Artificial Neural Network for DTMF decoders

Andrea Aiello [1], Pasquale Daponte [2], Domenico Grimaldi [1], Linus Michaeli [3]

[1] Dip. di Elettronica, Informatica e Sistemistica, Università della Calabria, 87036 Rende (CS), Italy. Ph.: ++39-0984-494712, Fax: ++39-0984-494713, E-mail: grimaldi@deis.unical.it.

[2] Facolta di Ingegneria, Università del Sannio, Piazza Roma, 82100 Benevento, Italy. Ph. +39-0824-305817, Fax: +39-0824-21866, E-mail: daponte@unina.it.

[3] Dept. of Electronics and Multimedial Telecommunications, University of Kosice, Letna 9/A, 04120 Kosice, Slovak Republic. E-mail: Linus.Michaeli@tuke.sk.

Abstract

In this paper, the pattern recognition characteristics of the Artificial Neural Networks (ANNs) are used to realise a real decoder for Dual Tone Multi Frequency signals used in the telecommunication field. A new neural architecture, the Multi Learning Vector Quantization (MLVQ) network, is proposed. It offers both greater efficiency in decoding and less sensitivity to noise. In order to solve the problem regarding input signal synchronisation, a pre-processing phase is organised. Respect of the timing parameters required by the international recommendations is assured by implementing a Finite State Machine (FSM). The prototype decoder has been realised by implementing the pre-processing phase, the MLVQ network and the FSM on the TMS320C30 Digital Signal Processor. The decoder has been tested according to the ITU-T Q.24 and Telcordia Recommendations by means of a PC-based automatic measurement station. The test results are given and compared with those obtained by a traditional decoder and by a decoder based on the Multi-layer Perceptron ANN.

Keywords: Artificial Neural Networks, DTMF, Telecommunication, Measurement.

1: Introduction

An increasing number of studies have been carried out on the Artificial Neural Networks (ANNs) in different areas of research. In some problems, they are more efficient than the conventional algorithms and represent an interesting tool for advanced research and applications. In particular, the ANNs promise to be more efficient at recognising and distinguishing complex vectors according to their ability to generalise and form some internal representations of the supplied input signal. These abilities make them very useful for the Dual Tone Multi Frequency (DTMF) signal decoding.

The DTMF signals are commonly used in touch-tone dialling applications [1], home automation via a personal computer [2], interactive banking and reservation systems. They correspond to one of twelve touch-tone digits (0-9, *, #) of the telephone keypad and are the sum of a low frequency tone (typically 697 Hz, 770 Hz, 852 Hz and 941 Hz) and a high frequency tone (typically 1209 Hz, 1336 Hz and 1477 Hz). All the DTMF frequencies have been carefully chosen in order to avoid problems with harmonics and distortion.

DTMF signal decoding is very difficult in real situations. Difficulties arise by a consequence of (i) noise presence, and (ii) frequency and amplitude errors in the generation of the two tones constituting the signals [3, 4]. Others difficulties arise from the international recommendations [5, 6] that establish the parameter values to be respected by real DTMF decoders. These parameters are very restrictive and concern both (i) the signal characteristics, and (ii) the decoding time interval.

Several ANNs could be utilised for DTMF signal decoders. The Learning Vector Quantization (LVQ) network [7] shows the greatest benefits with regard to the characteristics of simplicity, model free classification, and performance.

Nevertheless, the original LVQ network does not permit the realisation of a decoder, which completely satisfies the requirements of the international recommendations.

This paper proposes a solution based on the new Multi Learning Vector Quantization (MLVQ) network. This architecture is characterised by several LVQ networks working in parallel, each one classifying the sampled signal synchronised at a different time instant.

The prototype of the neural-based DTMF decoder has been realised by implementing the MLVQ network on a Digital Signal Processor (DSP). This prototype permits testing of the decoder's performance according to the international recommendations. Respect for the timing parameters is assured by organising and implementing a Finite State Machine (FSM).

All the testing procedures require subjecting the decoder to several DTMF signals with specific characteristics and, consequently, are both complex and very time consuming.

In order to speed up the testing procedures, a computer-based Automatic Measurement Station (AMS) has been designed and realised.

