Design and Application of Intelligent Control Schemes for the Performance Enhancement of a Web Server

Malik Loudini, Sawsen Rezig, Walid-Khaled Hidouci, Yahia Salhi

Ecole Nationale Supérieure d'Informatique (ESI), Laboratoire de Communication dans les Systèmes Informatiques (LCSI),
B.P 68M, 16270 Oued Smar, El Harrach, Algiers, Algeria
(e-mail:m_loudini@esi.dz, s_rezig@esi.dz, w_hidouci@esi.dz, yah_alg@yahoo.fr)

Abstract: This paper considers the design problem of efficient feedback control schemes to enhance the quality of service (QoS) of a web server (WS) whose input/output dynamic behavior is modeled by discrete mathematical representations. Discrete models allowing the digital simulation of the WS responses to different scenarios of client requests are adopted. The capabilities to guarantee performing service delays, under dynamic workload variations, are the main investigations of this work. To try to provide prospective solutions, advanced feedback control strategies are proposed. First, a fuzzy logic controller (FLC) is implemented as a regulating solution in a closed-loop control architecture. Then, the tabu search (TS) algorithm (TSA) is used to optimize the FLC parameters with innovative tuning procedures. The TS optimized FLC (TSOFLC) is also implemented and applied to attempt to improve the WS QoS. To demonstrate the effectiveness of the adopted closed-loop intelligent control strategies, digital simulation experiments are carried out and examined.

Keywords: Quality of service, Web server, differentiated service, service delay guarantee, absolute delay (AD), relative delay (RD), difference equation (DE), fuzzy logic controller, tabu search.

1. INTRODUCTION

Applying feedback control schemes to enhance the performance of software processes is becoming an attractive research area. Indeed, the main advantage offered by feedback control is its robustness to modeling inaccuracies, system nonlinearities, and time variation of system parameters. These types of uncertainties are very common in unpredictable poorly modeled environments such as the Internet. Detailed discussions about the application of feedback control to computing systems can be found in (Abdelzaher *et al.,* 2003; Hellerstein *et al.,* 2004; Abdelzaher et al., 2008; Parekh, 2010).

Fuzzy logic (Zadeh, 1988) based controllers (Lee, 1990) are well known for their ability to adapt to dynamic imprecise and bursty environments such that of the web traffic. Indeed, it is well known that web workloads are stochastic with significant parameter variations over time. So, a challenging problem is how to provide efficient and realistic performance control over a wide range of workload conditions knowing the highly nonlinear behavior of a WS in its response to the allocated resources.

In this work, WS QoS improvement solutions based on fuzzy logic based intelligent control strategies are investigated.

This paper is organized as follows. In Section 2, we briefly describe how web servers (WSs) operate. We also introduce the QoS and the semantics of delays and service delay guarantees in WSs. In Section 3, the dynamic mathematical modeling of the WS system is described and different discrete models are given. In Section 4, we present the adopted global feedback control strategy aimed to satisfy the desired performances of the WS. The derivation and implementation details of the proposed intelligent control strategies are presented in Section 5. In Section 6, we present the simulation results and discuss the obtained performances. Section 7 gives an overview of the related work. Finally, Section 8 concludes the paper.

2. QOS IN WEB SERVERS

WSs are commonly defined as computers that deliver web pages. Having an IP address and generally a domain name, a WS is software responsible for accepting HTTP (Gourley and Totty, 2002) requests from clients and offering them services as HTTP responses. HTTP lies behind every web transaction. A HTTP transaction consists of three steps: TCP (Kozierok, 2005) connection setup, HTTP layer processing and network processing. The TCP connection setup is performed through a three way handshake, where the client and the server exchange TCP SYN, TCP SYN/ACK and TCP ACK messages. Once the connection has been established, the client sends a request for an object (static HTML files, image files, various script files …). The WS handles the request and returns the object or the results of these queries. Finally, the TCP connection is closed by sending TCP FIN and TCP ACK messages in both directions (Andersson, 2005).

It is well known that WSs adopt either a multi-threaded or a multi-process model to handle a large number of users simultaneously. Processes or threads can be either created on demand or maintained in a pre-existing pool that awaits incoming TCP connection requests to the server. In HTTP 1.0, each TCP connection carried a single web request. This resulted in an excessive number of concurrent TCP connections. To remedy this problem the improved version of HTTP, called HTTP 1.1 (Fielding *et al.,* 1999), reduces the number of concurrent TCP connections with a mechanism called persistent connections, which allows multiple web requests to reuse the same connection (Lu *et al.,* 2001).

As pointed out in (Lu *et al.,* 2006), a multi-process model with a pool of processes is assumed, which is the model of Apache, the most commonly used WS today (Netcraft, 2013).

Various QoS techniques have been suggested to enhance the internet service levels. Among these, two major mechanisms or classes of services have emerged: the Integrated Services (IntServ) and the Differentiated Services (DiffServ) which have been proposed by the IETF Differentiated Services Working Group (Braden *et al.,* 1994; Blake *et al.,* 1998).

DiffServ is a computer networking protocol or architecture that allows different levels of services on a common network in order to provide a better QoS. In other words, it supports a manageable and scalable service differentiation for class-based aggregated traffic in IP networks.

Two approaches exist in DiffServ architecture:

Absolute DiffServ: This model seeks to guarantee end-to-end QoS. In this architecture, the user receives an absolute service profile and the network administrator attempts to maintain the absolute metric spacing between the users classes.

Relative DiffServ: This model seeks to provide relative or proportional services. In other words, it aims to guarantee to a higher priority class of users better (proportionally ratioed) service performances than those provided to a lower priority class.

Every HTTP request being supposed to belong to a class k $(0 \le k < N)$, two main delays are defined as:

Connection delay: It is the time interval between the arrival of a TCP connection (establishment) request and the time where the connection is accepted (dequeued) by a server process. The connection delay includes the queuing delay.

Processing delay: It is the time interval between the arrival of an HTTP request to the process responsible for the corresponding connection and time the server completes transferring the response.

In other words, the connection delay of class k at the m^{th} sampling instant, denoted by $C_k(m)$, is defined as the average connection delay of all established connections of class k within the time interval $[(m-1)T_s, mT_s]$, where T_s is a constant sampling period.

The delay differentiation being applied to connection delays, the adopted QoS metrics in this work are the connection delay guarantees. Using, for simplicity, delay to refer to connection delay, they are defined as follows:

Relative delay guarantee: A desired RD W_k is assigned to each class k. A RD guarantee $\{W_k \mid 0 \le k < N\}$ requires that $C_j(m)/C_l(m) = W_j(m)/W_l(m)$ for classes j and l $(j \ne l)$.

Absolute Delay Guarantee: A desired AD W_k is assigned to each class k. An AD guarantee $\{W_k \mid 0 \le k < N\}$ requires that $C_j(m) \le W_j(m)$ for any class j if there exists a lower priority class $l > j$ and $C_l(m) \le W_l(m)$ (a lower class number means a higher priority). Note that since system load can grow arbitrarily high in a WS, it is impossible to satisfy the desired delay of all service classes under overload conditions. The AD guarantee requires that all classes receive satisfactory delay if the server is not overloaded; otherwise desired delays are violated in the predefined priority order, i.e., low priority classes always suffer guarantee violation earlier than high priority classes.

3. SIMULATION MODELS

The systematic design of feedback systems requires an ability to quantify the effect of control inputs (e.g., buffer size) on measured outputs (e.g., response times), both of which may vary with time. Indeed, developing such models is at the heart of applying control theory in practice (Hellerstein *et al.,* 2004). The models obtained are also used to make numerical simulations as needed in this work.

Our control schemes will be tested based on the WS dynamic models established in (Lu *et al.,* 2006) based on a statistical (black-box) method. The deriving process is referred to as system identification (Ljung, 1999).

As described in (Lu *et al.,* 2006), the server queues can be considered as integrators of flow (which gives rise to difference equations). Therefore, the system to be controlled was modeled as a DE with unknown parameters.

