Reconfigurable Network Control using Fuzzy Logic for Magnetic Levitation Case Study

Benítez-Pérez H.*, Quiñones-Reyes P.++, Mendez-Monroy E.+,
Cardenas-Flores F.+ and García-Nocetti, F.+

++Instituto Tecnológico de Jiquilpan, Av. Tecnológico, S/N, CP 59510, Jiquilpan, Michoacán, México.
*+Departamento de Ingeniería de Sistemas Computacionales y Automatización, IIMAS, UNAM, Apdo. Postal 20-726, Del. A. Obregón, México D. F., CP. 01000, México.
Fax: ++52 55 5616 01 76, Tel: (*) ++52 55 5622 36 39
Email: (*) hector@uxdea4.iimas.unam.mx

Abstract: Nowadays dynamic behaviour performed by a computer network system shows the needs to address it from the perspective of a control system. One strategy to be followed it is the real-time modeling. Having obtained a way to represent a computer network system, next stage is to how control approach can be affected and modified. In that respect, this paper proposes a control reconfiguration strategy from the definition of an automata considering computer network reconfiguration. Several stages are studied, how computer network takes place as well as how control techniques is modified.

1. Introduction

Control reconfiguration is presented as an available approach for fault coverage in order to keep system performance. In here reconfiguration is pursued as response of time delay modification modification rather than fault appearance although this is the basic for control reconfiguration.

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. Reconfigurable control has been studied from the point of view of structural modification since fault appearance as presented by Blanke et al., (2003) where a logical relation between dynamic variables and faults are established. From the point of view that reconfigurable control performs a combined modification of system structure and dynamic response as studied by Benítez-Pérez et al., (2005a), Benítez-Pérez et al., (2005b) and Thompson (2004). It presents the advantage of bounded modifications over system response.

Some considerations need to be stated in order to define this approach. Firstly, faults are strictly local in peripheral elements and these are tackled by just eliminating the faulty element. In fact, faults are catastrophic and local. Time delays are bounded and restrictive to scheduling algorithms. Global stability can be reached by using classical control strategy, several fuzzy control strategies for online time delays.

The objective of this paper is to present a strategy for control reconfiguration based upon time delay knowledge as well as local fault effects within a distributed system environment considering the magnetic levitation challenge.

2. Structural Reconfiguration Algorithm

This paper is placed as a strategy for reconfigurable systems as shown in Fig. 1. In fact, this paper is focused into reconfigurable control law due to the presence of local faults and time delays as consequences. Time delays are measurable and bounded according to a real-time scheduling algorithm. In this case the scheduling algorithm is the well known earliest deadline first (EDF) algorithm. According to Fig. 1, structural reconfiguration takes place as result of EDF (using a ART2A neural network) performance and related user request. This action provokes a control law modification. How this is modified is the scope of this paper by using several Mamdani approaches.

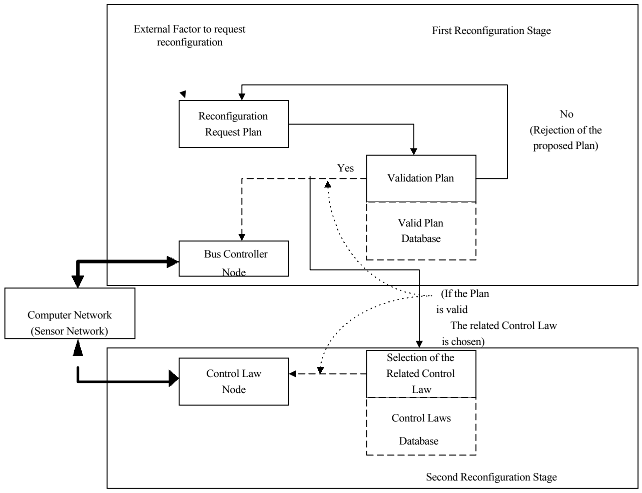

Fig. 1. General structure of Reconfigurable System over a Computer Network

The core of this algorithm is to perform on-line reconfiguration based upon a review of the proposed plan. The review uses a ART2 neural network in order to classified valid and non-valid plans. First, the ART2 neural network is trained offline using valid and non-valid plans from EDF evaluation and case study response. Based on this training procedure two main regions are determined, one related to suitable reconfigurations and other that holds non-trustable reconfigurations. During online stage ART2 network allows classification from new plans. If the response of the network belongs to valid plans it will be reconfigured, otherwise the proposed plan will be rejected. The current approach is based on two stages as shown in Fig. 2.

