RIO 2005 inter·noise ENVIRONMENTAL NOISE CONTROL
The 2005 Congress and Exposition on Noise Control Engineering
07-10 August 2005 - Rio de Janeiro - Brazil

An Open-Source Platform for Musical Room Acoustics Research

Leo Kazuhiro Ueda[a], Fabio Kon[b], Fernando Iazzetta[c]

[a,b]Department of Computer Science, University of São Paulo, Brazil
[c]Department of Music, University of São Paulo, Brazil

[a]lku@ime.usp.br, [b]kon@ime.usp.br, [c]iazzetta@usp.br
http://gsd.ime.usp.br/acmus

Abstract AcMus is an ongoing project to develop a software for estimation, measurement, analysis, and simulation of rooms specially designed for musical performance. The system under development is organized in modules. All modules shall be integrated in a way as to allow seamless flow of data regardless of the variety of data types and structures involved in the system. Portability and accessibility of the software developed have been major concerns in the implementation of the system and, for this reason, it has been developed as an open source project, so that users and software developers from different groups can collaborate with its further improvement. The measurement system is based on the sine sweep method and provides tools for the acoustic optimization of rooms designed for musical performance. It also provides a toolbox to help with the analysis and design of environments such as music halls, lecture rooms, etc. for the specific use of music appreciation. The system core is being implemented in Java, with great care to make it platform neutral. The implementation is based on the sophisticated Eclipse Platform (www.eclipse.org), which will facilitate the system's extensibility and flexibility, enabling the construction of an open platform for Acoustics experimentation, freely available to researchers in the field.

1. INTRODUCTION

In the last few years the research in the field of acoustics has experienced an expressive growth in Brazil. Although the research groups are mainly interested in issues related to environmental comfort and vibration and noise control, there is also an increasing concern about the acoustic qualities offered by spaces designed for music production and diffusion such as recording studios, movie theaters, and lecture halls. Many companies and professionals are getting specialized in room acoustics, but usually their activity is based on the use of imported standard solutions, which are not necessarily well adapted to our climatic, economic, and socio-cultural context.

In 2001, we created a research group in acoustics at the University of São Paulo with the aim of gathering researchers interested in the connection between music and acoustics. One year later we decide to formalize a project on room acoustics called AcMus in collaboration with individuals from different fields, such as Music, Architecture, Engineering, Physics, and Computer Science. Our main purpose is to start a regular work on musical acoustics and to combine the efforts of many individuals with similar interests.

The core of the AcMus project is the implementation of a software for the analysis, simulation, and optimization of acoustic parameters for small and medium sized music rooms [1]. The development of the software has been guided by some specific concerns. First, since we intend to promote the exchange of the results of our research with other groups, the software is completely open source. In the same direction, we are trying to use, whenever possible, source code and algorithms that are already made available by other research groups. The Eclipse Platform (see Section 2.1) will greatly help us to accomplish this. Second, the system must not be limited to a single platform. For this reason, the software has been developed mostly in Java so that it can run easily on the Linux, Solaris, MacOS, and Windows platforms. If the need arises, specific computationally-intensive functions can be implemented in C++ for performance reasons.

In this paper, we describe the current implementation of the AcMus software, more specifically, the measurement system implemented so far and the graphical interface offered to the user.

2. THE ACMUS INTEGRATED PLATFORM

We aim to build an open source software environment that integrates different tools for acoustic analysis, simulation, and optimization. This environment is divided in three main modules.

1. **Measurement Module:** it provides a graphical interface that helps the user to perform and organize acoustic response measurements taken from various rooms. Using different methods, it also calculates the room's impulse response (IR) and a large number of parameters related to the room acoustic behavior. We investigated two methods for acquiring the IR. One of them is the MLS (Maximum-Length Sequence) method [2, 3], which uses a signal based on a periodic sequence of randomly distributed positive- and negative-going impulses with a flat energy distribution in the frequency domain. The MLS is largely used in many acoustic measurement systems. The other method, which we call LSF (Log Sweep FFT) [4, 5], is a more recent development and can be understood as an evolution of the TDS method [6] and uses a logarithmic sweep as a excitation signal. These methods offer a much better signal-to-noise ratio in comparison to other methods that use mechanically produced impulses. Besides, numerical methods such as MLS, TDS and LSF allow for the repeatability of tests since they work with an stable and previously known signal. The Measurement Module has already been implemented as a Matlab prototype and has been successfully tested in acoustic measurements of different rooms at the University of São Paulo and in several theaters in the city of São Paulo.

