

Hybrid Fuzzy/Crisp-Logic Control of Manufacturing Systems

B PORTER

Department of Industrial and Manufacturing Systems
Engineering
The University of Hong Kong
Hong Kong

H MOI

Department of Industrial and Manufacturing Systems
Engineering
The University of Hong Kong
Hong Kong

Abstract — In recent years, techniques such as dynamic programming, the maximum principle, linear programming, and genetic algorithms have been used to synthesise optimal control policies for manufacturing systems. However, such techniques are frequently rather opaque and often yield control policies that are implemented by open-loop rather than closed-loop control systems. In this paper, it is therefore shown that closed-loop systems incorporating hybrid fuzzy/crisp-logic controllers can be readily designed for manufacturing systems. This hybrid approach is illustrated by reference to the closed-loop control of a simple manufacturing system producing a single part type.

I. INTRODUCTION

In modern industry, it is important that manufacturing systems be controlled so as to satisfy production schedules and to minimise manufacturing costs. Therefore, in recent years, various techniques such as dynamic programming, the maximum principle, and linear programming have been used to synthesise such optimal control policies for manufacturing systems [1] - [6]. Such synthesis techniques are needed because the state and control vectors involved in the mathematical models of manufacturing systems must satisfy numerous complicated constraints, which create major difficulties both for the design and for the operation of conventional multivariable control systems. However, synthesis techniques like dynamic programming [1] and the maximum principle [5] entail rapidly escalating computational difficulties as system complexity increases. Such computational difficulties are avoided by the use of linear programming [6] to synthesise optimal control policies for manufacturing systems; but this tractability is achieved at the expense of having to deal only indirectly with the dynamical complexities of such systems.

In order to circumvent these difficulties, it has recently been shown by Porter and Allaoui [7] that the genetic design methodology developed by Porter [8] can be readily used to synthesise optimal control policies for manufacturing systems. However, the optimal control policies thus obtained are

unable to function automatically for different part demand rates since they are implemented by open-loop rather than closed-loop control systems. In this paper, it is therefore shown that closed-loop systems incorporating hybrid fuzzy/crisp-logic controllers can be readily designed for manufacturing systems. This hybrid approach is illustrated by reference to the closed-loop control of a simple manufacturing system producing a single part type.

II. CLOSED-LOOP HYBRID CONTROL

The manufacturing systems to be controlled comprise m machines (with n associated buffers) and produce p part types. The dynamical behaviour of such systems is governed by linear differential equations of the forms [6][9]

$$\dot{q}(t) = A_1 u(t) + A_2 i(t) \quad (1)$$

and

$$\dot{x}(t) = A_3 u(t) - d(t) \quad (2)$$

for the buffer dynamics and for the production dynamics, respectively. In these equations, $q(t) \in \mathbb{R}^n$ is the vector of buffer levels, $x(t) \in \mathbb{R}^p$ is the vector of finished parts, $u(t) \in \mathbb{R}^n$ is the vector of buffer production rates, $i(t) \in \mathbb{R}^p$ is the vector of part release rates, and $d(t) \in \mathbb{R}^p$ is the vector of part demand rates. In addition, $A_1 \in \mathbb{R}^{n \times n}$ is the buffer routing matrix such that $A_1 u(t)$ represents the flows of parts between buffers; $A_2 \in \mathbb{R}^{n \times p}$ is the buffer loading matrix such that $A_2 i(t)$ represents the arrivals of external parts at the buffers; and $A_3 \in \mathbb{R}^{p \times n}$ is the output matrix such that $A_3 u(t)$ represents the flow of finished parts. Finally, if τ_j is the processing time of parts in buffer j and $B^{(k)}$ is the set of buffers for machine k ($k=1,2,\dots,m$), then the buffer production rates must satisfy the capacity constraints

$$\sum_{j \in B^{(k)}} \tau_j u_j(t) \leq 1 \quad (k = 1, 2, \dots, m) . \quad (3)$$

In addition, the state and control vectors must satisfy the constraints

$$q(t) \geq 0 , \quad (4)$$

$$u(t) \geq 0 , \quad (5)$$

and

$$i(t) \geq 0 . \quad (6)$$

However, $x(t)$, the state vector of finished parts, is not required to satisfy such a non-negativity constraint (since there may be either a surplus or a backlog of finished parts).

