GENETIC DESIGN OF FUZZY-LOGIC CONTROLLERS FOR ROBOTIC MANIPULATORS

B Porter and N N Zadeh
Intelligent Machinery Division
Research Institute for Design, Manufacture, and Marketing
University of Salford
Salford M5 4WT
England

ABSTRACT

In this paper, genetic algorithms are used to design multivariable fuzzy-logic controllers for robotic manipulators using no information regarding the dynamical parameters of such manipulators. In particular, it is shown that genetic algorithms provide a very effective means of determining both the optimal set of fuzzy rules and the domains of the associated fuzzy sets for such fuzzy-logic controllers. It is demonstrated, in the case of a particular direct-drive two-link robotic manipulator, that such fuzzy-logic controllers can be readily robustified.

1. INTRODUCTION

In industrial applications, the computed-torque method (Craig (1989)) is frequently used in the control of non-redundant robotic manipulators. If the dynamical characteristics of such manipulators are known precisely, it is possible to introduce computed-torque/proportional-plus-derivativecontrollers. However, in practice the values of the dynamical parameters required in such controllers are never known precisely. Therefore, explicit identification schemes have been introduced in connection with adaptive computed-torque controllers in order to avoid these difficulties (Craig (1989)).

Such adaptive controllers for robotic manipulators are, however, rather too complicated for routine industrial applications. There is therefore much scope for the development of simple and reliable robust non-adaptive controllers for such manipulators. Thus, for example, a new class of computed-torque/fuzzy-logic controllers was introduced by Porter and Zadeh (1995). It was demonstrated by Porter and Zadeh (1995) that such computed-torque/fuzzy-logic controllers are more robust than computed-torque/proportional-plus-derivativecontrollers, even when such controllers are genetically tuned in the sense of Porter and Jones (1992) and Porter, Mohamed, and Jones (1993). However, the computed-torque/fuzzy-logic controllers of Porter and Zadeh (1995) still require precise estimates of the dynamical parameters.

In this paper, this need for such parameter estimates is circumvented by using multivariable fuzzy-logic controllers governed by non-linear control-law equations of the form

$$\tau = \Xi\ (e,\dot{e}) \qquad (1)$$

expressed in terms of fuzzy logic, where $\tau \epsilon R^n$ is the vector of actuator torques and $e \epsilon R^n$ is the vector of joint-angle errors. It is shown that genetic algorithms (Goldberg (1989)) provide a very effective means of determining both the optimal set of fuzzy rules and the domains of the associated fuzzy sets for such fuzzy-logic controllers. It is demonstrated that such controllers can be readily robustified to cope with uncertainties in the dynamical characteristics of robotic manipulators.

2. GENETIC DESIGN PROCEDURE

The multivariable fuzzy-logic controllers under investigation are governed by sets of rules of the following form:

If e_1 *is* $P_1^{(j_1)}$ *and* e_2 *is* $P_2^{(j_2)}$ ... *and* e_n *is* $P_n^{(j_n)}$ *and* $\dot{e}_1$ *is* $Q_1^{(k_1)}$ *and* $\dot{e}_2$ *is* $Q_2^{(k_2)}$ ... *and* $\dot{e}_n$ *is* $Q_n^{(k_n)}$ *then* τ_1 *is* $R_1^{(l_1)}$ *and* τ_2 *is* $R_2^{(l_2)}$ ... *and* τ_n *is* $R_n^{(l_n)}$. (2)

The entire sets of fuzzy sets in the e_i, $\dot{e}_i$, and u_i (i = 1, 2, ..., n) spaces are, respectively, $P_i = \{P_i^{(1)}, P_i^{(2)}, ..., P_i^{(p)}\}$, $Q_i = \{Q_i^{(1)}, Q_i^{(2)}, ..., Q_i^{(q)}\}$, and $R_i = \{R_i^{(1)}, R_i^{(2)}, ..., R_i^{(r)}\}$. (i = 1, 2, ..., n). In this notation $j_i \epsilon \{1, 2, ..., p\}$, $k_i \epsilon \{1, 2, ..., q\}$, and $l_i \epsilon \{1, 2, ..., r\}$ (i = 1, 2, ..., n). In addition, these entire fuzzy sets are symmetric and are, respectively, defined on the domains $[-\alpha_i, \alpha_i]$, $[-\beta_i, \beta_i]$, and $[-\gamma_i, \gamma_i]$ (i = 1, 2, ..., n).

