crossSampler – A Convolution Cross-Synthesis Musical Instrument

Jimmy Eadie
Dept. Electronic and Electrical Engineering
Trinity College Dublin
Ireland
eadiejj@tcd.ie

***Abstract*—Convolutional Cross Synthesis (CCS) has emerged as a powerful technique for blending input signals to create hybrid sounds. It has significantly expanded the horizons of digital signal processing, enabling artists to explore novel audio effects. However, the conventional applications of CCS primarily revolve around reverberation and room simulation, rather than being utilized as a creative synthesis method. In this paper, we present the design of a digital instrument *CrossSampler* that harnesses a parametric approach to convolution cross synthesis which involves using adjustable parameters to control the blending of audio signals through convolution. These parameters allow for customization of the resulting sound, offering greater creative control and flexibility. It enables users to shape the output by manipulating factors such as duration, intensity, and spectral characteristics. This approach facilitates experimentation and exploration in sound design and opens new sonic possibilities.**

Keywords—Convolution, Synthesis, Sampling, Virtual Instrument)

I. Introduction

The project explores the convergence of convolution and sampling techniques in a parametric convolution cross synthesis (CCS) instrument developed using the JUCE framework. Its main objective is to design and implement a VST instrument that leverages CCS as a foundation for sound synthesis. The instrument empowers users with real-time control over the synthesis process, facilitating the creation of unique sounds. By incorporating user audio files, the instrument opens avenues for sound designers and musicians to delve into timbre exploration and sound manipulation. The proposed instrument capitalizes on the potential of convolution and sampling techniques, allowing for the generation of fresh and innovative sounds. Through its parametric framework, the instrument enables precise control over various synthesis parameters, facilitating the fine-tuning of desired audio characteristics. This novel approach contributes to the field of digital signal processing, broadening the creative possibilities for musicians and sound designers. By integrating a sampler instrument with a parametric CCS approach, the instrument empowers musicians to shape the synthesis process, incorporate their own audio files, and create original sounds. The paper provides a comprehensive analysis of the instrument's design, capabilities, and potential for sound design and innovation in digital music production [1]. The utilization of JUCE, a powerful framework for audio development, ensures the efficient and flexible realization of the CCS VST instrument.

II. Cross-Synthesis

Cross-synthesis is a versatile family of digital signal processing techniques that involve the combination of two or more signals, resulting in a hybrid signal that retains specific timbral properties from the original signals. Can be represented by:

$$X_{\text{output}}(k) = X_{\text{input 1}}(k) \cdot X_{\text{input 2}}(k) \qquad (2.0a)$$

In this equation 2.0a, Xoutput (k) represents the complex frequency-domain coefficient at index k of the output signal. Xinput1(k) and Xinput2 (k) represent the complex frequency-domain coefficients of the input signals 1 and 2, respectively. The operation · denotes the multiplication of the complex coefficients. Cross-synthesis has diverse applications in digital audio, such as the vocoder, which applies parameters from a "voice" signal to a "receiver" signal, creating a distinct vocal effect [2]. Its origins can be traced back to the late 1930s research at Bell Telephone Laboratories, leading to the development of the vocoder for speech compression and synthesis [3]. The phase vocoder, introduced in the late 1960s, enabled digital analysis and re-synthesis of audio signals, facilitating frequency domain processing [4]. These advancements laid the groundwork for purely digital cross-synthesis, including the phase vocoder in the late 1960s. The breakthrough in cross-synthesis led to novel musical applications, renowned for generating new and unique timbres [5]. When performing cross-synthesis, the output signal inherits characteristics from the input signals, primarily relying on amplitude or spectral envelopes. In simple terms, cross-synthesis refers to imparting the sonic qualities of one sound onto another. However, it's important to note that the resulting output signal may not directly resemble either input signal. In the 1980s, Richard Boulanger demonstrated the musical potential of convolution, expanding the concept of impulse response to encompass any recorded sound, thereby broadening the scope of convolution cross-synthesis beyond traditional reverberation contexts [6]. Curtis Roads and William Gardner made significant contributions to the field in the 1990s, exploring musical applications and proposing efficient methods for real-time implementation of cross-synthesis [7] [8].

