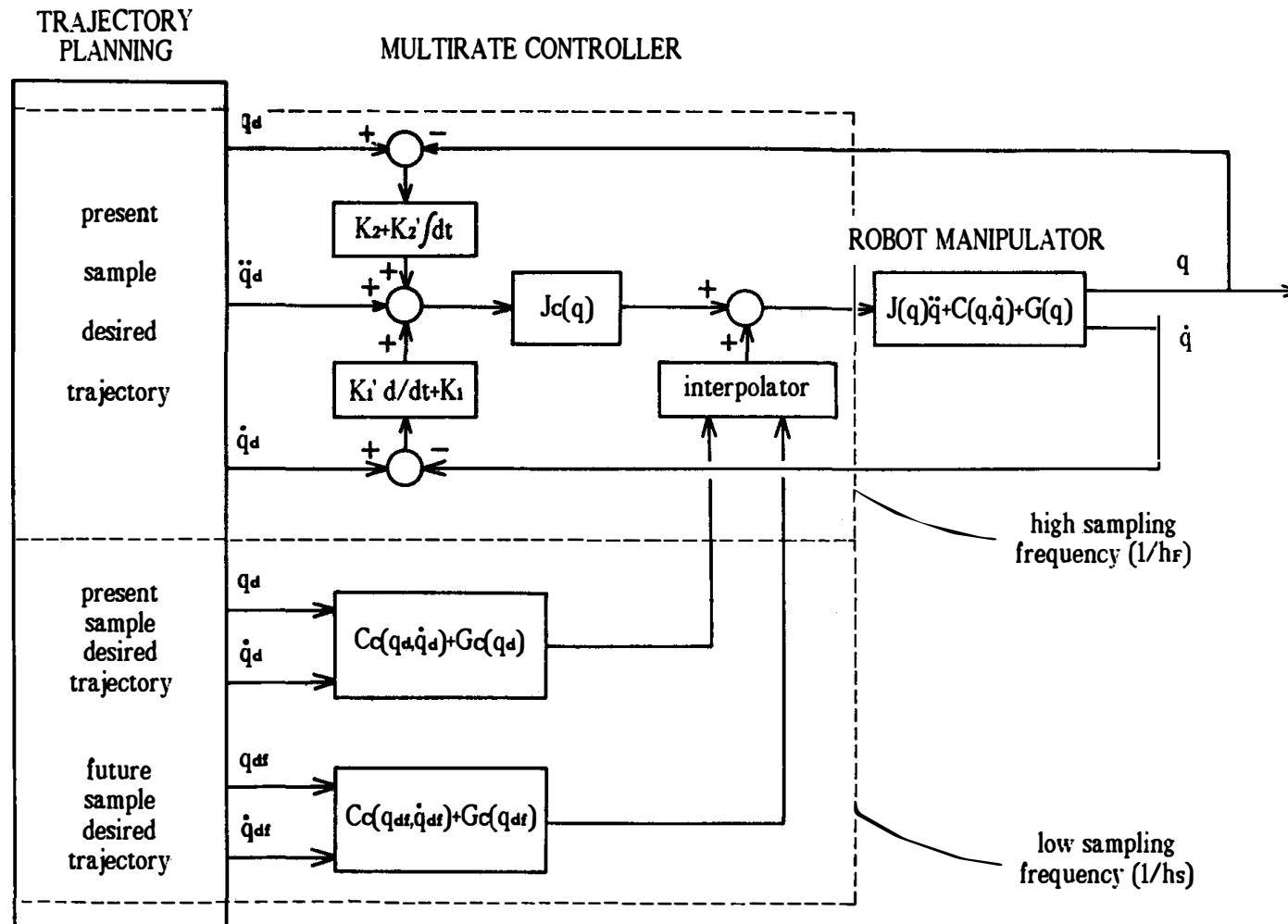
$$(\mathbf{I} + \mathbf{K}_1') \ddot{\mathbf{e}} + \mathbf{K}_1 \dot{\mathbf{e}} + \mathbf{K}_2 \mathbf{e} + \mathbf{K}_2' \int \mathbf{e} dt = \mathbf{0} \quad (13b)$$

