

Auditory Model Implementation on a DSP32C-Board

P. Cosi (*), L. Dellana (**), G.A. Mian (***) and M. Omologo (**)

(*) Centro di Studio per le Ricerche di Fonetica - C.N.R. P. zza G. Salvemini, 13 - 35131 Padova (Italy)

(**) I.R.S.T. Ist. per la Ricerca Scientifica e Tecnologica Pantè di Povo - 38050 Trento (Italy)

(***) Dip. di Elettronica e Informatica Via Gradenigo, 6 - 35100 Padova (Italy)

RÉSUMÉ

Nous avons utilisé une carte à virgule flottante DSP pour la mise en oeuvre d'une version légèrement modifiée du modèle auditif Seneff. Nous allons décrire la structure de matérielle e de logicielle du système de même que examiner sa performance comparée au système original développé et simulé à l'aide de langages telles que FORTRAN ou C.

INTRODUCTION

A joint synchrony mean-rate auditory model was recently developed [1] and successfully applied [2] as a front-end for speech recognition systems. Then some preliminary speech recognition experiments in noisy environments were performed [3]. This system seems to represent a powerful way to analyse speech, its main drawback being a very high computational complexity.

In order to better investigate its usefulness in speech recognition, such a system, originally developed in high-level languages such as FORTRAN or C, was implemented on a floating-point DSP board. The choice of floating-point processor technology is mandatory for the very high dynamic of the signals involved in the system.

The DSP board that was used is based on the AT&T DSP32C processor, capable of executing up to 25 million arithmetic operations per second. The board was interfaced with a PC-AT machine. The available library of the DSP32C source code routines was widely exploited in the described application and the software was all written in the C-language of the DSP.

The system implemented on the DSP32C was 20 times faster than the original simulation (C version running at 400 times the real time on a SUN 3). Presently the system runs at 20 times the real time.

This paper describes how such a system was realised and how the model was changed in order to operate with the arithmetic of the DSP.

DSP32C PROCESSOR AND SYSTEM CARD

The DSP board utilized to implement the system is based on the AT&T DSP32C floating-point processor and is produced by Loughborough Sound Images Ltd.. As already underlined in the introduction floating-point processor technology was chosen because of the very high dynamic of the signals involved in the system.

The DSP32C is a 32-bit floating-point processor entirely built in CMOS .75 μ m technology. The core of the processor is the 32-bit floating-point Data Arithmetic Unit (DAU), capable of executing up to 25 million arithmetic operations per second (Mflop/sec), at a 50 MHz clock frequency [5]. A 4-stage pipeline is implemented by the DAU unit, giving high performance while executing floating-point arithmetic operations. However, the nominal power of 25 Mflops/sec is obviously reduced to an average of about 5-7 Mflops/sec, because the DAU executes, not only pure arithmetic operations, but also other instructions such as various controls and memory or external word transfer operations, i.e. Parallel Input Output (PIO), Serial Input Output (SIO) or Direct Memory Access (DMA). The Control Arithmetic Unit (CAU) executes logic and control operations with 16 or 24 bit fixed-point arithmetic. CAU functioning can be autonomous from the DAU, e.g. while transferring data, or dependent, e.g. while creating memory addresses for the DAU operands. The internal floating-point representation of a number is different from the standard IEEE format and consists of a 24-bit mantissa, normalized between 1.0 and 2.0, and of an 8-bit exponent, leading to a range of 1500 dB.

The general hardware architecture of the Loughborough DSP32C SYSTEM CARD is shown in Figure 1. The board is a typical 16-bit single slot PC-AT (80286/386) board, with DMA facility and two 16-bit sample&hold A/D and D/A converters with 200 kHz sampling frequency, programmable via a Counter/Timer from 153.85 kHz to 0.15259 kHz. For a complete description of the DSP32C SYSTEM CARD

ABSTRACT

A floating-point DSP board was used to implement a slightly modified version of the Seneff auditory model. The hardware and software architecture of the system are described and its performance are examined in comparison with the original high-level language developed system.

refer to [6].

