

Tworzenie i optymalizacja filtrów FIR

Raport 1 (25.03.2016)

Uniwersytet Jagielloński – Zakład Fotoniki

1 Wstęp

Celem niniejszego raportu jest podsumowanie i przedstawienie wyników uzyskanych przy analizie filtrów typu FIR z myślą o zastosowaniu w układach FPGA (field-programmable gate array), jak również podkreślenie istotnych problemów napotkanych po drodze.

Zapis współczynników filtru w układach FPGA zazwyczaj ogranicza się do kilkunastu bitów na liczbę celem uzyskania możliwie największej wydajności, dlatego istotnym problemem staje się znalezienie ‘kompromisu’ pomiędzy liczbą współczynników oraz liczbą bitów (zwanej dalej dyskretyzacją) na których zostają zapisane.

Raport składa się z dwóch części: generowanie współczynników filtru oraz ich analizy.

2 Znalezienie współczynników filtru

Współczynniki filtru FIR powinny zostać znalezione za pomocą kodu uruchamianego na CPU układu. Z tego powodu powstał program (afilters) w języku C++, mający na celu wygenerowanie współczynników filtru z zadanej, dyskretnej transmitancji.

Program używa istniejącej i otwartej biblioteki do transformaty fouriera zwanej KISS FTT (licencja BSD – <http://kissfft.sourceforge.net>).

Obecnie kod znajduje się w prywatnym repozytorium pod adresem:

<https://github.com/masteranza/SignalProcessing>

oraz posiada szereg funkcji diagnostycznych ułatwiających szybkie testowanie filtrów, takie jak generacja filtrów low-, mid-pass.

2.1 Format plików wyjściowych

Współczynniki generowane przez program współczynniki są rzeczywiste i jest ich nieparzysta ilość. Zdyskretyzowane współczynniki również są rzeczywiste (nie całkowite), dokładny opis procesu dyskretyzacji znajduje się w dalszej części.

Na obecną chwilę ilość danych w plikach wyjściowych jest również nadmiarowa.

Każdemu wierszowi odpowiada jeden współczynnik, jednak kolumn (dla potrzeb analizy i korekcji błędów jest wiele), poniżej przedstawiono pełne zestawienie poszczególnych (numerowanych) kolumn. Najistotniejszymi kolumnami są jednak kolumna: 1 (transmitancja), 9 (współczynniki) i 11 (zdyskretyzowane współczynniki)

1 – zadana na wejściu transmitancja używana przy obliczaniu współczynników, z możliwymi dodatkami, np. wygładzaniem stromych zboczy, domyślnie wyłączone

- 2 – część rzeczywista transmitancji obliczonej pomocniczo z obliczonych współczynników
- 3 – część urojona transmitancji obliczonej pomocniczo z obliczonych współczynników
- 4 – część rzeczywista transmitancji obliczonej pomocniczo z obliczonych, dyskretyzowanych współczynników
- 5 – część urojona transmitancji obliczonej pomocniczo z obliczonych, dyskretyzowanych współczynników
- 6 – wartość bezwzględna współczynników ($\sqrt{\text{Re}\{\text{wsp}\}^2 + \text{Im}\{\text{wsp}\}^2}$)
- 7 – wartość bezwzględna dyskretyzowanych współczynników
- 8 – numer współczynnika począwszy od 0.
- 9 – część rzeczywista współczynników
- 10 – część urojona współczynników (0)
- 11 – część rzeczywista zdyskretyzowanych współczynników
- 12 – część urojona zdyskretyzowanych współczynników (0)
- 13 – dodatkowa kolumna na zapis niezmodyfikowanej transmitancji w przypadku dodatków (domyślnie 0)

2.2 Przykład

Kompilacja kodu źródłowego:

```
gcc -g -lm afilters.c kiss_fft.c kiss_fftr.c -o afilters;
```

Uruchomienie i wygenerowanie filtru low-pass o szerokości 10%:

```
./afilters 0.1
```

Uruchomienie i wygenerowanie filtru mid-pass o szerokości 10% i przesunięciem 30% (0% \equiv low-pass):

```
./afilters 0.1 0.3
```

Efekt działania programu jest wygenerowanie szeregu plików (używanych później w dalszej analizie) o różnych długościach filtrów i różnej liczbie bitów użytych do dyskretyzacji.

