Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 927878, 9 pages
doi:10.1155/2012/927878

Research Article

Fuzzy Networked Control Systems Design Considering Scheduling Restrictions

H. Benítez-Pérez,[1,2] A. Benítez-Pérez,[2,3] J. Ortega-Arjona,[2,4] and O. Esquivel-Flores[1,2]

[1] *Departamento de Ingeniería de Sistemas Computacionales y Automatización, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Apdo. Postal 20-726. Del. A. Obregón, 01000 México, DF, Mexico*
[2] *Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México, DF, Mexico*
[3] *Cecyt 11, Instituto Politecnico Nacional, Avenida de los Maestros S/N, México, DF, Mexico*
[4] *Departamento de Matemáticas, Facultad de Ciencias, Universidad Nacional Autónoma de México, DF, Mexico*

Correspondence should be addressed to H. Benítez-Pérez, hector@uxdea4.iimas.unam.mx

Received 23 May 2012; Revised 17 July 2012; Accepted 17 October 2012

Academic Editor: Sendren Sheng-Dong Xu

Nowadays network control systems present a common approximation when connectivity is the issue to be solved based on time delays coupling from external factors. However, this approach tends to be complex in terms of time delays. Therefore, it is necessary to study the behavior of the delays as well as the integration into differential equations of these bounded delays. The related time delays needs to be known a priory but from a dynamic real-time behavior. To do so, the use of priority dynamic Priority exchange scheduling is performed. The objective of this paper is to show a way to tackle multiple time delays that are bounded and the dynamic response from real-time scheduling approximation. The related control law is designed considering fuzzy logic approximation for nonlinear time delays coupling, where the main advantage is the integration of this behavior through extended state space representation keeping certain linear and bounded behavior and leading to a stable situation during events presentation by guaranteeing stability through Lyapunov.

1. Introduction

Nowadays real-time restrictions are the most certain definitions in terms of time delays where general considerations tend to be periodic and repeatable.

The control design and stability analysis of network-based control systems (NCSs) have been studied in recent years, based upon codesign strategy [1]. The main advantages of this kind of systems are their low cost, small volume of wiring, distributed processing, simple installation, maintenance, and reliability.

In a NCS, one of the key issues is the effect of network-induced delay in the system performance. The delay can be constant, time varying, or even random; this depends on the scheduler, network type, architecture, operating systems, and so forth. One strategy to be followed is the codesign since it takes both desired procedures to be followed. Nilsson analyzes several important facets of NCSs. Nilsson [2] introduces models for the delays in NCS, first as a fixed delay, afterward as an independently random, and finally like a Markov process. The author introduces optimal stochastic control theorems for NCSs based upon the independently random and Markovian delay models. In [3], Walsh et al. 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*), which is the longest time after which a sensor should transmit a data. Walsh et al. [3] derived bounds of the MATI such that the NCS is stable. This MATI ensures that the Lyapunov function of the system under consideration is strictly decreasing at all times. In [4], Zhang et al. extend the work of Walsh, they developed a theorem which ensures the decrease of a Lyapunov function for a discrete-time LTI system at each sampling instant, using two different bounds. These results are less conservative than those of Walsh, because they do not require the system

Lyapunov function to be strictly decreasing at all time. Nevertheless, a number of different linear matrix inequality (LMI) tools for analyzing and designing optimal switched NCSs are introduced.

Alternatively Zhu [5] takes into consideration both the network-induced delay and the time delay in the plant a controller design method is proposed by using the delay-dependent approach. An appropriate Lyapunov functional candidate is utilized to obtain a memoryless feedback controller; this is derived by solving a set of Linear Matrix Inequalities (LMIs). In [6], Wang and Sun, model 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, the H∞ state feedback controller can be constructed via a set of LMIs. Recently Fridman and Shaked [7] introduce a new (descriptor) model transformation for delay-dependent stability for systems with time-varying delays in terms of LMIs, and they also refine recent results on delay-dependent H∞ control and extend them to the case of time-varying delays. Based upon this review, this paper defines a model that integrates the time delays for a class of nonlinear system, therefore, this paper presents Fuzzy Control for NCSs [4, 8] considering time delay induced by the computer network as result of online reconfiguration the stability analysis is revised as well.