Finally, the results of the tests performed by the AMS are given and discussed. These results are also compared with those obtained by a decoder based on the Multi-layer Perceptron ANN [8, 9] and by a traditional decoder based on a modified Discrete Fourier Transform (DFT) [10]. All the decoders tested are implemented on the same DSP board.

2: The neural DTMF decoder

The block diagram of the neural DTMF decoder is depicted in fig.1.

Before applying the signal to the MLVQ network, the *pre-processing* phase occurs. In the *normalising* block, the sampled DTMF signal is normalised.

Successively, in the *synchronising* block, the synchronising value among the samples is detected in order to synchronise the input signal for the successive classification. Finally, in the *vector organising* block, the appropriate input vector is applied to the MLVQ network.

The *MLVQ* block performs the signal classification. Each block is designed according to the MLVQ network's working characteristics.

In what follows, therefore, the MLVQ network is examined before the pre-processing block.

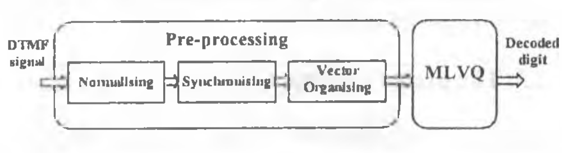

Fig. 1 The Block diagram of the neural decoder.

2.1: The LVQ and the MLVQ network

The MLVQ network is a particular neural structure based on the original LVQ network proposed by T. Kohnen [7]. This last has been modified in order to realise a neural structure able to decode the DTMF signals in the real situations according to the requirements of the international recommendations.

Given N classes of d-dimensional vectors and a vector $x \in \Re^d$, the LVQ neural network classifies x by individualising the class to which it belongs. For this purpose, the ANN needs a training phase, in which a reference vector, called the codebook, is determined for each class. Next, the LVQ network classifies a vector individualising the codebook, which best matches it. The most commonly used matching function (or metric of similarity) is the Euclidean distance defined as:

$$d_e(x,y)=\sum_{j=1}^{n}(x_j-y_j)^2 \qquad (1)$$

The matching function used in this paper is, however, the normalised dot product defined as:

$$d_x(x,y)=\sum_{j=1}^{n}(x_j \cdot y_j) \qquad (2)$$

where the vectors x and y are previously normalised. Experimental tests have shown that the dot product is more accurate than the Euclidean distance in the case under examination.

The input layer of the LVQ network (fig.2) is connected directly to the hidden layer without weighting the signal. The neurons of the hidden layer calculate the matching function between the input vector and their associated codebook according to (2). Each codebook corresponds to a digit (0-9, *, #).

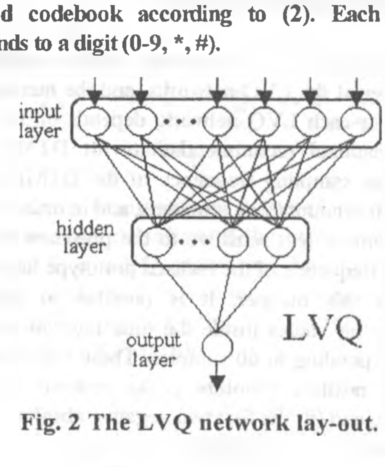

Fig. 2 The LVQ network lay-out.

The output layer neuron calculates the winning matching function as being the largest amongst the output of each hidden layer neuron in order to determine the detected dual-tone. In order to correctly classify the DTMF signals, the LVQ network needs the input vector to be opportunely synchronised. The input vector is applied to the LVQ network in such a way that the synchronising value is the input to a pre-fixed neuron, chosen as reference one.

The absolute peak value is used as the synchronising value. Problems arise from the presence of noise, which causes errors in decoding the absolute peak and consequently an incorrect synchronisation. One solution is to consider more than one synchronising value. In this manner, the DTMF signal is classified by matching the codebooks with the real signal in the neighbour of each synchronising value. Several LVQ networks working in parallel are used. Each LVQ network evaluates the winning matching function (2) for the input vector corresponding to each synchronising value. At the end, the valid decoded DTMF signal is established selecting the largest of the winning matching functions.

The realised decoder is organised by employing the Multi Learning Vector Quantization (MLVQ) network. The MLVQ network consists of a neural structure in which more than one identical LVQ network can be identified. Fig. 3 shows the block diagram of the used MLVQ constituted by four identical LVQ networks working in parallel in order to classify the input vector. The output of each LVQ is the couple constituted by the number associated with the decoded DTMF signal and the value of the winning matching function. The *Selector* block evaluates the largest of the values of the winning matching function belonging to the four LVQ networks. Only if this value exceeds a fixed threshold, is the corresponding DTMF signal assumed to be valid.

