International Journal of Innovative Computing, Information and Control
Volume **8**, Number **12**, December **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 8515–8526

NETWORKED CONTROL SYSTEMS DESIGN CONSIDERING SCHEDULING RESTRICTIONS AND LOCAL FAULTS

H. Benítez-Pérez[1], A. Benítez-Pérez[2] and J. Ortega-Arjona[3]

[1]Departamento de Ingeniería de Sistemas Computacionales y Automatización
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
[3]Departamento de Matemáticas
Facultad de Ciencias
Universidad Nacional Autónoma de México
Apdo. Postal 20-726, Del. A. Obregón, México D. F., C.P. 01000, México
hector@uxdea4.iimas.unam.mx

[2]Centro de Estudios Científicos y Tecnológicos número 11
Instituto Politécnico Nacional
Av. De los Maestros S/N, México D. F., México

Received October 2011; revised February 2012

Abstract. *Nowadays, Network Control Systems represent a common solution when solving connectivity issues for distributed control systems. However, this approach tends to increase the time complexity in terms of time delays, and thus, making necessary the study of the behavior of such time delays and the differential equations which model such a behavior. Time delays need to be known a priori, but from a dynamic, real-time behavior. To do so, this paper presents the use of a dynamic, priority exchange scheduling. The objective is to show how to tackle multiple time delays, as well as faults that are bounded, and the dynamic response from real-time scheduling approximation. The novelty of this approach is to use both, the appearance of faults and time delays, as perturbations considering nonlinear behavior through a fuzzy TKS approach, in a codesign strategy. The related control law is designed considering fuzzy logic for nonlinear time delays coupling and inherent local fault appearance.*
Keywords: Network, Control systems, Fuzzy TKS

1. **Introduction.** Real-time restrictions are the most certain time constraints when the behavior of a system tends to be periodic and repeatable. Considering this, the control design and stability analysis of Network Control Systems (NCSs) have been studied, in which a key scheduling restriction is the effect of the network-induced delays in the NCS performance. This delay can be constant, time-varying, or even random, depending on the scheduler, network type, architecture, operating systems, etc. Further, when a local fault occurs during the operation of an NCS, a respective local fault tolerance strategy has to be applied. Nevertheless, applying such a fault tolerance strategy impacts the overall performance of the NCS, since dynamic conditions are modified. Therefore, it seems necessary to take into account current local conditions, in order to keep NCS performance, even if it is degraded.

For example, let us suppose a general NCS. Different local faults may simultaneously occur during its operation. Normally, due to the distributed nature of the NCS, these faults are locally tackled using a scheduling algorithm, which introduces time delays due to the local recuperation strategies that, for safety reasons, may be bounded to a maximum allowable transfer interval (MATI) [15]. Hence, the local application of the scheduling algorithm is needed for the partial recovery of the NCS, but this implies that the overall performance of the NCS is now degraded. Given this, a solution may be to model each bounded time delay due to each local fault as well as the local impact of the fault itself. Based on this simple idea, it is possible to model the global and nonlinear effects of the time delays, in order to propose each time a controller that results adequate given each particular situation.

This paper proposes a novel codesign strategy, which considers bounded time delays for NCSs. Codesign implies the concurrent development of both, hardware and software components. The use of codesign here attempts to improve the main advantages of NCSs, such as their low cost, small volume of wiring, distributed processing, simple installation, maintenance, and reliability [2]. In this codesign strategy, an NCS is reconfigured according to bounded time delays. Reconfiguration is as "a transition that modifies the structure of a system so it changes its representation of states" [12]. Reconfiguration is used here as a feasible approach for modifying time delays, as well as taking into account local fault conditions (such as sensor faults), in order to maintain the performance of the NCS. It is necessary to represent a nonlinear NCS with these two key characteristics: time delays (bounded by a scheduler algorithm) and the appearance of local faults. Thus, in this paper, time delays and the appearance of local faults are taken to develop a fuzzy model of the controller, capable of solving this nonlinear behavior. This control approach is feasible, since local faults are known as well as bounded time delays.

2. Related Work.

In control systems, several modeling strategies for managing time delays within control laws have been studied by different research groups. For example, [11] proposes the use of a time delay scheme integrated to a reconfigurable control strategy, based on a stochastic methodology. [7] describes how time delays are used as uncertainties, which modify pole placement of a robust control law. [6] presents an interesting case of fault tolerant control approach related to time delay coupling. [3] studies reconfigurable control from the point of view of structural modification, establishing a logical relation between dynamic variables and the respective faults. Finally, [1,14] consider that reconfigurable control strategies perform a combined modification of system structure and dynamic response, and thus, this approach has the advantage of bounded modifications over system response.

