Physical Audio Digital Filters

Stefano Fasciani
S^3 Research Center
Faculty of Engineering and Information Sciences
University of Wollongong in Dubai
stefanofasciani@stefanofasciani.com

ABSTRACT
We propose an approach to insert physical objects in audio digital signal processing chains, filtering the sound with the acoustic impulse response of any solid measured in real-time. We model physical objects as a linear time-invariant system, which is used as an audio filter. By interacting with the object or with the measuring hardware we can dynamically modify the characteristics of the filter. The impulse response is obtained correlating a noise signal injected in the object through an acoustic actuator with the signal received from an acoustic sensor placed on the object. We also present an efficient multichannel implementation of the system, which enables further creative applications beyond audio filtering, including tangible signal patching and sound spatialization.

Author Keywords
NIME, Tangible Filters, System Identification, Linear Filtering.

ACM Classification
H.5.5 [Information Interfaces and Presentation] Sound and Music Computing - Signal analysis, synthesis, and processing, H.5.2 [Information Interfaces and Presentation] User Interfaces, C.3 [Special-Purpose and Application-Based Systems] Signal processing systems.

1. INTRODUCTION
In digital signal processing, filters allow changing, reducing or enhancing temporal and spectral characteristics of signals. In digital musical instruments and interactive computer music systems, filters are used to implement a wide range of audio effects, from equalizers to reverberators. A digital filtering structure can deliver significantly different alterations of the input signal, and this depends only on the selected filter coefficients. A single structure can selectively remove frequency components or insert delayed replicas of the input signal. The relationship between filter coefficients and the resulting sonic effect is not intuitive. Indeed several algorithms has been proposed to compute the set of coefficients from functional specifications of the filter [1]. In musical performance and composition, filters are manipulated to produce dynamic timbral and spatial variation of the sound. Due to the electrical (analogue or digital) nature, the filtering process is abstract and intangible when compared to the physicality of acoustic resonators and instruments.

Physical spaces are used as audio filters, such as in Alvin Lucier's piece "I am sitting in a room". When we convolve a signal with the impulse response of a room we virtually place the sound source in that specific physical environment. This is equivalent to propagating the sound through the room from a speaker to a microphone, which picks up direct sound and reflections (i.e. reverberation). This approach has severe practical limitations, such as the colocation with the preforming space and audio feedback loops, when aiming to measure or modify the room response in real-time for providing interaction wit the digital musical system.

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

Here we propose a similar approach to filter audio signals with the response of solid objects, which represent tangible elements that we insert as linear filters in the audio digital signal processing chain. Audio filtering can be achieved by propagating acoustic waves through a solid by the mean of electromechanical acoustic transducers. Assuming that these components are transparent, the sonic response depends on material and shape of the object, as well as on the positioning of the transducers. Manipulating the object is possible to vary these features, within physical constraints, determining a different response, hence a different audio signal alteration. However this approach, other than requiring digital-to-analog and analog-to-digital conversion, presents detrimental shortcomings for the interactive use in musical contexts. Firstly the transducer that senses the filtered sound and the loudspeakers that reproduce it can determine an unstable feedback loop. Secondly, the manipulation of object and transducers generates undesired vibrations throughout the system, with frequency and energy content in the audible range, adding noise bursts to the filtered audio signal.

In our method the physical object is fed with a continuous acoustic stimulus used only for the instantaneous and continuous estimation of its impulse response, which samples represent the coefficients of the filter that we use to process another sound source. This approach does not present feedback loops because stimulus and filter input are uncorrelated. Moreover in this way we also achieve robustness against noise bursts generated by tangible interaction with the filtering object. The rest of this paper is organized as follows: related works are reviewed in the next subsection; Sections 2 discusses the modeling of physical objects as digital filters and the technique used to estimate the response; in Section 3 we present the implementation of a proof-of-concept prototype; possible applications are considered in Section 4, followed by discussion and future work in 5.