2.2: MLVQ network design

The number of synchronising values, corresponding to the number of the LVQ networks, and the number of input neurons for each LVQ network, depends on (i) satisfying the international recommendations for DTMF decoders, and (ii) the sampling frequency of the DTMF signal. By taking into account these problems, and in order to make the decoder robust, less sensitive to the presence of noise, the sampling frequency of the realised prototype has been set to 8kHz. In this manner, it is possible to choose four synchronising values inside the time interval equal to 7.5 ms, corresponding to 60 samples. These values are: (i) the first two positive absolute peaks ordered according to amplitude, and (ii) the first two negative absolute peaks also ordered according to amplitude.

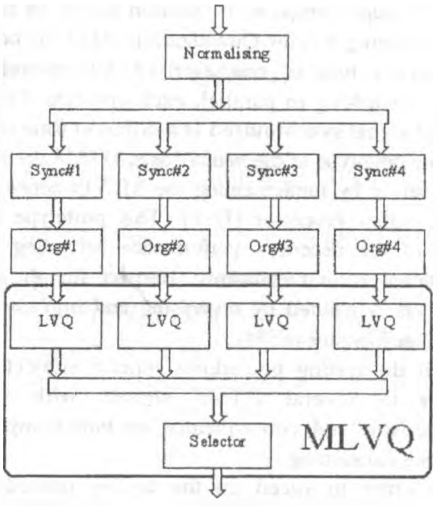

Fig. 3 The flow scheme of the DTMF decoder realised employing the MLVQ.

Several experiments have shown that the MLVQ network with 90 samples for each LVQ network is able (i) to decode the signal in satisfying the international recommendations, and (ii) to minimise the memory required once implemented on the DSP.

The LVQ network is trained in such a way that the input vector includes some elements both preceding and following the synchronising value. The optimal choice, according to the working characteristics of the LVQ network, is to set the reference neuron in the 60th position of the input layer; this neuron's input being the synchronising value. The sampled vector, therefore, is organised, in the *organising* block of fig. 3, so that

- the synchronising value is sent to the 60th neuron;
- the samples before the synchronising value are sent to the neurons preceding the 60th;
- the samples after the synchronising value are sent to the neurons following the 60th.

With this organisation of the input vector, the number of neurons of the input layer is set according to the worst case:

1. with synchronising value in the 60th position, the input vector is sent to the neurons from 1 to 90;
2. with synchronising value in first position, the input vector is sent to the neurons from 60 to 149.

In conclusion, the LVQ network with 149 input layer neurons must be used (fig. 4).

2.3: The pre-processing phase

Once the characteristics of the MLVQ network are established, each block of the pre-processing phase can be designed.

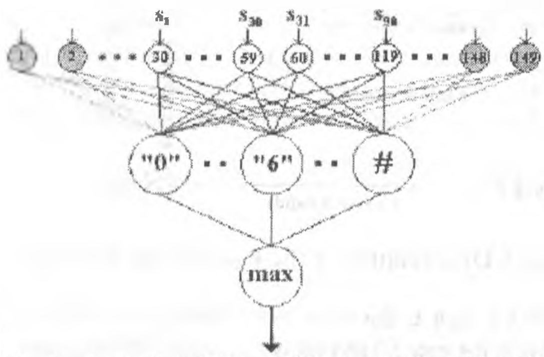

Fig. 4 Layout of the LVQ network with 149 input neurons, 12 hidden neurons, and 1 output neuron. The 60th neuron is the reference neuron.

In the *normalising* block, the signal is sampled and the Euclidean norm of the input vector is evaluated. The normalised vector is then calculated. The output of the *normalising* block is, therefore, the normalised vector of the 90 elements sampled at 8kHz.

The four *synchronising* blocks operate in parallel and in a similar way. The only difference is the synchronising value assumed in each block. In particular: (i) *Synch#1* refers to the absolute maximum peak value in the first 60 elements of the input vector (Fig.5a), (ii) *Synch#2* to the second maximum peak ordered according to amplitude in the first 60 elements of the input vector (Fig.5b), (iii) *Synch#3* to the absolute negative peak value in the first 60 elements of the input vector (Fig.5c), and (iv) *Synch#4* to the second negative peak ordered according to amplitude in the first 60 elements of the input vector (Fig.5d).

