The Design of a Fuzzy Logic Controller of an ATM Switch and its Simulation

FRANCE CHEONG, RICHARD LAI*, YONG SOO KIM+,
DAE HOON HWANG+, HAE SUK OH+ AND JI HONG KIM+
School of Business Information Technology
RMIT University
Victoria 3000, Australia
**Department of Computer Science and Computer Engineering*
La Trobe University
Victoria, 3086, Australia
+College of Software
Kyungwon University
Songnam, Kyunggi-Do, 461-701, Korea

Available Bit Rate (ABR) streams are carried whenever bandwidth is available to do so. When transporting MPEG streams over ATM networks, the link utilization cannot be brought close to 100% no matter how efficient the connection control admission scheme is, but can be improved by ABR traffic. Due to the unpredictable fluctuations and burstiness of the MPEG streams, it is difficult to accurately control the ABR flow rate. A traffic control mechanism based on fuzzy logic is therefore required to effectively transport ABR traffic so that network operators can use most of the unused bandwidth. In this paper, we present the design of an optimized Fuzzy Logic Controller (FLC) of an ATM switch for the rate control of an ABR stream. Unlike the previous work where synthetic data and trial and error were used for tuning an FLC, the merits of our method are that we used simulations and real MPEG data to generate the input-output data for the optimal design of the FLC, and that our technique enables the output of the FLC to be symmetrical.

Keywords: fuzzy logic controller, simulation, differential evolution, control system, ATM

1. INTRODUCTION

The ATM standards [1] specify four main types of ATM layer service classes: Constant Bit Rate (CBR), Variable Bit Rate (VBR), Available Bit Rate (ABR) and Unspecified Bit Rate (UBR). These different types of services have different Quality of Service (QoS) requirements and thus require different flow control methods. The requirements of CBR and VBR sources for fixed bandwidths are handled by the Connection Admission Control (CAC) mechanism which ensures that sufficient bandwidths are allocated at call set-up time. However, there are network and transport layer protocols that can adapt to a time-varying available bandwidth. This type of traffic is known as ABR stream which is carried whenever there is bandwidth available. In order to obtain the highest possible link utilization, all the excess bandwidth remaining after transporting VBR streams are made available for transporting ABR streams. When transporting MPEG [2] streams over an

Received May 12, 2005; revised September 5, 2005 & January 24, 2006; accepted March 1, 2006.
Communicated by H. Y. Mark Liao.

ATM network, the link utilization cannot be brought close to 100% no matter how efficient the CAC scheme is. In such a case, the utilization can be improved by ABR streams which consist mostly of data streams that are not sensitive to delay, but are highly sensitive to cell losses. Most data traffic can regulate their transfer rates as a function of bandwidth availability.

A traffic control mechanism is required to effectively transport ABR traffic so that network operators can make use of the unused bandwidth without affecting the performance of the CBR and VBR connections. The network provides feedback control information about the availability of bandwidth resources in Resource Management (RM) cells. The feedback information can be either binary or explicit. Binary information is used to request the source to increase/decrease the transmission of cells by pre-specified amounts, while explicit feedback information specifies the number of cells to be transmitted. Due to the unpredictable fluctuations and burstiness of VBR traffics, it is difficult to determine the explicit number of ABR cells that have to be transported at any point in time. This value is often determined by heuristic means such as in the case of the Enhanced Proportional Rate Control Algorithm (EPRCA) proposed by ATM Forum [3]. Due to the unpredictable fluctuations and burstiness of the MPEG streams, it is difficult to accurately control the ABR flow rate at all times. Thus, a control policy based on fuzzy logic is expected to perform better than EPRCA, provided the controller is adequately designed and tuned.

Fuzzy Logic Controllers (FLCs) were first introduced by Assilian and Mamdani [4]. FLCs are rule-based expert systems using human-like intelligence. They can easily capture the approximate and qualitative aspects of human knowledge and reasoning. The advantage of FLCs is that no mathematical modelling knowledge is required and this is useful for the control of complex and ill-defined systems. There are two main approaches to constructing an FLC (or any other type of fuzzy system) [5]: the *system identification approach* [6] and the *direct approach* [7]. Prior to the numerous attempts for producing optimized FLCs, the FLC design process used a combination of expert knowledge of the controlled system, heuristics and trial and error. This lack of systematic design procedure [8, 9] has long been recognized and several attempts have been made over the years for producing optimized FLCs.

Procyk and Mamdani [10] were among the first to propose a method for the automatic generation of FLCs. They realized that when complex processes with nonlinear behaviours and higher dimensions than single input-output are involved, a control policy (rule base) cannot always be developed whether from common sense or from an experienced operator. Wang and Mendel [11] developed a procedure for generating fuzzy rules by learning from examples. Although a mathematical model of the system is not required, sample input-output data are required and this is not always easy to obtain. Holve and Protzel [12] used Wang and Mendel's method [11] of training data for optimizing the rule base of an FLC which was used for acceleration control of an adaptive cruise control system. Lekova *et al*. [13] developed an FLC for traffic control. The control problem consists of finding the duration of the green light to minimize the waiting queues of vehicles.

