WHITE PAPER

Texas Instruments

A Prototype Lab Box with DSK 'C6711/13 for Rapid DSP Algorithm Development

By
Graziano Bertini and
Massimo Magrini
Inst. of Science and Technology of
Information "A. Faedo"
(ISTI-CNR) National Research Council
Pisa, Italy

Abstract

A system based on the TMS320C6711 DSK, enclosed in a chassis, for real-time DSP algorithms testing, is described. Some evaluation of software rapid prototyping tools for a "Virtual Dubbing" application voice is also given.

Summary

Rapid prototyping is a new approach in Digital Signal Processing systems development. With the advent of MATLAB®'s Real-Time Workshop® (RTW) it is now possible to compile, load, and execute graphically designed The MathWorks Simulink® models on an actual DSP platform, without spending many workdays coding in typical DSP-oriented languages or C/C++ dialects (compilers).

RTW supports the powerful Texas Instruments TMS320C6000™ DSP platform, including the TMS320C6711 DSK we are employing in our lab. Here we describe our experience in using these tools to develop and test real-time voice transformation algorithms proposed by the Musical and Architectural Acoustic Lab of FSSG-CNR (Fondazione Scuola San Giorgio-Consiglio Nazionale delle Ricerche, Venice, Italy).

DSP Support Tools Evolution

Real-time digital signal processing made considerable advances after the introduction of specialized DSP processors. Suitable starter kits, with a specific DSP processor and related software tools such as assemblers, simulators, debuggers, and so on, were provided in order to make system design and application development easier.

However, the first generations of starter kits were very cheap, including only the DSP and a few hardware devices. The system design for a specific application required the realization of a PCB including necessary hardware components around the DSP chip. The lab prototyping was expensive and the final results were not achieved in a "time-to-market" manner.

Figure 1 on the following page shows an example of the old starter kit, TMS320C26 DSK, integrated into a system using infrared devices, in order to control sounds and music production by hand movements as realized in our laboratory.

Subsequently, very suitable DSKs with audio codecs, static and/or dynamic memories and faster PC interfaces became available. Moreover, improvements were made to software tools and more complete libraries were included. More recently, one could compile and execute DSP algorithms from graphic models and schemas. For example, Simulink Real-Time Workshop is able to work directly with TI DSPs.

During the TI Europe One-Day Workshop organized in Florence in April 2003 we discovered the features of the TMS320C6711 DSK and we also obtained, as a grant, a kit for starting the experimentation. The following article details our experience with the implementation of complex voice transformation algorithms using the state-of-the-art hardware/software tools for rapid protoytping, in other words, the evaluation board with powerful floating-point DSPs and the support of a high-level software compiler (Simulink with Real-Time Workshop).

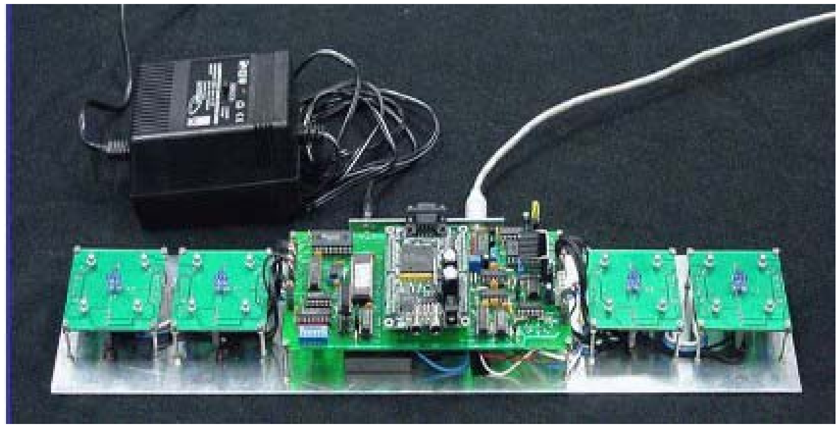

Figure 1. TMS320C26 DSP Starter Kit (in the center of the prototype) integrated in a system for gestural control of computer music via MIDI (Musical Instruments Digital Interface) protocol.

A System Based on the DSK Board for a Complex DSP Application: A Case Study

In 2004 we collaborated as consultants with the team from FSSG-CNR within the framework of the RACINE-S European Project (IST 2001-37117). The goals of this project were the reconstruction of highly damaged segments in movie films, both audio tracks (speech) and video sequences. The project involved various research/university units and industrial partners, each working on a different aspect: image-audio reconstruction and audio rendering improvement using intensimetric acoustic ambience reconstruction.

Our collaboration only focused on the development of a DSP system for supporting the speech restoration functions. More precisely, the main goal of our work, here described, was to verify the possibility of performing real-time voice transformation using a DSP-embedded system for implementing the proposed algorithms.

