Switching Fuzzy Logic Control for a Reconfigurable System considering Communication Time Delays

Benítez-Pérez H. and García-Nocetti F.+*

**+Departamento de Ingeniería de Sistemas Computacionales y Automatización,*
IIMAS,UNAM, Apdo. Postal 20-726. Del. A.Obregón, México D.F., 01000, México.
Fax: ++52 55 5616 01 76, Tel: (*) ++52 55 5622 36 39
*Email: (*) **hector@uxdea4.iimas.unam.mx***
*(+) **fabian@uxdea4.iimas.unam.mx***

Abstract: Nowadays classical strategies for communication time delays treat them as a stochastic variable. In here, time delays are treated as a nominal value part of control law. This control approach is based upon fuzzy classical approach. Time delay variable is integrated to fuzzy approach as part of inference structure. This new variable is measured from time behaviour of each element with respect to scheduling strategy. Scheduler is based upon a static strategy, where different configuration topologies are define off-line. This work addresses the problem of time variations as nominal value for control parameter reconfiguration.

Keywords: Fuzzy Logic Control, Time Delays, Pattern Recognition, Reconfigurable Control.

1.- Introduction

The emergence of smart sensor and actuator technology removes the need for centralised control with feedback loops to dumb peripheral actuators replacing it with a databus connection (Benítez-Pérez et al., 1998). This gives an autonomous actuator installation (Masten, 1997) as well as local control, self- calibration and health monitoring.

Several strategies for managing time delay within control laws have been studied for different research groups. For instance Nilsson (1998) proposes the use of a time delay scheme integrated to a reconfigurable control strategy based upon a stochastic methodology. On the other hand, Wu (1997) proposes a reconfiguration strategy based upon a performance measure from a parameter estimation fault diagnosis procedure. Another strategy has been proposed by Jiang et al., (1999) where time delays are used as uncertainties, which modify pole placement of a robust control law. Izadi et al., (1999) present an interesting view of fault tolerant control approach related to time delay coupling. Present approach takes time delays due to communication as deterministic measured variables. These measures are normalized from zero to one hundred percent. Where zero means non-time delay presence and hundred percent means worst case scenario from time delay point of view.

In here, control law is viewed as fuzzy logic control taken into account a nominal time delay from a deterministic reconfigurable communication approach. Inference rules has been proposed, firstly, by a try and error approach. Thereafter by a cluster method (Höppnner et al., 2000). In this case, Control law follows a reconfigurable control law based upon communication time delay. As mention before this nominal time delay is based upon a deterministic static scheduling approach. Reconfigurable communication is determined due to fault appearence within certain peripheral elements. Previous revisions are presented by Benítez-Pérez, (1999) and Benítez-Pérez et al., 2001. The purpose of this paper is to study how time delays are integrated to a control law approach rather than reviewing fault issues.

Those time delays are the result of communication performance. Proposed fuzzy control laws are coupled to the system getting a good performance. The evolution of time delays are known and established due to different fault and non-fault scenarios.

As case study, a ball and beam plant is used with two arrays of sensors and two actuators. This paper has been divided in six sections. First section is the current introduction. Second section is an overview of how time delays have been measured. Third section is presents the actual fuzzy control law approach. Fourth section, shows a case study. Finally, fifth section presents concluding remarks.

2.- Time Delays Modelling

This section describes how time delays from communication issues are measured. This distributed system is based upon time stamp relation, a scheduling algorithm is established for each element among the system. This static scheduler is presented by the use of time graphs.

The distributed system is integrated for four different types of elements, sensors, decision making module, controller and actuators. The scheduling approach (Krishna et al., 1997) provides sporadic communications due to the presence of fault scenarios. There are several kinds of time delays to be considered. These are exposed in Tables 1.a and 1.b.

Three scenarios are considered, firstly, fault free scenario. Secondly, fault scenario before control reconfiguration approach. Finally, fault scenario after control reconfiguration approach. Time diagram related to first scenario is shown in Fig. 1. Fig 2 presents second and third scenario respectively. Time delays are determined according to these graphs. Where time variables are explained in Table 1.

