COMMUNICATIONS

IMPLEMENTATION OF THE VITERBI ALGORITHM USING FUZZY ARITHMETIC

Nikos V. Kokkalis — Christos S. Koukourlis *

In this paper the incorporation of fuzzy arithmetic into Viterbi Algorithm is exploited. Especially the length estimation of trellis paths is based on fuzzy arithmetic. Taking advantage of the newly developed fuzzy processors, this paper aims to show the applicability of fuzzy arithmetic into Viterbi Algorithm by simulating the whole procedure in Borland Delphi. The algorithm exploited a good conformity with the expected results when compared to the optimum (conventional) implementation. We show that it is possible to incorporate fuzzy arithmetic into Viterbi Algorithm, obtaining similar results regarding BER *vs* E_b/N_0.

K e y w o r d s: Viterbi algorithm, fuzzy arithmetic, fuzzy logic, V.32, TCM, fuzzy processor

1 INTRODUCTION

Viterbi Algorithm (VA) is a powerful algorithm used in many standards of PSTN modem implementations, starting back from standard V.32, which is usually implemented in DSP processors. Viterbi Algorithm, when used in convolutional decoders, uses the so-called soft decision. In other words, the receiver does not decide which bits (or symbol) have been received, but it postpones the decision until other supplementary data have been received. The final decision takes place after some symbol intervals, when two paths of the trellis diagram merge on one node of the trellis. This loosening of decision brings in mind the fuzziness concept of fuzzy arithmetic. On the other hand, during the last years fuzzy processors are being developed that will facilitate the implementation of fuzzy arithmetic on these devices.

In this paper we exploit the application of fuzzy arithmetic into the Viterbi Algorithm by simulating the whole procedure in Borland Delphi. The simulation took place in a usual processor (computer), not dedicated for real time applications. Although the implementation took place using such an ordinary processor, which does not accommodate fuzzy operations, and therefore it lacks in efficiency and speed, our study gave the result that it is worth to implement Viterbi Algorithm incorporating fuzzy arithmetic from the point of view that the BER *vs* E_b/N_0 is maintained as soon as a fuzzy processor is involved.

2 FUZZY PROCESSORS

As mentioned previously, the Viterbi Algorithm is usually implemented in DSP processors [1]. Also, chips like the TMS320C54V90 from Texas Instruments implement full-featured modem technology intended for use in several modulation standards. Nowadays, powerful microcontrollers incorporating fuzzy arithmetic are developed. One example is the ST's 8-bit DuaLogic^{TM} microcontroller family which incorporates a dedicated fuzzy processor, an ALU for Boolean operation and a set of peripheral functions [2]. Also, several RISC approaches can be found in the literature [3, 4] in high-speed applications or for the processing of large amount of data as in image analysis where computation time becomes a critical issue. Their architecture consists of a RISC as a core processor with special hardware functional units for fuzzy related operations either integrated or placed as a co-processor. They can be considered as microprocessors tailored for information processing based on fuzzy logic and fuzzy set theory, equipped with application specific instruction set primitives optimized for fuzzy processing. These devices are able to perform in an efficient way both Boolean and fuzzy algorithms in order to reach the best performance that the two methodologies allow. They belong to the so-called "third generation fuzzy processors" currently appeared in the market. A commercial example is the Motorola 68HC12, an upgrade of the well-known 68HC11 microcontroller [5]. The dominant characteristic is a general purpose processor with fuzzy logic support, either in the form of on-board coprocessors or enhanced general-purpose instruction sets to accommodate fuzzy logic operations. A similar case is the appearance of new devices called Fuzzy Digital Signal Processors (FDSPs) where the fuzzy part acts as a controller or an adaptive element to a DSP system in order to implement adaptive filters, noise cancellers, channel equalizers, *etc* [6].