Since NCS is modified according to time delays, reconfiguration is a transition that modifies the structure of a system so it changes its representation of states. Here, it is used as a feasible approach for time delay modification.

In control systems, several modeling strategies for managing time delay within control laws have been studied by different research groups. Nilsson [2] proposes the use of a time delay scheme integrated to a reconfigurable control strategy, based on a stochastic methodology. Jiang and Zhao [9] describe how time delays are used as uncertainties, which modify pole placement of a robust control law. Izadi-Zamanabadi and Blanke [10] present an interesting case of fault tolerant control approach related to time delay coupling. Blanke et al. [11] study reconfigurable control from the point of view of structural modification, establishing a logical relation between dynamic variables and the respective faults. Finally, Thompson [12] and Benítez-Pérez and García-Nocetti [13] 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.

Normally, when a fault occurs during the operation of a system, a respective fault tolerance strategy is applied. However, applying such a fault tolerance strategy is not enough to maintain the performance of the system, since dynamic conditions are modified. Therefore, it is seems necessary to take into account current conditions in order to keep system performance, even degraded. Thus, this paper proposes a novel technique based on Fuzzy control and considering bounded variable time delays.

Table 1: First example for Priority Exchange.

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

The objective of this paper 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 novelty of this approximation is to guarantee schedulability as well as stability in the presence of bounded time delays. This is feasible since time delays are bounded according to scheduler response.

2. Scheduling Approximation

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

$$c_1 \longrightarrow c_n T_1 \longrightarrow T_n, \tag{1}$$

where priority is given as the well known Earliest Deadline First [14] algorithm which established as the process with the closest deadline has the most important priority. However, when a nonperiodic tasks appear it is necessary to deploy other algorithms to cope with concurrent conditions. To do so, Priority Exchange algorithm is pursuit in order to manage spare time from EDF algorithm. Priority exchange [15] algorithm 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 behaviour 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 time, as well as period, is given in terms of units that are entered. Remember that 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 upon this dynamic scheduling algorithm, time delays are given as current calculus in terms of task ordering. In this case, every time that scheduling algorithm takes place global time delays are modified in the short and long term. This behavior allows time delays to be known and bounded for different periods of time since current and future responses are established. On the other hand, if any aperiodic event would take place, this will be considered in terms of the server to be attended in a global periodic manner with the related time delay cost. For instance, follow next example where four tasks are settled and two aperiodic tasks take place at different times, giving different events with different time delays (see Table 2).

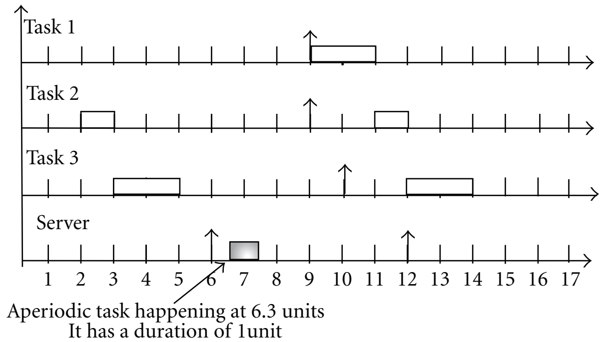

Figure 1: Related organization for PE with respect to Table 1.

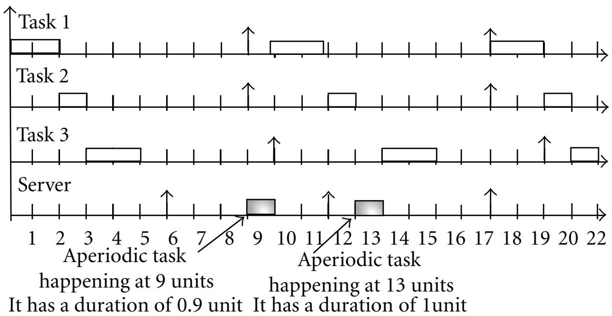

Figure 2: Task organization considering second example for PE.