2. **Utilities Module:** it is an acoustic tool box that offers functions that can be useful for the design of rooms, acoustic measurement, and audio processing. For example, these

functions include the generation of different types of audio signal, filtering and spectral analysis, and also a function to help the design of Schröder diffusors [7, 8, 9, 10] and low frequency absorbers. Many tools have already been implemented as independent Java applications and will now be included in the AcMus integrated platform.

3. **Simulation and Optimization Module:** this complex module will help the design of rooms by performing computer simulations on given models. Optimization techniques will be applied to allow the automatic search of solutions, which, at the same time, satisfy all constraints and can be shown to be the best ones available for a set of given criteria. This module is still in its initial stage of development. Two well known acoustic simulation methods will be implemented: ray tracing and image source, both based on geometrical acoustics. These two methods will be used in association. The software will provide the impulse response for the simulated rooms as well as the acoustic parameters calculated from this response. Also, we intend to build a module that convolves binaural impulse responses with anechoic audio for auralization.

2.1 Implementation

The current implementation of the AcMus Platform is almost entirely focused on the Measurement Module. In fact, even the implementation of tools for the Utilities Module is being guided by the needs of the Measurement Module.

In a measurement, a microphone and a loudspeaker, representing a listener and a sound source, are placed at chosen positions inside a room. The software then feeds the loudspeaker with a signal, which may differ depending on the method applied. At the same time, the microphone captures the room's response to this sound. After that, the software processes this recording to calculate parameters that describe the room's acoustic characteristics for that particular position of listener and source. Usually, the measurements are taken in a large number of positions and repeated several times.

A major concern of the implementation effort has been the quality of the user interface. We seek to ease the tedious job of taking countless measurements from a room. In the next sections, we show the stage of development.

The Eclipse Platform

Eclipse (`http://www.eclipse.org`) is a powerful, generic, extensible, open set of computer tools for developing programs. It is an "*IDE* (Integrated Development Environment) *for anything and yet nothing in particular*."

It has the same basic purpose of any other IDE (for example, Visual Studio, Delphi, and NetBeans), which is to help computer programmers build software by integrating development tools in a single environment. But Eclipse actually goes beyond, it can be largely extended with contributed plug-ins written in Java. It provides a foundation for constructing and running these plug-ins, allowing extensions built by different people to integrate seamlessly and become part of the Eclipse Platform.

The AcMus Software is being implemented as a set of Java plug-ins integrated into the Eclipse Platform. Applications built this way usually end up having an IDE look, partly because of the reuse of the Eclipse plug-in infrastructure, as one may notice from the following sections. The new Eclipse Rich-client Platform would allow us to have a more generic look but it prevents us from reusing some useful plug-ins.

Utilities

We built some tools to support the Measurement Module that may be suitable for other purposes. Therefore, they are part of the Utilities Modules. Figure 1 shows the *Audio Player*, which draws the waveform and the spectrum of an audio file. We are also implementing a DSP library that currently has a few FFT and filtering functions.