for the first and second case, respectively. For a second order with critical damping type response the controller parameters are depicted in Table II. Nevertheless, as shown by the system (nonideal) characteristic Eq. (12c), the overall computational delay  $T_d$ , may produce unstable systems; therefore, the Nyquist diagram was used in order to investigate system stability. For the referred set of parameters and  $K = 1$ , the phase margin was found to be of 72 and 60 degrees respectively. Moreover, the Nyquist diagram also enables to find the relation between  $T_d$  and the double pole magnitude ( $W_n$ )

$$T_d \leq \beta / W_n \quad (14)$$

with  $\beta = 0.651$  for the conventional computed torque algorithm and  $\beta = 0.119$  for the new controller, which were proved to be correct based on simulations using the exact Eqs. (2) and (3).





**Figure 4.** The new multirate nonlinear controller.

**Table II.** Controller design parameters.

Controller type	Computed torque	Multirate
Design equation	$\ddot{\mathbf{e}} + \mathbf{K}_1 \dot{\mathbf{e}} + \mathbf{K}_2 \mathbf{e} = \mathbf{0}$	$(\mathbf{I} + \mathbf{K}'_1) \ddot{\mathbf{e}} + \mathbf{K}_1 \dot{\mathbf{e}} + \mathbf{K}_2 \mathbf{e} + \mathbf{K}'_2 \int \mathbf{e} dt = \mathbf{0}$
$s$ plane pole location	$s_{1,2} = -W_n$	$s_{1,2} = -W_n; s_3 = -a W_n$
Diagonal coefficients	$K_1 = 2 W_n; K_2 = W_n^2$	$K_1 = (1 + K'_1)(2 W_n + a)$ $K_2 = (1 + K'_1)(W_n + 2a) W_n$ $K'_2 = (1 + K'_1) a W_n^2$
Numerical values	$W_n = 4$	$W_n = 4; a = 10; K'_1 = 0.1$
Sampling period (sec)	0.005	$h_F = 0.005$ for the fast loop $h_s = 0.16$ for the slow loop
Phase margin (degrees)	72	60

To verify the system performances, a linear test trajectory  $s(t)$ , from point  $(-1, 1)$  up to point  $(1, 0)$ , was used, having a bang-bang time evolution of the type

$$\ddot{s}(t) = \begin{cases} \sqrt{5} & 0 \leq t < 1 \\ -\sqrt{5} & 1 \leq t < 2 \\ 0 & 2 \leq t \end{cases} \quad (15a)$$

$$\dot{s}(0) = 0; \quad s(0) = 0 \quad (15b)$$

The resulting position tracking error

$$e_p = [(x_d - x)^2 + (y_d - y)^2]^{1/2} \quad (16)$$

is shown in Figure 5 for an overestimation ( $\hat{m}_L = 12$  Kg), a correct estimation (i.e.,  $\hat{m}_L = 6$  Kg) and an underestimation ( $\hat{m}_L = 0$  Kg) of the load mass. As noted by Refs. 8, 9, 45, the payload overestimation implies a lower error than the opposite case. This conclusion is easily inferred from Eqs. (12b) and (12c), because an overestimation of the payload corresponds to  $K > 1$ : a gain increase makes the output  $Q(s)$  (Fig. 3) less sensitive to the disturbance  $P(s)$ . The modified controller eliminates the steady state error and reduces the position tracking error  $e_p$ ; in the simulated experiments, the maximum tracking errors are scaled down by a factor of 35.5, 1.4, and 24.7, respectively.

The simulation results depicted in Figure 5(b) confirm the assumptions drawn from Eqs. (6) and (7) in Section II. With a precise load estimation, the ultimate model error comes from the finite sampling frequency. In this case, the improvements on the  $\mathbf{J}_c(\mathbf{q})$ ,  $\mathbf{G}_1(s)$ ,  $\mathbf{G}_2(s)$  loop structure are of minor influence, resulting the major performance degradation from the noncontinuous action of the aforementioned digital control correcting loop. If we

