ADAPTIVE CODING OF TIME VARYING TRANSMISSION CHANNELS USING NEURO-FUZZY NETWORKS

Sorina Zahan
Department of Communications,
Technical University of Cluj-Napoca
26, G. Baritiu str., 3400- Cluj-Napoca, Romania
E-mail: sorina.zahan@com.utcluj.ro

Abstract.

This paper aims to present a system that controls the adaptive coding scheme of time varying transmission channels, e.g. radio channels. It is implemented by the neuro-fuzzy network GARIC [3] and is able to adapt itself both to short and long term changes of the channel characteristics. Another important advantage is that the system does not rely on any specific model of the channel. A large part of the paper is devoted to the experimental work. All the results are new and are being published for the first time. Many conclusions can be drawn from the experiments. Very important is that, due to its adaptation capability, the system proves a large degree of flexibility and, in the same time, performances comparable or even better than the ones obtained using systems that "know" the model of the channel they are working on.

1. INTRODUCTION

Telecommunication channels depend upon a large number of parameters that are time and/or frequency varying. This is the case of the radio channels - mobile or fixed, terrestrial or satellite [8-10], cables, magnetic recording systems, etc. Together with techniques like equalization or diversity, channel coding is used to improve the quality of the received signal.

Channel coding, or error control, is usually achieved by one of the two well-known coding schemes, namely automatic-repeat-request (ARQ) and forward error correction (FEC). The first one relies on error detecting codes, performs the retransmission of the corrupted information and requires a return channel. The second scheme (FEC) is based on error correcting codes and does not require a return channel. A third scheme can be also employed for the channel coding, namely hybrid ARQ. This is a mixture of the first two, and implies both error correction and error detection. All these schemes use *non-adaptive error control*, i.e. the selected code(s) has a fixed detection/correction capability, generally providing an acceptable error probability at the end user.

Both from the quality and transmission efficiency points of view, a much more attractive approach is the *adaptive error control (AEC).* The channel coding is adaptive if the detection/correction capability (CC) varies according to the state of the channel, attempting thus to minimize the redundancy introduced by the control information. Either FEC or hybrid ARQ schemes can be adaptive but in both cases the return channel is necessary for the control of the coder. Relative to non-adaptive schemes, code rate improvements up to 40 % are reported. Interesting reviews of such schemes can be found in [6], [12].

An adaptive scheme has some implementation requirements that differ from the non-adaptive case. Not only that the return channel is compulsory, but both CC/CD coder and decoder should be able to change on line the code. Possible solution could be to use dedicated chips or, better, DSP's. In the case of high-speed transmissions a neural architecture could also be employed to implement the decoders [14].

When an adaptive coding scheme is implemented there are two main (and quite different) problems to be solved:

a) the estimation of the current state of the channel

b) the selection of the optimal code corresponding to the actual (estimated) state of the channel or, more generally, the selection of the optimal family of codes.

A lot of work in adaptive channel coding was dedicated to the second problem. Nevertheless, it is clear that a good estimation of the channel state, based on realistic premises, is a prerequisite for the optimality of the scheme. Related to this problem of estimation, there are two ideas that characterize the reported approaches, independent upon the classification or operating principle of the adaptive channel coding system. Firstly, *the controller* (the estimator) *relies on some more or less general model of the channel*, usually a probabilistic approximation like finite-state Markov chain [12]. Secondly, *the decision is made on threshold logic bases*. This decision is generally made taking into consideration the number of corrupted blocks observed during a certain amount of time.

In this paper a different approach to the estimation of the channel state or, more precisely, the estimation of the current correcting requirements of the transmission system is presented. The device that controls the correcting capability of the channel coder is based on fuzzy logic. It has a neural implementation, the technique underlying the on line adaptation of the system being reinforcement learning.

In the following we will focus on the principles of the neuro-fuzzy adaptive error control. The implementation of the controller, its modular structure and operating principles are then presented. Special attention will be paid to the experimental results that will be presented in some detail in section 4. The most important conclusions that can be drawn from the experiments will be given in the final section.

2. ADAPTIVE CHANNEL CODING CONTROLLED BY THE NEURO-FUZZY SYSTEM

During the last years we have been witnessing an increasing interest in the so-called blind signal processing, with emphasize on blind system identification and estimation. This means that a system is identified without knowledge regarding its input, or the inputs are estimated without knowing the model of the channel. In the case of wireless communication, there are several motivations behind this interest. First, it is desirable to eliminate the redundancy introduced by the training sequences

that are typically used to build the model of the channel. Second, rapid variations of the propagation channel seriously affect the accuracy of the built model. Third, if a processing relies on a general model, the performance may be modest because of the inherent approximation of the model [11].

In respect of blind processing for time varying channels a lot of work has been done and impressive results have been obtained mainly related to equalization and diversity techniques [4]. Nevertheless, blind approaches are certainly attractive for adaptive channel coding, too.

In [15] we proposed a model independent algorithm for adaptive channel coding (mainly oriented to FEC adaptive schemes). It is based on fuzzy logic and is implemented by a fuzzy adaptive system. It presents some important advantages, like:

- it does not rely on a channel model
- it uses more extensively the information related to the channel state because it takes into consideration not only the number of corrupted blocks, but also the average corruption degree (computed during the observation interval using fuzzy quantifiers)
- it is able to adapt itself to the channel (it "learns" the channel)
- it demands a very simple and accessible description of the adaptation strategy.

Nevertheless, it presents also some drawbacks:

- the tuning of the fuzzy adaptive system (namely its knowledge base) had to be done manually, which, on one side, is a tedious task and, on the other side, affects the adaptation performances of the whole system
- the response of the system to channel state changes is rather inertial, mainly when these changes are dramatic in terms of bit error probability

To remove these drawbacks, still maintaining the important advantages of the fuzzy approach, a neural implementation of the fuzzy system was considered [16]. The resulting neuro-fuzzy network is able to adapt itself to the changes of the controlled system (the communication channel). Although unsupervised learning is the natural choice for blind signal processing [1], reinforcement learning can lead to better performances, if applicable. This is the reason why we chose Berenji's GARIC (*Generalized Approximate Reasoning-Based Intelligent Control)* architecture - which is based on reinforcement learning, to implement the kernel of the adaptive coding system.