DSP32C System Card

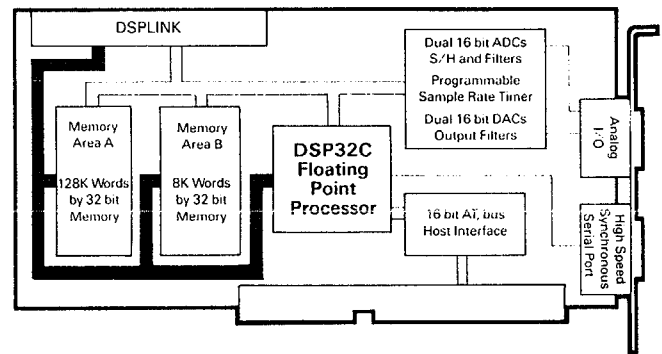


Figure 1. Block diagram of the Loughborough DSP32c System Card.

As for the software architecture, a full set of interface functions from the board and the host PC are furnished with the system (Loughborough), together with a graphic Monitor-Debugger, a C-language system, various optimized assembly language libraries supporting the C-compiler (FFT, filtering, etc.) and a processor Simulator (AT&T). The various steps of the complete DSP32C software generation process are described in Figure 2.

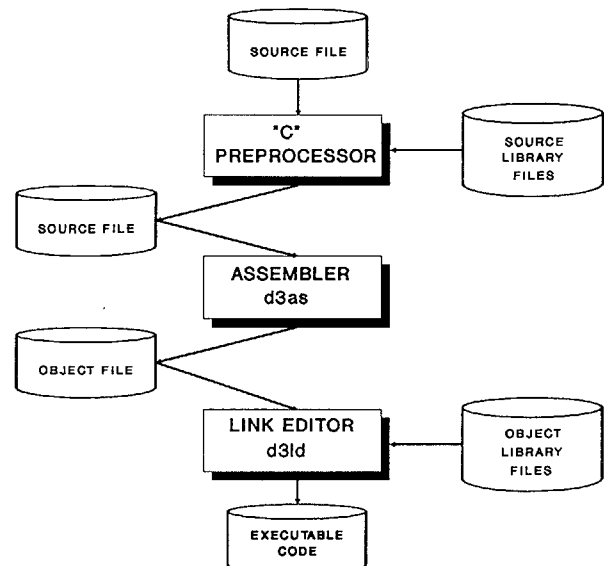


Figure 2. Software generation procedure for the DSP32C.



MASTER-SLAVE SOFTWARE SYNCHRONISATION

The global software organisation of the system is based on a typical synchronised master-slave process, as graphically shown in Figure 3. The master process provides speech segments to the slave process every 5 ms while the slave process, which corresponds to the whole auditory system, gives as output two 40-element vectors representing the envelope and the synchrony detector responses [1].

