Lightweight Speech Recognition for Multimodal Tongue Drive System

Yung-An Hsieh

School of Electrical and Computer Engineering

Georgia Institute of Technology

Atlanta, United States

I. Introduction

Multimodal Tongue Drive System (mTDS) [1] is an assistive technology for people with tetraplegia that uses speech recognition (SR), tongue and head motion to control devices such as a wheelchair or a computer. Currently, the speech recognition part of the system is done on a computer because of large computational requirements. However, this is not an ideal solution for wearable assistive devices. In this project, my goal is to implement a lightweight supervised learning-based speech recognition algorithm for mTDS that can classify about 20 words from the speech signal in real time. Currently, the algorithm is implemented on a laptop for testing and modifying.

In this article, some background knowledge about speech recognition will be introduced in part II. The dataset and the framework of the algorithm will be introduced in part III and IV. Several modification on the framework will then be tested and discussed according to the cross validation accuracy of the trained model in part V. Finally, the algorithm will be validated on a testing dataset for both prediction accuracy and execution time in part VI.

II. Background Knowledge

1. Speech:

Speech signal can be divided into two main categories, voiced and unvoiced sound. Voiced sound is produced by periodic source, which has a fundamental frequency called pitch. Unvoiced sound is aperiodic and noisy signal that has a lower amplitude and no well-defined pitch. The difference between voiced and unvoiced signal is a useful information for speech signal extraction.

The basic sounds of a language are called phonemes, for example, the “a” sound in the word “father”. A typical speech utterance consists of a string of vowel and consonant phonemes whose temporal and spectral characteristics change with time. Phonemes can help distinguish one word or meaning from another, therefore, they are important features for speech recognition.

2. Audio Features:

Audio features can be divided into two different levels, short-term frame-level and long-term clip-level. Both frames and clips may overlap with their previous ones. Fixed length clips are usually one to two seconds, for single word recognition in this project, I focused on frame-level features.

There are lots of frame-level features for audio signal analysis [2]. For example, volume is useful for speech-silent detection and zero-crossing rate can distinguish voiced and unvoiced sound. There are also spectral features, such as mel-frequency cepstral coefficients (MFCC), which are useful features in speech recognition. The goal of this project is to implement the algorithm on embedded hardware, thus a time domain feature, Linear Predicative Code (LPC), was chosen. Since LPC only requires time domain signal for calculation, it is computationally less expensive than using spectral features.

3. Linear Predictive Code:

LPC coefficients maps the audio to a coefficient set that has good correlation to the utterances. LPC is based on the physical model of human speech, that the shape of the vocal tract determines the sounds we

make. It aims to model human's vocal tract as a linear time-invariant (LTI) filter H(z):

$$H(z) = \frac{1}{1+a_1 z^{-1}+a_2 z^{-2}+\cdots+a_K z^{-K}} \quad (1)$$

The LPC model predicts the current sample by a linear combination of its previous K samples:

$$\hat{x}(n) = \sum_{i=1}^{K} a_i x(n-i) \quad (2)$$

Where $\hat{x}(\mathrm{n})$ is the predicted signal, $\mathrm{x}(\mathrm{n})$ are the previous signal, and a_i are the weights of each pervious signal, which are also the coefficients in equation (1). The coefficients a_i can be solved by minimizing the prediction error of equation (2) with autocorrelation method:

$$\begin{bmatrix} r_x(0) & r_x^*(1) & \cdots & r_x^*(K-1) \\ r_x(1) & r_x(0) & \cdots & r_x^*(K-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(K-1) & r_x(K-2) & \cdots & r_x(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix} = \begin{bmatrix} r_x(1) \\ r_x(2) \\ \vdots \\ r_x(K) \end{bmatrix} \quad (3)$$

Where r_x is the autocorrelation of signal and K is the number of LPC coefficients extracted.

After the coefficients are extracted, the spectrum of equation (1) represents a smoothed spectrum of the original speech signal. Also, the spectrum of equation (1) matches better to the original signal when more coefficients are extracted (Fig. 2).

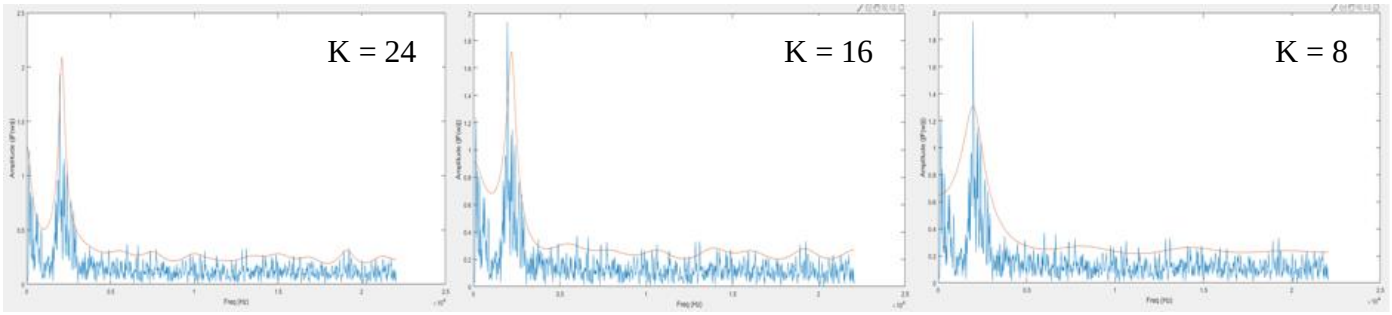

Fig. 2 Spectrum of H(z) (red) and spectrum of original signal (blue) with different K values.