III. Convolution

Convolution is a mathematical operation that combines input signals through a weighted summation process. It plays a fundamental role in signal processing by generating an output signal that captures the interaction between the input signals. Mathematically, it involves integrating the product of the two functions, with one function being reversed and shifted [9] :

$$y(n) = \sum_{m=-\infty}^{\infty} x(m) \cdot h(n-m) \qquad (2.0b)$$

In the above equation, 2.0b, y(n) represents the output signal at index n resulting from the convolution operation. x(m) and h(n-m) represent the input signal and impulse response, respectively, with m and n denoting the indices of the respective signals. The convolution operation involves multiplying each sample of the input signal, x(m), with a time-reversed and shifted version of the impulse response, h(n-m). The resulting products are then summed over all possible values of m to obtain the output signal, y(n). This equation captures the essence of convolution by mathematically describing the process of combining input signals through the weighted summation of their samples.

Filters play a crucial role in digital signal processing [10]. They are systems that generate an output signal based on an input signal [11]. Test signals like the impulse signal, consisting of a single maximum amplitude sample, are commonly used to evaluate and understand digital signal processing systems. The resulting output when an impulse signal is processed is known as the impulse response, representing the system's behaviour in the time domain [9]. The impulse response can be analysed in the frequency domain by computing the Fast Fourier Transform (FFT) to obtain the frequency response. Both the impulse response and frequency response provide insights into the system's characteristics. To understand the relationship between convolution and a system's impulse response, it is essential to grasp the concept of a linear, time-invariant system. A linear system adheres to additivity and homogeneity of degree principles [11]. Specifically, a system f(x) satisfies linearity if the following equations hold true:

$$f(x + y) = f(x) + f(y) \tag{2.1}$$

$$f(ax) = af(x) \tag{2.2}$$

$$f(ax + by) = af(x) + bf(y) \tag{2.3}$$

In audio signal processing, time invariance implies that if an input signal is applied to a system at time n seconds and then again at time n + t seconds, the output of the system will be nearly identical in both cases. The only difference is that the output in the second case will be shifted in time by t seconds [11]. A system x(n) → f[x] → y(n) satisfies the constraint of time invariance if the following equation holds true:

$$f[x(n-t)] = y(n-t) \tag{2.4}$$

The sound transformation effect of a linear, shift-invariant system is achieved by convolving the input signal with its impulse response. When an arbitrary audio signal is provided as input, the system produces an output. However, when an impulse signal is used as input, the system generates its impulse response. The output signal y(n) can be obtained by convolving the input signal f(n) with the impulse response i(n) [9]. This relationship is represented by equations (2.5-2.7) as follows:

Relationship Between a Function $x(n) \rightarrow S[x] \rightarrow y(n)$ and Impulse Response $u(n)$

$$S[x(n)] = y(n), \tag{2.5}$$

$$S[u(n)] = i(n), \tag{2.6}$$

$$x(n) * i(n) = y(n) \tag{2.7}$$

where:
u(n) is a sequence of samples u(n) = {1,0,0,0, . . . }
x(n) is an arbitrary sequence of samples
S[x] is a linear, time-invariant function which transforms its input, creating an output y(n)

IV. Basic Operations of Convolution

Examining how basic convolution operations transform an arbitrary sound can provide insight into the convolution of two musical signals [12]. One fundamental case involves the cross-synthesis of an arbitrary signal a(n) with an impulse signal u(n), where u(n) is defined as {1, 0, 0, 0, ...} [13]. In this scenario, convolution becomes an identity operation, as demonstrated by Equation 2.8:

$$a[n] * u[n] = a[n] \tag{2.8}$$

When the impulse signal is scaled by a constant s, the resulting convolution is equivalent to scaling the signal a[n] by the same constant s [13]. This relationship is illustrated in Equation 2.9:

$$a[n] * (s \times u[n]) = s \times a[n] \tag{2.9}$$

Another important fundamental case of convolution is the use of a delay function. When the impulse signal u(n) is shifted in time by t samples, a delay effect is achieved. In this case, the convolved signal is identical to the input signal a[n] shifted in time by t samples [13]:

$$a[n] * u[n-t] = a[n-t] \tag{2.10}$$

Every convolutional result can be understood as a sequence of mathematical operations, as depicted in Equation 2.9 and Equation 2.10 [13]. For example, the recurrence or echo effect of a signal a[n] can be achieved by convolving it with another signal b[n], where b[n] contains two well-spaced