Regarding NCS, [11] also analyzes several important facets of NCSs, by introducing models for the delays in NCS: first as a fixed delay, then as an independently random, and finally, like a Markov process. Optimal stochastic control theorems for NCSs are introduced, based upon the independently random and Markovian delay models. [15] introduces static and dynamic scheduling policies for transmission of sensor data in a continuous-time LTI system. They introduce the notion of the maximum allowable transfer interval (MATI). This is the longest time after which a sensor should transmit a data. They also derive bounds of the MATI, such that the NCS is kept stable. This MATI ensures that the Lyapunov function of the system under consideration is strictly decreasing at all times [18]. [18] extends the work of [15] developing a theorem which ensures the decrease of a Lyapunov function for a discrete-time LTI system at each sampling instant, by using two different bounds. These results are less conservative than those of Walsh, since here it is not required that the system Lyapunov function should be strictly decreasing at all time. Further, a number of different linear matrix inequality (LMI) tools for analyzing and designing optimal switched NCSs are introduced. Alternatively, [19] takes into consideration both the network-induced delay and the time delay in the plant, and thus, introduces a controller design method, using the delay-dependent approach. An appropriate Lyapunov functional candidate is used to obtain a memoryless feedback controller, derived by solving a set of Linear Matrix Inequalities (LMIs) [18]. [16] models the network induced delays of the NCSs as interval variables governed by a Markov chain. Using the upper and lower bounds of the delays, a discrete-time Markovian jump system with norm-bounded uncertainties is presented to model the NCSs. Based on this model, an H_∞ state feedback controller can be constructed via a set of LMIs. Recently, [5] introduces a new (descriptor) model transformation for delay-dependent stability, for systems with time-varying delays in terms of LMIs. He also refines recent results on delay-dependent H_∞ control, and extends them to the case of time-varying delays.

All these previous works have significance to understand how to treat the time delays, regardless of their nature, as well as the impact of local faults. Based upon this review, this paper presents a model that integrates the time delays for a class of nonlinear system and a fuzzy control for NCSs [12,13,18], considering time delays induced by the computer network as a result of online reconfiguration. This modeling approach is presented to bound time delays from computing behavior and taking local faults as a key issue, enhancing this modeling approximation. Also, the stability analysis is revised as well.

3. **Scheduling Approach.** The objective here is to present a reconfiguration control strategy developed from the time delay knowledge, following scheduling approximation where time delays are known and bounded according to used scheduling algorithm. The scheduling strategy proposed here pursues to tackle local faults in terms of fault tolerance. In this situation, current time delays would be inevitable.

Classical Earliest Deadline First (EDF) plus Priority Exchange (PE) algorithm is used here to decompose time lines and the respective time delays when present. For instance, time delays are supervised for a number of tasks as follows:

$$C_1 \to C_n T_1 \to T_n \tag{1}$$

Priority is given as the well known EDF algorithm, which establishes that the process with the closest deadline has the most important priority [10]. However, when an aperiodic task appears, it is necessary to deploy other algorithms to cope with concurrent conditions. To do so, the PE algorithm is used to manage spare time from the EDF algorithm. The PE algorithm [4] uses a virtual server that deploys a periodic task with the highest priority in order to provide enough computing resources for aperiodic tasks. This simple procedure gives a proximity, deterministic, and dynamic behavior within the group of included processes. In this case, time delays can be deterministic and bounded. As an example, consider a group of tasks as shown in Table 1. In this case, consumption times as well as periods are given in terms of integer units. Remember: the server task is the time given for an aperiodic task to take place on the system.

The result of the ordering based upon PE is presented in Figure 1.

Based on this dynamic scheduling algorithm, time delays are given as current calculations in terms of task ordering. In this case, every time that the scheduling algorithm takes place, the global time delays are modified in the short and long term. For instance,

Table 1. First example for PE algorithm

Name	Consumption (in units)	Period (in units)
Task 1	2	9
Task 2	1	9
Task 3	2	10
Server	1	6

consider the following example, in which four tasks are set, and two aperiodic tasks take place at different times, giving different events with different time delays.

The following task ordering is shown in Figure 2, using the PE algorithm, where clearly time delays appear.

