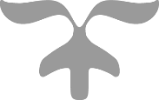
21AIE315 : AI IN SPEECH PROCESSING



**EPOCH ESTIMATION**

USING DMD



TEAM 11

CB.EN.U4AIE19014 – **ARUN PRAKASH J**

CB.EN.U4AIE19016 – **ASSWIN C R**

CB.EN.U4AIE19024 – **DHARSHAN KUMAR K S**

CB.EN.U4AIE19028 – **AVINASH DORA**

GUIDED BY

**MR. JYOTISH LAL G**



BTECH CSE-AI

CEN DEPARTMENT

AMRITA SCHOOL OF ENGINEERING

**ACKNOWLEDGEMENT**

This project would not have been possible without the support and guidance of Mr. Jyothish, who has provided us with sufficient knowledge and guidelines.

We would also like to thank our Computer Science and Engineering (Artificial Intelligence) – CEN department for giving us this opportunity to improve our skills.

Furthermore, we would like to thank the Amrita Vishwa Vidyapeetham management for providing ample resources to avail our project needs.

Our thanks and appreciations also go to colleagues in developing the project and people who have willingly helped us out with their abilities unconditionally.

**TABLE OF CONTENTS**

1. [ABSTRACT](#_Toc43146547) 4

2. INTRODUCTION5

3. [LITERATURE REVIEW](#_Toc43146551) 6

4. [NOVELTY](#_Toc43146551) 7

5. [PROPOSED METHOD USING DMD](#_Toc43146551) 8

6. [DMD](#_Toc43146551) 10

7. [RESULT AND DISCUSSION](#_Toc43146551) 13

8. [CONCLUSION](#_Toc43146551) 14

9. [MATALB CODE](#_Toc43146551) 15

10. [REFERENCES](#_Toc43146551) 25

**ABSTRACT**

This is a report based on end semester 6 project, Epoch Detection using DMD under the subject “AI IN SPEECH PROCESSING” – 19AIE302.

The project begins with a brief introduction towards the Epoch Estimation.

The main objective of this paper is to estimate the epoch or GCI (Glottal Closure Instant) locations from the speech signal using a decomposition method, DMD (Dynamic Mode Decomposition) which assumes the speech production system as non-linear. This paper also contributes in estimating instantaneous pitch frequencies of the modes obtained from DMD. DMD algorithm captures the center frequency close to the fundamental frequency defined for each glottal cycle of speech utterances through its modes. Based on the obtained mode frequency, a sine wave will be constructed and negative-to-positive zero-crossings will be estimated as epoch locations. Experimental result shows that the DMD approach is very effective in identifying the epoch locations more accurately when comparing with the existing methods.

**INTRODUCTION**

The lateral vibration of the vocal folds modulates the air from the lungs and generates the carrier signal of speech. The movement of the vocal folds is quasi-periodic for normal or modal voice. During the production of voiced sound, the significant excitation which occurs in the vocal tract is known as Epoch or GCI (Glottal Closure instance). The instantaneous pitch is used as a metric in various speech applications. For a given signal, the pitch is the inverse of the time interval between successive epoch locations. The accurate estimation of epoch locations is required for finding the correct pitch of the speech signal.

Due to the varying nature of the excitation characteristics in speech signal, many existing algorithms for epoch detection show degraded performance. The main challenge in the epoch detection task is the interaction of vocal tract response with the excitation signal, which causes the influence of various vocal tract frequencies mixed with the excitation signal. Effective removal of vocal tract influence is needed for a robust epoch estimation system. Hence, our work focuses on decomposition of speech signal into different modes using DMD in which the mode which closely resembles the excitation signal will be chosen for epoch estimation.

**Dataset Description:**

CMU Arctic dataset was chosen for performing our experiments which consists of 1150 utterances of both male and female speakers in US English language. Both speech signals as well as EGG signals were provided in the dataset and was recorded in 16 bit and sampling frequency of 32Khz.

**LITERATURE REVIEW**

Analysis of speech signals has become a field of interest due to its wide range of applications in classifications of disorders [1,2,3,4,5,6], emotion classification [7,8,9,10], classification of gender and age [11,12,13]. Speech signals are excitations caused by the vibrations in the vocal folds. These quasiperiodic excitations have regions with significant excitations called epochs. The estimation and analysis of epochs has great importance in all the above-mentioned applications. Epochs are also called glottal closure instances (GCIs). However, this estimation is not an easy task due to interaction of excitations and vocal tract resonant frequencies.