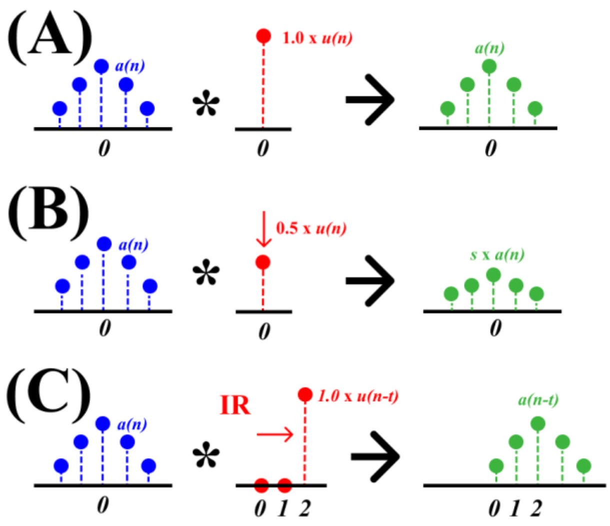

Fig. 1. Basic convolution operations from Equations 2.8, 2.9, and 2.10 corresponding to diagrams (A), (B), and (C) respectively.

impulses. If signal b[n] is replaced by signal c[n], which contains two impulses closer together than the length of signal a[n], the scaled repetitions of a[n] start to overlap as the impulses in c[n] get closer. This operation results in the convolution a[n] ∗ c[n], where the signal a[n] appears smeared in the time domain due to its interaction with a delayed instance of itself. Both the echo effect and time smearing effect are illustrated in Fig. 1 and Fig. 2.

V. Applications of Convolution in Digital Audio

Convolution offers transformative effects in cross-rhythm mapping, spatialization, filtering, and reverberation [14]. It has wide-ranging applications in music and audio engineering[15]. Convolution allows simulation of various systems' effects on audio signals, such as reverberation and analog hardware modelling[16]. Convolutional reverberation effects are popular among musicians and commercially available from software developers; Eventide, Waves, and Altiverb and Izotope and many more. Convolution allows for modelling physical spaces and electronic components. By considering any audio signal as the impulse response of a hypothetical system, cross-synthesis of two signals can create a new hybrid signal, offering diverse sonic possibilities.

VI. Convolution in Practise

For two sequences of samples, f[n] and g[n], each with lengths N and M respectively, equation 2.11 provides the mathematical definition of convolution. It's worth noting that convolution is a commutative operation, meaning that f ∗ g is equivalent to g ∗ f [13]:

$$(f * g)[n] = \sum_{m=0}^{M-1} f[n-m]g[m] \qquad (2.11)$$

The method of calculating the convolution of two signals, as defined in equation 2.11, is referred to as direct convolution. In this operation, the sum of each (m) × g(n - m) is computed at each point. To ensure a normalized output, the result of the convolution operation is typically scaled. It is important to note that the output sequence of the convolution operation will always be equal to or longer than the length of either of the input signals [9]. The length of the output sequence produced by direct convolution can be determined as follows:

$$length(output) = N + M - 1 \qquad (2.12)$$

The law of convolution is a fundamental principle in signal processing. It states that the Fourier transform of the convolution of two signals is equal to the point-wise

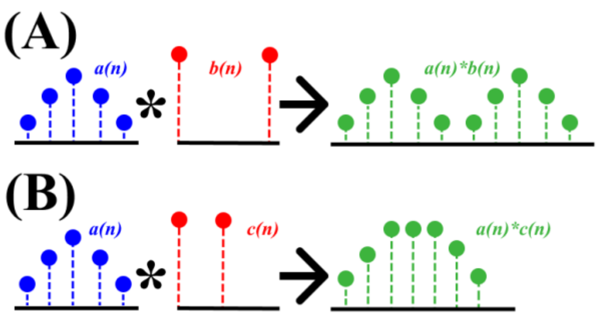

Fig. 2. Convolution operations achieving (A) echo and (B) time smearing effects.