In this paper, we present the design of an optimized Fuzzy Logic Controller (FLC) of an ATM switch for the rate control of an ABR stream; and the rate we control is the update rate. The inputs to the FLC are the error and its rate of change and the output of the FLC is the rate of change of the control variable. Differential Evolution (DE) [14] was employed to generate the best FLCs for a number of different situations. Control tables are then generated for the FLC and are merged into a single table for determining the optimized membership functions. Unlike the previous work where synthetic data and trial and error were used for tuning an FLC, the merits of our method are that we used simulations and real MPEG data to generate the input-output data for the optimal design of the FLC, and that our technique enables the output of the FLC to be symmetrical.

2. MPEG TRACE DATA

Contrary to many studies which use synthetic CBR and VBR streams for simulation purposes, we used real MPEG CBR audio streams and VBR video streams. We favoured the use of real MPEG trace data since it is well known that MPEG VBR video streams are hard to simulate because of their long range dependencies.

We collected MPEG-1 audio and video trace data (frame sizes) by processing five video CDs (*Independence Day*, *Cut Throat Island*, *Highlander*, *Maximum Risk* and *Mistrial*), but since this number is insufficient, we also used 14 trace data for a variety of MPEG-1 encoded movies found at the University of Würzburg in Germany. The MPEG trace files are: *asterix* (Cartoon), *atp* (ATP Tennis final 1994), *bond* (James Bond), *bowl* (Super Bowl), *dino* (Jurassic Park), *fuss*, *lambs* (The Silence of the Lambs), *mr.bean* (Mr Bean), *mtv* (Music Clips), *race* (Formula 1 race), *simpsons* (Cartoon), *soccer* (World Cup Soccer), *starwars* (Start Wars), and *terminator* (Terminator).

3. DESIGN METHOD

3.1 Control Philosophy

The problem under study is analogous to a fluid level control situation. The output buffer is considered as a tank holding the liquid (ATM cells) and the role of the controller is to maintain the liquid level close to a set point. Fluid level controllers use the error in liquid level and its rate of change as inputs. The error is defined as the set point minus the actual liquid level while the rate of change is measured as the change in error between successive time instants. The output of the controller can either be the control variable (ABR flow rate in our case) or the change in control variable.

Maintaining the output buffer level at the set point is a means to ensure that the maximum amount of ABR cells are transferred onto the ATM output link at every time instant. If the buffer level rises above the set point, this means that the ABR source has transmitted more cells than could be evacuated on the output link. This also implies that the output link bandwidth was fully utilized. If the buffer level rises too much, then there is the risk of ABR cells being lost by overflow. This is not a major problem since data applications keep track of transmitted cells and request re-transmission of any lost cells. However, in our context, output buffer overflow is undesirable since re-transmission of cells reduces bandwidth utilization.

If the buffer level falls below the set point, bandwidth is still fully utilized provided that the buffer is not completely empty. If the buffer is completely empty, then we cannot know whether the output link could have evacuated more ABR cells than were transmitted. Thus, the exact buffer level is not important so long as it does not overflow or underflow.

3.2 FLC Inputs and Output

Unlike the work of Pitsillides and Sekercioğlu [15], and Hu *et al*. [16], we do not use the buffer length and its rate of change as inputs to the FLC and the ABR flow rate as the output of the FLC. Instead, we use the error in buffer length and its rate of change as the inputs and the change in flow rate as the output. This allows us to use a standard MacVicar-Whelan rule base. In order to obtain the error in buffer length, we have to define a set point which is considered as the desired buffer level to be maintained at all times. Since the choice of this set point is not critical, we choose it to be the mid-point level of the buffer.

We do not use any other inputs such as the additional ones used by Hu *et al*.. We believe that this complicates the tuning of the FLC unnecessarily, and furthermore, it is simply not required. Maintaining the buffer level at the set point is all that is required to ensure the maximum utilization of bandwidth while minimizing cell losses. We also use more fuzzy sets per input/output variable. In particular, we use 7 fuzzy sets per input/output variable since we found out in previous studies that this number results in a smooth control table and is adequate for most control problems.

3.3 Type of FLC

We used a PD-like Mamdani FLC in which the inputs are the error in buffer level and its rate of change. The output of the FLC is the change in ABR flow rate and this has to be added to the previous value of the flow rate to obtain the new flow rate that is required to maintain the level at the set point.