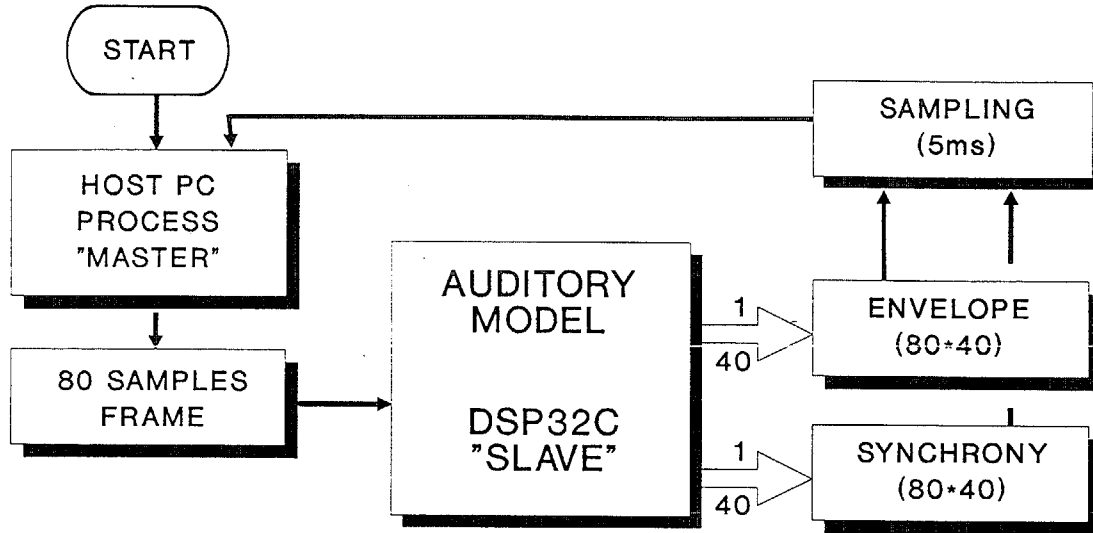


Figure 3. Master-slave implementation of the auditory model on the DSP32C System Card.

AUDITORY MODEL IMPLEMENTATION ON THE DSP32C

The auditory model implemented in this work refers entirely to the S. Seneff's original version [1]. Only a few modifications in the filter-bank design were introduced [4].

The output of the model consists of two 40-element vectors, both producing spectral representations appropriate for various tasks of a speech recognition system. One vector represents the average rate of neural discharge for each channel, while the other represents the level of dominance of periodicities at each channel's center frequency. These parameters are extracted by an ENVELOPE and a SYNCHRONY detector module respectively [1]. A block-diagram of the original model [1] is illustrated in Figure 4, where each block describes also the corresponding functions. All these functions were implemented on the DSP32C SYSTEM CARD with only a few structural modifications [7] to the original system formulation [1].

In Figure 4, in the 40-channel filter bank module, (a), (b), (c) refer to the initial prefiltering stage, adopted to eliminate the very high and the very low frequency components, to the IIR and to the FIR components respectively (see [1]). In the second module (Hair Cell Synapse Model) (a), (b), (c), (d) refer to the half-wave rectification, to the adaptation, to the synchrony reduction and to the rapid adaptation components respectively (see [1]).

Due to the very high dynamic of the signals involved in the filtering process (a 16-bit input signal can produce channel outputs in the range 10^{10} - 10^{14}), it was found necessary to equalize the dynamic range of the FIR outputs. To this purpose an attenuation factor K_i was computed for each filter tap [1] and introduced in the FIR chain [1]. In the original C-language implementation with double-precision variables, the attenuation factors were instead inserted at the output of each IIR filter. Another modification to the original C-language system, was made in the half-wave rectification module of the model second stage (see Figure 4). In fact, the original "arctan" function was substituted by an approximating piecewise linear function without distortions in the system outputs and with a considerable reduction of the computational load. Another very demanding module, corresponding to the Generalized Synchrony Detector (GSD) [1], was designed with a very efficient data structure [7], especially suitable to the DSP32C processor. Other modifications to the original high-level language version were introduced but only at the level of programming "tricks" which were always applied with the aim of speeding up execution time [7]. The filter bank module,

and all other modules using IIR and FIR filters, like those requiring the computation of the envelope or the mean of their related input signals, were realized using canonical direct-form filter structures like those illustrated in Figure 5.

Table 1 gives the DSP32C execution-time in usec for each building block of the model depicted in Figure 4.

