2024 edition

Deep Learning for Music Analysis and Generation

DDSP

({audio, MIDI} → audio)

Yi-Hsuan Yang Ph.D.
yhyangtw@ntu.edu.tw

Outline

- Differentiable digital signal processing (DDSP)
  - Uses a neural network to convert a user's input into complex DSP controls that can produce realistic signals
  - It’s a general idea
- MIDI-DDSP
  - MIDI-to-audio

Reference 1: ISMIR 2023 Tutorial

https://intro2ddsp.github.io/intro.html

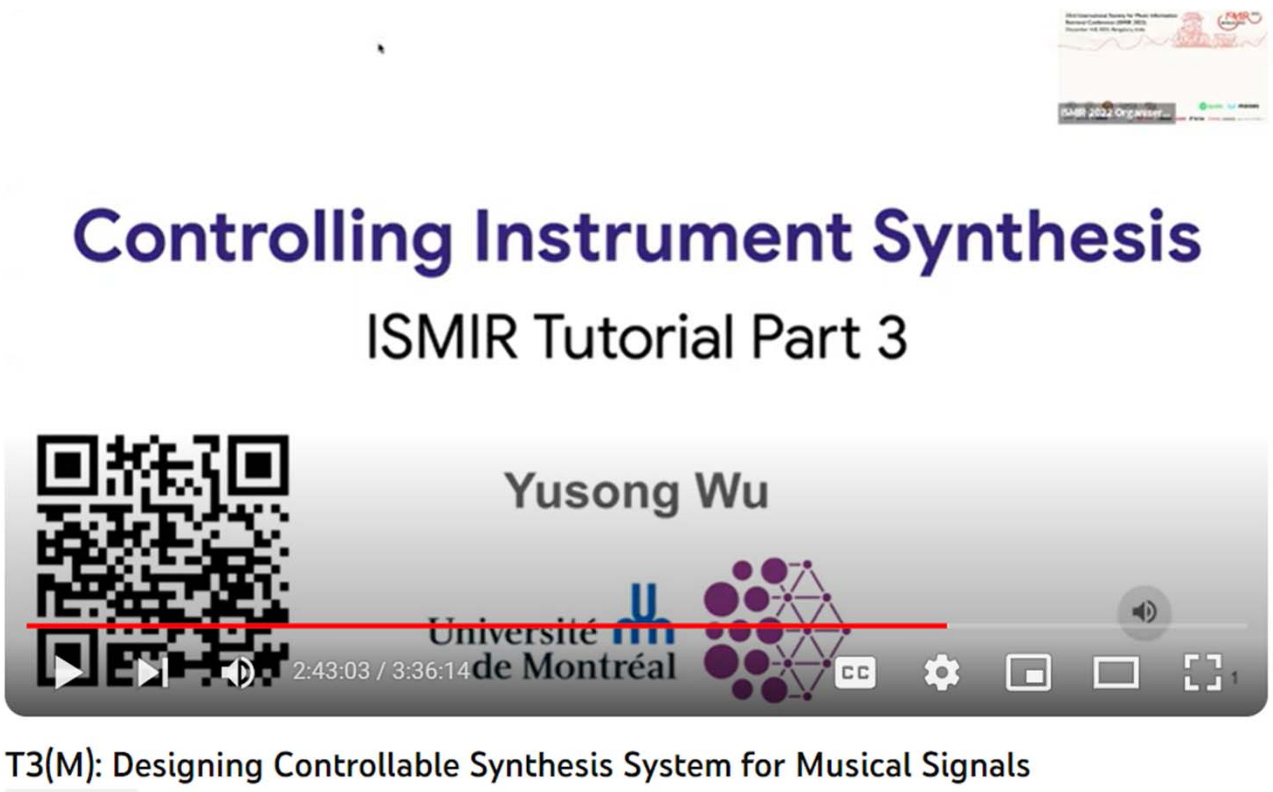
https://github.com/intro2ddsp/intro2ddsp.github.io

https://docs.google.com/presentation/d/1o9RWWmKX0yVVQii4-dtH3OlGZwqrfhEDgLo3582JnfM/edit#slide=id.p

Reference 2: ISMIR 2022 Tutorial

https://github.com/lukewys/ISMIR2022-tutorial

https://youtu.be/7U-zDL5con8?si=HcD7YDN66YPlyGCN&t=9783



Outline

- **Differentiable digital signal processing (DDSP)**
  - https://intro2ddsp.github.io/intro.html
- MIDI-DDSP

DSP & Audio Synthesis

https://intro2ddsp.github.io/background/neural-audio-synthesis.html

What Is DDSP?

https://intro2ddsp.github.io/background/what-is-ddsp.html

- "For example, a neural network might **output a value which is used as the cutoff frequency of a filter**, which is implemented differentiably"
- "During training, a loss function is computed on the output of the filter and, using the backpropogation algorithm, its gradient with respect to the neural network's parameters is computed."
- "In order to perform this computation, the derivative of the filter's output with respect to its cutoff frequency must be evaluated. That is to say, the filter forms a part of the computation graph, and its gradient is a factor of the chain rule decomposition of the loss gradient."

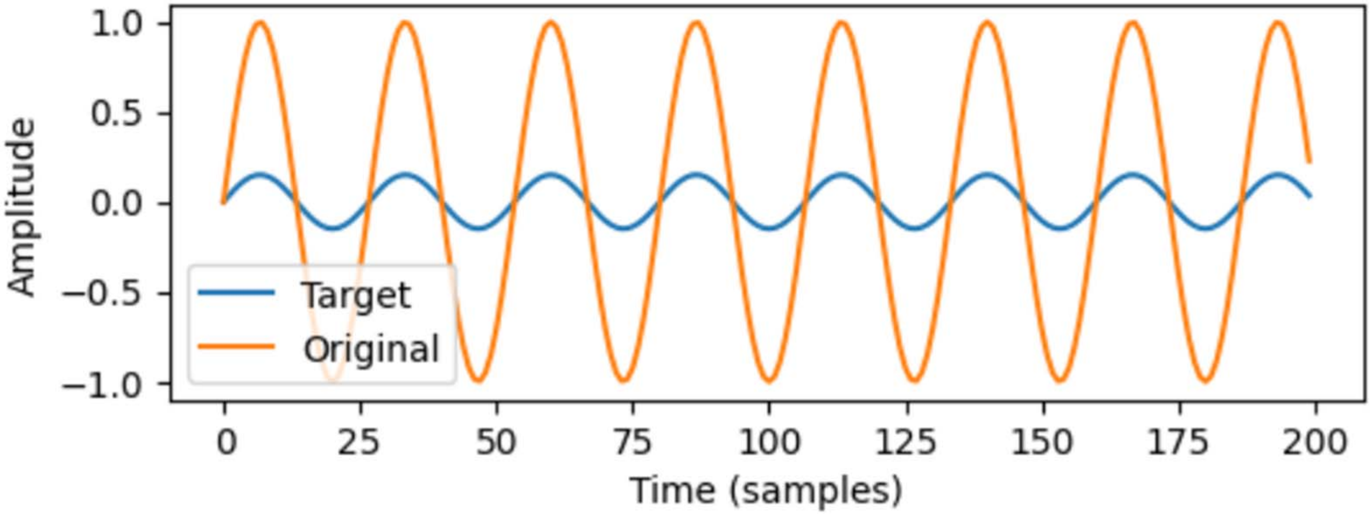
Why DDSP?

https://intro2ddsp.github.io/background/what-is-ddsp.html

1. We have prior knowledge about the class of signal we are interested in
2. We wish to infer the parameters of a particular signal processor or signal model
3. We are concerned about inference-time latency
4. We wish to allow human control over model outputs