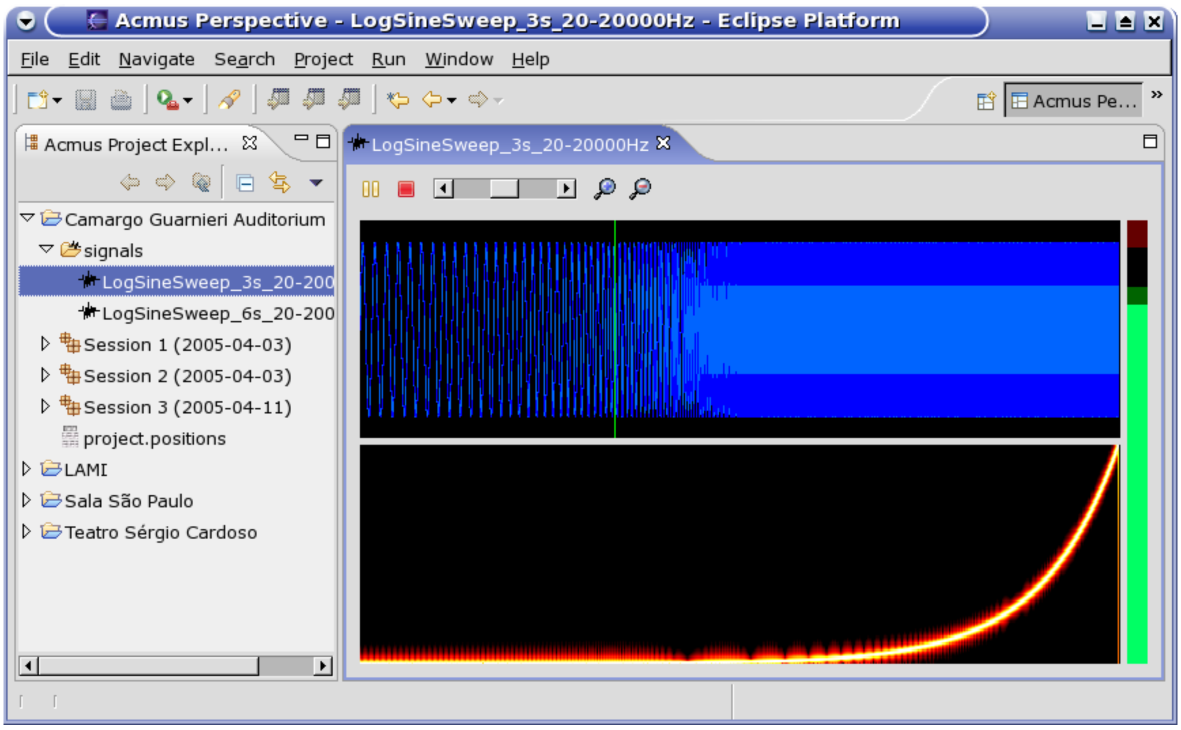


Figure 1: The Audio Player.

Measurement Manager

The Measurement Module takes advantage of the Eclipse Resources Plug-in to let the user store measurements taken from different rooms in hierarchical folders. Figure 2 shows the kinds of folders that induce a certain organization.

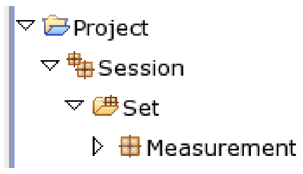


Figure 2: The hierarchical folders that store measurements.

The Project folder represents a room. The Session folder groups measurements taken at a specific period in time. The Set folder stores the repetitions of the same measurement. The Measurement folder stores the audio file of the room's response and the output of the response analysis. Figure 3 shows an actual screen shot.