The WS has been stimulated with pseudo-random digital white-noise input and a least squares estimator (Ljung, 1999) was used to estimate the model parameters. Details about the conducted experiments and the obtained results can be found in (Lu *et al.,* 2006). The authors established that the WS system can be modeled as a second order DE with adequate accuracy for the purpose of control design. A brief presentation is given hereafter.

The WS is modeled as a DE with unknown parameters, i.e., an n^{th} order model described by:

$$V(m) = \sum_{j=1}^{n} a_j V(m-j) + \sum_{j=1}^{n} b_j U(m-j). \tag{1}$$

In an n^{th} order model, there are $2n$ parameters $\{a_j, b_j \mid 1 \le j < n\}$ that need to be decided by the least squares estimator.

The DE model reflects the fact that the actual output of the open-loop system $(V(m))$ depends on previous inputs $(U(m-j))$ and outputs $(V(m-j))$ (i.e., recent and previous delays are correlated and depend on the recent allocated process budgets).

To stimulate the dynamics of the open-loop server, a pseudorandom digital white noise generator has been used to randomly switch two classes' process budgets between two input values. The input values to the white noise generator are selected based on the estimated range of the control inputs (the process ratio or process budget) at run time.

The identification results established that, the controlled system can be modeled by the following second order DE:

$$V(m) - a_1 V(m-1) - a_2 V(m-2) = b_1 U(m-1) + b_2 U(m-2). \qquad (2)$$

The DE based system model defined by (2) can be easily converted to a description by a discrete transfer function $G(z)$ from the control input $U(z)$ to the output $V(z)$ in the z -domain, given below:

$$G(z) = \frac{V(z)}{U(z)} = \frac{b_1 z + b_2}{z^2 - a_1 z - a_2}. \qquad (3)$$

The stimulation of the WS being carried out based on SURGE (Barford *et al.,* 1998) as the HTTP requests generator, two sets of experiments (two classes of users: 0 and 1) has been conducted, using three workloads (A, B and C) with different user populations, for each of the AD and RD approaches (see Table 1). Note that the variation of user populations (2 classes) is aimed to evaluate the sensitivity of the model parameters to workloads.

For each experience, a DE based dynamic model has been established (after determination of a_1 a_2, b_1 and b_2). The resulting discrete transfer functions, having the form of (3), are given in Table 1.

Table 1. Experiments data and corresponding discrete transfer functions.

		Workload A	Workload B	Workload C
RD case	Class 0	200	150	300
	Class 1	200	250	300
	$G(z)$	$\frac{0.95z - 0.12}{z^2 - 0.74z + 0.37}$	$\frac{2.28z + 0.08}{z^2 - 0.31z + 0.27}$	$\frac{0.47z + 0.21}{z^2 - 0.56z + 0.26}$
AD case	Class 0	100	150	200
	Class 1	400	250	300
	$G(z)$	$\frac{-0.82z - 0.52}{z^2 + 0.13z + 0.03}$	$\frac{-0.36z - 0.15}{z^2 - 0.14z + 0.05}$	$\frac{-0.49z - 0.25}{z^2 - 0.25z + 0.03}$

4. FUZZY LOGIC BASED FEEDBACK CONTROL ARCHITECTURE

The adopted WS fuzzy logic based feedback control architecture is illustrated by Fig. 1.

The main component appearing in this scheme is the Mamdani's PI type FLC (Mamdani, 1974).

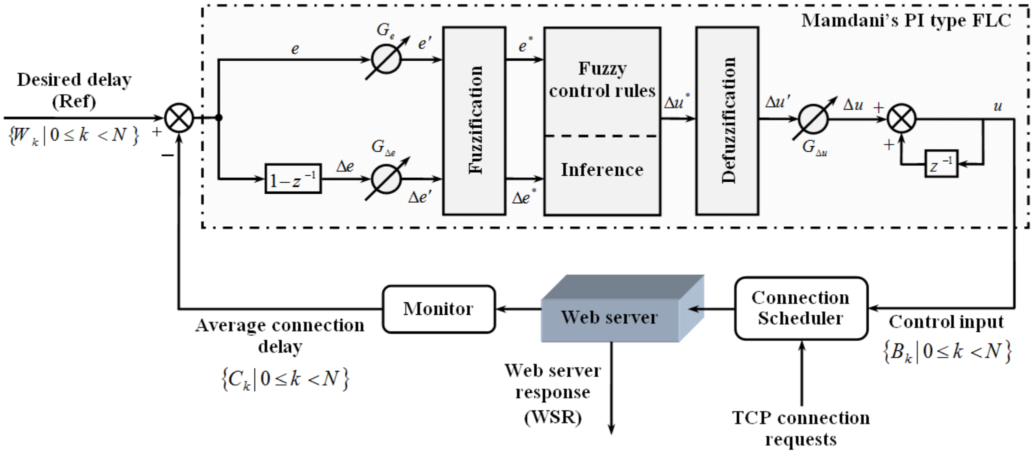

Fig. 1. WS fuzzy logic based feedback control architecture.

The controlled system being the WS, a key component is the connection scheduler. It serves as an actuator for controlling the delays of different classes. It listens to the well-known port, accepts every incoming TCP connection request, and uses an adaptive proportional share policy to allocate server processes to handle TCP connections from different classes.

At every sampling instant m , every class k $(0 \le k < N)$ is assigned a process budget, $B_k(m)$. The connections from class k should be served by at most $B_k(m)$ server processes at any time instant in the m^{th} sampling period (Lu *et al.,* 2006).

At each sampling instant, the connection scheduler transmits the control input effort in terms of process budgets $\{B_k \mid 0 \le k < N\}$ generated by the controller based on the error values provided by the feedback loops. These errors result from the comparisons of the desired delays $\{W_k \mid 0 \le k < N\}$ and the measured delays or sampled connection delays $\{C_k \mid 0 \le k < N\}$ computed by the monitor.

For each of the AD and RD approaches, the control key variables are explicitly summarized in Table 2.

Table 2. Main variables of the feedback control architecture for the AD and RD cases.

	Reference W_k	Output C_k (V)	Control input B_k (U)
RD case	Desired delay ratio between class k and $k-1$	Measured delay ratio between class k and $k-1$	Ratio between the process budgets of classes $k-1$ and k
AD case	Desired delay of class k	Measured delay of class k	Process budget of class k

The FLC variables are given as:

- $e(mT_s) = Ref\ (mT_s) - C_k\ (mT_s)$: loop error
- $\Delta e(mT_s) = \dfrac{e(mT_s) - e[(m-1)T_s]}{T_s}$: error's rate of change
- $\Delta u\ (mT_s)$: change in control setting.

where $Ref\ (mT_s)$ is the reference input at the m^{th} sampling instant.

G_e , $G_{\Delta e}$ are the inputs scaling factors and $G_{\Delta u}$ is the output scaling factor.

The PI type fuzzy logic command is given by

$$u(mT_s) = u[(m-1)T_s] + G_{\Delta u} * \Delta u(mT_s). \qquad (4)$$

5. DERIVATION AND APPLICATION OF THE PROPOSED FUZZY LOGIC BASED CONTROL SCHEMES

5.1 Controlling the Web server by a Mamdani PI Type FLC

In order to validate the proposed FLC based control scheme, digital simulations have been carried out on the basis of the adopted discrete-time process transfer functions for each of the AD and RD approaches.

The simulation study has been conducted according to the basic feedback control system architecture shown in Fig. 2, with variations of the WS workloads (A, B, C) for a better effectiveness and robustness evaluation:

- Workload A at $t = 0s$ (initial workload)
- Workload B at $t = 10s$
- Workload C at $t = 20s$.

