FXIVE: INVESTIGATION AND IMPLEMENTATION OF A SOUND EFFECT SYNTHESIS SERVICE

Parham Bahadoran[1,2], Adán L. Benito[1,2], Will Buchanan[1,3] and Joshua D. Reiss[2]

[1]FXive, [2]Queen Mary University of London, [3]RPPtv
United Kingdom

ABSTRACT

FXive is a real-time online sound effect synthesis service. It replaces the need for reliance on sound effect sample libraries in sound design. In this paper, we describe the motivation, framework, sound synthesis techniques, interfaces and means of interaction for FXive. The system is comprised of a library of synthesis models, audio effects, post-processing tools, temporal and spatial placement functionality for the user to create scenes from scratch or from pre-existing pre-sets. The real-time nature allows the user to manipulate multiple parameters to shape the sound at the point of creation. Semantic descriptors are mapped to low level parameters in order to provide an intuitive means of user manipulation. Post-processing features allow for the auditory, temporal and spatial manipulation of these individual sound effects.

1. INTRODUCTION

High quality audio is essential to the creative industries. Sound effects are often defined as non-musical, non-speech sounds used in creative contexts such as film, games or virtual reality. Sound effects add vital cues for listeners and viewers across a range of media content. The process of sourcing such sounds is often achieved through Foley [1] or via use of large databases of pre-recorded audio samples.

The development of more realistic, interactive, immersive sound effects is an exciting and growing area of research. Sound synthesis, where sound is generated through artificial means, either in analogue or digital or a combination of the two, is a promising approach for higher quality sound effects. Many synthesis techniques completely avoid reliance on samples. Lloyd [2] demonstrated that synthesized sounds can often be indistinguishable from real recordings.

Procedural audio, where sound is created in real-time according to a set of programmatic rules and live inputs can be viewed as a subset of sound synthesis with a particular emphasis on control and interaction. In subjective evaluation experiments, Bottcher and Serafin [3] demonstrated that in an interactive gameplay environment, 71% of users found synthesis methods more entertaining than audio sampling.

Yet sound synthesis and procedural audio have yet to gain widespread popularity in practice. Synthesis of high quality effects is often too computationally expensive to be considered a viable alternative, and general-purpose synthesisers may result in the design process being arduous for sound designers.

The aim of this paper is to demonstrate that reliance on sample libraries can be avoided by the use of lightweight and versatile sound synthesis models. Such models do not rely on

stored samples and provide a rich range of sounds. Rather than searching libraries and attempting to fit samples to a desired goal, sounds can be shaped at the point of creation.

We introduce an online real-time sound effects synthesis platform, FXive [4,5], which demonstrates this concept. We present its different components, including sound source modules and post-processing tools. The platform reduces the technical knowledge required to produce high quality, realistic sound samples through synthesis while alleviating the restrictions imposed by pre-recorded audio.

A large spectrum of sounds is procedurally [6] generated through custom DSP synthesis algorithms. Table 1 lists all bespoke synthesis models that have been integrated into the platform. Many of the models have been evaluated in terms of their quality and realism using the framework described in [7]. Two further models outside the standard framework are also provided; a sinusoidal model allowing the user to synthesise sounds based on sample analysis, and a latent force model demonstrating this for selected impact and musical instrument sounds [8].

2. ARCHITECTURE

FXive's framework is written entirely in JavaScript and relies on a client-side architecture. Sound generation and manipulation is achieved through the employment of the Web Audio API (WAA)[9] and customised JavaScript processors and functions to handle interactions. Every synthesis model and audio effect is encapsulated using the JSAP[10] audio plugin standard and audio chains are handled by JSAP's Plugin Factory. This provides increased flexibility by allowing the creation of complex interconnected audio graphs in a number of configurations.

The framework takes handles the sequencing, loading and creation of different chains and resources and creates automatic connections between interface elements and plugin's parameters. Sounds generated using the models and post-processing tools can be recorded by the user and rendered to a stereo track to be downloaded as a wav file.