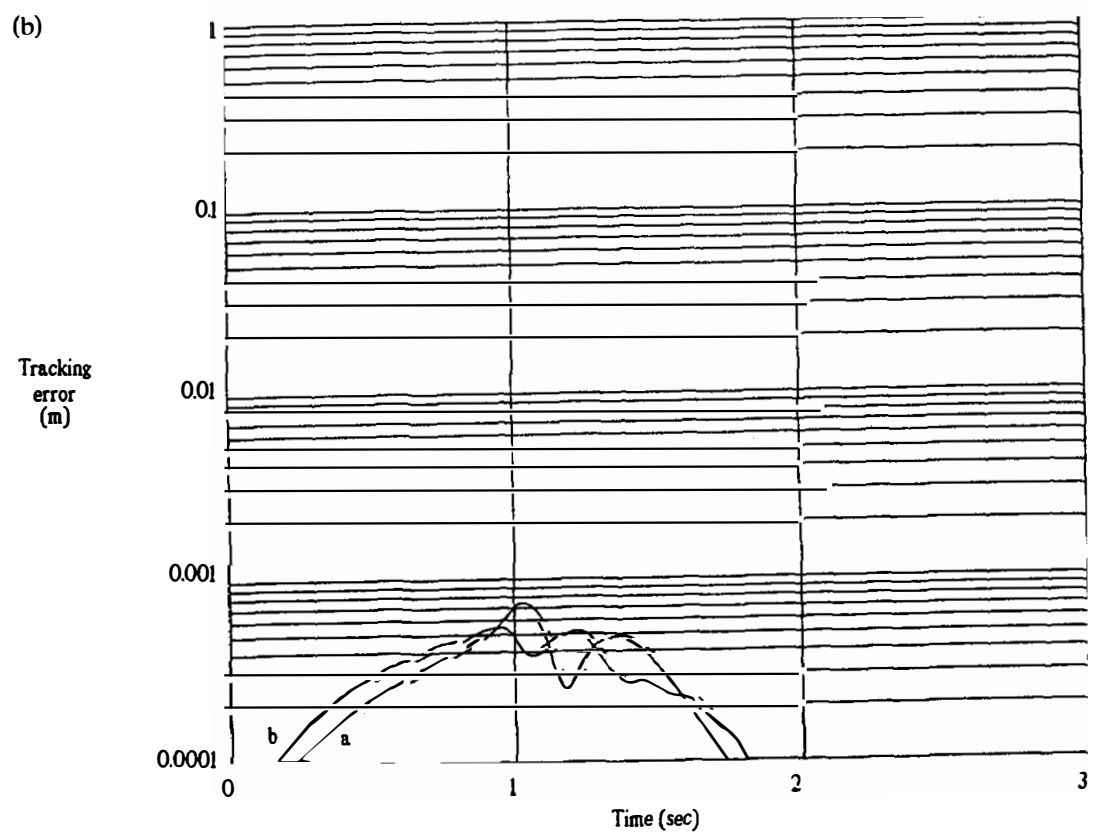
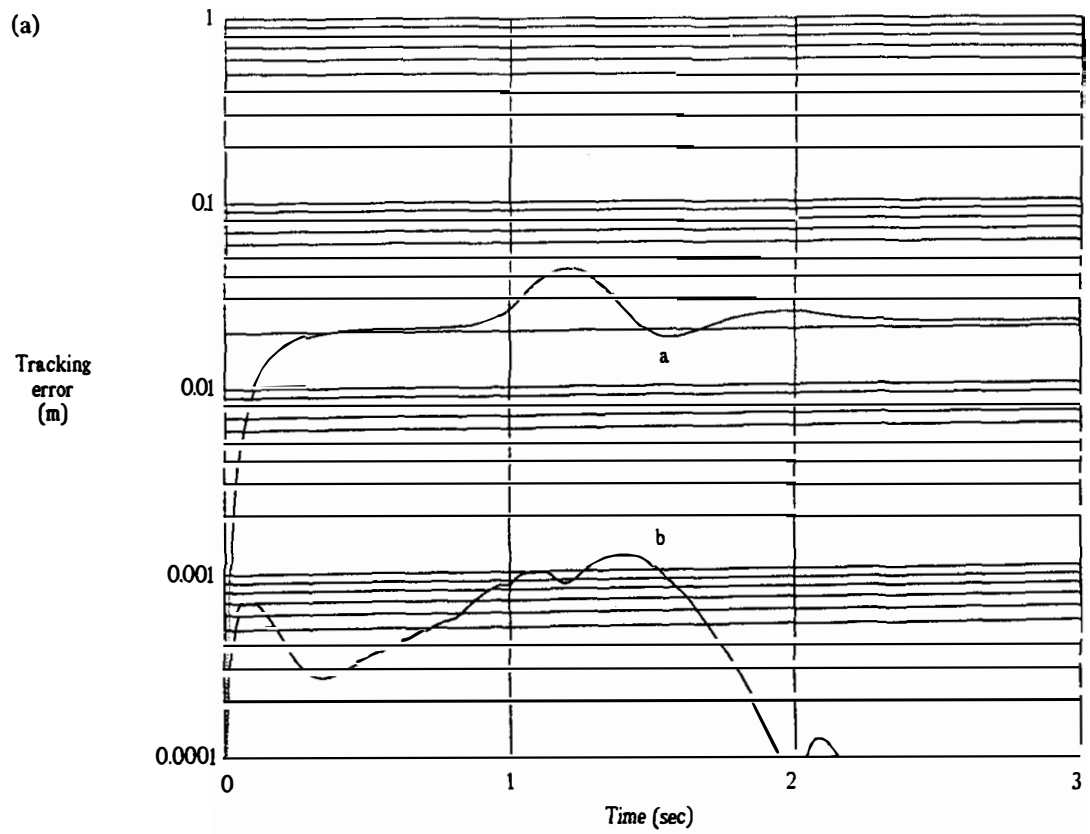