1.1 Related Works
Tangible physical objects augmented with acoustic sensors and actuators have been used in conjunction with computational algorithms for interaction between human and computers. Previous works have explored this approach for musical instruments, musical controllers, as well as for general-purpose interfaces. The induction of audible-range mechanical vibrations in the body acoustic musical instruments has been proposed and explored to provide novel form of intuitive and tangible computer music interaction [2]. The electromechanical transducers installed on actuated musical instruments enable the superposition of computer-generated sounds radiated through the same instrument body, equipped with contact microphones for a closed loop control [3]. Vibration sensing and stroke recognition throughout an interfacing physical object has been proposed to overcome the limitation of location-oriented striking of drum-oriented musical controllers [4]. The tangible acoustic interfaces technique described in [5] turns solids of arbitrary shape in interactive objects by detecting the contact position using source location and acoustic imaging.

Acoustic and vibration principles have been used also for general-purpose interfaces not explicitly designed for musical purposes. The Acoustruments [6] are inexpensive and passive plastic extensions for smartphones that add tangible functionalities to handheld devices, using only the existing device's microphone. A similar strategy, not restricted to smartphones and based on strike vibration sensing, has been proposed in Lamello [7]. Vibration speaker and a piezoelectric microphone are paired as a single sensor in [8] and proposed an generic hardware configuration for implement interactive objects with touch capability, supporting the recognition of different gestures. The frequency shift of air-borne inaudible acoustic waves determined by hand reflection is used in [9] to detect gesture, leveraging speakers and microphones already embedded in most consumer devices.

2. MODELING AND ESTIMATION

For the aim of this work, we model solid objects as a causal Linear and Time-Invariant (LTI) system. This implies that acoustic waves travelling through the physical object between two arbitrary locations can only be delayed and attenuated. All frequency components at the output must be present at the input too. Each component presents its individual delay and attenuation (or gain). With this assumption we are ignoring the parametric array, which is a nonlinear propagation transduction mechanism generating narrow and nearly side lobe-free beams of low frequency components, through the mixing and interaction of high frequency waves, effectively overcoming the diffraction limit associated with linear acoustics [10]. This determines the production of harmonics and mixed tones not present in the original sound. The advantage of LTI modeling is the availability of computational methods to estimate the system response, and to compute the output of a known system given an arbitrary input.

To filter an audio signal with a physical object modeled as a LTI, we need to estimate its Impulse Response (IR), and then perform the linear convolution between the incoming signal and the IR. This is equivalent to letting the sound through the solid, overcoming the limitations of a direct acoustic filtering as discussed in the introduction. In the next sections we detail the method to estimate the IR of an LTI and the limitations of the measurement setup.

2.1 LTI System Identification

In the time-discrete domain, the output of an LTI system $y(n)$ is the linear convolution between the input $x(n)$ and the IR of the system $h(n)$. To estimate $h(n)$ we use the property of the correlation sequence and LTI systems in (1), that equates the cross correlation between output and input $r_{yx}(l)$ to the input autocorrelation $r_{xx}(l)$ convolved with the IR. In (1) l represents the lag parameter or independent variable of the correlation sequence.

$$r_{yx}(l) = r_{xx}(l) * h(l) \tag{1}$$

The autocorrelation sequence of zero-mean white noise with a spectral density that is equally distributed across the whole frequency range is equal to its variance σ^2 when the lag l is zero, and it is and null elsewhere (i.e. equal to $\sigma^2\delta(l)$). Therefore, feeding the LTI system with white noise, the cross-correlation $r_{yx}(l)$ is exactly the IR scaled by the noise variance as in equation (2) and illustrated in Figure 1.

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} x(n)\, y(n-l) = \sigma^2\delta(l) * h(l) = \sigma^2 h(l) \tag{2}$$

This method is impractical because, as shown in (2), the crosscorrelation has to be computed over an infinite number of samples. However to get an accurate estimation of $h(n)$ is sufficient to compute the correlation on a finite number of samples, and average the $h(n)$ over multiple measurements. The considered number of samples must be at least equal or greater than the IR length. An advantage of this approach is that the estimated $h(n)$ represent a Finite Impulse Response (FIR) filter that will determine any stability problem when used to filter an audio signal, since it only presents zeros and no poles.

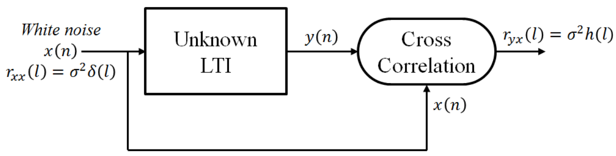

Figure 1: LTI system identification via correlation analysis.

