Real Time System for Instrumental Sound Extraction and Recognition

Adrian Ioan Lita[1)], Laurentiu Mihai Ionescu[2)], Alin Gheorghita Mazare[2)], Gheorghe Serban[2)], and Ioan Lita[2)]

[1)] Politehnica University of Bucharest
[2)] University of Pitesti,
ioan.lita@upit.ro

***Abstract**: This article presents a "smart sound sensor" solution. More specifically it is a system on a chip capable of extracting sound patterns from an input signal captured by a microphone, patterns that correspond to certain musical instruments. The aim is to identify sounds like: voices, ambient sounds or musical sounds. Such applications generally involve the presence of a computer that analyzes offline the audio sequence. There are also smartphone security applications that rely on voice recognition templates that were previously stored and which work by processing the input sequence offline, after recording. Unlike these methods, our solution enables online real time identification of a group of musical instruments and allows extracting of certain sound sequences that belong to a particular instrument so that it can be isolated for sole playing. The solution was implemented in an integrated system "sound sensor" that analyzes the input signal in real time and can give output data without additional elements.*

1. Introduction

Sound recognition is an elaborate research subject, addressed by a large number of papers. Currently speaking, a number of instruments purposed for sound recognition are already available: speech recognition which runs on PCs and smartphones, security (authentication) applications for voice recognition and also specialized instruments for recognizing sound patterns in different areas, including the music.

Software tools [1] that allow the identification of musical accords from a set of instruments already exist. Also, applications that determine the accuracy of sound sequences [2] are already built: they are not only identifying the accord type, but they also compare it to stored accords and determine the accuracy rate. Furthermore, by analyzing the sound, certain estimations can be computed for determining the base frequency of a song [3]. Recent papers are dedicated to building instruments for real time sound quality evaluation [4] or for analyzing melodic lines [5]. The system proposed by the authors is a hardware “sound sensor”. It consists of parallel processing algorithms for fast source identification using the sound footprint of a certain musical instrument. The system is used to recognize both the musical instrument and the notes played with that instrument, either online, during an audition, either offline, by playing pre-recorded music. The system was designed as a circuit that can be connected to other devices through various interfaces including Ethernet and Bluetooth or it can be used as a standalone product.

The system has numerous applications, not only for the analysis of instrumental sound or the quality of interpretation, but also in other domains where is necessary to identify certain sound generators in real time. In the upcoming years more and more research subjects on integrated sound sensors are funded by the industry [6]. Such intelligent and integrated solutions can improve the quality in production when are used on industrial level [7].

The following section describes the system architecture while section 3 presents the results obtained by the use of this system.

2. System Architecture

A system block schematic is presented in the figure below (fig. 1).

978-1-5090-1389-0/16/$31.00 ©2016 IEEE

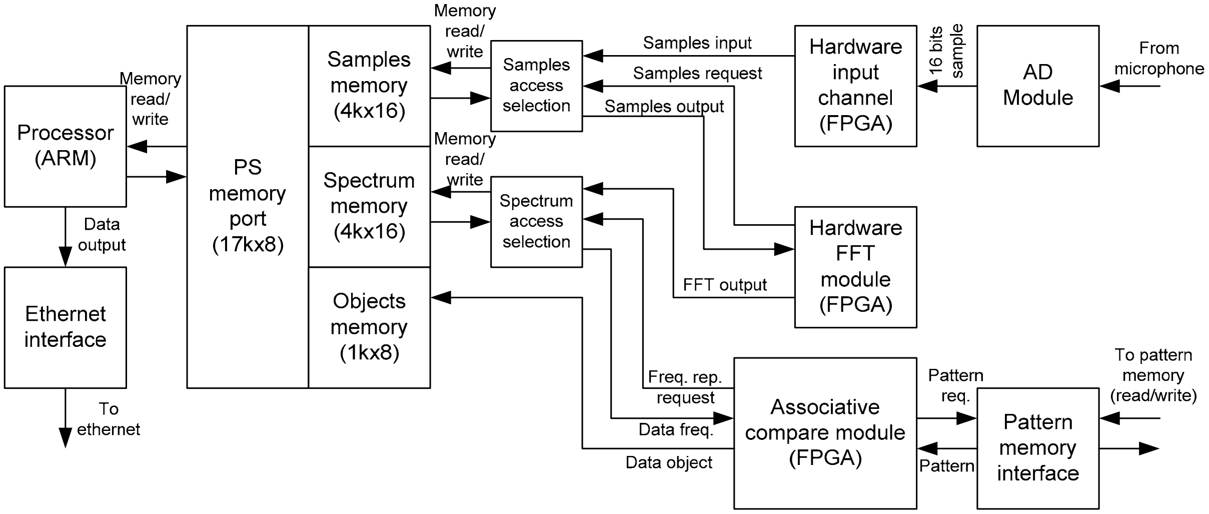


Fig. 1. Sound sensor block diagram.

