SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Wavelety a Banky filtrov

ZADANIE 1

Meno: Matúš Jakube

OIČ: 35411

Rok: 2012/2013

Text zadania:

1. Časť I – priestory a bázy

Majme

- 2 vektory ktoré formujú ortonormálnu bázu A , sú v pravom uhle a majú jednotkovú veľkosť pre každého študenta bude "báza" otočená o phi=10*cislo_studenta stupňov
- 2 vektory ktoré formujú bázu B , majú jednotkovú veľkosť pre každého študenta budú vektory v báze zvierať uhol phi=5*cislo_studenta stupňov
- signál x(I)= (2,2)

1.1. overte, že bázy môžu byť naozaj bázy (LNZ), či sú ortonormálne a prečo atď...
1.2. nakreslite bázy, použite "demo" (programček v adresári LIN_ALG), použite "axis equal"
1.3. zistite reprezentáciu x(A), x(B) a zrekonštruujte (graficky aj numericky)
1.4. zistite duálne bázy k A a B a nakreslite ich
1.5. zistite či platí parsevalova rovnosť, či veľkosť x(I) = veľkosť x(A) = veľkosť x (B)

2. Časť II - Spektrogram/škálogram

2.1. vytvorte si pílový signál s frekvenciou N khz v jednej časti a frekvenciou 2N khz v druhej časti
2.2. analyzujte ho pomocou škálogramu a spektrogramu, pričom sa snažte odhadnúť frekvenciu signálu v jednotlivých častiach a odhadnite v čase polohu bodu zmeny frakvencie

3. Časť III - Waveletove Rady

3.1. vypočítajte waveletove rady zo zadaného signálu (všetky nenulové koeficienty d_m,n)
3.2. pomocou matlabu urobte rekonštrukciu - ukážte ako postupným pribúdaním d_mn*psi_mn(t) sa aproximácia signálu vylepšuje. Signál trvá 4 sekundy a má jednosmernú zložku

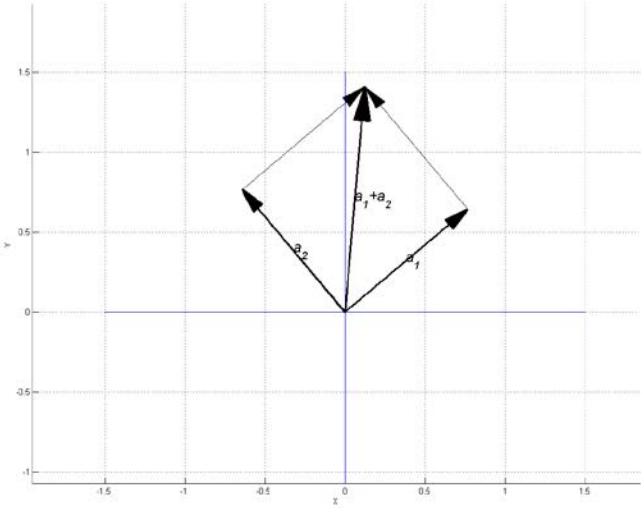
Obsah

1 Riešenie

1.1 Časť 1 – priestory a bázy

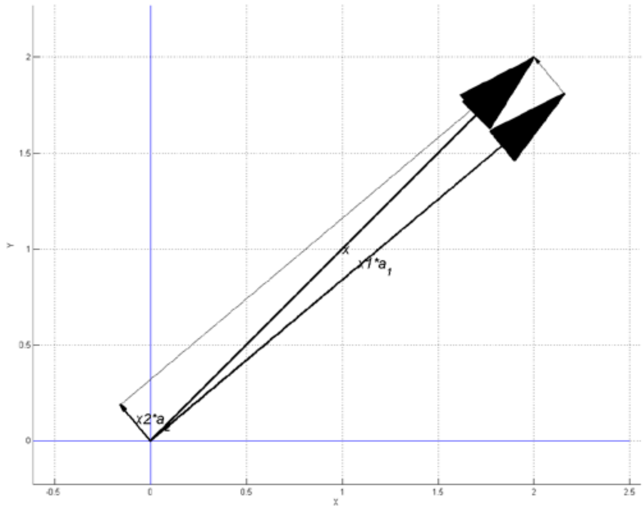
1.1.1 Ortonormálna báza

Mám zadané 2 vektory formujúce ortonormálnu bázu A. Tieto vektory majú jednotkovú veľkosť a spolu zvierajú pravý uhol. Z môjho zadania vyplýva, že báza je otočená o uhol 40° čo tvorí 2/9*π. Tiež mám zadaný vektor x(I)=(2,2).
V prvom kroku bolo potrebné si zistiť súradnice jednotlivých bázových vektorov a_1 a a_2. Podľa vzorcov **(1)** a **(2)**. Následne som vykonal kontrolu, či tieto vektory tvoria bázu. Kontrola spočíva overením lineárnej nezávislosti vektorov **(3)**. Z **Tab.1** je možné vidieť, že koeficienty k1 a k2 sú rozdielne, čo znamená, že vektory sú lineárne nezávislé. Ďalej som overil skutočnosť, že sa jedná o ortonormálnu bázu. Množina A = {a_i}, je ortonormálny systém v priestore E ak platí, že skalárny súčin vektorov $< a_1, a_2 > = 0$ a súčasne $< a_1, a_1 >= 1$, $< a_2, a_2 >= 1$. Na **Obr.1** je vidieť že súradnice vektorov zvierajú uhol 90° a zároveň, že súradnice vektorov boli vypočítané správne.