The four *organising* blocks operate in parallel. In particular, (i) *Org#1* provides the corresponding LVQ network with the input vector in which the positive absolute peak value is the input of the reference neuron (Fig.5a), (ii) *Org#2* provides the corresponding LVQ network with the input vector in which the second positive peak value ordered according to amplitude is the input of the reference neuron (Fig.5b), (iii) *Org#3* provides the corresponding LVQ network with the input vector in which the negative absolute peak value is the input of the reference neuron (Fig.5c), and (iv) *Org#4* provides the corresponding LVQ network with the input vector in which the second negative peak value ordered according to amplitude is the input of the reference neuron (Fig.5d). *Org#3 and #4* operate the sign inversion of the samples. In this manner, because the DTMF is a symmetrical signal, the need to train the LVQ#3 and #4 with different training sets is avoided.

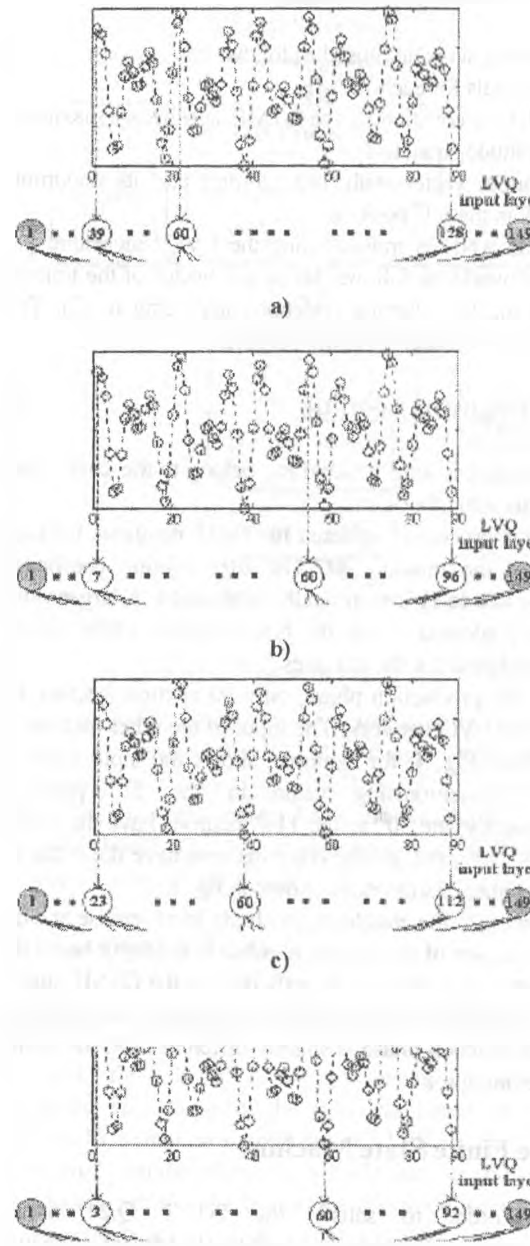

Fig. 5 Synchronising values detected in the first 60 elements and organisation of the input vectors to be sent to the four LVQ networks. The synchronising value, sent to the 60th reference neuron, corresponds to a) the 22nd sample, b) the 54th sample, c) the 38th sample, d) the 58th sample.

3: LVQ network training

The training set is organised as follows:

1. 10 signals for each DTMF;
2. each tone constituting the DTMF signal has a maximum amplitude equal to 1;
3. an input vector with 149 samples and its maximum peak in the 60th position.

The ANN is trained using the LVQ2 algorithm [7]. This last works as follows: let be x a vector of the training set and m_w the winning codebook according to (2). This codebook is updated by using the rule:

$$m_w(t+1)=m_w(t)\pm\alpha(t)[x-m_w(t)] \qquad (3)$$

The plus sign is used if x and m_w belong to the same class, the minus sign otherwise.

This process is repeated for 20-25 iterations for each vector of the training set. The $\alpha(t)$ training coefficient (ranging in 0 to 1) is empirically determined. A larger value is usually adopted during the first iterations, while a small one is preferred for the last ones.