As it can be observed, the system is based on the SoC Zynq 7000 device. The SoC is connected to an A/D converter for capturing of sound and inside is implemented a digital sample storage system, an associative compare module and a spectral representation with digital sound extraction system.

The comparison process is based on associative memory cells. In this way, from the sample sequences, with a defined time-window, certain patterns which belong to certain instruments can be recognized (patterns are already stored). If there is a match between a pattern that needs to be identified and another sound (from other instrument, or even noise), the specific pattern can be extracted through associative comparisons.

The system is built from two types of modules. The analog to digital converter, as well as the fast Fourier transform unit are already built modules. The A/D converter is an external circuit especially designed for audio processing systems. The FFT module is built into the Zynq 700 SoC and it is allocated via the Vivado Suite by using library components.

The second type of modules is the modules built by the authors. Some reflect the usage of SRAM memory (block RAM memory) that resides in the programmable logic of the SoC – the module responsible for storing the samples generated by the A/D converter, as well as for storing the FFT results. The module responsible for the comparison between the spectral representation of the input signal and the pre-stored samples of instruments are done based on associative memory, designed by the authors.

2.1. Allocated modules for "sound sensor"

The A/D converter used in the present paper is made by Analog Device (SSM2603). It is 24 bits A/D converter which allows sampling sound at frequencies that also include 96 kHz. Using this converter, the module can sample and store 4096 samples with 16 bits each.

The HFFT (Hardware Fast Fourier Transform) module is responsible for running of the discrete fast Fourier transform with 4096 samples accumulated over time. The module is based on hardware digital signal processing cores which offers in about 200 us the spectral values for the 4096 samples, assuming a 100MHz input clock frequency is fed into the programmable logic.

2.2. Built modules for "sound sensor"

As figure 1 depicts, the system contains memory circuits used for storing samples, spectral values and identified objects – instruments that were recognized. For all these, SRAM memory blocks were used in bi-port operating mode. The SRAM blocks (also called Block RAM) consists of high speed memory modules inside the programmable logic. These modules can run at working frequency (100MHz) and use only one clock cycle, which gives a 10ns access time for reading and writing. Another characteristic is that the

Block RAM can be configured in bi-port mode: one port is used by the programmable logic for storing samples and spectral values, while the other port is accessed by the programmable system (processor).

The comparison between the spectral values of a sample sound and the spectral values of an instrument template is done by an associative memory module, presented in figure 2.

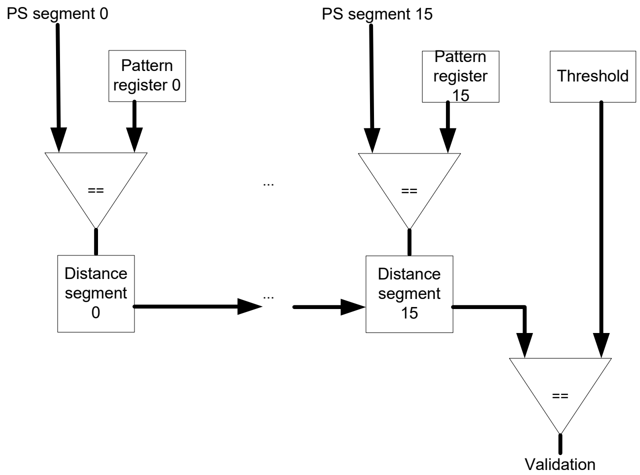


Fig. 2. Associative compare module.

The module contains 16 blocks capable of parallel processing 256 bits information. More precisely, each block takes a 16-values segment (each value has 16 bits) - the output of the spectral values memory and compares it with 16 values that belong to the stored template instrument. Each separately value represents the ratio of power to reference (sound at 0dB) for a certain frequency. The result is that the associative memory will extract the power differences that appear for each frequency. Every of these differences will be compared to a threshold. On the output of the comparator is generated a 512 bytes validation word – one bit for each frequency, which will show if the difference between the input and the template is below the threshold (0) or above it (1). This word represents the degree of recognition for certain instrument, and is stored in the object memory to be analyzed by the programmable system (processor).

