۱) می خواهیم ضرایب فیلترهای h و g را برای N=۸ به دست آوریم. برای این منظور داریم:

$$K=\frac{N}{۲}=۴ \quad \rightarrow \quad p(y)=\sum_{k=۰}^{\frac{N}{۲}-۱}\binom{\frac{N}{۲}-۱+k}{k}y^k=\sum_{k=۰}^{۳}\binom{۳+k}{k}y^k=۱+۴y+۱۰y^۲+۲۰y^۳$$

$$|Q(\omega)|^۲=p(y) \quad \xrightarrow[\text{حوزه Z}]{} \quad Q(z)Q(z^{-۱})=p(y(z))$$

از طرفی طبق تعریفی که داشتیم می دانیم:

$$y=sin^۲\left(\frac{\omega}{۲}\right)=\frac{۱-cos\omega}{۲}=\frac{۱-\frac{e^{i\omega}+e^{-i\omega}}{۲}}{۲}=\frac{۲-e^{i\omega}-e^{-i\omega}}{۴} \quad \rightarrow \quad y(z)=\frac{۲-z-z^{-۱}}{۴}$$

$$\rightarrow \quad Q(z)Q(z^{-۱})=p\left(\frac{۲-z-z^{-۱}}{۴}\right)=۱+۴\left(\frac{۲-z-z^{-۱}}{۴}\right)+۱۰\left(\frac{۲-z-z^{-۱}}{۴}\right)^۲+۲۰\left(\frac{۲-z-z^{-۱}}{۴}\right)^۳$$

پس از ساده سازی، عبارت بالا، بدین شکل در می آید:

$$Q(z)Q(z^{-۱})=\frac{-۵z^۶+۴۰z^۵-۱۳۱z^۴+۲۰۸z^۳-۱۳۱z^۲+۴۰z-۵}{۱۶z^۳}$$

باید ریشه های عبارت بالا را محاسبه نماییم:

$$\rightarrow \quad -۵z^۶+۴۰z^۵-۱۳۱z^۴+۲۰۸z^۳-۱۳۱z^۲+۴۰z-۵=۰$$

که پس از حل، ریشه ها به صورت زیر می باشند:

۳.۰۴۰۷ + ۰.۰۰۰۰i

۲.۰۳۱۱ + ۱.۷۳۹۰i

۲.۰۳۱۱ − ۱.۷۳۹۰i

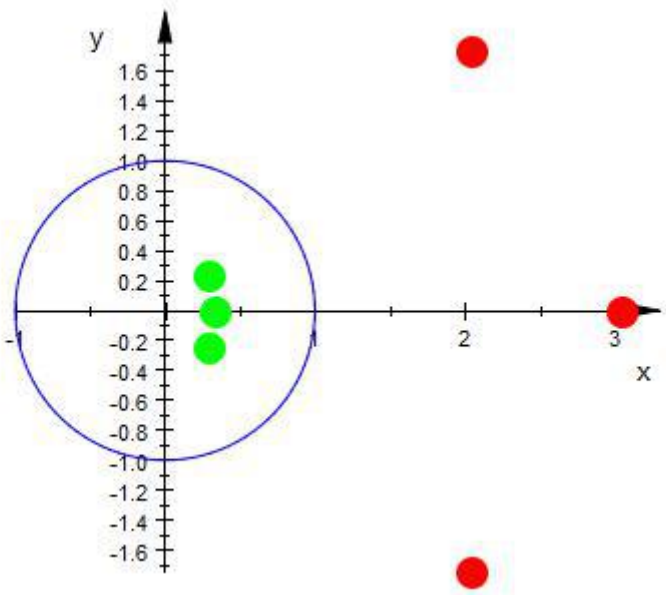
۰.۲۸۴۱ + ۰.۲۴۳۲i

۰.۲۸۴۱ − ۰.۲۴۳۲i

۰.۳۲۸۹ + ۰.۰۰۰۰i

حال باید بخشی از این ریشه ها را برای $Q(z)$ در نظر بگیریم. برای این منظور، از میان این ۶ ریشه که بخشی داخل دایره واحد و بخشی خارج آن، برای $Q(z)$، مجموعه ای از فاکتورها را انتخاب میکنیم که داخل دایره واحد باشند. این کار فیلتر $H(z)$ را می نیمم فاز میکند که در این صورت، حداکثر انرژی ضرایب فیلتر در حداقل تعداد ضرایب متمرکز می شوند. این کار عملیات پیاده سازی را بسیار راحت تر و کم هزینه تر می نماید.

برای اینکه موقعیت ریشه ها را نسبت به دایره واحد، بهتر ببینیم به شکل زیر توجه نمایید:



به وضوح مشخص است که ۳ ریشه در داخل دایره واحد قرار می گیرند. این سه ریشه را برای $Q(z)$ در نظر می گیریم.
بنابراین $Q(z)$ به فرم زیر در خواهد آمد:

$$Q(z) = \left(\left(۰.۲۸۴۱ \; + \; ۰.۲۴۳۲\mathrm{i}\right) z^{-۱} - ۱\right)\left(\left(۰.۲۸۴۱ \; - \; ۰.۲۴۳۲\mathrm{i}\right) z^{-۱} - ۱\right)(۰.۳۲۸۹\, z^{-۱} - ۱)$$

از طرفی می دانیم که پس از محاسبه ی $Q(z)$ با توجه به رابطه ی $H(z) = \alpha(\frac{۱+z^{-۱}}{۲})^{K} Q(z)$ به راحتی می توان $H(z)$ و در نتیجه ضرایب آن را به دست آورد. داریم:

$$H(z) = \alpha\left(\frac{۱ + z^{-۱}}{۲}\right)^{۴} Q(z)$$

$$\rightarrow H(z) = \alpha(\frac{۱ + z^{-۱}}{۲})^{۴}\left(\left(۰.۲۸۴۱ \; + \; ۰.۲۴۳۲\mathrm{i}\right) z^{-۱} - ۱\right)\left(\left(۰.۲۸۴۱ \; - \; ۰.۲۴۳۲\mathrm{i}\right) z^{-۱} - ۱\right)(۰.۳۲۸۹\, z^{-۱} - ۱)$$

برای محاسبه مقدار α ، با توجه به اینکه فیلتر پایین گذر بوده و در $\omega = ۰$ $(z = ۱)$ مقدار $\sqrt{۲}$ دارد، با جایگزین کردن این مقدار می توان به α رسید. بنابراین:

$$H(۱) = \sqrt{۲}$$

$$\rightarrow \sqrt{۲} = \alpha\left(\left(۰.۲۸۴۱ + ۰.۲۴۳۲\mathrm{i}\right) - ۱\right)\left(\left(۰.۲۸۴۱ - ۰.۲۴۳۲\mathrm{i}\right) - ۱\right)(۰.۳۲۸۹ - ۱) \rightarrow \quad \boxed{\alpha = -۳.۶۸۶}$$

پس در نهایت، H در حوزه ی Z به طور کامل به دست آمد:

$$H(z) = -۳.۶۸۶\,(\frac{۱ + z^{-۱}}{۲})^{۴}\left(\left(۰.۲۸۴۱ + ۰.۲۴۳۲\mathrm{i}\right)z^{-۱} - ۱\right)\left(\left(۰.۲۸۴۱ - ۰.۲۴۳۲\mathrm{i}\right)z^{-۱} - ۱\right)(۰.۳۲۸۹\,z^{-۱} - ۱)$$