2.2 Impulse Response Measurement Setup

The signals $x(n)$ and $y(n)$ in Figure 1 represent output and input of the computation environment executing the correlation analysis that estimate $h(n)$. Everything in the between contribute to the overall measured $h(n)$, not only the physical filtering object. In Figure 2 we provide a detailed breakdown of the typical components of the unknown LTI of Figure 1.

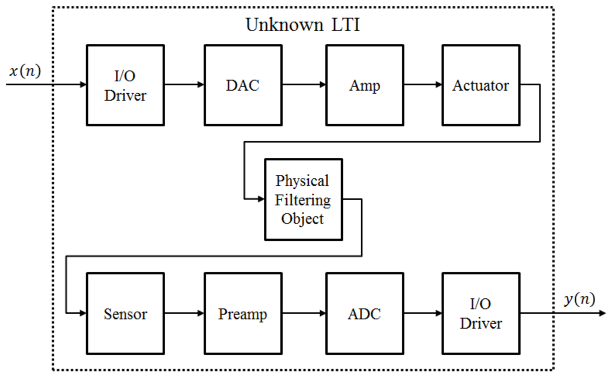

Figure 2: Breakdown of components contributing to the measured impulse response.

The $x(n)$ zero-mean white noise generated from the computer-based computation environment goes through: the operating system layers down to the I/O drivers, the digital-to-analog converter of the sound card, an amplifier, and finally the actuator transducing the signal into an acoustic wave injected into the physical solid object. Then it follows a complementary chain of components: the transducer that senses the filtered acoustic vibration on the object, a stage of signal conditioning and pre-amplification, the analog-to-digital converter, and the I/O driver that provides the samples of $y(n)$ to the computation environment for the correlation analysis.

If all components of the measurement in Figure 2 are transparent (i.e. no delay, flat magnitude response, linear phase), the estimated $h(n)$ exactly corresponds to the IR of the filtering object. This is an unlikely scenario, especially when using consumer-grade devices. Software application for real-time audio digital signal processing generally works with buffers of samples. The operating system and I/O drivers use buffering as well. This introduces a delay for each buffer in the chain equal to the number of samples in the buffer multiplied by the sampling period. Another significant source of delay is in the sigma-delta analog-to-digital converters, featured in all audio-grade ADC. These include a decimation FIR filter with approximately 40 to 60 taps, which contribute to delay the signal by a number sampling periods identical to number of taps. Delta-sigma modulators can be used in digital-to-analog converter as well, although this is less common, contributing with an additional delay due to the presence of an interpolating FIR filter. The electromagnetic moving coil in the actuating transducer determines another delay (range of microseconds and often negligible), which is proportional to the size of the driver.

Regarding the frequency response of the components in Figure 2, actuator and sensor have the greatest impact in coloring the signal.

The contribution of the other components is generally negligible. The actuator and sensor that we use, as most of loudspeakers and microphones, do not present a flat magnitude response in the audible frequency range. Delay and frequency response should be compensated, especially the latter, when trying to estimate the $h(n)$ of the physical object that we use as an audio digital filter. In our implementation we do not estimate the overall delay and frequency response from the documentation of the devices, but we measure these on the closed measuring loop by installing the sensor directly on the actuator. In theory, it is sufficient to include another pre-whitening filter in the measuring chain that compensates for the uneven frequency response.

In the ideal scenario with transparent devices and no filtering object, $y(n)$ is a zero-mean white noise signal as $x(n)$. We measure the whiteness W of $y(n)$ as in (3), where M is the length of the signal. This quantity is zero for white noise signal of infinite length. In our implementation we measure and display W for both $x(n)$ and $y(n)$ to facilitate the tuning of the pre-whitening filter that aims at reducing the non-ideality of the measuring loop.

$$W = \left(\sum_{l=-M}^{-1} r_{yy}(l)^2 + \sum_{l=1}^{M} r_{yy}(l)^2 \right) \Big/ r_{yy}(0)^2 \qquad (3)$$