3. Experimental results

A block schematic of the overall experimental system is depicted in figure 3.

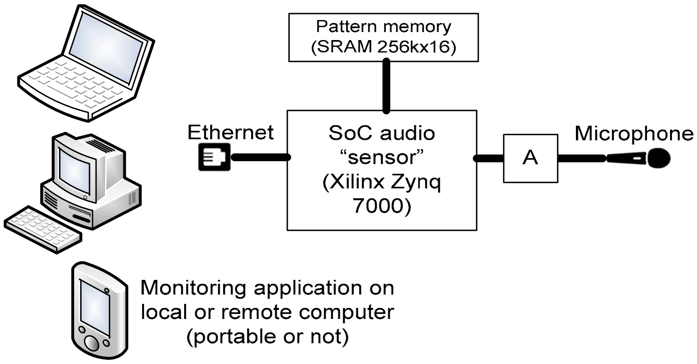
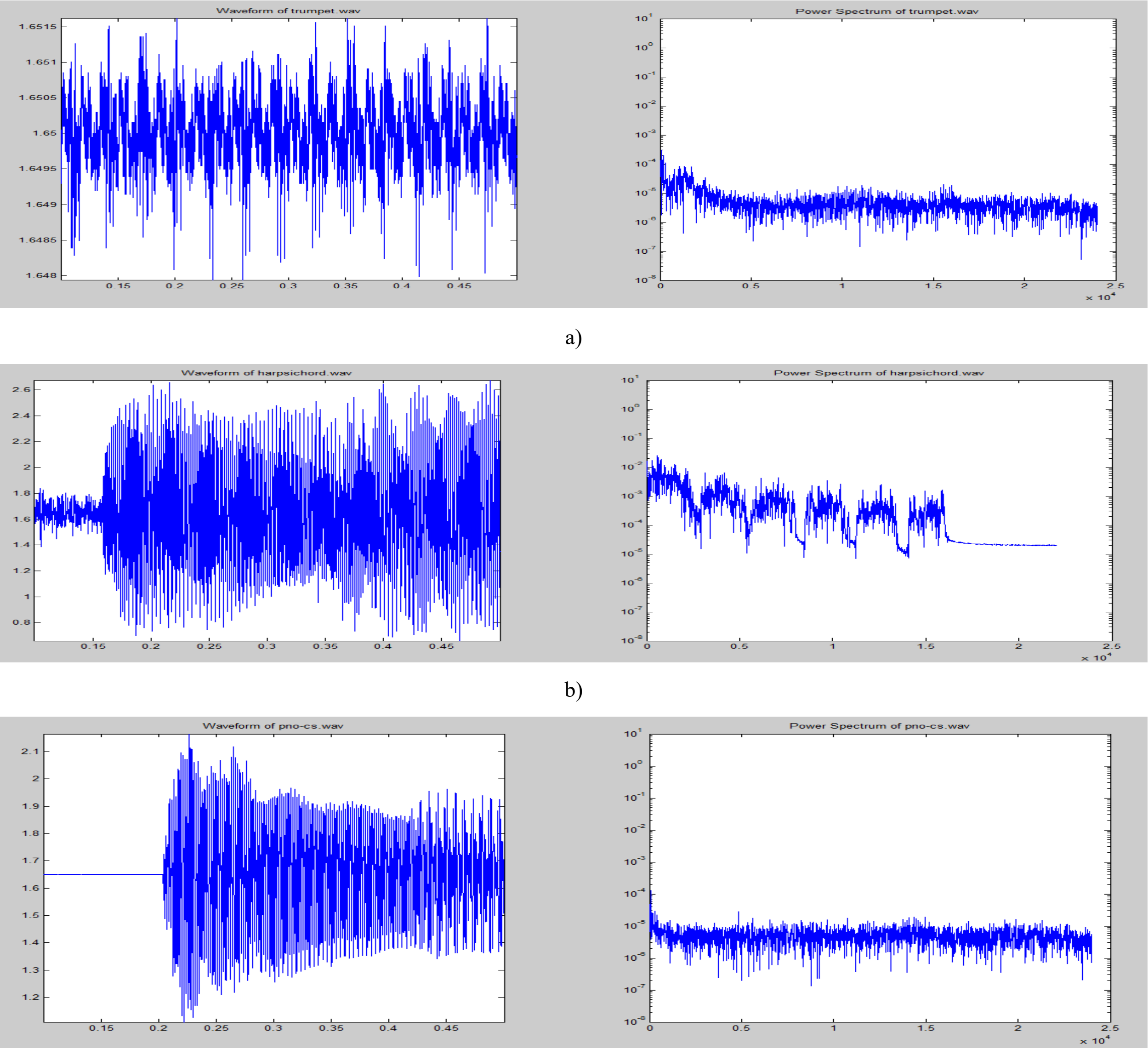

Pattern memory (SRAM 256kx16)
Ethernet
SoC audio "sensor" (Xilinx Zynq 7000)
A
Microphone
Monitoring application on local or remote computer (portable or not)

Fig. 3. Experimental system.