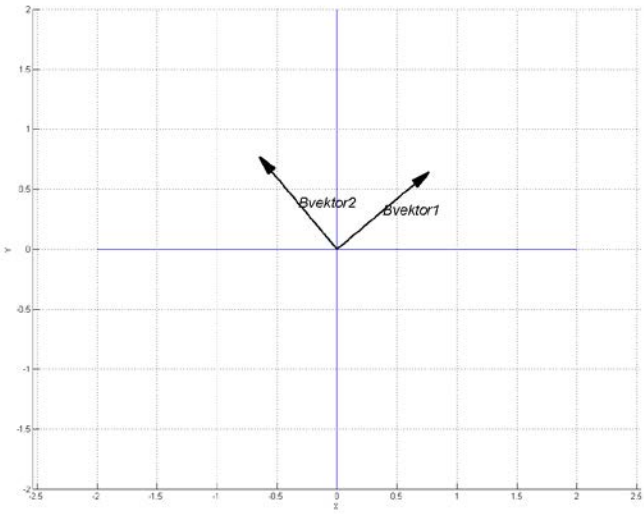
Obr. 1 bázové vektory

Ďalej bolo potrebné vypočítať súradnice zadaného vektora x(I) v báze A pomocou vzorcov **(6)**. Matica P_{IA} sa nazýva matica prechodu od bázy I k báze A, pričom I je jednotková matica a teda platí, že P_{IA}=A. Vlastnosť ortonormálnej báze je taká, že inverzná matica A^{-1} je rovnaká ako transponovaná matica A^T vďaka čomu sa výpočet zjednodušil. Správnosť výpočtu som si overil vynásobením matice A s maticou A^{-1} kde mi vyšla jednotková matica J. Po týchto krokoch som bol schopný zrekonštruovať pôvodný vektor x(I) graficky aj výpočtom ako je znázornené na **Obr.2**.



Obr. 2 Rekonštrukcia vektora x(I)

Nakoniec som vypočítal duálnu bázu $\tilde{A}$ **(7)**, ktorá je zobrazená na **Obr.3** a taktiež som zistil, že platí Parsevalova rovnosť **(8)**. Keďže sa jedná o ortonormálnu bázu, duálna báza je zhodná s bázou A.



Obr. 3 Duálna báza

Tab. 1 Výstup z matlabu pre Ortonormálnu bázu

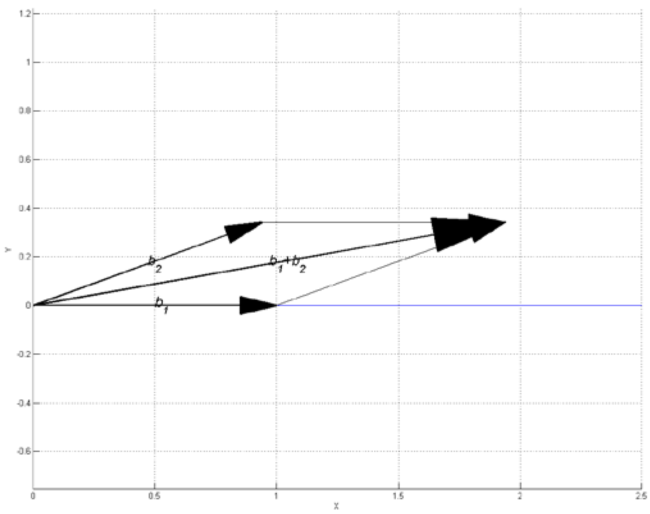
Výstup z Matlabu
a1 = 0.7660 0.6428
a2 = -0.6428 0.7660
k1 = -1.1918
k2 = 0.8391
SS1 = 0
SS2 = 0

```
SS3 =    1.0000
SS4 =    1.0000
J =    1.0000    0.0000
       0         1.0000
xA =    2.8177    0.2465
xR =    2.0000    2.0000
x1 =    2.8177
x2 =    0.2465
A_dualna =    0.7660   -0.6428
              0.6428    0.7660
dualb1 =    0.7660    0.6428
dualb2 =   -0.6428    0.7660
VxI =     8
VxA =    8
```

1.1.2 Neortonormálna báza

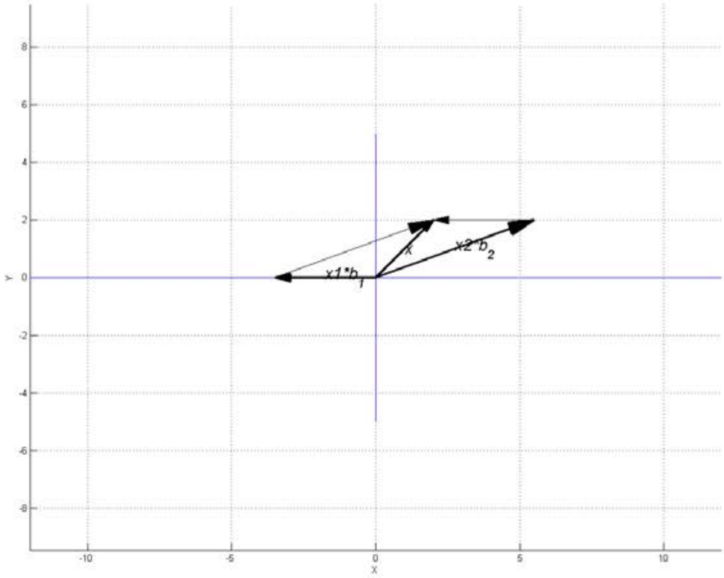
Zo zadania vyplýva, že máme 2 vektory, ktoré tvoria neortonormálnu bázu B a majú jednotkovú veľkosť. Vektor x(I)= (2,2) a vektory v báze zvierajú uhol 20° teda 1/9*π.