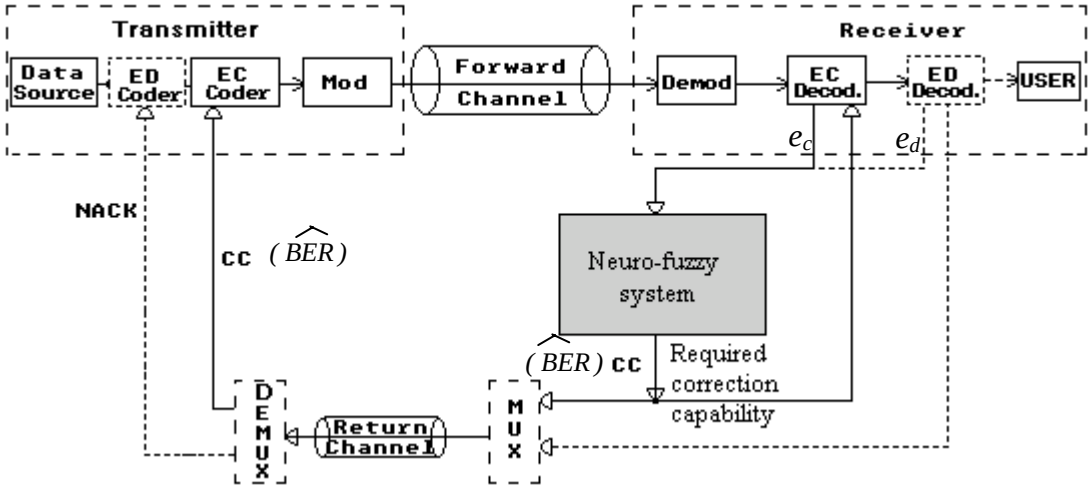

Figure 1. Adaptive channel coding controlled by the neuro-fuzzy system

The structure of the system is depicted in figure 1. The error correction coder (EC) that is located at the transmitter is controlled by the neuro-fuzzy system, which is located at the receiver. The controller makes its decision without taking into consideration any mathematical model of the channel. *Due to its neural component, it is able to adapt itself both to short term and long term changes of the channel characteristics*. The only information that reflects the current state of the channel and is available at the input of the system represents the number of corrected/detected errors e_c/e_d (per block or codeword). Its value is directly available at the output of the Viterbi decoders and, as shown in [12], can be easily computed for any other type of hard or soft decoders. The number of errors accurately reflects the state of the channel as long as the information was sufficiently protected, i.e. the state of channel was not underestimated in terms of bit error rate (BER). If the number or errors exceeded the correction/detection capability of the code in use, the number of "corrected"/"detected" errors can not lead to an accurate appreciation of the current state of the channel. For this reason, the controller's input is usually the number of erroneous blocks (counted within an observation interval). In [16] we showed that taking into consideration both the number of corrupted blocks and the average corruption degree, the information related to the channel state is used more extensively. The average corruption degree was computed during the observation interval using fuzzy quantifiers.

Based on the actual input, the neuro-fuzzy system estimates the current state of the channel. It can provide at the output the bit error rate or simply the required correction capability CC and transmits it through the return channel to the EC coder. Here, the right correction code will be selected and applied to the following data, till a new command will be received from the other end

of the telecommunication system. In the case of a hybrid ARQ scheme (dashed lines) the repetition request has to be also transmitted over the return channel.

3. THE NEURO-FUZZY SYSTEM

The structure of the neuro-fuzzy system is depicted in figure 2. It is composed by two subsystems:

- The *neuro-fuzzy network* (GARIC) that actually performs the control action
- The *observation duration subsystem*, which is used to establish the observation duration for every decision cycle.

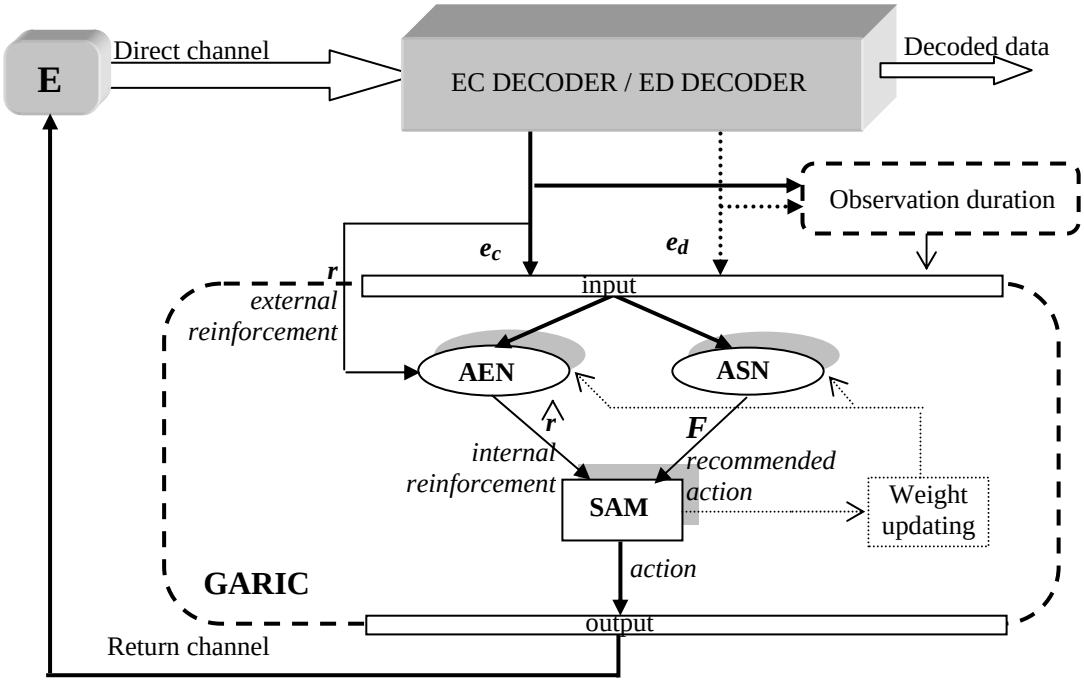

Figure 2. Internal structure of the neuro-fuzzy system

3.1. The neuro-fuzzy network

3.1.1. Structure

As we have already mentioned, the GARIC neuro-fuzzy network [2,3] represents a neural implementation of a fuzzy system whose adaptation relies on reinforcement learning.