Besides the implementation of a set of methods, such as LPC/VC (Linear Prediction Coding / Voice Conversion), the FSSG-CNR team proposed innovative methods for improving the quality of synthesized voice. The entire set of operations represents a particular implementation of the so-called "Virtual Dubbing" procedure. The basic steps of the complete project included: designing a method for high-quality voice transformation, implementing a suitable algorithm in MATLAB® and Simulink® and finally, translating it into DSP target code by means of a rapid prototyping approach. The original code was developed in MATLAB and so we used The MathWorks Real-Time Workshop® (RTW) DSP platform for rapid prototyping.

Real-Time Implementation Based on DSP

Since the complete implementation of the method has a great computational complexity, it is necessary to test the possibility of implementing it in real time.

As we developed the basic algorithm of the method at a research level, we needed to perform tests on a flexible platform that allows the implementation of the non-optimized algorithms with a reasonable effort. For this reason, we decided to implement them on a DSP hardware platform—Texas Instruments floating-point 32-bit processor (the TMS320C6711 DSP).

This DSP processor is based on the VLIW (Very Large Instruction Word) technology, which allows fast parallel computing jointly using its optimized "C" compiler. For a rapid evaluation of the TMS320C6711 processor, a Developer Starter Kit (DSK) is available from Texas Instruments, comprising a board and the software tools. The board must be connected to a standard PC running under its development environment [Code Composer Studio™ Integrated Development Environment (IDE)].

This board (Figure 2) includes 16 MBytes of SDRAM, plus 128 KB of external Flash. Along with other features, the most relevant for us is the presence of an audio codec (AD/DA converter), the TLC320AD535. Although its quality is rather poor (16-bit, 8-kHz sampling rate), it is sufficient for testing DSP applications in the speech audio band. In addition, the board can host additional daughter boards with better performance A/D converters: for example the PCM3003 daughter board, which allows 48-kHz sampling rate (16-bit stereo). In a second development phase we also tried to use the DSK board based on the slightly faster TMS320C6713 DSP. This second board has a USB link with the PC, which is much faster and reliable than the parallel connection.

The MathWorks' Real-Time Workshop® builds applications from Simulink® diagrams for prototyping, testing and deploying real-time systems on a variety of target computing

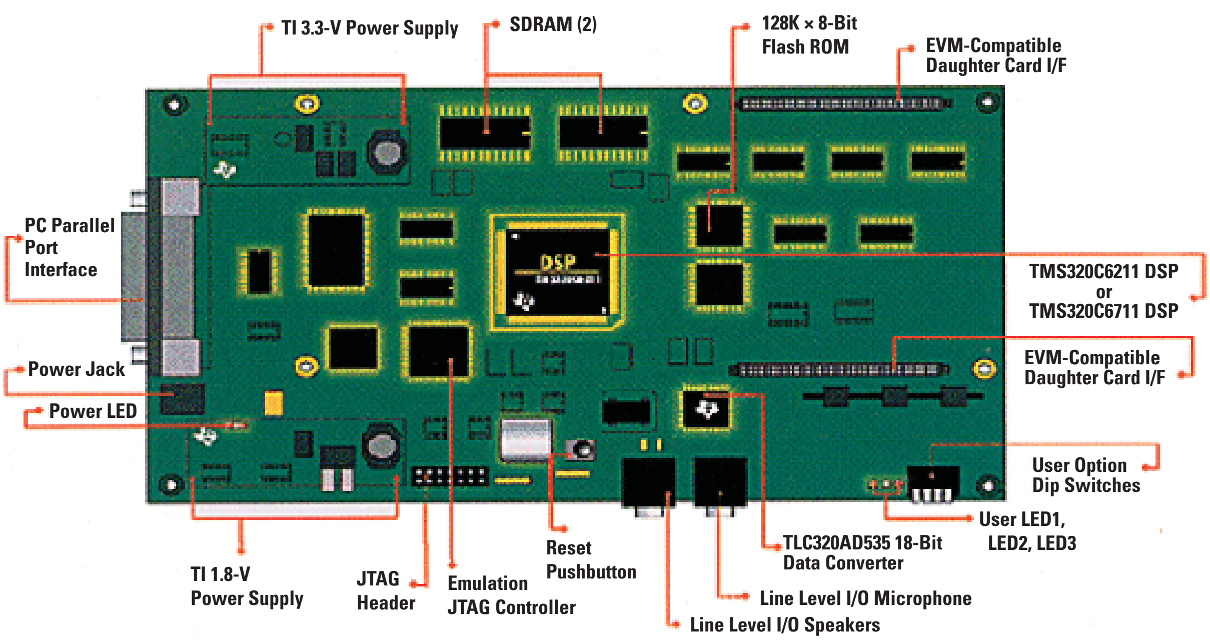

Figure 2. View of the TMS320C6711 DSK board.

platforms, including Texas Instruments TMS320C6000™ DSP platform of processors (*Embedded Target for TI C6000™ DSP*). This library offers a set of Simulink® blocks, implementing the behavior of various devices on the board (Figure 3).

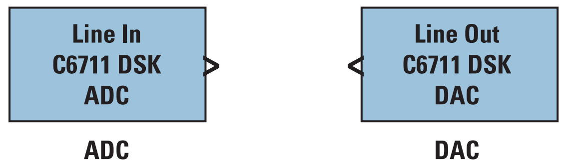

Figure 3. Line In and Line Out Simulink blocks.

