A57074_1, (2024.11.1a_updated, answer for Q15 updated), For year 24-25, cmsc5707

Description After you complete the assignment, you may get a mark (probably 0), it is a dummy result and not the real result because the true answers have not been input to the systems yet. The real result will be published 2 weeks or more after the assignment deadline.

Instructions Hard deadline: late submission is not allowed.

Please click on the "save and submit" button at the end of the page after you completed the assignment.

If your answer is not an integer, give it to the nearest 3 decimal places.
If your answer is an integer, adding decimal points has no harm. E.g. 3.0=3.00=3.000 etc.

Please complete the assignment before the deadline. Multiple attempts are allowed.

A57074.1.1

A speech signal (single channel) is sampled at 44100 Hz and the resolution is 16 bits. How many bytes are used to store [x] seconds of this signals?

answer1: 44100*2*x (ok4)

+/- 1

Range: x= 30->100

//

A57074.1.2

A speech signal is sampled at 44100 Hz . The frame size is [y] ms and the overlapping ratio is 70%. Calculate the number of overlapped samples of adjacent frames. Your answer may not be an integer, give your answer in decimal points, however, our accepted answer is +/-1 sample.

Answer2: ((y/1000)/(1/44100))*0.70 (ok4)

+/- 0.5

range: y=300 -> 600

//

A57074.1.3

A speech signal is sampled at [Z] Hz . A low pass filter (from 0 to X Hz) should be placed between the microphone amplifier and the analogue-to-digital converter to avoid antialiasing (failure to sample). Calculate the value of X.

Answer3: Z/2. (ok4)

+/- 1

range: Z=1000 -> 2000

==

A57074.1.4

The transformation from signal to Cepstral coefficients is achieved by the following 4 processors. Please select the order of execution on the left side of the processor shown below (order 1 is the first being processed etc.) .

Answer4:

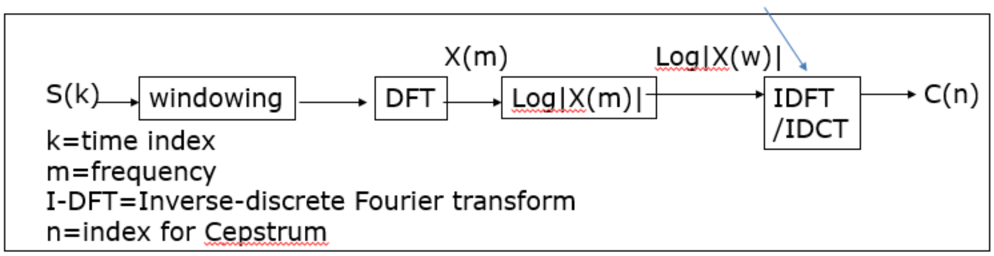
BOX-A, BOX-B, BOX-C, BOX-D =

order 1.windowing,

order 2. DFT,

order 3. LOG|X(m)| ,

order 4. IDFT



==

A57074.1.5

A frame segment of a sound signal X has a sequence of time domain samples:
X=[1,3,6,1,6,7,1,8,3,6,6,7,9,5,8]
The whole sequence is one segment, no frame blocking and framing is required here. You can calculate the auto-correlation parameters r0, r1, r2, r3 and r4 from X. Find (r0 + r1 + r2 + r3 + r4) .

Answer5: 1855 (ok4)

```
clear
clc
x=[1,3,6,1,6,7,1,8,3,6,6,7,9,5,8]
n=length(x)
% r0 -------------------------
for j=1:5
    offset=j-1;
    clear c
    for i=1:n-offset;
        c(i)=x(i)*x(i+offset);
    end
    r(j)=sum(c);
end
r
sum(r)
%r =  497   358   386   346   268
%=        1855
```

==

A57074.1.6

Linear predictive coding LPC coefficients--

A frame segment of a sound signal X has a sequence of time domain samples:
X=[1,3,6,1,6,7,1,8,3,6,6,7,9,5,8]
The linear predictive coding LPC coefficients of order 4 are a1, a2 , a3 and a4. Find a4.

```
%answer6: -0.3653 (ok4)

%+/- 0.001

clear
clc
x=[1,3,6,1,6,7,1,8,3,6,6,7,9,5,8]
n=length(x)
% -------------------------
for j=1:5
    offset=j-1;
    clear c
    for i=1:n-offset;
        c(i)=x(i)*x(i+offset);
    end
    r(j)=sum(c);
end
r0=r(1) %matlab index starts from 1 not 0, so adjust it
r1=r(2)
r2=r(3)
r3=r(4)
r4=r(5)

A=[r0,r1,r2,r3
   r1,r0,r1,r2
   r2,r1,r0,r1
```

```
   r3,r2,r1,r0];

B=[r1,r2,r3,r4]';
lpc_code=inv(A)*B
%sum(lpc_code)
% %lpc_code = [a;a2;a3;a4]=
%     0.3093
%     0.6643
%     0.2406
%    -0.3653
```