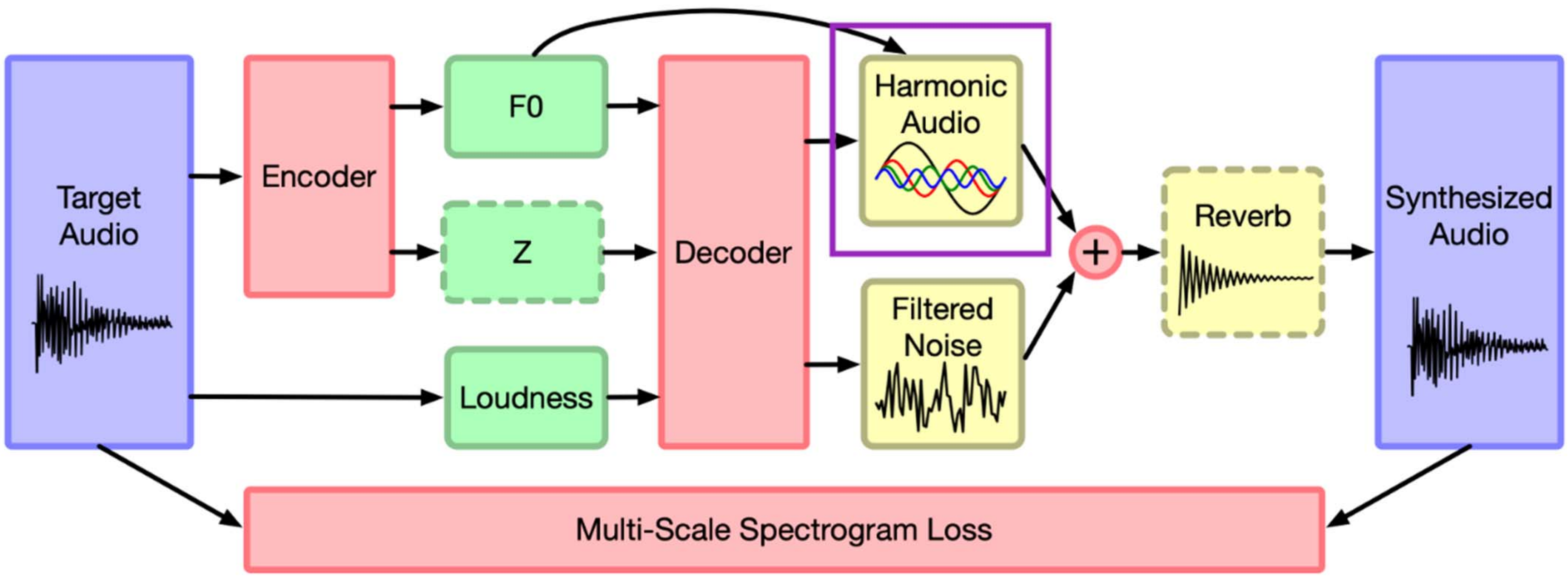
A Differentiable Gain Control

https://intro2ddsp.github.io/first-steps/diff_gain.html



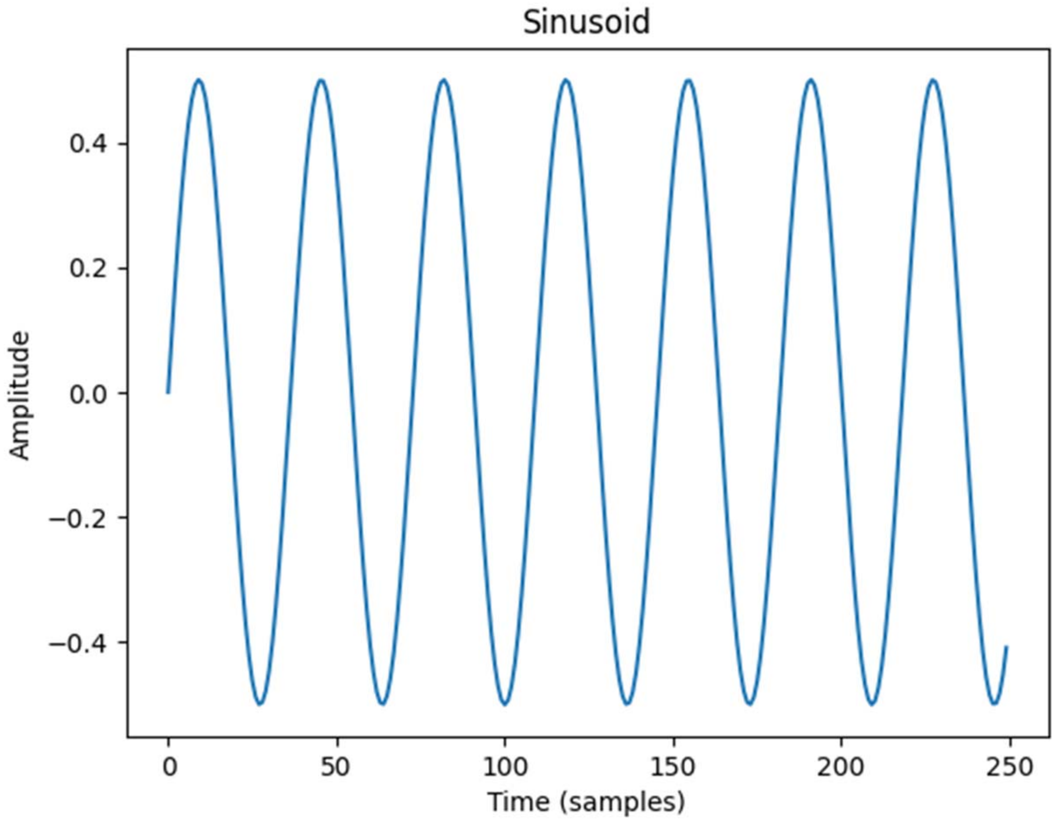
Sinusoidal Modelling Synthesis

https://intro2ddsp.github.io/synths/introduction.html



Writing a Differentiable Oscillator in PyTorch

https://intro2ddsp.github.io/synths/oscillator.html



Optimizing Parameters for the Differentiable Oscillator

https://intro2ddsp.github.io/synths/oscillator.html

- Optimizing amplitude → easy
- Optimizing frequency → difficult due to many local minima

Additive Synthesis

https://intro2ddsp.github.io/synths/additive.html

$$y[n] = \sum_{k}^{K} \alpha_k[n] \sin\left(\phi_k + \sum_{m=0}^{n} \omega_k[m]\right)$$

$$y[n] = \sum_{k=1}^{K} \alpha_k[n] \sin\left(\phi_k + k\sum_{m=0}^{n} \omega_0[m]\right)$$

$$\sum_{k=1}^{K} \hat{\alpha}_k[n] = 1 \text{ and } \hat{\alpha}_k[n] > 0$$

$$y[n] = A[n] \sum_{k=1}^{K} \hat{\alpha}_k[n] \sin\left(k\sum_{m=0}^{n} \omega_0[m]\right)$$

Harmonic Synthesizer

https://intro2ddsp.github.io/synths/harmonic_optimize.html

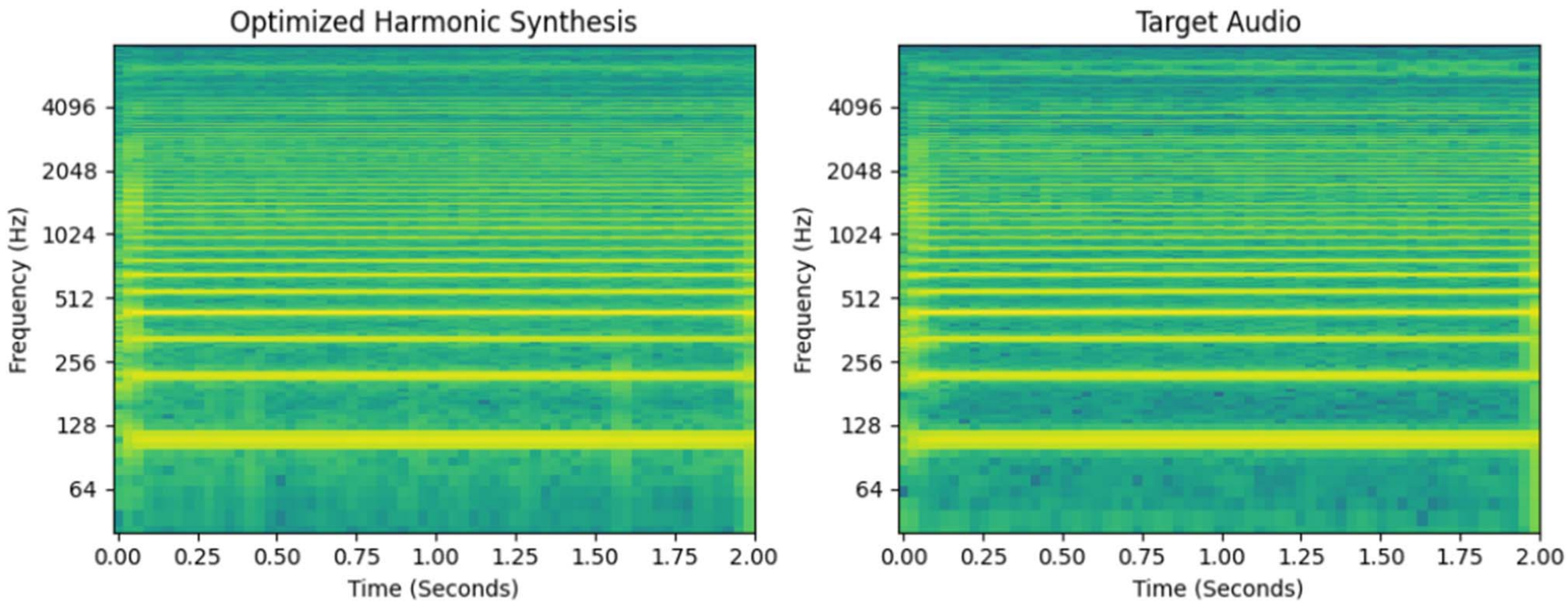
1. Constraining harmonic amplitudes to sum to one
2. Adding a global amplitude parameter
3. Parameter scaling to constrain the possible range of amplitudes
4. Removing frequencies above the Nyquist frequency which will result in aliasing

$$y[n] = A[n] \sum_{k=1}^{K} \hat{\alpha}_k[n] \sin\left(k \sum_{m=0}^{n} \omega_0[m]\right)$$

