

A fuzzy approach for the network congestion problem

Article

Accepted Version

Di Fatta, Giuseppe, Lo Re, Giuseppe and Urso, Alfonso (2002) A fuzzy approach for the network congestion problem. Lecture Notes in Computer Science (2329). pp. 286-295. ISSN 0302-9743 doi: https://doi.org/10.1007/3-540-46043-8_28 Available at <https://centaur.reading.ac.uk/6146/>

It is advisable to refer to the publisher's version if you intend to cite from the work. See [Guidance on citing](#).

To link to this article DOI: http://dx.doi.org/10.1007/3-540-46043-8_28

Publisher: Springer

Publisher statement: The original publication is available at www.springer.com/lncs

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the [End User Agreement](#).

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

Reading's research outputs online



A Fuzzy Approach for the Network Congestion Problem

Giuseppe Di Fatta, Giuseppe Lo Re, Alfonso Urso

CERE, Centro di studio sulle Reti di Elaboratori, C.N.R.,
viale delle Scienze, 90128 Palermo, Italy
{ difatta, lore, urso }@cere.pa.cnr.it

Abstract. In the recent years, the unpredictable growth of the Internet has moreover pointed out the congestion problem, one of the problems that historically have affected the network. This paper deals with the design and the evaluation of a congestion control algorithm which adopts a Fuzzy Controller. The analogy between Proportional Integral (PI) regulators and Fuzzy controllers is discussed and a method to determine the scaling factors of the Fuzzy controller is presented. It is shown that the Fuzzy controller outperforms the PI under traffic conditions which are different from those related to the operating point considered in the design.

Keywords: *AQM, Congestion Control, Fuzzy Control.*

1 Introduction

In the recent years, the unpredictable growth of the Internet has moreover pointed out the congestion problem, one of the problems that historically have affected the network. The network congestion phenomenon is induced when the amount of data injected in the network is larger than the amount of the data which can be delivered to destinations. Two different approaches, which can be considered complementary parts of a single main strategy, can be adopted to solve such a problem. The approach which historically has represented the beginning of network congestion control is a so called end-to-end approach; in this approach when data sources infer congestion occurrences from packet losses, they properly reduce their transmission rate. This is, for instance, the approach adopted by the Transmission Control Protocol. The effectiveness of a control system where sources are responsible for congestion control is based essentially on the fact that all, or at least most of the applications running through the network respect its rules. In this environment non compliant flows can obtain larger bandwidth against the ones which correctly obey the control laws. End-to-end peers detect congestion level by inferring it from packet losses. A packet loss could mean that one of the intermediate routers does not have enough memory space to host it before its retransmission on the appropriate link towards the destination. The simplest and also most deployed policy adopted by a router to

manage its queues, is a First Come First Served policy which is implemented by means of a First In First Out queue management. Such a policy, known as Drop Tail for router queue management, presents several disadvantages such as, the high delays suffered by packets when they go through longer queues. To eliminate the Drop Tail disadvantages and to anticipate the source answers to incipient congestion situations, authors in [1] proposed the adoption of a Random Early Detection (RED) policy. RED is an active policy of queue management which involves the dropping, or marking, of packets when the queue average length ranges between a minimum and a maximum threshold. The probability of packet dropping/marking is obtained from the average queue length accordingly to a linear law. In the last years, the active queue management policies have been object of a large interest in the scientific networking community and several proposals have been presented to find more effective control policies than RED. Among these we will refer to REM (Random Exponential Marking) [5] and PI (Proportional Integral) [8]. Nevertheless, these AQM policies suffer the disadvantage that they are unable to maintain their performances as the number of TCP flows increases. In order to avoid such a problem, it is possible to use a congestion control algorithm based upon a fuzzy logic controller. Namely, for high-order nonlinear systems, they often produce better results than those obtained by classical control techniques. Accordingly, many research efforts have been carried out to develop fuzzy logic controllers [9],[10],[11]. On the other hand, the design of a fuzzy logic controller is not straightforward, because of the heuristic involved in control rules and membership functions; moreover, the tuning of the parameters of a fuzzy logic controller, such as scaling factors, membership functions and control rules is a very complex task. Currently there are not automatic methods available for the design of the fuzzy knowledge base and for the tuning of a fuzzy logic controller. Therefore, the designers have to devise a fuzzy knowledge base by heuristic methods, employing experience. Accordingly, the parameters of a fuzzy control system are tuned by a trial and error method. This leads to a well-known fact that the design of a fuzzy logic controller is more difficult than the design of a conventional controller. In this paper, in order to obtain a method for the tuning of a fuzzy controller, the analogy between Proportional Integral (PI) regulators and fuzzy controllers is discussed and a method to determine the scaling factors of the fuzzy controller is presented. The remainder of the paper is structured in the following way. In section 2 we will examine more deeply RED, REM and PI. Section 3 deals with the design of a Fuzzy controller and how its project is related to the PI one. In section 4 we present the set of experiments we carried out and discuss the results. Finally, section 5 is devoted to conclusions.