///

A57074.1.7

Given a 2-D data set of

```
X=[2.2 1.3
   2.3 0.7
   0.2 0.6
   2.4 5.6
   4.5 7.8
   2.3 2.6
   5.5 5.6
   5.7 8.9
   1.2 3.4
   4.5 4.7];
```

Use the Binary-split K-means method with perturbation e = 0.01 to cluster the data. The first centroid is found to be c =(cx,cy) . In the first stage, the data X is then split into two sets, the centroids of the two sets (set 1 and set2) are (ex1,ey1) and (ex2,ey2), respectively. Find ex1+ey1+ex2+ey2.

Answer7 =14.4000 (ok4)

```
%%%
clc
clear

X=[2.2 1.3
   2.3 0.7
   0.2 0.6
   2.4 5.6
   4.5 7.8
   2.3 2.6
   5.5 5.6
   5.7 8.9
   1.2 3.4
   4.5 4.7];


m=mean(X);
e=0.01;

me1=m*(1+e);
me2=m*(1-e);

N=length(X);
d1=X-repmat(me1,N,1);
d2=X-repmat(me2,N,1);

figure(1)
clf
plot(X(:,1),X(:,2),'+')

jj=1;
kk=1;
for i=1:N
```

```
  dist(i)=norm(d1(i,:))-norm(d2(i,:));
   if (dist(i) >0)
     c1(jj,:)=X(i,:);
     jj=jj+1;
   else
    c2(kk,:)=X(i,:);
     kk=kk+1;
   end
end
mean_c1=mean(c1);
mean_c2=mean(c2);

ex1=mean_c1(1)
ey1=mean_c1(2)
ex2=mean_c2(1)
ey2=mean_c2(2)

ex1+ey1+ex2+ey2


% %Answer:
% ex1 =     1.6400
% ey1 =     1.7200
% ex2 =    4.5200
% ey2 =    6.5200
%
% ans = 14.4000
```

//
+++

A57074.1.8

Dynamic programming

The codes of a reference sound segment, and an unknown input segment are listed below.

```
reference =      [1  5  7  8  4  2  5  1  5  7]; %reference sound segment
input =          [1  4  6  8  4  3  4  2  5  6] %input segment
```

Note: The distortion measure between two sound of code A and B is (A – B)^2.

Use the dynamic programming method to find the optimal distortion between the two sounds.

Answer8= 4 (ok4)

Answer:

```
clear
clc
clf
```

```
close all
N_templates=1

r =     [1 5 7 8 4 2 5 1 5 7]; % reference sound segment
input = [1 4 6 8 4 3 4 2 5 6] %input segment

N_j=length(r)
N_i=length(input)
disp('find error matrix)')
for j=1:N_j
    for i=1:N_i
        % err_abs(j,i)=abs(   input(i)-r(j) ); %use abs
        err_abs(j,i)=(   input(i)-r(j) )^2; %use abs
mse
    end
end
err=err_abs;
err_ud=flipud(err) %for display only

%-------------------------------------------------------
disp('find accumulated error matrix)')
%do first y=1, for all x

%1111111111111111111111111111
%step1: do (1,1)
acc_err(1,1)=err(1,1);

%222222222222222222222222222222
%step2: do i=1 all i
j=1
for i=2:N_i
    acc_err(j,i)=err(j,i)+ acc_err(j,i-1);
end

%3333333333333333333333333333333333
%step3: do i=1 all j
i=1
for j=2:N_j
    acc_err(j,i)=err(j,i)+ acc_err(j-1,i);
end

%4444444444444444444444444444444444444
for j=2:N_j
    for i=2:N_i
        acc_err(j,i)=err(j,i)+ min([...
            acc_err(j-1,i),acc_err(j,i-1),...
            acc_err(j-1,i-1)]);
    end
end
acc_err;
acc_err_ud=flipud(acc_err)
first_row_min=min(acc_err_ud(1,:))
last_column_min=min(acc_err_ud(:,end))

ans=min(first_row_min,last_column_min)
%  acc_err_ud =
%
%    179    50    42    42    26    31    21    37     9     6
%    143    41    41    50    17    15    12    14     5     6
%    127    40    52    76    16    11    13     5    19    28
%    127    31    27    35     7     7     4    12     3     4
%    111    30    26    46     6     3     7     3    12    24
%    110    26    10    18     2     3     3     7     8    12
```

```
%    101     26      6      2     18     37     41     61     39     31
%     52     10      2      3     12     28     25     42     30     27
%     16      1      2     11     12     16     17     26     26     27
%      0      9     34     83     92     96    105    106    122    147

%answer= 4
```