پس از ساده سازی :

$$\mathrm{H(z)} = ۰.۲۳۰۳۹۴ + ۰.۷۱۴۸۹\,z^{-۱} + ۰.۶۳۰۸۹۷\,z^{-۲} - ۰.۰۲۸۰۲۵\,z^{-۳} - ۰.۱۸۷۰۷\,z^{-۴}$$
$$+۰.۰۳۰۸۴۱\,z^{-۵} + ۰.۰۳۲۸۸۶\,z^{-۶} - ۰.۰۱۰۵۹۸\,z^{-۷}$$

و بنابراین ضرایب فیلتر در حوزه زمان بدین شکل است:

$$\rightarrow h_{۰} = ۰.۲۳۰۳۹۴ \;\;,\;\; h_{۱} = ۰.۷۱۴۸۹ \;\;,\;\; h_{۲} = ۰.۶۳۰۸۹۷ \;\;,\;\; h_{۳} = -۰.۰۲۸۰۲۵ \;\;,$$
$$h_{۴} = -۰.۱۸۷۰۷ \quad h_{۵} = ۰.۰۳۰۸۴۱ \;\;,\;\; h_{۶} = ۰.۰۳۲۸۸۶\,,\;\; h_{۷} = -۰.۰۱۰۵۹۸$$

اکنون می توان با توجه به حقیقی بودن ضرایب و با توجه به رابطه ی :

$$g(n) = (-۱)^{n}\,h^{*}(N - ۱ - n) = (-۱)^{n}\,h(۷ - n)$$

ضرایب فیلتر g را نیز به راحتی به دست آورد.

$$g_{۰} = h_{۷} = -۰.۰۱۰۵۹۸$$
$$g_{۱} = -h_{۶} = -۰.۰۳۲۸۸۶$$
$$g_{۲} = h_{۵} = ۰.۰۳۰۸۴۱$$
$$g_{۳} = -h_{۴} = ۰.۱۸۷۰۷$$
$$g_{۴} = h_{۳} = -۰.۰۲۸۰۲۵$$
$$g_{۵} = -h_{۲} = -۰.۶۳۰۸۹۷$$
$$g_{۶} = h_{۱} = ۰.۷۱۴۸۹$$
$$g_{۷} = -h_{۰} = -۰.۲۳۰۳۹۴$$

در ادامه می خواهیم ضرایبی که در بالا به صورت محاسباتی به دست آوردیم، از طریق کد زیر توسط برنامه متلب رسم نموده و با نتایج خود مقایسه نماییم. کد مربوطه در ادامه آورده شده است:

```
clc;
clear;
close all;

%%% calculate coeficients of filters

[h g] = wfilters('db٤','r');
figure;
subplot(٢١١); stem(h);  title('coefficents of h filter (db٤)')
subplot(٢١٢); stem(g);  title('coefficents of g filter (db٤)')
```

پس از اجرای برنامه، ضرایب h و g بدین شکل به دست می آیند:

$h =$

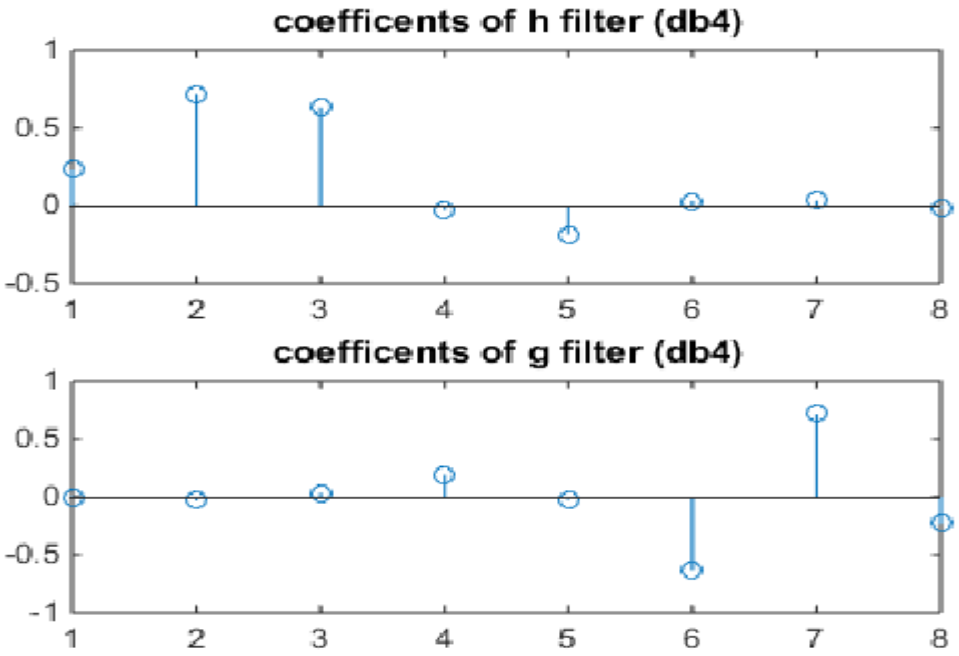
٠.٢٣٠۴ ٠.٧١۴٨ ٠.۶٣٠٩ − ٠.٠٢٨٠ − ٠.١٨٧٠ ٠.٠٣٠٨ ٠.٠٣٢٩ − ٠.٠١٠۶

$g =$

−٠.٠١٠۶ − ٠.٠٣٢٩ ٠.٠٣٠٨ ٠.١٨٧٠ − ٠.٠٢٨٠ − ٠.۶٣٠٩ ٠.٧١۴٨ − ٠.٢٣٠۴

که مشاهده می شود، با ضرایب به دست آمده از طریق محاسبات، برابر است. بناب کاملا یکسان است . بنابراین نتایج محاسبات و شبیه سازی کاملا یکسان می باشد.

می توان این ضرایب را روی یک نمودار نیز نمایش داد که در ادامه نمایش داده شده است:

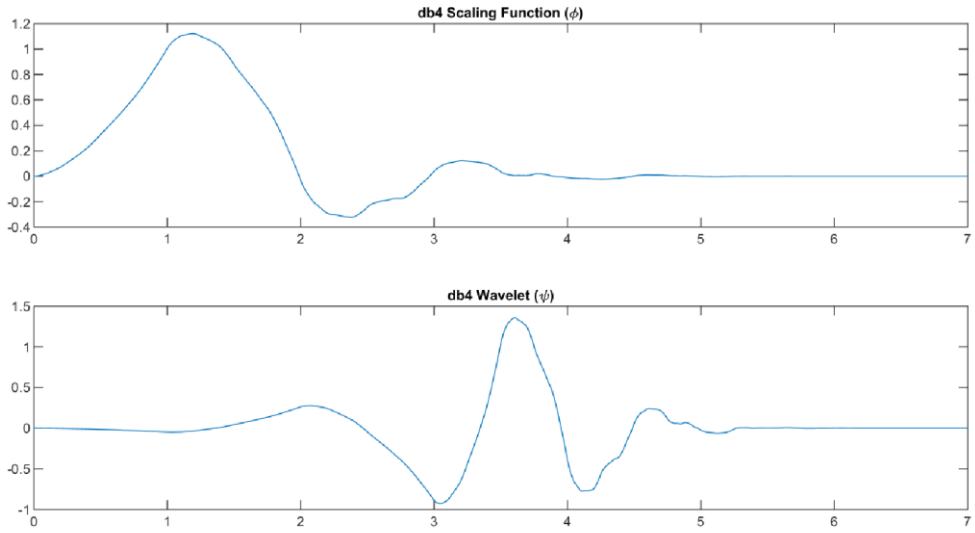


با توجه به ضرایب به دست آمده برای فیلترها، شکل موجهای $\varphi(x)$ و $\psi(x)$ را از طریق کد زیر، رسم می نماییم:

```
%%% plot scaling function and wavelet

[phi,psi,xval] = wavefun('db٤',١٠); %% for ١٠ iteration
figure;
subplot(٢١١);
plot(xval,phi);
title('db٤ Scaling Function (\phi)');
subplot(٢١٢);
plot(xval,psi);
title('db٤ Wavelet (\psi)');
```

که در ادامه رسم شده اند:



حال می خواهیم تحقیق کنیم که آیا روابط $\sum h(n) = \sqrt{۲}$ و $\sum_k h(n)h(n-۲k) = \delta_k$ درست می باشند؟ داریم:

$$\sum h(n) = ۰.۲۳۰۴ + ۰.۷۱۴۸ + ۰.۶۳۰۹ - ۰.۰۲۸۰ - ۰.۱۸۷۰ + ۰.۰۳۰۸ + ۰.۰۳۲۹ - ۰.۰۱۰۶ = ۱.۴۱۴$$

پس این رابطه صحیح می باشد.

برای رابطه بعدی داریم:

$$k = ۰ \rightarrow \sum_k h(n)h(n-۲k) = \sum \left(h(n)\right)^۲$$

$$= ۰.۲۳۰۴^۲ + ۰.۷۱۴۸^۲ + ۰.۶۳۰۹^۲ + ۰.۰۲۸۰^۲ + ۰.۱۸۷۰^۲ + ۰.۰۳۰۸^۲ + ۰.۰۳۲۹^۲ + ۰.۰۱۰۶^۲ = ۱$$

در نتیجه به ازای k=۰ این رابطه برقرار است.

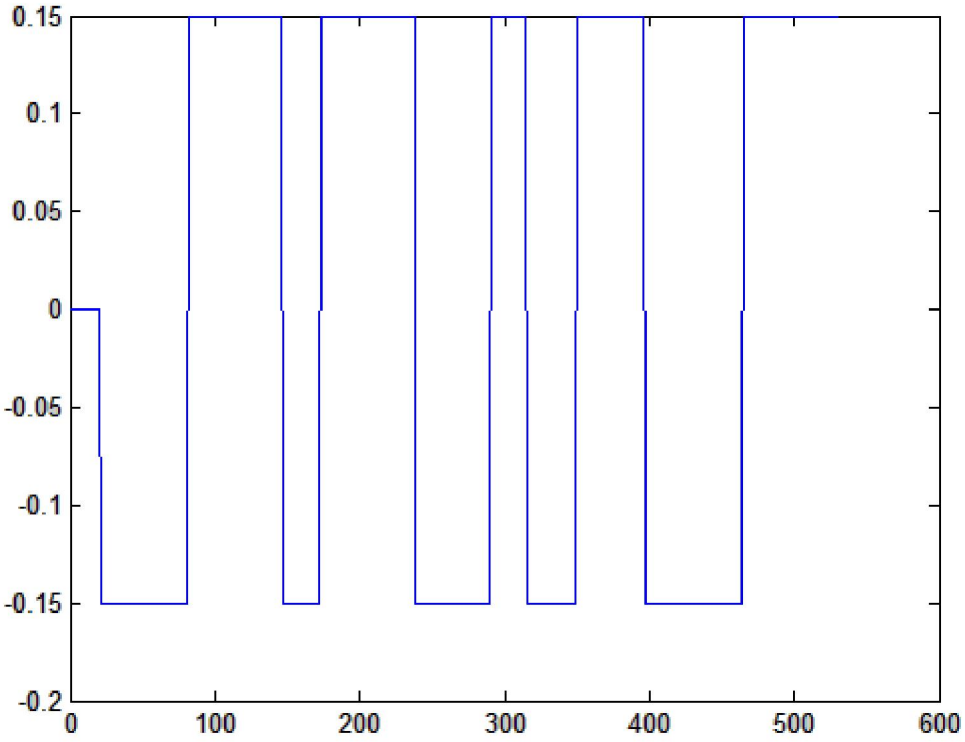
$$k \neq ۰ \rightarrow \sum_k h(n)h(n-۲k) = \sum_k h(n)h(n-۸) = ۰$$

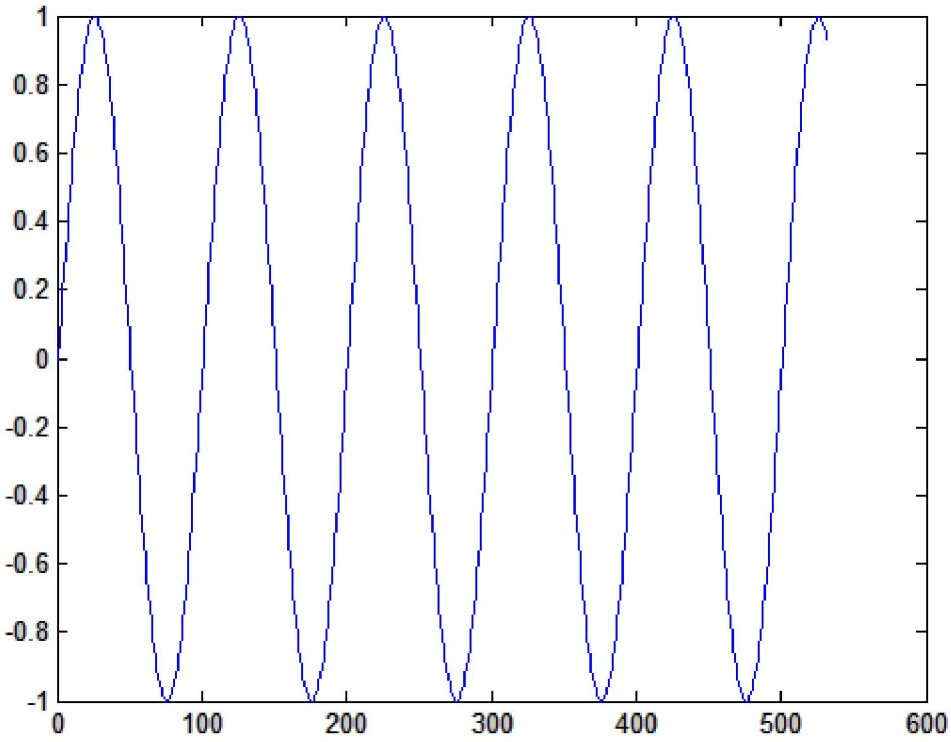
عبارت بالا به هیچ وجه مقدار ندارد چون شامل ضرایب فیلتر ما نمی شود. بنابر این رابطه دوم هم صحیح می باشد.