The control problem is to find, over some time period of duration T , the vectors $i(t) \in \mathbb{R}^p$ and $u(t) \in \mathbb{R}^n$ of part release rates and buffer production rates in response to a specified vector $d(t) \in \mathbb{R}^p$ of part demand rates. More precisely, the objective is to choose the control vectors $i(t) \in \mathbb{R}^p$ and $u(t) \in \mathbb{R}^n$ so as to minimise the cost function

$$\Gamma = \int_0^T [\lambda q(t) + \mu^+ x^+(t) + \mu^- x^-(t)] dt , \quad (7)$$

where

$$x^+(t) = \max \{x(t), 0\} \in \mathbb{R}^p \quad (8a)$$

is the parts surplus vector,

$$x^-(t) = \max \{-x(t), 0\} \in \mathbb{R}^p , \quad (8b)$$

is the parts backlog vector, and $\lambda \in \mathbb{R}^{1 \times n}$, $\mu^- \in \mathbb{R}^{1 \times p}$, $\mu^+ \in \mathbb{R}^{1 \times p}$ are weighting vectors for the buffer contents and the finished parts surplus or backlog. It is evident that the control vectors $i(t) \in \mathbb{R}^p$ and $u(t) \in \mathbb{R}^n$ that minimise this cost function, Γ , are optimal in the sense that the entire cost function associated with work-in-progress (as measured by the buffer levels), production surplus, and production backlog is minimised. However, the solution of this optimisation problem is non-trivial because the control vectors $i(t) \in \mathbb{R}^p$ and $u(t) \in \mathbb{R}^n$ must satisfy the constraints (3), (5), and (6) whilst the state vector $q(t) \in \mathbb{R}^n$ must satisfy the constraint (4).

This problem has, nevertheless, recently been solved by Porter and Allaoui [7] using the genetic design methodology developed by Porter [8]. However, the resulting genetically synthesised optimal control policies are implemented by open-loop rather than closed-loop control systems. In order to facilitate the closed-loop control of manufacturing systems, it is possible to introduce hybrid fuzzy/crisp-logic controllers. Since it is required that such controllers function automatically for different part demand rate vectors, $d(t) \in \mathbb{R}^p$, it is convenient to control the vector of finished parts, $x(t) \in \mathbb{R}^p$, by using a fuzzy-logic controller to generate the vector of buffer production rates, $u(t) \in \mathbb{R}^n$; and to control the vector of buffer levels, $q(t) \in \mathbb{R}^n$, by using a crisp-logic controller to generate the vector of part release rates, $i(t) \in \mathbb{R}^p$. The design of both the fuzzy-logic components and the crisp-logic components of such hybrid controllers so as to ensure that both $\|x(t)\| \rightarrow 0$ and $\|q(t)\| \rightarrow 0$ is straightforward because of the special structure of the linear differential equations (1) and (2).

III. ILLUSTRATIVE EXAMPLE

This general approach to the hybrid fuzzy/crisp-logic control of manufacturing systems can be conveniently illustrated by considering a simple system in which $m=2$, $n=2$, and $p=1$ [7]. In this case, equation (1) for the buffer dynamics assumes the scalar forms

$$\dot{q}_1(t) = i(t) - u_1(t) \quad (9)$$

and

$$\dot{q}_2(t) = u_1(t) - u_2(t) , \quad (10)$$

whilst equation (2) for the production dynamics assumes the scalar form

$$\dot{x}(t) = u_2(t) - d(t) . \quad (11)$$

It is assumed that $\tau_1 = 0.5$ and $\tau_2 = 0.5$ so that, in view of the inequalities (3), the production rates must satisfy the constraints

$$u_1(t) \leq 2 \quad (12)$$

and

$$u_2(t) \leq 2 . \quad (13)$$

In addition, it follows from the inequalities (4), (5), and (6) that the state variables and control variables must satisfy the constraints