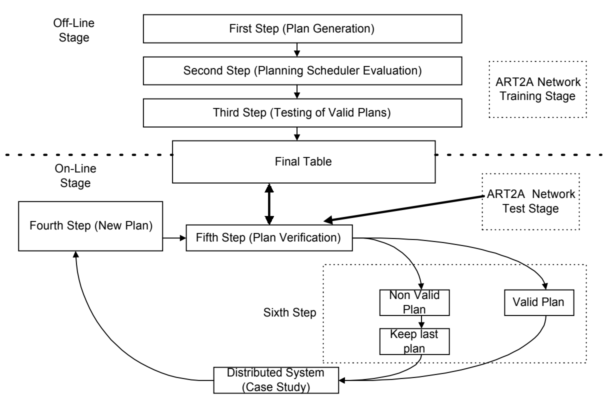

Fig. 2. Algorithm Proposal

It is important to mention that ART2 network cannot learn new plans during online stage as safety precaution.
The communication network plays a key role in order to define the behaviour of the dynamic system in terms of time variance giving a nonlinear behaviour. In order to understand such a nonlinear behaviour, time delays are incorporated by the use of real-time system theory that allows time delays to be bounded even in the case of causal modifications due to external effects. Several algorithms can be pursued such as Rate Monotonic, Deadline Monotonic or Earliest Deadline First ((Lian et al., (2002), Benítez-Pérez et al., (2003)), flexible time triggered (FTT) ((Almeida *et al.*, 2002), and Liu et al., (1973)). The use of EDF is pursued in here due to flexibility of task reorganization during online performance. The main difference is that the static scheduler defines during the off-line process the allocation of task, whereas the dynamic scheduler allocates tasks based on current conditions considering a time slot.
Basic procedure of EDF requires several characteristics from each task such as deadlines, consumption times and priorities. The difference between them is marked by the way tasks are ordered. It depends on the application which method for ordering tasks is the most suitable for a particular example. Those algorithms already mentioned are divided into two categories as static and dynamic schedulers.
For real-time purposes, it is best to use static schedulers because of its deterministic behavior. Recently, quasi-dynamic scheduling algorithms (such as FTT) have been defined to give certain flexibility to the static communication approach.
Earliest Deadline First defines task ejecution based on the proximity of each local deadline named as the difference between current time as local deadline. The smallest value amongst all tasks is the winner. For instance consider a group of three tasks with time distribution as shown in Fig. 3.

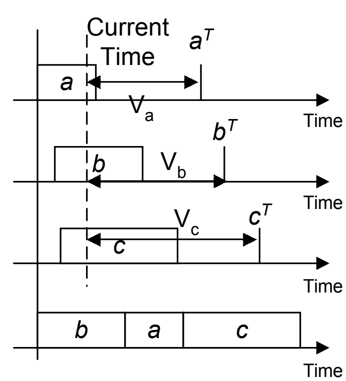

Fig. 3. EDF approximation

Where

$$\text{Current.time} - a^T = V_a$$
$$\text{Current.time} - b^T = V_b$$
$$\text{Current.time} - c^T = V_c$$
$$\min(V_*) = V_w$$

V_w is the task to be executed.

An ART2 is a self-organizing neural network based upon the angle between a prototype and input pattern to find a fitting cluster. This network has been implemented following the approach presented by Frank et, al. (1998). This scheme is shown in Fig. 4. The idea is to identify already classified fault patterns and categorize new faults based upon the classification of new patterns. The use of a new group of patterns does not overcome the identification of the physical meaning of the new classified fault. This work still should be performed off-line by the designer.