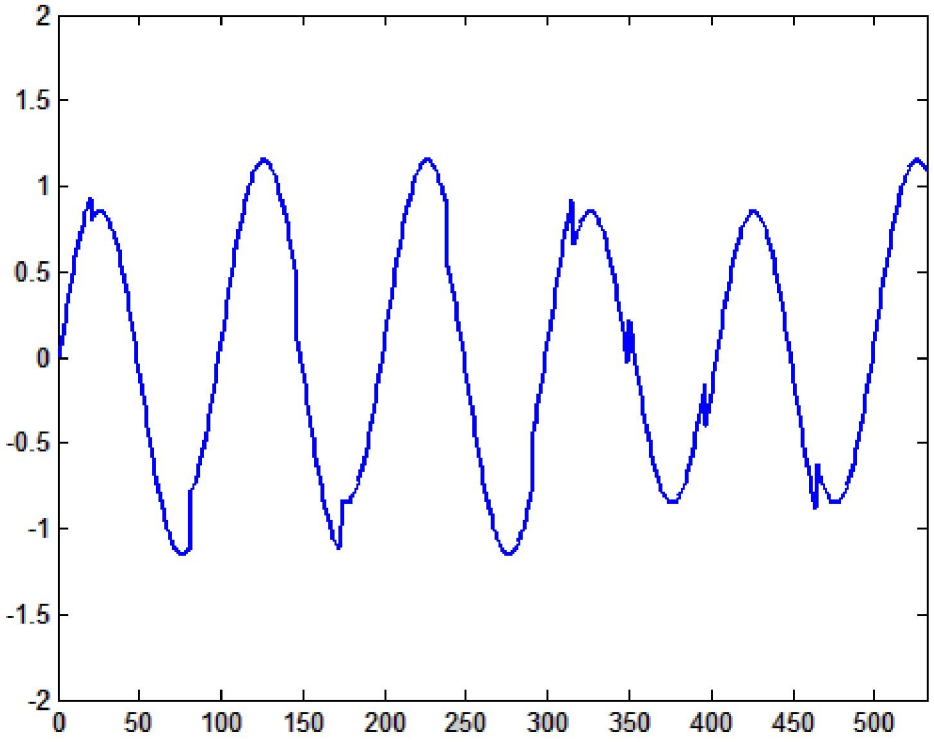
تحلیل سوال ۲:

قسمت الف:

دراین قسمت توضیح چگونگی کار برنامه داده شده خواسته شده است ک در خط اول برنامه یعنی x=zeros(۱,۲۰) ,۲۰عددصفردرx قرارمیگیرد.سپس یک عدد تصادفی به صورت رندوم بین ۲۰تا۷۰میخواهیم تولید کنیم ک برای اینکار ابتدا با کمک دستور rand یک عدد تصادفی بین صفر تایک تولید میکنیم که همانطور که میدانیم توزیع یکنواخت هم دارد.بعد مقدار تصادفی تولید شده بین صفر ویک رادر۵۰ضرب میکنیم ومقدار حاصل شده راگرد میکنیم.سپس مقدار تولیدشده را با عدد ۲۰جمع میکنیم که حاصل درنهایت درr ذخیره میشود.درخط بعد برنامه mod مقدار rرامحاسبه میکنیم ک حاصلش یکی از دومقدار صفر ویک میتواند باشدکه اگرصفر باشد همانطور که میدانیم آن عدد زوج بوده و درغیر اینصورت آن عدد فرد بوده است.درصورتی که مقدار تولید شده زوج بوده باشد با دستورones به تعداد r به x مقداریک اضافه میکند.طبق حلقه for متوجه میشویم که ۱۰ بار جهش خواهیم داشت.هربار باr تولید شده همین روند را تکرار میکند.هم چنین اگر r فرد بودبهx قبلی به تعدادr مقدار منفی یک اضافه میکند.همانطور که دربرنامه مشخص است این xتولید شده طولی خواهد داشت که معمولا بین ۴۰۰تا۵۰۰میباشد که به دلیل رندوم بودنrمقدار این طول ثابت نخواهد بود ودرنتیجه مقدارx هم که به rوابسته است ثابت نخواهد بود.سپس درخط بعدی برنامه میاییم یک سینوسی با فرکانس ۰.۰۱تولید میکنیم.وبا ۰.۱۵برابر مقدارxجمع میشودک درنهایت عددی بین یک ومنفی یک خواهد بود. به استثنای آن ۲۰ مقدار.که همانطور که درشکل ها هم مشخص است به دلیل جمع شدن x با یک مقدار سینوسی پریدگی هایی وجود دارد که خیلی هم زیاد نیستند چون این سینوسی را با۰.۱۵ برابر x جمع میکنیم که شکل های زیردرنهایت حاصل میشوند:


0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
100
200
300
400
500
600


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
100
200
300
400
500
600



کدبرنامه قسمت الف:

```
%%%%%%%%%%%%%%%%%%%%% Problem 2  %%%%%%%%%%%%%%%%
x=zeros(1,20);
index_jump= length(x);
for n=1:10
    r=round(50*rand(1))+20;          %% between 20-70
    k=mod(n,2);
    if k==0                          %% k even
        x=[x,ones(1,r)];
    else                             %% k odd
        x=[x,-ones(1,r)];
    end
    index_jump=[index_jump length(x)];
end
```

```
L=length(x);                          %% x is a sequence
from 0 and 1
t=0:L-1;
y=sin(2*pi/100*t);                    %% create a sinuosly
wave with length of L
z=0.15*x + y;                         %% sum .15 or -.15
with sinosal wave (number of discontuonous's point is
10)

figure;
plot(.15*x);
figure;
plot(y);
figure;
plot(z,'LineWidth',2);
axis([0 length(z) -2 2]);
```

قسمت ب:

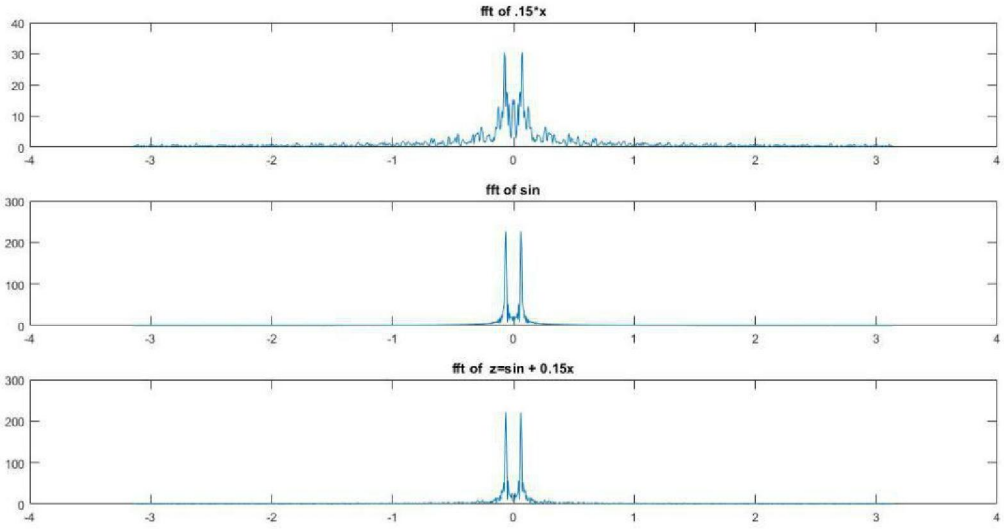
دراین قسمت میخواهیم برای یک نمونه از zمحل دقیق وقوع جهش ها را در سیگنال سینوسی بدست آوریم. برای اینکار میخواهیم نقش نوع موجک را درتعیین دقیق وقوع جهشها بررسی کنیم. تعداد سطوح تجزیه را۴ انتخاب میکنیم و موجک ها را به ترتیب db1,db2,db4,db8 انتخاب میکنیم.