$$q_1(t), q_2(t) \geq 0 , \quad (14)$$

$$u_1(t), u_2(t) \geq 0 , \quad (15)$$

and

$$i(t) \geq 0 . \quad (16)$$

The initial state of the system is such that

$$q_1(0) = 5 , \quad (17a)$$

$$q_2(0) = 0 , \quad (17b)$$

and

$$x(0) = 0 . \quad (17c)$$

It is assumed that control is to be exercised on the time interval $[0,10]$, and that the part demand rate on this interval is

$$d(t) = 1 \quad (0 \leq t \leq 10) . \quad (18)$$

The objective of such control is to generate $i(t)$, $u_1(t)$, and $u_2(t)$ so as to minimise the cost function in equation (7) with $T = 10$, $\lambda = [5,10]$, and $\mu^+ = \mu^- = 5$.

In order to achieve this objective, the finished parts level, $x(t)$, is controlled by a fuzzy-logic controller embodying the following five rules:

- FR1. If x is P, then Δu_2 is NB.
- FR2. If x is N, then Δu_2 is PB.
- FR3. If x is Z and \dot{x} is Z, then Δu_2 is Z.
- FR4. If x is Z and \dot{x} is P, then Δu_2 is NS.
- FR5. If x is Z and \dot{x} is N, then Δu_2 is PS.

However, the buffer levels, $q_1(t)$ and $q_2(t)$, are controlled by a crisp-logic controller embodying the following four rules:

- CR1. If $q_1 > 0$, then $i = 0$.
- CR2. If $q_1 = 0$, then $i = u_1$.
- CR3. If $q_2 > 0$, then $u_1 = 0$.

- CR4. If $q_2 = 0$, then $u_1 = u_2$.

The hybrid controller embodying these nine rules provides a simpler means of controlling the multivariable manufacturing system than does a completely fuzzy controller. Indeed, it is evident from equation (9) that $q_1(t)$ depends directly on both $i(t)$ and $u_1(t)$; and from equation (10) that $q_2(t)$ depends directly on both $u_1(t)$ and $u_2(t)$. But the hybrid controller essentially decouples these control tasks by first using the fuzzy-logic controller to generate $u_2(t)$, and then using the crisp-logic controller to generate $u_1(t)$ in terms of $u_2(t)$ and $i(t)$ in terms of $u_1(t)$.

The results obtained when this hybrid fuzzy/crisp-logic controller is implemented are shown in Figs 1 and 2 when the system is in the initial state defined by equations (17) and the part demand rate is given by equation (18). Thus, the state variables $q_1(t)$, $q_2(t)$, and $x(t)$ are as shown in Figs 1(a), 1(b), and 1(c), respectively; whilst the controls $i(t)$, $u_1(t)$, and $u_2(t)$ generated by the hybrid controller are as shown in Figs 2(a), 2(b), and 2(c), respectively. It is evident that this closed-loop behaviour is very similar to the following behaviour of the optimally controlled system (for which the cost function, Γ , is equal to 62.5):

- (i) $q_1(t)$ reduces linearly from 5 to 0 when $t = 5$, and then remains equal to 0;
- (ii) $q_2(t)$ equals 0 when $0 \leq t \leq 10$;
- (iii) $x(t)$ equals 0 when $0 \leq t \leq 10$;
- (iv) $i(t)$ equals 0 when $0 \leq t < 5$ and 1 when $5 < t \leq 10$.
- (v) $u_1(t)$ equals 1 when $0 \leq t \leq 10$.
- (vi) $u_2(t)$ equals 1 when $0 \leq t \leq 10$.

However, it is important to note that the near-optimal results shown in Figs 1 and 2 were obtained automatically by the closed-loop system incorporating the hybrid fuzzy/crisp-logic controller.