The FLC for rate control of ABR is shown in Fig. 1. Two output buffers are shown in the figure, one for CBR/VBR streams and another one for ABR streams. The two buffers are sized to achieve different purposes. The CBR/VBR buffer is small and is sized to limit the maximum delay to 125 μs as required by audio and video applications. The ABR buffer is much bigger and serves to compensate for any inaccuracies in the determination of the correct ABR flow rate. The size of the CBR/VBR buffer is critical, but the size of the ABR buffer is not so long as it is reasonably big. At each sampling time instant, the error in the level of the ABR buffer is measured and the change in error is computed. These two values are input to the FLC which produces an output. Since this output represents the change in flow rate, it has to be added to the previous flow rate to obtain the new flow rate. The new flow rate is then inserted in an RM cell which travels from the ATM switch back to the ABR source. The data application transmitting the ABR traffic is expected to adapt its transmission rate to match the rate specified in the RM cell.

Modeling more than one ABR user can be done using the same principle that we have used for one user. However, modeling how many users is not the key issue, but the validity of the assumptions that have been made.

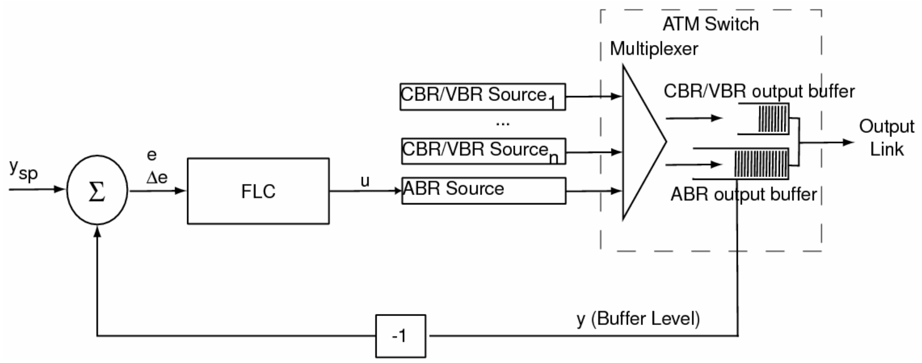


Fig. 1. Fuzzy logic control of ABR.

3.4 Tuning Philosophy

We use a new approach called the *control table approach* to design and tune the FLC. A control table contains output values of the FLC for the different combinations of the inputs over their whole range of possible values. A control surface is the surface obtained by plotting the values in the control table. If the FLC has two inputs, the control surface can be easily visualized as a three-dimensional plot. The greater the dimensions of the inputs, the harder it is to visualize the control surface.

In many studies, the control surface of the FLC is considered as the outcome of the design and tuning processes. The design process starts with the formulation of the rule base usually using expert knowledge or heuristics. This is followed by the formulation of the membership functions, either using some optimization technique or simply using trial and error. Finally, the outcome is the FLC's control surface. The process is iterated several times until satisfactory control is obtained.

We consider the control table/surface both as the input and the outcome of the tuning process. We select and use appropriate sample input-output data to define the desired control table as input to the tuning process. It might be argued that our approach is the same thing as using sample input-output data for tuning the FLC. It is not, because we do not use *just any* training data that we happen to obtain. Instead, we are *selective* in our use of training data. In particular, we divide the inputs into a range of values depending on the desired granularity of control. For each possible combination of the inputs, we seek the required output value. If we have several output values for the same combination of inputs, we reduce them to only one value by some means such as average, maximum, minimum, *etc*. The particular method use for reducing output several values depends on the application and the control philosophy.

The number of subdivisions used for dividing the input range depends on the desired accuracy of control and is thus application dependent. Unsatisfactory control will result if the subdivisions are too coarse. They should not be too fine as well because of difficulties of obtaining and handling large amounts of training data. We found a granularity of 0.1 to be adequate in the range [0.0, 1.0]. In many cases, it will not be possible to obtain data for the complete control table. In such cases, expert knowledge, heuristics, logging of events during simulation can be used to fill in the missing data.

DE was employed to tune the membership functions in order to achieve the desired control surface. Revision of the design may be required if the desired control surface is not reached. The whole process may be iterated several times until satisfactory control is achieved, but this should be rare if the sample data are representative of the required control system.

3.5 Building the Control Table

In order to build the control table for ABR control we randomly combine MPEG audio and video trace data into several aggregate streams. Each aggregate stream contains a random number of MPEG trace data files selected from an available list of 19 movies. Furthermore, in an attempt to simulate the random arrival of connection requests of MPEG movies that occur in real-life situations, we do not aggregate the first frames of all selected movies, then all the second frames, *etc*. Instead, we randomly select the starting aggregation points of all the movies, except for the first one which starts at the first frame. All the movies are of equal duration and when the end of a movie is reached while the aggregation is not over, aggregation of that movie continues from its first frame.

Each aggregated stream is then used to determine an optimized FLC for that particular stream using Differential Evolution (DE). Since we are using the error and the change in error as inputs to the FLC and the change in control variable as output, we can use a generic MacVicar-Whelan rule base [17] for the purpose. The relationship of the error and the change in error in formulating the change in output is well-defined by a MacVicar-Whelan rule base. We have proved the adequacy of this rule base in applications ranging from the cart-pole problem [18] to plant processes [19-21] and we shall prove it again for the ABR application.