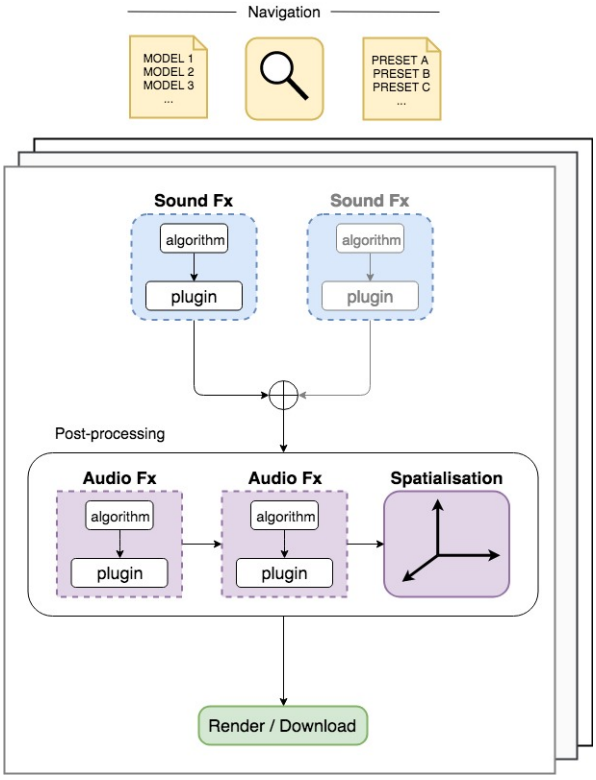

Figure 1 – Architecture of FXive

Two global chains are differentiated in this architecture: the sound-effect chain, composed by one or more sound effect plugins in parallel, and the post-processing chain, which is comprised of a series of audio processing and spatialisation tools (see Fig 1).

Sound Effects

All sounds found on FXive are built with real-time sound synthesis models tuned to synthesise an accurate representation of a class of sounds while exposing parameters that control the sound qualities. Though synthesis models are tailored to generate a a diverse range of sounds within a sound class and do not require use of samples, the availability of control parameters allows production of large banks of pre-sets. A portion of the interface for one such synthesis model, the siren, is depicted in Fig 2. Additional benefits over pre-recorded audio files include the ability to manipulate the audio source. This means characteristics such as pitch and duration may be determined by the user without the need to pitch shift and time stretch any audio. The mapping of low level features to perceptually intuitive controls allows for quick manipulation in order to fine-tune sound effects to suit a particular situation.