2 Active Queue Management Policies

The acronym AQM (Active Queue Management) indicates those policies of router queue management that allow: a) the detection of network congestion, b) the notification of such occurrences to the hosts on the network borders, c)

the adoption of a suitable control policy. With reference to the congestion notification, two different approaches can be followed. The first one involves the setting of a bit called ECN (Explicit Congestion Notification) in a sample of the packets flowing through the router. In turn, the destination will transmit such information to the source piggybacking it into the acknowledgement message. The second approach, used for those protocols that are not able to manage the ECN bit, involves a more drastic action, i.e. a probabilistic dropping of packets with the aim to induce a reaction of the data sources. The current version of TCP (Transmission Control Protocol) is not able to manage the ECN bit and, as a consequence, only the second solution may be adopted by Internet routers. The first AQM policy proposed for the Internet has been RED that, as previously mentioned, involves the dropping of packets accordingly to a probability law which ranges linearly when the average queue length is varying between a minimum and a maximum threshold. RED calculates the average queue length by assigning different weights to old value and current measure. This means the adoption of a low pass filter to reduce the high frequency variation of the instantaneous queue. This behavior is a precise design choice of the authors to overcome the oscillations which may be induced by isolated bursts. In the last years several objections have been raised against RED, among which the difficulty of setting proper RED parameters according to network conditions, and the dependence of the queue length in steady state from the number of flows. Namely, a growth of the flow number involves that of the average queue length, which, in turn, could exceed the maximum threshold, and all packets would be dropped. However, it is not possible to increase too much the maximum threshold, because this would mean higher queueing delays. On the other hand, if the maximum threshold is set to a low value, this would mean a bad usage of the link because of severe buffer oscillations.

From these considerations follows that it is very difficult to find out the proper trade-off, and it is not possible to tune RED to achieve both high link utilization and low delay.

Several proposals have been presented in the recent years which introduce RED improvements [2] [3].

Recently different approaches than RED have been proposed; among these two proposals REM and PI which, although obtained independently and following full different theoretical approaches, seem to represent, accordingly to their authors, the same solution [8]. REM is the result of a linear modelling of the problem and of its resolution in its dual form. Differently from RED, REM differentiates between the congestion measure of each router and the dropping probability. In REM the authors introduce a measure called price that eliminates the dependence of the dropping probability from the current value of the queue size. The derived algorithm uses the current queue size and its difference from a desired value to calculate the dropping probability accordingly to an exponential law. Such a feature owns the additivity property, so thus a source can calculate the price of the whole path using the knowledge of the total number of packets dropped along the path.

The PI controller for AQM uses classical control system techniques to design well suited control law for the router queue management. In particular, a nonlinear dynamic model for TCP/AQM has been developed in [6]. Once the model is linearized around an operating point, a stable PI linear controller is designed in order to satisfy the project specifications. The authors in [8] show that in the PI controller the Proportional part is equivalent to RED when the input low-pass filter is removed. The usage of a proportional controller leads to a lower time of response but also to lower stability margins; moreover, the proportional controller has a steady state regulation error, where such an error is defined as the difference between the steady state output queue and the reference value. In order to overcome the above disadvantages, the integral term is added which has the characteristic to give steady state error equal to zero and to give higher stability margins.

