Reconfigurable Fuzzy Takagi Sugeno Networked Control using Cooperative Agents over a Dsitributed environment

Ramírez-Gonzalez T.+++, Quiñones-Reyes P.++, Benítez-Pérez H.*, Laureano-Cruces A.+++ and García-Nocetti, F.+
+++Departamento de Sistemas, Universidad Autonoma Metropolitana-Azcapotzalco, Av. San Pablo 180, Col. Reynosa Tamaulipas, CP. 02200, Del. Azc., México DF, MEXICO.
++Instituto Tecnológico de Jiquilpan, Av. Tecnológico, S/N, CP 59510, Jiquilpan, Michoacán, MEXICO.
*+Departamento de Ingeniería de Sistemas Computacionales y Automatización, IIMAS, UNAM, Apdo. Postal 20-726, Del. A. Obregón, México D. F., CP. 01000, MEXICO.

Abstract: Nowadays dynamic behaviour performed by a computer network system shows the posibility to be addressed from the perspective of a control system. This paper discusses the use of Fuzzy Takagi Sugeno real time control with hardware-in-the-loop (HIL) magnetic levitator (maglev) using xPC Target. Here xPC Targe is used as operating environment for real time processing and to connect a computer network system. In that respect, this paper proposes a control reconfiguration based upon a distributed system strategy from the definition of a Takagi-Sugeno approach, considering computer network reconfiguration. Several stages are studied, how computer network takes place as well as how control techniques are modified using Fuzzy Takagi-Sugeno Control.

1. Introduction

Control reconfiguration is presented as an available approach for fault coverage in order to keep system performance. In here reconfiguration is pursued as response of time delay modification rather than fault appearance although this is the basis for control reconfiguration. This paper is focused into reconfigurable control law due to the presence of local faults and time delays as consequences from distributed system iterations.

To define the communication network performance, the use of several Target is necesary. This strategy achieves network implementation based on message passing, that are based on the real-time workshop, xPC Target and Simulink toolboxes from MATLAB by MathWorks Inc. Two kind of computer networks are used, CANbus and Ethernet. These present no further time delays difference because network size is quite small.

Four computers (one host and three targets) and a compatible data acquisition card are required to provide interface between the software and the hardware to be controlled (a nonlinear, open loop unstable, and time varying maglev device). Measured signals can be displayed in real time or saved for later analysis.

Several strategies for managing time delay within control laws have been studied by different research groups. For instance, [1] proposes the use of a time delay scheme integrated to a reconfigurable control strategy based upon a stochastic methodology.

In this paper reconfiguration is performed by the use of cooperative agents communicating between each other reaching an agreement through a blackboard.

The control algorithm is designed and implemented by the Matlab/Simulink software package (Release 14) with Real-Time Workshop and xPC Target Toolbox. These tools are able to automatically generate stand-alone real-time applications from the Simulink models that run on the so-called Target PC, while their development is carried out on separate host computer (Host PC). The Target PC's are based on a special real-time operating system (Real-Time Kernel) and they communicate with the Host PC through serial link or network connection.

The objective of this paper is to present a strategy for control reconfiguration based upon time delay knowledge through the use of cooperative agents and a blackboard within a distributed system environment considering the magnetic levitation challenge in a real time environment using xPC

Target [2] connected to CANBUS and Ethernet.

2. Structural Reconfiguration Algorithm

The goal of this algorithm is to switch computing tasks over different frequencies and priorities, in other words, there is no a static or predefined frequency to transmit.

A distributed system has several tasks working on a local way, each of them have a specific work and the collaboration amongst them produces a valid result. In this case each module is an agent, that produces scheduling algorithm, assigning one adequate frequency per tasks. To know if the frequency is adequate, the agent makes a selection supported in the information of the blackboard, and sent a message to the others.

2.1 Cognitive Analysis

The cognitive analysis shows that each module have two agents one autonomous (is who takes the decisions), an the other is no autonomous (is who executes the decision of the autonomous agent). The goal of both can be active or passive, an it means than can change their environment or not. At the design time of the cooperative system, some questions arise in order to perceive their environment.

Autonomous Agent

Scheduler

Kind of the goal: active. (it means that must change their environment)
Goal: schedule the tasks with respect to their own bandwidth.
Perceptions
Action that perceives:
(a) Select the adequate frequency in order to transmit the task, based on the available frequencies from the blackboard and from their own bandwidth.
(b) Updates the Blackboard taking account the messages from the other agents.
(c) Modify the frequency in which the task it is transmitted each period of time (specified on the blackboard in the priorities table by the duration attribute).
What can perceive?
The frequencies (busy or available), the communication messages for agents, the data transmitted, the priorities for each task.
What will perceive ?
The frequency (busy or available), and the priorities for each task.
Action
Makes the scheduling for the tasks to be transmitted on an specific frequency selected from the blackboard and updating it, reviewing if the other agents are using o releasing a region from the bandwidth.