Reinforcement learning performs the adaptation of the neural network parameters (weights) according to a performance measure called *reinforcement signal.* To be applicable, some rough information regarding the effect of the dynamic system's actions (good, poor, etc.) has to be available. This effect is expressed by the (external or primary) reinforcement signal and evaluated by the *critic* in a way that is useful to the control objective. The critic is an internal evaluator that, according to the reaction of the controlled environment, produces the internal (heuristic) reinforcements associated to the specific input state [5].

GARIC neuro-fuzzy network consists of three subsystems:

1. **The action evaluation network (AEN).** AEN plays the role of the critic. Therefore, the input vector has to encode the state of the process that is controlled by the fuzzy system. According to this input and to the external reinforcement signal *r*, AEN produces the internal reinforcement. The latter represents a performance indication or evaluation of the state of the process and is used both for updating the weights of the two neural networks (AEN and ASN) and to produce the actual action upon the system (the GARIC decision). AEN is a feedforward neural network, having a single hidden layer and a one-node output layer. The nodes from the input layer are connected both to the hidden neurones and to the output node.
2. **The action selection network (ASN).** ASN is the neural network that actually implements the fuzzy system. ASN choose the appropriate action F according to the current inputs and taking into consideration the knowledge base (which contains the fuzzy If-THEN rules). *During the adaptation process the fuzzy sets from the term set of the output variable (i.e. pertaining to the rules' consequence) are modified.*
3. **The stochastic action modifier (SAM).** Taking into account the recommended action given by ASN and the internal reinforcement given by AEN, SAM stochastically generates the final action of the neuro-fuzzy system. The final decision is a gaussian random variable having the standard deviation equal to the internal reinforcement signal and the mean equal to the ASN decision F. This solution leads to a better exploration of the state space but it introduces a certain amount of imprecision. As the experiments revealed, in certain circumstances the cost of this imprecision might become high.

3.1.2. Inputs

The inputs of the neuro-fuzzy system may differ depending upon the type of the adaptive coding. The simplest and most general choice is to take into consideration the number of erroneous blocks (codewords) counted during the observation interval, possibly using a fuzzy quantifier like "most" [15]. Another choice could be to use the relative corruption degree (RCD) [16] as a measure of the degree of corruption affecting the received blocks. RCD represents the number of previously corrected/detected errors relative to the actual CC. Fuzzy quantifiers can be also used.

The approach presented in this paper is dedicated to applications when both forward error correction and error detection coding are used, a common choice for many digital wireless communication systems [8]. In this case, we take as the input of the neuro-fuzzy system a combination of e_c (the indication of the error correction decoder) and e_d (the indication of the error detection decoder). e_c indicates the number of corrected errors/block whilst e_d indicates merely the

presence (e_d = 1) or absence (e_d = 0) of detected errors after correction. The input of the neuro-fuzzy system is called *estimation error* (EE).

We denote by N the number of blocks in the current observation interval, by e_{ci} (e_{di}) the indication of the error correction (detection) decoder corresponding to the ith block and by CC the actual value of the correction capability. If no errors are detected within the N blocks, EE is given by (1).

$$\mathrm{EE} = \frac{\mathrm{CC} - \max_i e_{ci}}{CC} \quad ; i = \overline{1,N} \tag{1}$$

If during the observation interval there are erroneous blocks detected after correction, i.e. exists a block i for which e_{di} = 1, the absolute value of EE will be given by the block error probability (P_b), measured after correction. For reasons related to the representation of the fuzzy variables we will take EE = - P_b. Therefore, in this case EE is given by (2).

$$\mathrm{EE} = -\frac{\sum_i e_{di}}{N} \quad ; i = \overline{1,N} \tag{2}$$

If EE = 0 there was a *correct estimation* of the channel state. If 0< EE ≤ 1 the state of the channel was *overestimated* in terms of BER (the real state of the channel was better than the estimated one). In this case, the efficiency of the transmission is low. If -1 ≤ EE < 0 the channel state (i.e. BER) was *underestimated* and consequently the user data contain errors.

3.1.3. Output and reinforcements

The output of the neuro-fuzzy system can be adapted to the application's needs. It can be the *recommended correction capability* CC (meaning the estimated number of bit errors per block) or *estimated BER* (computed as the ratio between CC and the length of the blocks). The system could directly provide an index to the correcting code to be used, if a table of correspondence CC-> code or BER->code was implemented.

The external reinforcement is positive if the previous decision was correct and negative in the other case. To define a correct decision different strategies can be implemented. For example, when the throughput efficiency is the most important, an optimal decision would be to have $maxe_c$=CC (i.e. EE=0), whilst a negative reinforcement is provided to the system when $maxe_c$<CC.When a high reliability has to be provided an overprotection would be considered optimal. To implement these strategies a threshold Θ can be set to positive values so that a positive external reinforcement will be given if $maxe_c$ = CC-Θ. An error detection indication or a different value of $maxe_c$ lead to a negative external reinforcement.

During the experiments that will be presented in the following section, we used external reinforcements of ± 0.05 and Θ=0. The learning rates for AEN network layers were both taken 0.001. The same value was used for the ASN learning rate.

3.2. The observation duration subsystem

The observation duration subsystem is a simple fuzzy system used to decide the optimal length of the observation interval. The inputs are EE and $maxe_c$. The output variable is the number of units (blocks or codewords) to be observed in order to take one decision. Its rules are mainly based on the presumption that a bad state *lasts less* than a good one [12]. They state that if the transmission system is in a good and adapted state ($maxe_c$ is small and $maxe_c$ = CC - Θ) the observation interval should increase. For a bad and/or unadapted state, the observation duration should decrease. In the following experiments the observation duration may vary between one block and five blocks.