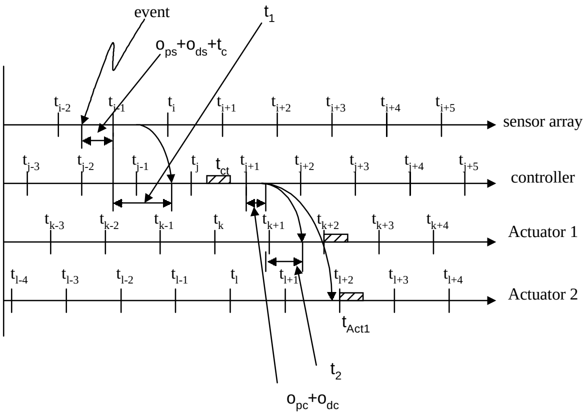

Fig. 1 Scheduler for Fault Free Scenario

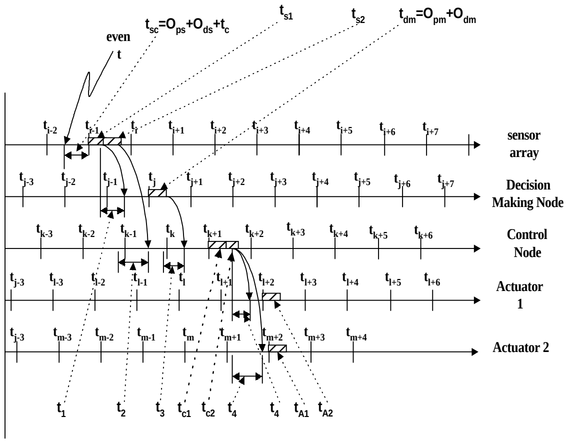

Fig 2 Scheduler for Fault Scenario

These time delays are transmitted from one to another nodes as time stamp of current node information. These results from each node are transmitted as part of the information flow of control system (sensor-control-actuator). Related to time delays, control node may produces either t_{ff} or t_{fsI} based upon this information, furthermore, it gets an estimation of time spent by actuator node and its communication ($\hat{t}_{A1}$). Having obtained these sources of time delay, control produces a global time delay Δt_*. This value is composed of time spent by sensor, communication time spent between sensor and control, time spent by control node. Δt_* has three different values as shown in eqns. 2, 4 and 5. These values depend on the current scenario. This Δt_* value is consider as an extra input for controller.

First scenario is named as fault free scenario. Fig. 1 presents a result of time performance. Total time spent during this scenario is 11.5 milliseconds according to table 1 and eqn. 1.

Var	Name	Time Consume (micro seconds)
c	Communication	450
b	Blocking	50
i	Interference	0
t_c	Capture sensor information	100
O_{ps}	Overhead time from pre-processing sensor information	3000
O_{ds}	Overhead time from post-processing sensor information	3000
t_1	Communication time from sensor node to control node	575
O_{pc}	Overhead of Pre-processing Information from control node	1000
O_{dc}	Overhead of Process Information from control node	1000
t_{ct}	Control Process Time	250
t_2	Communication time from control node to actuator node	575
t_{A1}	Processing time from actuator 1 and 2	2000

Table 1.a Time variable from Fault Free Scenario

Var	Name	Time Consume (micro seconds)
O_{ps}	Overhead time from pre-processing sensor information	1000
O_{ds}	Overhead time from post-processing sensor information	1000
t_c	Capture sensor information	1000
c	Communication	450
b	Blocking	50
i	Interference	0
t_1	Communication time from sensor node to decision making module	575

t_2	Communication time from sensor node to control node	575
t_{s1}	Processing time before sending information	2000
t_{s2}	Processing time before sending information	2000
O_{dm}	Overhead of Pre-processing Information	3000
O_{pm}	Overhead of Process Information	3000
t_{dm}	Processing time before sending information from Decision Making to Controller	1000
$t_{c1}= t_{c2}$	Processing time from control node	1000
t_3	Communication time from Decision making node to control node	575
t_4	Communication time from control node to actuator node	575
t_{A1}	Processing time from actuator 1 and 2	2000

Table 1.b Time variables from Fault Scenario

$$t_{ff}= t_1+o_{ps}+o_{ds}+t_{ct} + o_{pc}+o_{dc}+ t_2+ \hat{t}_{A1} \tag{1}$$

Where t_{ff} is the total time spent during fault free scenario. This time is a measure related to scheduling scheme shown in Fig. 1. Each node has its own internal clock that is synchronized by bizantine clock strategy (Krishna et al., 1997). Time delays related to message communication are determined accordingly to the associated time graph shown before.

