

A Fuzzy Elevator Group Controller with Linear Context Adaptation

Ricardo Gudwin
DCA-FEEC-UNICAMP
gudwin@dca.fee.unicamp.br

Fernando Gomide
DCA-FEEC-UNICAMP
gomide@dca.fee.unicamp.br

Márcio Andrade Netto
DCA-FEEC-UNICAMP
marcio@fee.unicamp.br

Abstract

In this paper we introduce a fuzzy elevator group controller using a linear context adaptation technique. We first describe the elevator group control problem and the schemes usually employed to solve it. We detail the fuzzy controller used in our development and an example system used in simulation experiments. The focus is on the comparison between the standard fuzzy controller and the fuzzy controller with linear context adaptation. Simulation results are included to show the usefulness of the fuzzy control strategy suggested.

1. Introduction

In traffic control of elevator systems, figure 1, a corresponding two level control hierarchy must solve two different control problems. The lower level task is to command each elevator to move up or down, to stop or start and to open and close the door. The higher level coordinates the movement of a group of elevators through a set of logical rules crafted to improve the system performance. This problem is solved by means of a group supervisory control system with the aid of a group supervisory control strategy (the set of rules defining the control policy).

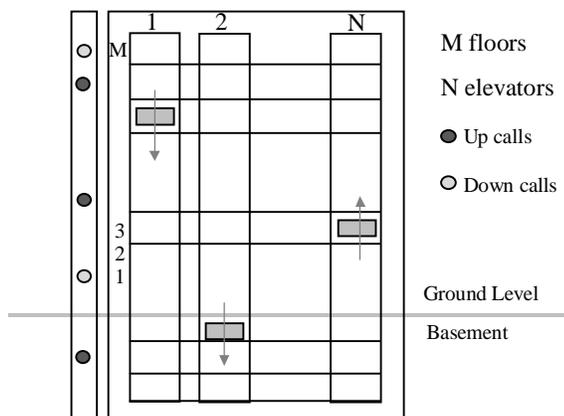


Figure 1 - Elevator group control problem

The main requirements of a group control system in serving both, car and hall calls, should be: to provide even service to every floor in a building; to minimize the time spent by passengers waiting for service; to minimize the time spent by passengers to move from one floor to another; to serve as many passengers as possible in a given time [1].

Due to the random nature of call time, call locations and the destination of passengers, there exist immense problems when attempting to satisfy all the above requirements. In general, the control strategy must be flexible enough to meet diverse conditions especially when changes in passenger demands, different traffic patterns (up-peak, down-peak, special services, etc.) are of concern.

A practical method widely used in group supervisory control systems consists in allocating cars to serve the building hall calls. Usually only new calls are allocated, remaining fixed once made. This method is known as call allocation strategy [1]. In car allocation, however, constraints must also be considered. For instance, a car may not pass a floor at which a passenger wishes to alight, a car may not reverse its direction of travel while carrying passengers, and a hall call cannot be served by a car going in the reverse direction.

In this paper we introduce a fuzzy group supervisory control of elevator systems. This approach is particularly useful because of the following:

- call allocation strategies are frequently expressed by a set of (fuzzy) rules provided by elevator design experts;
- different sets of rules can be easily grouped into a structured knowledge base to cover diverse traffic conditions such as up-peak, down-peak, heavy sector demand, heavy demand floor, balanced traffic, off-peak, etc.;
- logical constraints can be easily included into the knowledge base and equally processed;

- meta-rules can be easily developed and incorporated to choose the most suitable car allocation rules for a given traffic pattern;
- it predictably responds to system events.
- fuzzy concepts like high, medium and low are very appropriate to be used and comprehended by experts.

In addition, fuzzy production rules are particularly useful to derive both, allocation strategies and constraints. The assignment of the allocation strategy to the traffic condition is handled by a focus of attention mechanism, using meta-rules for traffic pattern characterization and rule base selection. Data acquisition and information coding functions are performed by a pre-processor. Actual call allocation decisions and control signals, generated after inference, are computed by a post-processor. This will be detailed in the next sections. For alternative developments see, e.g. [2,3,4].