AUDITORY SYSTEM MODULE	FUNCTION	EXEC-TIME (usec)
FILTER BANK	pre-filtering FIR	1.9
	principal cycle	92.8
	FIR filtering	56.8
	IIR filtering	64.8
	overhead	4.3
	total	220.6
EAR CELL SYNAPSE	half-wave rectif. (average)	140.0
	short-term adapt. (max)	60.8
	synchrony reduct.	93.6
	rapid AGC	178.4
	overhead	4.2
	total	477.0
ENVELOPE DETECTOR	envelope extr.	91.3
SYNCHRONY DETECTOR	total	540.8

Table 1. DSP32C execution-time in usec for each single module of the auditory model (50 MHz clock). Some of the measures refer to the average and some others to the maximum execution-time, instead of referring to the actual execution-time, because these are strictly dependent from the input signal shape. Only the total is given for the synchrony detector module. For the execution time of the various sub-modules implementing the GSD refer to [7].

IMPLEMENTATION PROBLEMS

The most critical step of this work was represented by the unexpected propagation of the arithmetic noise through the model, due to the fact that even the 32-bit accuracy of the DSP floating point is critical for this task. In fact, the original C-language implementation used

double precision, e.g. a 64-bit floating-point accuracy. The discrepancies between single and double precision become intolerable in the 40-channel critical band filter bank, due to the very low bandwidth of the higher order channel responses. The gain of these filters can reach 200 dB. Figures 6a and 6b show the frequency response of the last five channel outputs corresponding to double and single precision arithmetic, respectively. The bump in the high frequency region of Figure 6b is apparent. A detailed analysis showed that this behaviour has to be attributed to roundoff noise accumulation in the corresponding FIR sections. Different strategies were considered to face this problem. The simple expedient of moving on $z = -1$ the filter zeros proved to be the most effective. Figure 6c gives the corresponding frequency response. It can be appreciated that the roundoff noise effect was reduced to 100 dB below the maximum.

CONCLUSIONS

At present the system allows the user to obtain, in a reasonable time, the synchrony and envelope outputs of the model. The system was speeded up 20 times compared to the original C-language simulation running on a SUN 3. The present version of the DSP software runs at 20 times the real time. Further improvements are expected after translation of some routines into assembler code. Further work is also envisaged in the reduction of the intrinsic complexity of the model, e.g. reducing the number of channels, with a fast check of the consequent possible reduction in recognition performance.