//////////////////////

A57074.1.9 MEL scale

The frequencies of two signals are 220 and [y] Hz. Find the difference between the two signals in MEL scale.

Input y=range= from 500 to 1000
Accepted answer:
+/- 0.5
ANSWER 9 : 2595*log(1+(y/700))-2595*log(1+(220/700)) (ok4)

///////////////////

A57074.1.10

Fourier Transform

A signal `s= [2 1 4 7 2 3 1 5 7 8 5 4 3 6 7 8 7 4 0 2 3 6 ]`

Using Fourier transform shown in the diagram to find ||Xm||, where m=2 (the energy of the frequency at m.) (Hint, you may use the link shown in the diagram to find the
code for the calculation)

The Fourier Transform FT method

(Assume Signal S_k is already smoothed by a hamming window to simplify the discussion)

- Forward Transform *(FT)* of *N* samples: $X_m = FT(s_k)$

$X_{m=0,1.,N-1}(\text{complex } numbers) = FT\{s_{k=0,1,2..,N-1}(\text{real numbers})\}$, *the equation is*

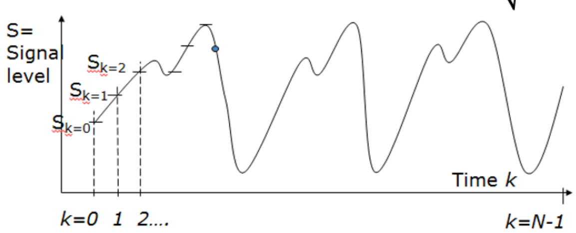
$$X_m = \sum_{k=0}^{N-1} s_k e^{-j\left(\frac{2\pi km}{N}\right)}, m = 0,1,2,3,\ldots,N-1, \text{and } e^{j\theta} = \cos(\theta) + j\sin(\theta), j = \sqrt{-1}$$

Input (is in time domain) $= s_{k=0,1,2,..N-1} = s_0, s_1, s_2, \ldots, s_{N-1,}$(total N samples)

Output (*is in* Frequency domain) after $FT = X_{m=0,}X_{m=1,}X_{m=2,}, \ldots, X_{m=N-1,}$ which are (N) complex numbers

$X_m = \|X_m\|_2 e^{j\theta_m}$,so X_m is complex. $\|\quad\|_2$ =Euclidean **norm**, or 2-norm, i.e.

If $Z = (z_1, z_2, \ldots, x_n)$, then $\|Z\|_2 = \sqrt{z_1^2 + z_2^2 + \ldots + z_n^2}$



Demo Matlab code:

- demo_dft_tutorial.rar
- https://ww2.mathworks.cn/matlabcentral/fileexchange/77789-discrete-fourier-transform