No Autonomous Agent

Transmition
Kind of the goal: passive
Goal: transmits the task on the pre-select frequency.
perceptions
Action that perceives:
Transmits the task.
What can perceive?
The frequency to transmit the task.
What will perceive ?
The frequency to transmit the task.
Action
Transmits the tasks on the frequency selected by the autonomous agent *Scheduler* .

2.2 The Algorithm

The algorithm is based on a cooperative agent system supported on a blackboard arquitecture in a full-fledged scheme [3], to handle information. In the blackboard, each agent can write to provide a route to the solution of a problem. The agents are the main actors who continuously are seeing the blackboard to find out what is happening in their enviroment. They are able to modify it whenever they make the decision to carry out a change.

The Blackboard used in this system is a table that has some restrictions to assign a bandwidth. It reflexes the state of the environment. All the agents are looking and modifying this table.

If there is a process that requires to execute some tasks, it should takes into account that it schedules only if there is an adequate frequency to guarantee the time when they are transmitted.

Inside the Blackboard there is a table, the priorities and the bandwidth for each task, limited by a minimal and maximum frequency. This range allows an adequate frequency for the task and restrict the options.

Another important attribute is the duration, that allows to change the frequency where the task is transmitted. Each period makes a new scheduling consulting with the other agents which frequencies they are using and sent a message to the agents with the selected option.

3. Case Study and Control Reconfiguration

Case study is a magnetic levitation system integrated to a computer network through software and hardware tools briefly introduced here. Detailed information can be found in [4].

A. Computers and Software

The Real time Workshop and xPC Target combination eliminates the writing of low level code for the controler and others components of the system. The controller is generated from a Simulink model in the Host PC and the resulting code is downloaded to the target PC through ethernet port. A Quanser DAQ board by Quanser in a target computer provides the interface between the target PC and the maglev.

B. Hardware-in-the-Loop

The magnetic levitator (maglev) case study. It is a nonlinear, open-loop unstable and time varying system. Thyis device contains an electromagnet, an infrared sensor for position measurment and a steel ball. The experimental setup, including the computers and the maglev, is shown in Fig. 1 in [5]. Using a free-body diagram of the maglev in [5], it is obtained the current-to-position transfer function of the electro-mechanical system as:

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

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

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

where:
g is the gravity force
I_{co} is the current of the coil
x_{bo} is the distance from coil to ball position.

In the obtaining linearized dynamic equation, the assumption is that the levitated objet operates around the static equilibrium of a nominal operating point (x_{bo}, I_{co}).

C. Control Reconfiguration Approach

For the implementation, a basic time diagram system is proposed in Fig. 4 and Fig. 5 in [6]. When a fault appears, the use of cooperative agents is performed in order to re-organize task execution according to basic time restrictions. Maximum time delays are bounded on these figures. Where s_1 and s_2 are optic sensor nodes, C is the control node and A_1 is the actuator node.

Both scenarios are local with respect to maglev system. As these two scenarios are bounded, the related consumption times are shown in Equations 3 and 4 (are related to Fig. 4 and Fig. 5 in [6], respectively) based on Table 1, where variable information is presented.

From both scenarios there is an element known as fault tolerance element that presents extra communication for control performance although it masks any local fault from sensors. Final implementation is shown in Figure 1.

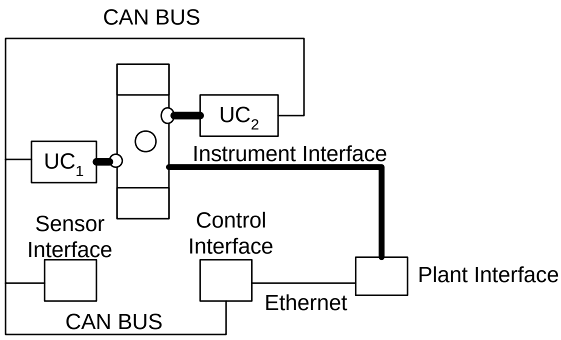

Figure 1. System Implementation

From this time boundary, including both scenarios, is feasible to implement some control strategies. There are two posible fault cases:

- One local fault;
- Several local faults.

Based on these two possible configurations, there is a worst-case scenario related to several local faults that has an impact on the global control strategy. Taking into account these two possible configurations, the local and global time delays are described in Table 1.

Table 1. Time delays related to local Communications

Fault Scenario	Local Time Delays	Global Time Delays
One Local Fault	1 ms	3 ms
Several Local Faults	1 ms	5 ms

As the time delays have been bounded, the plant model is defined based on equations 1 and 2 and Fig. 2.

