

An Implementation of Reconfigurable Network Control based upon Automata Proposal for Three Conveyor Belt Case Study

Benítez-Pérez H., Cárdenas-Flores F., García-Nocetti F.

Abstract: Online reconfiguration performed by a computer network system needs to be addressed from several perspectives due to complexity onto the system. This paper proposes different modeling approximations to obtain a holistic view of reconfiguration onto complex systems. First model is dynamic system modeling, second is an automaton in order to bound possible scenarios and third model is a Real Time scheduling algorithm to match possible configurations and related control laws.

1 Introduction

One of the main issues in fault tolerance is to keep availability even in hazard situations. A way to guarantee this is by reconfiguration where several consequences are expected. In that respect, either fault coverage, masking or tolerance are strategies to use during reconfiguration. Reconfiguration is a need in order to keep safety during fault scenarios. The results of this action modify several structures within complex systems like communication, dynamic behaviour and predictable response. In order to cover every aspect of these requirements, modeling becomes a crucial issue to get enough information by performing reconfiguration. In here, three different modeling techniques are followed, dynamic system modeling, automaton modeling and scheduling representation. These three strategies, by there own, lacks of a holistic view of the effects of a process such as reconfiguration. For instance, automaton strategy allows a structural view of reconfiguration without determining the effects of this action into dynamic behaviour of the system. Alternatively, dynamic system modeling provides a formal view of the effects of time delays and the loose of certain dynamic elements, however, reconfiguration is not a predetermined action only the effects of it. The novelty of this approximation is to integrate various modeling strategies to accomplish reconfiguration and its effects.

Firstly, dynamic system is modeled in order to determine where time delays play a key role. Secondly, automaton modeling is pursued in order to bound scenarios during fault and fault-free situations. From this automaton representation and taking into account real-time requirements, real-time scheduling through a special representation is implemented. The combination of these three allows system modeling even in complex situations like reconfiguration.

The objective of this paper is to present a strategy for control reconfiguration based upon time delay knowledge using a scheduling algorithm. Fault effects are local within a distributed system environment. The use of a case study is pursued to accomplish this objective. The novelty of this work is the amalgamation of scheduling and control techniques to get this strategy.

In particular, for the case of dynamic system modeling, several strategies for managing time delay within control laws have been studied for different research groups. For instance Nilsson [8] proposes the use of a time delay scheme integrated to a reconfigurable control strategy based upon a stochastic methodology. On the other hand, Wu [10] 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. [6] where time delays are used as uncertainties, which modify pole placement of a robust control law. Izadi et al. [5] 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. [2] where a logical relation between dynamic variables and faults are established. Alternatively reconfigurable control may performs a combined modification of system structure as studied by Benítez-Pérez et al. [1] and Thompson [9]. Another technique

like gain scheduling (Khalil, [7]) may give an interesting approximation to several time delay scenarios, however complexity related to system modeling during fault conditions is out the scope of this paper.

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 for online time delays.

2 Case Study and Control Reconfiguration Approach

Due to the complexity of this approach a particular implementation is pursued in order to present the potential of complementary modeling for reconfiguration. The Case study is based on three conveyor belts (Gudmundsson, [4]) integrated as follows, it comprises 3 conveyors belts, 4 actuators, and 12 sensors. It has 16 computing elements that considers the controller and the bus controller. Fig. 1 shows a diagram of this implementation. The procedure of the example is next; conveyor belt 1 detects a box, it modifies its speed up to a certain level to transport it in a faster way the box. MC is stated for Micro-Controller

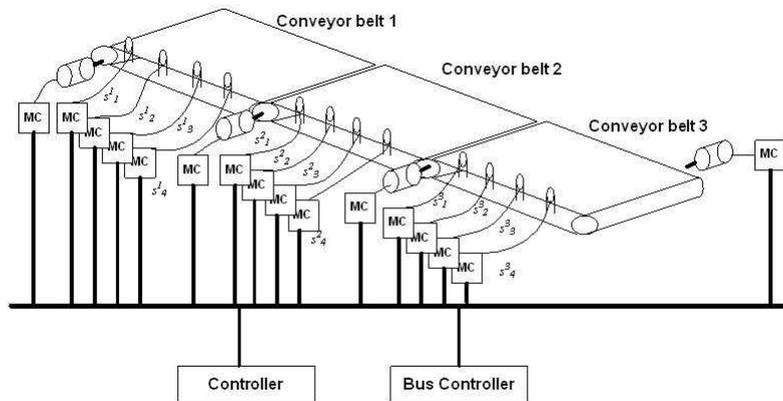


Figure 1: Conveyor belt example

When this box arrives to conveyor belt 2, its speed is modified up to another level to transport this box in a faster manner. Similar behavior is presented at conveyor belt 3. The sensor vector is used to detect the current position of box in any of these conveyor belts. Furthermore, actuator 4 has the task of pushing the current box when it arrives at this position.