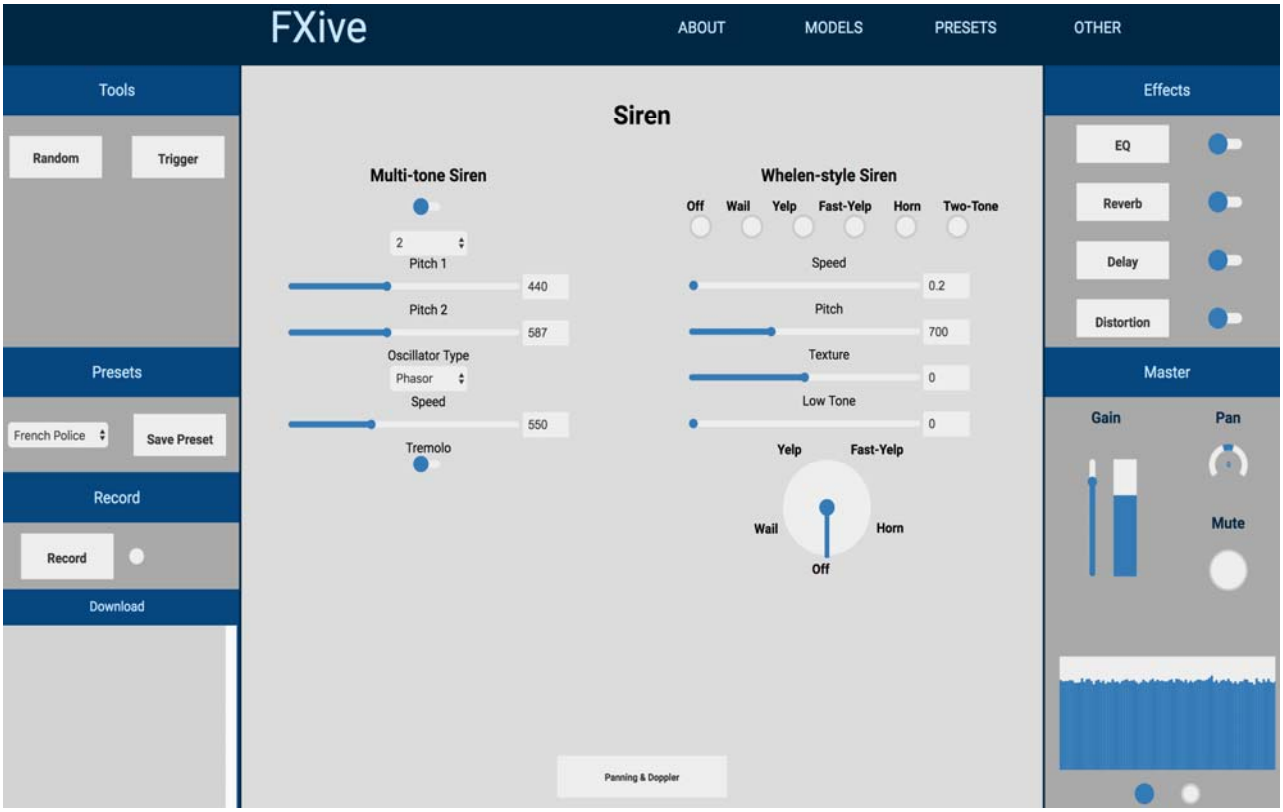

Figure 2 – Interface for the siren synthesis model. Not shown are audio effects, randomisation, timeline and spatialization.

Soundscapes

Sound scenes (or soundscapes) are offered within the framework as a conjunction of parallel synthesis models forming the sound effect chain. These combinations of sounds form more complex textures and illustrate what can be achieved with the platform. Each synthesis model has parameters exposed to the user in addition to overall control and processing of the soundscape. Example soundscapes models incorporated into FXive include a stormy day with rain, wind, stream and thunder models, and a night campsite scene with fire, wind, crickets and insects.

Navigation

The platform provides several forms of interaction and aims to provide an intuitive user experience and an easy path to access what different users may need. Different users may prefer to search for and download a sound effect in a few seconds, manipulate a synthesis model to edit and adjust a pre-set, or shape a new sound from scratch.

A pre-set search functionality is available from the homepage. It allows searching for a particular sound which will populate a list of pre-sets from all models matching the searched keyword, e.g. pre-determined parameters for the sound of an electric toothbrush within the synthesis model for DC motor sounds. A dedicated Pre-sets page also exists for browsing through all pre-sets from different models. There are currently over 130 pre-sets available.

For each pre-set there are options of either downloading a short snippet or visiting the model's page with the pre-set loaded and ready to manipulate or download. For a more detailed approach, a Models page is provided that can be browsed for a full list of all synthesis models (see Table 1). The models incorporate the full range of sound effects, including sound textures, impact sounds and soundscapes.

3. INTERACTIONS AND INTERFACES

Due to the nature of the development techniques discussed in Section 2, models can be regarded as self-contained sound source applications. This provides the flexibility to interact with them in different ways. One or more synthesis model can be chained together and manipulated using an interface. Furthermore, a URL query system can be used on an external host to trigger different actions from the models without the need for a user interface.