W przypadku filtrów low pass generowane są pliki postaci:

filtr <długość filtru> – <liczba bitów>.dat,

gdzie <długość filtru> przybiera wartości: 31, 63, 127, 255, 511, 1023,

a <liczba bitów> wartości: 0, 2, 4, ..., 64.

Dla filtrów mid-pass sytuacja wygląda analogicznie, zmieniają się tylko nazwy plików, dla uniknięcia kolizji: filtr <długość filtru> m – <liczba bitów>.dat

2.3 Proces dyskretyzacji

Przy użyciu m bitów największą liczbą całkowitą możliwą do zapisania jest liczba $2^m - 1$ (pierwszy bit użyty jest do zapisu znaku). Z tego powodu w procesie dyskretyzacji najpierw znajduje się

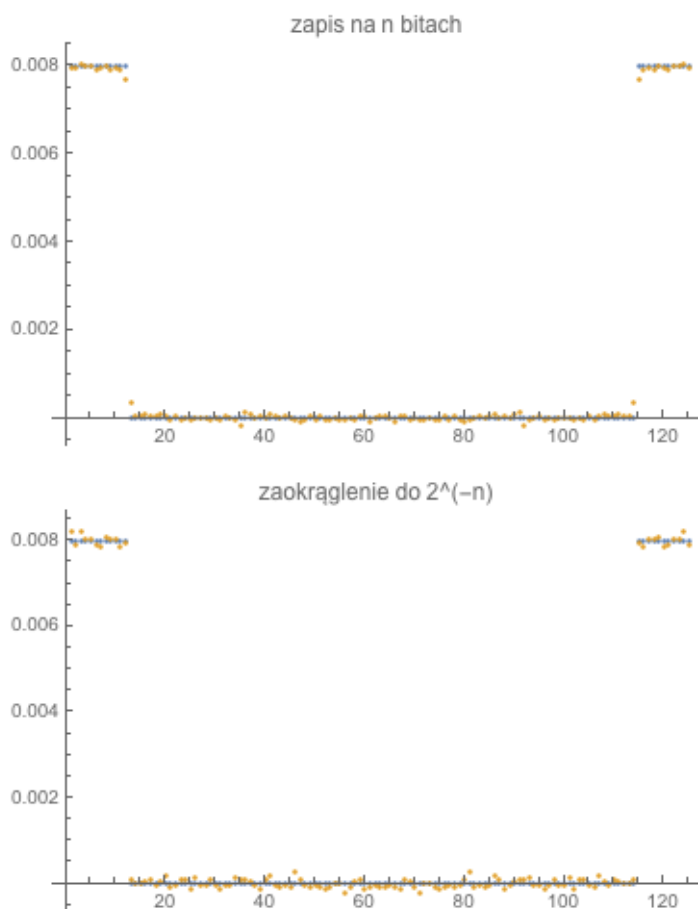
największa (\max) spośród liczb zapisanych w tablicy x . Następnie wszystkie liczby zostają podzielone/unormowane przez tę liczbę, a następnie pomnożone przez maksymalną liczbę ($2^{m-1} - 1$) i zaokrąglone do najbliższej liczby całkowitej również ustalając jej znak. Aby z tak zdyskretyzowanej liczby uzyskać liczbę, odpowiadającą oryginalnej liczbie mnożymy przez czynnik odwrotny, a więc $\max / (2^{m-1} - 1)$.

Procedurę tę ilustruje pseudokod:

```
xd[i] = ((x[i] > 0) - (x[i] < 0)) * round ((fabs (x[i]) / max) * (pow (2, m - 1) - 1))
* max / (pow (2, m - 1) - 1);
```

Podczas prac nad analizą filtrów okazało się również, że ten proces dyskretyzacji, jest różny od znacząco różny od zwykłego obcięcia liczb po przecinku.

Remark



3 Analiza filtrów w programie Mathematica

W tej części przedstawiony zostaje końcowy zestaw skryptów użytych do analizy filtrów wraz z wynikami.

Aby skrypty mogły zostać ponownie przeliczone, ten notes aplikacji Mathematica powinien znajdować się w folderze zawierającym aplikację afilters.