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 (ap2)	1.0	It occurs at 13

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

Now from this resulting ordering different tiny time delays are 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 beforehand 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 amongst dynamical systems. These relations are the actual and possible delays bounded as marked limit of possible and current scenarios. Then, delays may be expressed as local summations with a high degree of certainty for each specific scenario. In this case, if any new event takes place its response would be delayed until the server would place sometime for its requirements, giving the system a guarantee in terms of time delays and current response.

Now, in terms of this last example, during second scenario total delay is given as follows:

$$\begin{aligned} \text{Total delay} = {} & \text{consumption time delay aperiodic task1} \\ & + \text{consumption time delay task1} + \text{tsc2} \\ & + \text{consumption time delay task2} \\ & + \text{consumption time delay aperiodic task2} \\ & + \text{consumption time delay task3}. \end{aligned} \tag{2}$$

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. Moreover, consumption_time_delay_task1, consumption_time_delay_task2, and consumption_time_delay_task3 are related to the actual time delays from Figure 3 when one particular scenario is presented. The same situation is presented with consumption_time_delay_aperiodic_task2. This simple example shows how total time delays play a key role in the dynamical system; however, these are no monolithic since are composed through different local and dynamic delays.

Since aperiodic as well as sporadic events are capable to be attended in terms of a virtual server per node involved on the network giving a bounded response, the resulting behaviour is only dependant on inherent bounded and systematic time delays that can be aggregated in laps. Now, the computer network is only dependant on the synchronization of the network, which is a topic that is out of the scope of this paper and to be reviewed in future work.

The important issue to be determined in this section is that communicating time delays are to be known and bounded even in sporadic situations. Since this modelling is possible, what is left is how to incorporate the aggregate delays (either local or global) onto the dynamic modelling of the system. This strategy is proposed thorough Fuzzy Control since this technique provides the necessary elements to guarantee current global stability even in conditions of sporadic time delays since these are bounded according to the use of virtual server.

3. Fuzzy Control Design Considering Time Delays

Having defined time delays as result of scheduling approximation, several scenarios are potentially presented following this time delay behavior since this is bounded. 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, Fuzzy Control strategy is based upon Takagi-Sugeno. Therefore based upon Fuzzy Control systems [16] stays as

$$1 < i < m, \quad 1 < j < m, \quad \mu_{ij}(x_i(t)), \tag{3}$$

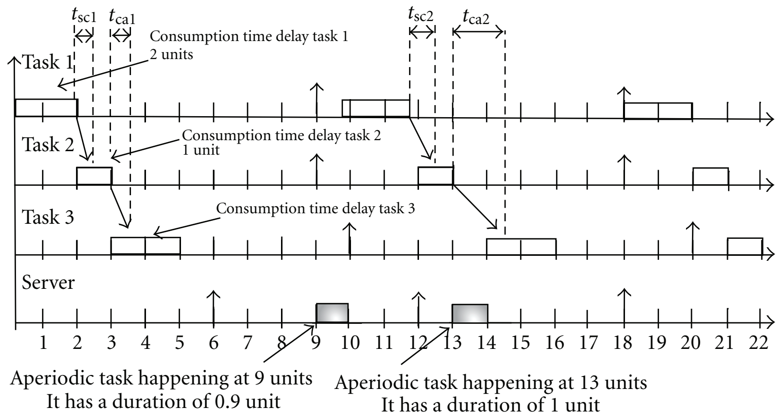

Figure 3: Related time delays are depicted according to both scenarios.

