A Fuzzy Neural CBR Channel Rate Controller for MPEG2 Encoders*

Maria C. F. De Castro, Fernando C. C. De Castro, Dalton S. Arantes and Dario F. G. Azevedo

Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações
Universidade Estadual de Campinas – UNICAMP
Caixa Postal 6101 – 13.083-970 – Campinas – SP – BRASIL
E-mail: cristina@decom.fee.unicamp.br, decastro@ee.pucrs.br, dalton@decom.fee.unicamp.br, dario@ee.pucrs.br

Abstract- A fuzzy algorithm is used as control surface for the buffer occupancy of a MPEG2 (Moving Picture Experts Group) video encoder. Based on scene features, a supervised algorithm trains a Radial Basis Function Neural Network (RBFNN). The so trained RBFNN acts as a predictor for the number of bits generated in a frame, so that the predicted buffer occupancy can be determined. The predicted and present buffer occupancies are applied to the fuzzy-generated control surface which yields the encoder quantizer step parameter. We compare the obtained results with the Test Model 5 standard rate control scheme.

1. Introduction

The MPEG-2 standard is a widely accepted video-compression technique that applies no standardization to the coding process, allowing the proposal of different approaches for the encoders. The MPEG coded data can be transmitted through constant bit rate (CBR) channels or through variable bit rate (VBR) channels. In the CBR case, due to the highly time-varying complexity of the video sequences, it is imperative the use of a buffer in order to properly stabilize the output rate to the channel [6]. As an example, perhaps one of the most demanding CBR buffer control tasks occurs in the transmission of medical images, where the requirements for image quality are severe and channel rates as low as 2 Mbps are not uncommon [8].

In a previous work [1] the authors developed a non-linear predictive video bit rate controller that uses a supervised RBFNN approach. The RBFNN training is carried out by updating the output neuron synapse weights, as well as the Gaussian centers, by means of the Stochastic Gradient (SG) algorithm [4]. The SG algorithm uses an error input derived from the desired output, which characterizes it as a supervised training process. The rate prediction at the RBFNN output is then applied to a non-linear mapping [2], which yields the encoder quantizer step (*mquant* parameter [3]).

In this work we present a fuzzy approach to determine the encoder quantizer step. The fuzzy controller implements a family of piecewise lines, each one with a different slope. The line set is conceived to properly control the buffer occupancy for time-varying complexity video signals. The obtained results with such simple rate control technique are similar to those previously obtained by the authors using the referred non-linear mapping technique [1]. In comparison with the Test Model 5 (TM5) standard rate control scheme [3], superior performance was obtained for the buffer occupancy and signal-to-noise ratio (SNR).

* Partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS).

2. RBF Neural Networks

A RBF neural network is a universal approximator [4]. It consists of an input layer with M nodes, a hidden layer with K neurons and an output neuron layer. In our case, since we are interested in estimating just the scene complexity on a frame by frame basis, the output layer has only one output neuron, as can be seen in Figure 1 [5].

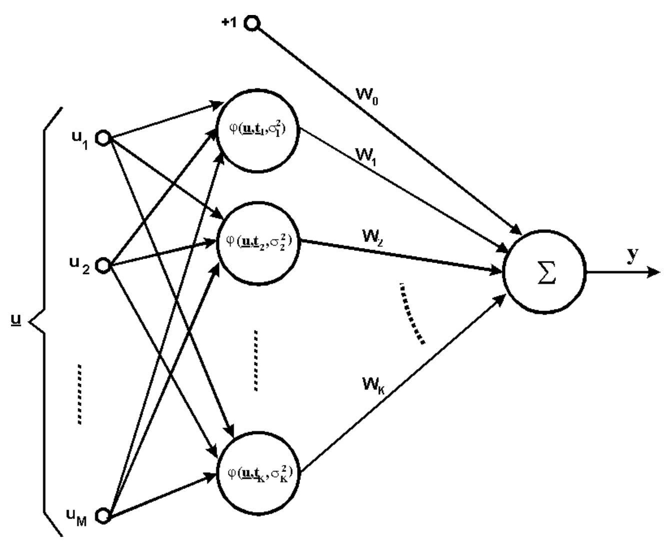

Figure 1: Radial Basis Function Network.