3.1 Inicjalizacja

3.2 Proste testy sprawdzające

Interesującym testem który można przeprowadzić jest wyrysowanie wygenerowanych współczynników oraz porównanie transformaty tych współczynników wykonanej wewnątrz programu afilters oraz w Mathematica.

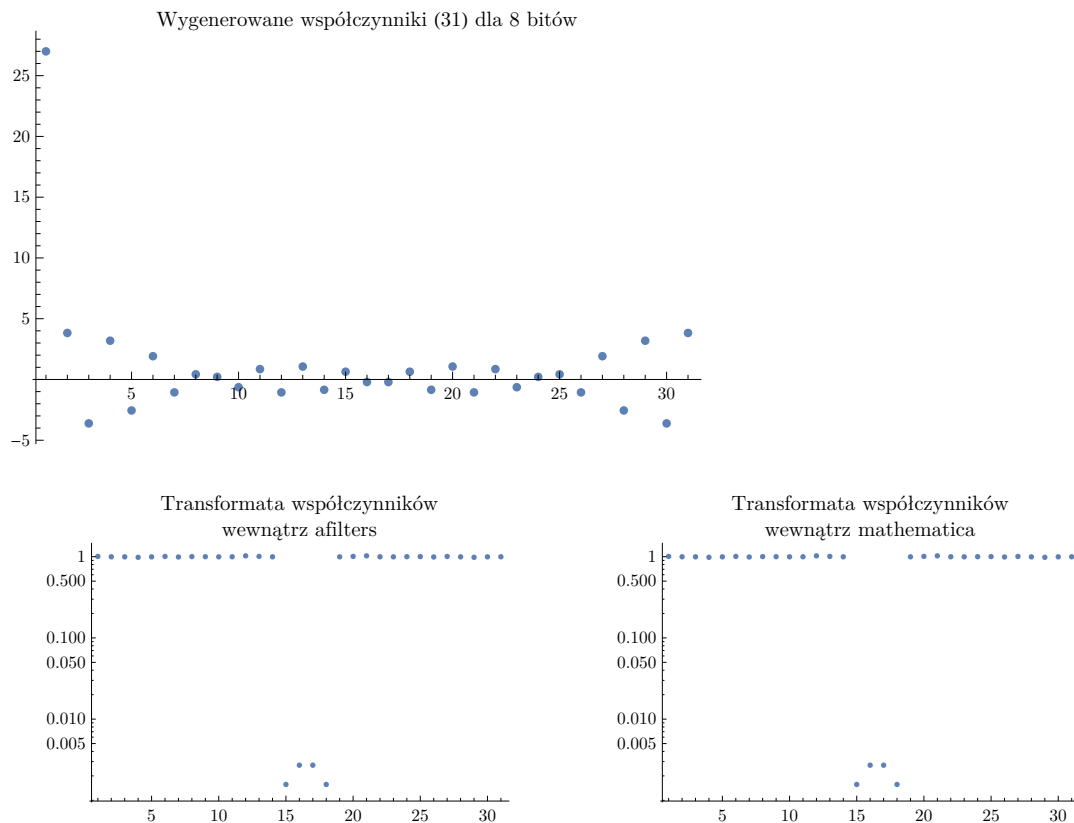
W tym celu na początek generujemy współczynniki filtrów szerokości 90% za pomocą komendy:

```
./afilters 0.9
```

lub bezpośrednio z notesu:

Ładujemy dane do notesu (należy zaczekać na wygenerowanie wszystkich plików):

Wybieramy dane dla przykładowej liczby współczynników: $wsp = 1$ (32) oraz liczby bitów: $bit = 4$ (8) i wyświetlamy je:



Wyniki są identyczne z dokładnością do stałej (liczby współczynników).

3.3 Użyta metoda szacowania błędu

Przy szacowaniu błędu dyskretyzacji współczynników wypróbowano wiele miar, będących wariacjami średniego błędu kwadratowego (porównując transformaty współczynników), jednak bardzo dogodną miarą okazało się obliczenie średniego błędu względnego na jeden współczynnik.