ب.aتوضیح دهید که کدام یک ازسطوح تجزیه برای تعیین دقیق محل وقوع جهش ها مناسب تر است؟ فرض میکنیم پاسخ شمادj است.

دربالا هم توضیح داده شد۴تا db تولید شده است. db1 در d1تاd4 ذخیره میشودو db2 هم درd5 تا d9 ذخیره میشود.به عبارتی دیگر ۴ تا ۴ تا ذخیره میشوند که در نهایت ۱۶ تا ذخیره میکنیم.البته ماتریسa4 هم جداذخیره میشود.برای تعیین اینکه کدام یک از سطوح برای تعیین محل وقوع جهش ها مناسب تر است

باید محتوای فرکانسی را بدست آوریم که ببینیم فرکانس بالا است یا پایین.که همانطور که درعمل میبینیم فرکانس بالا میباشد که در d1 است .درواقع d1سطح تجزیه مناسب برای تعیین محل دقیق جهش ها میباشد. که از نظر تیوری هم صحت این مطلب درزیربررسی شده است:

تبدیل فوریه ی تعدادی از پالس هارا ک سینک میشود با تبدیل فوریه ی سینوسی ک ضربه میشود را با هم جمع میکنیم ومشاهده میکنیم که محتوای فرکانسی بالا میشود.



همان طور که از طیف های فرکانسی پالس و سینوسی مشخص است مشاهده میشود که طیف فرکانسی پالس فرکانس بالا میباشد و طیف فرکانسی سینوسی فرکانس پایین میباشد.وجمع هردویشان هم در نهایت فرکانس بالامیباشد.

کدبرنامه:

```
%%%%%%%%%%%%%%      Decomposition     %%%%%%%%%%%%%%
for i=0:3

    str = sprintf('db%d',2^i);

    [c,l]=wavedec(z,4, str );


d{4*i+1}=c(l(1)+l(2)+l(3)+l(4)+1:l(1)+l(2)+l(3)+l(4)+l(
5));
    d{4*i+2}=c(l(1)+l(2)+l(3)+1:l(1)+l(2)+l(3)+l(4));
    d{4*i+3}=c(l(1)+l(2)+1:l(1)+l(2)+l(3));
    d{4*i+4}=c(l(1)+1:l(1)+l(2));
    a_4{i+1}=c(1:l(1));

end
%%%%%%%%%%%
```

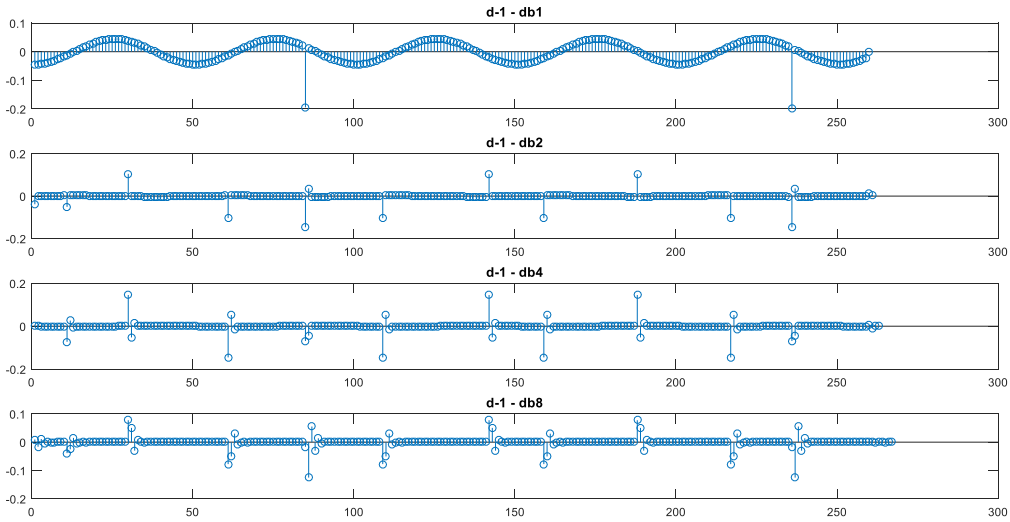
ب (b)

در قسمت اول از ما خواسته شده است که شکل های مربوط به d1 چهار ویولت db1 , db2 , db4 , db8 را در کنار هم رسم کنیم.

در زیر کد و شکل مربوط به رسم این جز تجزیه را مشاهده می کنیم.

```
q=1;
figure;
for i=0:3
    temp=d{4*i+q};
    subplot(4,1,i+1);
    stem(temp);
    str = sprintf('d-%d - db%d',q,2^i);
    title(str)
end
```

و شکل مربوطه آن:



همان طور که در شکل فوق نیز مشهود است db8 توانسته بخوبی نقاط جهش را پیدا کند اما هر چه به سمت ویولت ها با طول کمتر می رویم این دقت ما کاهش یافته است. همانطور که مشاهده میشود بدترین قدرت تفکیک در پیدا کردن نقاط جهش مربوط به db1 است.

علت اینکه db8 بسیار بهتر از db1 توانسته است این نقاط را تشخیص دهد این است که به علت اینکه ممان های بیشتری از ویولت آن صفر شده اند فیلتر های آن به حالت ایده آل نزدیک تر بوده(تیز تر می باشند) و قدرت تفکیک بسیار بهتری دارند.

در نتیجه برای بدست آوردن نقاط جهش از بردار d1 مربوط به ویولت db8 استفاده میکنیم.

برای بدست آوردن اندیس هایی که جهش رخ داده است ، روش کار به این گونه است ابتدا اندیس های غیر صفر بردار d1 (ما از d1 مربوط به db8 برای بدست آوردن نقاط استفاده می کنیم) را توسط دستور find متلب بدست می آوریم البته باید توجه شود که همانطور که در شکل فوق مشخص است در d1 برای یک جهش چند نقطه غیر صفر در کنار هم ممکن است وجود داشته باشد که این امر به علت این است که دقت تفکیک db8 ایده آل نیست حال از میان این نقاط که فاصله ایندکس آن ها کمتر از ۵ است تنها یکی را به عنوان نماینده انتخاب میکنیم.

اگر طول سیگنال را L در نظر گرفته شود، جز تجزیه D1 ، دارای طول $\frac{L}{2}$ می شود در نتیجه نقاطی را که به عنوان ایندکس انتخاب کردیم را باید در دو ضرب کنیم تا ایندکس ها به مقدار حقیقی خود نزدیک شوند.