For the case of multiple boxes in this example, the aim is that the boxes never crashed between each other. From this explanation, Table 1 shows the modification of speeds.

This case study is peculiar in that each conveyor belt has two different speeds as shown in Table 1.

These speeds are dependent on the sensor situation. This sensor situation is depicted as low and high, which is a semaphore for determining the presence of an object.

The second peculiarity is related to the difference between HS as follows:

$$HS^1 < HS^3 < HS^2, \quad (1)$$

where the middle conveyor belt is the fastest, then third conveyor belt, and so on. As the reader may realize, there are four motors, three for the conveyor belts and the fourth is to pull any object presented at its region.

Based on this case study, the response of the three actuators is shown in Fig. 2. Different speed-ups are shown assuming that a box is presented during a certain time. For instance, the first conveyor belt

Table 1: Speed selection

	Conveyor belt 1	Conveyor belt 2	Conveyor belt 3
Sensors $S_*^1 = \text{Low}$	LowSpeed	Low Speed	Low Speed
Sensors $S_*^1 = \text{High}$	High Speed HS^1	Low Speed	Low Speed
Sensors $S_*^2 = \text{Low}$	Low Speed	Low Speed	Low Speed
Sensors $S_*^2 = \text{High}$	Low Speed	High Speed HS^2	Low Speed
Sensors $S_*^3 = \text{Low}$	Low Speed	Low Speed	Low Speed
Sensors $S_*^3 = \text{High}$	Low Speed	Low Speed	High Speed HS^3

presents a faster speed-up during the first 3000 seconds, in comparison with the low speed-up during 3000 to 6000 seconds. This speed-up is shown as a change of slope of the current graphic. Similar behavior is presented for both conveyor belts as 2 and 3 are modified, because HS^2 is bigger than HS^3 .

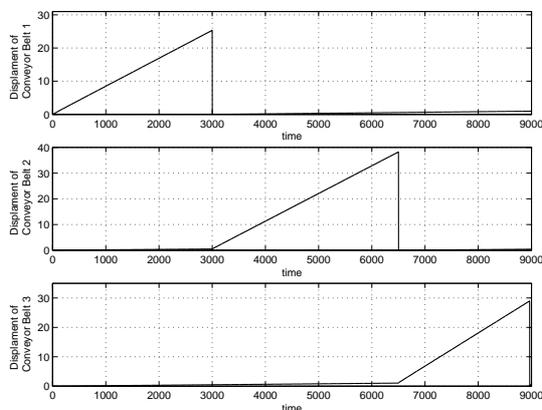


Figure 2: Related displacement when a box is present in each conveyor belt

2.1 First modeling approach

First modeling approach is based on control law modification taken into account time delay appearance, in particular for current case study. The schematic setup is based on Fig. 3 considering system response and control implementation.

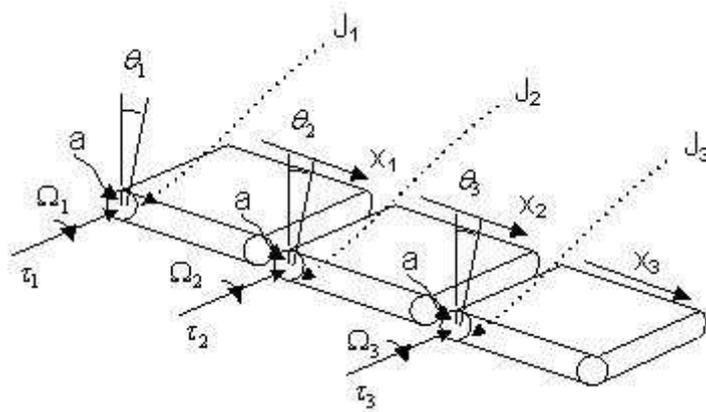


Figure 3: **Dynamic System Implementation**

In Fig. 3,

- x is the linear displacement
- θ_* is the angular displacement
- ω_* is the angular velocity
- F is the lineal force
- J_* is the lineal inercy
- k is the parameter
- τ_* is the torque
- a is the radius
- v is the lineal velocity

In this case, the plant presents two cases with or without a box per belt. As the second case is trivial, the first case is expressed per belt considering the mass of the box (referred to as m). The first conveyor belt is expressed as

