

# Opracowanie algorytmu generacji grafu DSP do rozwiązania problemu syntezy dźwięku

Design of a DSP graph generation algorithm for solving the sound  
synthesis problem

Autor pracy: Mateusz Bączek

Opiekun pracy: Dr Inż. Maciej Hojda

Seminarium Dyplomowe – prezentacja 1

2023

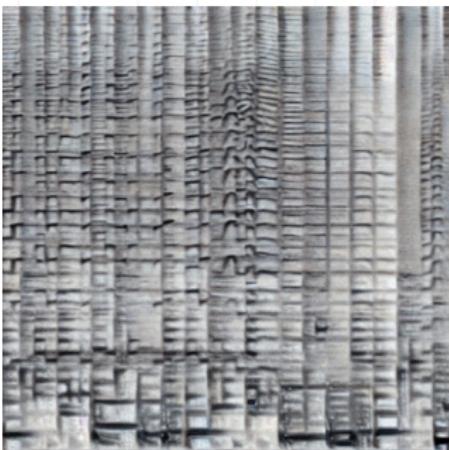


Politechnika Wrocławskiego

# Sztuka generatywna, algorytmiczna kompozycja muzyki



Rysunek: Obraz wygenerowany za pomocą algorytmu *Stable Diffusion*.



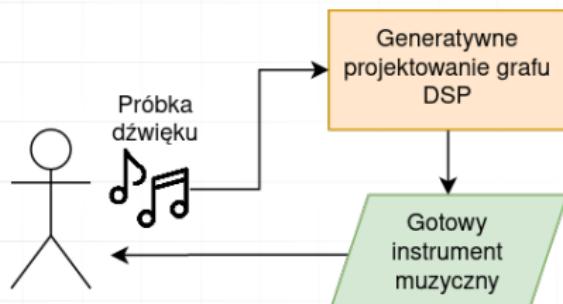
Rysunek: Spektrogram wygenerowany za pomocą algorytmu *Stable Riffusion*.

# Popularne podejścia do kompozycji algorytmicznej

1. Generowanie zapisu nutowego [Zhang, 2023],
2. Generowanie pełnego pliku audio [Forsgren, 2023],
3. Symulacja instrumentów za pomocą sieci neuronowych [Engel, 2017].

# Cel pracy

Wytwarzanie grafu przetwarzania sygnałów, który generuje dźwięk o określonej **barwie**.

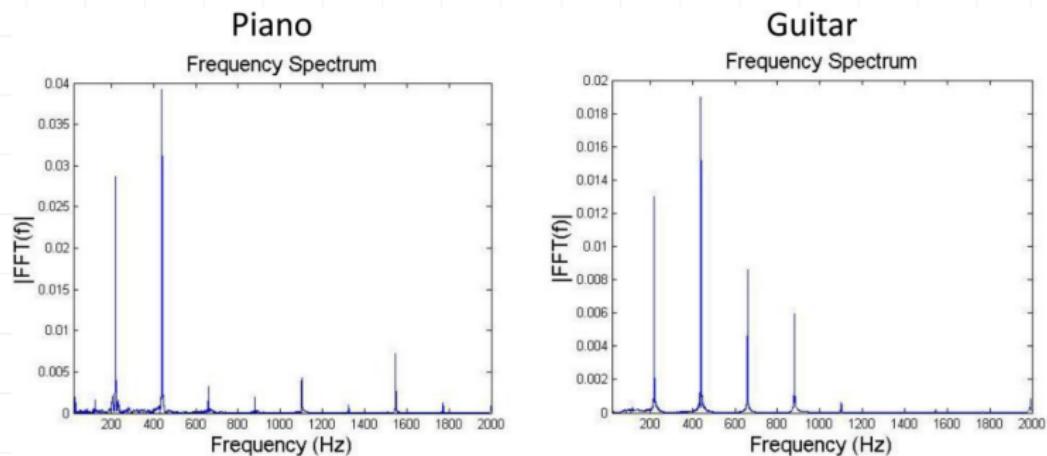


Rysunek: Ilustracja przypadku użycia.



Rysunek: Lutmistrz przy pracy.

# Czym jest barwa dźwięku?



Rysunek: Porównanie transformaty Fouriera dla tej samej nuty granej na pianinie i na gitarze.

# Problemy rozwiązywane w ramach pracy

1. Jak wygenerować dźwięk?
2. Jak ocenić, czy dźwięk spełnia zadane wymagania?
3. Jak zmodyfikować generowany dźwięk, aby zbliżyć się do wymaganego brzmienia?