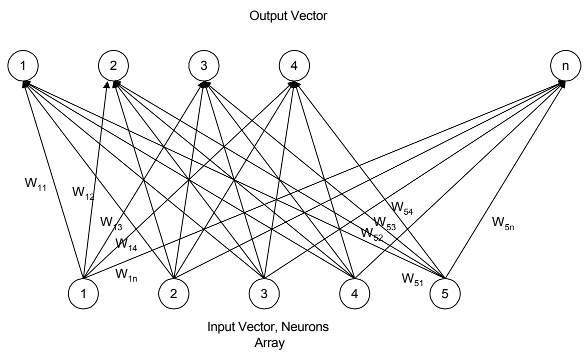

Fig. 4 Typical ART2 Network

3. Case Study and Control Reconfiguration Approach

Case study is a magnetic system integrated to a computer network as shown in Fig. 5 (Wincon, 2003).

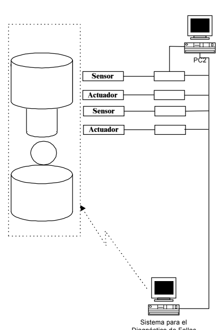

Fig. 5. Magnetic Levitator Case Study

The dynamics of case study expressed in transfer function is

$$G_{bi}(s) = \frac{k_{bdc} w_b^2}{s^2 + w_b^2}$$

$$k_{bdc} = \frac{x_{co}}{I_{co}} \qquad (1)$$

$$w_b = \sqrt{2}\sqrt{\frac{g}{x_{bo}}}$$

where

g is the gravity force

I_{co} is the current of the coil

x_{bo} is the distance from coil to ball position.

Following this implementation basic time diagram system is proposed as Figs. 6 and 7. When a fault appears the use of EDF through ART2A network is performed in order to re-organize task execution according to basic time restrictions shown in Figs. 6 and 7. Maximum time delays are bounded on these Figures.

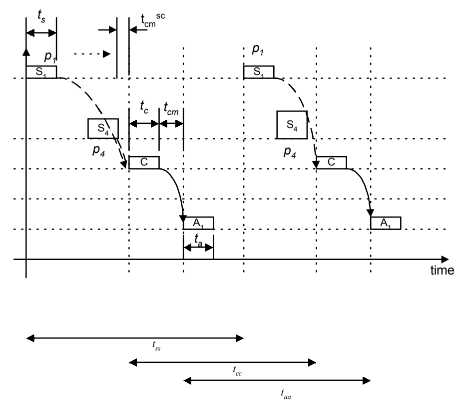

Figure 6. Fault-free scenario

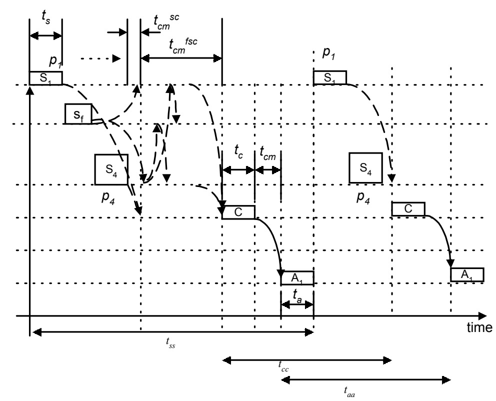

Figure 7. Fault scenario considering fault masking

Both scenarios are local with respect to magnetic levitation. As these two scenarios are bounded, the respective consumption times are shown in Equations 2 and 3 (Figures 6 and 7, respectively) based on Table 2, where variable information is presented. For fault free scneario, time diagram is expressed as:

$$tt = t_s * 4 + t_{cm}^{sft} + t_{ft} + t_{cm}^{sc} + t_c + t_{cm} + t_a \qquad (2)$$

where:

t_s is the consumed time by sensors

t_{cm}^{sc} is the consumed time by communication between sensor and control

t_{cm}^{sft} is the communication time between sensor and fault tolerance module.

