

١

٢

٣

٤



: -

%٦. (٥/٣)

%١

[]

%٠,١

- - - - -

[Y]

[Y]

%\

α

$D^{\alpha} D^{\beta}$

[Y]

: $D^{\alpha} D^{\beta} \alpha$

[Y]()

CVSD-OPT

MatLab

.()

...

s[n]

(206) 128 - 128

...

۳۲...
 . ۳۲...
 $D^۲ \quad D^۱ \quad \beta \quad \alpha$

F.M. [%]

$$F.M. = \left[\sum_{n=1}^{3200} (s[n])^2 \right] / \left[\sum_{n=1}^{3200} (s[n] - s(n))^2 \right] \quad ()$$

:s[n] ۳۲... F.M.

S[n] Hz	α	β	$D^۱$	$D^۲$	F.M.
۳۲۰	۰.۸۸	۰.۹۰	۱.۴	۰.۴	۲۹.۸۹۸
۱۰۰۰	۰.۸۸	۰.۹۰	۱.۴	۰.۴	۲۹.۸۹۸
۲۰۰۰	۰.۸۳	۰.۹۰	۱.۴	۰.۱	۸.۴۲۰
۴۰۰۰	۰.۴۵	۰.۹۲	۱.۱	۰.۵	۱.۴.۴۳۱
۶۰۰۰	۰.۲۲	۰.۹۲	۱.۰	۴.۴	۶.۰۳۱
۸۰۰۰	۰.۶۱	۰.۹۵	۱.۲	۵.۱	۱۷.۱۰۵

۱

ξ

ξ ۱

:

$$\alpha = 0.76, \quad \beta = 0.90, \quad D = \quad , \quad D = \quad ,$$



..

.

:

-

-

۲۵۰

۳۲۰۰۰

)

.(

-

.

-

.[۱]

-

)

%

.(

:

-

/ -
:
"

٢٥٦

٣٢٠٠٠

) %٦

(٥\١

: (٢)

٥\٤	١	٤	٥\٣	٢	١
٥\٣	١	٥	٥\٤	١	٢
٥\٤	١	٦	٥\٣,٥	١	٣

" -

- - - - -

:
 ۲۴...
 %۶
 ۶...
 : ۲

۰ ۴.۰	۱	۴	۰ ۳.۰	۱	۱
۰ ۴	۱	۰	۰ ۳	۱	۲
۰ ۴.۲۰	۱	۶	۰ ۴.۰	۱	۳

۳۲...
 ۶...

:(۴)

۰ ۴.۰	۱	۴	۰ ۳.۷۰	۱	۱
۰ ۰	۱	۰	۰ ۴.۰	۱	۲

0/4.0	1	6	0/4.0	1	3
-------	---	---	-------	---	---

4

: %12

' ' "

"

: (0)

0/4	1	4	0/4	1	1
0/3	1	0	0/4.0	1	2
0/4	1	6	0/2	1	3

0

:

•

•

•

•

0/1

)

0/3

.(

- - - - -

•

%) .

:

-

CVSD_TRNP_S

-

%) 2 % 6

:

-

-

%) 2 .

-

DSP

FPGA

\

```

% *** CVSD_OPT ***
clear;
for n=1:300;
    s(n)=round(1^k*sin(pi*n/100));
end
for N=1:100;
    z=input('Enter values for z in brackets:');
    alpha=z(1);
    beta=z(2);
    D1=z(3);
    D2=z(4);
    if s(1)>=0;
        c(1)=1;
        delta1=D2;
    else
        c(1)=0;
        delta1=-D2;
    end
    s_car1=delta1;
    if s(2)>=s(1);
        c(2)=1;
        delta2=beta*abs(delta1)+D2;
    else
        c(2)=0;
        delta2=-(beta*abs(delta1)+D2);
    end
    s_car2=alpha*s_car1+delta2;