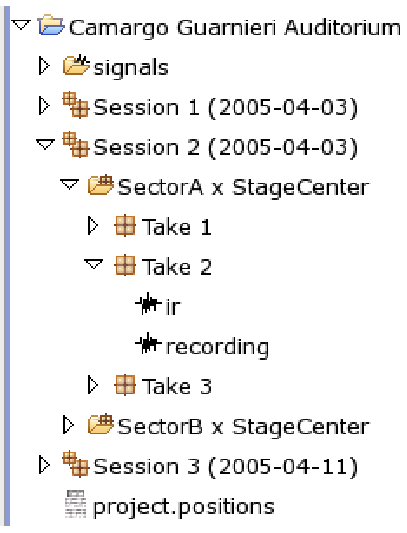
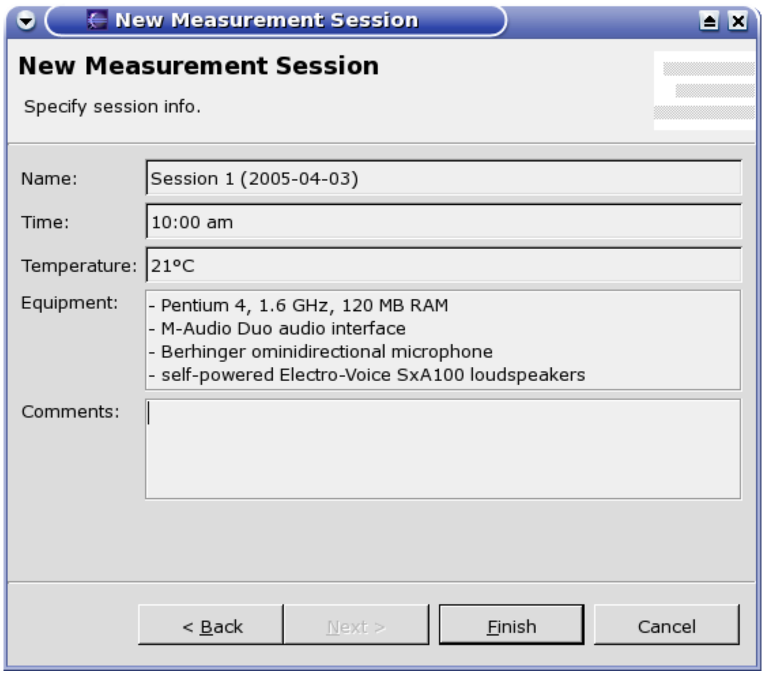


Figure 3: An example of the organization of the folders.

The Measurement Module provides wizards for the creation of folders. There is one wizard for each kind of folder, but the process of creation does not vary much. In all cases the user may input additional information about the folder such as date, time, equipment, comments, and so on. Figure 4 depicts the wizard for the creation of a Session folder.



New Measurement Session

New Measurement Session

Specify session info.

Name: Session 1 (2005-04-03)

Time: 10:00 am

Temperature: 21°C

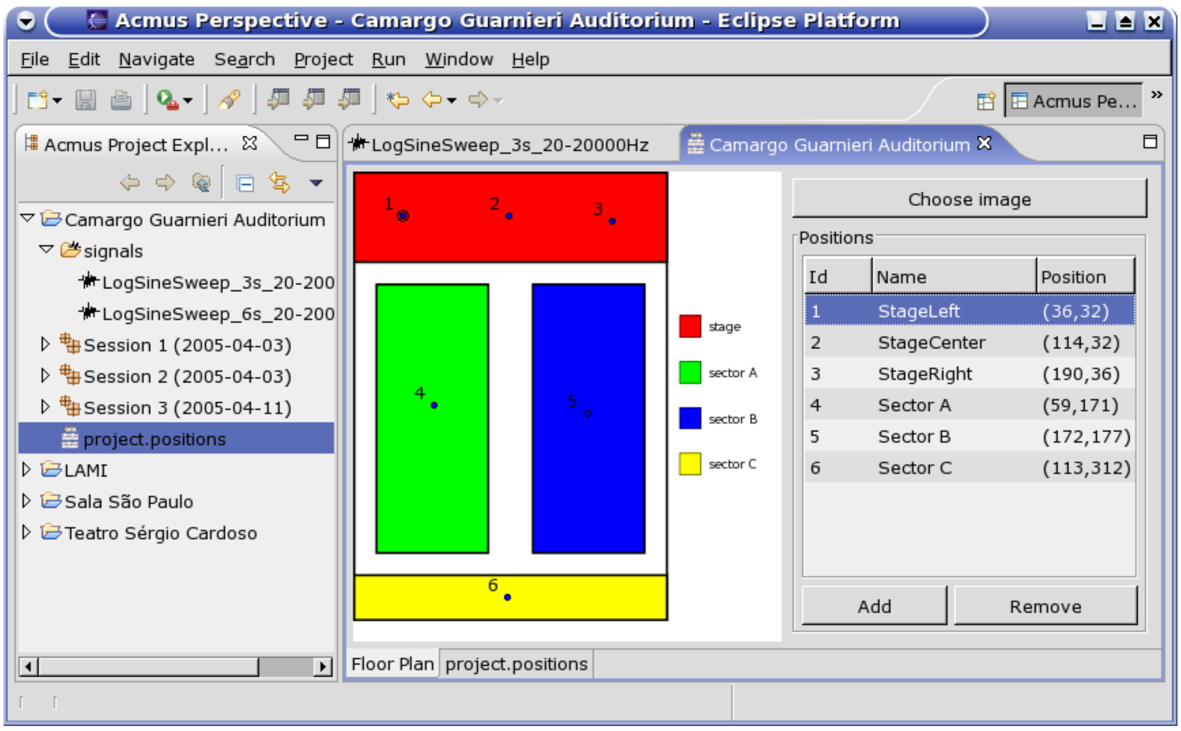
Equipment:
- Pentium 4, 1.6 GHz, 120 MB RAM
- M-Audio Duo audio interface
- Berhinger ominidirectional microphone
- self-powered Electro-Voice SxA100 loudspeakers

