

GENETIC DESIGN OF COMPUTED-TORQUE/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, a new class of computed-torque/fuzzy-logic controllers is introduced and designed using genetic algorithms. It is demonstrated that these computed-torque/fuzzy-logic controllers are more robust than computed-torque/proportional-plus-derivative controllers.

1. INTRODUCTION

In practice, the computed-torque method (Craig (1989)) is frequently used in the control of non-redundant robotic manipulators. The dynamical behaviour of such manipulators is governed by non-linear vector-matrix differential equations of the form

$$M(\theta)\ddot{\theta} + v(\theta, \dot{\theta}) + g(\theta) + f(\dot{\theta}) = \tau \dots (1)$$

if all elastic effects are neglected. In equation (1), $M(\theta) \in R^{n \times n}$ is the inertia matrix, $v(\theta, \dot{\theta}) \in R^n$ is the vector of centrifugal and coriolis torques, $g(\theta) \in R^n$ is the vector of gravitational torques, $f(\dot{\theta}) \in R^n$ is the vector of friction torques, $\tau \in R^n$ is the vector of actuator torques, and $\theta \in R^n$ is the vector of joint angles. If the dynamical characteristics of such manipulators are known precisely, it is possible to introduce computed-torque / proportional-plus-derivative controllers governed by control-law equations of the form

$$\tau = M(\theta)(\ddot{\theta}_d + K_1\dot{e} + K_2e) + h(\theta, \dot{\theta}) \dots (2)$$

In equation (2), $\theta_d \in R^n$ is the vector of desired joint angles, $e = \theta_d - \theta \in R^n$ is the vector of joint-angle errors, $K_1 \in R^{n \times n}$ is the proportional

gain matrix, $K_2 \in R^{n \times n}$ is the derivative gain matrix, and

$$h(\theta, \dot{\theta}) = v(\theta, \dot{\theta}) + g(\theta) + f(\dot{\theta}) \in R^n \dots (3)$$

It is then evident from equations (1) and (2) that, under the action of such computed-torque control, the vector of joint-angle errors is governed by the linear vector-matrix differential equation

$$\ddot{e} + K_2\dot{e} + K_1e = 0 \dots (4)$$

Equation (4) clearly indicates that the controller matrices $K_1 \in R^{n \times n}$ and $K_2 \in R^{n \times n}$ can be readily selected so as to produce linear second-order error dynamics with any required characteristics, eg, if $K_1 \in R^{n \times n}$ and $K_2 \in R^{n \times n}$ are diagonal, then the error dynamics of each joint of the robotic manipulator are decoupled.

However, in the case of practical robotic manipulators, the dynamical parameters required to formulate the control-law equation (2) are never known precisely. Such computed-torque/proportional-plus-derivative control-law equations will therefore in practice not have the form of equation (2) but rather the form

$$\tau = \hat{M}(\theta)(\ddot{\theta}_d + K_1\dot{e} + K_2e) + \hat{h}(\theta, \dot{\theta}), \dots (5)$$

where $\hat{M}(\theta) \in R^{n \times n}$ is the estimated value of the inertia matrix and $\hat{h}(\theta, \dot{\theta}) \in R^n$ is the estimated value of the centrifugal, coriolis, gravitational, and frictional vector defined in equation (5). In such practical cases, it is evident from equations (1), (3), and (5) that the vector of joint errors is

governed by the vector-matrix differential equation

$$\ddot{e} + K_2 \dot{e} + K_1 e = \hat{M}^{-1}(\theta) \times \left[\{h(\theta, \dot{\theta}) - \hat{h}(\theta, \dot{\theta})\} + \{M(\theta) - \hat{M}(\theta)\} \dot{\theta} \right] \dots (6)$$

Equation (6) clearly reduces to equation (4) in the special case when $\hat{M}(\theta) = M(\theta)$ and $\hat{h}(\theta) = h(\theta)$. However, in the general case when $\hat{M}(\theta) \neq M(\theta)$ and $\hat{h}(\theta) \neq h(\theta)$, it is evident that the choice of $K_1 \in R^{n \times n}$ and $K_2 \in R^{n \times n}$ made in connection with equation (4) will not normally be sufficient to ensure that the error dynamics of the robotic manipulator governed by equation (6) are acceptable. Therefore, explicit identification schemes have been introduced in combination with adaptive computed-torque controllers in order to provide good estimates of $M(\theta)$ and $h(\theta)$. However, since such adaptive controllers for robotic manipulators are rather complicated for routine industrial applications, there is much scope for the development of simple and reliable robust non-adaptive controllers for robotic manipulators.