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{\theta}_1 \end{bmatrix} = \begin{bmatrix} J/m & 0 \\ 0 & J_1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \dot{x}_1 \end{bmatrix} - \begin{bmatrix} 1/m \\ 0 \end{bmatrix} \tau_1 \quad (2)$$

$$y = \dot{x}_1.$$

The second conveyor and the third conveyor belt follow simmilar dynamics From these considerations, discrete plants are defined next by considering the presence of the box

$$x(k+1) = Ax(k) + \sum_{i=0}^l B_i^k u(k-i) \quad (3)$$

$$B_i^k = \int_{t_i^k}^{t_{i+1}^k} \exp(A(T-\tau)) B d\tau,$$

where $l = 1$ because the maximum number of sensors with delays is just one. Therefore, the A matrix is expressed as:

$$A_* = \begin{bmatrix} \exp(J/m) & 0 \\ 0 & \exp(J_*) \end{bmatrix}, \quad (4)$$

where T is the inherent sampling period, and t_0^k , t_1^k , and t_2^k are the related delays of the plant. For the case of local control laws, these are expressed next as:

$$\begin{aligned} x_c(k+1) &= A_c x_c(k) + B_c u_c(k) \\ y_c(k) &= C_c x_c(k - \tau_c - D_c u_c(k - \tau_c)), \end{aligned} \quad (5)$$

giving the delays as a result of decomposition from sensor and actuators, which are expressed as τ_{sc} and τ_{ca} respectively. The augmenting representation is given next:

$$\begin{aligned} u_c(k) &= y_p(k - \tau_{sc}) \\ u_p &= y_c(k - \tau_{ca}), \end{aligned} \quad (6)$$

where states are augmented as:

$$z = \begin{bmatrix} x_p(k) \\ x_c(k) \end{bmatrix}, \quad (7)$$

and expressed as:

$$\begin{aligned} z(k+1) &= \begin{bmatrix} A_p & 0 \\ 0 & A_c \end{bmatrix} z(k) + \begin{bmatrix} 0 & 0 \\ B_c C_p & 0 \end{bmatrix} z(k - \tau_{sc}) + \begin{bmatrix} B_p D_c C_c & 0 \\ 0 & 0 \end{bmatrix} z(k - \tau_{sc} - \tau_{ca} - \tau_c) \\ &+ \begin{bmatrix} 0 & B_p C_c \\ 0 & 0 \end{bmatrix} z(k - \tau_{ca} - \tau_c). \end{aligned} \quad (8)$$

By modifying Eqn. 8 to define the stability of network control, the next configuration is proposed:

$$\begin{aligned} F^j &= \begin{bmatrix} A_p^j & 0 \\ 0 & A_c^j \end{bmatrix} \\ F_1^j &= \begin{bmatrix} 0 & 0 \\ B_c^j C_p^j & 0 \end{bmatrix} & F_2^j &= \begin{bmatrix} B_p^j D_c^j C_p^j & 0 \\ 0 & 0 \end{bmatrix} \\ F_3^j &= \begin{bmatrix} 0 & B_p^j C_c^j \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Therefore, the state vector is modified for these time delays where the system is asymptotically stable based on $F^j + \sum_{i=1}^3 F_i^j$. Based upon a single control loop is stable (Eqn. 8), it is possible to define stability for every loop as shown in Eqn. 10.

$$\tau < \frac{\sigma}{\delta \sum_{i=1}^3 \left| F_i^j \left(F^j + \sum_{i=1}^3 F_i^j \right) \right|}, \quad (9)$$

where τ is the maximum value from all possible time delays at all loops. The absolute value is used in order to guarantee positive response with respect to current time delay.

$$\begin{aligned} \sigma &= \frac{\lambda_{\min}(Q)}{2\lambda_{\max}(P)} \\ \delta &= \left[\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)} \right]^{\frac{1}{2}}, \end{aligned}$$

where $\lambda_{\max}(P)$ and $\lambda_{\min}(Q)$ are the maximum and minimum eigenvalues of P and Q matrices respectively. λ_{\min} and λ_{\max} can not be complex values since P and Q are bounded to be real values. The proposed configuration is presented

$$\left(F^j + \sum_{i=1}^3 F_i^j \right)^T P + P \left(F^j + \sum_{i=1}^3 F_i^j \right) = -Q, \quad (10)$$

where P, Q are positive definite symmetric matrices and are eigenvalues of the matrix and where the chosen Equation is

$$V(x) = \frac{1}{2}x^{jT}(t)Px^j(t). \quad (11)$$

Based upon Lyapunov proposed Equation (Eqn. 11) and by its derivative as shown in Eqn. 12, where \dot{x} is substituted by the enhanced representation of \dot{z} which contains both states, from the plant as well as the controller.