One of the existing methods for epochs estimation include using linear prediction analysis on the extracted residual signal from the speech signal [14,15]. The limitation with this approach is bipolar nature of peaks present in the linear prediction residual. A solution to this problem was to introduce Hilbert envelop for unambiguous epoch estimation [16]. The limitations of linear prediction-based approach are due to the major influence of the vocal tract system.

Zero crossing of the phase slope of the residual LP is another method flowed by the use of the dynamic programming phase slop algorithm [17,18,19]. YAGA [18] applies the phase slope on the wavelet transform of the source signal to detect the epochs. ZFF (zero frequency filtering) based method uses the impulse excitation. ZFFs identify the locations of positive zero crossings as the needed locations.

The SEDREAMS algorithm uses the average pitch period of the mean of the speech signal [19]. LP analysis is performed on this output to obtain the epoch locations. MMF is another technique that uses the locations of lower singularity exponents to find the epoch locations [20]. The GEFBA algorithm uses the structure of the glottal flow to estimate epochs in the voice speech frame in a given speech signal [21].

In most of the beforementioned methods, more than one epoch location is found in one glottal cycle, which is followed by a candidate selection procedure. This motivates the need for a more robust method which correctly estimates the epoch locations.

**NOVELTY OF OUR WORK**

Most of the well-known epoch extraction methods assume the speech signal to come from a linear model. The performance of epoch estimation in these methods will be affected since this assumption is not appropriate for the speech signal which is produced due to non-linear interaction of excitation signal with vocal tract responses. So, our proposed model using DMD assumes the speech signal to come from a non-linear speech production system and so accurate estimation of epoch locations can be obtained. Other methods that relay on this assumption of non-linear speech production is Empirical Wavelet Transform (EWT), empirical mode decomposition (EMD) and variable mode decomposition (VMD) [10]. Using DMD has its advantage over VMD, since in epoch estimation work done by Jyothish et al. [24] uses iterative method of Variable Mode Decomposition. This may cause mode frequencies missing and can’t effectively get our desired mode for epoch extraction. Our proposed method which uses DMD does only single decomposition which is computationally lighter than the VMD pipeline proposed by [24].

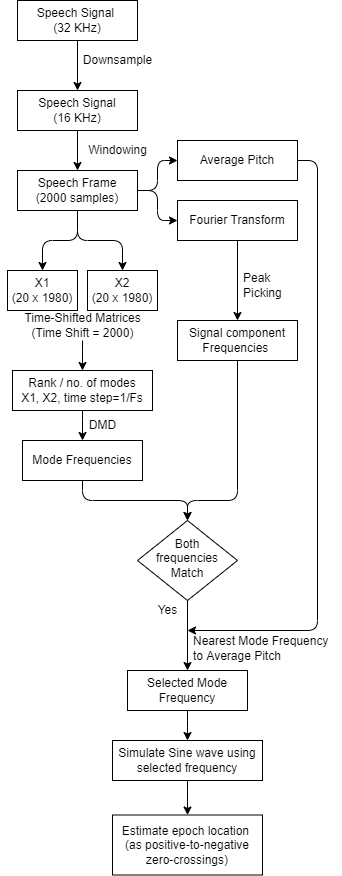


Fig. 2: Pipeline for VMD methodology

**PROPOSED METHOD USING DMD**

The proposed framework is illustrated in Fig. 1. Two phases were illustrated in the framework. Phase 1 is to find the instantaneous pitch frequencies for each mode obtained from DMD. Phase 2 uses the obtained pitch frequency of modes to simulate a sine wave which closely resembles the EGG signal. Epoch locations will be estimated from the positive-negative zero crossings of the sine wave and the various metrics needed for further speech signal analysis will be calculated based on the estimated epoch locations.

