xCoAx 2020
Computation,
Communication,
Aesthetics & X
2020.xCoAx.org
Graz, Austria

Augmented Drums: Digital Enhancement of Rhythmic Improvisation

Matteo Amadio
matteoamadio@outlook.com
Conservatory of music C. Pollini,
Padova, Italy

Alberto Novello
albynovello@gmail.com
Conservatory of music C. Pollini,
Padova, Italy

Keywords: Augmented Instruments, Drum Set, Music, Interaction, Improvisation, Live Electronics.

This paper presents a set of real-time modules that digitally enhance the performance of a drummer. The modules extract rhythmic information from the multichannel audio acquired using simple microphones onto the different drum parts. Based on the predicted tempo, the modules generate complex patterns that can be manually controlled through high-level parameters or can be left automatic adapting to the drummer's specific style. Such an interactive system is intended mainly for an improvised solo performance confronting a human drummer with a computer; however, it could be effectively employed in improvisations with larger ensembles or installations.

1. Historical Background and Related Work

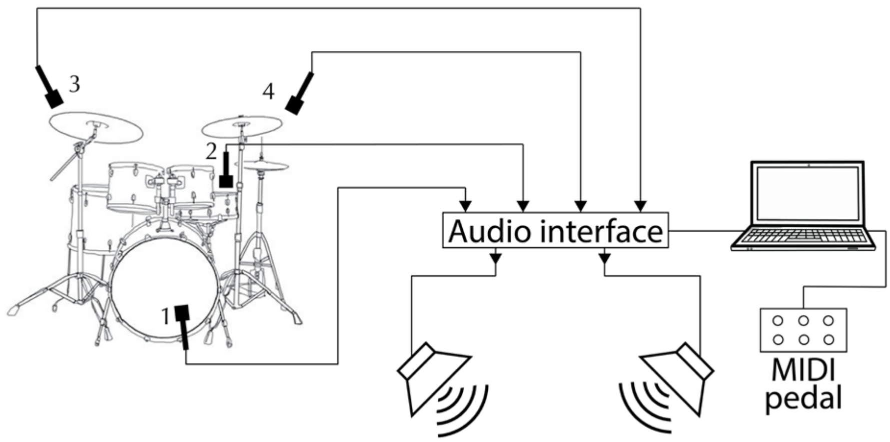
In the early years of electronic music, when the new technologies began to be considered as valid and innovative means for musical expression, percussion instruments had a fundamental role in the transition between the traditional, "keyboard-influenced" musical language and the "all-sound music of the future" (Cage 1937). Several composers considered percussion a great match with electronics because of their common characteristics. Both instrument families can generate a broad spectrum of unpitched sounds which encouraged composers to focus on timbre and rhythm. Another aspect is the modularity that characterises both setups of the percussionist and the electronic musician: the possibility to swap parts of an instrument chain has an impact on the creation process, and opens new possibilities in the performative approach (Cage 1937) (Varese 1966) (Stockhausen 1996). In the following years, the diffusion of digital technologies, computers and real time digital audio processing allowed the birth of new strategies for composition and live electronics. In the late 70's Chadabe started experimenting with digital "interactive composing systems": algorithms capable of responding in complex ways to the actions of the performer (Chadabe 1984). In his pieces *Solo* and *Rhythms*, for example, the performer provides high-level input data to the system that elaborates them in order to produce the whole musical output. The performer influences the final result but is unable to control each single event, placing himself in a position where he needs to listen and react to the gestures of the computer. In 1983 George Lewis was working at IRCAM developing an interactive software that could automatically generate instrumental music and also analyse the performance of human musicians in order to play along with them (IRCAM 1997) (Lewis 2018): this work established the base of his *Voyager* system. The idea of independence of the computer from the human performer led to the abolition of the "human leader/computer follower" hierarchy, in order to create the possibility to communicate only using the musical language (Lewis 2000). As a result, some intentions and emotions expressed by the human performer could also be found in the electronic performance, confirming the achievement of an authentic man-machine musical interaction. This last concept was pointed out also by Robert Rowe in the description of one of his early works in the field of human-machine musical interaction, *Hall of Mirrors* (Baisnee 1986). He describes the feedback loop generated by mutual imitation as two (or more) mirrors facing each other. Rowe developed his own interactive system named *Cypher*. In *Cypher* the user has to decide how the listener will interact with the player, allowing a high-level control on the actions of the software (Rowe 1990). In the '90s both *Cypher* and *Voyager* were modified to include the use of MIDI data as input and output, in order to easily process all the necessary data (Steinbeck 2019). In