Global time delay (Δt_*) is defined from the occurrence of an event until the information reaches control node. Following eqn. 1 actuator its time consumption and time communication are estimated from previous event. Eqn 2 shows this result.

$$t_{ff} - t_{A1} = \Delta t_{ff} \tag{2}$$

Where t_{ff} represents global time spent, t_{A1} represents time delay spent by actuator at fault free scenario and Δt_{ff} represents time delay at fault free scenario. In nominal conditions Δt_{ff} value is zero. For fault scenario I, see Fig. 2, the summation of this graph is as follows

$$t_{fsI}=t_{dm}+t_{s2}+t_{s1}+ t_2+t_3+t_{c2}+t_4+ t_{c1}+ \hat{t}_{A1} +t_1+ t_{sc} \tag{3}$$

This case presents another time delay result due to the appearance of an extra element identified as decision maker module. New communication transactions between sensor and control nodes appear due to this extra element. As a result of this interaction an extra time delay is sum as shown in eqn. 3. As soon as last time delay from actuator node $\hat{t}_{A1}$ is estimated from previous scenario. Final result is equal to equation 4.

$$t_{fsI} - t_{A1} = \Delta t_{fsI} \tag{4}$$

This time delay represents how long control action is taken to be ready before actuator node acts upon the plant. In nominal conditions this value represents 20- 40% from worst case scenario.

For second fault scenario shown in Fig, 2. A similar situation of former case is exposed due to appearance of extra elements. Eqn. 5 shows total time consumed in this scenario.

$$t_{fsII}=t_{dm}+t_{s2}+t_{s1}+ t_2+t_3+t_{c2}+t_4+ t_{c1}+ \hat{t}_{A1} +t_1+ t_{sc} \tag{5}$$

This third scenario is shown as

$$t_{fsII} - t_{A1} = \Delta t_{fsII} \tag{6}$$

Although t_{fsII} and t_{fsI} are similar in nominal terms, it is expected to be modified due to fault conditions. Nevertheless, the differences between scenarios are not explored in this paper. As result of these three scenarios three time delays are obtained. For the case of this simulation, CANbus standard is used to establish the communication between elements and clock synchronisation is time stamping over each communication process. Time delays related to sensing, controlling and processing information into the actuator are based upon the response of the dynamics tested in a PIII computer. The implementation of this scheduler is based upon State-Flow toolbox from Matlab 5.3 (MATLAB, 1998).

3.- Fuzzy Logic Control Approach

In order to integrate this variable established as global time delay (Δt_*), firstly it is necessary to consider its nominal value. That means a percentage between 0 to 100 %. Zero percent is a fault free scenario. 100 percent represents fault scenario of case II. This Δt_* value is any value produced as time delay less than or equal to that explained in eqn. 6.

Having defined global time delays as its nominal value. Fuzzy control structure is proposed as Fig. 3.

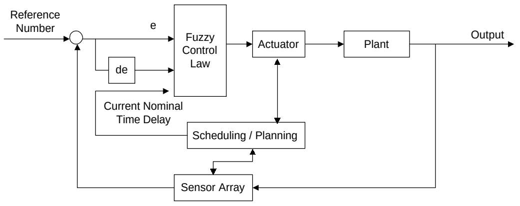

Fig. 3. Fuzzy Control Law.

Fuzzy control has been chosen rather than gain-scheduler controller and smith's predictor because it has a smooth transition between scenarios. Furthermore, the chosen operating points are the reference elements of proposed fuzzy control. Thus, any degradation from time delays would degrade control law but the plant keeps a stable response. Time delay degradation is bounded from communication protocol as explained by Lian et al., (2002).

Current approach follows Mamdani strategy rather than Takagi Sugeno (TKS) proposal. Further on TKS is focused into future work pursued as the integration of time delays into subsequent part of fuzzy rules.

The actual structure of this controller for fault free scenario is proposed in Fig. 4. This is based upon Driankov et al., (1994). Membership functions are gaussian bells, where *e* variable has six membership functions (PB, PM, PS, NS, NM, NB), *de* has 6 membership functions (PB, PM, PS, NS, NM, NB). The output variable has eight membership functions (PB, PM, PS, PZ, NZ, NS, NM, NB). Additional variable named Current Nominal Time Delay (CNTD) has three membership functions (N, Z, P).

