EC519 CELP Code

Jiangyu Wang

May 4, 2017

```
% Author: Jiangyu Wang
% Date: Apr 26 2017
% Email: jiangyu@bu.edu

clear;

%% Parameters Setting
bitrate = 4800;
[x,fs] = audioread('s5.wav');
lporder = 10;
winlen = 240; % 30ms
winshift = 240; % Non-overlapping
frame_num = round((length(x)-3*winlen-150)/winlen);
load('pp5.mat');
bit_alc = [7 7 7 6 6 5 5 4 4 3];
Gain_max=[5 3.5 2.5 1.5 1.5 2.5 1 1.5 1.5 1]';

if bitrate == 4800
    bit_alc = [16 16 15 14 14 14 13 13 11 9];
    bits_adpcb = 6;
elseif bitrate == 9600
    bit_alc = [17 16 16 13 11 11 11 11 10 10];
    bits_adpcb = 7;
end

%% Spectrum
```

```
coeff(1:10,1:frame_num)=0;
for i = 1:frame_num
    frame_procs=x(1+(i-1)*winlen:i*winlen);
    coeff_frameu=lpc(frame_procs,10);
    coeff_frame=-coeff_frameu(2:11)';
    coeff(:,i)=coeff_frame';
end
G_unquantize = []; sig_quantize = [];
alpha(1:lporder,1:lporder)=0; kk(1:lporder,1)=0;
for k = 1:frame_num
coeff_frame=coeff(:,k);
alpha(1:lporder,1:lporder)=0; alpha(1:lporder,lporder)=coeff_frame;
kk(1:lporder,1)=0; kk(lporder)=alpha(lporder,lporder);
for i=lporder:-1:2
    for j=1:i-1
        alpha(j,i-1)=(alpha(j,i)+kk(i)*alpha(i-j,i))/(1-kk(i).^2);
    end
    kk(i-1)=alpha(i-1,i-1);
end
g(1:lporder,1)=0;
for i=1:lporder
    g(i,1)=log((1-kk(i,1))/(1+kk(i,1)));
end
sig_frame(1:lporder,1)=0;
for i=1:lporder
   if g(i)<0
       sig_frame(i)=0;
   else sig_frame(i)=1;
   end
end
sig_quantize=[sig_quantize sig_frame];
G_unquantize = [G_unquantize g];
end

G_quantize(lporder,1:frame_num)=0; ks_g(1:lporder,1:frame_num)=0;
for i=1:frame_num
    for k=1:lporder
        ks_g(k,i)=compand(abs(G_unquantize(k,i)),40,Gain_max(k,1),'mu/c
```

```
        G_quantize(k,i)=quantiz(ks_g(k,i),0:Gain_max(k,1)/2^(bit_alc(1,l
    end
end
[pp,nn]=size(G_quantize); g_bint(1:pp,1)=0;
k_unquan(1:pp,1:nn)=0; g_unquan(1:pp,1:nn)=0; coeff_unquan(1:pp,1:nn)=0
for i=1:nn
    for k=1:pp
        g_unquan(k,i)=compand(G_quantize(k,i)*g_bint(k,1),40,Gain_max(k
        if sig_quantize(k,i)==0
            g_unquan(k,i)=-1*g_unquan(k,i);
        end
        k_unquan(k,i)=(1-exp(g_unquan(k,i)))/(1+exp(g_unquan(k,i)));
    end
    alpha(1:pp,1:pp)=0;
    for ii=1:pp
        alpha(ii,ii)=k_unquan(ii,i);
        if (ii > 1)
            for j=1:ii-1
                alpha(j,ii)=alpha(j,ii-1)-k_unquan(ii,i)*alpha(ii-j,ii-
            end
        end
    end
    coeff_unquan(:,i)=alpha(1:pp,pp);
end
 [h1,w1]=freqz(1,[1 -coeff(:,21)']); ww=w1*fs/pi/2; H1=10*log10(abs(h1)
 plot(ww,H1); title('Power Spectrum'); hold on;
%  [h2,w2]=freqz(1,[1 -coeff_unquan(:,21)']); ww=w2*fs/pi/2; H2=10*log10
%  plot(ww,H2); title('Power Spectrum');

%% d(n),e(n),d(n)',e(n)' Unquantized
 d_unquan(1:240*frame_num,1)=0; e_unquan(1:240*frame_num,1)=0;
 dd_unquan(1:240*frame_num,1)=0; ee_unquan(1:240*frame_num,1)=0;
 coeff(1:10,1:frame_num)=0; adpcb_enc(1:240,1)=0;
 randn('seed',0); stocb=randn(60,2^8);
for i=1:frame_num
    frame_procs=x(1+(i-1)*winlen:i*winlen);
    coeff_frameu=lpc(frame_procs,10);
    coeff_frame=-coeff_frameu(2:11)';
```

```
coeff(:,i)=coeff_frame';
d_unquan(1+(i-1)*240:240*i,1)=filter([1 -coeff_frame'],1,frame_procs

coeff_perc=coeff_frame*0.85;
adp_result_frame(1:4,1)=0; adp_gain_result_frame(1:4,1)=0;
sto_result_frame(1:4,1)=0; sto_gain_result_frame(1:4,1)=0;
e_unquan_frame(1:240,1)=0; d_unquan_frame(1:240,1)=0; ee_unquan_fram

for j=1:4

    X_ref=frame_procs(1+(j-1)*60:j*60);
    adpcb_subf(:,1:2^bits_adpcb)=toeplitz(adpcb_enc(2^bits_adpcb:2^
    b+59),flipud(adpcb_enc(1:2^bits_adpcb)));
    adpcb_ref=filter(1,[1 -coeff_frame'],adpcb_subf);

    adpcb_eng=sum(adpcb_ref.^2);
    adpcb_corr=X_ref'*adpcb_ref;
    kk= find(adpcb_corr==max(adpcb_corr));
    aa = length(kk);
    if (aa>0)