Optimizing a Harmonic Synthesizer

https://intro2ddsp.github.io/synths/harmonic_optimize.html

https://intro2ddsp.github.io/synths/harmonic_results.html



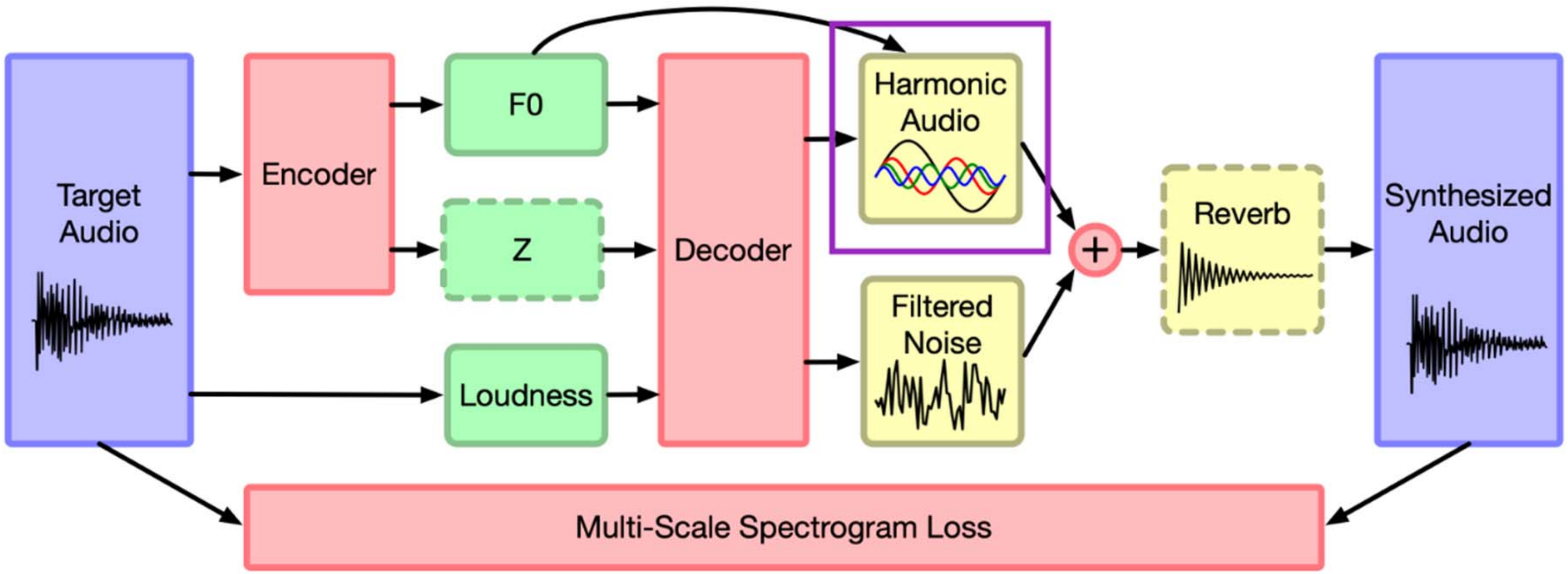
Differentiable Synthesis Libraries

https://intro2ddsp.github.io/synths/libraries.html

- https://github.com/magenta/ddsp
- https://github.com/acids-ircam/ddsp_pytorch
- https://github.com/torchsynth/torchsynth
- https://github.com/PapayaResearch/synthax
- https://github.com/csteinmetz1/dasp-pytorch

DDSP for Tone Transfer

- Essentially doing **audio-to-audio** generation
- Can we adapt the model to do **MIDI-to-audio** generation?



Outline

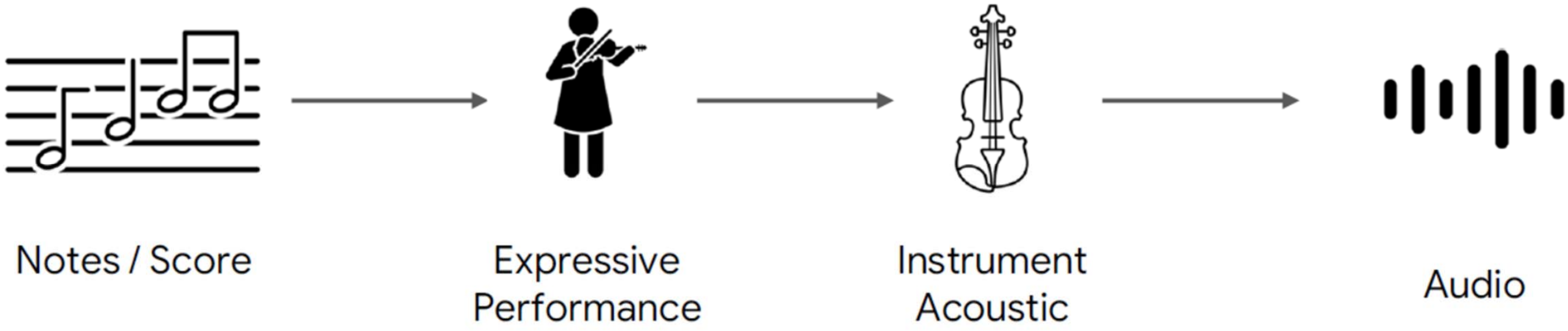
- Differentiable digital signal processing (DDSP)
  - Uses a neural network to convert a user's input into complex DSP controls that can produce realistic signals

- **MIDI-DDSP (ICLR'22)**
  https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

Ref: Wu et al, "MIDI-DDSP: Detailed control of musical performance via hierarchical modeling," ICLR 2022

