Active Control of String Instruments using Xenomai

Simon Benacchio, **Robert Piéchaud**
Adrien Mamou-Mani, **Victor Finel**
IRCAM-CNRS-UPMC, UMR STMS 9912
1 Place Igor Stravinsky, 75004 Paris, France
benacchio@ircam.fr, piechaud@ircam.fr

Baptiste Chomette
CNRS, UMR 7190
UPMC Université Paris, UMR 7190
Institut Jean Le Rond d'Alembert, 4 Place Jussieu, F-75005 Paris, France

Abstract

Real time system is a necessary component of an active control implementation. For example, the sound control, which became a real issue in a lot of fields, needs deterministic and fast properties. This research study presents an application using the Xenomai framework in order to control the vibration of a string instrument's soundboard. A control system is designed and applied to this structure with the aim of changing the instrument sound. The acquisition board used to measure and send the signals on the soundboard is linked to a modeling software running under the Xenomai real time framework, ensuring the necessary active control properties.

1 Introduction

Structural vibration is a natural phenomenon which is often considered as a drawback in the current technological issues. Different methods are used to reduce these vibrations. A usual one is the active control of structural vibrations. Its principle is to use a measurement of the disturbing vibrations to find a control signal which is sent on the structure to reduce it. For example this technique is used for the comfort of passengers in air transports. More recently, this control method has been applied to string instruments. In this case, the idea is not to reduce but to modify the vibration of the instrument's soundboard in order to change the sound or the quality of the instrument. A convenient way to apply this method is to use the modal active control [1]. This technique enables the modification of the modal parameters of the instruments' soundboard which are believed to impact the instruments' sound [2].

One of the main point in active control is the time issue. Indeed, to apply an efficient control on a structure, the analog to digital and digital to analog conversion and the control signal computation steps have to be deterministic with an extremely short time period. These constraints match perfectly the real time issues. Several products can be used to develop real time application. One of them are the Linux Real Time systems which offer the abilities to answer to the active control problems. A lot of experimental projects have been carried thanks to the Real Time Linux frameworks like RT-Linux or RTAI [3]. In this study, Xenomai is used to make the control system. This framework not only enables a deterministic computation with low latency, but also allows to take care of the portability and the maintainability of these services [4]. These reasons and the fact that more and more people use the Xenomai framework lead us to use it.

This study proposes the application of an active control method thanks to an experimental control system using Xenomai. Its aim is to use a method called the modal active control to modify the modal parameters of a string instrument's soundboard. After a quick presentation of the modal active control principle, the experimental setup used to apply the control is presented. Then some details of the control system is given. Some tests are done to give characteristics of this system. Finally, experimental results are presented.

2 Active Control and Experimental Setup

For the string instruments' sound production, modal parameters of the soundboard are very important. The modification of these parameters has been studied in many ways and by many people. For example, luthiers try to mechanically modify it, changing the different characteristics of the soundboard. Indeed, the quality and the sound of the instrument change with the wood used to make it, the dimensions of the soundboard or those of the ribs [5]. Scientists also try to modify the modal parameters of the soundboards to understand what is considered as a high quality instrument. They often study the effects of modal parameters thanks to sound synthesis [2]. The modal active control is a convenient way to address these different issues. Indeed, it enables the modification of modal parameters of the controlled structure without mechanical change. In this section, the general principle of modal active control is presented first. Then, a description of the experimental setup used in this study is given.

2.1 Modal Active Control

The modal active control method is based on the state space approach which describes a system using first order equations governing its state variables [1]. Figure 1 presents an example of state space approach feedback loop.

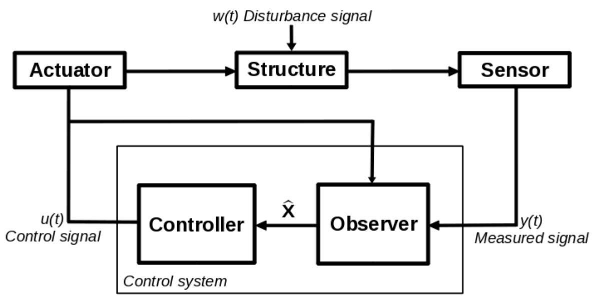

FIGURE 1: *Modal active control feedback loop.*

The structure to control is disturbed by a signal $w(t)$. Sensors are used to give the measured signal $y(t)$ of the structural vibrations. This signal is sent to the control system. The modal state $\hat{\mathbf{X}}$ of the structure is estimated thanks to an observer and used by the controller to give the control signal $u(t)$. This last one is added to the disturbance signal thanks to actuators and involves a new structure dynamics. Hence, the modal properties of the structure can be controlled. If the vibrating structure is a string instrument soundboard, its radiated sound can be modified. It must be noted that the observer is made with a model matching the vibration modes of the experimental structure. The influence of the modes' number in this model is studied later.

