15th Triennial World Congress, Barcelona, Spain

FUZZY TCP: A PRELIMINARY STUDY

Pablo Carbonell *,1 **Zhong-Ping Jiang** **,2 **Shivendra Panwar** **,3

* *Dept. of Systems Eng. and Control, Polytech. Univ. of Valencia, Plaça Ferràndiz i Carbonell, 2, E-03801 Alcoi, SPAIN*
** *Dept. of Electrical and Computer Eng. and Center for Advanced Technology in Telecommunications (CATT), Polytechnic University, Six Metrotech Center, Brooklyn, NY 11201, USA*

Abstract: Implementing efficient TCP for the Internet necessarily has to cope with the problem that the source does not know in advance which window allocation policy should be the best to use for a given network condition. In this paper, an on-line adaptive fuzzy system is used at the source in an objective to find the best possible weighted combination among the available policies.

Keywords: TCP, congestion control, fuzzy logic, communication protocols.

1. INTRODUCTION

Many successful applications of the control of complex systems based on fuzzy logic have been reported in recent years (Passino and Yurkovich, 1996), for instance auto-focus cameras, chemical processes, home appliances, automatic driving of vehicles, etc. The widespread application of fuzzy control is due to the ability of this technique to encapsulate the knowledge of any *ad hoc* or heuristics based control rules. Moreover, fuzzy logic can generalize the control procedure derived from these rules and it allows automatic fine tuning, usually with remarkable improvements on the performance of the controlled system.

Fuzzy logic-based control of telecommunications systems and networks is an emergent application which recently has been addressed in the scientific literature. A survey of recent advances can be found in (Ghosh *et al.*, 1998). Some applications are: modelling of queuing systems, access control of battlefield communications networks, fuzzy logic-based policing mechanisms. Special mentioning deserved for the issues of congestion and call admission control in ATM-based networks, where many fuzzy logic-based proposals have appeared in recent years (Douligeris and Develekos, 1997).

This paper is focused on congestion control using fuzzy control for end-to-end TCP. Empirical evidence has shown deficiencies in the performance, including cyclic behavior, for the standard Internet TCP congestion avoidance algorithm proposed by Jacobson (Jacobson, 1988). The approach here presented is different from other previous work on fuzzy TCP as in (Loukas *et al.*, 2000), (Gan *et al.*, 1999), which were mainly devoted to a fuzzy random early detection (RED).

In the next section, some relevant aspects of the fuzzy logic theory will be reviewed. The general control problem of TCP is then stated, following an analysis of suitability of the fuzzy logic approach. Next, the proposal is formalized. The paper will end with the study of a simple TCP case, along with some simulations we carried out.

[1] This work was done when this author was visiting the ECE Dept. at Polytechnic University under a grant from the New York State Center for Advanced Technology in Telecommunications (CATT). It has also been partially supported by Polytechnic University of Valencia and by Generalitat Valenciana under Grant GV00-088-14.
[2] Supported in part by the NSF under Grants INT-9987317, ECS–0093176 and ANI-0081527.
[3] Supported in part by the NSF under Grant ANI-0081527.

2. MOTIVATION

The widely accepted congestion avoidance algorithms for the Internet Transmission Control Protocol (TCP) is based on a variable-structure window size update law. For instance, we can consider the simplified version of Jacobson's congestion avoidance algorithm:

$$\text{IF (CONGESTION IS ON) THEN } W = W/2 \qquad \text{ELSE } W = W + 1/W$$

where, as congestion is detected, the multiplicative decrease is applied to the window size; otherwise, an additive increase is used.

The strategy above assumes that: a) there are just two possible network states: NORMAL OPERATION and CONGESTION; b) the network state can be inferred from the feedback information; and c) the window updates will drive eventually the source to the optimum throughput.

This previous logic rule can also be expressed in terms of the following expression:

$$W_k = W_{k-1} + \omega_i f_i(W_{k-1}) + \omega_d f_d(W_{k-1}) \qquad (1)$$

where f_i and f_d stand for the increase and decrease update laws, and ω_i, ω_d are logic values which represent the normal and congestion operating modes respectively and they are mutually exclusive, that is: $\omega_i, \omega_d \in \{0,1\}$; $\omega_i = \overline{\omega}_d$.