The aim of this paper is to propose or predict the usefulness of the combination of the two methodologies by showing that fuzzy arithmetic can be introduced in processing of power- demanding applications like the Viterbi Algorithm. The idea is in the direction of the trend to de-

* Democritus University of Thrace, Electrical and Computer Engineering Dept., Telecommunications Systems Laboratory, GR-671 00, Xanthi, Greece, E-mail: ckoukou@ee.duth.gr

ISSN 1335-3632

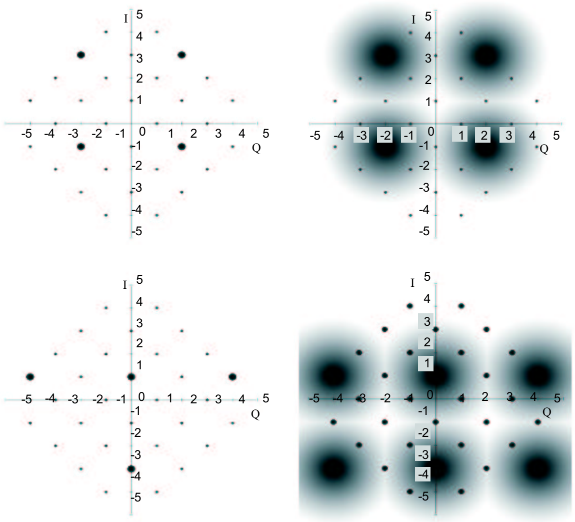

Fig. 1. Fuzzy sets for the two kinds of point-groups for the V.32 standard.

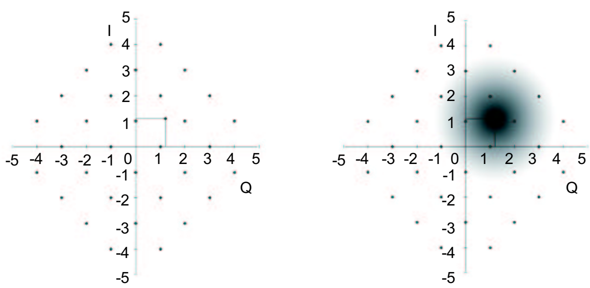

Fig. 2. Fuzzy set of a received input symbol.

velop special VLSI architectures that serve specific purposes [7].

3 TRELLIS CODED MODULATION AND VITERBI ALGORITHM

Trellis Coded Modulation (TCM) has been evolved during the last two decades as a combination of coding and modulation for the transmission of data through bandlimited channels. Its main advantage is that this combination allows significant coding gain compared to the concatenated coding and modulation occupying the same bandwidth. TCM is an amplitude/phase modulation combined with a convolutional encoder. At the receiver, the signal affected by noise is demodulated and decoded using the VA, which is based on a soft decision process. A simple four- state TCM modulation can give an equivalent of 3 dB improvement in the required S/N, without any additional bandwidth demand or reduction of the transmission rate.

The VA [8-10] has been used in almost all modem standards after the V.32. This standard was developed to transmit data through the public switched telephone network (PSTN). The Viterbi decoder as applied in TCM uses the so-called Euclidean distance in order to rank the conformity of a received symbol to one of the available points of the constellation diagram. As previously stated, the receiver does not take a hard decision, but in one sense it considers as possible all the neighbouring points of the diagram. A usual processor or DSP calculates this distance by making some mathematical operations and stores the results in the memory until a final decision is taken. The contribution of this paper is to show that this measure of conformity can be judged by using fuzzy arithmetic.

4 FUZZY ARITHMETIC

In the past, several alternative sub-optimal methods of convolutional decoding [11] have been developed, in order to reduce the complexity of VA. Moving in this direction, we examine the same problem from another point of view, *ie*, by combining different techniques of processing. The application of fuzzy arithmetic is to the extent that the computation of Euclidean distance is based on the intersection — a fuzzy operation — of fuzzy sets. The soft decision, which is used in the algorithm, lends itself to the implementation by fuzzy arithmetic due to its inherent fuzziness. In this paper we will consider the complicated case that the VA is applied in conjunction to QAM modulation, which results in the known as Trellis Coded Modulation (TCM) technique. For simplification we follow the 32-point constellation diagram of the V.32 standard, which is divided into eight groups of four points each. The VA aims to find which of the eight groups is the most likely that the input signal belongs. Next, by hard decision, it decides which is the correct point (one out of four of the group). That means that the decoder is concerned to know, by the use of a metric, the distance of the received point from each one of the eight groups. The metric that we use is the intersection of triangles that we define as membership functions.