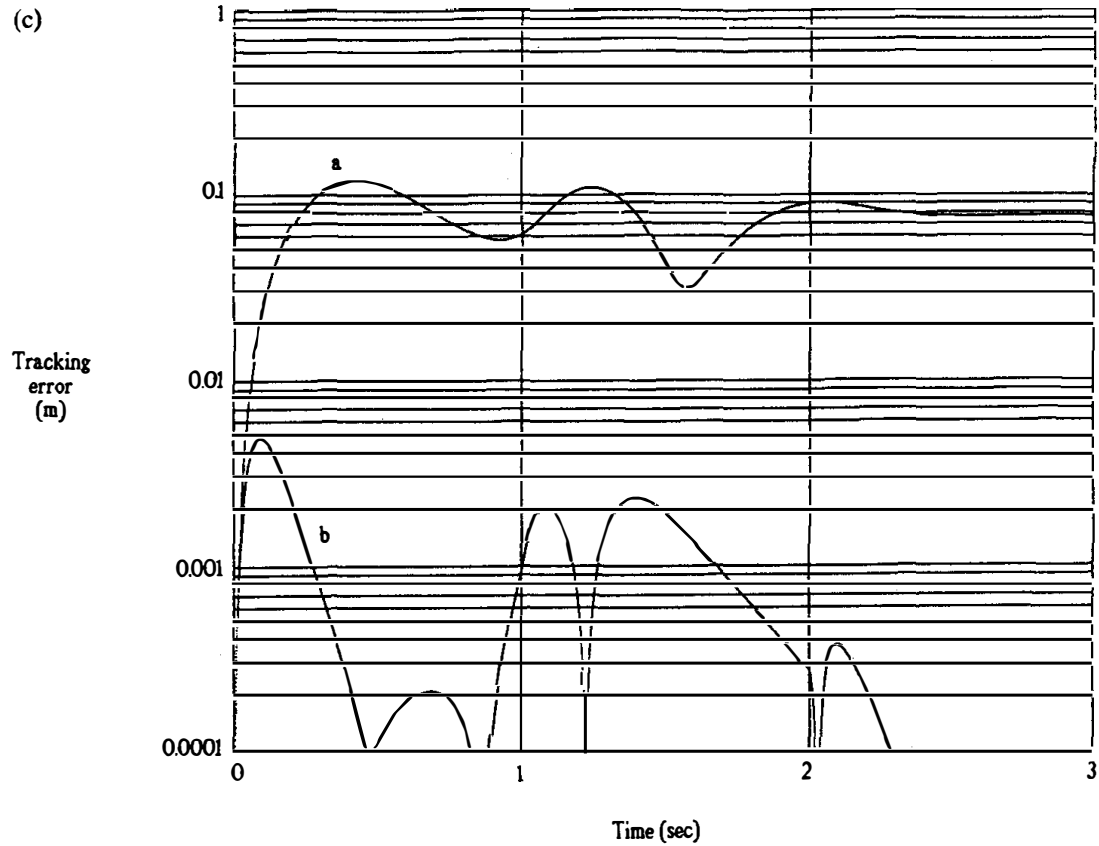