The k^{th} hidden neuron, $k=1,\ldots,K$, computes the distance between the center vector $\underline{t}_k \in \Re^M$ and the n^{th} input vector $\underline{u}(n) \in \Re^M$ presented to the RBF. The resulting distance is utilized as argument to a Gaussian function φ with variance $\sigma_k^2(n)$ as given by

$$\varphi(\underline{u}(n),\underline{t}_k(n),\sigma_k^2(n)) = \exp\left[-\frac{1}{\sigma_k^2(n)}\left\|\underline{u}(n)-\underline{t}_k(n)\right\|^2\right] \qquad (1)$$

The output of the k^{th} hidden neuron is applied to the output neuron through the weighting synapses w_k. The RBF output y to the input vector $\underline{u}$ is given by the weighted sum of all hidden neurons outputs, according to

$$y = \sum_{k=1}^{K} w_k \varphi\left(\underline{u},\underline{t}_k,\sigma_k^2\right) + w_0 \qquad (2)$$

The term $\varphi\left(\underline{u},\underline{t}_k,\sigma_k^2\right)$ in equation (2) is formally defined as the k^{th} Radial Basis Function, which computes the distance $D = \left\|\underline{u}-\underline{t}_k\right\|$ between the input vector $\underline{u}$ and the center vector $\underline{t}_k$ [4]. Notice that the output of each hidden neuron is a non-linear function of D. The synapse weight w_k represents the gain of the path which connects the output of the k^{th} hidden neuron to the output neuron. The constant term w_0 is the bias applied to the RBF.

The RBF training is performed by presenting a set of training vectors $\underline{u}$ to the network input. Upon the presentation of the n^{th} training vector $\underline{u}(n)$ and the associated desired output $y_d(n)$, the RBF free parameters w_k, $\underline{t}_k$ and σ_k^2 are updated by means of the SG algorithm [4][5], as shown by

$$w_k(n+1) = w_k(n) + \mu_w e(n)\varphi(\underline{u}(n),\sigma_k^2(n),\underline{t}_k(n)) \qquad (3)$$

$$\underline{t}_k(n+1) = \underline{t}_k(n) + 2\mu_t e(n) w_k(n)\varphi(\underline{u}(n),\sigma_k^2(n),\underline{t}_k(n))\frac{\underline{u}(n)-\underline{t}_k(n)}{\sigma_k^2(n)} \qquad (4)$$

And

$$\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma e(n) w_k(n)\varphi(\underline{u}(n),\sigma_k^2(n),\underline{t}_k(n))\frac{\left\|\underline{u}(n)-\underline{t}_k(n)\right\|^2}{\sigma_k^2(n)} \qquad (5)$$

where $e(n)=y_d(n)-y(n)$ is the error associated with the n^{th} presented training vector $\underline{u}(n)$ and μ_σ, μ_t, μ_w, are the learning rates for the Radial Basis Functions variances, for the center vectors and for the synapse weights, respectively.

As the training proceeds, the instantaneous error is minimized and the RBF acquires information about the underlying process associated with the training set. Specifically, the so trained RBF acts as a natural interpolator of the underlying process. It is capable to generate an output y with minimum error (in the mean square sense) for a presented vector $\underline{u}$ which is not a training vector but lies in its neighborhood. In this sense, the RBF is a non-linear predictor [5].

3. The Fuzzy Predictive Video Bit Rate Controller

Figure 2 shows the block diagram of the fuzzy neural predictive video CBR controller [1][2][5]. The n^{th} training vector at the input of the RBF neural network has 9 components: the variances of F_n, F_{n-1} and F_{n-2}; the variances of $(F_n - F_{n-1})$, $(F_{n-1} - F_{n-2})$ and $(F_{n-2} - F_{n-3})$; and T_n, T_{n-1} and T_{n-2}; where F_n is the n^{th} video frame at the encoder input and T_n is the type of F_n (1 for I-*frames*, 0 for P-*frames* and –1 for B-*frames* [6]).