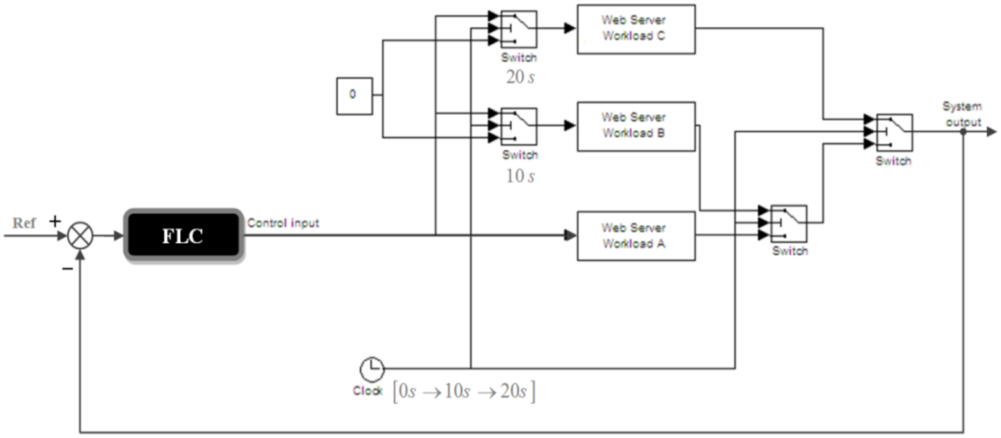

Fig. 2. FLC-based feedback control architecture.

After long series of trial-and-error tests, the following characteristics have been fixed for the two cases of the PI-type Mamdani's FLC based WS control; i.e. absolute service delay and relative service delay guarantees:

- Five fuzzy subsets (FSs) for each of the FLC variables (e , Δe , Δu) given hereafter by their mnemonics and linguistic values in the usual fuzzy logic terminology: Positive Big (PB), Positive Medium (PM), Zero (ZE), Negative Medium (NM), Negative Big (NB).
- The same triangular shapes have been assigned to the membership functions (MFs) of the FLC variables with a uniform distribution and a 50% overlap has been provided for the neighboring FSs (see Fig. 3). Therefore, at any given point of the universe of discourse, no more than two FSs will have non-zero degree of membership. As usual, the universes of discourse for each variable are normalized to the interval [-1,+1].

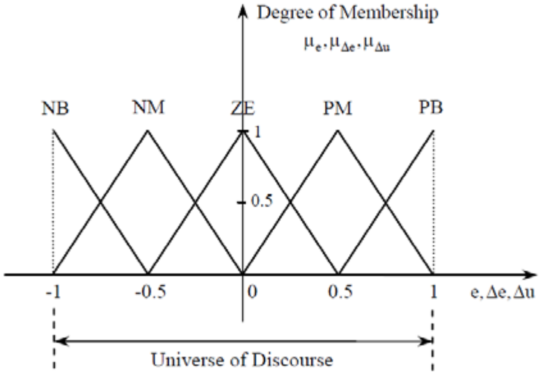

Fig. 3. FLC Membership functions.

- The set of decision rules forming the “rule base” which characterizes our strategy to control the studied dynamic process is organized in a matrix form (see Table 3) based on Mac Vicar-Whelan's diagonal decision table (Mac Vicar-Whelan, 1976).

Table 3. 5X5 Mc Vicar-Whelan decision table.

Δu		e				
		NB	NM	ZE	PM	PB
Δe	NB	NB	NB	NB	NM	ZE
	NM	NB	NB	NM	ZE	PM
	ZE	NB	NM	ZE	PM	PB
	PM	NM	ZE	PM	PB	PB
	PB	ZE	PM	PB	PB	PB

- After a tedious trial-and-error process the scaling factors best values have been determined, for each of the studied approaches, as :
  - AD case : $G_e = 0.40$, $G_{\Delta e} = 3$, $G_{\Delta u} = 0.010$
  - RD case : $G_e = 0.29$, $G_{\Delta e} = 1$, $G_{\Delta u} = 0.012$.
- The adopted inference method is based on the Mamdani's implication mechanism, and is also called SUPremum-MINimum composition principle (Pedrycz, 1993).
- To obtain crisp values of the inferred fuzzy control actions, we have selected the well known Centre-Of-Gravity (COG) defuzzification technique which is the most commonly employed.

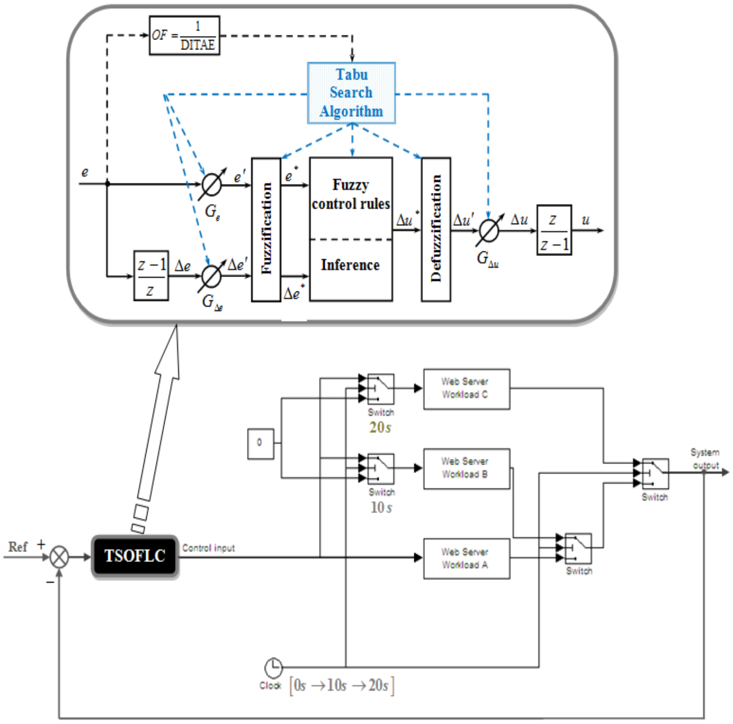

Fig. 4. TSOFLC-based feedback control system architecture.

5.2 *Controlling the Web server by the TSOFLC*

The TSOFLC-based feedback control system architecture is shown in Fig. 4.

5.2.1 *Tabu search algorithm*

Tabu search is a powerful stochastic local search algorithm belonging to the class of intelligent optimization techniques (Pham and Karaboga, 2012).

Introduced by Glover (Glover, 1989; Glover, 1990) in the late eighties, the TS search procedure is based on neighbourhood mechanism. It is a metaheuristic that guides a local heuristic search procedure to explore the solution space of a problem beyond local optimality. Its main specificity is its deterministic approach that can escape local optima by using a list of prohibited solutions known as *tabu list*. To generate the neighbourhood of any given solution, the local search procedure uses an operation (a mathematical function) called *move*. The information about the past steps of the search is kept in the tabu list. In every iteration of TS, the best solution is selected for the next iteration. In order to obtain the tabu list, TSA uses three basic elements: recency memory, frequency memory and aspiration criteria. The recency-based memory prevents cycles of length less than or equal to a predetermined number of iterations. Using the frequency based memory, the number of change of the solution vector elements is adjusted. If all solution vector elements are classified as tabu, then a least tabu solution element is removed from the tabu list (aspiration criteria). The flowchart of the TSA procedure is shown in Fig. 5.

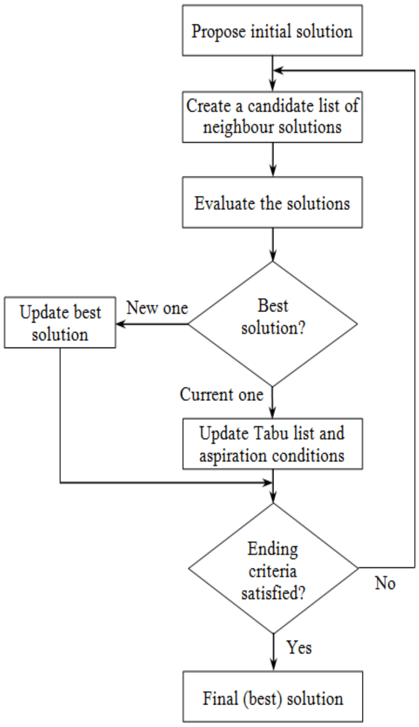

Fig. 5. Flowchart of the TSA procedure.

5.2.2 Derivation of the TSOFLC

The TS optimization algorithm is used to automatically adjust the following FLC parameters:

- Number of MFs for each FLC variable
- MFs shapes for each FLC variable
- MFs distribution for each FLC variable
- Decision table rules
- Scaling factors values.

