Chapter 2

Pitch Detection Algorithms

Tone information mainly depends on the pitch of the speech. Pitch detection is the first task for tone classification. Frequency components of the speech include the fundamental frequency and the harmonic frequency called formant. Pitch is determined by the fundamental frequency (F0) of the speech signal. To extract fundamental frequency, both frequency domain methods and time domain methods can be used.

In this Chapter, the backgrounds of pitch detection are introduced first. Then the implementations are described. The experiments and discussion are presented after that. Finally it is the summary of this chapter.

2.1 Backgrounds

In this section, first the pre-processing techniques of Pitch Tracking are introduced. Then it is the Pitch Tracking Algorithm. After that the smoothing technique of pitch contour is presented. Finally it is feature extraction of pitch contour.

2.1.1 Pre-processing of Pitch Tracking

The speech signal includes very rich harmonic components. The minimum F0 is about 80 Hz and the maximum is about 500 Hz. Most of them are in the range of 100-200 Hz. Thus the signal may involve 30-40 harmonic components. And the F0 component is often not the strongest one. Because the first formant usually is between 300-1000 Hz. The 2-8 harmonic components usually stronger than fundamental component (Wang *et al.*, 2001). The rich harmonic components let the pitch tracking become very complex. It usually has the harmonic errors and sub-harmonic errors. To improve the reliability some pre-processing of signal is necessary.

Since, the range of F0 is generally in the range of 80-500 Hz, then the frequency components above 500 Hz is useless for pitch detection. Thus a low-pass filter with pass-band frequency above 500 Hz would be useful in improving the performance of pitch detection. Generally, we use the low-pass-filter with 900 Hz (Liang *et al.*, 1999).

Also to reduce the effects of the formant structure, the nonlinear processing is usually used in pitch tracking.

$$y(n) = C[x(n)] \tag{2-1}$$

Where $x(n)$ is the original signal. $y(n)$ is the processed signal. $C[\ ]$ is the nonlinear function.

One of the nonlinear technique is center-clipping (Rabiner *et al.*, 1977; Kechu *et al.*, 2000) of speech which is first introduced by *M. M. Sondhi* (Cited by Rabiner *et al.*, 1977). The relation between input $x(n)$ and $y(n)$ is:

$$y(n) = clc[x(n)] = \begin{cases} (x(n) - C_L), & x(n) \geq C_L \\ o, & |x(n)| < C_L \\ (x(n) + C_L), & x(n) \leq C_L \end{cases} \tag{2-2}$$

Another nonlinear clipping we call is infinite-peak-clipping (Rabiner *et al.*, 1977; Kechu *et al.*, 2000). The function is described in (2-3):

$$y(n) = \operatorname{sgn}[x(n)] = \begin{cases} 1, & x(n) \geq C_L \\ 0, & |x(n)| < C_L \\ -1, & x(n) \leq C_L \end{cases} \tag{2-3}$$

where C_L is the clipping threshold. Generally C_L is about 30% of the maximum magnitude of signal. In application the C_L should be as high as possible. To get the high C_L, we can catch the peak value of the first 1/3 and the last 1/3 of signal and use the less one to be the maximum magnitude. Then we set the 60-80% of this maximum magnitude to be C_L.

The effect of center-clipping and infinit-peak-clipping is clearly shown in the Figure 2-1 (a, b, c). From Figure 2-1 (b), after center-clipping, the autocorrelation only leave several pulse that show the reduction of the confused secondary peak. From Figure 2-1 (c), the first peak is very clear. Also the secondary peak value is reduced. All of these

(a) $x(n)$ and Auto-correlation $x(n)$

(b) $clc[x(n)]$ and Auto-correlation $clc[x(n)]$

(c) sgn[$x(n)$] and Auto-correlation sgn[$x(n)$]

Figure 2-1 $x(n), clc[x(n)], \text{sgn}[x(n)]$ and the Auto-correlation

show that the center-clipping and infinite-clipping is effective in reducing the effects of the formant structure.