The RBF output $y(n)$ is a prediction of the number of bits that will be stored in the buffer by the end of the n^{th} frame coding process. The RBF supervised learning via SG algorithm uses the error $e(n)$ obtained from $y(n)$ and from the encoder output bits $cbf(n)$ generated after the n^{th} frame coding.

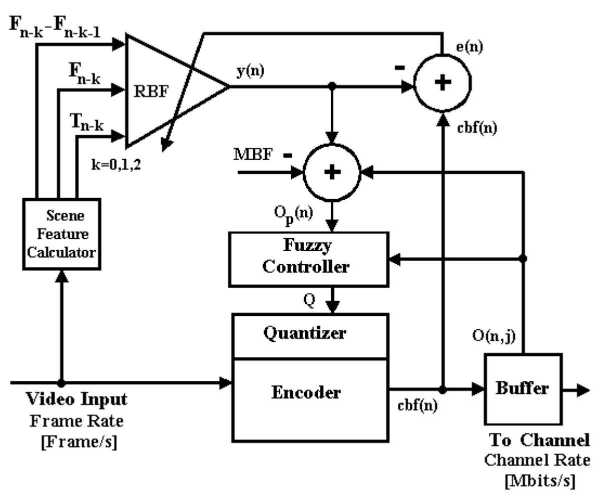

Figure 2: Fuzzy neural predictive video bit rate controller

Notice that adding $y(n)$ to $O(n,j)$ – the present buffer occupancy or the occupancy after the j^{th} macroblock of the n^{th} frame is coded – and subtracting the *mean bits-per-frame* MBF, we obtain $O_p(n)$ – the predicted buffer occupancy. $O_p(n)$ is an estimate of the buffer occupancy after the j^{th} macroblock of the $(n+1)^{th}$ frame is stored in the buffer [6]. MBF is defined as the ratio of the channel rate [Mbits/s] to the frame rate [frames/s].

The quantizer step Q or *mquant* [3] is the main control parameter for the buffer occupancy. The *mquant* control is achieved by means of a fuzzy controller. To perform the control task, the controller is provided with the information about the present buffer occupancy $O(n,j)$ and the predicted buffer occupancy $O_p(n)$. Based on this information it decides which quantizer step ($Q(n)$) will be applied to the encoder.

The input fuzzy linguistic variables (O and O_p) and the output fuzzy linguistic variable (Q) assume linguistic values in the set S={L_0, L_1, ..., L_{12}}. The set S is in ascending order, i.e., L_0 is associated with the lowest linguistic value and L_{12} is associated with the highest linguistic value. These thirteen linguistic values define the discrete fuzzy partition of the input and output spaces and were experimentally determined, according to the nature of the control problem. Due to the high sensitivity of the human visual system to image details, we had to use a larger number of values to create the discrete linguistic variable spaces than the usual seven or less partitions [7]. The differential variances between consecutive frames have been already considered at the RBFNN input. Therefore, the use of fuzzy linguistic variables that might take into account the buffer occupancy velocity is a redundant procedure. Figure 3 shows the membership functions of the fuzzy sets O, O_p and Q. All the values in the universes of discourse O, O_p and Q are normalized to the interval [0,…,1].

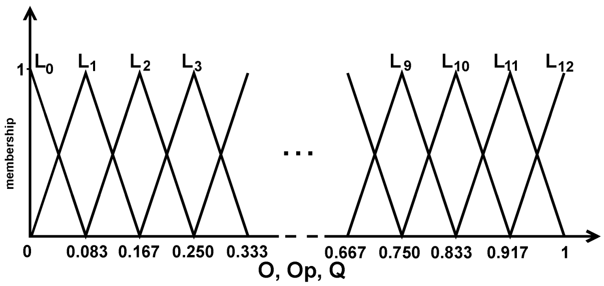

Figure 3: Fuzzy sets associated with the present buffer occupancy *O*, the predicted buffer occupancy O_p, and the quantizer step *Q*.