```

```

- - - - -


---


for n=Υ:length(s);
  s_bar=alpha*s_carΥ;
  d=s(n)-s_bar;
  if d>=0;
    c(n)=1;
  else
    c(n)=0;
  end
  if c(n-Υ)==c(n-1) & c(n-1)==c(n);
    deltaΥ=beta*abs(deltaΥ)+D1;
  else
    deltaΥ=beta*abs(deltaΥ)+DΥ;
  end
  if c(n)==0;
    deltaΥ=-deltaΥ;
  else
    end
    d_car=deltaΥ;
    s_carΥ=alpha*s_carΥ+d_car;
    deltaΥ=deltaΥ;
    s_carΥ=s_carΥ;
  end
  if c(1)==1;
    delta1=DΥ;
  else
    delta1=-DΥ;
  end
  s_carp(1)=delta1;
  if c(Υ)==1;
    deltaΥ=beta*abs(delta1)+DΥ;
  else
    deltaΥ=-(beta*abs(delta1)+DΥ);
  end
  s_carp(Υ)=alpha*s_carp(1)+deltaΥ;
for n=Υ:length(c);
  if c(n-Υ)==c(n-1) & c(n-1)==c(n);
    deltaΥ=beta*abs(deltaΥ)+D1;
  else
    deltaΥ=beta*abs(deltaΥ)+DΥ;
  end
end

```

```

if c(n)==0;
    delta_r=-delta_r;
else
end
s_carp(n)=alpha*s_carp(n-1)+delta_r;
delta_r=delta_r;
end
F.M=sum(s.^r)/sum((s-s_carp).^r);
F.M
end

```

2

```

% * * * CVSD_TRNP_S * * *
clear;
w(1:10)=0,0.4-0.1,0.4*cos(r*pi*(1:10)/10);
load EZA_r_k; s=round(10^4*EZA_r_k);
al=0.76; bt=0.9; D1=1; D_r=0.3; c=[1 0];
d_r=0; s_r=0; dd_r=0; sc(r)=0;
N=1;
for n=r:10:length(s)-10;
    I=n-r;
    for k=r:10^4;
        d=s(I+k)-al*s_r;
        if d>=0;
            c(k)=1;
        else
            c(k)=0;
        end
        if c(k-r)==c(k-1) & c(k-1)==c(k);
            d_r=bt*abs(d_r)+D1;
        else
            d_r=bt*abs(d_r)+D_r;
        end
        if c(k)==0;
            d_r=-d_r;
        else
            end
            s_r=al*s_r+d_r;
            d_r=d_r;
        end
    end
    ca=[c(r:10);c(10^4:10^4+10);c(10^4+10:10^4+20);c(10^4+20:10^4+30);...]

```

```

c(٦٧:٨٢);c(٨٣:٩٨);c(٩٩:١١٤);c(١١٥:١٣٠);...
c(١٣١:١٤٦);c(١٤٧:١٦٢);c(١٦٣:١٧٨);...
c(١٧٩:١٩٤);c(١٩٥:٢١٠);c(٢١١:٢٢٦);...
c(٢٢٧:٢٤٢);c(٢٤٣:٢٥٨)];
ci=[ca(:,١);ca(:,٢);ca(:,٣);ca(:,٤);ca(:,٥);...
ca(:,٦);ca(:,٧);ca(:,٨);ca(:,٩);...
ca(:,١٠);ca(:,١١);ca(:,١٢);ca(:,١٣);...
ca(:,١٤);ca(:,١٥);ca(:,١٦)];
for h=٢٤١:٢٥٦;
if rand>=٠,٥;
ci(h)=٠;
else
ci(h)=١;
end
end
cb=[ci(١:١٦),ci(١٧:٣٢),ci(٣٣:٤٨),ci(٤٩:٦٤),...
ci(٦٥:٨٠),ci(٨١:٩٦),ci(٩٧:١١٢),...
ci(١١٣:١٢٨),ci(١٢٩:١٤٤),ci(١٤٥:١٦٠),...
ci(١٦١:١٧٦),ci(١٧٧:١٩٢),ci(١٩٣:٢٠٨),...
ci(٢٠٩:٢٢٤),ci(٢٢٥:٢٤٠),ci(٢٤١:٢٥٦)];
cd=[c(١),c(٢),cb(١,:),cb(٢,:),cb(٣,:),cb(٤,:),...
cb(٥,:),cb(٦,:),cb(٧,:),cb(٨,:),...
cb(٩,:),cb(١٠,:),cb(١١,:),cb(١٢,:),...
cb(١٣,:),cb(١٤,:),cb(١٥,:),cb(١٦,:)];
for m=٣:١٦:٢٤٣;
M=N+m-٤;
J=m-٣;
for k=٣:١٧;
f=M+k;
f١=J+k;
if cd(f١-٢)==cd(f١-١) & cd(f١-١)==cd(f١);
dd٢=bt*abs(dd١)+D١;
else
dd٢=bt*abs(dd١)+D٢;
end
if cd(f١)==٠;
dd٢=-dd٢;
end
sc(f)=al*sc(f-١)+dd٢;
z١(k)=sc(f);
dd١=dd٢;
end
z٢=z١(٣:١٧).*w;
R(١)=sum(z٢(١:١٥).*z٢(١:١٥));