از طرفی میدانیم اولین جهش ما حتما در نقطه ۲۰ باید رخ دهد اما مشاهده می شود که همواره در این روش بیان شده این نقطه جهش را با اندیس ۲۶ نشان می دهد که با توجه به اینکه جهش در نقطه ۲۰ را می توان به عنوان معیار گرفت و از تمام اندیس های بدست آمده ۶ اندیس کم میکنیم (با چند بار تست کردن مشاهده شد که همواره جواب بهتری بدست می آید).

در زیر کد مربوطه و نتایج روش پیشنهادی را مشاهده میکنیم

```
temp=d{13};  %%% d1 db8
r= find(temp > .01);
r=r(r>5);
r=[r(1) r];
k=1;
No=0;
temp2=[0];
for i=1:length(r)-1

    temp1=r(i+1)-r(i);
    if temp1 > 5
        temp2(k)= r(i);
        k=k+1;
    end
```

```
end
index=[temp2 r(end)];
index=[2*temp2 2*r(end)];
index_enhancement=index-6;
```

		1	2	3	4	5	6	7	8	9	10	11
اندیس های بدست امده اولیه	index	13	26	57	74	98	111	137	155	180	206	*
اندیس ها * ۲	2* index	26	52	114	148	196	222	274	310	360	412	*
از هر اندیس ۶ واحد کم میکنیم	2* index - 6	20	46	108	142	190	216	268	306	354	406	*
اندیس هایی که واقعا در آن ها جهش رخ داده	Real index	20	48	108	144	190	218	268	301	354	408	465

مشاهده می شود روش پیشنهادی به خوبی و با دقت بسیار بالایی توانسته است مکان اندیس ها را تخمین بزند.

تنها نکته ای که می توان به آن اشاره کرد این است که d1 متعلق به db8 توانسته است مکان ۱۰ پرش از ۱۱ پرش موجود را تخمین بزند با توجه به اینکه مشاهده شده با افزایش طول فیلتر و استفاده از دابشیز های مرتبه بالاتر توانسیتم تعداد نقاط بیشتری را تخمین بزنیم احتمالا اگر بجای db8 از db16 یا دابشیز های مرتبه بالاتر استفاده کنیم بتوانیم مکان پرش ۱۱ را نیز مشخص کنیم.

به طور کلی همانطور که در جدول فوق مشاهده میکنیم روش پیشنهادی با دقت بالایی توانسته است مکان نقاط پرش را تخمین بزند.

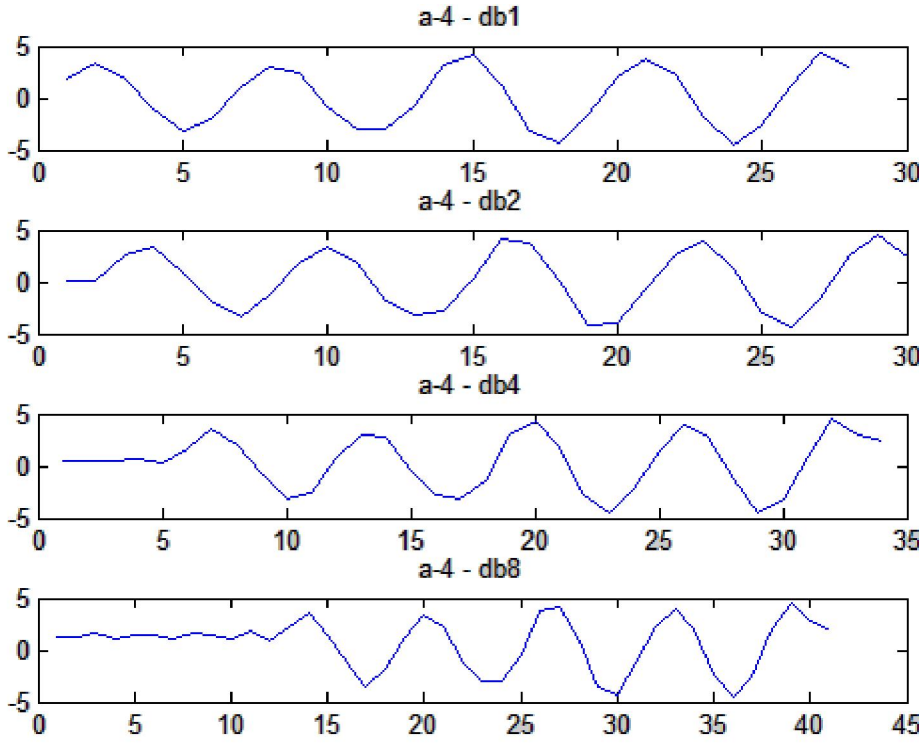
۲-ب:حال تعداد سطوح تجزیه را ۴و موجک را ب ترتیبdb1,db2,db4,db8انتخاب کنید

وتغییراتی که درa4رخ میدهد را نمایش دهید وآن ها را مقایسه کنید .

با مشاهده ی شکل ها میتوان گفت که درdb1 وضعیت بهتر است نسبت سایر dbها.چون محتوای فرکانس

پایین است ونسبت بهdb8بهتر عمل میکند.

```
%%%% plot a coefficents for 4 Decomposition
figure;
for i=1:4
    temp=a_4{i};
    subplot(4,1,i);
    plot(temp);
    str = sprintf('a-4 - db%d',2^(i-1));
    title(str)
end
```



کدبرنامه:

```
%%% B b)
%%%%% plot d1 for 4 Decomposition
q=1;
figure;
for i=0:3
    temp=d{4*i+q};
    subplot(4,1,i+1);
    stem(temp);
    str = sprintf('d-%d - db%d',q,2^i);
    title(str)
end

%%% calculate jump index
temp=d{13};  %%% d1 db8
r= find(temp > .01);
r=r(r>5);
r=[r(1) r];
k=1;
No=0;
temp2=[0];
for i=1:length(r)-1

    temp1=r(i+1)-r(i);
    if temp1 > 5
        temp2(k)= r(i);
        k=k+1;
    end
end
index=[temp2 r(end)];
index=[2*temp2 2*r(end)];
index_enhancement=index-6;
index_jump;
%%%%%%%%%%%%
```

قسمت ج:

یکبار میخواهیم موج سینوسی را از جهش های موج پالسی شکل پاک کنیم وبار دیگر موج سینوسی