4. EXPERIMENTAL RESULTS

Experiments have been performed using software simulation of the transmission system. An adaptive coding application using both forward error correction and error detection coding was considered. The channel was simulated by finite-state Markov chain. The system is by no means aware of the channel model, the input of the neuro-fuzzy system being the estimation error EE given by (1) and (2), respectively. The variation of CC (CCvar) was considered as output. CCvar represents the recommended increasing/decreasing of CC relative to the actual value. To study the behavior of the controller, during the experiments have been determined:

a) *underestimation error* (UE) computed as the number of blocks that have been underprotected relative to the total number of transmitted blocks. For perfect codes, UE actually represents the block error probability after EC decoder.
b) *overestimation error* (OE) computed as the number of blocks that have been overprotected relative to the total number of transmitted blocks. For a threshold Θ=0 (see 3.1.3), overprotection means a recommended CC strictly greater than the number of bit errors/block.
c) the duration of the adaptation *to a new state of the channel*
d) the adaptation performed within ASN *upon the fuzzy sets*

Different channels, different term sets of the fuzzy variables and different rule bases have been experimented.

Experiment 1

In the first experiment, a strategy in which reliability is (slightly) more important than redundancy was tested. A single rule base has been used. It contains five rules:

IF EE IS VS THEN CCvar IS largepos

IF EE IS S THEN CCvar IS smallpos

IF EE IS M THEN CCvar IS zero

IF EE IS L THEN CCvar IS smallneg

IF EE IS VL THEN CCvar IS largeneg

The experiments were carried on using two term sets for EE (figure 3 and figure 4). They differ only by the spread of the term set *medium*. The initial term set for the output variable is the same (figure 3). The modal values of the fuzzy sets are also the same. They are listed in Table1.The modal value of the fuzzy set *medium* corresponding to the input variable (0.1) indicates a strategy in which reliability is (slightly) more important than redundancy.

A poor quality two-state channel was considered. In one state BER=0.1. In the other one BER=0.08. Data rate is 9600bps. The block length is 148 bits.

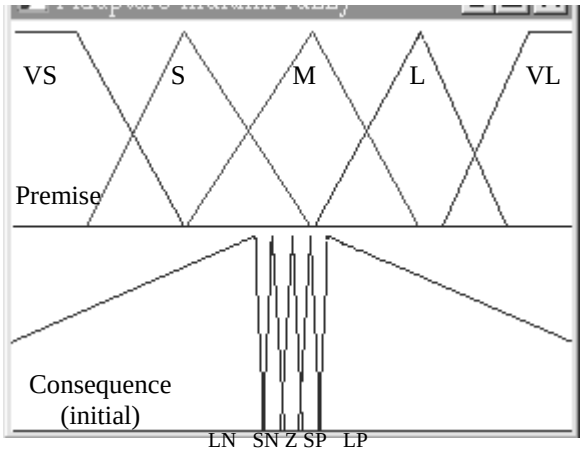

Figure 3. First (initial) term sets in exp. 1

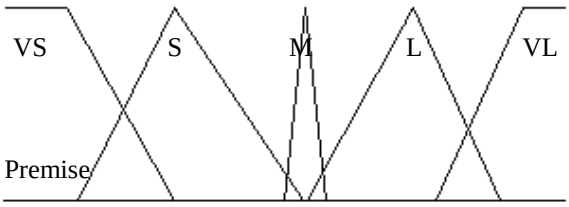

Figure 4. Second term set of EE in experiment 1.

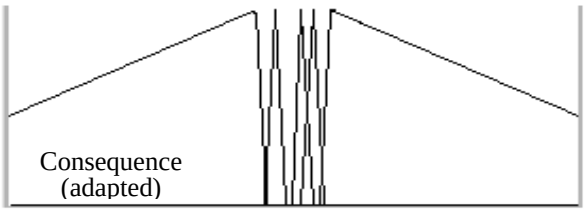

Figure 5. Adaptation of fuzzy sets after 30000 received blocks

After 10000 received blocks the UE rates were 4×10^{-2} for the first term set and 4.7×10^{-3} for the second term set. OE rates were 0.18 and 0.08, respectively. Afterwards, both systems presented the tendency to significantly increase the redundancy, accompanied by a reduction of the adaptation duration (usually 3-4 blocks, but also 0 blocks due to Stochastic Action Modifier). After 25000 received blocks the behavior seems rather stable, significant adaptation of the fuzzy sets taking place no more. The reinforcement of the overprotection tendency is clearly marked by the right shifting of the fuzzy set Z (figure 5). On the other hand, the sets SN and Z do not overlap anymore.

Table 1. Initial modal values of the fuzzy sets

VS	**S**	**M**	**L**	**VL**

-1	-0.5	0.1	0.6	1
LN	**SN**	**Z**	**SP**	**LP**
-2	-1	0	1	2

After 60000 blocks, the overall UE rates were 1.8×10^{-2} and 3.9×10^{-3}, respectively. As expected, UE rate became extremely high: 0.95 and 0.85, respectively. The maximum adaptation duration was 4 blocks.

Some conclusions can be drawn. First, the modification of the system (of the fuzzy sets) is slow. Second, the tendency is to reduce the error increasing rather dramatically the redundancy. Finally, the second term set is obviously better both in terms of error and redundancy.

What can be done to improve the performances? To increase the speed of adaptation to a new state of the channel, a larger initial distance between the fuzzy sets corresponding to the output variable should be considered. As it could be observed, because the third rule *(M->zero)* is much more often activated than the others, the neuro-fuzzy network can not significantly change the other fuzzy sets (LN, SN, SP, LP). For this reason, the initial choice for these sets remains very important. To ameliorate the redundancy, a null modal value for the fuzzy set *medium* may be more appropriate.

A final problem represents the action of the Stochastic Action Modifier (SAM). Interesting enough, after a while (around 12500 received blocks) although the redundancy increased significantly, the information being thus overprotected almost all the time, the underestimation error did not decrease as strong as expected. The reason was the action of SAM. The effect of its action for a situation in which the channel state remains constant for a longer duration is illustrated in figure 6. In this figure are plotted the system's decision (CC), the maximum number of errors that actually occurred in the blocks that were received during each observation interval and the input of the neuro-fuzzy network (EE). The strength of the SAM's action is remarkable, although in this case it did not introduce underprotection (EE stays positive).