multiplication of their respective Fourier transforms. This relationship is described by Equation 2.13:

$$F(f * g) = F(f) \cdot F(g) \qquad (2.13)$$

Direct convolution can be computationally expensive, but fast convolution, which utilizes the Fast Fourier Transform (FFT) algorithm, offers a more efficient implementation. Fast convolution reduces the complexity from N^2 operations to N log(N) operations for signals longer than 32 samples [4]. Direct convolution of two 4-second audio signals sampled at 44.1kHz requires 31,116,960,000 multiply-add operations, while fast convolution can achieve the same result with less than 2,722,000 operations. In terms of computational intensity, direct convolution is more than 11,400 times more demanding than fast convolution for longer audio signals [17].

VII. Convolutional Cross Synthesis

Convolution is a cross-synthesis technique that combines input signals, emphasizing shared frequencies and altering the resulting signal's spectral and temporal characteristics [18]. The operation involves the multiplication of samples in both the frequency and time domains, resulting in a cross-synthesized signal influenced by the input signals' frequency and temporal qualities. Convolution accentuates attributes such as brightness in signals with similar timbres and influences the temporal characteristics, including the attack [14]. Spectrograms, which represent the average frequency content of a signal over time, provide insights into the spectral and temporal effects of convolution in CCS.

Fig. 3 demonstrates the spectrogram and waveform representations of two convolved audio signals, revealing the transformative spectral effects of convolution. Convolution in the time domain is evident in waveform representations of input signals and their resulting convolution. Spectrally, convolution captures the average amplitude of each frequency throughout the signal, accommodating dynamic audio signals. By manipulating frequency-time characteristics, convolution enables diverse sonic modifications, such as modulation, resonance models, attack smoothing, echoes, room simulation, spectral smearing, and reverberation [13].

VIII. Problems Associated with Traditional CCS

The potential for convolving pairs of discrete audio signals is virtually limitless, offering an extensive range of digital

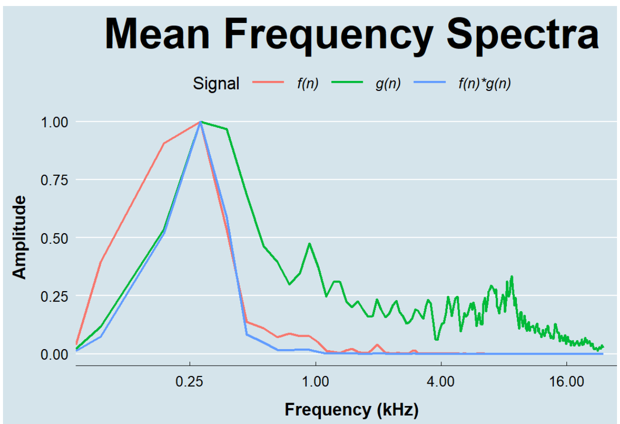

*Fig. 3. Spectrogram of(n), g(n), and f(n)*g(n) with normalised amplitudes*

audio effects. However, traditional implementations of (CCS) have limitations as a musical tool for sound design. One key limitation is that conventional CCS approaches produce a single output signal regardless of the number of input signals. This means that musicians have limited control over the synthesized signal. The only way to affect the output of CCS is by modifying the input signals. As noted by Roads [13] there is no straightforward method to weight a convolutional effect towards a specific input signal through signal transformation. Simply reducing the amplitude of one signal results in overall attenuation of the convolutional output, without altering the balance between the signals. Convolution in CCS tends to accentuate low frequencies and attenuate high frequencies, resulting in darker synthesized sounds. The constructive and destructive interference of magnitude spectra during convolution creates resonances, with low-frequency peak interference producing higher amplitudes. This characteristic exacerbates the lack of control in traditional CCS implementations.

IX. Parametrized Convolution Techniques for Cross-Synthesis

Traditional implementations of CCS lack flexibility and control over the output [9], [13], [17], [19]. However, a weighted approach to CCS introduced by Donahue et al. [17] addresses these limitations. This parameterized style of convolution, denoted as *ˆ, allows musicians to have greater control over the convolutional output. Equation 2.14 provides a comprehensive definition of this extended approach to CCS. Let F(f *ˆg) denote the complex result of the operation between signals f and g.