A set of fuzzy control rules is derived from a family of $Q \times O$ lines parameterized by O_p. Each family member is a pair of connected line segments labeled by the O_p value, as shown in Figure 4. Each segment in a particular member of the line family has its own slope (which was experimentally determined), and an upper and a lower saturation region limit each family member.

Notice that the higher the predicted occupancy O_p, the higher the slope of each $Q \times O$ line segment. This aims to prevent the buffer overflow for the situation of a large increase in buffer occupancy such as in drastic scene changes. On the other hand, for an almost empty buffer prediction (lower O_p levels), the $Q \times O$ line segments present minimum slope, yielding low *Q* values for a wide range of all possible *O* values, which avoids buffer underflow. The quantizing operation implies that lower levels for Q results in higher SNR.

The fuzzy control rules derived from Figure 4 are shown in Table 1. Based on Table 1 we can describe the rules in a linguistic form, for instance, "if *O* is $\boldsymbol{L_6}$ and O_p is $\boldsymbol{L_6}$, then *Q* is $\boldsymbol{L_6}$". This specific rule has the meaning: If the present buffer occupancy is at the medium level and the predicted buffer occupancy is at the medium level, then the quantizer step will have the medium level value.

In this work we present the results obtained from this particular set of fuzzy control rules. However, the implementation of the fuzzy controller allows a great number of different fuzzy rule sets. Each set of rules originated from different line families or from any other family of curves is able to generate new control surfaces for specific purposes.

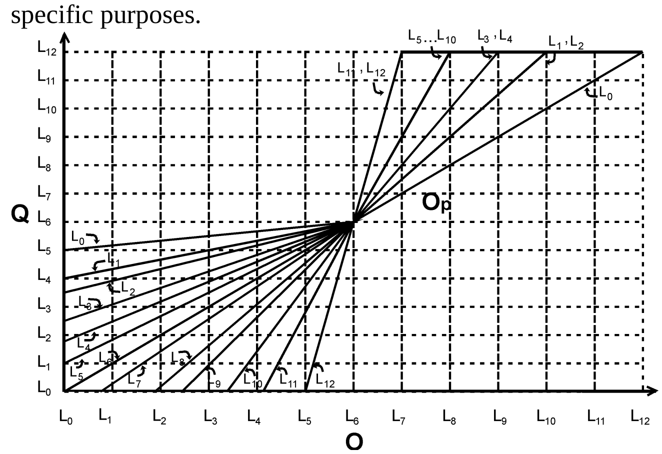

Figure 4: Family of $Q \times O$ lines parameterized by O_p.

O \ O_p	L_0	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}
L_0	L_5	L_4	L_4	L_3	L_2	L_1	L_0	L_0	L_0	L_0	L_0	L_0	L_0
L_1	L_6	L_5	L_4	L_4	L_3	L_2	L_1	L_1	L_0	L_0	L_0	L_0	L_0
L_2	L_6	L_5	L_5	L_4	L_4	L_3	L_2	L_2	L_0	L_0	L_0	L_0	L_0
L_3	L_6	L_5	L_5	L_5	L_4	L_4	L_3	L_3	L_1	L_0	L_0	L_0	L_0
L_4	L_6	L_6	L_6	L_5	L_5	L_5	L_4	L_4	L_3	L_2	L_2	L_0	L_0
L_5	L_6	L_6	L_6	L_6	L_6	L_5	L_5	L_5	L_5	L_4	L_4	L_3	L_0
L_6	L_6	L_6	L_6	L_6	L_6	L_6	L_6	L_6	L_6	L_6	L_6	L_6	L_6
L_7	L_7	L_8	L_8	L_8	L_8	L_9	L_9	L_9	L_9	L_9	L_9	L_{12}	L_{12}
L_8	L_8	L_9	L_{10}	L_{10}	L_{10}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}
L_9	L_9	L_{11}	L_{11}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}
L_{10}	L_{10}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}
L_{11}	L_{11}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}
L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}	L_{12}

Table 1: Fuzzy control rules for Q.