2.2 The Simplified Soundboard

Before to apply the control on a real instrument, its effects are studied on a simplified soundboard. The experimental setup used in this study is shown in Figure 2.

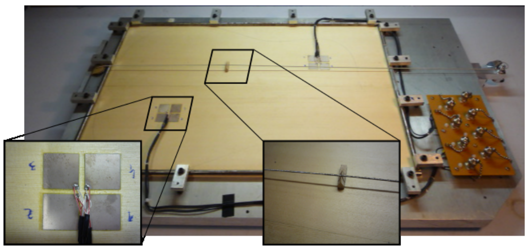

FIGURE 2: *Simplified soundboard used as experimental setup. Zoom on the bridge and on the piezoelectric transducers.*

It consists of a rectangular spruce plate identical to those used by string instruments' makers to make soundboards. Its edges are under clamped boundary conditions. A single string is tied on and connected to the bridge almost parallel with the soundboard plane. The transducers are piezoelectric patches made in PZT-5H and are bonded by group of four side by side.

3 Control System

To apply an active control on this soundboard, different steps are necessary. First one is the computation of the observer and controller gains. This step can be done in a pre-computation step according to the control target. Then these gains are used in the control system to give the signal $u(t)$. This last step requires hard time constraints. Indeed, the control system has to be very fast to have an efficient effect on the structure. Usually, the sampling frequency of the control system has to be at least ten time higher than the highest frequency to control [6]. More than the speed of the computation, the control system has to be deterministic. These constraints match these of the real time systems. This section describes the different parts of the control system, which is shown in Figure 3, linked to the structure to control.

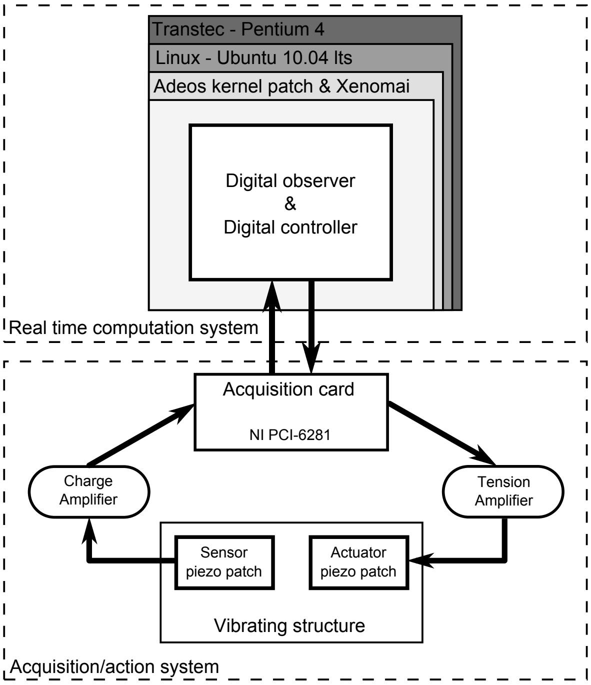

FIGURE 3: *Control system linked to the vibrating structure. Separation in the acquisition/action part and in the real time computation part.*

Two parts are presented separately. The first one is the acquisition/action system with the transducers, the amplifiers and the acquisition card. The second one is the real time computation system with the digital observer and the digital controller.

3.1 Acquisition/Action System

The transducers used to apply the control on the structure are piezoelectric patches. In Figure 2, one block of four patches are used as actuators and one patch of the other block as sensor. These piezoelectric patches are linked to a tension amplifier for actuators and to a charge amplifier for sensor. These amplifiers are then connected to the output and to the input of the acquisition card. The characteristics of the amplifiers and of this card is given in Table 1.

Tools	Type	Characteristics
Tension amplifier	Trek Model PZD350A	Gain 35
Charge amplifier	Brul & Kjaer Type 2635	Gain 2
Acquistion card	National Instruments PCI-6281	Input range $\pm$ 10V Output range $\pm$ 10V Max sampling frequency 625 kHz

TABLE 1: *Characteristics of the acquisition/action system tools.*

This part of the control system doesn't lead problem for the application of active control. Indeed, the two amplifiers are analog and don't introduce delay in the feedback loop. The acquisition card has a high sampling frequency which enables fast reading, analog to digital and digital to analog conversion and writing steps. The delay introduced by these steps are much lower than the delay of the computation system. The only limitation of the acquisition/action system is the fact that the amplifier have to be correctly set. Indeed, if their gains are not correctly chosen, the amplifiers can saturate and give wrong signals.