3 Fuzzy Controller

The AQM policies described in the above section suffer from the disadvantage that they are unable to maintain performance, in terms of speed of response, as the number of TCP flows increases. From a control point of view, with reference to the PI controller, this disadvantage is essentially due to the fact that the high frequency gain of the open loop transfer function is fixed because the controller design is carried out considering a particular value of the TCP flows. Namely, when the load increases the high frequency gain decreases and the system bandwidth becomes lower which implies a slower system in terms of rise and settling time. To overcome this disadvantage we use a fuzzy logic controller. In recent years, fuzzy logic controllers, especially Fuzzy Proportional-Integral (FPI) controllers, have been widely used for processes control owing to their heuristic nature associated with simplicity and effectiveness for both linear and nonlinear systems [14], [15]. In fact, for single-input single-output systems, fuzzy logic controllers can be essentially seen as PI type associated with nonlinear gain. Because of the nonlinear property of control gain, FPI controllers can achieve better system performance than the conventional PI controllers. On the other hand, due to the existence of non linearity, it is usually difficult to conduct theoretical analyses to explain why FPI controllers can achieve better performance. Consequently, it is useful to explore the nonlinear control properties of FPI controllers to improve the closed-loop performance.

Moreover, a method which allows to obtain the parameters of the fuzzy controllers is needed. Systematic methods for the determination of FPI scale factors have been developed in [12] [13] [16], which take advantage from the analogy of FPI and conventional PI controllers. More precisely, a set of relationships between the scale factors and the gains of PI controller are obtained from such an analogy. Then, the gains of PI controller are determined so as to satisfy, for instance, requirements on bandwidth of the control loop. Finally, scale factors are computed from the gains of PI controller.

As already said, fuzzy controllers taken into account in this paper are the PI-

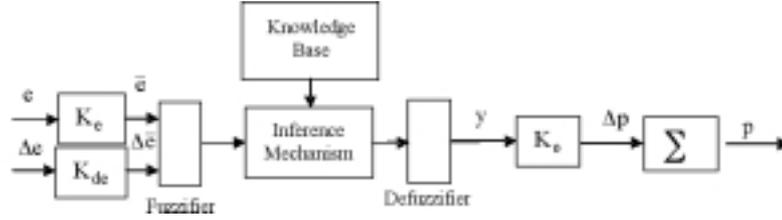


Fig. 1. Structure of FPI controller

type fuzzy controllers, as depicted in figure 1, where all quantities are considered at the generic discrete instant kT_s , with T_s the sampling period, $e = q - q_r$ is the error on the controlled variable q (queue size), $\Delta e = e(kT_s) - e((k-1)T_s)$ is the variation of the error, Δp is the increment of the control variable p (probability of packet marking/dropping) and K_e , K_{de} , and K_o are scaling factors to be determined.

It should be noticed that these fuzzy controllers have an intrinsic PI action. As a consequence, the steady-state behavior in the operating point cannot be different in the two controllers.

Once, the relationships between the parameters of the fuzzy controller and those of the standard PI one have to be determined; then a method of synthesis of the standard controller has to be chosen. In this paper, with the aim to synthesize the standard PI controller, we adopt the dynamic model of TCP behaviour using fluid-flow and stochastic differential equation analysis developed by Hollot et al.[6]. Furthermore, the synthesis of the PI controller is carried out following the guidelines to design stable controllers given in [8].

Finally, we can derive the relationships between the gains of the PI controller and the scale factors of the fuzzy one. Let us suppose that a set of rules and membership functions on normalized axes have been assigned, the fuzzy controller generates a non linear surface of the form:

$$y = f(\bar{e}, \Delta \bar{e}), \quad (1)$$

whereas the relation $p - y$ is that of a discretized integrator of gain K_o . The corresponding standard PI regulator can be seen as a plane whose equation, using the trapezoidal integration method and with reference to e , Δe and Δp axes, is:

$$\Delta p = K_I T_s e + \left(K_P + \frac{K_I T_s}{2} \right) \Delta e, \quad (2)$$

where K_P and K_I are the gains of the PI regulator. Moreover, it is possible to approximate the surface (1) with the plane:

$$y = K'_e \bar{e} + K'_{de} \Delta \bar{e}, \quad (3)$$

where K'_e and K'_{de} are obtained by means of minimization of the following index:

$$J = \int \int_{-\epsilon}^{\epsilon} \left(K'_e \bar{e} + K'_{de} \Delta \bar{e} - f(\bar{e}, \Delta \bar{e}) \right)^2 d\bar{e} d\Delta \bar{e}, \quad (4)$$

and where the parameter ϵ defines a well suited (small) interval around the operating point.