Comments:

Figure 4: The Session creation wizard.

Position Editor

Inside each project folder there is a file called project.positions. This file keeps a list of possible positions for the microphone and the loudspeaker. It is a text file, but the software provides a graphical editor, which is shown in Figure 5.



Id	Name	Position
1	StageLeft	(36,32)
2	StageCenter	(114,32)
3	StageRight	(190,36)
4	Sector A	(59,171)
5	Sector B	(172,177)
6	Sector C	(113,312)

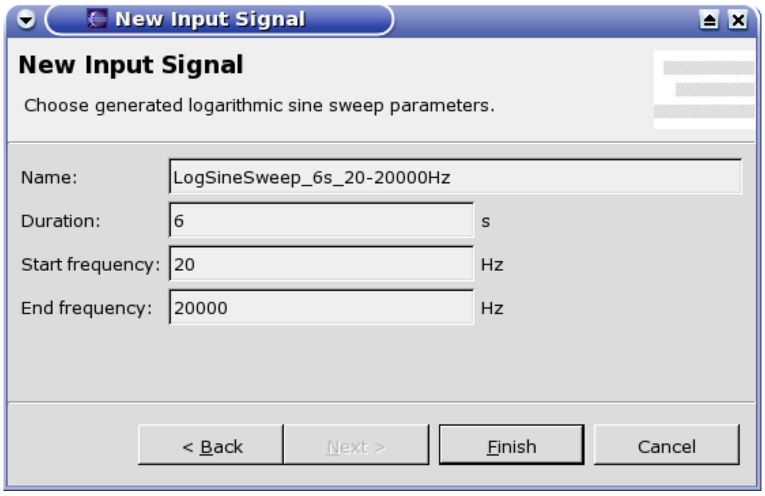
Figure 5: The Position Editor.

In this interface, the user may choose an image to represent the floor plan of the room. The positions are created as points plotted on top of this image. It is important to state that all this information is for documentation purposes only, it helps the user in the managing of the measurements.

Input Signal Manager

There is also a special folder inside the project folder called signals. Inside this folder the user may create and keep the input signals that will be played by the loudspeaker during measurements. The input signal can be of one of two types: a signal that is generated by the software or a given audio file. Currently, the only type of signal that the software generates is the logarithmic sine sweep. Figure 6 shows the wizard for the creation of this type of signal.

The user needs to choose the initial and ending frequencies, as well as the total duration of the sweep. The signal will only be generated when the user selects it in the Measurement Interface, it will not be generated in real time during the measurement nor it will be stored in the hard disc.



New Input Signal

New Input Signal

Choose generated logarithmic sine sweep parameters.