Figure 5(a,b).



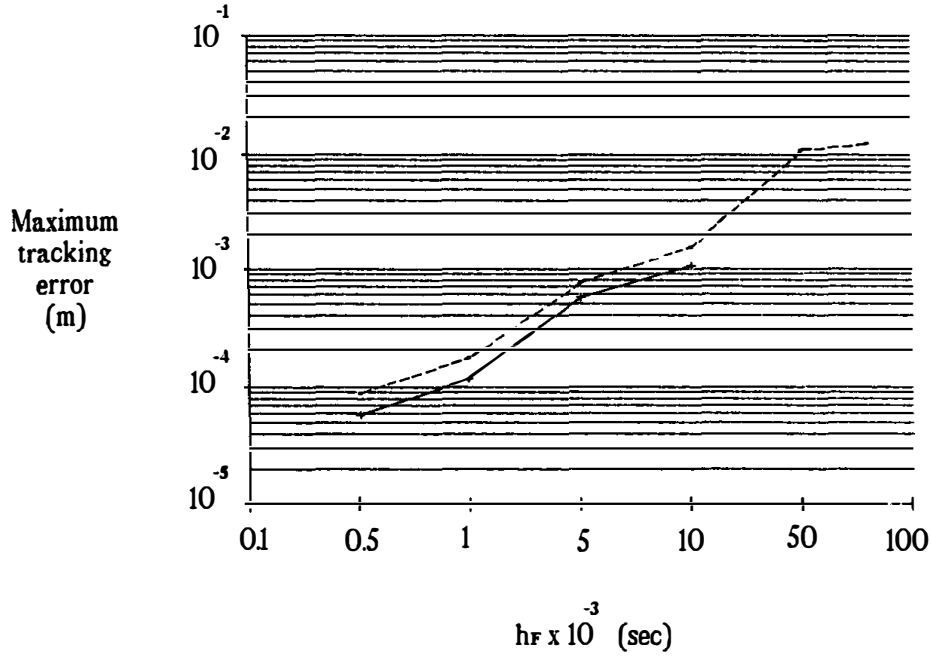
**Figure 5(c).** Resulting tracking error for the test trajectory  $s(t)$ .  $T1 = 0.0005$  sec,  $h_F = 0.005$  sec,  $h_s = 0.16$  sec, a—computed torque controller, b—multirate controller, (a) for an overestimation of the payload ( $\hat{m}_L = 12$  Kg), (b) for a precise estimation of the payload ( $\hat{m}_L = 6$  Kg), (c) for an underestimation of the payload ( $\hat{m}_L = 0$  Kg).

choose as our figure of merit, the maximum tracking error, then it is straightforward to understand the results of Figure 6.

Another point of discussion is how to design the multirate scheme, as well as its influence on stability and robustness. It was proved previously that for stability performances only the fast loop is of importance and consequently, from Eqs. (12d) and (14) we can find a restriction of the type

$$T1 + h_F/2 \leq \beta/W_n \quad (17)$$

where  $T1$  and  $\beta$  depend both on the software code efficiency and the microcomputer hardware capabilities. As far as robustness is concerned both sampling periods (i.e., the fast and slow loop periods  $h_F$  and  $h_s$ ) should be as small as possible; in the fast loop due to the fact that it corresponds to improve the feedback action, and in the slow loop as it implies a reduction on the amplitude of the perturbation signal  $P(s)$ . Finally, both sampling periods are



**Figure 6.** Maximum tracking error versus sampling period for a precise load estimation, using the test trajectory.  $T1 = 0.1 h_F$ , ----- computed torque controller, ——— multirate controller.

related by the expressions

$$T1 + pT0 \leq h_F, \quad 0 < p < 1 \quad (18a)$$

$$p = h_F / h_s \quad (18b)$$

where  $p$  represents the percentage of computation of the slow loop, included in each period of the fast one. These constraints (i.e., (17) and (18)) define a region of possible implementation that if not achieved implies the need of a more powerful software/hardware controller environment.