$$\begin{aligned} \dot{V}(x) &= \frac{1}{2}\dot{z}^{jT}(t)Pz^j(t) + \frac{1}{2}z^{jT}(t)P\dot{z}^j(t) \\ &\leq -\frac{1}{2}z^{jT}(t)Qz^j(t) + \\ &\quad \left| z^{jT}P \sum_{i=1}^3 F_i \int_{t_i^j}^0 \left[F^j z^j(t+\theta) + \sum_{i=1}^3 F_i^j z^j(t-\tau_i^j + \theta) \right] d\theta \right|. \end{aligned} \quad (12)$$

From control law expression, the related time delays are defined as τ_{sc}, τ_{ca} and τ_c where their respective values will be incorporated later in this section through splitting the time delay (Eqn. 8).

2.2 Second Modelling Approach

Having shown local control laws structures, second modeling approach is the global structure in terms of an automaton (Fig. 4). Where reconfiguration is expressed for the formal event manager. In this case, two states are possible with several events, which are managed by the sensor vector for each belt (first, second, and third belts) and expressed as $S_{1 \leq i \leq N}^1, S_{1 \leq i \leq N}^2$ and $S_{1 \leq i \leq N}^3$, respectively considering fault free scenario. It is important to mention that $S_{1 \leq i \leq N}^1, S_{1 \leq i \leq N}^2$ and $S_{1 \leq i \leq N}^3$ are independent conditions between conveyors, then, $S_{1 \leq i \leq N}^1 = 0$ means there is no box on conveyor belt 1. At the same time $S_{1 \leq i \leq N}^2 \neq 0$ can be stated meaning there is a box in second conveyor belt. Other condition can be presented as $S_{1 \leq i \leq N}^1 \neq 0, S_{1 \leq i \leq N}^2 \neq 0$ and $S_{1 \leq i \leq N}^3 \neq 0$ and where three boxes are presented on the system, each box per conveyor belt. The same sensor conditions are presented in Fig. 5.

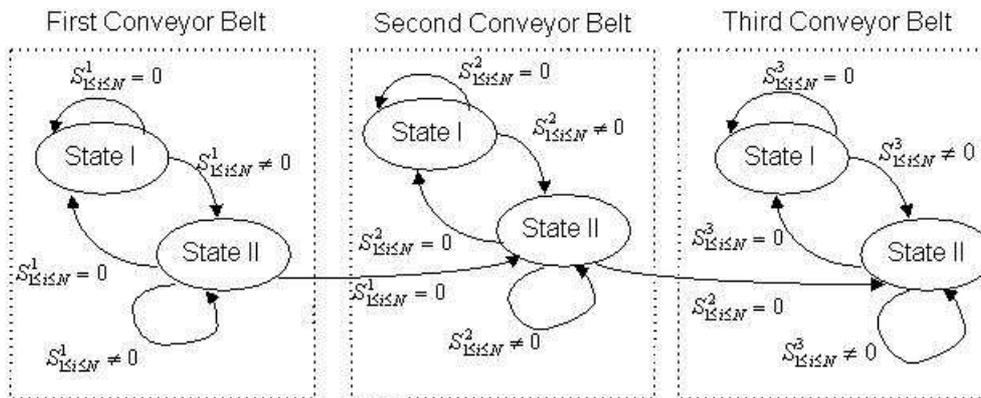


Figure 4: Fault-free scenario in terms of global structure

The switching effect is neglected in this fault-free scenario. In this scenario two cases are defined, when a box is presented (case II) or the other case (case I). For the second case the chosen control is to maintain the conveyor belt in zero speedup. For first case, the chosen controller is related to certain speed up depending on each conveyor belt.

For the case of a fault scenario, a new state appears for global control (Fig. 5) related to the action pursued when a fault is presented. The necessary event for reaching such a state is $S_{1 \leq i \leq N}^1 \neq 0$, and the fault's last event is composed of local information given by each local sensor with a relation to the health condition measures.