As shown, the experiments were conducted for 3 separate template instruments: trumpet, piano and harpsichord. The samples used were provided for experiments by Stanford University [8]: the piano was recorded for 20 seconds, the harpsichord for 18 seconds and the trumpet for 17 seconds. The wave files (.wav) contain accords for playing several octaves. The sample rate is 1.536 kbps.

These files were read and 4096 components FFT were calculated for them. The spectral values, totaling 4096, were stored in an external 256k x 16 SRAM memory. The memory is organized by pages of 4k x 16, exactly the size of the template. There is a total of 3 memory pages used.



a)

b)

c)

Fig.4. Time representation (left) and power spectrum representation (right) for templates of trumpet (a), piano(b) and harpsichord (c) – the display of the results it was done in MATLAB.

The left side windows represent a chuck of the time representation (more precisely, the first 500ms) for each of the 3 templates. The vertical axis represents the voltage for the measured sample. In the right side windows are presented the spectral values resulted by applying the fast Fourier transform, keeping in mind that on the vertical axis is represented the power.

Each template represents a pool of attraction for values in its neighborhood. This means that if a value is in the pool of attraction of a certain template (clearly speaking, if the value is close enough to a template) then the system will be capable of associating that value with the template, meaning that it will identify the instrument as the one in the template.

To better understand what "close enough to a template" means, the distance between templates has been determined. The next step was determining the minimum and the maximum of distances between templates. The values obtained are depicted by tables 1 and 2 (absolute values, in Watts).

Table 1. The minimum of distances between templates (W)

Trumpet – piano	Piano – harpsichord	Harpsichord – trumpet
6.1465 10^{-9}	1.0438 10^{-6}	7.3144 10^{-9}

Table 2. The maximum of distances between templates (W)

Trumpet – piano	Piano – harpsichord	Harpsichord – trumpet
1.7542 10^{-4}	0.0254	0.025

These differences between the minimum and maximum distances illustrate that for some frequencies sound patterns are much closed but, in same time, we have higher differences for other frequencies. The goal is to identify the number of frequencies (points in spectrum analysis) where the patterns are different enough.

In respect to this, the associative memory has a threshold value of 10^{-7} W (50 dB), threshold which corresponds to the level of background noise in a concert hall when there is no play – it corresponds to the background noise during the break. Under these circumstances 38 points (frequencies) with the spectral difference less than the threshold value between the trumpet and the piano were determined, while between harpsichord and piano there were no points and between the harpsichord and the trumpet one difference point. The results are listed in table 3.

Table 3. "Common points" with differences under the 50dB threshold.

	Trumpet	Piano	Harpsichord
Trumpet		38	1
Piano	38		0
Harpsichord	1	0	

The experiments were done by capturing sounds in a medium noise hall (concert pause), by identifying the instruments that produced the respective sounds. This way, the experiments confirmed the obtained results, which are presented in table 4.

Table 4. The probability of detection in "low noise" conditions.

Identifying trumpet	Identifying piano	Identifying harpsichord
98% (0.981)	98%(0.981)	99%(0.999)

After this stage, different experiments were done by raising the hall noise level (adding more noise levels to the measurement hall). A graph of the results obtained under the new experiments is presented by figure 5.

