| 02 | **Cyfrowe Przetwarzanie Sygnałów**  Transformacje ortogonalne prof. dr hab. inż. Tomasz Zieliński, dr inż. Jarosław Bułat | 2018.03.06 |
| --- | --- | --- |

# 1. Macierze transformacji (1 pkt)

Według poniższego wzoru wygeneruj wzorce kosinusowe w postaci wektorów i utwórz z nich macierz transformaty – macierz analizy **A**transformaty DCT-II. Niech

*, N=20, k=0...N-1, n=0...N-1,*

oznacza *k*-ty wiersz macierzy analizy **A** (20x20) będącej bazą pewnej transformaty.

Sprawdź czy wszystkie wektory (wiersze macierzy) są do siebie ortonormalne (czy iloczyn skalarny wszystkich par jest równy zero: suma iloczynów odpowiadających sobie próbek).

Poniżej podano przykład obliczania iloczynu skalarnego różnymi metodami.

clear all; close all

w1 = [ 0 0 1 0 0 0 0 0 ]; % Wektor 1

w2 = [ 0 0 0 0 1 0 0 0 ]; % Wektor 2

w12 = w1 .\* w2; % Iloczyn odpowiadających sobie próbek

prod1 = sum(w12) % ,,0'' oznacza że wektory są ortogonalne

prod2 = dot( w1, w2 ) % w przestrzeni Euklidesowej

prod3 = w1\*w2' % bezpośrednie obliczenie (mnożenie wektorowe)

# 2. Transformacja odwrotna – perfekcyjna rekonstrukcja (1+0.25 pkt)

Wygeneruj macierz odwrotną (syntezy) **S**=IDCT do macierzy DCT z pkt. 1 (transponuj macierz **A**, czyli zamień wiersze na kolumny), sprawdź czy **SA==I** (macierz identycznościowa), a następnie mając **A** i **S** wykonaj analizę:

oraz rekonstrukcję (syntezę):

sygnału sygnału losowego (funkcja randn()), sprawdź czy transformacja posiada właściwość perfekcyjnej rekonstrukcji, (**xs==x** ?).

Dla dociekliwych (+0.25 pkt): wygeneruj macierz kwadratową **A** za pomocą funkcji randn() dla *N=*20. Sprawdź ortonormalność jej wierszy (czy norma wierszy=1). Wyznacz macierz odwrotną S=inv(A). Sprawdź, czy **AS==I**?, czyli czy sekwencja operacji **y**=**Ax**, **xs**=**Sy** posiada właściwość perfekcyjnej rekonstrukcji. Dokonaj analizy i syntezy dowolnego sygnału losowego jak powyżej oraz sprawdź czy **xs**==**x** ?

„Zepsute” DCT: wygeneruj macierz **A** dla DCT, podstawiając niepoprawne indeksy (wartości) częstotliwości, np. zastąp „*k*” przez „*k+*0.25” (wzór w pkt. 1). Sprawdź ortogonalność tej macierzy, sprawdź wynik analizy oraz perfekcyjną rekonstrukcję na sygnale szumowym i harmonicznym.

# 3. Analiza częstotliwościowa (3 pkt)

Przyjmij liczbę próbek sygnału *N=*100 i częstotliwość próbkowania *fs=*1000 Hz. Wygeneruj sygnał *x* będący sumą trzech sinusoid o częstotliwościach *f1=*50*, f2=*100*, f3=*150 Hz i amplitudach *A1=*50*, A2=*100*, A3=*150 (odpowiednio).

Zbuduj macierze **A**=DCT i **S**=IDCT dla *N=*100 (patrz wzór w ćwiczeniu 1). Wyświetl w pętli wartości wszystkich wierszy macierzy **A** i kolumn macierzy **S**, tzn. pierwszy wiersz **A**, poniżej pierwsza kolumna **S**, drugi wiersz macierzy **A** i druga kolumna macierzy **S**, itd. Wyświetlaj oba przebiegi na jednym wykresie, użyj pętli i instrukcji pause.

Wykonaj analizę **y=Ax** i wyświetl wartości y(1:N): porównaj wartości współczynników niezerowych  
z wartościami amplitud składowych sygnału oraz porównaj numery współczynników niezerowych  
z wartościami częstotliwości składowych sygnału. Wyskaluj oś poziomą w częstotliwości: zastąp n=1:N przez f=(0:N-1)\*fs/N. Czy teraz wynik analizy (pokazywane częstotliwości i amplitudy składowych sygnału) jest poprawny? Sprawdź perfekcyjną rekonstrukcję (**xr=Sy**, **xr==x**?).

Zmień częstotliwość *f2=*100 Hz drugiej składowej sygnału na *f2=*105 Hz, wykonaj analizę sygnału sumarycznego (**y=Ax**) i wyświetl wyskalowany w hercach wykres *y(f)*. Składowa o *f2=*105 Hz jest teraz rozmyta, ponieważ jej wzorca nie ma w zestawie funkcji bazowych (w wierszach macierzy **A**). Sprawdź czy mimo to jest możliwa perfekcyjna rekonstrukcja sygnału (**xr=Sy**, **xr==x** ?).

Zwiększ wszystkie częstotliwości sygnału o 2.5 Hz (przesunięcie spowodowane przez zastosowanie niepoprawnej częstotliwości w konwerterze sygnału telekomunikacyjnego do pasma podstawowego wokół 0 Hz). Wyświetl wynik analizy. Zwróć uwagę na rozmycie wszystkich składowych.