Answer 10: 5.4444 (ok4)
+/- 0.002

```
% assignment question
%Question:
%A signal s= [2 1 4 7 2 3 1 5 7 8 5 4 3 6 7 8 7 4 0 2 3 6 ]
%Using Fourier transform shown in the diagram to find ||Xm||, where m=2 (
%the energy of the frequency at m.)
%(Hint, you may use the link shown in the diagram to find the
%code for the calculation)

%A tutorial to show how Discrete Fourier transform works, khwong 14.9.2016
%Kiss approach: simple and stupid, the purpose is to illustrate the idea
%but not for efficiency, rewrite if you want to use it for serious work.
% http://www.cse.cuhk.edu.hk/~khwong/www2/cmsc5707/5707_02_preprocess.pptx
%Discrete Fourier transform theory tutorial
clearvars
clear
close all
% %[s,fs]=audioread('s1.wav'); %fs=sampling frequency
% %[s,fs]=audioread('A4_oboe.wav'); %fs=sampling frequency%
% %[s,fs]=audioread('A4_oboe_11025.wav'); %fs=sampling frequency
%
% %[s,fs]=audioread('A5_flute.wav'); %fs=sampling frequency
%
% %[s,fs]=audioread('A4_violin.wav'); %fs=sampling frequency
%
% %you may want to resample the signal to other sampling rate, but it is slow
% % [orginal_s,orginal_fs]=audioread('s1.wav'); %fs=sampling frequncy
```

```
% % resample_factor=1
% % fs=orginal_fs/resample_factor
% % s=resample(orginal_s,fs,resample_factor);%resample(s,p,q)%resample p/q times
%
% N=2048%256%example, can change to other numbers such as 512 1024,.. 44100
%
%
% %N is the Fourier window, since sample is fs, if N=fs,the Fourier
% %processing window is it will be one second. E.g. FS=44100, N=Fs=44100, it
% %is 1 second. Proof: time between two samples is Ts=1/44100,then fs*Ts=1
% %sec.
%
% start=11777 %by inspection this frame has vowel (oscillating frequency)
%
% x=s(start:start+(N-1));%obtain a frame of sound, length N (fourier proc.windows)
%
% clear s % make sure you are using the right data


%numerical data

start=1
s= [2 1 4 7 2 3 1 5 7 8 5 4 3 6 7 8 7 4 0 2 3 6 ]
x=s
fs=2 %this is not used for find frequency ,
%but can be used to convert into real freq. in Hz
N=length(s)


%
%for m=0:(N/2) % m is from m=0 to m=N/2 as defined in wiki
for m=0:(N-1) % m is from m=0 to m=N/2 as defined in wiki

    % see https://en.wikipedia.org/wiki/Discrete_Fourier_transform
    clear real_tmp imag_tmp cos_basis sin_basis cos_part sin_part
    real_tmp=0;
    imag_tmp=0;

    %beware matlab matrix index starts from 1 not 0. Fix: +1 to make it work
    %ie, x(0) in the formula is x(1) here in the program.
    for k=0:N-1
        theta=2*pi*k*(m)/N;

        cos_basis(k+1)=cos(theta);%cos_basis(k) starts from k=1 not k=0
        cos_part(k+1)=x(k+1)*cos_basis(k+1); % so as other arrays

        sin_basis(k+1)=sin(theta);
        sin_part(k+1)=x(k+1)*sin_basis(k+1);

        real_tmp=real_tmp+cos_part(k+1); %fr_x=fourier of x
        imag_tmp=imag_tmp+sin_part(k+1); %fr_x=fourier of x
    end
    % pause

    fr_x(m+1)=abs(sqrt(real_tmp^2+imag_tmp^2));

    %    pause
    figure(1)
    clf

    subplot(6,1,1) %-------------- subplot 1 ----------------------------
    plot([0:N-1],x)%show correct index on the xaxis,
    ylabel('Signal x')
```

```
    text1=sprintf('time in samples, N=%d samples, Fs=%5d, time between 2
samples=1/Fs=%0.6f sec.',N,fs,1/fs)
    xlabel(text1);
    % text1=sprintf('m=%d,equivalent freq=%5.2f',m, fs*m/N)    ;
    %title(text1)%    legend(text1)


    subplot(6,1,2)%-------------- subplot 2 ----------------------------
    line([0 0 ],[ 1 -1]), hold on
    plot([0:N-1],cos_basis)% to show correct index of the data
    ylabel('cos');

    subplot(6,1,3)%-------------- subplot 3 ----------------------------
    line([0 0 ],[ 0.5 -0.5]), hold on
    plot([0:N-1],cos_part)% to show correct index of the data
    ylabel('Real');

    subplot(6,1,4)%-------------- subplot 4 ----------------------------
    line([0 0 ],[ 1 -1]), hold on
    plot([0:N-1],sin_basis)
    ylabel('sin');

    subplot(6,1,5)%-------------- subplot 5 ----------------------------
    line([0 0 ],[ 0.5 -0.5]), hold on
    plot([0:N-1],sin_part)
    ylabel('Img')

    subplot(6,1,6)%-------------- subplot 6 ----------------------------
    % plot(fr_x(m+1), fs*m/N,'+')
    plot(fr_x);
    hold on
    % plot([0:(fs/2)/(N/2):fs/2],fr_x);
    text1=sprintf('Freq: x-axis is in m=%d, each m represents Fs/N=%f Hz',m, fs/N)
;
    xlabel(text1)%    legend(text1)
    ylabel('|Y|')
    % pause

end
'answer to question, kit key to continue'
fr_x
'show result_1 fr_x(m=2)'
fr_x(2+1) %for m=2, index starts from 0

pause
figure(2)
clf
'test6'
size(fr_x)
%[1:(fs/2)/N:fs/2]
%size([1:1+((fs/2)/(N/2)):fs/2])
%pause
%plot(fr_x)
%plot([1:(fs/2)/(N/2):fs/2],fr_x);%each m unit is Fs/N Hz, from 0 to fs/2 Hz.
plot([0:(fs/2)/(N/2):fs/2],fr_x);%each m unit is Fs/N Hz, from 0 to fs/2 Hz.

title('power spectrum  of discrete fourier transofrm')
%text1=sprintf('m=%d,equivalent freq=%f',m, fs*m/N)    ;
text2=sprintf('In Hz, fs=%d',fs);
xlabel(text2);
[cc,ii]=max(fr_x)
%multiple Fs/N is to convert to frequency fro m m
'max frequceny is '
```

```
max_freq=(ii-1)*fs/N% minus 1 is because the index has been increased,

figure(3)% compare with fft() in matlab, seems to be ok,>> help ftt
clf
%%%%%%%%%%%%%%%%
Fs=fs;
y=x;
NFFT = N%2^nextpow2(L); % Next power of 2 from length of y
L=N;
Y = fft(y,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
plot(f,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')

% 5.4444
```