recent years, great progress has been made in the development of new interactive music systems and digital tools for the analysis of human performances. The application of these principles also led to the development of musical robots capable of interacting with human musicians in complex ways. The one-man Indian computer music system for example, applies machine listening techniques to various kind of signals produced by the human performance (Kapur 2006). The musician plays the *ESitar*, a modified sitar equipped with various kinds of sensors that provide gestural data: two examples are thumb pressure and fret detection. Also, the sound of the instrument is used as input and some accelerometer-based sensors are attached to the performer's body. This data is then used to synthesise sounds, control effects and generate beat accompaniment for the sitar performance. The result is an articulate and expressive system that reacts in interesting ways to the sitar performance, as can be seen in some demonstrative videos on Kapur's YouTube channel (Kapur 2010). Another musical robot is the one described in (Hoffmann 2010) which is an interactive marimba player that can improvise by listening and reacting to a human piano performance. Particularly interesting is the description of the third interaction module that focuses on the imitation of the rhythmic aspects of the human performance. This is done by weighing every point of the rhythmic grid of a bar by detecting if piano notes were played or not in those moments. These values are then stored in an array of probabilities that also keeps track of the past history. In *Agumenta* by Alessandro Guerri an electronic drum kit is used to send continuous information to a software that extracts general parameters like tempo, density and dynamics (Guerri 2015). This data is then used to control the generation of rhythmic patterns related to the performance of the drummer. In these drum kits, all the information about the notes played by the drummer are sent as MIDI messages, so their analysis can produce very accurate results. However, on the downside, the use of electronic drums implies a drastic reduction of the timbral expressivity of the instrument. The augmented drum kit developed by Christos Michalakos is based on an acoustic jazz drum kit with drum triggers, contact microphones and a speaker mounted on it (Michalakos 2012). The software includes sound processing and synthesis modules that can be turned on and off in various ways: manually by using a MIDI controller, automatically after the recognition of specific acoustic elements or through a pre-programmed cue. The variety of sound processing modules and the presence of different modalities of interaction with the software allow the drummer to control a rich electronic section, capable of enhancing the performance in several different ways, but requires a complex set of transducers, thus a long setup time. In this paper we present an approach for an augmented acoustic drum kit combining and expanding the ideas of the aforementioned systems. Our main goal is to sample the acoustic sound of the drums with all its nuances, then extract high-level features such as tempo, cue onsets and density to rhythmically enrich the performance by counterbalancing the drummer style. In the design process we prioritized the ease of setup and transportation of the system, requiring only a few common microphones and a sound card.

2. System Structure and Description

2.1. Hardware

We use the acoustic drum kit in two ways: as an acoustic source and as a control interface for the software. We also added a self-built MIDI foot controller with six buttons to control the desired software parameters. The acoustic sound of the drums is captured using four microphones: one placed inside the kick drum, one close to the snare, and two overheads. The number and the position of the overhead microphones can be modified, but the kick and snare microphones are essential for the software to work properly because their signals are analysed in order to retrieve important data regarding the performance of the drummer.

Fig. 1. Scheme of the hardware used.