Tak ako v predchádzajúcom prípade pri ortonormálnej báze bolo najprv potrebné si vypočítať súradnice bázových vektorov b_1 a b_2. Vektor b_1 leží na osi X a má súradnice (1,0). Podľa vzorcov **(2)** som získal súradnice druhého vektora. Pomocou lineárnej nezávislosti vektorov na základe vzorca **(3)** som zistil, že vektory tvoria bázu. **Tab. 1** zobrazuje koeficienty k1 a k2 ktoré sú rozdielne čo znamená, že vektory tvoria skutočne bázu. Báza je neortonormálna, pretože $< b_1, b_2 > \neq 0$.



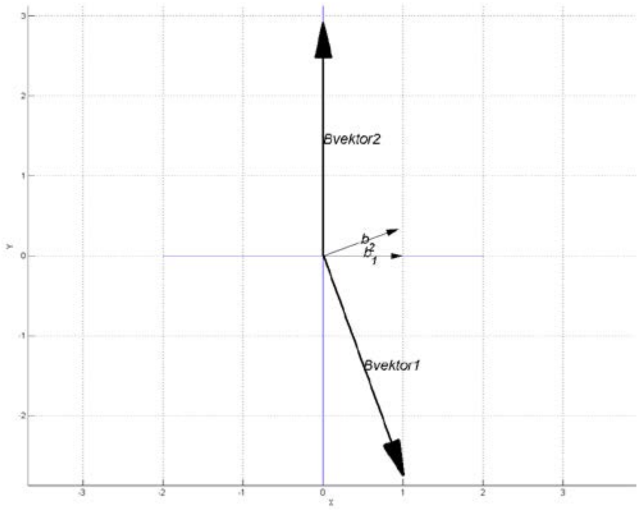
Obr. 4 Zobrazenie bázových vektorov

Ďalej som vypočítal súradnice vektora x(I) báze B pomocou vzorcov **(6)**. Taktiež platí, že P_{IB}=B. Pre výpočet inverznej matice v neortonormálnej báze je potrebné použiť vzorec **(9)**. Vektor x(I) som zrekonštruoval výpočtom a aj graficky na **Obr.5**.



Obr. 5 Rekonštrukcia vektora x(I)

V závere som pokračoval rovnako ako v predchádzajúcom prípade pri ortonormálnej báze. Vypočítal som duálnu bázu $\tilde{B}$ **(7)**, overil platnosť Parsevalovej rovnosti **(8)** a graficky som ju znázornil na **Obr.6**. Na základe skutočnosti, že táto báza je neortonormálna, Parsevalova rovnosť neplatí. **Obr.7** znázorňuje, že platí: $\widetilde{b_2} \perp b_1$ a tiež $\widetilde{b_1} \perp b_2$.



Obr. 6 Duálna a pôvodná báza

Tab. 2 Výstup z Matlabu pre Neortonormálnu bázu

Výstup z Matlabu

```
b1 =    1     0
b2 =    0.9397    0.3420
k1 =    1.0642
k2 =     0
SS1 =    0.9397
SS2 =    0.9397
B =    1.0000    0.9397
            0           0.3420
B1 =    1.0000   -2.7475
              0           2.9238
J =     1     0
           0     1
xB =   -3.4950    5.8476
xI =     2     2
x1 =   -3.4950
x2 =    5.8476
B_dualna =    1.0000         0
                      -2.7475         2.9238
dualb1 =    1.0000   -2.7475
dualb2 =         0    2.9238
VxI =     8
VxB =   46.4092
```

1.1.3 Rovnice

Súradnice vektorov:

$$\cos\alpha = \frac{a_{11}}{||1||} \qquad \sin\alpha = \frac{a_{12}}{||1||} \tag{1}$$

$$\cos\alpha = \frac{a_{22}}{||1||} \qquad -\sin\alpha = \frac{a_{21}}{||1||} \tag{2}$$

Lineárna nezávislosť:

$$a_1 = \beta \, . \, a_2 \tag{3}$$

Veľkosť vektora:

$$\left|\left|x\right|\right| = \sqrt{< x, x >} \tag{4}$$

Skalárny súčin:

$$< x, y > = \sum_i x_i \, y_i \qquad (5)$$

Reprezentácia x(A):

$$A = I.P_{IA}$$
$$\bar{x}(A) = P_{IA}^{-1}.\bar{x}(I)$$
$$\bar{x}(A) = A^{-1}.\bar{x}(I) \qquad (6)$$