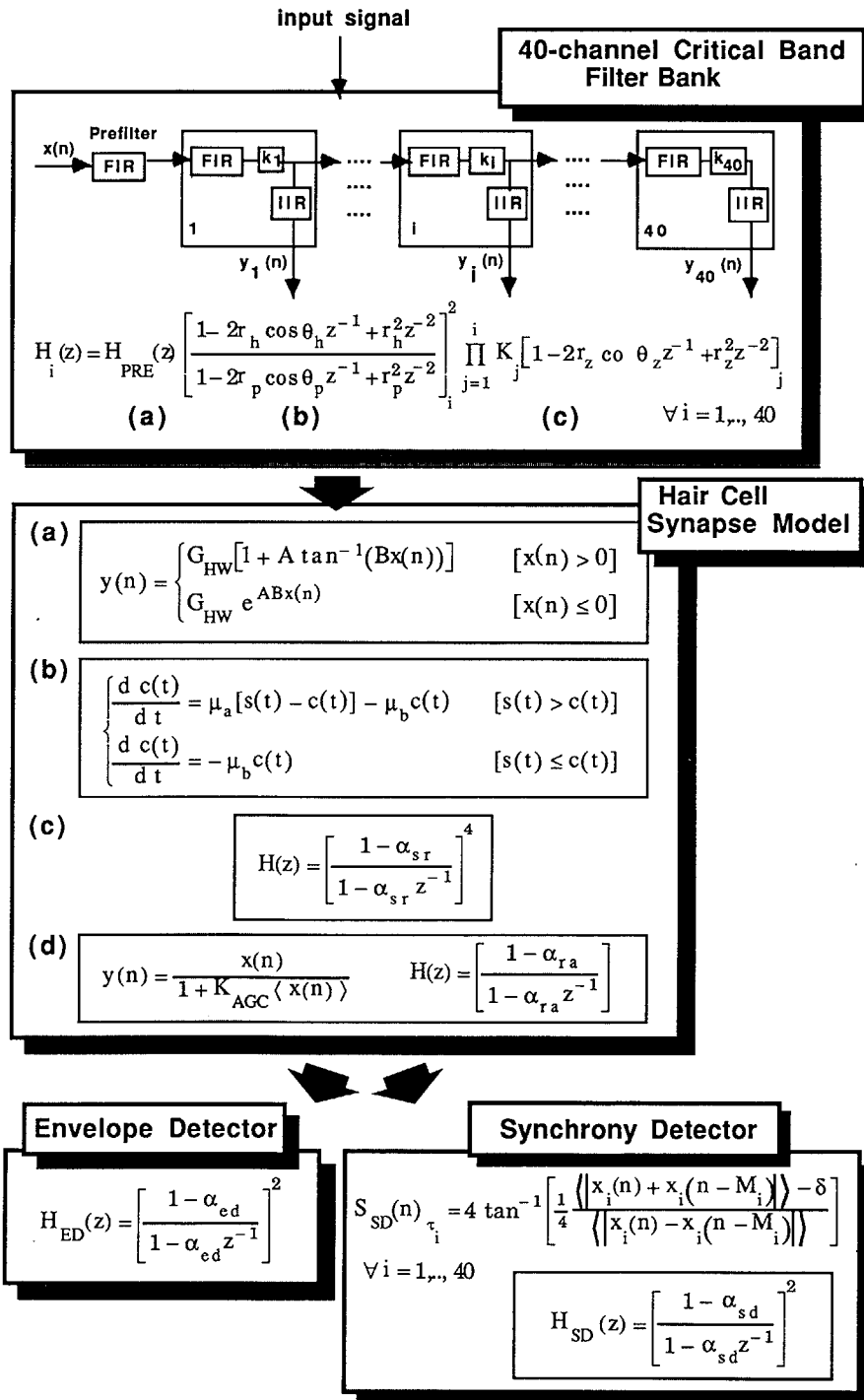


Figure 4. Auditory Model's block diagram.