Parameter Controls

The low-level parameters of sound synthesis algorithms are mapped to semantically meaningful descriptors which are exposed to the user. Each model is given parameters which either describe a physical or real-world characteristic of the sound being generated (e.g. density of rain, crackling of fire) or a semantic descriptor commonly used for sound design (e.g. warmth, depth). This allows the user to manipulate the synthesis engine without the need for deep understanding of the algorithm being employed.

A graphical user interface is designed using NexusUI [11] with a variety of control objects for manipulating one or a combination of parameters. The objects are mostly those used in typical audio software, e.g. buttons, sliders and knobs. Other control objects are sometimes used to concisely convey elaborate control parameters. Device orientation and motion tracking is also proposed to create more complex combinations of parameters with temporal and spatial attributes. This allows aspects of sound synthesis to be controlled by movement of eligible devices (e.g. a mobile phone).

Table 1: Summary of models offered on the platform, their classification, and the works from which they were informed. 'O' indicates that it is based on an original design.

Category	Model	Basic Description	Based on
Action	Applause	Distributed and synchronised hand-claps based on filtered, shaped noise bursts	[12]
	Creaking door	Creaky sound of a door opening/closing, controllable door movement	[13, 14]
	Footsteps	Quasi-periodic sequence of shaped impacts on various surfaces	[15, 16]
	Shaker	Pseudorandom overlapping and adding of small grains of sound	[17]
	Swinging object	Aeroacoustic model of objects swung in the air based on compact sources	[18-20]
	Whistle	Noise, modulated and passed through resonant filters	[21]
Animal	Birds	FM and AM synthetic bird call sounds	[22]
	Insects	A composite model made of several families of insect sounds	[22, 23]
	Mammal sounds	Wolf howl and lion roaring based on physical models of vocal chords	[24]
Artificial	Alert	Frequency sweep modulated by fast rising and falling amplitude envelope, delay based resonators and filters	[22]
	Beep	Oscillators multiplied by control signal that alternates between 0 and 1 at fixed intervals	O
	Droid	Shaped oscillators imitating droids talking sound	[22]
	Lightsabre	Sheath and unsheath noise made of delayed noise for phase interaction. Hum and swing sounds from swept oscillator frequencies	[22]
	Ringtones	Four-tone frequency-modulated ring sounds	[22]
	Siren	Two Siren models with tunable parameters for many types of emergency vehicle siren sounds	O
	Telephone	Modern (DTMF) and old dialing and ringing phone sounds	[22]
	Teleport	Frequency modulated sawtooth sent to flanger and filterbank	[22]
	Whoosh	Bandpass-filtered noise with amplitude and frequency envelopes	O
Environmental	Aeolian tones	Aeroacoustic model of air passing through a string	[18, 25]
	Bubbles	Pseudorandom overlapping and adding of small grains of sound	[21]
	Electricity	Electric hum and spark noises with phasing	[22]
	Fire	Sound of fire, controllable crackle, hissing and lapping components	[22]
	Pouring water	Sound of pouring liquid into a container with bubbles and waterflow components	[22]
	Rain	Sound of rain, controllable density, rumble and ambience	[22]
	Stream	Sound of running water, controllable bubble density and sound texture	[22]
	Thunder	Combination of noise sources, staggered delays, waveshaping, and delay-based echo.	[22]
	Wind	Howling wind, gusts and wails based on air passing by different objects	[22]
Impact	Bouncing	Filtered noise mimics impact material. Bounce intervals calculated using coefficient of restitution according to object and initial height	O
	Explosion	Filtered white and pink noise modulated using exponential ADSR envelopes	O
	Gun	AK47 with shell detonation and gas explosion, generates variety of other gun sounds	[22]
	Gunshot	Modal synthesis with incorporated residual	[26]
	Metal impact	Modal synthesis of metal impacts	O
	Ricochet	Fast filter sweep of noise source to mimic a doppler effect	[22]
	Rocket	Sound of complete rocket launch, chamber, exit noise and rocket components	[22]
	Rolling	Physical model of rapid sequence of impact sounds	[27]
Mechanical	Clock	Filtered noise with delay-line based resonators	[22]
	Electric motor	Physical model of a DC motor	[28]
	Fan	Blade, Motor and noise components, produces a range of propeller and motor sounds	[22]
	Jet	Turbine comprised of clipped oscillators, burn comprised of filtered and clipped noise. Both modulated according to engine speed	[22]
	Propeller	Aeroacoustic model of propeller-powered aircraft sounds	[18, 29]
	Squeaky toy	Sequence of filtered, modulated noise bursts	O
	Switch	Rapid sequence of short click sounds	[22]
Soundscape	Factory	Adjustable sound scene of various machinery and artificial sounds	O
	Night scene	An adjustable sound scene comprised of light wind, fire and cricket sounds	O
	Stormy day	Sound scene created combining thunder, wind, rain and stream sounds	[22]