t_{ft} is the consumption time from fault tolerance module

t_c is the consumed time by control node

t_{cm} is the consumed time by communication between controller and actuator

t_a is the consumed time by actuator

For fault scenario time diagram is epxressed as:

$$tt = t_s * 4 + t_{cm}^{sc} + t_{cm}^{ft} + t_{ft} + t_{cm}^{ft} + t_c + t_{cm} + t_a \quad (3)$$

where:

t_{cm}^{sc} is the assumed time by communication between sensor and control

t_{cm}^{fsc} is the time consumed for the fault sensor to send messages to its neighbor and produce agreement

t_c is the consumed time by control node

t_{cm}^{sft} is the communication time between sensor and fault tolerance module.

t_{ft} is the consumption time from fault tolerance module

t_{cm} is the consumed time by communication between controller and actuator

t_a is the consumed time by actuator

From both scenarios there is an element known as fault tolerance element that present extra communication for control performance although it mask any local fault from sensors.

From this time boundary, including both scenarios, it is feasible to implement some control strategies. A remarkable issue is related to a particular sensor fault related. Considering this configuration two cases are:

- One local fault;
- Several local faults.

Based on these two possible configurations, there is a worst-case scenario related to several local faults that has an impact on the global control strategy. The other configuration present a minor degradation for the global control strategy. Despite this performance degradation, the system keeps normal functionality due to the inherent fault tolerance strategy and the local controllers.

Taking into account these two possible configurations, the local and global time delays are described in Table 1.

Table 1. Time delays related to local Communications

Configuration 1 One Local Fault	Local Time Delays	0.110 ms
	Global Time Delays	0.110 ms
Configuration 2 Several Local Faults	Local Time Delays	0.110 ms
	Global Time Delays	0.220 ms

As the time delays have been bounded, the plant model is defined based on Figure 8.

From these cases automation approach is used in order to switch from one controller to another as shon in Fig. 11.

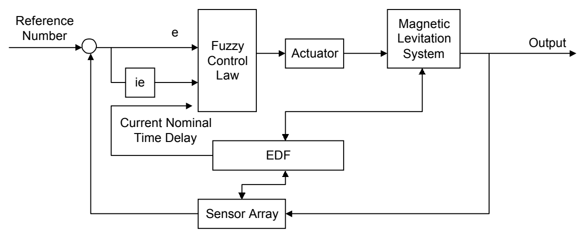

Fig. 8. Fuzzy Control Law

Fuzzy control has been chosen rather than gain-scheduler controller and Smith's predictor because it has a smooth transition between scenarios. 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.

The actual structure of this controller for fault free scenario is proposed in Fig. 9. 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), *ie (Integral of the error)* 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). Stability issue is not pursued in this paper. The interested reader may consult (Nguyen et al., 2003).

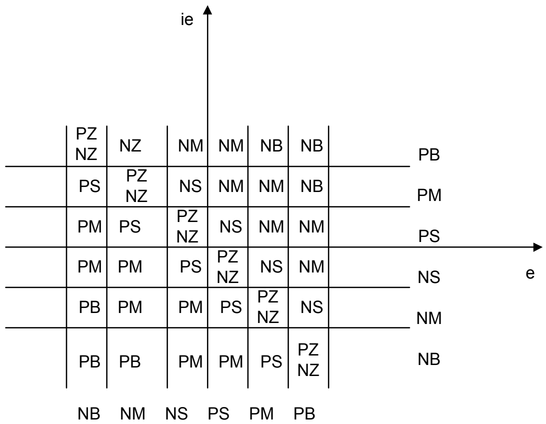

Fig. 9. Classical Structure for Fuzzy Control Law

This implementation is a common approach for fuzzy control. For the case of second scenario, Table 2 shows actual implementation.

TABLE 2. MODIFICATION FOR FUZZY CONTROL LAW

u_c		ierror				
		VHP	HP	M	LP	VLP
error	HP	VHP	VHP	VHP	HP	M
	LP	VHP	VHP	HP	M	LP
	Z	VHP	HP	M	LP	VLP
	LN	HP	M	LP	VLP	VLP
	HN	M	LP	VLP	VLP	VLP

Where

- VHP Very High Positive
- HP High Positive
- M Medium
- LP Low Positive
- VLP Very Low Positive
- Z Cero
- LN Low Negative
- HN High Negative
- VHN Very High Negative

Table 2 shows different possibilities at the same condition. This case is proposed due to the possible situation that may be presented at next stage. Current control Surface is presented in Fig. 10.