III. Dataset:

In this project, the dataset contains 21 different words that can be used as basic instructions for mTDS. The word list is shown in Table 1. The words are recorded in WAV format, which has a sampling frequency of 44100Hz. Each record file is about two second long, and a single word was being spoken in the each file.

In training dataset, there are about 40 record files per word, which are recorded by three different speakers. These files are used to train the speech recognition model. A testing dataset, which is not used for training, is used in the validation part of the project. The testing dataset is further divided into three subsets, more information about these subsets will introduced in part VI.

Wheelchair	Computer	Smartphone
Off , Start , Stop		
Left , Right		
Turn	Up , Down	
Forward , Backward	Scroll, Zoom	
	Select, Hold, Return	
	Keyboard	
	Home	Click

Table 1. Word list for speech recognition on mTDS.

IV. Framework

The speech recognition algorithm in this project is based on the framework shown in Fig. 3. A brief introduction of each component in the framework is described below.

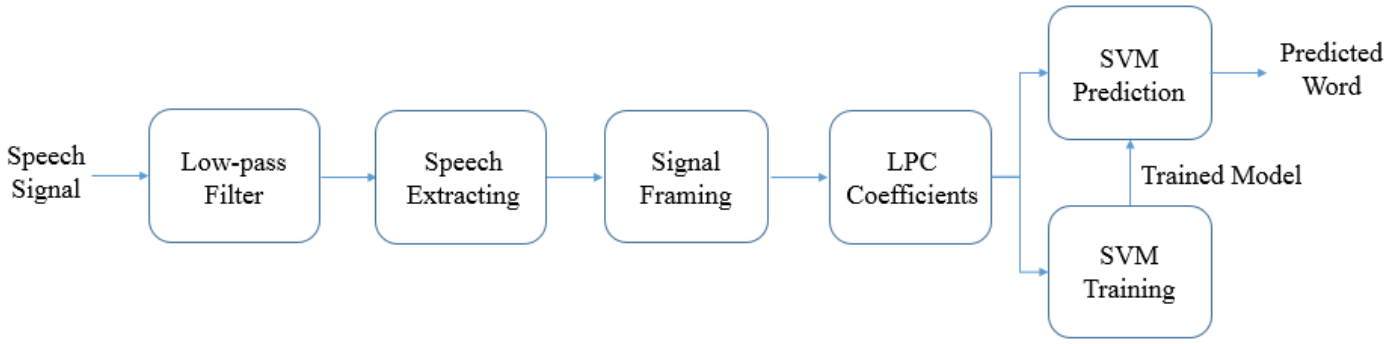

Fig. 3 Framework of the speech recognition algorithm.

1. Low-pass Filter:

After the record file is input, the speech signal is first normalized by its maximum amplitude, which makes the amplitude range of signal become [-1,1]. The signal is then passed through an infinite impulse response (IIR) low-pass filter to filter out noise. Since the human speech has a frequency range about 100 to 8000Hz, different passband frequencies of the filter within this range are tested in part V.

2. Speech Extraction:

For each record file, the speech signal has to be differentiated from silent and extracted before it is sent to generate LPC coefficients (Fig. 4). After the speech signal is extracted, it is zero-padded to about 0.8 second, which is the time frame that has been tested to be able to include the longest word “Computer” in the dataset. Several speech extraction methods are tested in this project. More details about this part will be discussed in part V.

3. Signal Framing:

The extracted 0.8 second speech signal is then separated into multiple frames. Each frame has one fourth of the frame length overlap with the pervious and next ones. Different frame lengths are tested for accuracy and execution time in part VI.

4. LPC coefficients:

For each frame of the speech signal, a set of LPC coefficients is generated using autocorrelation method. To minimize the signal discontinuities at the beginning and end of each frame, a Hamming window is applied to each frame before generating LPC coefficients. After the coefficients are generated in each frame, all of the coefficients are combined according to the frame order. This forms the coefficient set for the whole speech signal that is used for SVM training and prediction (Fig. 4). Different numbers of coefficients per frame are also tested for accuracy and execution time in part VI.

5. Support Vector Machine (SVM):

SVM [3] is used as the learning model in this project because it is computationally less expensive and requires less memory when compared with deep learning model. The word of each record file is input as the label of SVM training, and the generated LPC coefficients are input as the features. The number of features depends on the frame number and the LPC coefficients per frame. For example, with a signal separated into 23 frames and 8 LPC coefficients per frame, 184 LPC coefficients are used as the features of a single speech signal.

In this project, the radial basis function (RBF) kernel is used. With different framework parameters, such

as frame length or coefficients per frame, different sets of training features are generated. For each training set, the penalty and kernel parameters used for SVM model training are adjusted to get the highest five-fold cross validation accuracy [4]. After a model is trained, it is used to predict record files that are not used in training but with the same parameters set in the framework.