```

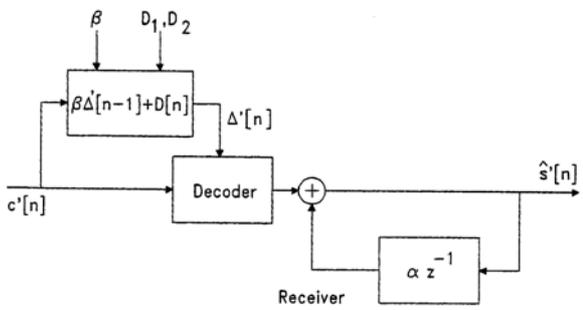
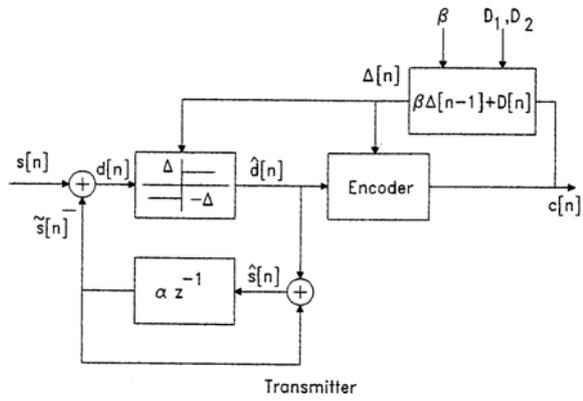
```

R(Υ)=sum(z^(1:1 ξ).*z^(Υ:1 0));
R(Ψ)=sum(z^(1:1 Ψ).*z^(Ψ:1 0));
R(Ξ)=sum(z^(1:1 Ξ).*z^(Ξ:1 0));
R(ϕ)=sum(z^(1:1 ϕ).*z^(ϕ:1 0));
R(Ϸ)=sum(z^(1:1 Ϸ).*z^(Ϸ:1 0));
R(Υ)=sum(z^(1:1 ϑ).*z^(Υ:1 0));
R(Λ)=sum(z^(1:1 Λ).*z^(Λ:1 0));
R(ϑ)=sum(z^(1:1 ϑ).*z^(ϑ:1 0));
R(1 Ϸ)=sum(z^(1:1 Ϸ).*z^(1 Ϸ:1 0));
R(1 ϕ)=sum(z^(1:1 ϕ).*z^(1 ϕ:1 0));
ph(1,Υ:1 Ϸ)=R(Υ:1 Ϸ);
ph(Υ,Υ:1 Ϸ)=R(Υ:ϑ);
ph(Ψ,Ξ:1 Ϸ)=R(Υ:Λ);
ph(Ξ,ϕ:1 Ϸ)=R(Υ:Υ);
ph(ϕ,Ϸ:1 Ϸ)=R(Υ:Ϸ);
ph(Ϸ,Υ:1 Ϸ)=R(Υ:ϕ);
ph(Υ,Λ:1 Ϸ)=R(Υ:Ξ);
ph(Λ,ϑ:1 Ϸ)=R(Υ:Ψ);
ph(ϑ,1 Ϸ)=R(Υ);
ph(1 Ϸ,1 Ϸ)=1;
phe=R(1)*eye(1 Ϸ);
ph^=ph';
ph^=ph+ph^+phe;
a=R(Υ:1 Ϸ)*inv(ph^);
p=dot(a,z^(1 Ϸ:-1:Λ));
f^=f^+1;
if p>=sc(f);
    cd(f^)=1;
else
    cd(f^)=0;
end
if cd(f^ - Υ)==cd(f^ - 1) & cd(f^ - 1)==cd(f^);
    dd^=bt*abs(dd^)+D^;
else
    dd^=bt*abs(dd^)+D^;
end
if cd(f^)==0;
    dd^=-dd^;
end
f=f+1;
sc(f)=al*sc(f-1)+dd^;
dd^=dd^;
end
c(1)=cd(Υ^Υ);

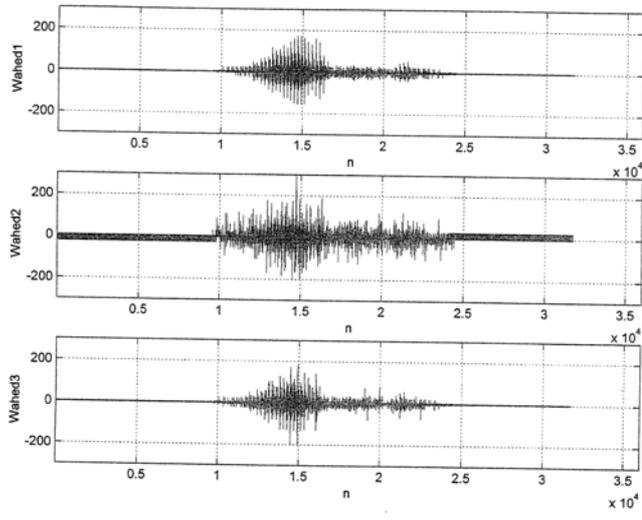
```