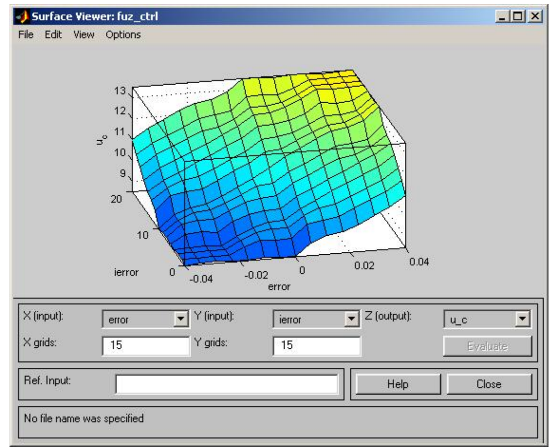

Fig. 10 Surface from Fuzzy control for fault Scenario

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 (Höpnner et al., 2000). The results are similar to those presented in Fig. 9 and Table 2.

The strategy followed in terms of control law reconfiguration is based upon Fig. 11. As mention in section 3, decision making module depends on neural Network.

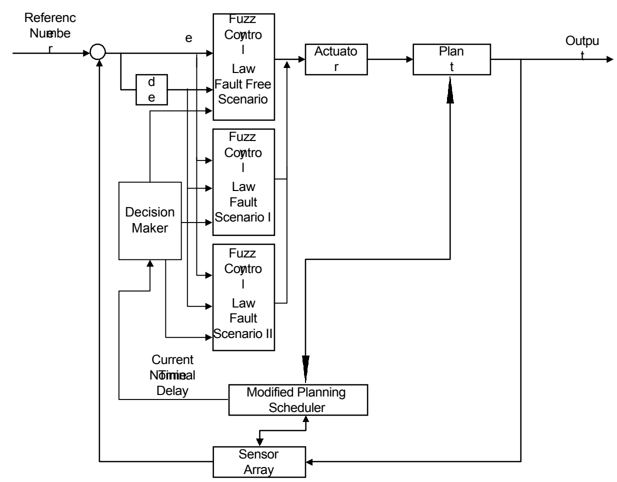

Fig. 11. Reconfigurable Control Scheme

4. Results

From this implementation several results are presented in terms of fault presence and the related action to overcome system lack of performance. How the system responds to these control strategies is presented in the following graphics taking into account fault-free, one local fault, respectively (Figs. 12, 13 and 14).

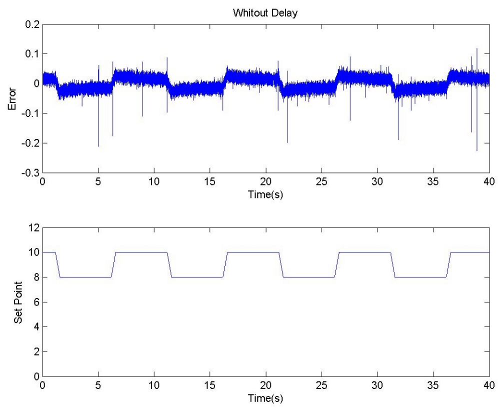

Fig. 12. Set point and Error from Case Study during Fault Free Scenario

Fault scenario presents two different time delays such as in Figs. 13 and 14. For both cases systems still performs feasible results as shown in Error response.

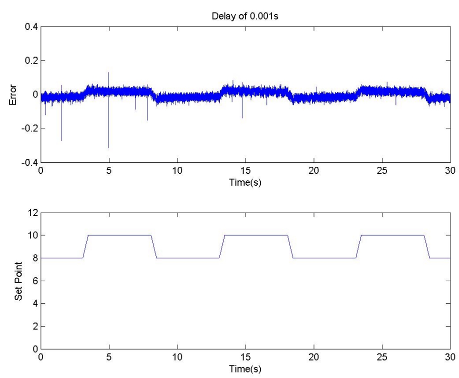

Fig. 13. Set point and Error from Case Study during Fault Scenario considering 1ms