2. Elevator group control system

The general functional structure of the elevator group control system is portrayed in figure 2. In the two level control hierarchy, the lower level task is to command each elevator (car) to move up and down, to stop or start, and to open and close the door. Actually this is a direct control level which performs the basic car movements. The higher level is a supervisory level to coordinate the movement of a group of cars to achieve a globally satisfactory performance. Coordinating decisions are based on system state (traffic conditions, hall calls, cabin calls, etc.). These decisions are translated into commands for the direct control level.

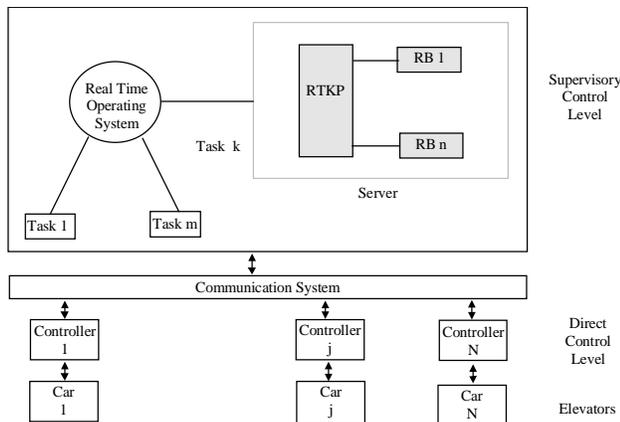


Figure 2 - Elevator group control system: general structure

The fuzzy group controller is a task of the overall control system encapsulated within a server to guarantee temporal isolation between itself and conventional real

time tasks. As highlighted in figure 2, it is composed by a real time knowledge processing module (RTKP), and several rule bases (RBi's) to support focus of attention mechanisms [5].

3. Fuzzy group controller

As depicted in figure 3, the fuzzy group controller is divided into four basic parts. First is a preprocessor module responsible for the transformation of input information into the internal representation model used. The postprocessor module translates the internal representation model into output information (commands) as required by the lower level controllers. Between these two modules there are an inference module and a metaknowledge base. These are for rule processing and rule base activation, respectively.

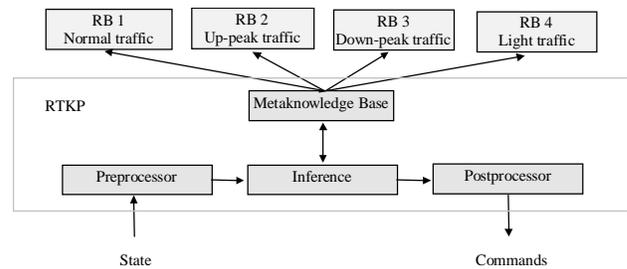


Figure 3 - Fuzzy group controller structure

Thus, the group controller constitutes four separate rule bases the selection of whose is performed by metarules. This specific rule base consists of two-valued predicates. The rules therein are of classificatory nature as they identify the current pattern of traffic and select one among the remaining rule bases. Each of these rule bases is responsible for handling a particular traffic pattern that is specific enough to deserve a separate control protocol. We have formed rule bases for the four types of traffic: normal, up-peak, down-peak, and light traffic. The detailed rules in the rule bases are summarized below.

Table 1. Rules for normal traffic

A_time(i,j) W_time(j)	Big_w	Medium_w	Small_w
Big_a	Medium	Low	Low
Medium_a	High	Medium	Low
Small_a	High	High	Medium

In Table 1, $W_time(j)$ and $A_time(i,j)$ are linguistic variables denoting the time the j -th hall call is waiting for service, and the time the i -th elevator expects to attend the j -th hall call, respectively. These rules are processed whenever there are hall calls to be attended,

and the number of passengers within the i -th elevator is less than its full service load. Otherwise, the corresponding priority is set at Zero. Note that priority is a decision variable whose linguistic terms are Zero, Low, Medium and High, being assigned for each pair (Elevator, Call) – Priority(i,j).