Once we simulate the Simulink model, it is possible to create some settings in model configurations in order to instruct MATLAB to start the DSP code building process. The resulting translation will transform the Simulink model into a Code Composer Studio™ IDE C language project and then, controlling the DSP compiler, into a DSP binary code, which can finally be downloaded on the DSK board. Before the translation, we tuned the model using The Mathwork's Model Advisor, a software tool that comprehensively analyzes the Simulink model to help the programmer to appropriately configure Simulink and Real-Time Workshop®.

Hardware Platform

In order to satisfy a specific request of the RACINE-S project team and for easier use of the system, we built a custom chassis for the starter kit board (Figure 4). This embedding

Figure 4. View of the system based on the DSK board, connected to the PC.

hardware is equipped with additional analog circuitry that provides analog input/output interfacing, a power supply, and signal conditioning (with a low-noise preamplifier). The front panel contains lamps for system state monitoring and a reset button (corresponding to those on the DSK board). The chassis also shields the system from electromagnetic interferences, improving the system's reliability.

Method for Virtual Dubbing

The starting point for virtual dubbing consists of digitized audio data arising from archived movie sequences that are badly damaged or even partially missing. In this last case, information is taken from segments positioned before and after the missing pieces of film. This audio data, which is supposed to be available as a standard audio file, contains the relevant information about the target voice which we need to reconstruct.

In a traditional dubbing studio a professional dubbing speaker dubs (Figure 5) the sentences pronounced by the target voice, if available, or simply reads the script, if the corresponding sequence is missing. The main idea at this point is to provide the voice of the professional dubbing speaker with the features of the target voice. This voice conversion process can be performed in real time during the dubbing process or off-line at an audio post-production level.

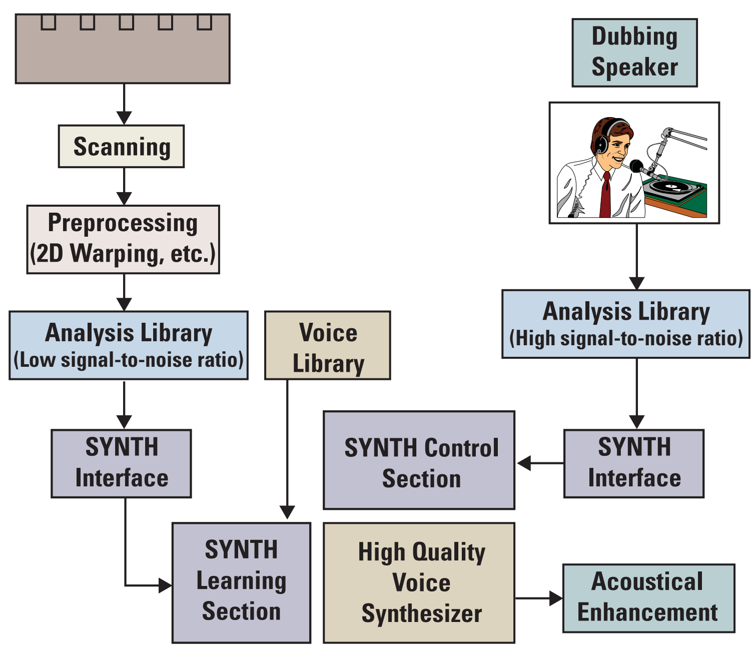

Figure 5. Virtual dubbing block diagram.

Algorithm Description

As previously stated, the project requires the conversion of a source speaker's voice into another, as if it were pronounced by a different (target) speaker. Today's voice-conversion algorithms mainly rely on residual-excited LPC synthesis[1,2] or on STFT-based synthesis.[3,4]

In both cases the core of the algorithm is the definition of a map F(x) that transforms a feature vector x onto a new vector y. In the former case, the feature vector is typically a set of LPC coefficients or, due to its better interpolation properties, a set of line spectral pairs coefficients (LSP).[1] An example of such a voice conversion algorithm is available in the FESTIVAL TTS system within the OGIresLPC synthesis plug-in. In the latter case, the feature vector can be some sort of spectrum magnitude representation, such as the FFT magnitude or the mel-cepstral coefficients.[3]

A simple definition of the map F can rely on some parametric non-linear function approximation tools, for example, neural networks or Radial Basis Function Networks (RBFN).[4] Another common approach to the definition of the map structure is to use a probabilistic, locally linear function. A widely used technique makes use of Gaussian Mixture Models (GMM), which are capable of embedding an acoustic model of the source speaker in the mapping function based on a Gaussian Mixture.[1,2]

The design of the conversion function requires a number of basic steps. Firstly, a database of sentences uttered by different speakers is generated. Then, the feature vectors by a frame-based LPC or sinusoidal analysis are computed. Before the training step, a dynamic time warping (DTW) procedure is required to time-align the feature vectors derived from the first speaker with the ones accounting for the second speaker. This guarantees that in the input-output training the first speaker's phoneme corresponds to the same phoneme of the second speaker. At this stage a conversion function based on an RBFN can be trained directly by identifying the parameters given by the input-output training pairs.