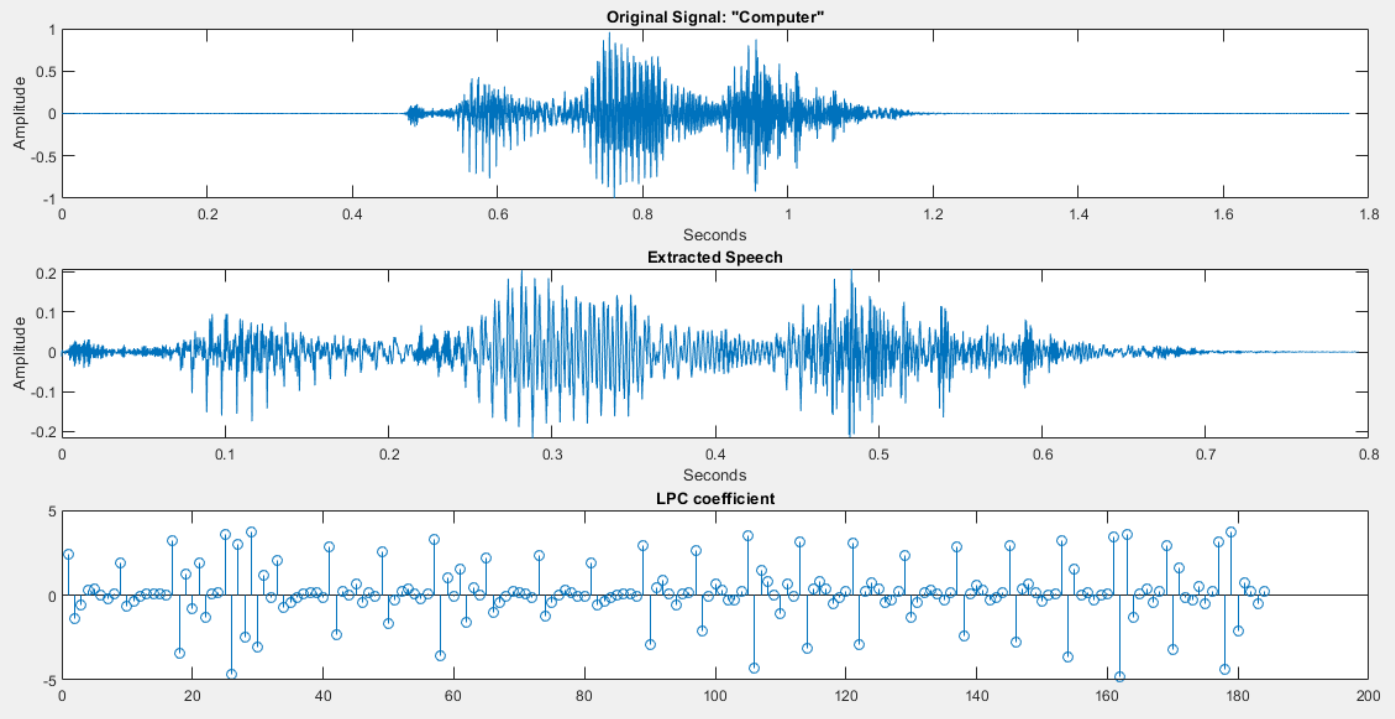

Fig. 4 The top plot shows the recorded signal that is input to the framework and the middle plot shows the extracted speech signal. The bottom plot shows the combination of LPC coefficients from all signal frames, which is sent to SVM model as training features. The LPC coefficients are put in the order of signal frames, with eight coefficients per frame in this figure, the first eight coefficients in the bottom plot are generated from frame one, the next eight coefficients are from frame two, and so on.

V. Framework Modification

In this part, three different modifications on the algorithm framework are tested, which include different speech extraction method, an added high-pass filter, and different passband frequency of low-pass filter. The testing is based on the five-fold cross-validation accuracy of the whole training dataset.

1. Speech Extraction Method:

(1) Direct Extraction:

Since the speech files in the training dataset were recorded in quiet environment, this method is first used to verify the LPC coefficients generating code. An amplitude threshold is set in this method and after a sample exceeds the threshold, a specific length, 0.8 second in this project, of the following signal is extracted directly. This method only works fine for ideal records, which have no background noise and the speaker is speaking in normal volume and pace.

(2) Mean Amplitude (MA) and Zero-Crossing Rate (ZCR):

To better extract speech in non-ideal records, which have background noise or the speaker is speaking the word with pause between syllables, the MA-ZCR method is used. In this method, the whole recorded signal is separated into frames before the speech is extracted. For each frame, the MA and ZCR values are calculated as:

$$\mathrm{MA(n)} = \frac{1}{N}\sum_{i=0}^{N-1}|s_n(i)| \tag{4}$$

$$\mathrm{ZCR(n)} = Number\ of\ times\ s_n\ crosses\ zero \tag{5}$$

Where n is the frame number, N is the frame length, and s_n is the signal.

Thresholds are set for both MA and ZCR, frames that have MA and ZCR values below the thresholds were eliminated. To set the thresholds of MA and ZCR, their values are first observed in several parts of speech signal, and the result is shown in Table 2. Using the observation, two MA thresholds and one ZCR threshold are set for speech extraction with the pseudo code shown in Fig 5. Different MA and ZCR thresholds are being tested using the cross validation accuracy of the trained model, and the thresholds that had the highest accuracy were selected (Table 3).

	MA	**ZCR**
Voiced Sound	High	Low
Unvoiced Sound	Low	High
Background Noise	Low	Low
Nearly Silent	Very low	High

Table 2 Comparison of MA and ZCR values.

If MA(i) $> MA_{thd\_high}$:
    Voiced sound → Extracted
Else if MA(i) $> MA_{thd\_low}$ and ZCR(i) $> ZCR_{thd}$:
    Unvoiced sound → Extracted
Else: Noise or Silent → Eliminated

Fig. 5 Pseudo code of speech extraction with MA-ZCR method.