Certain assumptions and constraints about the decision table and the FLC variables MFs to be optimized are given here:

- The number of FSs for each variable can take only one of the following possible values: 3, 5, 7 or 9.
- The FSs will be symbolized (labelled) by the standard linguistic designation and indexed in ascending order. If, for example, the number of FSs of a linguistic variable is equal to 5, the corresponding FSs will be NB, NM, ZE, PM and PB, and indexed from 1 to 5. The FSs NB and NM are considered as the opposites to PB and PM respectively (symmetrically with respect to ZE).
- All the FLC variables universes of discourses are normalized to lie between -1 and +1.
- The first and the last MFs have their apexes at -1 and +1, respectively.

The TSA optimization process starts with a first FLC "FLC_0" as an initial solution and begins the iterative evaluation of the generated new solutions by an objective function denoted by OF .

Chosen to maximize the inverse of the best-known and most-adopted performance index: Integral of Time-weighted Absolute Error (ITAE) (Graham and Lathrop, 1953), abbreviated here by DITAE for its discrete form, OF is written as:

$$OF = \frac{1}{DITAE} = \frac{1}{\displaystyle\sum_{m=m_0}^{m=m_f} \left\{ mT_s * \underbrace{\left|Ref\left(mT_s\right) - C_k\left(mT_s\right)\right|}_{\left|e\left(mT_s\right)\right|} \right\}} \quad (5)$$

where m_0 and m_f are the initial and final discrete times of the evaluating period and T_s is the sampling period.

Decision Table Deriving Method

The method on which we have based the decision rules table construction has been proposed in (Loudini, 2007; Loudini, 2013). The main details are given in the following.

First, a kind of grid (see Fig. 6) is constructed using two spacing parameters PSG_e and $PSG_{\Delta e}$ relative to the FLC two inputs e and Δe . PSG_e (resp. $PSG_{\Delta e}$) fixes the grid node X-axis coordinates (resp. Y-axis coordinates) in the interval [-1, +1] (normalized universe of discourse) with a simple computing formula given in the next paragraph. Each abscissa (resp. ordinate) represents a FS of the variable e (resp. Δe). The number of the grid constitutive nodes is then equal to the product result between the two FLC input FSs numbers.

Once the nodes are fixed, we introduce the output points on a straight line corresponding to the FLC output variable Δu . Now, the points (output) represent the FSs and not their coordinates. The number of points is equal to the output variable FSs number.

A third spacing parameter $PSG_{\Delta u}$ fixes the output point X -axis (Y -axis) coordinates similarly, whereas the Y -axis (X -axis) coordinates are calculated by an angular parameter, denoted by $Angle$. This determines the slope of the straight line supporting the output points with respect to the horizontal. This angular parameter varies in the interval $[0, \pi/2]$ anticlockwise.

Each of the grid nodes represents a case of the decision table and each output point represents an FS of the control variable Δu .

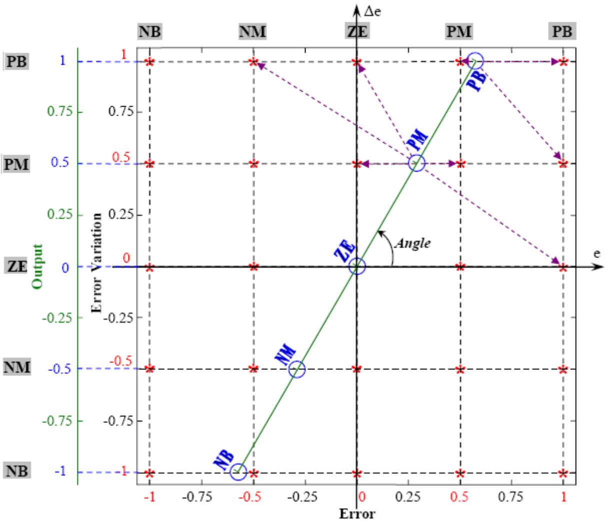

Fig. 6. Grid construction.

Once the coordinates of all the grid points (red stars) are computed, we can proceed to the assignment by determining the minimal distance among all the distances separating each node of the grid from all the output points (blue circles) situated on the straight line. Then, we assign to each node of the grid the closest output point. Consequently, the decision table case corresponding to this node will contain the FS representing the selected output point. Nevertheless, an assignment conflict could arise in the case of equality between two minimal distances separating a node and two output points. We propose to select the output point which has the lower FS index in the case of the upper part with respect to the table diagonal, or the output point which has the greater FS index in the case of the lower part. It should be noted that no more than two output points can be at the same distance from a given node of the grid, since all the output points are on the same straisght line.

The grid spacing parameter PSG specifies how the positions C_i of the intermediate points (between the centre and the extreme of each graduated axis) are spaced with respect to the central point. It offers flexibility to vary spacing. The more it is greater than 1, the more the point positions are close to the centre and vice versa. At the value 1, the positions are uniformly distributed in the universe of discourse interval [-1, 1].

The number of positions C_i being obviously the same as the number of FSs, we propose a formulation of the spacing law according to the spacing parameter PSG .

As a first stage, the positions C_i , being equidistant, are denoted by CEq_i and computed by:

$$CEq_i = 2\left(\frac{i-1}{NEF-1}\right)-1\,;\; i = 1,\cdots NFS\,. \tag{6}$$

The C_i values are then determined in terms of the spacing parameter PSG as follows:

$$C_i = sign(CEq_i) * \left|CEq_i\right|^{PSG} \tag{7}$$

with $sign(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$; $PSG = \left(PSG_1\right)^{PSG_2}$ with PSG_2 that can take the values +1 or -1.

Detailed illustrative examples are given in (Loudini, 2013).

MFs Deriving Method

Determination of the FLC MFs using the TSA takes place in three phases:

1. Creation of primary MFs of the FLC input/output variables $\left(e, \Delta e / \Delta u\right)$
2. Parameterization
3. Adjustment of the MFs.

Three types of MF shapes are considered:

- Triangular
- Trapezoidal, which generalize the triangular type
- "Two-sided" Gaussian with flattened summit.

The triangular shape is defined by three parameters, $\left[P1 \quad P2 \quad P3\right]$, which represent, respectively, the left abscissa of the triangle base, the peak abscissa, and the right abscissa of the triangle base. Each triangle base begins at the preceding triangle peak abscissa and ends at that of the following one.

The trapezoidal shape is defined by four parameters, $\left[P1 \quad P2 \quad P3 \quad P4\right]$, representing, respectively, the base left abscissa, the summit left abscissa, the summit right abscissa, and the base right abscissa. It is then framed by four points with the coordinates $\left(P1,0\right)$, $\left(P2,1\right)$, $\left(P3,1\right)$ and $\left(P4,0\right)$ (see Fig. 7). Note that if $P2 = P3$, we obtain a triangular shape.

We also define the two-sided Gaussian shape (Fig. 8) by four parameters, $\left[Sig1 \quad G1 \quad G2 \quad Sig2\right]$.

The left and right sides of the Gaussian are respectively defined by $G(x) = e^{-\frac{(x-G1)^2}{2(Sig1)^2}}$ and $G(x) = e^{-\frac{(x-G2)^2}{2(Sig2)^2}}$. To be able to use it within the framework of our optimizing method, it must be bounded by the same points used for the trapezoidal shape. In other words, we must define the two-sided Gaussian shape in terms of the parameters $\left[P1 \quad P2 \quad P3 \quad P4\right]$ instead of $\left[Sig1 \quad G1 \quad G2 \quad Sig2\right]$. For that purpose, we adopted a very small positive real number ε ($\varepsilon = 0.01$ was quite suitable) such that:

- The Gaussian left curve includes the points $\left(P1,\varepsilon\right)$ and $\left(P2,1\right)$
- The Gaussian right curve includes the points $\left(P3,1\right)$ and $\left(P4,\varepsilon\right)$.