        adpcb_index_subf = kk(1)+240-2^bits_adpcb;
        adpgain_subf=abs(adpcb_corr(kk(1))/(adpcb_eng(kk(1))+10*eps
        if adpgain_subf>1.4
          adpgain_subf=1.4;
        end
    else
        adpcb_index_subf = 240-2^bits_adpcb;
        adpgain_subf=1.4;
    end

    excit_subf = adpgain_subf*adpcb_enc(240-adpcb_index_subf+1:240-
    e_unquan_frame(1+(j-1)*60:60*j)=excit_subf;
    X_subf=X_ref-filter(1,[1 -coeff_frame'],excit_subf);
    X_ref=X_subf;
    stocb_ref = filter(1,[1 -coeff_frame'],stocb);

    stocb_eng=sum(stocb_ref.^2);
    stocb_corr=X_ref'*stocb_ref;
```

```
        stocb_index_subf=find(stocb_corr==max(stocb_corr));
        bb = length(stocb_index_subf);
        if (bb>0)
            stogain_subf=stocb_corr(stocb_index_subf(1))/stocb_eng(stocb
            ee_unquan_frame(1+(j-1)*60:60*j)=stogain_subf*stocb(:,stocb
            excit_subf = stogain_subf*stocb(:,stocb_index_subf(1))+adpg
            -adpcb_index_subf+1:240-adpcb_index_subf+60);
            sto_result_frame(j)=stocb_index_subf(1);
        else
            stogain_subf=stocb_corr(1)/stocb_eng(1);
            ee_unquan_frame(1+(j-1)*60:60*j)=stogain_subf*stocb(:,1);
            excit_subf = stogain_subf*stocb(:,1)+adpgain_subf*adpcb_enc(
            -adpcb_index_subf+60);
            sto_result_frame(j)=1;
        end
        d_unquan_frame(1+(j-1)*60:60*j)=excit_subf;
        adpcb_enc = [adpcb_enc(61:240); excit_subf];

        adp_result_frame(j)=adpcb_index_subf(1);
        adp_gain_result_frame(j)=adpgain_subf;

        sto_gain_result_frame(j)=stogain_subf;
        X_syn_uq_frame(1+(j-1)*60:60*j)=filter(1,[1 -coeff_frame'],exci

    end

    X_syn_uq(1+(i-1)*240:240*i,1)=X_syn_uq_frame;
    e_unquan(1+(i-1)*240:240*i,1)=e_unquan_frame;
    dd_unquan(1+(i-1)*240:240*i,1)=d_unquan_frame;
    ed_unquan(1+(i-1)*240:240*i,1)=ee_unquan_frame;

    adp_result(1:4,i)=adp_result_frame; adp_gain_result(1:4,i)=adp_gain
    sto_result(1:4,i)=sto_result_frame; sto_gain_result(1:4,i)=sto_gain
end
plot(1:240,d_unquan(1+(21-1)*240:21*240),1:240,d_quan(1+(21-1)*240:21*24
legend('Unquantized','Quantized');
title('Short term LP residual');
```

```
plot(1:240,e_unquan(1+(21-1)*240:21*240),1:240,e_quan(1+(21-1)*240:21*24
legend('Unquantized','Quantized');
title('Short and Long term LP residual');

%% d(n),e(n),d(n)',e(n)' Quantized
synthiezed_quan(1:240*n,1)=0; adpcb_dec(1:240,1)=0;
d_quan_frame(1:240,1)=0; e_quan_frame(1:240,1)=0;
dd_quan_frame(1:240,1)=0; ee_quan_frame(1:240,1)=0;

for i=1:frame_num
    synthized_quan_frame(1:240,1)=0;
    excit_subf(1:60,1)=0;
    for j=1:4
        e_quan_frame(1+(j-1)*60:60*j)=adp_gain_result_exp(j,i)*adpcb_dec
        -adp_result(j,i)+60);
        ee_quan_frame(1+(j-1)*60:60*j)=sto_gain_result_exp(j,i)*stocb(:
        excit_subf=sto_gain_result_exp(j,i)*stocb(:,sto_result(j,i))+
         adp_gain_result_exp(j,i)*adpcb_dec(240-adp_result(j,i)+1:240-a

        dd_quan_frame(1+(j-1)*60:60*j)=excit_subf;
        adpcb_dec=[adpcb_dec(61:240); excit_subf];
        synthized_quan_frame(1+(j-1)*60:60*j)=filter(1,[1 -al_exp(:,i)'
    end
    d1_q(1+(i-1)*240:240*i,1)=d_quan_frame; e1_q(1+(i-1)*240:240*i,1)=e
    d2_q(1+(i-1)*240:240*i,1)=dd_quan_frame; e2_q(1+(i-1)*240:240*i,1)=
    synthiezed_quan(1+(i-1)*240:240*i,1)=synthized_quan_frame;
end
plot(1:240,dd_unquan(1+(21-1)*240:21*240),1:240,dd_quan(1+(21-1)*240:21*
legend('Unquantized','Quantized'); title('Optimal Noise from Codebook')

plot(1:240,ee_unquan(1+(21-1)*240:21*240),1:240,ee_quan(1+(21-1)*240:21*
legend('Unquantized','Quantized'); title('CELP Excitation')

%% Vocoded Speech Plotting
plot(1:240,x(1+(21-1)*240:21*240),1:240,synthiezed_quan(1+(21-1)*240:21*
legend('Unquantized','Quantized'); title('Vocoded Speech')
```