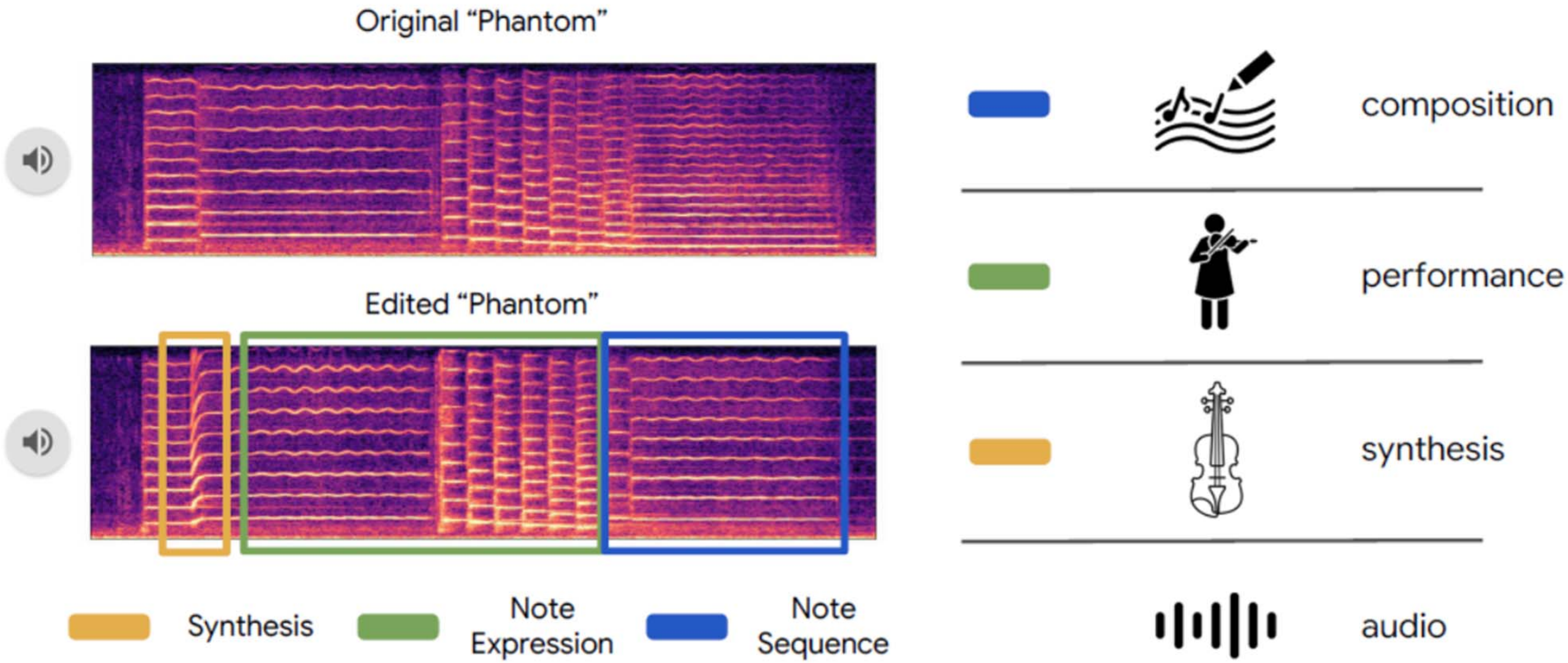
Human Instrument Performing Process

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925



MIDI-DDSP: Controlling Instrument Synthesis

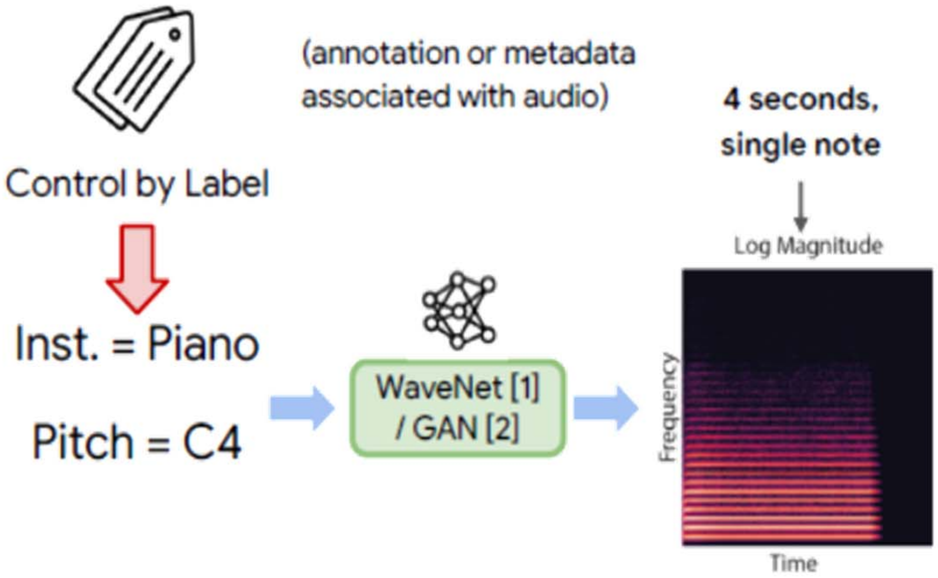
https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925



Note to Audio

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

- Generate single note condition on **pitch and instrument**.
- Modelling mainly **timbre**.

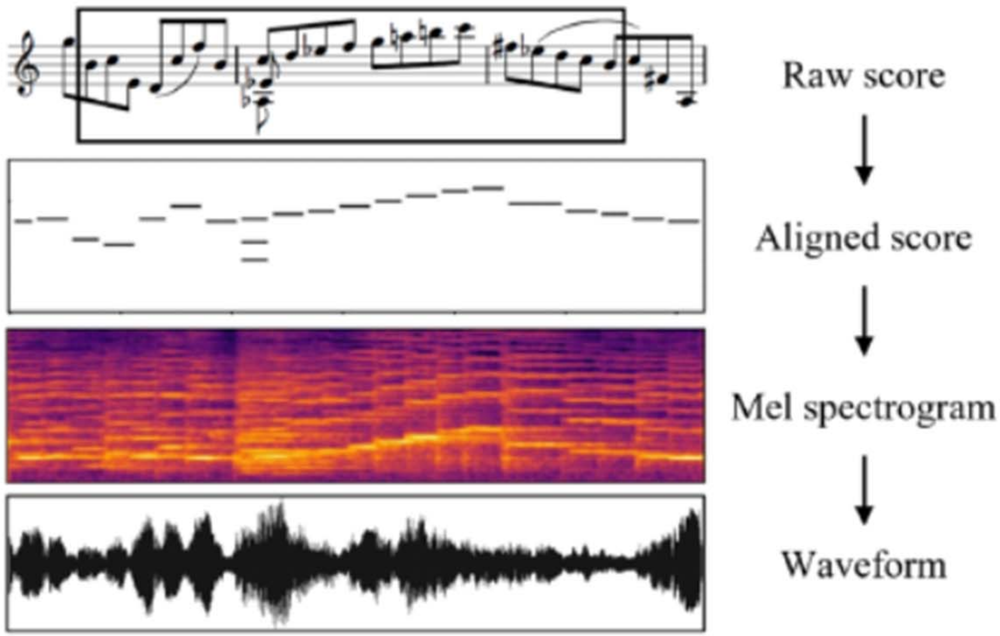


NSynth [1] & GANSynth [2]

Score to Audio

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

- Music score → Audio
- Generates **timbre** and **expressive performance** together.



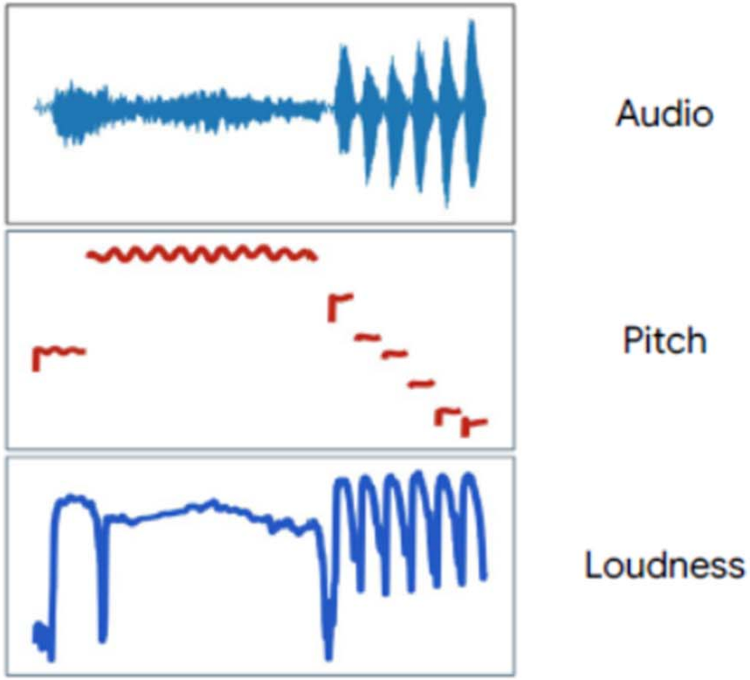
Deep Performer [3]