It is evident from the generic rule (2) that there are altogether $(pq)^n$ rules each with

0-7803-2759-4/95/$4.00

a different antecedent for each of which the appropriate consequent must be determined in terms of the entire sets of fuzzy sets $R_i (i = 1, 2, \ldots, n)$. In addition, there are $3n$ domains of the fuzzy sets for each of which the appropriate parameter must be determined. The genetic design procedure accordingly represents each fuzzy-logic controller governed by equation (1) as an entire string of $\{(pq)^n + 3n\}$ concatenated sub-strings of binary digits. The Darwinian fitness, Φ, of the controller represented by each such string of binary digits can be conveniently obtained by evaluating the cost function

$$\Gamma = \int_0^T \|e(t)\| dt \tag{3}$$

when the robotic manipulator performs a task of duration. T Indeed,

$$\Phi = \nu - \Gamma \tag{4}$$

where $\nu \epsilon R^+$ is an appropriately large number.

The evolutionary process involved in the genetic design of these multivariable fuzzy-logic controllers for robotic manipulators starts by randomly generating an entire population of binary strings. Then, using the standard genetic operations of selection, crossover, and mutation (Goldberg (1989)), entire successive populations of binary strings are caused to evolve. In this way, fuzzy-logic controllers of progressively increasing fitness are produced until no significant further improvement is achievable. The controller with the largest achievable fitness, Φ_{max}, is clearly (in view of equation (4)) the required multivariable fuzzy-logic controller with the smallest achievable cost function, Γ_{min}. This evolutionary process thus provides the optimal controller when the robotic manipulator performs the specified task.

3. ILLUSTRATIVE EXAMPLE

This general genetic methodology can be conveniently illustrated by designing multivariable fuzzy-logic controllers for the practical direct-drive two-link robotic manipulator located in the laboratories at the University of Salford. In this case, n = 2, p = 3, q = 3, and r = 7. It follows that the fuzzy sets involved in the rules for the controller are $P_1 = \{P_1^{(1)}, P_1^{(2)}, P_1^{(3)}\}$ $P_2 = \{P_2^{(1)}, P_2^{(2)}, P_2^{(3)}\}$, $Q_1 = \{Q_1^{(1)}, Q_1^{(2)}, Q_1^{(3)}\}$, $Q_2 = \{Q_2^{(1)}, Q_2^{(2)}, Q_2^{(3)}\}$ $R_1 = \{R_1^{(1)}, R_1^{(2)}, R_1^{(3)}, R_1^{(4)}, R_1^{(5)}, R_1^{(6)}, R_1^{(7)}\}$, and $R_2 = \{R_2^{(1)}, R_2^{(2)}, R_2^{(3)}, R_2^{(3)}, R_2^{(4)}, R_2^{(5)}, R_2^{(6)}, R_2^{(7)}\}$ with corresponding domains $[-\alpha_1, \alpha_1]$, $[-\alpha_2, \alpha_2]$, $[-\beta_1, \beta_1]$, $[-\beta_2, \beta_2]$ $[-\gamma_1, \gamma_1]$, and $[-\gamma_2, \gamma_2]$. In this case, these are therefore $(3 \times 3)^2 = 81$ different antecedents in these rules together with $3 \times 2 = 6$ associated domains. However, it is assumed that any optimal set of such rules includes the following rule (where Z = fuzzy set ZERO):

If e_1 is Z and e_2 is Z and $\dot{e}_1$ is Z and $\dot{e}_2$ is Z then τ_1 is Z and τ_2 is Z. (5)

The removal of this rule and the assumption of symmetry amongst the remaining 80 rules thus leads to the need to determine the appropriate consequent in only 40 rules. In each such rule, the sub-sets of P_1 and P_2 can each be represented by 3 bits so that the rules involved in each multivariable fuzzy-logic controller can be represented by a sub-string of 40 x 6 = 240 bits. If, in addition, each of the 6 domains is represented by 10 bits, the domains involved in each fuzzy-logic controller can be represented by a further sub-string of 6 x 10 = 60 bits. Therefore, each fuzzy-logic controller is represented by a string of 240 x 60 = 300 bits. In applying the genetic algorithm, it was found that excellent results were obtained using a population N = 20, a crossover probability $p_c = 0.8$, a mutation probability $p_m = 0.05$, with evolution occurring over 1000 generations.