In the production phase, only 90 element vectors are sent to the LVQ network. The input of the other neurons is set to zero. Fig. 4, for example, shows the input vector s with a synchronising value in the 31st position. Consequently, the 30th to the 119th neurons have the vector element s_j as input; all the other neurons have the input set to zero. Other examples are shown in fig. 5.

Owing to the architecture's ability to generalise and the optimal choice of the neuron number in the input layer, the LVQ network is able to correctly decode the DTMF signal, opportunely synchronised, even if the input vector utilised in the production phase is slightly different from the vector of the training set.

4: The Finite State Machine

In order to satisfy the ITU-T Q.24 timing specifications, a Finite State Machine (FSM) (fig. 6), which supervises the decoding process, is implemented. In order to guarantee the minimum accepted dual-tone length, the input signal is assumed to have been decoded if in two successive time decoding windows the result of the MLVQ network is the same. Moreover, the FSM allows avoidance of erroneous double-registration of a signal if reception is interrupted by a short break in transmission.

The FSM is controlled by two Boolean variables, *valid and same*, computed in each time decoding window. *Valid* is true if a valid DTMF is detected, while *same* is true if the

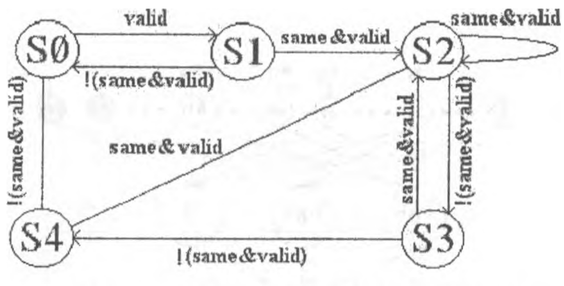

Fig. 6 Organisation of the Finite State Machine.

decoded digit is the same as the previous one. The FSM starts in the state S0 and waits for a valid DTMF signal.

If all the ITU-T Q.24 timing specifications are satisfied, the dual-tone is completely decoded in transition to state S2. From the state S2, the FSM awaits a pause signal before returning to state S0.

5: DTMF decoder implementation on the DSP

Once trained, the pre-processing phase and the MLVQ network are implemented on the DSP TMS320C30 by Texas Instruments [11]. This DSP is a 32-bit floating-point processor, which can execute operations at a performance rate of 33.3 MFLOPS (millions of floating point instructions per second) and 16.7 MIPS (millions of instructions per second). It is installed on an Evaluation Module Board (EVM) equipped with the additional chip TLC32044 used for the sampling section. The maximum sampling frequency of the ADC is 19.2 kHz.

The codebooks are recorded on the DSP memory into an array. The use of the array instead of the matrix makes processing more efficient because only one access is needed instead of the two accesses required by a matrix. The required data memory is 1788 words. In order to efficiently implement the decoding process, a pipeline is realised on the DSP between signal sampling and processing. While the signal is being sampled, the processor analyses the signal sampled in the previous time decoding window. This approach allows the implementation of the FSM and, consequently, the satisfaction of the timing requirements.

6: The automatic measurement station

In order to test the DTMF neural decoder according to all the international recommendations, the Automatic Measurement Station (AMS) has been developed. The AMS uses as its software environment LabVIEW [12] and as its hardware, a PC equipped with a Data Acquisition (DAQ) board made by National Instruments (fig. 7).

It is able both to generate accurate DTMF signals and to acquire the digital signal generated by the decoder, in order to determine the correspondence with the generated DTMF signal. In this way, it is possible to both easily and quickly judge the decoder performance. At the end of each test, a report is furnished in order to summarise the results.

6.1: Hardware design

The DAQ board is the LabPC-1200 [13]. This is a multifunction I/O, compatible ISA board. Its voltage input range is software programmable for 0-10 V (unipolar) or ±5 V (bipolar). The LabPC-1200 has a 12-bit ADC with a maximum analogue signal resolution of 24.4 μV. The single-channel-sampling rate of the ADC is 100 kS/s. This board has two double-buffered 12-bit DACs that are connected to two analogue output channels. Each channel can be independently configured through software for either unipolar (0-10 V) or bipolar (±5 V) operation. The resolution of the 12-bit DAC is 2.44 mV in both cases.