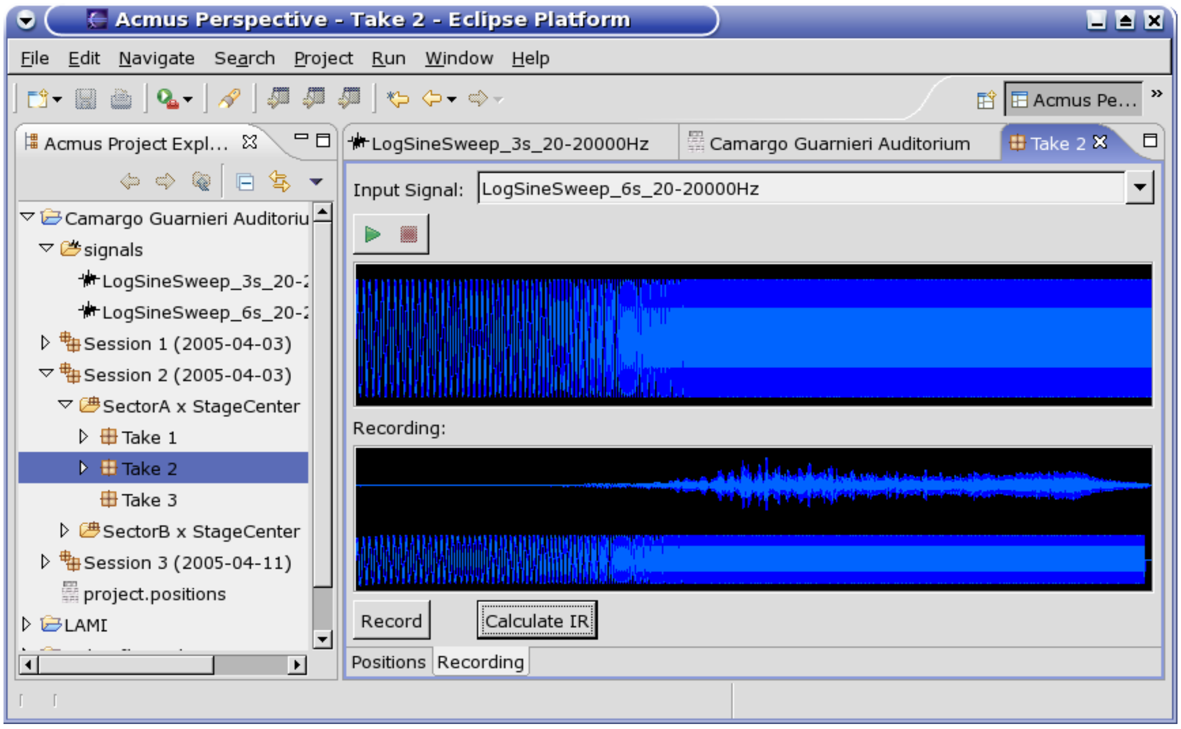
Name:	LogSineSweep_6s_20-20000Hz	
Duration:	6	s
Start frequency:	20	Hz
End frequency:	20000	Hz

< Back | Next > | Finish | Cancel

Figure 6: The Signal Wizard.

The Measurement Interface

A double-click on a measurement folder opens an interface for the management of this measurement. The interface, shown in Figure 7, allows the user to perform a measurement, that is, to play the chosen input signal and record the room response.



Acmus Perspective - Take 2 - Eclipse Platform

File Edit Navigate Search Project Run Window Help

Acmus Project Expl...

- Camargo Guarnieri Auditoriu
  - signals
    - LogSineSweep_3s_20-2
    - LogSineSweep_6s_20-2
  - Session 1 (2005-04-03)
  - Session 2 (2005-04-03)
    - SectorA x StageCenter
      - Take 1
      - Take 2
      - Take 3
    - SectorB x StageCenter
  - Session 3 (2005-04-11)
  - project.positions
- LAMI

LogSineSweep_3s_20-20000Hz | Camargo Guarnieri Auditorium | Take 2

Input Signal: LogSineSweep_6s_20-20000Hz

Recording:

Record | Calculate IR

Positions | Recording

Figure 7: The Measurement Interface.