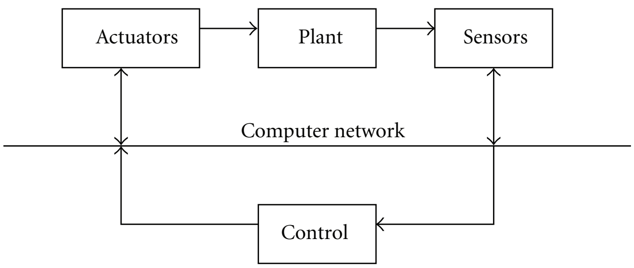

Figure 4: Network control system diagram.

where x are the states, m is the number of inputs, and μ is the related membership function. One has

$$g_j = \prod_{i=1}^{m} \left\{ \mu_{ij} \left[x_i(k) \right] \right\}, \tag{4}$$

$$h_j = \frac{g_j}{\sum_{j=1}^{m} g_j}, \tag{5}$$

$$x(k+1) = \sum_{j=1}^{m} \left[h_j \left\{ A_j^p x(k) + B_j^p u(k) \right\} \right], \tag{6}$$

where $x(k)$ and $u(k)$ are the state and input vectors, and A_j^p and B_j^p are the plant representation per scenario according to current time delays following Figure 4.

Now, considering current time delays as t_{caj} which is current time delay from controller to actuator and t_{scj} which is current time delay from sensor to controller. In here, current time delays are local aggregations of current behavior from scheduling strategy in any condition regardless of the event as long as this is prevented onto virtual server processes. One has

$$\begin{aligned} x_p(k+1) &= \sum_{j=1}^{m} \left[h_j \left\{ A_j^p x_p(k) + B_j^p u_p\left(k - t_{caj}\right) \right\} \right], \\ x_c(k+1) &= \sum_{j=1}^{m} \left[h_j^p \left\{ A_j^c x_c(k) + B_j^c u_c\left(k - t_{scj}\right) \right\} \right], \end{aligned} \tag{7}$$

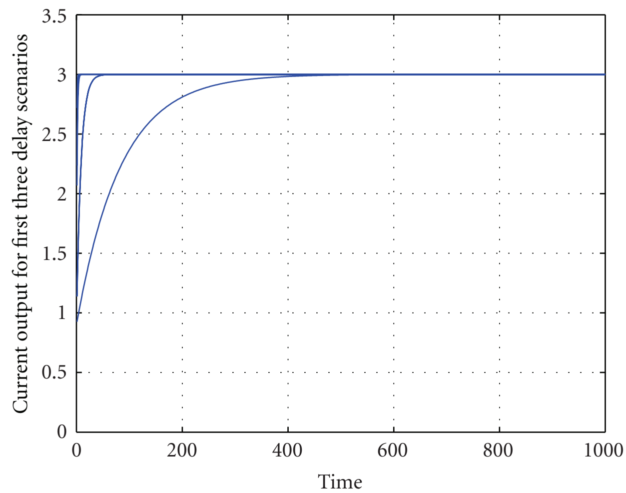

Figure 5: Systems response to those different time delay scenarios.

where A_j^c and B_j^c are the controller representation per scenario, $x_p(k+1)$ and $u_p(k+1)$ are state vector and input vector of the plant, and $x_c(k+1)$ and $u_c(k+1)$ are state vector and input vector of the controller.

From [17] remember that time delay representation in terms of discrete observe the following equations:

$$\begin{aligned} B_j^p &= \sum_{i=1}^{l_p} \left[\int_{t_i}^{t_{i+1}} B_j^c e^{A_j^c t} dt \right], \\ B_j^c &= \sum_{i=1}^{l_c} \left[\int_{t_i}^{t_{i+1}} B_j^p e^{A_j^c t} dt \right]. \end{aligned} \tag{8}$$

Remember that l_p and l_c are the total number of local time delays that appears per scenario. These are defined in last section as local time delays that can be aggregated as in (1) or they maybe presented as shown in (8). In any case final result is shown in (10) and (11). One has

$$y_{p_i} = c_p^i x_p(k), \quad y_{c_i} = c_c^i x_c(k), \tag{9}$$

where l_p and l_c are the number of local time delays; c_c^i and c_p^i are the gains related to observable states; the outputs are gathering as