In the current project, a Gaussian Mixture voice conversion model is used, in which two training sets of LPC coefficients extracted by the time window series and containing the source and the target (voice) signal respectively form the conversion function. This function is then used for mapping source into the target LPC coefficient sets.

Our LPC/VC model synthesizes the target voice by filtering the LPC residual of the source voice using target LPC coefficients obtained by converting the correspondent source LPC parameters by means of F. Careful design of the LPC/VC model which requires the voice signal to be separated into "voiced" and "unvoiced" time windows. At this stage of the project, a standard segmentation of the signal is divided into small segments of equal time lengths, which are sufficient to test the LPC voice conversion.

When used to model the speech process, the components of the GMM represent different phonetic events. Let us suppose that a sequence of P-dimensional column vectors {[x\vec]t}, t=1,...,T, which represents the time-varying spectral envelope of a source signal, has been fitted by a GMM. Moreover, let us assume that a sequence of P-dimensional column vectors {[y\vec]t}, t=1,...,T, having the same length of the source signal, is the target of the conversion. We defined a spectral conversion function as a map F to transform each vector in the input sequence into the vector which occupies the same position in the output sequence, thus preserving the time information of the input and output data.[9]

Implementation Using Simulink® / Real-Time Workshop®

Following the Rapid Prototyping approach, we implemented our algorithms using Simulink, which provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations (Figure 6).

The voice signal is first divided into frames having a strong overlapping factor in order to prevent the appearance of major artefacts occurring during the transition from one frame to another. For our specific purpose, here we have chosen a frame size of 1024 samples and a hop-size of 64 samples, along with a Blackman windowing that provides smooth transitions along adjacent frames. These figures seem to be widespread enough to provide sufficiently accurate results in the majority of the test cases. Then, each frame

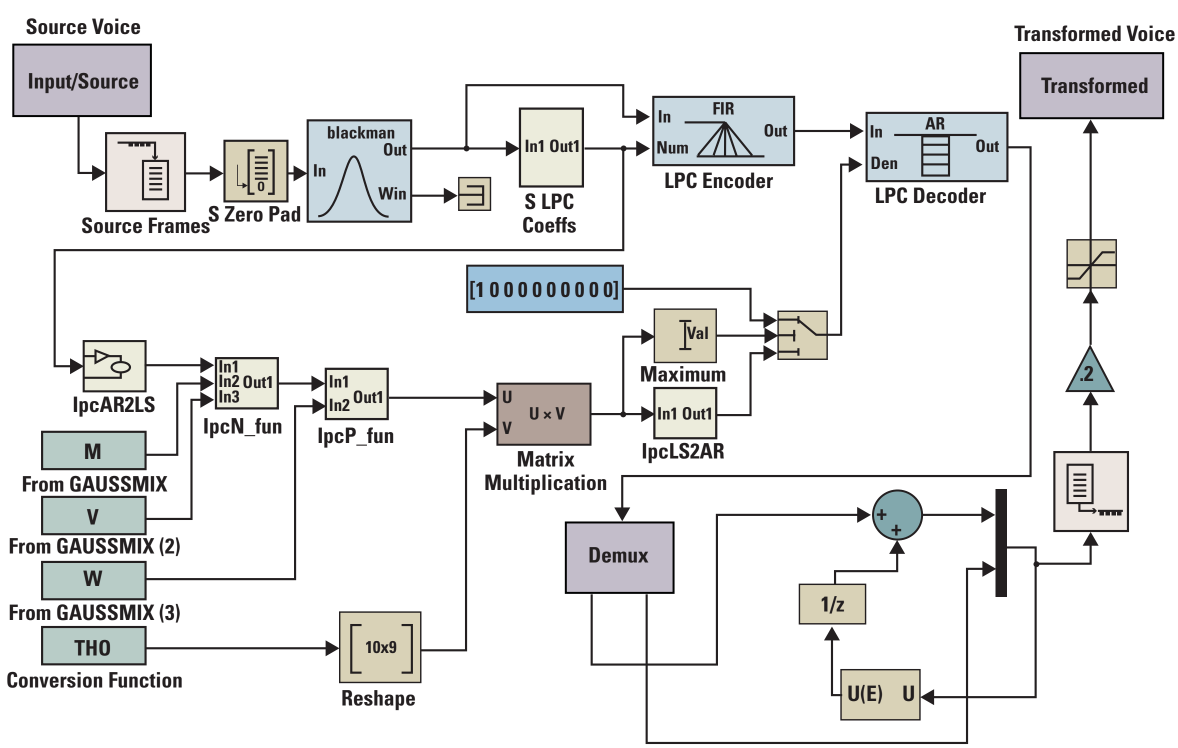

Figure 6. Complete voice conversion algorithm.

is LPC-encoded. An LPC order equal to 10 is considered suitable enough in most literature. For LPC analysis a classic extraction algorithm is employed that performs the autocorrelation of the input signal followed by a Levinson-Durbin LPC extraction algorithm.