Other Aspects of Instrument Synthesis

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

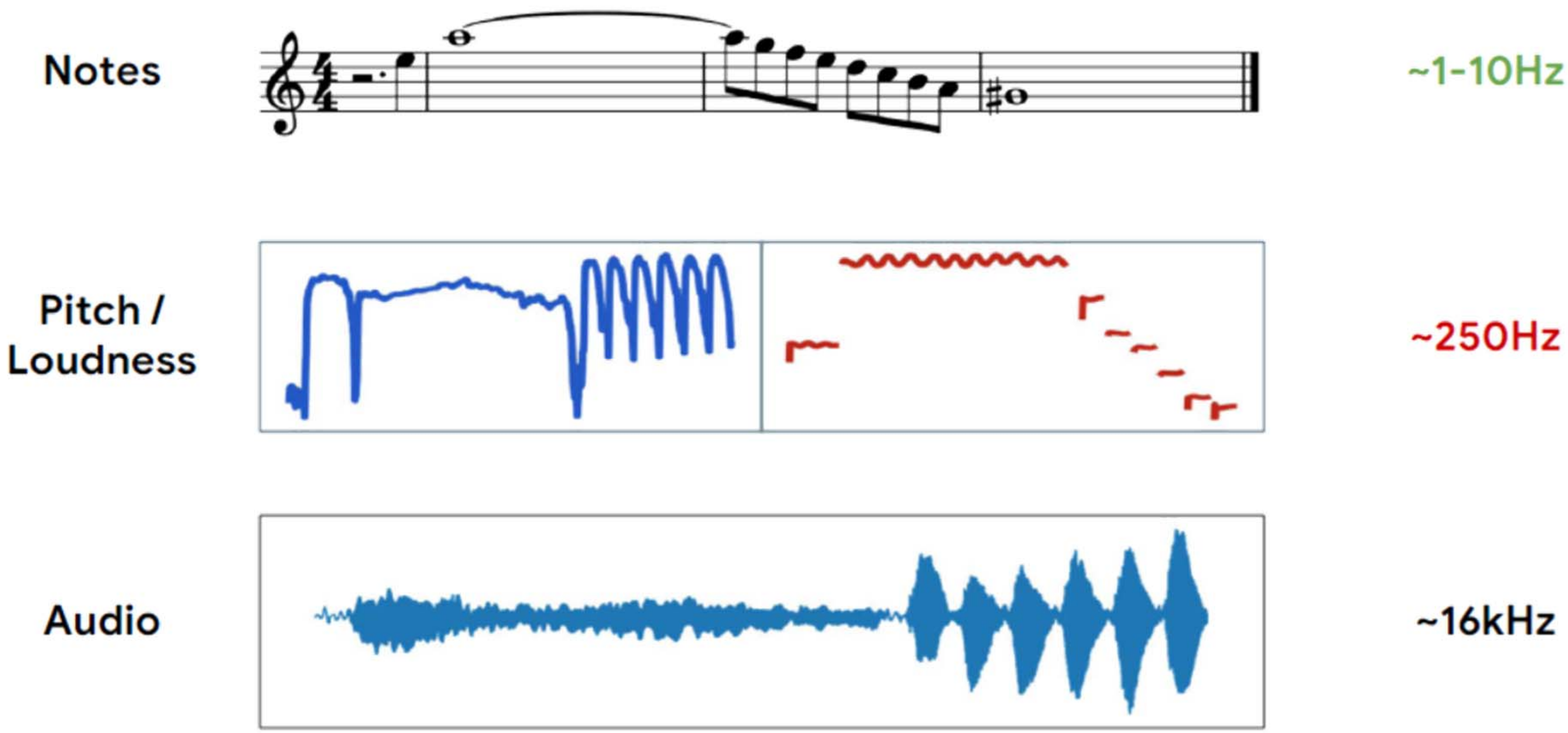
- **Low-level** quantities that changes **frequently**.
- E.g: pitch, loudness, expressive performance, etc.
- **No "labels"** available.

*labels: annotation or metadata associated with audio



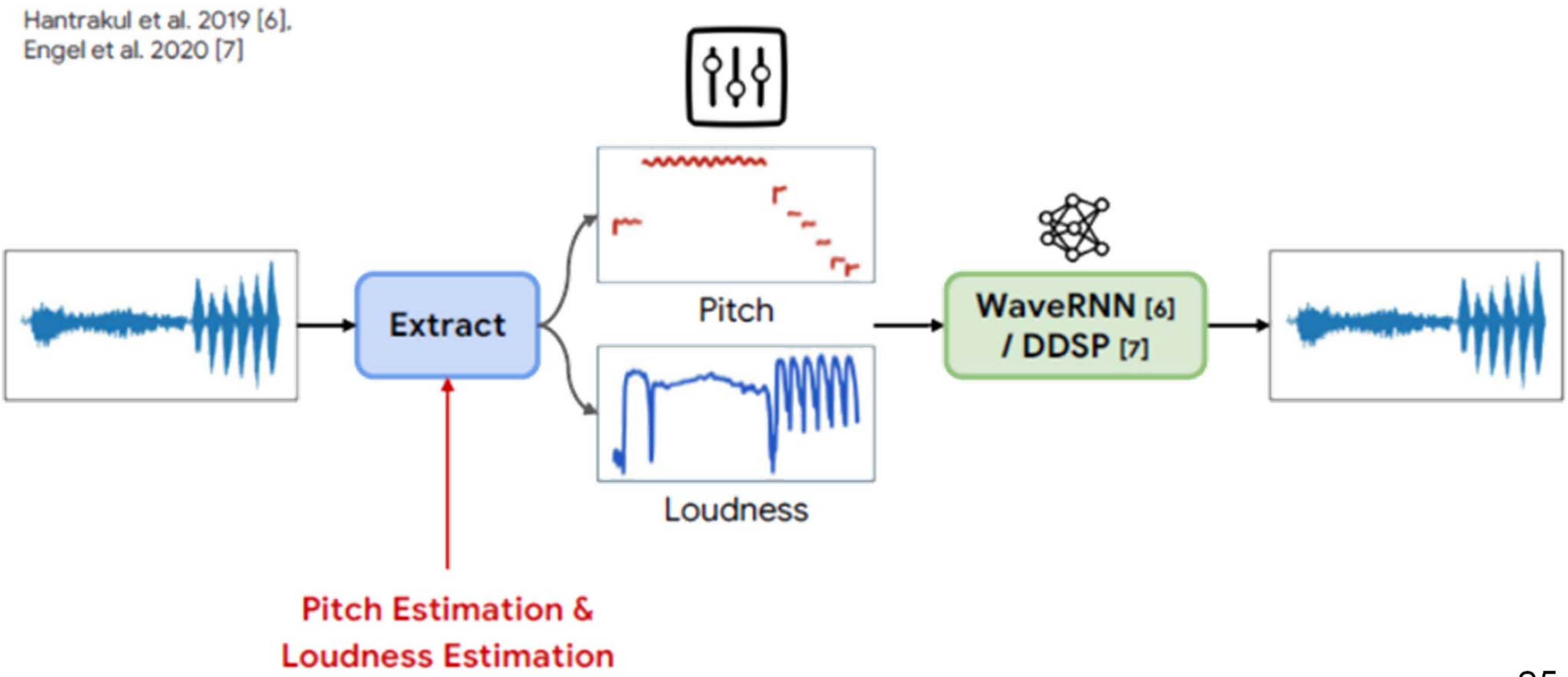
Low-level Quantities

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925



Extract the Label: Pitch and Loudness

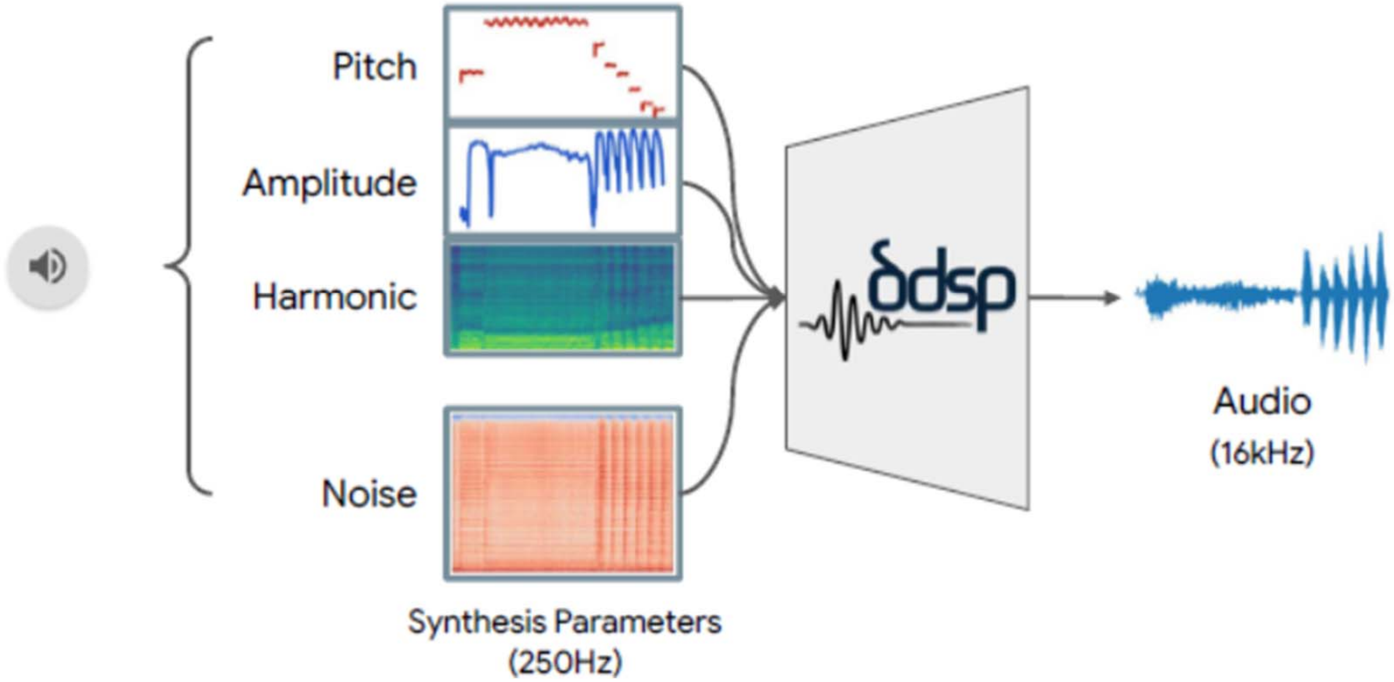
https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925



