

PCPS — Opis systemów dyskretnych, filtry FIR i IIR

Krzysztof Mazur, Małgorzata Michalczyk, Piotr Krauze

10 grudnia 2018


Wprowadzenie


Dokument ten zawiera informacje pomocne w wykonaniu ćwiczenia 2: „Opis systemów dyskretnych, filtry FIR i IIR” z Laboratorium Podstaw Cyfrowego Przetwarzania Sygnałów. Przed ćwiczeniem należy zapoznać się z celem ćwiczenia oraz wymaganą wiedzą i umiejętnościami wymaganymi na zajęciach. Zapoznanie się z planem ćwiczenia nie jest konieczne, choć może ułatwić zaliczenie ćwiczenia. Dokument ten zawiera także dodatkową wiedzę oraz zadania, których wykonanie jest dodatkowo premiowane. Wykonanie ich nie jest obowiązkowe. Treści te zostały oznaczone poprzez znak:



1 Cel ćwiczenia

Celem ćwiczenia jest sprawdzenie swojej wiedzy, za pomocą kartkówki, z zakresu opisu systemów dyskretnych, oraz zastosowanie tej wiedzy wraz z wiedzą dotyczącą filtrów FIR i IIR do problemu zmiany częstotliwości próbkowania.

 Problem zmiany częstotliwości próbkowania jest bardzo powszechny. Bardzo często jest związany ze stosowaniem nadpróbkowania. Nadpróbkowanie jest skuteczną metodą redukcji szumów, zwiększenia efektywnej rozdzielności przetwornika, oraz zmniejsza wymagania na analogowe filtry antyaliasingowe i rekonstrukcyjne. Biegłość w tym zakresie pozwala na stosowanie prostszych i tańszych filtrów analogowych oraz tańszych przetworników analogo-cyfrowych i cyfrowo-analogowych (także takich wbudowanych w mikrokontroler), a także uzyskanie lepszych efektów niż rozwiązanie klasyczne, np. charakterystyka filtrów analogowych zależy od parametrów elementów analogowych, których parametry nigdy nie są idealne i także mogą się zmieniać w czasie. Poza tym w niektórych systemach określona częstotliwość próbkowania jest wymuszona (sprzęt został zaprojektowany na określoną częstotliwość lub określoną minimalną częstotliwość).

 Problem zmiany częstotliwości próbkowania jest bardzo podobny do problemu samego próbkowania sygnału ciągłego. W pewnym przybliżeniu można uznać, że sygnał ciągły to sygnał z nieskończenie dużą częstotliwością próbkowania. W przypadku przechodzenia z sygnału ciągłego na dyskretny lub dyskretnego na analogowy konieczne jest stosowanie filtrów **ciągłych** w czasie (zwykle „analogowych”). Znacząco to ogranicza możliwości możliwych do uzyskania charakterystyk filtrów.

2 Wymagana wiedza i umiejętności

Na zajęciach **wymagana** jest następująca wiedza oraz umiejętności:

1. znajomość teorii o próbkowaniu sygnałów: twierdzenie Shannona-Kotelnikova, częstotliwość próbkowania, częstotliwość Nyquista, co to jest aliasing?
2. GWM (Gęstość Widmowa Mocy) sygnałów dyskretnych: jak można je obliczyć, jak wygląda w całym paśmie częstotliwości, jakie pasmo częstotliwości w przypadku sygnałów rzeczywistych wystarcza?
3. opis systemów dyskretnych: równanie różnicowe, transmitancja, charakterystyki częstotliwościowe filtrów, opóźnienie fazowe filtru, opóźnienie grupowe filtru
4. podstawowa wiedza z projektowania filtrów FIR oraz IIR, podstawowe filtry: Butterwortha, Czebyszewa I-go i II-go rodzaju, filtry Eliptyczne, filtr „brick-wall” (sinc)
5. umiejętność programowania w języku Matlab. Umiejętność obsługi środowiska Matlab lub/i GNU Octave. Zalecana znajomość następujących funkcji: `abs()`, `angle()`, `fft()`, `filter()`, `freqz()`, `pwelch()`, `rand()`, `sinc()`, `sin()`, `unwrap()`, `zeros()`.

3 Plan ćwiczenia

3.1 Generacja sygnału

Wygeneruj 16000 próbek sygnału dyskretnego x_1 będącego sumą trzech składowych sinusoidalnych: f_1 , f_2 oraz f_3 o amplitudzie 0.25. Częstotliwość próbkowania wynosi $f_s = 8000$ Hz.

🔍 GWMy chcemy wykreślać z rozdzielczością 1 Hz, bin częstotliwości ma mieć 1 Hz. Musimy mieć wtedy tyle próbek ile wynosi częstotliwość próbkowania. Dlaczego generujemy 2 razy więcej? Będziemy wykorzystywać filtry, które wykorzystują także próbki wcześniejsze, w przypadku filtrów IIR nieskończenie wiele próbek wcześniejszych, choć wpływ dalszych próbek jest coraz mniejszy. Z tego powodu musimy odrzucić początkowe próbki przefiltrowanego sygnału, dla uproszczenia odrzucamy połowę próbek, czyli potrzebujemy oryginalny sygnał dwukrotnie dłuższy. Należy nie zapomnieć o normalizacji względem liczby próbek (podzielić przez liczbę próbek).