A least square solution for the minimization of the above index it is possible by the discretization of the integral in (4). Finally, the output Δp of the PI controller (2) is made equal to the output of linear approximation (3) scaled by K_o . Therefore, the final analogy relationships can be given as follows:

$$\begin{cases} K_P = K_o K'_{de} K_{de} - \frac{K_I T_s}{2} \\ K_I = \frac{K_o K'_e K_e}{T_s} \end{cases} \quad (5)$$

Given K_P and K_I from the synthesis of the PI controller, the equations (5) provide a degree of freedom to determine the parameters K_e , K_{de} and K_o of the fuzzy controller. We use this degree of freedom to set the parameter K_o in order to fix the maximum allowed variation of the control variable Δp .

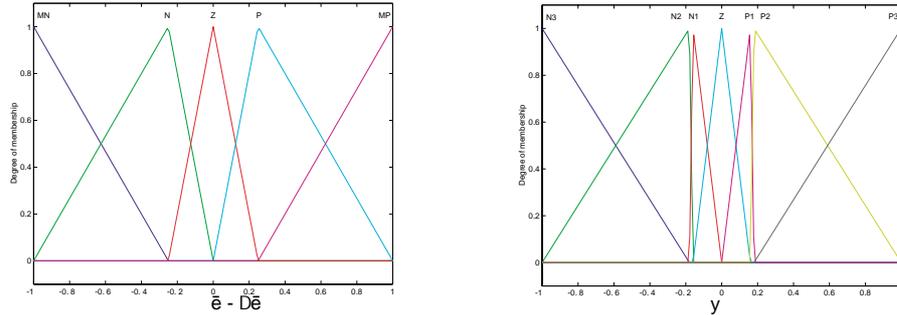


Fig. 2. Fuzzy Membership Functions of Input and Output Variables.

With reference to the fuzzy knowledge base, the membership functions of the input and output fuzzy variables are adopted as a triangular shape and they are depicted in figure 2. The determination of the rules requires the knowledge or experience of experts about the particular problem to be faced. In our design we select both the peak values of the membership functions and the rules in order to obtain a behaviour which reproduces that of the PI controller in an interval around the operating point ($\bar{e} = 0$), whereas differs from the PI one, obtaining a higher gain, when the error is different from zero. The selected rules are reported

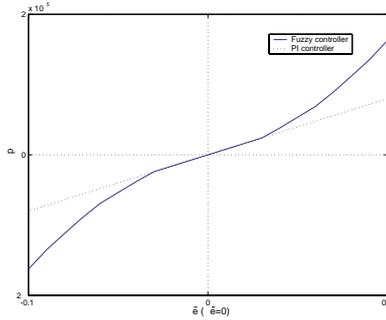


Fig. 3. The comparison between Fuzzy and PI output around $\bar{e} = 0$ and with $\Delta\bar{e} = 0$.

de \ e	MN	N	Z	P	MP
MN	N3	N3	N3	Z	P1
N	N3	N2	N1	Z	P2
Z	N3	N1	Z	P1	P3
P	N2	Z	P1	P2	P3
MP	N1	Z	P3	P3	P3

Table 1. Fuzzy Rules

in table 1 and the comparison between the output variable Δp generated by fuzzy controller and that one generated by the PI, with $\Delta\bar{e} = 0$, is shown in figure 3.

4 Experiments and Performance Evaluation

We have implemented the fuzzy controller as an active queue manager under the well known *ns2* simulator [17]. In this environment we also implemented the PI controller according to the pseudo-code reported in [8].

The network topology adopted in the experiments is reported in figure 4. The network load is generated by FTP and HTTP sources. All the flows are conveyed in the bottleneck link with 15 Mbps bandwidth capacity between the AQM router R_0 and the router R_1 . The routers R_2, R_3, R_4, R_5 are introduced to measure the different FTP and HTTP input and output traffic in the bottleneck link. HTTP flows are short lived flows with a bursty behaviour and can not be easily controlled by the congestion control mechanism. We adopted them as noise traffic. FTP flows, with their intensive data transfers, represent the traffic load to be controlled in the fluid flow dynamic model.