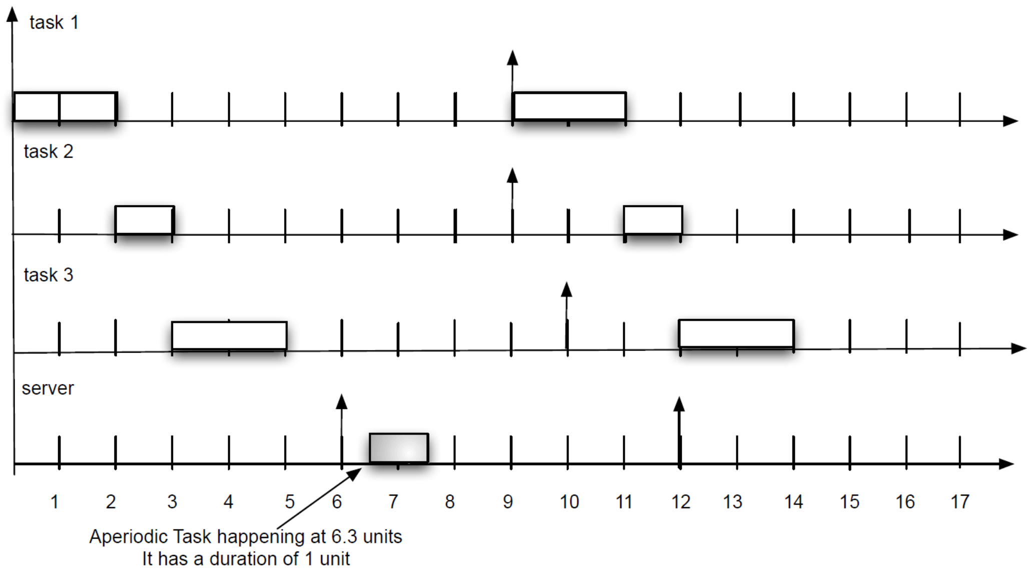

FIGURE 1. Related organization for PE of tasks in Table 1

TABLE 2. Second example of PE

Name	Consumption (in units)	Period (in units)
Task 1	2	9
Task 2	1	9
Task 3	2	10
Server	1	6
Aperiodic task 1 (ap1)	0.9	It occurs at 9
Aperiodic task 2 (ap2)	1.0	It occurs at 13

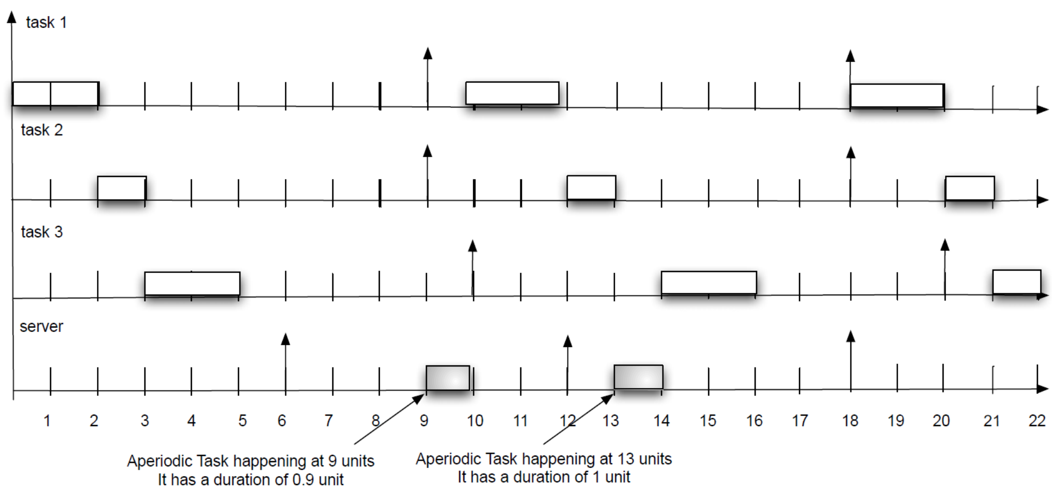

FIGURE 2. Task organization considering the second example for the PE algorithm

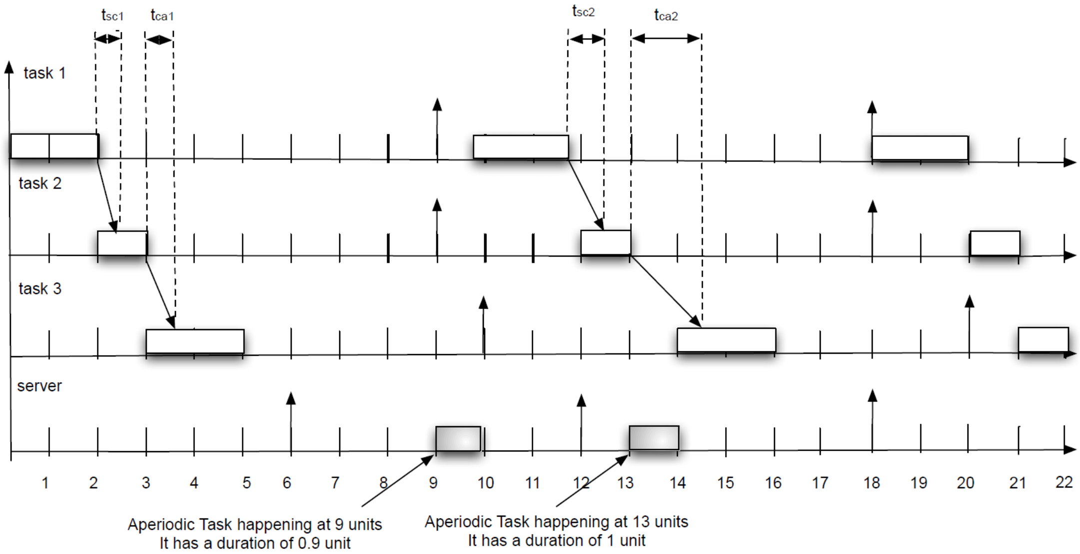

FIGURE 3. Related time delays are depicted according to both scenarios

Now, from this, a resulting ordering of different tiny time delays is given for two scenarios, as shown in Figure 3.