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

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

$$u_p\left(k - t_{caj}\right) \longrightarrow y_c = c_c x_c(k), \tag{12}$$

$$u_c\left(k - t_{scj}\right) \longrightarrow y_p = c_p x_p(k), \tag{13}$$

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

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

From last equation, the related dynamics are expressed A_j^C as B_j^C and C_c^j where j is the index with respect to each scenario. These scenarios are the related events presented in last section and are the result of local time delays and possible use of virtual server. In any case, (16) shows the holistic representation of the plant in conditions of potential time delays as well as the current dynamic modifications result from each scenario. One has

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

from state of the controller

$$x_c(k+1) = \sum_{i=1}^{m}\left[h_i\{A_i^c x_c(k) + B_i^c u_c(k - t_{sci})\}\right],$$

$$x_c(k+1) = \sum_{i=1}^{m}\left[h_i\left\{A_i^c x_c(k) + B_i^c\left(\sum_{j=1}^{m} h_j\left[c_p^j x_p\left(k - t_{scj}\right)\right]\right)\right\}\right] \tag{17}$$

and in terms of the plant x_p

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

$$x_p(k+1) = \sum_{j=1}^{m}\sum_{i=1}^{m}\left[h_j h_i\left[B_i^p\left(c_c^i x_c(k - t_{cai})\right)\right] + h_j A_j^p x_p(k)\right]. \tag{18}$$

For x_c,

$$x_c(k+1) = \sum_{j=1}^{m}\left(h_j\left\{A_j^c x_c(k) + h_i B_j^c\left(\sum_{j=1}^{m} h_j\left(c_p^i x_p\left(k - t_{scj}\right)\right)\right)\right\}\right),$$

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

$$x_c(k+1) = \sum_{j=1}^{m}\sum_{i=1}^{m}\left[h_j h_i B_j^c\left(c_p^i x_p(k - t_{sci})\right) + h_j A_j^c x_c(h)\right], \tag{19}$$

where the t_{caj} is current time delay from controller to actuator and t_{scj} is current time delay from sensor to controller. Moreover, A_j^c is the related dynamic Matrix of Control law. Now, the second main point presented in this work is the following: since the delays are bounded and can be known it is possible to develop a dynamic representation by using augmented states in terms of current control law and the related state space representation.

$$X = \begin{bmatrix} x_c \\ x_p \end{bmatrix},$$

$$\begin{matrix} x_c(k+1) \\ x_p(k+1) \end{matrix}$$

$$= \begin{cases} \sum_{j=1}^{m}\sum_{i=1}^{m}\left[h_j h_i\left[B_j^p\left(c_c^i x_c(k - t_{cai})\right)\right] + h_j A_j^p x_p(k)\right] \\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left[\left(h_j h_i B_j^c\left(C_p^i x_p(k - t_{sci})\right) + h_j A_j^c x_c(k)\right)\right]. \end{cases} \tag{20}$$

Now, the delays (t_{cai}, t_{sci}) are independent based upon the time obtained from scheduling approximation. This condition is very important for two reasons; firstly time delays are strictly local and may be aggregated differently per scenario or event and secondly these are bounded to inherent sampling time of dynamic benchmarking. Therefore, any aggregation must be bounded as presented.

$$t_{ca1} + t_{sc1} < t_{ca2} + t_{sc2} < \cdots < t_{cam} + t_{scm} < T. \tag{21}$$