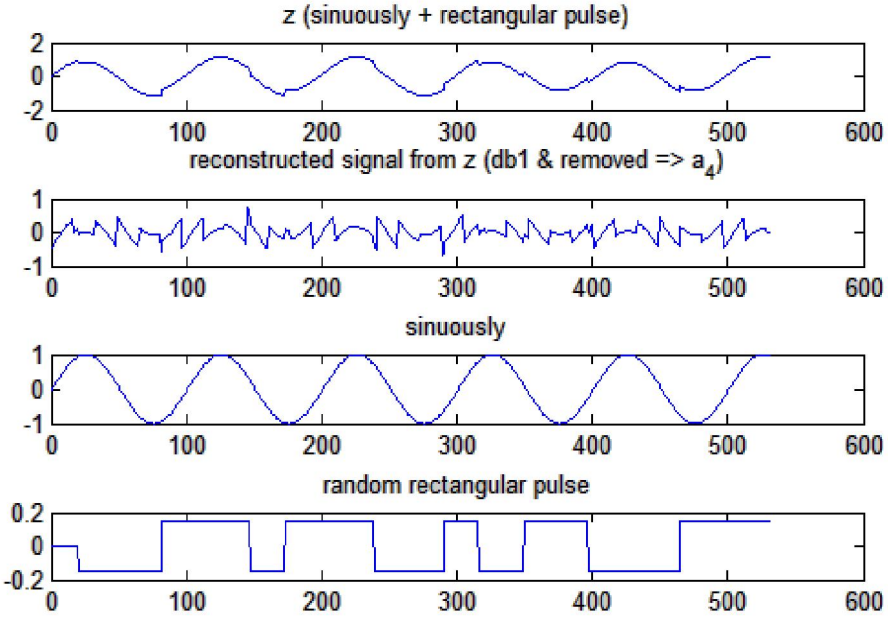
را حذف کنیم.وسیگنال پالسی شکل را استخراج کنیم وتحقیق کنیدو توضیح دهید برای هر کدام از مقاصد فوق چه روشی مناسب است؟

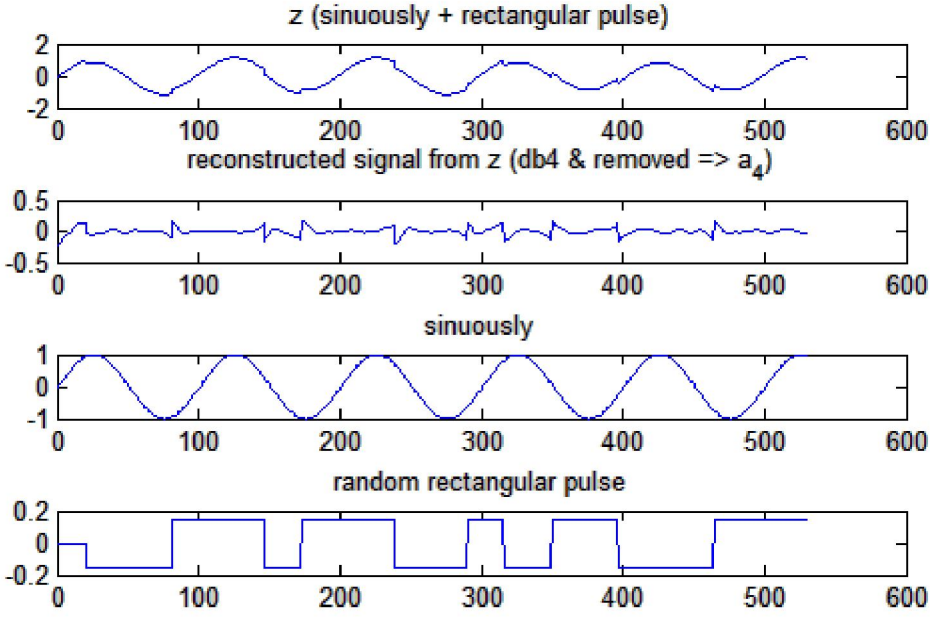
طبق برنامه نوشته شده در زیر برای استخراج سینوسی خالص یا پالس خالص به این صورت عمل میکنیم که ابتداویولت را مشخص میکنیم به این صورت که اگرu=0 باشد یعنی db1 مدنظر است و اگر u=1باشد یعنی db2 مدنظر است وبه همین صورت برای بقیه.که درنهایت ۴ تا ویولت با کمک مقادیر مختلفuمشخص میشوند.هر کدام از این ویولت ها هم ۵ حالت دارند.چون برای استخراج سینوسی یا پالس خالص باید یکی از ۵ حالتd1,d2,d3,d4,a4 حذف شوند.در حلقه دوم اگر a=0 باشدa4حذف میشود واگرa=1باشدd1حذف میشود.و به همین ترتیب برای بقیه به همین صورت است.سپس با کمک شکل های حاصل شده میتوان متوجه شد که با حذفdb8-d2,db4-d3 تقریبا میتوان سینوسی خالص استخراج کردوبا حذفdb8-a4,db4-a4 میتوان پالس را استخراج کرد.

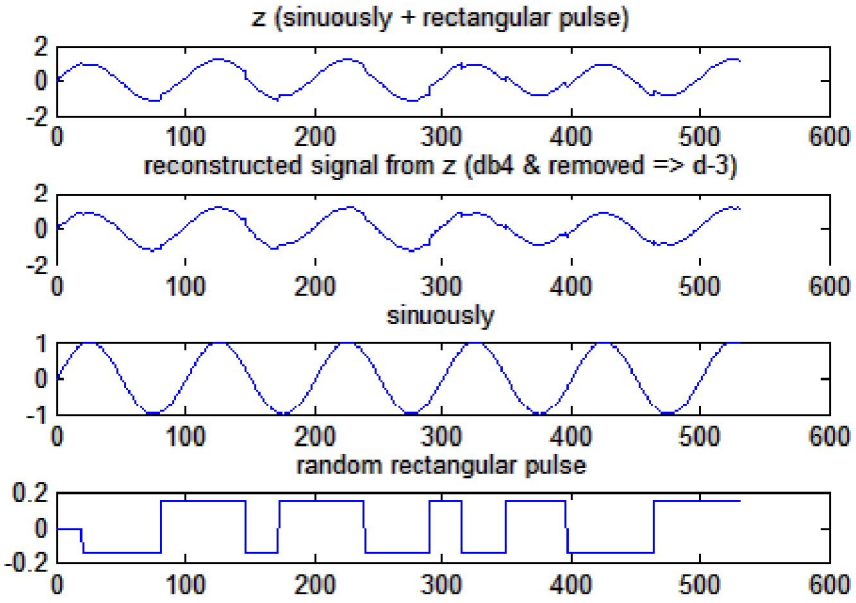
میتوان برای مشخص نشدن واضح لبه ها درdb8-a4به دقیق نبودن فیلتر ها و وجودdidelobe هااشاره کرد..همچنین میدانیم هرچقدر dbبالاتر رود فیلتر ها تیزتر میشوند و استخراج جهش ها بهتر میشود. موج سینوسی را با دقت وصحت بیشتری میتوان استخراج کرد چون تک فرکانس میباشد و نسبت به پالس طیف فرکانسی بهتری دارد.ولی در پالس ها به علت رندوم بودن وکوچک و بزرگ شدن طول سینک هاو متفاوت بودن طول ها فرکانس ها بهم میریزدونسبت به سینوسی تک فرکانس نتیجه خوبی نمیدهد.