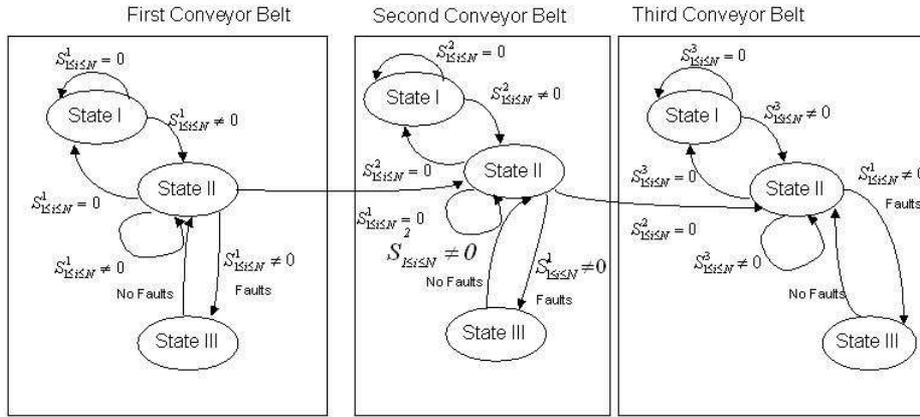


Figure 5: Local fault scenario for the global structure

Considering individual modeling, there is one type of local fault to be considered, which is that one of the sensors is faulty with no consideration of the type of fault. It is assumed that the fault is detectable and measurable (This is a condition in this paper).

2.3 Third Modeling Approach

Third modeling approach is related to the use of scheduling algorithm in order to determine possible time delays between processes. It is important to remember that the fault tolerance strategy is based on the use of consecutive sensors to mask the fault using extra communication to perform lateral agreement. Therefore, this approach provides two different time graphs, one for each scenario (fault and fault free), as shown in Figs. 6 and 7, respectively. The cases from Figs. 4 and 5 are related to bounded time delays from decision maker (Benítez-Pérez et al., [1] and have an effect into control and plan modeling as shown before. At the end of this section (Eqn. 15) it is shown how time delays are modified based on time diagram representation (Figs. 6 and 7). The reader should realize that time delays are bounded by the use of scheduling algorithm through ART2 network.

Both scenarios are local with respect to one belt. It is considered that the other two belts do not present faulty conditions. As these two scenarios are bounded (fault and fault-free), the respective total consumption times (tt and tt_f respectively) are shown in Eqns. 13 and 14 (Figs. 6 and 7, respectively), where variable information is presented.

$$tt = t_s * 4 + t_{cm}^{sc} + t_c + t_{cm}^{ca} + t_a \tag{13}$$

where:

- t_s is the consumed time by sensors
- t_{cm}^{sc} is the consumed time by communication between sensor and control
- t_c is the consumed time by control node
- t_{cm}^{ca} is the consumed time by communication between controller and actuator
- t_a is the consumed time by actuator

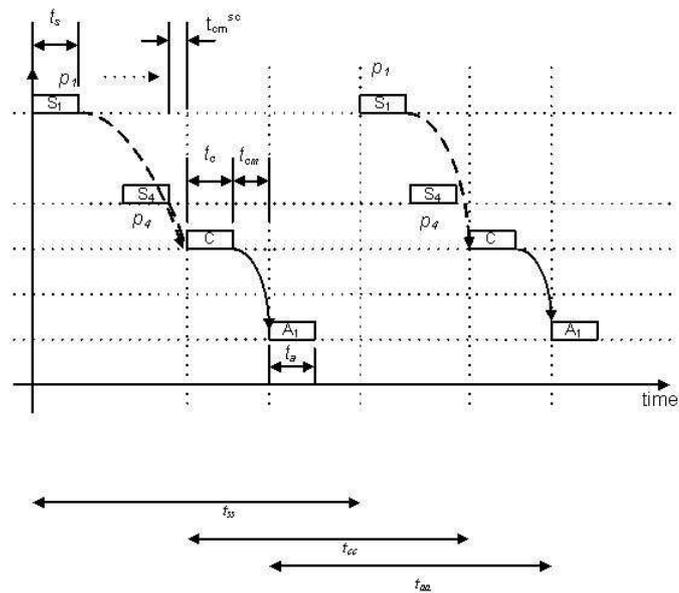


Figure 6: Fault-free scenario

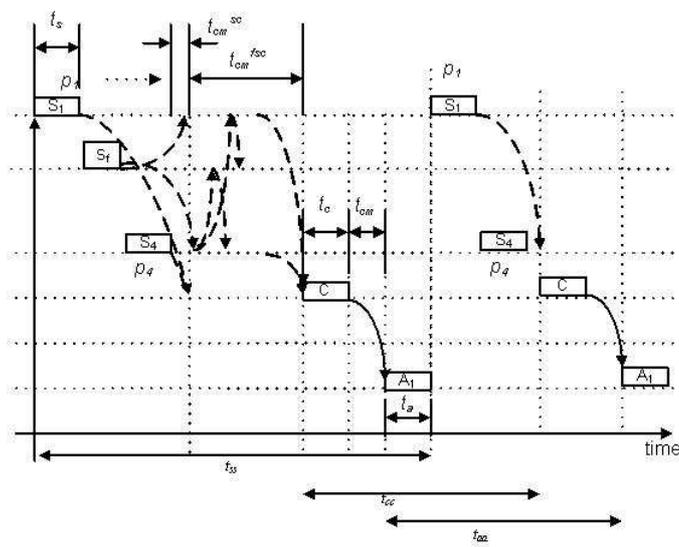


Figure 7: Fault scenario considering fault masking

$$tt_f = t_s * 4 + t_{cm}^{sc} + t_{cm}^{fsc} + t_c + t_{cm}^{ca} + t_a \quad (14)$$