Charts in the left side of figures show the queue lengths obtained by AQM controllers, whereas charts in the right side show the corresponding values of the dropping probability. Figures 5 confirms that PI outperforms RED, as reported

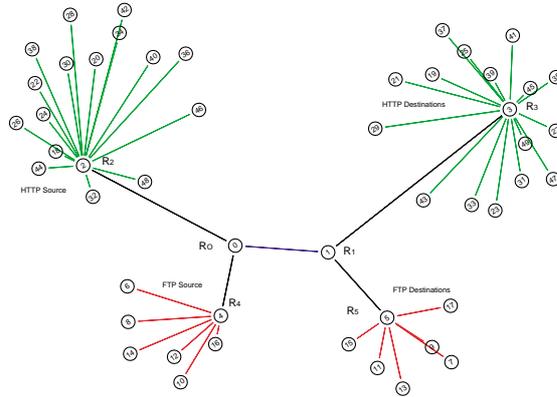


Fig. 4. *The experimental topology*

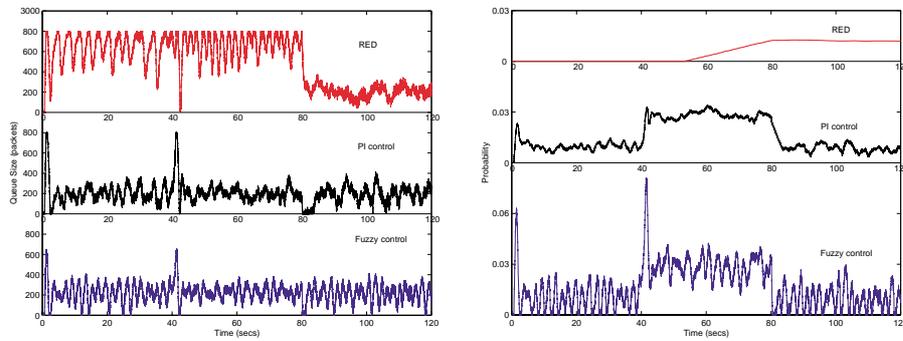


Fig. 5. *RED, PI and Fuzzy Queue Lengths and Dropping Probabilities with 60 additional FTP flows.*

in [8] and shows also the same behaviour of the Fuzzy and PI controllers. The comparison between Fuzzy and PI will be the subject of the remainder of this section. Figures 5 and 6 show that in steady traffic conditions both the PI and Fuzzy controllers are able to maintain the queue length at the reference value (200 packets).

The propagation delay of the paths between sources and destinations is uniformly distributed in the range 160-240 msec. The maximum queue length in the AQM router R_0 is 800 packets. The parameters of the PI and Fuzzy controllers are determined for operating conditions where the number of flows is 60 and the desired value of the queue length is 200 packets.

In all the experiments we adopted a first group of 60 FTP flows (*design conditions*) and 180 HTTP flows (*noise traffic*) which last the whole simulation time (from 0 to 120 seconds) and a second group of additional FTP flows (*over-*

load) which transmit in the time interval between 40 and 80 seconds. We carried out three different experiments, where the additional number of FTP flows are respectively 60 (figure 5), 240 (figure 6.a), and 480 (figure 6.b).

In operating conditions near to the design ones (figure 5) both the controllers have similar performances, as expected. The effectiveness of the Fuzzy controller becomes evident when a greater increment of the load brings the operating point far away from the one considered in the design (figures 6.a and 6.b). Specifically, in such cases the Fuzzy controller shows a lower time of response than the PI one. Furthermore, when the additional load stops at 80 seconds, the PI controller brings the queue length to zero for a longer time than the Fuzzy one, thus producing a longer underutilization of the link. This phenomenon is confirmed by the faster response of the Fuzzy control variable (the dropping probability) shown in the charts on the right.