2.1.2 Pitch Tracking Algorithms

Basically, pitch detection algorithms use short-term analysis techniques. For every frame x_m we get a score $f(T/x_m)$ that is a function of the candidate pitch periods T. Algorithm determine the optimal pitch by maximizing (2-4).

$$T_m = \arg\max_T f(T/x_m) \tag{2-4}$$

A commonly used method to estimate pitch is based on detecting the highest value of the auto-correlation function (Rabiner *et al.*, 1976; Rabiner *et al.*, 1977; Kechu *et al.*, 2000) in the region of interest. Given a discrete time signal $x(n)$, defined for all n , the auto-correlation function is generally defined in (2-5):

$$R_x(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)x(n+m) \tag{2-5}$$

The autocorrelation function of a signal is basically a (non-invertible) transformation of the signal that is useful for displaying structure in the waveform. Thus, for pitch detection, if we assume *x(n)* is exactly periodic with period *P*, i.e., $x(n) = x(n+P)$ for all *n*, then it is easily shown that:

$$R_x(m) = R_x(m+P) \tag{2-6}$$

i.e., the autocorrelation is also periodic with the same period. Conversely, periodicity in the autocorrelation function indicates periodicity in the signal.

For a nonstationary signal, such as speech, the concept of a long-time autocorrelation measurement as given by (2-5) is not really meaningful. Thus, it is reasonable to define a short-time autocorrelation function, which operates on short segments of the signal as:

$$R_x(m)=\frac{1}{N}\sum_{n=0}^{N'-1}\left[x(n+l)w(n)\right]\left[x(n+l+m)w(n+m)\right],\quad 0\le m\le M_0 \qquad (2\text{-}7)$$

where $x(n)$ is an appropriate window for analysis, N is the section length being analyzed, N' is the number of signal samples used in the computation of $R(m)$, M_0 is the number of autocorrelation points to be computed, and l is the index of the starting sample of the frame. For pitch detection applications N' is generally set to the value in (2-8):

$$N'=N-m \qquad (2\text{-}8)$$

So that only the N' samples in the analysis frame (i.e., $x(l),x(l+1),...,x(l+N-1)$) are used in the autocorrelation computation. Values of 200 and 300 have generally been used for M_0 and N, respectively, it is corresponding to a maximum pitch period of 20 ms (200 samples at a 10 kHz sampling rate) and a 30 ms analysis frame size.

A variation of autocorrelation analysis for measuring the periodicity of voiced speech uses the AMDF (Rabiner *et al.*, 1976; Kechu *et al.*, 2000), defined by the relation in (2-9):

$$D_m=\frac{1}{L}\sum_{n=1}^{l}\left|x(n)-x(n-m)\right|,\qquad m=0,1,...,m_{\max} \qquad (2\text{-}9)$$

Where *x(n)* are the samples of input speech and *x(n-m)*, are the samples time shifted *m* seconds. The vertical bars denote taking the magnitude of the difference $x(n)-x(n-m)$. Thus a difference signal *Dm*, is formed by delaying the input speech various amounts, subtracting the delayed waveform from the original, and summing the magnitude of the differences between sample values. The difference signal is always zero at delay = 0, and is particularly small at delays corresponding to the pitch period of a voiced sound.

The AMDF is a variation of ACF (Autocorrelation Function) analysis where, instead of correlating the input speech at various delays (where multiplication and summations are formed at each value), a difference signal is formed between the delayed speech and the original, and at each delay value the absolute magnitude is taken. Unlike the autocorrelation or cross-correlation function, however, the AMDF calculations require no multiplication, a desirable property for real-time applications.

For each value of delay, computation is made over an integrating window of N samples. To generate the entire range of delays, the window is "cross difference" with the full analysis interval. An advantage of this method is that the relative sizes of the nulls tend to remain constant as a function of delay. This is because there is always full overlap of data between the two segments being cross difference.

In extractors of this type, the limiting factor on accuracy is the inability to completely separate the fine structure from the effects of the spectral envelope. For this reason, decision logic and prior knowledge of voicing are used along with the function itself to help make the pitch decision more reliable.

2.1.3 Smoothing

Generally, the pitch determination described above is still error-prone. The erroneous voiced/unvoiced decisions and inaccurate voiced pitch hypotheses can lead to noisy and undependable feature measurements. Then a smoothing stage is necessary in improving the performance of the system.