	NB	NM	NS	PS	PM	PB
PB	PZ NZ	NZ	NM	NM	NB	NB
PM	PS	PZ NZ	NS	NM	NM	NB
PS	PM	PS	PZ NZ	NS	NM	NM
NS	PM	PM	PS	PZ NZ	NS	NM
NM	PB	PM	PM	PS	PZ NZ	NS
NB	PB	PB	PM	PM	PS	PZ NZ

de

e

Fig. 4. Classical Structure for Fuzzy Control Law

This implementation is a common approach for fuzzy control. For the case of second and third scenarios, Fig. 5 shows actual implementation.

	NB	NM	NS	PS	PM	PB
PB	PZ NZ	NZ	NM	NM	NB	NB
PM	PS	*	NS	NM	**	NB
PS	PM	PS	PZ NZ	NS	NM	NM
NS	PM	PM	PS	PZ NZ	NS	NM
NM	PB	***	PM	PS	****	NS
NB	PB	PB	PM	PM	PS	PZ NZ

de

e

*	PZ NZ PM NM	**	NM NB NS
***	PM PB PS	****	PZ NZ PM NM

Fig. 5. Modification for Fuzzy Control Law

Fig. 5 (Fault Scenario II) shows different possibilities at the same condition. This case is proposed due to the possible situation that may be presented at next stage. This is at 100 percent time delay. For instance, condition *de* is NM and *e* is PM has a result NZ, PZ for fault free scenario (Fig. 4). However, Fig. 5 presents same scenario with four possible solutions NM, PM, NZ and PZ. This is the result of considering where *e* and *de* suppose to be with 100 percent delayed. In this case every new state in terms of fuzzy control is considered as equally possible.

Both control laws have been established firstly from try and error approach, afterwards, the use of a classical cluster technique such as fuzzy C-Means is used in order to validate both control laws (Höppnner et al., 2000). The results are similar to those presented in Figs. 3 and 4.

In here consequent membership functions are selected due to expected variations for current time delays. As mention before these combination have been selected by using a common clustering technique where these fuzzy rules are the most common during clustering training.

4. Case Study

The strategy followed in this paper is based upon Fig. 7.

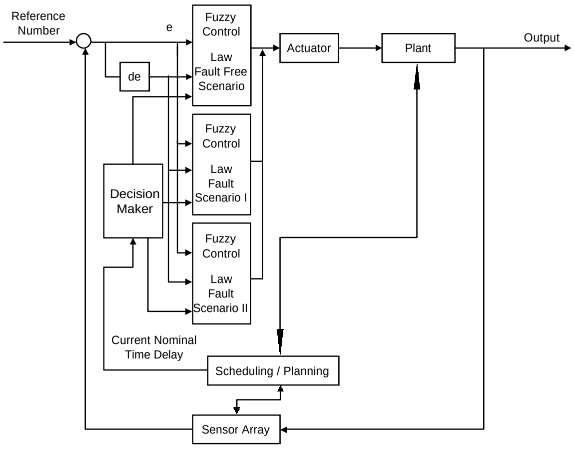

Fig. 7 Reconfigurable Control Scheme

The case study is based upon a ball and beam example as shown in Fig. 8.

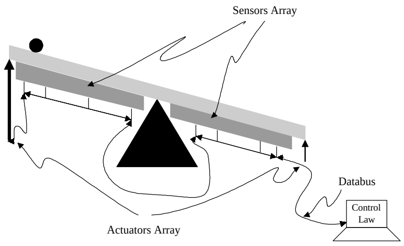

Fig. 8 Plant Scheme

There are two arrays of sensors and two actuators. Each of them on each side of the beam. The model of the plant uses one sensor who is reporting the actual position of the ball and one actuator who is moving the beam.

The plant dynamics is shown next:

$$
\begin{aligned}
\dot{X} &= AX + Bu(t) \\
y &= CX
\end{aligned}
\tag{7}
$$

$$
A = \begin{vmatrix} 0.9956 & -.482 & -.2097 & .7928 \\ 0.0163 & .9741 & 1.0572 & -.6585 \\ .0001 & -.0203 & .8762 & -1.5178 \\ 0.0 & 0.0004 & 0.0192 & 0.6868 \end{vmatrix}
$$

$$
B = \begin{vmatrix} -.0172 \\ .015 \\ .0042 \\ -.0003 \end{vmatrix}
$$

$$
C = \begin{vmatrix} -0.3706 & -0.566 & .5668 & .419 \end{vmatrix}
$$

Where *y* is the output, *u* is the input, *A*, *B* and *C* are the representative polynomials of the system.