To obtain the resulting fuzzy rules we apply the min t-norm to model the connective *and* and the Larsen fuzzy conjunction to model the decision-making logic [7]. To perform the aggregation rule we use the union operator, modeled by the max s-norm. The center of area method is applied for defuzzification [7].

4. Experimental Results

In this section we present two examples in order to compare the performance of the RBF fuzzy predictive video CBR controller with that of the Test Model 5 (TM5) heuristic.

In both cases we used standard video sequences, of size 720×480 pixels, a video input frame rate of 30 frames/s and an output channel rate of 3.5Mbits/s. The RBF learning rates were set to: μ_t=0.15, μ_σ=0.5 and μ_w=0.1. The buffer size adopted was $2 \times MBF = 233333$ bits, as suggested in the TM5 [3].

In the first case we used the standard video sequence Mobile. Figures 5 and 6 show respectively the SNR and the percent buffer occupancy versus the frame index obtained for this sequence.

Notice that the nominal buffer occupancy as well as the buffer occupancy variance attains values substantially smaller for the fuzzy neural controller than for the TM5 controller. For the fuzzy neural controller, the buffer occupancy is kept around 50% along almost all coding process. However, as we can see in Figure 5, the buffer stabilization does not imply in SNR losses.

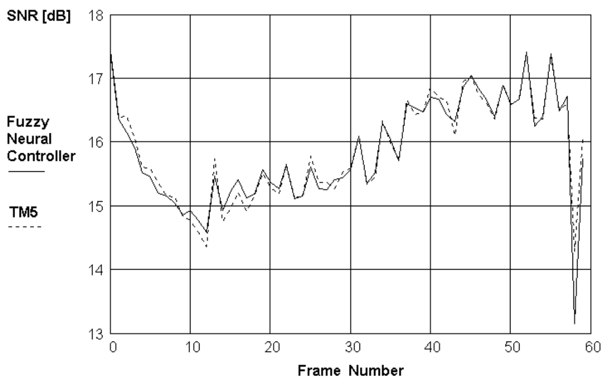

Figure 5: SNR [dB] (Mobile sequence).

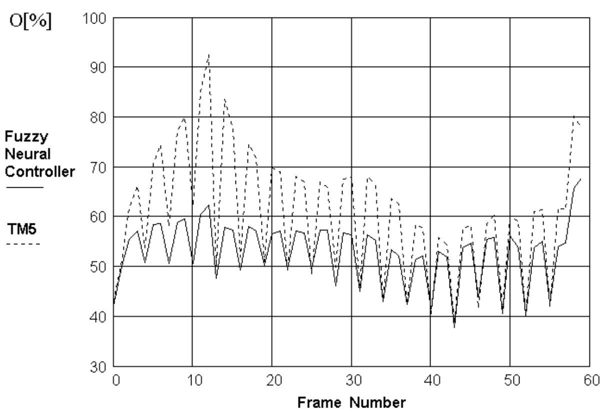

Figure 6: Percent buffer occupancy for Mobile sequence.

Figures 7 and 8 show respectively the SNR and the percent buffer occupancy versus the frame index for the juxtaposed sequences Mobile & Kiel. This sequence is obtained with the 30 first frames of the Mobile sequence and the last 30 frames of the Kiel sequence appended to it. The other fuzzy parameters, the RBF parameters and the video parameters are all kept the same.

It is important to note that the two sequence composition introduces a drastic scene change [5]. However, in despite of the drastic scene change, the fuzzy neural controller has kept the buffer occupancy in considerable lower levels, if compared with the TM5 control, for the same SNR levels.