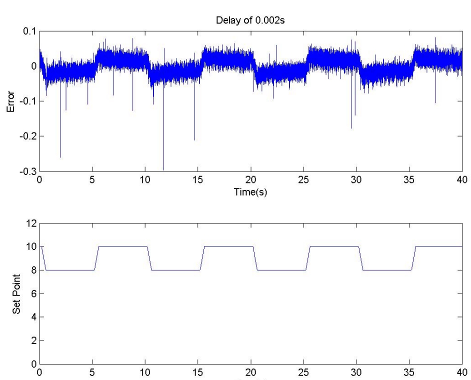

Fig. 14. Set point and Error from Case Study during Fault Scenario considering 2ms

This last example presents two different control cases in which control reconfiguration is based on the decision-maker module following decision ART2 network. Which is simple because it is dependent on the fault presence and on the related time delays. This reconfiguration approach becomes feasible due to the knowledge of fault presence and the consequence of time delays. Its consumption time is neglected, and it is considered part of control performance. It is obvious that fault presence is measurable; if this local fault localization approach cannot detect faults, this strategy becomes useless. Alternatively, local time delay management refers to the use of a quasi-dynamic scheduler to propose dynamic reconfiguration based on current system behavior rather than on predefined scenarios.

In this case, fault and fault-free scenarios are the same as in the first approach; however, in this case, these belong to the scheduler strategy that is performed on-line. The selected scheduler strategy is performed on-line. The selected scheduler algorithm is a modification of earliest deadline first (EDF) to define fixed nonpreemptive tasks like controllers and actuators. For both tasks, time behavior is defined by their necessities. Taking into account these assumptions, the scheduler performs task reorganization based on their consumption times and fault presence. In this case, the followed algorithm is EDF implementation based upon ART2 network.

First nonpreemptive tasks are considered $\{c_1 \dots c_p\}$, and $\{p_1 \dots p_p\}$ where $p \le n$ and n is the total number of tasks. The rest of the tasks are considered faulty elements, masking elements, and neighbor elements $\{c_{p+1} \dots c_n\}$. From this last group of tasks, there is one condition related to one inherent communication among faulty elements, fault masking modules, and neighbor elements. The time window spent (Δt) for reconfiguration needs to obtain the sensor fault's response evaluation and time performance evaluation from the same elements; with this information, the EDF implementation is performed to determine system configuration (Fig. 15).

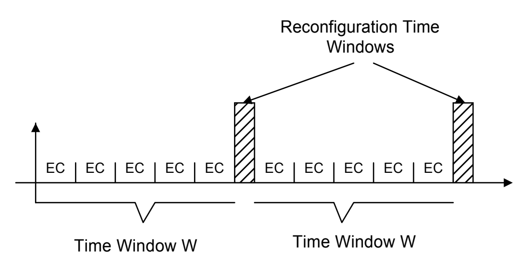

Fig. 15. Time Window Reconfiguration

Furthermore, during this time window, control law strategies are chosen based on current time and sensor demands; here, control laws may be defined in a similar fashion to that exposed before. However, the decision-maker module becomes basic in order to define a precise current global control law. Time delays are encapsulated and expressed as

$$t_{sc}^{f} = t_{cm}^{sc} + t_{FMM} + \Delta t \quad (4)$$

where t_{sc}^{f} is the time delay as a result of the EDF algorithm. Local control laws are expressed in the same form as expressed before, with the only modification that of time delay, which is t_{sc}^{f}; at this moment, reconfigurable control is expressed following the same form as the global control law event structure.

To define the communication network performance, the use of the **True-Time** is pursued. This strategy achieves network simulation based on message transactions that are based on the real-time toolbox from MATLAB. Extended information from this tool is available at (Cervin et al., 2003); the true time main characteristics are shown next. In the **True-Time** model, computer and network blocks are introduced in Figure 16.