Now, in terms of the stability which is necessary to guarantee system response in several conditions, it is pursued the use of classical Lyapunov candidate since one the main conditions is system bounded response as linear inherent behavior. Therefore, the derivative of a candidate Lyapunov function is expressed as

$$\Delta u(k) = V(k+1) - V(k), \tag{22}$$

and the related Lyapunov function is proposed as

$$V(k) = X(k)^T P X(k). \tag{23}$$

Now in terms of the augmented states and the related fuzzy rules

$$V(k) = \begin{bmatrix} x_c \\ x_p \end{bmatrix}^T P \begin{bmatrix} x_c \\ x_p \end{bmatrix}, \tag{24}$$

where each of the fuzzy rules is given as an expression of local delays (which are the results of local time delays that can be aggregate per event) from current condition from plant towards controller and vice verse. One has

$$\begin{bmatrix} x_c \\ x_p \end{bmatrix} = \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}. \tag{25}$$

Now for each rule, it exists a delay related to a particular condition (which is expressed as event in terms of Section 2) involving the plant and controller. This delay is unique on every specific time. In this case, these are associated to a particular relationship of last equation.

$$\begin{aligned} V(k+1) - V(k) &= \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} \\ &\quad - \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} &V(k+1) - V(k) \\ &= \begin{bmatrix} \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^P\left(c_c^i x_c\left(k-t_{caj}\right)\right)\right) + h_j A_j^P x_p(k)\right) \\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^c\left(c_p^i x_p\left(k-t_{scj}\right)\right)\right) + h_j A_j^c x_c(k)\right) \end{bmatrix}^T P \\ &\quad \times \begin{bmatrix} \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^P\left(c_c^i x_c\left(k-t_{caj}\right)\right)\right) + h_j A_j^P x_p(k)\right) \\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^c\left(c_p^i x_p\left(k-t_{scj}\right)\right)\right) + h_j A_j^c x_c(k)\right) \end{bmatrix} \\ &\quad - \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}. \end{aligned} \tag{26}$$

Remember h_j and h_i are defined following (5). Therefore

$$\begin{aligned} V(k+1) - V(k) &= \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} \\ &\quad - \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}^T \\ &\quad \times P \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}. \end{aligned} \tag{27}$$

Now considering the fuzzy system representation in terms of local time delays as well as local plants and control laws,

$$\begin{aligned} &V(k+1) - V(k) \\ &= \begin{bmatrix} \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^P\left(c_c^i x_c\left(k-t_{caj}\right)\right)\right) + h_i A_i^P x_p(k)\right) \\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^c\left(c_p^i x_p\left(k-t_{scj}\right)\right)\right) + h_i A_i^c x_c(k)\right) \end{bmatrix}^T \\ &\quad \times P \begin{bmatrix} \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^P\left(c_c^i x_c\left(k-t_{caj}\right)\right)\right) + h_i A_i^P x_p(k)\right) \\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left(h_j h_i\left(B_j^c\left(c_p^i x_p\left(k-t_{scj}\right)\right)\right) + h_i A_i^c x_c(k)\right) \end{bmatrix} \\ &\quad - \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}. \end{aligned} \tag{28}$$

Now if only one of the time delays is considered. This condition is possible since time delays are bounded and strictly less than sampling time of dynamic system. Therefore at any case following inequality is always kept true. One has

$$0 > \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} - \begin{bmatrix} x_c(k) \\ x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \end{bmatrix}. \quad (29)$$

Therefore this may be expressed as follows:

$$\begin{aligned} 0 > &\left\{ \begin{bmatrix} h_j h_i \left(B_j^P (c_c^i)\right) & h_i A_i^P \ \ 0 \ \ 0 \\ 0 & 0 \ \ h_j h_i \left(B_j^c \left(c_p^i\right)\right) \ \ h_i A_i^c \end{bmatrix} \times \begin{bmatrix} x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \\ x_c(k) \end{bmatrix} \right\}^T \\ &\times P \begin{bmatrix} h_j h_i \left(B_j^P (c_c^i)\right) & h_i A_i^P \ \ 0 \ \ 0 \\ 0 & 0 \ \ h_j h_i \left(B_j^c \left(c_p^i\right)\right) \ \ h_i A_i^c \end{bmatrix} \times \begin{bmatrix} x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \\ x_c(k) \end{bmatrix} \\ &- \begin{bmatrix} x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \\ x_c(k) \end{bmatrix}^T P \begin{bmatrix} x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \\ x_c(k) \end{bmatrix}. \end{aligned} \quad (30)$$