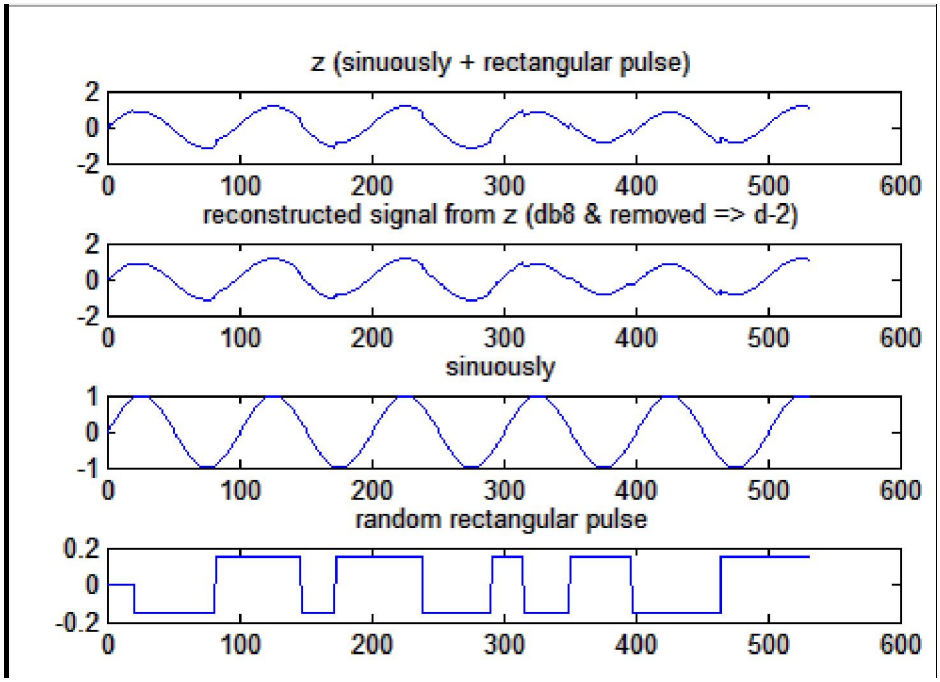
درزیر شکل ها آورده شده اند ک با دقت در آنها میتوان ب نتایج بالا دست یافت.

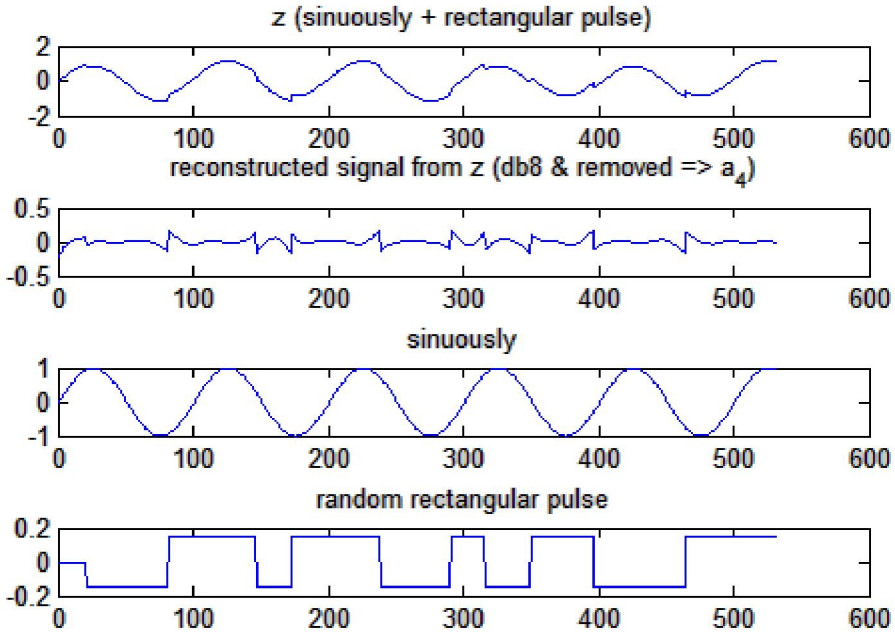
نکته:با کمکwavelet packetهم می توان محل دقیق لبه ها را ب خوبی تشخیص داد.


z (sinuously + rectangular pulse)
reconstructed signal from z (db1 & removed => a_4)
sinuously
random rectangular pulse


z (sinuously + rectangular pulse)
reconstructed signal from z (db4 & removed => a_4)
sinuously
random rectangular pulse


z (sinuously + rectangular pulse)
reconstructed signal from z (db4 & removed => d-3)
sinuously
random rectangular pulse


z (sinuously + rectangular pulse)
reconstructed signal from z (db8 & removed => d-2)
sinuously
random rectangular pulse




```
%%%%% C
%%%% plot fft


%%% u=0 => db1 ** u=1 => db2 ** u=2 => db4 ** u=3 =>
db8
for u=0:3
    q=4*u;

    %%% a=1 => d_1 **  a=2 => d_2 **  a=3 => d_3 **
a=4 => d_4 **  a=0 => a_4
    for a=0:4


        switch a
            case 1 %% d1
                d_1_New=zeros(1,length(d{q+1}));

                C=[a_4{u+1}  d{q+4}  d{q+3}  d{q+2}
d_1_New];
```

```
                L=[length(a_4{u+1})  length(d{q+4})
length(d{q+3})  length(d{q+2}) length(d_1_New)
length(z)];
                db_type = sprintf('db%d',2^u);
                z_New=waverec(C,L, db_type);

                title_data=sprintf('%s & removed => d-
%d',db_type,a);
            case 2 %% d2
                d_2_New=zeros(1,length(d{q+2}));

                C=[a_4{u+1}  d{q+4}  d{q+3}  d_2_New
d{q+1}] ;
                L=[length(a_4{u+1})  length(d{q+4})
length(d{q+3}) length(d_2_New) length(d{q+1})
length(z)];
                db_type = sprintf('db%d',2^u);
                z_New=waverec(C,L, db_type);

                title_data=sprintf('%s & removed => d-
%d',db_type,a);
            case 3 %%d3
                d_3_New=zeros(1,length(d{q+3}));

                C=[a_4{u+1}  d{q+4}  d_3_New d{q+2}
d{q+1} ];
                L=[length(a_4{u+1})  length(d{q+4})
length(d_3_New) length(d{q+2})  length(d{q+1})
length(z)];
                db_type = sprintf('db%d',2^u);
                z_New=waverec(C,L, db_type);

                title_data=sprintf('%s & removed => d-
%d',db_type,a);
            case 4 %% d4
                d_4_New=zeros(1,length(d{q+4}));

                C=[a_4{u+1}  d_4_New d{q+3}  d{q+2}
d{q+1}];
```

```
                L=[length(a_4{u+1})  length(d_4_New)
length(d{q+3})  length(d{q+2})  length(d{q+1})
length(z)];
                db_type = sprintf('db%d',2^u);
                z_New=waverec(C,L, db_type);

                title_data=sprintf('%s & removed => d-
%d',db_type,a);
            case 0 %% c4
                a_4_New=zeros(1,length(a_4{u+1}));

                C=[a_4_New d{q+4} d{q+3}  d{q+2}
d{q+1}];
                L=[length(a_4_New)  length(d{q+4})
length(d{q+3})  length(d{q+2})  length(d{q+1})
length(z)];
                db_type = sprintf('db%d',2^u);
                z_New=waverec(C,L, db_type);

                title_data=sprintf('%s & removed =>
a_4',db_type);
        end

        figure('units','normalized','outerposition',[0
0 1 1])
        subplot(411)
        plot(z);  title('z (sinuously + rectangular
pulse)')
        subplot(412)
        plot(z_New);  str=sprintf('reconstructed signal
from z (%s)',title_data);  title(str);
        subplot(413);
        plot(y);      title('sinuously')
        subplot(414);
        plot(.15*x);      title('random rectangular
pulse')

    end
end
```