The basic concept of a linear smoother is the separation of signals based on their non-overlapping frequency content. For nonlinear smoothers it is more convenient to consider separating signals based on whether they can be considered smooth or rough (noise-like). Thus a signal *x(n)* can be considered as

$$x(n) = S[x(n)] + R[x(n)] \tag{2-10}$$

where $S[\ ]$ is the smooth part of the signal *x(n)* and $R[\ ]$ is the rough part of the *signal x(n)* . The candidate proposed by Tukey (Cited by Rabiner *et al.*, 1975) for extracting $f[x(n)]$ from $x(n)$ was to use running medians of the data. Running medians have several good properties which make them good candidates for a smoother. These include the following properties.

Property 1: $\text{Median}[ax(n)] = a\ \text{median}[x(n)]$.

Property 2: Medians will not smear out sharp discontinuities in the data, as long as the duration of the discontinuity exceeds some critical duration.

Property 3: Medians will approximately follow polynomials.

Although median smoothing preserves sharp discontinuities in the data, it fails to provide sufficient smoothing of the undesirable noise-like components for which the smoothing was originally designed. A fairly good solution is a smoothing algorithm based on a combination of running medians and linear smoothing. Since the

running medians provide a fair amount of smoothing already, the linear smoothing can consist of a fairly low-order system and still give adequate results. Tukey proposed the use of a 3-point Hanning window as one candidate for the linear smoother.

2.1.4 Feature Extraction

Due to the smoothness of a pitch contour, the 3rd order polynomial using least-mean-square and Orthogonal polynomial approximation are chosen to fit the pitch contour.

For least-mean-square approximation (Kechu *et al.*, 2000), it expresses the approximation function as the sum of weighted observation value.

$$f_{LMS} = \sum_{j=1}^{N} a_i x_i \tag{2-11}$$

Where f_{LMS} is the estimated function, a_j is the weighted coefficients, x_i is the observation item $(x_1 = 1, x_2 = x, x_3 = x^2, x_4 = x^3)$. Then it is to minimize the expectation value of approximation error $(e = f_{LMS} - f)$ to get the weighted coefficients a_i. In order to minimize the expectation value, we need to get the derivation of the expectation value and set it to zero. Then the coefficients will be calculated through equation (2-12).

$$\sum_{j=1}^{N} E\left(x_i x_j\right) a_j = E\left(f x_i\right) \tag{2-12}$$

Orthogonal polynomials are defined in terms of their behavior with respect to each other and throughout some predetermined range of the independent variable. In the case of the vectors, if the set was complete it was said to span a vector space and any vector in that space could be expressed as a linear combination of orthogonal basis vectors. The first four discrete Legendre polynomials (Wang *et al.*, 2001; Chen *et al.*,1990) can be chosen to represent the pitch contour. They are shown in equation (2-13).

These polynomials are normalized in length to [0,1]. Where i is from 0 to N, $N+1$ is the length of pitch contour and N should be bigger than three. Legendre polynomials is a kind of Orhtogonal polynomials with the simplest weight function which is equal to 1. They are chosen to represent the pitch contour because they resemble to the basic pitch contour patterns. A pitch contour segment $f(i/N)$, can then be as (2-14).

$$\phi_0\left(\frac{i}{N}\right)=1,$$

$$\phi_1\left(\frac{i}{N}\right)=\left[\frac{12\times N}{N+2}\right]^{1/2}\left[\frac{i}{N}-\frac{1}{2}\right],$$

$$\phi_2\left(\frac{i}{N}\right)=\left[\frac{180\times N^3}{(N-1)(N-2)(N-3)}\right]^{1/2}\left[\left(\frac{i}{N}\right)^2-\frac{i}{N}+\frac{N-1}{6\times N}\right],$$

$$\phi_3\left(\frac{i}{N}\right)=\left[\frac{2800\times N^5}{(N-1)(N-2)(N+2)(N+3)(N+4)}\right]^{1/2}\bullet$$
$$\left[\left(\frac{i}{N}\right)^3-\frac{3}{2}\left(\frac{i}{N}\right)^3+\frac{6N^3-3N+2}{10\times N^2}\left(\frac{i}{N}\right)+\frac{(N-1)(N-2)}{20\times N^2}\right],$$

(2-13)

$$\hat{f}\left(\frac{i}{N}\right)=\sum_{j=0}^{3}a_j\times\phi_j\left(\frac{i}{N}\right),\qquad 0\le i\le N \tag{2-14}$$

Where

$$a_j=\frac{i}{N+1}\sum_{i=0}^{N}f\left(\frac{i}{N}\right)\times\phi_j\left(\frac{i}{N}\right), \tag{2-15}$$

The reconstructed pitch contour will not lose much information since orthogonol polynomials up to degree of three are used to fit it.