5 INPUT AND CONSTELLATION DIAGRAM FUZZIFICATION — MEMBERSHIP FUNCTIONS

In order to estimate the metric we represent the constellation points as cones with their peaks at the exact points of the diagram and a base extended around them (Fig. 1). Actually, two of the eight groups are shown (one square-shaped and one "T"-shaped), consisting of four points each, and also the corresponding two groups of cones, in two-dimensional grey-scale representation. The bottom group, which originally is "T" shaped, seems to have two more cones at the bottom right and left sides, which come from the way that it is implemented in the software, but this does not harm the operation.

The position of the received symbol on the constellation diagram is represented as a fuzzy set, *ie* as another cone with its peak at the point the receiver considers that the received symbol lies. As sated previously, the receiver does not decide which the received symbol is. Instead,

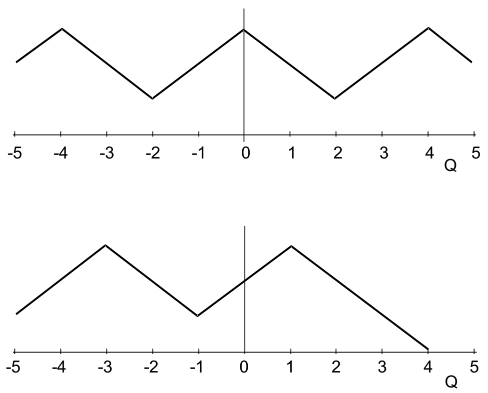

Fig. 3. Decomposition (projection) of the two dimensional fuzzy set on the two axes, I and Q. Symbol μ (vertical axis) is the degree of membership on I and Q axes.

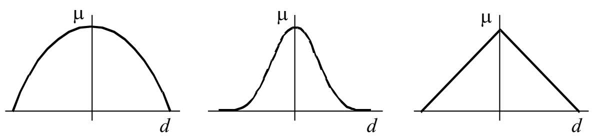

Fig. 4. Alternative membership functions: $y(x) = (1 - x^2)$, Gaussian, triangular.

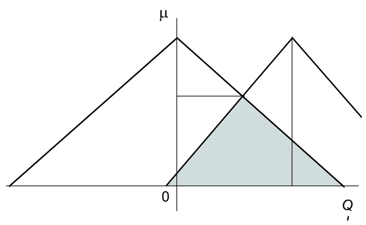

Fig. 5. Intersection of triangles of input point (signal) and constellation point membership functions

it considers the (Euclidean) distance from all eight constellation points, one from each of the eight groups, and inserts this value into the algorithm. This is the so-called soft decision. In our case, the metric that is to be inserted in the algorithm is no more the Euclidean distance but the intersection of the signal cone with the cones of the neighbouring constellation points.

One step further, and in order to simplify things and to avoid the computation of two-dimensional intersections, we represent each point (constellation or input) with a pair of one-dimensional fuzzy sets. Actually, this representation is the projection of the two-dimensional membership function on each axis of the constellation diagram. The pairs of these fuzzy sets are shown in Fig. 3.

Figure 3 shows the decomposed two-dimensional fuzzy set, corresponding to the right hand set of Fig. 1. The degree of membership, μ, is shown on the vertical axes, which is actually the projection of the cones on each direction.

Additionally, other possible membership functions (fuzzy sets) have been considered. These are the $y(x) = (1-x^2)$ and the Gaussian curve, shown in Fig. 4. It seems that the most appropriate is the $y(x) = (1-x^2)$, because it is directly analog to the logarithm of the probability density function. But the use of the conic (or pyramid) function, which gives a triangular projection on the axes, simplifies the calculation without significant loss of precision.

