DMRN+16: DIGITAL MUSIC RESEARCH NETWORK
ONE-DAY WORKSHOP 2021

QUEEN MARY UNIVERSITY OF LONDON
TUE 21 DECEMBER 2021

Building style-aware neural MIDI synthesizers using simplified differentiable DSP approach

Sergey Grechin[1] and Ryan Groves[2]

[1]Infinite Album, grechin.sergey@gmail.com
[2]Infinite Album, ryan@infinitealbum.io

***Abstract*— We explore how simplified differentiable DSP approach can be used to build realistically sounding virtual MIDI-controllable monophonic synthesizers. The simplification involves directly using MIDI data as input to the DDSP decoder. On top of that, we show how incorporating additional style-based and temporal channels can be used to imitate various aspects of performance and improve realism. We further demonstrate the results of applying the approach to the task of modelling the sound of electric guitar. The presented results were obtained with a model trained on less than 12 minutes of manually MIDI-annotated audio. The source code is released along with the prepared dataset.**

***Index Terms*—** Deep Learning, DDSP, virtual synthesizers, MIDI

I. MODEL ARCHITECTURE AND INPUTS

The model was built on top of [1] which in turn is a simplified version of original DDSP design [2]. In contrast to the original implementation, in our model audio is not directly used to produce inputs to the decoder. Instead, we use MIDI annotations to heurustically generate fundamental frequency curve (F0), loudness, additional timing signals (”distance from onset” , ”distance to offset”), proposed in [3], and arbitrary CC (continuous controller) values. CC values are used to capture various performance characteristics such as ”openness” - the degree of how muted the guitar string is when plucked. Additional inputs are passed through dedicated stacks of dense layers before being concatenated with the traditional DDSP decoder inputs.

II. DATASET AND RESULTS

For training, 12 minutes of playing chromatic scales on an electric guitar were recorded. MIDI annotations were made manually with CC55 used to represent open (127) and muted (1) sound. For training, velocity values were obtained using A-weighted loudness-based approach proposed in [3]. On inference stage velocity was taken from the source MIDI.

We encorage the reader to visit the online supplement page [4] to listen to the generated examples and access additional resources such as the source code and the dataset.

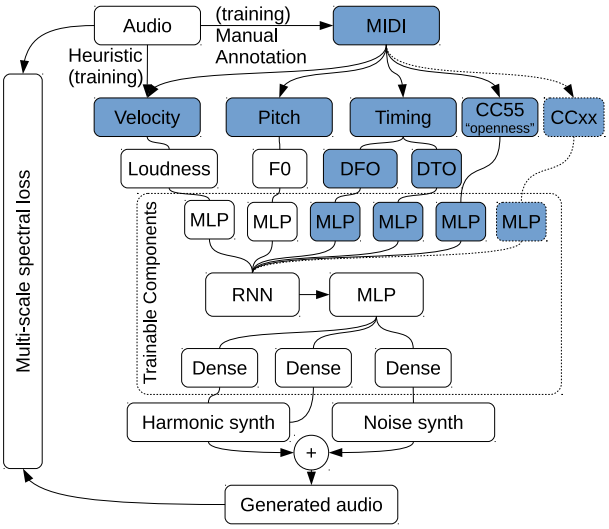

Figure 1: Componenets of the model. The white blocks represent the componements of original DDSP design [2]. MLP - Multilayer perceptron, DFO - distance from onset, DTO - distance to offset, CC - continuous controller

Figure 2: Waveforms generated on C-5 for two MIDI notes with closed and open sound. This example demonstrates that the model has succeeded in learning temporal characteristics of the guitar sound, including the modelling of initial transient.

III. FUTURE RESEARCH

We plan to research if other aspects of playing style can be captured using this approach, such as playing with chords. For that, we plan to add additional harmonic synth stacks with independently learnable parameters.

IV. REFERENCES

[1] DDSP simplified repository. [Online]. Available: https://github.com/raraz15/ddsp_simplified

[2] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts, “DDSP: Differentiable digital signal processing,” in *ICLR*, 2020.

[3] Nicolas Jonason, Bob Sturm, and Carl Thom, “The control-synthesis approach for making expressive and controllable neural music synthesizers,” in *Joint Conference on AI Music Creativity*, 2020.

[4] Online supplement. [Online]. Available: https://grechin.org/neural_synthesizers_with_simplified_ddsp.html