In first phase of the pipeline, we will follow a window-based approach where the whole speech signal is divided into non-overlapping speech frames. The sample length of 2000 was found to give best results through experiments. The 1-dimensional signal vector will be converted into a 2-dimensional matrix by time-shifting as a pre-processing step before sending it into DMD algorithm. The speech signal vector will be shifted by 1 unit for each row of the time-shifted matrix. Each row of the matrix will contain ‘N-s’ values in it, where ‘N’ is the speech signal vector length and ‘s’ is the time-shift coefficient value. Through various experiments, we found that ‘s’ value of 200 and rank value of 6 gave the best results. We will find the all the sub-signal components of our speech frame through its linear spectrum using Fourier transform. If the frequencies obtained through DMD approximately matches with the frequencies obtained through linear spectrum, then we choose the DMD mode frequency nearest to the average pitch of the speech signal. So, this method requires information of average pitch to find the final mode frequencies corresponding to the excitation signal. Epoch locations can be estimated from the constructed sine wave with final mode frequency.

Fig. 1: Pipeline for proposed methodology

**DYNAMIC MODE DECOMPOSITION (DMD)**

**Dynamic Mode Decomposition:**

Dynamic mode decomposition is a versatile and strong matrix decomposition technique that originated in fluid dynamics and nonlinear waves and developed by peter Schmid [25]. It was developed in order to comprehend, control, or simulate inherently complex, nonlinear systems without necessarily knowing the underlying governing equations that drive the system completely or partially. When we see DMD in the last few years alone, it has made immense progress in both theory and application. The DMD algorithm provides spatio-temporal decomposition of data into its corresponding dynamic modes via the snapshots of a given system and has a connection to non-linear dynamics of the system through Koopman operator which helps in mapping time-delayed snapshots. DMD produces a least-square regression to reduce the high-dimensional system to lower dimensional system in the form of DMD modes and its corresponding eigenvalues. The DMD modes represents the coherent spatial structure of the data and its corresponding eigen values shows the systems evolving behavior .There are two methods for obtaining these eigenvalues and modes. The first is Arnoldi-like, which is useful for theoretical analysis due to its connection with Krylov methods. The second is a singular value decomposition (SVD) based approach that is more robust to noise in the data and to numerical errors. The procedure involved in DMD algorithm is explained as follows.

**Algorithm:**





**Algorithm Explanation:**

Step 1: Collect the Data.

Step 2: The Data is been organized into 2 matrices, X1 and x2. the columns of X1 are the snapshots of the data reshaped into very tall column vectors which evolve in time. Hence the columns are evolving in time along with the dynamics of the system.

Step 3: Next X2 is the same Data Matrix, but advanced one-time step in the future (shifted 1 delta T ahead).

Step 4: Now, DMD finds the best fit linear operator A, that maps X2 and X1 and can be expressed as mentioned in eqn(1)

Step 5: DMD approximates the leading(dominating) eigen values and eigen vectors of the A matrix.

Step 6: once we obtain the eigen vectors of the A matrix, it will have the same shape as that of one of the columns of X1 and can be reshaped into a flow field (eigen flow field) – dominant flow field coherent modes.

Step 7: Thus, we obtain the leading eigen decomposition gives the spatial temporal modes and the time dynamics (eigen values).

Step 8: Hence by knowing the leading eigen decomposition of A, we can predict how the system will evolve in the future.

**RESULTS AND DISCUSSION**

A single speech frame of 2000 samples were taken for this approach and sine wave was constructed for this frame according to the obtained mode frequency. The sine waves corresponding to all the speech frames of the signal were concatenated together to form the simulated excitation signal from which the epoch locations can be extracted from the negative-to-positive zero-crossings.

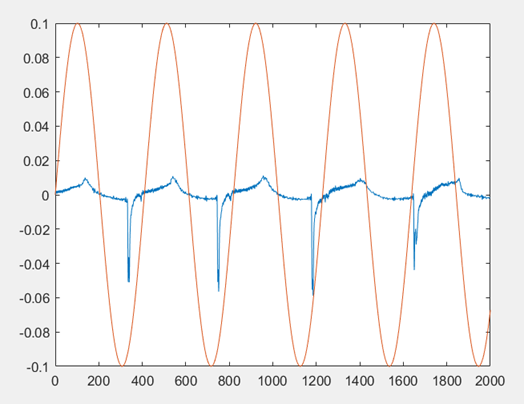


Fig. 2: Sine wave (orange) and DEGG (blue)