Magnitude Calculation: (2.14a)

||F(f *ˆg)|| = (||F(f)||^p · ||F(g)||^(1-p))^2q

Phase Calculation: (2.14b)

$$\angle F(f \hat{*} g) = 2s(r\angle F(f) + (1 - r)\angle F(g))$$

In the above equations, ||F(f)|| represents the magnitude and ∠F(f) represents the phase angle of the complex frequency-domain coefficient of signal f. Similarly, ||F(g)|| represents the magnitude and ∠F(g) represents the phase angle of the complex frequency-domain coefficient of signal g. The constants p, q, r, and s determine the weights and scaling factors for the magnitude and phase calculations. To achieve a brighter sound in CCS, [17] introduced a parameter q that controls the interference effect when multiplying spectra. As q decreases, the frequency response becomes smoother, resulting in a flatter spectrum. When q increases, the constructive interference between spectra intensifies, leading to resonant sounds resembling simple tones.

$$||F(f \hat{*} g)|| = (||F(f)|| \cdot ||F(g)||)q \qquad (2.15)$$

The extended approach to convolution allows for the emphasis of one sound over the other in cross-synthesis. Fast Convolution separates the magnitude and phase spectra of the input signals, enabling separate weighting. The extended convolutional output can be skewed towards one magnitude spectrum while the other becomes more noise-like. The parameter q is multiplied by 2 to maintain the brightness scale.

$$||F(f \hat{*} g)|| = (||F(f)||p \cdot ||F(g)||(1-p))2q \qquad (2.16)$$

The extended convolution approach includes parameter r to control the combination of phase spectra. Equation 2.17 defines this extension, with r = 1 emphasizing the phase spectrum of signal f. The multiplication coefficient of 2 ensures consistency with Fast Convolution, and when r = 0.5, the combined phase spectrum resembles that of Fast Convolution.

$$\angle F(f \hat{*} g) = r\angle F(f) + (1 - r)\angle F(g) \qquad (2.17)$$

Donahue et al. (2016) introduces parameter s to the formula as described in Equation 2.18. This parameter influences the combined phase spectrum ∠F(f *ˆ g), similar to how parameter q affects the combined magnitude spectrum. When s = 1, no phase scattering occurs. As s approaches zero, more of the combined phase spectrum is eliminated, resulting in substantial time-domain cancellation and a signal resembling an impulse. When q increases beyond one, the combined phase spectrum converges towards a uniform phase distribution, creating drone-like ambient sounds with a flat temporal envelope.

$$\angle F(f \hat{*} g) = 2s(r\angle F(f) + (1 - r)\angle F(g)) \qquad (2.18)$$

The extensions to CCS involve nonlinear manipulation of the phase and magnitude components of Fourier transforms F(f) and F(g). This manipulation does not preserve the exact frequency component structure, leading to a circular shifting of the onset of a zero-padded signal, as depicted in Fig. It is important to note that such a signal will always exhibit cyclical behaviour.

The CCS method see Fig 4 by Donahue et al. [17] provides parameterized control over cross-synthesis. It offers musicians the ability to emphasize specific spectra, resulting in unique sonic possibilities. However, artifacts such as circular shifting in zero-padded signals arise, which can

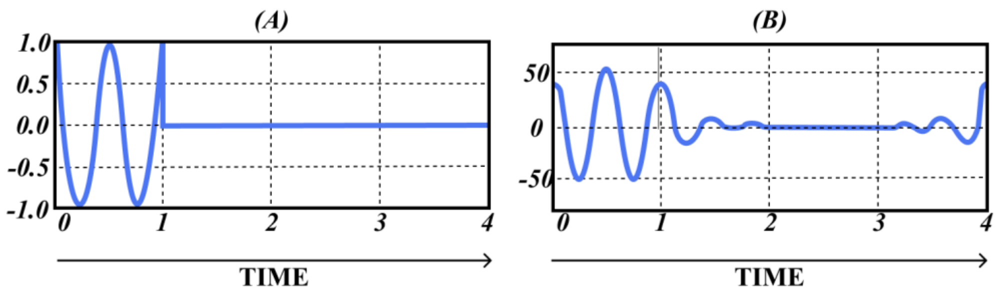

Fig. 4. Circular Artifact resulting from Nonlinear Manipulation of Phase and Magnitude Components [Donahue et al. (2016). Figure (A) shows a zero padded sample reconstructed after a linear Fourier transform. Figure (B) shows the reconstruction of a samples.