Thus, DE is used to find the optimum membership functions for each FLC operating on a particular aggregation of MPEG streams. The objective function used to guide the search is the sum of squares of the error (SSE). Each trial FLC is used to control the ABR flow rate while the transportation of the aggregate stream is in progress. The error in the buffer level is measured at each sampling instant and the SSE is accumulated for the duration of the transportation of the aggregated stream. The trial FLC which minimizes the SSE is deemed to be the best one for that particular aggregated stream. We used a population of 50 individuals, $\alpha = 0.7$ (parameter used in calculating the perturbation vector), and $Cr = 0.7$ (crossover probability) as system parameters.

When optimizing the membership functions, we make use of the fact that the FLC for ABR control needs to be symmetrical (outputs of same magnitude but different directions for inputs of same magnitude and different directions). In order to achieve this, the parameters for the membership functions are generated only in the range [0.0, 1.0] and they are replicated to the range [− 1.0, 0.0]. The parameters at the extreme positions are fixed to − 1.0 and 1.0 respectively while the middle parameter is fixed to 0.0.

After having obtained several FLCs optimized for transporting the maximum amount of ABR traffic on random aggregations of CBR/VBR streams, we need to combine these FLCs into a single one. How do we do combine several FLCS into one? The answer is the control table. Each FLC is made to generate a control table at the required granularity level and the control tables are merged into a single one. There are many ways to combine several entries into single ones and this is problem dependent. In the ABR situation, we averaged the values since they did not show much variation. The resulting control table is shown in Fig. 2.

change in error \ error	-1.0	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9	-0.8	-0.7	-0.7	-0.6	-0.6	-0.6	-0.5	-0.3	0.0
-0.9	-1.0	-1.0	-0.9	-0.9	-0.9	-0.9	-0.9	-0.9	-0.9	-0.9	-0.9	-0.8	-0.7	-0.7	-0.6	-0.6	-0.5	-0.4	-0.3	-0.1	0.2
-0.8	-1.0	-0.9	-0.8	-0.8	-0.8	-0.8	-0.8	-0.8	-0.8	-0.8	-0.8	-0.7	-0.7	-0.6	-0.6	-0.5	-0.4	-0.3	-0.1	0.1	0.4
-0.7	-1.0	-0.9	-0.7	-0.7	-0.7	-0.7	-0.7	-0.7	-0.7	-0.7	-0.7	-0.7	-0.6	-0.6	-0.5	-0.4	-0.2	-0.1	0.1	0.3	0.6
-0.6	-1.0	-0.9	-0.7	-0.7	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.5	-0.3	-0.2	0.0	0.1	0.3	0.4	0.5	0.6	0.7
-0.5	-1.0	-0.9	-0.7	-0.7	-0.6	-0.6	-0.6	-0.6	-0.6	-0.5	-0.5	-0.3	-0.2	0.0	0.2	0.3	0.4	0.5	0.6	0.7	0.7
-0.4	-1.0	-0.9	-0.7	-0.7	-0.6	-0.6	-0.6	-0.6	-0.5	-0.5	-0.4	-0.2	-0.1	0.1	0.2	0.4	0.5	0.6	0.6	0.7	0.8
-0.3	-1.0	-0.9	-0.7	-0.7	-0.6	-0.6	-0.6	-0.6	-0.5	-0.4	-0.3	-0.1	0.0	0.2	0.3	0.4	0.5	0.6	0.7	0.7	0.8
-0.2	-1.0	-0.9	-0.7	-0.7	-0.6	-0.6	-0.6	-0.6	-0.5	-0.3	-0.2	0.0	0.1	0.3	0.4	0.5	0.5	0.6	0.7	0.8	0.9
-0.1	-1.0	-0.9	-0.7	-0.7	-0.6	-0.6	-0.6	-0.5	-0.4	-0.2	-0.1	0.1	0.2	0.4	0.5	0.6	0.6	0.6	0.7	0.8	0.9
0.0	-1.0	-0.9	-0.7	-0.7	-0.6	-0.6	-0.6	-0.5	-0.4	-0.2	0.0	0.2	0.4	0.5	0.6	0.6	0.6	0.7	0.7	0.9	1.0
0.1	-0.9	-0.8	-0.7	-0.6	-0.6	-0.6	-0.5	-0.4	-0.2	-0.1	0.1	0.2	0.4	0.5	0.6	0.6	0.6	0.7	0.7	0.9	1.0
0.2	-0.9	-0.8	-0.7	-0.6	-0.5	-0.5	-0.4	-0.3	-0.1	0.0	0.2	0.3	0.4	0.6	0.6	0.6	0.6	0.7	0.7	0.9	1.0
0.3	-0.8	-0.7	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	0.0	0.1	0.3	0.4	0.5	0.6	0.6	0.6	0.6	0.7	0.7	0.9	1.0
0.4	-0.8	-0.7	-0.6	-0.6	-0.5	-0.4	-0.2	-0.1	0.1	0.2	0.4	0.5	0.5	0.6	0.6	0.6	0.6	0.7	0.7	0.9	1.0
0.5	-0.7	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	0.0	0.2	0.3	0.5	0.5	0.6	0.6	0.6	0.6	0.6	0.7	0.7	0.9	1.0
0.6	-0.7	-0.6	-0.5	-0.4	-0.3	-0.1	0.0	0.2	0.3	0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.7	0.7	0.9	1.0
0.7	-0.6	-0.3	-0.1	0.1	0.2	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.9	1.0
0.8	-0.4	-0.1	0.1	0.3	0.4	0.5	0.6	0.6	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.9	1.0
0.9	-0.2	0.1	0.3	0.4	0.5	0.6	0.6	0.7	0.7	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	1.0	1.0
1.0	0.0	0.3	0.5	0.4	0.6	0.6	0.6	0.7	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Fig. 2. Control table with granularity of 0.1.