As previously stated, the design and the training of the VC model are performed off-line, with a number of GMMs set to 10. The design phase first needs the target voice LPC coefficients that are treated the same way as before. These coefficients, transformed into line spectral pairs, are then used for the GMM modelling procedure, obtaining a set of transformation parameters (M, V, W, TH0) (Figure 6) that we will use for the re-synthesis process.

The core steps of the re-synthesis are:

- **lpcar2ls** converts LPC coefficients into line spectral pairs. This requires a roots computation and a deconvolution.
- **lpcN_fun** and **lpcpi_fun** compute parameters used for calculating the set of line spectral pairs (lsp) encoding the transformed voice according to a specific formula.
- **lpcls2ar** converts the line spectral pairs back to transformed LPC coefficients.

The reported algorithms represent an important improvement over the usual LPC techniques since by means of the voice transformation algorithms, a detailed description of the spectral properties of the target and source signals is taken into account.[5]

The complete model was simplified in order to develop in parallel the DSP application and to test the system (Figure 7). This reduced model lacks the GMM-based conversion

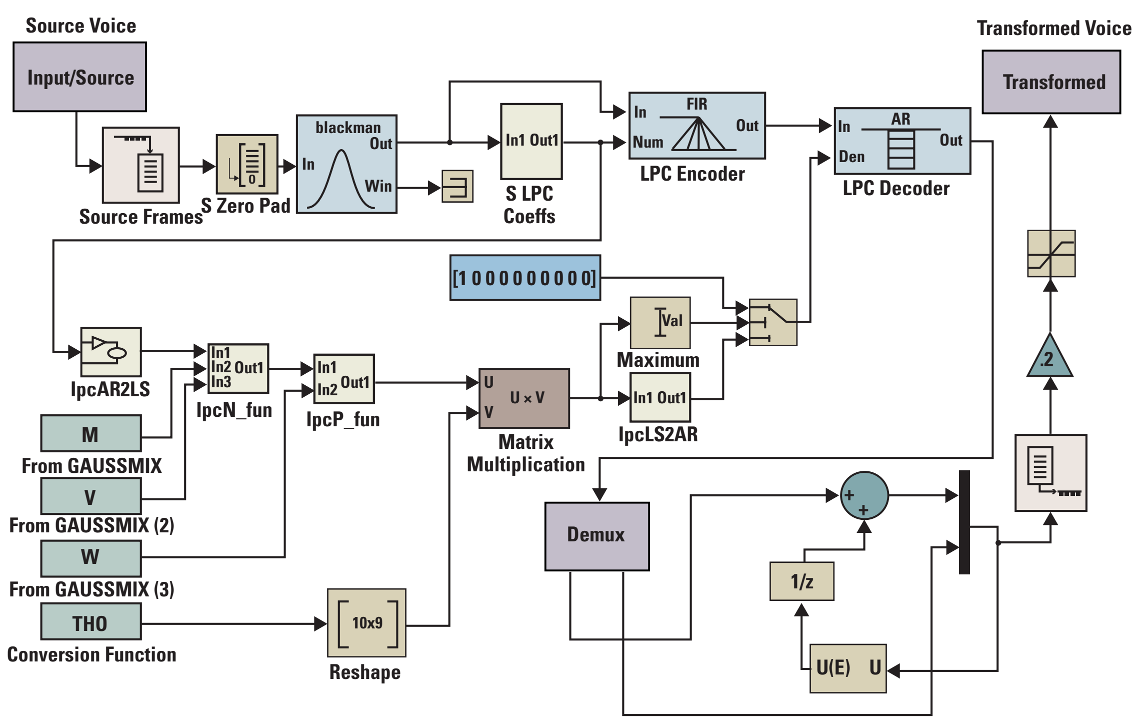

Figure 7. Simplified voice conversion algorithm.

part and requires, as input, synchronized source and target voices. For the test run we used a short pre-recorded sequence of vowels that is uploaded in the DSP hardware together with the code of the model. The output of the model is connected with the DAC line out.

Hardware Prototype

We assembled a chassis of dimension Rack U2 standard with front and back panels. The system includes a TMS320C6711 DSK (Figure 8), together with a power supply and a small board created in our lab, containing a stereo preamplifier, with adjustable voltage gain, necessary to allow full codec dynamic.[8] It was realized using an NE5532 low-noise op-amp. The characteristics of the preamp are: gain 0–20 dB, power supply ± 12 V, input impedance 47 KΩ, and bandwidth 20–30 KHz (± 3 dB).