This formulation leads to the following two systems of equations:

$$\begin{cases} e^{-\frac{(P1-G1)^2}{2*(Sig1)^2}} = \varepsilon \\ e^{-\frac{(P2-G1)^2}{2*(Sig1)^2}} = 1 \end{cases} ; \quad \begin{cases} e^{-\frac{(P3-G2)^2}{2*(Sig2)^2}} = 1 \\ e^{-\frac{(P4-G2)^2}{2*(Sig2)^2}} = \varepsilon \end{cases} . \tag{8}$$

The resolution of (8) gives:

$G1 = \mathrm{P2}$; $G2 = \mathrm{P3}$;

$$Sig1 = \sqrt{-\frac{(\mathrm{P1}-\mathrm{P2})^2}{2*\log\varepsilon}}\;;\; Sig2 = \sqrt{-\frac{(\mathrm{P4}-\mathrm{P3})^2}{2*\log\varepsilon}}.$$

Note that ε has been used, since the two sides of the Gaussian never pass a null abscissa.

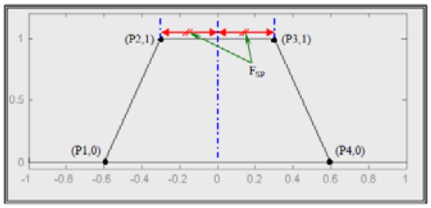

Fig. 7. Trapezoidal MF.

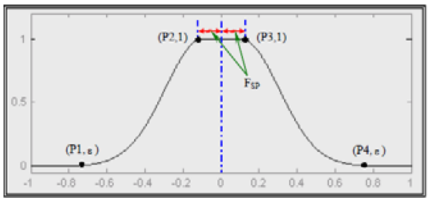

Fig. 8. Two-sided Gaussian MF.

MFs shape optimization

Inspired by the works of (Park *et al.,* 1995; Cheong and Lai, 2000; Foran, 2002), we have proposed in (Loudini, 2007) a new technique for the MFs shape optimization which we present in the following.

It is based on a design parameter called shape parameter (SP). This optimizing parameter gives possibilities of diversification (hybridization) of MF shapes on the universe of discourse of each of the FLC input/output variables.

SP is considered as a real number belonging to the interval [0, 2]. Its integer part, denoted by I_{SP} , will determine the shape of the MF, and its fractional part, denoted by F_{SP} , will determine the spacing with respect to the centre of the MF.

The MF shape is specified by I_{SP} and F_{SP} as follows:

- $I_{SP}=0$: trapezoidal or triangular shape
- $I_{SP}=1$: two-sided Gaussian shape.
- F_{SP} determines the symmetric space with respect to the centre of the MF as shown in Fig. 7 and Fig. 8. As we can see in Fig. 7, if the spacing is equal to zero the trapezoidal shape reduces to a triangular one.

Being optimized by the TSA, the number of MFs, denoted by NFS , for each of the FLC input/output variables, is not constant. Consequently, it is not feasible to allocate a spacing parameter to each MF. That is why we have proposed a solution, which consists in allocating a shaping parameter, denoted by SP_M , for the MF of the middle of the universe of discourse, and another, denoted by SP_E , for the extreme MF. The intermediate MF shaping parameters, denoted by SP_I , are then deducted from SP_M and SP_E so that they will have equidistant intermediate values.

The i^{th} shape parameter $SP_I(i)$, corresponding to the i^{th} intermediate MF, is determined by:

$$SP_I(i)=SP_M+2(i-1)\frac{SP_E-SP_M}{NFS-1};\ i=1,\ldots,\frac{NFS+1}{2}. \qquad (9)$$

We can observe that $SP_I(1)=SP_M$ and $SP_I\left(\frac{NFS+1}{2}\right)=SP_E$. So, two parameters are enough for any number of FSs.

The previous MF shaping parameters are allocated to the FLC input/output variables as follows:

- e → $SP_M e$, $SP_M \Delta e$ and $SP_M \Delta u$
- Δe → $SP_E e$, $SP_E \Delta e$ and $SP_E \Delta u$
- Δu → $SP_I e$, $SP_I \Delta e$ and $SP_I \Delta u$.

Note that if the medium and extreme MF shaping parameters are equal, all the universe of discourse MFs will have the same shape generated by the parameters. It is also important to prevent important overlapping between the generated MFs, which is undesirable in fuzzy control (flattening phenomenon) (Bühler, 1994). For this purpose, we have fixed a maximum value to the space F_{SP} equal to the half of the minimal distance between the two nearby summits.

MFs width spacing optimization

The summit abscissae of the different shapes are calculated by the same principle of parameter spacing used in the determination of the grid nodes and point coordinates for the decision table derivation. The FLC input/output variable MF spacing parameters are, respectively, denoted by PSF_e , $PSF_{\Delta e}$ and $PSF_{\Delta u}$.

During the search process, the TSA looks for the optimal setting of the FLC controller parameters which minimize the cost function OF . Solutions with low DITAE are considered as the fittest.

The TSA parameters chosen for the tuning purpose are given in Table 4 and the adopted parameter encoding is shown in Table 5.

Table 4. TSA parameters.

TS property	Method/value
Neighborhood generation method	swap of two elements
Neighbor list size	70
Maximum number of iterations	10

Table 5. Parameters adopted for encoding.

Parameter	Interval	Precision	Number of encoding bits
NFS	[3 , 9]	2	2
PSG_1	[0.1 , 1]	0.01	7
PSG_2	[-1 , 1]	2	1
$Angle$	[0 , $\pi/2$]	$\pi/512$	9
PSF_1	[0.1 , 1]	0.01	7
PSF_2	[-1 , 1]	2	1
SP	[0 , 1.99]	0.01	8
G_e ,$G_{\Delta e}$	]0 , 50]	0.01	13
$G_{\Delta u}$	]0 , 50]	0.1	9

Parameters of the Derived TSOFLC

After the TSA based optimization process, the main characteristics have been fixed for the AD and RD cases. The resulting scaling factors, MFs and decision tables are given in Table 6, Fig. 9, Table 7 and Table 8 respectively.

Table 6. TSOFLC scaling factors.

	G_e	$G_{\Delta e}$	$G_{\Delta u}$
AD control	0.5906	0.1811	-0.733
RD control	0,3968	0,3968	0,4285

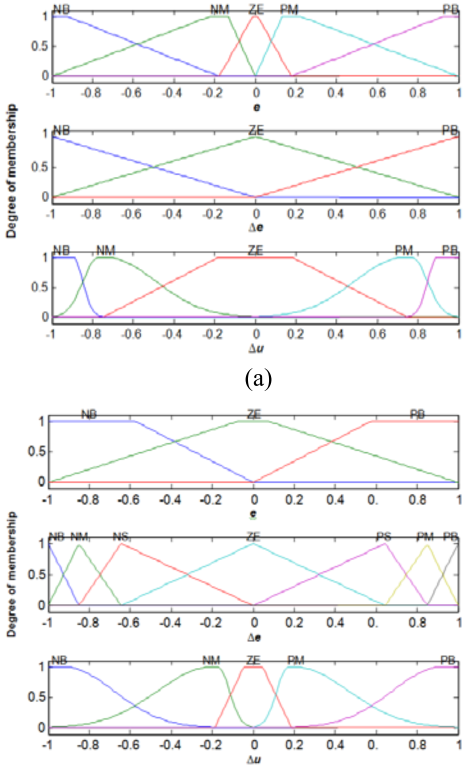

Fig. 9. MFs of the TSOFLC: (a) AD case; (b) RD case.

Table 7. Decision table of the TSOFLC (AD case).

Δu		e				
		NB	NM	ZE	PM	PB
Δe	NB	NB	NB	NB	NM	NM
	NM	NB	NB	NM	ZE	PM
	ZE	NB	NM	ZE	PM	PB

Table 8. Decision table of the TSOFLC (RD case)

Δu		e		
		NB	ZE	PB
Δe	NB	NB	NM	PM
	NM	NB	NM	PM
	NS	NB	ZE	PM
	ZE	NB	ZE	PB
	PS	NM	ZE	PB
	PM	NM	PM	PB
	PB	NM	PM	PB

6. SIMULATION RESULTS

In order to test the proposed intelligent control schemes, digital simulations have been carried out based on the process adopted dynamic mathematical models (discrete-time transfer functions) given in Table 1.