Learn to Extract Synthesis Parameters: DDSP

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

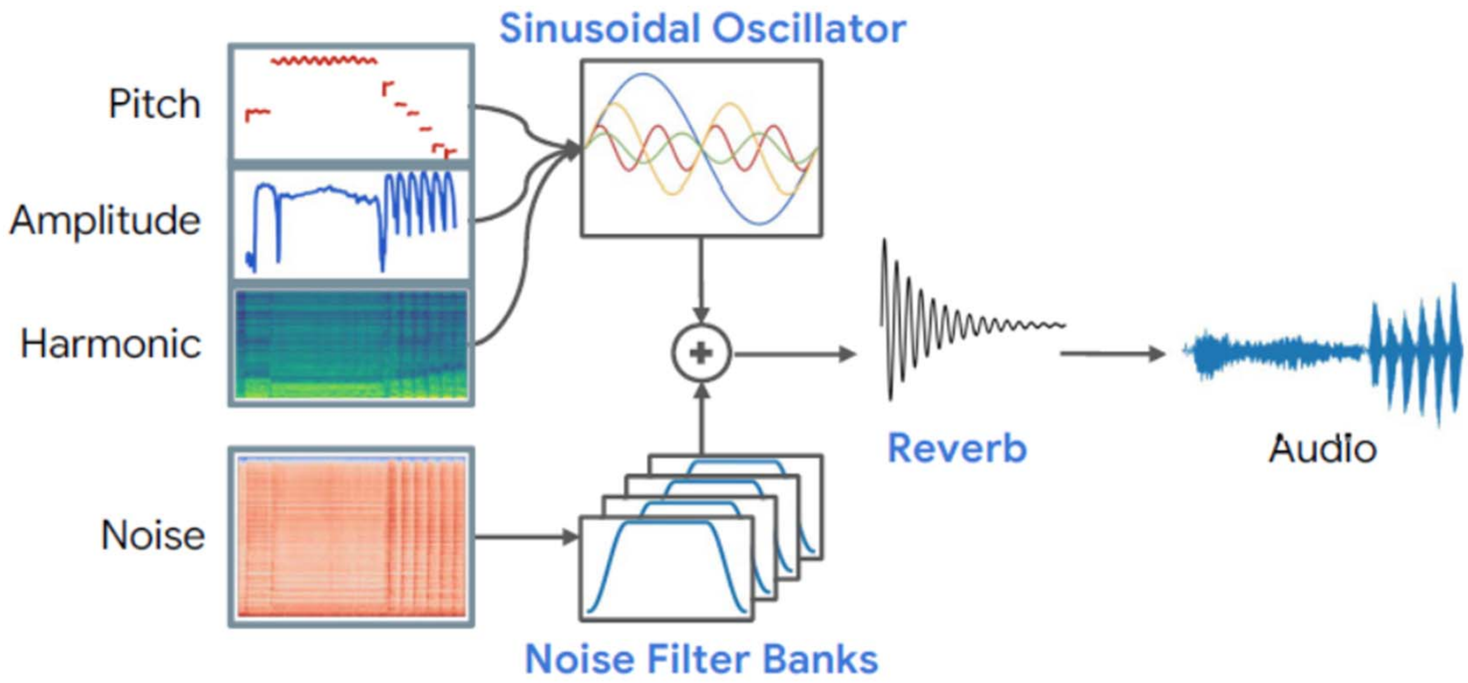
Engel et al. 2020 [7]



DDSP: Differentiable Digital Signal Processing

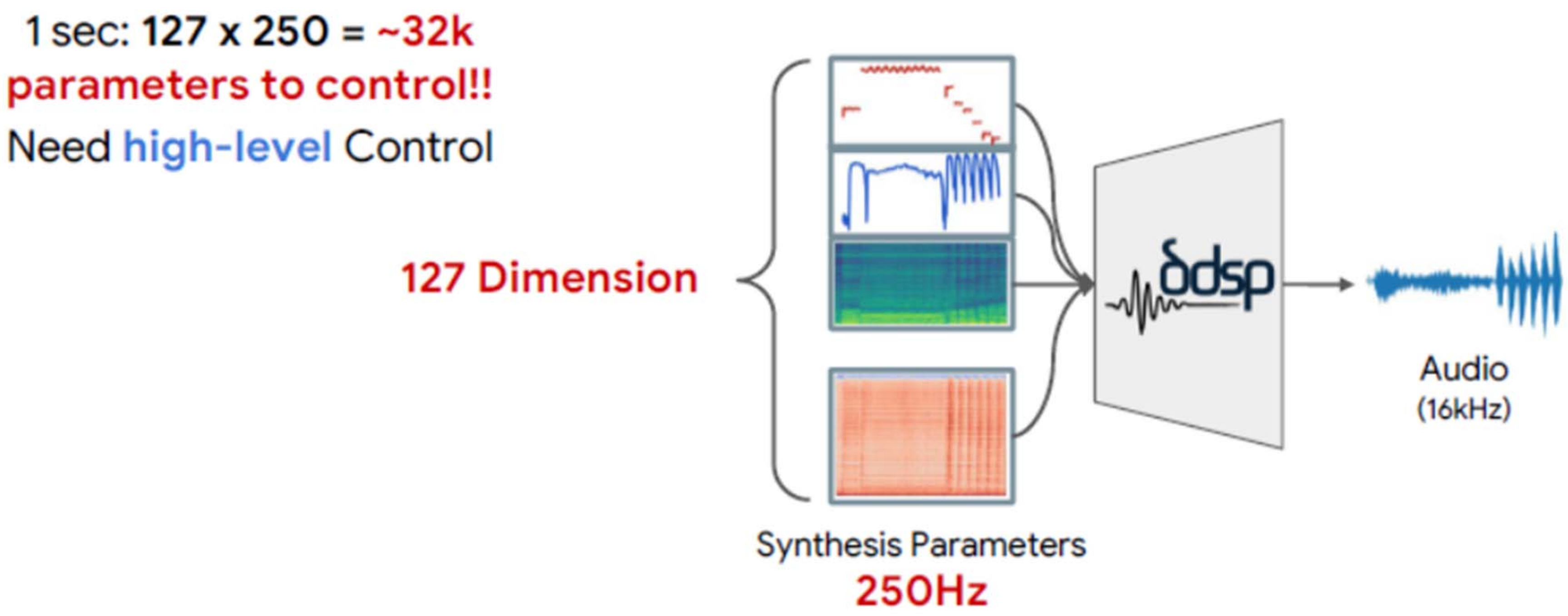
https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

Engel et al. 2020 [7]