#### IV. CONCLUSION

The computed torque algorithm has been analyzed with the purpose of giving insight on the requirements towards better nonlinear model-based controllers. Then a more robust controller, that is a controller capable of better performances in the presence of model inaccuracies, was derived. Our approach has also shown that it is possible to conjugate accurate results with continuous time models in computer controlled systems. By avoiding the Z-transform approach we have been able to use the well established s-plane design methods and, consequently, capitalize our engineering intuition. This design philosophy provided a framework where several scheduling actions were developed. Our concern for an efficient allocation of the computational resources lead to several modifications in the conventional computed torque

controller, such as: multirate sampling, possible use of memory or of dedicated compiler schemes, and predictive action.

As a result, a new multirate and robust controller enabling real time performances with today's microcomputers was engineered. Moreover, it was developed a set of tuning procedures, which are compatible with present day industrial control practice.

The authors would like to thank the anonymous referees for their valuable comments.

## LIST OF SYMBOLS

$n$	number of degrees of freedom
$\mathbf{q}$	$n$ vector of joint positions
$\dot{\mathbf{q}}$	$n$ vector of joint velocities
$\ddot{\mathbf{q}}$	$n$ vector of joint accelerations
$\mathbf{J}(\mathbf{q})$	positive-definite, symmetric inertial $(n \times n)$ matrix
$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$	Coriolis/centripetal $n$ vector torque
$\mathbf{G}(\mathbf{q})$	gravitational $n$ vector torque
$\mathbf{T}$	joint $n$ vector torque
$\mathbf{J}_c(\mathbf{q})$	computational model of the $(n \times n)$ inertial matrix
$\mathbf{C}_c(\mathbf{q}, \dot{\mathbf{q}})$	computational model of the Coriolis/centripetal $n$ vector torque
$\mathbf{G}_c(\mathbf{q})$	computational model of the gravitational $n$ vector torque
$\mathbf{T}_c$	computed joint $n$ vector torque
$r_1$	length of link 1
$r_2$	length of link 2
$m_1$	mass of link 1
$m_2$	mass of link 2
$m_L$	payload mass at the top of link 2
$m_0 = m_2 + m_L$	total mass at the tip of link 2
$J_1$	inertia of link 1
$J_2$	inertia of link 2
$\mathbf{u}(t)$	commanded acceleration $n$ vector
$\mathbf{K}_1, \mathbf{K}_2$	diagonal feedback $(n \times n)$ matrices of constant, for the velocity and position feedback gains, respectively
$\mathbf{K}'_1, \mathbf{K}'_2$	diagonal feedback $(n \times n)$ matrices of constant, for the velocity derivative and position integral feedback gains, respectively
$\mathbf{W}(t)$	dynamics error driving, $n$ vector
$\mathbf{q}_d$	present sample, desired position
$\dot{\mathbf{q}}_d$	present sample, desired velocity
$\ddot{\mathbf{q}}_d$	present sample, desired acceleration
$\mathbf{q}_{df}$	next future sample, desired position
$\dot{\mathbf{q}}_{df}$	next future sample, desired velocity

$\ddot{\mathbf{q}}_{df}$	next future sample, desired acceleration
$T_d$	overall computational delay
$T_1$	computational delay of the fast loop
$T_0$	computational time of the slow loop
$p$	percentage of computation of the slow loop, included in each period of the fast loop
$h$	sampling period
$h_F$	sampling period of the fast loop
$h_s$	sampling period of the slow loop
$C_1 = \cos(q_1)$ ; $C_2 = \cos(q_2)$ ; $C_{12} = \cos(q_1 + q_2)$ ; $S_2 = \sin(q_2)$	