Using Matlab-Simulink software, the simulation studies have been conducted according to the feedback control architectures shown in Fig. 2 for the Mamdani PI type FLC application and in Fig. 4 for the TSOFLC, for different web server workloads (A, B, C). In fact, two abrupt workload variations have been considered at the instants 10 s and 20 s. These sudden variations allow better effectiveness and robustness evaluation of the adopted closed-loop control strategies.

6.1 Obtained Performances with the Mamdani PI type FLC

The obtained system responses and the corresponding control inputs generated by the Mamdani PI type FLC are shown, respectively, in Fig. 10 and Fig. 11 for the AD and RD cases. As we can see, the first FLC succeed to make the system output converge to the desired delay in an acceptable delay and maintain it at the vicinity of the reference before and after the two changes of workload. However, at these instants, inevitable but minor overshoots and undershoots occur due to the workload burst variations. These oscillations around the desired steady state value are more pronounced in the RD case. Nevertheless, the FLC shows rather good robustness in the face of these situations.

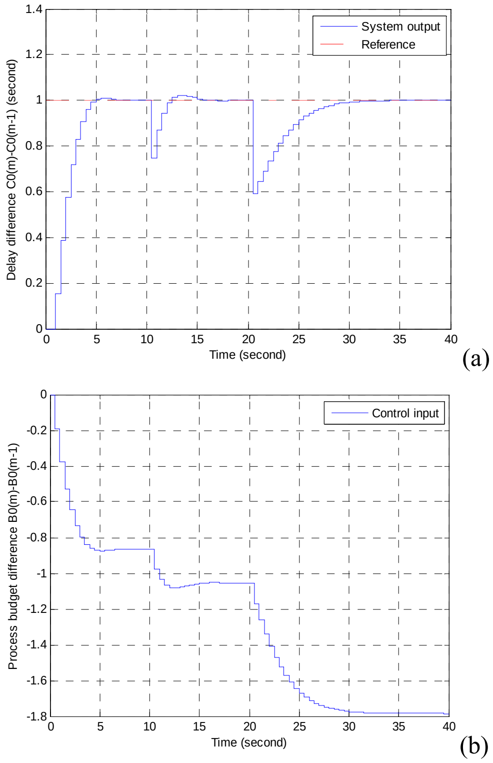

Fig. 10. FLC-based WS response for the AD case: (a) Response; (b) Control input.

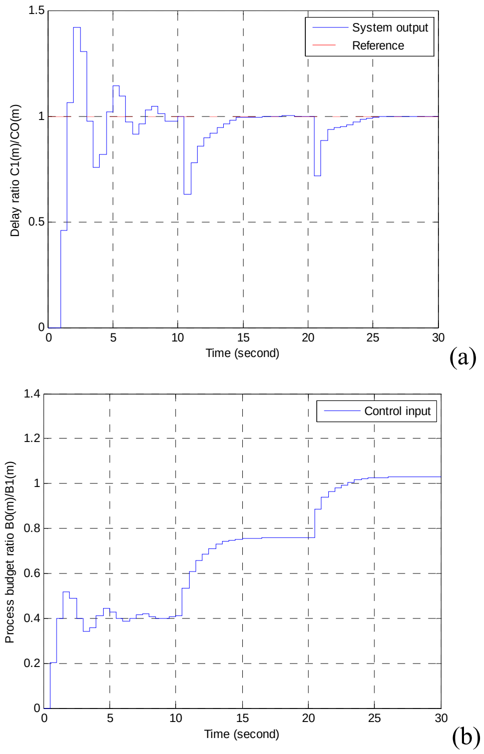

Fig. 11. FLC-based WS response for the RD case:
(a) Response; (b) Control input.

6.2 *Obtained Performances with the TSOFLC*

The digital simulation results, illustrating the performances of the implemented TSOFLC, are shown in Fig. 12 (AD case) and Fig. 13 (RD case).

As can be seen from these figures, the TSOFLC exhibits rather better step response performance in terms of rise time, overshoot magnitude, oscillations around the reference (desired delay difference (ratio)) and response (settling) time. We can also see that the TSOFLC shows an improvement in terms of robustness when faced to the simulated sudden workload variations, particularly for the RD case.

Under the TSOFLC strategy, the closed-loop controlled WS enforces, succesfully, the absolute (relative) delay guarantee by satisfying the required delay difference (delay ratio) for the high priority classes (class 0 and class 1) with an obvious superiority than the standard Mamdani PI type FLC.

7. RELATED RESEARCH

Establishing more and more complete and accurate dynamic models on one hand and synthesizing efficient control schemes on the other hand for the performance enhancement of software processes, particularly the WSs, is becoming an attractive research area. Even though several works have extensively investigated different QoS enhancing mechanisms supporting service differentiation, few research works addressing the application of feedback control methodologies are available.

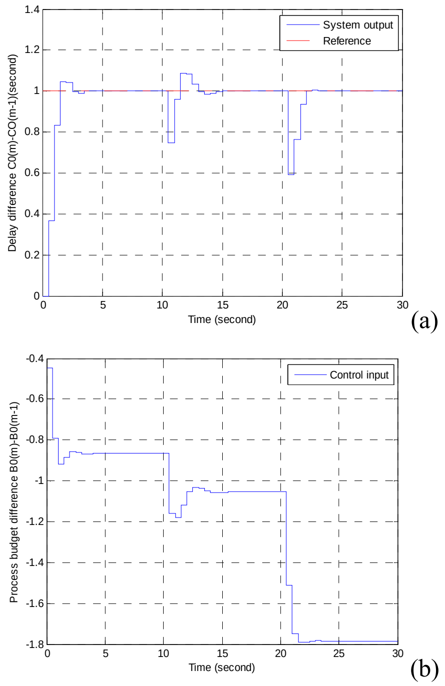

Fig. 12. TSOFLC-based WS response for the AD case:
(a) Response; (b) Control input.

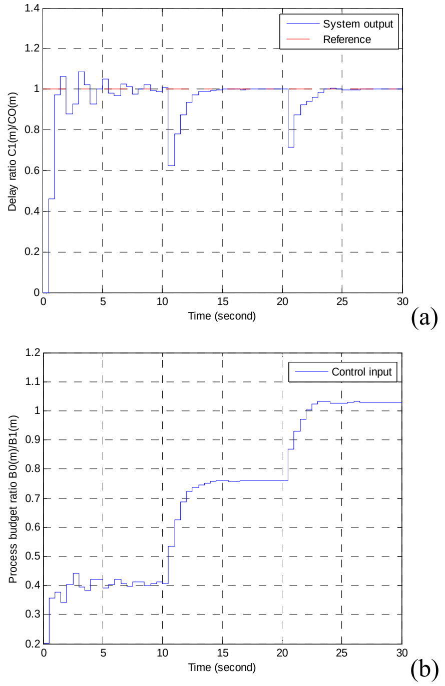

Fig. 13. TSOFLC-based WS response for the RD case:
(a) Response; (b) Control input.

Our description on literature review of related research works starts by mentioning significant ones that have used service delay differentiation approaches as mechanisms of QoS enhancement. We have found very interesting the investigations in (Leung *et al.*, 2001; Lee *et al.*, 2004; Rashid *et al.*, 2005; Wei *et al.*, 2006; Bourasa and Sevasti, 2007; Wu *et al.,* 2008; Garcia *et al.,* 2009; Dimitriou and Tsaoussidis, 2010; Gao *et al.,* 2011; Varela *et al.,* 2012).

The closest works to our investigation being those using feedback control techniques to improve WSs QoS, we briefly present some relevant ones in a chronological order.

In (Andersson *et al.,* 2003), a combination of queuing theory and control theory has been adopted. The Apache WS has been modeled as a GI/G/1-system. Then, a standard PI-controller was employed as an admission control mechanism.

In (Henriksson *et al.,* 2004), a contribution is presented as an extension of the classical combined feedforward/feedback control framework where the queuing theory is used for feedforward delay prediction. They replace the queuing model with a predictor that uses instantaneous measurements to predict future delays. The proposed strategy was evaluated in simulation and by experiments on an Apache WS.