The ability of the same hybrid controller to deal automatically with the same part demand rate when the system is in the different initial state

$$q_1(0) = 3 , \quad (19a)$$

$$q_2(0) = 3 , \quad (19b)$$

and

$$x(0) = 1 \quad (19c)$$

is demonstrated by the results shown in Figs 3 and 4. Thus, the state variables $q_1(t)$, $q_2(t)$, and $x(t)$ in this case are as shown in Figs 3(a), 3(b), and 3(c), respectively; whilst the controls $i(t)$, $u_1(t)$, and $u_2(t)$ generated by the hybrid controller are as shown in Figs 4(a), 4(b), and 4(c), respectively. This hybrid controller is equally effective if the part demand rate is increased by 50%, such that

$$d(t) = 1.5 \quad (0 \leq t \leq 10), \quad (20)$$

when the system is in the initial state defined by equations (19). Indeed, the state variables $q_1(t)$, $q_2(t)$, and $x(t)$ in this case are as shown in Figs 5(a), 5(b), and 5(c), respectively; whilst the controls $i(t)$, $u_1(t)$, and $u_2(t)$ generated by the hybrid controller are as shown in Figs 6(a), 6(b), and 6(c), respectively.

IV. CONCLUSION

Techniques for the synthesis of optimal control policies for manufacturing systems are frequently rather opaque and often yield control policies that are implemented by open-loop rather than closed-loop control systems. In this paper, it has therefore been shown that closed-loop systems incorporating hybrid fuzzy/crisp-logic controllers can be readily designed for such systems. This hybrid approach has been illustrated by reference to the closed-loop control of a simple manufacturing system producing a single part type.

V. REFERENCES

- [1] S.B. Gershwin, *Manufacturing Systems Engineering*, Prentice-Hall, 1994.
- [2] J.G. Kimemia, and S.B. Gershwin, "An algorithm for the computer control of a flexible manufacturing system", *IIE Transactions*, vol 15, pp 353-362, 1983.
- [3] R. Akella, and P.R. Kumar, "Optimal control of production rates in a failure-prone manufacturing system", *IEEE Trans. Automatic Control*, vol. AC-31, pp. 116-126, 1986.
- [4] T. Bielecki, and P.R. Kumar, "Optimality of zero-inventory policies for unreliable manufacturing systems", *Operations Research*, vol. 36, pp. 533-541, 1988.
- [5] J.B. Sousa, and F.L. Pereira, "A receding horizon strategy for the hierarchical control of manufacturing systems", *Proc. 4th International Conference on Computer Integrated Manufacturing and Automation Technology*, pp 443-450, 1994.
- [6] A. Sharifnia, "Stability and performance of distributed production control methods based on continuous-flow models", *IEEE Trans. Automatic Control*, vol AC-39, pp 725-737, 1994.
- [7] B. Porter, and C. Allaoui, "Genetic synthesis of optimal control policies for manufacturing systems", *Proc. World Automation Congress*, Montpellier, France, May 1996.
- [8] B. Porter, "Genetic design of control systems", *J.SICE*, vol 34, pp 393-402, 1995.
- [9] K. Egilmez, and A. Sharifnia, "Optimal control of a manufacturing system based on a novel continuous-flow model with minimal WIP requirement", *Proc. 4th International Conference on Computer Integrated Manufacturing and Automation Technology*, pp 113-118, 1994.