In this paper, a class of computed-torque/fuzzy-logic controllers for such manipulators is therefore introduced. These new controllers differ from the conventional computed-torque/proportional-plus-derivative controllers governed by equation (2) in that the linear control law

$$u = K_1 e + K_2 \dot{e} \dots (7)$$

incorporated in equation (2) is replaced by a non-linear control law of the form

$$u = K(e, \dot{e}) \dots (8)$$

expressed in terms of fuzzy logic. It is demonstrated that such computed-torque/fuzzy-logic controllers are more robust than computed-torque / proportional-plus-derivative controllers. This demonstration is effected in the case of a particular direct-drive two-link robotic manipulator in accordance with the following procedure:

- (i) both controllers are first genetically tuned (Porter and Jones (1992), Porter, Mohamed, and Jones (1993)) when all the dynamical characteristics of the manipulator are known precisely;
- (ii) both controllers are then deployed (without further tuning) for the same trajectory-tracking task when all the dynamical characteristics are not known precisely.

2. GENETIC DESIGN PROCEDURES

2.1 Computer-Torque/Proportional-Plus-Derivative Controllers

In this case, the controllers are governed by control-law equations of the form

$$\tau = M(\theta)(\ddot{\theta}_d + u) + h(\theta, \dot{\theta}) \dots (9a)$$

where

$$u = K_1 e + K_2 \dot{e} \dots (9b)$$

and it is assumed that both $M(\theta)$ and $h(\theta, \dot{\theta})$ are known precisely. The problem is to determine the proportional controller matrix $K_1 \in R^{n \times n}$ and the derivative controller matrix $K_2 \in R^{n \times n}$ such that the cost function

$$\Gamma = \int_0^T \|e(t)\| dt \dots (10)$$

is minimised when the robotic manipulator performs a specified trajectory-tracking task of duration, T. This problem can be readily solved using genetic algorithms (Porter and Jones (1992), Porter, Mohamed, and Jones (1993)) by representing each proportional-plus-derivative controller governed by equation (9b) as a string of $2n^2$ concatenated sub-strings of binary digits. Each such sub-string represents in binary coded form the individual scalar elements of the controller matrices $K_1 \in R^{n \times n}$ and $K_2 \in R^{n \times n}$. The Darwinian fitness, Φ , of the controller represented by each such entire string of binary digits is obtained by evaluating the cost function defined in equation (10) and then calculating

$$\Phi = 1/\Gamma \dots (11)$$

Then, using the standard genetic operations of selection, crossover, and mutation (Goldberg (1989)), entire populations of binary strings are caused to evolve. In this way, proportional-plus-derivative 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 (11)) the required proportional-plus-derivative controller with the smallest achievable cost function, Γ_{min} . This provides the genetically optimal computed-torque/proportional-plus-derivative controller when the robotic manipulator performs the specified trajectory-tracking task.

2.2 Computed-Torque/Fuzzy-Logic Controllers

In this case, the controllers are governed by control-law equations of the form

$$\tau = M(\theta)(\ddot{\theta}_d + u) + h(\theta, \dot{\theta}) \quad \dots (12a)$$

where

$$u = K(e, \dot{e}) \quad \dots (12b)$$

and it is assumed that both $M(\theta)$ and $h(\theta, \dot{\theta})$ are known precisely. The problem is to express the non-linear controller function $k(e, \dot{e})$ in equation (12b) in terms of fuzzy logic such that the cost function (10) is again minimised when the robotic manipulator performs the same specified trajectory-tracking task of duration, T . This problem can be readily solved by using genetic algorithms to select the domains of fuzzy sets defined in e , \dot{e} , and u spaces in the fuzzy-logic representation of the control-law equation (12b). This representation can be achieved in many different ways; but, in the present context, the genetic design procedure can be conveniently explained in terms of the decoupled version