After that, by clicking on the “Calculate IR” button, the user asks the software to calculate the impulse response from the captured sound. The result is shown in Figure 8.

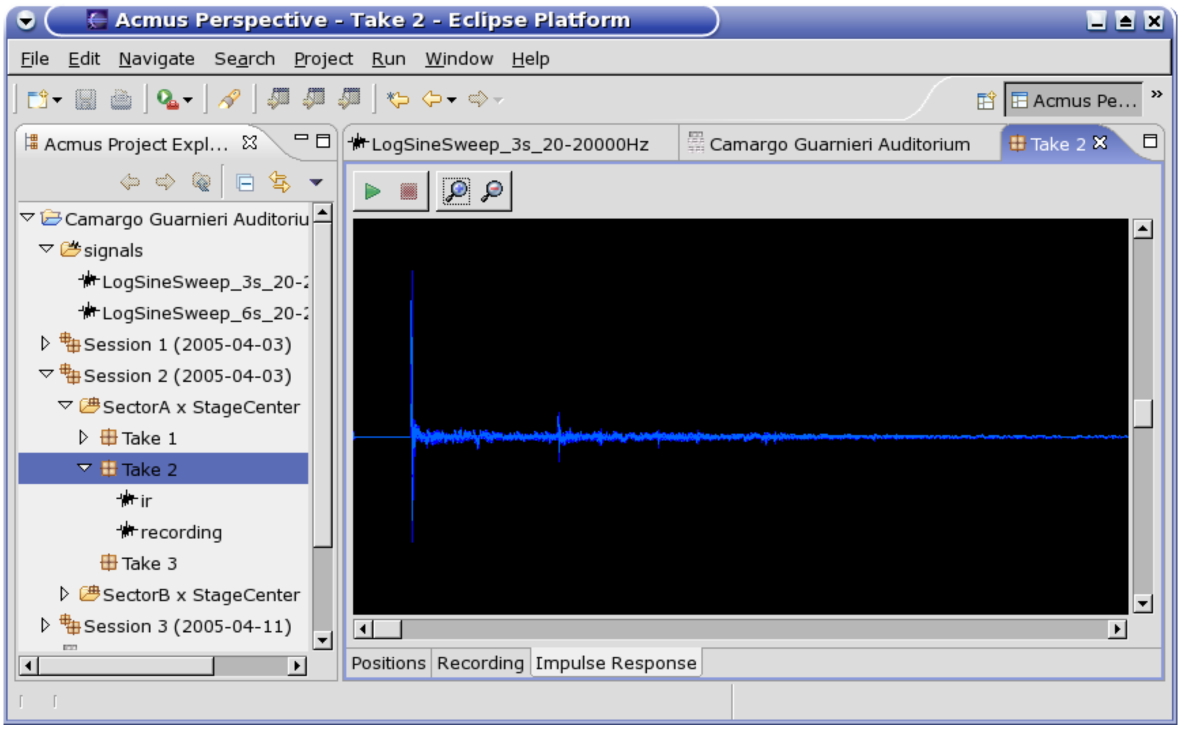


Figure 8: The calculated impulse response.

3. CONCLUSIONS AND FUTURE WORK

We showed the implementation of part of the AcMus software stating the importance of it being open, extensible, and platform-independent. There is clearly a lot of work to be done, but working source code is already available at the project's Web site: `http://gsd.ime.usp.br/acmus`.

One important missing feature is the automation of the measuring process. We plan to improve the Measurement Interface (Figure 7) so that it allows the user to issue a single command that performs all the measurement repetitions for the chosen microphone-loudspeaker positioning.

Besides that, of course, we need to implement the processing of the impulse response. First, the obtained IR needs some treatment in order to look more like an ideal IR. We will offer the methods already implemented in our Matlab prototype: Lundeby [11], Chu [12], and Hirata [13]. After this treatment, the software will be able to calculate the acoustic parameters: reverberation time (RT), early decay time (EDT), sound strength (G), clarity (C), definition (D), center time (T), initial time-gap delay (ITGD). The output will look like Table 1, which shows the parameters generated by our Matlab prototype for the Camargo Guarnieri Auditorium.