Duálna báza:

$$\tilde{A} = A^{-1^{T}} \qquad (7)$$

Parsevalova rovnosť:

$$||x(I)|| = ||x(A)|| \qquad (8)$$

$$x(I).\bar{x}(I) = x(A).\bar{x}(A)$$

Inverzná matica pre neortonormálnu bázu:

$$B^{-1} = \frac{1}{det}.\begin{pmatrix} \searrow & +/- \\ +/- & \nwarrow \end{pmatrix} \qquad (9)$$

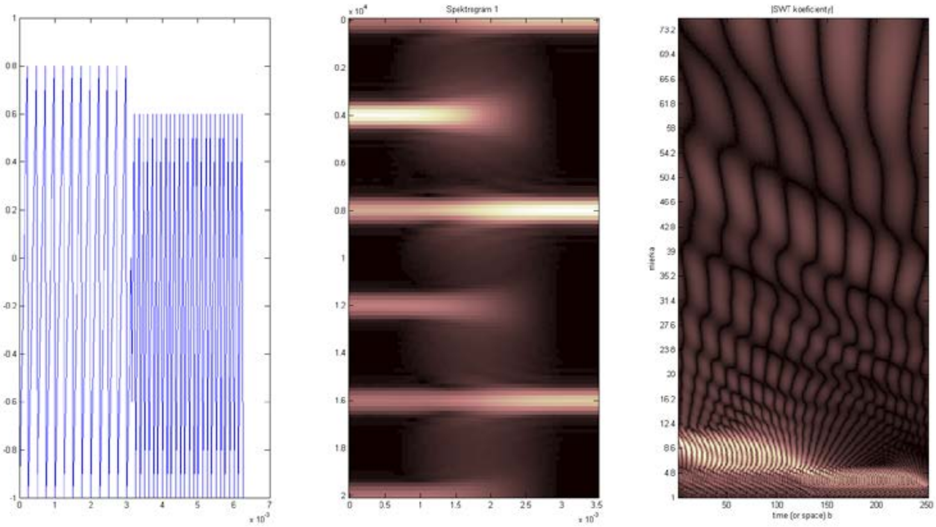
1.2 Časť 2 – škálogramy/spektrogramy

Najprv som si vykreslil pílovitý signál s frekvenciou 4kHz, ktorá sa po určitom čase mení na 8 kHz. Tento signál som navzorkoval 40 000 vzorkami. Následne som pristúpil k zobrazeniu spektrogramu, pri ktorom som použil Hanningovo okno o veľkosti 110 s prekryvom okien 108. Potom som si zobrazil škálogram za použitia waveletu „db44“.(**Obr.7**)

Po úspešnom zobrazení škálogramu a spektrogramu som z nich odhadol frekvenciu signálu v jednotlivých častiach. Na spektrograme vidieť, že frekvencie v prvej časti signálu vzrastajú po 4kHz a v druhej časti po 8kHz. S použitím príkazu *centfrq* som odhadol strednú frekvenciu na f_0=0.6437Hz.Na škálograme som potom odčítal hodnoty a_1=6,45 a a_2=3,25 (stredy elipsoidov) a vypočítal približné hodnoty frekvencii s použitím nasledujúceho vzorca:

$$f_1 = \frac{f_0}{a_1} * 40\,000 \simeq 3991{,}94Hz$$

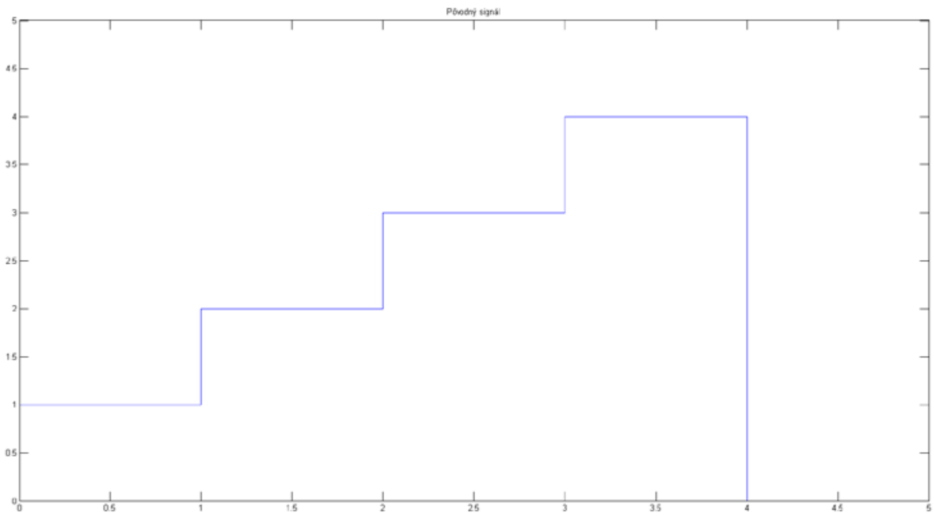
$$f_2 = \frac{f_0}{a_2} * 40\,000 \simeq 7922{,}5Hz$$