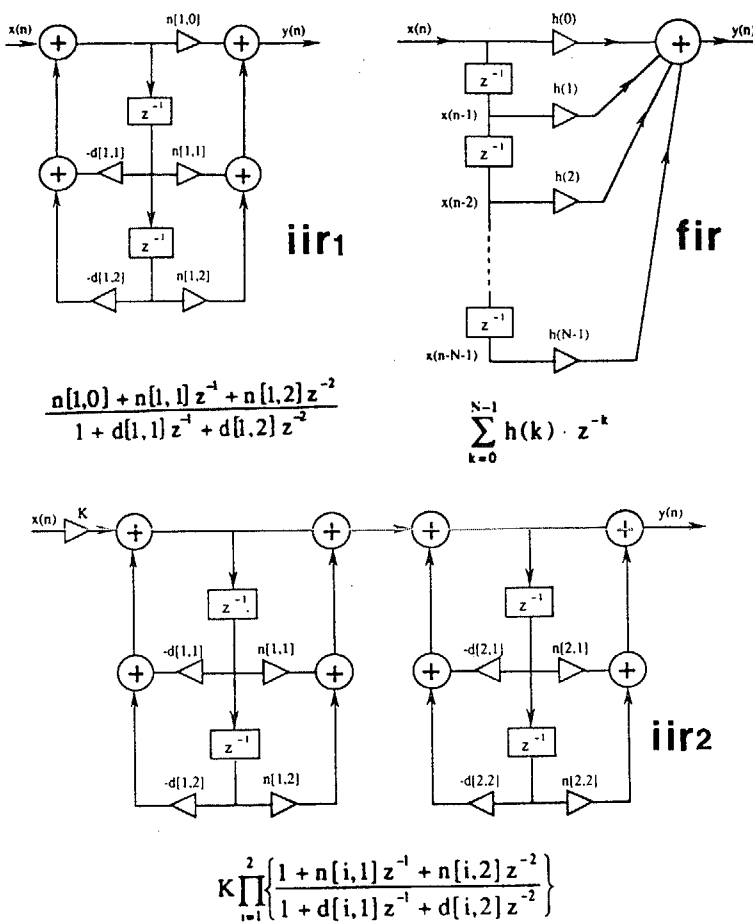


Figure 5. DSP32C IIR and FIR structures.

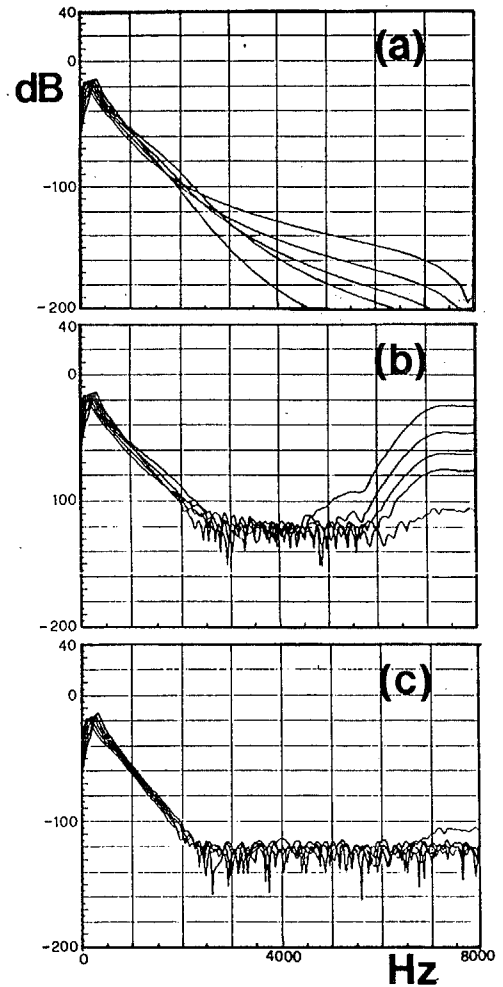


Figure 6. Frequency response of the last five channel outputs corresponding to double (a) and single (b) precision arithmetic. The corrected frequency response (zeros in $z = -1$) is given in (c).

REFERENCES

[1] S. Seneff (1988), "A joint synchrony/mean rate model of auditory speech processing", *Journal of Phonetics* (1988) 16, pp. 55-76.

[2] R. De Mori, Y. Bengio and P. Cosi, "On the Generalisation Capability of Multi-Layered Networks in the Extraction of Speech Properties", *Proceedings of IJCAI*, August 20-25, 1989, Detroit, pp. 1531-1536.

[3] P. Cosi, D. Falavigna, G.A. Mian and M. Omologo, "A Comparison between Mel-scale Cepstrum and Auditory Model Representation for Noisy Speech Recognition", *Proceedings of EUSIPCO-90*, Barcelona, Spain, pp. 1199-1202.

[4] P. Cosi, "Phonetically-Based Multi-Layered Neural Networks for Vowel Classification", *Speech Communication*, Vol. 9, 1990, pp. 15-29.

[5] AT&T, *WE DSP32C Digital Signal Processor - Information Manual*, December, 1988.

[6] Loughborough Sound Images, Ltd, *DSP32C PC System Board; User Manual*, Issue 1.0, July 1989.

[7] L. Dellana, "Realizzazione di un Modello del Sistema uditivo Periferico su Processore DSP32C", *Tesi di Laurea*, Corso di Laurea in Ingegneria Elettronica, Dipartimento di Elettronica e Informatica, Università di Padova, Facoltà di Ingegneria, A.A. 1989-1990.