signal of single speech frame

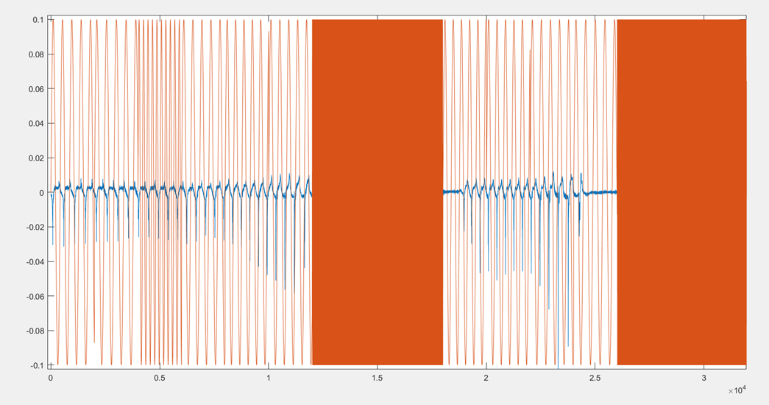


Fig. 3: Sine wave of speech signal

From the Fig. 2, we can see that the epoch locations (negative peaks) in DEGG signal closely resembles the epoch locations (negative-to-positive) in sine wave. In Fig. 3, we can see multiple speech frames concatenated together for a whole speech frame. The low frequency regions correspond to the voiced region from which epoch estimation can be done and the high frequency regions corresponds to the unvoiced regions.

**CONCLUSION**

We show that the application of DMD algorithm iteratively on the speech signal helps to capture the required center frequency of the mode. The center frequency of the selected mode is found to be near to the fundamental frequency of the glottal excitation signal. Accurate estimation of epoch locations is possible due to this center frequency characteristic of the DMD mode. We compared the effectiveness of the proposed method with the state-of-the-art epoch estimation methods and found that our proposed method shows better results than the existing methods. This work would be extended to epoch estimation in emotional speech signal using our proposed approach where the problem of low accurate predictions from existing methods in highly aroused emotions such as anger and happiness may be solved.

**MATLAB CODE**

clc;

clear all;

close all;

% Time domain representation of signal

path = "C:\Users\Dharsankumar\Desktop\Speech paper\Codes\cmu\_us\_bdl\_arctic-WAVEGG\cmu\_us\_bdl\_arctic\orig\";

[y1, Fs] = audioread(path+"arctic\_a0002.wav"); %reading hte wav file

sp1=y1(:,1); % Signal with 225920 samples

egg1=y1(:,2);

sp=resample(sp1,Fs,16000); % Resample to 16KHz

egg=resample(egg1,Fs,16000); % Resample to 16KHz

Fs=16000;

all\_freqs = [];

start\_limit = 20000;

end\_limit = 52000;

for k=start\_limit:2000:end\_limit-2000

range1 = k

range2 = k+2000

final\_mode\_freq = simulate\_signal(sp,egg,range1,range2, Fs);

all\_freqs = [all\_freqs final\_mode\_freq];

end

%% Plot sine wave for whole signal

fs = 16000; % Sampling frequency (samples per second)

dt = (1/fs); % seconds per sample

StopTime = ((62.5\*2\*length(all\_freqs))/1000); % seconds % Change '2'

t = (0:dt:StopTime)'; % seconds

data\_all = [];

for i=1:length(all\_freqs)

final\_mode\_freq = all\_freqs(i);

data = find\_sine(final\_mode\_freq);

data\_all = [data\_all; data];

end

data\_all = data\_all(1:length(data\_all)-length(all\_freqs)+1);

figure;

egg\_part = egg(start\_limit:end\_limit);

plot(diff(egg\_part));

hold on;

plot( t\*100000/6.25 , data\_all );

%%

k = 24000;

range1 = k

range2 = k+2000

final\_mode\_freq = simulate\_signal(sp,egg,range1,range2, Fs);

function [final\_mode\_freq] = simulate\_signal(sp,egg,range1,range2, Fs)

sp = sp(range1:range2-1); % Take 2000 samples from full signal

egg = egg(range1:range2-1);

% pitch

%-------

% Calculate pitch for the speech signal

fx=fxrapt(sp,16000);

avg\_F0 = mean(fx)

if(isnan(avg\_F0)==1)

%final\_mode\_freq = 1000;