MA_{thd_high}	MA_{thd_low}	ZCR_{thd}	**Accuracy**
0.05	0.002	50	57.69%
0.05	0.002	100	53.54%
0.05	0.005	50	58.81%
0.05	0.008	50	55.56%
0.03	0.005	50	58.80%
0.07	0.005	50	58.81%

Table 3 Comparison of different MA and ZCR thresholds

(3) Amplitude Thresholding:

In this method, a threshold is set for every samples in the speech signal, samples that have amplitudes below the threshold are eliminated. The method is similar to using mean amplitude threshold of each frame, but the frame length was now set to a single sample. This avoids the situation of losing speech signal when a frame consists of mostly silent but also small portion of speech signal in MA-ZCR method. However, this method causes some discontinuities in the signal, and it also doesn't perform well in non-ideal records.

To see how this method affect the LPC coefficients generation and thus the speech prediction accuracy, the spectrum of LPC filter and original signal in three different parts of the signal were observed (Fig. 6). The observation results are discussed in Table 4.

	Discussion
Unvoiced Sound	Unvoiced sound has low amplitude and high zero crossing rate, thus many samples will be eliminated by amplitude threshold method. This causes the discontinuities of signal in amplitude threshold method, which can be seen in the expended spectrum in Fig. 6a. However, the main part of the spectrum in the original signal (Fig. 6d) is still preserved, so the LPC coefficients can still preserve the speech information.
Voiced Sound	Voiced sound has high amplitude and low zero crossing rate, thus it is slightly affected by the discontinuities caused by amplitude thresholding. Both the spectrum of the LPC filter and the original signal maintain nearly the same when using amplitude thresholding (Fig. 6b and Fig. 6e).

Silent	In original signal, the silent part is actually still having slightly random noise. Thus the generated LPC coefficients form a filter that tries to match the noise spectrum (Fig. 6f). This causes LPC coefficients be generated randomly in different frames that both represent silent. With amplitude thresholding, these small random noise is eliminated, which makes the silent part become completely zero in all samples. Thus, all the LPC coefficients in silent frames will be zero, which forms a constant LPC filter shown in Fig. 6c.

Table 4 Discussion of the spectrum of LPC filter and original signal in different parts of signal.

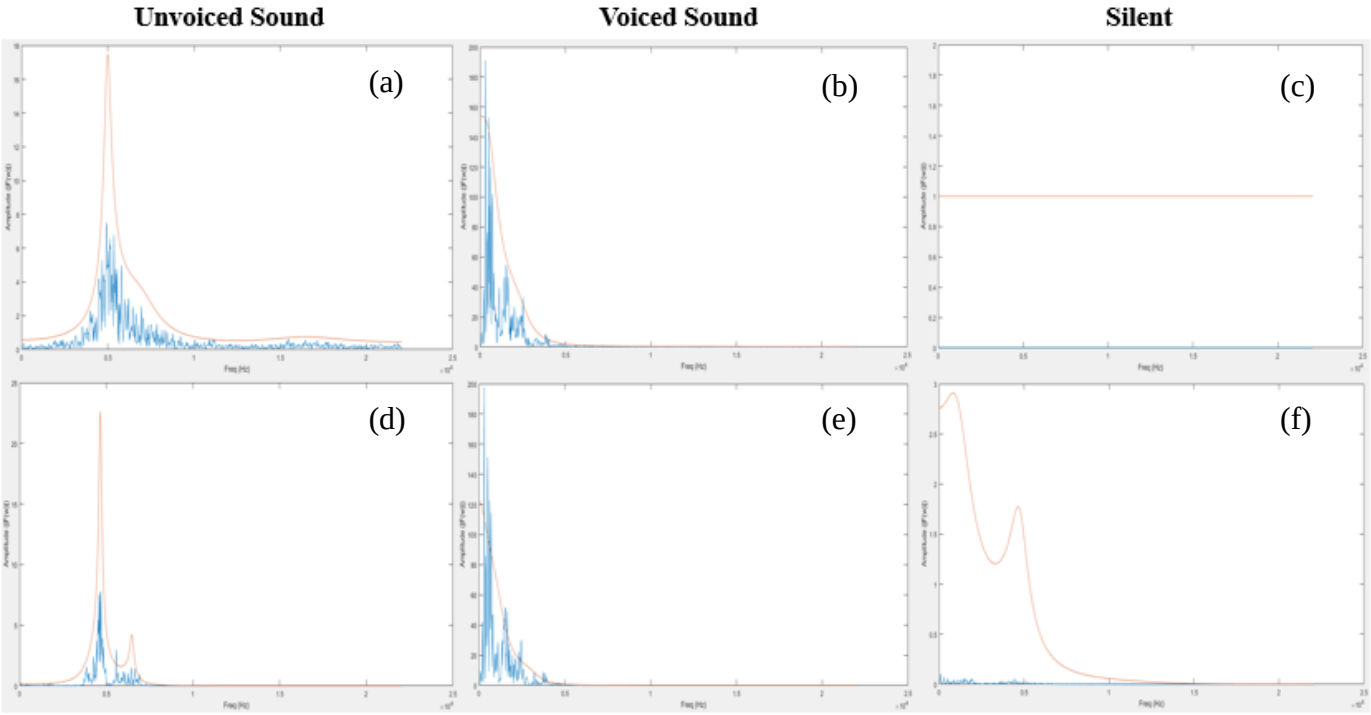

Fig. 6 The spectrum of the LPC filter (red) and the original signal (blue) in three different parts of the signal are presented. The first row contains the spectrums of speech using amplitude thresholding method, and the second row contains the spectrums of the original speech signal.