Unfortunately, the Internet congestion appears to be very complex and a simple update law like (1) does not always give satisfactory results. From the observation of the network behavior what can be found is that: a) there are many complex possible network states; b) the feedback information: duplicate ACK and time-outs, do not provide enough information in order to determine the exact current network state; c) the ad-hoc window update laws of additive increase and multiplicative decrease, even in cases when they bring the system to a stable mode, do not necessarily correspond to the optimum throughput at the source.

From the example above what is apparent is that if binary logic update rules are used, then, there is no possibility of tuning because just one of the increase or decrease laws can be selected at each update time.

A fuzzy logic rule can be stated in the same fashion as that of the example, though now the states NORMAL and CONGESTION are represented by means of a fuzzy set, and they will have associated fuzzy membership functions. So, any notification from the feedback information will result in a weighted combination of both update laws.

The hard part of this approach is how to find the membership functions for every state that better describe the network condition and, furthermore, that should be capable of driving the system to the optimum throughput. In order to do that we should define an objective function with a global maximum. As regards the membership functions, two alternatives are possible: non-adaptive (off-line tuning) or adaptive (on-line tuning). The first ones have less computation cost, but at the expense of poorer global results. In the next section, the problem will be stated in a general form.

3. FUZZY TCP

As it has been shown in the previous section, congestion avoidance is in essence a selection process of the best window size update for a given network condition observed from some feedback data. In the approach presented in this paper, this selection will actually be a fuzzy selection, that is a weighted combination of the update laws inferred from the defuzzification of a set of fuzzy rules.

In our proposed fuzzy TCP scheme, the window size update law is selected from a set of n different available allocation policies:

$$W_k = W_{k-1} + \sum_{i=1}^{n} \omega_i(p(kT),\theta) f_i(W_{k-1}) \qquad (2)$$

where $p(t)$ corresponds to the network feedback notifications: duplicate acknowledgements, time-outs, etc.; $f_i(W_{k-1})$ are the available allocation policies at the source: additive increase, multiplicative decrease, slow-start, fast retransmit and fast recovery, etc.; and $\omega_i = \omega_i(p,\theta)$ stand for the fuzzy basis functions (FBF) (Wang, 1993) with adjustable parameters θ and so that:

$$0 \leq \omega_i \leq 1 \quad ; \quad \sum_{i=1}^{n} \omega_i = 1 \qquad (3)$$

In any of the current TCP implementations, the functions $\omega_i(p,\theta)$ are nonadaptive, mutually exclusive and binary. So, in terms of logic, the rules governing (1) are based on binary or bivalent logic. In the method proposed here, fuzzy logic is used instead of binary logic. So, there will be a gradation between 0 (FALSE) and 1 (TRUE) of the value of the functions ω_i in (3).

The TCP is therefore governed by means of a non-linear Takagi-Sugeno fuzzy controller (Passino and Yurkovich, 1996) with the following fuzzy rule base:

$$R_i: \quad \text{IF } p(kT) \text{ is } P_i \quad \text{THEN } \Delta W_k = f_i(W_{k-1}), \quad i = 1 \ldots n \qquad (4)$$

Let $\mu_i(p(t))$ be the membership function associated with the fuzzy set P_i. These functions can be interpreted as the degree of likelihood of the network being at the operating mode represented by P_i (NORMAL, CONGESTION, etc.), for the current observed network conditions $p(t)$.

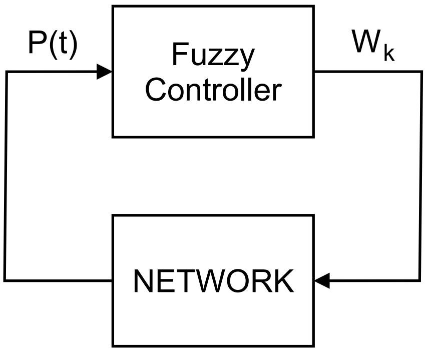

Fig. 1. The network feedback loop.

In order to verify (3), functions μ_i should be normalized to ω_i (FBF):

$$\omega_i(p) = \frac{\mu_i(p)}{\sum_{i=1}^{n} \mu_i(p)} \tag{5}$$

4. THE OPTIMIZATION PROCESS

The objective of the fuzzy controller in Figure 1 is to find the optimal combination of the update laws. This optimality can be expressed by using a function objective U:

$$U = \phi_a(X_a) - \phi_l(X_l) \tag{6}$$

where $\phi_a(X_a)$ is, as in (Kelly, 2001), the *utility* function, an increasing and strictly concave function of the rate of acknowledged packets X_a, and ϕ_l is the *cost* function, an increasing and strictly convex function of the rate of lost packets X_l. The maximum of this function, that is, the point where the utility most exceeds the cost, is the optimum.