2.2 Implementations

Here, first the implementation of AMDF algorithm is described. Then the implementation of pitch-tracking algorithm of auto-correlation is introduced. Finally the framework of classification are presented.

2.2.1 AMDF

We only implement a coarse quantization. Figure 2-2 shows a block diagram of the AMDF pitch detector. The speech signal, is initially sampled at 10 kHz. Then the signal pass a low-pass filter (0-900 Hz) and set the first 20 samples to be zero. The clipping threshold is then calculated and the center-clipping is done on the signal. Then average magnitude difference function is computed on the center-clipped speech signal at the lag (20—140 samples) through the signal from 20 to 160 samples. The pitch period is identified as the value of the lag which the minimum AMDF occurs. Thus a fairly coarse quantization is obtained for the pitch period.

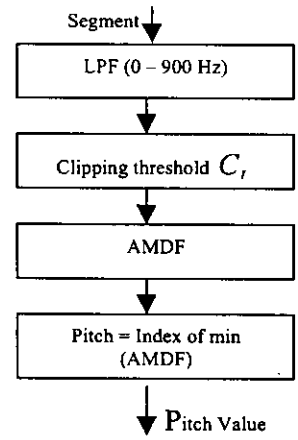
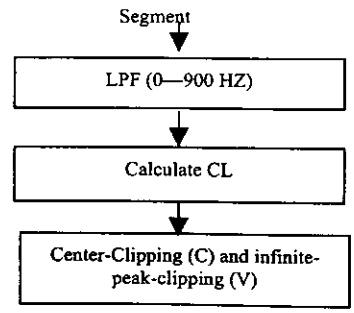


Figure 2-2 Block Diagram of the Coarse Pitch Detection using AMDF

Also the five-point median-filter and feature extraction using LMS and Orthogonal polynomials are implemented according to the introduction above.

2.2.2 Auto-correlation

The modified auto-correlation pitch detector based on the center-clipping method and infinite-peak-clipping is used in our implementation. Figure 2-3 shows a block diagram of the pitch detection algorithm. The speech signal is sampled at 10 kHz. The method requires that the speech be low-passed filtered to 900 Hz and sectioned into overlapping 30-ms (300 samples) sections for processing. Since the pitch period computation for all pitch detectors is performed 100 times/s i.e, every 10 ms, adjacent sections overlap by 20 ms or 200 samples. The first stage of processing is the computation of a clipping threshold C_L for the current 30-ms section of speech. The clipping level is set at a value which is 68 percent of the smaller of the peak absolute sample values in the first and last 10-ms portions of the section. Following the determination of the clipping level, the 30-ms section of speech is center clipped, and then infinite peak clipped. Following clipping the auto-correlation function for the 30-ms section is computed over a range of lags from 20 samples to 160 samples (i.e., 2-ms-20-ms period). The location of the maximum in auto-correlation function is chose asthe pitch period.



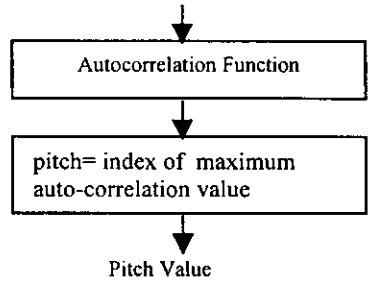


Figure 2-3 Block Diagram of Pitch Detection Algorithm using Modified Auto-correlation Method

2.2.3 Classification Framework

The pitch extraction program extracted pitch according to the speech wave file and corresponding label information. The extracted pitch is smoother using smoothing program. The smoothed the pitch contour is feed into feature extraction program to extract pitch feature. Here we use the 4 coefficients of 3-rd order polynomials. Finally the extracted feature is feed into the pre-trained NN classifier for classification and the accuracy is calculated correspondingly.