These two scenarios present two different local time delays that need to be taken into account before hand, in order to settle the related delays according to scheduling approach and control design. These time delays can be expressed in terms of local relations between both dynamical systems. These relations are the actual and possible delays, bounded as marked limits of possible and current scenarios. Then, delays may be expressed as local summations with a high degree of certainty.

In this last example, during the second scenario, a total delay is given as:

Total delay = consumption_time_delay_aperiodic_task1
+ consumption_time_delay_task1 + task2
+ consumption_time_delay_task2
+ consumption_time_delay_aperiodic_task2
+ consumption_time_delay_task3

Now, from this example, l_p is equal to 2 and l_c is equal to 3. l_p and l_c are the total number of local delays within one scenario from sensor to control and from control to actuator respectively.

4. **Fuzzy Control Design Considering Time Delays and Local Faults.** Having defined time delays as a result of a scheduling approximation as well as local faults considering sensor faults, since faults are bounded, several scenarios are potentially presented following this time delay behavior. In fact, the number of scenarios is finite, since the combinatorial formation is bounded. Therefore, any strategy in order to design a control law needs to take into account gain scheduling approximation. To do so, a fuzzy control strategy based upon Takagi-Sugeno is applied. Based on fuzzy control systems [10] it stays as:

$$\begin{array}{l} \mu_{ij}(x_i(t)) \\ 1 \rightarrow i \rightarrow m \quad \text{and} \quad 1 < j < m \end{array} \tag{2}$$

where x are the states, m is the number of inputs and m is the related membership function. Thus:

$$g_j = \prod_{i=1}^{m} \{\mu_{ij}[x_i(k)]\} \tag{3}$$

$$h_j = \frac{g_j}{\sum_{j=1}^{m} g_j} \tag{4}$$

$$X(k+1) = \sum_{j=1}^{m} [h_j\{A_j^p x(k) + B_j^p u(k)\}] \tag{5}$$

where A_j^p and B_j^p are the plant representation per scenario according to current time delays following Figure 4, and ρ_j is the relation of fault vector.

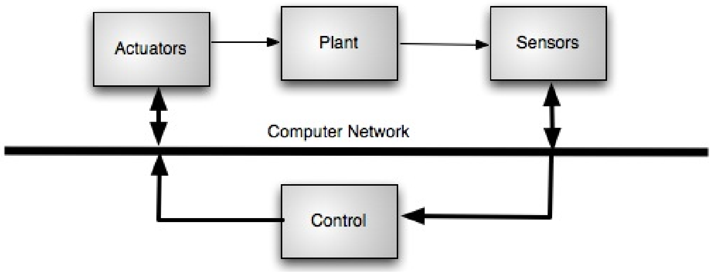

FIGURE 4. Network control system diagram

Now let t_{caj} be the current time delay from controller to actuator, and t_{scj} be the current time delay from sensor to controller, and ρ_j^p and ρ_j^c be the relations of fault presence through the event.

$$x_p(k+1) = \sum_{j=1}^{m} [h_j\{A_j^p x_p(k) + \rho_j^p B_j^p u_p(k - t_{caj})\}] \tag{6}$$

$$x_c(k+1) = \sum_{j=1}^{m} [h_j^p\{A_j^c x_c(k) + \rho_j^c B_j^c u_c(k - t_{scj})\}] \tag{7}$$

From [12], the time delay representation in terms of discrete view is:

$$B_j^p = \sum_{i=1}^{l_p} \left[\int_{t_i}^{t_{i+1}} \rho_j^c B_j^c e^{A_j^c t} dt \right] \tag{8}$$

$$B_j^c = \sum_{i=1}^{l_c} \left[\int_{t_i}^{t_{i+1}} \rho_j^p B_j^p e^{A_j^c t} dt \right] \tag{9}$$

Remember that l_p and l_c are the total number of local time delays that appears per scenario:

$$y_{p_i} = c_p^i x_p(k)$$
$$y_{c_i} = c_c^i x_c(k)$$

where l_p and l_c are the number of local time delays and the outputs are gathered as:

$$y_p = \sum_{j=1}^{m} [h_j\{c_p^j x_p(k)\}] \tag{10}$$

$$y_c = \sum_{j=1}^{m} [h_j\{c_c^j x_c(k)\}] \tag{11}$$

$$u_p(k - t_{caj}) \rightarrow y_c = c_c x_c(k) \tag{12}$$

$$u_c(k - t_{scj}) \rightarrow y_p = c_p x_p(k) \tag{13}$$

$$x_p(k+1) = \sum_{j=1}^{m} [h_j\{A_j^p x_p(k) + \rho_j^p B_j^p u_p(k - t_{caj})\}] \tag{14}$$

$$= \sum_{j=1}^{m} [h_j\{A_j^p x_p(k) + \rho_j^p B_j^p c_c x_c(k - t_{caj})\}] \tag{15}$$

From this last equation, the related dynamics are expressed as A_j^c, B_j^c, and C_j^c, where j is the index with respect to each scenario:

$$x_p(k+1) = \sum_{j=1}^{m} \left[h_j \left\{ A_j^p x_p(k) + \rho_j^p B_j^p \left\{ \sum_{i=1}^{m} h_j(c_c^i x_c(k - t_{cai})) \right\} \right\} \right] \tag{16}$$