3.6 Tuning the Rule Base

As done in a previous work [22], we do not use expert knowledge or automated techniques based on optimization for designing the rule base. The rule base is simply built by mapping the entries in the control table to the rules of the rule base. In this particular case, there is even no need to perform this mapping because a generic MacVicar-Whelan rule base can be used instead. This is because the control table was generated by aggregating several control tables generated by several FLCs which used MacVicar-Whelan rule bases. The standard 7 by 7 MacVicar-Whelan rule base used for the ABR FLC is shown in Fig. 3. The linguistic variables (*Negative Small*, *Negative Medium*,

changeInError \ Error	1	2	3	4	5	6	7
1	1	1	1	1	2	3	4
2	1	2	2	2	3	4	5
3	1	2	3	3	4	5	6
4	1	2	3	4	5	6	7
5	2	3	4	5	5	6	7
6	3	4	5	6	6	6	7
7	4	5	6	7	7	7	7

Fig. 3. 7 by 7 MacVicar-Whelan rule base.

Negative Big, *Zero*, *Positive Small*, *Positive Medium* and *Positive Big*) are represented as values from 1 to 7. The first rule (in the first cell of the matrix) is read as: If the error is *Negative Small* and the change in error is *Negative Small* then the change in output is *Negative Small*.

Since there were no great variation among the entries of the individual control tables, the aggregated control table still mapped to a standard MacVicar-Whelan rule base. Should this be no longer the case, then a rule base has to be designed by mapping it from the control table. It should be noted that even if the control table does not map to a MacVicar-Whelan rule base exactly, it is still possible to use it since the membership functions can be optimized to produce the desired control surface. However, this is subject to some reasonable limits.

3.7 Tuning the Membership Functions

The membership functions are tuned using Differential Evolution. We used the sum of squares of errors (SSE) as objective function, a population of 50 individuals, $\alpha = 0.7$, and $Cr = 0.7$. We used triangular membership functions with a degree of overlapping of two [19-21]. We also used symmetrical fuzzy sets [18] so that the FLC produces symmetrical outputs (same magnitude but different directions for inputs of same magnitude and different directions). The universe of discourse of all the FLC variables is normalized to the range [– 1.0, 1.0]. The resulting membership functions are shown in Fig. 4 and the control surface of the FLC is shown in Fig. 5.

	Parameters – Vertices of fuzzy sets/Singletons						
	1	2	3	4	5	6	7
Error	– 1.0	– 0.760287	– 0.341034	0.0	0.341034	0.760286	1.0
ChangeInError	– 1.0	– 0.692270	– 0.582427	0.0	0.582427	0.692270	1.0
ChangeInOutput	– 1.0	– 0.664575	– 0.601014	0.0	0.601014	0.664575	1.0

Fig. 4. Parameters for membership functions.

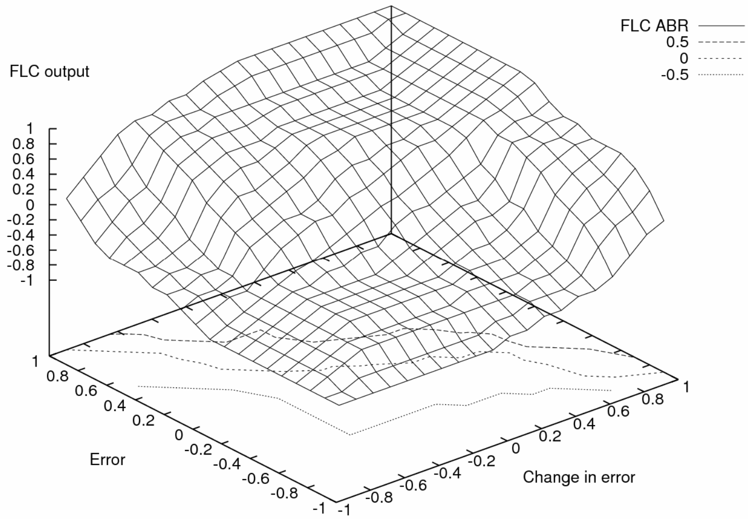


Fig. 5. Control surface.

