**Obraz zawierający logo, tekst, Czcionka, Grafika

Opis wygenerowany automatycznie**

**Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,**

**INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

Raport

**Podstawy telekomunikacji**

Autor: Grzegorz Lis, Karolina Sawosz

Kierunek studiów: Mikroelektronika w Technice i Medycynie

# Kraków, 2023

**Laboratorium 2**

***Opis problemu***

Celem laboratoriów było sprawdzenie, czy baza jest ortogonalna oraz ortonormalna, sposób jej normalizacji, rozwinięcie szeregu Haara, weryfikacja twierdzenia Parsevala, rozwinięcie szeregu Walsha, używanego do aproksymacji sygnałów sinusoidalnych oraz trójkątnych, kodowanie Manchester.

***Podstawy matematyczne rozwiązania***

Ortogonalność – dwa wektory w określonej przestrzeni są ortogonalne, gdy ich iloczyn skalarny wynosi 0.

Ortonormalność – dwa wektory są ortonormalne, gdy są ortogonalne oraz jednostkowe (ich długość wynosi 1).

Norma wektora – długość wektora.

Szereg Haara:

Twierdzenie Parsevala - suma (lub całka) kwadratu funkcji równa się sumie (lub całce) kwadratu jej transformaty.

Dla każdego ortonormalnego ciągu {} w przestrzeni , zachodzą trzy warunki:

1. Skończone liniowe kombinacje funkcji {} tworzą zbiór gęsty z przestrzeni ,
2. Jedyny element należący do przestrzeni , który spełnia warunek E(X) = 0 dla każdego n, jest X = 0,
3. Dla każdego X należącego do zachodzi własność Parsevala

Szereg Walsha (funkcja podstawowa):

***Symulacje i obserwacje***

Zadanie 1

a) Listing 1. Sprawdzenie ortogonalności wektorów.

function y = orthogonal(a, b)

if length(a) == length(b)

c = a.\*b;

c = sum(c);

if round(c,10) == 0

y=1;

else

y=0;

end

else

y=0;

end

end

function y = orthogonal3(a, b, c)

v1 = orthogonal(a,b);

v2 = orthogonal(b,c);

v3 = orthogonal(a,c);

if (v1 == 1) && (v2 == 1) && (v3 == 1)

y=1;

else

y=0;

end

end

Iloczyn skalarny wektorów a, b, c jest równy 0, zatem wektory są ortogonalne. W podanym rozwiązaniu zastosowano funkcję round(\*,10) zaokrąglającą do 10 liczb po przecinku, ponieważ jest to dokładność wystarczająca, a MATLAB zaokrągla liczby niewymierne (np. ), przez co zwracał fałsz przy porównywaniu wyników z liczbą 1

Listing 2. Sprawdzenie ortonormalności wektorów oraz ich normalizacja.

y = orthonormal3(a,b,c)

y\_normalized = orthonormal3(normalize(a),normalize(b),normalize(c))

function y = vector\_length(a)

a = a.^2;

y = round(sqrt(sum(a)),10);

end

function y = normalize(a)

y = a ./ vector\_length(a);

end

function y = orthonormal3(a, b, c)

if (orthogonal3(a, b, c) == 1) && (vector\_length(a) == 1) && (vector\_length(b) == 1) && (vector\_length(c) == 1)

y = 1;

else

y = 0;

end

end

We wcześniejszym etapie zadania została sprawdzona ortogonalność tych samych wektorów, więc powyższy listing zawiera etap dotyczący wyłącznie ortonormalizacji. Jak zostało wspomniane w matematycznych podstawach, wektory są ortonormalne, gdy są ortogonalne oraz ich długość (norma) wynosi 1. Normę wektora należy rozumieć przez pierwiastek z sumy kwadratów współrzędnych wektora.

Obraz zawierający zrzut ekranu, linia

Opis wygenerowany automatycznie

Rysunek . Wynik funkcji orthonormal bez normalizacji oraz z normalizacją wektorów.