$$u_i = k_i(e_i, \dot{e}_i) \quad (i = 1, 2, \dots, n) \quad \dots (14)$$

of equation (12b). Then, each of the n decoupled fuzzy-logic controllers is governed by rules of the following form:

$$\text{If } e_i \text{ is } P_i^{(j)} \text{ and } \dot{e}_i \text{ is } Q_i^{(k)} \text{ then } u_i \text{ is } R_i^{(l)}$$

The entire sets of fuzzy sets in the e_i , \dot{e}_i , and u_i spaces are, respectively, $P_i = \{P_i^{(1)}, P_i^{(2)}, \dots, P_i^{(N)}\}$, $Q_i = \{Q_i^{(1)}, Q_i^{(2)}, \dots, Q_i^{(N)}\}$ and $R_i = \{R_i^{(1)}, R_i^{(2)}, \dots, R_i^{(N)}\}$ ($i=1, 2, \dots, n$). These entire fuzzy sets are symmetric and are, respectively, defined on the domains $[-\alpha_i, \alpha_i]$, $[-\beta_i, \beta_i]$, and $[-\gamma_i, \gamma_i]$ ($i=1, 2, \dots, n$). The genetic design procedure represents each fuzzy-logic controller represented by equations (14) as a string of $3n$ concatenated sub-strings of binary digits. Each such sub-string represents in binary coded form the individual domains of the entire fuzzy sets P_i , Q_i , R_i ($i=1, 2, \dots, n$). The Darwinian fitness, Φ , of the controller represented by each such entire string of binary digits is obtained by evaluating the cost function, Γ , defined in equation (10) and then calculating $\Phi = 1/\Gamma$.

Then, using again the standard genetic operations of selection, crossover, and mutation (Goldberg (1989)), entire 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 (16)) the required fuzzy-logic controller with the smallest achievable cost function, Γ_{min} . This provides the genetically optimal computed-torque/fuzzy-logic controller when the robotic manipulator performs the specified trajectory-tracking task.

3. ILLUSTRATIVE EXAMPLE

These genetic methodologies can be conveniently illustrated by designing computed-torque controllers for the practical direct-drive two-link robotic manipulator located in the laboratories at the University of Salford. The dynamical characteristics of this manipulator are known very precisely as a result of using the genetic identification procedure of Porter, Sangolola, and Zadeh (1994). It is therefore possible to use the previously described procedures to design both computed-torque/proportional-plus-derivative and computed-torque/fuzzy-logic controllers for this manipulator.

Thus, in the case of the genetically designed computed-torque / proportional-plus-derivative

controller, the excellent time-domain behaviour in the Cartesian x,y -plane of the end effector of this robotic manipulator is as shown in Figure 1(a). In this case, the actual trajectory is very close to the desired trajectory of the end effector with an associated cost function of

$$\Gamma_1 = 54.4 \times 10^{-4} \dots (15)$$

However, if this same proportional-plus-derivative controller is now used when the manipulator carries an additional 4 kg payload, the resulting time-domain behaviour in the Cartesian x,y -plane of the end effector is as shown in Figure 1(b). In this case, the actual trajectory oscillates conspicuously in the vicinity of the desired trajectory of the end effector with an associated cost function of

$$\Gamma_2 = 487.0 \times 10^{-4} \dots (16)$$

It is evident that $\Gamma_2 > \Gamma_1$ and that the computed-torque/proportional-plus-derivative controller is therefore not robust in the presence of unmodelled payload dynamics.