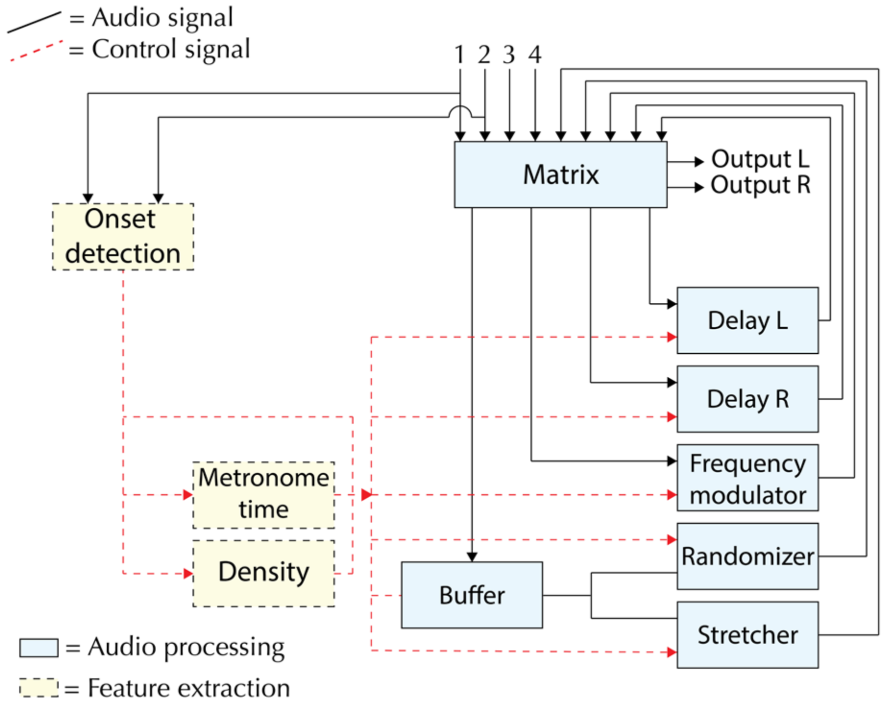


2.2. Software

The software is programmed in Max/MSP[1] and is divided in two main sections: the first one analyses the inputs to extract rhythmical information regarding the performer's style in order to generate control signals, while in the second one the audio coming from the microphones is digitally processed to produce rhythmical effects related to the data retrieved. This allows the drummer to easily influence the action of the effects by just modifying, for example, the groove or by changing the metric subdivision of a fill. The electronic effects are completely controlled by the data extracted from the performance, except for a few manual controls: *Reset* button which immediately sets the patch to its initial state, and a *Performance end* message that causes the electronic effects to slowly fade out and can be used whenever the drummer wants to end its exhibition. These messages can be sent through the use of the MIDI pedal and are not subject to any kind of variation operated by the software.

1. Max/Msp by Cycling '74: https://cycling74.com.

Fig. 2. The structure of the software.



Features Extraction

In the first section the sound captured by the kick and snare microphones is analysed in order to detect the onset of the single strokes and to send a message whenever one is identified. This is done by calculating the RMS amplitude over a 64 sample window and then routing the resulting signal to the *thresh~* object, which acts as a Schmitt trigger: an onset is identified whenever the signal exceeds the higher threshold, but the object won't be able to detect another one until the input falls beneath the lower threshold. After this data is extracted, it will be used to control the modules of the second section, but also reanalysed in order to retrieve information about the rhythmical aspects of the performance, in particular its tempo and density. *Tempo* is estimated by measuring the distance in milliseconds between each onset and the previous one. We then arrange this data in a histogram displaying the number of occurrences per value vs the time interval in milliseconds. The highest peaks are then selected and analysed to generate the tempo value. *Density* is estimated as the number of kick and snare onsets received over a two-seconds temporal window. The onset messages are routed into a counter: every time a message is received, the carried count is increased by one. At the same time, each message is delayed by 2 seconds and then routed back to the counter, this time causing it to decrease the count by one. By doing that, we obtain a parameter that is highly reactive to the actions of the drummer and computationally light. The resulting signal is then smoothed and normalised in order to reduce the value to a floating-point number varying between 0 and 1.