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

From this time boundary based on these three modeling approaches, it is feasible to implement the control strategy as presented in Eqn. 10. As mentioned before, time delays take place in three representations τ_{sc} , τ_{ca} and τ_c , therefore, by decomposing Eqn. 14, time delays are expressed as:

$$\begin{aligned} \tau_{sc} &= t_s * 4 + t_{cm}^{sc} \\ \tau_c &= t_{cm} + t_c + t_{cm}^{ca} \\ \tau_{ca} &= t_a, \end{aligned} \quad (15)$$

A remarkable issue is related to a particular sensor fault related to any of three belts. Considering this configuration, three cases are possible:

- One local fault;
- Two local faults, one per belt;
- Three local faults, one per belt.

Based on these three possible configurations, there is a worst-case scenario related to three local faults that has an impact on the global control strategy. The other two configurations 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 (fault masking between sensors) and the local controllers. Taking into account these three possible configurations, the local and global time delays are described in Table 2.

Table 2: Time delays related to local and global effects

Configuration 1	Local Time Delays	110 ms
One Local Fault	Global Time Delays	110 ms
Configuration 2	Local Time Delays	110 ms
Two Local Fault	Global Time Delays	220ms
Configuration 3	Local Time Delays	110 ms
Three Local Fault	Global Time Delays	400 ms

3 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 and two local faults, respectively (Figs. 8, 9 and 10). First scenario presents a fault free situation where local controller response is shown, it is important to highlight that fourth actuator response presents a normal response.

When first fault appears (one local sensor does not response and the masking approach is followed) a local time delay takes place where its effects are shown in first local control response. Similar situation