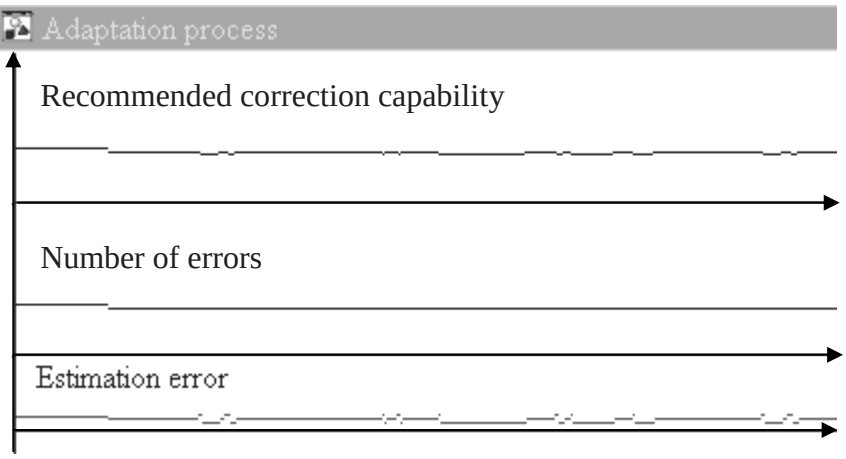

Figure 6. The action of Stochastic Action Modifier (SAM)

Experiment 2

In order to decrease the redundancy, experiment 1 is performed again using a null modal value for the fuzzy set *medium* and a different spread for the fuzzy sets. The term sets used in experiment

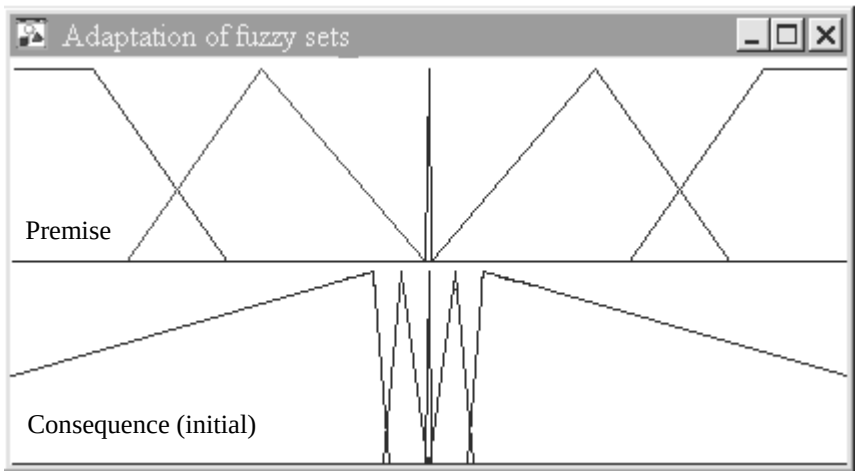

Figure 7. Term sets used in experiment 2

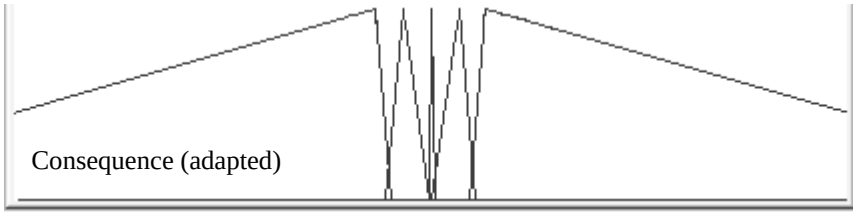

Figure 8. Adapted fuzzy sets in experiment 2 (after 25000 received blocks)

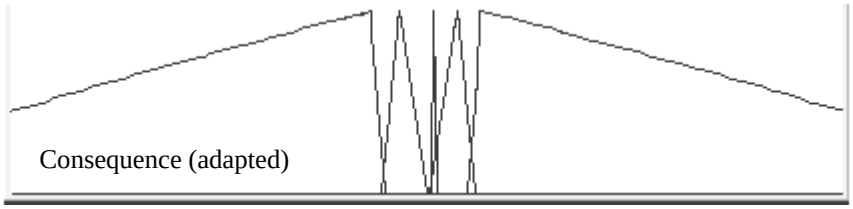

Figure 9. Adapted fuzzy sets in experiment 2 (after 120.000 received blocks)

2 are depicted in figure 7.

The UE rate was more than twice higher than in the previous experiment (UE<0.01) but the redundancy is much smaller (OE< 0.0128). EE =0 during 99% of the time. The adaptation duration is higher (9 ÷ 21 blocks, typically 9 ÷12 blocks). The system manifests overprotection only during the adaptation regime and sometimes for longer constant channel states (over 1000 received blocks) because of SAM's action. During the adaptation to a new channel state both over and underprotection may occur. During adaptation UE is 0.4 ÷ 0.8 when the channel moves to a worse state and 0.1 ÷ 0.3 when the channel moves to a better state. It decreases in time because of the adaptation of the fuzzy sets, reaching after 20 000 received blocks half of these values. The adaptation duration is slightly smaller for a transition to a better state (9 ÷15 blocks, typically 9 ÷10 blocks). The adaptation of the fuzzy sets is less impressive and even slower. The situation after 25000 received blocks is depicted in figure 8. Interesting, the stochastic action of the modifier becomes visible only after 20000 received blocks.

After 25000 received blocks the channel was changed to a better one: a three-state channel having BER_1= 0.001, BER_2= 0.024, BER_3= 0.048. The overall UE (including the underprotected blocks from the previous phase of the experiment) decreased to 6.5×10^{-3} at an overall OE of 0.33. It is important to mention that the errors are not dependent upon the order of the channel states. The adaptation duration (in blocks) is presented in table 2.