Wynikiem funkcji orthonormal3() bez normalizacji jest 0, więc wektory nie są ortonormalne, natomiast po ich normalizacji ten wynik wynosi 1, zatem wektory są ortonormalne zgodnie  
z przypuszczeniami.

b) Listing 3. Implementacja macierzy przejścia w bazie Haara.

y = haar(8)

function y = haar(number)

sets = zeros(number,2);

k=0;

j=0;

for i = 0 : number-1

sets(i+1,:) = [j,k];

if k == (2^j)-1

j = j+1;

k = 0;

else

k = k+1;

end

end

y = zeros(number, number);

y(:,1) = 1;

for i = 2 : number

for t = 0: number-1

if (t/number) >= (sets(i-1,2))/(2^sets(i-1,1)) && (t/number) < (0.5+sets(i-1,2))/(2^sets(i-1,1))

y(t+1,i) = 1;

elseif (t/number) >= (0.5+sets(i-1,2))/(2^sets(i-1,1)) && (t/number) < (1+sets(i-1,2))/(2^sets(i-1,1))

y(t+1,i) = -1;

else

y(t+1,i) = 0;

end

end

end

end

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek . Generacja pierwszych ośmiu wektorów w bazie Haara.

Mając określony sygnał w postaci ośmioelementowego wektora, należałoby go transponować, a następnie przemnożyć przez wyliczoną macierz przejścia.

Listing 4. Weryfikacja twierdzenia Parsevala.

signal = sin(2\*pi\*t);

dft\_sig = dft(signal);

area\_sig = sum(abs(signal).^2);

area\_dft = sum(abs(dft\_sig).^2)/length(signal);

round(area\_sig,10) == round(area\_dft, 10)

function y = dft(x)

y = zeros(size(x));

for i=0:(length(x)-1)

for j=0:(length(x)-1)

y(i+1) = y(i+1) + x(j+1)\*exp(-1i\*2\*pi/length(x)\*i\*j);

end

end

end

Twierdzenie Parsevala zostało udowodnione dla sygnału sinus, ponieważ powierzchnia pod funkcją jest równa powierzchni pod widmem DFT tej funkcji.

Obraz zawierający zrzut ekranu, biały

Opis wygenerowany automatycznie

Rysunek 3. Wynik potwierdzający teorię Parsevala.

c) Listing 5. Aproksymacja sygnału trójkątnego i sinusoidalnego za pomocą funkcji Walsha.

N = 1024;

hadamardMatrix = hadamard(N);

HadIdx = 0:N-1; % Hadamard index

M = log2(N)+1; % Number of bits to represent the index

binHadIdx = fliplr(dec2bin(HadIdx,M))-'0'; % Bit reversing of the binary index

binSeqIdx = zeros(N,M-1); % Pre-allocate memory

for k = M:-1:2

% Binary sequency index

binSeqIdx(:,k) = xor(binHadIdx(:,k),binHadIdx(:,k-1));

end

SeqIdx = binSeqIdx\*pow2((M-1:-1:0)'); % Binary to integer sequency index

walshMatrix = hadamardMatrix(SeqIdx+1,:); % 1-based indexing

L = 4;