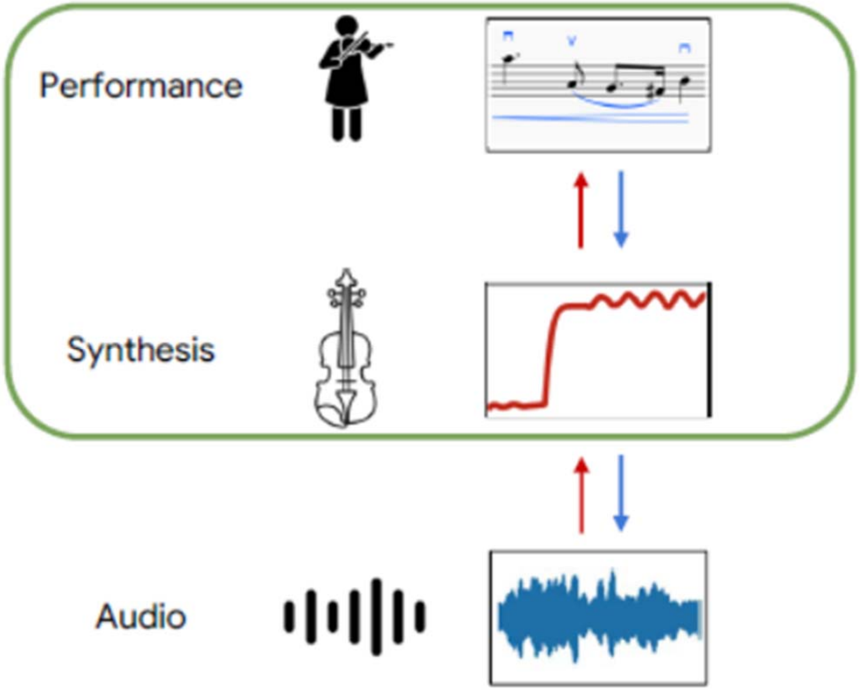
Problem of Low-level Control

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925



Extract Performance Parameter

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

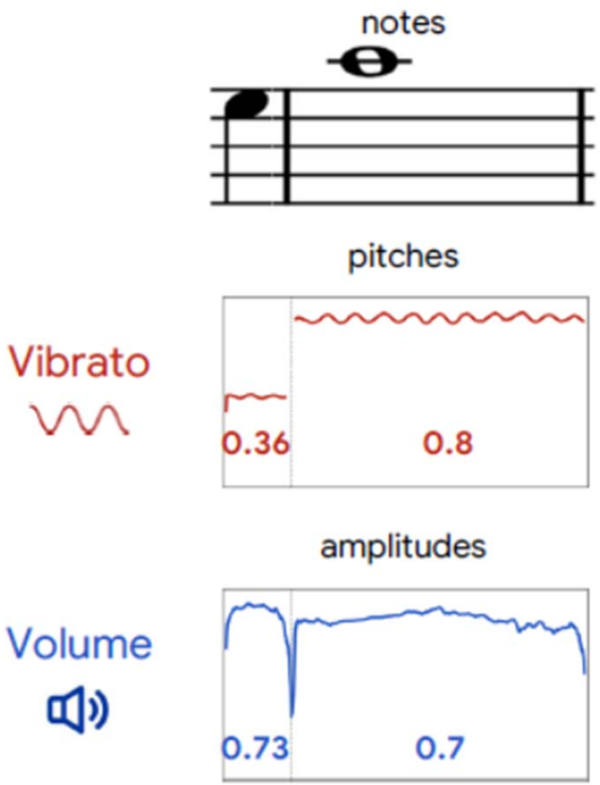


Summary statistics pooled over notes
6-D **scalar features**, scaled [0,1]:

- Volume
- Vibrato
- Brightness
- Attack Noise
- Volume Peak Position
- Volume Fluctuation

Extract Performance Parameter

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

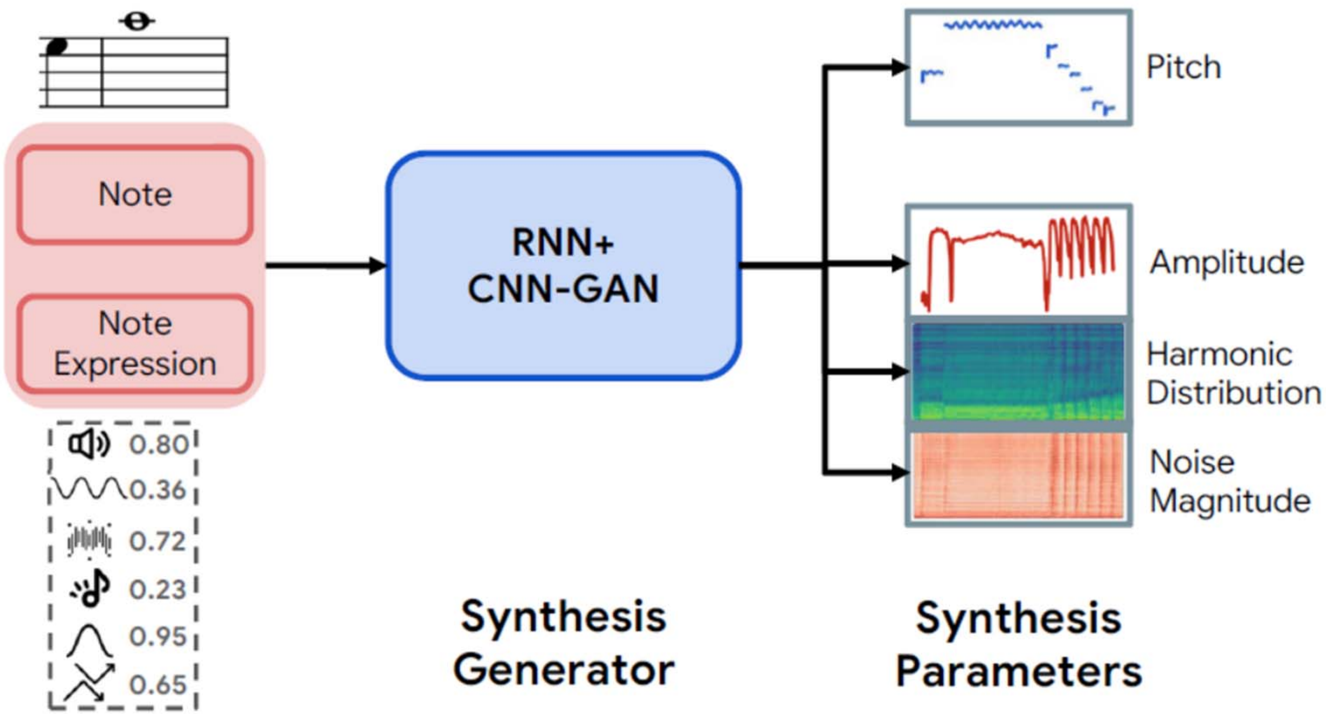


Summary statistics pooled over notes
6-D **scalar features**, scaled [0,1]:

- Volume
- Vibrato
- Brightness
- Attack Noise
- Volume Peak Position
- Volume Fluctuation