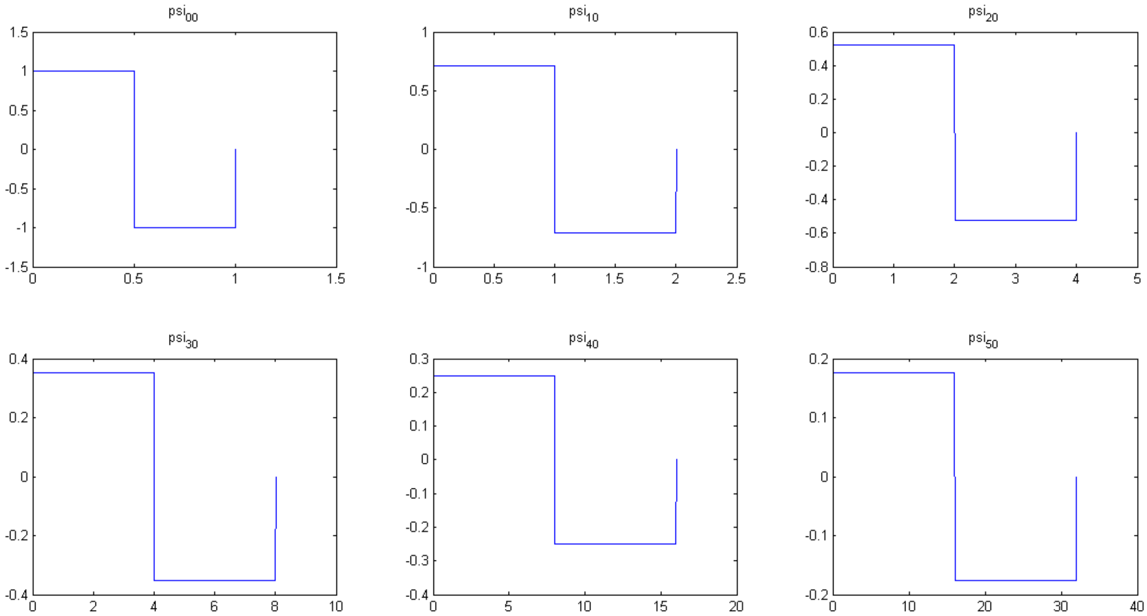
Obr. 7 Zadaný signál, spektrogram a škálogram

1.3 *Časť 3 – Waveletove Rady*

Ako prvé som si ručne vypočítal a nakreslil wavelety $\psi_{m,n}$(**Obr. 9**). Pre výpočet som použil vzorce **(10)**. Výsledné wavelety vznikli zmenou Haarovho wavletu, konkrétne jeho posunutím a zrýchlením, resp. spomalením. Počas porovnania môjho zadaného signálu **Obr.8** a zobrazených wavletov som si určil, ktoré koeficienty je potrebné počítať. Pri pohľade na zadaný signál je zrejmé, že signál obsahuje jednosmernú zložku. Z toho dôvodu má signál nekonečne veľa koeficientov $d_{m,n}$. Na základe týchto zistení som si počet nenulových koeficientov $d_{m,n}$ vypočítal po m=5 podľa vzorca **(11)**.