//

A57074.1.11

speech signal frame blocking

A speech signal is 2.25 seconds in duration. A frame blocking method is applied to obtain the frames, the frame size is 30ms and non-overlapping region is 20ms. Estimate how many frames (+/-1 accuracy allowed) will you obtain for this signal.

Answer11: 111 (ok4)

Answer range: +/- 2

(2250-30)/20=111 frames

//

A57074.1.12

: MFCC (multiple answer question)

Which of the following statement is/are true for MFCC Mel-frequency cepstrum coefficients.

1) The term MFCC0 (index 0) is smaller than other MFCC terms.
2) The term MFCC0 (index 0) is bigger than other MFCC terms.
3) The term MFCC0 (index 0) represents the highest frequency term of the signal.
4) MFCC0 (index 0) is not used in speech recognition because it represents the energy term and irrelevant to the frequency content of the signal.

Answer 12: True 2,4 (ok4)

A57074.13 Dynamic programming, ch5

An accumulated distortion score matrix (using the format same as the lecture notes) of matching 2 sequences is shown in the table below.

The optimal distortion score of this accumulated distortion score matrix is P. The number of elements of the optimal path is N (including the first and last elements of the path). Find N+P. Hints: The diagonal path is preferred (see lecture notes for the definition), and assume we do not apply any restricted region in the optimal search.

60	60	69	70	34	49	37	46	26	26
59	71	107	91	30	24	21	25	22	46
55	71	99	75	29	20	21	21	21	61
39	35	39	39	20	38	27	39	12	13
35	51	79	47	11	2	3	3	12	61
19	15	31	11	2	11	15	24	24	40
18	6	10	2	6	31	47	72	76	75
17	5	1	5	30	79	89	120	95	71
1	1	10	11	15	40	56	75	70	71
0	4	20	24	25	41	50	66	67	76

Answer13: 23 (ok4)

+/- 0

P=13

N=10

Optimal path is underlined

60	60	69	70	34	49	37	46	26	26
59	71	107	91	30	24	21	25	22	46
55	71	99	75	29	20	21	21	21	61
39	35	39	39	20	38	27	39	12	$\underline{13}$
35	51	79	47	11	$\underline{2}$	$\underline{3}$	$\underline{3}$	$\underline{12}$	61
19	15	31	11	$\underline{2}$	11	15	24	24	40
18	6	10	$\underline{2}$	6	31	47	72	76	75
17	5	$\underline{1}$	5	30	79	89	120	95	71
1	$\underline{1}$	10	11	15	40	56	75	70	71
$\underline{0}$	4	20	24	25	41	50	66	67	76