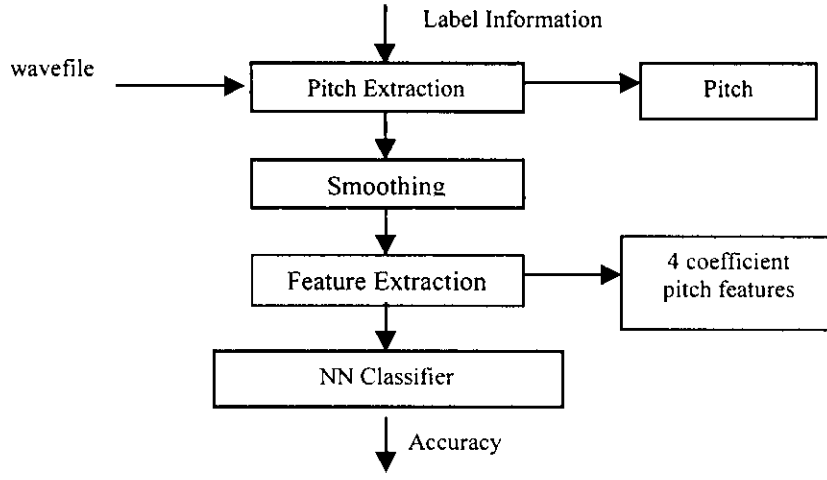


Figure 2-4 Classification Framework for Pitch Detection Algorithms

2.3 Experiments and Discussions

2.3.1 Experiments Setting

The experiments mainly include two parts. First part is emphasis on the observation of the results of these two pitch detection algorithms. And the pre-processing effects like the processing of low-pass-filter and center-clipping. The voiced/unvoiced determination in auto-correlation method is also tested. The speech

that we used in our experiments is from Thai continuous speech database. Here for observing the effects, we have done the above experiments for some speeches from the database. Considering almost all of them shows the similar results. Here we only use one continuous speech with information "07229" and one single Madarine speech "hao(3)" which is considered more difficult in pitch tracking because of its big variation. Second part is worked on a small database that is based on 4-continuous-Thai-digit sentence. The sentences are chosen according to the general distribution of 3 tones in Thai digit. It includes 14 sentences with 23 1st tone, 10 2nd tone and 23 4th tone. To consider this is only for testing, we record the sound in the office environment. Sampling frequency is 16K Hz. And we collect 4 male's sound and 2 rounds per person. Finally we get 112 speeches. All of the speech is hand labeled with the wave-surfer software. In our testing, we use the 1st round speech of each person as training set. And the following are as the testing set. We use our implementation to detect the pitch contour and extract the pitch feature. A classifier using 3-layer feed-forward neural network is implemented. The input layer includes four neurons corresponding to the four extracted tone feature. To represent five tones in Thai speech, the output layer consists of five separate units. The size of hidden layer is task dependent and is determined empirically. Different number of hidden neurons is tested, say 10, 15, 20, 25 etc. and it was found that 15 hidden neurons gave the best performance.

Figure 2-5 Waveform of Thai Digit("07229")

Figure 2-6 Waveform of Mandarin Speech "hao" with 3rd tone

2.3.2 Experiments Results and Discussions

To observe the difference between AMDF and Auto-correlation method, we test both of them through a Thai continuous digit "07229", which is shown in Figure 2-5. The pitch is shown in Figure 2-7. From the figure, the pitch information mainly lies on the voiced part in the speech signal. The silence part of the pitch is shown as the big variation. In the voiced part the pitch tracking show continuously and smoothly. Then the voiced/unvoiced decision is proved to be a very important part of pitch detection. Also although the pitch track shown in Figure 2-7 can describe the

trend of the pitch, it still exists some error points which need further processing, that we say, smoothing. Also in Figure 2-7, it shows both results for Auto-correlation method and AMDF. We can see that both methods can give us accepted result.