ACKNOWLEDGEMENTS

This research is financially supported by FAPESP, the São Paulo State Research Agency, proc. #02/02678-0. The AcMus project has also the support from Roland Brazil.

Table 1: Acoustical parameters of the Camargo Guarnieri Auditorium.

Parameters	Frequency [Hz]										
	63	125	250	500	1000	2000	4000	8000	A	C	Linear
C50 [dB]	-10.880	-16.110	-11.870	1.730	3.290	3.560	5.300	8.730	5.350	4.330	5.120
C80 [dB]	-1.970	-1.060	-2.170	5.140	5.680	5.690	7.640	10.690	7.650	6.980	7.650
D50 [%]	7.550	2.390	6.110	59.850	68.080	69.440	77.230	88.190	77.430	73.050	76.470
D80 [%]	38.840	43.940	37.760	76.570	78.730	78.780	85.320	92.150	85.330	83.300	85.330
CT [ms]	112.950	97.000	124.070	60.780	57.240	52.480	37.340	21.310	38.350	44.080	38.840
EDT [s]	1.015	0.807	1.225	0.963	1.174	1.142	1.128	0.798	1.062	1.049	1.016
T20 [s]	1.234	1.128	1.178	1.203	1.213	1.255	1.137	0.897	1.134	1.147	1.119
T30 [s]	1.377	1.184	1.229	1.267	1.306	1.323	1.139	0.893	1.186	1.212	1.181
T40 [s]	1.377	1.237	1.245	1.309	1.422	1.364	1.137	0.857	1.248	1.271	1.226

Bass ratio	0.954
Treble ratio	0.990
ITGD [ms]	11.500

REFERENCES

[1] Fernando Iazzetta, Fabio Kon, Marcelo Gomes de Queiroz, Flávio Soares Correa da Silva, and Marcio de Avelar Gomes. AcMus: Computational tools for measurement, analysis and simulation of room acoustics. In *Proceedings of the European Acoustics Symposium*, Portugal, September 2004.

[2] S. N. Y. Gerges and A. G. Gomes. Modelling of room acoustic parameters using MLS technique and numerical simulation. In *7th International IBPSA Conference*, Rio de Janeiro, Brazil, 2001.

[3] John Vanderkooy. Aspects of MLS measuring systems. *J. Audio Engr. Soc.*, 42(4):219–231, April 1994.

[4] Angelo Farina. Simultaneous measurements of impulse response and distortion with a swept sine technique. In *AES Conference*, France, 2000.

[5] S. Müller and P. Massarani. Transfer function measurements with sweeps. *J. Audio Eng. Soc.*, 49:443, 2001.

[6] Richard C. Heyser. Acoustical measurements by time delay spectrometry. *J. Audio Engr. Soc.*, page 370, October 1967.

[7] Manfred R. Schröder. Diffuse sound reflection by maximum length sequences. *J. Acoust. Soc. Am.*, 57:149–151, 1975.

[8] Manfred R. Schröder. Progress in architectural acoustics and artificial reverberation: Concert hall acoustics and number theory. *J. Audio Engr. Soc.*, 32(4):194–203, April 1984.

[9] Peter D'Antonio and John H. Konnert. The reflection phase grating diffusor: design theory and application. *J. Audio Eng. Soc.*, 32(4):228–238, April 1984.

[10] Peter D'Antonio and John H. Konnert. The Schroeder quadratic-residue diffusor: design theory and application. In *AES 74th Convention*, New York, October 8-12 1983.

[11] A. Lundeby, T.E. Vigran, H. Bietz, and M. Vorländer. Uncertainties of measurements in room acoustics. *Acustica*, 81:344–355, 1995.

[12] W. T. Chu. Comparison of reverberation measurements using Schroeder's impulse method and decay-curve averaging method. *J. Acoust. Soc. Am.*, 63(5):1444–1450, May 1978.

[13] Y. Hirata. A method of eliminating noise in power responses. *J. Sound Vib.*, 82(593–595), 1982.