A Real-Time DSP-based Reverberation System with Computer Interface

Fernando A. Beltrán, José R. Beltrán, Javier Arregui, Belén Gracia
Department of Electronic Engineering and Communications
University of Zaragoza, Spain
beltran@posta.unizar.es, http://www.unizar.es

Abstract

This paper describes a highly versatile, low-cost reverberation system comprising two main elements: a computer for building and editing the desired reverberation effect impulse response, and a commercial DSP-based board, to run the algorithm in real-time, allowing the evaluation of the results. The main parameters of the reverberation algorithm can be modified by means of a dedicated graphic interface in the host computer.

1 Introduction

One of the most important acoustic effects in a listening experience is the contribution to the sound of the physical space where the source and the listener are located. We typically perceive not only the original sound, but a lot of reflected sound due to the room reverberation [1]. Reverberation creates an ambient space in the perception of the listener.

Thus, we are used to hear the sound with some acoustic characteristics introduced by the surrounding space, and it is a classical effort to reproduce the effect of different auditoriums for sound recording, sound listening or living performances.

We present here an experimental system where it is possible to work with a computer based graphic interface in order to define a virtual room, the walls, ceiling and floor sound reflection percentage, to inspect the resulting amplitude-time diagram of the impulse response of the room, to modify amplitude and distance in the early reflections, and reverberation time and density in the continuous reverberation. Once the desired reverberation is defined, it can be heard in a real time DSP based unit; this unit may be connected to the computer for loading the reverberation parameters, or it can work in stand-alone mode.

This paper is organized as follows: after the introduction we present in section 2 the reverberation algorithms used for this work; in section 3, the user interface developed for a PC platform is explained; section 4 presents the commercial DSP-based board used to run in real time the reverberation algorithm, and finally we present some conclusions, further work and references.

2 Reverberation Algorithms

If we create a virtual listening space with a virtual sound source and a listener in it, and different obstacles, we should be able to simulate the reflections that affect the progress of the sound waves from the source to the listener.

We would then had computed the impulse response corresponding to that virtual space (*Figure 1*). However, for a general shaped space with arbitrary obstacles, this calculation involves the execution of a rather complex and time-consuming ray-tracing algorithm [2].

Because we are looking for a specialized tool designed to develop reverberation effects, rather than a real architecture acoustic simulation, a first approach to generate a reverberation pattern is to work with a simple parallelepipedic-shaped room. In this case, the calculation is highly simplified, and the impulse response is quickly obtained by means of the source image method.

Once the virtual space has been characterized, our objective is to digitally process the source sound in order to add a reverberation effect similar to the one

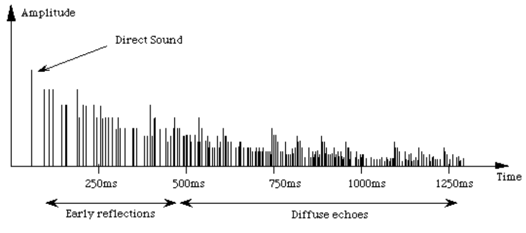

Figure 1. Typical impulse response of a reverberating ambient.

obtained if we actually were inside the room we have designed.

If we had large memories and faster processors we could just implement a long impulse response filter with so many taps as reflections we wanted to consider, so that the effect will exactly match that of the designed room.

Unfortunately this approach has to be discarded because it would produce very poor reverberation effects if implemented with actual Digital Signal Processors. Thus, one more time this result encourages to use simplified room modeling, because, even if we had an exact impulse response description of a very complicated room, the final DSP algorithm would not be able to count for the minor details of individual reflections. Anyway, in most cases our ears wouldn't either. To dramatically increase the number of repetitions, feedback must be used (this leads to definition of Infinite Impulse Response filters).

Two main simple IIR filters are used in the reverberation algorithms we present here. In a comb filter a sample is delayed and fedback to the input, producing a series of echoes. In an all-pass filter, feedback and forward paths over a delay line are used, producing the effect of many reflections and a flat frequency response.

Schroeder [3][4] designed a reverberation algorithm using comb filters disposed in parallel and all-pass filters disposed in series. However, this reverberation system can still be improved to increase echo density and frequency response. Gardner and other researchers of the Massachusetts Institute of Technology [5] proposed three more sophisticated reverberation algorithms (for small, medium and large rooms) based on nested and cascaded all-pass filters with an outer comb and low-pass filter. They will serve as a starting point for our design. These IIR filters simulate the dense echo response of the reverberation. However we have also included a FIR filter to render some early reflections before the dense rumor appears [6]. This early echoes provide information about the geometry of the room that is lost as time increases.

The general structure of the reverberation algorithm implemented for our system is shown in *Figure 2*. Input audio samples are organized in a circular buffer. The first part of the algorithm (left hand in the picture) is a FIR filter to produce the early reflections. The low-pass filter in this area is to simulate the air absorption and the effect (filtering) of the walls over the high frequencies. In this part, there is also a linear filter to "feed" the IIR block, necessary to generate a smooth transition from the early reflections to the diffuse reflections area. The second part of the algorithm (right hand in the picture) is the IIR filter (based in nested all-pass filters) and a feedback loop with a low-pass filter (comb filter), to produce the dense echo effect.