## References

1. D. F. Golla, S. C. Garg, and P. C. Hughes, "Linear State Feedback Control of Manipulators," *Mech. Machine Theory*, **16**, 93–103 (1981).
2. H. Seraji, M. Jamshidi, Y. K. Tim and M. Shahinpoor, "Linear Multivariable Control of Two-Link Robots," *J. of Robotic Systems*, **3**, 349–365 (1986).
3. K.-K. D. Young, "Controller Design for a Manipulator Using Theory of Variable Structure Systems," *IEEE Trans. Syst., Man, Cybern.*, **SMC-8**, 101–109 (1978).
4. J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Trans. Automat. Contr.*, **25**, 468–474 (1980).
5. E. Freund, "Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators," *The Int. J. Robotics Research*, **1**, 65–78 (1982).
6. M. Sahba and D. Q. Mayne, "Computer-Aided Design of Nonlinear Controllers for Torque Controlled Robot Arms," *IEE Proc.*, **131**, 8–14 (1984).
7. Chang-Huan Liu and Yen-Ming Chen, "Multi-Microprocessor-Based Cartesian-Space Control Techniques for a Mechanical Manipulator," *IEEE J. Robot., Automat.*, **RA-2**, 110–115 (1986).
8. V. D. Tourassis and C. P. Neuman, "Robust Nonlinear Feedback Control for Robotic Manipulators," *IEE Proc.*, **132**, 134–143 (1985).
9. C. P. Neuman and V. D. Tourassis, "Robust Discrete Nonlinear Feedback Control for Robotic Manipulators," *J. Robotic Systems*, **4**, 115–143 (1987).
10. R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*, The MIT Press, Cambridge, MA, 1981.
11. M. Brady et al., *Robot Motion: Planning and Control*, The MIT Press, Cambridge, MA, 1982.
12. H. Faessler, "Computer-Assisted Generation of Dynamical Equations for Multi-body Systems," *The Int. J. Robotics Research*, **5**, 129–141, (1986).
13. M. C. Leu and N. Hemati, "Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators," *ASME J. Dynamic Syst., Meas., Contr.*, **108**, 172–179 (1986).
14. J. Koplik and M. C. Leu, "Computer Generation of Robot Dynamics Equations and the Related Issues," *J. Robotic Systems*, **3**, 301–319 (1986).
15. C. P. Neuman and J. J. Murray, "Customized Computational Robot Dynamics," *J. Robotic Systems*, **4**, 503–526 (1987).
16. R. G. Morgan and U. Ozguner, "A Decentralized Variable Structure Control Algorithm for Robotic Manipulators," *IEEE J. Robot., Automat.*, **RA-1**, 57–65 (1985).
17. J. A. Tenreiro Machado and J. L. Martins de Carvalho, "A Smooth Variable Structure Control Algorithm for Robot Manipulators," *IEE Control 88 Conference*, Oxford, U.K., April, 1988.