% If first method fails, use 2nd method

fx\_pitch = pitch(sp,16000);

avg\_F0\_method2 = mean(fx\_pitch);

if((50>avg\_F0\_method2)&&(avg\_F0\_method2>150))

final\_mode\_freq = 1000

return;

end

end

fx\_pitch = pitch(sp,16000);

avg\_F0\_method2 = mean(fx\_pitch)

% Spectrum

%---------

y=sp;

L=length(y);

NFFT = 2^nextpow2(L);

seg\_fft = fft(y,NFFT);

z = seg\_fft(1:1+NFFT/2);

z= abs(z(1:NFFT/2+1));

f\_scale = (0:NFFT/2)\* Fs/NFFT;

figure;

plot(f\_scale,z);

% Estimate peaks from spectrum

%-----------------------------

if(max(z)>70)

cut = 40;

else

cut = 20;

end

[pks,locs] = findpeaks(z,'MinPeakHeight',max(z)-cut);

findpeaks(z,'MinPeakHeight',max(z)-cut)

if(length(locs)<3)

cut = 50;

[pks,locs] = findpeaks(z,'MinPeakHeight',max(z)-cut);

end

if(length(locs)<3)

final\_mode\_freq = 1000

return;

elseif(pks(1)<10)

final\_mode\_freq = 1000

return;

end

% Pick frequencies with more amplitude

[M1,I1] = sort(mink(pks,3));

locs\_new = [];

for i=1:length(I1)

locs\_new = [locs\_new locs(I1(i))];

end

locs = locs\_new;

needed\_freq = [];

for i=1:length(locs)

needed\_freq = [needed\_freq f\_scale(locs(i))];

end

needed\_freq = sort(maxk(needed\_freq,3)) % Pick top 3 frequnecy values

% Create snapshots for DMD

%-------------------------

X1=[];

Xa=[];

Xb=[];

window\_length=20; % 'k' value

for i=1:window\_length+1

X1 =[X1;reshape(sp(i:i+length(sp)-window\_length-1),[1,length(sp)-window\_length])];

end

Xa=X1(1:window\_length,:); % Taking 1st column till 'n-1'th column of time-shifhted 'X' matrix

Xb=X1(2:window\_length+1,:); % Taking 2nd column till 'n'th column of time-shifhted 'X' matrix

X1=Xa;

X2=Xb;

% Dynamic mode decompsition

%--------------------------

% STEP 1: singular value decomposition (SVD)

r = 6; % rank-r truncation

[U, S, V] = svd(Xa, 'econ');

Ur = U(:, 1:r);

Sr = S(1:r, 1:r);

Vr = V(:, 1:r);

% STEP 2: low-rank subspace matrix

% (similarity transform, least-square fit matrix, low-rank subspace matrix)

Atilde = Ur'\*X2\*Vr\*Sr^(-1);

% STEP 3: eigen decomposition

% W: eigen vectors

% D: eigen values

[W, D] = eig(Atilde);

% STEP 4: real space DMD mode

Phi = X2\*Vr\*Sr^(-1)\*W; % DMD modes

dt=1/Fs;

lamdba = diag(D); % eigen value

omega = log(lamdba)/dt; % log of eigen value

Om=log(diag(D))/(2\*pi\*dt);

Om2=abs(imag(Om));

%figure;

%plot(real(Phi));

% Keep window length fixed & change for different ranks

%-------------------------------------------------------

% Window length w1

window\_length = 300;

[X1,X2] = DMD\_v2(sp, window\_length);

Xa = X1;

% Loop - wl is constant & no. of mode varies

rank1 = 5;

rank2 = 120;

all\_ranks = zeros(rank2-rank1,window\_length);

for r=rank1:rank2

[U, S, V] = svd(Xa, 'econ');

Ur = U(:, 1:r);

Sr = S(1:r, 1:r);

Vr = V(:, 1:r);

Atilde = Ur'\*X2\*Vr\*Sr^(-1);

[W, D] = eig(Atilde);

Phi = X2\*Vr\*Sr^(-1)\*W; % DMD modes

dt=1/Fs;

lamdba = diag(D); % eigen value

omega = log(lamdba)/dt; % log of eigen value

Om=log(diag(D))/(2\*pi\*dt);