Three different methods of speech extraction are compared using five-fold cross-validation accuracy of the training model, the result is shown in Table 5. The amplitude threshold method yields the highest accuracy. This means that the discontinuities caused by amplitude thresholding method have little affection on preserving speech information with LPC coefficients. Also, making LPC coefficients in silent frames as zeros rather than random values that try to match the spectrum of random noise help increase the accuracy.

	Direct Extraction	**MA and ZCR**	**Amplitude Threshold**
Cross-Validation Accuracy	52.53%	58.81%	77.55%

Table 5 Comparison of the three speech extraction methods.

2. High-pass Filter:

Since the high frequency components of the speech signal are more susceptible to noise and have lower amplitude, a high-pass filter is added to boost the high frequency components after the speech is extracted. Two different high-pass filters are tested on cross-validation accuracy, the frequency response of these filters are shown in Fig. 7. The difference between the two filters is that the second filter balance more between low and high frequency components as it reduces more on low frequency components. The comparison result with different and also no high-pass filter in the framework is shown in Table 6. The second filter, which balanced more between low and high frequency components, yields the highest frequency.

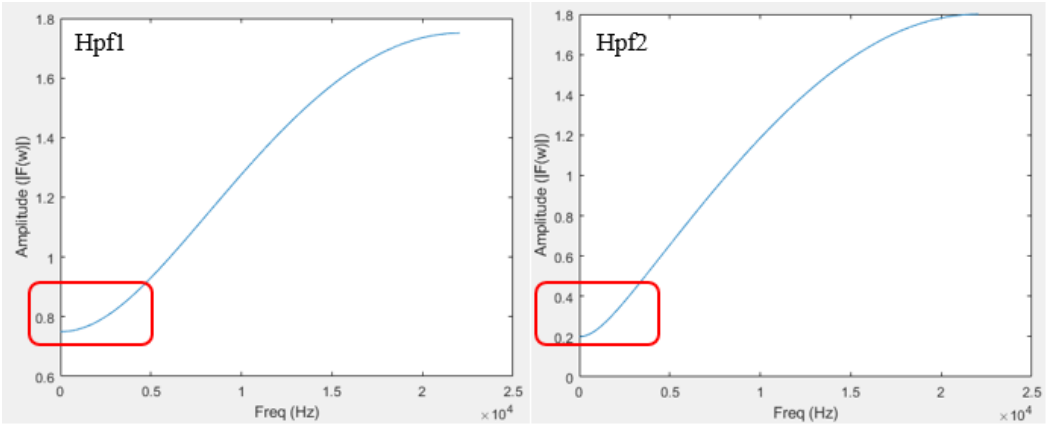

Fig. 7 The frequency response of two different high-pass filter. The main difference between these filters is labeled with the red box, where the second filter (Hpf2) reduces more on low frequency components.

	No high-pass filter	Hpf1	Hpf2
Cross-Validation Accuracy	77.55%	83.50%	86.08%

Table 6 Comparison of different high-pass filter.

3. Passband Frequency of Low-pass Filter:

The low-pass filter in the beginning of the system framework is used to remove noise in the input signal. With lower passband frequency, more noise can be eliminated. However, more speech information in high frequency components will also be lost. To see how the tradeoff between noise and speech information affects the prediction of word, several passband frequency were set in the framework for testing. Table 7 shows the cross-validation accuracy with different passband frequency of the low-pass filter. With passband frequency set to 5000Hz, the tradeoff between speech information preserving and noise eliminating yields the highest cross-validation accuracy.

Passband Frequency	500Hz	2000Hz	5000Hz	8000Hz
Cross-Validation Accuracy	74.86%	81.14%	86.08%	85.30%

Table 7 Comparison of different passband frequency of the low-pass filter.

VI. Validation

In this part, a testing dataset which is not included in model training is used for accuracy validation. The testing dataset is further divided into three different subsets with five record files per word, the difference between each subsets is shown in Table 8. Since the ultimate goal of this project is to implement the speech recognition algorithm onto a microprocessor and predict words in real-time, timing complexity is also an important aspect. So besides the accuracy, the execution time of the algorithm for word recognition, which used the trained SVM model for prediction, was also tested on a laptop in this part.

Testing Subset	Description
Set 1	A speaker in major part of the training dataset, with ideal records.
Set 2	A speaker in minor part of the training dataset, with non-ideal records. Non-ideal records are record files that have background noise or the speaker is speaking in low volume or abnormal pace.
Set 3	A speaker not in the training dataset, with non-ideal records.

Table 8 Description of three different testing subsets.

1. Frame Length and LPC Coefficients:

Two framework parameters, frame length and LPC coefficients per frame (labeled as K in the following article), are being adjusted to different values for testing the prediction accuracy and execution time. Three different frame lengths are tested in this part. For each frame length, four different numbers of K are tested. The number of frames is also adjusted in different frame length to get the same length of the whole speech signal, for example, 23 frames were used for 2000 samples per frame and 31 frames for 1500 samples per frame to both get about 0.8 seconds of the speech signal. Each frame has one fourth of the frame length overlap with each other in every frame length. The K values are also selected to generate four different LPC sets in different frame length, which have nearly the same total number of coefficients for the whole signal in the same set. The total number of LPC coefficients in different LPC sets are shown in Table 9.