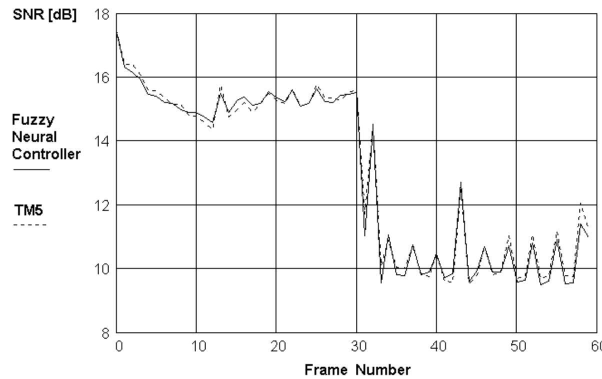

Figure 7: SNR [dB] (Mobile & Kiel sequence).

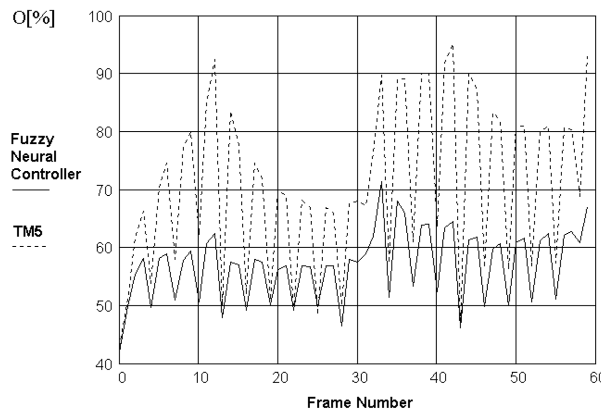

Figure 8: Percent buffer occupancy for Mobile & Kiel sequence.

5. Conclusions

A fuzzy technique was proposed in this paper to perform the buffer rate control of a predictive video CBR controller for MPEG2 encoders. The fuzzy control was applied to determine the encoder quantizer step parameter, as a function of the present and predicted buffer occupancies.

Concerning the buffer occupancy and the SNR, the results obtained with the proposed technique are similar to those obtained by the authors with the non-linear mapping technique described in [1]. If compared with the Test Model 5 rate control standard technique [3], the results obtained are superior with respect to the nominal buffer occupancy and the buffer variance, for the same SNR.

The fuzzy techniques applied in this work are quite simple and make room for the use of more elaborate techniques, aiming to improve the fuzzy controller performance. For instance, a controller based on adaptive fuzzy sets may be derived from this implementation. This approach improves the controller performance since it can adapt itself to the video signal statistics.

However, the main contribution of this work is to make possible the generation of empirically or experimentally conceived control surfaces, for specific purposes other than the CBR goal. The fuzzy controller circumvents the need of analytical functions to describe the control surface, since any conceivable surface can be generated by simply changing the set of fuzzy rules.

References

[1] Fernando C. C. De Castro, Maria C. F. De Castro and Dalton S. Arantes. A Supervised neural constant bit rate video controller for MPEG2 encoders. *IEEE Proceedings of ITS'98, 1998.*

[2] Y. Saw, P. M. Grant, J. M. Hannah and B. Mulgrew. Nonlinear predictive rate control for constant bit rate mpeg video coders. *IEEE Proceedings of ICASSP'97, 1997.*

[3] ITU-T SG15. Video codec test model *TM5*, January 1995.

[4] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, New Jersey, USA, 1996.

[5] Fernando C. C. De Castro, Maria Cristina F. De Castro and Dalton S. Arantes. *Controle de Taxa CBR em Sistemas MPEG com Redes Neurais Preditivas RBF*. RT-002-UNICAMP-TELEBRÁS-825/97, 1997.

[6] ISSO/IEC-JTC1/SC29/WG11 and ISSO/IEC-DS-13818-2. Information technology – generic coding of moving pictures and associated audio information: Video. 1994.

[7] W. Pedrycz and F. Gomide. *An Introduction to Fuzzy Sets – Analysis and Design*. The MIT Press, Cambridge, Massachusets, USA, 1998.

[8] http://www.nren.nasa.gov/echo.html