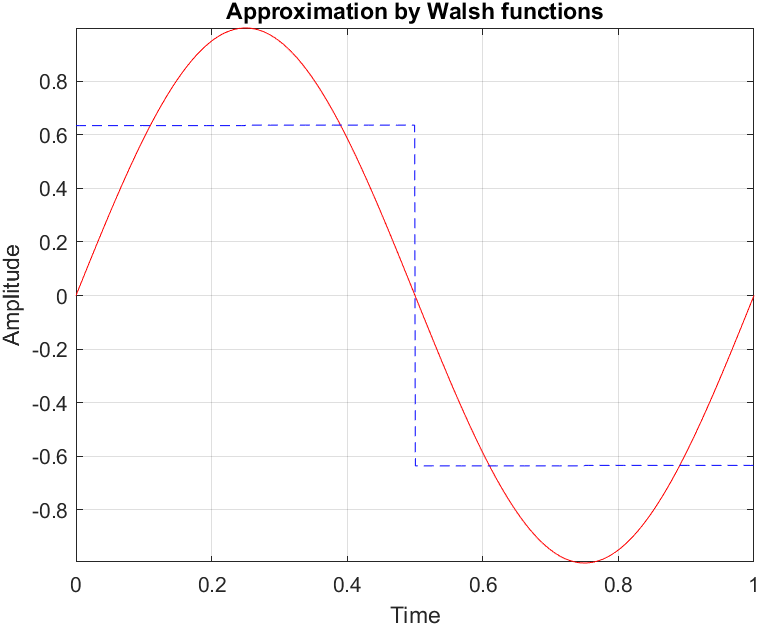
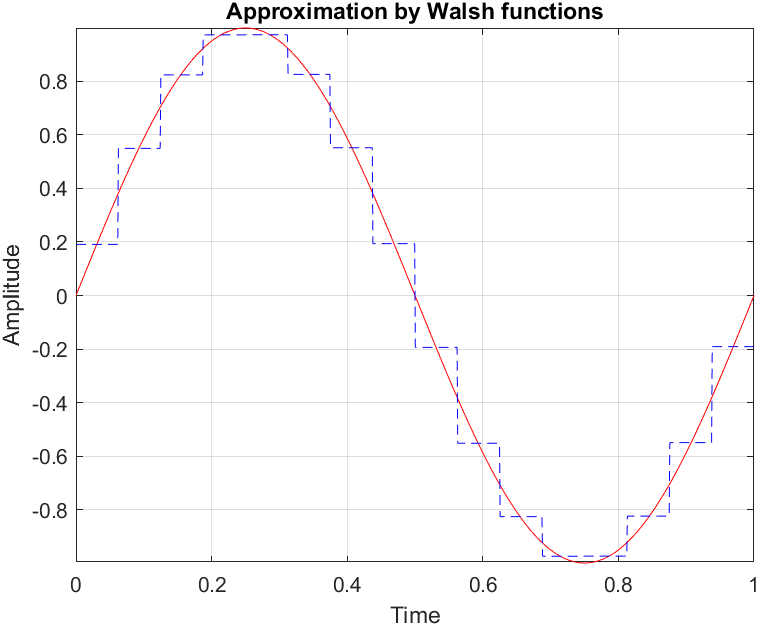
approximation = zeros(size(t));

for n = 1 : L

approximation = approximation + dot(signal, walshMatrix(:,n))\*walshMatrix(:,n);

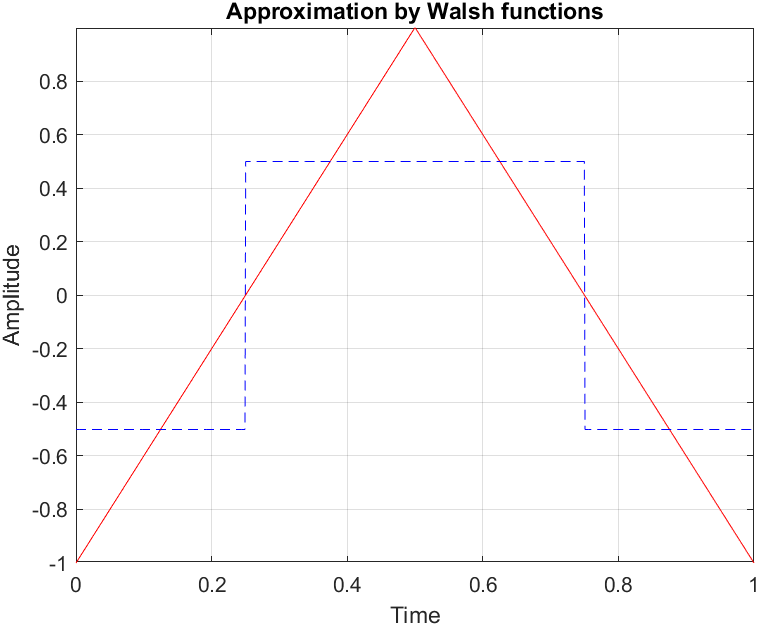
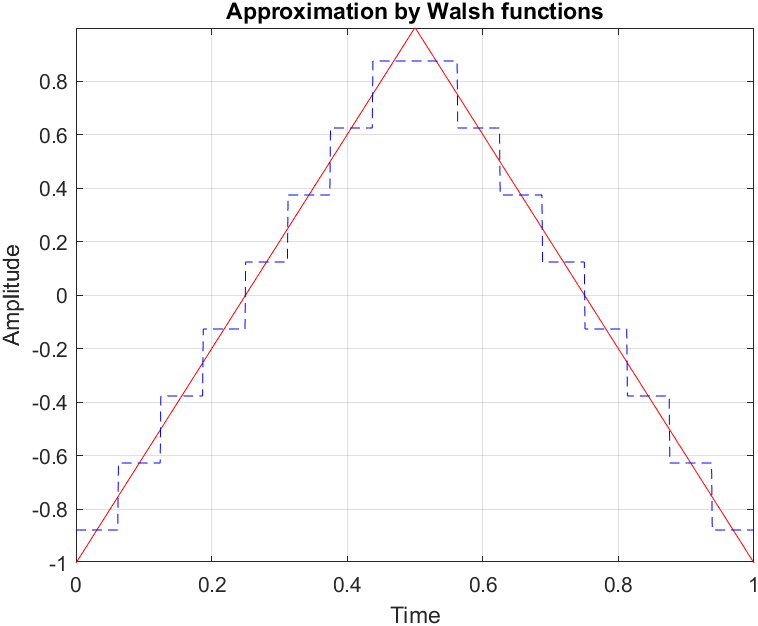
end