3. PROOF OF CONCEPT PROTOTYPE

3.1 Hardware Configuration

For the implementation of a proof of concept prototype we used the following setup. For the soundcard we selected a Behringer UMC 404 soundcard, which provides 4 channels analog-to-digital and 4 channels digital-to-analog conversion, and it includes 4 microphone preamplifiers. We use large piezoelectric sensor or Korg CM-200 contact microphone as acoustic transducers. For the amplifier, we selected a class D based on the Maxim MAX9744. For the acoustic actuators we use surface transducers of two different sizes: the LB07, with a nominal power of 5 W and diameter of 44 mm, and the LB16 with a nominal power of 3 W and diameter of 30.5 mm. These are shown in Figure 3. Both have 4 Ω impedance and nominal frequency response between 100 Hz and 15 kHz.

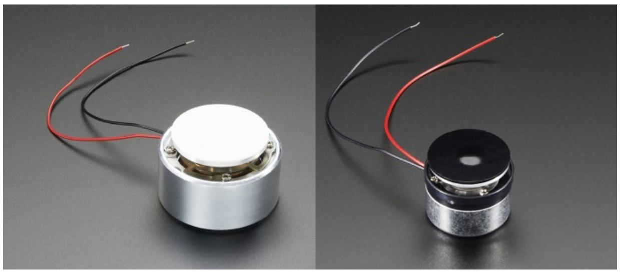

Figure 3: LB07 (left) and LB16 (right) surface transducers.

Using the hardware described above, we measured the loop delay and frequency response without including any physical filtering object, placing the sensor on the surface transducer. As a computational platform we used Max/MSP running on OSX 10.12. We measured a delay of approximately 350 samples with a sampling rate of 48 kHz and I/O vector size (buffer size) of 32 samples, which is the lowest supported by the UMC 404. This delay is equivalent to 7.2 ms, but can be reduced to approximately 3.6 ms when increasing the sampling rate to 192 kHz, as the overall delay in number of samples is almost constant, but the minimum supported buffer size grows to 64. Using the LB16 we observer slightly lower delay (few samples), due to its smaller size. However the delay of the impulse response estimation loop is not critical because it determines only a time shift in the measured IR of the physical object. The time shift can be removed detecting the attack of the impulse response, and eliminating the preceding samples. However we have to ensure that the length of the signal we use for the correlation analysis is greater than the IR of the physical object plus the measuring loop delay.

The frequency response of the measuring loop using the LB07 and LB16 with the piezoelectric sensors and CM-200 are shown in Figure 4 and it is evident that these are far from being transparent, contributing to color the signal and the measured $h(n)$. When using the piezoelectric sensor, the LB07 provides a better response at the lower end of the frequency spectrum, due to the larger size of the driver. However, both transducers are relatively small and have a weak response below 200 Hz. The response using the LB16 appears more uniformly distributed across the audible spectrum, but it is more irregular than the LB07, hence it is more challenging to compensate it using the cascade of basic filters, such as biquads. We prefer this approach rather than a higher order FIR compensating the whole measuring loop response due to the weak response and high harmonic distortion of transducers at low and high frequencies. According to the documentation, the piezoelectric sensors have a resonance at 1300 ± 500 Hz and 4000 ± 500 Hz. The latter one is clearly visible in both top spectrums in Figure 4, while the peaks at 300 Hz and 400 Hz are due to resonances in the body of the LB07 and LB16. With the CM-200 we obtain a better response at low frequencies, but more peaks and valleys across the audible frequency range, which makes challenging a compensation using basic filters. In all cases the response at frequencies above 5 kHz is poor.