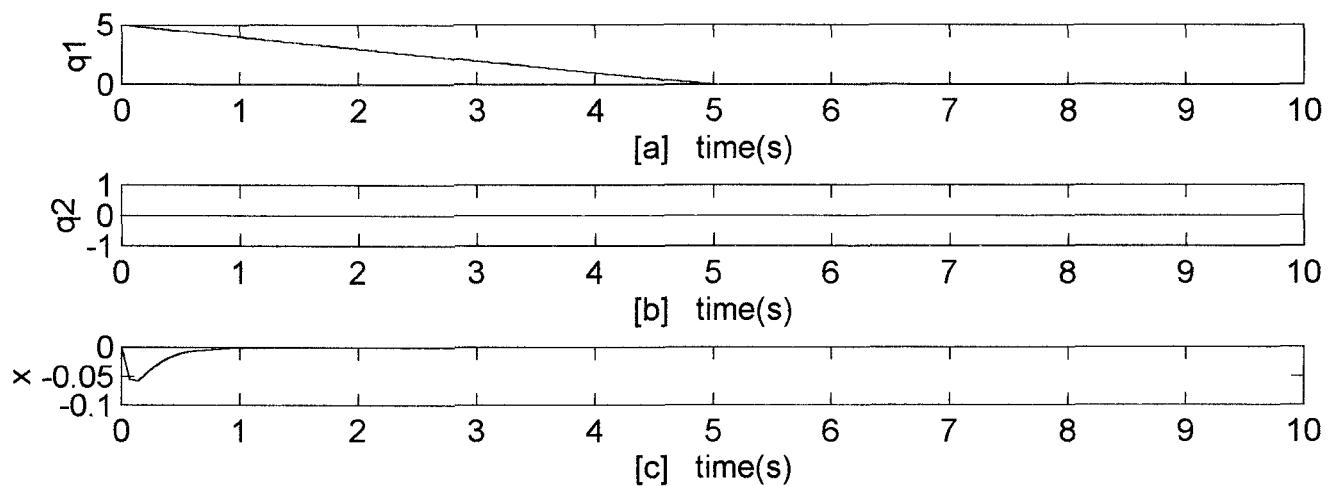


Figure 1: Time-domain behaviour of q_1 , q_2 and x . [$q_1(0)=5$, $q_2(0)=0$, $x(0)=0$, $d(t)=1$]

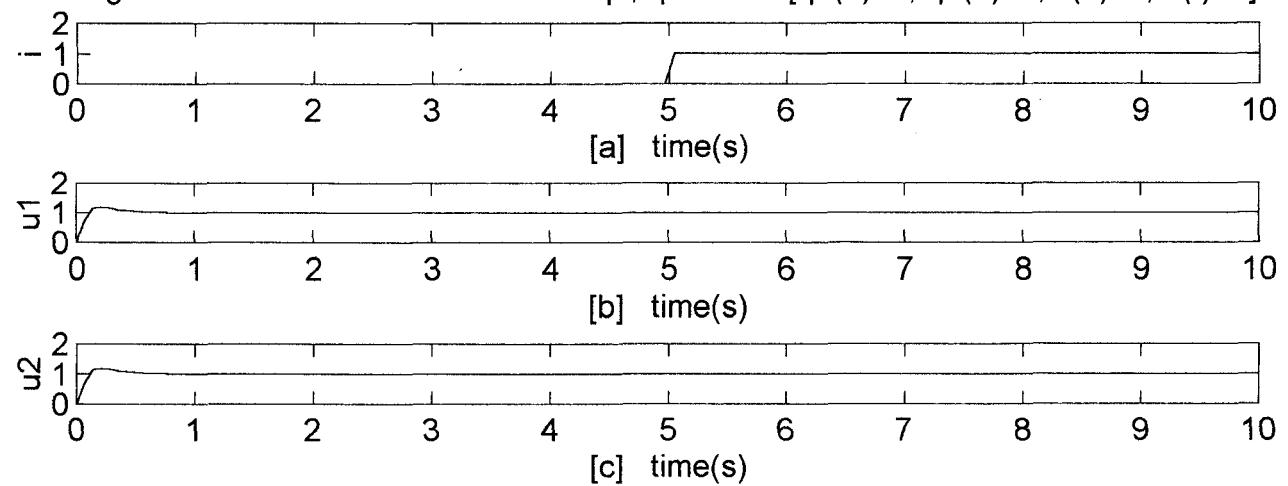


Figure 2: Time-domain behaviour of i , u_1 and u_2 . [$q_1(0)=5$, $q_2(0)=0$, $x(0)=0$, $d(t)=1$]