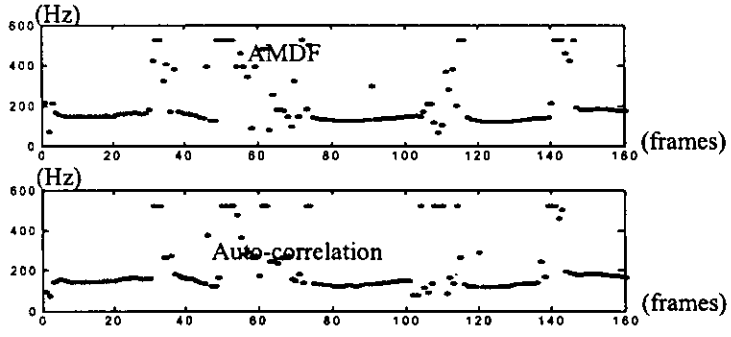


Figure 2-7 Pitch Track Using Auto-correlation Method and AMDF

2.3.2.1 Voice/Unvoiced Decision and Smoothing

In the implementation of auto-correlation method, we use 0.55 of the frame energy as the threshold to detect the voiced/unvoiced decision. Figure 2-8 shows the experiment's results of it. From Figure 2-8, it can detect the voiced part of the speech basically although some decision logic need to be further studied.

The smoothing of the pitch contour is necessary after a single pitch contour. The smoothing procedure is done on a segment-by-segment basis. The pitch mean of a segment is calculated first. Then the difference of pitch values in two continuous frames is examined. If it is greater than a predetermined threshold, the one lies farther away from the mean value is treated as a double, triple or half pitch error and corrected. The above process is done twice, forward and backward, for each segment in order to ensure the smoothness of pitch contour. After this, it is the median filter and a 3-point hanning window. Finally the linear interpolation is done for the very short pitch to extract the 3-order polynomials feature.

Here we chose a single speech word to be the object. Generally the pitch of 3rd tone in Mandarin is more difficult to classify than other tone because of its big variations. We chose the Mandarin word "hao" with 3rd tone in the experiment. The waveform is shown in Figure 2-6. The pitch using Auto-correlation method is smoothed using the smoothing technique described above.

2.3.2.2 Effects of Pre-processing

In order to observe the effects of low-pass-filter and center-clipping, we did some experiments on the speech "hao" which is the third tone in Chinese. For AMDF algorithm, we did not find significant effect of the pre-processing. But the pre-processing reduced the data and then increased the processing speed. But the effects

of LPF in auto-correlation method are quite clear and shown in Figure 2-10. In Figure 2-10, the error points reduced from 10 to 4 after adding the processing of LPF.