From the state of the controller:

$$x_c(k+1) = \sum\nolimits_{i=1}^{m} [h_i\{A_i^c x_c(k) + \rho_j^c B_i^c u_c(k - t_{sci})\}] \tag{17}$$

$$x_c(k+1) = \sum\nolimits_{i=1}^{m} \left[h_i\left\{A_i^c x_c(k) + \rho_j^c B_i^c\left\{\sum\nolimits_{j=1}^{m} h_j[c_p^j x_p(k - t_{scj})]\right\}\right\}\right] \tag{18}$$

For x_p:

$$x_p(k+1) = \sum\nolimits_{j=1}^{m} [h_j\{A_j^p x_p(k) + \sum\nolimits_{i=1}^{m} h_i \rho_j^p B_j^p (c_c^i x_c(k - t_{cai}))\}] \tag{19}$$

$$x_p(k+1) = \sum\nolimits_{j=1}^{m} \sum\nolimits_{i=1}^{m} [h_j h_i[\rho_j^p B_i^p (c_c^i x_c(k - t_{cai}))] + h_j A_j^p x_p(k)] \tag{20}$$

For x_c:

$$x_c(k+1) = \sum\nolimits_{j=1}^{m} \left(h_j\left\{A_j^c x_c(k) + h_i \rho_j^c B_j^c\left(\sum\nolimits_{j=1}^{m} h_j(c_p^i x_p(k - t_{scj}))\right)\right\}\right) \tag{21}$$

$$x_c(k+1) = \sum\nolimits_{j=1}^{m} \sum\nolimits_{i=1}^{m} \left[h_j h_i\left[\rho_j^p B_j^p\left(c_c^j x_c(k) + \sum_{i=1}^{m} h_i \rho_j^c B_i^c\left(\sum\nolimits_{j=1}^{m} h_j(c_p^i x_p(k - t_{scj}))\right)\right)\right]\right] \tag{22}$$

$$x_c(k+1) = \sum\nolimits_{j=1}^{m} \sum\nolimits_{i=1}^{m} [h_j h_i \rho_j^c B_j^c (c_p^i x_p(k - t_{sci})) + h_j A_j^c x_c(k)] \tag{23}$$

Now, as augmented states:

$$X = \begin{bmatrix} x_c \\ x_p \end{bmatrix} \tag{24}$$

$$\begin{matrix} x_c(k+1) \\ x_p(k+1) \end{matrix} = \left\{ \begin{matrix} \sum_{j=1}^{m} \sum_{i=1}^{m} [h_j h_i[\rho_j^p B_j^p (c_c^i x_c(k - t_{cai}))] + h_j A_j^p x_p(k)] \\ \sum_{j=1}^{m} \sum_{i=1}^{m} [h_j h_i \rho_j^c B_j^c (c_p^i x_p(k - t_{sci})) + h_j A_j^c x_c(k)] \end{matrix} \right. \tag{25}$$

Here, the delays are independent based upon the time obtained from scheduling approximation:

$$t_{ca1} + t_{sc1} < t_{ca2} + t_{sc2} < \cdots < t_{cam} + t_{scm} < T \tag{26}$$

where the derivative of a candidate Lyapunov function is expressed as:

$$\Delta u(k) = V(k+1) - V(k) \tag{27}$$

and the related Lyapunov function is proposed as:

$$V(k) = X(k)^T P X(k) \tag{28}$$

In terms of the augmented states and the related fuzzy rules:

$$V(k) = \sum\nolimits_{i=1}^{m} \begin{pmatrix} x_{ci} \\ x_{pi} \end{pmatrix}^t P \begin{pmatrix} x_{ci} \\ x_{pi} \end{pmatrix} \tag{29}$$

where each of the fuzzy rules is given as an expression of local delays from current condition from plant towards controller, and vice versa.

$$\begin{bmatrix} x_c \\ x_p \end{bmatrix} = \begin{bmatrix} x_c(k) \\ x_c(k - t_{ca1}) \\ x_c(k - t_{ca2}) \\ \vdots \\ x_c(k - t_{cam}) \\ x_p(k) \\ x_p(k - t_{sc1}) \\ x_p(k - t_{sc2}) \\ x_p(k - t_{sc3}) \\ \vdots \\ x_p(k - t_{scm}) \end{bmatrix} \tag{30}$$

For each rule, there is a delay related to a particular condition to the plant and controller. This is unique on every specific time. In this case, these are associated to a particular relationship of last equation.

$$V(k+1)-V(k)=\begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} - \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix} \tag{31}$$

$$\begin{aligned} V(k+1)-V(k)=&\begin{bmatrix} \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^p B_j^p (c_c^i x_c(k-t_{caj}))) + h_j A_j^p x_p(k)) \\ \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^c B_j^c (c_p^i x_p(k-t_{scj}))) + h_j A_j^c x_c(k)) \end{bmatrix}^T P \\ &\begin{bmatrix} \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^p B_j^p (c_c^i x_c(k-t_{caj}))) + h_j A_j^p x_p(k)) \\ \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^c B_j^c (c_p^i x_p(k-t_{scj}))) + h_j A_j^c x_c(k)) \end{bmatrix} \\ &- \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix} \end{aligned} \tag{32}$$