18. Douglas P. Glasson, "Development and Application of Multirate Digital Control," *IEEE Control Syst. Magazine*, 2-8, Nov., (1983).
19. J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-Line Computational Scheme for Mechanical Manipulators," *ASME J. Dynamic Syst., Meas., Contr.*, **102**, 69-76, (1980).
20. J. Y. S. Luh and C. S. Lin, "Scheduling of Parallel Computation for a Computer-Controlled Mechanical Manipulator," *IEEE Trans. Syst., Man, Cyber.*, **SMC-12**, 214-234 (1982).
21. H. Kasahara and S. Narita, "Parallel Processing of Robot-Arm Control Computation on a Multimicroprocessor System," *IEEE J. Robot., Automat.*, **1**, 104-113 (1985).
22. Eli E. Binder and James H. Herzog, "Distributed Computer Architecture and Fast Parallel Algorithms in Real-Time Robot Control," *IEEE Trans. Syst., Man, Cyber.*, **16**, 543-549, (1986).
23. T. Watanabe, M. Kametani, K. Kawata, and K. Tetsuya, "Improvement in the Computing Time of Robot Manipulators Using a Multimicroprocessor," *ASME J. Dynamic Syst., Meas., Contr.*, **108**, 190-197 (1986).
24. C. P. Neuman and J. J. Murray, "Symbolically Efficient Formulations for Computational Robot Dynamics," *J. Robotic Systems*, **4**, 743-769 (1987).
25. J. S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *ASME J. Dynamic Syst., Meas., Contr.*, **97**, 220-227 (1975).
26. J. S. Albus, "Data Storage in the Cerebellar Model Articulation Controller (CMAC)," *ASME J. Dynamic Syst., Meas., Contr.*, **97**, 228-233 (1975).
27. Marc H. Raibert, "Analytical Equations vs. Table Look-Up for Manipulation: A Unifying Concept," *Proc. IEEE Conf. Decision Contr.*, New Orleans, Dec. 1977, pp. 576-579.
28. M. H. Raibert, "A Model for Sensorimotor Control and Learning," *Biological Cybernetics*, **29**, 29-36 (1978).
29. M. H. Raibert and B. K. P. Horn, "Manipulator Control Using the Configuration Space Method," *The Industrial Robot* **5**, 69-73 (1978).
30. S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering Operation of Robots by Learning," *J. Robotic Systems*, **1**, 123-140 (1984).
31. G. Hirzinger, "Robot Systems Completely Based on Sensory Feedback," *IEEE Trans. Ind. Electron.*, **IE-33**, 105-109 (1986).
32. F. Miyazaki, S. Kawamura, M. Matsumori, and S. Arimoto, "Learning Control Scheme for a Class of Robot Systems with Elasticity," *Proc. IEEE 25th Conf. Decision Contr.*, Athens, Greece, Dec., 1986.
33. W. Thomas Miller III, "Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm," *IEEE J. Robot., Automat.*, **RA-3**, 157-165 (1987).
34. W. T. Miller III, F. H. Glanz, and L. G. Kraft III, "Application of a General Learning Algorithm to the Control of Robotic Manipulators," *The Int. J. Robotics Research*, **6**, 84-98, (1987).
35. J. A. Tenreiro Machado, J. L. Martins de Carvalho, J. A. Silva Matos, and Antonio M. C. Costa, "A Real-Time System for Robot Manipulator Inverse Dynamics Computation," *15th IFAC/IFIP Workshop on Real-Time Programming*, Valencia, Spain, 1988.
36. J. A. Tenreiro Machado, J. L. Martins de Carvalho, Antonio M. C. Costa, and J. A. Silva Matos, "Robot Manipulator Dynamics—Towards Better Computational Algorithms," *IFAC Symp. on Robot Control SYROCO'88*, Karlsruhe, FRG, 1988.
37. Jose S. Matos and John V. Oldfield, "Binary Decision Diagrams: From Abstract Representations to Physical Implementations," *20th IEEE/ACM Design Automation Conference*, Miami Beach, U.S.A., 1983.



38. K. Yoshimoto and K. Wakatsuki, "Application of the Preview Tracking Control Algorithm to Servoing a Robot Manipulator," *Robotics Research, The First Int. Symp.*, 1984, pp. 883–897.
39. K. Yoshimoto and H. Sugiuchi, "Trajectory Control of Robot Manipulator Based on the Preview Tracking Control Algorithm," *Robotics Research, The Second Int. Symp.*, 1985, pp. 171–178.
40. C. D. Vinante, C. Bermudez, and F. Tarre, "Predictive Compensation Implemented with a Microprocessor," *IEEE Control Syst. Magazine*, **6**, 40–43 (1986).
41. H. B. Kuntze, A. Jacubasch, J. Richalet, and C. Arber, "On the Predictive Functional Control of an Elastic Industrial Robot," *Proc. IEEE 25th Conf. Decision Contr.*, Athens, Greece, Dec., 1986.
42. J. Richalet, S. Abu El Ata-Doss, and C. Arber, "Predictive Functional Control: Application to Fast and Accurate Robots," *Proc. 10th IFAC World Congress on Automat. Contr.*, Munich, FRG, 1987.
43. C. H. An, C. G. Atkeson, and J. M. Hollerbach, "Experimental Determination of the Effect of Feedforward Control on Trajectory Tracking Errors," *Proc. IEEE Int. Conf. Robotics and Automat.*, San Francisco, U.S.A., April, 1986.
44. P. K. Khosla and T. Kanade, "Experimental Evaluation of the Feedforward Compensation and Computed-Torque Control Schemes," *Proc. American Control Conference*, Seattle, U.S., June, 1986.
45. J. R. Hewit and J. S. Burdess, "Fast Dynamic Decoupled Control for Robotics, Using Active Force Control," *Mech. Machine Theory*, **16**, 535–542 (1981).