However, in the case of the genetically designed computed-torque/fuzzy-logic controller, the excellent time-domain behaviour in the Cartesian x,y -plane of the end effector of the robotic manipulator with no payload is as shown in Figure 2(a). In this case, the actual trajectory is very close to the desired trajectory of the end effector with an associated cost function

$$\Gamma_3 = 39.7 \times 10^{-4} \dots (17)$$

However, if this same fuzzy-logic controller is now used when the manipulator carries an additional 4 kg payload, the resulting time-domain behaviour in the Cartesian x,y -plane of the end effector is as shown in Figure 2(b). In this case, the actual trajectory does not oscillate and is quite close to the desired trajectory with an associated cost function

$$\Gamma_4 = 160.0 \times 10^{-4} \dots (18)$$

It is evident that, although $\Gamma_4 > \Gamma_3$, nevertheless $\Gamma_4 / \Gamma_3 < \Gamma_2 / \Gamma_1$. This means that the computed-torque/fuzzy-logic controller is more robust in the presence of unmodelled payload

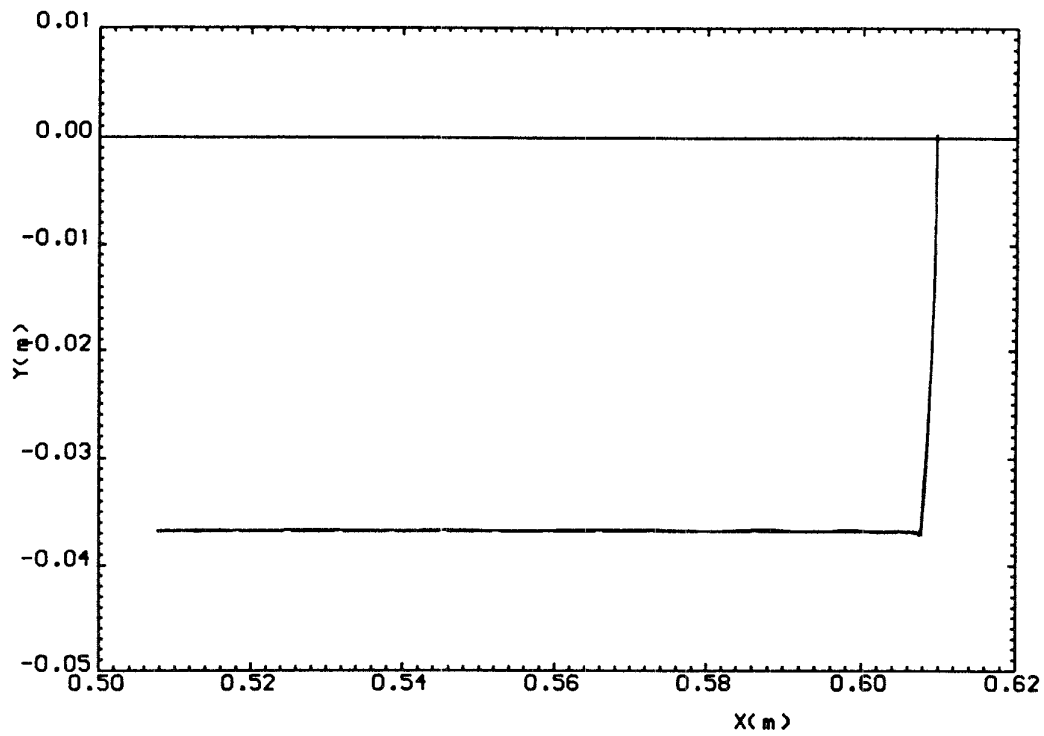
dynamics than is the computed-torque/proportional-plus-derivative controller. Note, however, that this increased robustness is achieved with no loss of performance in the design case since $\Gamma_3 < \Gamma_1$.