The instantaneous rate of lost packets X_l should be determined from the feedback network information $p(t)$ (duplicate acks, time-out, etc.), while the instantaneous rate of ack packets X_a can easily be derived afterwards:

$$X_a(kT) = X(kT) - X_l(p(kT)) \tag{7}$$

where $X(kT) = \frac{W_k}{T}$ is the total instantaneous throughput.

Problem statement Given an observed network conditions $p(kT)$, adapt the parameters θ of the FBF $\omega_i(p(kT), \theta)$ in order to allocate the window size W which maximizes the objective function U.

The problem stated above is a nonlinear programming problem. It is necessary to use a search technique, like the steepest ascent gradient method:

$$\theta_k = \theta_{k-1} + \sigma \left.\frac{\partial U}{\partial \theta}\right|_k \tag{8}$$

σ is the learning rate and it can be made variable if it is used a variable step-size algorithm.

The gradient of the objective function $\frac{\partial U}{\partial \theta}$ in (8) has to be estimated. By using the chain rule, and assuming a constant round-trip time, the gradient can be decomposed as follows:

$$\begin{aligned} \left.\frac{\partial U}{\partial \theta}\right|_k &= \left.\frac{\partial U}{\partial X_a}\right|_k \left.\frac{\partial X_a}{\partial \theta}\right|_k - \left.\frac{\partial U}{\partial X_l}\right|_k \left.\frac{\partial X_l}{\partial \theta}\right|_k \\ &= \left.\frac{\partial U}{\partial X_a}\right|_k \left.\frac{\partial X}{\partial \omega}\right|_k \left.\frac{\partial \omega}{\partial \mu}\right|_k \left.\frac{\partial \mu}{\partial \theta}\right|_k \\ &\quad - \left[\left.\frac{\partial U}{\partial X_a}\right|_k + \left.\frac{\partial U}{\partial X_l}\right|_k \right] \left.\frac{\partial X_l}{\partial \omega}\right|_k \left.\frac{\partial \omega}{\partial \mu}\right|_k \left.\frac{\partial \mu}{\partial \theta}\right|_k \end{aligned} \tag{9}$$

$$\left.\frac{\partial X}{\partial \omega}\right|_k = \frac{1}{T} \frac{\partial W_k}{\partial \omega} \tag{10}$$

$$\frac{\partial W_k}{\partial \omega} = f(W_{k-1}) \tag{11}$$

$$\left.\frac{\partial \omega}{\partial \mu}\right|_k = \frac{1}{\sum \mu_i} - \frac{\mu}{(\sum \mu_i)^2} = \frac{\sum \mu_i - \mu}{(\sum \mu_i)^2} \tag{12}$$

where, for the sake of clearness, the membership function indices i have been dropped.

The analytic expression $X_l(p(kT))$ is unknown, so this gradient has to be estimated. The simplest estimate that can be used is derived from the method of finite differences:

$$\left.\frac{\partial X_l}{\partial \omega}\right|_k \approx \begin{cases} \dfrac{X_{lk} - X_{lk-1}}{\omega_k - \omega_{k-1}} & |\omega_k - \omega_{k-1}| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The main advantage of this estimate is its simple algorithm, however the estimation is very noise sensitive. So, it may not be very accurate in a practical implementation, and other gradient estimators should be employed, as those based on stochastic perturbation (Spall, 1998).

If the round-trip time is not constant, a new unknown term $\frac{\partial T}{\partial \omega}$ will appear in (10). It can be estimated in the same way as (13).

5. A SIMPLE TCP CASE

In this section, a simple case is considered based on Jacobson's algorithm with just two update policies: additive increase and multiplicative decrease (no FRFR, ECN (Floyd, 1998), RED (Floyd and Jacobson, 1993), etc.), with round-trip time T constant. As shown in (Kelly, 2001), the window size update behavior can be approximated by means of a continuous ow model:

$$W_k = W_{k-1} + (1 - p(t)) - p(t)\frac{W_{k-1}^2}{2} \tag{14}$$

where $p(t)$ is the probability of congestion, the first term stands for the additive increase policy, whereas the second term stands for the multiplicative decrease policy. In the fuzzy approach, the following equation replaces this last expression:

$$W_{k-1} = W_{k-1} + \omega_i - \omega_d \frac{W_{k-1}^2}{2} \quad (15)$$

where now the ω_i, ω_d functions are the fuzzy basis functions (5).