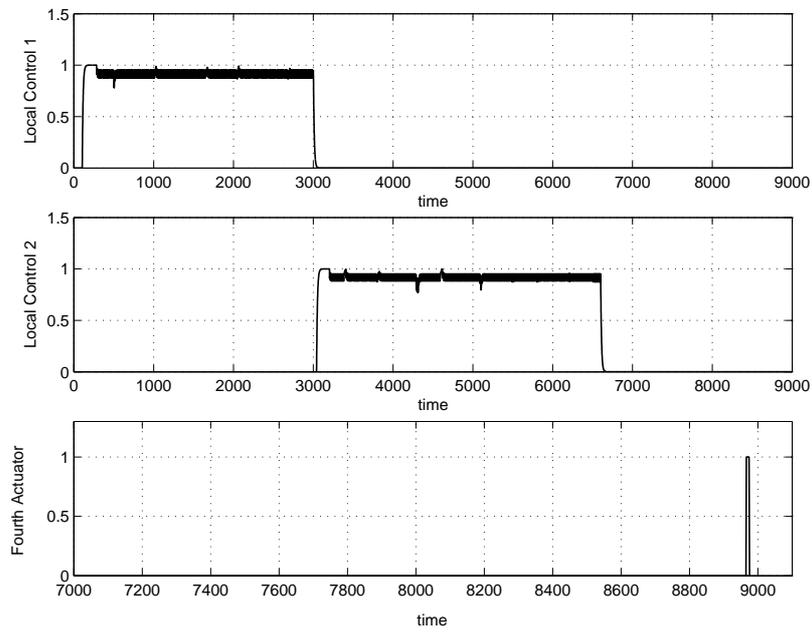


Figure 8: **Fault-free scenario**

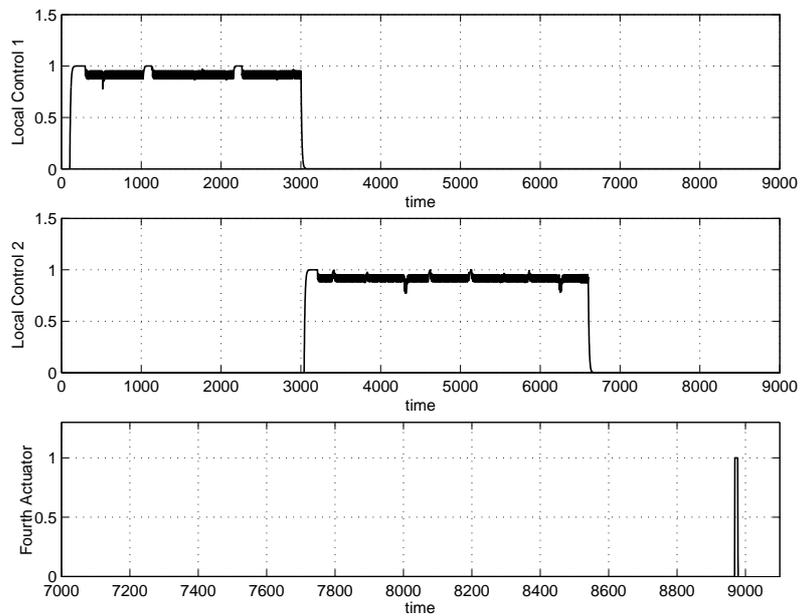


Figure 9: **First local fault appearance and related global effects**

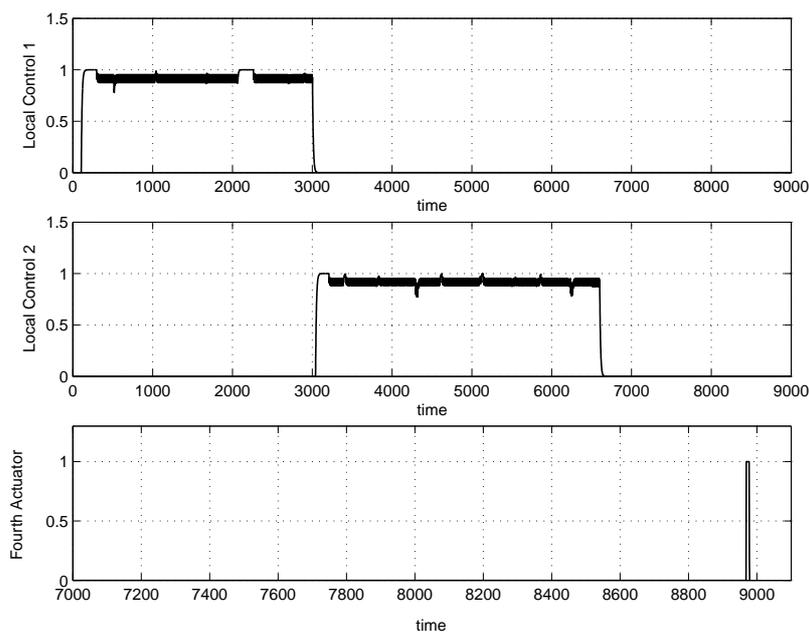


Figure 10: **Second local fault appearance and related global effects**

is presented when a second fault appears and first fault is still active (Fig. 10). In this case no relevant modification is shown due to time switching controller takes place.

This example presents three local control cases with a global automaton in which control reconfiguration is based on the scheduling algorithm, 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. It is obvious that fault presence is measurable; if this local fault localization approach cannot detect faults, this strategy becomes useless.

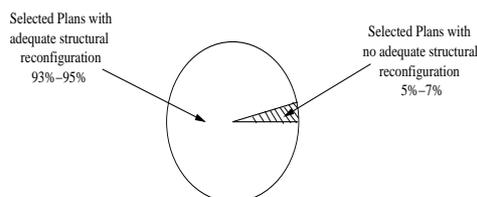


Figure 11: **Percentage of Selected Valid Plans for Structural Reconfiguration**

Moreover, 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. The scheduler performs task reorganization based on their consumption times and fault presence.

The number of accepted plans is presented taking into those selected with no adequate response from structural reconfiguration (Fig. 11). For instance, some tasks would not have enough time to be sampled and executed. This result is presented as the percentage of the adequate use of structural reconfiguration during on-line stage. In this case, current control law is modified according to time delays status.

Having defined the percentage related to those adequate plans during structural reconfiguration, this is taking as 100 % and is evaluated in terms of control law performance. The results are presented in Fig. 12. In here, 97% of the valid plans have a valid response in terms of the mean square error response

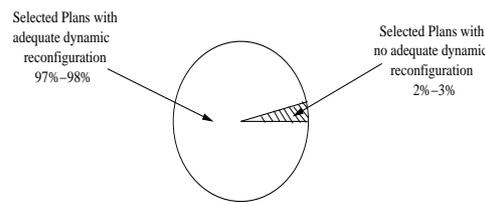


Figure 12: **Percentage of Selected Valid Plans for Control Law Reconfiguration**

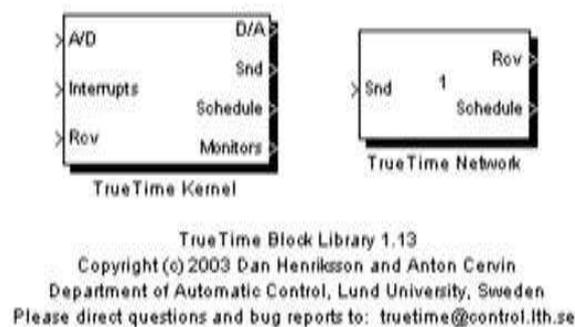


Figure 13: **Basic model of true time**

from the dynamic response of case study.