approximation = approximation/length(approximation);



Rysunek 5. Sygnał sinusa przybliżony 16 funkcjami Walsha.

Rysunek 4. Sygnał sinusa przybliżony 4 funkcjami Walsha.

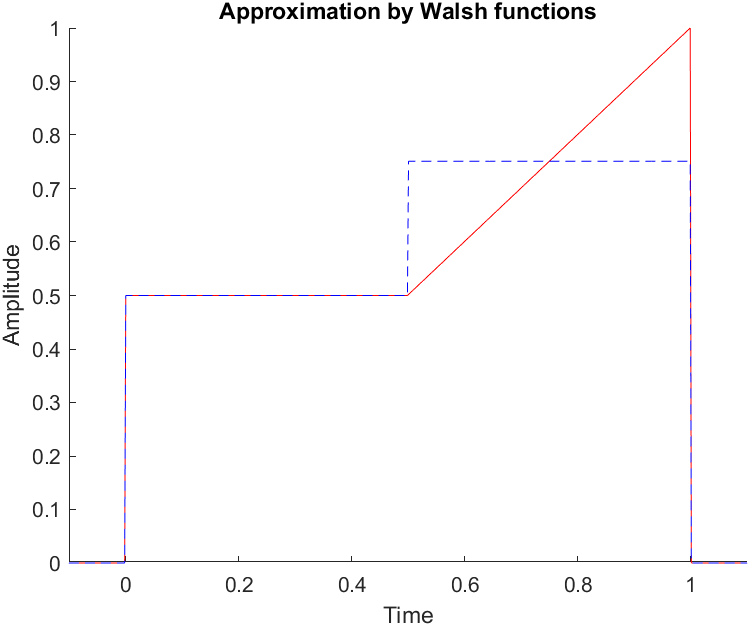


Rysunek 7. Sygnał trójkątny przybliżony 16 funkcjami Walsha.

Rysunek 6. Sygnał trójkątny przybliżony 4 funkcjami Walsha.

Sygnał jest znormalizowany poprzez podzielenie wyniku aproksymacji przez długość otrzymanego sygnału. Przybliżenie jest tym dokładniejsze, im więcej funkcji Walsha zostanie wykorzystane.

Zadanie 2

****

Rysunek 8. Aproksymacja podanego sygnału.

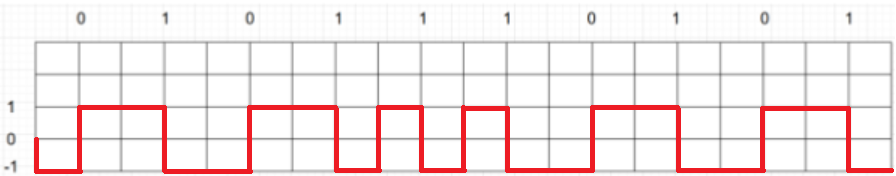
Do aproksymacji sygnału został wykorzystany kod z poprzedniego zadania ze zmienionym sygnałem wejściowym. Jak widać dla 4 pierwszych funkcji Walsha kawałek sygnału prostokątnego jest bardzo dobrze przybliżony, natomiast sygnał trójkątny nie. Można natomiast stwierdzić, że powierzchnia pod sygnałami jest taka sama.