Table 2. Rules for up-peak traffic

A_time(i,j)\W_time(j)	Big_w	Medium_w	Small_w
Big_a	Medium	Low↑ (Zero↓)	Low↑ (Zero↓)
Medium_a	High	Medium↑ (Zero↓)	Low↑ (Zero↓)
Small_a	High	High↑ (Zero↓)	Medium↑ (Zero↓)

The rules in Table 2 looks the same as before, but rule firing is very different. First, during up-peak conditions rules are processed when the number of passengers within elevators are below the maximum capacity only. Otherwise, priorities are set to Zero. Second, the first row rules are processed independently of the elevator direction of movement whereas the decisions are set at Zero at the shaded columns if hall calls are down calls, or as shown in the Table, if hall calls are up calls. In addition, a two-valued rule immediately fires to send elevators to the main floor whenever it becomes available.

Table 3. Rules for down-peak traffic

A_time(i,j)\W_time(j)	Big_w	Medium_w	Small_w
Big_a	Medium	Low↓ (Zero↑)	Low↓ (Zero↑)
Medium_a	High	Medium↓ (Zero↑)	Low↓ (Zero↑)
Small_a	High	High↓ (Zero↑)	Medium↓ (Zero↑)

Down-peak rules, shown in Table 3 are processed under the similar conditions of up-peak rules, but with the entries of the shaded columns reversed, and no car is sent to the main floor when it becomes available.

Rules for light-traffic conditions are similar to the normal traffic conditions except that the number of elevators in service may change for, e.g., energy saving and maintenance purposes.

The membership functions are as in figure 4,5 and 6.

Rules are viewed as fuzzy conjunctions of min type, and inference is proceeded via the max-min composition. Rules aggregation is performed after active rules firing, using the max aggregation operator. The center of area

was the defuzzification method used. In other words, the procedures of encoding, decoding, and mapping mechanism are:

- encoding: possibility computations,
- mapping: fuzzy conjunction,
- decoding: center of gravity

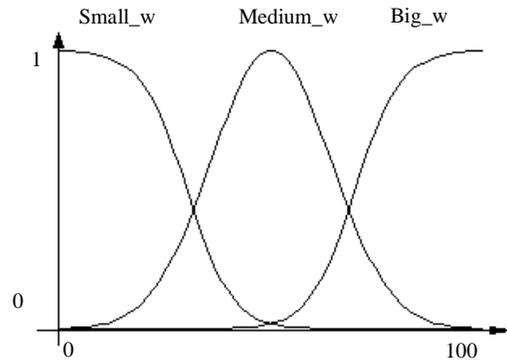


Figure 4 – Membership Functions for Linguistic Variable W_time(j)

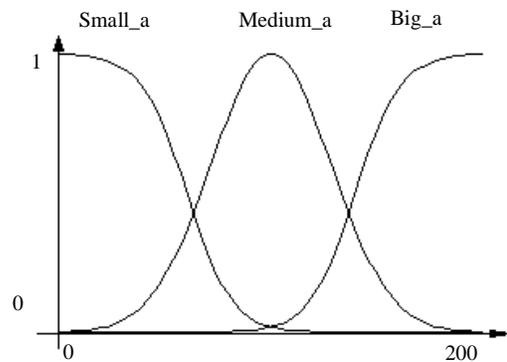


Figure 5 – Membership Functions for Linguistic Variable A_time(i,j)

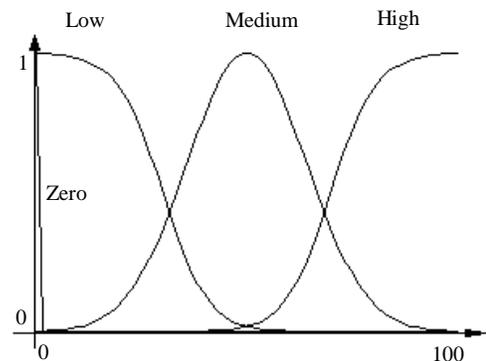


Figure 6 – Membership Functions for Linguistic Variable Priority(i,j)

After defuzzification, a priority ranging from 0 to 100 is assigned to each pair (Elevator, Call). Next, an allocation algorithm tries to allocate, for each elevator, the call with the maximum priority. If two or more elevators are qualified to serve the same call, they compete with each other, based on their A_time values. If the A_time is the same, then the algorithm makes a random choice. This mechanism is illustrated in figure 7 and proceeds as follows.