Audio Routing and Effects

All the calculated data is sent to the elements of the second section of the patch, which is formed by an audio routing matrix and five sound processing modules that we named *Delay*, *Frequency Modulator*, *Buffer*, *Stretcher* and *Randomizer*. In order to easily route all the audio signals and define their way from the microphones to the stereo output, we arranged a matrix that can be modified manually or in an automatic way. The *Delay* is made of a time varying delay line with two feedback loops. One of the delay lines uses a pitch shifter with a transposition ratio randomly chosen between 0.062 and 4 times of the original pitch. This effect generates ascending or descending "quantised glissandos", creating rhythmical and timbral variety whenever the effect is activated. The incoming audio signal is analysed using the *bonk~* [2] object in order to detect the presence of onset transients. If an onset is detected, the Delay effect can be triggered depending on a probability value set by the user. The delay time and the length of the amplitude envelope of the effect are chosen probabilistically from multiples or submultiples of the estimated tempo. The *Frequency Modulator* uses the audio input as the modulator signal for the frequency of a square wave oscillator. This signal is amplified and then the central frequency value is added; these two operations transform the input in order to bring it in a range where the modulation can be audible and effective on the carrier oscillator. The output of this effect is not continuous but, similarly to the delay, its amplitude envelope is triggered whenever the randomly generated number exceeds the gate imposed by the *Probability* value. The effect is triggered by snare onset messages. The *Randomizer* and *Stretcher* use a monophonic audio buffer which is the recording of the signal coming from one of the audio outputs of the matrix, and can operate in a manual or automatic way. When in automatic mode, the recording is armed by the kick onsets with a probability of 3% and the recording starts on the next tempo tick. Manual commands override the automatic mode. While recording to an audio buffer, we temporarily deactivate the other effects that use it for playback, in order to avoid clicks. At the end of each recording, the *Randomizer* analyses the buffer through the use of the *slice~* [3] object that divides the buffer in various segments based on the detection of transients in the signal. The buffer is rhythmically scrambled to create a playing queue where all the segments are permutated in time. The reproduction of this queue is automatically managed by comparing the *Density* parameter with a threshold set by the user. The queue is played whenever the *Density* parameter goes beneath the threshold and it is either paused when the threshold is exceeded or after five seconds. The *Stretcher* responds to the need of balancing all the short and percussive sounds with longer and softer ones. The audio output of this effect is mostly located in the lower region of the spectrum and is characterised by slow and

2. bonk~ : https://cycling74.com/forums/64-bit-versions-of-sigmund-fiddle-and-bonk.

3. slice~ : http://naotokui.net/2014/10/maxmsp-objects-on-github/.

often undefined attacks that create continuity in the performance. It uses the results of the analysis performed by the *slice~* object and stretches the audio segments by playing them at a much lower speed, causing them to be transposed and lengthened. The effect is triggered by the snare onset messages and the *Probability* parameter is controlled by the *Density* value. The probabilistic gate is similar to the one of the *Frequency Modulator.* Whenever the threshold is exceeded, we select a new segment to be played and generate a new random speed.

2.3. System Setup

Setting the system correctly is fundamental for the performance and it has to adapt to the playing style of the drummer. The gains for the kick and snare microphones are of critical importance as they influence the corresponding onset detection algorithms. Once the correct functioning of the onset detection algorithm is verified, it will be sufficient to reset the patch and recall the desired preset in order to start the performance. The entire procedure typically takes a few minutes.

3. Applicative Scenarios

The probabilistic gates make the interventions of the software unpredictable for the drummer. This aspect suggests an improvisational approach to the performance, in which the musician has to constantly modify his musical material in order to adapt to the electronic effects. The solo performance is the most intuitive way to use this system, however one could improvise with other musicians and use the drums to generate the control signals to process the overall sound. Another possible application of this research is an interactive installation in which the audience could modify the parameters of the processing engine in order to add or remove disturbing elements to the performance.