The results of prediction accuracy and execution time are shown in Table 9-11 and Fig. 8. The accuracy of Test_set1 is very close to the cross-validation accuracy of the training dataset, and the increasing of accuracy also becomes slower while more LPC coefficients are used. With non-ideal records in Test_set2 and Test_set3, the accuracy is much lower, and the increasing of LPC coefficients has no improvement on the accuracy. By computing the average accuracy of three testing subsets, the algorithm using 23 frames, which has the largest frame length, yields the highest accuracy. The execution time of the algorithm is also the lowest when 23 frames are used.

Frame length = 2k samples	**5-fold cross-validation**	**Test Set1**	**Test Set2**	**Test Set3**	**Test Sets Average**	**Execution Time per prediction**
K = 4 (92 LPC – Set1)	83.28%	78.10%	44.76%	21.90%	48.25%	0.0187 sec
K = 8 (184 LPC – Set2)	86.08%	83.81%	38.10%	23.81%	48.57%	0.0262 sec
K = 12 (276 LPC – Set3)	85.52%	89.52%	41.90%	23.81%	51.74%	0.0341 sec
K = 16 (368 LPC – Set4)	86.20%	91.43%	42.86%	21.90%	52.06%	0.0434 sec

Table 9 Validation result of frame length equals to 2000 samples with 23 frames.

Frame length = 1.5k samples	**5-fold cross-validation**	**Test Set1**	**Test Set2**	**Test Set3**	**Test Sets Average**	**Execution Time per prediction**
K = 3 (Set1)	81.82%	82.86%	43.81%	22.86%	49.84%	0.0209sec
K = 6 (Set2)	85.19%	84.76%	32.38%	25.71%	47.62%	0.0287 sec
K = 9 (Set3)	85.86%	88.57%	40.95%	21.90%	50.47%	0.0360 sec
K = 12 (Set4)	86.08%	88.57%	40.00%	27.62%	52.06%	0.0443 sec

Table 10 Validation result of frame length equals to 1500 samples with 31 frames.

Frame length = 1k samples	**5-fold cross-validation**	**Test Set1**	**Test Set2**	**Test Set3**	**Test Sets Average**	**Execution Time per prediction**
K = 2 (Set1)	78.45%	79.05%	37.14%	20.95%	45.71%	0.0220sec
K = 4 (Set2)	81.59%	83.81%	42.86%	17.14%	47.94%	0.0288 sec
K = 6 (Set3)	86.42%	81.90%	35.24%	19.05%	45.40%	0.0364 sec
K = 8 (Set4)	86.42%	88.57%	35.24%	20.00%	47.94%	0.0448 sec

Table 11 Validation result of frame length equals to 1000 samples with 46 frames.

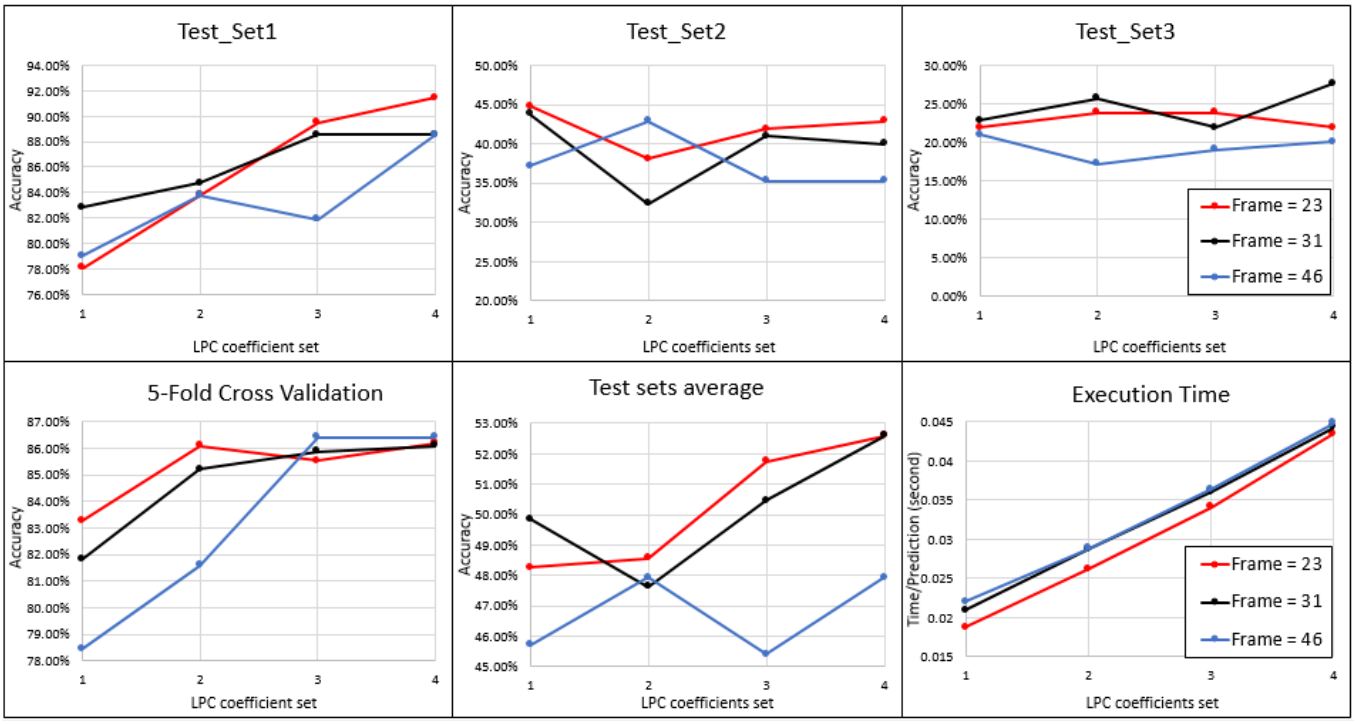

Fig. 8 Validation result of different frame lengths (frame numbers) and LPC coefficient sets.

