



**AKADEMIA GÓRNICZO-HUTNICZA**

**AGH**

Dokumentacja do projektu

**Biblioteka szablonowa do obsługi  
filtrów cyfrowych oraz ich kaskad**

z przedmiotu

**Języki Programowania Obiektowego**  
Elektronika i Telekomunikacja III rok

*Adam Franaszek*

Środa 8:00

prowadzący: mgr. Jakub Zimnol

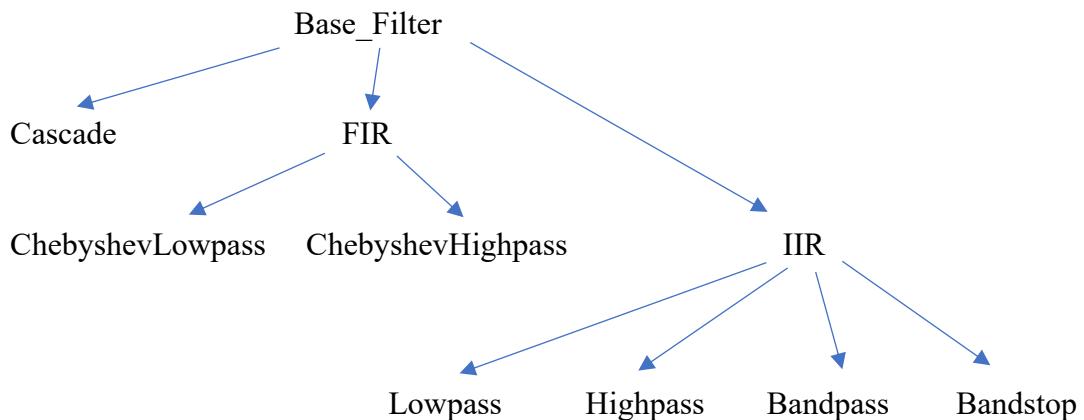
13.01.2026

# 1. Wprowadzenie

Celem projektu było zaimplementowanie w strukturze obiektowej biblioteki szablonowej do obsługi podstawowych filtrów cyfrowych oraz tworzenia ich kaskad. Głównym zastosowaniem jest szybkie prototypowanie systemów DSP w środowisku C++.

## 2. Struktura projektu

Strukturę dziedziczenia klas projektu można przedstawić graficznie w następujący sposób:



Opis poszczególnych klas:

Warstwa 0:

- **Base\_Filter<T>** - Wirtualna klasa bazowa, stanowi podstawę i dziedziczą po niej wszystkie inne klasy

Warstwa 1:

- **FIR<T>** - Klasa dziedziczy po bazowej, implementuje mechanizmy filtracji i obsługi parametrów filtrów typu FIR. Możliwe jest utworzenie obiektu tej klasy.
- **IIR<T>** - Klasa dziedziczy po bazowej, implementuje mechanizmy filtracji i obsługi parametrów filtrów typu IIR. Możliwe jest utworzenie obiektu tej klasy.
- **Cascade<T>** - Klasa dziedziczy po bazowej, przechowuje wektor unikalnych wskaźników na kopie filtrów. Umożliwia mechanizmy jednoczesnej filtracji i zarządzania pamięcią wszystkich filtrów składających się na kaskadę.

Warstwa 2:

- **Lowpass<T>**, **Highpass<T>**, **Bandpass<T>**, **Bandstop<T>** - klasy dziedziczące po FIR. Implementują automatyczne wyliczanie współczynników filtrów na podstawie prostych algorytmów DSP.
- **ChebyshevLowpass<T>**, **ChebyshevHighpass<T>** - klasy dziedziczące po IIR. Implementują uproszczony algorytm wyliczania współczynników filtra typu Chebyshev I biquad.

## 3. Skrócony opis mechaniki działania

Głównym aspektem wyróżniającym ten projekt jest klasa **Cascade**, która umożliwia tworzenie kaskad filtrów wszystkich obiektów, których klasy dziedziczą po klasie bazowej **Base\_Filter**. Jej działanie polega na tworzeniu idealnych kopii filtrów dodawanych do kaskady (by zabezpieczyć się przed usunięciem filtru źródłowego), a następnie przechowywania ich unique pointerów. Wywołanie kaskady jest również uproszczone dzięki przeciążeniu operatora “+”.

## 4. Opis uruchomienia

Projekt oparty jest na CMake oraz Makefile, zatem aby skompliować go wystarczy wpisać w terminal komendę “make”, a zostanie utworzony folder /build, z którego możemy uruchamiać poszczególne pliki pokazowe. Dodatkowo Komenda “make run” uruchamia od razu main, który jest krótkim wprowadzeniem do używania biblioteki.