Om2=abs(imag(Om));

for j=1:length(Om2)

all\_ranks(r-rank1+1,j) = Om2(j);

end

end

% Get mode frequencies

%---------------------

s = size(all\_ranks);

mode\_freq = [];

for i=1:s(1)

for j=1:s(2)-4-1

temp = [];

if(needed\_freq(1)-20 < all\_ranks(i,j)) && (all\_ranks(i,j) < needed\_freq(1)+20)

if(needed\_freq(2)-20 < all\_ranks(i,j+2)) && (all\_ranks(i,j+2) < needed\_freq(2)+20)

if(needed\_freq(3)-20 < all\_ranks(i,j+4)) && (all\_ranks(i,j+4) < needed\_freq(3)+20)

temp = [temp ;all\_ranks(i,j)];

temp = [temp ;all\_ranks(i,j+2)];

temp = [temp ;all\_ranks(i,j+4)];

Distance = euclidean(temp,needed\_freq);

temp = [temp; sum(sum(Distance))]; %Get euclidean distance

mode\_freq = [mode\_freq temp];

end

end

end

end

end

if(length(mode\_freq)==0)

for k=1:10

window\_length = window\_length + 50;

[all\_ranks] = get\_all\_ranks(window\_length, X1, X2, sp);

[mode\_freq] = get\_freq(all\_ranks,needed\_freq);

if(length(mode\_freq)>0)

break

end

end

end

% Choose the frequency values with least euclidean distance

[M,I] = min(mode\_freq(4,:));

mode\_freq = mode\_freq(:,I);

mode\_freq = mode\_freq(1:length(mode\_freq)-1)

% Find final mode

%----------------

% Close to signal pitch

dist = [];

for i=1:length(mode\_freq)

Distance = euclidean([mode\_freq(i),1],[mode\_freq(i),1]);

dist = [dist Distance(1,2)];

end

[M,I] = min(dist);

final\_mode\_freq = mode\_freq(I)

% Simulate sine wave

%-------------------

%egg=y1(:,2);

%egg = egg(29000:32000-1);

%egg = egg(30000+70:32000-1+70);

figure;

%subplot(2,1,1)

%plot(egg)

%hold on;

plot(diff(egg));

fs = 16000; % Sampling frequency (samples per second)

dt = (1/fs); % seconds per sample

StopTime = ((62.5\*2)/1000); % seconds % Change '2'

%StopTime = 1;

t = (0:dt:StopTime)'; % seconds

t\_0 = cat(1,zeros(100,1), t);

%F = 50; % Sine wave frequency (hertz) % Change 'F'

F = final\_mode\_freq;

data = 0.1\*sin(2\*pi\*F\*t);

data\_0 = cat(1,zeros(100,1), data);

%figure;

hold on;

plot( t\_0\*100000/6.25 , data\_0 )

end

%% Functions

function [X1,X2] = DMD\_v2(sp, window\_length)

X1=[];

Xa=[];

Xb=[];

%window\_length=20; % 'k' value

for i=1:window\_length+1

X1 =[X1;reshape(sp(i:i+length(sp)-window\_length-1),[1,length(sp)-window\_length])];

end

Xa=X1(1:window\_length,:); % Taking 1st column till 'n-1'th column of time-shifhted 'X' matrix

Xb=X1(2:window\_length+1,:); % Taking 2nd column till 'n'th column of time-shifhted 'X' matrix

X1=Xa;

X2=Xb;

end

function pitch1 = find\_pitch(range11,range22)

diff1 = range22-range11;

diff1\_in\_ms = diff1/16;

pitch1 = (1/diff1\_in\_ms)\*1000;

end

function Distance = euclidean(x,y)

Distance = zeros(length(x) , length(y));

for i = 1:length(x)

for j = 1:length(y)

if i ~= j

Distance(i,j) = sqrt((x(i)-x(j))^2 + (y(i)-y(j))^2);

end

end

end

end

function [data]=find\_sine(final\_mode\_freq)

% Simulate sine wave

%-------------------

fs = 16000; % Sampling frequency (samples per second)

dt = (1/fs); % seconds per sample

StopTime = ((62.5\*2)/1000); % seconds % Change '2'

%StopTime = 1;

t = (0:dt:StopTime)'; % seconds

t\_0 = cat(1,zeros(100,1), t);

%F = 50; % Sine wave frequency (hertz) % Change 'F'

