Real-time FM tone transfer with Bela

Franco Caspe
Centre for Digital Music
Queen Mary University of London
f.s.caspe@qmul.ac.uk

Andrew McPherson
Centre for Digital Music
Queen Mary University of London
a.mcpherson@qmul.ac.uk

Mark Sandler
Centre for Digital Music
Queen Mary University of London
mark.sandler@qmul.ac.uk

***Abstract*—Tone transfer is a Neural Audio Synthesis approach that employs Deep Neural Networks (DNNs) to transform the timbre of a musical instrument into another. In this progress report, we analyze the deployment process of a lightweight tone transfer approach that resynthesizes audio employing a frequency modulation (FM) algorithm. The low computational cost, and the compactness of the synthesis controls of the FM generator allow us to employ a DNN architecture small enough to be deployed for real-time execution on Bela, a low-latency audio processing platform. We implement the tone transfer algorithm for a round-trip latency of 8ms, small enough to provide an intimate control with the synthesizer. Finally, we discuss potential uses for this novel method for continuous control of a well-known synthesis algorithm.**

FM Tone Transfer

Neural Audio Synthesis (NAS) algorithms are data-driven approaches that use Deep Neural Networks (DNNs) to generate realistic audio. In recent years, a new branch of controllable NAS algorithms have proposed control affordances by conditioning the DNNs with signals such as MIDI [1], raw audio [2], or features such as pitch and loudness [3]–[5], turning them in powerful synthesizers that can be played either through a standard music interface, or arbitrary audio sources.

A subset of these techniques employ differentiable synthesis blocks such as spectral modelling algorithms [3], [6] or waveshapers [4], [5] that are controlled by DNNs to produce audio. This combination biases the DNN towards audio generation, yielding lightweight models that can be executed in real-time with low latency [4], [7]. Moreover, such algorithms can be trained to resynthesize sound from a corpus of audio when conditioned with pitch and loudness features, making them playable by any audio source. This method is known as tone transfer [6]. Although the results are impressive, the complexity of the synthesis models make the process difficult to intervene by an user: it is usually required to retrain the network on a new dataset to access to new sounds.

In this progress report, we will discuss the deployment of a tone transfer algorithm that employs FM synthesis for audio generation, a well-known sound design algorithm that can generate complex spectra with low computational cost and a compact set of parameters, namely the frequency, modulation index and output levels of sinusoidal oscillators. We exploit the compactness of the timbre representation to deploy a DNN model of only 20k parameters. The small size of the neural network and efficiency of the synthesis method allow us to implement a full tone transfer pipeline (including loudness and pitch tracking) in C++ that can be executed in less than 4ms in Bela [8], an embedded computing platform for low-latency audio processing and generation.

The pipeline runs at 16kHz, with a frame size of 64 samples, yielding a total latency of 8ms including buffering, potentially enabling intimate control [9] of the NAS architecture for live use. Furthermore, the DNN describes timbre in terms of well-known FM sound design primitives that can be intervened at run time by an user to allow timbre exploration.

The end result is an FM synthesizer that is playable in real-time by an audio source. This novel approach of continuous control of the well-known FM synthesis architecture, historically driven by keyboard or triggers, enables the integration of classic synthesizer sound design strategies, such as oscillator envelope reshaping, re-routing, feedback and frequency manipulation in a synthesis pipeline controlled by arbitrary sound signals, including musical instruments. We will outline such uses and showcase a basic application, expecting to generate a conversation on potential uses of this synthesis strategy. As the computing power of embedded systems increases, and NAS algorithms get more efficient, we expect to see further continuous control strategies for synthesizers with DNNs presented by the NIME community.

The author is funded by EPSRC and UKRI under the Centre for Doctoral Training in Artificial Intelligence and Music at Queen Mary University of London (Grant EP/S022694/1).

References

[1] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel, "MIDI-DDSP: Detailed control of musical performance via hierarchical modeling," *ICLR*, 2022.
[2] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv:2111.05011*, 2021.
[3] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *ICLR*, 2020.
[4] B. Hayes, C. Saitis, and G. Fazekas, "Neural waveshaping synthesis," *ISMIR*, 2021.
[5] M. Michelashvili and L. Wolf, "Hierarchical timbre-painting and articulation generation," *ISMIR*, 2020.
[6] M. Carney, C. Li, E. Toh, P. Yu, and J. Engel, "Tone transfer: In-browser interactive neural audio synthesis," in *ACM IUI 2021 Workshops*, 2021.
[7] ACIDS - IRCAM, "Differentiable Digital Signal Processing," https://github.com/acids-ircam/ddsp_pytorch, 2022.
[8] A. McPherson and V. Zappi, "An environment for submillisecond-latency audio and sensor processing on beaglebone black," in *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.
[9] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer music journal*, vol. 26, no. 3, pp. 11–22, 2002.