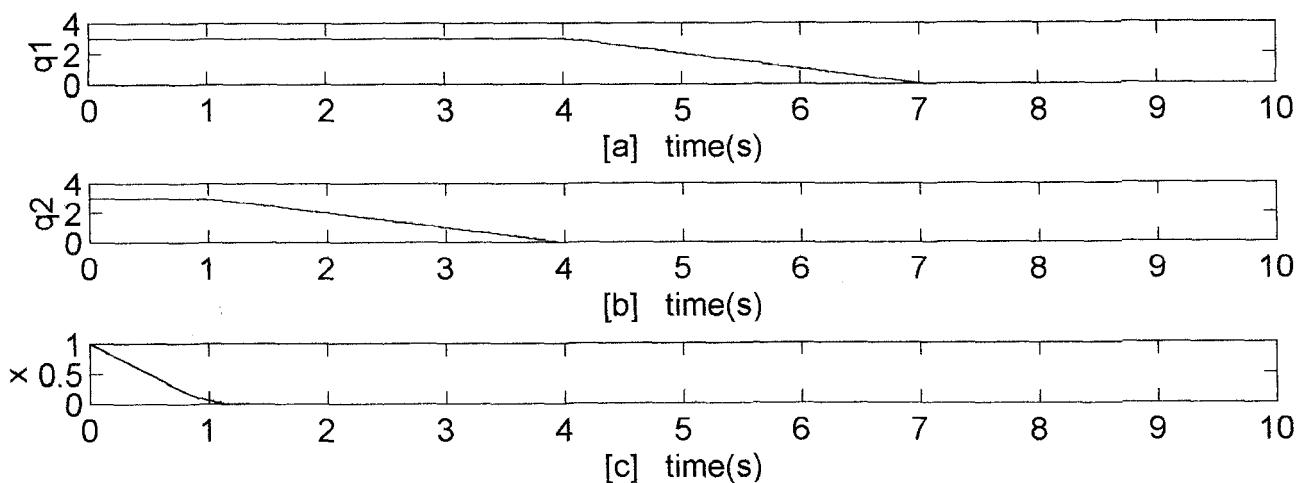


Figure 3: Time-domain behaviour of q_1 , q_2 and x . [$q_1(0)=3$, $q_2(0)=3$, $x(0)=1$, $d(t)=1$]

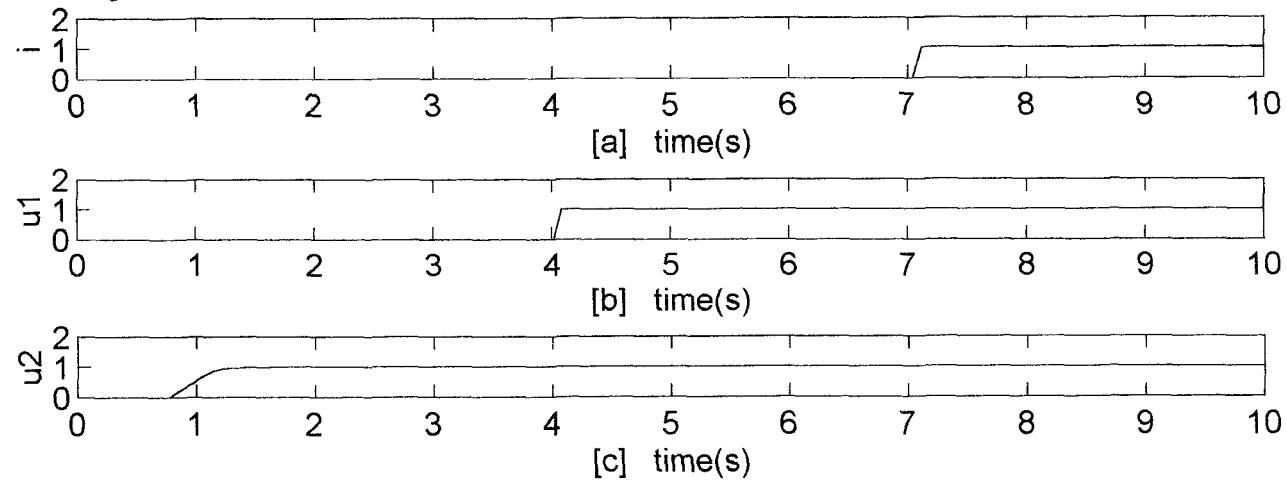


Figure 4: Time-domain behaviour of i , u_1 and u_2 . [$q_1(0)=3$, $q_2(0)=3$, $x(0)=1$, $d(t)=1$]