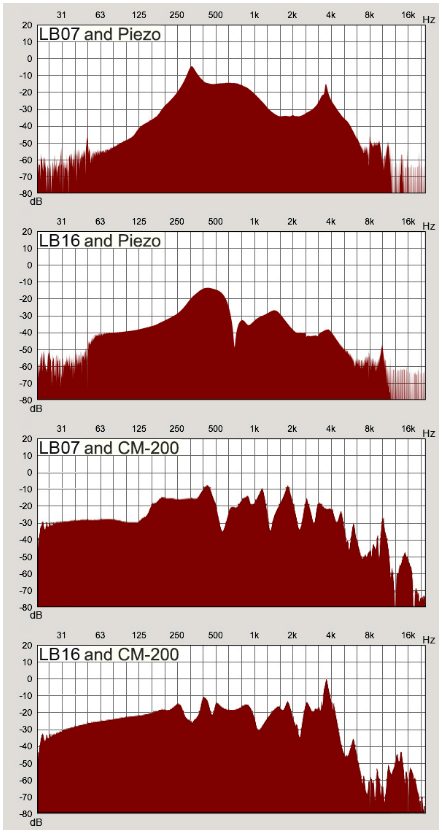

Figure 4: Frequency response of the measuring loop with the LB07 and LB16 paired with piezo sensor and CM-200.

3.2 Software Implementation

We implemented a Max/MSP application[1] to estimate the impulse response and use it to filter arbitrary sound sources. It provides

[1] http://stefanofasciani.com/padf.html

features to tune the system and to explore the creative musical application of the proposed approach. The computation of correlation and linear convolution (FIR filtering) are performed in the frequency domain to reduce the computational complexity, enabling multiple channels to efficiently run in parallel, also when using long filters (up to 4096 taps). In Figure 5 we show a simplified functional diagram of the signal processing for each channel that we implemented in the Max/MSP application. The graphical user interface is shown in Figure 6. Here we discuss the key features we provide to users. The system supports mono and stereo mode. In stereo mode we simultaneously estimate two $h(n)$, and these are used to filter the left and right output channels of the source. It is possible to use two transducers to sense acoustic vibration from the physical object in different locations, or to include also another actuator, filtering each channels with a different objects or different regions of a larger one.

The FFT size ranges from 512 to 4096 and it determines the length of the signal for the correlation analysis, therefore also the $h(n)$ length. This parameter and the sampling rate have a drastic impact to the filter estimation. Autocorrelation and whiteness measurement, as in (3), are visualized for one output and one input channel. These respectively represent the white noise $x(n)$ propagated by the actuator and the $y(n)$ captured by the sensor. Autocorrelation and whiteness measurement help to manually tune the pre-whitening biquad filter, which can be extended to a cascade of biquad filters.

For a more accurate estimation of $h(n)$ we use the running average on a number of consecutive measurements set by users. However this determines a slower system response, hence users have to find their optimal tradeoff. Moreover users can the further smooth the impulse response using a single pole low-pass filter (logarithmic sliding). The estimated $h(n)$ are displayed on screen together with their spectrums. Different signals can be routed to the main audio output: the filtered sound, the original sound source, or the IR for debugging purposes or for other sonic creative applications. Finally, we included an option to freeze the filters, taking a snapshot of the $h(n)$, which are cropped and stored in buffers, as visible in the bottom part of Figure 6. These static buffers can be read at variable rates (forward or backward) set by users to expand the sonic potential of our implementation.

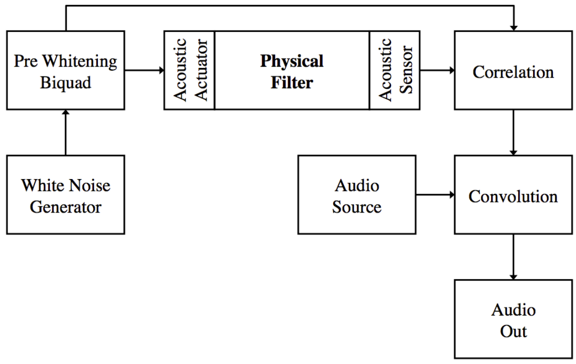

Figure 5: Simplified functional diagram of the signal processing for a single channel.