4. Conclusion and Future Work

We described an augmented drum system for interactive electroacoustic improvisation between a drummer and the computer. The computer listens to the acoustic sound of the drums and generates effects and rhythmical patterns based on the analysis of the performance. We wanted to establish a close relationship between the actions of the drummer and the software in order to create a strong sensation of interaction between drums and electronic parts. However, in order to achieve a more complex interaction between acoustic and electronic events, we decided to implement an architecture using probabilistic gates. The presence of an audio routing matrix and the possibility to store presets allows one to create different performance scenarios. For the current implementation, the hardware used is a basic drum kit and a few common microphones. We designed this system so that the musicians only need their own software, computer, and soundcard. However, improvement can be done in order to create variety in timbre by adding new elements to the kit. An important aspect we will be studying in the future is the type and positioning of the microphones. In particular we envision the use of two microphones that are placed far away from the drum kit in order to capture and process the room's ambient response. Another possible improvement could be the use of drum triggers to replace the onset detection algorithm to achieve more stability and simplifying the detection task. On the software side, we are thinking of new mapping possibilities through the predisposition of another routing matrix for control signals. This revision will introduce the possibility to easily set the action of the electronics to manual or automatic mode. In manual mode there will be the possibility to give control of the software to a second performer through the use of a MIDI controller as in the case of a traditional electroacoustic concert setup. New feature extraction algorithms will be created to provide further descriptions of the ongoing performance: a more precise beat tracker, spectral features to capture musical gestures, and a new rhythmic classifier. We imagine new effects and sound processing modules, such as dynamic filtering, ring modulation, granulation, and spatialization of the electronic sounds. The last addition to the software will be a cue-based interaction system. Cues will be useful to organise long improvisations, as well as to store presets and recall them whenever a specific sound environment is needed.

Fig. 3. Me vs Me? — Interactive drum performance https://youtu.be/taSawYaJ9YU.

References

Baisnee, P. F., Barrière,
J. B., Koechlin, O., Rowe, R.
1986. Real-time Interaction between Musicians and Computer: Live Performance Utilisations of the 4X Musical Workstation. In: ICMC 1986 Proceedings, pp. 238-239.

Cage, John
1937. *The Future of Music: Credo.*

Chadabe, Joel
1984. *Interactive Composing: an Overview.* In: *Computer Music Journal Vol. 8, No. 1*, pp. 22-27. DOI: 10.2307/3679894.

Guerri, Alessandro
2015. *AGUMENTA - an interactive drumming experience.* https://www.youtube.com/watch?time_continue=6&v=oQtuLBVuv64

Hoffman, G., Weinberg, G.
2010. *Gesture-based Human-Robot Jazz Improvisation.* In: IEEE International Conference on Robotics and Automation, pp. 582-587.

IRCAM Ressources
1997. *George Lewis.* http://brahms.ircam.fr/george-lewis#bio, http://brahms.ircam.fr/works/work/10042/.

Kapur, A., Tzanetakis, G. et al.
2006. *Towards the One-Man Indian Computer Music Performance System.*

Kapur, A
2010. *Experiments with ESitar, Electronics, Robotics, and Animation*. https://www.youtube.com/watch?v=l8UsRkWhNa4

Lewis, George E.
2000. *Too Many Notes: Computers, Complexity and Culture in Voyager*. In: *Leonardo Music Journal, Vol. 10*, pp. 33-39.

Lewis, George E.
2018. *Why Do We Want Our Computers to Improvise?* https://www.youtube.com/watch?v=wDP8FsjyCaA CTM Festival.

Michalakos, Christos
2012. *The Augmented Drum Kit: an Intuitive Approach to Live Electronic Percussion Performance.* ICMC

Rowe, Robert
1990. Feature Classification and Related Response in a Real-Time Interactive Music System. In: ICMC 1990 Proceedings, pp. 202-204.

Steinbeck, P.
2019. George Lewis's Voyager. In: The Routledge Companion to Jazz Studies, Routledge, pp. 261-270.

Stockhausen, Karlheinz, Jerome Kohl
1996. *Electroacoustic Performance Practice.* In: *Perspectives of New Music Vol. 34, No. 1*, pp. 75-105. DOI: 10.2307/833486.

Varese, Edgard, Chou Wen-chung
1966. *The Liberation of Sound.* In: *Perspective of New Music Vol. 5, No. 1.* DOI: 10.2307/832385.