%code

clear

clc

clf

close all

N_templates=1

```
r=      [5 6 9 6 4 1 7 1 3 6]; %reference
input = [5 7 9 7 4 1 2 1 4 8] %input
N_j=length(r)
N_i=length(input)
disp('find error matrix)')
for j=1:N_j
    for i=1:N_i
        % err_abs(j,i)=abs(   input(i)-r(j) ); %use abs
        err_abs(j,i)=(   input(i)-r(j) )^2; %use abs mse
    end
end
err=err_abs;
err_ud=flipud(err) %for display only

%-----------------------------------------------------
disp('find accumulated error matrix)')
%do first y=1, for all x

%1111111111111111111111111111
%step1: do (1,1)
acc_err(1,1)=err(1,1);

%2222222222222222222222222222222
%step2: do i=1 all i
j=1
for i=2:N_i
    acc_err(j,i)=err(j,i)+ acc_err(j,i-1);
end

%3333333333333333333333333333333333
%step3: do i=1 all j
```

```
i=1
for j=2:N_j
    acc_err(j,i)=err(j,i)+ acc_err(j-1,i);
end

%4444444444444444444444444444444444444444
for j=2:N_j
    for i=2:N_i
        acc_err(j,i)=err(j,i)+ min([...
            acc_err(j-1,i),acc_err(j,i-1),...
            acc_err(j-1,i-1)]);
    end
end
acc_err;
acc_err_ud=flipud(acc_err)
first_row_min=min(acc_err_ud(1,:))
last_column_min=min(acc_err_ud(:,end))

ans=min(first_row_min,last_column_min)

% acc_err_ud =
%
%    60    60    69    70    34    49    37    46    26    26
%    59    71   107    91    30    24    21    25    22    46
%    55    71    99    75    29    20    21    21    21    61
%    39    35    39    39    20    38    27    39    12    13
%    35    51    79    47    11     2     3     3    12    61
%    19    15    31    11     2    11    15    24    24    40
%    18     6    10     2     6    31    47    72    76    75
%    17     5     1     5    30    79    89   120    95    71
%     1     1    10    11    15    40    56    75    70    71
%     0     4    20    24    25    41    50    66    67    76
```

////////////////////// q13-16 are programming exercises ////////////

A57073.1.14 :

Autocorrelation programming exercise --

You need to use MATLAB for the programming exercises.

Obtain the data from http://www.cse.cuhk.edu.hk/~khwong/www2/cmsc5707/data57074_1.zip
Unzip the files and follow the instructions in readme4b.docx. Run the code "demo_sign1_question".

You will obtain a signal frame "x_frame", the autocorrelation parameters of x_frame are r0,r1,r2,r3,r4 etc. Find r2.

Answer14: 3.5989(ok4)

```
function demo_sig1()
clear
clc
clf
close all
fname='s4.wav';


start=20000; %starting location (sample)
frame_size=1024; % one frame
[x_windowed,x_raw,fs]=read_x1(fname,start,frame_size);
size(x_windowed)
size(x_raw)
%%fill in you code from here
size(x_raw)
'you may use x\_windowed for assignments'

%%%%% answer to be filled %%%%%%
lpc(x_windowed,8) %just for interest
N=length(x_windowed)
p=2
temp=0
for i=1:N-8
    temp=temp+x_windowed(i)*x_windowed(i+p);
end
'auto correlation of order '
p
'is'
temp
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% get data for assignments: 57074.1, you may use
% matlab online: https://ww2.mathworks.cn/en/products/matlab-online.html
% https://ww2.mathworks.cn/en/support/learn-with-matlab-tutorials.html
function [x_windowed,x_raw,fs]=read_x1(fname,start,frame_size)


[sig,fs]=audioread(fname);%x=data,fs=sampling_rate,nbits=num_bits
sound(sig,fs);%you can listen to the sound

x_raw=sig(:,1); %get one channel from stereo sound incase it is stereo
x_frame=x_raw(start:start+frame_size-1);
figure(1)
plot(x_frame)
ylabel("x raw")
xlabel("time samples")

figure(2)
clf
subplot(3,1,1);
hamm=hamming(frame_size);
plot(x_frame)
ylabel("x\_raw")
xlabel("time samples")

subplot(3,1,2);
hamm=hamming(frame_size);
plot(hamm)
ylabel("hamming")
xlabel("time samples")

subplot(3,1,3);

x_windowed=hamm.*x_frame;
plot(x_windowed)
ylabel("x\_frame windowed")
xlabel("time sample")

'you may use x1\_frame (windowed) and s\_all for assignments'




p =

     2


temp =

     0
```

```
ans =

    'auto correlation of order '


p =      2
temp =    3.5989
```