Audio Effects and Processors

A chain of common post-processing effects has been included accompanying each model for further sculpting and manipulation of the source. This chain is comprised of a distortion effect that permits precise shaping of its clipping curve, feedback delay, convolutional reverb that offers a selection of impulse responses, 5-band parametric equaliser, master gain and panning. Each audio effect can be bypassed accordingly and the sound effect muted if necessary.

Randomisation

Values of sliders on the interface can be randomised by the user via a dedicated set of controls. Randomisation takes part within the sliders' range and always employs the last user input as a reference point to calculate the random variations. A control parameter allows the user to select a percentage of the range over which the new values will span. A button permits triggering of randomisation at will.

Spatialisation

A Spatialisation feature is built using the WAA's PannerNode which places the sound source around the listener on a 3D Cartesian coordinate. This technique allows manipulating the location and movement of sound sources and may be used as a powerful sound design tool to further sculpt sound scenes.

The spatialisation model takes into account the sound objects' direction, orientation, velocity and distance from the listener. A two-part interface to place the sound source in a 3D coordinate system. A two dimensional slider is used to place the sound horizontally (left/right) and vertically (above/below) around the listener and an additional slider controls the third dimension (front/back).

Timeline

As shown in Figure 3, a timeline feature is implemented for triggering a model at specific times. Individual timeline tracks can be generated and used to trigger each of the parameters of the model. The current version of this feature provides discrete sequencer tracks with adjustable step size and sequence length, and an option for looping.

Each trigger-based parameter can be triggered at desired times by marking those points on the timeline of the trigger track. Parameters which can be used include model trigger parameters (e.g. fire, explode, hit, etc), any of the pre-sets and the Randomiser. The Randomiser can be triggered at specific intervals to applying a small percentage of randomisation to the model parameters and create a natural variation in the synthesis over time.