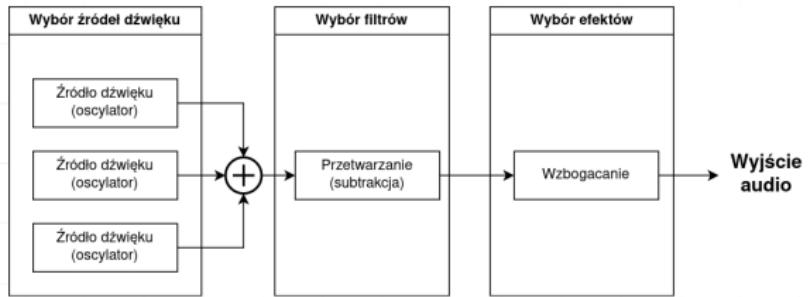
Rysunek: Przebieg sygnału dźwiękowego na oscyloskopie.

# Skąd bierze się barwa dźwięku?

- ▶ Materiał, z którego wykonane są struny,
- ▶ pudło rezonansowe (lub jego brak),
- ▶ kształt pudła rezonansowego,
- ▶ materiał, z którego wykonany jest instrument,
- ▶ materiał, który uderza w struny,
- ▶ ... i wiele innych cech fizycznych instrumentu.

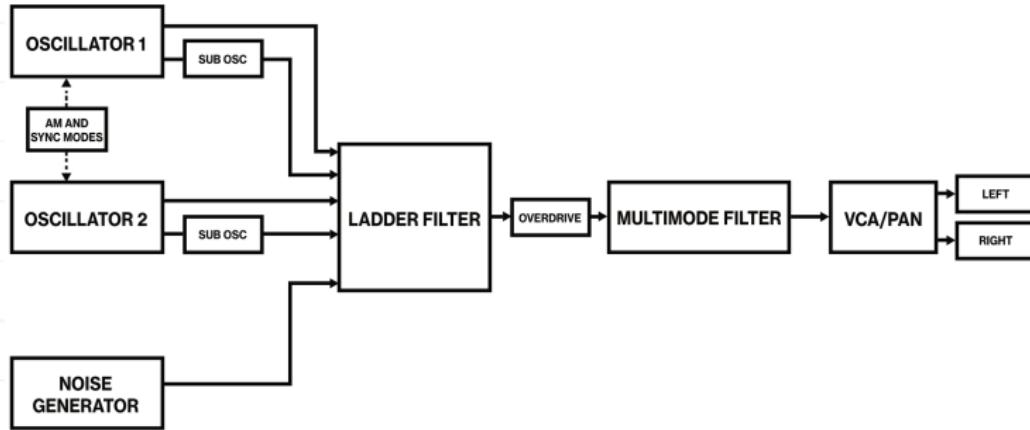
# Architektura instrumentu muzycznego

- ▶ Generacja sygnału,
- ▶ filtracja sygnału,
- ▶ efekty dźwiękowe.



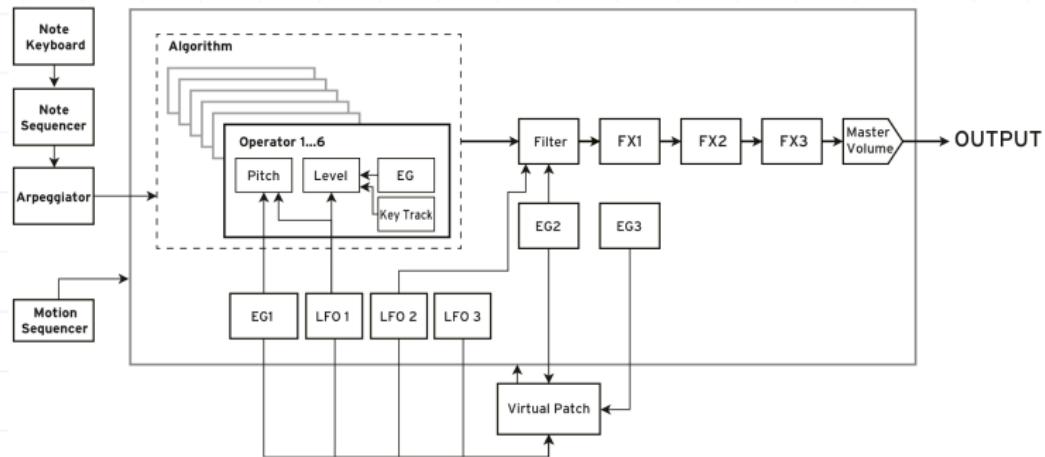
Rysunek: Architektura instrumentu muzycznego.