In order to demonstrate the ease with which robust multivariable fuzzy-logic controllers can be genetically designed for robotic manipulators <u>using no information regarding the dynamical parameters of such manipulators,</u> it is instructive to perform the following series of three designs:

(i) The controller is first designed in the absence of any payload, and the resulting good time-domain behaviour of the end effector in the cartesian x-y plane is as shown in Figure 1(a) for which the associated cost function is

$$\Gamma_1 = 144.2 \times 10^{-4}. \tag{6}$$

If this controller is now used when the manipulator carries a payload of 4kg, then the resulting very oscillatory time-domain behaviour of the end effector is as shown in Figure 1(b) for which the associated cost function is

$$\Gamma_2 = 360.0 \times 10^{-4}. \quad (7)$$

(ii) The controller is then designed in the presence of a payload of 4kg, and the resulting good time-domain behaviour of the end effector is as shown in Figure 2(b) for which the associated cost function is

$$\Gamma_3 = 236 \times 10^{-4}. \quad (8)$$

If this controller is now used in the absence of any payload, then the resulting rather oscillatory time-domain behaviour of the end effector is as shown in Figure 2(a) for which the associated cost function is

$$\Gamma_4 = 317 \times 10^{-4}. \quad (9)$$

(iii) The controller is then designed in the case of a composite task consisting of the task in the absence of any payload together with the task in the presence of a payload of 4kg. If this controller is now used in the absence of any payload, then the resulting good time-domain behaviour of the end effector is as shown in Figure 3(a) for which the associated cost function is

$$\Gamma_5 = 185.8 \times 10^{-4}. \quad (10)$$

Finally, if this controller is now used in the presence of a payload of 4kg, then the resulting acceptable time-domain behaviour of the end effector is as shown in Figure 3(b) for which the associated cost function is

$$\Gamma_6 = 259.7 \times 10^{-4}. \quad (11)$$

These results indicate that the genetic design procedure involved in (iii) readily produces robust controllers. Thus, $\Gamma_5 < \Gamma_4$ even though $\Gamma_5 > \Gamma_1$; and $\Gamma_6 < \Gamma_2$ even though $\Gamma_6 > \Gamma_3$: in addition,

$\Gamma_5 + \Gamma_6 < \Gamma_1 + \Gamma_2$ and $\Gamma_5 + \Gamma_6 < \Gamma_3 + \Gamma_4$.

4. CONCLUSION

In this paper, genetic algorithms have been used to design multivariable fuzzy-logic controllers for robotic manipulators using no information regarding the dynamical parameters of such manipulators. In particular, it has been shown that genetic algorithms provide a very effective means of determining both the optimal sets of fuzzy rules and the domains of the associated fuzzy sets for such fuzzy-logic controls. It has been demonstrated, in the case of a particular direct-drive two-link robotic manipulator, that such fuzzy-logic controllers can be readily robustified to cope with uncertainties in the dynamical characteristics of robotic manipulators.

REFERENCES

[1] J J Craig, "Introduction to Robotics", Addison-Wesley, 1989.

[2] D E Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.

[3] B Porter and A H Jones, "Genetic tuning of digital PID controllers", Electronics Letters, vol 28, p 843, 1992.

[4] B Porter, S S Mohamed, and A H Jones, "Genetic tuning of digital PID controllers for linear multivariable plants", Proc European Control Conference, Groningen, The Netherlands, June/July 1993.

[5] B Porter and N N Zadeh, "Genetic design of computed-torque/fuzzy-logic controllers for robotic manipulators", Proc IEEE International Symposium on Intelligent Control, Monterey, USA, August 1995.