**Kodowanie Manchester**

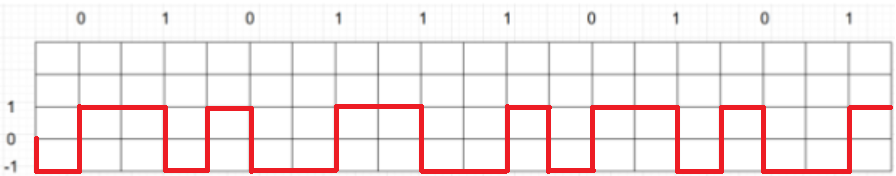
Zadanie 1Obraz zawierający linia, Wykres, zrzut ekranu, numer

Opis wygenerowany automatycznie

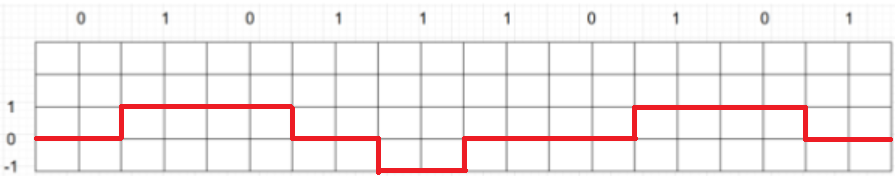
Rysunek 9. Wiadomość zakodowana za pomocą kodowania Manchester według standardu G. E. Thomas'a.



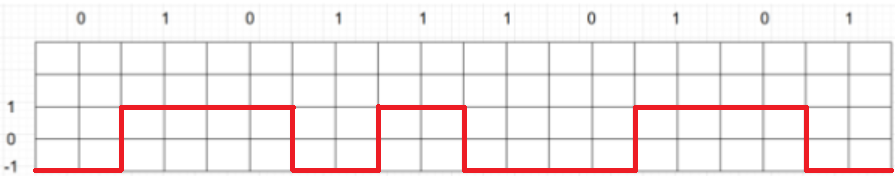
Rysunek 10. Wiadomość zakodowana za pomocą kodowania Manchester według standardu IEEE.



Rysunek 11. Wiadomość zakodowana za pomocą kodowania różnicowego Manchester.



Rysunek 12. Wiadomość zakodowana za pomocą kodowania MLT-3.



Rysunek 13. Wiadomość zakodowana za pomocą kodowania NRZI.

Zadanie 2

a) Rozkodowanie wiadomości za pomocą kodu Manchester (IEE):

1110101110

b) Rozkodowanie wiadomości za pomocą kodu Manchester różnicowy:

1000110101

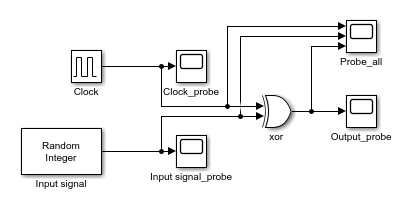
c) Rozkodowanie wiadomości za pomocą kodu MLT-3:

0010011001

d) Rozkodowanie wiadomości za pomocą kodu NRZI:

1110110101

Zadanie 3



Rysunek 14. Schemat, który koduje wiadomość kodem Manchester.

Obraz zawierający zrzut ekranu, Prostokąt, kwadrat, linia

Opis wygenerowany automatycznie

Rysunek 15. Wykresy przedstawiające zegar, dane oraz dane zakodowane.

Wiadomość została zakodowana kodem Manchester standardu IEEE.

***Wnioski***

Zadanie 1

W przypadku więcej niż dwóch wektorów są one ortogonalne, gdy każdy z nich jest ortogonalny z każdym.

Im więcej funkcji Walsha zastosujemy do aproksymacji sygnału, tym sygnał jest dokładniejszy. Dzieje się tak z powodu zwiększającej się przedziałów, na które podzielone jest t.

Zadanie 2

Macierze Haara i Walsha nie są ortogonalne, ponieważ wszystkie kolumny muszą być między sobą ortogonalne, co jest już niespełnione na pierwszych kolumnach.

**Kodowanie Manchester:**

Zadanie 1

W kodowaniu Manchester różnicowym głównym czynnikiem, na który powinno się zwracać uwagę jest jego zbocze, a nie poziom, przez co poziom przed nadaniem wiadomości jest nieistotny.

Zadanie 2

Wiadomość zakodowana jest dwa razy dłuższa od oryginalnego komunikatu, ponieważ reaguje na każde przejście ze stanu niskiego do wysokiego, lub z wysokiego do niskiego.