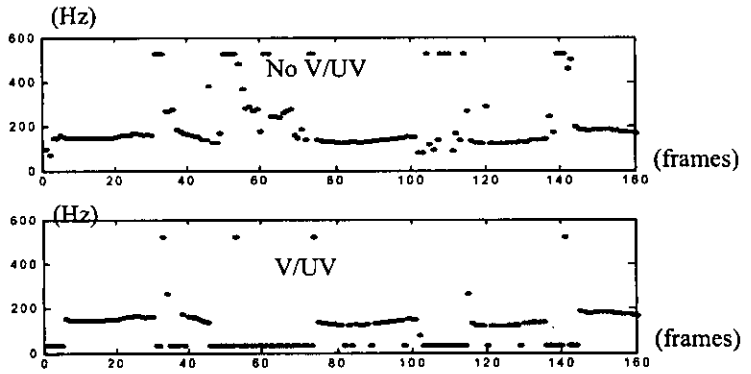


Figure 2-8 Voice/Unvoiced Detection in Autocorrelation Method

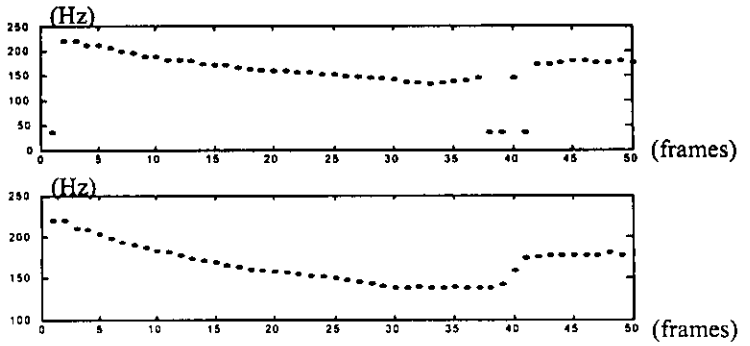


Figure 2-9 Smoothing of "hao" (3rd tone in Chinese) Pitch Contour

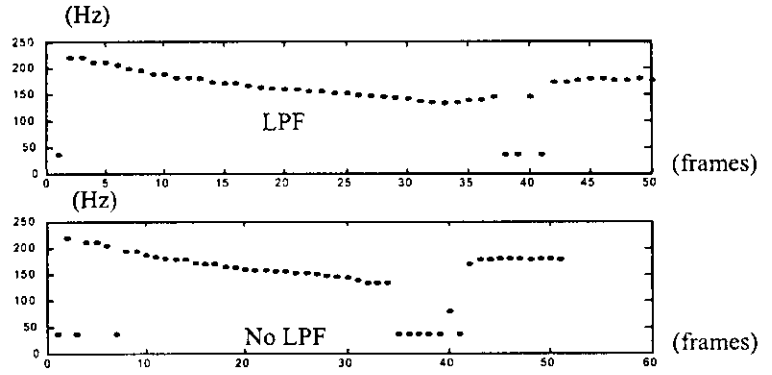


Figure 2-10 Effects of LPF in Auto-correlation method

2.3.2.3 Feature Extraction

Pitch information mainly lies on the trend of pitch contour. As introduced above, two methods, LMS and Orthogonol polynomial, are used to extract the pattern of pitch contour. The experiment is shown in Figure 2-11. From the figure, both of

them are working well. But finally which one can get better performance in recognition system needs the further research and experiments. Also Figure 2-12 shows the shape of the four discrete Legendre bases for the space of pitch contour length. From Figure 2-12, we can see that the four discrete Legendre bases can be used to express the basic pitch contour.

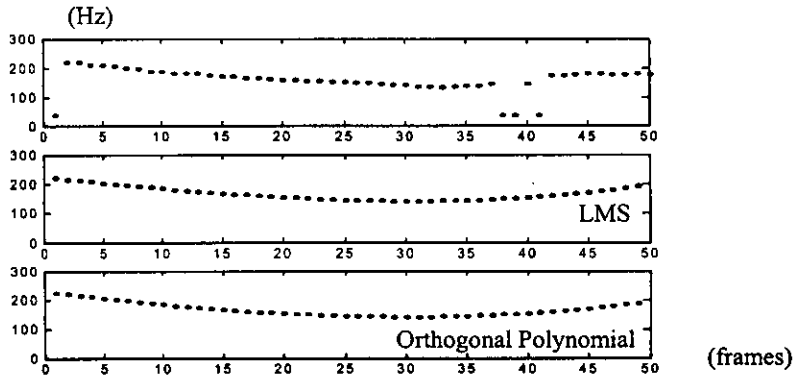


Figure 2-11 The pitch pattern extracted

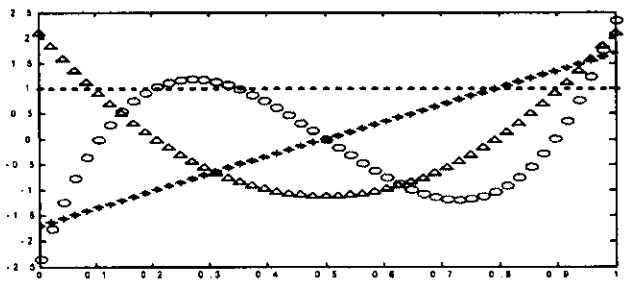


Figure 2-12 Four Discrete Legendre bases
(Point: base 1 *: base 2 Δ: base 3 o: base 4)

2.3.2.4 Classification

This is the second part of the experiments in this Chapter. A three-layer feed-forward neural network is used for classification. According to our observation, we use the auto-correlation pitch detection and orthogonal polynomial for our testing. All feature vectors are normalized to lie between −1.0 and 1.0 using the min-max normalization shown in equation 2-14.

$$normf_i = 2.0 \times \left(\frac{F_i - \min F_i}{\max F_i - \min F_i} \right) - 1.0, \qquad (2\text{-}14)$$

The total performance of the testing for Thai digit is about 79.02% (177 from 224). The confusion-matrix is shown in Table 2-1 from where the confusion between tone 1(low) and tone 4 (high) are found. Since only 3 tones exists for all Thai digit, there are only 3 tones shown in the confusion-matrix.

