



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Biblioteka do obsługi filtrów cyfrowych

z przedmiotu

Języki Programowania Obiektowego

EiT Rok III

Michał Janik

Grupa: Piątek 9:45

prowadzący: Jakub Zimnol

09.01.2025

1. Opis Projektu

Projekt implementuje podstawowe filtry sygnałów — FIR (Finite Impulse Response) i IIR (Infinite Impulse Response), które są powszechnie stosowane w dziedzinie przetwarzania sygnałów. Celem projektu jest zaprezentowanie podstawowych operacji takich jak filtracja sygnałów, analiza stabilności filtrów IIR, oraz zarządzanie współczynnikami filtrów FIR i IIR.

Ta biblioteka może być wykorzystywana w różnych zastosowaniach, takich jak:

- Filtrowanie sygnałów: FIR i IIR są szeroko stosowane w filtrach sygnałów, takich jak filtry dolnoprzepustowe, górnoprzepustowe, pasmowe i zaporowe.
- Przetwarzanie audio: Usuwanie zakłóceń w sygnałach audio, poprawa jakości dźwięku.
- Systemy komunikacyjne: Filtrowanie sygnałów w systemach transmisji danych w celu eliminacji zakłóceń.

2. Opis Zaimplementowanych Klas

- **Klasa filter:** Klasa bazowa, która dostarcza podstawowe funkcje do przetwarzania sygnałów. Zawiera metody:
 - **conv():** wykonuje splot dwóch wektorów.
 - **sum_abs2():** oblicza sumę kwadratów wartości absolutnych w zadanym zakresie wektora. Wykorzystywana do sprawdzania stabilności filtru.
 - **next_power_of_2():** znajduje najbliższą większą potęgę liczby 2 dla danego argumentu. Wykorzystywana jako ustawienie domyślnej wartości dla długości sygnału wyjściowego w zależności od wejściowego.
- **Klasa FIR:** Klasa implementująca filtr FIR. Umożliwia filtrację sygnału wejściowego oraz ustawianie współczynników odpowiedzi impulsowej:
 - **OutSignal():** oblicza sygnał wyjściowy filtracji za pomocą splotu sygnału wejściowego z odpowiedzią impulsową.
 - **setX(), setH():** umożliwiają ustawienie nowych sygnałów wejściowych i współczynników.
 - **getX(), getH():** zwracają obecny sygnał wejściowy i współczynniki odpowiedzi impulsowej.
- **Klasa IIR:** Klasa implementująca filtr IIR. Zawiera współczynniki do przetwarzania sygnału w formie sprzężenia zwrotnego i przekazu prostego (feedforward i feedback).
 - **OutSignal():** oblicza sygnał wyjściowy przy użyciu współczynników feedforward i feedback.
 - **stability():** sprawdza stabilność filtru na podstawie analizy energii sygnału wyjściowego pobudzonego skokiem jednostkowym dla aktualnie ustawionych współczynników.
 - **SetB(), SetA(), SetL():** umożliwiają ustawienie współczynników feedforward, feedback oraz długości sygnału wyjściowego.
 - **getB(), getA(), getL():** zwracają aktualne współczynniki feedforward, feedback i długość sygnału wyjściowego.

3. Kompilowanie projektu za pomocą CMake

- **Struktura katalogów:**

src/: zawiera pliki źródłowe (main.cpp, FIR.hpp, IIR.hpp, filter.hpp).
CMakeLists.txt to plik konfiguracyjny CMake.

- **Kroki kompilacji:**

Należy otworzyć terminal w katalogu głównym projektu, następnie utworzyć katalog budowania i przejść do niego np.:

mkdir build

cd build

Skonfigurować projekt za pomocą CMake:

cmake ..

Skorzystać z make, aby skompilować projekt:

make

- **Uruchomienie aplikacji:**

Po zakończeniu kompilacji uruchom aplikację poleceniem **./Filters**