3.2 Decymacja sygnału

Przeprowadź decymację sygnału poprzez wybór co N-tej próbki z sygnału x_1 , tworząc sygnał x_2 . Decymację można przeprowadzić w języku Matlab za pomocą:

```
x_2 = x_1(1:N:end);
```

Wykreśl przebieg czasowy sygnałów: x_1 oraz x_2 . Wykreśl na jednym wykresie GWM ostatnich f_s próbek sygnałów x_1 oraz x_2 . GWM należy wykreślić od zera do częstotliwości Nyquista dla każdego z sygnałów. Opisz czym różnią się widma tych sygnałów, oraz co jest przyczyną tych różnic.

Ostatnie f_s próbek w języku Matlab można uzyskać za pomocą:

```
y = x(end-(f_s-1):end);
```

3.3 Projektowanie filtru

Wyjaśnij jakie zjawisko jest powodem pojawiania się nowych częstotliwości w sygnale. Zaproponuj filtr cyfrowy H eliminujący to zjawisko, a zarazem nie mający istotnego wpływu na sygnał poprawnie odtwarzany. Przyjmij następujące kryteria:

- częstotliwości mniejsze niż $\alpha \frac{f_s}{2N}$ mają być przenoszone z błędem nie większym jak R_p decybeli
- częstotliwości nie mniejsze niż $\frac{f_s}{2N}$ mają być tłumione co najmniej R_s decybeli

Parametry α , R_c oraz R_s są podane przez prowadzącego.

Zastosuj filtr Butterwortha. Parametry w postaci wielomianowej $H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}$ filtru Butterwortha n -tego rzędu, o znormalizowanej częstotliwości odcięcia f_c (wartość 1 to częstotliwość Nyquista) można wygenerować za pomocą:

```
[B, A] = butter(n, f_c);
```

Do określenia minimalnego rzędu oraz częstotliwości odcięcia spełniające powyższe wymagania można się posłużyć funkcją `buttord()`.

Przefiltruj sygnał x_1 przez filtr H , tworząc sygnał x_3 .

2 Zastosuj alternatywnie filtr „brick-wall” (sinc). Parametry odpowiedzi impulsowej tego filtru można wyznaczyć za pomocą: $i=(-M:M)'$; $H = f_c * \text{sinc}(f_c * i)$; , gdzie M to parametr określający liczbę parametrów filtru, filtr ma $2M+1$ elementów odpowiedzi impulsowej. f_c to znormalizowana częstotliwość odcięcia filtru. Uzasadnij dobór f_c oraz M .

3.4 Decymacja przefiltrowanego sygnału

Przeprowadź decymację przez wybór co N -tej próbki sygnału x_3 , tworząc nowy sygnał x_4 .

Wykreśl na jednym wykresie GWM sygnałów x_1 , x_3 oraz x_4 . GWM należy wykreślić od zera do częstotliwości Nyquista dla każdego z sygnałów. Opisz czym różnią się widma tych sygnałów, oraz co jest przyczyną tych różnic. W czym sygnał x_4 jest „lepszy” od sygnału x_2 ?

3.5 Zwiększenie częstotliwości próbkowania sygnału

Zwiększ N -krotnie częstotliwość próbkowania sygnału x_4 oraz zwiększ jego amplitudę N -krotnie uzyskując sygnał x_5 . Sygnał ten powinien mieć długość równą długości sygnału x_3 (sygnał x_4 przed decymacją) oraz powinien być uzupełniony zerami. Można to przeprowadzić w języku Matlab za pomocą:

```
x_5 = zeros(size(x_3));  
x_5(1:N:end) = N * x_4;
```

Wykreśl na jednym wykresie GWM sygnałów x_3 , x_4 oraz x_5 . GWM należy wykreślić od zera do częstotliwości Nyquista dla każdego z sygnałów. Opisz czym różnią się widma tych sygnałów, oraz co jest przyczyną tych różnic. Czy widma sygnałów x_4 oraz x_5 się tak naprawdę czymś istotnie różnią?

3.6 Filtracja sygnału

Przefiltruj sygnał x_5 przez wcześniej projektowany filtr H , tworząc sygnał x_6 .

Narysuj na jednym wykresie czasowym fragment (kilka okresów) sygnałów x_3 oraz x_6 . Czym różnią się te sygnały?

Wykreśl na jednym wykresie GWM sygnałów x_3 , x_4 oraz x_5 oraz x_6 . GWM należy wykreślić od zera do częstotliwości Nyquista dla każdego z sygnałów. Do którego z sygnałów x_3 , czy x_4 jest teraz podobny sygnał x_6 ?

Wykreśl charakterystykę amplitudową filtru H . Charakterystykę filtru można otrzymać za pomocą:

```
[H, f] = freqz(B, A, 4000);
```

gdzie B i A to współczynniki wielomianów transmitancji, 4000 to liczba binów dla których charakterystyka jest wyznaczana, H to wektor z uzyskaną charakterystyką, a f to wektor częstotliwości odpowiadającym elementom wektora H . Charakterystykę można wykreślić za pomocą:

```
plot(w, 20*log10(abs(H)));
```

⚡ Przeprowadź powyższe rozważania także dla alternatywnego filtru H , „brick-wall” (sinc).

3.7 Opóźnienie filtrów

⚡ Narysuj charakterystykę opóźnienia filtru H , w próbkach. Jakie opóźnienie ma filtr H dla częstotliwości f_1 ?

Opóźnienie filtru w próbkach można uzyskać za pomocą:

```
Hd = -unwrap(angle(H(2:end))) ./ f(2:end);
```

Uwaga, opóźnienie dla 0 Hz zostało pominięte by nie dzielić przez 0.

⚡ Skompensuj opóźnienie filtru H w sygnale x_6 . Narysuj na jednym wykresie czasowym sygnał x_3 oraz odpowiednio przesunięty sygnał x_6 bazując na wyznaczonym opóźnieniu dla f_1 .