Therefore,

$$\begin{aligned} V(k+1)-V(k) \;=\;& \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} \\ &- \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix} \end{aligned} \tag{33}$$

Considering the fuzzy system representation:

$$\begin{aligned} V(k+1)-V(k)=&\begin{bmatrix} \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^p B_j^p (c_c^i x_c(k-t_{caj}))) + h_i A_i^p x_p(k)) \\ \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^c B_j^c (c_p^i x_p(k-t_{scj}))) + h_i A_i^c x_c(k)) \end{bmatrix}^T P \\ &\begin{bmatrix} \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^p B_j^p (c_c^i x_c(k-t_{caj}))) + h_i A_i^p x_p(k)) \\ \sum_{j=1}^m \sum_{i=1}^m (h_j h_i(\rho_j^c B_j^c (c_p^i x_p(k-t_{scj}))) + h_i A_i^c x_c(k)) \end{bmatrix} \\ &- \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix} \end{aligned} \tag{34}$$

If only one of the time delays is considered:

$$0 > \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} - \begin{bmatrix} x_c(k) \\ x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \end{bmatrix} \tag{35}$$

Therefore, this can be expressed as follows:

$$\begin{aligned} 0 > &\left\{ \begin{bmatrix} h_j h_i(\rho_j^p B_j^p(c_c^i)) & h_i A_i^p & 0 & 0 \\ 0 & 0 & h_j h_i(\rho_j^c B_j^c(c_p^i)) & h_i A_i^c \end{bmatrix} \begin{bmatrix} x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \\ x_c(k) \end{bmatrix} \right\}^T P \\ &\begin{bmatrix} h_j h_i(\rho_j^p B_j^p(c_c^i)) & h_i A_i^p & 0 & 0 \\ 0 & 0 & h_j h_i(\rho_j^c B_j^c(c_p^i)) & h_i A_i^c \end{bmatrix} \begin{bmatrix} x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \\ x_c(k) \end{bmatrix} \\ &- \begin{bmatrix} x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \\ x_c(k) \end{bmatrix}^T P \begin{bmatrix} x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \\ x_c(k) \end{bmatrix} \end{aligned} \tag{36}$$

and based upon this particular case, state representation is given as:

$$\begin{aligned} 0 > &\begin{bmatrix} x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \\ x_c(k) \end{bmatrix}^T \left[\begin{bmatrix} h_j h_i(\rho_j^p B_j^p(c_c^i)) & h_i A_i^p & 0 & 0 \\ 0 & 0 & h_j h_i(\rho_j^c B_j^c(c_p^i)) & h_i A_i^c \end{bmatrix}^T P \right. \\ &\left. \begin{bmatrix} h_j h_i(\rho_j^p B_j^p(c_c^i)) & h_i A_i^p & 0 & 0 \\ 0 & 0 & h_j h_i(\rho_j^c B_j^c(c_p^i)) & h_i A_i^c \end{bmatrix} - p \right] \begin{bmatrix} x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \\ x_c(k) \end{bmatrix} \end{aligned} \tag{37}$$

since only a single specific delay is possible on the current time. Therefore, only one state condition is available, and it is expressed as before, following an LMI conditions matrix G_j^i:

$$G_j^i = \begin{bmatrix} h_j h_i(\rho_j^p B_j^p(c_c^i)) & h_i A_i^p & 0 & 0 \\ 0 & 0 & h_j h_i(\rho_j^c B_j^c(c_p^i)) & h_i A_i^c \end{bmatrix} \tag{38}$$

The core of current representation is expressed as:

$$\begin{aligned} 0 > &\left[\begin{bmatrix} h_j h_i(\rho_j^p B_j^p(c_c^i)) & h_i A_i^p & 0 & 0 \\ 0 & 0 & h_j h_i(\rho_j^c B_j^c(c_p^i)) & h_i A_i^c \end{bmatrix}^T P \right. \\ &\left. \begin{bmatrix} h_j h_i(\rho_j^p B_j^p(c_c^i)) & h_i A_i^p & 0 & 0 \\ 0 & 0 & h_j h_i(\rho_j^c B_j^c(c_p^i)) & h_i A_i^c \end{bmatrix} - P \right] \end{aligned} \tag{39}$$

and

$$0 > G_j^{i^T} P G_j^i - P \tag{40}$$

If a local fault appears, the system modifies its structure according to Equation (39). However, stability still can be reached, since faults are strictly local and unitary in terms of similar fault presence, and bounded on time. This assumption is quite restricted on real life, albeit possible for incipient faults. Therefore, how to cope in degradable systems due to intermittent faults is out of the scope of this paper. Faulty conditions, as well as inherent structure characteristics, are as well out of the scope of this paper. The focus of current approximation is to establish fault effects into system structure considering time deadlines. Following this restricted condition, it is possible to establish stability requirements following one sensor missing and the corresponding local time delay, since Equation (39) is not affected by the structure of the system. Now, the structure of ρ_j^* is a matrix (N×N) which is coherent on the dimensions of B_j^* as expressed as:

$$\rho_i^* = \left[\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \right] \tag{41}$$