# Jak kontroluje się barwę w syntezatorach dźwięku



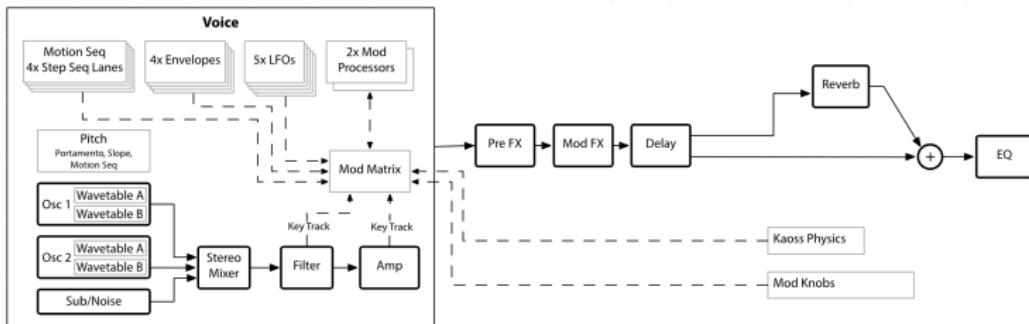
Rysunek: Diagram blokowy pojedynczego głosu w syntezatorze Elektron Analog Four.

# Jak kontroluje się barwę w syntezatorach dźwięku



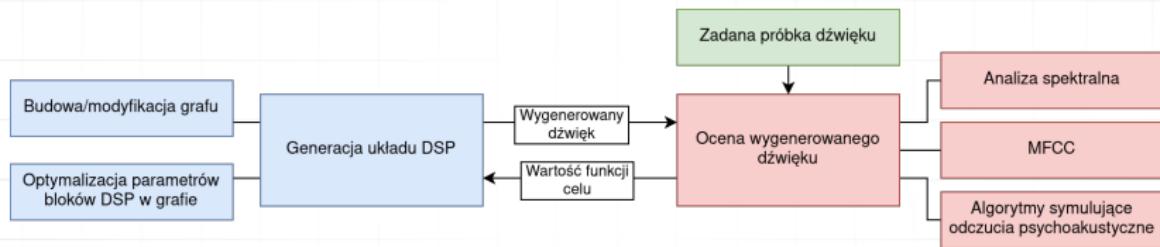
Rysunek: Diagram blokowy pojedynczego głosu w syntezatorze Korg Opsix.

# Jak kontroluje się barwę w syntezatorach dźwięku



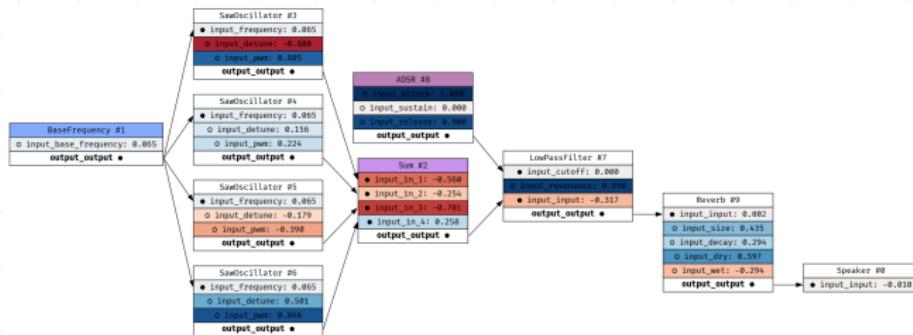
Rysunek: Diagram blokowy pojedynczego głosu w syntezatorze Korg Modwave.

# Jak algorytmicznie wytworzyć układ DSP generujący zadany dźwięk?



Rysunek: Diagram algorytmu realizowanego w ramach pracy.

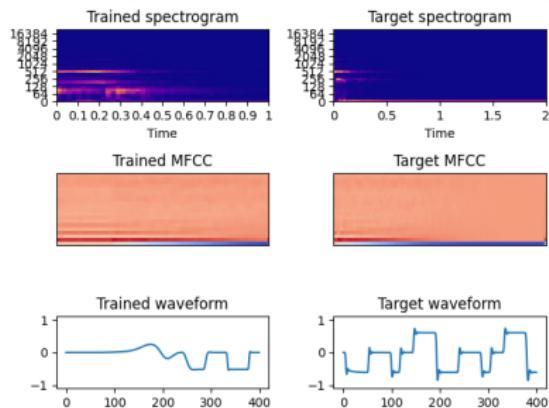
# Owoce pracy - silnik syntezy



Rysunek: Przykładowy graf DSP wygenerowany w środowisku eksperymentalnym.