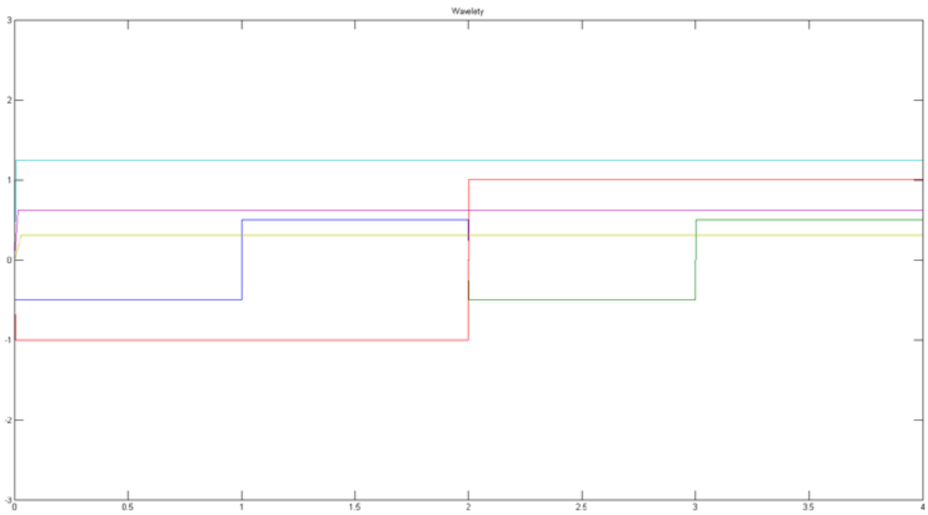
Obr. 8 Zadaný signál



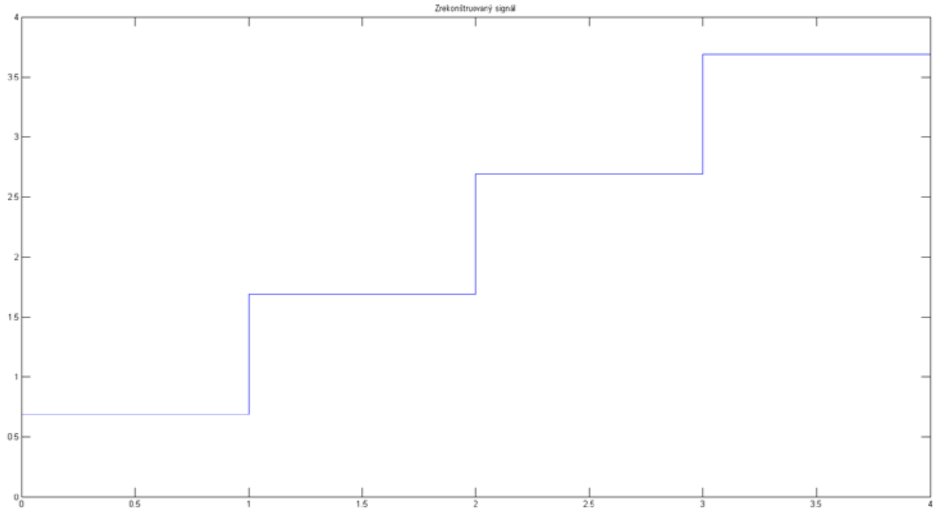
Obr. 9 Wavelety $\psi_{m,n}$

Ručne som na papieri vypočítal hodnoty: $d_{1,0} = -\frac{\sqrt{2}}{2}$, $d_{1,1} = -\frac{\sqrt{2}}{2}$, $d_{2,0} = -2$, $d_{3,0} = \frac{5\sqrt{2}}{2}$, $d_{4,0} = \frac{5}{2}$ a $d_{5,0} = \frac{5\sqrt{2}}{4}$. Následne som vypočítané koeficienty použil pri rekonštrukcii signálu pomocou vzorca **(12)**.

Obr.10 znázorňuje wavelety ktoré sú vynásobené koeficientami, ktoré som vypočítal. Na obrázku **Obr.11** som zobrazil výsledný zrekonštruovaný signál.



Obr. 10 Jednotlivé wavelety vynásobené príslušnými koeficientami



Obr. 11 Zrekonštruovaný signál

1.3.1 Rovnice

$$\psi_{m,n} = 2^{-\frac{m}{2}}.\psi(2^{-m}.t - n) \tag{10}$$

$$d_{m,n} = < f(t) ; \psi_{m,n} > = \int_{-\infty}^{\infty} f(t) . \psi_{m,n} (t) \, dt \tag{11}$$

$$f(t) = \sum_m \sum_n d_{m,n}.\psi_{m,n} \tag{12}$$

2 Záver

V prvej časti mojej práce som si pomocou programu matlab vypočítal hodnoty bázových vektorov a porovnal ich s hodnotami, ktoré som vypočítal na papieri. Keďže som sa počas výpočtu na papieri nezmýlil, hodnoty sa rovnali. Tiež som si pomocou programu skontroloval tvorbu duálnej bázy a zobrazenie vektora x(l) v zadanej báze.

Druhá časť, ktorá bola venovaná spektrogramom a škálogramom mi v programe graficky znázornila frekvencie môjho signálu. Taktiež pomocou týchto znázornení som mohol odhadnúť strednú frekvenciu signálu pomocou ktorej som vypočítal vzorkovaciu frekvenciu.

V poslednej časti som si ručne vypočítal waveletove rady zo zadaného signálu tj. všetky nenulové koeficienty $d_{m,n}$, avšak len po hodnotu $d_{5,0}$, pretože by ich bolo nekonečne veľa. Následne som spätne vykonal rekonštrukciu zadaného signálu. Pomocou využitia funkcii matlabu som si zrekonštruovaný signál vykreslil v matlabe.

3 Príloha: Zdrojové kódy pre Matlab

Zoznam modulov:

- plot_2d_axis.m – funkcia na kreslenie osi
- plot_2d_vector.m – funkcia na kreslenie vektora
- Ortonormalna_baza.m
- Neortonormalna_baza.m
- Spectrogram_skalogram.m
- Waveletove_rady.m

plot_2d_axis.m

```
function plot_2d_axis(minX, maxX, minY, maxY)
figure;
hold on;
axis([minX maxX minY maxY]);
grid on;
axis equal;
xlabel('X');
ylabel('Y');
plot([minX,maxX],[0,0]);
plot([0,0],[minY,maxY]);
```