- - - - -

```
c(Y)=cd(Y^o^A);  
N=N+Y^o^V;  
end
```

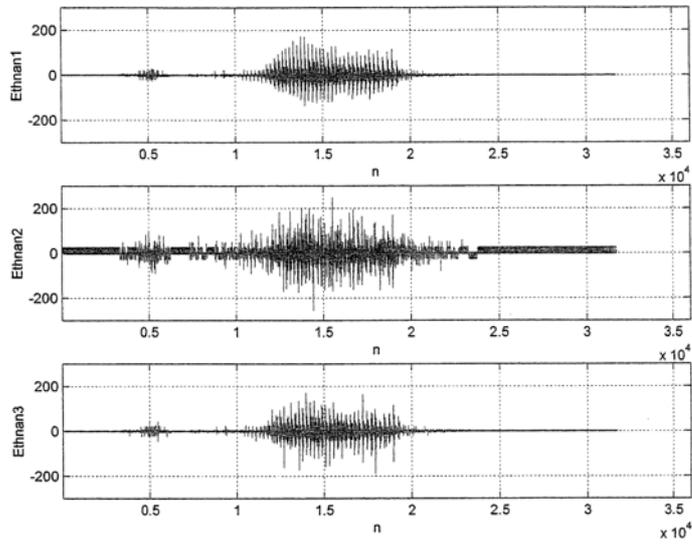


(CVSD)



" " " " " "

: Wahed¹
 : Wahed²
 : Wahed³

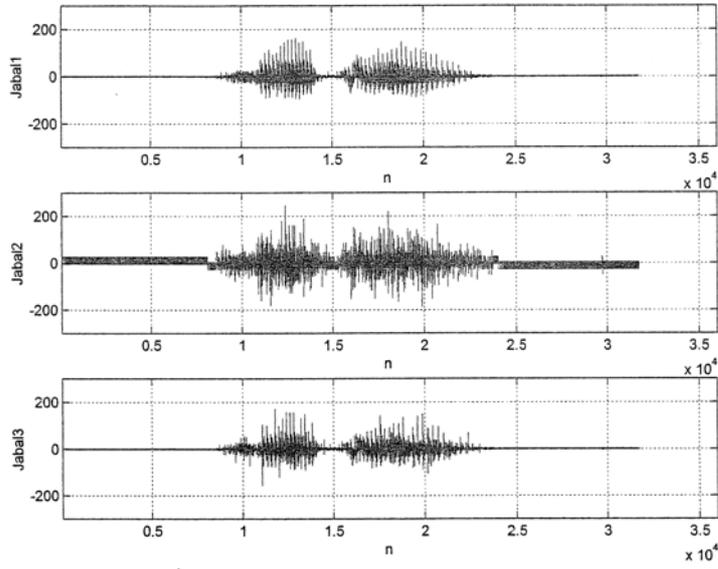


" " "