*Dlaczego niektóre współczynniki analizy (***y***) mają duże wartości? Ponieważ analizowany sygnał dobrze koreluje się (iloczyn skalarny) z niektórymi wierszami macierzy* **A** *(wzorcami częstotliwości).*

*Dlaczego możliwa jest rekonstrukcja sygnału? Ponieważ wiedząc „ile” (***y***) każdego wzorca częstotliwości jest w sygnale (wzorce znamy), można te wzorce częstotliwości wymnożyć przez „ile” i zsumować przeskalowane sygnały (***xr=By***), odtwarzając w ten sposób analizowany sygnał (pierwsza próbka sygnału zrekonstruowanego jest sumą pierwszych próbek wszystkich przeskalowanych wzorców, druga ... sumą drugich, itd.; w kolumnach macierzy B mamy wzorce, które są wykorzystywane do rekonstrukcji, y to informacja o „ile”).*

# 4. Sygnały rzeczywiste (opcjonalnie – dla dociekliwych) (+0.25 pkt)

Wczytaj do Matlaba sygnał z pliku mowa.wav, lub inny własny plik, wykorzystując funkcję  
[x, fs] = audioread('mowa.wav') i wyświetl go (w osi x numer próbki). Następnie wybierz wzrokowo z tego sygnału *M=*10 różnych fragmentów *xk=x*(*n1:n2*) o długości *N=*256 próbek, *k=*1,2,3*,..M*, oblicz dla nich *yk = Axk*. Następnie wyświetl w pętli dane na podzielonym rysunku: góra – *k*-ty fragment *xk*(*n1:n2*), dół – wynik analizy *yk*(*f*), wyskalowany w hercach. Jako macierz analizy przyjmij DCT z punktu 1.

# 5. DAB (+1 pkt)

W zadaniu 4 laboratorium 1 analizowano odebrany sygnał czasowy radia DAB, idealny i zniekształcony. Wykorzystywano w nim funkcję PhaseRefSymbGen(), która zwraca referencyjny sygnał czasowy sigPhaseRefSymb, występujący w sygnale DAB, oraz jego widmo Fouriera PhaseRefSymb. Widmo to jest otrzymywane za pomocą równania **X=Ax’** z sygnału czasowego (jak wyżej, gdzie **x** to końcowe 2048 wartości sygnału sigPhaseRefSymb; pierwsze 504 wartości odrzucamy, gdyż są one specjalnie wykonanym powtórzeniem ostatnich 504 próbek sygnału; jest to tzw. cykliczny prefiks). W tym przypadku macierz **A** jest przeskalowaną macierzą **F** dyskretnej transformacji Fouriera (patrz zadanie 2 w ćwiczeniu 4 TOwNiT). Wygeneruj macierz **F** dla *N=*2048, podstaw , oblicz **X=Ax’** i porównaj wynik z PhaseRefSymb (alternatywnie w Matlabie: X=fft(x)). Następnie obróć każdą NIEZEROWĄ wartość wektora **X** (zespoloną) o losowo wybrany jeden kąt ze zbioru: {*+π*/4*, +3π* /4*, +5π*/4 *albo +7π*/4} (czyli wykonaj mnożenie ),co odpowiada przesłaniu na każdej używanej częstotliwości jednej z par bitów {00, 01, 10 albo 11} metodą różnicowej modulacji fazowej 4-DQPSK (*Differential Quadrature Phase Shift Keying*). Otrzymany w ten sposób wektor oznacz jako **Y**. Potem oblicz **y=SY** dla (alternatywnie w Matlabie: y=ifft(Y)) oraz skopiuj na początek jego ostatnich 504 wartości. Oznacz sygnał otrzymany w ten sposób przez SigSymb (2552 próbki). W taki sposób był otrzymany sygnał DAB, który analizowałeś w zadaniu 4 laboratorium DSP nr 1. W radiostacji zaczynano od widma Fouriera PhaseRefSymb, a następnie je obracano kątowo po raz pierwszy, kodując bity w tych obrotach, potem wynik pierwszego obrotu obracano jeszcze raz, itd., (łącznie 75 razy). Łącznie mieliśmy 76 bloków po 2552 próbki (prefiks 504 + sygnał 2048).

Co dalej? Znasz sygnały sigPhaseRefSymb oraz SigSymb, w sygnale DAB występują one jeden za drugim (przed nimi jest NullSymbol). Wykorzystaj je do obliczenia jaką sekwencję bitów otrzymano i porównaj ją z sekwencją bitów nadaną. W tym celu oblicz: widma obu sygnałów metodą macierzową (po usunięciu cyklicznego prefiksu, czyli ich pierwszych 504 wartości), ich obrót kątowy względem siebie i przelicz kąty na bity (za pomocą funkcji sign(), działającej na części rzeczywistej real() i urojonej imag() obliczonych zespolonych mnożników kątowych). Co się stanie jeśli:

1. do sygnałów sigPhaseREfSymb oraz SigSymb dodał się szum gaussowski (w Matlabie funkcja randn()) o amplitudzie równej 1%, 5%, 10% albo 25% amplitudy sygnału przesyłanego?
2. jesteś źle zsynchronizowany, tzn. źle odcinasz cykliczny prefiks i nie zaczynasz od 505 próbki tylko np. od 500. Dlaczego przesunięcie to nie pogarsza wyniku detekcji bitów?