F = final\_mode\_freq;

data = 0.1\*sin(2\*pi\*F\*t);

end

function [all\_ranks] = get\_all\_ranks(window\_length, X1, X2, sp)

Fs=16000;

% Keep window length fixed & change for different ranks

%-------------------------------------------------------

% Window length w1

%window\_length = 300;

[X1,X2] = DMD\_v2(sp, window\_length);

Xa = X1;

% Loop - wl is constant & no. of mode varies

rank1 = 5;

%rank2 = 20;

rank2 = 120;

all\_ranks = zeros(rank2-rank1,window\_length);

for r=rank1:rank2

%r = 5; % rank-r truncation

[U, S, V] = svd(Xa, 'econ');

Ur = U(:, 1:r);

Sr = S(1:r, 1:r);

Vr = V(:, 1:r);

Atilde = Ur'\*X2\*Vr\*Sr^(-1);

[W, D] = eig(Atilde);

Phi = X2\*Vr\*Sr^(-1)\*W; % DMD modes

dt=1/Fs;

lamdba = diag(D); % eigen value

omega = log(lamdba)/dt; % log of eigen value

Om=log(diag(D))/(2\*pi\*dt);

Om2=abs(imag(Om));

%all\_ranks = [all\_ranks Om2];

for j=1:length(Om2)

%disp(j);

all\_ranks(r-rank1+1,j) = Om2(j);

end

%all\_ranks = horzcat(all\_ranks,Om2);

end

end

function [mode\_freq] = get\_freq(all\_ranks,needed\_freq)

s = size(all\_ranks);

mode\_freq = [];

for i=1:s(1)

for j=1:s(2)-4-1

temp = [];

if(needed\_freq(1)-20 < all\_ranks(i,j)) && (all\_ranks(i,j) < needed\_freq(1)+20)

%mode\_freq = [mode\_freq all\_ranks(i,j)];

if(needed\_freq(2)-20 < all\_ranks(i,j+2)) && (all\_ranks(i,j+2) < needed\_freq(2)+20)

%mode\_freq = [mode\_freq all\_ranks(i,j+2)];

if(needed\_freq(3)-20 < all\_ranks(i,j+4)) && (all\_ranks(i,j+4) < needed\_freq(3)+20)

%disp(all\_ranks(i,j))

%disp(all\_ranks(i,j+2))

%disp(all\_ranks(i,j+4))

temp = [temp ;all\_ranks(i,j)];

temp = [temp ;all\_ranks(i,j+2)];

temp = [temp ;all\_ranks(i,j+4)];

Distance = euclidean(temp,needed\_freq);

temp = [temp; sum(sum(Distance))]; %Get euclidean distance

mode\_freq = [mode\_freq temp];