For the system presented here, three complete reverberation algorithms (for small, medium and large rooms) have been optimized to run in real time using the Motorola DSP56007 [7].

3 User Interface

The computer based graphic interface is a PC program developed in Visual Basic.

This application allows the user to design a virtual simple geometry room (a parallelepipedic room) with assignable dimensions, set the amount of sound reflection for each wall as well as for the floor and ceiling, locate a virtual sound source and a listener in the 3D space within the room and calculate the real impulse response of the room by means of the source image method.

After that, the application allows on screen viewing of the impulse response file obtained in the previous step. By simply inspecting the impulse response, the user may decide to go back to the virtual room screen to modify some parameters, or to generate a reverberation algorithm to match this impulse response. In this case, the program characterizes this impulse response through the reverberation time, sound intensity decay and echo density evolution. It then suggests the DSP algorithm that best matches the above mentioned parameters. The selection of the algorithm relies on the following set of steps.

1) An IIR reverberation filter is chosen. The possible implemented choices are Small, Medium and Large room based on nested and cascaded all-pass and comb filters.

2) The standard filter is time-scaled to match the echo density pattern of the virtual pre-defined room.

3) The outer feedback gain is chosen so that the reverberation time and intensity decay conditions are also met.

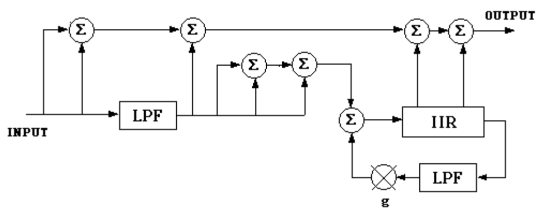

Figure 2. Reverberation algorithm block diagram.

4) A number of early echoes are added to the algorithm by means of a FIR filter. By default, these early echoes are set to be equal to those of the pre-defined room, with the following exceptions:

- If a reflection in the virtual room is below a minimum intensity level, it is not taken into account.
- If two reflections are very close together, they are rendered as a single one with equivalent intensity.

To adapt the generic reverberation algorithms described in the previous section to the desired one, a vector of 68 parameters is used. They serve to define, for instance, the cut-off frequency and position of the input low-pass filter, the brightness (cut-off frequency of the outer feedback loop), amplitude and position of the early reflections, reverberation time, etc.

Once the algorithm is calculated, the user is allowed to change all the suggested parameters and even set individual position and amplitude of each one of the early echoes rendered. It is done with the help of the screen shown in *Figure 3*. The final set of parameters may be saved in a file for further retrieval.

To listen to the results of the final reverberation effect defined by the user, we use a commercial board from Sample Rate Systems, based in the Motorola DSP56007. The computer is connected via RS 232 to a development system (LINK 56004) from Domain Technologies. Inc. [8], which comprises a hardware box and an associated software. This system is connected to the DSP board through the Motorola OnCE (On Chip Emulator) port. The user interface program described in this section makes the development system software transparent for the user, so the vector of parameters used to define the DSP reverberation algorithm are directly sent to the DSP and the user does not need exiting the main program window.

4 Sample Rate Systems Board

The Sample Rate Systems model M4-0202-A [9] is a stand-alone digital signal processing module for prototyping and small scale production of digital audio equipment. This board has two audio preamplifiers, one stereo 18-bit A/D converter, digital signal processing section based on Motorola DSP56004/7 processor, one stereo 18-bit D/A converter and two-channel digitally controlled output level control.

The board accepts either differential or single-ended inputs. Differential inputs are connected to a type of connectors which accept both XLR and 6.35 mm jack male plugs. Single-ended inputs are connected via RCA-type connectors. The selection between differential and single-ended inputs is done with a jumper.

The differential pre-amplifiers are designed using Analog Device's SSM-2017 integrated audio pre-amplifier. The amplifier accepts either single-ended or differential inputs. The gain of the amplifier is controlled with a trimmer potentiometer, one per channel. The gain range is from 0 to 60 dB.

The A/D conversion is done using a Crystal Semiconductor's CS5389 converter. This chip is a complete analog-to-digital converter for stereo digital audio systems. It performs sampling, A/D conversion and anti-aliasing filtering, generating 18-bit digital words for both left and right inputs in serial form. The output word rate can be up to 50 kHz per channel.