contribute to interesting musical effects. Direct manipulation of these artifacts is challenging, but fast convolution with specific parameter values allows some control over the outcome. The artifact of circular shifting in extended CCS shapes the frequency and temporal characteristics of the resulting sound. While it can create interesting musical effects, manipulating these artifacts directly is challenging, limiting control for musicians.

X. CrossSampler Instrument Overview

The current prototype of *CrossSampler* system consists of two sections the sampler instrument and the convolution engine. Both are launched when the VST[1] is loaded. The sampler plug-in triggers loaded sounds via MIDI input in a chosen DAW. The audio output of the sampler is directly routed to the input of the convolution engine, convolution synthesis in real time takes place and outputs to the track the VST is loaded on, see Fig. 5 for signal flow.

Fig. 6 shows the user interface for the sampler instrument with usual parameter controls accessed through knobs, most of these controls are found on many VST samplers [20]. Users can add local files to the input section by clicking the import range in the file manager section. The sampler allows playback control, and the waveform display component visualizes the file's waveform. This plug-in component functions as a digital sampler instrument, where an audio file or sample is loaded and stored for playback at different pitches.

Fig. 7 shows the user interface for the convolution engine which outputs the convolved signal from both the engine and the sampler to the selected output track. The program utilizes interactive objects such as buttons and sliders inherited from the JUCE SDK[2]. While traditional reverberation techniques typically rely on short impulse responses, this instrument allows for the loading of extended audio files, up to 20 seconds in length. This expanded capability enables sound designers to explore new possibilities in creating unique timbres and sonic textures. By leveraging the convolution process with longer audio segments, the instrument opens up avenues for experimentation and artistic exploration,

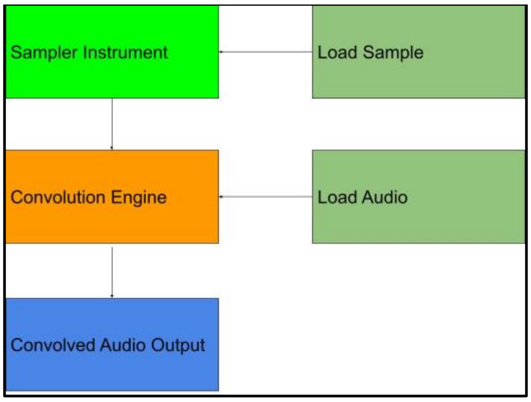

Fig. 5. Signal Flow

empowering sound designers to craft intricate and captivating soundscapes. With its focus on providing extended audio loading capabilities, this convolution sampler instrument is specifically tailored to meet the needs of sound designers seeking to push the boundaries of sonic creativity.

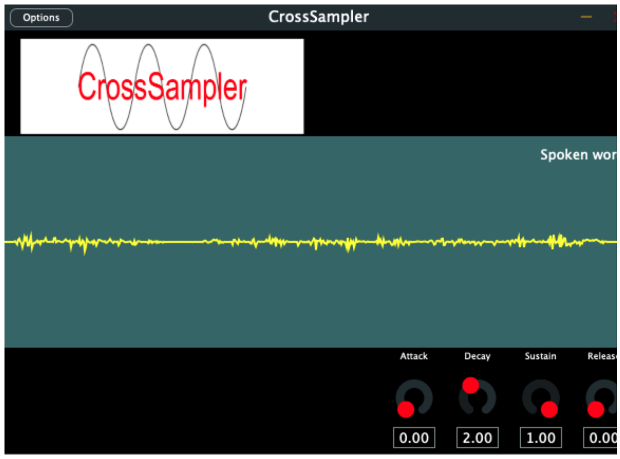

Fig. 6. Sampler Instrument displays control parameters and sample loading.