3.2 Real Time Computation System

The part which is essential for the efficiency of the control system is the computation part. Indeed, this part is a digital one and is subjected to discret time limitations. As said in an earlier section, these constraints match these of the real time systems. A description of this part is given in Figure 4.

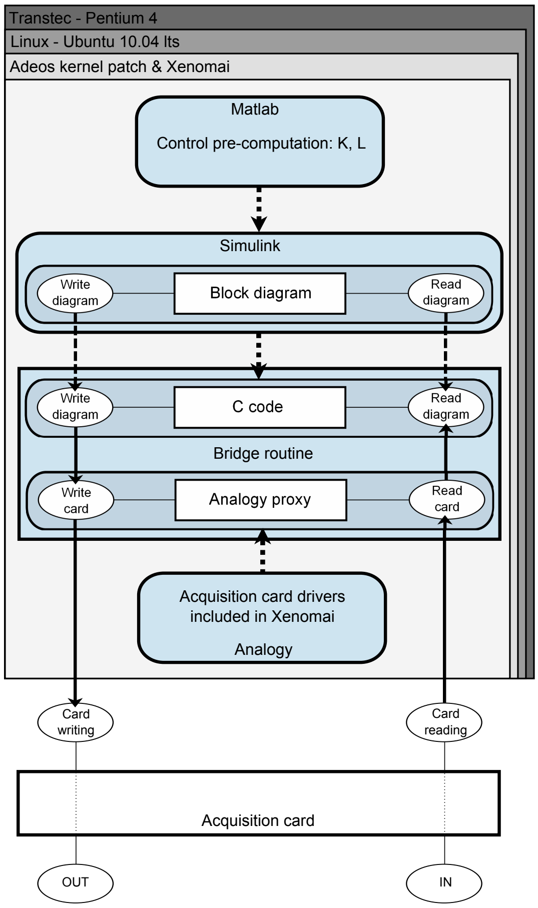

FIGURE 4: *Links between each component of the computation system.*

Hardware

Different hardwares as DSP or microcontrollers are usable to make a real time system. In this study, a computer processor used with a real time framework is chosen. The characteristics of the computer are given in Table 2.

Computer	Transtec
Processor	Intel - Pentium 4
Cores	2 virtual cores
Frequency	2.40 GHz

TABLE 2: *Characteristics of the computer used the computation part.*

The operating system used is an Ubuntu 10.04 lts which is a Linux distribution. The kernel is then patched with Adeos which is an interrupt management environement [7] and Xenomai which is the real time framework enabling to reach the desired time constraints [4]. The kernel is then configured and compiled to give the real time system. Details of the configuration and of the Xenomai framework are given in [8].

Analogy_proxy

Analogy is used to link the acquisition/action system and the real time framework. It is a free software included in Xenomai, used for managed data acquisition devices in real time [9]. The driver codes have to be configured to match the card characteristics. Then a C routine nammed Analogy_proxy is created to initialize the ports of the NI card used as input and output. The reading and writing functions are given in this routine. The function to turn the control computation in real time is also written in this routine with the highest scheduling priority. These C functions are very simple and use basic functions of the Xenomai library given in Table 3.

Task	Function	Description
Read	a4l_sync_read	Perform a synchronous acquisition read operation
Write	a4l_sync_write	Perform a synchronous acquisition write operation
Turn on real time	rt_task_shadow	Turns the current Linux task into a native Xenomai task

TABLE 3: *Xenomai library functions used in Analogy_proxy.*

Pre-computation Step

Before to design the control system, a pre-computation step is necessary to find the controller and the observer gains K and L according to the control target. Matlab is used to find these gains thanks to usual control algorithms [10].

Design of the Control System

Then, the control system, which is built with the observer and the controller, is modeled with a block diagram thanks to the software Simulink. This model is shown in Figure 5 and can be studied in time simulations to observe the effects of the control.