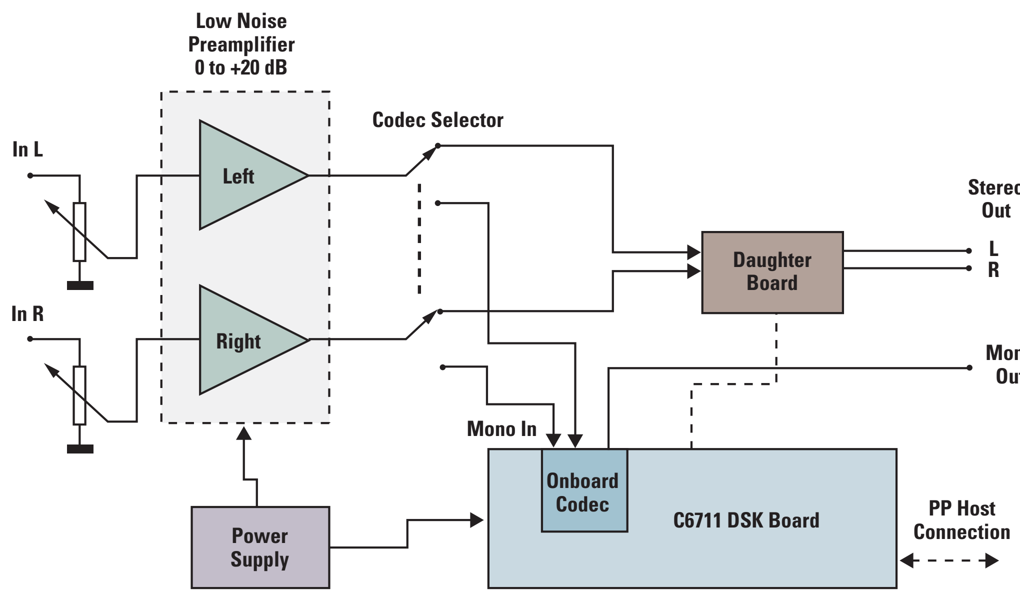

Figure 8. Hardware architecture.

On the front panel were set switches, a power supply indicator, potentiometers for setting input level, various stereo jacks for inputs and outputs, user LEDs, user switches, and a reset push button, which carry out the same tasks of the corresponding parts present on the board (Figure 9 on the following page). The LEDs driver current was derived from the on-board D-latch SN74LVTH16374 output, paying attention to the fan-out.

Control dynamics of the DA/AD converter were implemented on the DSP using software procedures, and the overflow condition was indicated at the same time by user LEDs on the board and front panel.

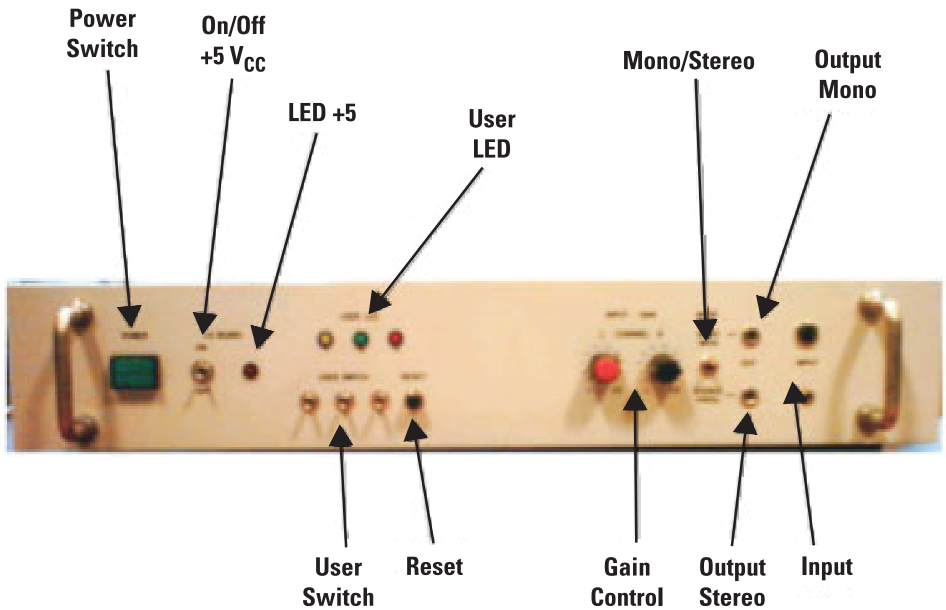

Figure 9. Front panel of the enclosing hardware.

Preliminary Results and Conclusions

We successfully converted and loaded the reduced model into the DSK boards and the signal at the DSK DAC output exactly reflects the simulation output (Figure 10).

The main goal of this project was to develop instruments and methods for reconstructing audio sequences starting from old, badly damaged movie films. We proposed a solution based on LPC/VC algorithms plus GMM transformation. This solution provides a successful approach to the physical modeling of the vocal tract, physical meaningfulness, and directivity of the control parameters. The quality of the resynthesized voice can be further improved using non-linear techniques.[6,7] We are investigating this possibility