2. Non-ideal Records:

According to the results in Fig. 8, amplitude thresholding method performs well with ideal records (Test_set1). However, since only a single amplitude threshold is set in this method, it is not able to eliminate large background noise in non-ideal records (Fig. 9c). This problem also happens when speakers speak in low volume, as the recorded signal being normalized, the originally small noise become large enough to cross the threshold. Thus with non-ideal records in Test_set2 and Test_set3, the prediction accuracy is low. On the other hand, MA-ZCR method has better performance on removing large background noise since a ZCR threshold is also set for speech extraction (Fig. 9b). However, as discussed in part V-1, amplitude thresholding still has a much higher cross-validation accuracy than MA-ZCR method.

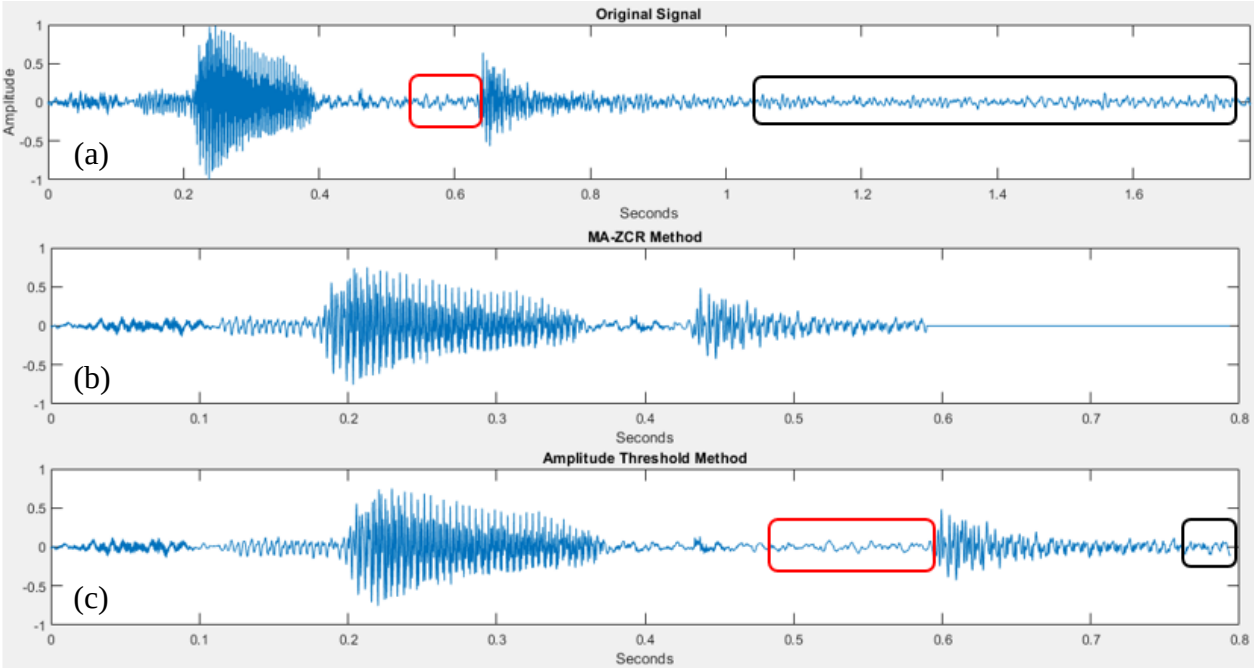

Fig. 9 Comparison between MA-ZCR method (b) and amplitude thresholding (c) methods in non-ideal record (a). The pause between two syllables (red box) and the background noise (black box) are eliminated in MA-ZCR method.

To try improving the performance of the algorithm in non-ideal records, amplitude thresholding and MA-ZCR methods are combined in this part. The MA-ZCR method is first used to eliminate frames with large background noise, and the amplitude thresholding method is then used to extract the final speech signal for LPC coefficients generation. Two different LPC sets (Set2 and Set3) in part VI-1 and three different frame length, thus three different frame numbers, are tested. The validation results are shown in Table 12-13. In Fig. 10-11 the results are compared with using only amplitude thresholding method.

In Fig. 10, the result shows that adding MA-ZCR method leads to a lower accuracy in ideal records (Test_set1), which matches the result in part V-1. In Test_set2, accuracy with MA-ZCR method is still lower, but as more frames are used with MA-ZCR method, the accuracy increases. In Test_set3, adding MA-ZCR method leads to a higher accuracy than only using amplitude thresholding method. Both Tset_set2 and Test_set3 indicate that, as the frame length decreases, adding MA-ZCR method can further improve the accuracy in non-ideal records. This improvement can be more obvious when the non-ideal records were spoken by speakers not in the training dataset.

LPC Num. = 184 (LPC Set2)	**5-fold cross-validation**	**Test Set1**	**Test Set2**	**Test Set3**	**Test Sets Average**	**Execution Time per prediction**
23 Frames	80.36%	71.43%	28.57%	25.71%	41.90%	0.0693sec
31 Frames	75.76%	72.38%	34.29%	31.43%	46.03%	0.0809 sec
46 Frames	67.68%	72.38%	38.10%	36.19%	48.89%	0.1023 sec

Table 12 Validation result of combining MA-ZCR and amplitude thresholding methods with 184 LPC coefficients.