The CS5389 uses 5th-order, delta-sigma modulation with 64x oversampling followed by digital filtering

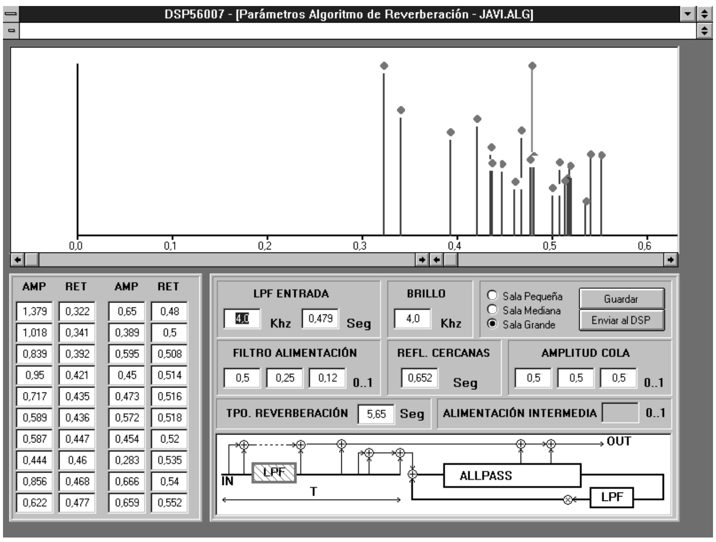

Figure 3. Screen for reverberation parameters setting.

and decimation, which removes the need for an external anti-alias filter. The ADC uses a differential architecture which provides excellent noise rejection.

The D/A converter used is Crystal Semiconductor's CS4328 converter. This chip is a complete stereo digital-to-analog output system. It includes an 8x digital interpolation filter followed by a 64x oversampled delta-sigma modulator. The modulator output controls the reference voltage input to a linear analog low-pass filter. This architecture allows the adjustment of sample rate between 1 kHz and 50 kHz while maintaining linear phase response simply by changing the master clock frequency.

The CS4328 includes a serial port, utilizing two select pins to support four different interface modes with the processor, accepting either 16 or 18-bit input data.

Following the D/A converter there is a digital volume control unit, implemented with the Crystal Semiconductor's CS3310. This chip has two independent, low distortion volume controls. The control is done by the DSP via the chip's serial port. Although the control of the volume is done digitally, the actual control function happens in the analog domain.

The control has a step size of 0.5 dB and ranges from 95.5 dB attenuation to 31.5 dB gain (-95.5 dB to +31.5 dB).

This board has both differential and single-sided outputs.

Program memory is provided as a 28 pin standard memory socket. The socket accepts most EPROMs, EEPROMs and non-volatile SRAMs IC's on the market, and can be used with memory IC's up to 32 kbytes.

Data memory capacity is provided on a standard 30 pin SIMM socket. The socket can accommodate standard dynamic SIMM modules up to 4 Mbytes. Both 8 or 9 bit modules can be used (on 9-bit modules, the 9th bit will be unused). The memory interface of the DSP can be programmed in the software according to the speed of the memory module used. In addition, the refresh of the dynamic memory is automatically handled by the DSP.

The module can be expanded externally by an expansion bus connector. The bus provides access to most of the connections of the DSP. In addition, there is 32 kbytes of RAM-type I/O space.

5 Conclusions

In this work we have presented an integrated development system oriented to the generation of high quality reverberation algorithms. The starting point is a simple geometry virtual room design, and the computed reverberation pattern is characterized by 68 different parameters. The user may experiment unlimited reverberation patterns by changing those parameters and listening to the results in real time, with the help of a DSP based commercial board.

Because the complexity of the reverberation algorithm implemented in this system, its main limitation is the lack of stereo signal processing capability using only one DSP.

Further work will consist in the implementation of a set of real time audio effects algorithms for the Motorola DSP56007 (chorus, flangers, filters, delays, etc.), with a computer graphic interface, so the user may configure their parameters and link them on the screen, with automatic generation of the corresponding DSP code, and listening to the results using the DSP board.

References

[1] Sabine, W.C. “Reverberation”, originally published in 1900. Reprinted in 1972, in R.B. Lindsay Editor, *Acoustics: Historical and Philosophical Development*, Dowden, Hutchinson and Ross, Stroudsburg, PA.

[2] Borish, J. 1984. Electronic Simulation of Auditorium Acoustics. Ph. D. Thesis, Center for Computer Research in Musical Acoustics, Dpt. of Music, Stanford University, CA.

[3] Schroeder. M. R., Logan, B. F. 1961. Colorless Artificial Reverberation. J. Audio Engineering Society. Vol. 9, No. 3.

[4] Schroeder. M. R. 1962. Natural Sounding Artificial Reverberation. J. Audio Engineering Society. Vol. 10, No. 3.

[5] Gardner, W. G. 1992. The Virtual Acoustic Room. Master’s Thesis. Massachussetts Institute of Technology.

[6] Beltrán, F. A., Fernández de Velasco, D. 1994. A DSP-Based Reverberation System with Graphic Interface. Proc. International Conference on Signal Processing Applications and Technology, Dallas, TX, pp.108-113.

[7] Motorola. 1992. DSP56000 Digital Signal Processor Family Manual.

[8] Domain Technologies, Inc. 1994. Link-56004 Manual. Development System for Motorola's DSP56004.

[9] Sample Rate Systems Oy. 1995. User's Manual M4-0202-A. Stand-Alone Audio DSP Module.