: Ethnan \

: Ethnan \

: Ethnan \

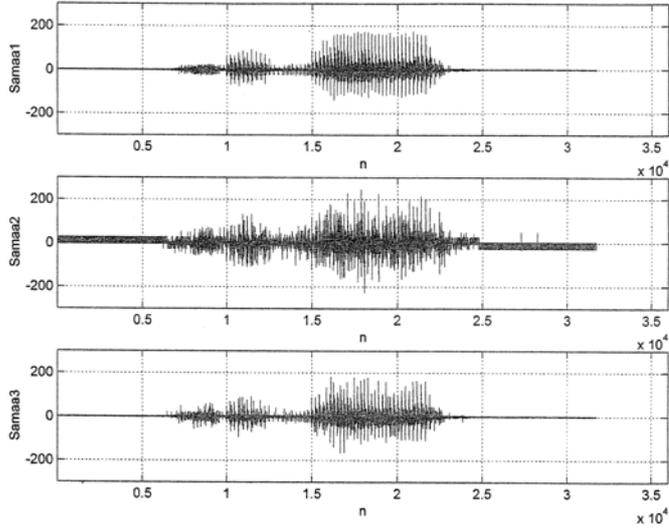


" " 4

: Jabal\

: Jabal^

: Jabal^

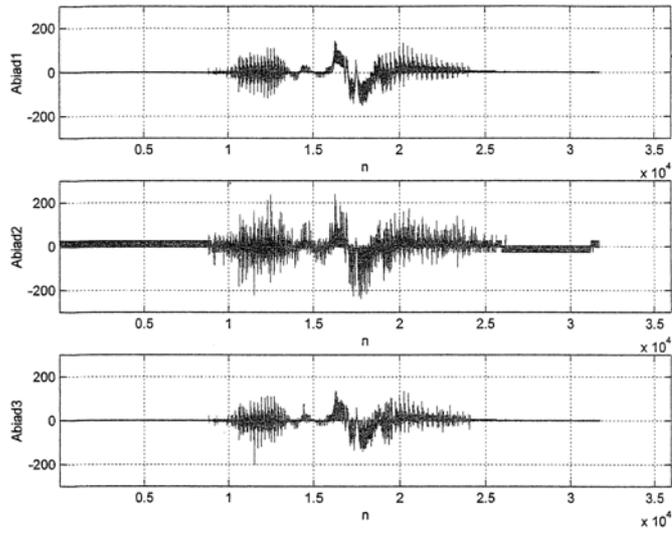


" " °

: Samaa¹

: Samaa²

: Samaa³



: Abiad^١

: Abiad^٢

: Abiad^٣

[١] - نور الدين إبراهيم إبراهيم, مقالة بعنوان " مساهمة في ترميم الإشارات الصوتية المرمزة بالتعديل التفاضلي ذي الميل المتغير باستمرار " , مجلة باسل الأسد للعلوم الهندسية , العدد ١٦ لعام ٢٠٠٢ .

[٢]- Barnwell, T. P., Nayebi, K.,Richardson, C. H., Speech Coding, A Computer Laboratory Textbook, John Wiley & Sons, Inc. ١٩٩٦.

[٣]- Lathi,B. P.,Modern Digital and Analog Communication Systems, CBS College Publishing, New York, ١٩٨٣.

[٤]- Rabiner, L. R., Schafer, R. W., Digital Processing of Speech Signal, Prentice Hall, New Jersey, ١٩٧٨.

[٥]- A. M. Kondoz, Digital Speech Coding for Low Bit Rate Communication Systems,John Wiley & Sons,١٩٩٨.

[٦]- Speech Signal Recovery in Packet- Switched Communication Networks, ١٧th Instrumentation and Measurement Technology Conference, IMTC, May ١-٤, ٢٠٠٠, Baltimore, Maryland, USA, Proceedings, pp. ٥٢-٥٦.

- - - - -



//