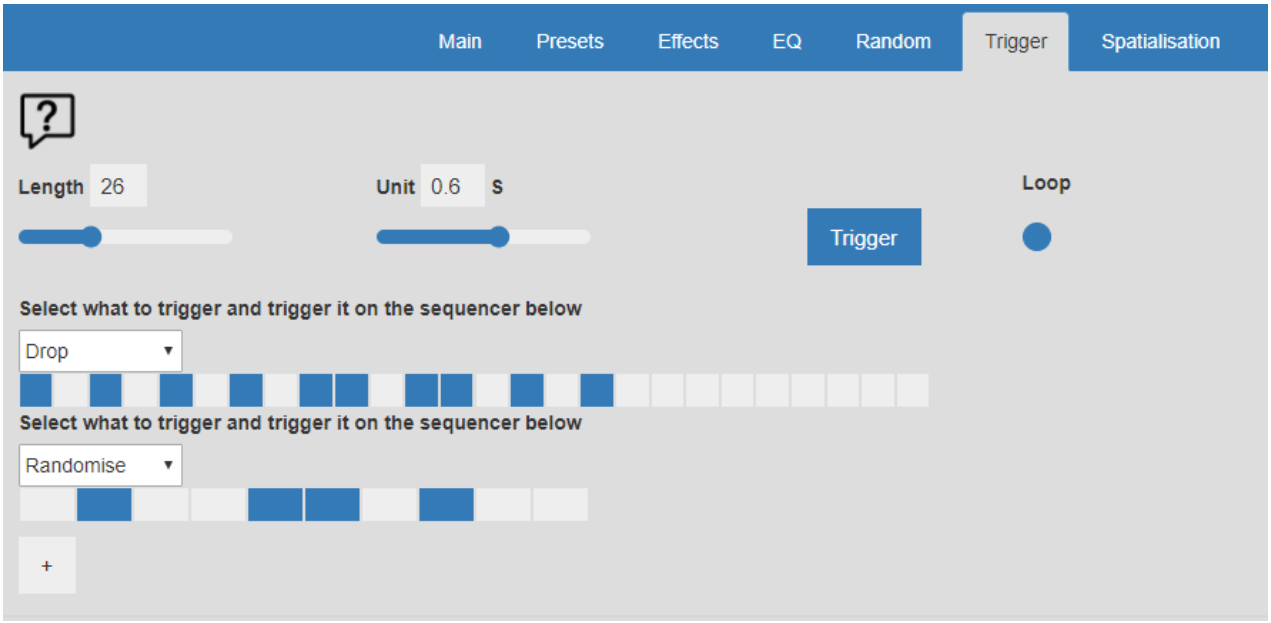

Figure 3 – Interface for the triggering of model parameters, allowing sounds and actions to be sequenced on a timeline.

Online query system

FXive's framework allows sound effects to be queried by a host via URL without the need for manual setting of their parameters. The framework decodes and interprets the query string and sets the different actions accordingly. The correct sequencing and loading of events related to both audio chains and processing is taken care of by the frame work. All interpretation and sequencing processes are automated by specific JavaScript routines once the URL has been loaded. Audio is still generated real-time on client-side and relies on browser compatibility with the WAA. This query system can be used to set pre-sets automatically, update and edit plugin states, trigger buttons and even download recorded excerpts.

4. SUMMARY

This paper introduced an online real-time sound effect synthesis service. The platform is built around a database of bespoke sound effect synthesis models, which are based on algorithms that have been fine-tuned and adapted for generating a large spectrum of sounds. Synthesis models are developed with the sound design process kept in mind, so meaningful control parameters are provided. This library of sound effect synthesis models, post-processing tools and temporal and spatial placement functionality provides the user with enough functionality to produce a wide range sound effects of high quality and flexibility, suitable for use in sound design projects.

Planned future developments include the ability to render output in multiple formats, especially object-based audio. Sound effects from sample libraries are typically delivered

as monaural or stereo, so an object-based format will move beyond another limitation of current approaches by allowing native rendering for immersive applications. A tagging feature would also allow user-generated and contributed parameter settings, thus greatly increasing the number of pre-sets.

FXive can be accessed online at http://fxive.com with the Chrome browser.

5. ACKNOWLEDGMENTS

This research was supported by the following InnovateUK grants in collaboration with RPPtv: Autonomous Systems for Sound Integration and GeneratioN (2017-18), SFX Synthesis Service (2016-18) and Real-time synthesised sound effects cloud service RTSFX (2015-16).

REFERENCES

[1] Ament, V. T., The Foley grail: The art of performing sound for film, games, and animation, CRC Press, 2014.

[2] Lloyd, D. et al.. "Sound synthesis for impact sounds in video games." ACM Symposium on Interactive 3D Graphics and Games, pp. 55-61, 2011.

[3] Niels Bottcher, N. and Serafin, S., "Design and evaluation of physically inspired models of sound effects in computer games," 35th Intl. Audio Engineering Society Conference: Audio for Games, London, 2009.