XI. CCS Implementation of VST Instrument

The method in the JUCE framework for convolution is the **process()** method provided by the **juce::dsp::Convolution** class. This method performs the actual convolution operation on the input audio data. The **processAudioBuffer()** function

Fig. 7. Convolution engine.

takes an input audio buffer (**inputBuffer**) and an output audio buffer (**outputBuffer**). The **process()** method is called on the **convolution** object, passing the array of read pointers for the input buffer (**inputBuffer.getArrayOfReadPointers()** and the array of write pointers for the output buffer (**outputBuffer.getArrayOfWritePointers()**.

The **crossSynthesis()** function takes two input audio buffers (**inputBuffer1** and **inputBuffer2**) and performs convolution cross synthesis on them. The resulting cross-synthesized audio is written to the **outputBuffer**. The function iterates through each channel and each sample of the input buffers. It adds the samples from both input buffers and passes the resulting cross-synthesized sample through the **convolution.processSample()** method to apply convolution

[1] https://www.techopedia.com/definition/266/virtual-studio-technology-vst

[2] https://docs.juce.com/master/index.html

using the loaded impulse response. Below are the spectrograms which display signals loaded into an instance of *CrossSampler*. Fig. 8 Speech loaded into the sampler engine. Fig. 9 Drums loaded into the Convolution engine, and the convolved output is shown in Fig. 10. This displays the effect spectrally of CCS of both signals.