In (Oottamakorn, 2005), the author proposed a resource management and scheduling algorithm to provide relative delays differentiated guarantees to classes of incoming requests at a QoS-aware WS. One of the key results of his work is the development of an efficient procedure for capturing the predictive traffic characteristics and performances by monitoring ongoing traffic arrivals. This allows the WS's resource management by determining sufficient server resource for each traffic class in order to meet its delay requirements. In order to achieve a self-stabilizing performance in delay QoS guarantees, he has implemented an adaptive feedback control mechanism.

In (Zhou *et al.,* 2006), the problem of providing proportional QoS differentiation with respect to response time on WSs has been investigated. They first present a processing rate allocation scheme based on the foundations of queueing theory. They designed and implemented an adaptive process allocation approach, guided by the queueing-theoretical rate allocation scheme, on an Apache server. They established that this application-level implementation shows weak QoS predictability because it does not have fine-grained control over the consumption of resources that the kernel consumes and hence the processing rate is not strictly proportional to the number of processes allocated. They then designed a feedback controller and integrated it with the queueing-theoretical approach. The adopted feedback control strategy adjusts process allocations according to the difference between the target response time and the achieved response time using a PID controller.

In (Qin and Wang, 2007), the authors applied a control-theoretic approach to the performance management of Internet WSs to meet service-level agreements. In particular, a CPU frequency management problem has been studied to provide response time guarantees with minimal energy cost. It was argued that linear time-invariant modeling and control may not be sufficient for the system to adapt to dynamically varying load conditions. Instead, they adopted a Linear-parameter-varying (LPV) approach.

In (Kihl *et al.,* 2008), the authors presented how admission control mechanisms can be designed with a combination of queuing theory and control theory. They modeled an Apache WS as a GI/G/1-system and validated their model as an accurate representation of the experimental system, in terms of average server utilization. Using simulations for discrete-event systems based on queuing theory and with experiments on an Apache WS, they compared a PI controller and an RST-controller, both commonly used in automatic control, with a static controller and a step controller, both commonly used in telecommunication systems. Note that the controllers were implemented as modules inside the Apache source code. They have also performed a nonlinear stability analysis for the PI-controlled system.

In (Yansu *et al.,* 2009), a self-tuning control framework to provide proportional delay differentiation guarantees on WS has been proposed. The approach updates the model and controller parameters based on the variations of object model to reduce system error and optimize the performances through an online identification.

In (Lu *et al.,* 2011), the authors considered providing two types of QoS guarantees, proportional delay differentiation and absolute delay guarantee, in the database connection pool in Tomcat WS application servers using the classical feedback control theory. To achieve these goals, they established approximate linear time-invariant models through system identification experimentally, and designed two PI controllers using the root locus method. These controllers are invoked periodically to calculate and adjust the probabilities for different classes of requests to use a limited number of database connections, according to the error between the measured QoS metric and the reference value.

In a recent work (Patikirikorala *et al.,* 2012), a new approach for QoS performance management and resource provisioning using an off-line identification of Hammerstein and Wiener nonlinear block structural model has been proposed. Using the characteristic structure of the nonlinear model, a predictive feedback controller based on a gain schedule technique is incorporated in the design to achieve the performance objectives.

Examples of earlier research investigations using fuzzy logic based feedback control can be found in (Diao *et al.,* 2002; Wei *et al.,* 2005; Wei *et al.,* 2006; Chan and Chu, 2007; Wei *et al.,* 2007; Tian *et al.,* 2010; Rao *et al.,* 2011).

8. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions In this article, we have presented in detail the design study and the application of some intelligent control methodologies for the QoS enhancement of a web server in the case of absolute and relative delay service differentiations.

To enforce desired absolute or relative connection delays via dynamic process reallocation, we have developed two closed-loop control schemes based on the traditional Mamdani PI type FLC and then, a Tabu Search optimized FLC.

We have implement and evaluated the proposed feedback control architectures on a simulated Web server using validated dynamic discrete-time mathematical models for different scenarios of workload variation.

The obtained simulation results demonstrate that our intelligent control-based strategies provide satisfactory robust delay guarantees even when the workload fluctuates abruptly and significantly.

We have also revealed some innovative procedures in tuning the FLC by the Tabu search optimization technique.

Further investigations to improve the obtained performances by other feedback control schemes as well as the optimization by other techniques will be conducted as well.

Finally, we propose, as an open problem, to investigate the mathematical modeling of the Web server dynamic process as a time-variable parameters system.

REFERENCES

Abdelzaher, T.F., Stankovic, J.A., Lu, C., Zhang, R. and Lu, Y. (2003). Feedback Performance Control in Software Services. *IEEE Control Systems Magazine*, 23(3), 74-90.

Abdelzaher, T.F., Diao, Y., Hellerstein, J.L., Lu, C. and Zhu, X. (2008). Performance Modeling and Engineering. 7: *Introduction to Control Theory and its Application to Computing Systems*, Springer, New York, 185-215.

Andersson, M., Kihl, M. and Robertsson, A. (2003). Modelling and Design of Admission Control Mechanisms for Web Servers Using Non-Linear Control Theory. In *Proc. of the ITCom's Conf. on Performance and Control of Next-Generation Communication Networks*, Orlando, FL, USA, 53-64.

Andersson, M. (2005). Introduction to Web Server Modeling and Control Research. Technical Report, Department of Communication Systems, Lund Institute of Technology.

Barford, P. and Crovella, M.E. (1998). Generating Representative Web Workloads for Network and Server Performance Evaluation. In *Proc. of the ACM SIGMETRICS Joint Int. Conf. on Measur. and Modeling of Computing Syst.*, Madison, WI, USA, 151-160.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998). An Architecture for Differentiated Services, *IETF RFC* 2475.

Bourasa, C. and Sevasti, A. (2007). An Analytical QoS Service Model for Delay-Based Differentiation. *Computer Networks*, 51(12), 3549-3563.

Braden, R., Clark, D. and Shenker, S. (1994). Integrated Services in the Internet Architecture: An Overview. *IETF RFC* 1633.

Bühler, H. (1994). *Réglage par Logique Floue*, Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland.

Chan, K.H. and Chu, X. (2007). Design of a Fuzzy PI Controller to Guarantee Proportional Delay Differentiation on Web Servers. In *Proc. of the 3rd Int. Conf. on Algorithmic Aspects in Information and Management*, Portland, OR, USA, 389–398.

Cheong, F. and Lai, R. (2000). Constraining the Optimization of a Fuzzy Logic Controller Using an Enhanced Genetic Algorithm. *IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics*, 30(1), 31-46.

Diao, Y., Hellerstein, J.L. and Parekh, S. (2002). Optimizing Quality of Service Using Fuzzy Control. In *Proc. of the 13th IFIP/IEEE Int. Workshop on Distributed Systems: Operations and Management*, In: *LNCS*, 2506, Springer-Verlag, Heidelberg, Berlin, 42-53.

Dimitriou, S. and Tsaoussidis, V. (2010). Promoting Effective Service Differentiation with Size-Oriented Queue Managemen. *Computer Networks*, 54(18), 3360-3372.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. (1999). Hypertext Transfer Protocol-HTTP/1.1. *IETF RFC* 2616.

Foran, J. (2002). *Optimisation of a Fuzzy Logic Controller Using Genetic Algorithms*, Master Project Report, School of Electronic Eng., Dublin City University.

Gao, A., Mu, D. and Hu, Y. (2011). A QoS Control Approach in Differentiated Web Cashing Service. *Journal of Networks*, 6(1), 62-70.

Garcia, D.F., Garcia, J., Entrialgo, J., Garcia, M., Valledor, P., Garcia, R. and Campos, A.M. (2009). A QoS Control Mechanism to Provide Service Differentiation and Overload Protection to Internet Scalable Servers. *IEEE Trans. on Services Computing*, 2(1), 3-16.

Glover, F. (1989). Tabu Search-part I. *ORSA Journal on Computing*, 3(1), 190–206.