To define the communication network performance, the use of the true-time network 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. [3]) the true time main characteristics are shown next. In the true time model, computer and network blocks are introduced in Fig. 13.

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. These have been developed by the Department of Automatic Control, Lund Institute of Technology, Sweden. 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 the network. This tool provides the necessary interruptions to simulate delay propagation as well as synchronization within the network.

4 Concluding Remarks

Present approach shows the integration of three modelling techniques in order to perform reconfiguration. Three approaches are followed, control law design, automaton modelling and RT scheduling strategy. 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 when one control is the adequate. This goal is reached by a strategy compose of three algorithms, one which is responsible for structural reconfiguration and it has been implemented in this paper as ART2A network. Second

algorithm is responsible for dynamic control design and third algorithm is based on an automaton technique to perform switching control. What 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.

5 Acknowledgements

The authors would like to thank the financial support of DISCA-IIMAS-UNAM, and UNAM-PAPIIT (IN101307 and IN105303) Mexico in connection with this work. And the High Performance Computing Project within the “Macroproyecto Tecnologías para la Universidad de la Información y la Computación” of the Universidad Nacional Autónoma de México (UNAM).

Bibliography

- [1] Benítez-Pérez H. and García-Nocetti F., Reconfigurable Distributed Control, *Springer Verlag*, 2005.
- [2] Blanke M. and Kinnaert M. and Lunze J. and Staroswiecki M., Diagnosis and Fault Tolerant Control, *Springer*, 2003.
- [3] Cervin A. and Henriksson D. and Lincoln B. and Eker J. and Arzén K., How Does Control Timing Affect Performance, *IEEE Control Systems Magazine*, Vol. 23, pp. 16-30, 2003.
- [4] Gudmundsson D. and Goldberg K., Tuning Robotic Part Feeder Parameters to Maximize Throughput, *Assembly Automation Publisher: MCB University Press*, Vol. 19, No. 3, pp216-221, 1999.
- [5] 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.
- [6] Jiang J. and Zhao Q., Reconfigurable Control Based on Imprecise Fault Identification, *Proceedings of the American Control Conference, IEEE*, pp. 114-118, San Diego, June, 1999.
- [7] Khalil H., Nonlinear Systems, *Prentice Hall*, 2002.
- [8] Nilsson, J., Real-Time Control with Delays, *PhD. Thesis, Department of Automatic Control, Lund Institute of Technology*, Sweden, 1998.
- [9] Thompson, H., Wireless and Internet Communications Technologies for monitoring and Control, *Control Engineering Practice*, vol. 12, pp. 781-791, 2004.
- [10] 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.

Benítez-Pérez H.
Universidad Nacional Autónoma de México
Departamento de Ingeniería de Sistemas Computacionales y Automatización
Apdo. Postal 20-726., Admón. No. 20
Del. A. Obregón, México D. F., CP. 01000, México.
E-mail: hector@uxdea4.iimas.unam.mx

Cárdenas-Flores F., García-Nocetti, F.
IIMAS, Universidad Nacional Autónoma de México
Apdo. Postal 20-726., Admón. No. 20
Del. A. Obregón, México D. F., CP. 01000, México

Received: August 1, 2007



Hector Benítez-Pérez is a full time researcher in the IIMAS UNAM (México). He obtained his BSc in electronic engineering at the Engineering Faculty UNAM in 1994 and his PhD at Sheffield University, UK en 1999. His areas of interest are in Real Time Control and Fault Diagnosis.



F. Cárdenas-Flores obtained his BSc in Biomedical Engineering at UAM-Iztapalapa 1990. He works in the Departamento de Ingeniería en Sistemas Computacionales IIMAS UNAM since 1994. He areas of interest are in Real time, Parallel computing and Digital Signal and Image Processing.



F. García-Nocetti BSc in electric engineering at the Engineering Faculty at UNAM in 1984. MBs and PhD at University of Wales, UK 1988, 1991. He is a full time researcher at IIMAS UNAM since 2000. His areas of interest are architectures and algorithms for High performance computing including real time applications. He is member of IEEE, IEE and IFAC.