E1	12↑	7↓	3↓	1↑	14↓	8↓	20↑
E2	20↑	8↓	14↓	1↑	3↓	7↓	12↑
E3	12↑	3↓	1↑	20↑	8↓	14↓	7↓
E4	7↓	14↓	8↓	20↑	20↑	12↑	1↑

Figure 7 – Allocation Algorithm

For each elevator, a queue is built to order the calls by priority. The algorithm tries to select the first element of each queue. In the example above, elevators E1 and E3 conflict in serving call 12↑. They compete and elevator E3 wins. Elevator E1 consider its 2nd option, 7↓, which is now conflicting with E4. If E4 loses, it proceeds to its 2nd option. At the end of the algorithm, each elevator serves a different call.

The fuzzy controller is also equipped with the mechanism of context adaptation [6,7,8]. Here, context adaptation is to adjust universes in such a way that what is meant to be, e.g., High or Low, depends on the traffic intensity. The scheme used is called the absolute limit context determination method, the simplest among a number of alternatives [6]. Briefly, to define the context for a situation, samples are taken to generate the upper and lower bounds of the underlying universes.

The scheme emphasizes the fact that the semantics (membership functions) of the linguistic terms are very much context-dependent and time-varying. This is because there is no universal meaning of such terms like low, medium, high, small, big and their definition may vary during the task of controlling the system. For instance, the meaning of small waiting time depends on the traffic pattern.

4. Simulation experiments

The series of simulation experiment was run in the following environment [9]:

- building and elevators:
 - 15 floors above main lobby and 3 floors below ground level,
 - height of each floor: 3.3 m,
 - 4 elevators.

- timings
 - time to open the door: 1 sec,
 - time to close the door: 1 sec,
 - time for each passenger to get in: 1 sec,
 - time the door remains open: 1 sec,
 - acceleration: 1m/s²,
 - desacceleration: 1m/s²,
 - maximum load: 1600 kg,
 - capacity: 20 persons.

The main patterns of traffic were considered; each experiment was carried out with context adaptation and without it. The fuzzy controller described here has already been compared with conventional strategies, with improvements of up to 30% in performance [10]. In all experiments, the arrival request at the floors (hall calls) was modeled using a Poisson distribution. The scenarios were as follows:

- 1 - up-peak: this traffic condition concerns people entering the building in the morning between 7:45 and 8:45,
- 2 - down-peak: this occurs between 17:15 and 18:45 and concerns the traffic out of the building after working hours,
- 3 - all day traffic: it covers all day from 6:00 to 20:00.

Two kinds of simulation were performed. The first used the fuzzy controller without context adaptation, and the other one with context adaptation.

The main results of group control are expressed via an average waiting time and maximum waiting time (both given in seconds) and are collected in Tables 4 and 5. In all experiments the adaptation of context resulted in substantial gains in system performance.

Table 4 Average and maximum waiting time, no context adaptation

	Average	Maximum
Down-peak	25.00	184.0
Up-peak	20.13	105.0
All day	14.95	209.0

Table 5 Average and maximum waiting time, context adaptation

	Average	Maximum
Down-peak	23.04	132.0
Up-peak	17.00	92.0
All day	12.93	156.0

The figures below show the traffic profile, the waiting time histogram for up (light bars) and down calls (dark bars), and samples of the elevator movements as well.