The associated membership functions will depend on all the available information from the network condition. In a typical TCP Reno application, that information comprises: Source state: window size W_k; Estimated round-trip time: T_k; Duplicate acknowledgments; Time-out.

It is assumed that a way to estimate the instantaneous lost packet rate X_l from the above information has been already implemented at the source. A new parameter is defined as $n = \frac{X_l}{X}$, i.e. the estimated rate of lost packets.

If the fuzzy controller that is to be used is nonadaptive, i.e., the tuning of the membership functions is carried out off-line, then it suffices to choose a simple asymmetrical z-shaped and s-shaped fuzzy membership functions pair. They are defined as follows:

$$\mu_i(n) = 1 - \mu(n, n_{i1}, n_{i2}) \quad (16)$$

$$\mu_d(n) = \mu(n, n_{d1}, n_{d2}) \quad (17)$$

$$\mu(x,\theta_1,\theta_2) = \begin{cases} 0, & x \le \theta_1 \\ 2\left(\dfrac{x-\theta_1}{\theta_2-\theta_1}\right)^2, & \theta_1 < x \le \theta_0 \\ 1-2\left(\dfrac{\theta_2-x}{\theta_2-\theta_1}\right)^2, & \theta_0 < x \le \theta_2 \\ 1, & x > \theta_2 \end{cases}$$

where $\theta_0 = \frac{\theta_2-\theta_1}{2}$, and μ_i and μ_d are the corresponding membership functions for the NORMAL and CONGESTION modes. Their shapes are depicted in Figure 2.

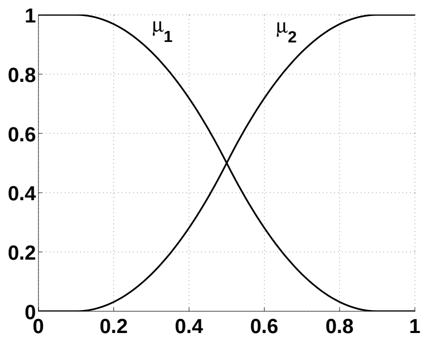

Fig. 2. Membership functions for simple TCP .

If a nonadaptive fuzzy controller does not give good results, adaptation or online learning is needed. An adaptive fuzzy controller requires continuous membership functions with strictly positive values for the domain of n, so that any change in the congestion indicator n can be detected because of a corresponding change in the membership value. A very common option is to use the sigmoid function, which is defined as follows:

$$\mu_i = \mu(1-n, c_{i0}, \sigma_{i0}) \quad (18)$$

$$\mu_d = \mu(n, c_{d0}, \sigma_{d0}) \quad (19)$$

$$\mu(x,\theta_1,\theta_2) = \frac{1}{1+\exp(-\theta_2(x-\theta_1))}$$

where μ_i and μ_d are the initial functions associated with the NORMAL and CONGESTION operation respectively. The parameters θ_1 and θ_2 are to be updated every round-trip time by the adaptive law (8).

6. SIMULATIONS

In order to show the crucial role played by the suitability of the parameters tuning in the fuzzy membership functions, a set of simulations has been carried out to compare the results obtained by the nonadaptive and the adaptive fuzzy controllers. In the case of the nonadaptive controller, two different parameter values of the membership functions in (18) are tested, as it is shown in Table 1. The initial values for the adaptive controller are also shown in this table.

Table 1. Parameter values.

	n_{i1}	n_{i2}	n_{d1}	n_{d2}
Case 1	0.1	0.9	0.1	0.9
Case 2	0.4	0.6	0.4	0.6
	c_{i0}	σ_{i0}	c_{d0}	σ_{d0}
Adaptive	0	50	1	50

The objective function (6) used in this simulations has some similarities with that used in (Kelly, 2001) and it is defined as follows:

$$U(t) = \frac{2}{\pi}\arctan(X_a) - n^2 \quad (20)$$

The first study case is a hypothetical situation where initially there is no congestion signal but after some time has elapsed ($t = 50$ s) a fixed and independent signal of $n = 0.2$ is detected. This can be interpreted as a situation when most of the bandwith is shared with other users, so that the change in window size at the source will not actually modify the congestion status. In the second case considered, the ow is limited up to a window size $W_0 = 50$. Therefore, any packet sent above this size will be lost.