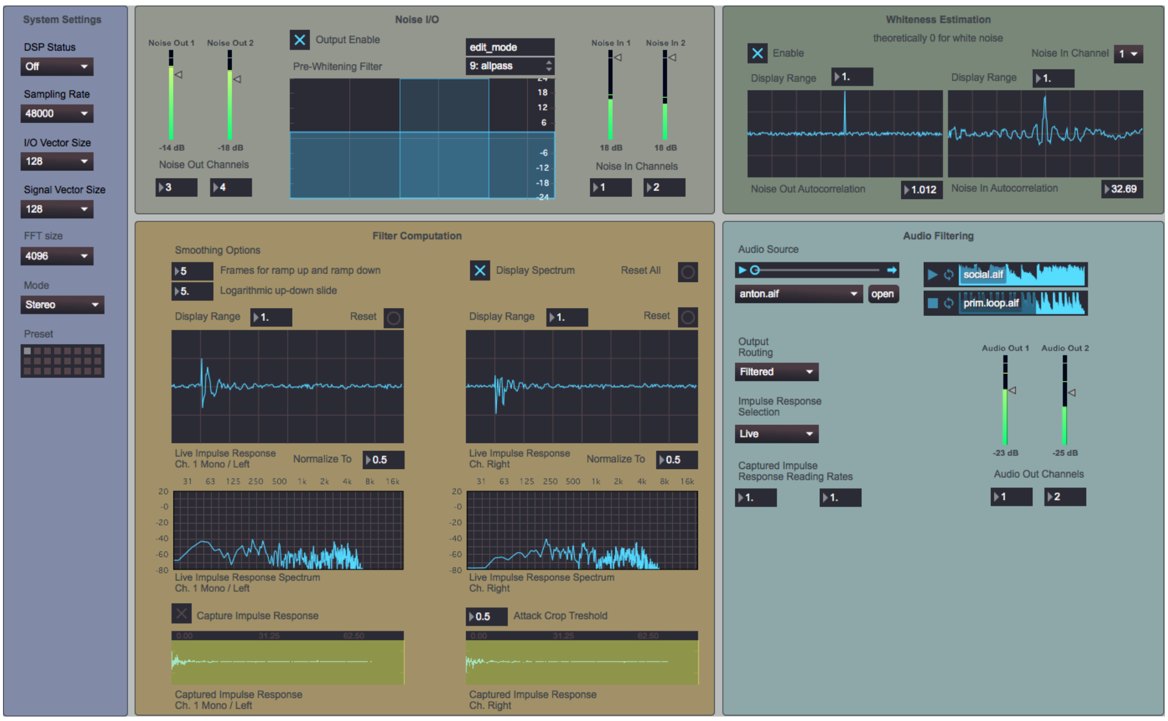

Figure 6: Graphical user interface of the Max/MSP system implementation.

4. APPLICATIONS

The approach we described in this paper allow to process audio signals with a tangible solid objects. The sonic result depends on the physical features of the object, such as material, size and shape. There are two ways for users to interact with the filtering system, hence to ways to modify the estimated object response. It is possible to alter the propagation of the acoustic vibration through the medium handling the physical object, for instance applying pressure or tapping the object in those regions between actuator and sensor. The density and the stiffness of the solid have a significant impact on the result of this approach. As expected, when interacting with thin and flexible surfaces we obtain wider filtering variations. We obtained interesting results when using object with internal cavities as physical filters (e.g. boxes), because these combine thin flexible surfaces, such as carton, plastic, or metal, with internal resonating chambers. The object mixture of air-borne and structure-borne vibrations provides interesting responses. A second way of tangible interaction is achieved moving sensor and actuators. The piezoelectric sensor and CM-200 have a high sensitivity and should not be handled while estimating the response. This can be moved offline only, and while the system is running it should be stick to a fixed position. Instead, the surface transducers can be freely and gradually moved to different position, determining continuous variations in the filter response. Moreover, users can apply vertical pressure to increase the energy transfer to the object, which determines an increase of the IR energy. With both actuators we used in this study we did not experience degrades in acoustic transduction during tangible interaction.

When lifting the surface transducer from object, the estimated $h(n)$ has all samples equal to zero, and therefore the resulting filter does not output any signal. This is equivalent to interrupting a signal path or opening a circuit. The path is restored when placing back the surface transducer. Therefore this approach can be used to signal patching and routing using physical objects. Our implementation already supports such application, and it can be scaled up to work with a higher number of channels, sensors and actuators.