Table 2. The adaptation duration (in blocks) for different channel transitions in experiment 2

To \ From	BER=0.001	BER=0.024	BER=0.048
BER=0.001	-	0 ÷1	1 ÷3
BER=0.024	1	-	3 ÷7
BER=0.048	10 ÷20	5 ÷13	-

There are three causes of overprotection: the condition to have a minimum RCC of 1, even for error-free blocks (99.2% of overprotection cases), transitions to a better state (0.8% of cases) and SAM's action during longer channel states (negligible). In order to have a more accurate value for OE, after a total of 75000 received blocks we continued the experiment for a three-state channel having a very low probability of error-free received blocks (BER_1= 0.012, BER_2= 0.024, BER_3= 0.048). The overall value of OE (including the previous phases) became lower than 0.01. Because of SAM action, UE has a slight increase from 0.0065 to 0.009. The adaptation to a new state took

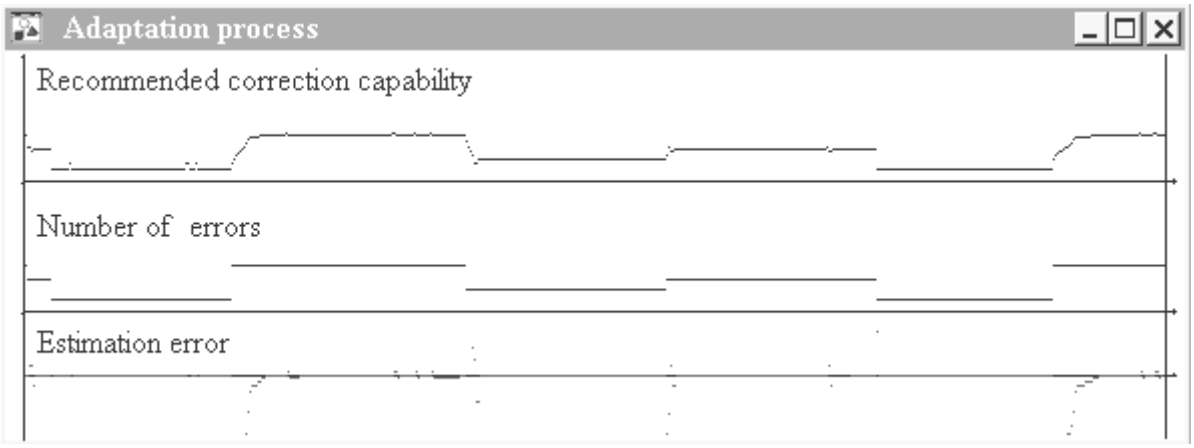

Figure 10. The behavior of the system after 30 minutes of transmission over different time varying channels

place in 10 ÷13 blocks for a worse state and 6 ÷7 blocks for a better state. The experiment continued with different other channels having 3 to 5 states. No major differences occurred in the behavior of the system. The long term tendency of the system is to have equal OE and UE (approx.

0.01) The shape of the output variable term set after 30 minutes of transmission (approx. 120000

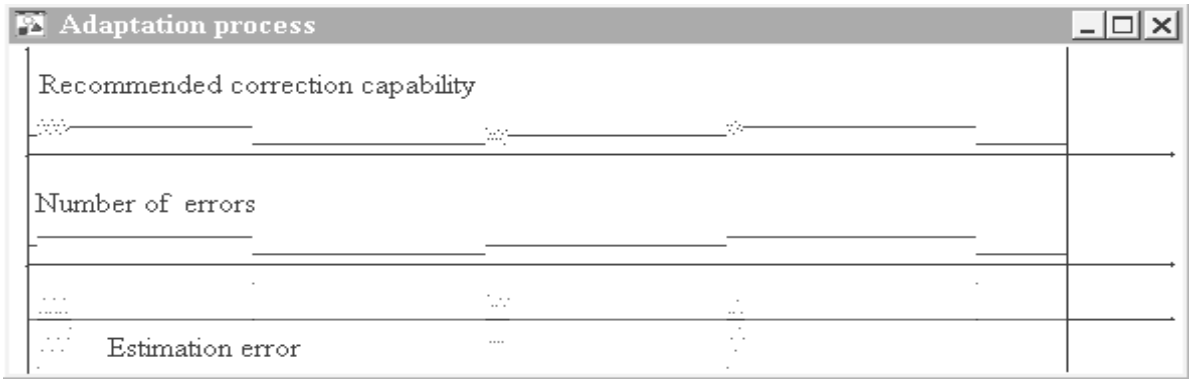

Figure 11. Adaptation evolution in experiment 3 after 30000 received blocks.

received blocks) is depicted in figure 9. The behavior of the system for a four-state channel (BER_1= 0.012, BER_2= 0.024, BER_3= 0.048, BER_4= 0.128) is presented in figure 10. The action of SAM is again visible.

Experiment 3

In order to decrease the estimation error (mainly UE) experiment 2 has been repeated using a different initial term set for the output variable. The left spread, modal value and the right spread of the fuzzy membership functions are listed below. A more dynamic adaptation to a new channel state is expected.

LN(24.0,-6.0,1.6,); SN(2.6,-3.0,2.99,); Z(0.1,0.0,0.1,); SP(2.99,3.0,2.6,); LP(1.6,6.0,24.0,)

After 25000 received blocks, UE reached 5×10^{-3} at an overestimation error of 5.2×10^{-3} (approximately half of the precedent value). Anyhow, after 20000 blocks, through the adaptation of the fuzzy sets, the system has the tendency to be less dynamic. The maximum adaptation duration is 10 blocks for a transition to a better state and 20 blocks for a transition to a worse state. This rather long interval can be explained by the fact that the dynamism of the system may cause oscillations during adaptation. These oscillations can be observed in figure 11 for a three-state channel having BER_1= 0.001, BER_2= 0.024, BER_3= 0.048.

After the movement of the fuzzy set *zero* to the right becomes significant, SAM's action becomes visible. It is benefic for the adaptation duration (after 100000 received blocks the maximum adaptation duration decreases from 20 to 10 blocks) but, on the other hand, it produces over/underestimation of the channel state when this is constant.

To avoid this disproportionate movement of the fuzzy set *zero* and, implicitly, the negative effect of SAM, it is necessary to have a more uniform distribution of the activation of the fuzzy

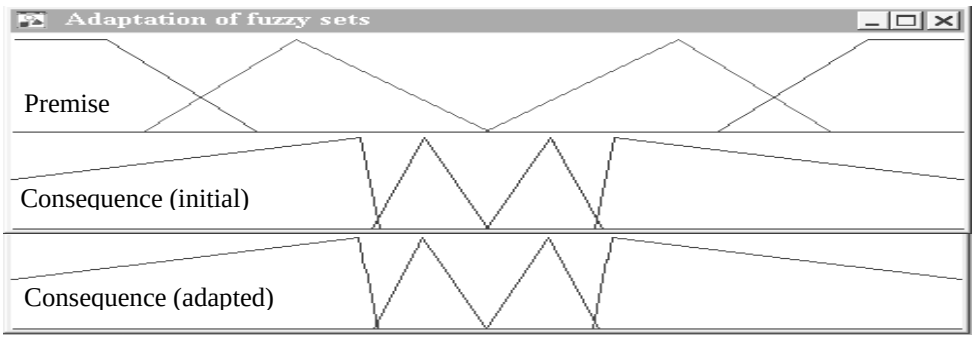

Figure 12. Initial term sets in experiment 4

Figure 13. Adapted fuzzy sets in experiment 4

rules. For this, we decided to eliminate the third rule and, implicitly, the fuzzy sets *medium* and *zero*. The previous experiments were repeated using this new knowledge base.