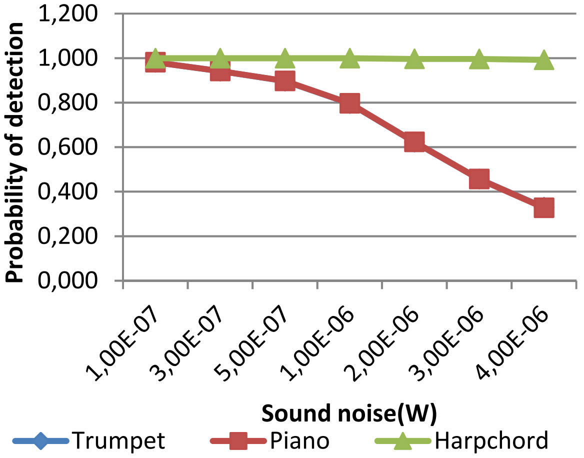


Fig.5. Evolution of detection probability when increase the noise

As it can be observed, the evolution of probability with noise increase behaves differently from one instrument to another. While the harpsichord probability remains unchanged, the trumpet and the piano seem to have very small differences to one another, with the overall probability decreasing quite fast with the increase of noise (according to [9] 10^{-6} W (60 dB) is the equivalent of a noisy room such as a barber shop or a restaurant). On a noise level of approximately 65dB it seems that the recognition probability between the two drops to 0.3, similar to the case where a random prediction is made (1 out of 3).

On the other hand, the harpsichord detection worked almost perfect since the probability rate never went below 0.98. This difference in identification can be assumed by the tone difference between instruments – some instruments, even if it does not seem that way can have similar tones – and on the quality of the template files (experiments can be redone with different template files).

The identification time was evaluated, taking in to consideration the following: the system allows acquiring the signal of microphone with frequency rate of 96000 Hz. This means that 4096 samples were acquired in 42.7 milliseconds. At the same time as the sound is sampled, it is also memorized. The hardware Fourier transforms (HFFT), which process 4096 samples, takes 221 us (including storage in the spectral memory). The associative comparison between the spectral waveform and a chosen pattern for a certain instrument takes about 700 us, time which also includes identification of difference between the pattern and the instrument (differences which may occur from the accord of instrument, background noise, the acoustic of room, etc). This means an identification time of about 1ms after receiving the samples. If we take into account the 42.7 ms for receiving the samples, a response time of less than 50 ms is achieved. The system, having a detection rate of 20 times per second can be successfully used in a real time sound recognition application.

The identified instruments as well as the tone difference from the standard pattern can be delivered to a final user in different ways. For example, our system has a built-in Ethernet interface and can transmit the results on a local network. The final user can use the same Ethernet interface to configure the system by transmitting sound patterns and selecting an instrument for detection.

4. CONCLUSIONS AND FUTURE TRENDS

The solution proposed by the authors brings an innovation element by implementing the acquisition and recognition of sound samples in an autonomous hardware structure. The system allows recognition of musical instruments via the spectrum analysis of the sound captured by a microphone, and, by comparing it to the spectrum of pre-recorded template instruments. The comparison is done using a parallel associative memory. In this way, the differences in spectrum can be detected with a certain probability degree. The detection capability is significant higher compared to other solutions, some of which are implemented on PC. Detection time is very small compared to other PC-based analysis solutions. Taking all the above into consideration, the overall system can be used as a true real-time sound sensor.

Further research directions that we have in mind are related of using more sound patterns in order to increase the number of instruments that the system can detect and classify.

REFERENCES

[1] T. Fujishima, „Realtime chord recognition of musical sound: A system using common lisp music”, Proc. ICMC, pp. 464-467, 1999

[2] M. Goto, „An audio-based real-time beat tracking system for music with or without drum-sounds”, Journal of New Music Research, Vol 30, No. 2, pp.159-171, 2001

[3] M. Goto, „A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals”, Speech Communication, No. 43, pp. 311-329 2004

[4] O. Romani Picas, H. Parra Rodriguez, D. Dabiri, „A Real-Time System for Measuring Sound Goodness in Instrumental Sounds”, Society Convention 138, 2015

[5] A. Arzt, A. Widmer, „Real-time music tracking using multiple performances as a reference”, Proc. of the International Society for Music, 2015

[6] E. Ceuca, A. Tulbure, A. Taut, O. Pop, I. Farkas, “Embedded System for Remote Monitoring of OBD Bus”,ISSE 2013 , 36th International Spring Seminar on Electronics Technology „*Automotive Electronics*” May 8 – 12, 2013, Alba Iulia, Romania

[7] Belu Nadia, Anghel Daniel Constantin, Rachieru Nicoleta, „ Application of Fuzzy Logic in Design Failure Mode and Effects Analysis”, Innovative Manufacturing Engineering, Book Series: Applied Mechanics and Materials, Volume: 371,Pages: 832-836, 2013

[8] https://ccrma.stanford.edu/~jos/Piano/ Piano_Harpsichord_ Sound _Examples.html

[9] http://www.engineeringtoolbox.com/sound-power-level-d_58.html, http://www.acoustic-glossary.co.uk/sound-power.htm, https://en.wikipedia.org/wiki/Sound_power