Glover, F. (1990). Tabu Search-part II. *ORSA Journal on Computing*, 1(2), 4–32.

Gourley, D. and Totty, B. (2002). *HTTP: The Definitive Guide*. O'Reilly Media, Sebastopol, CA, USA.

Graham, D. and Lathrop, R. C. (1953). The Synthesis of Optimum Transient Response: Criteria and Standard Forms. *Trans. of the American Institute of Electrical Engineers*, *2*: *Applications and Industry*, 72, 273–288.

Hellerstein, J.L., Diao, Y., Parekh, S. and Tilbury, D.M. (2004). *Feedback Control of Computing Systems*. IEEE Press-Wiley, Hoboken, NJ, USA.

Henriksson D., Lu, Y. and Abdelzaher, T. (2004). Improved Prediction for Web Server Delay Control. In *Proc. of the 16th Euromicro Conf. on Real-Time Syst.*, Catania, Sicily, Italy, 61-68.

Kihl, M., Robertsson, A., Andersson, M. and Wittenmark, B. (2008). Control-Theoretic Analysis of Admission Control Mechanisms for Web Server Systems. *World Wide Web*, 11(1), 193-116.

Kilkki, K. (1999). *Differentiated Services for the Internet*. Macmillan Technical Pulishing, Indianapolis, IN, USA.

Kozierok, C.M. (2005). *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. No Starch Press, San Fransisco, CA, USA.

Lee, C.C. (1990) Fuzzy Logic in Control Systems: Fuzzy Logic Controller- part I & part II. *IEEE Trans. on Systems, Man and Cybernetics*, 20(2), 404-435.

Lee, S.C.M., Lui, J.C.S. and Yau, D.K.Y. (2004). A Proportional-Delay Diffserv-Enabled Web Server:

Admission Control and Dynamic Adaptation. *IEEE Trans. on Parallel and Distributed Syst.*, 15(5), 385-400.

Leung, M.K.H., Lui, J.C.S. and Yau, D.K.Y. (2001). Adaptive Proportional Delay Differentiated Services: Characterization and Performance Evaluation. *IEEE/ACM Trans. on Networking*, 9(6), 80-817.

Ljung, L. (1999). *System Identification - Theory for the User*. PTR Prentice Hall, Upper Saddle River, N.J., USA.

Loudini, M. (2007). *Contribution à la Modélisation et à la Commande Intelligente d'un Bras de Robot Manipulateur Flexible*. Ph.D. thesis, Ecole Nationale Polytechnique, Algiers, Algeria.

Loudini, M. (2013). Modeling and Intelligent Control of an Elastic Link Robot manipulator. To appear in *Int. Journal of Advanced Robotic Systems*, 10, 1-18.

Lu, C., Abdelzaher, T.F., Stancovic, J.A. and Son, S.H. (2001). A Feedback Control Approach for Guaranteeing Relative Delays in Web Servers. In *Proc. of the 7th IEEE Real-Time Technology and Applications Symposium*, Taipei, Taiwan, 51-62.

Lu, C., Lu, Y., Abdelzaher, T.F., Stancovic, J.A. and Son, S.H. (2006). Feedback Control Architecture and Design Methodology for Service Delay Guarantees in Web Servers. *IEEE Trans. on Parallel and Distributed Systems*, 17(9), 1014-1027.

Lu, J., Dai, G., Mu, D., Yu, J. and Li, H. (2011). QoS Guarantee in Tomcat Web Server: A Feedback Control Approach. In *Proc. of the 2011 Int. Conf. on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Beijing, China, 183-189.

Mac Vicar-Whelan, P.J. (1976). Fuzzy Sets for Man Machine Interactions. *Int. J. of Man Mach. Studies*, 8(6), 687-697.

Mamdani, E.H. (1974). Applications of Fuzzy Algorithms for Control of a Simple Dynamic Plant. *Proceedings of the IEE*, 121(12), 1585-1588.

Netcraft (2013). Web server survey. Available at: http://news.netcraft.com/archives/2013.

Oottamakorn, C. (2005). Class-Based Guarantees of Relative Delay Services in Web Servers. In *Proc. of the IASTED Int. Conf. on Parallel and Distributed Computing and Networks*, Innsbruck, Austria.

Parekh, S. (2010). *Feedback Control Techniques for Performance Management*. Ph.D Dissertation, University of Washington, Seattle, WA, USA.

Park, Y.J., Cho, H.S. and Cha, D.H. (1995). Genetic Algorithm-Based Optimization of Fuzzy Logic Controller Using Characteristic Parameters. In *Proc. of the IEEE Int. Conf. on Evolutionary Computation*, Perth, WA, 831-836.

Patikirikorala, T., Wang, L., Colman, A. and Han, J. (2012). Hammerstein–Wiener Nonlinear Model Based Predictive Control for Relative Qos Performance and Resource Management of Software Systems. *Control Engineering Practice*, 20(1), 49-61.

Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems*. John Wiley & Sons Inc., New York, NY, USA.

Pham, D.T. and Karaboga, D. (2012). *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing, and Neural Networks*. Springer, London, UK.

Qin, W. and Wang, Q. (2007). Modeling and Control Design for Performance Management of Web Servers via an LPV Approach. *IEEE Trans. on Control Systems Technology*, 15(2), 259-275.

Rashid, M.M., Alfa, A.S., Hossain, E. and Maheswaran, M. (2005). An Analytical Approach to Providing Controllable Differentiated Quality of Service in Web Servers. *IEEE Trans. on Parallel and Distributed Syst.*, 16(11), 1022-1033.

Rao, J., Wei, Y., Gong, J. and Xu, C.-Z. (2011). DynaQoS: Model-Free Self-Tuning Fuzzy Control of Virtualized Resources for Qos Provisioning. In *Proc. of the 9th Int. Workshop on Quality of Service*, San Jose, CA, USA.

Tian, F., Xu, W. and Sun, J. (2010). Web QoS Control Using Fuzzy Adaptive PI Controller. In *Proc. of the 9th Int. Symp. on Distributed Computing and Applic. to Business Engineering and Science*, Hong Kong, China, 72-75.

Varela, A., Vazão, T. and Arroz, G. (2012). Providing Service Differentiation in Pure IP-Based Networks. *Computer Communications*, 35(1), 33-46.

Wang, Z. (2001). *Internet QoS. Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann, San Fransisco, CA, USA.

Wei, Y., Lin, C., Chu, X., Shan, Z. and Ren, F. (2005). Class-Based Latency Assurances for Web Servers. In *Proc. of the 1st Int. Conf. on High Performance Computing and Communications*, Sorrento, Italy, in: LNCS, 3726, Springer-Verlag, Heidelberg, Berlin, 388-394.

Wei, Y., Lin, C., Chu, X. and Voigt, T. (2006). Fuzzy Control for Guaranteeing Absolute Delays in Web Servers. *Int. J. of High Performance Computing and Networking*, 4(5-6), 338-346.

Wei, J., Xu, C.-Z., Zhou, X. and Li, Q. (2006). A Robust Packet Scheduling Algorithm for Proportional Delay Differentiation Services. *Computer Communications*, 29(18), 3679-3690.

Wei, J. and Xu, C.Z. (2007). Consistent Proportional Delay Differentiation: A Fuzzy Control Approach. *Computer Networks*, 51(5-6), 2015-2032.

Wu, C.-C., Wu, H.-M. and Lin, W. (2008). High-Performance Packet Scheduling to Provide Relative Delay Differentiation in Future High-Speed Networks. *Computer Communications*, 31(10), 1865-1876.

Yansu, H., Guanzhong, D., Ang, G. and Wenping, P. (2009). A Self-Tuning Control for Web Qos. In *Proc. of the Int. Conf. on Information Engineering and Computer Science*, Wuhan, China, 1-4.

Zadeh, L. A. (1988). Fuzzy Logic, *Computer*, 21(4), 83-93.

Zhou, X., Cai, Y. and Chow, E. (2006). An integrated Approach with Feedback Control for Robust Web QoS Design. *Computer Communications*, 29(16), 3158-3169.