and based upon this particular case, state representation may given as

$$\begin{aligned} 0 > &\begin{bmatrix} x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \\ x_c(k) \end{bmatrix}^T \\ &\times \left[\begin{bmatrix} h_j h_i \left(B_j^P (c_c^i)\right) & h_i A_i^P \ \ 0 \ \ 0 \\ 0 & 0 \ \ h_j h_i \left(B_j^c \left(c_p^i\right)\right) \ \ h_i A_i^c \end{bmatrix}^T \times P \begin{bmatrix} h_j h_i \left(B_j^P (c_c^i)\right) & h_i A_i^P \ \ 0 \ \ 0 \\ 0 & 0 \ \ h_j h_i \left(B_j^c \left(c_p^i\right)\right) \ \ h_i A_i^c \end{bmatrix} - P \right] \\ &\times \begin{bmatrix} x_c\left(k - t_{caj}\right) \\ x_p(k) \\ x_p\left(k - t_{scj}\right) \\ x_c(k) \end{bmatrix}. \end{aligned} \quad (31)$$

Because only one specific delay is possible on current time, only one state condition is available and is expressed as before following LMI conditions matrix; G_j^i.

$$G_j^i = \begin{bmatrix} h_j h_i \left(B_j^P (c_c^i)\right) & h_i A_i^P \ \ 0 \ \ 0 \\ 0 & 0 \ \ h_j h_i \left(B_j^c \left(c_p^i\right)\right) \ \ h_i A_i^c \end{bmatrix}. \quad (32)$$

The core of current representation is expressed as

$$0 > \left[\begin{bmatrix} h_j h_i \left(B_j^P (c_c^i)\right) & h_i A_i^P \ \ 0 \ \ 0 \\ 0 & 0 \ \ h_j h_i \left(B_j^c \left(c_p^i\right)\right) \ \ h_i A_i^c \end{bmatrix}^T \times P \begin{bmatrix} h_j h_i \left(B_j^P (c_c^i)\right) & h_i A_i^P \ \ 0 \ \ 0 \\ 0 & 0 \ \ h_j h_i \left(B_j^c \left(c_p^i\right)\right) \ \ h_i A_i^c \end{bmatrix} - P \right], \quad (33)$$

$$0 > {G_j^i}^T P G_j^i - P. \quad (34)$$

Remember that in terms of LMI this consideration should be globally stable in terms of index performance.

4. Experimental Setup

The following is a setup to demonstrate how achievable this combination to make a suitable approximation for time delays managements is. The number of periodic tasks is equal to 5 and the number of aperiodic tasks is 7. Following table presents tasks conditions.

Now based upon plant dynamics this is given as

$$\begin{aligned} A &= \begin{bmatrix} -0.3 & 0 & 3 \\ -4 & -2 & 0.1 \\ 0.1 & 0.3 & -1 \end{bmatrix}, \\ B &= \begin{bmatrix} 0.1 \\ 0.3 \\ 0.2 \end{bmatrix}, \\ \dot{x} &= Ax + Bu, \\ y &= cu. \end{aligned} \quad (35)$$

Time delays are determined per scenario with each local delay considered. The number of scenarios is 13 (as shown in Figure 6), where each columns is one scenario and the related time delays are amongst each sensor following (8). The given control design following (34) is expressed as follows, where time delays tend to be constant per scenario:

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]

Figure 6

$$k = \begin{bmatrix} 0.1 & 0.11 & 0.001 & 0.01 & 0.1 & 0.01 & -0.001 & -0.01 & 0.0 & -0.0 & -2.2 \\ -0.2 & -0.11 & 0.1 & -1.1 & -0.2 & -0.5 & -1.1 & -0.9 & 0.1 & -0.01 & -0.1 \\ -1.2 & -1.1 & -1.1 & -0.2 & -0.13 & -2.1 & -0.9 & -2.1 & -1.01 & -1.1 & -0.2 \end{bmatrix}. \quad (36)$$

Fuzzy variables as well as the number of rules are determined following Méndez-Monroy and Benítez-Pérez [16]; here final approximation is determined by similar error following time delay approach and the related system response. Now the response of system according to First output is shown in Figure 5.