- ▶ Implementacja w wydajnym, komplikowanym języku Rust,
- ▶ Natywna biblioteka dla języka Python,
- ▶ Generuje struktury danych z pakietu obliczeniowego numpy.

# Owoce pracy - porównanie sygnałów pod względem barwy, czyli funkcja celu



**Porównywane cechy sygnału:**

1. Spektrum Fouriera,
2. *Mel-frequency cepstrum*,
3. CQT [**Shoerkhuber**].

Rysunek: Porównanie sygnałów.

## Osiągnięte cele

1. Wytworzenie elastycznego silnika do tworzenia grafów DSP,
2. Ewolucja parametrów grafu wykonującego syntezę FM,
3. Ewolucja parametrów grafu wykonującego syntezę virtual analog,
4. Ewolucja struktury grafu wykonującego właściwy typ syntezy (zależnie od próbki wejściowej).

## Dalszy harmonogram realizacji pracy

- 1.** Cel został osiągnięty: opracowany algorytm generuje grafy DSP, które upodobniają się do zadanych próbek dźwięku:
  - ▶ Wciąż możliwe jest poprawienie dokładności,
  - ▶ Wciąż możliwe jest przyspieszenie czasu trenowania,
  - ▶ Wciąż można implementować nowe bloki DSP,
- 2.** Postęp w pisaniu pracy: ~15%.

# Potencjalne kierunki dalszego rozwoju

1. Wykorzystanie STFT [**Jacobsen**] do syntezy dźwięków perkusyjnych,
2. Przetestowanie kolejnych bloków DSP:
  - ▶ *Physical Modeling,*
  - ▶ *Multimode filters,*
  - ▶ *Reverb with pitch shifting,*
3. Implementacja funkcji celu na GPU,
4. Szkolenie sieci neuronowej na przestrzeni dźwiękowej danego syntezatora, na wzór [**Faronbi, 2023**],
5. Inkrementalne trenowanie na coraz dłuższych fragmentach docelowej próbki,



Zhang, Li and Callison-Burch, Chris

Language Models are Drummers: Drum Composition with  
Natural Language Pre-Training

<https://arxiv.org/abs/2301.01162>



Forsgren, Seth\* and Martiros, Hayk\*

Riffusion - Stable diffusion for real-time music generation

<https://riffusion.com/about>,



Jesse and Resnick, Cinjon and Roberts, Adam and Dieleman,  
Sander and Eck, Douglas and Simonyan, Karen and Norouzi,  
Mohammad

Neural Audio Synthesis of Musical Notes with WaveNet  
Autoencoders

<https://arxiv.org/abs/1704.01279>

-  Faronbi, Daniel and Roman, Iran and Bello, Juan Pablo,  
Exploring Approaches to Multi-Task Automatic Synthesizer  
Programming  
[10.1109/ICASSP49357.2023.10095540](https://doi.org/10.1109/ICASSP49357.2023.10095540)
-  Yan Ke and Hoiem, D. and Sukthankar, R.  
Computer vision for music identification  
[10.1109/CVPR.2005.105](https://doi.org/10.1109/CVPR.2005.105)
-  M. Pasquier, P.  
Automatic Design of Sound Synthesizers as Pure Data Patches  
using Coevolutionary Mixed-typed Cartesian Genetic  
Programming  
<https://metacreation.net/wp-content/uploads/2015/08/p309-macret.pdf>

-  Caspe, Franco and McPherson, Andrew and Sandler, Mark  
DDX7: Differentiable FM Synthesis of Musical Instrument Sounds  
<https://arxiv.org/abs/2208.06169>
-  Jacobsen, E. and Lyons, R.  
The Sliding Fourier  
10.1109/MSP.2003.1184347
-  Christian, and Anssi Klapuri.  
Constant-Q transform toolbox for music processing  
7th Sound and Music Computing Conference,  
Barcelona, Spain. 2010



Rysunek: Obraz wygenerowany przez DALL-E dla promptu „sound synthesizer drawn in style of van Gogh”.

Dziękuję za uwagę!

# Deklaracja

Deklaruję, iż uda się ukończyć pracę przed  
wyznaczonym terminem obrony.

~ Mateusz Bączek