6.2: Software design

The software of the AMS is organised according to the flow scheme of fig. 8. The main blocks are *Test Selection, Test Procedure, I/O Interface and Test Results.*

Test Selection allows the test to be both the ITU-T Q.24 and Telcordia Recommendations. The ITU-T Q.24 Recommendation includes the recommendations of the NTT, AT&T, ETSI and the Australian and Brazilian Administrations. Moreover, the user can fix both the number and characteristics of the test signal in order to guarantee respect of parameters imposed by the recommendations.

Test Procedure is organised to automatically set the characteristics of the test signals according to the previous selected test. This procedure includes the following: *Frequency Deviation Test, Power Level Test, Twist Test,* and *Timing Test.*

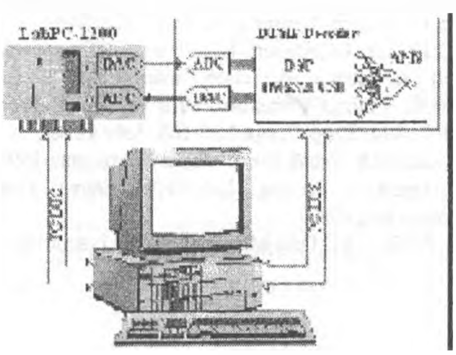

Fig.7 Measurement station for testing the decoder.

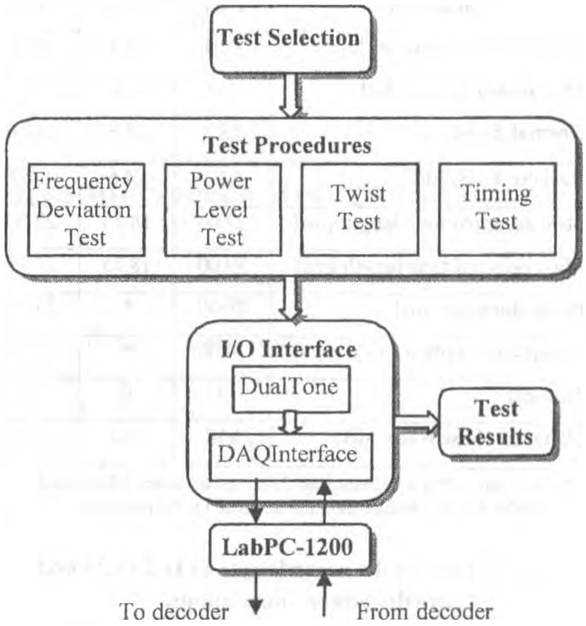

Fig. 8 The flow scheme of the AMS.

The *I/O Interface* includes the two procedures: (i) *DualTone* creates the DTMF signal with assigned characteristics, and (ii) *DAQInterface* administers communication between the AMS and the decoder by using the DAQ board. The *Test Results* generates the final report in order to evaluate the results.

7: Experimental results

In order to test the MLVQ based DTMF decoder implemented on the DSP, the AMS has been mounted as shown in fig.7. In tab.1, the test results are shown and compared with those of the decoders based on a MLP network [8, 9] and on a modified DFT [10].

Compared with the others, the MLVQ decoder shows:

- less sensitivity to frequency distortion of each tone constituting the DTMF signal;
- less sensitivity to the power level of the signal; indeed, it is able to decode signals with lower amplitudes;
- greater capability to decode dual-tone with a difference in power level between the two tones (twist);
- less sensitivity to noise presence on the DTMF signals.

Moreover, the DTMF test tapes have shown that the talk-off performances are good. Indeed, the MLVQ decoder has no false detection when invalid DTMF signals (such as speech) were input. For this test, the Telcordia test tapes were used.

The test results show that the MLVQ decoder meets the ITU-T Q.24 and Telcordia Recommendations.

Parameters	DFT	MLP	MLVQ
Frequency Tolerance [%]	≤2.1	≤1.8	≤1.9
Min. power level [dBm]	-30	-31	-32
Normal Twist [dB]	≤8.3	≤4.6	≤11.0
Reverse Twist [dB]	≤4.3	≤4.6	≤9.0
Min. accepted tone length [ms]	27.00	18.75	22.50
Max. rejected tone length [ms]	39.00	18.75	33.75
Pause duration [ms]	29.00	*	33.75
Signal interruption [ms]	14.00	*	22.50
Talk-off	141	0	0
Signal-to-Noise Ratio [dB]	≥18	≥14	≥12

* These parameters are not evaluated because the tested MLP-based decoder does not include the satisfaction of timing parameter.