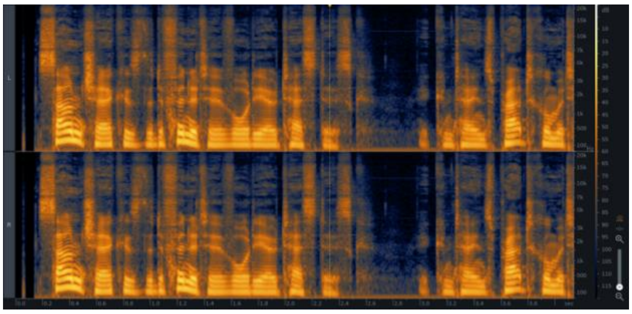

Fig. 8 Spectrogram a

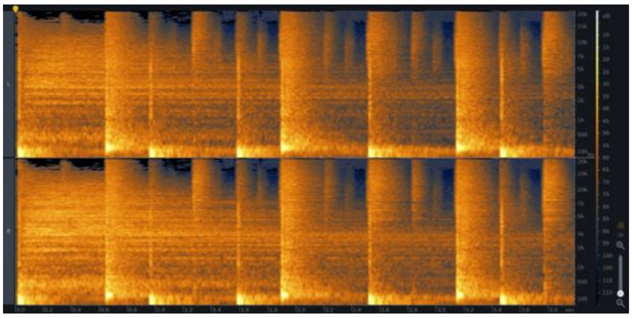

Fig. 9 Spectrogram b

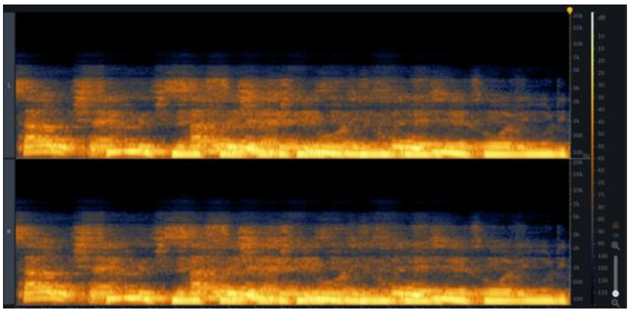

Fig. 10 Spectrogram c

XII. Conclusion

Convolutional Cross Synthesis (CCS) has emerged as a powerful technique in digital signal processing, allowing for the blending of input signals to create hybrid sounds. While CCS has traditionally been associated with reverberation and room simulation effects, this paper has presented the design of a parametric convolution cross synthesis instrument, called *CrossSampler*, which extends the application of CCS to sound synthesis. By incorporating adjustable parameters, *CrossSampler* provides users with enhanced control over the blending of audio signals through convolution, enabling customization of the resulting sound. This parametric approach offers greater creative control, flexibility, and the ability to shape the output by manipulating parameters such as duration, intensity, and spectral characteristics. The instrument facilitates experimentation and exploration in sound design, opening up new possibilities for creating fresh and innovative sounds. The integration of the JUCE framework ensures efficient and flexible realization of the *CrossSampler* VST instrument. Through the convergence of convolution and sampling techniques, *CrossSampler* contributes to the field of digital signal processing, expanding the creative horizons for musicians and sound designers. By enabling real-time control and the incorporation of user audio files, *CrossSampler* empowers users to create original sounds and explore new timbres. The instrument offers a novel platform for musicians and sound designers for sonic exploration and experimentation.

XIII. Declarations

Availability of data and materials - The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests - The authors declare that they have no competing interests.

Funding -Not applicable.

Authors' contributions - sole authorship JE**.**

Acknowledgements - Not applicable.

References

[1] H. Davies, 'A history of sampling', *Org. Sound*, vol. 1, no. 1, pp. 3–11, Apr. 1996.

[2] H. Dudley, 'Remaking Speech', *The Journal of the Acoustical Society of America*, vol. 11, no. 2, pp. 169–177, Jun. 2005.

[3] J. O'Smith, 'Cross-Synthesis'. https://ccrma.stanford.edu/~jos/sasp/Cross_Synthesis.html (accessed Jun. 09, 2023).

[4] J. Laroche and M. Dolson, *Phase-vocoder: about this phasiness business*. 1997.

[5] J. A. Moorer, 'The Use of Linear Prediction of Speech in Computer Music Applications', Audio Engineering Society, Feb. 1978.

[6] M. Dolson, 'Recent advances in musique concrète at CARL', in *Proceedings of the International Computer Music Conference, 1985*, in Proceedings of the International Computer Music Association. San Francisco: International Computer Music Association, 1985, pp. 55–60.

[7] C. R. Poli Stephen Travis Pope, Aldo Piccialli, Giovanni De, Ed., *Musical Signal Processing*. New York: Routledge, 2013.

[8] W. G. Gardner, 'Efficient Convolution without Input/Output Delay', *Journal of The Audio Engineering Society*, Mar. 1995.

[9] W. Pirkle, *Designing Audio Effect Plugins in C++: For AAX, AU, and VST3 with DSP Theory*, 2nd edition. New York, NY: Routledge, 2019.

[10] L. Rabiner *et al.*, 'Terminology in digital signal processing', *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 5, pp. 322–337, Dec. 1972.

[11] S. W. Smith, *The Scientist & Engineer's Guide to Digital Signal Processing*. San Diego, Calif: California Technical Pub, 1998.

[12] C. Pina, 'Convolution and convoluted acoustic spaces', Oct. 09, 2019. https://www.academia.edu/44002739/Convolution_and_convoluted_acoustic_spaces

[13] C. Roads, 'Musical Sound Transformation by Convolution', presented at the ICMC Proceedings 1993, Michigan, 1993.

[14] C. Roads, *Composing Electronic Music: A New Aesthetic*. Oxford, New York: Oxford University Press, 2015.

[15] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*, 3rd ed. edition. Cambridge: Cambridge University Press, 2014.

[16] D. G. Myers, *Digital Signal Processing: Efficient Convolution and Fourier Tranform Techniques*. New York: Prentice Hall, 1991.

[17] C. Donahue, T. Erbe, and M. Puckette, 'Extended Convolution Techniques for Cross-Synthesis', presented at the International Conference on Mathematics and Computing, 2016.

[18] B. Truax, 'Sound, Listening and Place: The aesthetic dilemma', *Org. Sound*, vol. 17, no. 3, pp. 193–201, Dec. 2012.

[19] E. Grinstein, N. Q. K. Duong, A. Ozerov, and P. Pérez, 'Audio Style Transfer', in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 586–590. doi: 10.1109/ICASSP.2018.8461711.

[20] S. O'Modhrain, 'A Framework for the Evaluation of Digital Musical Instruments', *Computer Music Journal*, vol. 35, no. 1, pp. 28–42, 2011, Accessed: Jun. 05, 2023. [Online]. Available: https://www.jstor.org/stable/41241705