Table 2-1 Confusion-matrix of Tone Classification for Thai Digit

Tone	1	2	4	Percent(%)
1	69	4	19	75
2	5	33	2	82.5
4	15	2	75	81.52

The same experiments is also done on vowel aa speech in the continuous speech database (Thongprasert *et al.*, 2002) which include 18 speakers with 20 utterances each. The vowel aa speech lying in the first 15 utterances of each speaker are taken as training data. The rest is testing data. The confusion-matrix is shown in Table 2-2.

Table 2-2 Confusion-matrix of Tone Classification for Vowel aa Speech

Tone	0	1	2	3	4	Percent(%)
0	82	3	5	4	6	82
1	5	54	5	2	14	67.5
2	4	11	34	49	2	34
3	3	0	3	42	1	85.71
4	1	13	1	4	25	56.8

Here we use tone 0,1,2,3, 4 represent tone mid, low, falling, high, rising separately. The lost of accuracy mainly lies in the confusion between the 1st tone (low) and the 4th tone (rising), the 2nd tone (falling) and the 3rd tone (high). The reason for such results can be found from the 5-Thai-tone contour shown in Figure 2-13 and the effects of continuous interaction.

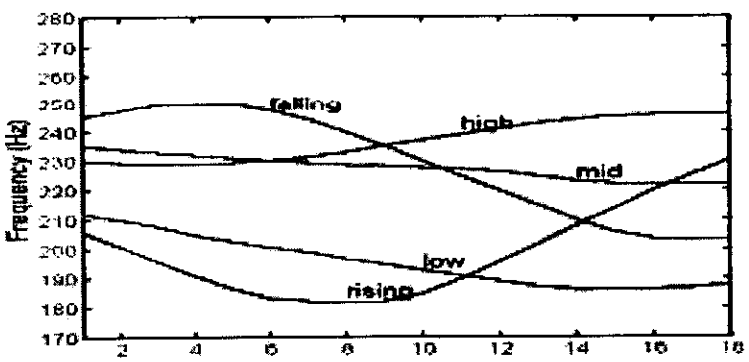


Figure 2-13 Average F0 contours of the five Thai tones produced in isolation

From the figure, we can see the initial level of tone 1 and tone 4, tone 2 and tone 3 is similar. Also because of the continuous interaction of speech, the trend of tone may not meet the final level for tone 4,2. Also here only 4 features are used in classification. So it's possible the accuracy will be improved if more features are added.

2.4 Summary

We described, in this chapter, the two pitch detection algorithms and the related techniques including preprocessing, post-processing and extraction of pitch pattern. According to our observing of the experiments. We found that both auto-correlation method and AMDF algorithm can provide the accepted results generally. Through the observing of preprocessing technique in both techniques, we didn't find the big effects of preprocessing on AMDF. But the obvious effects of low-pass-filter is shown in the experiment using auto-correlation method. At the same time, we have tested the smoothing using median-filter and voiced/unvoiced decision in auto-correlation method. Both of them showed the positive results. Finally, we used two methods to extract the pitch pattern through the smoothed pitch contour. According to the experiments figure, both of them works quite well. But in this case we need the smoothed pitch segment. For pitch detection the voice/unvoiced determination and the segmenting of pitch contour are another important issue that we did not discussed much here. The reason is that we are using the labeled speech data. The voiced/voiceless information is known from the label information. A simple classification testing has been done on our implementation. The results show the basic working of our implementation. The 79.02% accuracy is reached. Big confusion lies between tone 1 and tone 4. From the typical shape of pitch contour, we can find that the beginning part of these two tones is going to be the same trend that makes them easily to be confused. Through the work described here, we have implemented 2 pitch detection algorithms. Both of them can give the satisfied pitch contour based on our experiments. Based on extracted pitch, the big variation of pitch contour still existed. The further processing of pitch contour to reduce the variation of pitch contour and improve the tone classification performance is necessary. In next Chapter, the configuration of tone feature is discussed that include the related issue about tone-critical segment, scaling, normalization, tone feature setting. The implementations and the experiments are also described.