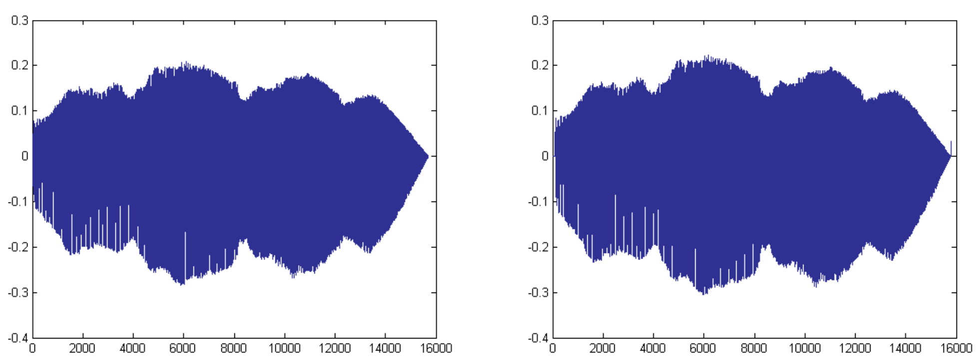

Figure 10. Simulated (left) vs. real-time-generated signal (right).

through the interfacing of an efficient non-linear pitch detection model with the core of LPC voice conversion algorithms. The dynamic model allows for an efficient emulation of the perceptive function with a reduced number of parameters.

These algorithms were first tested by The MathWorks Simulink® environment. High-quality voice reconstruction tests have been carried out with good results. According to one of the project's goals, we tested the possibility of a real-time implementation using a modern DSP platform provided by Rapid Prototyping tools. Considering the great complexity of the whole algorithm, together with some limits of the hardware/software development platform used, real-time implementation has been carried out only in a simplified form. It is necessary to optimize certain parts of the algorithm, writing them directly into S-Functions (see MATLAB® documentation), in order to test the complete algorithm.

We achieved an interesting result of becoming familiar with the rapid prototyping approach: the workflow we used was very powerful and is promising for further investigations and improvements.

About the Authors

Graziano Bertini (1943) has worked at the Istituto Elaborazione dell'Informazione of the Italian National Research Council (IEI-CNR) since 1962. In 1975 he received a degree in Computer Science from the University of Pisa: his thesis topic was the design of the polyphonic, real-time, Audio Terminal-TAU2, then used by musicians and researchers for computer music experiments (1975–1985). From 1985 to the present, his research involves the development of DSP-based systems, mainly for electroacoustics and base-band telecom applications.

Massimo Magrini (1966) graduated from the University of Pisa with a degree in Computer Science in 1994. Then he has worked as a freelance consultant for several private companies and for academic institutes (ISTI-CNR, IFC-CNR, SNS-Pisa), mainly in the fields of digital signal processing and multimedia.

A Short History of Signal Processing Research Activity at ISTI–CNR

The activity on signal processing started during the 1970s at the Institute of Information Processing (Ist. di Elaborazione dell'Informazione, IEI-CNR) in Pisa, Italy. The research mainly concerned images and signal processing in the biomedical and the computer music fields. The systems we developed were based on analog and discrete devices. The emerging class of DSPs available on the market during the 1980s made it possible to solve complex applications by using algorithms and digital systems, that is in a very competitive way with respect to the corresponding analog technology.

The research was mainly conducted in the Signals & Images Processing Lab of the IEI-CNR in cooperation with other CNR institutes, universities and national and international groups. Later, DSP technology has obviuously been adopted by other CNR departments, i.e., IFC-CNR (Ist. di Fisiologia Clinica for biomedical applications) and IBF-CNR (Ist. di Biofisica for biophysical applications).

During the year 2000, all the CNR Institutes of Pisa (over 500 people) were grouped together in the science park, "Area della Ricerca CNR di Pisa" (www.area.pi.cnr.it). IEI and CNUCE-CNR (Centro Nazionale di Calcolo Elettronico) Institutes merged in 2002 becoming the ISTI-CNR (Institute of Science and Technology of Information "A. Faedo").

Our lab investigated various topics regarding the development of DSP systems for audio synthesis additive techniques, sound effects, and active noise control. The know-how gained from these experiences generated wider interests and also activated collaborations with private companies, dealing with: digital audio (Leonardo spa, SEED srl, Massa), digital treatment of voice (Italtel spa, Milano), musical signals (Bontempi-Farfisa spa, Colleferro) and quality control of denim fabrics (Legler Tex spa, Bergamo).

Several prototypes based on PC workstations and DSP cards have been fully designed and carried out in our laboratories. A number of DSP-based systems have also been realized and dissemination activity carried out (see Appendix).

Acknowledgements

The work was partially supported by the RACINE-S (IST 2001-37117) and the MUSCLE EU (FP6-507752) projects. The authors would like to thank Diego Gonzales for the algorithm modeling and the team leading; Federico Fontana and Lorenzo Grassi for the MATLAB®-to-Simulink® translation and implementation.

References

1. Kain and M. W. Macon: "Spectral voice conversion for text-to-speech synthesis," Proceedings of International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 285–288, 1998.
2. A. Kain and M. W. Macon: "Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction," Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2001.
3. Y. Stylianou, O. Cappé, and E. Moulines: "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, March 1998.
4. N. Ivahashi and Y. Sagisaka: "Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks," *Speech Communications,* vol. 16, pp. 139–151, 1995.