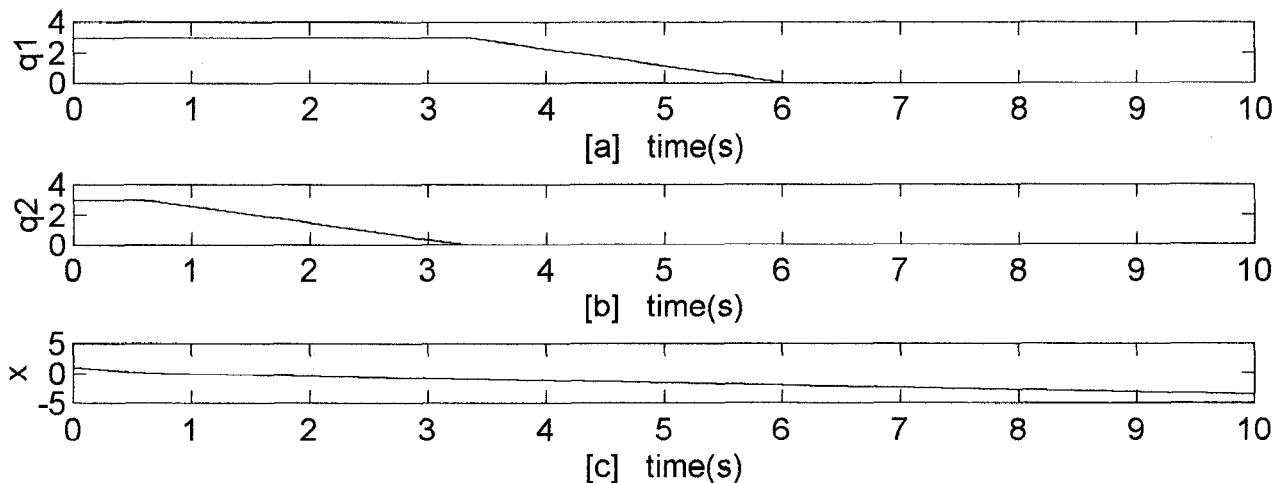


Figure 5: Time-domain behaviour of q_1 , q_2 and x . [$q_1(0)=3$, $q_2(0)=3$, $x(0)=1$, $d(t)=1.5$]

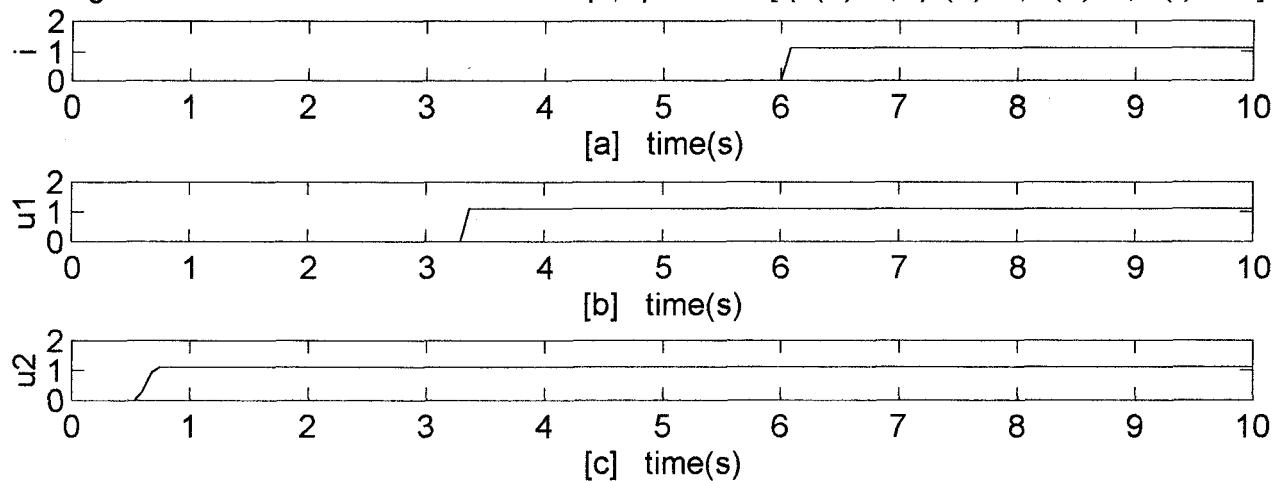


Figure 6: Time-domain behaviour of i , u_1 and u_2 . [$q_1(0)=3$, $q_2(0)=3$, $x(0)=1$, $d(t)=1.5$]