Szybka transformata kosinusowa

Naszym zadaniem będzie zastosowanie szybkiej transformaty kosinusowej do odszumienia sygnału periodycznego. Sygnał zaszumiony generujemy zgodnie z poniższym algorytmem:

a) sygnał okresowy nie zaszumiony ma postać

$$y_0(i) = cos(\omega \cdot i) + cos(2\omega \cdot i) + cos(3\omega \cdot i) \tag{1}$$

gdzie: i - numer próbki sygnału (numer elementu w wektorze),

$$\omega = 2\frac{2\pi}{n} \tag{2}$$

n - ilość próbek

b) tworzymy zmienną losową imitującą szum

$$a = 2sign \cdot X \tag{3}$$

gdzie:

$$X = \frac{rand()}{RAND\_MAX + 1.0} \tag{4}$$

jest liczbą pseudolosową o rozkładzie równomiernym w przedziale (0,1).
Znak zmiennej określamy następująco: losujemy drugą zmienną losową Y (podobnie jak X) następnie dokonujemy wyboru

$$sign = \begin{cases} +1, & Y > \frac{1}{2} \\ -1, & Y \leqslant \frac{1}{2} \end{cases} \tag{5}$$

c) sygnał zaszumiony konstruujemy następująco

$$y(i) = y_0(i) + a \tag{6}$$

wyznaczając wartość a dla każdego indeksu i z osobna

Zadania do wykonania:

1. Zapisać zaszumiony sygnał do wektora typu float. Długość wektora wynosi $n = 2^k$, przyjmujemy $k = 10$
2. Wykonać transformatę kosinusową sygnału korzystając z funkcji
   cosft2(float data[],int n,int typ)
   gdzie: $typ = 1$ dla transformacji zwykłej i $typ = -1$ dla transformacji odwrotnej
3. Dokonać dyskryminacji w transformacie na poziomie 25% wartości maksymalnej
4. Po dyskryminacji (wyzerowaniu sygnału o amplitudzie niższej od progu dyskryminacji) wyznaczyć transformatę odwrotną tj. użyć jeszcze raz tej samej procedury (**cosft2**), ale sygnał wyjściowy przemnożyć przez $2/n$ (normalizacja)
5. Wykonać rysunki:
   - jeden wykres sygnału zaszumionego
   - jeden wykres transformaty w pełnym zakresie
   - jeden wykres transformaty na którym wyraźnie widoczne będą piki pochodzące od modów dominujących w sygnale - znajdują się one na początku wykresu transformaty
   - na jednym rysunku narysować sygnał niezaszumiony y_0 oraz sygnał po odszumieniu.

6. Sprawdzimy która z metod zadziała szybciej dla naszego problemu - transformacja kosinusowa czy standardowa zespolona FFT radix-2. Czas wykonywania transformacji kosinusowej możemy policzyć w poniższy sposób:

```
#include <time.h>
.
.
.
clock_t begin = clock();
cosft2(data,n,1);
clock_t end = clock();
double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
```

Do wykonania zespolonej FFT wykorzystujemy funkcję:
four1(float data2[],int n,int typ)
gdzie data2 ma 2*n elementów, części rzeczywiste próbek sygnału znajdują się w elementach nieparzystych tablicy, części urojone w parzystych. Wyliczamy czas wykonywania four1 analogicznie do cosft2. Do sprawozdania: która metoda działa szybciej i dlaczego?