LPC Num. = 276 (LPC Set3)	**5-fold cross-validation**	**Test Set1**	**Test Set2**	**Test Set3**	**Test Sets Average**	**Execution Time per prediction**
23 Frames	80.58%	78.10%	28.57%	24.76%	43.81%	0.0783 sec
31 Frames	78.68%	71.43%	34.29%	26.67%	44.13%	0.0893 sec
46 Frames	63.85%	69.52%	36.19%	31.43%	45.71%	0.1043 sec

Table 13 Validation result of combining MA-ZCR and amplitude thresholding methods with 276 LPC coefficients.

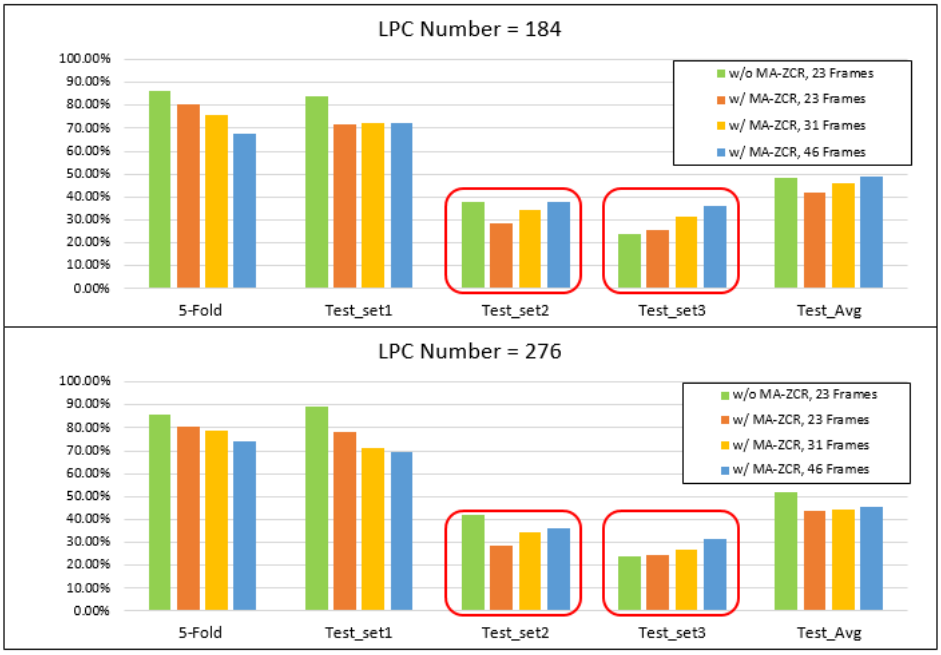

Fig. 10 Comparison of prediction accuracy between results in Table 12-13 and using amplitude thresholding method only (green).

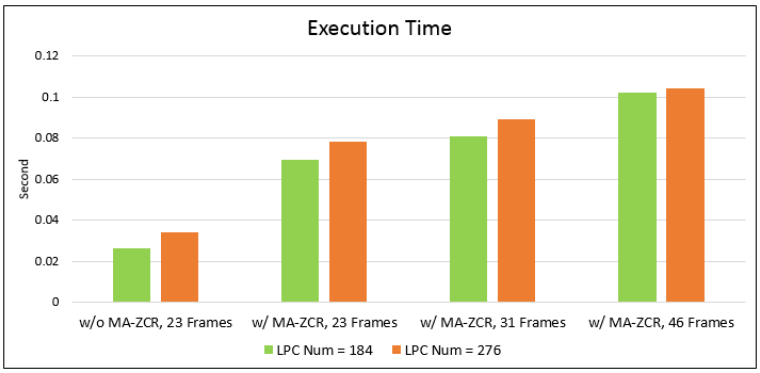

Fig. 11 Comparison of execution time between results in Table 12-13 and using amplitude thresholding method only. By adding MA-ZCR method, the algorithm requires much more time than using only amplitude thresholding method.

VII. Conclusion

In this article, a speech recognition algorithm framework using Linear Predictive Code (LPC) and Support Vector Machine (SVM) is presented. With amplitude thresholding method used for speech extraction and a high-pass filter that balances the low and high frequency components of speech signal, the accuracy of the algorithm can be improved. The algorithm performs well in ideal records, with more LPC coefficients are used, the accuracy is increased but with slower increasing rate. With same amount of LPC coefficients, using 23 frames with 2000 samples per frame has a slightly higher accuracy and lower execution time. For non-ideal records, the MA-ZCR method can be added to improve the accuracy but with increased execution time.

VIII. Reference

[1] M.N. Sahadat, A. Alreja, M. Ghovanloo, “Simultaneous multimodal PC access for people with disabilities by integrating head tracking, speech recognition, and tongue motion”, IEEE Trans. Biomed. Circuits Syst., vol. 12, no. 1, pp.192-201, Feb. 2018.

[2] Santosh K.Gaikwad, Bharti W.Gawali, Pravin Yannawar, “A Review on Speech Recognition Technique”, International Journal of Computer Applications (0975 – 8887) Volume 10– No.3, November 2010.

[3] C.-C. Chang, C.-J. Lin, "LIBSVM: a library for support vector machines", no. 5, 2001, [online] Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] C.-W. Hsu, C.-C. Chang, C.-J. Lin, “A practical guide to support vector classification”, Tech. rep., 2003, Department of Computer Science, National Taiwan University.