In Figure 3 the evolution of the objective function (6) $U(t)$ is shown. Before the congestion signal is detected, the fuzzy TCP and Reno algorithms have the same increasing evolution. After the congestion starts, Reno and fuzzy case 1 offer poor results, while fuzzy case 2 and the adaptive fuzzy controller are able to soften the loss.

Figure 4 shows the results obtained for the second study case. In this case, the network cannot accept any window size above $W_0 = 50$, therefore congestion is detected when the window size exceeds this value. TCP Reno shows a good behavior, although nonadaptive fuzzy case 1 and the adaptive case yield

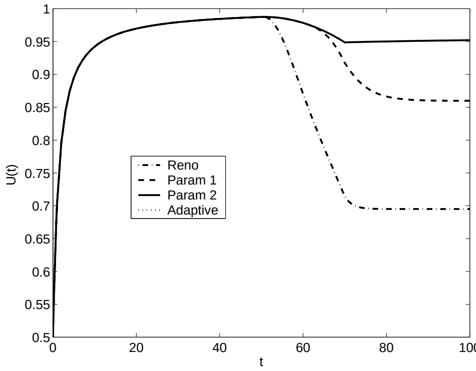

Fig. 3. Values of the objective function U for different TCP schemes when a fixed congestion $n = 0.2$ is detected.

similar results. It is obvious that in this situation the best solution is to keep the congestion at level $n = 0$, because any increase in the window size would reduce the value of the objective function U. The worst result corresponds to nonadaptive fuzzy case 2, because the centers of the membership functions are too close to detect a small congestion signal.

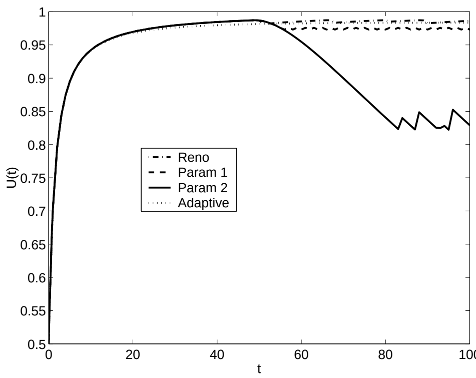

Fig. 4. Values of the objective function U for different TCP schemes for a maximum window size $W_0 = 50$.

Finally, in Figure 5 the evolution on the shape of the membership functions for the adaptive controller is shown. When a fixed congestion signal is detected, the function associated with the normal operation is kept around its original position, whereas the congestion function is shifted into the full congestion area. The reason for this result is that in the present network condition the decreasing action should be kept close to zero, while the increasing action, that is already at full value, does not need any further adaptation. When the window size has a maximum value, both functions are shifted to the low-congestion area, that is, the increasing region is progressively reduced whereas the decreasing region is enlarged until an equilibrium point is reached.

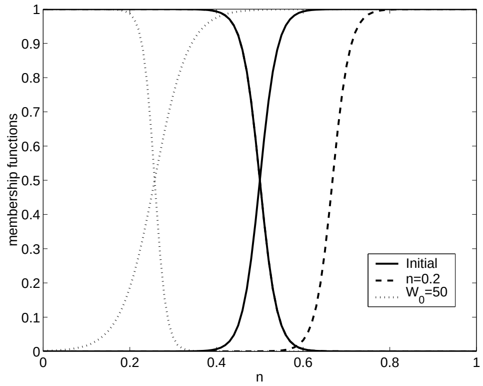

Fig. 5. Initial distribution of the membership functions and after adaptation.

7. FUZZY SLOW-START

In this section, we outline the design process for considering the slow-start algorithm implementation within a fuzzy framework.

In the slow start phase, two update laws have to be considered:

$$f_{ss0}(W_k) = -W_k + 1 \tag{21}$$

$$f_{ss1}(W_k) = W_k \tag{22}$$

f_{ss0} stands for the initial setting of the window to 1. f_{ss1} stands for the exponential window increase.