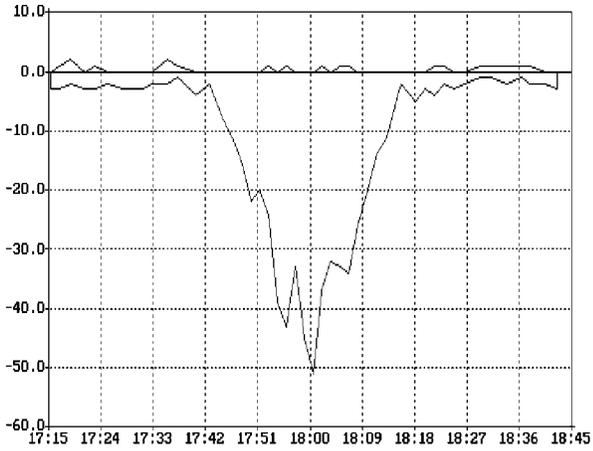


Figure 8 - Profile for down-peak traffic

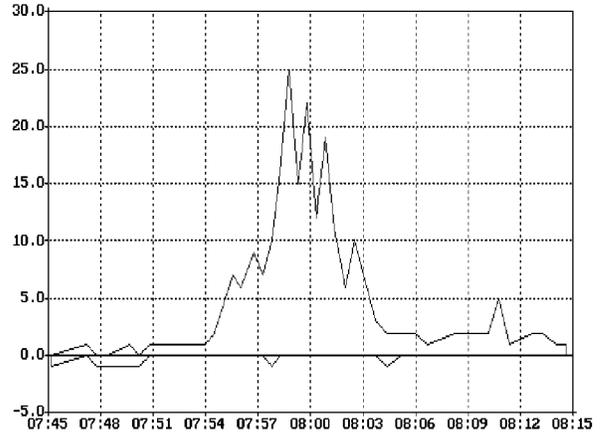


Figure 11 - Profile for up-peak traffic

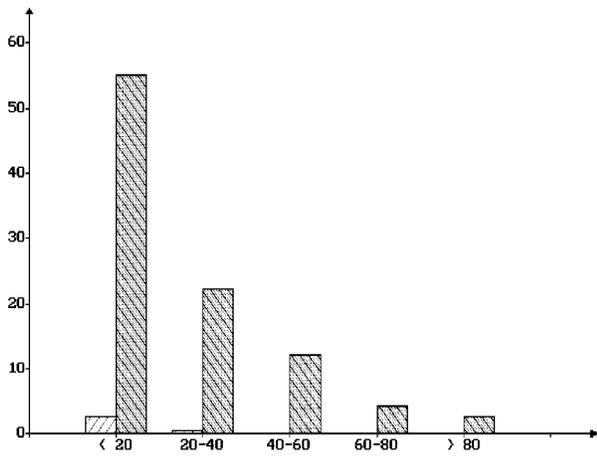


Figure 9 - Histogram for down-peak with context adaptation

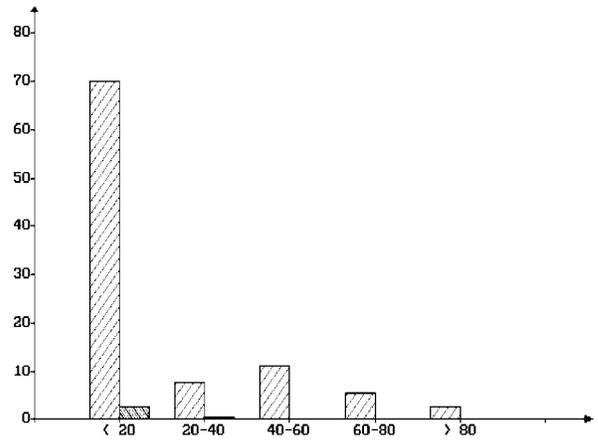


Figure 12 - Histogram for up-peak with context adaptation

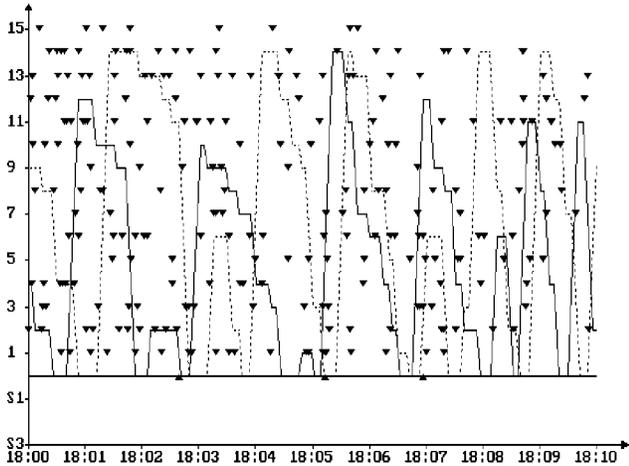


Figure 10 - Sample of the trajectories for two elevators, down-peak

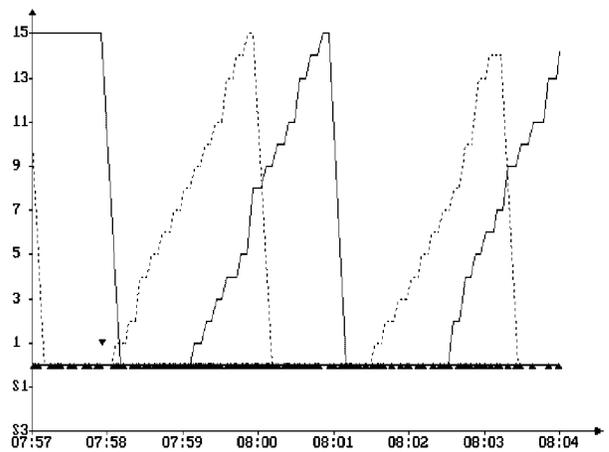


Figure 13 - Sample of the trajectories for two elevators, up-peak

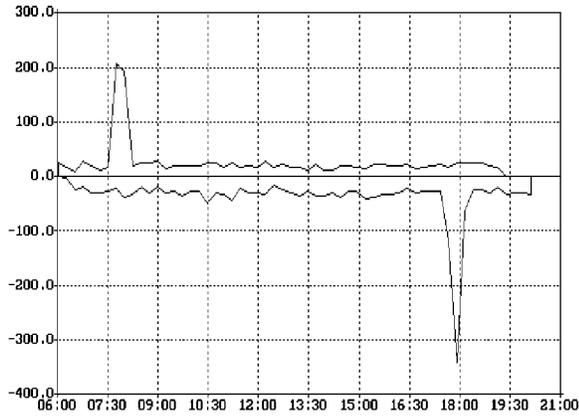


Figure 14 - Profile for all day traffic

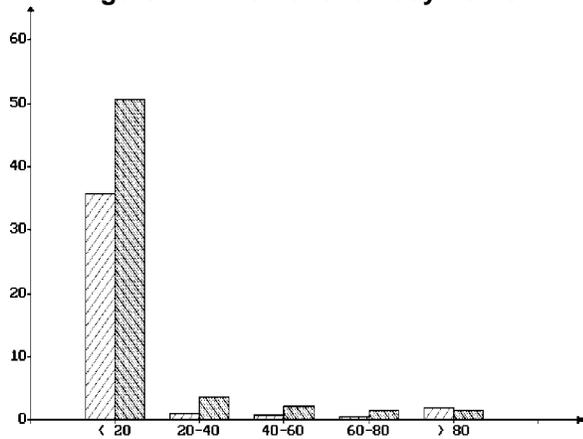


Figure 15 - Histogram for all day traffic with context adaptation

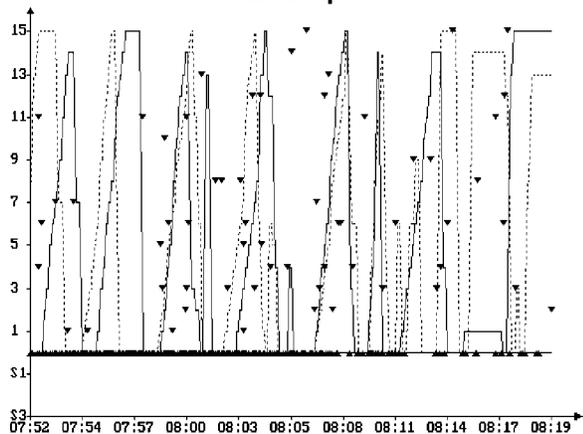


Figure 16 - Sample of the trajectories for two elevators, all day traffic

5. Conclusions

In this paper, we have introduced a fuzzy controller with linear context adaptation, and compared its performance with a standard fuzzy controller. The main purpose was to use it as a part of an elevator group