3.3.1 Średni błąd względny na jeden współczynnik

Oznaczmy spektrum filtru idealnego (składający się z N liczb) o szerokości w przez $T(w)$, a przez $T_M(w)$ filtr (również składający się z N liczb) uzyskany przy zapisie na M bitach. Wtedy nasza miara wyraża się przez:

$$\sqrt{\frac{\sum_{i=1}^M (T_M(w)[i] - T(w)[i])^2}{\sum_{i=1}^M (T(w)[i])^2}} \quad (3.1)$$

lub w przypadku ciągłym:

$$\sqrt{\frac{\int_0^\pi dx (T_M(w)[x] - T(w)[x])^2}{\int_0^\pi dx (T(w)[x])^2}} \quad (3.2)$$

Oczywiście aby możliwe było obliczenie błędu w przypadku ciągłym, należy wykonać najpierw DTFT (Discrete Time Fourier Transform) na zestawie współczynników.

3.4 Generowanie filtru low-pass i szacowanie błędu w procesie dyskretyzacji na M bitach

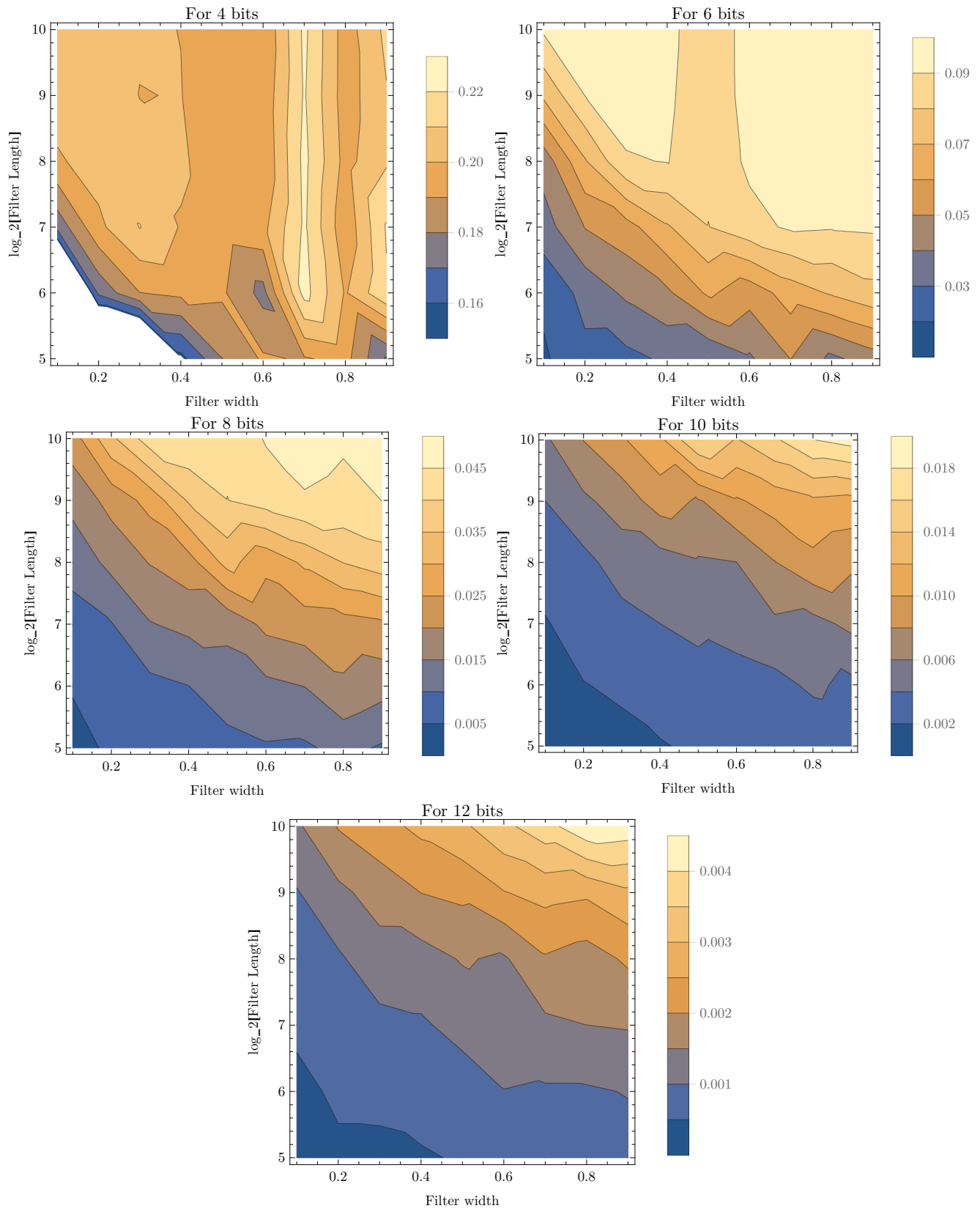
Poniższy kod pozwala na zewnętrzne wygenerowanie współczynników filtru (poprzez program `afilters`), a następnie na obliczenie błędów opisanego wcześniej procesu dyskretyzacji, w zależności od szerokości i długości filtru lub ilości bitów.