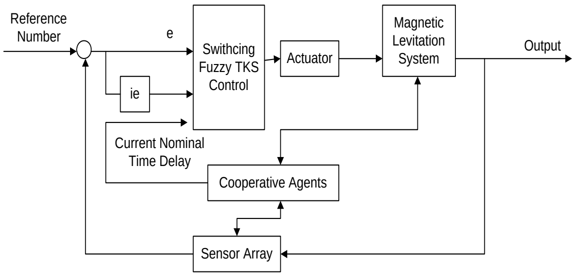

Figure 2. Plant and control law integration

The control law is defined as a group of Fuzzy TKS control law [7] related to each local linear system. The general structure of each fuzzy rule is:

r_i if x_1 is A_{1i}^c and x_2 is A_{2i}^c and … x_l is A_{li}^c then

$$f(k) = k_p^i e(k) + k_i^i \int e(k)dt \qquad (4)$$

where $i = \{1,...., N\}$, N is the number of fuzzy rules, $\{x_1,....,x_l\}$ are current states of the plant, A_{ij}^c are the gaussians membership functions like:

$$A_{ij}^c = \exp\left(-\frac{\left(x_i - c_{ij}^c\right)^2}{\sigma_{ij}^{c\,2}} \right) \qquad (5)$$

where: c_{ij}^c and σ_{ij}^c are constants to be tuned. Further explanation can be found in [8].

Using the proposed dynamic plant based on the structure, see [8] where:

$$x(k+1) = a^p x(k) + B^p u(k)$$

$$y = c^p x(k)$$

and for fault scenario:

$$B_i^p = B_i \sum_{j=1}^{M} \int_{\tau_j^i}^{\tau_{j-1}^i} e^{-a^p(t-\tau)} d\tau$$

that considers local faults and related time delays, with the condition of: (see equations 3 and 4)

$$\sum_{j=1}^{N} t_j^i \le T$$

and following the representation of the plant in [8]:

$$h_i = \prod_{j=1}^{l} A_{ij}(x_j)$$

and replacing for the defuzzification:

$$x(k+1) = \frac{\sum_{i=1, j=1}^{N} h_i w_j \left(\left(a_i - c_i (k_p^i e(k) + k_i^i \int e(k)dt) B_i^p \right) x(k) + B_i^p ref \right)}{\sum_{i=1, j=1}^{N} h_i w_j} \qquad (6)$$

where *ref* is the reference to be followed by controller and the variables i and j are used due to fuzzy rules interconnections as the representation of different linear plants and respective controllers.

4. Results

From this implementation several results are presented in terms of fault presence and the related action to overcome system lack of performance from control constants K_i and K_p. How the system responds to control strategy is presented in the following graphics showing error response from different frequencies for three different tasks. Fig. 3 shows fault-free scenario with the modification of the frequencies of the tasks as well as the overall control error response.

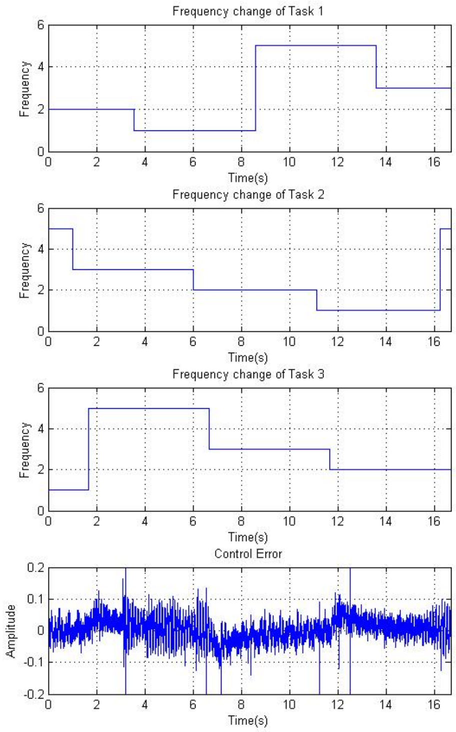

Fig. 3. Error response from fault-free scenario

Fig. 4 shows the response of the tasks over the same change of the frequencies shown in previous fig.