Audio spatialization is also possible using our prototype. When using a single transducer, two sensors on the same object, and measuring the IRs of the independent filters applied to the left and right output channels, we obtain also a panning effect. The IR related to the sensor closer to the transducer has higher energy. The scenario is inverted when moving the transducer towards the other sensor. Since the energy of the impulse response is proportional to the loudness at the output of the filter, moving the transducer between two sensors we obtain linear panning. We extended this approach implementing also a system for quadraphonic spatialization, working with four piezoelectric sensors, four filters and four output channels. In Figure 7 we show an example of four sensors fixed at the corners of a small blackboard, representing a small-scale version of a rectangular room with four speakers located at the corners. Panning is determined by the position of the surface transducer moved by the user. The impulse responses of the four independent filters, one for each output channel, are also visible in the figure.

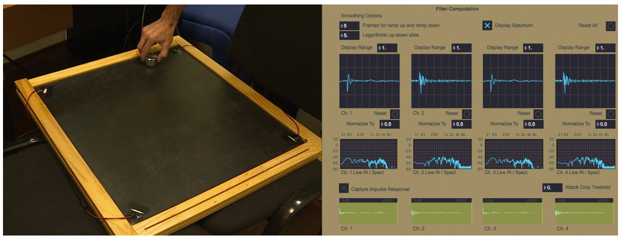

Figure 7: Quadraphonic panning and filtering application.

5. DISCUSSION AND FUTURE WORK

We presented a method and a proof of concept prototype to use physical objects as digital filters to process audio signals. Our approach allows users to touch and manipulate the filtering object without degrading the filtering process or generating noise bursts to the filtered sound. We introduced possible musical applications providing tangible interaction with computer-based digital audio processing systems. The white noise we inject into the physical object can be heard in the proximity of the system. The noise level is controllable and the minimum level depends on material and size of the filtering object. However, in musical context this side effect is negligible because the white noise is masked by the loudspeakers' sound or is eventually noticeable by the performer only.

Future work addressing limitation of the current implementation could provide significant usability improvements. The time to estimate an accurate impulse response of the physical object can be reduced using longer and overlapping signals for the correlation analysis, but this will increase the computational complexity of the system. The transparency of the current IR measuring loop presents plenty of room for improvement. A significant delay reduction can be achieved using an I/O device with drivers supporting smaller buffer size, and conversion analog-to-digital or digital-to-analog not based on sigma-delta modulators. In our proof of concept hardware we used low cost and small size sensors and actuators and we discussed their detrimental effect on the frequency response of the measuring loop. Automatic compensation techniques or adaptive filtering could provide a significant improvement in the estimation of the object response, as well using of sensor and actuator with a wider and transparent frequency response.

6. REFERENCES

[1] U. Zölzer, Digital audio signal processing, 2nd ed. Chichester U.K.: Wiley, 2008.

[2] D. Overholt, E. Berdahl, and R. Hamilton, "Advancements in actuated musical instruments," Organised Sound, vol. 16, no. 2, 2011.

[3] O. Lähdeoja, "Active Acoustic Instruments for Electronic Chamber Music," in Proceedings of the 16th international conference on New Interfaces for Musical Expression, Brisbane, Australia, 2016.

[4] K. Jathal and T. H. Park, "The HandSolo: A Hand Drum Controller for Natural Rhythm Entry and Production," in Proceedings of the 16th international conference on New Interfaces for Musical Expression, Brisbane, Australia, 2016.

[5] A. Crevoisier and P. Polotti, "Tangible Acoustic Interfaces and Their Applications for the Design of New Musical Instruments," in Proceedings of the 5th international conference on New Interfaces for Musical Expression, Singapore, 2005.

[6] G. Laput, E. Brockmeyer, S. E. Hudson, and C. Harrison, "Acoustruments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, New York, USA, 2015.

[7] V. Savage, A. Head, B. Hartmann, D. Goldman, G. Mysore, and W. Li, "Lamello: Passive Acoustic Sensing for Tangible Input Components," in CHI '15: CHI Conference on Human Factors in Computing Systems, 2015.

[8] M. Ono, B. Shizuki, and J. Tanaka, "Touch & Activate: Adding Interactivity to Existing Objects Using Active Acoustic Sensing," in Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology, New York, USA, 2013.

[9] S. Gupta, D. Morris, S. Patel, and D. Tan, "SoundWave: Using the Doppler Effect to Sense Gestures," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, USA, 2012.

[10] R. T. Beyer, Nonlinear Acoustics. Woodbury, NY: American Institute of Physics, 1974.