4. SIMULATION

The simulation of the transportation of ABR and CBR/VBR streams was performed as a time-driven simulation of an ATM switch with an output link having a bandwidth of 155.52 Mbits/s. We used an output buffer of 1,024 cells for holding ABR cells and we selected the mid-point of the buffer to be the set point, that is, 512 cells. When selecting the sampling time interval (also the time interval for the simulation), we have to consider the fact that if the interval is too long, the burstiness of the VBR traffic may not be captured and if it is too short, the network will need to send feedback signals more frequently. We selected the time interval to be one milli-second. This represents the time to fill half of the buffer capacity at the rate of 155.52 Mbits/s.

Thus, every milli-second, the following is performed:

- computing the number of CBR/VBR cells to transmit using the aggregated MPEG frame sizes and displaying rate (frames per second);
- computing the number of ABR cells to transmit using the error in buffer level and the change in error (done by FLC);
- updating the maximum number of CBR/VBR and ABR cells on the output link and the output buffer levels;
- computing the error and change in error; and
- updating the statistics for computing the link utilization.

In our simulation we assume that ABR cells are always available for transmission. We believe that this is a reasonable assumption for our simulation studies since we interested in finding out the performance of our FLC as measured by the link utilization. If the availability of ABR cells is allowed to fluctuate, then this will also have an impact on the link utilization. It is much simpler to remove the influence of ABR availability on link utilization and to analyze only the effect of the FLC on link utilization.

We simulated only one ATM switch because it is trivial to extend the method to a network of switches. In the case of a network of switches, each switch in a transmission path determines the rate of ABR that it is capable of handling and only the minimum rate determined by these switches is transmitted across the network.

5. RESULTS OBTAINED

We tested the performance of our FLC in controlling the flow rate of ABR traffic on the aggregations of MPEG trace data described earlier. We used different aggregations for training and testing and found out the link utilization to be very close to 100%. We also tested the transient response of the FLC when sudden load changes occur in the flow of CBR/VBR streams. We simulated this by increasing the set point from 512 to 768 cells. It can be seen from Fig. 6 that the new buffer level is attained quickly and smoothly.

In many studies, the transient response of FLCs is demonstrated by a step increase (as done above). It is assumed that the FLC will perform equally well for a step decrease. This is very unlikely because different parts of the rule base and membership functions are activated for step increases and decreases. The only way to make the FLC perform equally for step increases and decreases is to explicitly make it symmetrical by using symmetrical rule bases (the MacVicar-Whelan is a symmetrical rule base) and membership functions. Symmetrical performance of the FLC is demonstrated in Fig. 6 by bringing down the set point to 256 cells. Our FLC performs equally well for both step increases and decreases. The small periodic disturbances shown in Fig. 6 are due to changes in frame size of the aggregated MPEG trace data. Every time the size of the aggregate frame changes, the FLC takes about 2 or 3 time instants to recover from the disturbance.

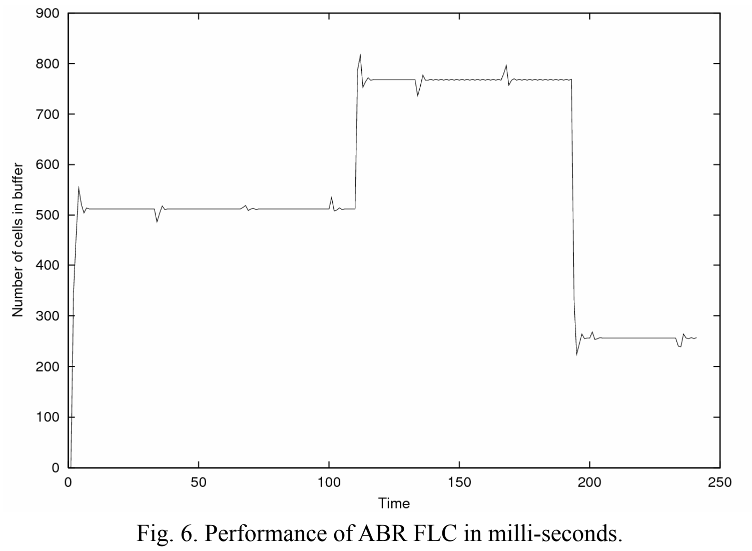


Fig. 6. Performance of ABR FLC in milli-seconds.

5.1 Evaluating the Results

Pitsillides and Sekercioğlu [15] proposed an FLC for ABR control that takes the output buffer length and its rate of change as input. The output of the FLC is the required flow rate of the ABR traffic. The main drawback of this work is that the FLC was designed and tuned by trial and error. Pitsillides *et al*. [23] improved their work by considering the transport of ABR traffic in the presence of CBR and VBR traffic. However, their FLC was still designed by trial and error.

