NEW IMPLEMENTATION METHOD FOR GENERALIZED FREQUENCY MODULATION SYNTHESIZER

Keiji Hirata
Future University Hakodate
hirata@fun.ac.jp

EXTENDED ABSTRACT

Generalized frequency modulation (GFM) synthesizer has the network architecture like neural networks that can be optimized by the well-known backpropagation technique (Figure 1) [2]. Many of commercial FM synthesizers [3, pp.224-250] provides the presets which are built-in connection patterns of oscillators with predefined parameters of amplitudes and carrier and modulating frequencies because some expert skills for manipulating parameters to generate new desired sounds are required. If we could imagine a single connection patterns of oscillators subsuming every preset, we would reach an idea of a general form of the FM synthesizer which looks like a neural network, the activation functions of which are vibrating functions.

Figure 1 presents how oscillator units are interconnected with weights w; the depth is M, the width is N, and Σ^w stands for weighted sum. The input to the network is vector $\{A_1, A_2, \cdots A_N\}$, and the output is waveform S. For an oscillator unit by a typical vibrating function, we first adopt a sinusoidal function $y = \sin\ 2\pi((x+c)\,t+p)$ with input x and output y (Figure 2, left). Each oscillator unit has two parameters c and p to be tuned corresponding to frequency and phase, respectively. All oscillator units share the identical timing signal t, which moves between starting time T_s and ending time T_e, to compute the output waveform (the red solid curve in Figure 3). The network attempts to fit the output waveform to the training waveform only between T_s and T_e (dark blue). Thus, in the ranges out of the interval between T_s and T_e, the network does not take care of the output waveform (red dashed curves).

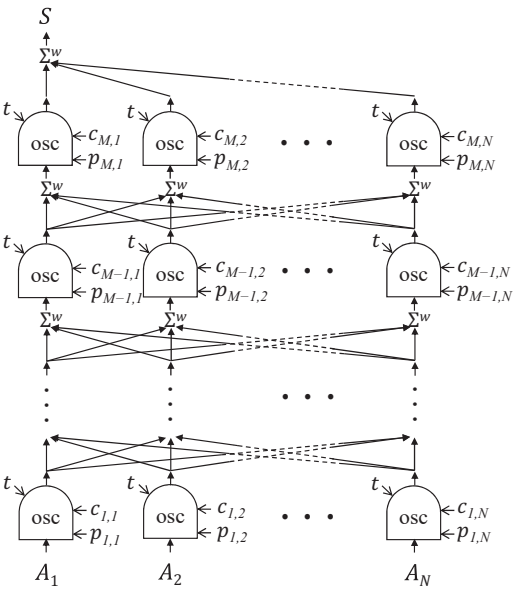

Figure 1. Network configuration

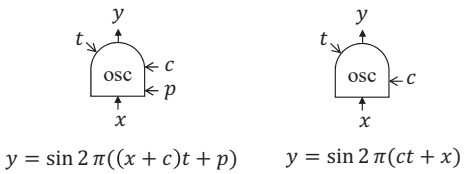

Figure 2. Oscillator units by frequency modulation (left) and phase modulation (right)

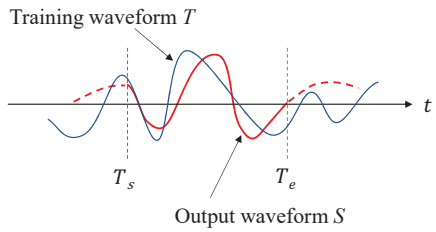

Figure 3. Output (red) and training (blue) waveforms

Theoretically, the input vector to the network can be either constants or any waveforms synchronized by the timing signal t. Here, without loss of generality, we assume all the elements of the input vector $\{A_1, A_2, \cdots A_N\}$ are constant values.

According to what signal processing teaches, phase modulation (PM) is essentially the same as frequency modulation (FM), and both the two modulations belong to angle modulation. Therefore, we could employ PM, instead of FM, and adopt a sinusoidal function (Figure 2, right), in which the number of parameters is one less.

The network attempts to approximate the output waveform to the training waveform by using the standard backpropagation technique to optimize the GFM synthesizer [1]. For optimization, we currently employ stochastic gradient descent, Adam, and L^2 regularization with soft threshold, although the technical details are omitted.

In Figure 4, we just demonstrate that the GFM synthesizer works well for small-sized artificial training waveforms, using two kinds of oscillator units, FM and PM, respectively. The training waveforms are two cycle of rectangle. The horizontal axis in the figure stands for time in the unit of time instant. The waves generated by the GFM synthesizer are drawn in the colored curves, and the training waves in black. In the left of Figure 4 (FM), we use the network of depth 5 by width 5 and obtain the output waveform at epoch 2700, while in the right (PM), depth 10 by width 10 at epoch 24276.

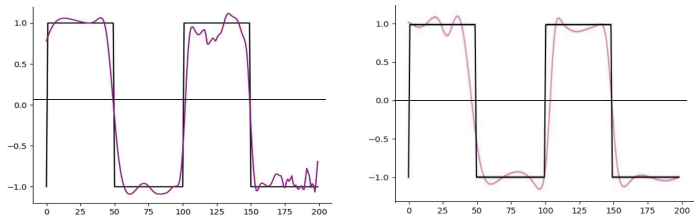

Figure 4. Output waveforms for rectangle given when training: (left) Oscillator unit by frequency modulation and (right) That by phase modulation

It is seen that the output waveform of the GFM synthesizer with oscillator units of FM rather contains some spiky, unstable elements, while that with PM is smoother. In particular, the latter half of the output waveform of FM is jaggier, and we have identified a theoretical basis for this phenomenon. It is also pointed out that the computational cost of the GFM synthesizer with FM is less.

Since the GFM synthesizer accepts an input vector, more than one pair of input vector and training waveform are given to the GFM synthesizer during training phase. Then, we have conducted a simple experiment and confirmed the GFM synthesizer may learn multiple waveforms at the same time, as in conventional neural networks, although the details of the experiment is omitted due to space limitation. We think that the tentative result of the experiment suggests a possibility to create new sounds; for example, some interpolated values among input vectors may yield the sound morphing among already existing sounds that have been trained.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[2] K. Hirata, M. Hamanaka, and S. Tojo. Feasibility study of deep frequency modulation synthesis. In *Proc. of Computer Music Multidisciplinary Resaerach (CMMR)*, page to appear, 2019.

[3] C. Roads. *The Computer Music Tutorial*. The MIT Press, 1996.