## 5. Przykłady działania filtrów

Poniżej znajdują się przykłady przefiltrowanych sygnałów prostokątnych przez filtry (pliki demo generują wektory tych przebiegów) zobrazowane za pomocą środowiska Matlab:

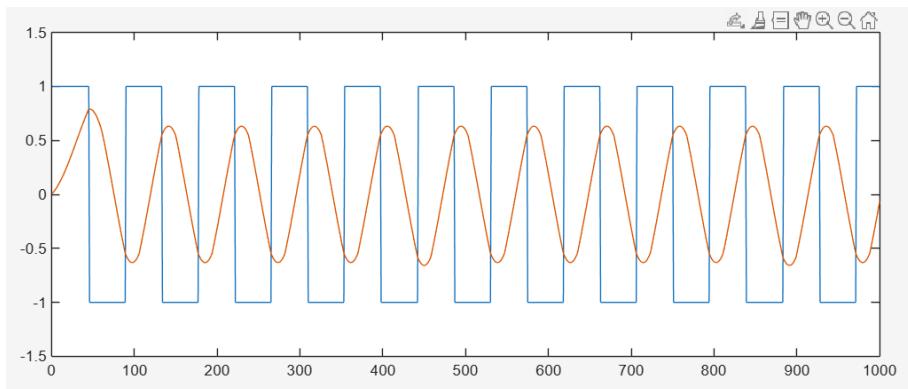


Figure 1- Lowpass, order = 60

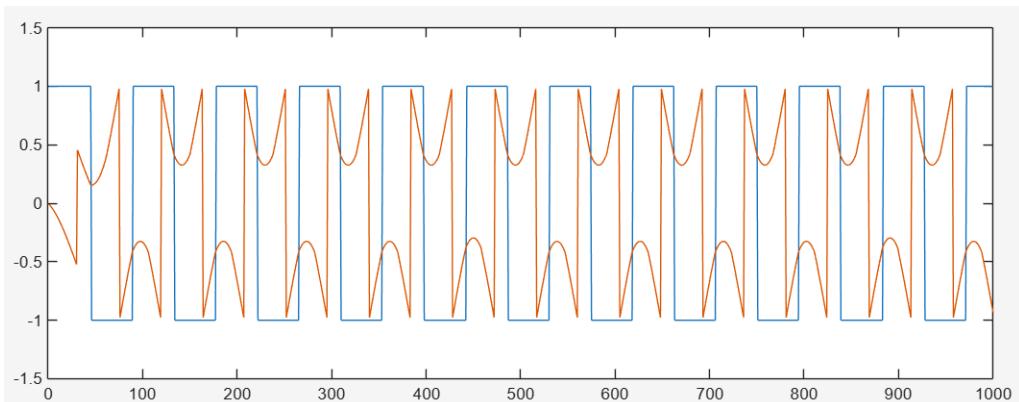


Figure 2 - Highpass, order = 60

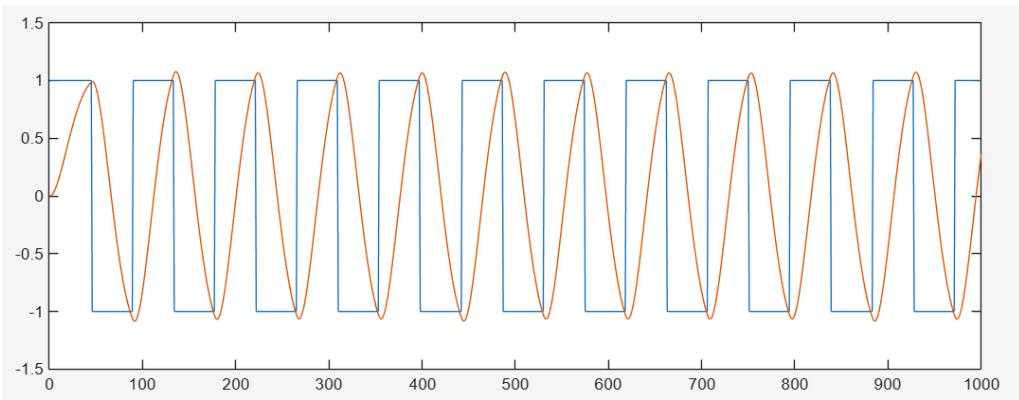


Figure 3- ChebyshevLowpass, order = 2

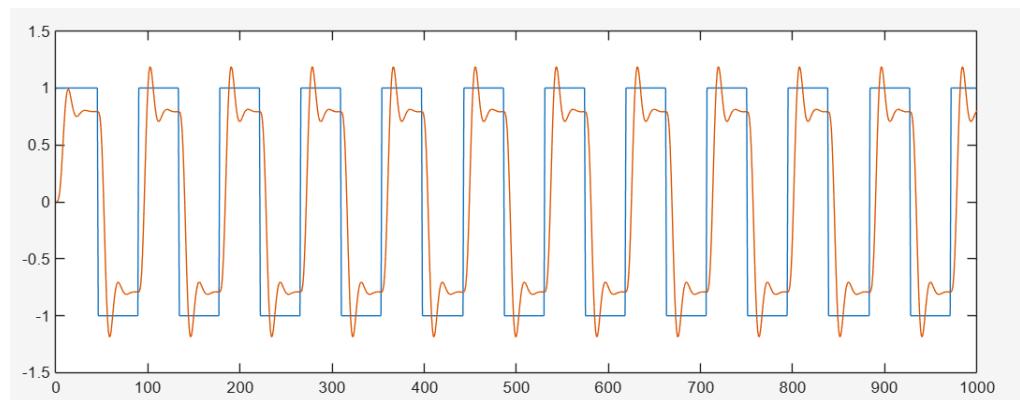


Figure 4 - Cascade = Highpass + ChebyshevLowpass