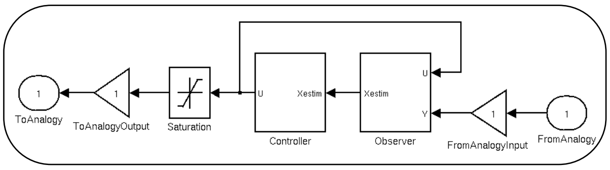

FIGURE 5: *Block diagram of the control system modeled using Simulink.*

The design of the control system can be made with a continuous or a discret time model. Indeed, the current microprocessors are so fast that a control system can be easly design in continuous time and then transform in a digital system. Moreover, digital controllers have several advantages. They can be easly modified and have a good stability in time [6]. This control system model can be used directly for experimental implementation thanks to the code generation tool of Simulink. A C code file can be created and used in the real time environement. Nevertheless, the input and the output of the computation system is not linked to the acquisition/action system yet. Specific names are given to these input and output and are used to make this link. As shown in Figure 5, the input is called *FromAnalogy* and the output *ToAnalogy*.

Bridge Routine

In order to enable the use of the Analogy_proxy functions by the C code generated by Simulink, a bridge routine is developed. Thus the computation part can communicate with the acquisition/action part of the experimental control system. This routine is a shell script which has to be executed each time a C code is generated by Simulink, typically when the model of the structure changes. It replaces the name of the input *FromAnalogy* and output *ToAnalogy* of the block diagram in Figure 5 by function calls and compiles the new code to create a programm which is used to execute the control.

The experimental real time system is now ready to be used. Before to use it on the experimental setup, some tests are conducted to study its real time efficiency.

4 Test of the Real Time System

Firstly, the global real time system is tested. Then the experimental control system latency (granularity) and determinism (precision) are studied. The tests used in this section are inspired by the tests given in [8] and can be found in [11]. It can be noted that to ensure a real time execution the value of the file *sched_rt_runtime_us* which can be found in */proc/sys/kernel/* has to be set at -1. Thus all the CPU time is allocated to the real time applications.

4.1 Real Time System Precision

Precision under Low CPU Load

To test the precision of the implemented real time system, a program using a timer is executed. The period of the timer is set to 1000 μs. The data are written in a text file to be studied after the execution. In this first test, the program is executed during around 20 minutes with a low CPU load. The statistics about the results are given in Table 4.

Measure number	1194793
Minimum	937 μs
Maximum	1063 μs
Average	1000 μs
Standard Deviation	6 μs

TABLE 4: *Statistics for the precision of the system under low CPU load.*

The timer triggers have effectively an average of 1000 μs. The worst values are happened 63 μs before and after the asking trigger. Nevertheless, the standard deviation is about 6 μs that is much better than for an usual timer. A histogram representation of these results are given in Figure 6.

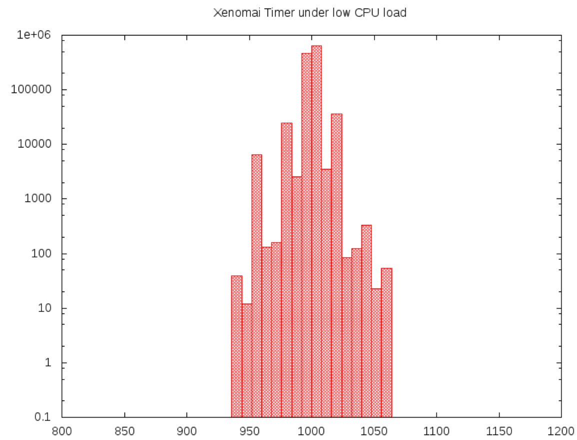

FIGURE 6: *Results given by the timer under low CPU load. (Abscissa: Duration between two triggers, Ordinate: Number of occurrence)*

The biggest part of the measurements happened effectively after 1000 μs.

Precision under High CPU Load

The same test is done with a high CPU load. The program *dohell*, which can be found with other test scripts in */usr/xenomai/bin*, is used to place the system in hard conditions. The statistics about the results are given in Table 5 and a histogram is given in Figure 7.