In TCP Reno, when the window size is less than ssthresh or when a timeout has occurred, the slow-start algorithm is initiated and performed until the window size arrives at ssthresh, when the congestion avoidance phase is resumed. We keep this binary and exclusive condition can be kept.

$$\text{IF TIMEOUT OR (W} < \text{ssthresh)} \qquad \text{THEN } W_k = W_{k-1} + f_{ss0}(W_{k-1}) \tag{23}$$

From a fuzzy perspective, we can consider a *fuzzy ssthresh*, like in Figure 6, where there can be found three different regions: for low values of the window size, the system is in the slow-start phase, for large values of the window size, the system is in the congestion avoidance phase, and for values around ssthresh, the system is in a transition region, where a weighted combination of both laws is performed. The sstresh value in Figure 6 can be adapted (as actually is carried out in TCP Reno). Furthermore, the span of the transition region can be adapted as well.

8. CONCLUSIONS

Due to the complexity of the congestion control problem in TCP for the Internet, fuzzy inference of the window size update law at the source seems to be a

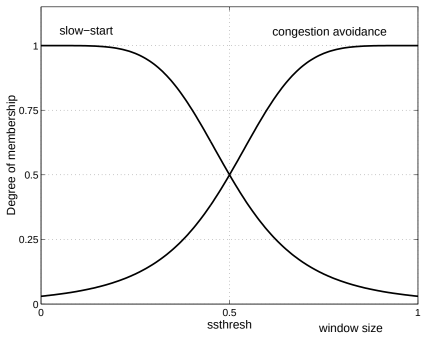

Fig. 6. Fuzzy slow-start threshold.

better solution than conventional TCP based on logical rules. This new approach has not been previously reported in the literature.

The application of a fuzzy logic-based control technique turns the design problem into an optimization process. The simulations carried out have shown that a nonadaptive fuzzy update law cannot perform adequately at some of the many different possible complex states of the network. Therefore, parameter adaptation is needed in order to allow the membership functions to reshape as a result of a change in the network condition. From the preliminary results obtained in the simulations, it can be concluded that the proposed Fuzzy TCP can perform better than the conventional TCP. Although the example given is a simplified version of TCP, the proposed algorithm can be easily extended to more general cases.

The object of our current research work is twofold: a theoretical analysis of the structural conditions of the fuzzy controllers that guarantee convergence, and a study of fairness, friendliness and optimality for the whole or a part of the network controlled with the proposed fuzzy algorithm. Furthermor e, the proposed controller is going to be tested with a more realistic simulator, where the use of a stochastic estimator for the gradient will be required.

9. REFERENCES

Douligeris, Christos and George Develekos (1997). Neuro-fuzzy control in atm networks. pp. 154–162.

Floyd, Sally (1998). Tcp and explicit congestion notication. *ACM Computer Communication Review* **4**, 10–23.

Floyd, Sally and Van Jacobson (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*.

Gan, Meng, Elmar Dorner and Jochen Schiller (1999). Applying computational intelligence for congestion avoidance of high-speed networks. In: *Proceedings of the 7th Workshop on Future Trends of Distributed Computing Systems*. pp. 23–28.

Ghosh, Sumit, Qutaiba Razouqi, H. Jerry Schumacher and Aivars Celmins (1998). A survey of recent advances in fuzzy logic in telecommunications networks and new challenges. *IEEE Trans. on Fuzzy Systems* **6**(3), 443–447.

Jacobson, Van (1988). Congestion avoidance and control. In: *Proceedings of the ACM SIGCOMM'88*. pp. 314–329.

Kelly, Frank (2001). *Mathematics Unlimited - 2001 and Beyond*. Chap. Mathematical modelling of the Internet, pp. 685–702. Springer-Verlag. Berlin.

Loukas, Rossides, Stefan Kohler, Pitsillides Andreas and Tran-Gia Phuoc (2000). Fuzzy red: Congestion control for tcp/ip diff-serv. In: *Proceedings of the 10th Mediterranean Electrotechnical Conference*. pp. 19–22.

Passino, Kevin M. and Stephen Yurkovich (1996). *The Control Handbook*. Chap. Fuzzy Control, pp. 1001–1017. CRC Press. Boca Raton, Fla.

Spall, James C. (1998). Adaptive stochastic approximation by the simultaneous perturbation method. In: *Proceedings of the 1998 IEEE Conference on Decision and Control*. Tampa, FLO. pp. 3872–3879.

Wang, Li-Xin (1993). Stable adaptive fuzzy control of nonlinear systems. *IEEE Transactions on Fuzzy Systems* **1**(2), 146–155.