6 OPERATIONS AMONG FUZZY SETS – METRIC CALCULATION

In both cases, the sets that can result from the operations can sometimes be expressed with sets of the same shape. We have to use sets defined in R^2. To avoid this, when we want to find the degree of membership of an input point, we find the degree of membership in each of the two sets corresponding to each axis (I and Q), by applying the operation of intersection separately. One step beyond is to calculate the distance from each of the eight points by using the intersection operation.

Different metrics have been adopted for two distinguished cases. For the case of triangular membership function the following metric is used:

$$\mathit{length} = \min\big(\sup \mu_I(I), \sup \mu_Q(Q)\big)$$

where $\mu_I(I)$ and $\mu_Q(Q)$ are the membership functions of the intersection of the input triangle and constellation point triangle that correspond to each axis. The function sup denotes the maximum value (supreme) of the corresponding function.

The intersection is the shaded triangle shown in Fig. 5.

For the case that the membership function of the constellation points is $y(x) = (1 - x^2)$, the metric is:

$$\mathit{length} = \sup \mu_I(I) + \sup \mu_Q(Q)\,.$$

At this point we must denote that the criterion of the VA is reversed, as the algorithm has to find the longest path in the trellis diagram instead of the shortest. This is because the degree of membership is larger when the points are closer, in contrast to the Euclidean distance (used in the normal VA), which is smaller.

7 SIMULATION PROCEDURE AND RESULTS

The system was simulated in baseband. This implies that the clock recovery and the modulation/demodulation procedures are considered to be ideal and thus not affecting the system performance. Since our objective is to compare only the Viterbi decoding schemes mentioned in the previous sections, which do not depend on these stages, this approach is sufficient.

Specifically, the simulation is implemented as follows. A random symbol is generated by a data source with statistically uniform distribution. This is encoded convolutionally for TCM or sent to the next stage unmodified *ie* in the case of QAM. Then it is mapped on the I–Q coordinates of the appropriate constellation diagram.

Each of the two coordinates of the signal is added with a Gaussian random variable giving a version of the signal

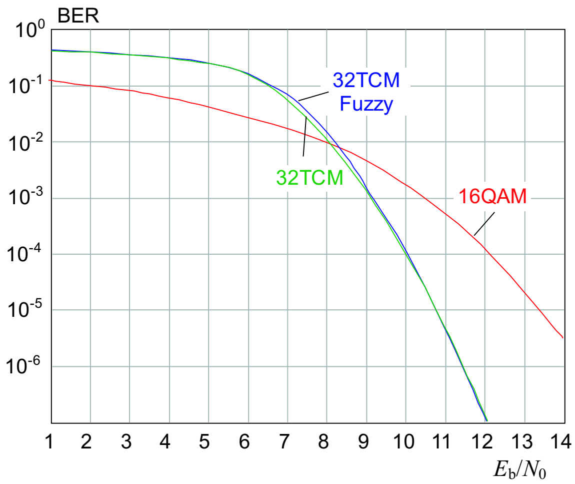

Fig. 6. Bit Error Rate (BER) *vs* E_b/N_0 of 32TCM with fuzzy metric $y(x) = (1 - x^2)$ membership function.

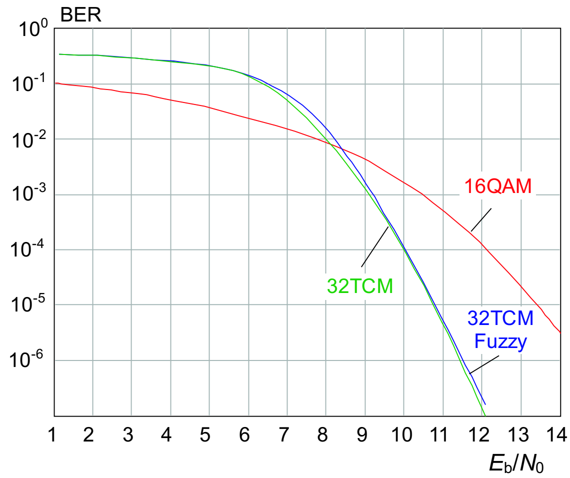

Fig. 7. BER of 32TCM *vs* E_b/N_0 with fuzzy metric — triangle membership function.

distorted by AWGN. The Gaussian random number generator is implemented with the Box-Muller [12] method and its variance depends on the desired E_b/N_0. These coordinates are then fed to the combined demodulator and Viterbi decoder for TCM or, in the case of QAM, they are translated to the closest symbol. The output symbol is compared to the input symbol and the number of different bits is measured. After a sufficiently large number of iterations, in the order of hundreds of millions, the BER is calculated.