control system. Simulation results have shown that the fuzzy controller with linear context adaptation performs better than the standard fuzzy controller. This is justifiable because the standard fuzzy controller remains fixed once tuned. Although we can set appropriate universes for each specific rule base to accommodate the different traffic patterns, the standard fuzzy controllers may still become insensitive to traffic fluctuations that occurs in a given situation. In contrast, the controller with context adaptation is continuously adapting the underlying universes and can keep tracking of the traffic pattern changes. An additional advantage of context adaptation is that we are not required to define precisely the universes (or to fine-tune the membership functions) because they are automatically adjusted at each step. In this sense, context adaptation translates into a simple, yet efficient mechanism to get a kind of adaptive fuzzy controller.

6. References

- [1] G.C. Barney and S.M. dos Santos, "Elevator Traffic Analysis Design and Control" – IEE Control Engineering Series 2, 2nd edition, London, Peter Peregrinus Ltd, 1985.
- [2] C. Kim, K. Seong, H. Lee-Kwang, J. Kim, Y. Lim, "A Fuzzy Approach to Elevator Group Control System" – IEEE Trans. On SMC, vol. 25, n. 6, June 1995, pp. 985-990.
- [3] Y. Umeda, K. Uetani, H. Ujihara, S. Tsuji – "Fuzzy Theory and Intelligent Options" – Elevator World, July, 1989.