5. Conclusions

Current time delays can be modeled using real-time dynamic scheduling algorithms; however the resulting delays are time varying and stationary, therefore related local control laws need to be designed according to this characteristic and 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 nonlinear combination is followed by current situation of the states which are partially delayed due to communication behavior.

The main contribution on this paper is the capability to determine local time delays that can be aggregated per event since a scheduling algorithm contributes to bound time response. Therefore Fuzzy Control may be attractive to guarantee global stability since any condition is bounded to be less than sampling period at the worst case scenario with no loose of generality.

The use of dynamic scheduling approximation allows the system to be predictable and bounded; therefore, time delays can be modeled in these terms. Moreover, the resulting dynamic representation tackles the inherent switching per scenario. This approximation has the main drawback that context switch may be invoked every time a periodic 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.

Acknowledgments

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, "Real-time distributed control: a fuzzy and model predictive control approach for a nonlinear problem," *Nonlinear Analysis: Hybrid Systems*, vol. 2, no. 2, pp. 474–490, 2008.

[2] J. Nilsson, *Real-time control with delays [Ph.D. thesis]*, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.

[3] G. C. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," in *Proceedings of the American Control Conference (ACC '99)*, pp. 2876–2880, San Diego, Calif, USA, June 1999.

[4] H. Zhang, D. Yang, and T. Chai, "Guaranteed cost networked control for T-S fuzzy systems with time delays," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 37, no. 2, pp. 160–172, 2007.

[5] X. Zhu, C. Hua, and S. Wang, "State feedback controller design of networked control systems with time delay in the plant," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 2, pp. 283–290, 2008.

[6] Y. Wang and Z. Sun, "H∞ control of networked control systems via LMI approach," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 2, pp. 343–352, 2007.

[7] E. Fridman and U. Shaked, "Delay-dependent stability and H∞ control: constant and time-varying delays," *International Journal of Control*, vol. 76, no. 1, pp. 48–60, 2003.

[8] K. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, John Wiley & Sons, New York, NY, USA, 2001.

[9] J. Jiang and Q. Zhao, "Reconfigurable control based on imprecise fault identification," in *Proceedings of the American Control Conference (ACC '99)*, vol. 1, pp. 114–118, San Diego, Calif, USA, June 1999.

[10] R. Izadi-Zamanabadi and M. Blanke, "A ship propulsion system as a benchmark for fault-tolerant control," *Control Engineering Practice*, vol. 7, no. 2, pp. 227–239, 1999.

[11] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault Tolerant Control*, Springer, 2003.

[12] H. A. Thompson, "Wireless and internet communications technologies for monitoring and control," *Control Engineering Practice*, vol. 12, no. 6, pp. 781–791, 2004.

[13] H. Benítez-Pérez and F. García-Nocetti, *Reconfigurable Distributed Control*, Springer, 2005.

[14] Y. Liu, *Real Time Systems*, Wiley, 2000.

[15] Buttazzo, *Real Time Systems*, Springer, 2004.

[16] P. E. Méndez-Monroy and H. Benítez-Pérez, "Supervisory fuzzy control for networked control systems," *International Journal Innovative Computing, Information and Control Express Letters*, vol. 3, no. 2, pp. 233–240, 2009.

[17] H. Benitez-Pérez, J. Ortega-Arjona, F. Cardenas-Flores, and P. Quiñones-Reyes, "Reconfiguration control strategy using Takagi-Sugeno model predictive control for network control systems—a magnetic levitation case study," *Proceedings of the Institution of Mechanical Engineers I*, vol. 224, no. 8, pp. 1022–1032, 2010.