Experiment 4

The initial term sets for the experiment 4 are depicted in figure 12. The left spread, modal value and right spread of the membership functions are listed below.

LN(24.0,-4.0,0.6,); SN(1.6,-2.0,2.01,); SP(2.01,2.0,1.6,); LP(0.6,4.0,24.0,)

To avoid the oscillations during the adaptation process a distance of 2 was choose between the modal values of LP/SP and LN/SN, respectively.

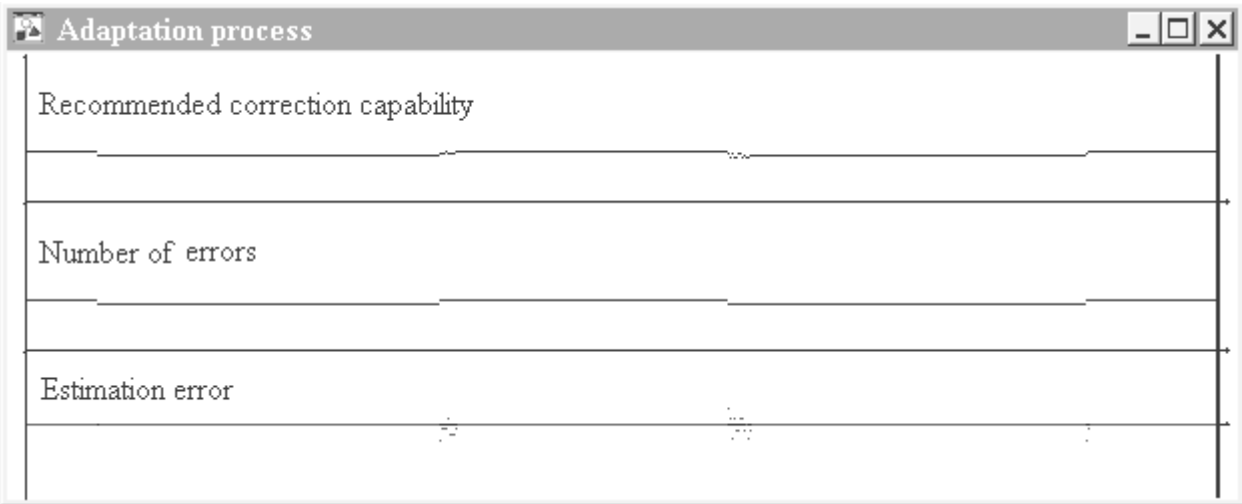

Figure 14. Adaptation evolution in experiment 4 after 35 000 received blocks

The system presents the best performances obtained so far. UE = 3.3×10^{-3}, OE= 3.0×10^{-3}. The adaptation duration is low, typically 5 blocks (which, depending upon the nature of the state may represent a single observation interval). The maximum adaptation duration is 10 blocks and does not depend significantly upon the type of transition (!). This behavior can be observed in figure 14. The adaptation of the fuzzy sets after 35000 received blocks is presented in figure 13. A slight

movement to the right is present, but the action of the stochastic modifier has more constructive effects (during the variation of the channel state) and less destructive effects (when the channel remains in a certain state).