Odpowiednie fragmenty mogą zostać w prosty sposób uaktywnione poprzez od-komentowanie.

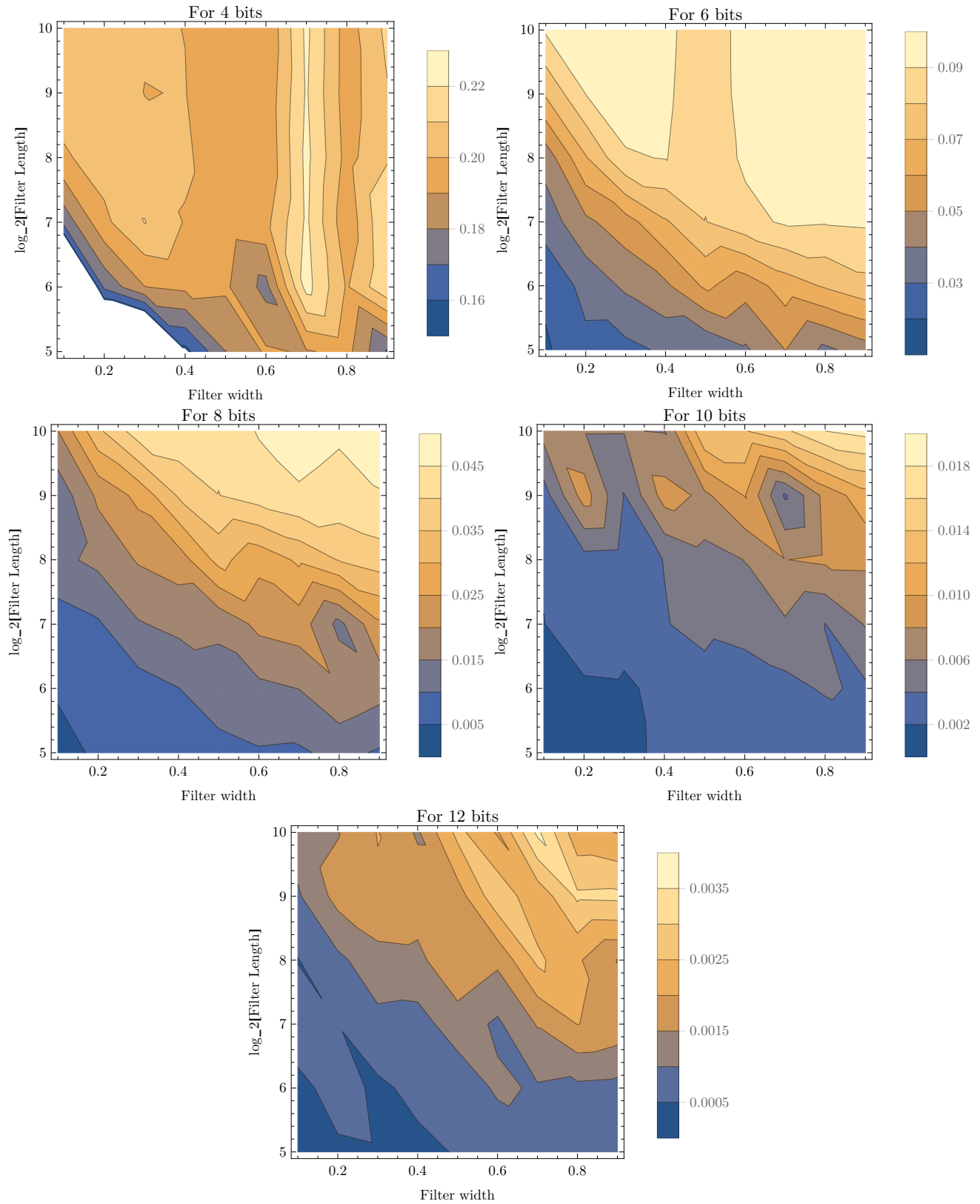
Za pomocą tego skryptu oraz:

Wygenerowano średnie błędy względne dla różnych liczb bitów zależące od szerokości i długości filtru:

3.4.1 Dla miary dyskretnej



3.4.2 Dla miary ciągłej

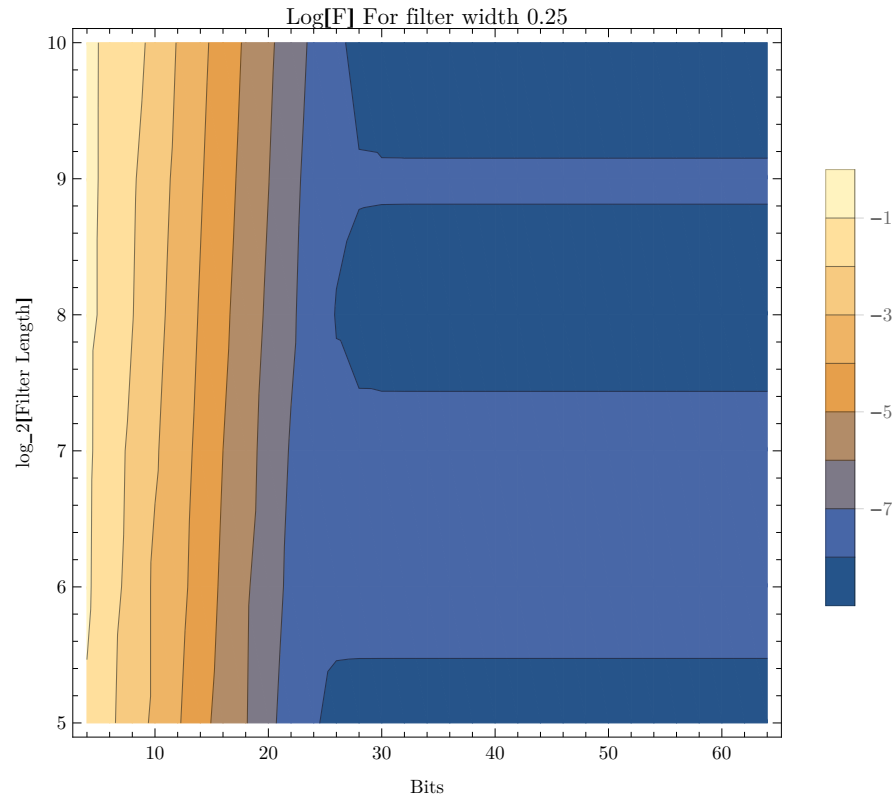


Z poniższych wykresów widać, iż obie miary pokrywają się ze sobą w dobrym stopniu.

3.5 Zanik błędu z rosnącą liczbą współczynników

Za pomocą podobnego skryptu:

Wygenerowano zależność średniego błędu względnego od długości filtru i liczby bitów użytej przy dyskretyzacji, otrzymując:



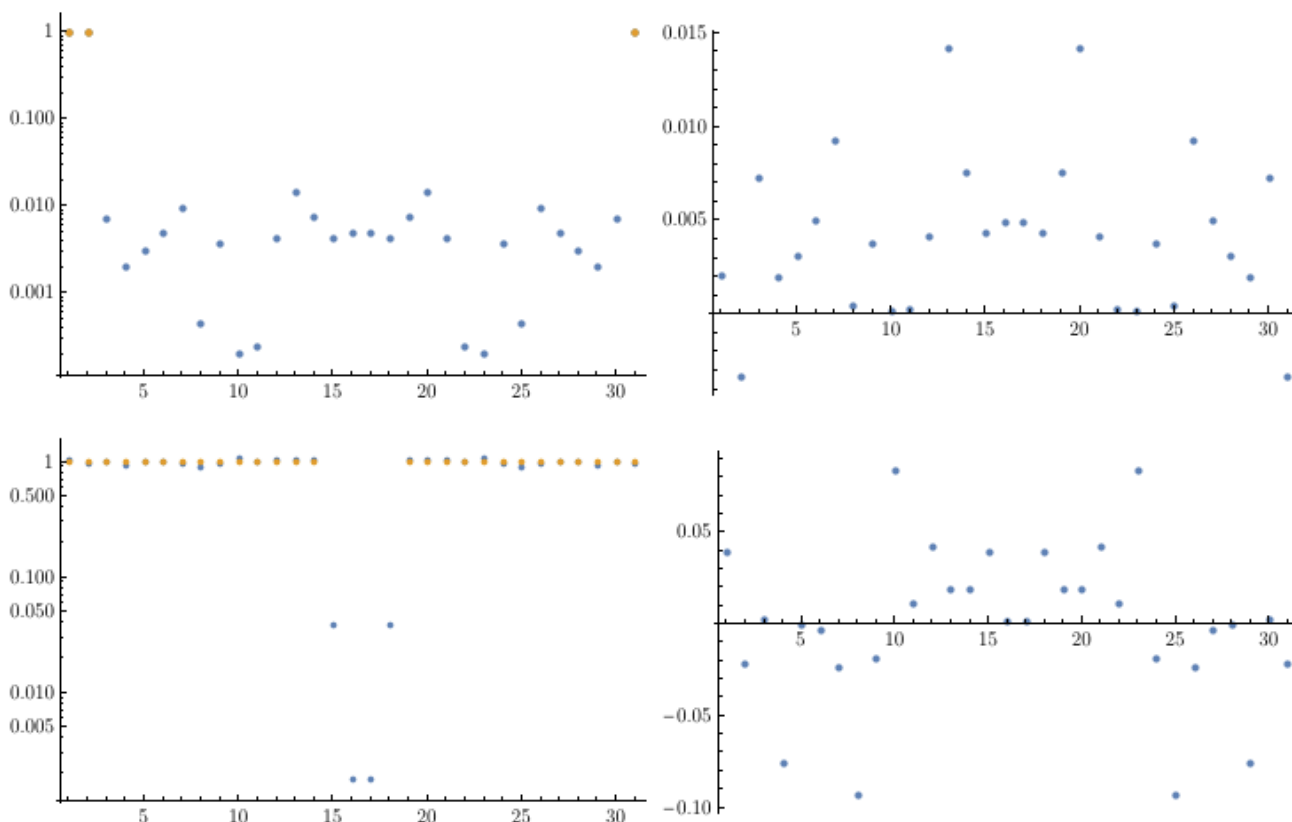
Wykres wskazuje (na spodziewany) zanik błędu z rosnącą liczbę bitów.

3.6 Problem rosnącego błędu z szerokością filtru

Nieco zastanawiający okazał się wzrost błędu przy rosnącej szerokości filtru, zwłaszcza (jak widać) w przypadku niskiej ilości bitów.

Do tej pory nie znaleziono zadowalającego wytłumaczenia tego fenomenu.

Prosta heurystyka, przeprowadzona przy 6 bitach i mierze dyskretnej daje następujące wyniki dla filtrów 10% (pierwszy wiersz) i 90% (drugi wiersz):



Pierwsza kolumna przedstawia spektrum idealne i spektrum zdyskretyzowane na jednym wykresie, zaś druga przedstawia różnicę między nimi.

Licznik równania (3.1) dla przypadku filtru 10% wynosi 0.00104495, a mianownik 3.

Dla filtru 90%, licznik wynosi 0.0555405, a mianownik 27.

Zatem błąd zdecydowanie staje się większy gdy próbujemy utworzyć szerszy filtr.

4 Podsumowanie

Przedstawiona tutaj metoda oraz zestaw skryptów pozwala na analizę jakości dowolnych filtrów z udziałem różnych miar – zarówno ciągłych jak i dyskretnych.

Dzięki uzyskanym wynikom można, w prosty sposób dopasować parametry pracy filtru FIR w zależności od możliwości dostępnego sprzętu.

Nasza wiedza jest jednak nie pełna. Do tej pory nie udało się wytłumaczyć, dlaczego średni błąd względny na jeden współczynnik rośnie wraz z długością filtru.