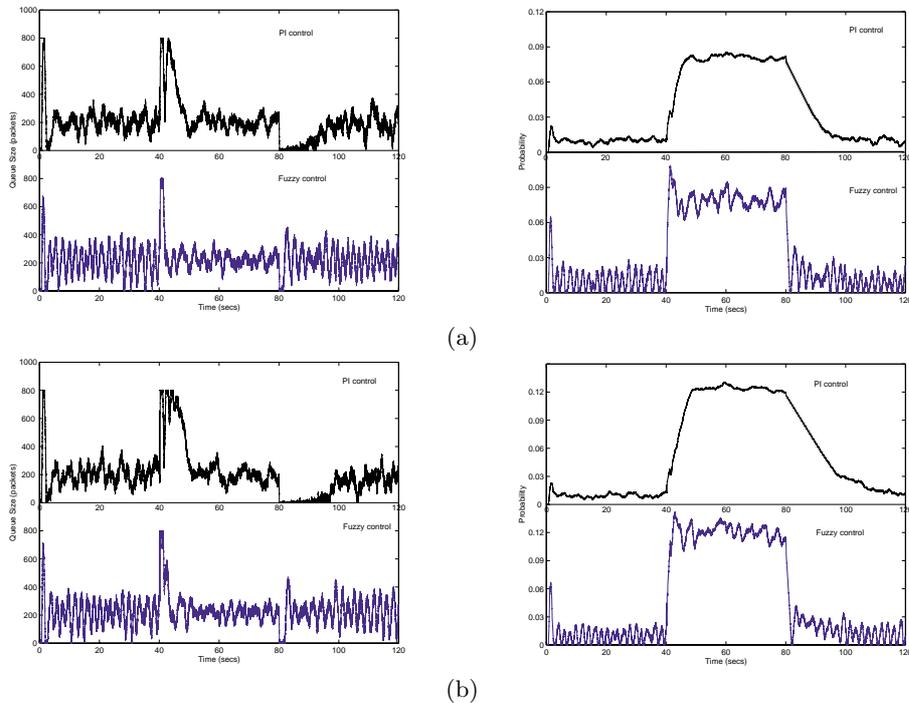


Fig. 6. PI and Fuzzy Queue Lengths and Dropping Probabilities with 240 (a), and 480 (b) additional FTP flows.

5 Conclusions

In this paper we developed a Fuzzy controller for active queue management. The design has been carried out in analogy with a PI controller and the experiments showed that it outperforms the PI one in situations of functioning different than those related to the operating point of the design. Such a better behaviour is due to the characteristic of the Fuzzy controller to change its closed loop gain under different operating conditions.

References

1. S. Floyd and V. Jacobson, "Random Early Detection Gateway for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, N. 4, Aug. 1993, pp. 397 - 413.
2. W. Feng, D. Kandlur, D. Saha, K. Shin, A Self-Configuring RED Gateway, *INFOCOM '99*, March 1999
3. T. J. Ott, T. V. Lakshman, and L. H. Wong, SRED: Stabilized RED, *Proceedings IEEE INFOCOM '99*, New York, March 1999.
4. S.H. Low and D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence", *IEEE/ACM Transactions on Networking*, Vol. 7, N. 6, Dec. 1999, pp. 861 - 874.
5. S. Athuraliya, V. Li, S.H. Low and K. Yin, "REM: Active Queue Management", *IEEE Network*, Vol. 15, N. 3, May/June 2001 pp. 48 - 53.
6. C. Hollot, V. Misra, D. Towsley, W. Gong, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", *Proc. of ACM SIGCOMM 2000* Aug. 2000, pp. 151 - 160.
7. C. Hollot, V. Misra, D. Towsley, W. Gong, "A Control Theoretic Analysis of RED", *Proc. of IEEE INFOCOM* Apr. 2001.
8. C. Hollot, V. Misra, D. Towsley, W. Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows", *Proc. of IEEE INFOCOM* Apr. 2001.
9. E. H. Mamdani, "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant", *Proc. Inst. Elect. Eng.*, vol. 121, no. 12, pp. 1585-1588, 1974.
10. D. A. Rutherford, G. C. Bloore, "The Implementation of Fuzzy Algorithms for Control", *Proc. IEEE*, vol 34, no. 4, pp. 572-573, 1976.
11. C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller -Part I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404-435, 1990.
12. Jian-Xin Xu, Chang-Chieh Hang, Chen Liu, "Parallel Structure and Tuning of a Fuzzy PID Controller", *Automatica* 36 (2000) pp. 673-684.
13. Misir D., Malki H. A., Chen G., "Design and Analysis of a Fuzzy Proportional-Integral-Derivative Controller", *Fuzzy Sets and Systems* 79 (1996) pp. 297-314.
14. Li W., "Design of a Hybrid Fuzzy Logic Proportional plus Conventional Integral-Derivative Controller", *IEEE Trans. on Fuzzy Systems* 6 (4) (1998) pp. 449 - 463.
15. Malki H. A., Li H. D., Chen G. R., "New Design and Stability Analysis of Fuzzy Proportional-Derivative Control System", *IEEE Trans. on Fuzzy Systems* 2 (4) (1994) pp. 245 - 254.
16. Alonge F., D'Ippolito F., Raimondi F. M., Urso A., "Method for Designing PI-Type Fuzzy Controllers for Induction Motor Drives", *IEE Proceedings on Control Theory and Applications*. vol. 148, n. 1, January 2001, p. 61-69
17. ns-2, network simulator (ver. 2), <http://www.isi.edu/nsnam/ns/>