plot_2d_vector.m

```
function handles = plot_2d_vector( origin,myVector,name,varargin )
x1=origin(1);
y1=origin(2);
x2=myVector(1);
y2=myVector(2);
text((x2-x1)/2,(y2-y1)/2,['\it', name],'HorizontalAlignment','left','FontSize',18);
%
% plot_2d_vector - plots an arrow to the current plot
%
% format:   handles = plot_2d_vector( x1,y1,x2,y2 [,options...] )
%
% input:    x1,y1   - starting point
%           x2,y2   - end point
%         options - come as pairs of "property","value" as defined for "line" and "patch"
%         controls, see matlab help for listing of these properties.
%      note that not all properties where added, one might add them at the end of this file.
%
%            additional options are:
%            'headwidth':  relative to complete arrow size, default value is 0.07
%            'headheight': relative to complete arrow size, default value is 0.15
%            (encoded are maximal values if pixels, for the case that the arrow is very long)

% output:   handles - handles of the graphical elements building the arrow
%
```

```
% Example:  plot_2d_vector( origin, vector,'linewidth',2,'color',[0.5 0.5 0.5], 'facecolor',
[0.5 0.5 0.5] );
%           plot_2d_vector( origin, vector,'linewidth',2,'headwidth',0.25,'headheight',0.33 );
% =============================================
% constants (can be edited)
% =============================================
alpha       = 0.15;   % head length
beta        = 0.07;   % head width
max_length  = 22;
max_width   = 10;

% =============================================
% check if head properties are given
% =============================================
% if ratio is always fixed, this section can be removed!
if ~isempty( varargin )
    for c = 1:floor(length(varargin)/2)
        try
            switch lower(varargin{c*2-1})
                % head properties - do nothing, since handled above already
            case 'headheight',alpha = max( min( varargin{c*2},1 ),0.01 );
            case 'headwidth', beta = max( min( varargin{c*2},1 ),0.01 );
            end
        catch
            fprintf( 'unrecognized property or value for: %s\n',varargin{c*2-1} );
        end
    end
end

% =============================================
% calculate the arrow head coordinates
% =============================================
den         = x2 - x1 + eps;                          % make sure no devision by zero occurs
teta        = atan( (y2-y1)/den ) + pi*(x2<x1) - pi/2;      % angle of arrow
cs          = cos(teta);                             % rotation matrix
ss          = sin(teta);
R           = [cs -ss;ss cs];
line_length = sqrt( (y2-y1)^2 + (x2-x1)^2 );               % sizes
head_length = min( line_length*alpha,max_length );
head_width  = min( line_length*beta,max_length );
x0          = x2*cs + y2*ss;                          % build head coordinats
y0          = -x2*ss + y2*cs;
coords      = R*[x0 x0+head_width/2 x0-head_width/2; y0 y0-head_length y0-
head_length];
% =============================================
% plot arrow  (= line + patch of a triangle)
% =============================================
h1          = plot( [x1,x2],[y1,y2],'k' );
h2          = patch( coords(1,:),coords(2,:),[0 0 0] );
```

```
% ==============================================
% return handles
% ==============================================
handles = [h1 h2];

% ==============================================
% check if styling is required
% ==============================================
% if no styling, this section can be removed!
if ~isempty( varargin )
    for c = 1:floor(length(varargin)/2)
        try
            switch lower(varargin{c*2-1})

             % only patch properties
            case 'edgecolor',   set( h2,'EdgeColor',varargin{c*2} );
            case 'facecolor',   set( h2,'FaceColor',varargin{c*2} );
            case 'facelighting',set( h2,'FaceLighting',varargin{c*2} );
            case 'edgelighting',set( h2,'EdgeLighting',varargin{c*2} );

            % only line properties
            case 'color'    , set( h1,'Color',varargin{c*2} );

            % shared properties
            case 'linestyle', set( handles,'LineStyle',varargin{c*2} );
            case 'linewidth', set( handles,'LineWidth',varargin{c*2} );
            case 'parent',    set( handles,'parent',varargin{c*2} );

            % head properties - do nothing, since handled above already
            case 'headwidth',;
            case 'headheight',;

            end
        catch
            fprintf( 'unrecognized property or value for: %s\n',varargin{c*2-1} );
        end
    end
end
```

Ortonormalna_baza.m

```
close all; clear all;
clc;
alfa=2/9*pi;
a11=cos(alfa); a12=sin(alfa);
a21=-sin(alfa); a22=cos(alfa);
a1=[a11 a12]
a2=[a21 a22]
xI=[2 2];
```

```
%a) overenie
% že sú to bázy (LNZ)
k1=a11/a21              %koeficienty k1 a k2 sa nesmú rovnať
k2=a12/a22
% že sú ortonormálne (SS-skalárny súčin)
SS1=a1*a2', SS2=a2*a1' %SS musí byť 0
SS3=a1*a1', SS4=a2*a2' %SS musí byť 1

%b)vykreslenie vektorov v 2D
plot_2d_axis(-1.5,1.5,-1.5,1.5);
lnw = 2.3;
orig = [0,0];
plot_2d_vector(orig, a1,'a_1', 'linewidth',lnw);
plot_2d_vector(orig, a2,'a_2', 'linewidth',lnw);
plot_2d_vector(a1, a1+a2,'', 'linewidth',1);
plot_2d_vector(a2, a1+a2,'', 'linewidth',1);
plot_2d_vector(orig, a1+a2,'a_1+a_2', 'linewidth',lnw);

%c) reprezentácia x pomocou A
A=[a11 a21; a12 a22];
A1=A^(-1);
J=A1*A      %kontrola správnosti A1-> J musí byť jednotková matica
xA=(A1*xI')'
% rekonštrukcia
xR=(A*xA')'
x1=xA(:,1)  %prvy stlpec matice xA
x2=xA(:,2)  %druhy stlpec matice xA
% graficky
plot_2d_axis(-2.5,2.5,-2.5,2.5);
lnw = 2.3;
orig = [0,0];
plot_2d_vector(orig, x1*a1,'x1*a_1', 'linewidth',lnw);
plot_2d_vector(orig, x2*a2,'x2*a_2', 'linewidth',lnw);
plot_2d_vector(x1*a1, xR,'', 'linewidth',1);
plot_2d_vector(x2*a2, xR,'', 'linewidth',1);
plot_2d_vector(orig, xR,'x', 'linewidth',lnw);

%d) duálna báza
A_dualna=transpose(A1) %pri ortonorm. báze sa báza zhoduje s duálnou bázou
%graficky
dualb1=A_dualna(:,1)'
dualb2=A_dualna(:,2)'
plot_2d_axis(-2,2,-2,2);
lnw = 2.3;
orig = [0,0];
plot_2d_vector(orig, dualb1,'Bvektor1', 'linewidth',lnw);
plot_2d_vector(orig, dualb2,'Bvektor2', 'linewidth',lnw);
%e) Parsevalova rovnosť
VxI=xI*xI'
VxA=xA*xA'
```