As soon as a local fault appears, ρ_j^* matrix is modified. The local fault is replaced by a one condition as shown as follows:

$$\rho_i^* = \left[\begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \right] \tag{42}$$

5. **Experimental Set Up.** The following experiment is set up to demonstrate how this combination is achievable, in order to make a suitable approximation for time delays managements. The number of periodic tasks is set to 5 and the number of aperiodic tasks is 7. This experiment has been presented, where typical results are shown in terms of time delays only. As for initial conditions, following table presents tasks conditions. Here, time delays are settled per scenario considering local faults and fault-free conditions. Current time delays are used to determine the related control law according to Equation (40). The novelty of this approximation is related to the use of Priority Exchange (PE) in order to bound process scheduling as well as time delays during appearance of local and known faults. Once PE bounds time delays, these are used to tackle system modeling following Equations (20) and (30). In order to guarantee stability by the use of a pre-define control law it is necessary to demonstrate stability through Equations (31) and (41).

Now, based upon plant dynamics, this is given as:

$$A = \begin{bmatrix} 3 & 0 & 3 \\ -4 & -2 & 0.1 \\ 0.1 & 0.3 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0.1 \\ 0.3 \\ 0.2 \end{bmatrix} \quad y = cu \tag{43}$$

Time delays are determined per scenario as shown next:

Time delays =[0.001 0.0000 0.0 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0012 0.0001 0.0003 0.00075 0.0001 0.000101 0.00009 0.00002 0.00002 0.000012 0.00003 0.00001 0.000109
0.003 0.0002 0.001 0.0009 0.00018 0.0001601 0.000109 0.000102 0.0000222 0.0000212 0.000063 0.000061 0.000209
0.004 0.0012 0.00232 0.00101 0.00021 0.0020101 0.000209 0.000302 0.000062 0.0000412 0.000073 0.000101 0.000309
0.005 0.002 0.004 0.0013 0.00031 0.0030101 0.000609 0.000602 0.000082 0.0000712 0.000303 0.000201 0.000609
0.00611 0.00302 0.0055 0.00244 0.00041 0.0040101 0.000809 0.001002 0.000102 0.0000812 0.000503 0.001001 0.001009
0.007 0.006 0.0065 0.0033 0.00101 0.0050101 0.001009 0.003002 0.0001202 0.0005012 0.000703 0.002001 0.003009
0.00811 0.0072 0.0085 0.0044 0.00201 0.0060101 0.003009 0.004002 0.000402 0.0008012 0.000903 0.003001 0.004009
0.0095 0.008 0.0095 0.0066 0.00401 0.0070101 0.0039009 0.006002 0.000602 0.0020012 0.002003 0.005001 0.005009
0.0099 0.009 0.0099 0.008 0.00501 0.0080101 0.006009 0.007002 0.004002 0.0050012 0.005003 0.007001 0.009009
0.010 0.01 0.01000 0.01 0.00701 0.008990101 0.007009 0.030002 0.040002 0.0070012 0.010003 0.070001 0.090009]

where each of the columns is one scenario, and the related time delays are between each sensor, following Equations (8) and (9). The given control design following Equation (40) is expressed as follows, where time delays tend to be constant per scenario:

$$k = \begin{bmatrix} 0.1 & & -2.2 \\ -0.2 & \cdots & -0.1 \\ -1.2 & & -0.2 \end{bmatrix}$$

Fuzzy variables, as well as the number of rules, are determined following the approach by [10], in which the final approximation is determined by a similar error, following time delays approach and the related system response. Now the response of system according to first output is shown in Figure 5.

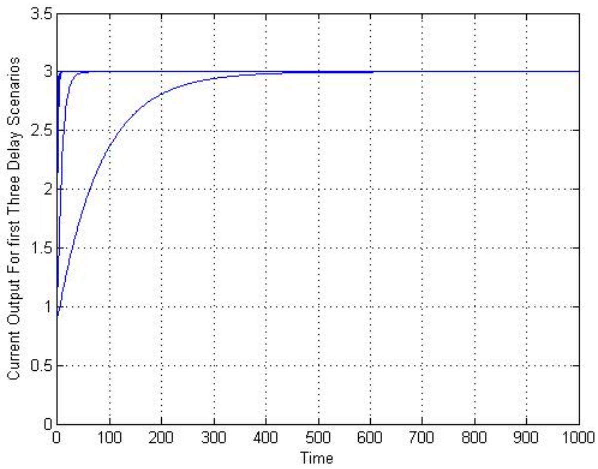

Figure 5. Systems response for those different time delay scenarios

It is shown that even in the case of fault scenario, time response is settled adequately as well as local faults following the proposed mathematical procedure of Equations (21)-(41).