Measure number	1122072
Minimum	937 μs
Maximum	1061 μs
Average	1000 μs
Standard Deviation	2 μs

TABLE 5: *Statistics for the precision of the system under high CPU load.*

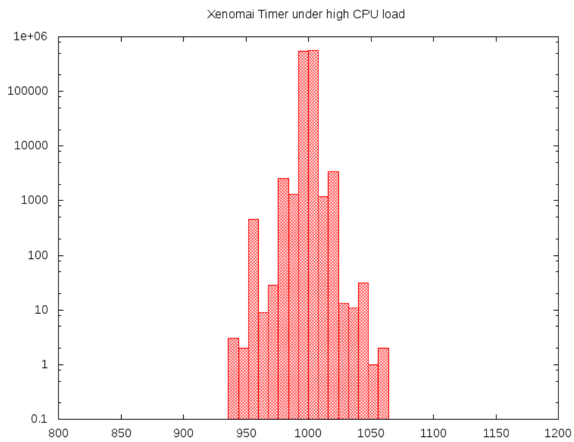

FIGURE 7: *Results given by the timer under high CPU load. (Abscissa: Duration between two triggers, Ordinate: Number of occurrence)*

In this case, the worst values are the same than the test with the low CPU load. But the standard deviation is better. That was not what we expected since the processor has a lot of task to manage.

4.2 Experimental Control System Efficiency

Precision and Latency

In this section, the precision and the latency of the experimental control system is studied. For the first test, the control system and more specifically the observer is made with a eight modes' model. This means that the control can be applied only on these eight modeled modes. A pre-allocated buffer is used to store 150000 values measuring the duration between two samples in the feedback loop and during the control. The statistics of these measurements are given in Table 6.

Measure number	150000
Minimum	16.1 μs
Maximum	21.5 μs
Average	17.0 μs

TABLE 6: *Statistics for the precision and the latency of the experimental system during the control.*

The histogram about these results is given in Figure 8.

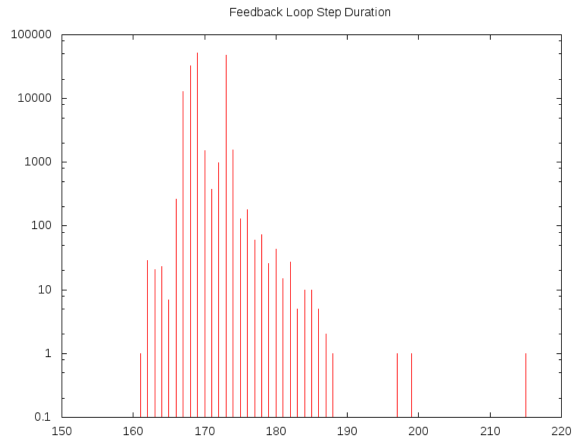

FIGURE 8: *Results given by the time measurement during the control and with a eight modes' model. (Abscissa: Duration between two triggers, Ordinate: Number of occurrence)*

The average duration between tow samples is 17 μs. The worst deviation is about 4 μs. This enable the control of a large frequency band in which are include the eight first vibration modes of the structure.

Influence of the Modes' Number

The Influence of the modes' number on the duration of the feedback loop is tested in this paragraph. The control applied to the structure is the same than in the last paragraph. Different models are used to study the modes' number effects. The smallest one is a 5 modes model and the biggest one a 25 modes model. The average, the minimum and the maximum durations for each studied models are given in

Figure 9.

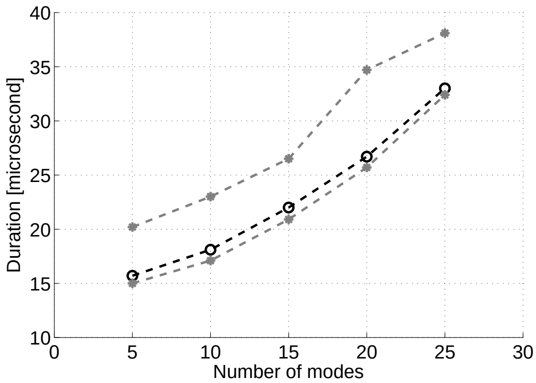

FIGURE 9: *Average (- - -), minimum and maximum (- - -) durations of the feedback loop for the models with 5, 10, 15, 20 and 25 modes.*

The duration of the feedback loop of the system increases with the number of modes in the model. However, these latencies are short and allows the control of large frequency bands. Moreover, the precision of the computation system is about 8 μs for the worst values that is a good result for an active control application in the perceptible domain of the frequencies. This real time system seems to be a efficient tool to apply acive control.

5 Experimental Results

In this section, the experimental control system described earlier is used to control the experimental setup of the Figure 2.

5.1 Control Target

The theoretical Frequency Response Function (FRF) of the soundboard between the disturbing and the measured signals is given in Figure 10.