//

A57074.1.15: MFCC programming exercise--

Obtain the data from http://www.cse.cuhk.edu.hk/~khwong/www2/cmsc5707/data57074_1.zip

Study the matlab documentation of the function “mfcc”. Run the following code

%% matlab code starts here %%%%%%%%%%%%%%%%%%%%%%%%%%

clear

[audioIn,fs] = audioread("s4.wav");

%the source .wav sound is stereo, the following uses one channel of the sound

[coeffs,delta,deltaDelta,loc] = mfcc(audioIn(:,1),fs);

% If W=window size in samples

% M=overlapping sample

% N=number of frames obtained.

% P=number of MFCC parameters obtained.

Find W+M+N+P . Hints: +/- 3 is acceptable.

Answer15: 2294 (updated 2024.11.1)

+/-3

Answer: read the workspace windows of the matlab windows(located at left-bottom window). We found that

Variable loc : 1323,1764,2205,….

~~Overlapping window size M=1764-1323=441. End of frame 1 and end2 are 1323 and 1764 respectively.~~

By reading Loc, we see that the first frame is from 1 to 1323. Hence frame_size=1323.

Non-overlapping window size Nonoverlapping_size =1764-1323=441. Because ends of frame 1 and end2 are 1323 and 1764 respectively.

M=frame_size - Nonoverlapping_size =1323-441=882

Size of coeffs is 76x14, that means there are N=76 frames and P=13 parameters because the first is LogEnergy.

In fact for the coeffs, the first column is the LogEnergy, then mfcc0,mfcc1,…mfcc12 . There are 13 mfcc parameters. You nay run the code :

coeffs2 = mfcc(audioIn(:,1),fs,"LogEnergy","Ignore");

To verify it.

W=1323 , M=882, N=76, P=13, W+M+N+P=2294

If you answer P=13 or P=14 are also considered correct since the coefficients have 14 parameters. Therefore I set the range to be more flexible, i.e. +/- 3. See above.

//

A57074.1.16: MFCC programming exercise--

MFCC programming exercise--

Based on the above questions, extraction

xa= the 40-th frame of sound file "s4.wav"

xb= the 50-th frame of sound file "s5.wav"

Find the mfcc_distortion between the MFCC vectors of xa and xb.

Hints:

(1) In the output "coeffs" of the MATLAB mfcc function, the n-th frame output is the n-th row of coeffs.

E.g. the 40th row of coeffs is the mfcc vector of frame 40 of the sound file.

(2) Assume the two mfcc vectors (coeffs) are [LogEnergy_m0, m0,m1,m2…,m12], and [LogEnergy_n0,n0,n1,n2,..,n12]. The mfcc_distortion=sqrt [(m1-n1)^2+(m2-n2)^2+,…,+(m12-n12)^2]. Note that LogEnergy and m0 and n0 terms are not used in the calculation. See details in

https://ww2.mathworks.cn/help/audio/ref/mfcc.html?overload=mfcc+false

(3) % How to index a matrix in matlab,

```
M=[1 2 3
   4 5 6
   7 8 9]
```

M(1,1) %=1

M(2,2) %=5

M(1,3) %=3

Answer16: 3.92 (ok4)
+/- 0.02

```
clear
clc
clf
[s4,fs] = audioread("s4.wav");
%the source .wav sound is stereo, the following uses one channel of the sound
[coeffs_s4,delta,deltaDelta,loc] = mfcc(s4(:,1),fs);

[s5,fs] = audioread("s5.wav");
%the source .wav sound is stereo, the following uses one channel of the sound
[coeffs_s5,delta,deltaDelta,loc] = mfcc(s5(:,1),fs);

vec_s4=coeffs_s4(40,3:14)
vec_s5=coeffs_s5(50,3:14)

result=norm(vec_s4-vec_s5)

%use of norm tutorial
v=[1, 2, 4]'
result1=norm(v) %=4.5826
result2=sqrt((1^2+2^2+4^2))%=4.5826
%result1 , result2 are the same

%3.92
```

//////////////////////////////////////