Sensors and actuators models are considered lineal by the inherent self calibration within these elements. Time delays produced by the communication scheduler are considered as part of the dynamic behaviour.

The only considered fault scenario is the degradation of any of these peripheral elements per time. Just one sensor is considered to be faulty. The condition of the fault scenario is noise over one peripheral element. There are three main possible scenarios, fault free, fault non-catastrophic and catastrophic.

Further revision of fault scenarios is pursued by Benítez-Pérez et al., (1999). The decision maker module shown in Fig. 7 is similar to that proposed by Yu et al. 2001. This is based upon eqn. 8 where e defines the error between two control node variables α and β which are constants defined as $\alpha = 0.92$ and $\beta = 0.08$.

$$\delta = \alpha e^2 + \beta \int_0^t e^2 dt \tag{8}$$

Where δ is normalized between 0 and 1. This measure switches from one control law to another based upon the use of a threshold.

The main advantage of this control approach is the avoidance of glitches during control law transitions. Unlike similar implementations (Benítez-Pérez et, al., 2001), current method eliminates undesirable glitches between fault scenarios.

The output of the system are the angle of the beam and position of the ball. The plant has been linearized to one operation point nearby the centre of the beam giving eqn. 7.

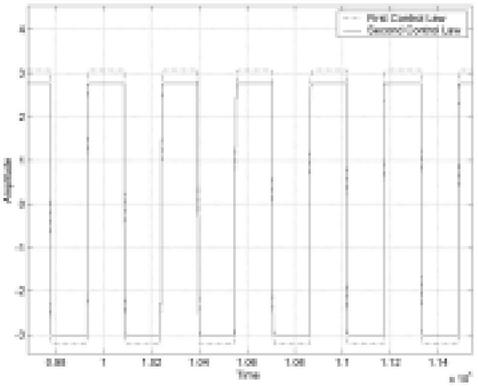

Fig. 9 Control Laws Response during a Fault Free Scenario

Results presented in Fig. 9 show how proposed approach is suitable as reconfiguration strategy in comparison to other proposed techniques such as the classical technique used in previous work (Benítez-Pérez et, al., 2001). In this case, first control law (depicted in Fig. 4) is presented as dashed line signal. Second control law (depicted in Fig. 5) is presented as solid line. The response of both control laws is according to fault free scenario. Although, there is a small difference between amplitudes, this is neglected. The amplitude of both signals is based upon the response of the system to train pulse input as target.

Next two Figs., present the response of both control laws during two different fault scenarios. First fault scenario presents a total time delay of 15.61 ms (Fig. 10). Second fault scenario presents a total time delay of 60 ms (Fig. 11). Although the difference is significant, both control laws keep the tracking of the system.

Fig. 10 shows a small degradation from first control law due to time delay presence. This degradation is shown in Fig. 11.

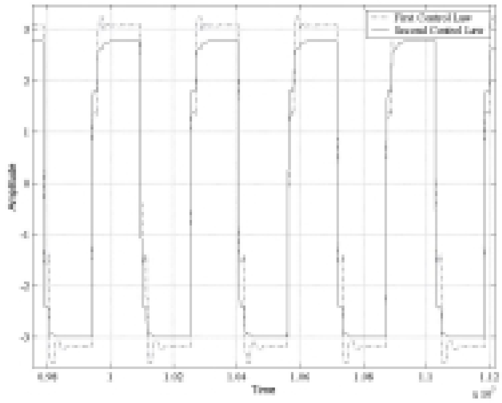

Fig. 10 Control Laws Response considering a Small Total Time Delay

As mentioned in previous sections both scenarios are the result of fault condition within the system. Nevertheless, these faults are not in the scope of this work.

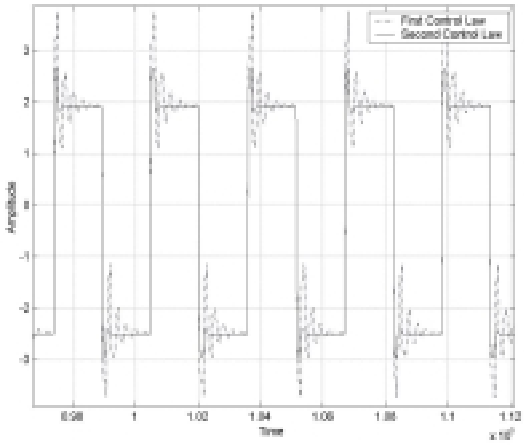

Fig. 11 Control Laws Response considering a Large Total Time Delay

As the reader may observe, Fig. 11 shows first control law has a major degradation in comparison to second control law. This degradation is observed in Fig. 10 with less impact into the system dynamics.