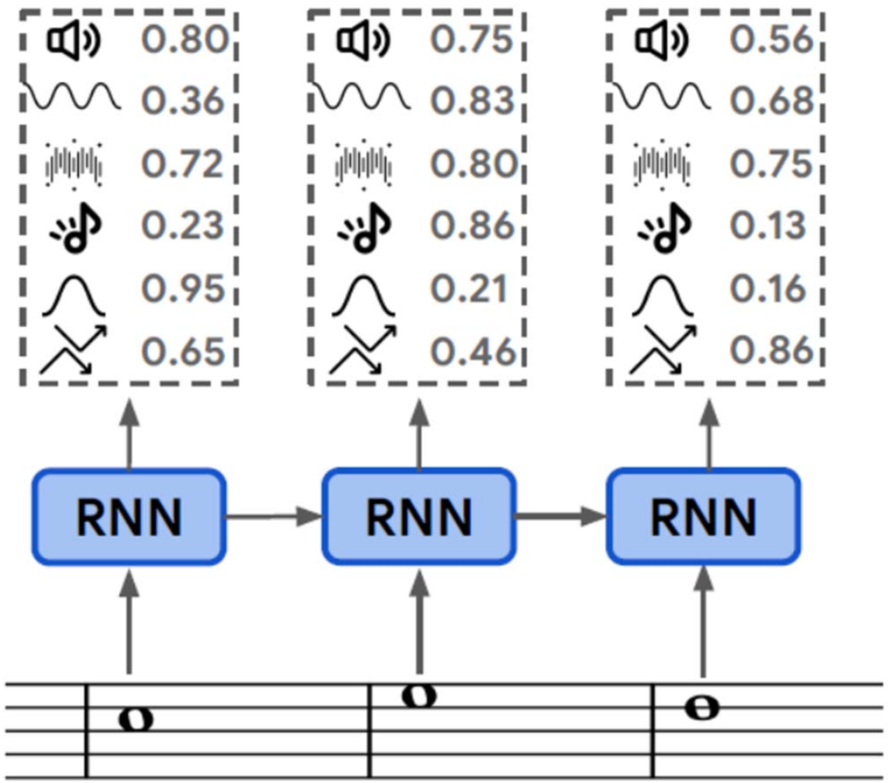
Synthesis Generator

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925



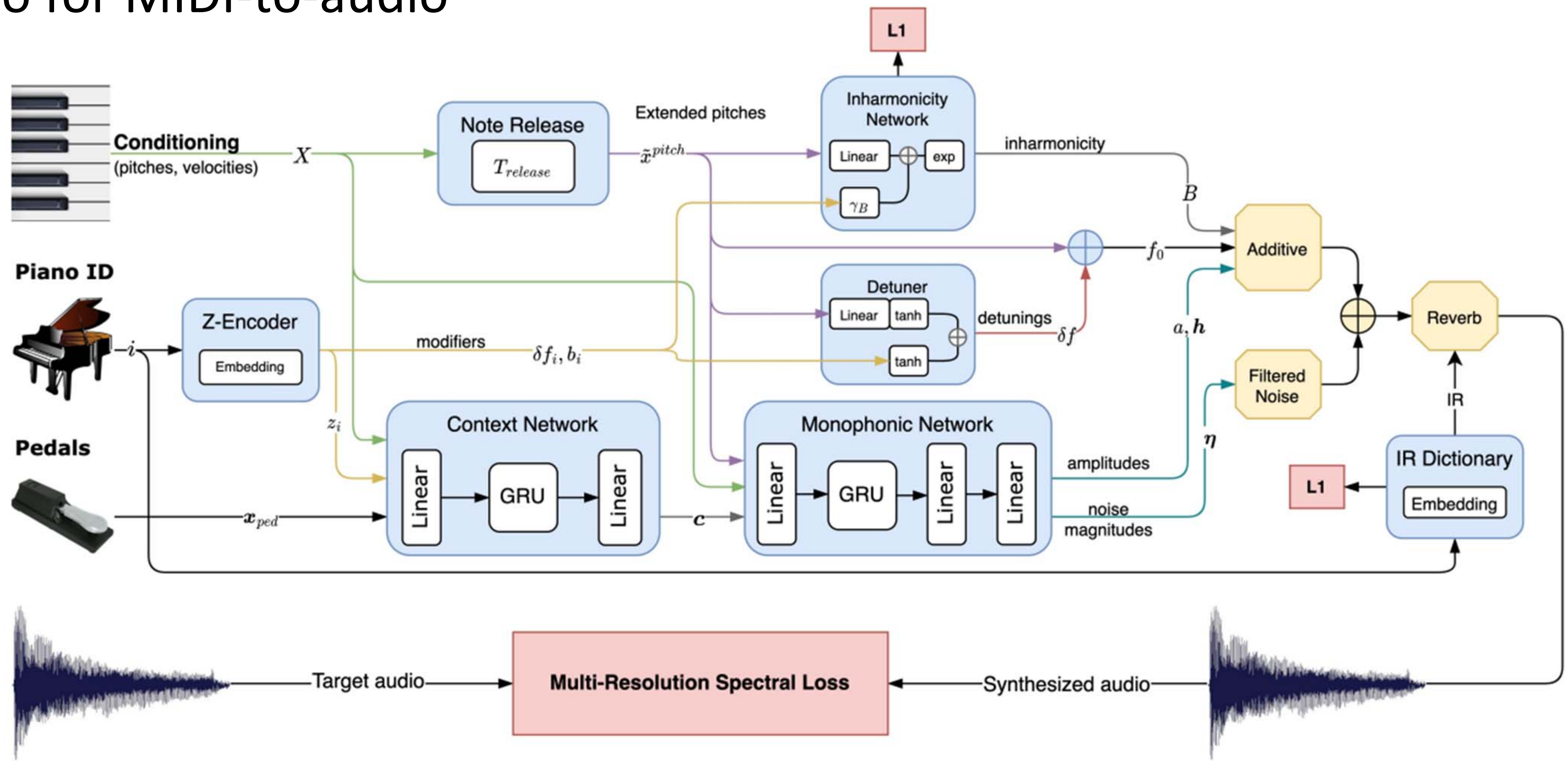
Autoregressive Prior on Expression Controls

https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925



DDSP-Piano (AES'23)

- Also for MIDI-to-audio

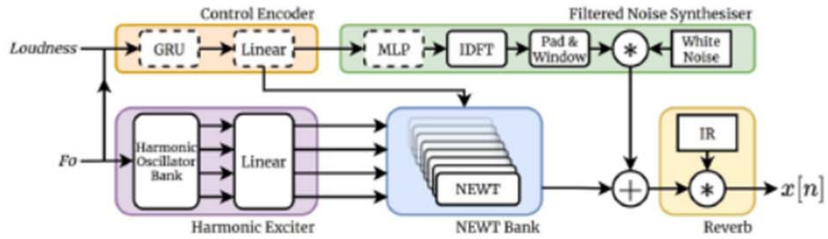


Ref: Renault et al, "DDSP-Piano: a neural sound synthesizer informed by instrument knowledge," AES 2023

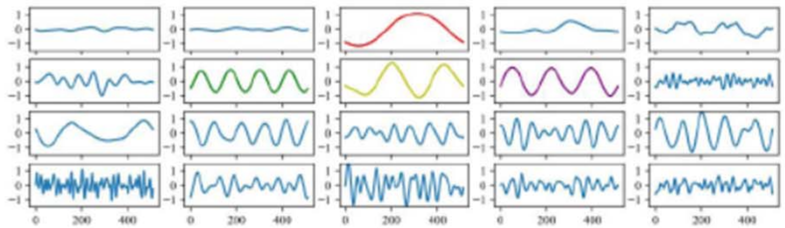
Other Differentiable Synthesis Works

- Also for MIDI-to-audio

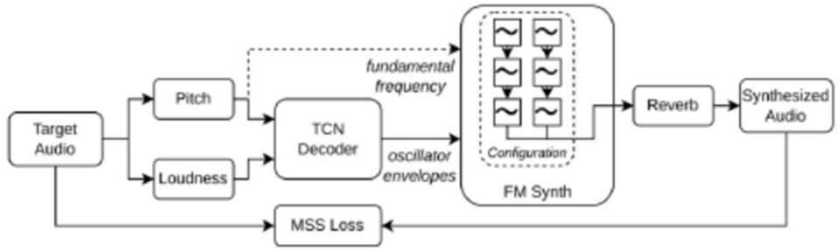
Waveshaping Synthesis [8]



Wavetable Synthesis [9]



FM Synthesis [10]

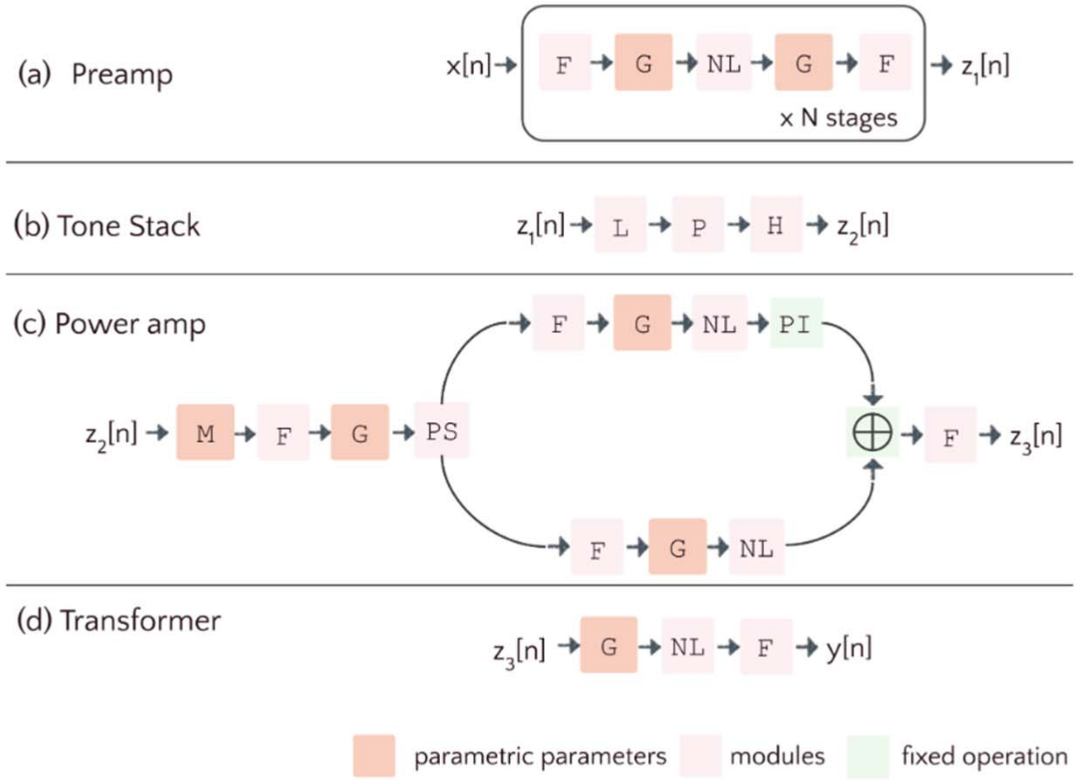


https://docs.google.com/presentation/d/1xrzeAIMnVOumSql_L2oIfVMXcJxOKd3F2u4_DEIkmbY/edit#slide=id.g1a484a50b88_1_1925

DDSP Guitar Amp (arXiv'24) (from our lab)

https://ytsrt66589.github.io/ddspGuitarAmp_Demo/

- Not for MIDI-to-audio
- Models the four components of a guitar amp using specific DSP-inspired designs
  - preamp
  - tone stack
  - power amp
  - output transformer



Ref: Yeh et al, "DDSP Guitar Amp: Interpretable guitar amplifier modeling," arXiv 2024