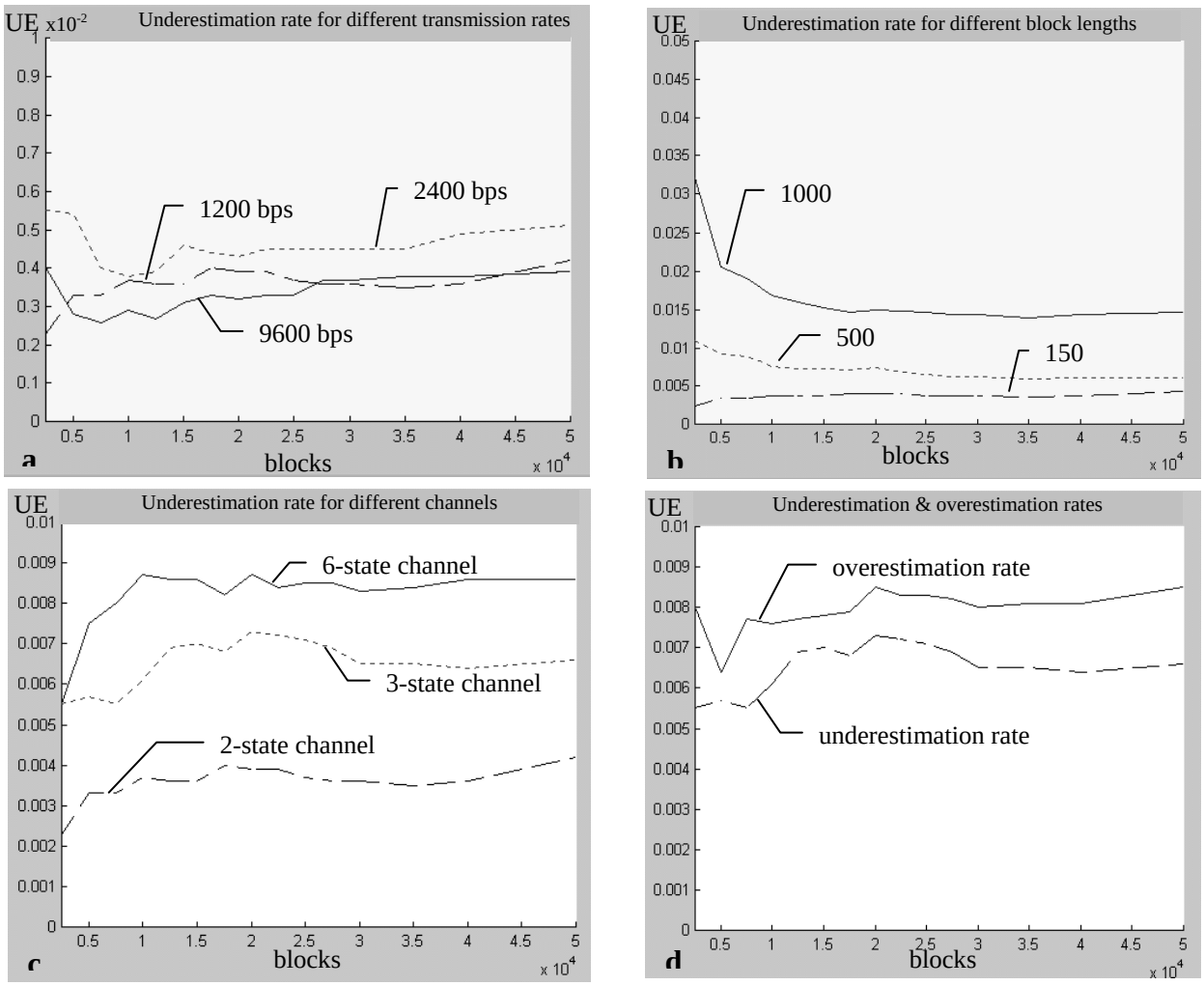

Figure 15. Comparative performances

5. CONCLUSIONS

Using this last system, further experiments have been carried on in order to study the dependency of the performances upon the data rate, the length of the blocks, the number and duration of the channel states, the quality of the channel states and so on. The main results are presented in figure 15. The dependence upon the transmission rate is low, UE and OE variation being less than 2×10^{-3} for data rates between 300 and 28800 bps. UE evolutions for 1200, 2400 and 9600 bps are depicted in figure 15.a. In figure 15.b the behavior for different block length is depicted. Considering UE equal to block error probability after the EC decoder, result the following BER's (after decoding) :

- 150 bits/block : P_b= 0,004, BER $\geq 2.5 \times 10^{-5}$
- 500 bits/block : P_b= 0,006, BER $\geq 1.2 \times 10^{-5}$

- 1000 bits/block: P_b= 0,015, BER ≥ 1.5×10^{-5}

It can be observed that the system performs better for longer blocks. Nevertheless, for longer blocks the adaptation duration is also longer, even in terms of blocks. For example, in case of a 6-state channel at 9600 bps, typical adaptation durations are 10 blocks for 151 bits/block, 9 ÷12 blocks for 500 bits/block and 15 ÷20 blocks for 1000 bits/block.

As expected, both underestimation and overestimation error are larger for frequent changes of the channel state. In figure 15.c is presented the UE evolution for channels varying over 2, 3 and 6 different states during the same amount of time. The experiments revealed that the mean value of UE per channel transition is virtually independent upon the number and the quality of the channel states but is dependent upon the difference between the quality of two successive states.

Other conclusions are that the performances still depend not only upon the choice for the fuzzy sets pertaining to the premises of the fuzzy rules, but also upon the initial choice for the term set of the output variable. Further, large overlapping of the fuzzy sets lead to a strong stochastic behavior of the system, which is not always benefic for the overall performances. The system has the tendency to stabilize on a value of OE slightly higher than UE, but these values are always comparable (see figure 15.d).

Due to its adaptation capability, the system proved a large degree of flexibility and, very important, performances comparable or even better than the ones obtained by controllers that rely on specific models of the channels [12]. These results are encouraging and future work will be devoted to the improvement of the system, mainly in respect of GARIC implementation. Some changes related to SAM's action has to be applied and a refinement of the external reinforcement will be taken into consideration.

REFERENCES

[1] S. -I. Amari, A. Cichocki, "Adaptive Blind Signal Processing – Neural Network Approaches", *Proc. of the IEEE,* vol. 86, no.10, oct. 1998, pp. 2026-2048.

[2] H.R. Berenji, "An architecture for designing fuzzy controllers using neural networks", *Int. J. of Approximate Reasoning*, vol.6, nr.2, 1992, pp. 267-292.

[3] H.R. Berenji, P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcement" *IEEE Trans. on Neural Networks*, vol.3, nr.5, 1992, pp.724-740.

[4] G.B. Giannakis, C. Tepedelenlioglu, "Basis Expansion Models and Diversity Techniques for Blind Identification and Equalization of Time-Varying Channels", *Proc. of the IEEE,* vol. 86, no.10, oct. 1998, pp. 1969- 1986.

[5] S. Haykin, *Neural networks - a comprehensive foundation*, Macmillan, 1995.

[6] S. Khalel, "Hybrid ARQ Protocols with Adaptive Forward Error Correction", *IEEE Trans. on Comm.*, Feb. 1994.

[7] Bart Kosko, *Neural networks and fuzzy systems: A Dynamically Approach to Machine Intelligence*, Prentice Hall, New Jersey, 1991.

[8] W.C.Y. Lee, *Mobile Cellular Telecommunications. Analog and Digital Systems,* McGraw-Hill, 1995.

[9] A.Miller, B. Vucetic, *Satellite Communication. Mobile and Fixed Services,* Kluwer, 1993.

[10] T.S. Rappaport, Wireless Communications, Prentice Hall, 1996.

[11] L.Tong, S. Perreau, “Multichannel Blind Identification: From Subspace to Maximum Likelihood Methods”, *Proc. of the IEEE,* vol. 86, no.10, oct. 1998, pp. 1951- 1968.

[12] B. Vucetic, "An adaptive coding scheme for time-varying channels", IEEE Trans. on Communications, vol. 39, no. 5, 1991, pp. 653-662.

[13] G. Wade, *Signal Coding and Processing*, Cambridge University Press, 1994.

[14] S. Zahan, "Neural Decoder for Error Correcting Block Codes", *Proc. of EUFIT'95*, Aachen, 1995, pp. 401-405.

[15] S. Zahan, “Adaptive error control system for time varying channels”, Proc. of. El. and Telecomm. Symp, Timisoara, 1996, pp. 46-51.

[16] S.Zahan, “Reinforcement Learning Applied to Adaptive Coding of Time Varying Transmission Channels”, *Proc. of NEUREL’97*, Belgrade, 1997, pp. 223- 227.