end

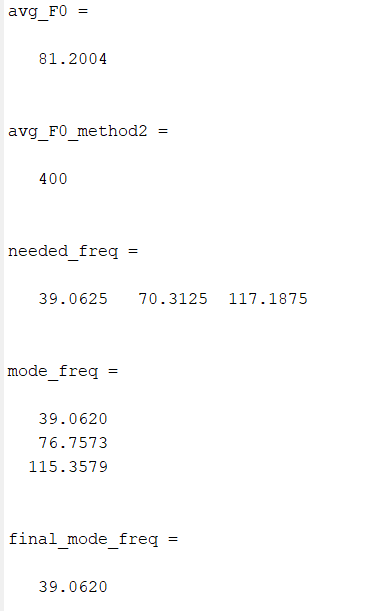
end

end

end

end

end



**REFFERENCE**

1. Karam, Zahi N., et al. "Ecologically valid long-term mood monitoring of individuals with bipolar disorder using speech." 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014.
2. Chaiani, Mounira, et al. "Voice disorder classification using speech enhancement and deep learning models." Biocybernetics and Biomedical Engineering 42.2 (2022): 463-480.
3. Dodd, Barbara. "Differential diagnosis of pediatric speech sound disorder." Current Developmental Disorders Reports 1.3 (2014): 189-196.
4. Shriberg, Lawrence D., et al. "Extensions to the speech disorders classification system (SDCS)." Clinical linguistics & phonetics 24.10 (2010): 795-824.
5. Waring, Rebecca, and R. Knight. "How should children with speech sound disorders be classified? A review and critical evaluation of current classification systems." International Journal of Language & Communication Disorders 48.1 (2013): 25-40.
6. Low, Daniel M., Kate H. Bentley, and Satrajit S. Ghosh. "Automated assessment of psychiatric disorders using speech: A systematic review." Laryngoscope Investigative Otolaryngology 5.1 (2020): 96-116.
7. F. Dellaert, T. Polzin, A. Waibel, Recognizing emotion in speech. Spoken Language, in ICSLP 96, pp, 1970–1973 (1996)
8. S. R. Kadiri, P. Gangamohan, S.V Gangashetty, B. Yegnanarayana, Analysis of excitation source features of speech for emotion recognition, in Interspeech, pp. 1324–1328 (2015)
9. S.G. Koolagudi, R. Reddy, K.S. Rao, Emotion recognition from speech signal using epoch parameters, in International Conference on Signal Processing and Communications (SPCOM), pp. 1–5 (2010)
10. S.R.M. Prasanna, D. Govind, Analysis of excitation source information in emotional speech, in Inter speech, pp. 781–784 (2010)
11. Meinedo, Hugo, and Isabel Trancoso. "Age and gender classification using fusion of acoustic and prosodic features." Eleventh annual conference of the international speech communication association. 2010.
12. Nitisara, Galih Rahagi, Suyanto Suyanto, and Kurniawan Nur Ramadhani. "Speech Age-Gender Classification Using Long Short-Term Memory." 2020 3rd International Conference on Information and Communications Technology (ICOIACT). IEEE, 2020.
13. Qawaqneh, Zakariya, Arafat Abu Mallouh, and Buket D. Barkana. "Age and gender classification from speech and face images by jointly fine-tuned deep neural networks." Expert Systems with Applications 85 (2017): 76-86.
14. T. Drugman, P. Alku, A. Alwan, B. Yegnanarayana, Glottal source processing: from analysis to appli cations. Comput. Speech Lang. 28(5), 1117–1138 (2014)
15. A.P. Prathosh, T.V. Ananthapadmanabha, A.G. Ramakrishnan, Epoch extraction based on integrated linear prediction residual using plosion index. IEEE Trans. Audio Speech Lang. Process. 21(12), 2471–2480 (2013)
16. T. Ananthapadmanabha, B. Yegnanarayana, Epoch extraction from linear prediction residual for iden tification of closed glottis interval. IEEE Trans. Acoust. Speech Signal Process. 27(4), 309–319 (1979)
17. P.A. Naylor, A. Kounoudes, J. Gudnason, M. Brookes, Estimation of glottal closure instants in voiced speech using the DYPSA algorithm. IEEE Trans. Audio Speech Lang. Process. 15(1), 34–43 (2007)
18. M.R.P. Thomas, J. Gudnason, P.A. Naylor, Estimation of glottal closing and opening instants in voiced speech using the YAGA algorithm. IEEE Trans. Audio Speech Lang. Process. 20(1), 82–91 (2012)
19. T. Drugman, T. Dutoit, Glottal closure and opening instant detection from speech signals, in Interspeech, pp. 2891–2894 (2009)
20. V. Khanagha, K. Daoudi, H. Yahia, Detection of glottal closure instants based on the microcanonical multiscale formalism. IEEE/ACM Trans. Audio Speech Lang. Process. 22(12), 1941–1950 (2014)
21. A.I. Koutrouvelis, G.P. Kafentzis, N.D. Gaubitch, R. Heusdens, A fast method for high-resolution voiced/unvoiced detection and glottal closure/opening instant estimation of speech. IEEE/ACM Trans. Audio Speech Lang. Process. 24(2), 316–328 (2016)
22. J. Gilles, Empirical wavelet transform. IEEE Trans. Signal Process. 61(16), 3999–4010 (2013)
23. N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.C. Yen, C.C. Tung, H.H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Royal Soc. Lond. A Math. Phys. Eng. Sci. 454, 903–995 (1988)
24. K. Dragomiretskiy, D. Zosso, Variational mode decomposition. IEEE Trans. Signal Process. 62(3), 531–544 (2014)
25. Mohan, Neethu, K. P. Soman, and Kumar S. Sachin. "A data-driven approach for estimating power system frequency and amplitude using dynamic mode decomposition." 2018 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE). IEEE, 2018.