4. CONCLUSION

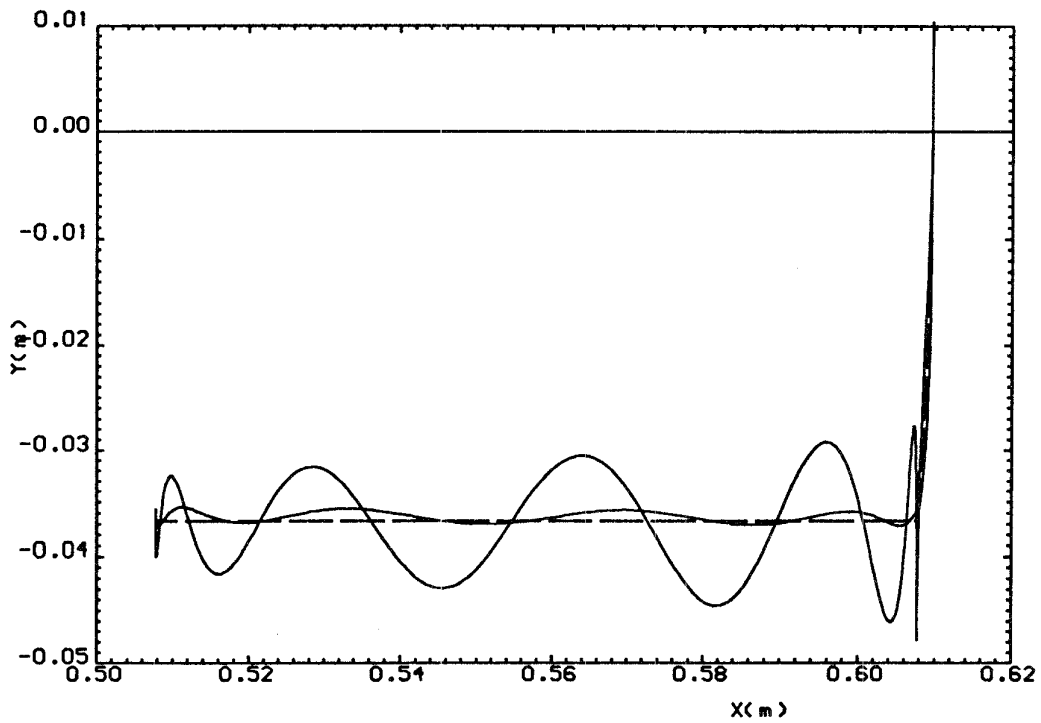
In this paper, a class of computed-torque/fuzzy-logic controllers for robotic manipulators has been introduced and designed using genetic algorithms. These new controllers differ from conventional computed-torque/proportional-plus-derivative controllers in that the linear proportional-plus-derivative control law in the computed-torque controller is replaced by a non-linear control law expressed in terms of fuzzy logic. It has been demonstrated that these computed-torque/fuzzy-logic controllers are more robust than computed-torque/proportional-plus-derivative controllers. This demonstration has been effected in the case of a particular direct-drive two-link manipulator.

REFERENCES

- J J Craig, "Introduction to Robotics", Addison-Wesley, 1989.
- D E Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.
- B Porter and A H Jones, "Genetic tuning of digital PID controllers", Electronics Letters, Vol 28, pp 843, 1992.
- 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.
- B Porter, B A Sangolola, and N N Zadeh, "Genetic design of computed-torque controllers for robotic manipulators", Proc IASTED International Conference on Systems and Control, Lugano, Switzerland, June 1994.