Neortonormalna_baza.m

```
clc;
close all;
clear all;
phi=1/9*pi;
b11=1; b12=0;
b21=cos(phi); b22=sin(phi);
b1=[b11 b12]
b2=[b21 b22]
xI=[2 2];

%a) overenie
% že sú to bázy (LNZ)
k1=b11/b21              %koeficienty k1 a k2 sa nesmú rovnať
k2=b12/b22
% že sú neortonormálne
SS1=b1*b2', SS2=b2*b1' %SS nesmie byť 0

%b) vykreslenie vektorov v 2D
plot_2d_axis(0,2.5,0,0.5);
lnw = 2;
orig = [0,0];
plot_2d_vector(orig, b1,'b_1', 'linewidth',lnw);
plot_2d_vector(orig, b2,'b_2', 'linewidth',lnw);
plot_2d_vector(b1,b1+b2,'', 'linewidth',1);
plot_2d_vector(b2, b1+b2,'', 'linewidth',1);
plot_2d_vector(orig, b1+b2,'b_1+b_2', 'linewidth',lnw);

%c) reprezentácia x pomocou B
B=[b11 b21; b12 b22]
B1=B^(-1)
J=B1*B      %kontrola správnosti B1
xB=(B1*xI')'
% rekonštrukcia
xI=(B*xB')'
x1=xB(:,1)  %prvy stlpec matice xB
x2=xB(:,2)  %druhy stlpec matice xB
% graficky
plot_2d_axis(-12,12,-5,5);
lnw = 2.3;
orig = [0,0];
plot_2d_vector(orig, x1*b1,'x1*b_1', 'linewidth',lnw);
plot_2d_vector(orig, x2*b2,'x2*b_2', 'linewidth',lnw);
plot_2d_vector(x1*b1, xI,'', 'linewidth',1);
plot_2d_vector(x2*b2, xI,'', 'linewidth',1);
plot_2d_vector(orig, xI,'x', 'linewidth',lnw);

%d) duálna báza
B_dualna=transpose(B1)
```

```
%graficky
dualb1=B_dualna(:,1)'
dualb2=B_dualna(:,2)'
plot_2d_axis(-2,2,-6,6);
lnw = 2.3;
orig = [0,0];
plot_2d_vector(orig, dualb1,'Bvektor1', 'linewidth',lnw);
plot_2d_vector(orig, dualb2,'Bvektor2', 'linewidth',lnw);
plot_2d_vector(orig, b1,'b_1', 'linewidth',1);
plot_2d_vector(orig, b2,'b_2', 'linewidth',1);

%e) Parsevalova rovnosť pri neortonorm. neplatí
VxI=xI*xI'
VxB=xB*xB'
```

Spektrogram_skalogram.m

```
clear all, close all, clc
f1 = 4000;
T1 = 1/f1;
dt = T1/10;
f2 = 8000;
T2 = 1/f2;
tmax = 50*T2;
t = 0:dt:tmax;
x = sawtooth(2*pi*f1*t).*(t<=(tmax)/2)+sawtooth(2*pi*f2*t).*(t>(tmax/2));
figure, subplot(131), plot(t,x)
[S,F,T]=specgram(x,250,40000,hanning(110),108);
subplot(132), imagesc(T,F,abs(S)), title('Spektrogram 1');
subplot(133)
ccfs = cwt(x,1:0.1:75,'db44','plot' );
title('|SWT koeficienty|')
ylabel('mierka')
centfrq('db44')
```

Waveletove_rady.m

```
clear all, close all, clc
[~,psi,xval] = wavefun('haar',10);
d10 = -sqrt(2)/2;
d11 = -sqrt(2)/2;
d20 = -2;
d30 = 5*sqrt(2)/2;
d40 = 5/2;
d50 = 5*sqrt(2)/4;
psi10 = sqrt(2)/2*psi;
xval10 = 2*xval;
psi11 = sqrt(2)/2*psi;
xval11 = 2*xval+2;
psi20 = 1/2*psi;
xval20 = 4*xval;
psi30 = sqrt(2)/4*psi;
```

```
xval30 = 8*xval;
psi40 = 1/4*psi;
xval40 = 16*xval;
psi50 = sqrt(2)/8*psi;
xval50 = 32*xval;
plot(xval10,d10*psi10,xval11,d11*psi11,xval20,d20*psi20,xval30,d30*psi30,xval40,d40*
psi40,xval50,d50*psi50); axis([0 4 -3 3]);
title('Wavelety');
T = [0 0 1 1 2 2 3 3 4 4];
F = [0 0.6875 0.6875 1.6875 1.6875 2.6875 2.6875 3.6875 3.6875 0];
figure
S = stairs(T,F)
title('Zrekonštruovaný signál')
```