6. Conclusions.

In an NCS, time delays can be modeled using real-time dynamic scheduling algorithms; however, the resulting delays are time varying and stationary, and therefore, related local control laws need to be designed according to this characteristic. Time integration is the key global issue to be taken into consideration. Global stability is reached by the use of Takagi-Sugeno fuzzy control design, where a nonlinear combination is followed by a current sequence of states, which are partially delayed due to the communication behavior.

Fault conditions are overcome through two strategies: (1) by scheduling approximation and (2) by control design, considering inherent time delays. In this case, this approach takes the sole effect of faults as a result of certain local malfunction, which is bounded to this. Fault nature, as well as fault treatment, is out of the scope of this paper, since it is necessary to establish a reliability study.

Fault status is considered to be only local and bounded, where the unique effect is the absence of current measure, as well as inherent time delays. The first condition is preserved by the use of ρ_j^*, and local time delays appear to be feasible by the use dynamic scheduling, since these are already known as part of time manipulation onto tasks belonging to studied system.

The use of a dynamic scheduling approach allows the system to be predictable and bounded, and therefore, time delays can be modeled in these terms. Moreover, the resulting dynamic representation tackles the inherent switching per scenario. However, this approach has the main drawback that context switch may be invoked every time that an aperiodic task takes place, and it is possible to be executed. In this case, inherent time delays to this action are taken into account to be processed as uncertainties. Thus, in this paper, time delays and the appearance of local faults are taken to develop a fuzzy model of the controller, capable of solving this nonlinear behavior. This control approach is feasible, since local faults are known as well as bounded time delays.

Acknowledgment. The authors acknowledge the support of UNAM-PAPIIT IN103310, ICyTDF PICCO 10-53 and Mr. Adrian Duran for his valuable help.

REFERENCES

[1] H. Benítez-Pérez and F. García-Nocetti, *Reconfigurable Distributed Control*, Springer Verlag, Germany, 2005.
[2] H. Benítez-Pérez, Real-time distributed control a diverse approach for nonlinear problem, *Nonlinear Analysis: Hybrid Systems and Applications*, vol.2, no.2, pp.474-490, 2008.
[3] M. Blanke, M. Kinnaert, J. Lunze and M. Staroswiecki, *Diagnosis and Fault Tolerant Control*, Springer, Germany, 2003.
[4] Buttazzo, *Real Time Systems*, Springer, Germany, 2004.
[5] E. Fridman and U. Shaked, Delay-dependent stability and H_∞ control: Constant and time-varying delays, *International Journal Control*, vol.76, no.1, pp.48-60, 2003.
[6] R. Izadi-Zamanabadi and M. Blanke, A ship propulsion system as a benchmark for fault-tolerant control, *Control Engineering Practice*, vol.7, pp.227-239, 1999.
[7] J. Jiang and Q. Zhao, Reconfigurable control based on imprecise fault identification, *Proc. of the American Control Conference, IEEE*, San Diego, CA, USA, 1999.
[8] F. Lian, J. Moyne and D. Tilbury, Network design consideration for distributed control systems, *IEEE Transactions on Control Systems Technology*, vol.10, no.2, pp.297-307, 2002.
[9] Y. Liu, *Real Time Systems*, Wiley, New York, 2000.
[10] P. E. Méndez-Monroy and H. Benítez-Pérez, Supervisory fuzzy control for networked control systems, *ICIC Express Letters*, vol.3, no.2, pp.233-240, 2009.
[11] J. Nilsson, *Real-Time Control with Delays*, Ph.D. Thesis, Lund Institute of Technology, Sweden, 1998.
[12] P. Quiñones-Reyes, H. Benítez-Pérez, F. Cárdenas-Flores and J. Ortega-Arjona, Reconfigurable fuzzy takagi sugeno model predictive control networked control (magnetic levitation case study), *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol.224, no.8, pp.1022-1032, 2010.
[13] K. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, Wiley, New York, 2001.
[14] H. Thompson, Wireless and Internet communications technologies for monitoring and control, *Control Engineering Practice*, vol.12, pp.781-791, 2004.
[15] G. C. Walsh, H. Ye and L. Bushnell, Stability analysis of networked control systems, *Proc. of American Control Conference*, San Diego, CA, USA, 1999.
[16] Y. Wang and Z. Sun, H-inf control of networked control systems via LMI approach, *International Journal of Innovative Computing, Information and Control*, vol.3, no.2, pp.343-352, 2007.
[17] W. Zhang, *Stability Analysis of Networked Control Systems*, Ph.D. thesis, Case Western Reserve University, 2001.
[18] H. Zhang, D. Yang and T. Chai, Guaranteed cost networked control for T–S fuzzy system switch time delays, *IEEE Transactions on Systems and Cybernetics part C: Applications and Reviews*, vol.37, no.2, pp.160-172, 2007.
[19] X. Zhu, C. Hua and S. Wanf, State feedback control design of networked control systems with time delay in the plant, *International Journal of Innovative Computing, Information and Control*, vol.4, no.2, 2008.