5. D. Bonsi, C. Drioli, F. Fontana, D.L. Gonzalez and D. Stanzial, Deliverable 3.2.1 RACINE-S, "Algorithms for dynamic control of sound synthesis and implementation of acoustic ambience extraction from tracks" (2003).
6. J. Cartwright, D.L. Gonzalez and O. Piro: "Nonlinear dynamics of the perceived pitch of complex sounds," *Physical Review Letters,* vol. 82, n. 26, pp 5389–5392 (1999).
7. J. Cartwright, D.L. Gonzalez and O. Piro, "Pitch perception: A dynamical-systems perspective," PNAS, vol. 98, n.9, pp. 4855–4859 (2001).
8. G. Bertini, L. Di Giovannantonio, D. Gonzales, L. Grassi, M. Magrini , M. Dedola: "Sistema di sviluppo con DSP floating-point per il rapid prototyping," Nota interna ISTI-CNR, B4-14, 2004.
9. G. Bertini, F. Fontana, D. Gonzalez, L. Grassi, M.Magrini. "Voice transformation algorithms with real time DSP rapid prototyping tools," Technical report ISTI-CNR, TR-70, 2004.

Appendix

Some DSP systems developed and realized at Signals & Images Lab. of the ISTI (ex IEI) - CNR (*)

- Leonard 'C25 PC cards, with one TMS320C25 DSP and one 14-bit codec. Option: external Hi-Fi stereo interfaces AES/EBU. Used for DSP educational and active noise control (1990–93).
- SeeboarD TI 'C25-50 PC cards, with on board Hi-Fi stereo delta Sigma codec, 16-bits, 44.1 KHz used for ADPCM voice compression at Italtel SPA, Milan (1995).
- Multiprocessor DSP systems with some 'C50 and one 'C40 Loughborough PC boards, for Tissue Quality Control (SEED Srl - Massa, and Legler SPA - Bergamo – European Bryte Project, 1995).
- SeeXover board with 2 DASP57002 and one TMS320C26 DSP, linked to PC via RS-232, for crossover and audio effects.
- Floating-point TMS320C31 DSP unit, used as accelerator for SeeboarD card; MultiDSP' C31 (1996–1997).

(*) Firmware and application software on the above-mentioned systems were mainly developed by SEED Srl, (SEED srl has been a third party of Texas Instruments).

Educational Activity on DSP

- Organization of the Course "Digital Signal Processing: Real-Time Hardware Design" jointly with a teacher of the ICS School of London, Consorzio Pisa-Ricerche, 19–22 June 1989.
- Lectures on "Leonard 'C25 DSP card architecture," course in "Tecnici Elettronici nella Elaborazione Digitale di Segnali" Leonardo spa, Massa (1993).
- Lectures on "Elaborazione digitale di segnali" and "Trattamento del segnale vocale" course "Telecomunicazioni numeriche" e "TV-SAT e GSM" Prov. di Pisa, (Aprile-luglio 1997).
- Lectures on "Sistemi per l'elaborazione digitale dell'informazione audio" Dip. di Neuroscienze Università di Pisa, Laurea di 1° livello in Audiometria e Audioprotesi, since 2002.

References on DSP Systems Realized at IEI/CNUCE-CNR:

1. G. Bertini, L. Tarabella: A DSP system and a Graphic Editor for audio synthesis algorithms. Proceed. Int'l Computer Music Conference ICMC '89, (Columbus Ohio, USA, 1989) pp. 312–315.
2. G. Bertini, L.M. Del Duca, M. Marani: "The Leonard C25 System for Real-Time Digital Signal Processing," Proceed. Int'l Workshop in Man Machine Interaction in Live Performance, (Pisa, 1991), pp. 107–117.
3. L. Bedini, G. Bertini, A. Ribolini: "Abbattimento attivo del rumore mediante l'uso di una scheda basata sul TMS320C30," Internal Report IEI-CNR, B4-24, 1993.
4. G. Bertini, M. Marani: "Methodology and Digital Systems for Electroacoustic Applications," Proceed. 2nd Int. Conf. on Acoustic and Musical Research CIARM95, (Ferrara, 1995), pp. 189–194.
5. G. Bertini, L. Leodori, M. Marani, V. Mazzacca: "AUDIOLab – una stazione per audiometria sperimentale in ambiente Windows," Atti del XXIV Congr. Naz. AIA – Associaz. Italiana Acustica, (Trento, 1996), pp. 461–464.
6. L. Tarabella, G. Bertini, T. Sabbatini: "The Twin-Towers: A remote sensing device for controlling live-interactive music," Proc. Int. Workshop M.C.P.A. Feb. 1997, Scuola Sup. di Studi Univ. S. Anna, Pisa (SSSUP), pp. 41–46.
7. F. De Bernardinis, R. Roncella, R. Saletti, P. Terreni & G. Bertini: "An Efficient VLSI Architecture for Real-Time Additive Synthesis of Musical Signals," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol.7:1 March 1999, pp. 105–110.

8. G. Bertini, V. Di Salvo, W. Gaffi, R. Neri, A. Rocchi: “Microprocessori Per DSP. Modelli recenti e tools di sviluppo – Alcune applicazioni nel settore telecom,” Internal Report ISTI-CNR, B4-10, Pisa, 2003.