It has been found that the application of the Gaussian membership function lacks in precision when compared to the other two functions examined here, and will not be considered further. When using the $y(x) = (1 - x^2)$ function, we get the result shown in Fig. 6.

The simulation results of Fig. 6 show that there is a very good coincidence among standard 32TCM and 32TCM-fuzzy. Also in Fig. 6, the corresponding BER for the simple 16QAM case is shown for comparison purposes.

When the triangular membership function is used, a slight deterioration can be observed. The corresponding simulation results are shown in Fig. 7. This figure shows a deterioration of about 0.2 dB at very high E_b/N_0.

From the above discussion we conclude that the adoption of the triangular membership function is preferable due to its simplicity, compared to the $y(x) = (1 - x^2)$ function, giving only a slight deterioration of the performance.

8 CONCLUSIONS

In this paper the Viterbi Algorithm has been simulated using fuzzy arithmetic to calculate the metric and rank the coincidence of received constellation points. Although our approach is generic, for simulation purposes the V.32 standard has been adopted. The results are compared to the optimum implementation and show that the use of fuzzy arithmetic greatly preserves the BER *vs* E_b/N_0 of the optimum implementation. So, the development of special VLSI architectures incorporating fuzzy processors for power-demanding applications like the Viterbi Algorithm is encouraged.

References

[1] MANSOOR, C.: Viterbi Implementation on the TMS320C5x for V.32 Modems, Digital Signal Processing Applications — Semiconductor Group, Texas Instruments Incorporated (1996).

[2] ST Microelectronics (1998) ST52x301, 8-bit OTP/EPROM DuaLogic™ MCUs with ADC, UART, timer, TRIAC and PWM driver. Advanced data sheet.

[3] HIROYUKI WATANABE: RISC Approach to Design of Fuzzy Processor Architecture, IEEE International Conference on Fuzzy Systems (Cat. No. 92CH3073-4), San Diego, CA, USA, 8–12 March, (1992), 431–441.

[4] VALENTINA SALAPURA: A Fuzzy RISC Processor, IEEE Transactions on Fuzzy Systems **8** No. 6 (2000), 781–790.

[5] VIOT, G.: Third Generation Fuzzy Processors, IEEE Technical Applications conference, Northcon/96, Conference Record, Seattle, WA, USA, (1996), 203–209.

[6] LABIB SULTAN: FDSP: A VLSI Core for Adaptive Fuzzy and Digital Signal Processing Applications, ISCAS'98 Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, Monterey, CA, USA, vol. 3, 123–126.

[7] ASCIA, G.—CATANIA, V.—RUSSO, M.: VLSI Hardware Architecture for Complex Fuzzy Systems, IEEE Transactions on Fuzzy Systems **7** No. 5 (1999), 553–570.

[8] FORNEY, G. D.: The Viterbi Algorithm, Proceedings of the IEEE **61** No. 3 (1973), 268–278.

[9] UNGERBOECK, G.: Channel Coding with Multilevel/Phase Signals, IEEE Trans. Information Theory **IT-28** (1982), 55–67.

[10] UNGERBOECK, G.: Trellis-Coded Modulation with Redundant Signal Sets, Part I: Introduction, IEEE Communications Magazine **25** No. 2 (1987), 5–11.

[11] PROAKIS, J. G.: Digital Communications, 3rd edition, McGraw Hill, New York, 1995.

[12] BOX, G. E. P.—MULLER, M. E.: A Note on the Generation of Random Normal Deviates, Ann. Math. Stat. **29** (1958), 610–611.

Received 2 May 2004

Nikos V. Kokkalis and **Christos S. Koukourlis**. Biographies not supplied.