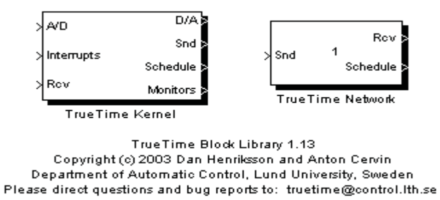

Figure 16. Basic model of true time

These blocks are event driven, and the scheduling algorithm is managed by the user independently of each computer block. True time simulation blocks are basically two blocks. Each kernel represents the interface between the actual dynamical model and the network simulation. Here, continuous simulation and digital conversion take place to transmit information through simulated network. This tool provides the necessary interruptions to simulate delay propagation as well as synchronization within the network.

5. Concluding Remarks

Present approach shows the integration of two techniques in order to perform reconfiguration. These two approaches are followed, in cascade mode, structural reconfiguration and control reconfiguration. Although there is no formal verification in order to follow this sequence, it has been adopted since structural reconfiguration provides settle conditions for control reconfiguration. The use of a real-time scheduling algorithm in order to approve or disapprove modifications on computer network behaviour allows time delays bounding during a specific time window. This local time delay bounding allows the design of a control law capable to cope with these new conditions. Preliminary results show that control reconfiguration is feasible as long as the use of a switching technique predetermines which control is the adequate. This goal is reached by a strategy compose of two algorithms, one which is responsible for structural reconfiguration and it has been implemented in this paper as ART2 network. The second algorithm is responsible for fuzzy control reconfiguration and it is based on an automaton technique. In this case switching control is perform through neural netork. What it is important for this last approach is that control conditions are strictly bounded to certain response.

Future work is focused to produce certain evaluation metrics that allows feasible comparison between different approaches. Fuzzy logic Takagi Sugeno approach need to be addressed.

ACKNOWLEDGEMENTS

The authors would like to thank the financial support of DISCA-IIMAS-UNAM, and UNAM-PAPIIT (IN106100 and IN105303) Mexico in connection with this work.

References

Almeida L., Pedreiras P., and Fonseca J. A.; "The FTT-CAN Protocol: Why and How"; IEEE Transactions on Industrial Electronics, Vol. 49, No. 6, pp. 1189-1201, 2002.

Benítez-Pérez, H. and Garcia-Nocetti F.; "Switching Fuzzy Logic Control for a Reconfigurable System Considering Communication Time Delays"; Proceedings, CDROM, European Control Conference; ECC 03 September, 2003.

Benítez-Pérez, H., and García-Nocetti, F.; "Reconfigurable Distributed Control "; Springer Verlag, 2005a.

Benítez-Pérez H., García-Zavala A and García-Nocetti F.; "Alternative Method based upon Planning Scheduler for On-Line Reconfiguration using System Performance"; Fifth IEEE International Symposium and School on Advance Distributed Systems ISSADS, Lecture Notes on Computer Science 3563, pp. 141-152, **ISBN:** 3-540-28063-4, México, 2005b.

Blanke, M., Kinnaert M., Lunze J., and Staroswiecki M.; "Diagnosis and Fault Tolerant Control"; Springer, 2003.

Cervin, A., Henriksson, D., Lincoln, B., Eker, J., and Arzén, K.; "How Does Control Timing Affect Performance?"; IEEE Control Systems Magazine, Vol. 23, pp. 16-30, 2003.

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

Frank, T., Kraiss, K.F., and Kuhlen, T; "Comparative Analysis of Fuzzy ART and ART-2A Network Clustering Performance"; IEEE Transactions on Neural Networks, Vol. 9, No. 3, May 1998.

Höppner, F., Klawonn, F., Kruse, R., and Funkler, T.; "Fuzzy Cluster Analysis"; John Wiley and Sons, 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.

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.

Liu L., and Layland L.; "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment" Journal of ACM., Vol. 20, pp. 46-61, 1973.

Nguyen H., Prasad N., Walker C., and Walker E.; "Fuzzy and Neural Control"; Ed. CRC, 2003.

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

Thompson, H.;"Wireless and Internet Communications Technologies for monitoring and Control"; Control Engineering Practice, vol. 12, pp. 781-791, 2004.

Winconhttp://www.quanser.com/english/html/solutions/fs_soln_software_wincon.html, 2003.

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