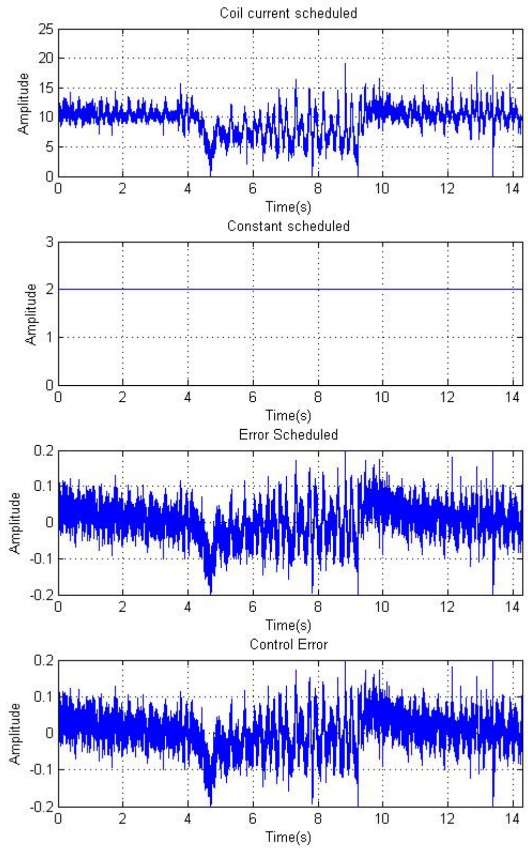

Fig. 4. Tasks Response from Fault-Free scenario

This last example presents control reconfiguration based on the decision-maker module implemented through the cooperative agents. Active switching control is performed based on Fuzzy Takagi Sugeno Control approach when a local fault appear and fault tolerant node takes place.

This reconfiguration approach becomes feasible due to the knowledge of fault presence and the consequence of time delays. Its consumption time is neglected, and it is considered part of control performance. It is obvious that fault presence is measurable; if this local fault localization approach cannot detect faults, this strategy becomes useless. Alternatively, local time delay management refers to the use of a quasi-dynamic scheduler to propose dynamic reconfiguration based on current system behavior rather than predefined scenarios.

In this case, fault and fault-free scenarios are the same as in the first approach. The selected scheduler strategy is performed on-line. The selected scheduler algorithm is the use of Cooperative system to define fixed nonpreemptive tasks like controllers and actuators. Local control laws are expressed in the same form as expressed before, with the only modification of this time delay; at this moment, reconfigurable control is in same form as the global control law event structure.

5. Concluding Remarks

Present approach shows the integration of two techniques in order to perform reconfiguration. These two approaches are followed, in cascade mode, structural reconfiguration and control reconfiguration. Although there is no formal verification in order to follow this sequence, it has been adopted since structural reconfiguration provides settle conditions for control reconfiguration. The use of a real-time scheduling algorithm in order to approve or disapprove modifications on computer network behaviour allows time delays bounding during a specific time window. This local time delay bounding allows the design of a control law capable to cope with these new conditions.

This scheduling algorithm is perform by the use of cooperative agents programmed within the simulink block by the use of an Sfunction.

Preliminary results show that control reconfiguration is feasible as long as the use of a switching technique predetermines which control is the adequate. This goal is reached by a strategy compose of two algorithms, one which is responsible for structural reconfiguration and it has been implemented in this paper as cooperative agent approach. The second algorithm is responsible for Fuzzy Takagi Sugeno PID control reconfiguration. What it is important for this last approach is that control conditions are strictly bounded to certain response. Future work is related to integrate dynamic shedulling algorithms and formal stability probe of this implementation.

ACKNOWLEDGEMENTS

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

6. References

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

[2] *xPC Target User's Guide*, MathWorks, Inc., MA, 2004.

[3] Daniel D. Corkill. Collaborating Software: Blackboard and Multi-Agent Systems & the future, In Proceedings of the international Lisp Conference, New York, New York, October 2003.

[4] Wincon;http://www.quanser.com/english/html/solutions/fs_soln_software_wincon.html, 2003.

[5] P. S. Shiakolas and D. Piyabongkarn, "Development of a real-time digital control system with a hardware-in-the loop magnetic levitation device for reinforcement of controls education," *IEEE Trans. Educ.*, vol. 46, pp. 79–87, Feb. 2003.

[6] Benítez-Pérez, H., Quiñones-Reyes, P., Mendez-Monroy, E., García-Nocetti, F. And Cardenas-Flores, F.; "Reconfigurable Fault Tolerant PID Networked Control for Magnetic Levitation Case Study "; Accepted paper for the IFAC Symposium on Safeprocess in Beijing, P.R. of China. August 2006.

[7] Abonyi J.; "Fuzzy Model Identification for Control"; Birkhäuser, 2003.

[8] Benítez-Pérez, H., Cardenas-Flores, F. and García-Nocetti, F.; "Reconfigurable Takagi-Sugeno Fuzzy Logic Control for a Class of Nonlinear System considering Communication Time Delays"; Accepted paper for the International Conference on Control. UK 2006.