The scope of this result presents performance of proposed control law as well as the switch approach. This last technique shows a good solution of switching strategies.

5. Concluding Remarks

This research work has shown the implementation of an on-line control reconfiguration approach considering a deterministic time delay among a distributed system. This reconfiguration its originated due to local faults presence which are not within the scope of this paper, The idea of measurements in terms of deterministic time delays is the reflection of this fault scenario. Furthermore, control strategy based upon fuzzy control law has been proposed from two different strategies. First is for fault free scenario. Second strategy is use for fault scenarios cases I and II respectively.

Fuzzy clustering is a feasible approach to study suitable scenarios for deterministic time delay integration. This strategy has been used in order to validate the proposed response. Fuzzy C-Means is used in order to generate a suitable fuzzy system based upon the desire system response.

Although this whole control approach may be seen as reconfigurable control, it is not the purpose of this paper to explore this issue. The main contribution of this paper is focused into time delay measurement incorporated to fuzzy logic control as a deterministic value.

Results presented in this paper are classified as preliminary. These are based upon a case study simulation.

Further research is proposed in order to explore the control law as reconfigurable approach based upon time delay variation as uncertainties. Besides, stability analysis is compulsory in order to formalize this strategy.

Acknowledgements
The authors would like to thank the financial support of DISCA-IIMAS-UNAM, and UNAM-PAPIIT (IN106100) Mexico in connection with this work.

References:

Benítez-Pérez, H., and García-Nocetti, F.; “Reconfigurable Distributed Control based upon Smart Peripheral Elements”; European Control Conference; ECC01, Porto, Portugal, pp. 3534-3539, 2001.

Benítez-Pérez, H., Thompson, H. A. and Fleming, P. J.; “Simulation of Distributed Fault Tolerant Heterogeneous Architectures for Real-Time Control”; 5th IFAC Workshop on Algorithms and Architectures for Real-Time Control AARTC’98, pp. 89-94, Cancún, México, 1998.

Benítez-Pérez, H.; “Smart Distributed Systems”; PhD. Thesis AC&SE Dept. University of Sheffield, UK, 1999.

Driankov, D., Hellendoorn, H., Reinfrank, M.; “ An Introduction to Fuzzy Logic Control”; Springer-Verlag, 1994.

Höppner F., Klawonn F., Kruse R. and Runkler T.; “Fuzzy Cluster Analysis”; J. Wiley, Fist Edition, USA, 2000.

Izadi-Zamanabadi R. and Blanke M.; “A Ship Propulsion System as a Benchmark for Fault-Tolerant Control”; Control Engineering Practice, Vol. 7, pp. 227-239, 1999.

Jiang J., and Zhao Q.; “Reconfigurable Control Based on Imprecise Fault Identification”; Proocedings of the American Control Conference, IEEE, pp. 114-118, San Diego, June, 1999.

Krishna C. M. and Shin K. G.; “Real-Time Systems”; Mc. Graw Hill, New York, USA, 1997.

Lian F. Moyne J. and Tilbury D. ; “Network Design Consideration for Distributed Control Systems”; IEEE Transactions on Control Systems Technology, Vol. 10, No. 2, pp. 297-307, March 2002.

Masten, M. K.; “Electronics: The intelligence in Intelligent Control”; IFAC Symposium on Intelligent Components and Instrument for Control Applications, Annecy, France, pp. 1-11, 1997.

Mathworks; “System Identification Toolbox User’s Guide”; MATLAB, 1998.

Nilsson, J.; “Real-Time Control with Delays”; PhD. Thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, 1998.

Wu N.; “Reliability of Reconfigurable Control Systems: A Fuzzy Set Theoretic Perspective”; Proceedings of the 36th Conference on Decision & Control, IEEE, TP15 5:10, pp. 3352-3356, San-Diego, USA, 1997.

Yu W. and Pineda J.; “Chemical Process Modeling with Multiple Neural Networks”; European Control Conference, Porto, Portugal, pp. 3735-3740, 2001.