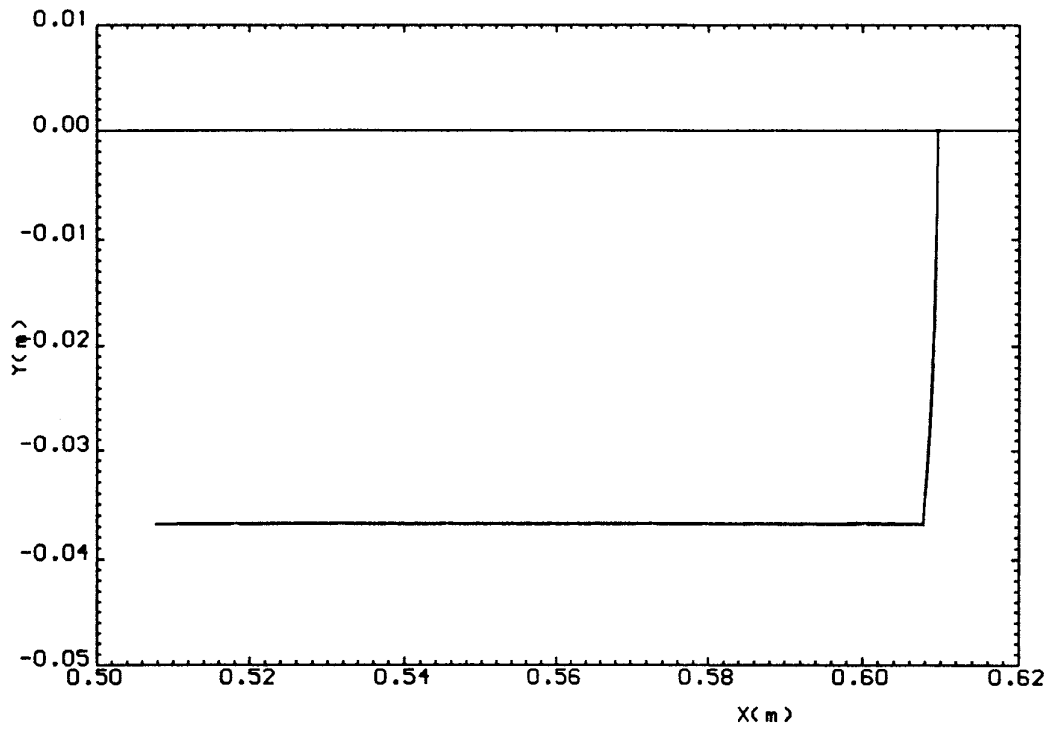


(a)

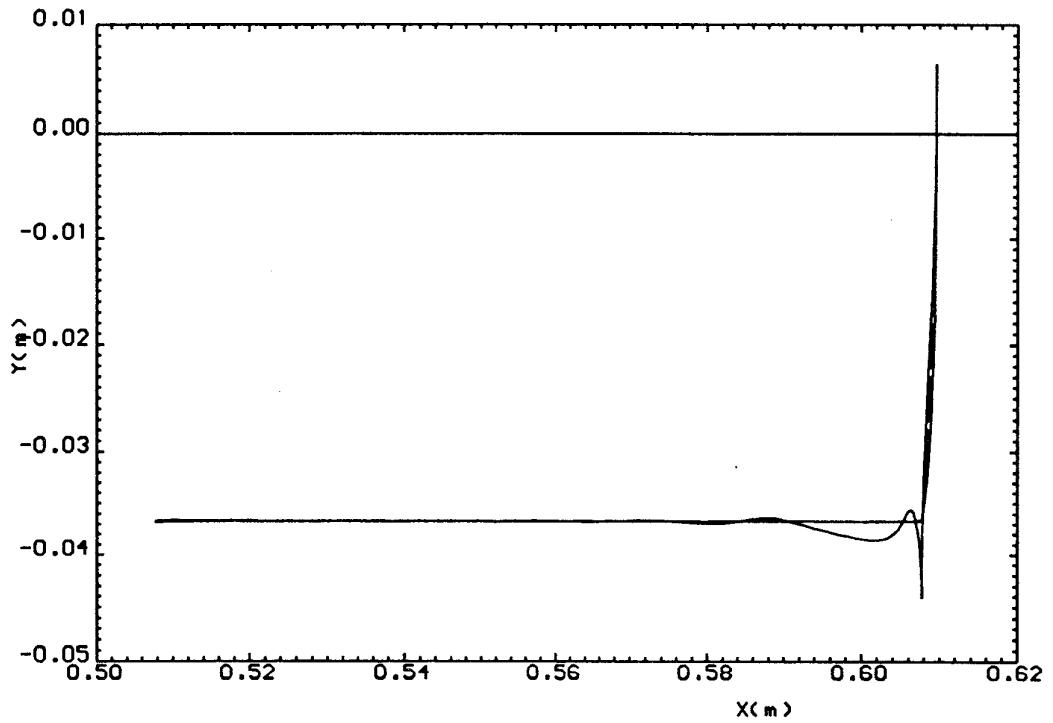


(b)

Figure 1 :Trajectory-tracking performance of
 computed-torque/PD controller
 (a) Without payload (b) With 4kg payload



(a)



(b)

Figure 2 : Trajectory-tracking performance of computed-torque/fuzzy logic controller
 (a) Without payload (b) With 4kg payload