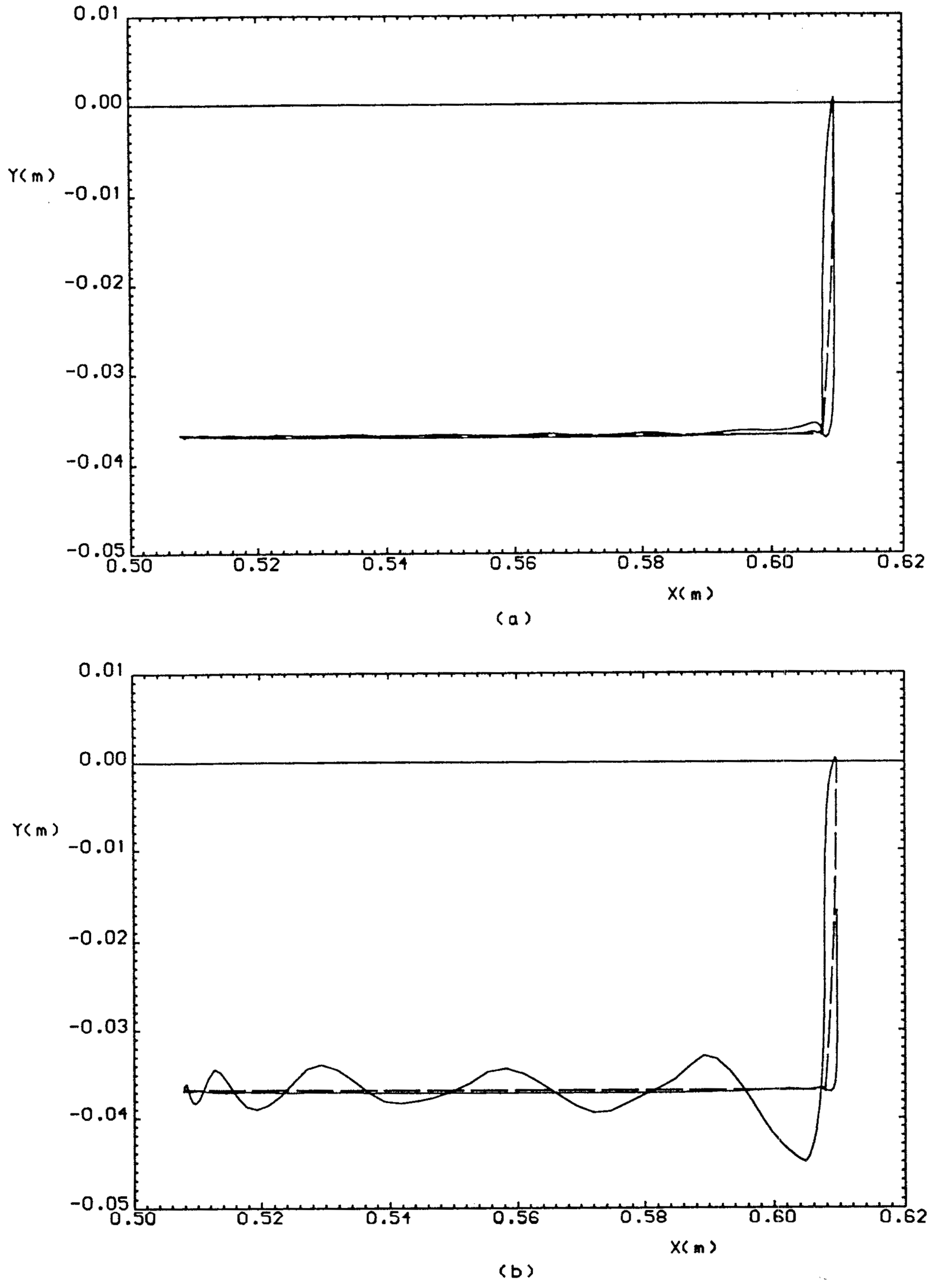


Figure 1 : Trajectory-tracking performance of controller tuned with no payload
(a) Without payload (b) With 4kg payload