Hu *et al*. [16] used a different FLC for rate control of ABR; they used a Takagi-Kang-Sugeno FLC [24] instead of the Mamdani FLC [4] which is commonly used. They used four input variables for their FLC: the queue length, the rate of change of the queue length, a derived measure which gives an indication of the demand of the ABR traffic, and the number of lost ABR cells. The drawbacks of this work are that there are too many input variables to the FLC for controlling the flow rate of ABR traffic, and that the use of two fuzzy sets per input variable will not give a control surface that is smooth enough for rate control of ABR traffic. Kwok and McLeod [25] used FLC input variables that are different from the traditional combination of a state variable and its rate of change. The rule base was also done by trial and error. Although, it was mentioned that training data was used to tune the membership functions, no details were given as to how they were obtained.

Our method for designing FLCs for ABR flow rate control is useful for systems without using a mathematical model. It is particularly useful when the control problem can be formulated such that the input variables to the FLC are the error and its rate of change and the output of the FLC is the change in control variable. Its merits are:

- It can be used for designing FLCs for systems without known mathematical models.
- Real MPEG audio and video trace data were used for the simulation.
- The FLC is able to monitor the ATM link utilization and maintain it close to 100%.
- The transient response of the FLC is quick, smooth and symmetric.
- The control table is obtained using a different simulation technique from the one used for call admission control. This resulted in formation of a complete control table, that is, one without any missing entries.
- The method can be used for designing both PI- and PD-like FLCs since when the control table approach is used, the rule base is simply derived from the control table.

Some limitations of the method are:

- Although the method can be used for designing both PI- and PD-like FLCs, we used it for designing only PI-like FLCs. To be more precise, the best FLCs determined by Differential Evolution for each simulation run used MacVicar-Whelan rule bases which are more appropriate for PI- than PD-like FLCs.
- We also need to further investigate how to merge several control tables into a single one as averaging the entries may not always be satisfactory.

It is well-known that on-line tuning using gradient descent is a very slow process. Furthermore, on-line optimization of certain parts of the FLC might have a negative bearing on other parts of the FLC. It is better to optimize the whole FLC structure in one step than in a long sequence of very small steps. Using Evolutionary Algorithms to optimize the FLC in one step ensures that the output of the FLC matches the desired control surface for every combination of input values.

6. CONCLUSIONS

In this paper, we have shown that if the control problem is properly formulated using sound control engineering principles, then the task of designing the FLC is somewhat simplified. In particular, if the inputs to the FLC are the error and its rate of change and the output of the FLC is the rate of change of the control variable, we can use a PD-like Mamdani FLC. Furthermore, we can use a standard MacVicar-Whelan rule base for the FLC such that only the membership functions need to be tuned. While tuning the membership functions, we used the knowledge gained by experimenting with the cart-pole problem [18] to make them symmetrical as this is a requirement of the particular control problem. We used a different approach to obtaining the entries in the control table. Since the control application is computationally tractable, we employed Differential Evolution to generate the best FLCs for a number of different situations. Control tables are then generated for each FLC and are merged into a single table for determining the optimized membership functions.

We have tested our design on several aggregations of real CBR/VBR MPEG data and we found out that the output of the FLC was symmetrical and the link utilization was close to 100%. We have also tested the transient response of the FLC by simulating a step increase as well as a step decrease. The response of the FLC is quick, smooth and symmetric. Future work includes: using the method to design PD-like FLCs; modelling the round trip delay; using the absolute rate; and formulating a more systematic method for merging several control tables into a single one.

REFERENCES

1. ATM Forum, *ATM User-Network Interface* (*UNI*) *Specification*, Version 3.1. Upper Saddle River, Prentice Hall PTR, NJ, 1995.
2. D. Pan, "A tutorial on MPEG/audio compression," *IEEE Multimedia*, 1995, pp. 60-74.
3. L. Roberts, "Enhanced PRCA (proportional rate-control algorithm)," An ATM Forum Contribution 94-0735R1, 1994.
4. S. Assilian and E. H. Mamdani, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, Vol. 7, 1974, pp. 1-13.
5. R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley and Sons, Inc, New York, 1994.
6. J. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition*, IEEE Press, New York, 1992.
7. P. Eyckhoff, *System Identification: Parameter and State Estimation*, Wiley, London, England, 1974.