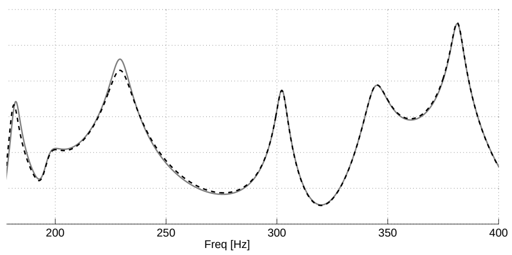

FIGURE 10: *Theoretical FRF of the soundboard without (—) and with control (- - -).*

Each peacks on this curve match a mode of vibration. The parameters of each mode can be modified thanks to the control system. The control target have to be defined before to apply the control. In this example, several modal parameters are decided to be controlled. These modifications are given in Table 7.

Mode	Modal parameter	Modification [%]
3	Frequency	-0.5
5	Damping	+25

TABLE 7: *Modal parameters modifications.*

The observer and controller gains are found thanks to the pre-computation step according to the chosen modal parameters modifications. The theoretical effects of this control can be observed on the theoretical FRF of the control soundboard given in Figure 10. Finally the computed gains are implemented in the experimental control system.

5.2 Results of the Experimental Control

To test the efficiency of the control system, the soundboard is disturbed with a chirp. Thus, the structure is excited in a frequency band between 0 and 400 Hz. The experimental FRF of the soundbaord measured between the disturbing and the measured signal is given in Figure 11.

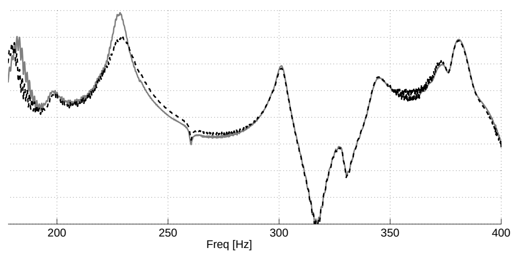

FIGURE 11: *Experimental FRF of the soundboard without (—) and with control (- - -).*

Then this experimental FRF is measured again while the control system is turned on. The resulted FRF is given in Figure 11. It can be observed that the control is efficient. Indeed, the effects applied by the experimental control system are quite the same than those of the Figure 10. The fact that the theoretical and the experimental FRF are not exactly the same is due to the location of the disturbing signal of the FRF. In the theoretical case, the input is sent at the same point that the control signal but not in the experimental case.

6 Conclusions

This study presents the use of a Linux real time framework to an active control application. Xenomai is used as the real time environment in which the experimental control system is develop. Observer and controller gains are found in a pre-computation step before to be implemented in the control system. Simulink is used to easily design this system with a block diagram. Then a routine is used to link this model to the input and output of the acquisition/action system. The real time framework enables to reach the active control time constraints. After some test on this system, experimental results measured on a simplified soundboard of a string instruments are finally given to test the efficiency of the control system. These results show that the Linux real time framework is an efficient way to make an experimental active control system. Their still are some developments to do in order to obtain a stable and a finished system. For example the link between the C code generated by Simulink and the input and output of the acquisition card has to be enhanced. Indeed, a sampling frequency can be chosen for the Simulink solver. If this frequency is chosen higher than the frequency of the real time system problems can appear. An other question is about the precision of the system during the timer tests since its accuracy is better whith a high CPU load. Finally, to study the precision and the latency of the experimental control system, time measurements could be done with the *rt_fprintf* function of the RTDK Library.

Acknowledgments

This work has been done during the PhD of Simon Benacchio, funded by the Agence National de la Recherche (ANR IMAREV project).

References

[1] *Dynamic and Control of Structures, A Modal Approach*, W. Gawronski, 1998, Springer Ed.

[2] *The acoustics and psychoacoustics of the guitar*, H. Wright, 1996, PhD Thesis

[3] *Control of sound radiation and transmission by means of passive piezoelectric networks: modelling, optimization and experimental implementation*, G. Rosi, 2010, PhD Thesis

[4] *http://www.xenomai.org/*

[5] *Guitar making - The acoustician's tale*, B. Richardson, 2010, Proc. of the Second Vienna Talk

[6] *Signal processing for Active Control*, S. Elliott, 2001, Academic Press

[7] *Adaptative Domain Environment for Operating Systems*, K. Yaghmour, 2001, Opersys inc.

[8] *Solutions temps réel sous Linux*, C. Blaess, 2012, Eyrolles

[9] *http://www.xenomai.org/index.php/Analogy_general_presentation*

[10] *Active control applied to string instruments*, S. Benacchio et al., 2012, Acoustics 2012, Nantes

[11] *www.blaess.fr/christophe/livres/solutions-temps-reel-sous-linux/*