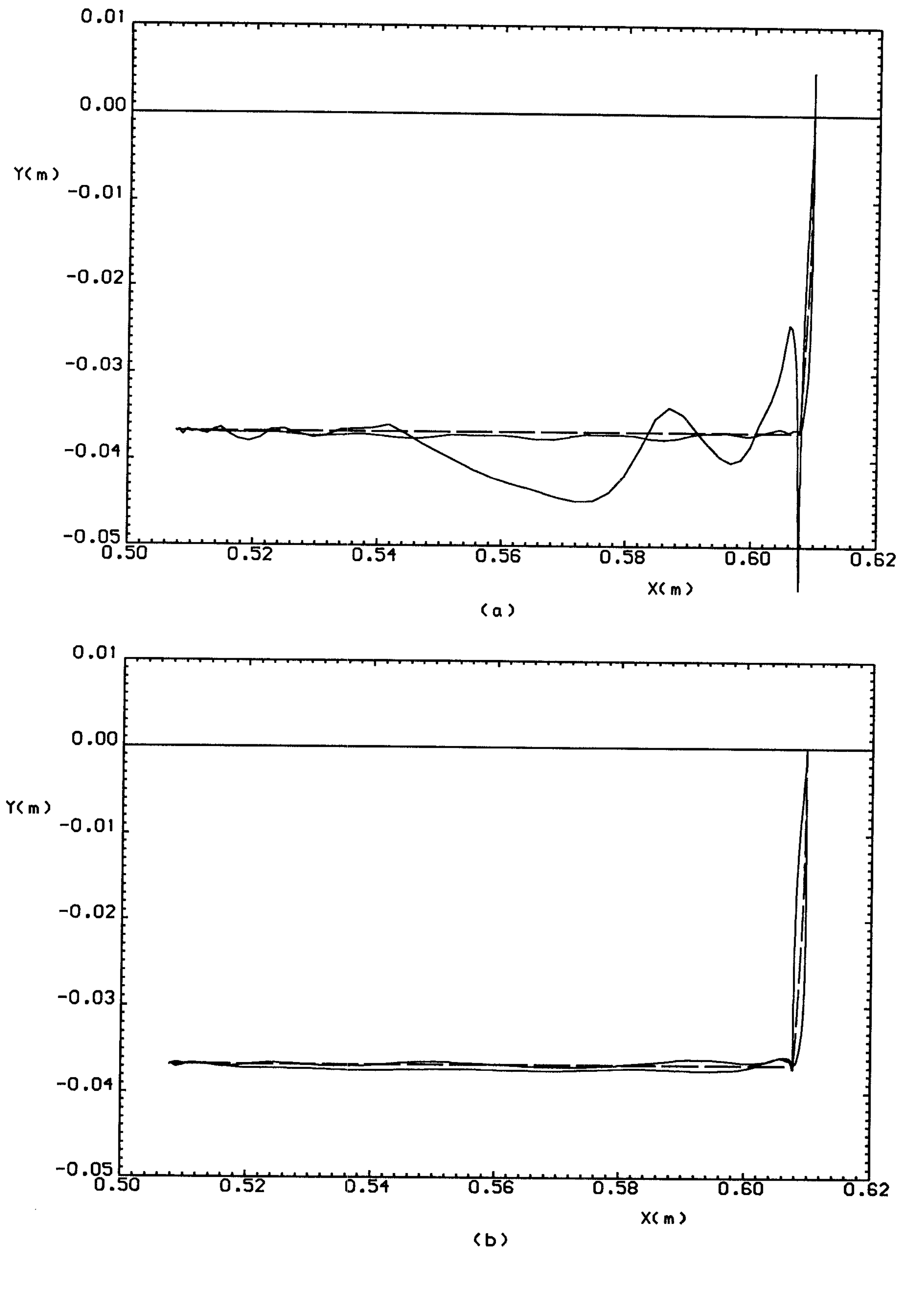
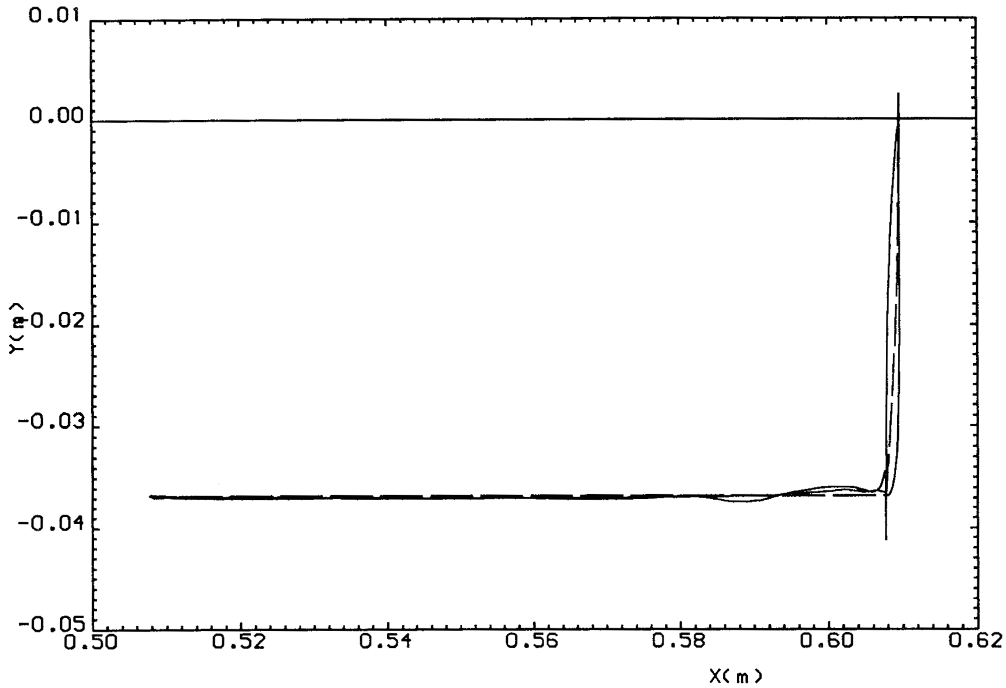
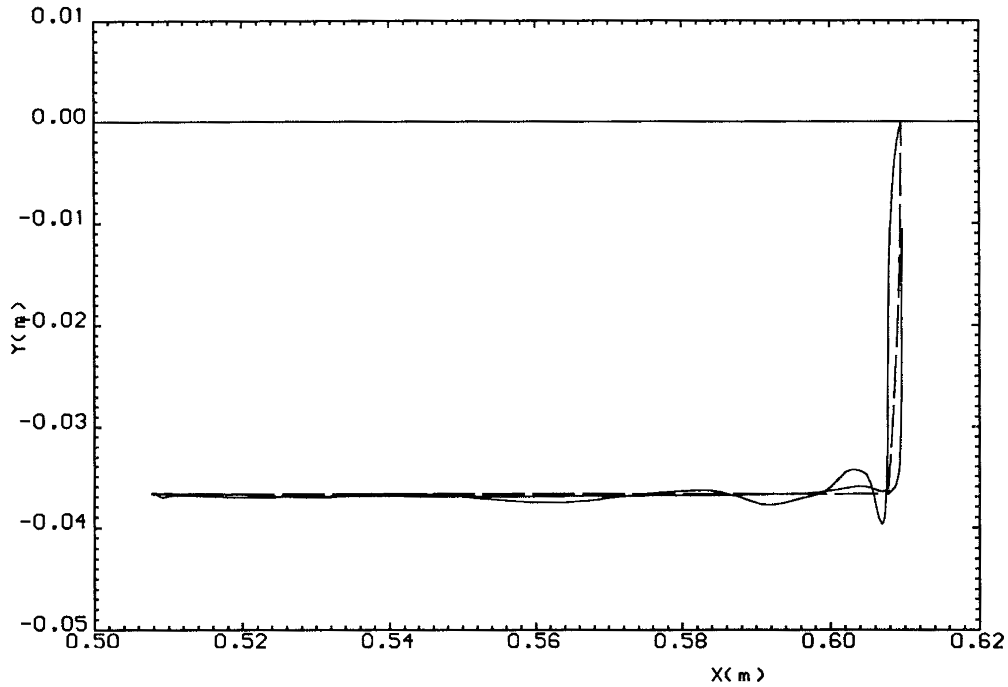


Figure 2 : Trajectory-tracking performance of controller tuned with 4kg payload
(a) Without payload (b) With 4kg payload



(a)



(b)

Figure 3 : Trajectory-Tracking performance of controller tuned for composite task (a) Without payload (b) With 4kg payload