[4] Bahadoran, P., et al., "A System for Online Sound Effect Synthesis," 2017, UK Patent 1719854.0 (Pending).

[5] P. Bahadoran, et al., FXive: A Web Platform for Procedural Sound Synthesis, 144th Audio Engineering Society Conv., May 2018.

[6] Farnell, A., "An introduction to procedural audio and its application in computer games," Audio Mostly, v. 23, 2007.

[7] Moffat, D. and Reiss, J. D., "Perceptual Evaluation of Synthesized Sound Effects," ACM Transactions on Applied Perception, 15 (2), April 2018.

[8] Wilkinson, W., et al., "Latent force models for sound: Learning modal synthesis parameters and excitation functions from audio recordings," 20th International Conference on Digital Audio Effects, 2017.

[9] Adenot, P. and Wilson, C., "Web audio API," W3C Working Draft, December 2015.

[10] Jillings, N., et al., "JSAP: A plugin standard for the web audio API with intelligent functionality," Audio Engineering Society Convention 141, 2016.

[11] Taylor, B. et al., "Simplified Expressive Mobile Development with NexusUI, NexusUp, and NexusDrop." NIME, pp. 257–262, 2014.

[12] Adami, A., et al., "On Similarity and Density of Applause Sounds," Journal of the Audio Engineering Society, 65(11), pp. 897–913, 2017.

[13] Heinrichs, C., et al., "Human performance of computational sound models for immersive environments," The New Soundtrack, 4(2), pp. 139–155, 2014.

[14] Heinrichs, C. and McPherson, A., "Mapping and Interaction Strategies for Performing Environmental Sound," IEEE VR Workshop: Sonic Interaction in Virtual Environments (SIVE), 2014.

[15] Turchet, L., et al., "What do your footsteps sound like? An investigation on interactive footstep sounds adjustment." Applied Acoustics, 111, pp. 77–85, 2016.

[16] Farnell, A., "Marching onwards: procedural synthetic footsteps for video games and animation," Pure Data Convention, 2007.

[17] Cook, P., "Physically informed sonic modeling (phism): Synthesis of percussive sounds," Computer Music Journal, 21(3), pp. 38–49, 1997.

[18] Selfridge, R., et al., "Creating Real-Time Aeroacoustic Sound Effects Using Physically Derived Models," Journal of the Audio Engineering Society (to appear), 2018.

[19] Selfridge, R., et al., “Sound synthesis of objects swinging through air using physical models,” Applied Sciences, 7(11), p. 1177, 2017.
[20] Selfridge, R., et al., “Real-time Physical Model for Synthesis of Sword Swing Sounds,” 14th Sound and Music Computing Conference, 2017.
[21] Cook, P. R, Real sound synthesis for interactive applications, CRC Press, 2002.
[22] Farnell, Andy, Designing sound, MIT Press, 2010.
[23] Pekonen, J. and Jylhä, A., “3-D Sound Synthesis of a Honeybee Swarm,” Audio Eng. Soc. Conv. 127, 2009.
[24] Wilkinson, W. and Reiss, J. D, “A Synthesis Model for Mammalian Vocalization Sound Effects,” 61st Audio Engineering Society Conference: Audio for Games, 2016.
[25] Selfridge, R., et al., “Physically derived synthesis model of an Aeolian tone,” Audio Engineering Society Convention 141, 2016.
[26] Mengual, L., et al., “Modal Synthesis of Weapon Sounds,” 61st Audio Engineering Society Conf.: Audio for Games, 2016.
[27] Rocchesso, D. and Fontana, Federico, The sounding object, Mondo estremo, 2003.
[28] Hendry, S. and Reiss, J. D., “Physical Modeling and Synthesis of Motor Noise for Replication of a Sound Effects Library,” 129th AES Convention, San Francisco, 2010.
[29] Selfridge, R., et al., “Physically Derived Sound Synthesis Model of a Propeller,” 12th Audio Mostly Conf., p. 16, ACM, 2017.