8. C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller – Part I," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, 1990, pp. 404-418.
9. C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller – Part II," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, 1990, pp. 419-435.
10. T. J. Procyk and E. H. Mamdani, "A linguistic self-organizing process controller," *Automatica*, Vol. 15, 1979, pp. 15-30.
11. L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, 1992, pp. 1414-1427.
12. R. Holve and P. Protzel, "Generating fuzzy rules for the acceleration control of an adaptive cruise control system," in *Proceedings of North American Fuzzy Information Processing Society*, 1996, pp. 451-455.
13. A. Lekova, A. Nabout, and L. Mikhailov, "An inductive method for learning fuzzy membership functions," in *Proceedings of CESA'96 IMACS Multiconference on Control, Optimization and Supervision*, 1996, pp. 934-937.
14. K. Price and R. Storn, "Differential evolution," *Dr. Dobbs Journal*, 1997, pp. 18-24.
15. A. Pitsillides, Y. A. Şekercioğlu, and G. Ramamurthy, "Fuzzy backward congestion notification (FBCN) congestion control in asynchronous transfer mode (ATM) networks," in *Proceedings of GLOBECOM*, 1995, pp. 280-285.
16. Q. Hu, D. W. Petr, and C. Braun, "Self-tuning fuzzy traffic rate control for ATM networks," in *Proceedings of EEE International Conference on Communications*, 1996, pp. 424-428.
17. P. J. MacVicar-Whelan, "Fuzzy sets for man-machine interaction," *International Journal of Man-Machine Studies*, Vol. 8, 1976, pp. 687-697.
18. F. Cheong and R. Lai, "Designing a hierarchical fuzzy logic controller using differential evolution," in *Proceedings of IEEE International Conference on Fuzzy Systems*, 1999, pp. I-277-I-282.
19. F. Cheong and R. Lai, "Optimizing a fuzzy logic controller with well-formed fuzzy sets and rules," in *Proceedings of Eighth International Fuzzy Systems Association World Congress*, 1999, pp. 603-607.
20. F. Cheong and R. Lai, "Constraining the optimization of a fuzzy logic controller using an enhanced genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 30, 2000, pp. 31-46.
21. F. Cheong and R. Lai, "On simplifying the automatic design of a fuzzy logic controller," in *Proceedings of Annual Meeting of North American Fuzzy Information Processing Society*, 2002, pp. 481-487.
22. F. Cheong and R. Lai, "A hierarchical fuzzy-CAC for ATM networks," in *Proceedings of International Conference on Fuzzy Information Processing Theories and Applications*, 2003.
23. A. Pitsillides, Y. A. Şekercioğlu, and G. Ramamurthy, "Effective control of traffic flow in ATM networks using fuzzy explicit rate marking (FERM)," *IEEE Journal on Selected Areas in Communications*, Vol. 15, 1997, pp. 209-225.
24. T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, 1985, pp. 116-132.
25. A. Kwok and R. McLeod, "ATM congestion control using a fuzzy neural network," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, 1996, pp. 814-817.

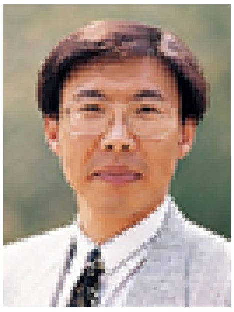
France Cheong obtained his Ph.D. in Computer Systems Engineering from La Trobe University in Australia. He is currently a senior lecturer in the School of Business IT at RMIT University in Australia. His current research interests include fuzzy systems, evolutionary computing and complex systems modeling and simulation.

Richard Lai holds a B.E.(Hons) and a M.E. from the University of New South Wales, and a Ph.D. from La Trobe University, Australia. He is currently with the Department of Computer Science and Computer Engineering at La Trobe University. He was ranked as the number one scholar in Systems and Software Engineering in a row of four years (1999-2002) by the editors of the Journal of Systems and Software, U.S.A. He has served on the program committees of many international conferences. He is a member of the editorial board of Computer Communications and the Journal of Systems and Software. His research interests include component based software engineering, software process, and software testing and measurement.

Yong-Soo Kim received the B.S. in Physics from Pusan University, Pusan, Korea, in 1979, and the M.S. in Computer Science from YonSei University, Seoul, Korea, in 1986. He received Ph.D. from Dongguk University, Seoul, Korea, in 1997. He is currently a Professor of College of Software at Kyungwon University, Korea. His research focuses on multimedia information management, performance modeling, and GRID computing systems.

Dae-Hoon Hwang received the B.S. in Mathematics from Dongkook University, Seoul, Korea, in 1977 and the M.S. in Computer Science from Choongang University, Seoul, Korea, in 1983. He received Ph.D. from Choongang University, Seoul, Korea, in 1991. He is currently a Professor of College of Software at Kyungwon University, Korea. His research focuses on multimedia information management, e-learning, and Internet application.

Hae-Seok Oh received his B.S., M.S. and Ph.D. degree in Computer engineering from Seoul University, Korea, in 1975, 1979 and 1989 respectively. He is currently a Professor of College of Software at Kyungwon University, Korea. His research interests are in the multimedia processing, HomeNetwork, sensor network and intelligent networks.



Ji Hong Kim received the B.E. in Electronics Engineering from KyungHee University, Seoul, Korea, in 1974, and the M.S. in Computer Science from California State University at Fullerton, California, in 1982. He received Ph.D. in Computer Engineering from KyungHee University, Seoul, Korea, in 1995. After working in Interpac Software at Santa Monica, California (1982-1989), he joined as a professor at KyungWon University. His research focuses on software engineering, and software product-line engineering.