- [4] M. Funabashi, K. Mori – "Knowledge Based Control Systems and Software for Building Expert Systems – EUREKA-II" – Hitachi Review vol. 37, n. 4, 1988.
- [5] R.R. Gudwin, F.A.C. Gomide, M.L. Andrade Netto, M.F. Magalhães, "Knowledge Processing in Control Systems" – IEEE Transactions on Knowledge and Data Engineering, February 1996, vol. 8, n. 1, pp. 106-119.
- [6] R. R. Gudwin, F.A.C. Gomide, "Context Adaptation in Fuzzy Processing" – Proceedings of the Brazil-Japan Joint Symposium on Fuzzy Systems, Campinas, SP – Brazil, July, 1994, pp. 15-20.
- [7] W. Pedrycz, R.R. Gudwin, F.A.C. Gomide, "Nonlinear Context Adaptation in the Calibration of Fuzzy Sets" – Fuzzy Sets and Systems 88 (1997) 91-97.
- [8] R.R. Gudwin, F.A.C. Gomide, W. Pedrycz, "Nonlinear Context Adaptation with a Genetic Algorithm" – Proceedings of the 7th IFSA World Congress, Prague, Czech Republic, June, 1997, pp. 300-305.
- [9] W. Pedrycz, F.A.C. Gomide, "An Introduction to Fuzzy Sets: Analysis and Design" MIT Press, 1998.
- [10] R. R. Gudwin – "A Kernel for Real Time Knowledge Processing" – M.S. Thesis – DCA-FEEC-UNICAMP, Febr. 1992 (in portuguese).

Acknowledgement: The authors acknowledge CNPq, the Brazilian National Research Council for its support and grant #300729/86-3, and FAPESP. They also acknowledge the anonymous reviewers, for their suggestions.