Tab. 1 Test results according to ITU-T Q.24 and Telcordia Recommendations.

In particular, differently from the MLP decoder, in the case of the MLVQ decoder the timing respect is assured by the FSM implementation.

In tab. 2, the implementation specifications of the DSP-based decoders are shown. The MLVQ decoder requires a greater data memory than the DFT decoder. However, the memory required by the MLVQ decoder is less than the available memory on the DSP. On the contrary, the MLVQ decoder is characterised by a processing time (sum of the sampling time interval and the algorithm processing time interval) shorter than the processing time of the MLP and DFT decoders, implemented on the same DSP board and using the same sampling frequency of 8kHz.

Other tests were executed in order to evaluate the minimum A/D Converter (ADC) resolution. For the MLVQ based decoder, the minimum ADC resolution is 6 bits, while the MLP-based decoder needs, at least, an 8 bit ADC to correctly decode all the DTMF signals as shown in [8, 9].

Parameters	DFT	MLP	MLVQ
Sampling interval time [ms]	26.60	18.75	11.25
Algorithm processing time [ms]	*	9.00	7.25
Processing time [ms]	26.60	27.75	18.50
Data memory [words]	150	3060	1788

*The DFT algorithm works completely in the time interval between two successive samples

Tab.2 Implementation specifications of the decoders.

8: Conclusions

This paper has proposed the use of a new ANN, the Multi Learning Vector Quantization (MLVQ) network, for DTMF decoders. The MLVQ decoder has been realised and implemented on the Texas Instruments DSP TMS320C30. Due to the intrinsic characteristics of the MLVQ network, the adopted solutions allow the possibility of obtaining a faster and more robust decoder.

The experimental tests have confirmed that the presence of speech and music is no longer a problem. The neural decoder meets the ITU-T Q.24 and Telcordia Recommendations. Moreover, the MLVQ decoder shows low sensibility to noise presence and, consequently, can be used successfully in real applications characterised by a high level of noise.

References

[1] G.Intrater, O.Falik, A single-chip controller for digital answering machines, IEEE Trans. on Customer Electronics, vol. 39 No.1, Feb 1993, pp.45-48

[2] B.Koyuncu, PC remote control of appliances by using telephone lines, IEEE Trans. on Customer Electronics, vol. 41 No.1, Feb 1995, pp.201-209

[3] R.Becker, J. Mulder, SIGFRED: A low power DTMF and signaling frequency detector, IEEE Journal of Solid-State Circuits, vol. 26, No. 7, July 1991, pp. 1027-1037

[4] S.Bagchi, S.K.Mitra, An efficient algorithm for DTMF decoding using the subband NDFT, Proc. of IEEE Symposium on Circuits and Systems, Seattle, WA, USA, 28 April-3 May, 1995, vol. 3, pp.1936-1939

[5] ITU Blue Book, Recommendation Q.24: Multi-Frequency Push-Button Signal Reception, Ginevra,

[6] DTMF Detector Performance http:\\www.ece.utexas.edu \~mason\codesing\DTMF_Performance.htm

[7] T.Kohnen, Self-Organising Maps, 2nd ed., Berlin: Springer-Verlag, 1997

[8] P.Daponte, D.Grimaldi, L.Michaeli, Neural based decoder of DTMF by DSP TMS320C30, Proceedings of the Second European DSP Ed. and Res. Conf., 1998, pp.289-294

[9] P.Daponte, D.Grimaldi, L.Michaeli, Neural network and DSP based decoder for DTMF signals, submitted to IEE Proc. on Science, Measurement and Technology

[10] M.D.Felder, J.C.Mason, B.L.Evans, Efficient dual tone multi frequency detection using the non-uniform discrete Fourier transform, IEEE Signal Processing Letters, vol. 5, no. 7, pp. 160-163, July 1998

[11] TMS320C3X, User's Guide, Texas Instruments, 1994

[12] M.Chugani, A.Samant, LabVIEW Signal Processing, Prentice Hall, 1998

[13] Lab-PC1200/AT, User Manual, National Instruments, 1996